DE-41 (DME)

YEAR 2023

NUST COLLEGE OF

ELECTRICAL AND MECHANICAL ENGINEERING

Digital Twin for Condition Monitoring

A PROJECT REPORT

DE-41 (DME)

Submitted by

ASHHAD ALI SHAHID

TABREZ ASIF

DAWOOD HAIDER KHAN

M. YOUSUF EISA

BACHELORS

IN

MECHANICAL ENGINEERING

YEAR

2023

PROJECT SUPERVISOR AND CO-SUPERVISOR

DR. IMRAN AKHTAR, DR. ZAFAR ABBAS BANGASH

NUST COLLEGE Of
ELECTRICAL AND MECHANICAL ENGINEERING
PESHAWAR ROAD, RAWALPINDI

DECLARATION

We hereby declare that no portion of the work referred to in this Project Thesis has been submitted in support of an application for another degree or qualification of this of any other university or other institute of learning. If any act of plagiarism found, we are fully responsible for every disciplinary action taken against us depending upon the seriousness of the proven offence, even the cancellation of our degree.

COPYRIGHT STATEMENT

- Copyright in text of this thesis rests with the student author. Copies (by any process) either in full, or of extracts, may be made **only** in accordance with instructions given by the author and lodged in the Library of NUST College of E&ME. Details may be obtained by the Librarian. This page must form part of any such copies made. Further copies (by any process) of copies made in accordance with such instructions may not be made without the permission (in writing) of the author.
- The ownership of any intellectual property rights which may be described in this thesis is vested in NUST College of E&ME, subject to any prior agreement to the contrary, and may not be made available for use by third parties without the written permission of the College of E&ME, which will prescribe the terms and conditions of any such agreement.
- Further information on the conditions under which disclosures and exploitation may take place is available from the Library of NUST College of E&ME, Rawalpindi.

ACKNOWLEDGMENTS

We are truly grateful to Allah Almighty who gave us these eyes so we could use them to observer our surroundings, these ears to hear the sounds around us, these hands we used to complete our work and write this document and our brains and enough wisdom to solve an engineering problem and present a solution to it. It could not have been without the help of our family members, especially our parents, whose staunch and never-ending support helped us to take part in higher education and then go on to work on a project requiring hard work, patience, persistence, and intelligence. Lastly, we would like to show our gratitude and appreciation to our supervisor Dr. Imran Akhtar, and co-supervisor Dr. Zafar Abbas Bangash, whose continuous guidance and calm attitude towards our mistakes helped in achieving a goal we had dreamed about for a long time.

ABSTRACT

A digital twin is an enhanced digital representation of a real system. Digital twins can mimic the operation of physical systems and use data captured from their sensors to detect abnormal conditions and diagnose the cause of the problem. This report outlines the development of a digital twin for condition monitoring of a vibrating overhanging beam. The digital twin is designed to collect real-time data from sensors, such as vibration, and uses Microsoft Azure Function to process and analyse the data. The system is then linked with Ansys software to perform simulations and predict potential faults or failures in the beam.

The report discusses the development process, including sensor selection, data collection and analysis, and the integration of Ansys simulations with the digital twin. Additionally, the report highlights the benefits of using digital twin technology for condition monitoring, such as improved equipment performance and reduced maintenance costs.

The results of this project demonstrate the effectiveness of digital twin technology in providing early detection of potential faults and failures in industrial equipment. The report also discusses the potential for further research and development in this field to optimize the digital twin technology for condition monitoring of other industrial systems.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1: Introduction

1.1 Background

The objective of this project is to develop a digital twin for condition monitoring of a vibrating beam. The digital twin is a virtual model of the beam that can be used to monitor its performance and predict any issues before they occur. The use of digital twins has become increasingly popular in recent years, particularly in the field of industrial automation, due to their ability to improve operational efficiency, reduce downtime, and enhance safety.

The pump is a critical component of many industrial processes, and its failure can result in costly downtime, loss of production, and safety hazards. Therefore, monitoring the pump's condition is essential to ensure its reliable operation and prevent costly breakdowns. The digital twin for the pump will be developed using computer-aided design (CAD) software and simulation tools. The model will take into account the pump's geometry, material properties, and operating conditions to predict its performance and detect any deviations from the expected behaviour.

This report presents the methodology, implementation, and results of the digital twin for condition monitoring of a pump. It discusses the development of the simulation model, the integration of the model into a monitoring system, and the analysis of the data collected. The report concludes with a discussion of the implications of the project, including its potential applications in industry and areas for future research.

1.2 Problem Statement

The condition monitoring of vibrating structures is crucial in various industrial applications such as aerospace, civil engineering, and mechanical systems. In these applications, early detection and accurate assessment of structural defects or anomalies can significantly enhance safety, prevent catastrophic failures, reduce maintenance costs, and optimize operational efficiency. However, the conventional methods for monitoring and assessing the condition of vibrating beams often rely on manual inspections or limited sensor data, leading to suboptimal results and increased downtime.

The need of the hour is to develop a comprehensive and robust solution that enables real-time monitoring and analysis of vibrating beams using a digital twin. A digital twin is a virtual representation of a physical asset or system that can replicate its behaviour, predict its performance, and monitor its condition in a virtual environment. By utilizing data from sensors, the digital twin can provide a holistic view of the vibrating beam's condition, enabling proactive maintenance and timely interventions. The primary objective of this project is to design and implement a digital twin framework specifically tailored for condition monitoring of vibrating beams. The framework should integrate various components such as data acquisition, signal processing, anomaly detection algorithms, and visualization tools to enable effective monitoring and analysis. The digital twin should accurately simulate the behaviour of the vibrating beam under different operational conditions and provide actionable insights based on real-time sensor data.

The digital twin will leverage historical data, collected from sensors placed on the beam. Additionally, the framework should support remote access and provide a user-friendly interface for operators and maintenance personnel to visualize the condition of the vibrating beam and make informed decisions regarding maintenance and repair actions. By developing a digital twin for condition monitoring of vibrating beams, this project aims to enhance the reliability and safety of vibrating structures while optimizing maintenance strategies and minimizing operational downtime.

1.3 Beams

A beam is a structural element that primarily resists loads applied laterally to the beam's axis (an element designed to carry primarily axial load would be a strut or column). Its mode of deflection is primarily by bending. The loads applied to the beam result in reaction forces at the beam's support points. The total effect of all the forces acting on the beam is to produce shear forces and bending moments within the beams, that in turn induce internal stresses, strains and deflections of the beam. Beams are characterized by their manner of support, profile (shape of cross-section), equilibrium conditions, length, and their material.

1.4 Types of Beams

There are various types of structural beams. Their usage is dependent on the application and the structure where they are to be accommodated. The difference in their shapes, sizes, supports and dimensions are dictated by the desired structural needs. The most important types of beams include:

1.4.1 Simply Supported Beams

Simply supported beams are a fundamental structural element commonly used in civil engineering and construction. These beams are supported at each end with simple and fixed connections, allowing them to freely rotate but preventing any horizontal movement. The simplicity of their design makes them popular for various applications, ranging from small residential structures to large industrial projects. Simply supported beams are crucial for distributing loads and maintaining structural integrity. They efficiently transfer forces and moments from one end to the other, allowing them to bear the weight and provide stability. Due to their versatility, simplicity, and cost-effectiveness, simply supported beams remain a vital component in the construction industry.

Figure 1: A Simply Supported Beam

1.4.2 Cantilever Beams

Cantilever beams are unique and distinctive structural elements extensively used in engineering and architecture. Unlike simply supported beams, cantilevers are supported at only one end, while the other end remains free and unsupported. This design creates a visually striking effect and allows for a wide range of creative and practical applications. Cantilever beams are renowned for their ability to extend beyond their supporting structure, creating overhangs and providing functional advantages such as increased space or aesthetic appeal. They are often used in structures like balconies, bridges, and architectural features. The load applied to a cantilever beam creates a bending moment, which is absorbed by the beam's internal stress distribution. By carefully designing the dimensions and materials used, engineers can ensure the stability and structural integrity of cantilever beams. Their unique characteristics make them a fascinating and valuable component in modern construction and design.

Figure 2: A Cantilever Beam

1.4.3 Continuous Beams

Continuous beams are a type of structural element widely used in civil engineering and construction. Unlike simply supported beams, continuous beams have more than two supports along their length. These supports, often referred to as intermediate supports or columns, provide additional points of stability and prevent excessive deflection or rotation. Continuous beams are designed to distribute loads and moments evenly throughout their length, creating a continuous load path. This characteristic makes them suitable for spanning long distances without the need for additional supports. The continuous nature of these beams enhances their load-carrying capacity and structural efficiency. Continuous beams are commonly employed in various structures, including bridges, multi-story buildings, and large industrial complexes. Their ability to handle complex loading conditions and provide enhanced structural stability makes continuous beams a preferred choice in many construction projects.

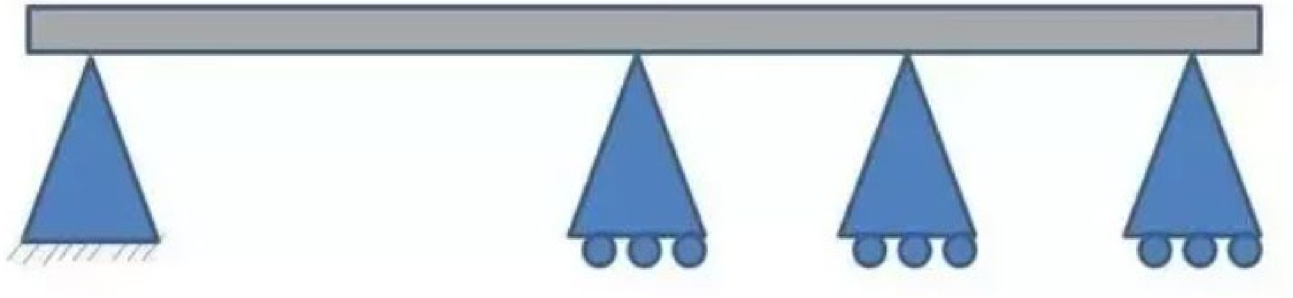


Figure 3: A Continuous Beam

1.4.4 Fixed Beams

A fixed beam is a structural element that is capable of supporting loads and resisting bending moments at its ends. It is designed to be rigidly connected to its supports, preventing any rotation or displacement. Fixed beams are commonly used in construction and engineering projects where stability and load-bearing capacity are essential. The fixed connections at the beam ends eliminate any rotational movement, resulting in high bending stiffness. This characteristic makes fixed beams suitable for applications where minimal deflection and precise load distribution are required. They are often utilized in bridges, buildings, and other structures where structural integrity is critical. Fixed beams play a crucial role in maintaining the overall stability and safety of a structure, making them an integral component in civil engineering and architectural designs.

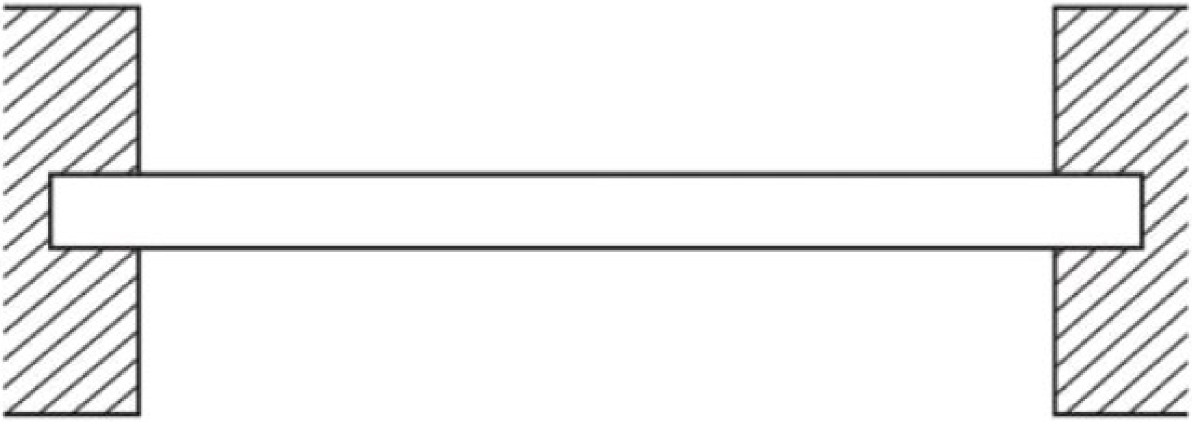


Figure 4: A Fixed Beam

1.4.5 Cantilever, Simply Supported Beams

A cantilever beam refers to a structural member that is supported at one end while the other end is left unsupported, projecting horizontally into space. In contrast, a simply supported beam is one that is supported at both ends, allowing for vertical movement and rotation. However, the term "cantilever simply supported beam" may seem contradictory at first. It could possibly describe a beam that is fixed or supported on one end and simply supported on the other, creating a hybrid configuration. This combination of support conditions can be seen in various engineering applications, where the beam's behaviour is influenced by both the cantilever and simply supported characteristics, resulting in unique load distribution and structural responses.

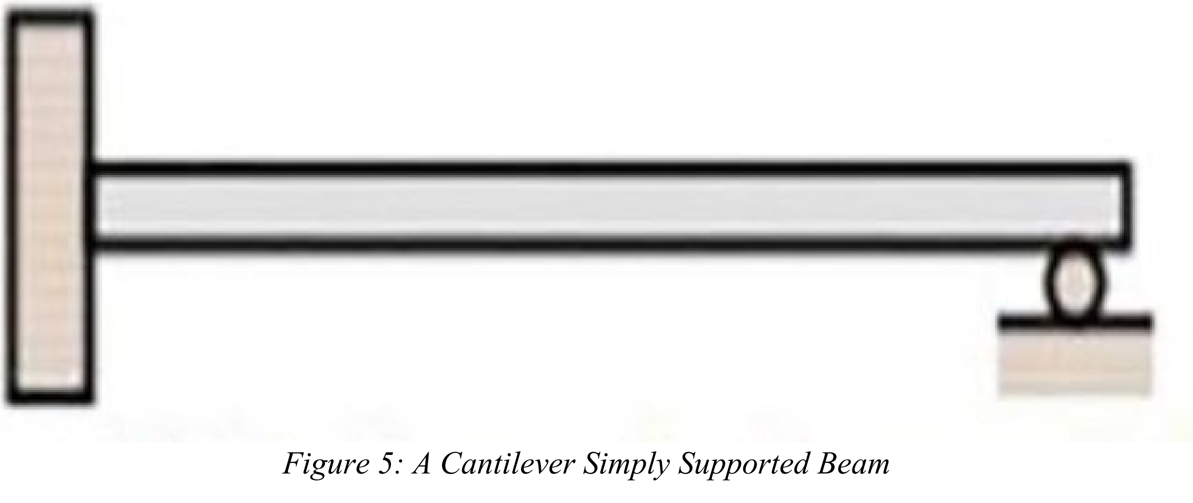


Figure 5: A Cantilever Simply Supported Beam

1.4.6 Overhanging Beam

Overhanging beams refer to structural elements that extend beyond their supports or the edges of a structure. These beams play a crucial role in construction and engineering, providing additional support, creating architectural aesthetics, and enabling the creation of larger and more versatile spaces. One of the primary functions of overhanging beams is to distribute the weight of the structure and any additional loads to the supporting columns or walls. By extending beyond the support points, they can transfer the load more effectively and reduce the stress on the primary structural members. Overhanging beams are commonly used in various architectural styles to create visually appealing designs. They can be found in both residential and commercial buildings, where they serve both functional and aesthetic purposes. The overhanging portion of the beam can provide shade or protection from the elements, such as extending over a window or entrance, offering shelter, and enhancing the building's overall functionality. In addition to their practical applications, overhanging beams can also be used to create striking architectural features. They can add depth, dimension, and visual interest to a structure, making it more visually appealing and memorable. Overhanging beams are often utilized in bridges, balconies, canopies, and pergolas to create unique designs and distinctive architectural elements. With advancements in engineering and construction techniques, overhanging beams can be engineered to withstand the forces and loads they experience, ensuring structural integrity and safety. Overall, overhanging beams are versatile structural components that combine functionality with aesthetics, allowing architects and engineers to create visually striking and efficient buildings while ensuring structural stability and load distribution.

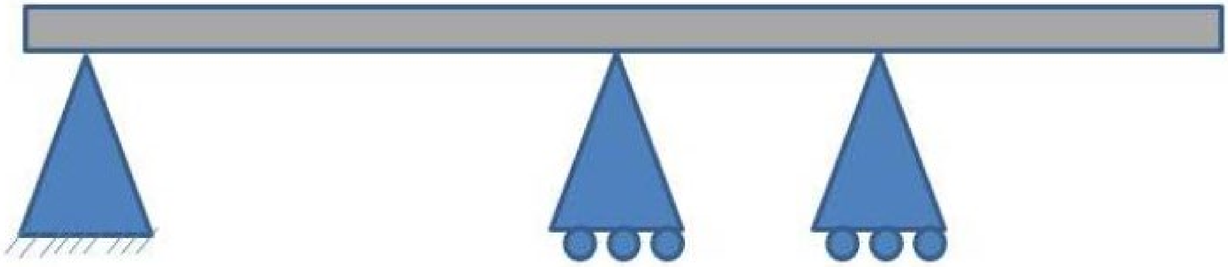


Figure 6: An Overhanging Beam

1.5 Objectives

I. Develop a digital twin model of a vibrating beam.
II. Implement Azure function for real-time data acquisition and monitoring.
III. Evaluate the effectiveness of the proposed approach for reducing downtime and improving efficiency in the context of the vibrating beam.

1.6 Sustainable Development Goals

The concept of the industrial digital twin has significant potential to contribute to the achievement of several Sustainable Development Goals (SDGs). One key SDG related to industrial digital twin technology is **Goal 9**: Industry, Innovation and Infrastructure. This SDG aims to promote inclusive and sustainable industrialization, enhance technological innovation, and improve infrastructure. Industrial digital twin technology can support the achievement of this goal by improving the efficiency of industrial processes and promoting innovation through the use of data analytics, simulation, and optimization.

Another SDG related to industrial digital twin technology is **Goal 12**: Responsible Consumption and Production. This SDG aims to ensure sustainable patterns of consumption and production by reducing waste, increasing resource efficiency, and promoting sustainable consumption. Industrial digital twin technology can contribute to this SDG by allowing for more efficient use of resources and reducing waste in industrial processes. The ability to simulate and optimize industrial processes using digital twins can help to identify areas of inefficiency and reduce resource consumption, leading to more sustainable production and consumption patterns.

Figure 7: SDG 9

Figure 8: SDG 12

Chapter 2: Development of Digital Twin

2.1 Background

The objective of this project is to develop a digital twin for condition monitoring of a vibrating beam. The digital twin is a virtual model of the beam that can be used to monitor its performance and predict any issues before they occur. The use of digital twins has become increasingly popular in recent years, particularly in the field of industrial automation, due to their ability to improve operational efficiency, reduce downtime, and enhance safety.

A beam is a critical component of many structures, and its failure can result in costly downtime, disasters, and safety hazards. Therefore, monitoring the beam's condition is essential to ensure its reliable operation and prevent costly breakdowns. The digital twin for the beam will be developed using Microsoft Azure. The model will take into account the beam's geometry, material properties, and operating conditions to predict its performance and detect any deviations from the expected behaviour. This report presents the methodology, implementation, and results of the digital twin for condition monitoring of a beam. It discusses the development of the simulation model, the integration of the model into a monitoring system, and the analysis of the data collected. The report concludes with a discussion of the implications of the project, including its potential applications in industry and areas for future research.

2.1.1 Literature Review

Digital twin technology has emerged as a promising approach to condition monitoring and predictive maintenance of industrial equipment. A digital twin is a virtual representation of a physical asset, which integrates data from various sources, including sensors, control systems, and historical data. By simulating the behaviour of the physical asset, a digital twin can provide real-time monitoring and analysis of the asset's performance, identify potential issues, and predict future behaviour.

2.1.1.1 Conceptual Framework of Digital Twins

The concept of digital twins originated from the fields of cyber-physical systems and the Internet of Things (IoT). It combines virtual models with real-time data from sensors and other sources to create a dynamic representation of the physical world. The digital twin acts as a bridge between the physical and digital realms, enabling real-time monitoring, analysis, and prediction.

2.1.1.2 A Specific Case - Pumps

The use of digital twins for condition monitoring of pumps has gained increasing attention in recent years. Pumps are critical components in many industrial processes, and their failure can lead to significant downtime and maintenance costs. Condition monitoring techniques, such as vibration analysis and temperature monitoring, have been used to detect potential issues in pumps. However, these techniques often require manual inspection and can only provide limited information on the pump's behaviour.

Several studies have investigated the use of digital twins for pump condition monitoring. For example, a digital twin was developed for a centrifugal pump, which integrated data from a variety of sources, including vibration sensors and motor current sensors. The digital twin was able to identify potential issues, such as cavitation and impeller wear, before they occurred, and provided real-time monitoring of the pump's performance.

In another study, a digital twin was developed for a reciprocating pump, which integrated data from pressure sensors and displacement sensors. The digital twin was able to predict the pump's behaviour under different operating conditions and identify potential issues, such as valve leakage and piston wear.

A review by discussed the potential benefits and challenges of using digital twins for pump condition monitoring. The review highlighted the ability of digital twins to provide real-time monitoring and analysis of the pump's performance, which can improve the pump's efficiency and reliability. However, the review also identified challenges, such as the need for accurate and reliable data, and the complexity of developing and integrating digital twins into the plant's control system.

Overall, the literature suggests that digital twin technology has significant potential for pump condition monitoring and predictive maintenance. The development of digital twins for pumps requires the integration of various data sources, the use of simulation and validation techniques, and the implementation of the digital twin into the plant's control system. Further research is needed to explore the potential benefits and limitations of digital twins for pump condition monitoring in different industrial contexts.

The development of digital twins for pumps has several advantages over traditional condition monitoring techniques. Digital twins can provide a more comprehensive and accurate picture of the pump's behaviour, allowing for early detection of potential issues and more effective maintenance strategies. Digital twins can also facilitate predictive maintenance, allowing for maintenance to be scheduled before a failure occurs, reducing downtime and maintenance costs.

The development of digital twins for pumps requires the integration of various data sources, including sensor data, historical data, and operational data. The data must be accurate and reliable, and the digital twin must be calibrated and validated to ensure accurate predictions. Several studies have investigated the use of machine learning algorithms to improve the accuracy and reliability of digital twins for pump condition monitoring.

One challenge in the development of digital twins for pumps is the complexity of the models and the need for accurate simulations. The models must be able to accurately simulate the behaviour of the physical asset under different operating conditions, including variations in flow rate, pressure, and temperature. The models must also be able to simulate the effects of different maintenance strategies, such as pump overhaul and component replacement.

Another challenge in the development of digital twins for pumps is the integration of the digital twin into the plant's control system. The digital twin must be able to communicate with the control system and provide real-time monitoring and analysis of the pump's performance. The digital twin must also be able to provide alerts and recommendations to the plant personnel when potential issues are detected.

Several studies have investigated the use of digital twins for specific types of pumps. For example, in a study, a digital twin was developed for a positive displacement pump, which integrated data from a variety of sources, including pressure sensors and displacement sensors. The digital twin was able to predict the pump's behaviour under different operating conditions and identify potential issues, such as valve leakage and piston wear.

In another study, a digital twin was developed for a submersible pump, which integrated data from vibration sensors and temperature sensors. The digital twin was able to identify potential issues, such as bearing wear and shaft misalignment, and provide recommendations for maintenance and repair.

The development of digital twins for pumps is an area of active research, with several challenges and opportunities for further investigation. One area of research is the integration of digital twins with other technologies, such as the Internet of Things (IoT) and artificial intelligence (AI). The integration of digital twins with IoT devices can provide more extensive and accurate data, while the integration of digital twins with AI algorithms can improve the accuracy and reliability of predictions.

Another area of research is the development of digital twins for pumps in different industrial contexts. The behaviour of pumps can vary significantly depending on the type of pump, the operating conditions, and the environment. The development of digital twins for pumps in different industrial contexts requires the adaptation of models and simulation techniques to the specific context, and the integration of data from different sources.

In summary, the literature suggests that digital twin technology has significant potential for pump condition monitoring and predictive maintenance. The development of digital twins for pumps requires the integration of various data sources, the use of simulation and validation techniques, and the integration of the digital twin into the plant's control system. Further research is needed to explore the potential benefits and limitations of digital twins for pump condition monitoring in different industrial contexts and the integration of digital twins with other technologies.

Literature Review: Digital Twins

2.1.1.3 Digital Twins in Manufacturing

One of the prominent domains where digital twins have gained traction is manufacturing. Digital twins of production systems or individual components allow for virtual testing, optimization, and predictive maintenance. By simulating the behaviour of manufacturing processes, organizations can enhance efficiency, reduce downtime, and improve product quality.

Figure 9: A Pictorial Representation of Use of Digital Twins in Manufacturing

2.1.1.4 Digital Twins in Healthcare

In the healthcare sector, digital twins have the potential to revolutionize patient care and medical research. Personalized digital twins can be created to capture an individual's physiological parameters, genetic makeup, and lifestyle data. These virtual replicas enable accurate diagnosis, personalized treatment plans, and the exploration of various medical scenarios.

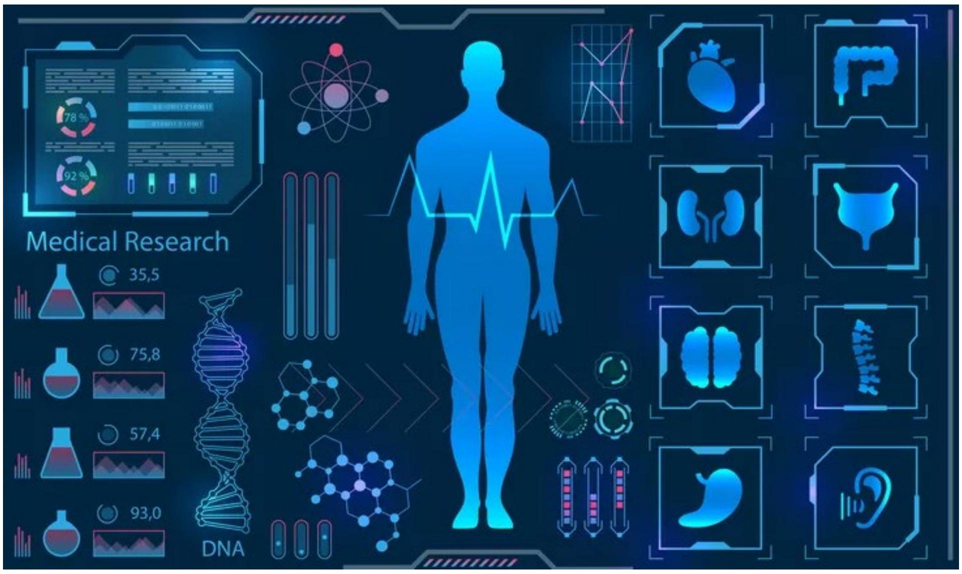


Figure 10: A Pictorial Representation of Use of Digital Twins Healthcare Sector

2.1.1.5 Digital Twins in Smart Cities and Infrastructure

Digital twins play a crucial role in the development and management of smart cities and infrastructure. By creating virtual replicas of physical assets, such as buildings, bridges, and urban systems, city planners and engineers can simulate scenarios, monitor structural health, and optimize resource allocation. Digital twins offer a range of benefits, including predicting and mitigating potential risks, improving urban sustainability, and enhancing the overall quality of life.

In the context of smart cities and infrastructure, digital twins enable city planners and engineers to better understand, manage, and optimize urban systems.

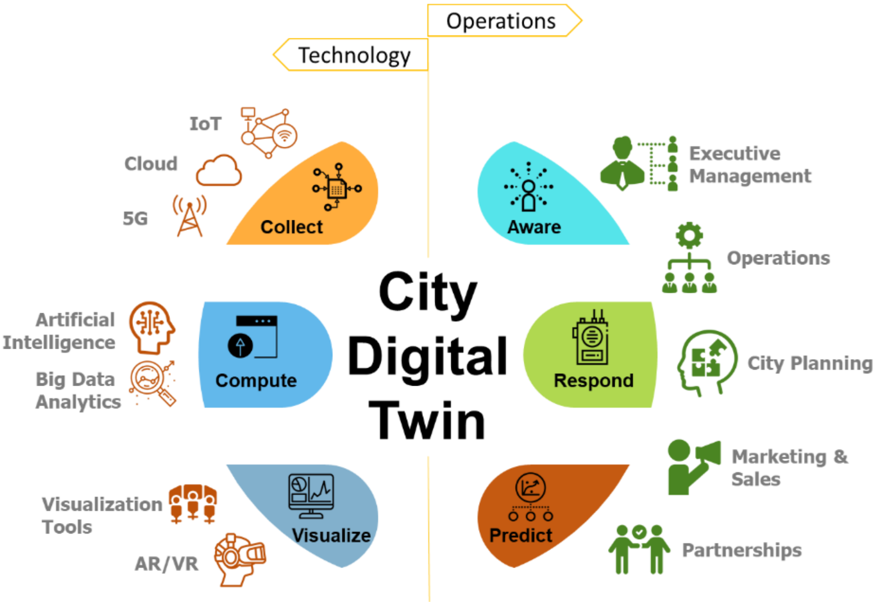


Figure 11: A Pictorial Representation of Use of Digital Twins in City Planning

Here are some key applications and benefits of digital twins in this domain:

I. **Simulation and Scenario Planning:** Digital twins allow city planners to simulate and analyse different scenarios before implementing changes or making decisions. For example, they can assess the impact of new construction projects, changes in traffic patterns, or modifications to utility systems. By simulating these scenarios, planners can evaluate the potential outcomes, optimize resource allocation, and ensure the most efficient use of urban space.

II. **Structural Health Monitoring:** Digital twins enable real-time monitoring of the structural health of buildings, bridges, and other infrastructure assets. Sensors embedded within physical structures collect data on parameters such as strain, temperature, vibration, and corrosion. This data is fed into the digital twin, allowing engineers to monitor the health of the asset, identify potential issues, and schedule maintenance activities proactively. By detecting problems early, digital twins can help prevent costly repairs and ensure the safety and longevity of infrastructure.

III. **Risk Assessment and Mitigation:** Digital twins facilitate risk assessment and mitigation strategies for urban systems. By integrating data from multiple sources, such as weather forecasts, sensor readings, and historical records, digital twins can predict potential risks and simulate their impact on the city. For instance, in flood-prone areas, digital twins can simulate various flood scenarios and evaluate the effectiveness of different mitigation measures. This helps city planners develop more robust disaster management plans, enhance resilience, and reduce the impact of natural disasters.

IV. **Resource Optimization:** Digital twins enable optimization of resource allocation in urban systems. By analysing real-time data from sensors and IoT devices, digital twins can identify inefficiencies and suggest improvements. For example, in smart grids, digital twins can monitor electricity usage patterns, identify peak demand periods, and optimize the distribution of electricity resources. Similarly, in water management systems, digital twins can analyse consumption patterns and identify areas of high-water usage, enabling better allocation of water resources.

V. **Urban Sustainability:** Digital twins contribute to urban sustainability efforts by providing insights into energy consumption, carbon emissions, and environmental impact. By integrating data on energy usage, transportation patterns, waste management, and air quality, digital twins can assess the sustainability of cities and identify areas for improvement. City planners can use this information to implement more sustainable practices, reduce carbon footprints, and enhance the overall quality of the urban environment.

VI. **Citizen Engagement and Participation:** Digital twins can also foster citizen engagement and participation in urban planning and decision-making processes. By visualizing data and simulating different scenarios, digital twins make complex information more accessible to the public. Citizens can provide feedback, input, and suggestions based on their interactions with the digital twin. This promotes transparency, inclusivity, and collaboration between city authorities and residents, leading to better-informed decisions and a sense of ownership among the community.

2.1.1.6 Advancements in Digital Twin Technologies

Recent advancements in technologies such as artificial intelligence, machine learning, and big data analytics have significantly enhanced the capabilities of digital twins. These technologies enable real-time data integration, complex simulations, and advanced analytics for accurate predictions and decision-making. Furthermore, the integration of virtual reality and augmented reality technologies provides immersive visualization and interaction with digital twin models.

2.1.1.7 Challenges and Future Directions

While digital twins hold great promise, several challenges need to be addressed for their widespread adoption. Data security, privacy concerns, interoperability, and scalability are among the key challenges. Additionally, the development of accurate and reliable models, data fusion techniques, and effective human-machine interfaces are areas that require further research. Future directions include the integration of digital twins across multiple domains, the development of standards and frameworks, and the exploration of ethical and legal implications.

2.2 Challenges in Developing Digital Twins for Beams

The development of digital twins for beams poses several challenges related to data acquisition, modelling and simulation, and integration with the plant's control system.

Data acquisition is a critical aspect of digital twin development, as the accuracy and reliability of the digital twin depend on the quality and quantity of data used. The acquisition of data from beams can be challenging due to several factors, such as the type of sensors used, and the working condition of the bema. Some beams may not have sensors installed, or the sensors may not provide sufficient data for modelling and simulation. In such cases, additional sensors may need to be installed, or alternative data sources may need to be used, such as historical maintenance records or operator feedback.

Modelling and simulation are also critical aspects of digital twin development, as the accuracy and reliability of the digital twin depend on the fidelity of the models used. The development of accurate models requires a deep understanding of the physics and mechanics of pumps, as well as the operating conditions and environmental factors that affect beam behaviour. The use of simulation techniques, such as finite element analysis (FEA), can improve the accuracy of the models and enable the prediction of the beam's behaviour under different operating conditions. However, the development of accurate models can be time-consuming and require significant expertise.

The integration of the digital twin with the actual physical structure is also a critical aspect of digital twin development, as the digital twin must be able to communicate with the control system and provide real-time monitoring and analysis of the beam's performance. The integration of the digital twin with the control system requires the development of appropriate software interfaces and protocols, as well as the integration of data from different sources. In addition, the integration of the digital twin with the control system may require modifications to the control system architecture, which can be challenging.

2.3 Opportunities for Developing Digital Twins

Despite the challenges posed by digital twin development for beams, there are several opportunities for further investigation and development. These opportunities include the integration of digital twins with other technologies, the development of digital twins and the use of digital twins for predictive maintenance.

The integration of digital twins with other technologies, such as IoT and AI, can provide significant benefits for pump condition monitoring and predictive maintenance. The integration of digital twins with IoT devices can provide more extensive and accurate data, while the integration of digital twins with AI algorithms can improve the accuracy and reliability of predictions. For example, the integration of digital twins with AI algorithms can enable the prediction of the remaining useful life of beams before fatigue takes place, which can support more efficient maintenance scheduling and reduce disasters.

The development of digital twins for pumps in different industrial contexts is another opportunity for further investigation. The behaviour of pumps can vary significantly depending on the type of pump, the operating conditions, and the environment. The development of digital twins for pumps in different industrial contexts requires the adaptation of models and simulation techniques to the specific context, and the integration of data from different sources.

Finally, the use of digital twins for predictive maintenance is an opportunity for further investigation. Predictive maintenance involves the use of data analysis and modelling to predict the occurrence of maintenance issues before they occur, enabling more efficient maintenance scheduling and reduced downtime. The use of digital twins for predictive maintenance can enable the prediction of potential issues based on real-time monitoring and analysis of pump performance, enabling more efficient maintenance scheduling and reduced downtime.

2.4 Methodology

The methodology for developing a digital twin for condition monitoring involves several steps, including data acquisition, model development, simulation, and validation. The following sections describe each step-in detail.

2.4.1 Data Acquisition

The first step in developing a digital twin for condition monitoring is data acquisition. This involves gathering data from the beam and its surrounding environment, such as vibration behaviour, stress values, strain measurement, and other relevant parameters. The data can be collected using various sensors, such as vibration sensors, strain gauges, and other relevant sensors.

The data can be collected manually or automatically, depending on the type of sensor and the location of the beam. For example, sensors located in the surrounding environment may require manual data collection. In addition, historical maintenance records and operator feedback can be used to supplement sensor data and provide additional insights into beam’s behaviour.

The collected sensor data needs to be pre-processed before being sent to the digital twin model. The pre-processing step involves filtering, scaling, and normalization of the sensor data. Filtering is done to remove any noise or outliers in the data. Scaling and normalization are done to ensure that all sensor data is within the same range and to avoid any bias in the model.

It is essential to ensure that the data collected is accurate, reliable, and representative of the beam’s behaviour under different operating conditions. The data collected should cover a wide range of operating conditions and environmental factors, such as variations in vibration frequencies, multitude of load values and different strain conditions. This will enable the development of accurate models and simulations that can accurately predict beam’s behaviour under different operating conditions.

2.4.2 Model Development

The second step in developing a digital twin for beam condition monitoring is model development. This involves developing a mathematical model of the beam's behaviour based on the data collected in the previous step. The model should be able to predict the beam's behaviour under different operating conditions and environmental factors accurately.

The model can be developed using various techniques, such as empirical modelling, data-driven modelling, or physics-based modelling. Empirical modelling involves developing a model based on experimental data, while data-driven modelling involves developing a model based on statistical analysis of the data. Physics-based modelling involves developing a model based on the underlying physical principles governing beam behaviours, such as structural dynamics.

The choice of modelling technique depends on the available data and the desired level of accuracy. In general, physics-based modelling provides the most accurate results but requires significant expertise and computational resources.

2.4.3 Azure Function

Azure Functions is a serverless compute service that enables developers to run code on demand in the cloud. Azure Functions allows developers to create event-driven functions that respond to events such as changes to data in a database, a message arriving in a queue, or a file being uploaded to storage. In this project, an Azure Function will be used to collect and pre-process the sensor data from the pump.

The basic process of creating a digital twin on Azure involves the following steps:

1. **Connecting to the physical system:** Using IoT sensors to connect the physical system to Azure IoT Hub to collect data about the system.
2. Modelling the physical system: Using Azure Digital Twins to model the physical system using a set of metadata that describes the system's behaviours, properties, and relationships.
3. **Storing and analysing data:** Using Azure Time Series Insights to store and analyse the data collected from the physical system in real-time. This allows the user to track the system's behaviours and detect anomalies or potential issues.
4. **Enabling remote monitoring and control:** Using Azure Stream Analytics and Azure Event Grid to enable remote monitoring and control of the physical system. This allows the user to view real-time data, receive alerts, and trigger actions based on the data collected.

2.4.4 Integration with ANSYS

The next step in creating the digital twin for condition monitoring is to integrate the digital twin model with ANSYS. ANSYS is a software platform used for engineering simulation and modelling. The digital twin model will be integrated with ANSYS to perform simulations and predictions of the pump's behaviours under different operating conditions.

2.4.5 ANSYS Twin Builder

ANSYS Twin Builder is a software tool used to develop and deploy digital twins. It provides a graphical user interface that allows users to create, simulate, and deploy digital twin models. ANSYS Twin Builder supports a wide range of physical systems, including mechanical, electrical, fluid, and thermal systems. In this project, ANSYS Twin Builder will be used to develop a digital twin model of the pump.

Once the digital twin model is integrated with ANSYS, simulations can be performed to predict the pump's behaviours under different conditions. The simulations can be used to detect any anomalies or faults in the pump's operation and provide recommendations for maintenance or repair.

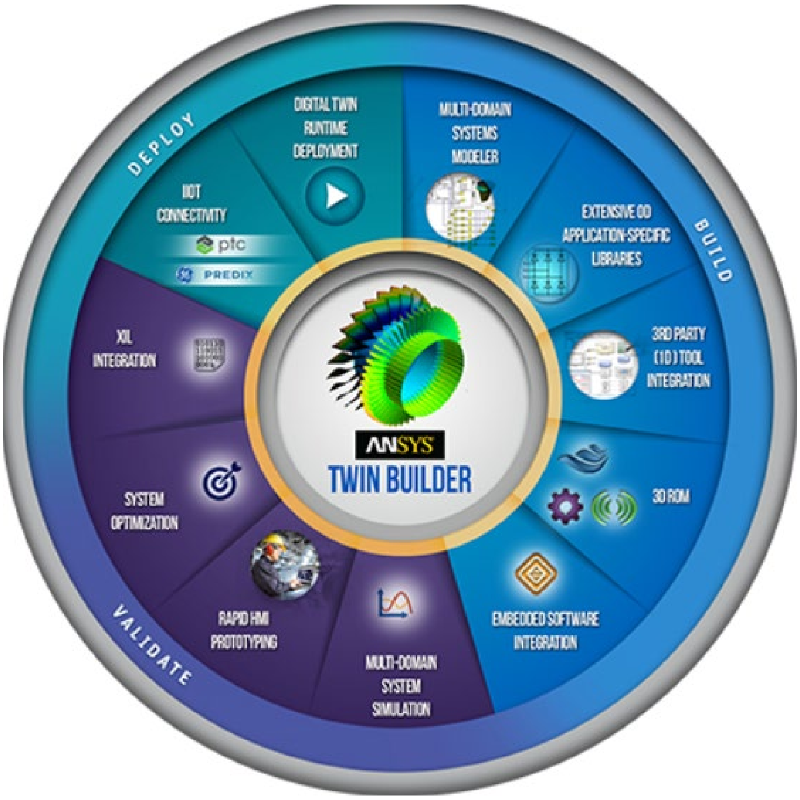


Figure 12: Ansys Twin Builder Breakdown

2.4.6 Simulations

The third step in developing a digital twin for condition monitoring is simulation. This involves simulating the behaviours of the beams using the model developed in the previous step. The simulation should be able to predict the beam's behaviours under different operating conditions accurately.

The simulation can be performed using various techniques, such as finite element analysis (FEA), or system dynamics modelling. System dynamics modelling involves simulating the interaction between different components of the beam set-up and its surrounding environment.

The choice of simulation technique depends on the available data and the desired level of accuracy. In general, FEA provides the most accurate results but require significant computational resources.

2.4.7 Validation

The fourth step in developing a digital twin for condition monitoring is validation. This involves comparing the simulation results with experimental data to ensure that the model and simulation accurately represent the pump's behaviours under different operating conditions.

The validation can be performed using various techniques, such as statistical analysis, error analysis, or sensitivity analysis. Statistical analysis involves comparing the simulation results with experimental data using statistical measures such as mean absolute error and root mean square error. Error analysis involves quantifying the error between the simulation results and experimental data. Sensitivity analysis involves quantifying the effect of different input parameters on the simulation results.

The validation process is essential to ensure the accuracy and reliability of the digital twin. Any discrepancies between the simulation results and experimental data should be addressed by refining the model or simulation technique.

Chapter 3: An Introduction to IoT

3.1 Introduction

The Internet of Things (IoT) has emerged as a transformative technology that connects physical objects and devices to the internet, enabling data exchange, communication, and automation. This literature review aims to provide an overview of the key concepts, applications, challenges, and advancements related to IoT.

3.2 Conceptual Framework of IoT

The Internet of Things (IoT) is a transformative concept that encompasses the interconnection of physical devices, sensors, and objects through a network infrastructure. This interconnected network enables the seamless exchange of data between devices, leading to real-time monitoring, control, and automation across various industries and domains. The IoT ecosystem comprises four fundamental components: physical devices, connectivity, data processing, and applications.

At the core of the IoT ecosystem are the physical devices themselves. These devices can range from simple sensors and actuators to complex machinery, appliances, and wearables. These devices are equipped with sensors and embedded systems that enable them to gather data from their surroundings, such as temperature, humidity, pressure, motion, and more. Additionally, they may have actuators that allow them to interact with the physical world by performing actions based on the data they receive.

Connectivity forms the vital infrastructure that enables communication and data exchange between IoT devices. Various connectivity technologies are utilized in IoT systems, including Wi-Fi, Bluetooth, cellular networks (such as 3G, 4G, and 5G), satellite connections, and low-power wide-area networks (LPWANs) like LoRaWAN and NB-IoT. These technologies provide different ranges, data rates, power requirements, and coverage areas, allowing IoT devices to be deployed in diverse environments.

The data collected by IoT devices needs to be processed and analysed to extract valuable insights and make informed decisions. This is where the data processing component of the IoT ecosystem comes into play. The massive amounts of data generated by IoT devices are transmitted to cloud platforms or edge computing systems for processing. Cloud platforms offer scalability, storage, and computational power, making them suitable for managing large volumes of data. Edge computing, on the other hand, involves processing data closer to the source, reducing latency and enabling real-time analysis and response. Data processing techniques, such as machine learning algorithms, artificial intelligence, and data analytics, are applied to derive meaningful information from the collected data.

Finally, the applications component of the IoT ecosystem utilizes the processed data to create value-added services and enable automation. These applications can be diverse and span across various domains, including smart homes, industrial automation, healthcare, agriculture, transportation, and more. For instance, in a smart home scenario, IoT devices like smart thermostats, lighting systems, and security cameras can be interconnected to enable energy efficiency, remote monitoring, and enhanced security. In industrial settings, IoT applications can optimize production processes, monitor equipment health, and enable predictive maintenance. In healthcare, IoT devices can track patient vitals, provide remote monitoring, and enable timely intervention. The possibilities are vast and continue to expand as technology advances.

The IoT ecosystem brings numerous benefits and opportunities. Real-time monitoring and control of devices enable proactive maintenance, improved efficiency, and cost savings. Automation and optimization of processes enhance productivity and accuracy. The ability to gather vast amounts of data from interconnected devices enables data-driven decision-making and predictive analytics, leading to improved outcomes and insights. Furthermore, the IoT ecosystem promotes connectivity and collaboration, enabling integration with other technologies and systems, such as cloud computing, big data analytics, and AI.

However, the IoT ecosystem also presents challenges and considerations. Security and privacy are critical concerns, as the vast amount of data being transmitted and processed can be vulnerable to unauthorized access or malicious attacks. Robust security measures, encryption, and authentication protocols are necessary to protect IoT devices and the data they generate. Additionally, interoperability and standardization are essential to ensure seamless integration and communication between different IoT devices and platforms. Moreover, the ethical implications of data collection, storage, and usage need to be carefully addressed to maintain trust and transparency.

3.3 IoT Applications

The applications of IoT (Internet of Things) are incredibly diverse and have the potential to revolutionize multiple domains. Here are some key applications of IoT in various sectors:

I. **Healthcare:** IoT plays a crucial role in the healthcare sector, enabling remote patient monitoring, wearable devices, and smart healthcare systems. IoT devices such as wearable health trackers can continuously monitor vital signs like heart rate, blood pressure, and sleep patterns, providing valuable data for healthcare professionals to remotely monitor patients and detect any anomalies. IoT-enabled smart pill dispensers can also help patients adhere to their medication schedules by providing reminders and alerts.

Figure 13: IoT Usage in Healthcare

II. **Agriculture:** IoT technology is transforming the agriculture industry through applications like precision farming, smart irrigation, and livestock monitoring. IoT sensors placed in fields can collect data on soil moisture, temperature, and nutrient levels, allowing farmers to optimize irrigation and fertilization processes, leading to improved crop yields and resource efficiency. Livestock monitoring devices equipped with IoT sensors can track animal health, location, and behaviour, helping farmers ensure the well-being of their livestock and detect any signs of disease or distress.

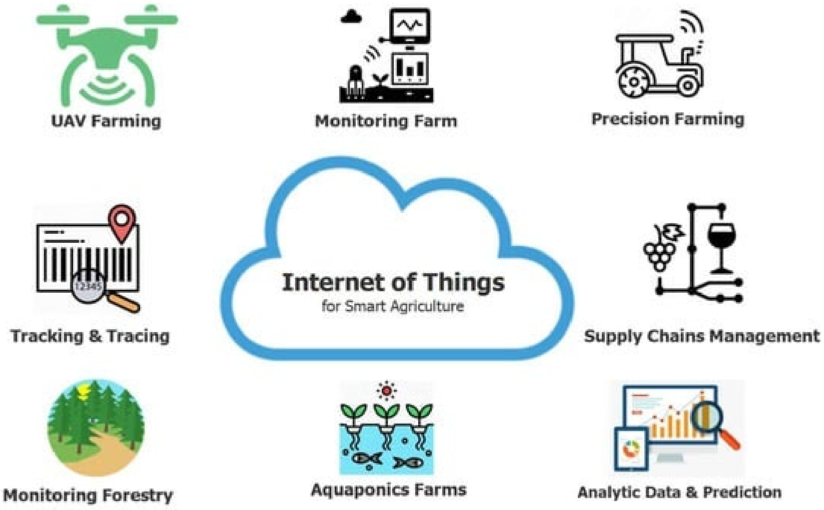


Figure 14: IoT Usage in Agriculture

III. **Smart Cities:** IoT is a key enabler in the development of smart cities. IoT devices and sensors deployed across cities can provide valuable data for optimizing various services and improving the quality of life for residents. For example, IoT-based traffic management systems can gather real-time data on traffic patterns, congestion, and accidents, enabling authorities to optimize traffic flow and reduce congestion. Smart waste management systems equipped with IoT sensors can monitor waste levels in bins, optimizing collection routes and reducing costs. Additionally, IoT-enabled energy management systems can help optimize energy consumption in buildings, reducing energy waste and promoting sustainability.

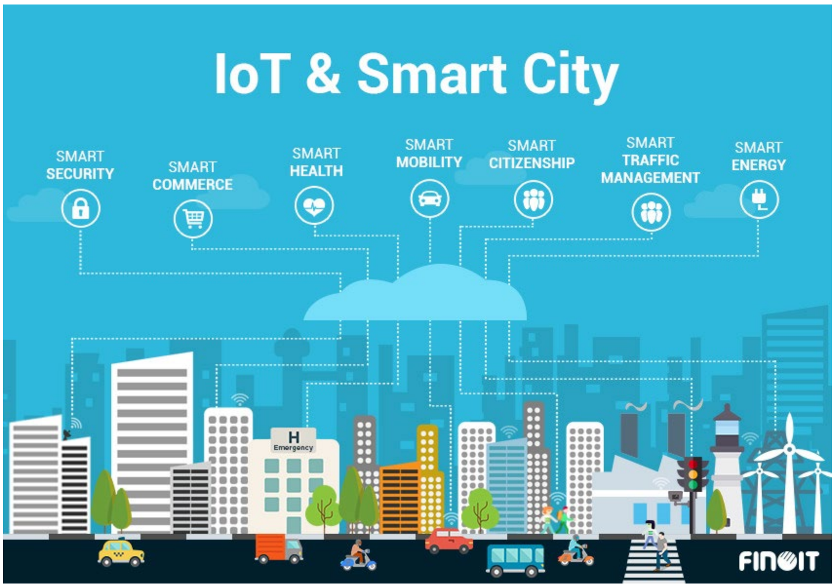


Figure 15: IoT Usage in Smart Cities

IV. **Industrial IoT (IIoT):** IIoT is the application of IoT technology in industrial settings, bringing automation, efficiency, and predictive maintenance to manufacturing processes. IoT devices and sensors embedded in machinery and equipment can monitor performance metrics, collect data on operating conditions, and detect signs of potential failures. This data enables predictive maintenance, where maintenance activities are scheduled based on actual equipment condition rather than a fixed schedule, minimizing downtime and optimizing maintenance costs. IIoT also facilitates asset tracking and inventory management, improving supply chain efficiency.

[1]

Figure 16: IoT Usage in IIoT

V. **Environmental Monitoring:** IoT plays a crucial role in environmental monitoring, allowing us to collect real-time data on air quality, water quality, and weather conditions. IoT sensors placed in cities, industrial areas, or natural environments can monitor pollutants, temperature, humidity, and other environmental parameters. This data enables authorities and researchers to assess environmental conditions, detect pollution sources, and take appropriate actions for sustainable environmental management [2].

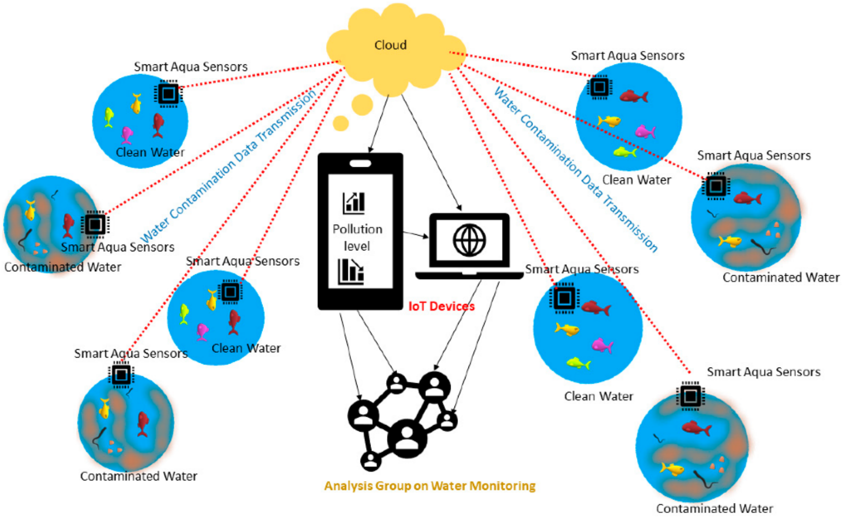


Figure 17: IoT Usage for Environmental Monitoring

VI. **Smart Homes:** IoT technology is transforming homes into smart, interconnected environments. IoT devices such as smart thermostats, lighting systems, and security cameras can be controlled and monitored remotely through mobile applications. This enables homeowners to optimize energy consumption, enhance security, and automate various tasks for convenience and comfort [3].

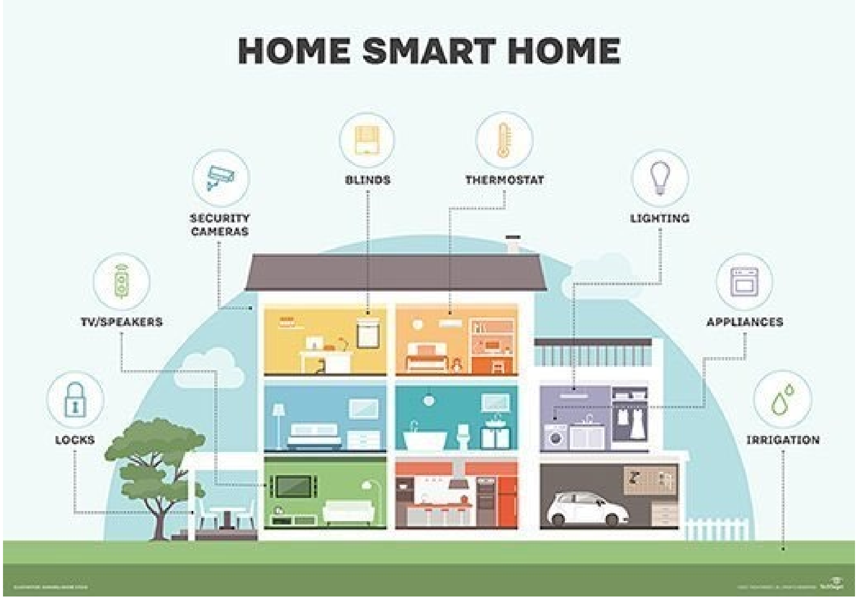


Figure 18: IoT Usage in Smart Homes

These are just a few examples of the wide range of applications of IoT technology across different sectors. As IoT continues to evolve and advance, we can expect to see even more innovative use cases and transformative applications that have the potential to reshape industries and improve our daily lives.

3.4 IoT Technologies and Protocols

The Internet of Things (IoT) has emerged as a transformative force, revolutionizing the way we interact with the world around us. By connecting everyday objects to the internet and enabling them to communicate and share data, IoT technologies have paved the way for innovative applications across various industries. Central to the success of IoT are the technologies and protocols that facilitate seamless connectivity, communication, and interoperability between devices. In this article, we will explore some of the key IoT technologies and protocols that power the IoT ecosystem.

I. **Wireless Connectivity:** Wireless connectivity is a fundamental component of IoT, enabling devices to communicate without the constraints of physical cables. Several wireless technologies are used in IoT deployments, including Wi-Fi, Bluetooth, Zigbee, Z-Wave, and Cellular (3G, 4G, and 5G). Wi-Fi is widely used for high-bandwidth applications and is ideal for devices operating within a local network. Bluetooth is commonly found in personal devices and short-range applications, while Zigbee and Z-Wave are popular for low-power and low-data-rate applications, such as home automation. Cellular connectivity, on the other hand, offers wide coverage and enables IoT devices to connect globally.

II. **IoT Protocols:** IoT devices use various protocols to establish communication and exchange data efficiently. Some of the widely used IoT protocols include MQTT (Message Queuing Telemetry Transport), CoAP (Constrained Application Protocol), HTTP (Hypertext Transfer Protocol), and AMQP (Advanced Message Queuing Protocol). MQTT and CoAP are lightweight protocols designed for constrained environments with low bandwidth and high latency. They are ideal for resource-constrained devices and enable efficient communication over unreliable networks. HTTP, a familiar protocol used in web applications, is also used in IoT for data exchange between devices and cloud platforms. AMQP is a robust protocol used for reliable messaging and queuing in IoT applications.

III. **Sensor Technologies:** Sensors play a crucial role in collecting data from the physical environment and enabling IoT devices to monitor and respond to changes. Various sensor technologies are employed in IoT applications, including temperature sensors, humidity sensors, motion sensors, pressure sensors, and light sensors, among others. These sensors detect and measure physical parameters and convert them into electrical signals that can be processed and transmitted by IoT devices. Sensor data provides valuable insights for applications like environmental monitoring, smart homes, healthcare, agriculture, and industrial automation.

IV. **Edge Computing:** The massive amount of data generated by IoT devices poses significant challenges in terms of storage, processing, and latency. Edge computing addresses these challenges by bringing computational power closer to the devices at the network edge. By performing data processing and analysis locally on edge devices, edge computing reduces the need for data transmission to the cloud, minimizing latency and bandwidth requirements. Edge computing also enhances data privacy and security by keeping sensitive information within the local network. It enables real-time decision-making, improved response times, and efficient resource utilization.

V. **Security and Privacy:** With the proliferation of connected devices and the exchange of sensitive data, security and privacy are of paramount importance in IoT deployments. IoT devices are vulnerable to cyber threats, and a compromised device can have severe consequences. To ensure secure communication and protect data integrity, various security measures are implemented, including authentication, encryption, access control, and secure bootstrapping. Additionally, privacy-preserving techniques, such as data anonymization and differential privacy, are employed to protect the identity and personal information of users.

VI. **Standardization:** Standardization is critical to ensure interoperability and seamless integration of IoT devices and systems from different manufacturers. Several organizations and consortiums are actively involved in defining standards for IoT technologies and protocols. For example, the Open Connectivity Foundation (OCF) and the Zigbee Alliance focus on standardizing interoperability for smart homes.

3.5 Data Analytics and Artificial Intelligence in IoT

The massive amount of data generated by IoT devices presents both opportunities and challenges. However, advanced data analytics techniques, such as machine learning and artificial intelligence (AI), enable organizations to leverage this data and extract valuable insights, leading to real-time insights, predictive analytics, and anomaly detection.

Machine learning algorithms, a subset of AI, are designed to analyse and identify patterns in large datasets. They can be applied to IoT data to uncover hidden insights and make predictions. By training these algorithms on historical IoT data, they can learn to recognize patterns and trends, allowing for real-time insights and predictive analytics. For example, in the manufacturing industry, machine learning algorithms can analyse sensor data from machinery to identify patterns indicating potential failures. By detecting anomalies in real-time, predictive maintenance can be performed, reducing costly downtime, and optimizing maintenance efforts.

Moreover, AI-powered data analytics enables organizations to gain a deeper understanding of their operations and customer behaviour. By processing and analysing IoT data, AI algorithms can identify correlations, trends, and anomalies that may not be apparent to humans. This helps organizations make data-driven decisions, optimize processes, and uncover new business opportunities. For instance, in retail, AI algorithms can analyse IoT data from sensors and cameras to understand customer behaviour, preferences, and shopping patterns. This information can be used to personalize marketing campaigns, optimize store layouts, and improve inventory management.

Real-time analytics is another significant advantage of advanced data analytics techniques in IoT. Traditional data analysis methods often involve batch processing, where data is collected and processed in batches, leading to delays in insights. However, with real-time analytics, IoT data can be processed and analysed as it is generated, providing immediate insights, and enabling timely actions. This is crucial in scenarios where quick decision-making and rapid response are essential. For example, in smart cities, real-time analytics of IoT data can help detect and address traffic congestion, monitor air quality, and respond to emergencies promptly.

Furthermore, advanced data analytics techniques can facilitate anomaly detection in IoT data. Anomalies are deviations from the expected patterns or behaviours in the data, which may indicate potential issues or threats. Machine learning algorithms can be trained to recognize these anomalies and raise alerts or trigger automated actions. For instance, in cybersecurity, AI algorithms can analyse network traffic data from IoT devices to detect unusual patterns that may indicate a cyber-attack or unauthorized access. By identifying anomalies in real-time, organizations can take immediate action to mitigate risks and protect their systems and data.

While advanced data analytics techniques offer significant opportunities, they also come with challenges. Handling and processing massive volumes of IoT data require scalable infrastructure and efficient algorithms. Organizations need to invest in robust data storage, processing, and cloud computing resources to handle the data influx. Additionally, ensuring data quality and reliability is crucial for accurate analysis and meaningful insights. Data pre-processing techniques, such as data cleaning, normalization, and feature engineering, may be required to prepare the data for analysis.

Another critical consideration is data privacy and security. IoT devices collect a wealth of sensitive data, including personal information and operational details. Organizations must implement robust security measures, including encryption, access controls, and secure data transmission protocols, to protect this data from unauthorized access and breaches.

3.6 Challenges and Future Directions

Despite the immense potential of IoT, several challenges hinder its widespread adoption. Interoperability and standardization issues, scalability concerns, and data management complexities pose challenges to the seamless integration and deployment of IoT solutions. Energy efficiency and sustainability of IoT devices and networks are also areas of focus. Future directions include the integration of IoT with emerging technologies such as 5G, blockchain, and edge computing to enhance connectivity, security, and data processing capabilities

Chapter 4: Development of Digital Twin for Vibrating Beam

4.1 Set-Up

The first step in the development of the digital twin is to form a physical system whose digital twin is to be made. For this purpose, our set-up consists of the following blocks:

- Beam base
- Beam
- ESP-32 Wi-Fi module
- Vibration sensor
- Vibration inducing motor
- Breadboard
- Jumper wires

The following lines will explain the purpose of these equipment:

4.1.1 ESP-32 Wi-Fi Module

The purpose of this module is to communicate data between the physical system and digital system. Using Wi-Fi signals, it transmits data from the beam system to digital platform over cloud. Similarly, it receives commands from the digital world that are given to regulate the physical system. The data is received via vibration sensor which is transferred to the dashboard. Any command given to physical system is received by ESP-32 and communicated to vibrating motor. Following picture describes the outlook of this module:

Figure 19: ESP-32 Module

4.1.2 Vibration Sensor

The purpose served by this sensor is to receive the information and behaviour of vibration of the beam. This sensor, names WS-420 senses the movement of the beam as it is pasted on it. Any movement in the beam causes a similar movement or vibration in this sensor. Thus, via ESP-32, it transmits the signals of vibration to the serial monitor where the information is post-processed to get the values of vibration frequency in hertz. The sensor's picture is shown below:

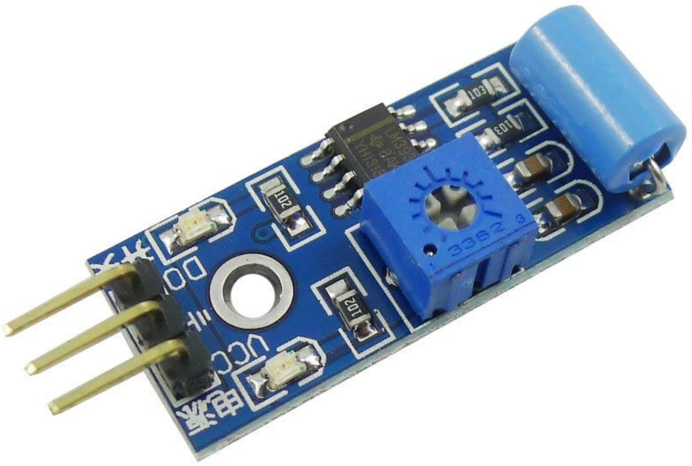


Figure 20: Vibration Sensor

4.1.3 Vibration Inducing Motor

The motor used in the system serves the purpose of generating controlled vibrations in the beam. By inducing these vibrations, it allows for the regulation of the beam's motion according to specific requirements. In contrast, manual, unautomated vibrations cannot be precisely controlled or regulated, making the motor an essential component in achieving the desired results. To ensure the motor operates according to the desired frequency of vibration, it receives commands from the digital world via an ESP-32 module. The ESP-32 acts as a communication interface between the digital control system and the motor. It receives commands from the digital world, typically through a serial monitor or other digital input sources.

The commands sent to the motor via the ESP-32 are typically in the units of hertz (Hz). Hertz is a unit of measurement used to quantify frequency, representing the number of cycles or vibrations occurring per second. In the context of the beam's vibration, the desired frequency of vibration is specified in Hz.

However, before the received command can be directly used to control the motor, it undergoes post-processing or conversion. This post-processing step ensures that the command is appropriately interpreted and translated into the required input for the motor to generate vibrations at the desired frequency.

The specific post-processing steps depend on the motor's design, control mechanism, and compatibility with the ESP-32 module. These steps may involve scaling, calibration, or conversion algorithms to map the command in Hz to the appropriate input range for the motor. For instance, the motor may have a specific range of frequencies it can generate, and the post-processing step adjusts the command to fit within that range.

Once the post-processing is completed, the modified command is sent to the motor, which then adjusts its operation accordingly. The motor's internal mechanisms, such as its control circuitry and drivers, interpret the command and generate the necessary vibrations in the beam at the desired frequency.

By integrating the motor with the digital control system via the ESP-32 module, precise control and regulation of the beam's vibrations are achieved. The ability to receive commands from the digital world and dynamically adjust its operation enables the motor to mimic the desired frequency of vibration accurately. The motor looks like:

Figure 21: Vibration Inducing Motor

The following pictures show the complete set-up from front and top views:

Figure 22: Front-View of the Set-Up for Project

Figure 23: Top-View of the Set-Up for Project

4.2 Code

The development of a digital twin involves creating a virtual replica of a physical system in a digital environment. This replication process is crucial as it allows for real-time monitoring, analysis, and simulation of the physical system. To achieve this, data from the physical system needs to be translated and integrated into the digital twin. This translation is typically done through programming, where the first step involves writing code for the sensors, motor, and ESP-32 module.

The sensors play a vital role in capturing various parameters or variables of the physical system. These sensors could include temperature sensors, pressure sensors, humidity sensors, and many others, depending on the nature of the system being replicated. Each sensor is connected to the ESP-32 module, which acts as a microcontroller capable of collecting data from the sensors.

To begin programming the digital twin, the code for the sensors needs to be written. This code typically involves configuring the sensors and establishing a communication protocol between the sensors and the ESP-32 module. The ESP-32 module serves as the interface between the physical system and the digital twin, facilitating the transfer of data from the sensors to the digital environment.

The next step is to write code for the motor or any other actuator present in the physical system. The motor is responsible for controlling the movement or operation of specific components. By writing code for the motor, it becomes possible to replicate its functionality in the digital twin. The code for the motor usually includes instructions for its operation, such as speed control, direction, and any other relevant parameters.

Once the code for the sensors and motor has been written, the next aspect of programming involves creating a dashboard. The dashboard serves as the user interface for visualizing the data collected from the physical system in real time. It allows users to monitor the parameters, analyse trends, and make informed decisions based on the information provided.

The creation of a dashboard involves designing a graphical interface that presents the data in a clear and intuitive manner. Various programming languages and frameworks can be used for this purpose, such as HTML, CSS, and JavaScript for web-based dashboards, or frameworks like PyQt or Tkinter for desktop applications. These tools provide the necessary functionality to create interactive and visually appealing interfaces.

To display the real-time values, the code for the dashboard needs to establish a connection with the digital twin. This is typically achieved through a communication protocol such as MQTT (Message Queuing Telemetry Transport) or HTTP (Hypertext Transfer Protocol). The code continuously receives data from the ESP-32 module, which is then processed and displayed on the dashboard.

In addition to real-time monitoring, the dashboard can also provide additional features for data analysis and simulation. For instance, it can include graphs, charts, and historical data analysis tools to visualize trends and patterns over time. This allows users to gain insights into the behaviour of the physical system and make predictions or optimizations based on the collected data.

4.2.1 C Code

When developing a digital twin that involves components such as vibration sensors, vibrating motors, and the ESP-32 module, the programming language used is C, and the platform of choice is the Arduino IDE (Integrated Development Environment). This combination provides a user-friendly environment and simplifies the process of writing code for these devices.

The Arduino IDE is specifically designed to support programming for Arduino boards, which are widely used in the development of electronic projects and prototypes. It offers a simple and intuitive interface, making it accessible even to those with limited programming experience. The IDE provides a set of libraries and functions that facilitate the interaction between various components, including sensors, actuators, and microcontrollers.

To start writing code for the digital twin, the first step is to set up the Arduino IDE and configure it for the specific Arduino board being used, such as the Arduino Uno or Arduino Mega, which are commonly utilized in projects involving sensors and actuators. In our case, the desired board is “DOIT ESP-32 DEVKIT” since we have connected the motor as well as the sensor directly to ESP-32.

Next, the code for the vibration sensor can be written. Vibration sensors are typically analog sensors that provide voltage outputs proportional to the intensity of vibrations. The Arduino IDE provides libraries that simplify the process of interfacing with these sensors. The code involves initializing the sensor, configuring its pins, and reading the analog input to obtain the vibration data. The obtained data can then be stored in a variable or transmitted to the ESP-32 module for further processing.

After writing the code for the vibration sensor, the next step is to write code for the vibrating motor. Vibrating motors are actuators that produce vibrations when supplied with power. They are often used to simulate physical vibrations or provide haptic feedback. Arduino boards have digital output pins that can be used to control the operation of the motor.

The code for the vibrating motor involves configuring the digital output pin to which the motor is connected, specifying the desired mode of operation (such as on/off or varying vibration intensity), and controlling the voltage supplied to the motor. By modifying the voltage level or pulsing the motor on and off, various vibration patterns can be achieved.

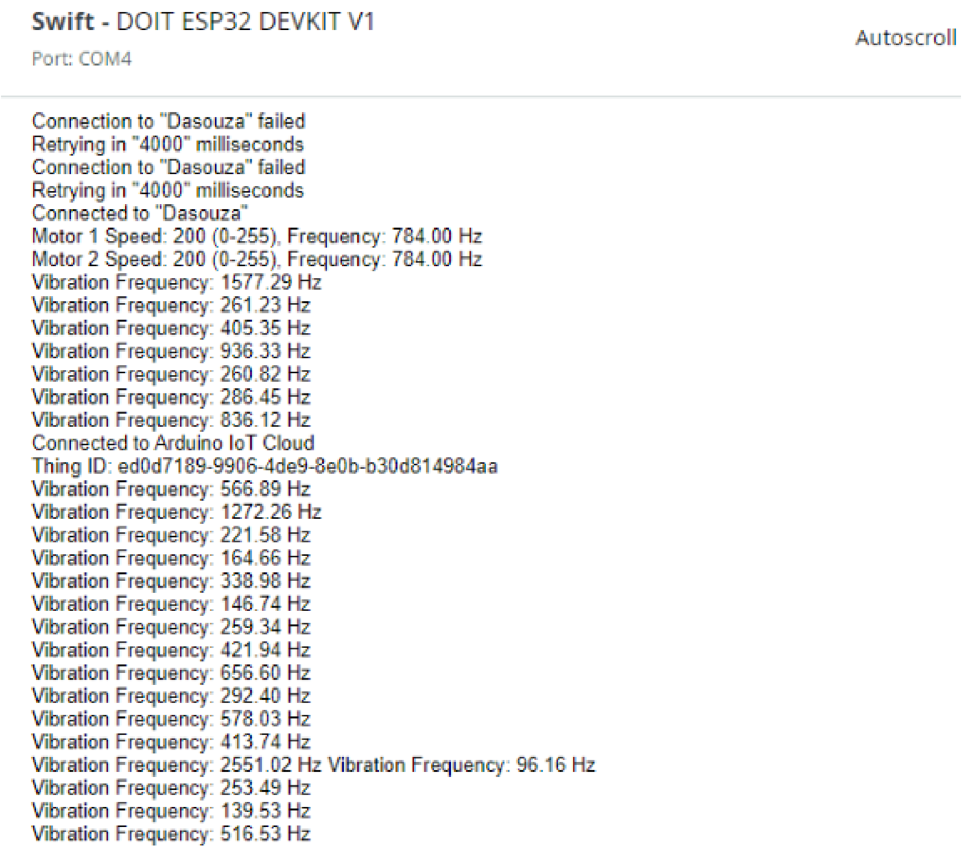
Once the code for the vibration sensor and vibrating motor has been written, the next step is to write code for the ESP-32 module. The ESP-32 is a microcontroller module that provides Wi-Fi and Bluetooth connectivity, making it suitable for transmitting data from the sensors to the digital twin environment.

The Arduino IDE supports the ESP-32 module and provides libraries that simplify the process of establishing communication with it. The code for the ESP-32 involves configuring the Wi-Fi or

Bluetooth settings, establishing a connection to a network or device, and defining the protocol for data transmission.

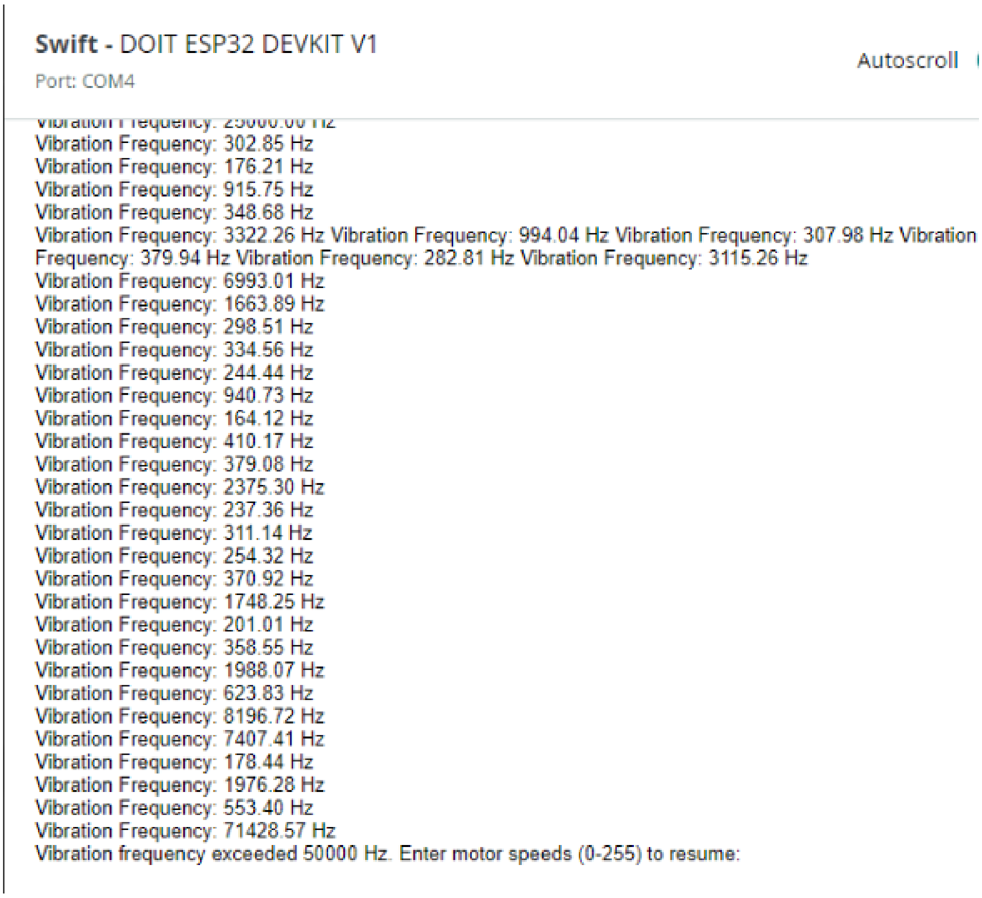
Overall, the combination of C programming language and the Arduino IDE provides a straightforward and accessible platform for writing code for vibration sensors, vibrating motors, and the ESP-32 module. The IDE's user-friendly interface, extensive library support, and compatibility with Arduino boards make it an ideal choice for developing digital twins involving these components. By following the provided libraries, functions, and example code, developers can easily interface with the hardware, configure their behaviour, and establish communication between them. The detailed code is provided in **Appendix 1**.

Pictures of the serial monitor of IDE are shown below which show the real time data acquired form the vibration sensor:



```
Connection to "Dasouza" failed
Retrying in "4000" milliseconds
Connection to "Dasouza" failed
Retrying in "4000" milliseconds
Connected to "Dasouza"
Motor 1 Speed: 200 (0-255), Frequency: 784.00 Hz
Motor 2 Speed: 200 (0-255), Frequency: 784.00 Hz
Vibration Frequency: 1577.29 Hz
Vibration Frequency: 261.23 Hz
Vibration Frequency: 405.35 Hz
Vibration Frequency: 936.33 Hz
Vibration Frequency: 260.82 Hz
Vibration Frequency: 286.45 Hz
Vibration Frequency: 836.12 Hz
Connected to Arduino IoT Cloud
Thing ID: ed0d7189-9906-4de9-8e0b-b30d814984aa
Vibration Frequency: 566.89 Hz
Vibration Frequency: 1272.26 Hz
Vibration Frequency: 221.58 Hz
Vibration Frequency: 164.66 Hz
Vibration Frequency: 338.98 Hz
Vibration Frequency: 146.74 Hz
Vibration Frequency: 259.34 Hz
Vibration Frequency: 421.94 Hz
Vibration Frequency: 656.60 Hz
Vibration Frequency: 292.40 Hz
Vibration Frequency: 578.03 Hz
Vibration Frequency: 413.74 Hz
Vibration Frequency: 2551.02 Hz Vibration Frequency: 96.16 Hz
Vibration Frequency: 253.49 Hz
Vibration Frequency: 139.53 Hz
Vibration Frequency: 516.53 Hz
```

Figure 24: Serial Monitor Showing Real Time Vibration Values



```
Vibration Frequency: 25000.00 Hz
Vibration Frequency: 302.85 Hz
Vibration Frequency: 176.21 Hz
Vibration Frequency: 915.75 Hz
Vibration Frequency: 348.68 Hz
Vibration Frequency: 3322.26 Hz Vibration Frequency: 994.04 Hz Vibration Frequency: 307.98 Hz Vibration
Frequency: 379.94 Hz Vibration Frequency: 282.81 Hz Vibration Frequency: 3115.26 Hz
Vibration Frequency: 6993.01 Hz
Vibration Frequency: 1663.89 Hz
Vibration Frequency: 298.51 Hz
Vibration Frequency: 334.56 Hz
Vibration Frequency: 244.44 Hz
Vibration Frequency: 940.73 Hz
Vibration Frequency: 164.12 Hz
Vibration Frequency: 410.17 Hz
Vibration Frequency: 379.08 Hz
Vibration Frequency: 2375.30 Hz
Vibration Frequency: 237.36 Hz
Vibration Frequency: 311.14 Hz
Vibration Frequency: 254.32 Hz
Vibration Frequency: 370.92 Hz
Vibration Frequency: 1748.25 Hz
Vibration Frequency: 201.01 Hz
Vibration Frequency: 358.55 Hz
Vibration Frequency: 1988.07 Hz
Vibration Frequency: 623.83 Hz
Vibration Frequency: 8196.72 Hz
Vibration Frequency: 7407.41 Hz
Vibration Frequency: 178.44 Hz
Vibration Frequency: 1976.28 Hz
Vibration Frequency: 553.40 Hz
Vibration Frequency: 71428.57 Hz
Vibration frequency exceeded 50000 Hz. Enter motor speeds (0-255) to resume:
```

Figure 25: Serial Monitor Showing Real Time Values With Limit Exceeded

4.2.2 Azure Code

Microsoft Azure is a cloud computing platform that offers a wide range of services, including tools and capabilities for creating digital twins. With Azure, developers and organizations can harness the power of the cloud to build, deploy, and manage digital twin solutions efficiently. Azure provides a platform that enables the collection of data from various sources, which can then be used to create and monitor digital twins.

To create a digital twin using Azure, the first step is to establish data ingestion from the physical system into the Azure cloud. This can be done through various methods such as IoT (Internet of Things) devices, sensors, gateways, or other data sources. These devices are configured to collect data from the physical system and transmit it to Azure using protocols like MQTT, HTTP, or AMQP.

Azure IoT Hub is a central component that facilitates secure and reliable bi-directional communication between the physical devices and the cloud. It acts as the message broker, allowing devices to connect and transmit data to the cloud in a scalable and efficient manner. IoT Hub supports a wide range of protocols and provides features for device management, security, and monitoring.

Once the data is ingested into Azure, it can be processed and analysed using various services. Azure Digital Twins is a dedicated service designed to help create, model, and manage digital twins. It provides a rich set of APIs and tools for defining the properties, relationships, and behaviours of digital twin components. Azure Digital Twins allows developers to model the physical system and its various elements, such as sensors, actuators, and devices, using a digital representation. The service also enables the creation of relationships between these components, allowing for a holistic view of the system.

The data collected and processed in Azure can be used to update and synchronize the state of the digital twin. By continuously feeding the digital twin with real-time data, it becomes a dynamic and accurate representation of the physical system. This synchronization allows for monitoring the current state of the physical system and enables real-time analysis and decision-making based on the digital twin's data.

Azure offers various visualization and dashboarding capabilities that can be utilized to create intuitive and customizable real-time dashboards for monitoring the digital twin. Services like Azure Dashboards and Power BI provide powerful visualization tools that can be integrated with Azure services. These tools allow developers to create interactive dashboards that display key performance indicators (KPIs), metrics, charts, and graphs, providing a comprehensive view of the digital twin and its associated data. The detailed code of Microsoft Azure is given in **Appendix 2**.

4.2.3 Dashboard

The dashboard created on Microsoft Azure is used to view the real time values of frequency of the vibrating beam. The data is displayed through a frequency-meter that is custom made for this twin. The data is also displayed on a graph that shows the frequency over a range of time for an easy comparison. The dashboard is created for desktop as well as mobile phone for easy remote monitoring. The following images of vibration gauge and vibration chart show real time values received from the sensor. The gauge shows a relative meter of vibration frequency and the chart shows a time-stamp history of vibration frequency values received from the sensor.

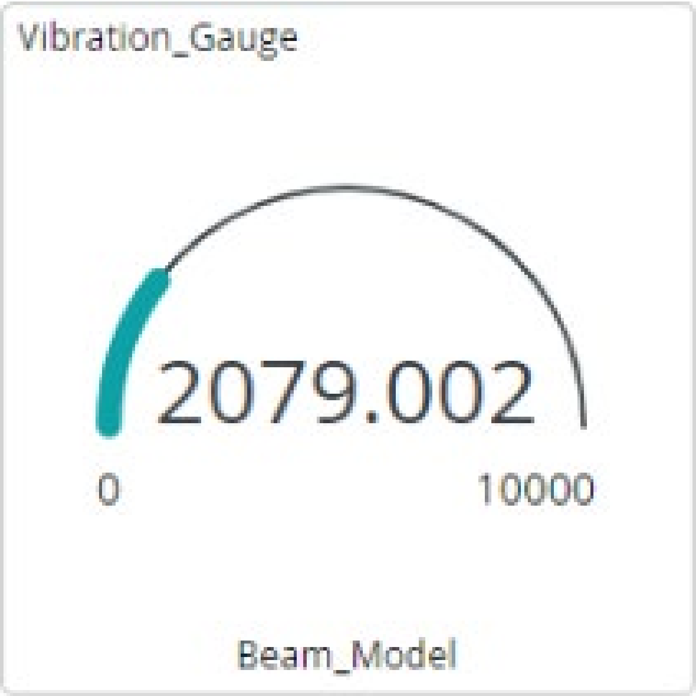


Figure 26: Vibration-Gauge Showing Real Time Values

Figure 27: Vibration-Chart Showing Real Time Values With Time-Stamp

In the above chart, the x-axis represents the time-stamp at which a particular value is received from the sensors and the y-axis represents the intensity of vibrations of the beam.

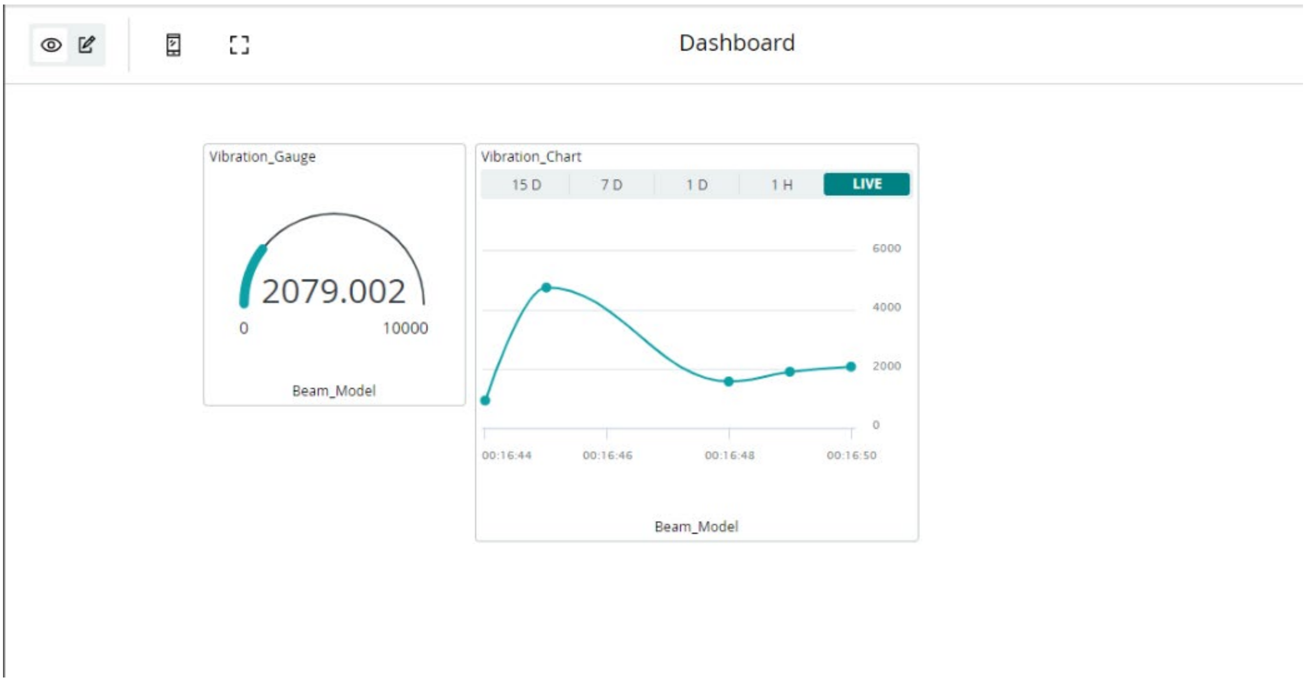


Figure 28: Azure Dashboard

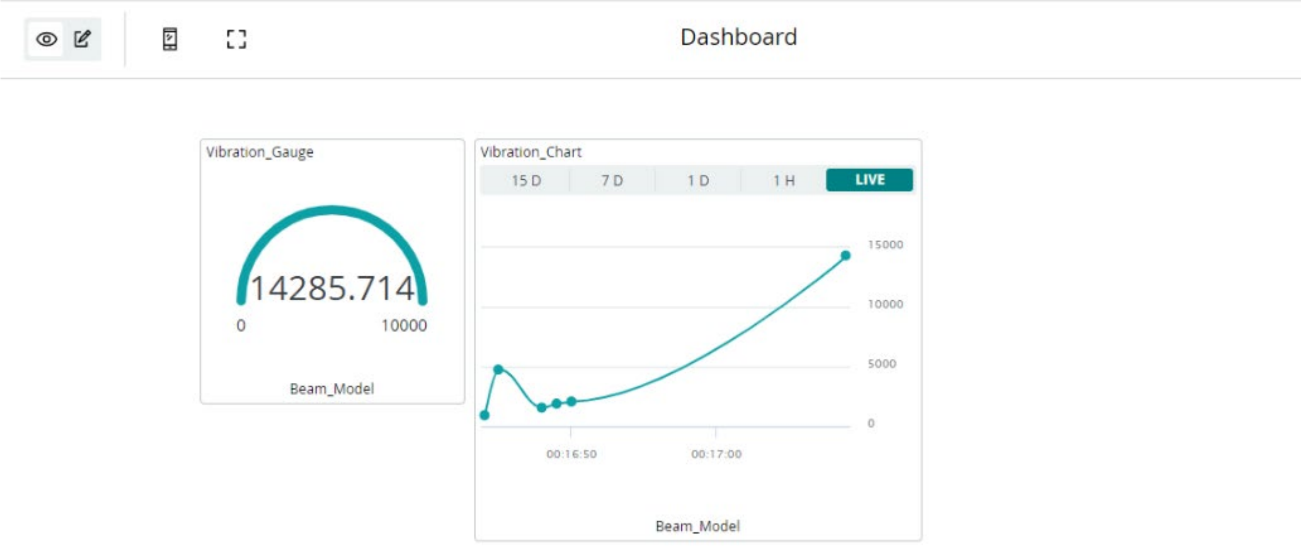


Figure 29: Azure Dashboard with Limit Exceeded

4.2.4 Mobile Dashboard

Just as a desktop dashboard, a mobile dashboard has also been created for the ease of monitoring of data. The dashboard looks like:

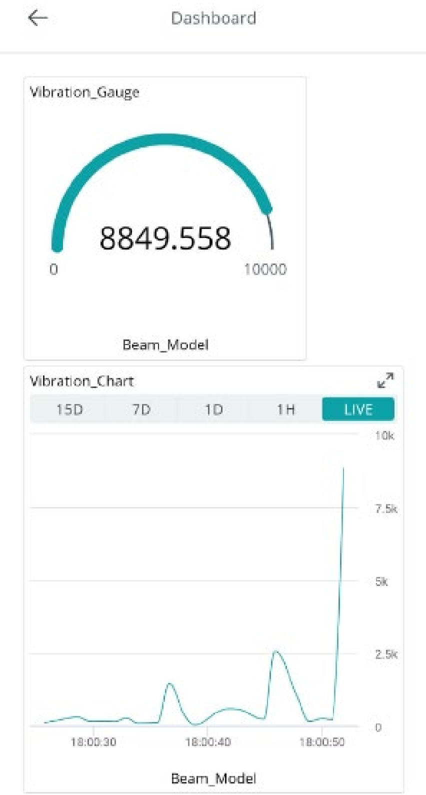


Figure 30: Mobile Dashboard

4.3 Conclusion

In conclusion, this final year project successfully developed a digital twin for the vibration analysis of an overhanging beam. The digital twin demonstrated its capability to upload real-time frequency values from the physical beam setup to a digital dashboard, providing valuable insights into the beam's vibrational behaviour.

The digital twin's ability to receive commands from the dashboard and mimic the vibration of desired frequencies opens up numerous possibilities for practical applications. Engineers and researchers can utilize this digital twin to study and analyse different vibration scenarios, optimize structural designs, and explore potential solutions to mitigate vibration-related issues.

By integrating the physical and digital domains, this project showcases the potential of digital twins in enhancing structural analysis and design processes. The digital twin serves as a powerful tool for monitoring and understanding the dynamic behaviour of the overhanging beam in real time, enabling proactive maintenance and ensuring the beam's performance and safety.

Furthermore, the project highlights the importance of data-driven decision-making in structural engineering. The real-time frequency data collected from the physical setup, coupled with the digital twin's analytical capabilities, enables informed and efficient decision-making to address vibration-related challenges.

Overall, this project contributes to the advancement of digital twin technology and its practical application in structural analysis. It lays the foundation for future research and development in this field, offering new possibilities for enhancing structural safety, performance, and efficiency.

4.4 Future Developments

Some possible developments in the future to further this project include:

i. Development of a digital twin for an industrial application such as for a compressor, pump, machines, etc.
ii. Integration of machine learning algorithms in these twin for predictive health maintenance rather than real time health monitoring.
iii. Improved communication between physical and digital worlds for better data collection and upload.

Appendix 1: IDE Code

```
#include "thingProperties.h"

const int motorPin1 = 2;         // Pin connected to the first motor
const int motorPin2 = 15;        // Pin connected to the second motor
const int vibrationSensorPin = 32;   // Pin connected to the SW-420 vibration sensor

void setup() {
  // Initialize serial and wait for port to open:
  Serial.begin(9600);
  // This delay gives the chance to wait for a Serial Monitor without blocking if none is found
  delay(1500);
   pinMode(motorPin1, OUTPUT);     // Set the first motor pin as output
  pinMode(motorPin2, OUTPUT);     // Set the second motor pin as output

  ledcSetup(0, 5000, 8);          // Configure LEDC channel 0 with a frequency of 5kHz and resolution
of 8 bits
  ledcSetup(1, 5000, 8);          // Configure LEDC channel 1 with a frequency of 5kHz and resolution
of 8 bits
  ledcAttachPin(motorPin1, 0);    // Attach LEDC channel 0 to the first motor pin
  ledcAttachPin(motorPin2, 1);    // Attach LEDC channel 1 to the second motor pin

  pinMode(vibrationSensorPin, INPUT);   // Set the vibration sensor pin as input

  initProperties();

  // Connect to Arduino IoT Cloud
  ArduinoCloud.begin(ArduinoIoTPreferredConnection);

  /*
    The following function allows you to obtain more information
    related to the state of network and IoT Cloud connection and errors
    the higher number the more granular information you’ll get.
```

```
    The default is 0 (only errors).
    Maximum is 4
 */
  setDebugMessageLevel(2);
  ArduinoCloud.printDebugInfo();
}

void loop() {
  ArduinoCloud.update();
  // Your code here
   static int intensity1 = 0;       // Intensity value for the first motor
  static int intensity2 = 0;       // Intensity value for the second motor

  if (Serial.available() > 0) {
    intensity1 = Serial.parseInt();  // Read the intensity value for the first motor from the serial monitor
    intensity2 = Serial.parseInt();  // Read the intensity value for the second motor from the serial
monitor

    // Map the input intensity values to the motor speed range
    int motorSpeed1 = map(intensity1, 0, 255, 0, 255);
    int motorSpeed2 = map(intensity2, 0, 255, 0, 255);

    ledcWrite(0, motorSpeed1);      // Set the speed of the first motor
    ledcWrite(1, motorSpeed2);      // Set the speed of the second motor

    float frequency1 = map(motorSpeed1, 0, 255, 0, 1000);  // Map motor speed to frequency range
(0Hz - 1000Hz)
    float frequency2 = map(motorSpeed2, 0, 255, 0, 1000);  // Map motor speed to frequency range
(0Hz - 1000Hz)

    Serial.print("Motor 1 Speed: ");
    Serial.print(motorSpeed1);
    Serial.print(" (0-255), Frequency: ");
    Serial.print(frequency1);
```

```
    Serial.println(" Hz");

    Serial.print("Motor 2 Speed: ");
    Serial.print(motorSpeed1);
    Serial.print(" (0-255), Frequency: ");
    Serial.print(frequency1);
    Serial.println(" Hz");

    while (Serial.available() > 0) {
      Serial.read();  // Clear any remaining input in the serial buffer
    }
  }

  // Read the vibration sensor value
  int vibrationValue = digitalRead(vibrationSensorPin);

  if (vibrationValue == HIGH) {
    float vibrationFrequency = calculateVibrationFrequency();
    vibrations = vibrationFrequency;
    Serial.print("Vibration Frequency: ");
    Serial.print(vibrationFrequency);
    Serial.println(" Hz");

    if (vibrationFrequency > 5000) {
      // Stop both motors
      ledcWrite(0, 0);
      ledcWrite(1, 0);

      Serial.println("Vibration frequency exceeded 5000 Hz. Enter motor speeds (0-255) to resume:");
    }
  }

  delay(500);
}
```

```
float calculateVibrationFrequency() {
  unsigned long startTime = micros();
  unsigned long endTime;

  while (digitalRead(vibrationSensorPin) == HIGH) {
    // Wait for the sensor to go LOW
  }

  endTime = micros();
  unsigned long pulseDuration = endTime - startTime;

  // Calculate vibration frequency in Hz
  float frequency = 1000000.0 / pulseDuration;
  return frequency;
}
```

Appendix 2: Azure Code

```
"@context": "dtmi:dtdl:context;2",
  "@id": "dtmi:com:example:dummy;2",
  "@type": "Interface",
  "displayName": "Practice model",
  "contents": [
    {
      "@type": "Property",
      "name": "Vibrations",
      "schema": "double"
    }
  ]
}
// Default URL for triggering event grid function in the local environment.
// http://localhost:7071/runtime/webhooks/EventGrid?functionName={functionname}
using System;
using Microsoft.Azure.WebJobs;
using Microsoft.Azure.WebJobs.Host;
using Microsoft.Azure.EventGrid.Models;
using Microsoft.Azure.WebJobs.Extensions.EventGrid;
using Microsoft.Extensions.Logging;
using Azure.DigitalTwins.Core;
using Azure.Identity;
using System.Net.Http;
using Azure.Core.Pipeline;
using Newtonsoft.Json.Linq;
using Newtonsoft.Json;
using Azure;
```

```
namespace yousufeisafyp
{
    public static class Function1
    {
        private static readonly string adtInstanceUrl =
Environment.GetEnvironmentVariable("ADT_SERVICE_URL");
        private static readonly HttpClient singletonHttpClientInstance = new HttpClient();

        [FunctionName("IOTHubtoTwins")]
        public async static void Run([EventGridTrigger]EventGridEvent eventGridEvent, ILogger
log)
        {
            if (adtInstanceUrl == null) log.LogError("Application setting \"ADT_SERVICE_URL\"
not set");
            try
            {
                var cred = new ManagedIdentityCredential("https://digitaltwins.azure.net");

                var client = new DigitalTwinsClient(
                new Uri(adtInstanceUrl),
                cred,
                new DigitalTwinsClientOptions
                {
                    Transport = new HttpClientTransport(singletonHttpClientInstance)
                });

                log.LogInformation($"ADT service client connection created.");

                if (eventGridEvent != null && eventGridEvent.Data != null)
```

```
                {
                    log.LogInformation(eventGridEvent.Data.ToString());

                    // convert the message into a json object
                    JObject deviceMessage =
(JObject)JsonConvert.DeserializeObject(eventGridEvent.Data.ToString());

                    // get our device id, temp and humidity from the object
                    string deviceId = (string)deviceMessage["systemProperties"]["iothub-
connection-device-id"];
                    var vibration = deviceMessage["body"]["Vibration"];

                    //log the vibration
                    log.LogInformation($"Device:{deviceId} Vibration is:{vibration}");

                    // Update twin with vibration for our esp 32>
                    var updateTwinData = new JsonPatchDocument();
                    updateTwinData.AppendReplace("/Vibration", vibration.Value<double>());
                    await client.UpdateDigitalTwinAsync(deviceId, updateTwinData);
                }
            }

            catch (Exception ex)
            {
                log.LogError($"Error in ingest function: {ex.Message}");
            }

        }
    }}
```

References

[1] TIBCO, “TIBCO,” TIBCO, [Online]. Available: https://www.tibco.com/reference-center/what-is-iiot.

[2] S. L. Ullo and G. Sinha, “Advances in Smart Environment Monitoring Systems Using IoT and Sensors,” 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/11/3113.

[3] S. Shea, “TechTarget,” [Online]. Available: https://www.techtarget.com/iotagenda/definition/smart-home-or-building.

[4] Guo, D., Zhao, D., Cheng, H., Wang, D., & Hu, J. (2021). Digital twin for condition monitoring and fault diagnosis of centrifugal pumps: A review. Renewable and Sustainable Energy Reviews, 137, 110669.

[5] Li, Y., Li, H., Li, M., & Li, S. (2019). Design of pump condition monitoring and fault diagnosis system based on digital twin. 2019 IEEE International Conference on Industrial Cyber Physical Systems (ICPS), 101-106.

[6] Cheng, H., Hu, J., Guo, D., Wang, D., & Zhao, D. (2021). A Review of Digital Twin-Based Fault Diagnosis Methods for Centrifugal Pumps. IEEE Access, 9, 772-789.

[7] Li, Y., Li, H., Li, M., & Li, S. (2020). Digital twin and its applications in condition monitoring and fault diagnosis of industrial equipment: A review. Measurement, 167, 108284.

[8] Sisworahardjo, N., & Nurhadi, L. (2020). A review of digital twin technology in condition monitoring and maintenance. IOP Conference Series: Materials Science and Engineering, 851(1), 012047.

[9] Zhao, D., Cheng, H., Wang, D., Guo, D., & Hu, J. (2020). Digital twin for predictive maintenance and prognostics of centrifugal pumps: A review. Energy

Procedia, 187, 102-107.

[10] N. Malik, A. Abbas, and M. H. Zahid, "Digital Twins: A Literature Review," IEEE Access, vol. 8, pp. 45412-45425, 2020.

[11] S. Nair and P. R. Nair, "Real-time Condition Monitoring and Fault Diagnosis of Pumps Using Internet of Things and Machine Learning: A Review," Measurement, vol. 142, pp. 273-283, 2019.

[12] X. Zhang, C. Wei, and K. Wang, "Digital Twin: A Review from Control Perspective," IEEE Access, vol. 7, pp. 24054-24070, 2019.

[13] Ansys. (2021). Ansys. [online] Available at: https://www.ansys.com/ [Accessed 15 Feb. 2023].

[14] Microsoft Azure. (2023). Azure Functions. [online] Available at: https://azure.microsoft.com/en-us/services/functions/ [Accessed 15 Feb. 2023].