Available online at www.sciencedirect.com

Manufacturing Letters

Manufacturing Letters 41 (2024) 1464–1475

52nd SME North American Manufacturing Research Conference (NAMRC 52, 2024)

Manufacturing Floor Mapping and Presence Tracking with a Physics-Based Game Engine

John Osho[a]*, Esther Akinrinde [a], Carson Tillery [a], Ethan Koenig[a], Kaylee Adams [a], Gregory T. Purdy [a]

Samuel Ginn College of Engineering, Auburn University, 345 W. Magnolia Ave., Auburn, AL 36849, USA

* Corresponding author. E-mail address: johnaosho@auburn.edu

Abstract

The manufacturing shop floor exists in a dynamic environment with constantly changing components regarding position and features. Presence tracking on the manufacturing shop floor is essential to enhance effective operations and safety. Internet of Things technology solutions have been developed to track and monitor machine and operators' movements and materials on the shop floor to enhance machine availability and worker safety and improve overall operational effectiveness. However, for SMMs with limited financial resources and infrastructure, there is a need for a financially friendly approach to shop floor presence tracking and monitoring. This work presents the use of Internet-of-things and physics-based engines to develop a scalable and customizable solution to track presence on a manufacturing floor. The paper aims to improve and promote the adoption of I-4.0 technologies and digital manufacturing by SMMs with the development of an affordable, repeatable, customizable, open-source solution. The article describes a methodology for mapping, modeling, monitoring, and reporting presence on a manufacturing floor using physics-based engines. The methodology describes mapping the manufacturing floor into zones, the design of the IoT solution, selecting and placing sensors, and presenting the data using dashboards and physics-based engines. The paper investigates the methodology by implementing a case study on the Interdisciplinary Center for Advanced Manufacturing Systems (ICAMS) manufacturing floor using the Unity3D game engine, open-source software, and off-the-shelf motion sensors.

Keywords: IoT; Smart manufacturing; I-4.0; Manufacturing floor monitoring; Physics-based game engine; Digital manufacturing

1. Introduction

Industry 4.0 (I-4.0) integrates advanced digital technologies into manufacturing processes and represents the next evolution of digital manufacturing. The transition to I-4.0 is occurring through the implementation of various technologies, including autonomous robots, simulation, horizontal and vertical system integration, the Industrial Internet of Things (IIoT), augmented reality, cybersecurity, big data analytics, cloud computing, and additive manufacturing [1].

I-4.0, driven by intelligence, connectivity, and flexible automation, aims to create a comprehensive, unified approach to managing processes through these drivers by integrating all processes in each system. This is achieved by integrating heterogeneous data and knowledge [2]. I-4.0 in the manufacturing industry provides the capacity to offer supreme oversight and control for operations and businesses for every aspect of the production cycle, from start to finish. This enhanced control extends to improved collaboration and information sharing among various stakeholders, including departments, partners, suppliers, products, and personnel. Moreover, integrating human labor into this automated environment fosters ongoing improvements and emphasizes activities that generate value.

The Internet of Things (IoT) is central to operating a highly digitalized system that I-4.0 aims to achieve [3]. The system is designed to facilitate a seamless and ongoing flow of data and information, enabling communication between machines and human interaction. This convergence bridges the physical and digital domains, resulting in the development of sophisticated, intelligent, and autonomous systems.

The foundation of decision-making and automation in manufacturing operations is monitoring activities by data collection from every component involved in the process [4]. Proper monitoring of operators' activities in manufacturing by generating data and information needed to make managerial decisions is significant to stakeholders. Decisions regarding routine or predictive maintenance, safety protocols, operations planning, compliance and reporting, and resource optimization can be made effectively when data monitoring systems are set in place. The benefits of tracking activities on the manufacturing floor include cost reduction, efficiency in operations, reduced risk of occupational hazard, effective maintenance, improved productivity, and effective planning. Manufacturing floor management is important in successfully integrating I-4.0 [5].

Shop floor monitoring solutions have been developed using sensors, IoTs, and digital twins to improve production and safety on the manufacturing floor and monitor various manufacturing processes [4, 5]. However, solutions proposed by [4] and [5] are commercially available solutions and are not focused on affordability, privacy, integration, or flexibility [6]. The lack of financial resources, management support, technological infrastructure, and fear of failure are some barriers preventing small and medium-sized manufacturers (SMM) from adopting I-4.0 [6]. This creates a need for more financially friendly and infrastructure-flexible approaches toward creating I-4.0 solutions for SMMs to promote market competitiveness.

To address the adoption concerns of SMMs, the study discussed in the manuscript develops an affordable, flexible, and scalable IoT solution focused on privacy and integration into existing environments or environments with limited IT resources. The study describes the implementation of a real-time presence monitoring system for manufacturing floor operations by using off-the-shelf sensors and harnessing the potential of IoT and physics-based game engines. Notably, this system is crafted using a physics-based game engine technology and an IoT sensor network solution to provide the real-time status of active zones on the manufacturing floor for monitoring and managing its processes. The aim of this study is to:

- Provide a methodology to model, map, and group a manufacturing shop floor into zones for effective presence tracking.
- Provide a methodology of using a physics-based game engine to select, place, and orient motion sensors on a manufacturing floor to maximize coverage.
- Provide a methodology to present real-time manufacturing floor presence data using a physics-based game engine.
- Developing an affordable, repeatable, customizable, open-source manufacturing floor monitoring solution for SMMs.

The remainder of this paper is structured as follows. Section 2 reviews related works, such as using I-4.0, IoT, sensors, and game engines in manufacturing. Section 3 provides the methodology. Section 4 provides a case study and discusses the process of sensor selection, manufacturing floor mapping for sensor placement, and in-depth integration of the manufacturing floor into the game engine. Section 5 is the discussion section, which details the proposed solution's effectiveness. Finally, Section 6 discusses some recommendations and a brief conclusion of the work.

2. Related Work

2.1. Industry 4.0 and digital manufacturing

The fourth technological revolution, or I-4.0, aims to increase operational efficiency and hinges on cyber-physical systems and the IoT [3]. This integration includes various advanced technologies such as additive manufacturing, the IoT, big data, machine learning, high-speed computing, and robotics. Adopting I-4.0 technologies has proven beneficial in navigating complex systems, facilitating digital transformation, and minimizing physical contact in operations. Real-time capability, interoperability, and service orientation are some of the principles of I-4.0 [2]. Digital manufacturing integrates innovative technologies from I-4.0, such as the IoT and artificial intelligence, in manufacturing processes.

These innovations are instrumental in enhancing manufacturing operations through immediate monitoring and control, anticipatory maintenance, optimized resource utilization, and the capability to tailor products and procedures. This transforms traditional manufacturing into a flexible, effective, and information-centric environment.

The need to achieve maximum efficiency and overall improvement of processes encourages the adoption of I-4.0 in manufacturing operations [7]. Tao & Zhang implemented a Digital Twin Manufacturing floor and discussed its key components to bridge the gap between the virtual and physical worlds in manufacturing [8]. Mourtzis et al. developed an IoT-Based Monitoring System for manufacturing floor control [4]. Cohen et al. suggested using a Digital twin and IoT technologies to facilitate solutions for disruptions experienced when operating an assembly line [9]. Sotskov suggested using Digital twins, and IoT can be extended to include manufacturing operations in a broader scope [10].

2.2. Manufacturing floor monitoring

A manufacturing floor typically involves humans and machines working together [11]. Filippo Bosi et al. emphasized the importance of keeping up with changes in the manufacturing floor environment and proposed a way to analyze results rapidly from machine data [11]. Bellavista et al. proposed a monitoring system for older manufacturing machines by quickly processing machine data to enhance the reactive services such as predictive maintenance and remote controlling in a smart factory [12].

According to Cohen et al., self-aware sensors can be used to avoid most of the disruptions experienced on an assembly line

[9]. IoT and sensors have been applied to many operational processes to enhance productivity, including business processes, healthcare, and manufacturing. The use of sensors makes a manufacturing facility more flexible and intelligent. To ensure efficiency, it is essential to identify the appropriate sensors based on the specific situation.

Oyekan et al. proposed a solution to monitor workers' ergonomic conditions in real time and mitigate the risk of musculoskeletal injuries with wearable sensors based on the adaptive control of thought-rational (ACT_R) [13]. Immediate corrections can be made to their work postures, thus improving work and overall productivity of workers. Prathima et al. proposed using electronic sensors for real-time data collection and analysis of the Overall Equipment Efficiency (OEE) of a CNC machine in manufacturing [14].

Islam et al. proposed using sensors for monitoring workers' motion and breathing patterns in a smart factory, particularly in a toxic production environment. The study discussed using ultra-wideband (UWB) sensors to collect real-time data on workers' breathing and motion, which is then analyzed using machine learning and deep learning methods [15]. Schmidt et al. used electroencephalography sensors to obtain information about the brain activities of operators while interacting with the two primary manufacturing floor management systems identified as the rigid structure approach and continuous improvement approach [5].

2.3. Game Engine in manufacturing

Game engines are software applications that allow the integration of multiple resources and use physics engines, scripting integration, multithreading capability, and scene management. Game engines can render graphics and simulate mechanical behavior, kinematics, momentum, and collision detection [16]. As a result, they are applied in modeling physics-based systems, real-time data visualization, simulation, virtual reality, and the creation of digital twins or virtual representations of the manufacturing floor and assets.

Jun et al. used game engines to enhance the collaboration between autonomous systems and humans to improve smart manufacturing environments [17]. Zhu built a virtual reality (VR) environment to improve the problem-solving skills of engineering students using the Unity game engine and HTC Vive VR [18]. Matsas & Vosniakos discussed using virtual reality simulation systems to simulate the safer collaboration between humans and robots using the Unity Game engine[19].

Wang et al. developed a system for effective task management on a manufacturing floor using Ubiquitous Augmented reality technology (SSUAR). Real-time sharing of data between machines, operators, and sensors provided by this system is used to manage tasks and for effective resource allocation [20].

Kuts et al. developed a digital twin solution using Unity3D as the data visualization and input handling platform and MQTT for connectivity with sensors/robots/machines on the manufacturing floor [21]. Li et al. also presented a solution for monitoring and visualizing the data gathered from machines/robots on the manufacturing floor using Unity3D. The study used Robot Operating System (ROS) to simulate the movements of the real robot in the visualization [22].

Samir et al. created a digital twin of the manufacturing floor with real-time updates using Unity3D, which provided visualization and real-world environments [23]. Zarco et al. recommended Unity3D and Unreal Engine, among others, because they met the requirements in each aspect of their study. [16]. Our study uses the Unity game engine to model, plan, and report real-time presence detection on a manufacturing floor.

3. Methodology

3.1. Define goals, requirements, and constraints

At this stage, the requirements and specifications of the proposed solution are identified. This can be accomplished by assessing the current state of the manufacturing floor and meeting with key stakeholders. The project's goals, budget, time, design, and technological constraints are documented.

3.2. Manufacturing floor mapping and modeling

Once key stakeholders sign off on the documented requirement, the manufacturing floor is mapped into zones. A two or three-dimensional model of the manufacturing floor depicting the mapped zones is created. Information about the building dimensions, manufacturing equipment, processes, staff qualification, and safety are used for mapping and modeling the manufacturing floor.

3.3. Planning

The IoT solution is designed based on the use cases, requirements, and constraints at this stage. Sensors, software, communication protocols, data management, reporting, and other components are identified. The system architecture is also designed. It is important that the IoT solution integrates into the current system and adheres to technological and design constraints.

3.4. Sensor selection

The appropriate sensors that meet the use cases identified in the previous stages are identified. The sensor technology, communication protocol, precision, limitations, and flexibility are factors to be considered when selecting a sensor. Experiments are designed for the sensors based on the use cases and are used to validate the manufacturers' specifications. The results and information gathered from the experiments are used to select the sensors.

3.5. Sensor modeling, placement, and orientation

A two or three-dimensional model of the sensor's capture volume is created using information about the sensor. Once the sensor model is created, it is imported into the model of the manufacturing floor. The position of the sensor can be adjusted for coverage. Additional experiments to gather more information about the orientation of the sensor can be carried out to improve the accuracy of the sensor's model and placement results.

3.6. Data collection and reporting

The IoT solution is created by developing and integrating all the individual components of the system architecture. The solution is integrated into the existing data collection system. The data collected is transformed into valuable information using dashboards and physics-based game engines. Information provided by the report should meet the stakeholders' needs and specifications.

3.7. Testing verification and experimentation

Once the IoT solution has been installed and integrated, tests and experiments are conducted to identify bugs, accuracy, blind spots, and limitations of the new IoT system. The results from the experiments, tests, and stakeholders' feedback are used to improve the system.

4. Case Study

The Interdisciplinary Center for Advanced Manufacturing Systems (ICAMS), a manufacturing facility at Auburn University, was used for this case study. The ICAMS facility trains and educates students as well as industry personnels in advanced manufacturing technologies and supports the university and local businesses with their machining needs. ICAMS is equipped with additive and subtractive industrial machinery. The case study aims to design, implement, and deploy motion monitoring systems to track people on ICAMS's manufacturing floors with game engines, open-source software, and affordable commercial off-the-shelf motion sensors. The objectives of the case study are:

- Track motion on the manufacturing floor.
- Determine the time spent in each zone.
- Provide a real-time representation of active zones.
- Generate notifications when motion is detected in restricted zones.

4.1. Defining goals, requirements, specifications, and constraints

The goals, requirements, budget, time, technological constraints, and the manufacturing floor's current state were determined by conducting on-site assessments and meeting with key stakeholders. The primary requirements of the manufacturing floor monitoring solution were to be affordable, reusable, customizable, scalable, and able to integrate with the existing data collection system.

An assessment of the current state of the manufacturing floor was conducted. Information about the dimensions of the manufacturing floor, network layout, existing monitoring and alert systems, and data collection technology were analyzed. A report was generated as described, and the information was added to the system requirements and constraints.

The ICAMS facility comprises two manufacturing buildings, 1480 and 1490, each 10,000 square feet and 50 feet apart. Building 1490 is fully networked, with ethernet jacks installed around the building and surveillance cameras placed at the entry and exit of the manufacturing floor. The building hosts an edge Windows server, and access to surveillance footage is limited to the facility's office. Building 1480 has no network connectivity and surveillance. The project aimed to track people in both buildings and provide all employees with information and statistics. The timeframe for the completion of the project was four months.

4.2. Manufacturing floor design, mapping, and modeling

The manufacturing floor was mapped into zones based on safety, purpose, and manufacturing process. The manufacturing process zones contain groups of machines, equipment, and tools used for a particular manufacturing process. ICAMS can perform cutting, engraving, welding, additive, and subtractive processes and is equipped with three and five-axis vertical machining centers, a multi-turret lathe, a laser engraver, metal additive machines, Electrical Discharge Machines (EDM), and a waterjet.

The safety zones are areas within the facility where unqualified individuals should not have access to unsupervised. In ICAMS, the manual equipment areas are restricted to untrained personnel.

The zones in ICAMS are grouped based on three purposes: storage, the location of a machine, and multipurpose locations. The areas were tagged, and the building plan was updated to reflect the zones and areas.

4.2.1. Two-Dimensional Manufacturing floor Modeling

A two-dimensional model of the manufacturing floor was created in Fusion 360 (colored zones overlaid in Microsoft PowerPoint) using information about the facility's dimensions and zone mapping. Figure 1 depicts a two-dimensional model of the ICAMS manufacturing floor plan in PowerPoint.

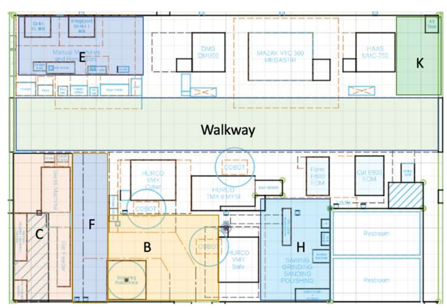

Figure 1 Two-dimension shopfloor model.

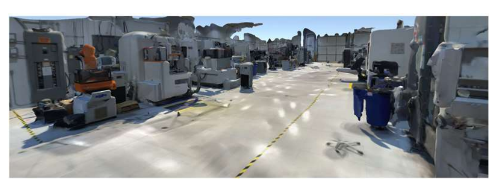

Figure 2 Three-dimensional shopfloor model.

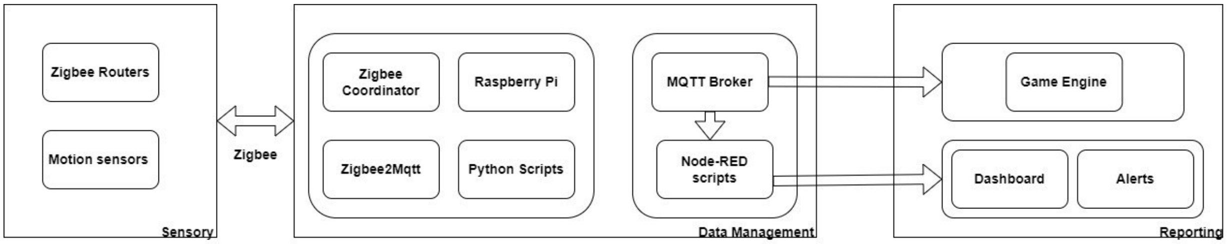

Figure 3. System architecture

4.2.2. Three-Dimensional Manufacturing floor Modeling

A three-dimensional manufacturing floor model was created using "Polycam" with an iPhone 14 Pro. Polycam is a 3D model creation software that utilizes modern mobile devices' LiDAR and camera capabilities.

An iPhone 14 Pro was used to scan the manufacturing floor, and the 3D model was exported as a .obj file. The artifacts of the 3D model (floating geometry, inaccurate models, etc.) were remedied using Blender – a 3D modeling software. The 3D model was imported into Unity3D, a physics-based game engine software. The textures of the 3D model were replaced with a single flat white color texture in Unity3D, and the floor of the 3D model was replaced with a flat plane. Figure 2 depicts a three-dimensional model of the ICAMS manufacturing floor.

4.3. Planning

The planning stage deals with the design of the manufacturing floor monitoring IoT solution. The team identified sensors, protocols, hardware, and software specifications required to support and integrate the IoT solution into the existing system.

4.3.1. Identify use cases

The use cases were identified from the information gathered about the project requirements from the stakeholder's meeting and the initial assessment report. The use cases are:

- Track motion on the manufacturing floor.
- Determine the time spent in each zone.
- Provide a real-time representation of active zones.
- Generate notifications when motion is detected in restricted zones.

The information was used for safety, planning, tracking, reporting, and alerts.

4.3.2. Identify the required components

A solution was designed to meet the required use cases and specifications and integrate with the existing network and data collection technologies. The solution comprises three subsystems: IoT, data aggregation, and game engine. The subsystems were grouped into three distinct categories: sensory, data management, and reporting. The sensory component comprises motion sensors, communication radios, communication protocols, and software that work together to detect and transmit sensor data. The data management component includes network protocols, open-source software, databases, and microcontrollers for receiving, formatting, processing, and storing sensory data. The reporting components are dashboards, alerts, and game engine tools used to produce alerts and provide users with reports and insights about the sensory data.

4.3.3. Design the system architecture

System architecture represents and describes the structure and behaviors of the system [24]. The manufacturing floor presence monitoring system comprises sensors, Zigbee routers, a Zigbee coordinator, a network node, a game engine, a dashboard, and alert tools. The motion data sensed by the motion sensors is transmitted to the network node by Zigbee routers and coordinators. The network node formats and transmits the data over MQTT to the game engine and the node-red scripts. The script generates alerts and transforms the sensor data into usable information reported on dashboards. Figure 3 represents the system architecture for the IoT solution.

4.4. Sensor selection

The motion sensors selected operate on passive infrared (PIR) technology. PIR sensors are electronic sensors that measure infrared (IR) light radiating from objects in their field of view. PIR sensors detect general movement but provide no information about who or what moved. Applications include motion detectors, security alarms, and automatic lighting applications [25]. PIR motion sensors were selected for the case study because of their affordability and accessibility.

Figure 4. PIR Sensors

Six affordable and market-accessible sensors were shortlisted for this case study: Smart Things, Philips, Xiaoming, Third Realty, TPlink, and Ikea. Figure 4 represents the six PIR sensors.

Three use cases were identified based on the floor map, zoning, and the stakeholders' needs. They are:

- Report motion or movement between zones, doors, and restricted areas.
- Detect and report motion within zones,
- Detect and report motion on walkways.

Experiments were conducted on the sensors by connecting them wirelessly to a network node over ZigBee and viewing the data with node-red. The selection of sensors for the use cases was made based on the results of the experiments.

The first experiment compared the response times of the sensors. The second experiment determined the sensors' reporting interval, which is the time the sensors take to report that motion is no longer detected. The third experiment determined the sensors' field of view, the maximum linear distance each sensor could detect.

4.4.1. Sensor response time

The sensors' response time is essential for accurate and real-time reporting. The test was performed to determine how long it takes for each sensor to detect motion. The experiment was conducted using two stopwatches set to measure how long the sensor took to report detection while an operator walked briefly in front of the sensors. The times were recorded, and an average response time for each sensor was obtained.

Table 1. Sensor Response Time

Sensor Name	Average Response Time (secs)	Standard Deviation	Min	Max	No of samples
Smart Things	0.38	0.09	0.23	0.53	10
Philips	0.23	0.09	0.13	0.40	10
Xiaomi	0.50	0.14	0.21	0.66	10
Third Realty	0.92	0.28	0.60	1.45	10
TPLink	N/A	N/A	N/A	N/A	N/A
Ikea	0.68	0.10	0.54	0.87	10

Table 1 shows the results obtained for the sensor response time experiment. Philips had the quickest response time, with an average of 0.23 seconds. TPlink was found to be too unstable in its response to movement and hence considered the least fit for use.

4.4.2. Sensor reporting interval

An experiment was conducted to determine how long the sensors take to report that motion is no longer detected by placing an object in the sensor's field of view. While most sensors reported a change of status when a moving object was no longer present, the TPLink and Ikea sensors only reported the presence of a moving object. This was used to disqualify the use of the sensors.

Table 2. Sensor Reporting Intervals.

Sensor Name	Average Duration (secs)	Standard Deviation	Min	Max	No of Samples
Philips	9.90	0.00	9.89	9.91	20
Smart Things	16.80	1.59	15.13	21.03	20
Third Reality	33.23	0.48	32.25	33.98	12
Xiaomi	90.01	0.01	90.01	90.03	17

Table 2 shows the results obtained for the sensor reporting experiment. Philips had the quickest reporting with an average of 9.90 seconds, while Xiaomi had the slowest with a mean reporting time of 90.01 seconds.

4.4.3. Sensor distance field of view

The experiment was carried out to determine the maximum linear distance the sensors can detect motion. Each sensor was tested at distances between 0 and 35ft in 5ft increments. A person walks into the sensor's field of view to see if the sensor reports the detection of motion.

Table 3. Sensor Distance Range

Sensor Name	Distance (feet)
Smart Things	35
Philips	30
Xiaomi	20
Third Realty	10
TPLink	N/A
Ikea	N/A

Table 3 shows the results obtained for the sensor distance field of view tests. Smart Things has the longest distance, covering 35 feet, while Third Realty detects motion not more than 10 feet away. Philips and Xiaomi can detect motion 30 and 20 feet away, respectively.

4.4.4. Defining sensors characteristics

Sensors were selected based on experiment results, cost, size, flexibility, and use case. Flexibility is defined as the sensor's orientation and ease of installation. The distance field of view is the maximum linear distance the sensor can detect the motion. The response time is the time it takes the sensor to detect and report motion. Table 4 describes the key characteristics of each sensor used in this case study.

Table 4. Sensor Characteristics

Sensor Name	Cost ($)	Average Response Time (secs)	Average Status Report Duration (Secs)	Distance Field of View (Feet)	Flexibility
Smart Things	39.99	0.65	16.8	35	Flexible
Philips	39.99	0.64	9.9	30	Slightly Flexible (90 degrees)
Xiaomi	24.99	1.11	90.01	20	Flexible
Third Realty	15.99	1.95	33.23	10	Not Flexible
TPLink	12.99	2.14	-	-	Slightly Flexible (90 degrees)
Ikea	18.95	1.66	-	-	Not Flexible

The information from Table 4 was used to score the sensors based on their characteristics. High cost is categorized as $30 and above, medium cost is between \$15 and \$30, while low cost is less than \$15. Smart Things and Philips were the most expensive, while TPlink was the cheapest.

The sensors were scored as small, medium, and large based on the size (weight and height) of the sensors. Third Reality and Xiaomi were rated as small, Philips, Smart Things, Ikea medium, and TP-Link as large.

The sensor’s average response time was used to group the sensors into fast and slow sensors using 0.5 seconds as the benchmark. Smart Things, Philips, and Xiaomi were regarded as fast sensors, while Third Realty and Ikea were slow sensors.

The distance field of view was used to group the sensor into short-, medium-, and long-range sensors. Sensors with a field of view above thirty feet were considered long-range, those between thirty and fifteen medium-range, and those less than fifteen feet were considered short. Smart Things and Philips scored as long-range sensors, Xiaomi medium-range, and Third Reality as short-range.

4.4.5. Selecting the sensors

The three primary criteria for reporting movement between zones were for the sensor to have a small field of view, small size, low cost, and a moderate reporting interval. The Third Reality sensor was selected because it is small, cheap, easy to install in tight spaces, and has a limited field of view.

The sensor’s flexibility, response time, distance field of view, and reporting intervals were the primary requirements for tracking motion within the zone. The use case required a sensor with a long-range, fast response time and reporting time to capture motion accurately and calculate the time spent in each zone. Motion capturing in zones also required flexible sensors for ease of installation and the ability to orient the sensor at different angles to capture planes. The Smart Things and Philips sensors scored highly in most categories, and even though the Philips sensor has a faster reporting time, the Smart Thing sensor was selected because it is very flexible and can be oriented in various angles and directions.

The sensor requirements to track the walkway motion were small size, flexibility, distance field of view, and response time. The Smart Things and Xiaomi sensors scored high in this category; the Xiaomi was selected because it was smaller and cheaper. TPlink and Ikea sensors were not considered for selection in any use cases because neither scored well in the experiment. Table 5 represents the sensor selection criteria based on the use cases.

Table 5. Sensor scores for selection criteria

Use Cases	Flexibility	Resp. Time	Range	Reporting Interval	Cost	Size
Zone tracking	Flexible	Fast	Long	Fast	N/A	N/A
Interzone tracking	N/A	N/A	small	moderate	low	small
Walkway tracking	Flexible	Fast	N/A	N/A	N/A	medium

4.5. Sensor modeling, placement, and orientation

To determine the best positioning for the sensors throughout the facility, experiments were carried out to determine the sensors' field of view and orientation. This information was used to determine the orientation and the total number of sensors needed to cover a given area.

4.5.1. Creating a model of the sensors

The number of sensors and their placement in each zone was determined by modeling the sensor and manufacturing floor. PowerPoint, Fusion 360, and Unity3D were used to create two- and three-dimensional models of the sensors and manufacturing floor. Information about the sensor’s field of view and orientation, building dimensions, and 3D scans were required to model the sensor on the manufacturing floor. The sensor model was created using the data gathered about the sensor from the distance and orientation field of view experiments. The details about the number of sensors are vital for planning and costing.

Figure 5 Sensor orientation apparatus.

4.5.2. Sensor orientation field of view test

This experiment investigated the orientation and capture volume of the sensor by varying the angle and distance. A plastic container shaped like a hollow cylinder with a flat section inside was used for testing. The plastic container restricted the sensor's field of view to selected angles. The sensors were tested at different angles and varying distances by placing the sensor into the container at the desired angle and distance and then determining if the sensor detected motion. The apparatus is shown in Figure 5.

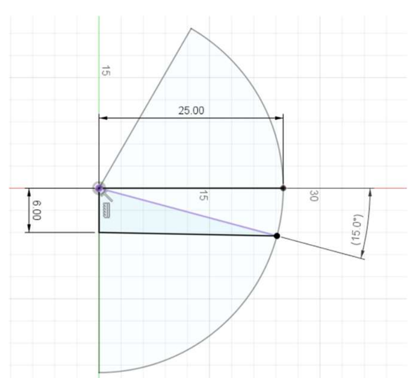

Figure 6 Sensor field of view model.

The sensor's field of view is restricted to angles between 0 and 75 degrees, and the sensor's motion detection was investigated at distances between 7.5 and 30 feet for each angle. The experiment result on the Smart Things motion sensor was a distance range of 25 inches and an angle range of 150 degrees. Sensor Modeling using the Unity game Engine. Figure 6 depicts the field of view of the sensor.

4.5.3. Sensor Modeling using the Unity game engine

The Unity engine's lighting system was used to model the field of view of the sensor as a spotlight using information about the dimension of the sensor's field of view. The sensor's linear distance and orientation range are 7.62 meters and 150°, respectively. The sensor's capture volume values can be adjusted in real-time in Unity3D, making the model customizable to different sensors. The floor distance covered by the sensor is measured with a tool from Microsoft's MRTK Graphics Tools package. Figure 7 shows a model indicating the dimensions of the capture volume of the field of view of the Smart Things sensor in Unity3D.

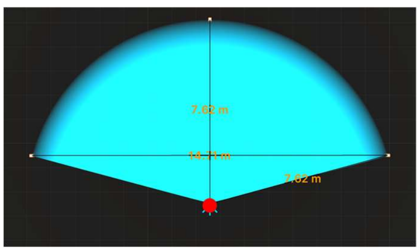

Figure 7 Sensor three-dimensional field of view model in Unity3D.

4.5.4. Sensor placement based on the use case and properties

The position and the number of sensors for each zone were determined by adjusting the position of the sensor model within the 3D model of the manufacturing floor in Unity3D. By utilizing Unity3D's lighting system, the area visible to the sensor can identify impedances to the sensor's capture volume. Objects in front of the sensor will cast a shadow on objects behind them. Figure 8 depicts the motion sensor's reaction to objects, showing the shadow created by a person standing (represented by a capsule) in the sensor's view.

Unity3D provides flexibility to place the sensor in different locations and angles, providing the ability to adjust height and angle to confine the capture volume of the sensor to a particular zone. Figures 9 and 10 show the three-dimensional model of the sensor's field of view as it moved up and down a wall on the shop floor. The sensor model was adjusted in various locations in Unity3D to determine the best location to capture motion in the area without blind spots after accounting for angles, orientation, distance, ease of installation, access, and obstruction by machines or objects.

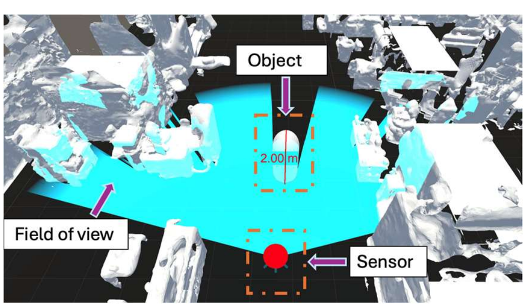

Figure 8. Unity shadow detection

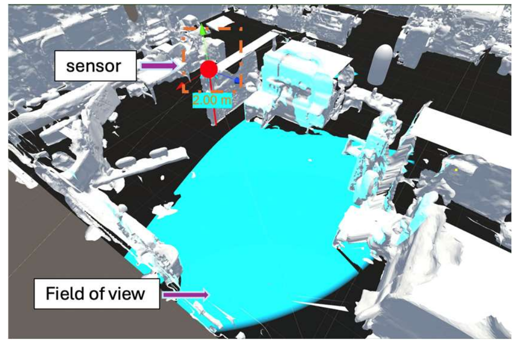

Figure 9. Sensor placement on top

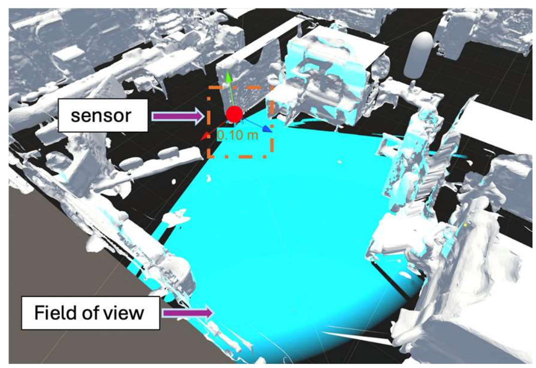

Figure 10. Sensor placement at the bottom

4.5.5. Sensor distribution on the Manufacturing Floor

The game engine determined the number of sensors required to cover each zone and their placement, direction, angle, and height. The study was able to integrate twenty-three PIR sensors from three different manufacturers, Third Reality, Smart Things and Xaiomi into the manufacturing floor. The number of sensors per zone and their distribution across the

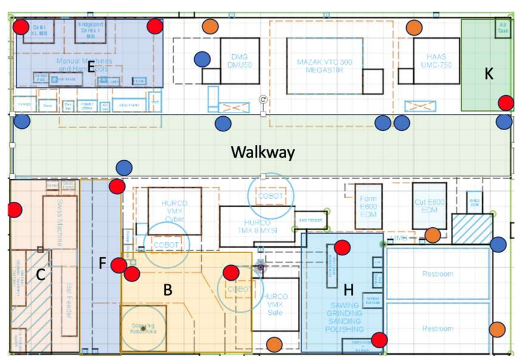

Figure 11 Shop floor sensor location.

manufacturing floor are shown in Figure 11. Zones B, F, E, and H have two sensors each, Zones C and K have one sensor, the walkway has six sensors and there are five sensors between zones. The red circles represent Smart Things sensors, blue Xiomi sensors, and brown circles Third reality sensors.

4.6. Data Collection and reporting

The collection of sensor data is an essential phase of the project. The data collected from the PIR sensors provides insight into the traffic on the manufacturing floor, which is vital for planning, defining thresholds, and generating alerts, notifications, reports, and dashboards. The data collection pipeline comprises Zigbee routers, a network node, node-red scripts, and a MongoDB database. The network node was developed using a Raspberry Pi running zigbee2mqtt, python scripts, and an MQTT broker.

Zigbee is an open standard wireless mesh networking protocol for wireless machine-to-machine (M2M) communication developed for low-power IoT devices and sensors [26]. Zigbee2Mqtt is an open-source project that enables communication with Zigbee devices from different vendors without the vendor's bridge or gateway and provides access to the data over MQTT [27]. MQTT is a lightweight publish/subscribe messaging protocol for IoT devices [28].

A ZigBee mesh network was built covering the two ICAMS buildings, each 10,000 square feet and 50 feet apart, with Zigbee2Mqtt and ZigBee routers. Motion data is sent from the PIR sensors to the ZigBee routers and passed to the ZigBee coordinator with ZigBee2Mqtt. The ICAMS ZigBee mesh network is represented in Figure 12, where the star node represents the coordinator, blue nodes represent the routers, and green nodes represent the sensors.

The sensor send data over MQTT to the node-red script once motion is detected. The sensor data is formatted, mapped into zones by the node-red scripts, and transformed into useful information by the game engine and dashboards. The occupancy statistics dashboard and game engine representation are accessible to users on the ICAMS network.

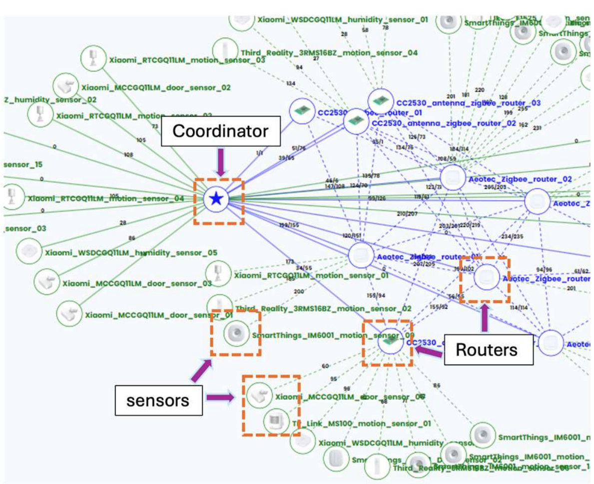

Figure 12 Sensor ZigBee network.

4.6.1. Integrating the unity system into the data collection pipeline

Unity3D systems were integrated into the active data collection pipeline with the M2MQTT open-source library. M2MQTT allows the Unity3D system to connect to an MQTT broker and subscribe to topics. An MQTT topic is defined for every zone on the manufacturing floor. The actions within the Unity3D system are controlled by the messages sent to the MQTT topics. The corresponding point on the manufacturing floor model is turned on or off based on the content of the message.

4.6.2. Dashboards

The Information about the zone status (active or inactive), the maximum duration spent in the zone at a given time, and the aggregate of the duration spent in a zone on a given day are calculated and stored in the MongoDB and InfluxDB databases using nod-red scripts. The information is retrieved from the databases and presented on the Grafana dashboard. Figure 13 depicts the Grafana dashboard showing the statistics of each zone.

4.6.3. Game engine presence representation

A custom shader based on Unity3D's collision system was used to represent heat map points on the manufacturing floor. Points were modeled as 3D spheres. The colliders of the sphere activate when the shader is enabled. The MQTT message received by the Unity3D MQTT receiver is used to turn on or off the corresponding heat map point.

A PNG image of the labeled manufacturing floor map is placed on the top-down, isometric view of the 3D model of the manufacturing floor. This was accomplished with Unity3D's "canvas" system, which allows for dynamic scaling of screen overlays based on the display resolution and aspect ratio.

Two versions of the game engine representation were developed: a WebGL version – which can be hosted as a web server and accessed by computers on the network – and a standalone Windows version. Figure 15 shows the manufacturing floor's real-time presence detection as heatmaps in Unity3D.

4.7. Testing verification and experimentation

The game engine solution was installed on the manufacturing floor. Tests were conducted on the game engine solution to confirm that correct zones were activated, sensor overlaps, blind spots, response time, constraints, and system limitations were identified. The game engine was presented to stakeholders, and questionnaires were used to obtain feedback.

4.7.1. Test to show zone coverage

The test aimed to confirm total coverage in each zone and prevent the sensor's capture volume from overlapping with other zones. The test was carried out by monitoring the dashboards as an operator walked within zones to test for areas

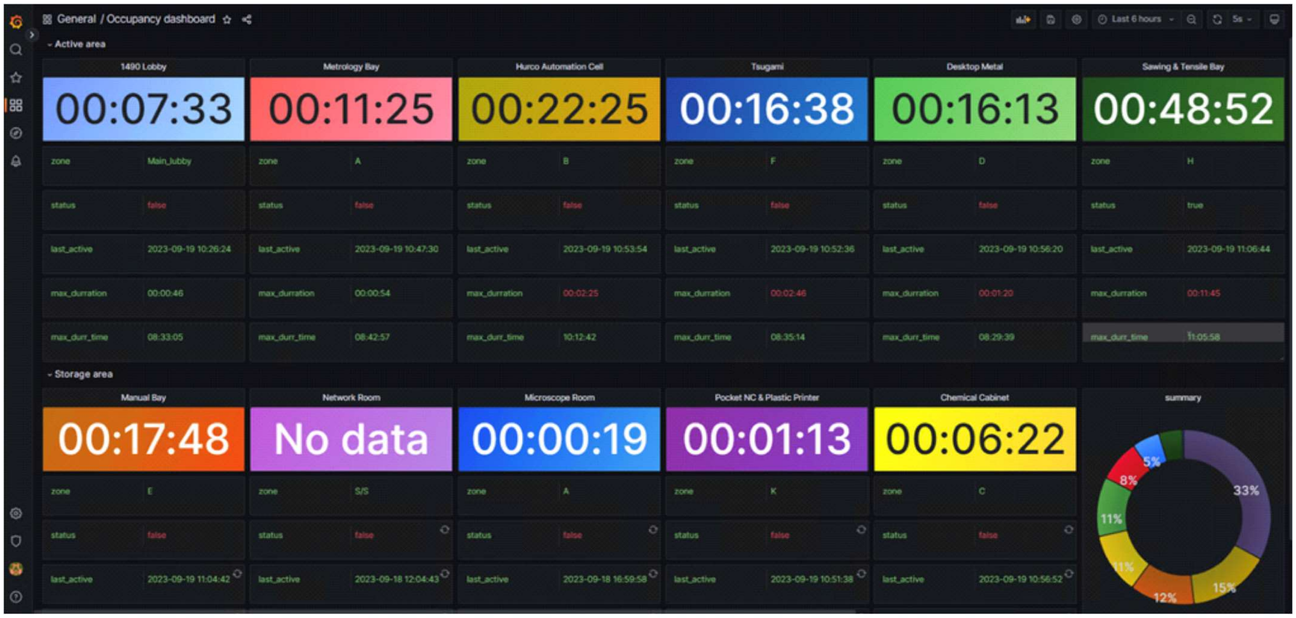

Figure 13 Presence statistics dashboard

of potential blind spots and across zone boundaries to test for overlaps. The dashboards were used to confirm if the motion was detected, and adjustments were made to the orientation of the sensor as required. Information such as positions, heights, view angles, and, most significantly, the effect of objects placed in the sensor's field of view on the capture volume provided by the game engine was valuable for the placement and orientation of the sensor.

The limitation of the sensor was used to our advantage as sensors were adjusted to use manufacturing machines to reduce the sensor's capture volume and confine the sensors' field of view to their respective zones. Zones with more than one sensor were oriented such that their field of view overlaps. Figure 14 depicts zones B, C, and E, as well as the location and direction of the sensors. The red dot represents the sensors, and the sectors represent their range and direction. Figure 14 will be used as an example to describe the zone overlap test.

As depicted in Figure 14, the sensor's field of view in Zone C overlaps with Zone F. The sensor is placed 1.7 ft above the ground at an angle of 0°, using the Swiss machine to restrict its field of view to zone C.

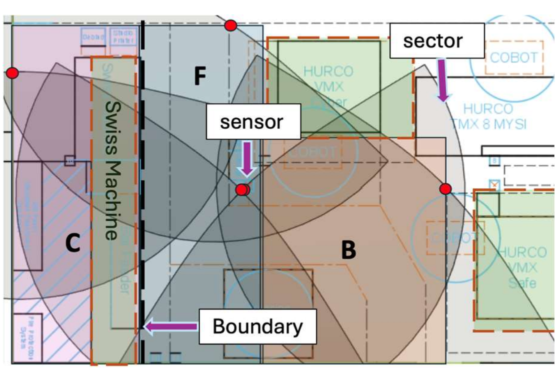

Figure 14 Sensor coverage test.

After the coverage tests, the two sensors in zone B were installed at 6 and 6.7 ft above the ground and oriented at angles 5° and 45°, respectively. The two sensors in zone F were installed at 2.7 and 6.5 ft above the floor and oriented at 0° and 35°, respectively. The sensor in zone C was installed at 1.7 ft at 0°.

4.7.2. Experimentation to show response time

The experiment is to determine the amount of time Unity3D takes to represent the data reported by the sensors. The response time of the game engine solution was obtained by experiment to compare with the average response time of the sensors used. The experiment involved moving an object in front of the sensor briefly to record the time it takes to activate the game engine representation. Eight trials were run with three stopwatches, making the total number of samples 24. The average time for the game engine representation to be activated in response to the motion is 1.91 seconds with a standard deviation of 0.4, a minimum time of 1.11 seconds, and a maximum of 2.8 seconds.

5. Discussion

This section focuses on the effectiveness of our approach in modeling and representing in real-time presence detection on the manufacturing floor. The methodology's practicality, affordability, flexibility, repeatability, and customizability were examined using the case study.

The study provides a methodology to model a manufacturing floor, capture the volume of a PIR sensor, and render the 3D models in the Unity3D game engine. Unity3D allows the sensor to be placed in different locations and oriented in position and angle to capture the required capture volume. Implementing the case study demonstrated that the

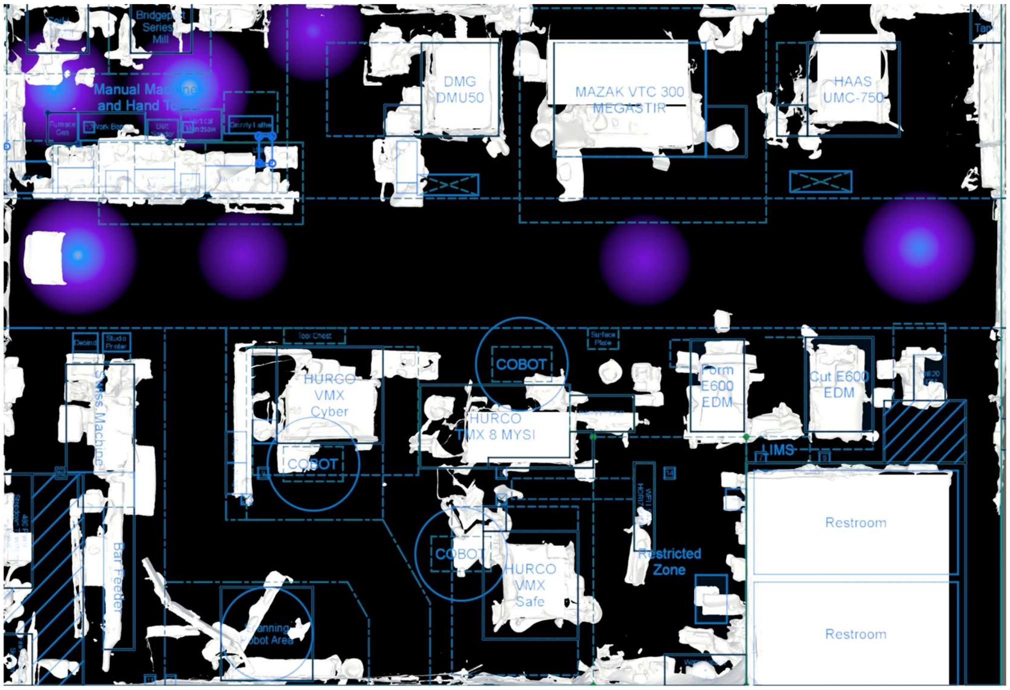

Figure 15 Shop floor representation

approach is customizable to manufacturing facilities of varied sizes.

The methodology is affordable, flexible, and repeatable because it incorporates and integrates multiple inexpensive, dissimilar, off-the-shelf sensors using open-source software and tools. The approach can include other Zigbee sensors, such as temperature and humidity.

The methodology proposed in this study is scalable and suitable for sites with limited IT infrastructure. The mesh network created by the Zigbee routers and coordinators can be extended by adding more Zigbee routers. The range of the Zigbee network coverage was extended to another building, 1480, without ethernet, wireless, or internet infrastructure in the case study.

The IoT solution reports and represents real-time motion or presence on the manufacturing floor with a dashboard and a Unity3D heat map. The average response time of the dashboard and the Unity3D representation are 0.65 and 1.91 seconds, respectively. The average response time of the Unity game engine representation is slightly slower than the dashboard. A response time of less than 2 seconds is acceptable for the Unity game engine representation in this case study because the system does not reply to it for calculation, reports, and alerts.

A limitation of the IoT solution is the PIR technology used by the sensors. The sensor's accuracy can be questioned when two operators are positioned parallel to the position of the sensor because it only detects the operator directly in front of it. As a result, cameras might be needed to augment the monitoring system.

6. Conclusion

The paper promotes the adoption of I-4.0 technologies by SMMs with the development of an affordable, repeatable, customizable, open-source solution. The study describes a customizable methodology for mapping, modeling, monitoring, and reporting presence on a manufacturing floor with physics-based engines. A case study was carried out to validate the methodology by demonstrating the design, planning, implementation, and deployment of real-time manufacturing floor monitoring IoT solutions. The main contributions of this paper are:

1. A methodology of using a physics-based game engine to design, place, and orient sensors on the shop floor for total coverage and minimize blind spots.
2. An affordable open-source solution to integrate multiple inexpensive dissimilar off-the-shelf sensors for small and medium enterprises.
3. The development of an IoT sensor network solution for tracking presence on the manufacturing shop floor.
4. A dashboard that tracks and presents the daily occupancy in each manufacturing floor area.
5. A methodology to represent in real-time the presence data on a manufacturing floor using a physics-based game engine.

Affordability, privacy, integration, flexibility, and lack of IT infrastructure are some barriers to the adoption of I-4.0 by SMM [6]. By harnessing the power of the physics-based engine and open-source software, the study was able to build an affordable, scalable, and flexible solution within a short time frame. The solution also achieves the integration and privacy objectives by integrating into the existing system and saving all data locally. The study described the mapping of the manufacturing floor into zones, the design of an IoT solution that integrates into the existing IT infrastructure, the placement of twenty-three PIR sensors from various manufacturers on the manufacturing floor, and the presentation of data using dashboards and physics-based engines.

7. Future work

Manufacturing floor monitoring aims to improve efficiency, enhance quality, and ensure safety, amongst other things. In this paper, we developed a presence monitoring IoT solution.

In the future, the solution can be improved by integrating Radio Frequency Identification (RFID), image recognition, and other technology to identify the people accessing restricted zones. Other sensors can also be integrated to capture more information about the environmental condition of the manufacturing floor, such as temperature and humidity.

The game engine representation can be further developed by integrating virtual reality technology. This will give users a virtual walkthrough of the manufacturing floor, allowing them to see live data from sensors and machines and view detailed statistics about the shop floor.

Implementing corrective actions during the monitoring process is essential. Hence, an automated system can be implemented to execute corrective actions such as sound alarms, emergency stops on machines, or manufacturing floor lockdowns in the case of extreme safety or security breaches.

Acknowledgements

This research was funded by Grant W52PLJ-20–9-3045, awarded to the Interdisciplinary Center of Advanced Manufacturing Systems (ICAMS) by the US Army.

References

[1] Michael Rüßmann et al., "Industry 4.0: The Future of Productivity and Growth in Manufacturing Industries," Boston Consulting Group, vol. 9, no. 1, 2015.
[2] Y. Lu, "Industry 4.0: A survey on technologies, applications and open research issues," Journal of Industrial Information Integration, vol. 6. Elsevier B.V., pp. 1–10, Jun. 01, 2017. doi: 10.1016/j.jii.2017.04.005.
[3] V. Roblek, M. Meško, and A. Krapež, "A Complex View of Industry 4.0," Sage Open, vol. 6, no. 2, Jun. 2016, doi: 10.1177/2158244016653987.
[4] D. Mourtzis, N. Milas, and A. Vlachou, "An internet of things-based monitoring system for shop-floor control," J Comput Inf Sci Eng, vol. 18, no. 2, Jun. 2018, doi: 10.1115/1.4039429.
[5] D. Schmidt, J. V. Diez, J. Ordieres-Meré, R. Gevers, J. Schwiep, and M. Molina, "Industry 4.0 lean shopfloor management characterization using EEG sensors and deep learning," Sensors (Switzerland), vol. 20, no. 10, May 2020, doi: 10.3390/s20102860.
[6] H. M. Elhusseiny and J. Crispim, "SMEs, Barriers and Opportunities on adopting Industry 4.0: A Review.," in Procedia Computer Science, Elsevier B.V., 2021, pp. 864–871. doi: 10.1016/j.procs.2021.12.086.
[7] A. Jamwal, R. Agrawal, M. Sharma, and A. Giallanza, "Industry 4.0 technologies for manufacturing sustainability: A systematic review and future research directions," Applied Sciences (Switzerland), vol. 11, no. 12. MDPI AG, Jun. 02, 2021. doi: 10.3390/app11125725.
[8] F. Tao and M. Zhang, "Digital Twin Shop-Floor: A New Shop-Floor Paradigm Towards Smart Manufacturing," IEEE Access, vol. 5, pp. 20418–20427, Sep. 2017, doi: 10.1109/ACCESS.2017.2756069.
[9] Y. Cohen, M. Faccio, F. Pilati, and X. Yao, "Design and management of digital manufacturing and assembly systems in the Industry 4.0 era," International Journal of Advanced Manufacturing Technology, vol. 105, no. 9. Springer, pp. 3565–3577, Dec. 01, 2019. doi: 10.1007/s00170-019-04595-0.
[10] Y. N. Sotskov, "Assembly and Production Line Designing, Balancing and Scheduling with Inaccurate Data: A Survey and Perspectives," Algorithms, vol. 16, no. 2. MDPI, Feb. 01, 2023. doi: 10.3390/a16020100
[11] Filippo Bosi et al., "Cloud-enabled Smart Data Collection in Shop Floor Environments for Industry 4.0," 2019 15th IEEE International Workshop on Factory Communication Systems (WFCS), 2019.
[12] Paolo Bellavista et al., "Design Guidelines for Big Data Gathering in Industry 4.0 Environments," Institute of Electrical and Electronics Engineers, Jun. 2019.
[13] J. Oyekan, Y. Chen, C. Turner, and A. Tiwari, "Applying a fusion of wearable sensors and a cognitive inspired architecture to real-time ergonomics analysis of manual assembly tasks," J Manuf Syst, vol. 61, pp. 391–405, Oct. 2021, doi: 10.1016/j.jmsy.2021.09.015.
[14] P. B A, S. P N, and S. P M, "Shop floor to cloud connect for live monitoring the production data of CNC machines," Int J Comput Integr Manuf, vol. 33, no. 2, pp. 142–158, Feb. 2020, doi: 10.1080/0951192X.2020.1718762.
[15] F. B. Islam, J. M. Lee, and D. S. Kim, "Smart factory floor safety monitoring using UWB sensor," IET Science, Measurement and Technology, vol. 16, no. 7, pp. 412–425, Sep. 2022, doi: 10.1049/smt2.12114
[16] L. Zarco, J. Siegert, T. Schlegel, and T. Bauernhansl, "Scope and delimitation of game engine simulations for ultra-flexible production environments," Procedia CIRP, vol. 104, pp. 792–797, Jan. 2021, doi: 10.1016/J.PROCIR.2021.11.133.
[17] M. B. G. Jun, H. Yun, and E. Kim, "Human Expertise Inspired Smart Sensing and Manufacturing," in Proceedings of International Conference on Electronics, Communications and Information Technology, ICECIT 2021, Institute of Electrical and Electronics Engineers Inc., 2021. doi: 10.1109/ICECIT54077.2021.9641237.
[18] R. Zhu, "Eye-Track Modeling of Problem-Solving in Virtual Manufacturing Environments," American Society for Engineering Education, 2021, 2021.
[19] E. Matsas and G. C. Vosniakos, "Design of a virtual reality training system for human–robot collaboration in manufacturing tasks," International Journal on Interactive Design and Manufacturing, vol. 11, no. 2, pp. 139–153, May 2017, doi: 10.1007/s12008-015-0259-2.
[20] X. Wang, A. W. W. Yew, S. K. Ong, and A. Y. C. Nee, "Enhancing smart shop floor management with ubiquitous augmented reality," Int J Prod Res, vol. 58, no. 8, pp. 2352–2367, Apr. 2020, doi: 10.1080/00207543.2019.1629667
[21] V. Kuts et al., "DIGITAL TWIN: UNIVERSAL USER INTERFACE FOR ONLINE MANAGEMENT OF THE MANUFACTURING SYSTEM," in ASME International Mechanical Engineering Congress and Exposition, 2021. [Online]. Available: http://asmedigitalcollection.asme.org/IMECE/proceedings-pdf/IMECE2021/85567/V02BT02A003/6826815/v02bt02a003-imece2021-69092.pdf
[22] Y. Li, Q. Zhang, H. Xu, E. Lim, and J. Sun, "Virtual monitoring system for a robotic manufacturing station in intelligent manufacturing based on Unity 3D and ROS," Mater Today Proc, vol. 70, pp. 24–30, Jan. 2022, doi: 10.1016/J.MATPR.2022.08.486.
[23] K. Samir, A. Maffei, and M. A. Onori, "Real-Time asset tracking; a starting point for Digital Twin implementation in Manufacturing," Procedia CIRP, vol. 81, pp. 719–723, Jan. 2019, doi: 10.1016/J.PROCIR.2019.03.182.
[24] J. Osho et al., "Four Rs Framework for the development of a digital twin: The implementation of Representation with a FDM manufacturing machine," J Manuf Syst, vol. 63, pp. 370–380, Apr. 2022, doi: 10.1016/j.jmsy.2022.04.014.
[25] S. Katta, S. Ramatenki, and H. Sammeta, "Smart irrigation and crop security in agriculture using IoT," in AI, Edge and IoT-based Smart Agriculture, 2021. doi: 10.1016/B978-0-12-823694-9.00019-0.
[26] Linda Rosencrance, "What is Zigbee? | Definition from TechTarget," techtarget.
[27] Koen Kanters, "zigbee2mqtt ." Accessed: Nov. 12, 2023. [Online]. Available: https://www.zigbee2mqtt.io/
[28] HiveMQ, "MQTT - The Standard for IoT Messaging." Accessed: Nov. 12, 2023. [Online]. Available: https://www.hivemq.com/mqtt/