A Digital Twin Ecosystem for Additive Manufacturing using a Real-Time Development Platform

Minas Pantelidakis

Auburn University

Konstantinos Mykoniatis (✉ kmykoniatis@auburn.edu)

Auburn University <https://orcid.org/0000-0003-2269-9628>

Jia Liu

Auburn University

Gregory Harris

Auburn University

Research Article

Keywords: Additive Manufacturing, Digital Twin, Fused Deposition Modeling, Simulation, Unity 3D

Posted Date: January 27th, 2022

DOI: <https://doi.org/10.21203/rs.3.rs-1270408/v1>

License: This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Version of Record: A version of this preprint was published at The International Journal of Advanced Manufacturing Technology on April 13th, 2022. See the published version at <https://doi.org/10.1007/s00170-022-09164-6>.

Abstract

Additive manufacturing is often used in rapid prototyping and manufacturing, allowing the creation of lighter, more complex designs that are difficult or too expensive to build using traditional manufacturing methods. This work considers the implementation of a novel digital twin ecosystem that can be used for testing, process monitoring, and remote management of an additive manufacturing fused deposition modeling machine in a simulated virtual environment. The digital twin ecosystem is comprised of two approaches. One approach is data-driven by an open-source 3D printer web controller application that is used to capture its status and key parameters. The other approach is data-driven by externally mounted sensors to approximate the actual behavior of the 3D printer and achieve accurate synchronization between the physical and virtual 3D printers. We evaluate the sensor-data-driven approach against the web controller approach, which is considered to be ground truth. We achieve near-real-time synchronization between the physical machine and its digital counterpart and have validated the digital twin in terms of position, temperature, and run duration. Our digital twin ecosystem is cost-efficient, reliable, replicable, and hence can be utilized to provide legacy equipment with digital twin capabilities, collect historical data, and generate analytics.

I. Introduction

Digital Twin (DT) technology is one of the fastest-growing concepts of Industry 4.0. According to a 2018 Gartner study conducted, 75% of organizations implementing IoT already use DT or plan to do so within a year [1]. The DT concept refers to a digital informational construct that mirrors the behavior of an observable physical or cyber-physical system in a real-time simulation environment [2]. The digital information is a “twin” of the information contained in the physical system and can be connected to the observable system throughout its lifecycle [3]. According to a recent study, the DT market was valued at \$3.8 billion in 2019 and is expected to increase to \$35.8 billion by 2025 [4]. The research in DT technology has been a hot area in both academia and industry, and a variety of tools, frameworks, and architectures have been proposed and built for accelerating DT applications [5] [6] [7] [8] [9]. Moreover, with the emergence of DTs in different industries, efforts have been made to review and categorize DTs by domain expertise [11] [12] or algorithms [13] for better understanding and developing DTs. As our awareness of the benefits of a DT grows and the tools for constructing a DT improve, the definition of the term DT continues to evolve. In this paper, DT is described as a fully connected functional digital representation of an observable manufacturing asset that runs in real-time (or near real-time) mode and is driven by data and a set of attributes from the physical system [2].

The DT market is being driven by the growth of Industry 4.0 technologies (e.g., IoT [14], cloud computing, Additive Manufacturing, Modeling & Simulation, Extended Reality, Artificial Intelligence [15]) and the desire to reduce costs and shorten product development time [16]. Additionally, DTs can assist with process monitoring, virtual commissioning, predictive maintenance, lifecycle management, process diagnosis, and production responses [17] [18] [19] [20] [21] [22]. DTs also provide information interfaces between humans and machines to improve operators’ capability and productivity in manufacturing [23]

[24] [25]. Moreover, DTs can address the difficulties of productivity and quality control in industrial processes and allow organizations to act on a digital model for testing, and optimization [26]. Executing testing and optimization can accelerate prototyping, generate optimal decisions, eliminate instabilities caused to the real system, and assess a system's performance and efficacy.

At the same time, additive manufacturing (AM) is quickly growing, while also improving scalability and strategic options. Leaders in manufacturing are implementing AM across their organization and taking advantage of the increased flexibility this technology provides. AM is considered an emerging and disruptive Industry 4.0 technology that offers superior customization and flexibility in designing and manufacturing complex products, greatly improves the competitiveness of manufacturing, and reduces lead time and material waste [27]. Manufacturers that desire benefit from AM initiatives may discover that success requires cross-functional engagement and a willingness to move beyond the status quo in terms of product and process design, and in support of novel and innovative approaches. Utilizing a 3D real-time development platform for generating a DT for an AM process facilitates design and operation for highly customized and low-volume production. Fused Deposition Modeling (FDM) [28], also known as material extrusion or fused filament fabrication, is the most widely used AM process in prototyping, low-volume production for non-critical parts, training, and education. This can be attributed to its low risk, convenience, low cost, and friendly user interface [29] [30] [31] [32]. An FDM AM machine typically heats filament of thermoplastic material (e.g., acrylonitrile butadiene styrene), extrudes it through a nozzle onto a movable build platform, and builds a 3D product in a layer-wise manner based on digital 3D models [31]. This process creates layers of material that adhere to one another. Layer bonding is controlled through pressure, temperature control, and chemical agents [27].

Interconnectivity and interoperability of industrial machinery are improving [33]. The wide adoption of data semantic standards (e.g., MTConnect [34]) further accelerates these changes. However, not every manufacturer has the resources or incentive to upgrade to a contemporary solution and therefore continue to utilize their trusted and consistent legacy equipment. Small and medium-sized enterprises (SMEs) in particular, could greatly benefit from solutions that bring legacy equipment to the digital era. Such solutions should undoubtedly be reliable, straightforward, inexpensive, and require minimum technical expertise.

In this work, we present a Digital Twin Ecosystem (DTE) for an FDM AM machine, a Lulzbot Taz Workhorse 3D printer. The DTE is comprised of two different approaches. The first approach uses data collected by Octoprint[1], an open-source printer host software. The second approach is a replicable sensor-data-driven approach, that can be generalized and assist in the digitalization of legacy machines. We verify and validate our sensor-data-driven approach against the Octoprint-data-driven approach, which we consider as ground truth, as Octoprint has direct access to the firmware of the printer. Both approaches rely on state-of-the-art open-source and free-of-charge software. The sensor-driven approach utilizes widely available, reliable, and cost-efficient sensors, mounted externally, in a non-disruptive manner. It also enables communication and synchronization with other devices and sharing of process data via a local network. Both approaches result in a novel data-driven soft-real-time DTE of the 3D

printer, implemented with the Unity 3D real-time game development platform. The simulated virtual environment is configurable and allows remote access to machine information such as key process parameters, operational conditions, built-in sensor data, and user-added sensor signals. It also provides on-demand reports of tracked process variables and can collect and store historical data. We assess the real-time mimicking ability of the sensor-data-driven DT module in terms of extruder position and temperature, by comparing it to the Octoprint-data-driven DT module. We also evaluate the DTE in terms of 3D printer task duration and status. To the authors' knowledge, the developed DTE is at the frontier of FDM DT research. We developed a digital replication of an FDM physical system in a real-time development platform, synchronized the activities between the physical process and its digital twin, and established an ecosystem with bidirectional information flow. This work provides insight to integrate real-time development platform capabilities, data acquisition, analytics, and advanced simulation into developing DTs of various manufacturing processes and systems.

The rest of this paper is structured as follows: In section II we provide an overview of the current state of research regarding DTs for manufacturing and explore the different approaches to build a DTE. In addition, we highlight the benefits and limitations of existing DT approaches and define the scope of our research in terms of the published literature. In section III we describe the implementation of the DTE and give a detailed explanation of our methodology. In section IV we evaluate the developed DTE and report the results. Finally, in section V we discuss our conclusions and future research directions.

[1] <https://octoprint.org/>

II. Related Work

A. DIGITAL TWIN BACKGROUND

DTs have been heralded as a major milestone towards the transition to Industry 4.0, where advanced digitalization drives informed decision-making. Manufacturing benefits from such a system in multiple ways (e.g., flexibility in design and manufacturing, dynamic recalibration of equipment, insights into performance aspects of the production line and manufacturing process, energy footprint reduction, etc.). As such, new production models have emerged, where ubiquitous connectivity of components provides data flow from product conceptualization and design to production and distribution [35].

The definition of a DT within AM is still controversial [36], with some researchers defining it as a virtual system that mimics a physical system [37] [38] [39] [40], while others adopt a simulation-centric approach [41] [42] [43] [44]. The physical behavior of FDM involves heat transfer, materials phase changes, and machine mechanics that complicate the development of DTs [45] [46]. Moreover, existing works about FDM mainly focus on individual parts of the FDM process (e.g., the cooling down of the deposited filament, the flow in the nozzle, the bond between layers [47] [48] [49] [50]) instead of examining the process as a whole.

There is a consensus of thought regarding the importance of incorporating sensor data into DTs, as a means towards the acquisition of real or near-real-time data from physical processes [51] [52] [53] [54]. However, this concept is still in its development stage. “DTs” developed with this concept only have sensor data from the physical FDM machine, but lack the digital replication of the physical system in the virtual space; therefore, they are somewhat difficult to be differentiated from the traditional sensor-based process monitoring, such as [55] [31].

In legacy equipment, that does not inherently implement closed-loop control with networking capabilities, integration of digitalization poses a bigger challenge. Installing external sensors and networking solutions seems like a promising direction [56] [57]. Another body of research explores side-channels (e.g., acoustics, energy footprint, etc.), as a non-intrusive way to reveal information about the state and behavior of physical entities [58]. However, such an approach would be out of the scope of this work.

A DTE system typically consists of two major components. The first component collects data from physical assets, processes, optionally stores locally or on the cloud, and serves them continuously (e.g., streaming), or when queried, via a network. This component also has access to the machine, therefore can potentially intervene in the process. This component can be split into multiple sub-components, as described in the work of [56] and [59]. In [56], the authors use five different layers: i) embedded devices, ii) data acquisition and control, iii) local data storage, iv) networking, v) cloud storage. In [59], the authors use the 5C structure, which consists of five layers: i) connection, which describes data acquisition, ii) conversion, which refers to transforming raw data to a useful format, iii) cyber, which describes the generation of analytics, iv) cognition, meaning self-awareness and autonomous failure detection and v) configuration, referring to self-adaption to improve efficiency. These layers provide functionality that is beneficial to the operation of the DT. However, an effective DTE could omit certain layers of the referenced architectures, depending on system requirements, the scale of the data being processed, and its level of maturity. At its basic form, this component should communicate with a machine, process data, and make them available via a network. Therefore, in our work, we refer to this component as the data acquisition-processing-distribution component (APDC). The methodology section describes APDC in more detail.

The second major component of a DTE is a system that utilizes available data to conduct a real-time simulation of the physical entity, presents the algorithmically extracted insight to users, and could be also used for optimization studies and “what-if” analysis. This component is referred to as layer vi) emulation and simulation in the work of [56]. In this research, we use the term virtual representation component (VRC), further described in the Methodology section.

In the last decade, a substantial body of research utilized 3D development platforms, such as Unity 3D and Epic Unreal game engines to conduct simulation studies and develop DT applications for various case studies, ranging from industrial simulation studies [60] [61] [62] to robotics [63] [64], autonomous vehicles [65], smart cities [66], and remote surgery [67]. 3D real-time development platforms feature a certain set of characteristics (e.g., networking, rendering engine, physics engine, event system, animation

system, custom logic design) that makes them particularly suitable candidates for DTs. This work utilizes the Unity 3D development platform to implement the VRC of a DT for an FDM AM machine and open-source software and cost-efficient hardware for the APDC.

B. ADDITIVE MANUFACTURING BACKGROUND

Additive manufacturing (also known as rapid prototyping or 3D printing) is a manufacturing technique that involves building up layers of a material to form a solid product. Fig. 1 illustrates a 3D printing process. While there are several 3D printing technologies (e.g., Direct Metal Laser Sintering, Stereolithography, Selective Laser Sintering, Digital Light Process, etc.), here we concentrate on the process of Fused Deposition Modeling (FDM), which is one of the most popular and accessible AM processes. During the COVID-19 pandemic, hundreds of FDM machines at different locations in the U.S. provided nearly one million pieces of safe personal protective equipment for local medical providers through the 3D Printing COVID-19 Rapid Response Initiative. It has great potential to be deployed in distributed manufacturing and to act as a catalyst for ubiquitous manufacturing in the U.S.

The general approach remains the same across all these AM methods, whether the end product is a rapid prototype or a fully functional part. The initial stage in the AM process is to create a digital model. The digital model is most typically created using Computer-Aided Design (CAD) tools. Through 3D or structured light scanning, reverse engineering can also be employed to create a digital model. When designing for AM, there are various factors to consider such as geometry limits, structural requirements, orientation, etc.

An important step in the AM process that differentiates it from subtractive manufacturing methods is the requirement to convert a CAD model into a stereolithography file (STL). An STL file describes an object's surfaces using triangles (polygons). Once an STL file has been created the file is imported into a slicer program (e.g., Cura LulzBot Edition) and is converted to G-code. G-code is a programming language for numerical control (NC), which is used to control automated machine tools in computer-aided manufacturing (CAM) including 3D printers, and CNC machines. The G-code commands are tailored for the printer's firmware and dictate all aspects of the printing process (e.g., initial calibration, temperature levels, acceleration, feed rate, extrusion rate, movement coordinates, etc.) Therefore, a G-code file is the most essential component of the printing process and an indirect way to monitor the system. It is worth noting that slicing engines can also generate estimates of the amount of time needed to print the 3D object. These estimates often lack accuracy but can be further improved with plug-ins available for most printer host software solutions.

Since most FDM 3D printing machines are composed of many small, delicate components, proper maintenance and regular calibration are essential for producing accurate prints. After the print has started, most AM machines do not need monitoring. The machines operate autonomously, with problems occurring usually when the machine runs out of material or there is a software issue.

lil. Methodology

This section describes the methodology that we followed to develop a novel DTE using an open-source 3D printer web controller application and Unity 3D real-time development platform. We implement the DTE in a modular manner so that each component is a self-reliant entity while being able to work as part of the ecosystem. The two major components of the DTE architecture are i) the data acquisition-processing-distribution component (APDC), and ii) the virtual representation component (VRC). The orchestrated interweaving of the above components enables the DTE for continuous synchronization between the physical system and its digital counterpart. The use of only open-source and free-of-charge technologies was one of the main principles behind our design and implementation. We strongly believe that DT enabling technologies should be widely accessible to the public so that the Technology Readiness Level (TRL) of these tools further matures. A high-level diagram of the DTE is illustrated in Fig. 2. The following sections describe the different components of our methodology in further detail.

A. DATA ACQUISITION-PROCESSING-DISTRIBUTION COMPONENT (APDC)

The data acquisition-processing-distribution component APDC consists of a digital interface that hosts Octoprint and implements a Representational State Transfer (REST) Application Programming Interface (API) to serve external sensor data. The APDC collects, processes, and distributes data collected both from Octoprint and from external sensors. Data collected from Octoprint are used to build a stateful and detailed DT approach, as well as verify and validate our sensor-data-driven DT approach. The external sensors are used to build a sensor-data-driven DT approach, meant for legacy equipment, with no inherent interconnectivity capabilities. The following sections introduce the subcomponents of the APDC and describe them in detail. A high-level view of the APDC architecture is illustrated in Fig. 3. The left rectangle area illustrates the APDC component which consists of the Octoprint web controller and the Sensor data REST API. The right rectangle area represents the 3D printer module that contains Marlin firmware, and the externally mounted sensors (thermocouple and IR distance sensors). The relationship of data exchange between Octoprint and Marlin is bidirectional, while the user-attached sensors are utilized to monitor the system through the Sensor data REST API.

1) ACQUISITION-PROCESSING-DISTRIBUTION COMPONENT

To build the APDC, we use Raspberry Pi 3, because of its small size, low cost, and adequate computational power. The Raspberry Pi is a bare Printed Circuit Board (PCB) equipped with all the building components of a typical computer, including the most common I/O, CPU, GPU, RAM, SD card support, and Wi-Fi connectivity to name a few. The Operating System of our Raspberry Pi 3 is Raspberry Pi OS, previously known as Raspbian. We developed the sensor API using the Flask[2] framework. Flask is a Python-based microweb framework that excels at rapid prototyping and easy maintenance. The sensor data become available once the sensor API is queried from an external service.

2) OCTOPRINT WEB CONTROLLER

The APDC hosts Octoprint, an open-source extendable web interface used to monitor and control consumer 3D printing machines. Apart from a web interface, Octoprint also provides a REST API that can be queried via Hypertext Transfer Protocol (HTTP) requests. Octoprint provides information regarding i) job operations, ii) file operations, and iii) printer operations. Below we provide more details on how we combine and utilize parts of these three information categories to collect data that enable the Octoprint data-driven DT module.

- Job operations are used to receive information related to the currently loaded file, its name, its size, and upload date. Job operations are also used to receive and verify the estimated printing time (calculated by the slicing engine), the type of printing filament, the current G-code line being interpreted as a percentage of the total file lines (print completion percentage), the current G-code file line, printing time elapsed, as well as remaining printing time.
- File operations: are used to retrieve a full list of uploaded files, along with their metadata, as well as statistics related to successful and failed prints for each loaded file.
- Printer operations are used for verification and validation (V&V) purposes as well as to retrieve current printer states and historical temperature data for the extruder head and the printing bed.

3) EXTERNAL sENSORS

A critical factor in a 3D printing process is to appropriately set up and maintain temperature, both for the printer head and the printing bed. The sensor-data-driven DT approach utilized external sensors for the following reasons: i) to verify and validate the integrity of our architecture and its capability to be applied on legacy equipment, and ii) to provide an environment for rapid prototyping and reduced data collection downtime. Replacing an internal sensor can be time-consuming. Therefore, we configured the system in a plug-and-play fashion, which simplified replacing a sensor in our pipeline. To receive real-time temperature data, we used the Yocto-Thermocouple USB temperature sensor. The sensor provides two inputs for thermocouples. We utilized one input for the printer head, and the other for the printing bed. The sensor also features internal flash memory, which can be used to store and retrieve historical temperature data. The response time of the sensor is less than 30ms, with a temperature range between -75°C and 260°C, which makes it a good option for our sensor-data-driven DT. Another vital component of the printing process is the precise movement of the various parts involved. Our FDM AM machine has two different moving parts, namely the extruder head and the printing bed. To monitor the movement of the aforementioned parts, we installed and tested infrared (IR) distance sensors, namely Adafruit VL53L0X Time of Flight Micro-LIDAR. An isometric view of our 3D printer that shows the sensor positions is illustrated in Fig.4. The temperature sensors are shown with a triangle symbol and the IR distance sensors with a circle. Fig. 5 illustrates an image compilation of a real 3D printer that showcases the different positions of the externally mounted sensors.

B. VIRTUAL REPRESENTATION COMPONENT (VRC)

Unity is a popular 3D real-time development platform, used in various applications ranging from virtual humans[68] and automotive showcases to training agents with Deep Reinforcement Learning [69]. It provides a realistic physics engine, a robust event system, animation system, advanced lighting settings, and the ability to implement custom logic using a programming language (e.g., C#). This platform was utilized to implement the virtual representation component of the DTE. A 3D model of the printer was imported into the 3D development platform. First, we fixed orientation issues, changed the pivot point, and grouped the related parts of the printer. For example, all parts of the extruder head were grouped under a new game object called Extruder. Game objects act as containers for components, which accomplish the real functionality. For example, the “Extruder” object is a self-contained modeling construct that allows defining that construct's characteristics, data, behavior, user interface, and animation. Creating a new game object also helps reorient the model, since the new object's pivot point can be altered without visual modifications to the original model's meshes. Unity's animation system was not only used for visualization purposes but also for V&V of different 3D printer processes. The Octoprint-data-driven approach is stateful, meaning that it keeps track of the current state of the printer, using data directly from its firmware. The sensor-data-driven approach is stateless and utilizes our external sensors to drive the DT, without keeping track of the current state of the printer. The different approaches are explained in subsections B (1) and B (2).

1) OCTOPRINT DATA-DRIVEN Digital Twin

The VRC accesses Octoprint (hosted in the APDC) using an open-source C# wrapper class of the Octoprint REST API. Any class that "wraps" or "encapsulates" the functionality of another class or component is referred to as a wrapper class. This is useful because it provides a level of abstraction from the underlying class or component's implementation. The C# project was compiled into a single Dynamic Link Library (DLL) file using MS Visual Studio. The DLL file is a library that contains code and data that can be used by many programs at the same time. Thus, we imported the DLL file into the Unity project's assets folder to enable access to the Octoprint REST API using high-level C# routines. To achieve synchronization between the LulzBot Workhorse and its stateful DT, a combination of event-based and periodic logic execution is used. Octoprint is queried multiple times per second to determine the printer state and the values of various variables of interest, such as current G-code command, extruder temperature, bed temperature, and estimated time of print completion. The G-code is loaded in a buffer, and the currently executed command is identified based on the feedback from Octoprint. The concept is that the stateful DT of the printer is engaged in some specific kind of action at any given time. The available actions depend on the type of actions the real 3D printer performs, such as idling, printing, translating on XYZ, etc. These actions are referred to as states, in the sense that the DT model is in a “state” where it is printing or idling, just to name a few. In general, the DT will have restrictions on proceeding to the next state rather than being able to switch immediately from a state to another. When a defined event occurs and certain conditions are met, a transition between two states is triggered. For example, an animation can only occur when the DT is printing and not when it is in the idle state. Thus, it should never switch straight from the idle state to the animation state. The options for the next state that our DT model can enter from its current state are referred to as state transitions. Taken into consideration,

the set of states, the transitions, and the variables of interest involved with the current state, comprise a finite state machine (FSM). We used a unified modeling language (UML), which is a general-purpose, developmental modeling language that is designed to provide a standard way to depict a system's architecture. Fig. 6 illustrates a UML state chart that is used to model the dynamic nature of the different DT states of the FSM. The most important states are the following: "Connected", which is when a successful connection is established with the printer; "Operational", meaning that the printer is ready to execute the printing process; "Printing", meaning that the printer is currently in the printing process; "Paused", meaning that the print has paused for some reason. The most critical state is the "Printing" state, where the DT remains for most of the print. When a new G-code command is received, the DT enters the "Animating" state. After the physical coordinates are mapped to virtual space, the movement animation curves per axis are generated in real-time before the animation is triggered. Animation curves provide the option to model movement more accurately (e.g., linear, constant, and ease-in-out movement). To simulate realistic animations, we identify the coordinate limits of the physical machine for each axis and translate them to limits within virtual space. Both the physical and the virtual position limits are displayed in Table 1 under columns "physical limit" and "virtual limit" respectively. The formula that converts physical to virtual coordinates is shown in (1). It receives a coordinate in physical space ($coord_p$), and, given physical and virtual coordinate minimum and maximum values ($min_p, max_p, min_v, max_v$), maps the physical coordinate to virtual space ($coord_v$). This equation is common for all three axes (X, Y, and Z). Physical and virtual limits are denoted by subscripts p and v respectively. Fig 7. Illustrates the DT of the AM FDM, as it is displayed in the Unity 3D development platform.

Table 1
Position limits for the physical machine and the unity DT.

<i>Type</i>	<i>physical limit</i>	<i>sensor val.</i>	<i>virtual limit</i>
X min	-49.9	386	-1.15
X max	295	0	1.15
Y min	-16.64	-20.0	-1.17
Y max	308	301	0.88
Z min	0	351	-1.24
Z max	299	1	1

$$coord_v = min_v + \frac{(coord_p - min_p)(max_v - min_v)}{max_p - min_p} \quad (1)$$

2) SENSOR DATA-DRiven Digital Twin

The sensor-data-driven DT utilizes signals from the externally-mounted sensors. The VCR accesses real-time sensor data using our sensor REST API (hosted in the APDC). External sensors can be mounted in

several ways depending on various criteria, such as sensor specifications and durability, a possible collision with moving parts, and the availability of mounting space. The external sensor mounting affects the position sensor values. Therefore, the relationship between position sensor values and coordinates is not fixed but varies depending on the installation setting.

To estimate the equivalent physical coordinates value given sensor readings we devised approximation functions. These functions are calculated by fitting the curves generated by plotting the empirical data of the physical coordinate values over sensor values for each axis. The approximation functions for all axes appear to be linear and are illustrated in (2) to (4).

$$X_{\text{physical}} = -0.8935 * X_{\text{sensor}} + 295 \quad (2)$$

$$Y_{\text{physical}} = 1.0113 * Y_{\text{sensor}} + 3.587 \quad (3)$$

$$Z_{\text{physical}} = -0.8544 * Z_{\text{sensor}} + 299.85 \quad (4)$$

where X_{physical} , Y_{physical} , and Z_{physical} are the physical coordinates for X-axis, Y-axis, and Z-axis respectively, and X_{sensor} , Y_{sensor} , and Z_{sensor} are the sensor values for each axis.

A raw sensor value is given as input to the corresponding approximation function (i.e., (2) to (4)), and the result is a physical coordinate. The physical coordinate is then mapped to virtual space using (1). Consequently, the DT adjusts its moving components, according to the new virtual coordinate. Regarding temperature, no manipulation is required. The temperature value in the DT comes from the temperature sensor reading. The sensor-data-driven DT does not utilize an FSM approach, but is stateless, meaning that the DT always monitors the position and temperature of the 3D printer, without identifying the current state of the machine.

[2] <https://flask.palletsprojects.com/en/2.0.x/>

IV. Evaluation

In this section, we discuss our findings and evaluate the sensor-data-driven DT in terms of extruder position and temperature, by comparing it to the Octoprint-data-driven DT. We also evaluate the Octoprint-data-driven DT in terms of print duration against manually collected measurements. We assume that the true machine values for position and temperature are reported via Octoprint as it has access to the printer firmware containing the instructions that tell the device how to interface with other computer hardware. Our experiments follow the methodology presented in Fig. 8. To ensure the correctness of both of the DT approaches we observed the behavior of the DTE through animation over time and compared it with the physical 3D printer. The sensor-driven DT properties matched the actual 3D printer properties during the simulation runtime. Therefore, we concluded that twinning between the virtual and the physical asset has been successfully achieved and the DTE is valid. The successful twinning and validation of the DTE can be observed in our online demo.

A. POSITION ASSESSMENT

To evaluate the position-mimicking capability of the sensor-data-driven DT, we placed the extruder head and bed on $(X, Y, Z) = (50, 53, 19)$ in physical space. Values are measured in millimeters. We then utilized a sensor API to compare the data from the sensor-data-driven DT against Octoprint data. We collected sensor readings and physical position approximation values in 30 replications. Table 2 presents the mean and standard deviation for sensor values. Table 3 shows the mean and standard deviation for physical coordinate approximations. For the X coordinate, which was initially set to 50, we get a mean of 47.31. For the Y coordinate, which was initially set to 53, we get a mean of 52.94. For the Z coordinate, which was initially set to 19, we get a mean of 22.2. After 30 replications the DT is on average 2.69 mm off on the X-axis, 0.06 mm off on the Y-axis, and 3.2 mm off on the Z-axis. These differences were expected due to the limitations of the currently installed IR distance sensors, which usually operate over a range of 50 to 1200 mm and can generally achieve between 3% and 12% ranging accuracy depending on the operational conditions and environmental factors (i.e., measurement distance, light conditions, angle, etc.). To verify and validate the DT position process and assess the correctness of our code and whether the DT behaves reasonably compared to the real 3D-printer positions we applied an Agent-Based Modeling graphical representation technique [70]. Fig. 9 presents a graphical representation of the position approximation experiments. Blue circles correspond to the approximated physical coordinate. The solid line is the mean of the measurements, and the dashed line is the true physical coordinate (Octoprint coordinate). As it is observed from the face validity[71] of the graphical representation we conclude that the DT positioning behaves sufficiently accurately.

Table 2
Mean and standard deviation of sensor readings for constant physical coordinates (X=50, Y=53, Z=19).

Sensor X-axis		Sensor Y-axis		Sensor Z-axis	
Mean	SD	Mean	SD	Mean	SD
277.2	2.565	48.8	1.827	325	3.5

Table 3
Mean and standard deviation of physical coordinate approximations for constant physical coordinates (X=50, Y=53, Z=19).

X_{physical} approx.		Y_{physical} approx.		Z_{physical} approx.	
Mean	SD	Mean	SD	Mean	SD
47.31	2.292	52.94	1.848	22.2	2.993

B. TEMPERATURE ASSESSMENT

To validate the temperature-mimicking capability of the sensor-data-driven DT, we gradually increase the temperature and observed the embedded extruder head sensor readings at the steady-state. Polylactic

acid (PLA) is one of the most common materials used in 3D printing and our filament material of choice. PLA works well when printing between 180° C to 230° C, which is typically the printer’s operation temperature. We performed two experiments, one at 180° C and another one at 230° C, with 30 replications for each temperature level. The results are illustrated in Table 4. The manufacturer specifies a $\pm 2.2^{\circ}$ C error limit for the yocto-thermocouple sensor. The observed sensor values are within machine specifications. Fig. 10 graphically illustrates the experiment replications. Blue circles correspond to the temperature values acquired by the external sensors. The solid line corresponds to the mean temperature across the experiments, while the dashed line is the temperature reported by Octoprint. As it is observed from the face validity of the graphical representation and the mean and standard deviation of Table 4 we conclude that the DT temperature behaves sufficiently accurately.

Table 4
Mean and standard deviation of the temperature sensor readings at two different temperature levels.

180 ° C		230° C	
Mean	SD	Mean	SD
181.82	0.191	230.89	0.092

C. STATE DURATION ASSESSMENT

To confirm that the state durations are accurately reflected in the Octoprint-data-driven DT we wrote a custom G-code file that describes a print with no temperature adjustments and executed 30 print repetitions. We manually measure the duration of the print by observing the printer and comparing it against the print duration generated in the Octoprint-data-driven DT. Fig. 11 shows side-by-side plots, one for each measurement method. Blue circles correspond to the measured values across the replications, and the solid line represents the mean of the measurements. Results, in tabular format, are illustrated in Table 5. The manually measured duration for the print has a mean of 25.7133 seconds and a standard deviation of 0.157. The Octoprint-data-driven DT duration for the print has a mean of 25.9865 seconds and a standard deviation of 0.219. Thus, we conclude that the print duration mimicking capability of the DT is sufficiently accurate.

Table 5
Print duration mean and standard deviation. Measurements are taken manually for the machine, and automatically from inside the Unity DT.

Print duration (manual)		Print duration (dt)	
Mean	SD	Mean	SD
25.71333 (sec)	0.1571	25.98651 (sec)	0.219

D. EXTRUDER SPEED ASSESSMENT

The current DTE setup allows the movement speed to be directly estimated from position feedback. If we know two consecutive positions, we can deduce speed by dividing the distance traveled by the elapsed time. However, if feedback is not available, speed estimation becomes a much harder task. For speed to be successfully estimated without feedback from the machine, one needs to know the feed rate setting of the printer, the acceleration setting, and the translation distance. As such, experiments were designed to determine and evaluate the effect of the aforementioned variables on the average speed. The results of our experiments are discussed below.

Initially, the acceleration was set to 500 mm/s². Performing 100 mm translations with a feed rate of 3000 the extruder can reach an average speed of 118.34 mm/s. However, when performing 250 mm translations with the same settings, it can reach an average speed of 171.23 mm/s. Given the above, we determine that a higher speed can be achieved when the extruder head travels for a longer distance.

When the translation distance is not large enough, increasing the feed rate does not increase the average speed. The maximum speed of the printer is 300 mm/s, however, this is not always achievable. The extruder must accelerate for a certain amount of time to reach the target speed. For example, reaching a speed of 300 mm/s with an acceleration of 500mm/s² would take 0.6 seconds. The distance covered within this time is 90mm. It is past the 90 mm point that the extruder head will travel at the maximum speed. The extruder also must decelerate before stopping. Deceleration requires some distance for which the printer does not operate at maximum speed. The longer the translation, the less acceleration, and deceleration affect the average speed. From the above, we determine that the average speed within a translation has a non-linear relationship with the feed rate for small translation distances and low acceleration settings. To determine how acceleration affects the relationship between feed rate and extruder speed, we set the acceleration to the maximum allowed value of 9000 mm/s² and performed the same experiment with the new acceleration setting. The results are presented in Fig. 12. The solid line corresponds to the experiment with acceleration set to 500mm/s² and distance to 100 mm. The dashed line corresponds to acceleration set to 500 mm/s² and distance to 250 mm. The dotted line corresponds to acceleration equal to 9000 mm/s² and distance 100 mm. The dashed-dotted line corresponds to acceleration equal to 9000 mm/s² and translation distance 250 mm. We can observe that the relationship between feed rate and average speed approximates a linear curve as translation distance and acceleration increase, with acceleration having the most significant effect.

V. Conclusions & Future Work

In this paper, we described the development of a novel digital twin ecosystem that can be used for testing, process monitoring, and remote management of an AM FDM machine in a simulated virtual environment. The developed DTE can capture the overall operation and performance of the FDM machine and therefore could be used for in-process analysis and optimization. To develop a realistic DTE that works independently of machine feedback, one needs to study, identify, and model the internal physics that govern the operation of the machine. In our work, we applied a DTE architecture, composed of two key

components: the APDC and the VRC. Using the APDC and VRC we were able to coordinate and interconnect the components that comprise the DTE and achieve continuous synchronization between the physical asset and its digital system. We verified and validated the DTE by assessing position, temperature, and duration, comparing sensor data with internal machine data. In the future, we plan to examine more variables, such as the relationship between feed rate, acceleration, and translation distance, as well as the temperature rise and drop rate. In addition, we plan to explore environmental factors (e.g., atmospheric temperature, humidity, light conditions), as well as the relationship between different filaments, extrusion speeds, and print quality. Our overall future goal is to use this DTE to predict the product quality even before it is printed, reduce the time and cost in experimental testing of optimal printing conditions, and assist in the optimization of part design and tool path.

IR distance sensors have a higher accuracy the closer they are placed to the measured object. Furthermore, according to the manufacturer, ambient lighting and the material of the measured object affect the quality of the readings. Best readings are obtained from shiny objects, in good lighting conditions. The environmental conditions during the experiments were not fully controlled. More specifically, printer surfaces and room lighting have not been adjusted to optimize sensor reading accuracy. Such modifications will be implemented and studied in future research endeavors.

Another future research direction is to augment the developed DT with Extended Reality (XR) capabilities. The transformation is straightforward since Unity 3D real-time development platform offers extensive XR support. An XR version of the 3D printer's DT has the potential to serve remote educational purposes. Schools, universities, and SMEs could benefit from such a system. For example, a remote Augmented Reality DT can be utilized when the environment is not suitable for machinery, or there are financial constraints. Furthermore, the interconnectivity capabilities of modern head-mounted displays (HMDs) allow for collaborative work between individuals and teams.

Declarations

Funding This work was partially supported by the National Institute of Standards and Technology, Grant 70NANB18H267, and the Interdisciplinary Center of Advanced Manufacturing Systems (ICAMS) funded by the U.S. Army, Grant W52p1j-20-9-3045.

Data availability Not applicable

Code availability Not applicable

Ethics approval Not applicable

Consent to participate Not applicable

Consent for publication Not applicable

Conflict of interest The authors declare no competing interests.

Acknowledgment The authors would like to acknowledge John Osho and Anna Hyre for assisting with the hardware setup and programming.

References

- [1] K. Costello and G. Omale, "gartner.com," Stamford, Conn., February 2019. [Online]. Available: <https://www.gartner.com/en/newsroom/press-releases/2019-02-20-gartner-survey-reveals-digital-twins-are-entering-mai>. [Accessed October 2021].
- [2] K. Mykoniatis and G. A. Harris, "A digital twin emulator of a modular production system using a data-driven hybrid modeling and simulation approach," *Journal of Intelligent Manufacturing*, pp. 1-13, 2021.
- [3] M. Grieves and J. Vickers, "Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems," in *Transdisciplinary perspectives on complex systems*, Springer, 2017, pp. 85-113.
- [4] MarketsandMarkets, "Digital Twin Market by Technology, Type (Product, Process, and System), Application (predictive maintenance, and others), Industry (Aerospace & Defense, Automotive & Transportation, Healthcare, and others), and Geography - Global Forecast to 2026," 2020.
- [5] N. Tiwari, M. J. Kaur and V. P. Mishra, "Architecture and use cases of digital twins towards smart manufacturing," *International Journal of Public Sector Performance Management*, vol. 8, no. 3, pp. 264-272, 2021.
- [6] D. Lehner, J. Pfeiffer, E. F. Tinsel, M. M. Strljic, S. Sint, M. Vierhauser, A. Wortmann and M. Wimmer, "Digital Twin Platforms: Requirements, Capabilities, and Future Prospects," *IEEE Software*, vol. 01, pp. 0-0, 2021.
- [7] J. Conde, A. Munoz-Arcenales, A. Alonso, S. Lopez-Pernas and J. Salvachua, "Modeling Digital Twin Data and Architecture: A Building Guide with FIWARE as Enabling Technology," *IEEE Internet Computing*, vol. 01, pp. 1-1, 2021.
- [8] D. Lehner, S. Sint, M. Vierhauser, W. Narzt and M. Wimmer, "AML4DT: a model-driven framework for developing and maintaining Digital Twins with automationML," in *2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Vasteras, Sweden, 2021.
- [9] R. Kostromin, A. Feoktistov and M. Voskoboinikov, *Service-oriented tools for automating digital twin development*, 2021.
- [10] A. Hyre, G. Harris, J. Osho, M. Pantelidakis, K. Mykoniatis and J. Liu, "Digital twins: Representation, Replication, Reality, and Relational (4Rs)," *Manufacturing Letters*, vol. 31, no. January, pp. 20-23, 2022.
- [11] F. Tao, H. Zhang, A. Liu and A. Y. Nee, "Digital twin in industry: State-of-the-art," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 2405-2415, 2018.

- [12] J. F. Uhlenkamp, K. Hribernik, S. Wellsandt and K. D. Thoben, "Digital Twin Applications: A first systemization of their dimensions," in *2019 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC)*, Valbonne Sophia-Antipolis, France, 2019.
- [13] G. Pronost, F. Mayer, B. Marche, M. Camargo and L. Dupont, "Towards a Framework for the Classification of Digital Twins and their Applications," in *2021 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC)*, Cardiff, United Kingdom, 2021.
- [14] K. Mykoniatis, "A real-time condition monitoring and maintenance management system for low voltage industrial motors using internet-of-things," *Procedia Manufacturing*, vol. 42, pp. 450-456, 2020.
- [15] M. Liu, S. Fang, H. Dong and C. Xu, "Review of digital twin about concepts, technologies, and industrial applications," *Journal of Manufacturing Systems*, vol. 58, pp. 346-361, 2021.
- [16] T. D. West and M. Blackburn, "Is digital thread/digital twin affordable? A systemic assessment of the cost of DoD's latest manhattan project," *Procedia computer science*, vol. 114, pp. 47-56, 2017.
- [17] T. H.-J. Uhlemann, C. Schock, C. Lehmann, S. Freiberger and R. Steinhilper, "The digital twin: demonstrating the potential of real time data acquisition in production systems," *Procedia Manufacturing*, vol. 9, pp. 113-120, 2017.
- [18] Z. Liu, N. Meyendorf and N. Mrad, "The role of data fusion in predictive maintenance using digital twin," in *AIP Conference Proceedings*, 2018.
- [19] Q. Qi and F. Tao, "Digital twin and big data towards smart manufacturing and industry 4.0: 360 degree comparison," *Ieee Access*, vol. 6, pp. 3585-3593, 2018.
- [20] T. H.-J. Uhlemann, C. Lehmann and R. Steinhilper, "The digital twin: Realizing the cyber-physical production system for industry 4.0," *Procedia Cirp*, vol. 61, pp. 335-340, 2017.
- [21] P. Aivaliotis, K. Georgoulas and K. Alexopoulos, "Using digital twin for maintenance applications in manufacturing: State of the Art and Gap analysis," in *2019 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC)*, Valbonne Sophia-Antipolis, France, 2019.
- [22] Y. Xu, Y. Sun, X. Liu and Y. Zheng, "A digital-twin-assisted fault diagnosis using deep transfer learning," *IEEE Access*, vol. 7, pp. 19990 - 19999, 2019.
- [23] A. Löcklin, T. Jung, N. Jazdi, T. Ruppert and M. Weyrich, "Architecture of a Human-Digital Twin as Common Interface for Operator 4.0 Applications," *Procedia CIRP*, vol. 104, pp. 458-463, 2021.
- [24] V. Holzwarth, C. Hirt, J. Gisler and A. Kunz, *Virtual Reality Extension for Digital Twins of Machine Tools*, 2021.

- [25] Y. Wang, X. Wang, F. Tao and A. Liu, "Digital twin-driven complexity management in intelligent manufacturing," *Digital Twin*, vol. 1, p. 9, 2021.
- [26] H. Zhang, Q. Liu, X. Chen, D. Zhang and J. Leng, "A digital twin-based approach for designing and multi-objective optimization of hollow glass production line," *IEEE Access*, vol. 5, pp. 26901-26911, 2017.
- [27] K. V. Wong and A. Hernandez, "A review of additive manufacturing," *International scholarly research notices*, vol. 2012, 2012.
- [28] S. C. Daminabo, S. Goel, S. A. Grammatikos, H. Y. Nezhad and V. K. Thakur, "Fused deposition modeling-based additive manufacturing (3D printing): techniques for polymer material systems," *Materials today chemistry*, vol. 16, p. 100248, 2020.
- [29] H. Guo, R. Lv and S. Bai, "Recent advances on 3D printing graphene-based composites," *Nano Materials Science*, vol. 1, pp. 101-115, 2019.
- [30] P. Parandoush and D. Lin, "A review on additive manufacturing of polymer-fiber composites," *Composite Structures*, vol. 182, pp. 36-53, 2017.
- [31] P. K. Rao, J. P. Liu, D. Roberson, Z. J. Kong and C. Williams, "Online real-time quality monitoring in additive manufacturing processes using heterogeneous sensors," *Journal of Manufacturing Science and Engineering*, vol. 137, 2015.
- [32] P. Rao, J. Liu, D. Roberson and Z. Kong, "Sensor-Based Online Process Fault Detection in Additive Manufacturing," in *ASME 2015 International Manufacturing Science and Engineering Conference*, Charlotte, North Carolina, USA, 2015.
- [33] T. R. Kurfess, C. Saldana, K. Saleeby and M. P. Dezfouli, "A Review of Modern Communication Technologies for Digital Manufacturing Processes in Industry 4.0," *Journal of Manufacturing Science and Engineering*, vol. 142, p. 110815, 2020.
- [34] A. Vijayaraghavan, W. Sobel, A. Fox, D. Dornfeld and P. Warndorf, *Improving machine tool interoperability using standardized interface protocols: MT connect*, 2008.
- [35] R. Rosen, G. Von Wichert, G. Lo and K. D. Bettenhausen, "About the importance of autonomy and digital twins for the future of manufacturing," *IFAC-PapersOnLine*, vol. 48, pp. 567-572, 2015.
- [36] K. Bartsch, A. Pettke, A. Hubert, J. Lakamper and F. Lange, "On the digital twin application and the role of artificial intelligence in additive manufacturing: a systematic review," *Journal of Physics: Materials*, 2021.
- [37] D. Montoya-Zapata, J. M. Rodriguez, A. Moreno, J. Posada and O. Ruiz-Salguero, "2D linear finite element simulation of laser metal heating for digital twins," *International Journal for Simulation and Multidisciplinary Design Optimization*, vol. 12, p. 11, 2021.

- [38] A. Shrivastava, S. Mukherjee and S. S. Chakraborty, "Addressing the challenges in remanufacturing by laser-based material deposition techniques," *Optics & Laser Technology*, vol. 144, p. 107404, 2021.
- [39] P. Stavropoulos, A. Papacharalampopoulos, C. K. Michail and G. Chryssolouris, "Robust Additive Manufacturing Performance through a Control Oriented Digital Twin," *Metals*, vol. 11, p. 708, 2021.
- [40] D. Gunasegaram, A. Murphy, M. Matthews and T. DebRoy, "The case for digital twins in metal additive manufacturing," *Journal of Physics: Materials*, vol. 4, p. 040401, 2021.
- [41] T. Debroy, W. Zhang, J. Turner and S. S. Babu, "Building digital twins of 3D printing machines," *Scripta Materialia*, vol. 135, pp. 119-124, 2017.
- [42] G. Knapp, T. Mukherjee, J. Zuback, H. Wei, T. Palmer, A. De and T. DebRoy, "Building blocks for a digital twin of additive manufacturing," *Acta Materialia*, vol. 135, pp. 390-399, 2017.
- [43] I. Sieber, R. Thelen and U. Gengenbach, "Enhancement of High-Resolution 3D Inkjet-Printing of Optical Freeform Surfaces Using Digital Twins," *Micromachines*, vol. 12, p. 35, 2021.
- [44] Z. Xi, "Model Predictive Control of Melt Pool Size for the Laser Powder Bed Fusion Process Under Process Uncertainty," *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part B: Mechanical Engineering*, vol. 8, p. 011103, 2021.
- [45] M. A. Azad, D. Olawuni, G. Kimbell, A. Z. M. Badruddoza, M. Hossain and T. Sultana, "Polymers for extrusion-based 3D printing of pharmaceuticals: A holistic materials–process perspective," *Pharmaceutics*, vol. 12, no. 2, p. 124, 2020.
- [46] B. Shaqour, M. Abuabiah, S. Abdel-Fattah, A. Juaidi, R. Abdallah, W. Abuzaina, M. Qarout, B. Verleije and P. Cos, "Gaining a better understanding of the extrusion process in fused filament fabrication 3D printing: a review," *The International Journal of Advanced Manufacturing Technology*, pp. 1-13, 2021.
- [47] H. Xia, J. Lu, S. Dabiri and G. Tryggvason, "Fully resolved numerical simulations of fused deposition modeling. Part I: fluid flow," *Rapid Prototyping Journal*, vol. 24, no. 2, pp. 463-476, 2018.
- [48] H. Xia, J. Lu and G. Tryggvason, "Fully resolved numerical simulations of fused deposition modeling. Part II – solidification, residual stresses and modeling of the nozzle," *Rapid Prototyping Journal*, vol. 24, no. 6, pp. 973-987, 2018.
- [49] B. J. Courter, V. Savane and C. J. Hansen, *Finite Element Simulation of the Fused Deposition Modelling Process*, 2017.
- [50] M. P. Serdeczny, R. Comminal, D. B. Pedersen and J. Spangenberg, "Numerical simulations of the mesostructure formation in material extrusion additive manufacturing," *Additive Manufacturing*, vol. 28, pp. 419-429, 2019.

- [51] A. Gaikwad, R. Yavari, M. Montazeri, K. Cole, L. Bian and P. Rao, "Toward the digital twin of additive manufacturing: Integrating thermal simulations, sensing, and analytics to detect process faults," *IJSE Transactions*, vol. 52, pp. 1204-1217, 2020.
- [52] R. M. Scheffel, A. A. Frohlich and M. Silvestri, "Automated fault detection for additive manufacturing using vibration sensors," *International Journal of Computer Integrated Manufacturing*, vol. 34, pp. 500-514, 2021.
- [53] H. Wei, T. Mukherjee, W. Zhang, J. Zuback, G. Knapp, A. De and T. DebRoy, "Mechanistic models for additive manufacturing of metallic components," *Progress in Materials Science*, vol. 116, p. 100703, 2021.
- [54] R. Zou, X. Liang, Q. Chen, M. Wang, M. A. Zaghloul, H. Lan, M. P. Buric, P. R. Ohodnicki, B. Chorpene and A. C. To, "A Digital Twin Approach to Study Additive Manufacturing Processing Using Embedded Optical Fiber Sensors and Numerical Modeling," *Journal of Lightwave Technology*, vol. 38, pp. 6402-6411, 2020.
- [55] M. Alizadeh, M. N. Esfahani, W. Tian and J. Ma, "Data-driven energy efficiency and part geometric accuracy modeling and optimization of green fused filament fabrication processes," *Journal of Mechanical Design*, vol. 142, no. 4, 2020.
- [56] A. Redelinghuys, A. H. Basson and K. Kruger, "A six-layer architecture for the digital twin: a manufacturing case study implementation," *Journal of Intelligent Manufacturing*, pp. 1-20, 2019.
- [57] N. Karanjkar, A. Joglekar, S. Mohanty, V. Prabhu, D. Raghunath and R. Sundaresan, "Digital twin for energy optimization in an SMT-PCB assembly line," in *2018 IEEE International Conference on Internet of Things and Intelligence System (IOTAIS)*, 2018, pp. 85-89.
- [58] S. R. Chhetri and M. A. Al Faruque, "IoT-enabled living digital twin modeling," in *Data-Driven Modeling of Cyber-Physical Systems using Side-Channel Analysis*, 2020, pp. 155-182.
- [59] B. Bagheri and J. Lee, "Big future for cyber-physical manufacturing systems," *Design world*, vol. 23, 2015.
- [60] V. Kuts, T. Otto, T. Tahemaa and Y. Bondarenko, "Digital twin based synchronised control and simulation of the industrial robotic cell using virtual reality," *Journal of Machine Engineering*, vol. 19, 2019.
- [61] M. Lacomblez, B. Jeanne, V. Havard and D. Baudry, "Co-simulation architecture between a digital twin and a virtual reality environment in an industrial context," in *Advances in Manufacturing Technology XXXII: Proceedings of the 16th International Conference on Manufacturing Research, incorporating the 33rd National Conference on Manufacturing Research, September 11–13, 2018, University of Skovde, Sweden*, 2018.

- [62] N. Meier, R. Muller-Polyzou, L. Brach and A. Georgiadis, "Digital twin support for laser-based assembly assistance," *Procedia CIRP*, vol. 99, pp. 460-465, 2021.
- [63] M. Matulis and C. Harvey, "A robot arm digital twin utilising reinforcement learning," *Computers & Graphics*, vol. 95, pp. 106-114, 2021.
- [64] T. Hassel and O. Hofmann, "Reinforcement Learning of Robot Behavior based on a Digital Twin.," in *ICPRAM*, 2020, pp. 381-386.
- [65] Z. Wang, K. Han and P. Tiwari, "Digital Twin Simulation of Connected and Automated Vehicles with the Unity Game Engine," in *2021 IEEE 1st International Conference on Digital Twins and Parallel Intelligence (DTPI)*, 2021, pp. 1-4.
- [66] G. White, A. Zink, L. Codeca and S. Clarke, "A digital twin smart city for citizen feedback," *Cities*, vol. 110, p. 103064, 2021.
- [67] H. Laaki, Y. Miche and K. Tammi, "Prototyping a digital twin for real time remote control over mobile networks: Application of remote surgery," *IEEE Access*, vol. 7, pp. 20325-20336, 2019.
- [68] K. Mykoniatis, A. Angelopoulou, M. D. Proctor and W. Karwowski, "Virtual humans for interpersonal and communication skills' training in crime investigations.," in *International Conference on Virtual, Augmented and Mixed Reality*, 2014.
- [69] L. Sun, J. Zhai and W. Qin., "Crowd navigation in an unknown and dynamic environment based on deep reinforcement learning," *IEEE Access*, vol. 7 , pp. 109544-10955, 2019.
- [70] X. Xiang, R. Kennedy, G. Madey and S. Cabaniss, "Verification and validation of agent-based scientific simulation models," in *In Agent-directed simulation conference*, 2005.
- [71] T. H. Naylor and J. M. Finger, "Verification of computer simulation models," *Management science*, vol. 14, no. 2, pp. B-92, 1967.

Figures

Figure 1

Image captured during the 3D printing process. The 3D printer extrudes layers of filament to produce a solid object.

Figure 2

High level diagram of the DTE. The two major components are i) the virtual representation component (VRC), and ii) the data acquisition-processing-distribution component (APDC).

Figure 3

The data acquisition-processing-distribution component (APDC) of the DTE. This component collects data directly from the printer and the externally mounted sensors.

Figure 4

Taz Workhorse 3D printer model that highlights external sensor installation positions.

Figure 5

a) Printer front view, b) Printer back view with y distance IR sensor, c) Printer top view, d) Z distance sensor from below, e) Thermocouple sensor attached to the nozzle of the printer head, f) x distance IR sensor.

Figure 6

UML state chart that describes the logic executed in the Octoprint-data-driven Unity DT.

Figure 7

The 3D printer and its stateful DT in Unity engine.

Figure 8

Methodology used for DT verification and validation. The DT is evaluated in terms of: i) position, ii) temperature, and iii) print duration.

Figure 9

Physical coordinate approximations from sensor values.

Figure 10

Temperature readings at two different levels.

Figure 11

Print duration measured manually (left) and in the DT (right).

Figure 12

Impact of acceleration, translation distance and feed rate on the average speed of the extruder.