School of Engineering and Digital Sciences
Department of Mechanical & Aerospace Engineering
Nazarbayev University

Sensor-Based Digital Twin for Fused Deposition Modeling (FDM) 3D Printers

Kemel Shomenov

Supervisors: Professor Essam Shehab and Associate Professor Md. Hazrat Ali

Submitted in fulfillment of the requirements
for the degree of Master of Science
in Mechanical & Aerospace Engineering

April 2024

DECLARATION

I hereby, declare that this manuscript, entitled "*Sensor-based digital twin for Fused Deposition Modeling (FDM) 3D printers*", is the result of my own work except for quotations and citations, which have been duly acknowledged.

I also declare that, to the best of my knowledge and belief, it has not been previously or concurrently submitted, in whole or in part, for any other degree or diploma at Nazarbayev University or any other national or international institution.

Name: Kemel Shomenov

Date: 05.04.2024

Abstract

The development of Digital Twin for 3D printing is crucial to optimize the printing process and achieve high-quality printed objects. It allows to improve the current limitations of Fused Deposition Modeling (FDM) 3D printing such as long printing time, need for monitoring, and defects of printed parts. There are many studies on Digital Twin development for FDM 3D printing, including IoT-based monitoring, machine learning, and image processing. However, sensor-based approaches with proper sensor selection, data transfer, and visualization have not been fully explored yet.

The aim of this work is the development of a Digital Twin for FDM 3D printing with improved accuracy, resulting in better control and optimization of the printing process. The main approach to building the proposed DT system consists of several important steps such as data collection, data transfer, data storage, data analysis, and a graphical user interface (GUI) that allows monitoring and control of the printing process. The system has two types of data which are data from a 3D printer and data from embedded sensors. Data from the printer were retrieved using Python, while sensor data were collected via Arduino modules and stored in a real-time database. Different sensors were compared for parameters like filament flow rate and nozzle/bed position. The Firebase database was chosen after comparison, and Unity 3D was selected as the GUI. The controller sends GCode commands to the printer line by line, enabling real-time editing and automatic defect detection. Key research results include successful integration of sensor data with printer data, selection of appropriate database and GUI platforms, and implementation of real-time control, monitoring and autonomous defect detection capabilities.

The novelty of this research is that it proposes the application of affordable and accurate sensors that have not been suggested before. Furthermore, it does not use third-party hosts to control the 3D printer but instead employs Python, which allows full flexibility for defect detection and print optimization.

Acknowledgements

I extend my deepest gratitude to Professor Essam Shehab for his invaluable guidance and unwavering support throughout the entire process of completing this thesis. I am truly grateful for his insights, constructive feedback, and constant encouragement, which have consistently pushed me to strive for the highest standards in both my thesis report and presentation.

I thank my co-supervisor Professor Md. Hazrat Ali, whose extraordinary support helped me throughout the thesis work. His intense commitment to my topic, prompt academic guidance, and knowledge of research methods have been instrumental in the completion of this thesis.

Furthermore, I am grateful to Master Thesis II Instructor Dmitriy Sizov for organizing progress presentations and rehearsals. His constructive feedback was helpful during the final presentation.

In addition, I want to thank Mr. Nursultan Jyeniskhan for his support and advice that were helpful to write high quality report.

Lastly, I would like to express my gratitude to my family for their unwavering belief in me and continuous support throughout my Bachelor's and Master's degree pursuits.

Table of Contents

List of Figures

Chapter 1 – Introduction

1.1. Background

1.1.1 Digital Twin

Currently, there is a global trend towards Industry 4.0, which involves the incorporation of advanced technologies, such as AI (artificial intelligence), machine learning, and the IoT (Internet of Things), into manufacturing and other industrial fields to establish more intelligent and effective systems. One of the terms that is mainly associated with Industry 4.0 is the digital twin. A Digital Twin is a virtual copy of a physical product, process, or system that allows real-time monitoring, analysis, and optimization of performance under various conditions (Grieves, 2011). The digital twin in Figure 1 for instance, is synchronized with real-time data from the physical robot arm and allows remote monitoring and control of its movements. Digital Twinning is a trending topic in the research community as the number of publications about it has increased exponentially over the past few years (Lim et al., 2020). The idea of the digital twin was first introduced by Michael Grieves in 2003 during his course on product lifecycle management at the University of Michigan. Initially, digital twins were mainly used in the military and aerospace industries. Presently, the concept of the digital twin is undergoing significant advancements and progress (Tao et al., 2022).

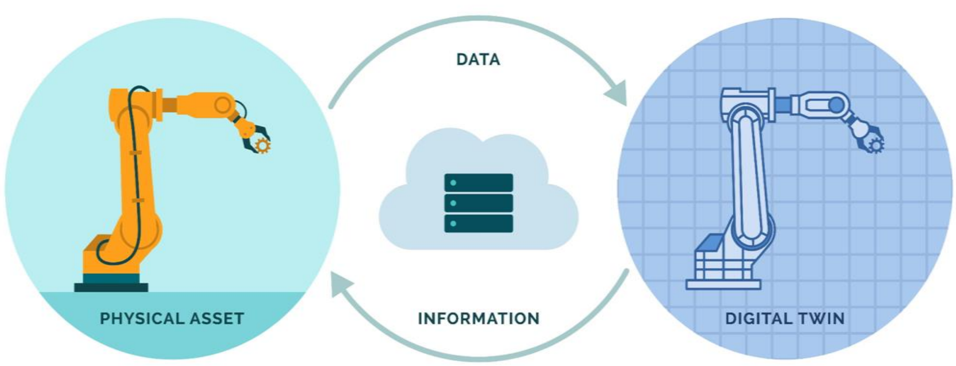

Figure 1: Digital twin of a robot arm (Korolov, 2022)

1.1.2 Additive Manufacturing (AM) and Fused Deposition Modeling (FDM)

Additive manufacturing refers to a manufacturing technique in which the model is constructed by adding material layer by layer. The CAD software's model is converted into an appropriate file format such as STL, which is then subjected to the slicing process. During this process, the slicer determines the G-code instructions required for the 3D printing hardware. There are many types of additive manufacturing depending on the technology or materials used. Figure 2 shows AM classification based on the materials used for 3D printing such as solid, powder, and liquid.

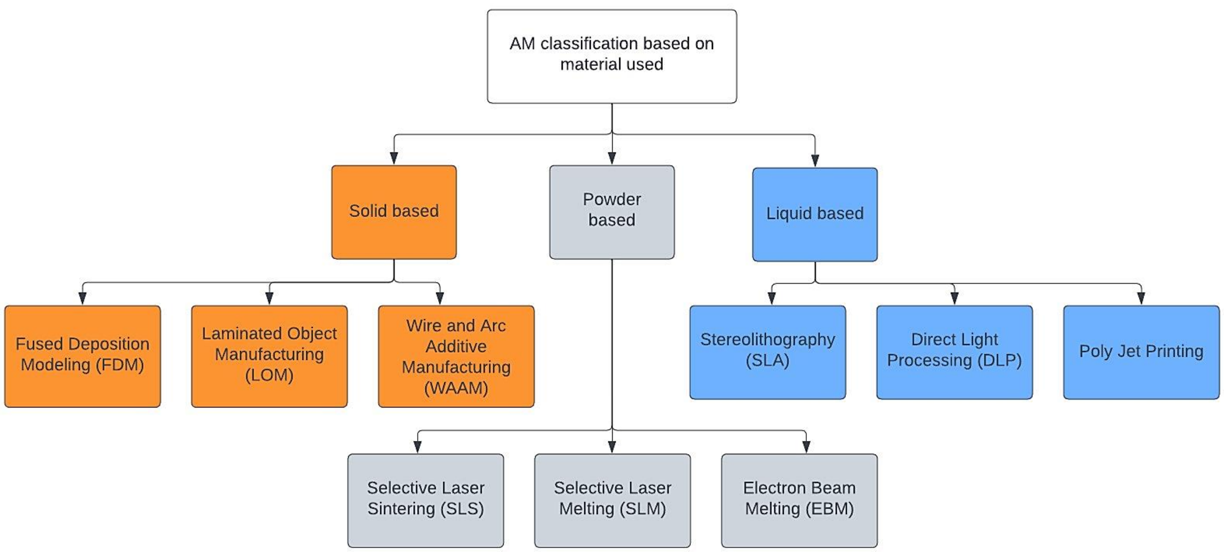

Figure 2: AM classification based on the material used (Salifu et al., 2022)

The most popular type of 3D printing is Fused Deposition Modeling (FDM). In this technique, objects are created layer by layer by extruding heated thermoplastic material through a nozzle. Different thermoplastics have different melting points, and they are used in the form of filaments. The most common filaments are Poly Lactic Acid (PLA), Acrylonitrile Butadiene Styrene (ABS), Nylon, and Polyethylene Terephthalate Glycol (PETG). Figure 3 shows the structure of a typical FDM 3D printer. Three stepper motors are used to move the nozzle and bed in the x, y, and z directions. There are different configurations such as stationary bed with nozzle moving in all three directions, bed moving in the axis with nozzle moving in x and z axes, and bed moving in z axis with nozzle moving in x and y directions.

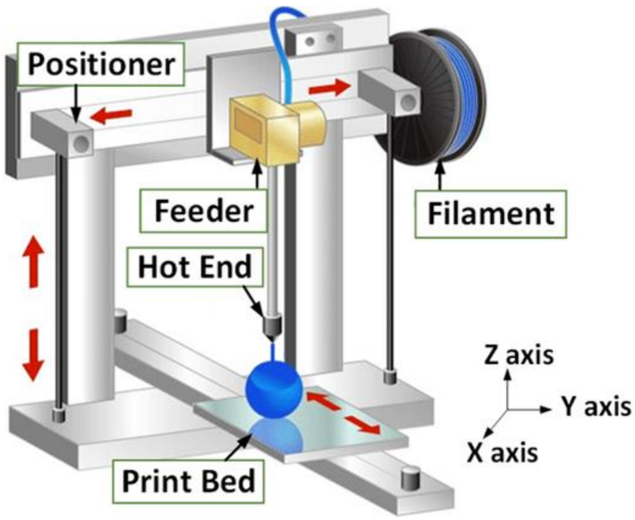

Figure 3: The FDM 3D printer structure (Li et al., 2018)

FDM 3D printers have two types of extruder setup which are Direct-drive and Bowden as shown in Figure 4. In a direct extrusion system, the extruder is positioned on the printhead and directly feeds the filament into the hot end. On the other hand, a Bowden system typically has the extruder mounted on the printer's frame, causing the filament to pass through a PTFE tube before reaching the printhead. Each of them has their advantages and disadvantages. Direct drive ensures reliable extrusion as the extruder is mounted directly onto the printhead, allowing the motor to effortlessly push filament through the nozzle. However, as the extrusion system is mounted on the hot end it adds extra weight which causes additional vibrations.

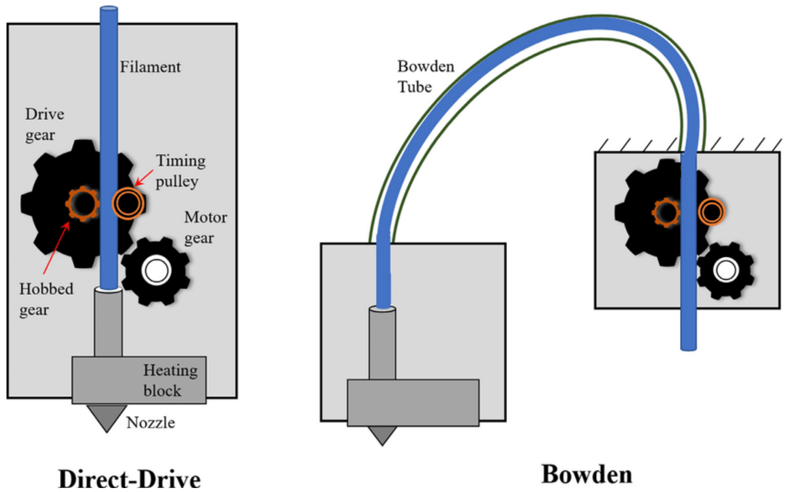

Figure 4: Direct-drive and Bowden tool head designs (Bhagia et al., 2021)

1.1.3 FDM 3D Printing Problems

FDM technology, like any other manufacturing method, has its limitations and challenges. There are several printing defects related to FDM technology that users should be aware of. One of the most common among them is a warping effect which is associated with the natural properties of thermoplastics. During cooling, the material starts to contract, and internal stresses develop within the part that cause warping as shown in Figure 5. Heating and cooling parameters should be selected properly to prevent this defect.

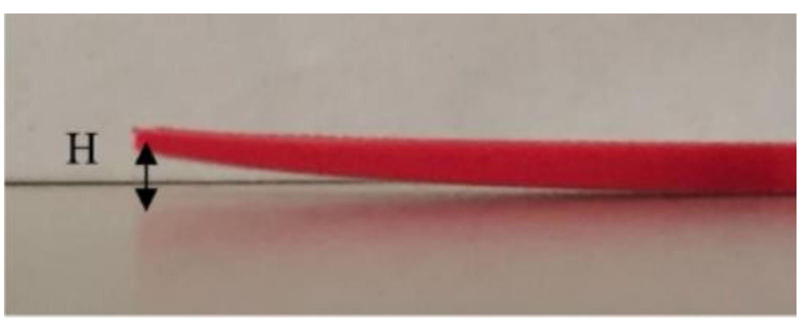

Figure 5: The warped edge of a printed sample (Syrlybayev et al., 2021)

The next limitation of FDM technology is extrusion problems such as under-extrusion or over-extrusion. The over-extrusion is mostly caused by wrong configuration in slicer settings, while under-extrusion is caused by several factors such as a partially clogged nozzle, less filament compression, and worn extruder gear. The under-extrusion results in poor geometry and weakened mechanical properties (Figure 6). Sometimes, filament might not be extruded at all because of clogged nozzle or filament breakage.

Figure 6: Printed part with under extrusion

Another issue of FDM 3D printing is layer shifting. Typically, FDM 3D printers have ball screws responsible for z-axis movement and belts connected to stepper motors for movements in x and y directions. Stepper motors accomplish movements of the nozzle and bed according to the Gcode, however 3D printer itself has no feedback about the actual location. Over time, the belts in the x or y direction might stretch which can cause layer-shifting defects (Figure 7). Furthermore, loosened fasteners of pulleys also might cause this problem.

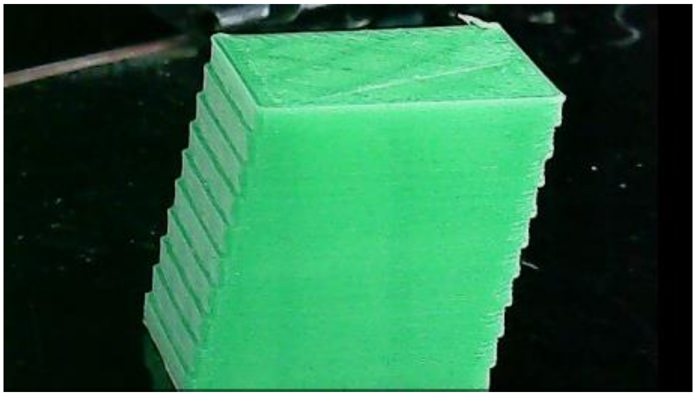

Figure 7: Layer shifting

One of the problems of FDM 3D printing which can be also caused by non-optimal printing configurations and mechanical issues is vibrations. Non-desired vibrations can occur because of loosened screws and other mechanical parts of a 3D printer. Furthermore, speed and acceleration settings in the slicer also can cause excessive vibrations.

1.2. Motivation

This study focuses on the development of Digital Twin for 3D printing, which is one of the popular non-traditional types of manufacturing. The FDM (Fused Deposition Modeling) was chosen for this study as it is the most widely used technique of 3D printing where objects are created layer by layer by extruding heated thermoplastic material through a nozzle. The FDM 3D printers have several limitations such as long printing time, the need for monitoring, and different defects of printed parts. The Digital Twin for 3D printers by using advanced technologies allows for the optimization of printing parameters and getting high-quality printed objects.

Several studies have been done in this field to develop digital twins by using advanced technologies. However, a sensor-based approach with proper sensor analysis for measuring different parameters such as temperature, vibration, and position has not been done yet. Furthermore, not all research works have fully developed digital twin frameworks with all components such as data acquisition, transfer, analysis, and GUI (graphical user interface). This research work aims to develop a digital twin with high-accuracy sensors by experimentally comparing different types of sensors for corresponding parameters.

The digital twin is one of the new concepts that can be used to make industrial processes more efficient. This research is aimed to develop a digital twin framework for FDM 3D printers which can be helpful for other researchers who are interested in integrating this new concept into other fields of industry. The methods used in this research for data acquisition, transfer, and analysis can be used for other research works as well.

1.3. Research Aim and Objectives

This research aims to develop a sensor-based digital twin for FDM 3D printers. There are several research objectives which are to:

- Evaluate existing sensors and technologies to determine the most suitable sensors for monitoring 3D printing process parameters;
- Design a data acquisition system that collects sensor data throughout the printing process;
- Develop a data transfer method between the physical model and the digital model;
- Create a digital twin model that mirrors the printer's behavior, incorporating the sensor data for real-time monitoring, control, and autonomous defect detection;
- Conduct a series of experiments to validate the accuracy and reliability of the developed system.

1.4. Thesis Structure

This thesis consists of five chapters. The comprehensive literature review is presented in Chapter 2. Several key pillars of Industry 4.0 are presented, and it is presented how they are utilized in developing a digital twin for FDM 3D printers. Chapter 3 describes detailed methodology of proposed digital twin development by compromising main stages such as sensor integration and data transfer. Chapter 4 shows results of the thesis work and discussion of the main outputs. The final chapter presents conclusions of the research and future works.

Chapter 2 – Literature Review

2.1. Introduction

Several research studies have been done on developing a digital twin for 3D printing. The general process workflow of digital twin for 3D printing in Figure 8 was described by Kantaros et al. (2021). It shows the interconnection between digital and physical systems where the main aim of the digital twin is to monitor and optimize process parameters to obtain high-quality prints. Different authors use different methods by integrating particular types of advanced technologies such as machine learning, image processing, and IoT (Internet of Things) to achieve this goal.

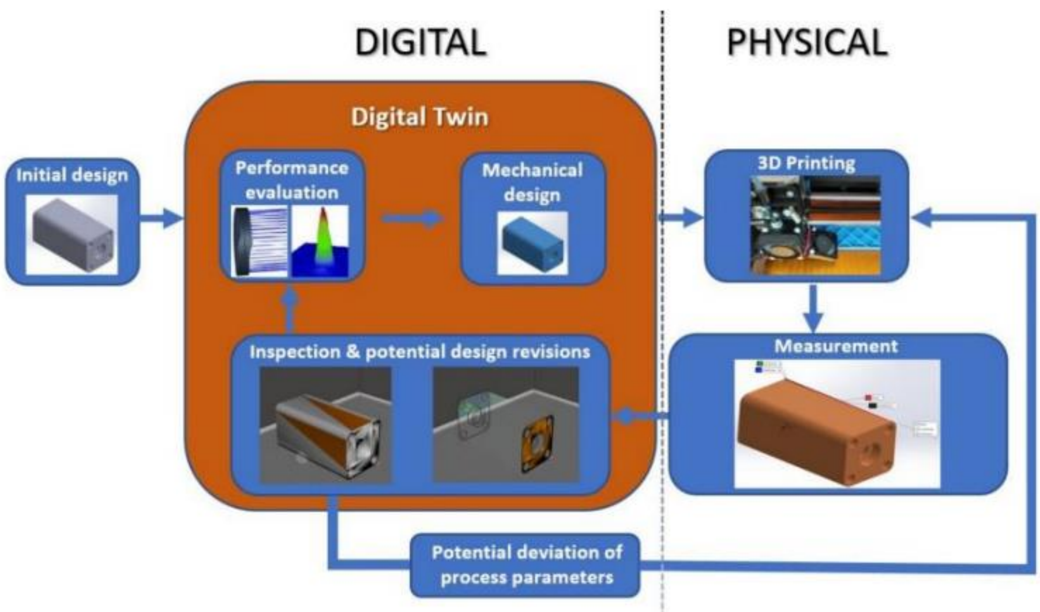

Figure 8: Process workflow of a digital twin for 3D printing (Kantaros et al., 2021)

Figure 9 shows pillar technologies for digital twin development described in this literature review. Machine learning algorithms analyze sensor data to predict defects and optimize parameters, while image processing techniques monitor printing progress and quality. IoT devices collect real-time data for remote monitoring and control, and augmented reality provides visualizations and overlays for interactive process management.

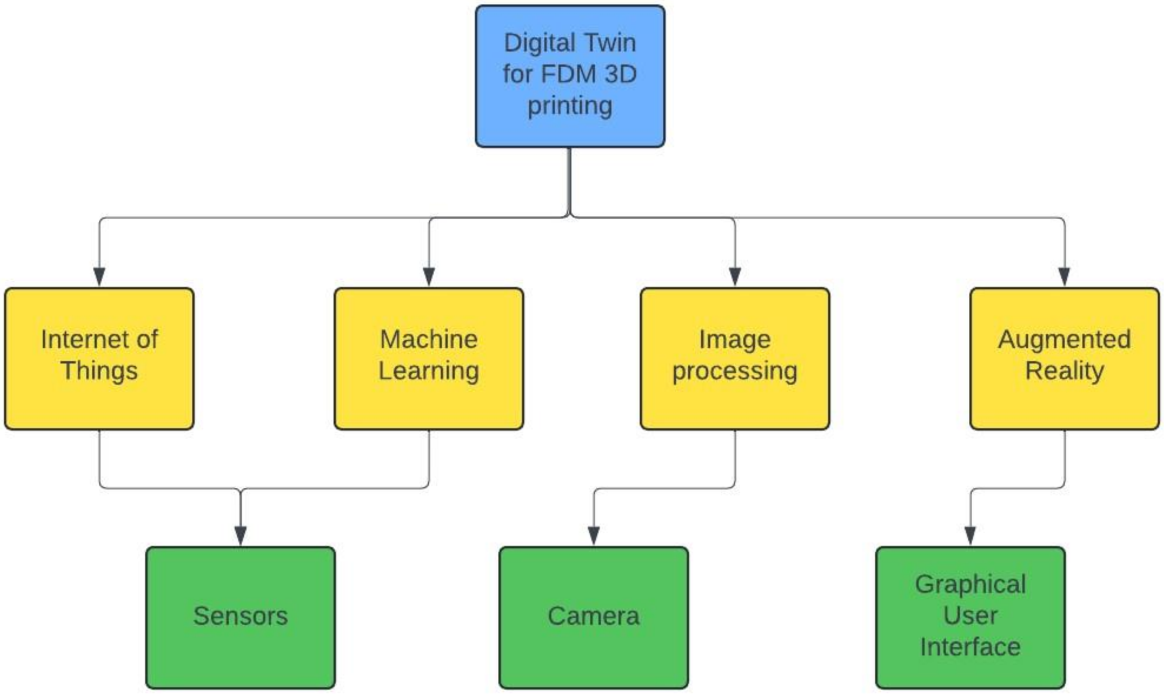

Figure 9: Digital twin pillar technologies

2.2. IoT-based Monitoring

Barbosa and Aroca (2017) have proposed IoT architecture for the control and monitoring of 3D printing processes. The main component of this architecture is Beacons installed on a 3D printer that allows data transfer via Bluetooth technology. Those beacons continuously send data to a web address where the user can remotely access all data about the 3D printer's parameters via mobile devices in real-time. In addition, users can control 3D printers remotely by using this technology. However, this architecture uses data only from the 3D printer's serial connection, which, for some parameters, shows preassigned values rather than real-world data. In comparison, Chhetri et al. (2019) have presented IoT-based methodology by utilizing some sensors to develop digital twin for 3D printing. Their method is based on using side channels to identify abnormal faults and deduce the quality of printed parts while remaining up to date. They used vibration, acoustic, power, and magnetic sensors to represent the physical status of their 3D printer (Figure

10). The authors concluded that their digital twin achieved an average accuracy of 83.09% for error localization. The next research work by Kakade et.al (2022) uses a rotary encoder to measure material flow and a load cell to measure the weight of available material. Their setup is designed to detect printing faults related to filament flow such as material jamming and filament runout. The authors used Octoprint installed in Raspberry Pi to control the 3D printer remotely during filament jamming or runout. Most of the papers integrate some sensors for particular printing parameters and develop IoT setups for real-time remote monitoring and control. Kazhymurat et al. (2022) for instance, installed thermistors, accelerometers, and cameras to be able to monitor and control their 3D printer remotely in real-time.   Overall, these studies present advanced monitoring of the 3D printing process but do not include optimization methods.

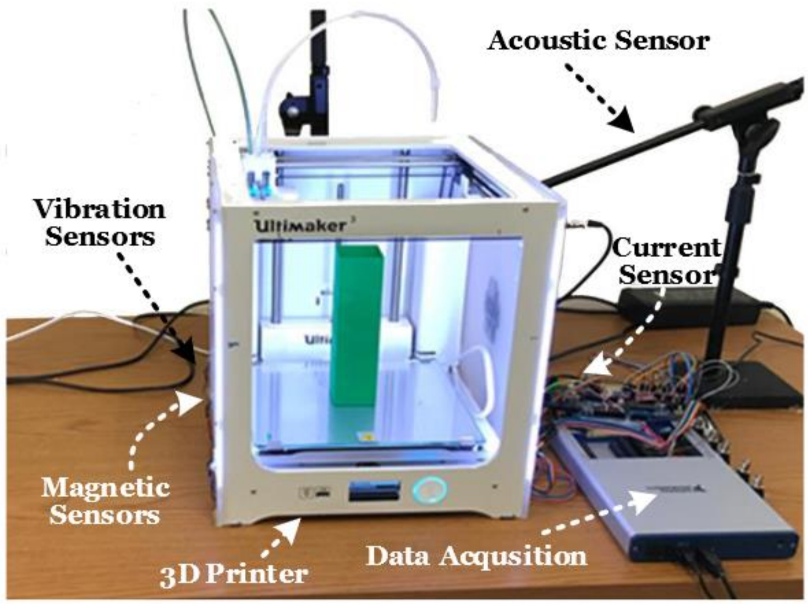

Figure 10: IoT setup for FDM 3D printer (Chhetri et al., 2019)

2.3. Machine Learning and Neural Networks

Only monitoring the 3D printer's parameters is not enough to build a fully functional digital twin. The data about the 3D printer's status should be analyzed to optimize necessary parameters. Sampedro et al. (2022) have presented machine machine-learning algorithm to detect nozzle clogging which is one of the undesired FDM 3D printer failures. Their machine-learning algorithm could predict nozzle clogging with 97% accuracy. Likewise, Butt and Mohaghegh (2022) also investigated filament extrusion from the nozzle of a 3D printer. However, they focused on the effect of the extrusion system on the quality of the print. They collected data by changing different parameters of the extrusion system such as nozzle temperature and filament feed rate and evaluating the quality of those prints by measuring surface roughness, tensile, and hardness. Those data were used for machine learning for analyzing and identifying patterns and trends. Another similar example of machine learning integration was done by Boschetto et al. (2013). Their developed neural network can predict surface roughness of the printed part which is the main indicator of print quality. They trained the neural network with collected data including the measured surface roughness values by profilometer and the corresponding printing parameters. Another model of Neural Networks was developed by Scheffel et al. (2021) for printing fault detection using vibration sensors. The authors identified the vibration patterns associated with faulty products, which demonstrate the viability of this approach. By employing convolutional neural networks (CNN) to independently learn data inspection without requiring specific domain knowledge, the processes become more streamlined and straightforward. To conclude, these articles show how machine learning and neural networks can be used in developing digital twin by accurately predicting printing defects.

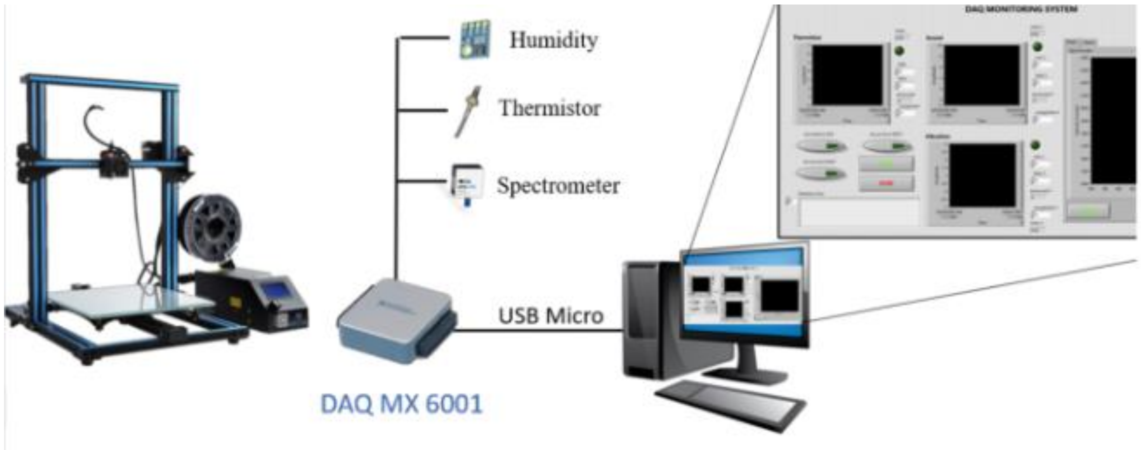

Figure 11: Data acquisition setup for investigating filament extrusion (Sampedro et al., 2022)

2.4. Image processing

Usually, data collection is done by using sensors that track particular parameters of a 3D printer such as position, temperature, and vibration. Installing a camera on the 3D printer provides an opportunity to not only observe the printing process but also integrate image processing and machine learning. Henson et al. (2021) proposed a new method of real-time distortion and failure detection by using cameras. The authors used three cameras to capture 3D-printed parts from three different positions. Those cameras are linked to MATLAB software to process images in real-time. The captured images from different perspectives are compared with a 3D model of the object for the presence of any geometrical distortion. In comparison, Moretti et al. (2021) used a digital video microscope to obtain optical imaging of a printed layer for its contour identification and stacking. The device was installed near the nozzle of the 3D printer to capture each printed layer from the top view as shown in Figure 12. Then image processing was applied to compare printed layer contours with preassigned ones in Gcode. A similar approach was also used by (Jin et al., 2019; Farhan Khan et al., 2021). They also installed a camera near the nozzle and with the help of real-time monitoring and machine learning algorithms, print defects are predicted. In conclusion, image processing via camera is also a useful tool for defect detection and can be used in developing a digital twin.

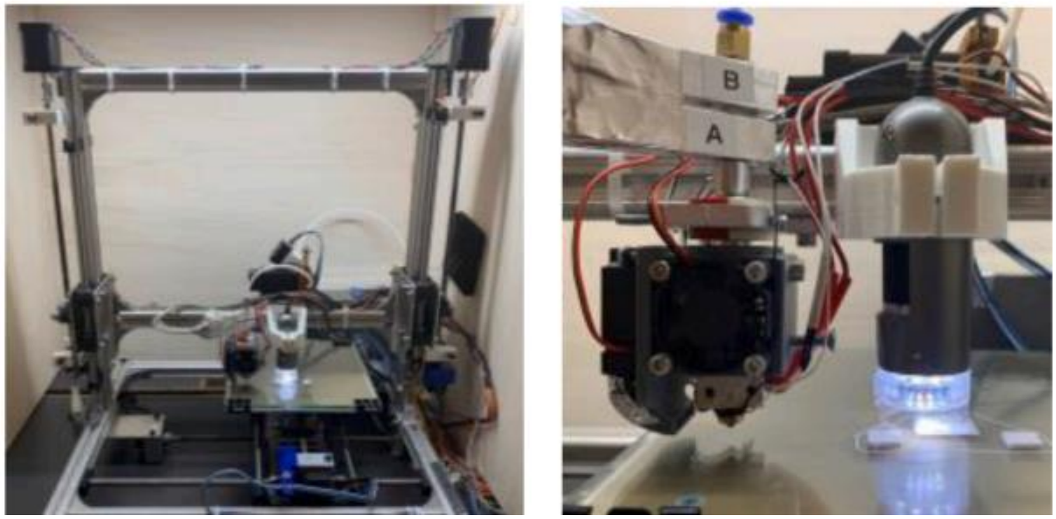

Figure 12: Optical system installed on the 3D printer (Moretti et al., 2021)

2.5. Augmented Reality Integration

Augmented Reality can be implemented not only for realistic visualization but also for monitoring and control of a 3D printer. Ceruti et al. (2017) integrated AR with FDM 3D printer to detect geometrical deviations in printed parts. The AR-Viz environment was used to compare the printed part with the CAD model by overlapping method (Figure 13). It is possible to superimpose a virtual model of the part being printed onto the video streaming from glasses. The studies in (Mourtzis et al., 2021; Paripooranan et al., 2020) also built a digital copy of the 3D printer that can be adapted to an AR environment. To conclude, AR is a great tool to visualize the digital twin of a 3D printer and also can include several useful functions.

Figure 13: 3D printer and Wuzix glasees (Ceruti et al., 2017)

2.6. Digital Twin frameworks with 3D printer visualization

Machine-human interaction is one of the important parts of the digital twinning process. Several researchers tried to develop their frameworks with the visualization of 3D printers. The main aim of building digital replicas that mimic real 3D printers is to enhance user experience and simplify monitoring and control. Pantelidakis et al. (2022) built their digital twin in Unity 3D that mimics the real 3D printer by using data from the printer itself and embedded sensors (Figure 14). They used Octoprint to extract data from the printer. Infrared distance sensors and thermocouples were used to measure nozzle/bed position and temperature respectively. Similarly, Rachmawati et al. (2023) proposed a digital twin platform that can be created in Unity 3D. They also showed that Lightweight Convolutional Neural Network (LCNN) can used for fault detection based on sensor data. The environmental sensor was used to measure the humidity and temperature of the room, while contactless temperature sensors were used to measure nozzle and bed temperatures. In comparison, Corradini and Silvestri (2022) visualized not only the 3D printer’s motion but also material extrusion as well. The authors obtained the data about main 3D printing parameters from OctoPrint and from the installed sensors. They used encoders embedded in the stepper motors to track the nozzle position of the 3D printer. While thermocouples are used to provide data about temperature. Additionally, accelerometers are used to sense vibrations on the 3D printer. Then, the authors used those data to visualize the 3D printing process by using Panda3D and Python libraries. Then, real 3D printed parts and visualized parts are compared and analyzed.

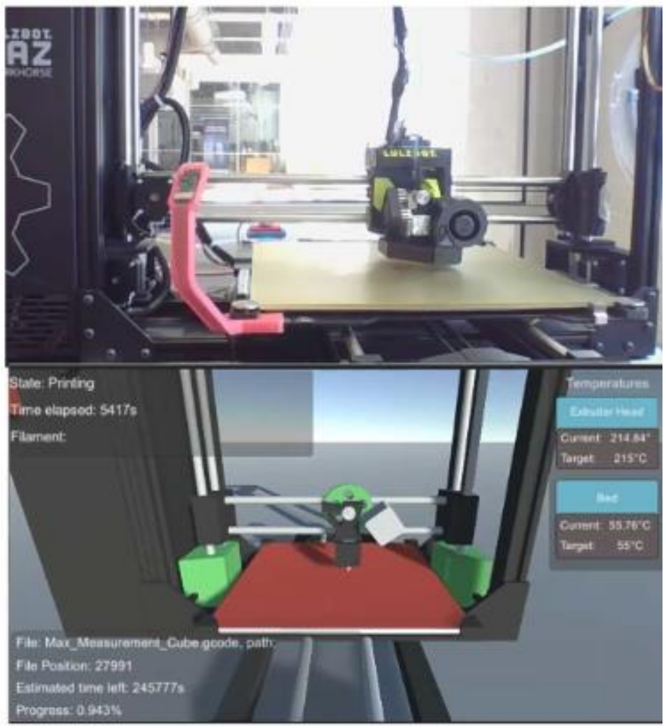

Figure 14: 3D printer and its digital replica in Unity 3D (Pantelidakis et al., 2022)

Table 1: Synthesis of publications

Authors & year	**Data acquisition method**	**Sensors used (parameters measured)**	**Data analysis method**	**Visualization method**
Pantelidakis et al. (2022)	Octoprint	• IR sensors (position) • Thermocouples (nozzle and bed temperature)	Sensor data and printer data comparison	Unity 3D

Corradini and Silvestri (2022)	Octoprint	• Encoders for each axis (position) • Thermistors (nozzle and bed temperature)	Comparison between CAD model and 3D model from sensor data	Panda3D
Yi et al. (2021)	Serial port	• ZMPT101B (Voltage) • ACS712 (Current)	Layer-by-layer comparison between AR-based DT and real print	Siemens NX
Rachmawati et al. (2023)	Raspberry Pi	• Environmental sensor (humidity and temperature) • MLX90614 (nozzle and bed temperature)	Lightweight Convolutional Neural Network (LCNN)	Unity 3D

2.7. Research Gap Analysis

After analyzing the studies listed above, several points were identified as a research gap. Firstly, most of the authors do not cover all important printing parameters while building their digital twin. Pantelidakis et al. (2022) for instance, did not integrate filament sensors to measure filament usage. It is important to consider all basic printing parameters to develop a fully functional digital twin of a 3D printer. Secondly, most of the researchers did not justify their choices on sensor selection. Most of the authors like Corradini and Silvestri (2022) used encoders to monitor nozzle/bed positions, however mechanical faults caused by belts or screws can not be detected by encoders. Lastly, most of the research work does not include optimization of printing processes despite that the main aim of DT development is to optimize the process of its physical object.

Chapter 3 – Methodology

3.1. Overview

The main parts of this research work are shown in Figure 15. The main component of every digital twin is data. Therefore, the first step is data collection. The primary element of any digital twin is data. In this research, two types of data are employed to construct the digital twin. The first type originates from a printer and can be obtained by sending specific Gcode commands through the serial port. For instance, temperature readings from thermistors under the bed and inside the hot end can be acquired using the M105 command, and the Gcode line being processed allows us to determine nozzle position and track the progress of a print. The second type of data comes from embedded sensors. Since most affordable 3D printers only have temperature sensors, additional external sensors are utilized to measure printing parameters. Following data collection, the next step involves analyzing the gathered information. The system is designed to independently analyze data and identify any printing issues. It should also have the capability to autonomously pause or stop the print in case of anomalies. The final component of the digital twin system is the graphical user interface (GUI) in platforms such as Unity. The GUI is intended to enable users to remotely monitor and control the 3D printing process. All data from the printer and sensors should be presented in an organized manner for user convenience.

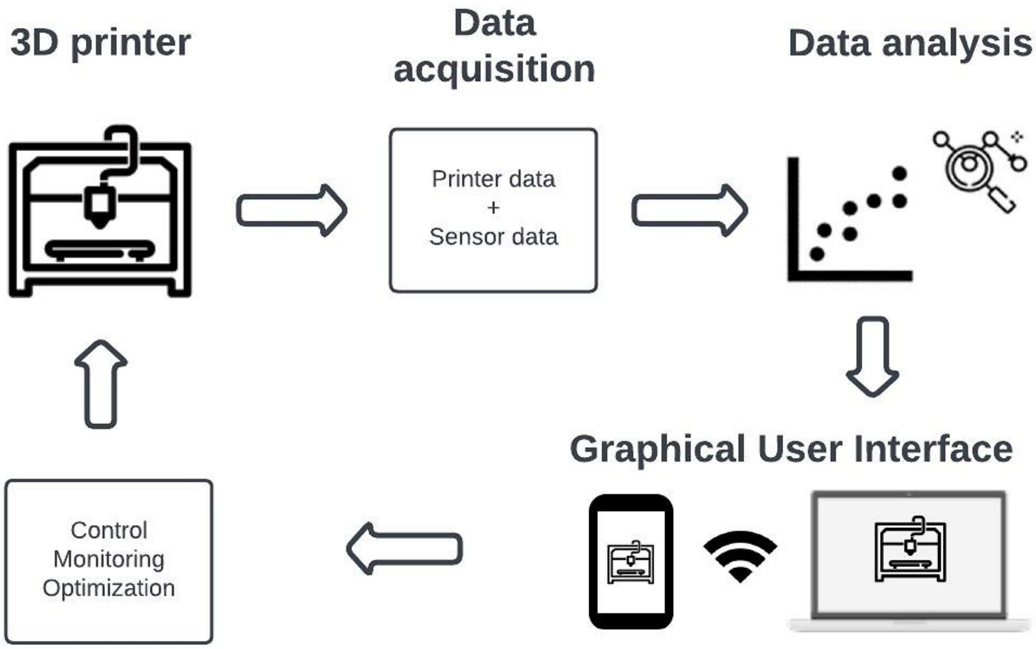

Figure 15: General workflow

3.2. Identification of Parameters

The first step to start data collection is to identify necessary printing parameters. The five process parameters are selected which are nozzle temperature, nozzle position, bed temperature, vibration, and filament flow rate (Figure 16). Those five parameters are crucial for every FDM 3D printer and are responsible for common printing defects (Figure 17). Nozzle temperature determines the material's melting characteristics, while nozzle position ensures precise layer deposition for accurate prints. Bed temperature influences initial layer adhesion, preventing warping issues. Vibration control is essential to avoid print defects caused by excessive printer movements. Lastly, filament flow rate directly impacts layer thickness and overall print quality. It's observed in the literature review that many researchers tend to concentrate on just one or two of these printing parameters. However, for building a fully functional digital twin, all five parameters should be covered.

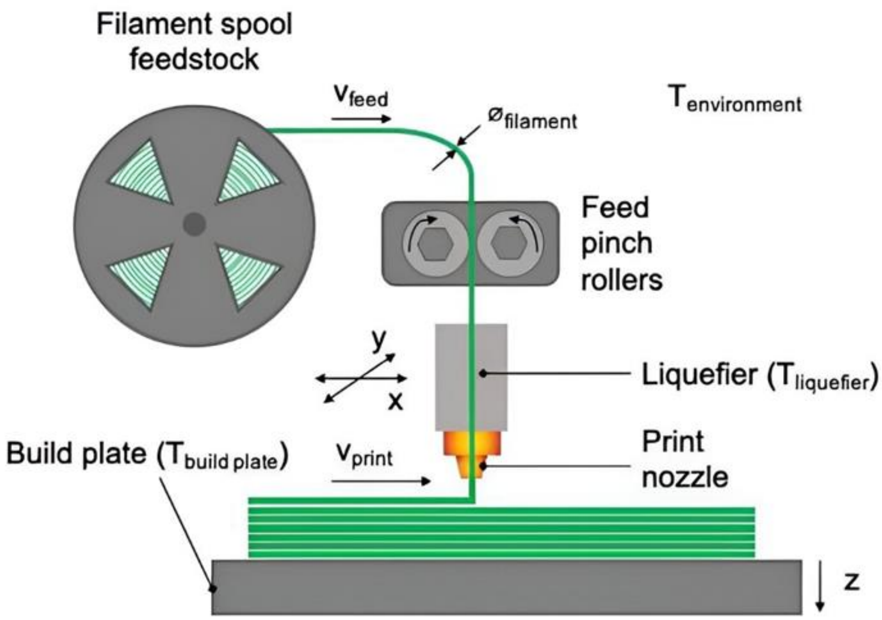

Figure 16: Working principle of the FDM and key process parameters (Vaes and Van, 2021)

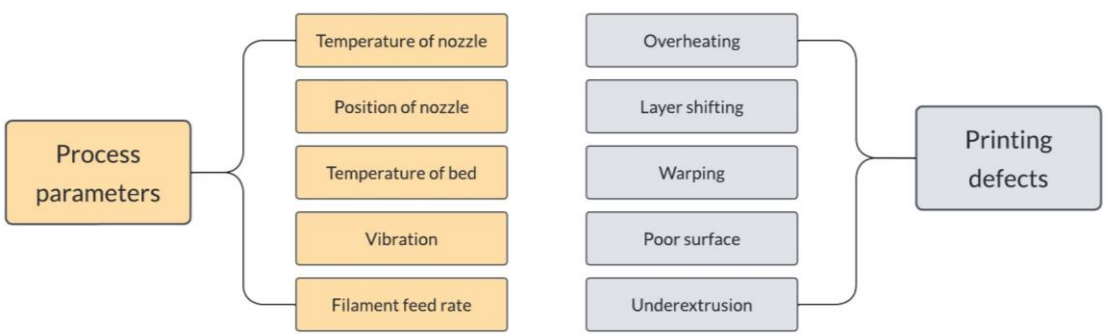

Figure 17: Process parameters and printing defects

3.3. Sensor Integration

3.3.1 Vibration

Vibration can have a significant impact on FDM (Fused Deposition Modeling) 3D printing. Unnecessary vibrations can cause the extruder to move unintentionally, resulting in the layer height being inconsistent, or the layers not bonding properly. It is one of the printing parameters that can affect the overall printing quality. The vibration can be measured by using special sensors based on sensors based on the principle of piezoelectricity. Piezoelectricity is a phenomenon in which certain materials generate an electric charge in response to applied mechanical stress. When the sensor is exposed to mechanical vibrations, the piezoelectric material generates a small electrical charge in proportion to the strength and frequency of the vibration. This charge is picked up by electrodes attached to the metal plates. After amplifying those signals they are processed by the sensor's electronics to produce a measurable output signal.

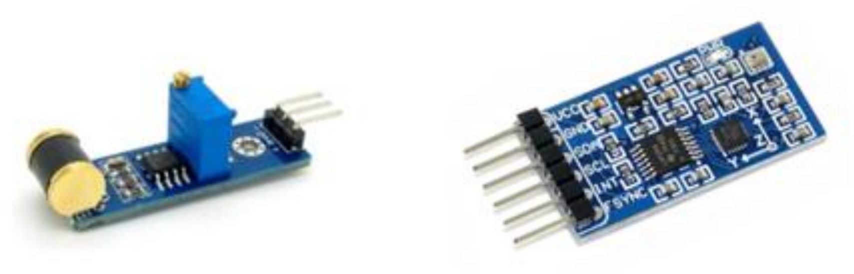

Figure 18: Sencera 801S vibration sensor and Waveshare 10DOF IMU sensor

Two types of sensors were available for vibration tracking as shown in Figure 18. The first one is the Sencera 801S vibration sensor which is a micro shock detector. The second one is the IMU sensor which has a built-in gyroscope and accelerometer. The accelerometer can sense acceleration in x,y, and z directions. The main advantage of using an accelerometer compared to a vibration sensor is its ability to sense vibration in three directions independently. Furthermore, most of the researchers have used accelerometers installed on their 3D printers to sense vibrations.

3.3.2 Filament flow

One of the important parameters of FDM 3D printers is the amount of filament flow rate. Most of the portable 3D printers do not have sensors that track the filament usage. In reality, it is an important parameter that is responsible for the quality of the printed part. Tracking the filament allows us to detect and predict under extrusion and filament run out or breakage. Some modern FDM 3D printers have filament sensors, but most of them are designed to detect only filament breakage or runout. The limit switches can be used to build simple filament runout sensor as shown in Figure 19. However, these sensors cannot be used to measure exact amount of filament being extruded.

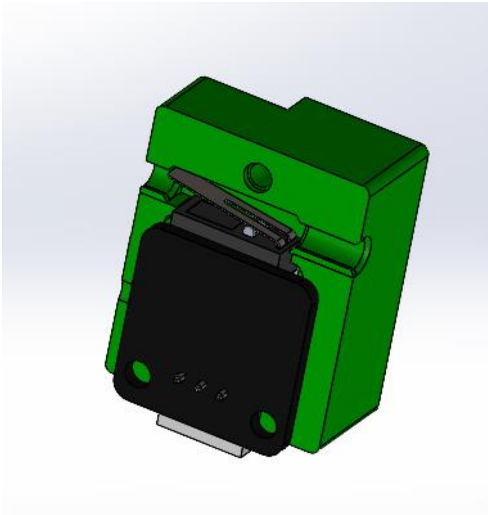

Figure 19: Filament sensor with limit switch

The sensor to measure filament usage is built by using L393 IR speed sensors (Figure 20). These sensors are usually used as encoders for rotating wheels. The sensor consists of an infrared emitter that emits an infrared beam in the direction of the rotating object. On the opposite side of the sensor, there is an infrared receiver positioned to receive the infrared beam. The sensor can count received signal through the holes in rotating wheel, which can be further used to calculate rotation speed.

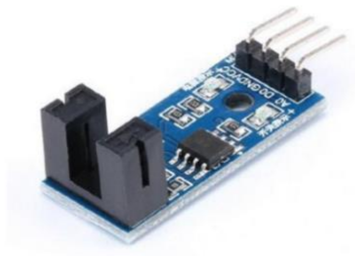

Figure 20: L393 IR speed sensor

The encoder disc with 24 holes is used to track the filament usage. When extruder pulls the filament, the disc rotates because of the friction. The gear and bearing are used to grab the filament. The spring is responsible for providing enough friction, so the movement of the filament rotates gear. The one sensor counts the holes, while the other is responsible for detecting the direction of rotation (Figure 21).

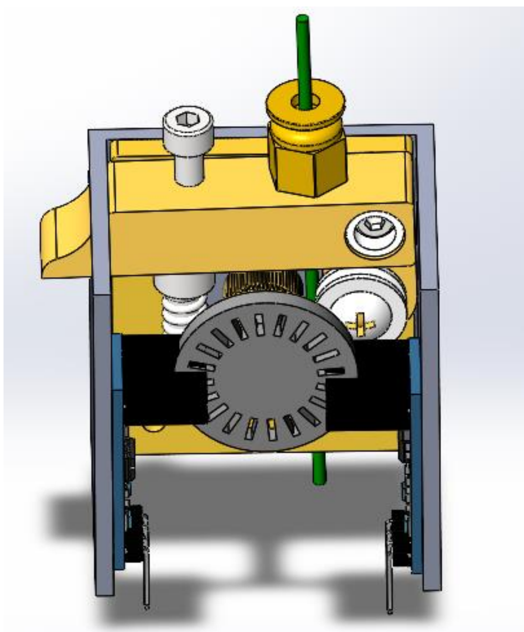

Figure 21: CAD model of the filament sensor

3.3.3 Nozzle/bed position

Some articles from the literature suggested to use laser distance sensors to measure nozzle position. The VL53LOX distance sensor was tested to measure nozzle position along the X axis (Figure 22). It works by emitting a laser and measuring the time it takes for the light to bounce back from an object to calculate the distance. The main advantage of this sensor is that it is compact and contactless.

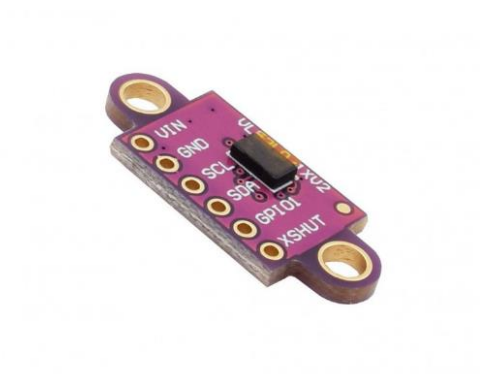

Figure 22: VL53LOX distance sensor

Figure 23 shows the test setup of the distance sensor for X-axis of the 3D printer. The sensor is fixed on the left side of the 3D printer and measures distance from the nozzle. A 3D printed PLA sheet is fixed to the nozzle to serve as a wall for the laser to bounce off.

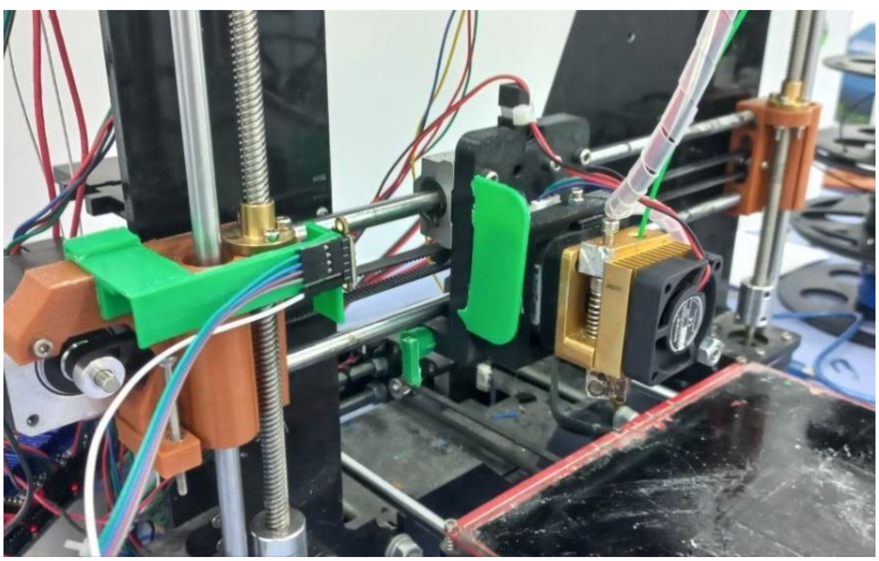

Figure 23: The distance sensor testing for X-axis

The results of the experiment (Figure 24) showed that the sensor outputs are not stable. They fluctuated over 3 mm range when nozzle is not moving. This is not acceptable for digital twin development as 3D virtual copy of the 3D printer should repeat exact nozzle movement in real time. Furthermore, the readings of the distance sensors depends on the material of the wall and can be affected by ambient light in the room.

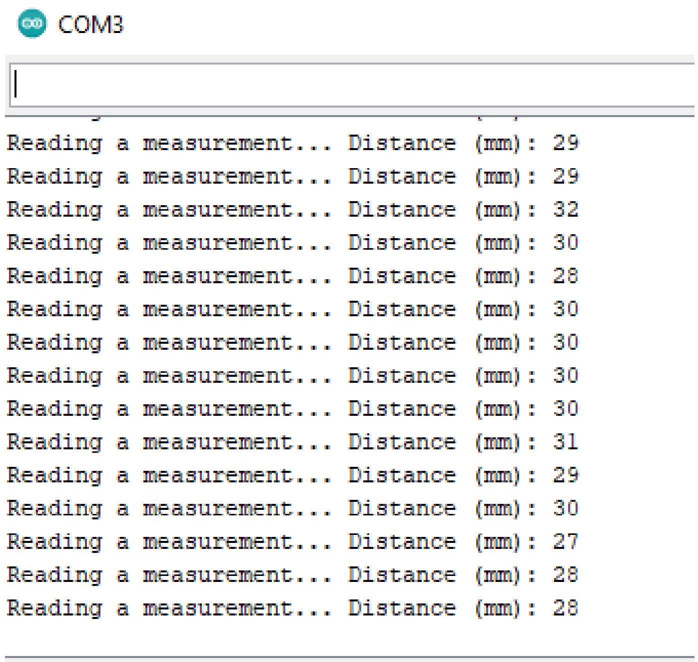

Figure 24: The distance sensor readings

The capacitive displacement sensors would be best option for nozzle position measurement. They give stable output and have 0.01-0.1 mm deviation depending on its price which is suitable for FDM 3D printers. Figure 25 shows structure of a capacitive displacement sensor. These sensors consist of a collection of grid-capacitances and signal processing circuit. The grid-capacitances are divided into stator grid-capacitance and sliding grid-capacitance. The stator grid-capacitance is created by etching a pattern of grids directly onto the top copper layer of circuit board stator. On the slider, another printed circuit board contains 48 separate etched grids known as emitters. Together, these printed circuit boards form two variable capacitors. As the slider moves, the capacitance changes in a linear and repeating pattern. To determine the precise position of the slider, the signal processing circuitry integrated into the slider counts the grids as the slider moves and performs a linear interpolation based on the magnitudes of the capacitors.

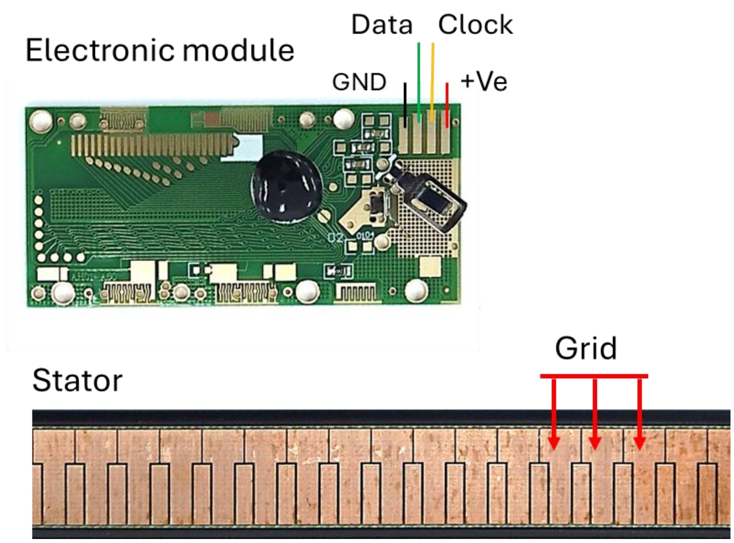

Figure 25: Structure of capacitive displacement sensor

Figure 26 shows how digital calipers can be connected to the Arduino to read their values. Four wires should be connected to four pins of the electronic module which are 5V, GND, ClOCK, and DATA. Furthermore, 200 ohm resistor, 10uF capacitor, and one diode are needed to regulate the supply voltage for the caliper.

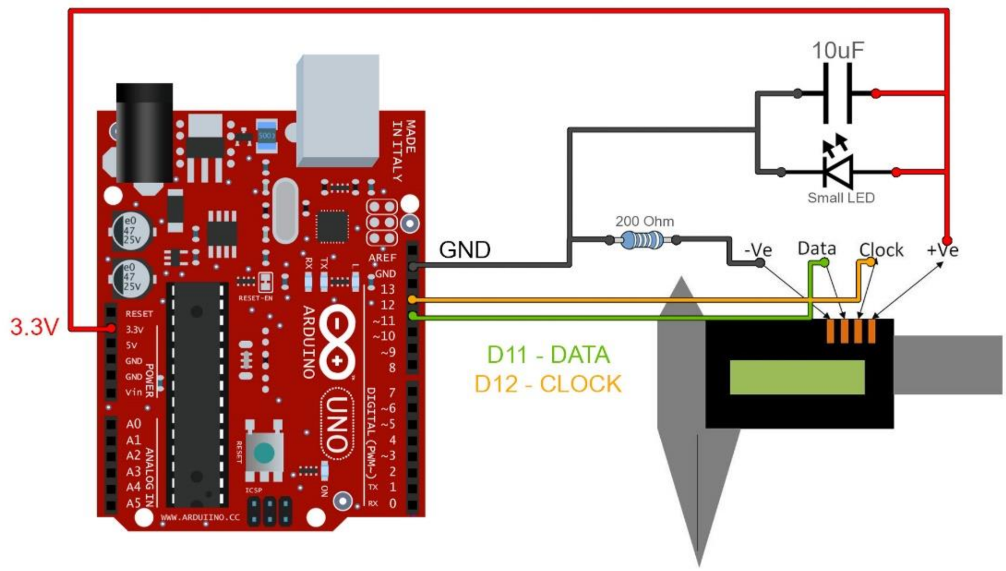

Figure 26: Connection scheme with Arduino

The Figure 27 below shows the placement of the position sensor along the X axis. The slider is fixed on the nozzle and moves along the scale. Several details are designed for 3D printing to fix position sensors (Figure 28).

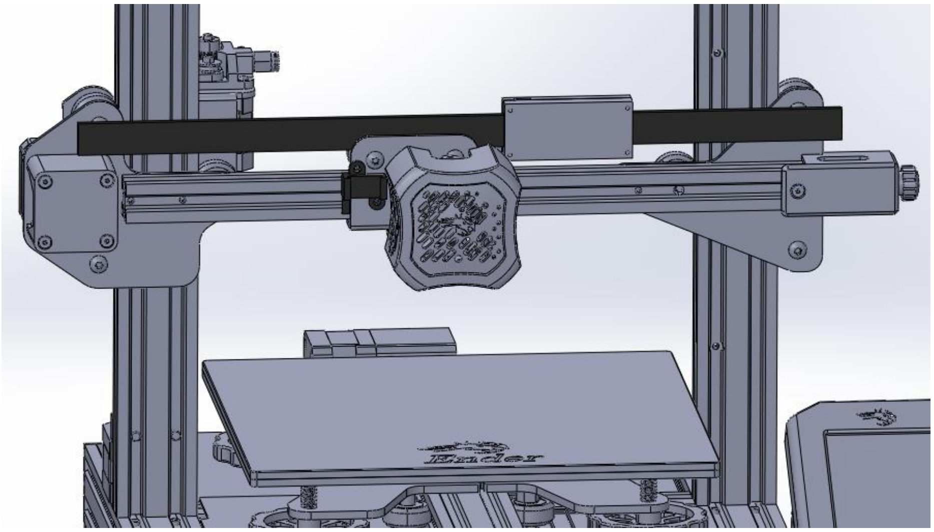

Figure 27: The CAD model of the position sensor for X-axis

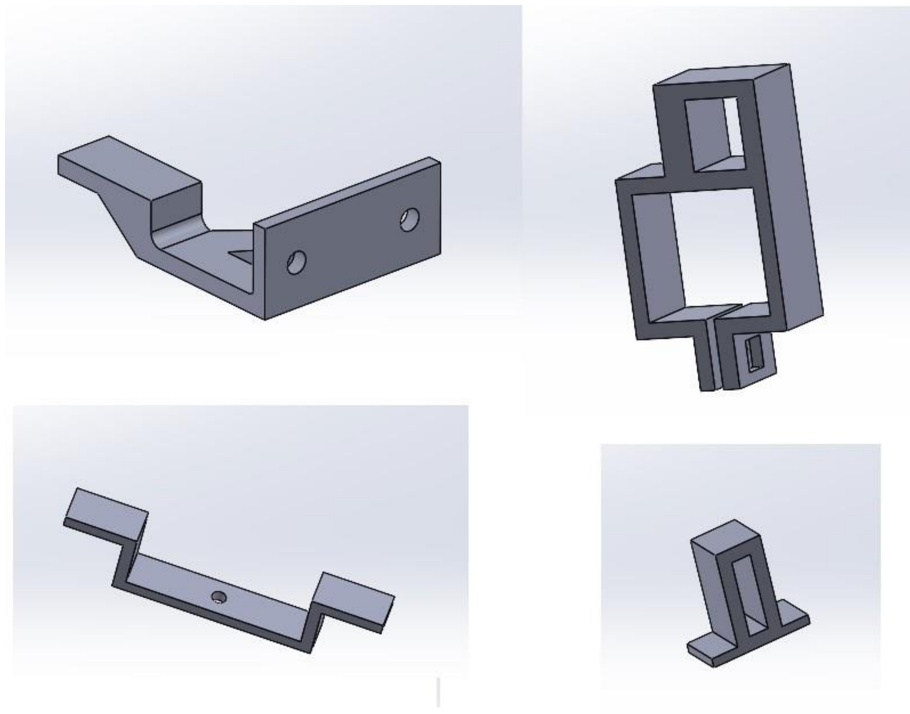

Figure 28: 3D models of the fixators

3.4. Data transfer

The one of crucial part of the system is data transfer between system components. The smooth data flow between real 3D printer and its digital twin makes the system reliable and provides satisfying user experience. Figure 29 represents the data flow between the digital twin components The 3D printer is connected to the computer via USB cable. The pyserial package in python allows to communicate with the 3D printer to retract and send data. Furthermore, Arduino modules are used to collect the sensor data and they are connected to the computer. The Python file receives data from the 3D printer and Arduino modules via a serial port and sends it to a database.

After comparing the various databases such as MySQL, Mongo DB, and Firebase, the last one was selected for this project. Firebase is mostly preferred for real-time applications, and it is easier to set up. It is a NoSQL database that syncs and stores data in real-time. While MySQL is an open-source relational database management system that uses SQL domain-specific language as its foundation. Managing enormous amounts of data is simple using Firebase.

The python also sends data from the 3D printer to the Firebase. The Unity is selected as graphical user interface. The data collected in Firebase can be easily accessed in Unity. Unity platform allows users to remotely monitor and control 3D printing process from mobile phone or laptop. All necessary data from the printer and sensors are presented in the GUI.

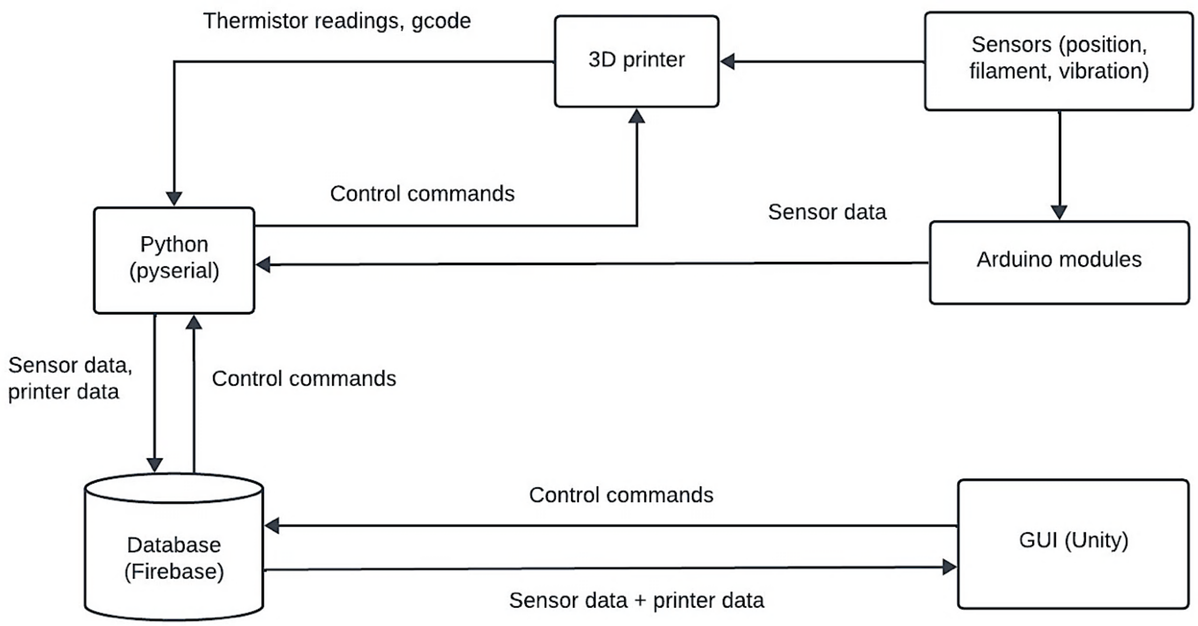

Figure 29: Data flow between system components

Chapter 4 – Results and Discussion

4.1 Sensor integration

4.1.1 Vibration

The comparison test was conducted between Sencera vibration sensor and Waveshare 10DOF IMU sensor for their ability to sense vibrations during the printing (Figure 30).

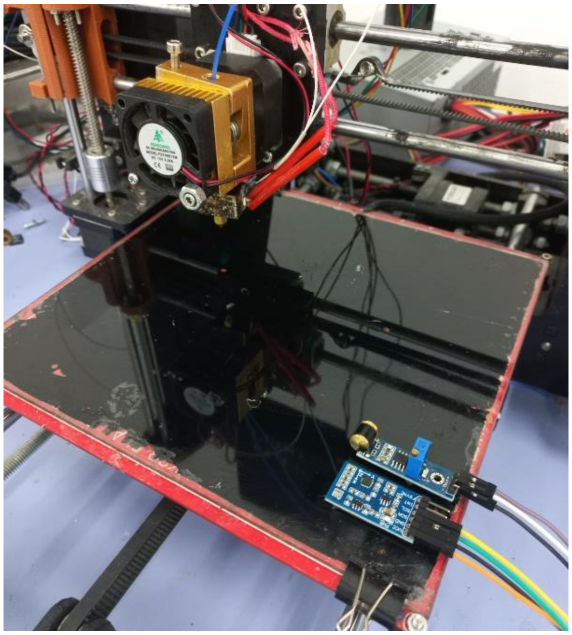

Figure 30: Vibration test setup

Both sensors were mounted on the 3D printer's bed and tested to compare their functionalities. The Arduino Uno was used to collect data from sensors and plot them on output window. Some external forces were applied to create vibrations manually during printing process.

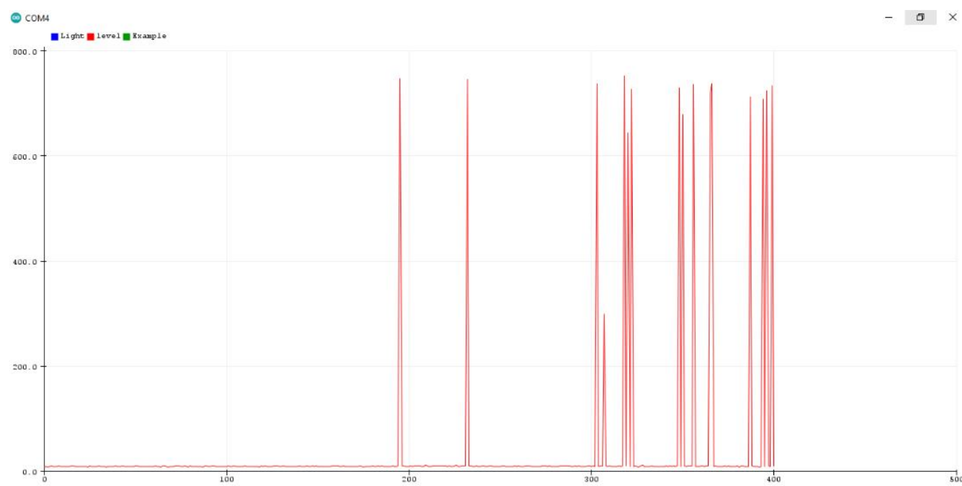

Figure 31: Output from Sencera 801S

Both sensors could sense applied vibrations as it can be seen from fluctuations. However, accelerometer showed better results as it could sense different vibration ranges. While, Sencera vibration sensor showed almost same values for different vibration ranges.

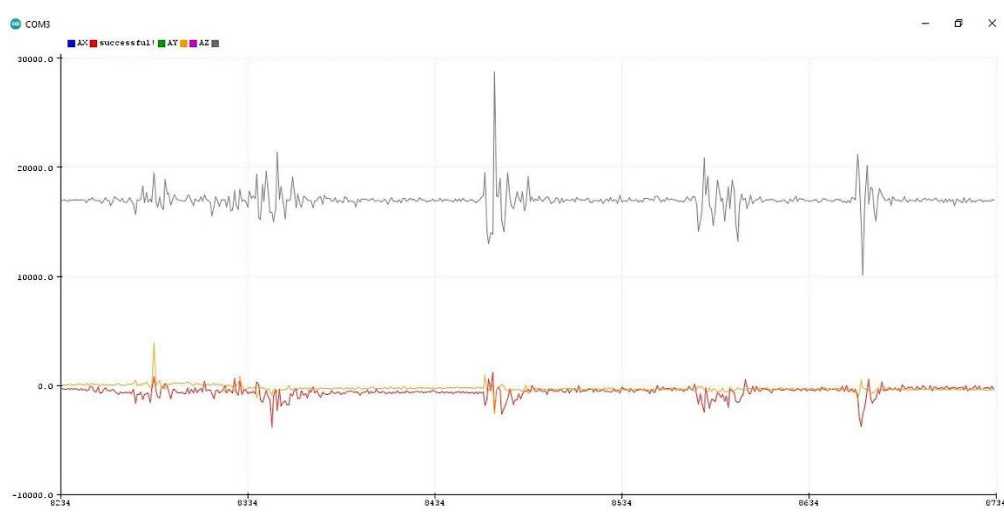

Figure 32: Output from the accelerometer

After comparing the two sensors, the WaveShare IMU sensor was selected for vibration tracking. The sensor is installed near the 3D printer nozzle as shown in Figure 33.

Figure 33: Installed vibration sensor

The vibration sensor was tested at various printing speeds. The typical print speed for most of the FDM 3D printers is 60 mm/s. Figure 34 shows acceleration values for the Z axis at 30 mm/s, 60 mm/s, and 120 mm/s. Higher print speed causes additional vibrations as can be seen from the plot. Furthermore, it causes overheating (Figure 35) as the nozzle fan could not cool the deposited layer on time. The acceleration values for typical print speed can be used to detect anomalies during 3D printing. The anomalies can occur due to issues with mechanical parts such as loose screws or belts. The typical acceleration values for 60 mm/s print speed are between 14000-19000 LSB/g according to the experiment. Furthermore, it can be used to optimize the print speed according to the vibration values.

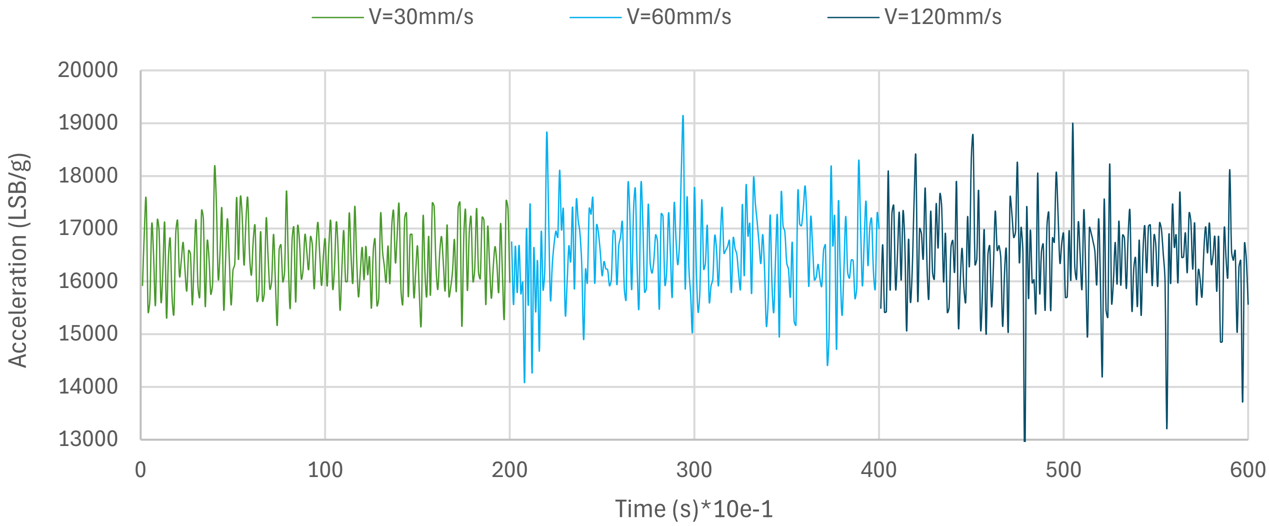

Figure 34: Vibration tracking at different printing speeds

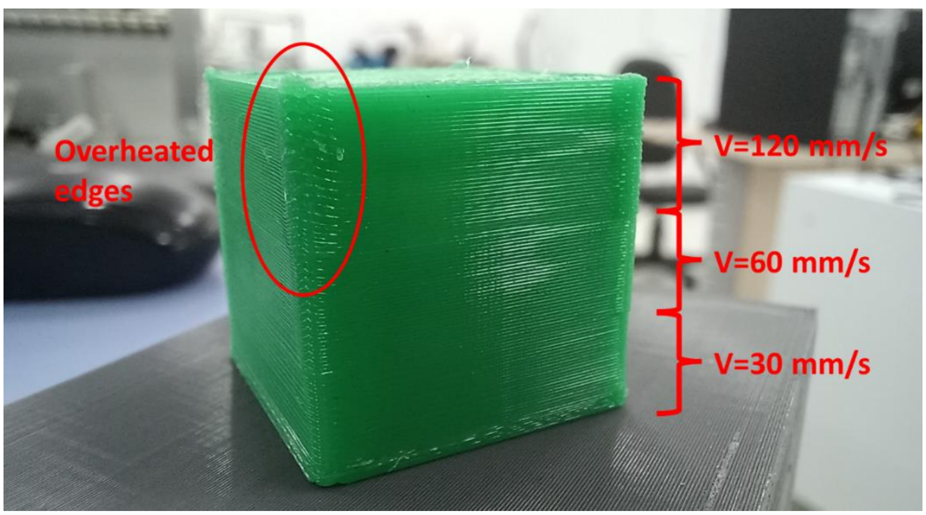

Figure 35: Sample cube printed with different print speed

4.1.2 Filament flow

The filament sensor is developed according to the CAD model shown in Figure 21. All parts of the sensor were 3D printed by using PLA filament. The bearing, gear and spring were taken from a typical FDM 3D printer’s extruder system. The final assembly of the filament sensor with optical encoders is shown in Figure 36.

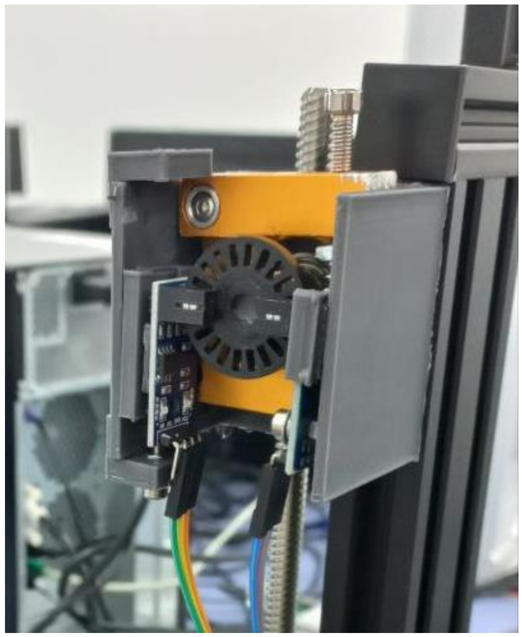

Figure 36: Filament sensor with optical encoder

The Arduino microcontroller was programmed in such way so it can detect direction of the rotation by comparing signals from two different sensors. The time difference when they receive signals should be different for clockwise and anticlockwise directions. The developed filament sensor can track length of filament used with 5mm accuracy as the disc rotates to one hole when 5mm of filament is extruded.

```
COM7

Filament length: 55
Hole Count: 12
Filament length: 60
Hole Count: 12
Filament length: 60
Hole Count: 13
Filament length: 65
Hole Count: 13
Filament length: 65
Hole Count: 14
Filament length: 70
Hole Count: 14
Filament length: 70
Hole Count: 15
Filament length: 75
```

Figure 37: Arduino output of the filament sensor

The proposed filament sensor with optical encoder showed instability during long prints and needed regular calibration. Therefore, optical encoder was replaced with magnetic encoder as shown in Figure 38. The extruder gear is connected to a shaft that rotates freely in a bearing. The magnetic encoder is also connected to this shaft and rotates when the filament is being extruded or retracted. The sensor is connected to the Arduino that converts the pulses to the length of filament in millimeters.

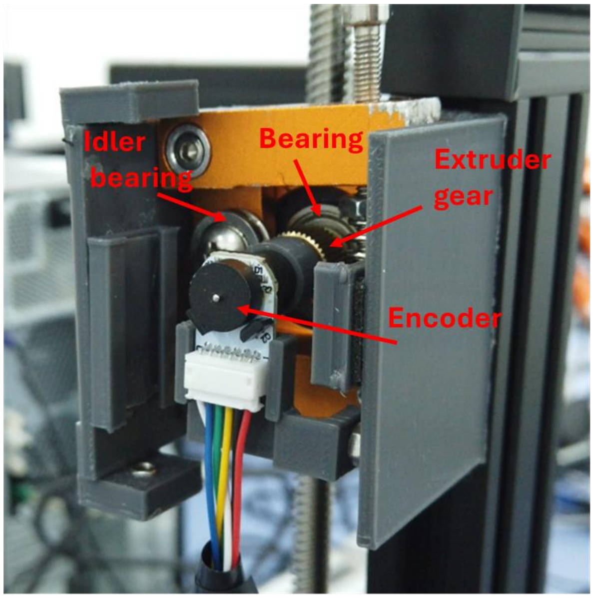

Figure 38: Filament sensor with magnetic encoder

Figure 39 shows the connection scheme of the magnetic encoder with the Arduino Uno. Motor pins are not necessary as only encoder is used. Pin A is used to count the pulses, while pin B is used to determine the direction of the rotation.

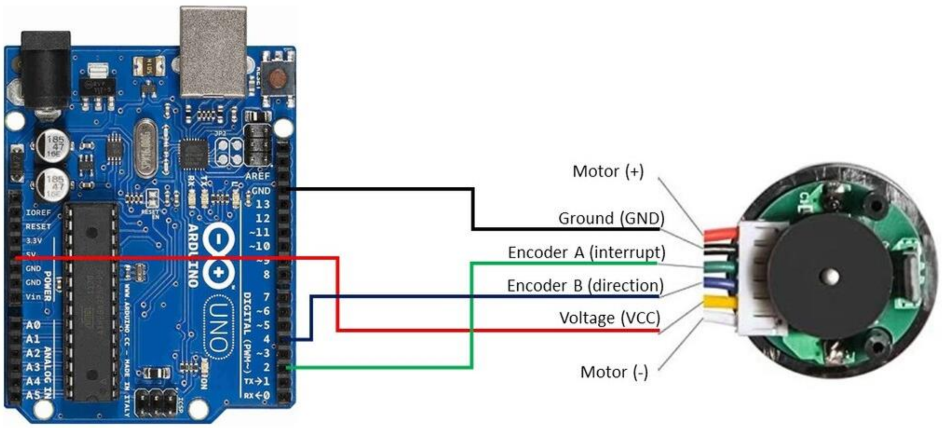

Figure 39: Encoder connection with Arduino

The filament sensor's performance was assessed by printing a sample with dimensions of 20x20x30 mm. According to the Cura slicer, the actual filament length needed for this sample is

451.2 mm. The sample was printed ten times, and the sensor readings were recorded (Figure 40). The test revealed that the filament sensor readings varied within a 16 mm range compared to the actual value (Figure 41). This difference in sensor recordings can be attributed to two factors. Firstly, the magnetic encoder's accuracy is approximately 5.3 mm, as it produces only 7 pulses per rotation. Secondly, the filament sensor was not installed right next to the extruder, resulting in a slight deviation due to the distance between the extruder and the filament sensor. Nevertheless, this test indicates that the filament sensor is capable of detecting under-extrusion defects during the printing process.

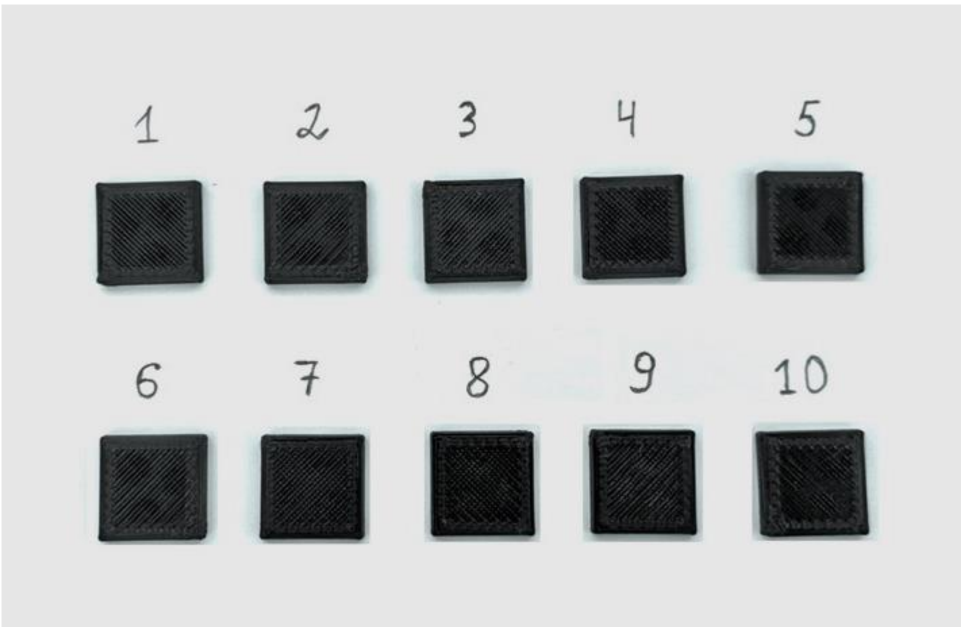

Figure 40: Printed samples for filament sensor test

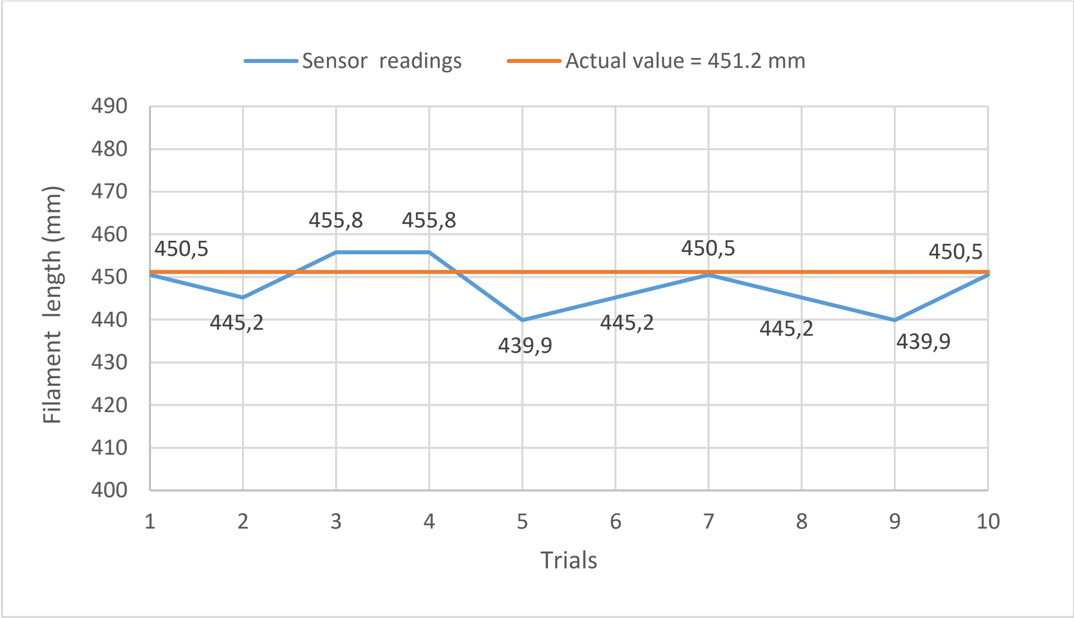

Figure 41: Filament sensor test results

4.1.3 Nozzle/bed position

The capacitive sensors are extracted from digital calipers. The case of the digital calipers are replaced with 3D modeled custom cases (Figure 42) to install them on the 3D printer. The electronic displays, batteries and buttons are not necessary as only electronic module is needed. Four wires are connected to four pins of the electronic module which are 5V, GND, ClOCK, and DATA. Those pins are connected to the Arduino to read their values.

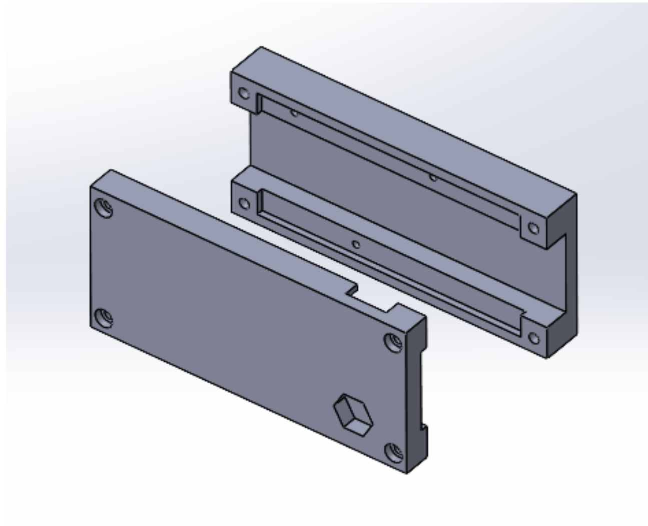

Figure 42: 3D model of the case

Figure 43: Position sensor

Those position sensors (Figure 43) are installed on the 3D printer for the X and Y axes. Two ends of each position sensor are fixed with 3D-printed connectors and bolts. The slider of the position sensor for the X axis is fixed with the nozzle, while the slider of the position sensor for the Y axis is fixed with the bed. They do not resist the movement of the 3D printer and slide freely.

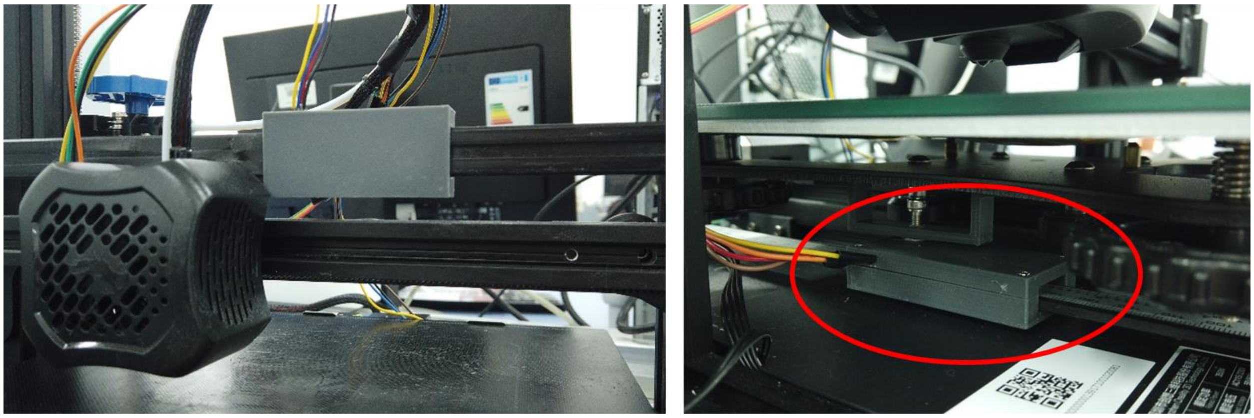

Figure 44: Installed position sensors for X and Y axes

Figure 45 shows real-time nozzle path visualization in Unity based on Gcode and position sensor readings. The black lines show the nozzle path according to the Gcode, while the red dots show sensor readings. The Python file receives sensor data from Arduino modules and sends it to the Firebase. Furthermore, it sends GCode commands to the Firebase after sending them to the 3D printer. Those data from Firebase are accessed in Unity. Unity interprets and displays this data, utilizing LineRenderer for the black lines and circular game objects for sensor data. The Python file ensures a smooth transfer of information between sensors, Firebase, and Unity. Notably, position sensor data is updated every second, resulting in a lower number of red dots at higher print speeds. Nevertheless, this setup enables real-time tracking of nozzle positions during the printing of outer layers, a critical factor in identifying layer-shifting defects.

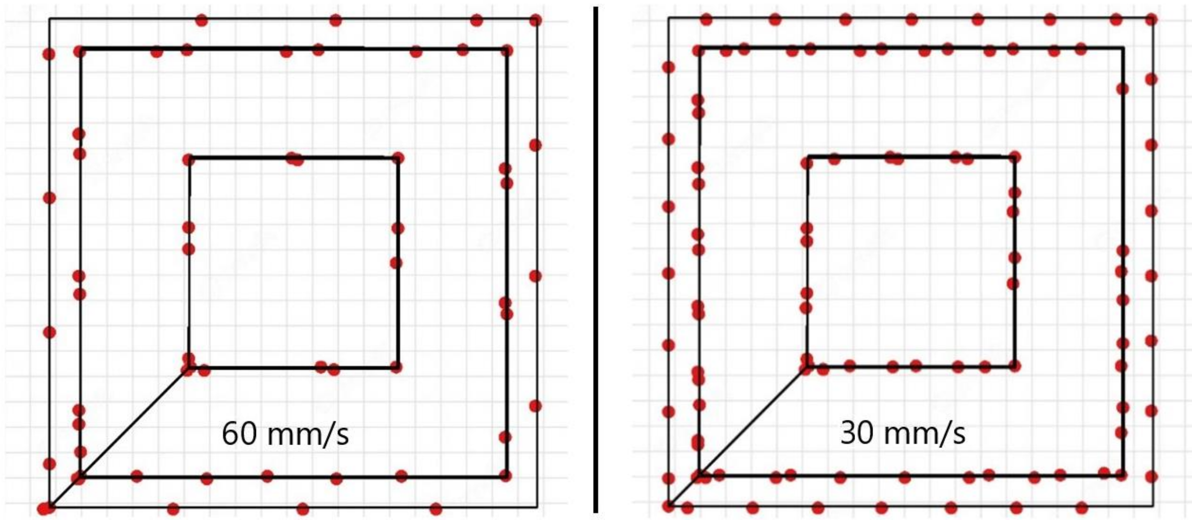

Figure 45: Nozzle path visualizations in Unity

4.2 Autonomous Defect Detection

The proposed Digital Twin system should autonomously detect a printing defect according to the sensor values. Autonomous defect detection can be done using Python directly or using Unity. In the first option, a Python script continuously monitors and compares sensor values with GCode data. It receives sensor data from Arduino modules via a serial port and reads GCode from the laptop. If an anomaly is detected in the sensor data, it should alert the user or pause the printing process. The second option utilized in this study involves using Unity for data comparison, as depicted in Figure 38. Unity retrieves data from the Firebase database and continuously checks for anomalies. If an anomaly is detected, it updates the value of the "isPaused" variable in Firebase. Simultaneously, the Python script continuously monitors the value of "isPaused", and if it changes to 1, it pauses the printing process. The three sensors were embedded in this study; therefore defects related only to those three sensors are considered.

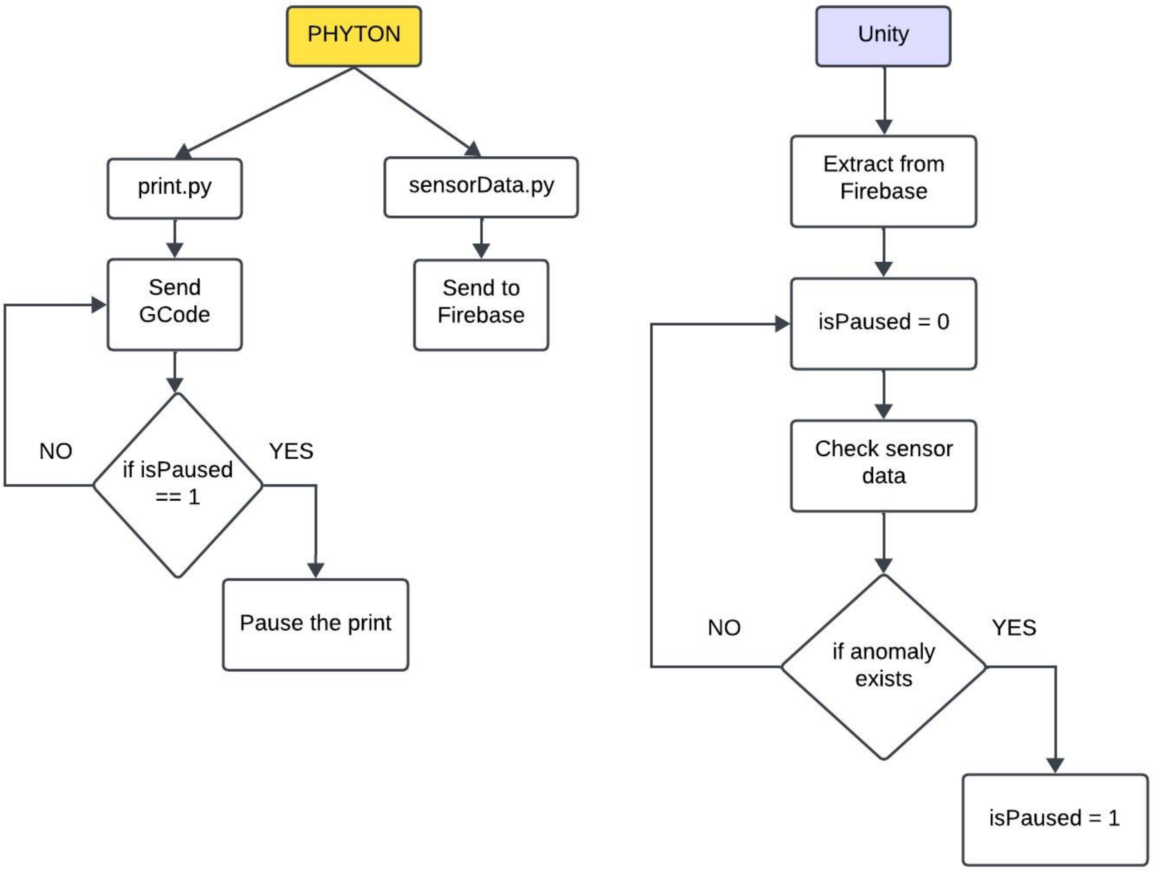

Figure 46: Defect detection control flowchart

Figure 47 shows an example of under-extrusion detection; when the difference between sensor data and Gcode is more than 15 mm it pauses the print by sending the M25 command to the 3D printer. This approach can detect all filament flow-related defects such as under extrusion, filament runout/breakage, and nozzle clogging.

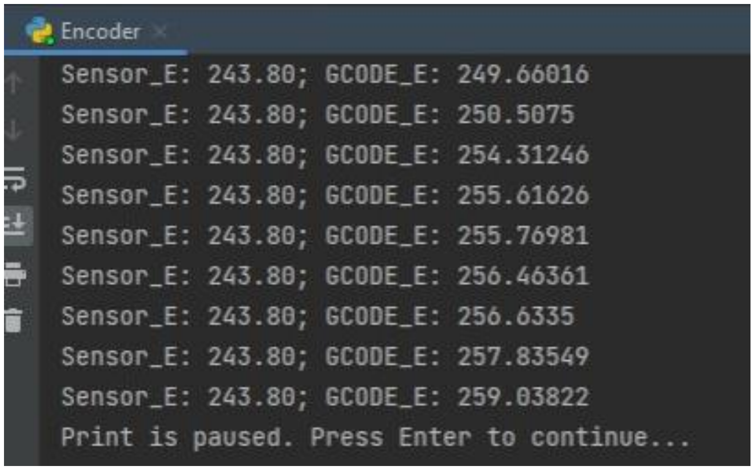

Figure 47: Under extrusion detection

The values from the accelerometer are also continuously checked by the Python file whether they fall within the typical range. If there is an anomaly in accelerometer values, the Python file pauses the print. It can detect the occurrence of mechanical faults that are causing that anomaly.

The position sensors can capture the outer walls of each layer which can be used to detect a layer shift during a print. Figure 48 shows such an example where a square formed by red dots (sensor values) is shifted to the left bottom side from the actual position shown by black lines.

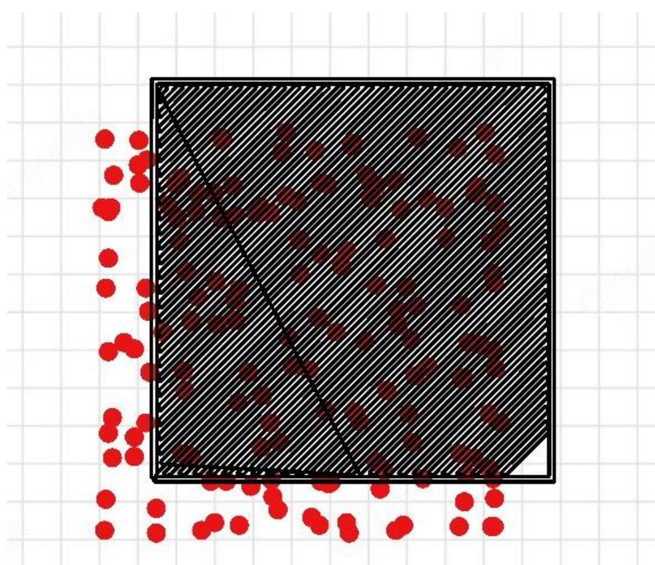

Figure 48: Layer shift detection example (V = 60 mm/s)

4.3 Data transfer and GUI

Figure 49 shows the components of the system between which data transfer occurs. Two Arduino modules are used for data acquisition. Arduino Mega is used to collect data from two position sensors (digital calipers), while Arduino Uno is used to collect data from filament sensor (magnetic encoder) and vibration sensor (accelerometer). Two Arduino modules and the 3D printer are connected to the laptop by using a USB hub. The python files communicate with Arduino modules and the 3D printer by using three different USB ports.

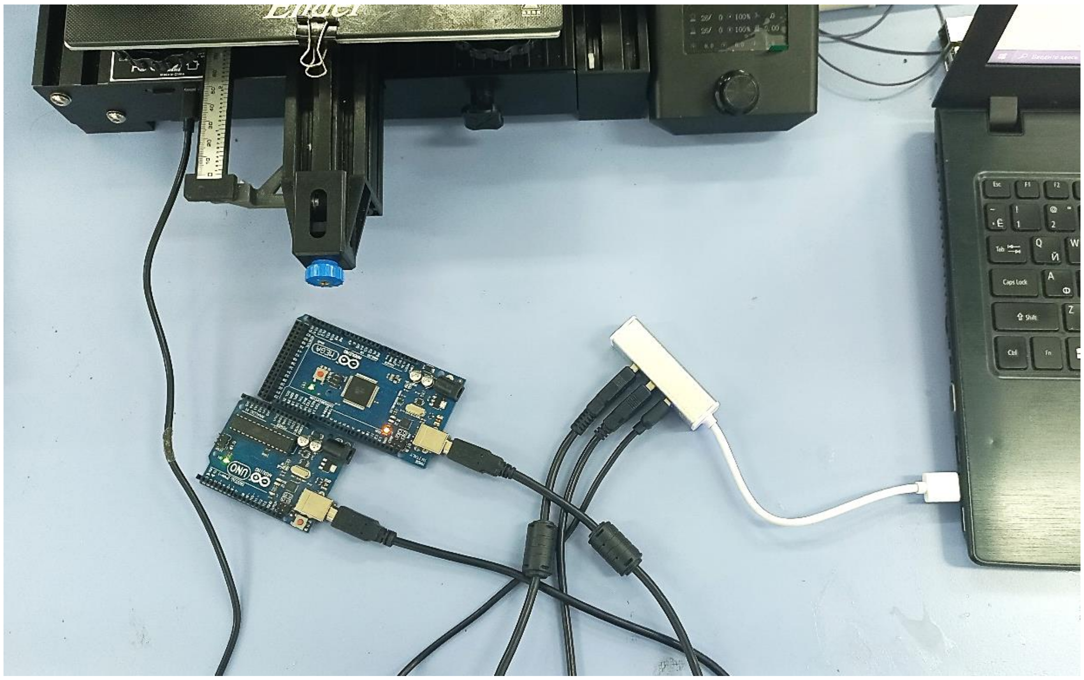

Figure 49: Data transfer components

The Firebase is used as a real-time database to store data from the printer and embedded sensors. In Firebase, two documents were created which are “PrinterData” and “SensorData” as shown in Figure 50. Each document has a collection of data. “PrinterData” has 10 data consisting of 4 data extracted from built-in thermistors and 6 data from Gcode lines. In comparison, “SensorData” has only 5 data obtained from vibration, filament, and position sensors. The Python file sends GCode to the 3D printer line by line. After sending the GCode line to the 3D printer, it should send it also to the database. However, a 3D printer can execute multiple GCode line commands within one second. In contrast, Python cannot transmit several GCode lines to the database in such a short amount of time, thereby slowing down the 3D printer. To solve this problem, a buffering method was used to collect GCode commands and send them to the database every second. Each second, a different amount of data is received by the database without interrupting the 3D printer. The data in the database is used for visualization in Unity.

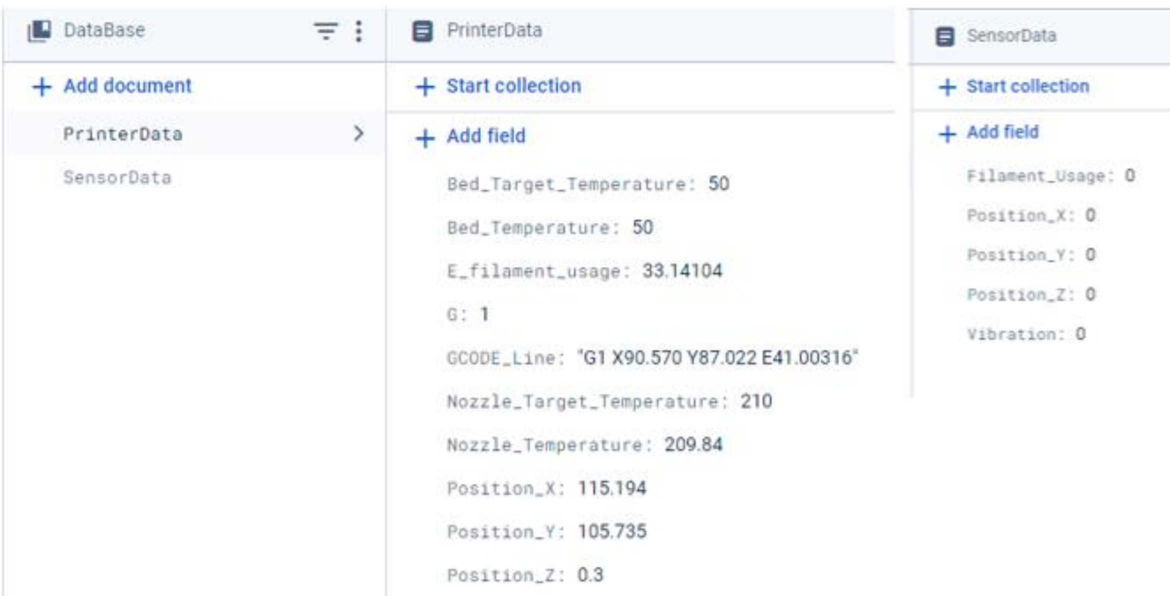

Figure 50: Firebase window

Data transfer between DT system components was successfully developed. Printer data are extracted by using the "pyserial" package in Python and can be sent to Firebase by using "firebase" packages. The Unity 3D can access Firebase data by using Firebase Unity SDK. Figure 51 shows a Graphical User Interface built in Unity. The interface contains all the necessary information about the print state. Furthermore, there are control commands available for a user like pause or cancel. The sensor data are also presented in the GUI to provide ground truth data. Vibration data is visualized in line chart form, while the nozzle path is visualized for each layer with black lines (GCode) and red dots (sensor data). Nozzle movement is visualized with a 3D copy of the 3D printer. However, only GCode coordinates are used for smooth movement of the nozzle as sensor data are not evenly distributed.

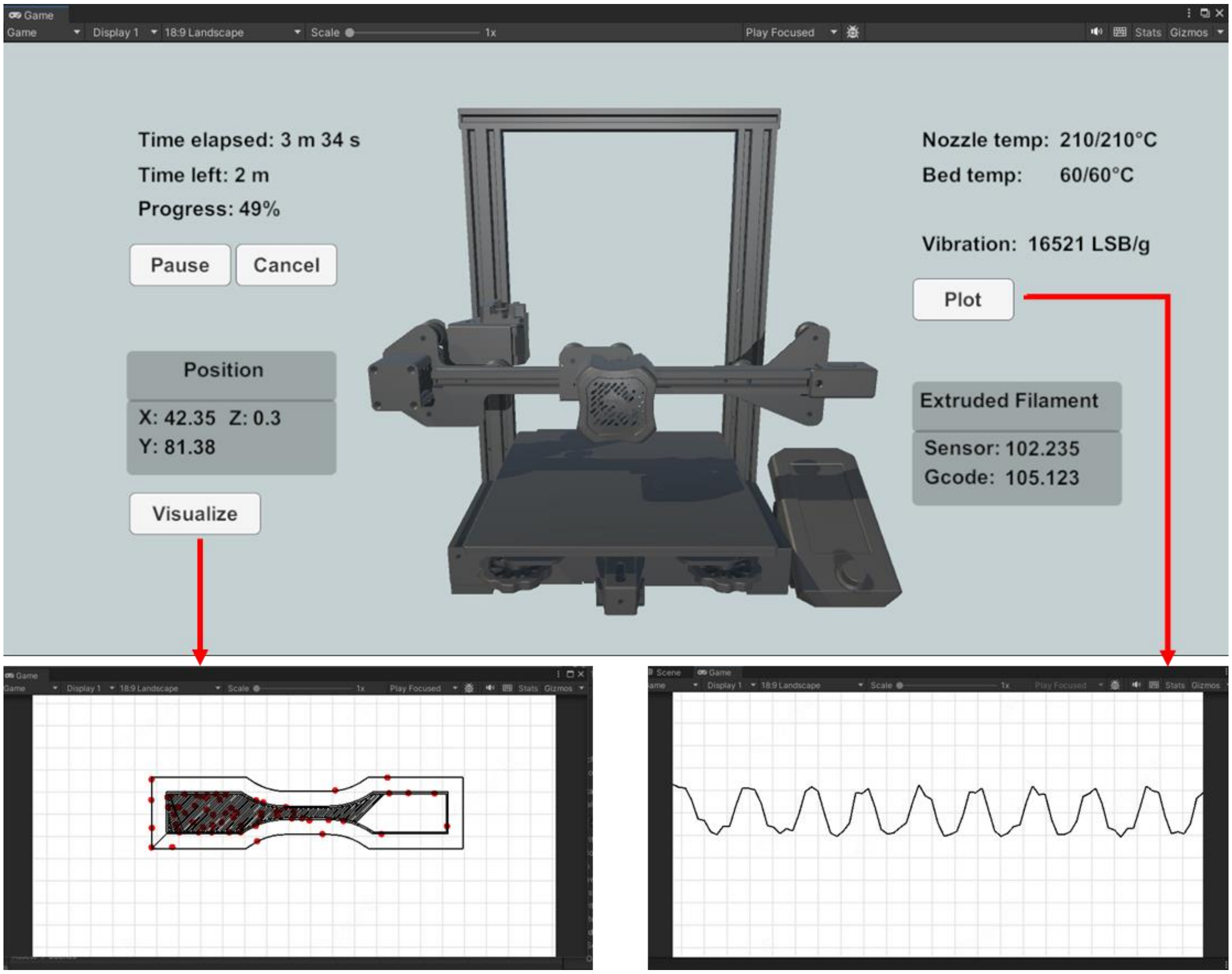

Figure 51: GUI in Unity 3D

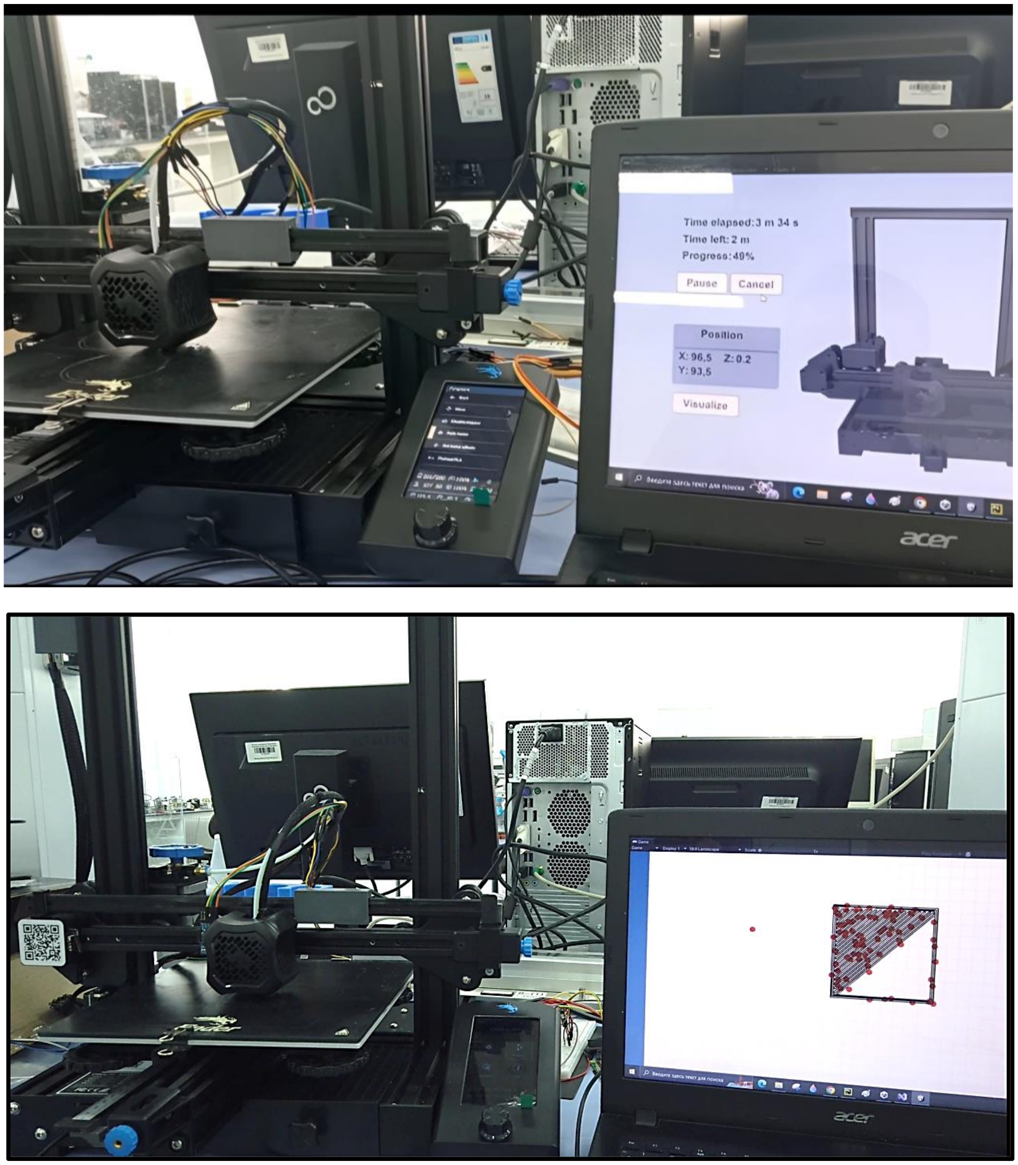

Figure 52: 3D printer and GUI in Unity

4.4 Discussion

The work offers a detailed examination of Digital Twins (DTs) customized specifically for the field of additive manufacturing. This study aims to tackle the limitations and obstacles faced in traditional Fused Deposition Modeling (FDM) 3D printing methods. Previous studies have mainly concentrated on specific functions of DTs for 3D printing, including monitoring process variables and detecting defects. However, these studies often neglect important printing parameters, rationale for sensor selection, and optimization of the printing process. Addressing these research gaps, this research studies the DT technology by preferring a comprehensive approach that covers data collection, sensor integration, real-time monitoring, and defect detection strategies. The proposed DT system shows potential in improving the efficiency, quality, and dependability of FDM 3D printing procedures through the integration of vibration sensors, custom-designed filament flow sensors, and nozzle/bed position sensors, with advanced data analysis methods and user-friendly interfaces.

Additionally, the results of the experiments demonstrate the efficiency and practicality of the suggested DT system in real-life scenarios. By conducting thorough tests and verification processes, the research illustrates the capabilities of the developed sensors in monitoring important printing parameters, identifying irregularities, and facilitating automated detection of defects. Notable achievements include the successful integration of vibration sensors for spotting anomalies, filament flow sensors for tracking materials, and nozzle/bed position sensors for monitoring the progress of printing in real-time. These experimental results not only confirm the effectiveness of the proposed DT system but also emphasize its potential to transform FDM 3D printing procedures by providing useful insights, improving quality control protocols, and enabling predictive maintenance strategies.

Chapter 5 – Conclusion and Future Works

5.1 Conclusions

In this research thesis, a sensor-based digital twin for FDM 3D printers is presented for real-time monitoring, control, and autonomous defect detection. The data transfer method for the DT is proposed that provides a smooth connection between of 3D printer, sensors, database, python, and Graphical User Interface. The proposed data acquisition system incorporates the use of affordable but accurate sensors to track printing parameters. Each sensor was tested and evaluated to verify its use in the proposed DT system. The application of Python allows manual and autonomous control of 3D printers according to the sensor values. In the proposed DT system, python sends GCode commands line by line which allows editing GCode commands in real time for optimization purposes. The Graphical User Interface built in Unity provides all necessary data about the 3D printer from the printer itself and the external sensors as well. Key research results include successful integration of sensor data with printer data, selection of appropriate database and GUI platforms, and implementation of real-time control, monitoring and autonomous defect detection capabilities.

5.2 Key Research Contribution to Knowledge

The current challenges that FDM 3D printing technology is facing include the need for continuous real-time monitoring and control during extended printing periods, as well as the implementation of an automatic defect detection system to conserve resources such as time, money, and materials. These challenges can be effectively tackled through the adoption of digital twin technology. The digital twin model developed in this study offers several valuable contributions to the existing knowledge. Firstly, it suggests use of affordable and accurate sensors for filament counting and nozzle position monitoring. Secondly, it proposes data transfer method between 3D printer, python, Arduino with sensors, and database. Lastly, it suggests transfer of GCode line by line that allows real time defect detection and optimization.

5.3 Future Works

This future work incorporates the optimization of printing parameters according to the sensor values. Machine learning algorithms can be used for that purpose. Optimization can be done by editing Gcode in real time according to the sensor values and machine learning suggestions. Furthermore, research effort is needed to employ machine learning in analyzing sensor data and predicting possible printing defects. The Graphical User Interface can be improved by adding material extrusion visualization based on the filament sensor data. Further study is required to integrate augmented reality into the developed digital twin.

References

Bhagia, S., Bornani, K., Agrawal, R., Satlewal, A., Ďurkovič, J., Lagaňa, R., Bhagia, M., Yoo, C. G., Zhao, X., Kunc, V., Pu, Y., Ozcan, S., & Ragauskas, A. J. (2021). Critical review of FDM 3D printing of PLA biocomposites filled with biomass resources, characterization, biodegradability, upcycling and opportunities for biorefineries. *Applied Materials Today, 24,* 101078. https://doi.org/10.1016/j.apmt.2021.101078

Boschetto, A., Giordano, V., & Veniali, F. (2013). Surface roughness prediction in fused deposition modelling by neural networks. *The International Journal of Advanced Manufacturing Technology*, *67*(9–12), 2727–2742. https://doi.org/10.1007/s00170-012-4687-x

Butt, J., & Mohaghegh, V. (2022). Combining Digital Twin and Machine Learning for the Fused Filament Fabrication Process. *Metals*, *13*(1), 24. https://doi.org/10.3390/met13010024

Ceruti, A., Liverani, A., & Bombardi, T. (2017). Augmented vision and interactive monitoring in 3D printing process. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, *11*(2), 385–395. https://doi.org/10.1007/s12008-016-0347-y

Chhetri, S. R., Faezi, S., Canedo, A., & Faruque, M. A. A. (2019). QUILT: Quality inference from living digital twins in IoT-enabled manufacturing systems. *Proceedings of the International Conference on Internet of Things Design and Implementation*, 237–248. https://doi.org/10.1145/3302505.3310085

Corradini, F., & Silvestri, M. (2022). Design and testing of a digital twin for monitoring and quality assessment of material extrusion process. *Additive Manufacturing*, *51*, 102633. https://doi.org/10.1016/j.addma.2022.102633

Farhan Khan, M., Alam, A., Ateeb Siddiqui, M., Saad Alam, M., Rafat, Y., Salik, N., & Al-Saidan, I. (2021). Real-time defect detection in 3D printing using machine learning. *Materials Today: Proceedings*, *42*, 521–528. https://doi.org/10.1016/j.matpr.2020.10.482

Gf, B., & Rv, A. (2017). An IoT-Based Solution for Control and Monitoring of Additive Manufacturing Processes. *Journal of Powder Metallurgy & Mining*, *06*(01). https://doi.org/10.4172/2168-9806.1000158

Grant Trebbin: Digital Calliper Teardown and Repair. (n.d.). Retrieved November 19, 2023, from https://www.grant-trebbin.com/2014/04/digital-calliper-teardown-and-repair.html

Grieves, M. (2011). *Virtually perfect: Driving innovative and lean products through product lifecycle management*. Space Coast Press.

Guo, L., Cheng, Y., Zhang, Y., Liu, Y., Wan, C., & Liang, J. (2021). Development of Cloud-Edge Collaborative Digital Twin System for FDM Additive Manufacturing. *2021 IEEE 19th International Conference on Industrial Informatics (INDIN)*, 1–6. https://doi.org/10.1109/INDIN45523.2021.9557492

Henson, C. M., Decker, N. I., & Huang, Q. (2021). A digital twin strategy for major failure detection in fused deposition modeling processes. *Procedia Manufacturing*, *53*, 359–367. https://doi.org/10.1016/j.promfg.2021.06.039

Jin, Z., Zhang, Z., & Gu, G. X. (2019). Autonomous in-situ correction of fused deposition modeling printers using computer vision and deep learning. *Manufacturing Letters*, *22*, 11–15. https://doi.org/10.1016/j.mfglet.2019.09.005

Kakade, S., Mulay, A., & Patil, S. (2022). IoT-based real-time online monitoring system for open ware FDM printers. *Materials Today: Proceedings*, *67*, 363–367. https://doi.org/10.1016/j.matpr.2022.07.210

Kantaros, A., Piromalis, D., Tsaramirsis, G., Papageorgas, P., & Tamimi, H. (2021). 3D Printing and Implementation of Digital Twins: Current Trends and Limitations. *Applied System Innovation*, *5*(1), 7. https://doi.org/10.3390/asi5010007

Kazhymurat, T., Shehab, E., & Ali, Md. H. (2022). IoT-Based Real-Time 3D Printing Monitoring System. *2022 International Conference on Smart Information Systems and Technologies (SIST)*, 1–5. https://doi.org/10.1109/SIST54437.2022.9945778

Korolov, M. (2022, October 27). *How to Secure Digital Twin Technology in Your Data Center*. https://www.datacenterknowledge.com/security/how-secure-digital-twin-technology-your-data-center

Li, Z., Rathore, A. S., Song, C., Wei, S., Wang, Y., & Xu, W. (2018). PrinTracker: Fingerprinting 3D Printers using Commodity Scanners. *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, 1306–1323. https://doi.org/10.1145/3243734.3243735

Lim, K. Y. H., Zheng, P., & Chen, C.-H. (2020). A state-of-the-art survey of Digital Twin: Techniques, engineering product lifecycle management and business innovation perspectives. *Journal of Intelligent Manufacturing*, *31*(6), 1313–1337. https://doi.org/10.1007/s10845-019-01512-w

Moretti, M., Rossi, A., & Senin, N. (2021). In-process monitoring of part geometry in fused filament fabrication using computer vision and digital twins. *Additive Manufacturing*, *37*, 101609. https://doi.org/10.1016/j.addma.2020.101609

Mourtzis, D., Togias, T., Angelopoulos, J., & Stavropoulos, P. (2021). A Digital Twin architecture for monitoring and optimization of Fused Deposition Modeling processes. *Procedia CIRP*, *103*, 97–102. https://doi.org/10.1016/j.procir.2021.10.015

Pantelidakis, M., Mykoniatis, K., Liu, J., & Harris, G. (2022). A digital twin ecosystem for additive manufacturing using a real-time development platform. *The International Journal of Advanced Manufacturing Technology*, *120*(9–10), 6547–6563. https://doi.org/10.1007/s00170-022-09164-6

Paripooranan, C. S., Abishek, R., Vivek, D. C., & Karthik, S. (2020). An Implementation of AR Enabled Digital Twins for 3-D Printing. *2020 IEEE International Symposium on Smart Electronic Systems (iSES) (Formerly iNiS)*, 155–160. https://doi.org/10.1109/iSES50453.2020.00043

Rachmawati, S. M., Putra, M. A. P., Lee, J. M., & Kim, D. S. (2023). Digital twin-enabled 3D printer fault detection for smart additive manufacturing. *Engineering Applications of Artificial Intelligence*, *124*, 106430. https://doi.org/10.1016/j.engappai.2023.106430

Rao, P. K., Liu, J. (Peter), Roberson, D., Kong, Z. (James), & Williams, C. (2015). Online Real-Time Quality Monitoring in Additive Manufacturing Processes Using Heterogeneous Sensors. *Journal of Manufacturing Science and Engineering*, *137*(6), 061007. https://doi.org/10.1115/1.4029823

Salifu, S., Ogunbiyi, O., & Olubambi, P. A. (2022). Potentials and challenges of additive manufacturing techniques in the fabrication of polymer composites. *The International Journal of Advanced Manufacturing Technology*, *122*(2), 577–600. https://doi.org/10.1007/s00170-022-09976-6

Sampedro, G. A. R., Agron, D. J. S., Amaizu, G. C., Kim, D.-S., & Lee, J.-M. (2022). Design of an In-Process Quality Monitoring Strategy for FDM-Type 3D Printer Using Deep Learning. *Applied Sciences*, *12*(17), 8753. https://doi.org/10.3390/app12178753

Scheffel, R. M., Fröhlich, A. A., & Silvestri, M. (2021). Automated fault detection for additive manufacturing using vibration sensors. *International Journal of Computer Integrated Manufacturing*, *34*(5), 500–514. https://doi.org/10.1080/0951192X.2021.1901316

Syrlybayev, D., Zharylkassyn, B., Seisekulova, A., Perveen, A., & Talamona, D. (2021). Optimization of the Warpage of Fused Deposition Modeling Parts Using Finite Element Method. *Polymers*, *13*(21), 3849. https://doi.org/10.3390/polym13213849

Tao, F., Xiao, B., Qi, Q., Cheng, J., & Ji, P. (2022). Digital twin modeling. *Journal of Manufacturing Systems*, *64*, 372–389. https://doi.org/10.1016/j.jmsy.2022.06.015

Yi, L., Glatt, M., Ehmsen, S., Duan, W., & Aurich, J. C. (2021). Process monitoring of economic and environmental performance of a material extrusion printer using an augmented reality-based digital twin. *Additive Manufacturing*, *48*, 102388. https://doi.org/10.1016/j.addma.2021.102388

D. Vaes and P. Van Puyvelde, “Semi-crystalline feedstock for filament-based 3D printing of polymers,” Prog. Polym. Sci., vol. 118, p. 101411, Jul. 2021, doi: 10.1016/j.progpolymsci.2021.101411

3D filament meter caliper Arduino i2c tutorial. (n.d.). https://electronoobs.com/eng_arduino_tut93.php

Appendices

Arduino code for accelerometer

```
#include <MPU9255.h>//include MPU9255 library

MPU9255 mpu;

char report[80];

void setup()
{
  Serial.begin(9600);

  if(mpu.init())
  {
  Serial.println("initialization failed");
  }
  else
  {
  Serial.println("initialization successful!");
  }

// To excel initializtion
  Serial.println("CLEARDATA"); // очистка листа excel
  Serial.println("LABEL, Time,Bed_X, Bed_Y, Bed_Z"); // заголовки столбцов
  //Serial.println("RESETTIMER");
}

void loop()
{


  mpu.read_acc();//get data from the accelerometer
  //mpu.read_gyro();//get data from the gyroscope
//  mpu.read_mag();//get data from the magnetometer

Serial.print("DATA, TIME");

 // snprintf(report, sizeof(report), "A: %6d %6d %6d",
 //   imu.a.x, imu.a.y, imu.a.z);
//  Serial.println(report);

Serial.print(" ,");
Serial.print(mpu.ax);
Serial.print(" ,");
Serial.print(mpu.ay);
Serial.print(" ,");
Serial.print(mpu.az);
Serial.println(" ,");


  delay(200);
}
```

Arduino code for filament sensor

```
#define Encoder_output_A 2 // pin2 of the Arduino
#define Encoder_output_B 4 // pin 3 of the Arduino
// these two pins has the hardware interrupts as well.

int Count_pulses = 0;
void setup() {
Serial.begin(9600); // activates the serial communication
pinMode(Encoder_output_A,INPUT); // sets the Encoder_output_A pin as the input
pinMode(Encoder_output_B,INPUT); // sets the Encoder_output_B pin as the input
attachInterrupt(digitalPinToInterrupt(Encoder_output_A),DC_Motor_Encoder,RISING);
}

void loop() {
  Serial.println(Count_pulses*5.3);
  delay(1000);
}

void DC_Motor_Encoder(){
  int b = digitalRead(Encoder_output_B);
  if(b > 0){
    Count_pulses++;


  }
  else{
    Count_pulses--;


  }
```

Arduino code for position sensor

```
#define CLOCK_PIN 12
#define DATA_PIN  11


void setup()
{
  Serial.begin(9600);
  pinMode(CLOCK_PIN, INPUT);
  pinMode(DATA_PIN, INPUT);

}

char buf[20];
unsigned long tmpTime;
int sign;
int inches;
long value;
float result;
bool mm = true; //define mm to false if you want inces values

void loop()
{
  while(digitalRead(CLOCK_PIN)==LOW) {}
  tmpTime=micros();
  while(digitalRead(CLOCK_PIN)==HIGH) {}
  if((micros()-tmpTime)<500) return;
  readCaliper();
  buf[0]=' ';
  dtostrf(result,6,3,buf+1); strcat(buf," in ");
  dtostrf(result*2.54,6,3,buf+1); strcat(buf," cm ");

  if(mm)
  {
    Serial.print(result); Serial.println(" mm");
    delay(100);
  }
  else
  {
    Serial.print(result); Serial.println(" in");
    delay(100);
  }
}
void readCaliper()
{
  sign=1;
  value=0;
  inches=0;
  for(int i=0;i<24;i++) {
    while(digitalRead(CLOCK_PIN)==LOW) {}
    while(digitalRead(CLOCK_PIN)==HIGH) {}
    if(digitalRead(DATA_PIN)==HIGH) {
    if(i<20) value|=(1<<i);
    if(i==20) sign=-1;
```

```
    if(i==23) inches=1;
  }
}
if(mm)
{
  float multiplier = pow(10, 1);
  float tempvalue=-(value*sign)/100.0;
  result=-round(tempvalue* multiplier) / multiplier;
}
else
{
result=(value*sign)/(inches?2000.0:100.0); //We map the values forr
}
```