UNIVERSITA' DEGLI STUDI DI PARMA

DOTTORATO DI RICERCA IN
INGEGNERIA INDUSTRIALE

CICLO XXXIV

DESIGNING, PROTOTYPING AND TESTING OF THE DIGITAL TWIN OF A 3D PRINTER TO MEET INDUSTRY 4.0 REQUIREMENTS.

Coordinatore:
Chiar.mo Prof. Gianni Royer Carfagni

Tutore:
Chiar.mo Prof. Marco Silvestri

Dottorando: Fabio Corradini

Anni Accademici 2018/2019 - 2020-2021

Table of Contents

Abstract

This thesis aims to explain the structure and operation of a multifunctional digital twin for a material extrusion 3D printer. Among its functionalities, the digital twin includes an advanced remote control and monitoring of the printing process, the ability to collect and sort job data, run simulations, react to errors and identify their cause, evaluate the geometric accuracy of the printed piece, monitor machine wear, and self-calibrate. The structure of the twin is modular, it has been built entirely with open-source software and does not rely on proprietary solutions so that it is easily replicable and extendable to different systems.

The printer used for testing is a Cartesian material extrusion printer equipped with encoders, accelerometers, additional thermometers, and a machine vision system. The data collected by the sensors during printing is sent to a data interface that sorts it into the twin core, and a remote storage system. The digital twin core houses the simulation of the printer's physics and control logic and performs simulation, monitoring, and diagnostic functions. This module is also capable of generating a detailed representation of the printed model in real-time from data collected during processing. This representation can be used to evaluate the geometric conformity of the obtained piece compared to an ideal printing model, all carried out through an automatic procedure.

An additional part of the digital twin is a GUI, that is intended to allow the user to remotely interact with the system, view collected data, and perform post-processing operations. Both the core and the GUI are able to communicate with the printer through a print host software, so that the system can secure the printer automatically, if an error occurs, and guide the user in a procedure to recover the interrupted job; moreover, the digital twin can perform an auto-calibration procedure, where error situations are created in a controlled way with the aim of identifying the working limits.

The functionalities of the digital twin are verified with a series of tests carried out at different printing speeds, both in simulation and in real prints. The simulations are able to predict the impact of the settings used on the duration of the print, on the axes load, on the wear of the components, and on the dimensions of the piece; on the other hand, from the real prints it is possible to obtain the actual data, the evaluation of the geometric conformity of the pieces, and the virtual model of the printed parts. Through the analysis of the latter, it is possible to investigate the influence of the chosen settings and of the printer structure on the geometric characteristics of the parts.

Chapter 1

Introduction

1.1 Industry 4.0

The term "Industrie 4.0" was first used in 2011 in Germany at the Hannover Trade Fair [1] to name the paradigm shift that was already possible to observe in the most advanced manufacturing industry and envisage a number of strategic goals from that perspective [2]. Subsequently, the concept was picked up by most developed countries and became known in the English version of "Industry 4.0" [3].

The term Industry 4.0 refers to the fact that the new technologies increasingly used at the industrial level are causing a change that is upsetting the way of conceiving the organization of production and, potentially, society itself. This can be compared to the impact that the industrial revolutions of the past centuries have had. In fact, every past industrial revolution was characterized by a progressive accumulation of technical innovations that leaded to a massive change in the technological mode, logistic and manufactured products [4, 5]

- **First revolution**: it took place during the 18-19th centuries due to the invention of steam engine and the improvement of mechanical loom. This forced a change in production technology from being based on manual labor to being based on industry, and required the development of a new logistics (steam transportation) since production is no longer distributed in private workers' homes but centralized in a factory [6].

- **Second revolution**: the second industrial revolution happened in the late 19th and early 20th centuries when the development of new technologies (steel, electricity, chemical industry) reached a critical point. Industrial production had to evolve according to the precepts of the division of labor and the conveyor production, while massive investments were needed in infrastructure for transport (rail and road).

- **Third revolution**: the third industrial revolution began at the end of the 20th century thanks to the advent first of electronic control technologies and then digital ones, and the development of information technology. It is characterized by major investments in the wired network infrastructure and relocation of production. Production technologies became more flexible thanks to automatic controls, so that the machines could support people in increasingly qualified tasks.

- **Fourth revolution**: the great changes that are taking place at the production level in most parts of the world have similarities with what happened during the previous industrial revolutions. The great innovations in the field of communication and information technology have allowed the development of the concept of Internet of Things and Cyber Physical System while the advent of Artificial Intelligence (AI) makes possible the analysis of large amounts of data that assume a significant value. The infrastructure needed to fully employ these new production paradigms is mainly related to the high-speed wireless network, and is still under development. The technological mode that is in construction tends to the total automation of the production starting from the manufacturing to the decisional part up to real factories without personnel (the so-called dark factories or lights out factories) [3]. The production continues the trend to be more and more flexible and distributed thanks to new additive manufacturing techniques. One of the differences compared to the revolutions of the past is that this was predicted a priori instead of observed a posteriori [6]. In the past, the nations that were able to better adopt the innovations brought by the previous revolutions were then leaders at an economic and political level for the following decades. For this reason, in many countries measures are being taken to better identify the challenges and the opportunities that this global change will bring and an huge amount of investment has been allocated [7].

The importance assumed by the digital transition can be well demonstrated by the fact that, worldwide, from 2011 to 2020, as many as 117 development initiatives have been launched in this direction by 56 countries [8]. The development of Industry 4.0 is based on a few key concepts that can be found recurrently in the literature, such as the **Internet of Things**, **Big Data Analytics**, **the cloud**, **autonomous robots**, **simulation**, **Augmented Reality**, **horizontal and vertical system integration**, **Additive Manufacturing**, and **cybersecurity** [9]. They are all related to the development of communications systems and globalization.

The development of communications infrastructure and the presence of cheap digital devices has greatly simplified the collection and distribution of data, as well as facilitated the possibility of remote control. The network, therefore, is no longer understood as an interface used by humans to exchange information but can be used to control and transmit data between automated systems (machine-to-machine) or between automated systems and users (machine-to-human, human-to-machine). These systems can correspond to sensors, machines, products, or entire industrial complexes. This concept takes the name of **Internet of Things** (IoT) and is not only extended to the industrial field but also to health, home automation, and civil administration [10].

The fact that the collection and dissemination of data no longer have to be done by hand, with long and expensive experimental campaigns, makes it possible to collect a huge amount of material that can be transmitted and stored efficiently and quickly. These data can then be analyzed to extract useful information from the point of view of design, organization, and optimization of products, machinery, and processes (**Big data analytics**), so much that the main source of automation is no longer considered the users, but the data itself [10].

Given the speed with which data are collected, their study with traditional methods is unthinkable if a valid amount of information is to be obtained. In this situation, it is witnessed a huge development of **Artificial Intelligence** methods, such as **Machine Learning**, which can efficiently assimilate the collected data and make decisions autonomously or guide users.

The increasing presence of representations of physical systems on the network makes the concept of **Cyber-Physical Systems** (CPSs) possible. A CPS is a system that is composed of a **simulated** digital part (**Digital Twin**) that mirrors the functionality and behavior of a real part to make possible predictions, monitoring, and diagnosis operations [11]. The fusion of the digital and real worlds is even more pronounced when considering the development of **Augmented Reality** systems that allow the user to physically see the digital part of a system.

The capabilities of self-learning and self-management make systems increasingly independent of human control, which is why there is a strong incentive to create autonomous robots and factories that are self-reliant both technically and in terms of management. The development of technology is making the traditional hierarchical business management model obsolete with an increasing decentralization of decisions [12], and therefore a **vertical integration** with the partial automation of the organizational structure is taking place. At the same time, there is an increasing **horizontal integration** concerning communications and extra-firm decisions, thanks to innovative supply-chain management techniques [13]. The goal of these innovations, in addition to increasing management efficiency and resource utilization, is to increase business adaptability and flexibility [3]. Indeed, in the modern globalized world, it is vital for a company to be able to chase customer demands as efficiently and quickly as possible. In this sense, **Additive Manufacturing** technologies have become increasingly important [14]. They, in fact, make it possible to use generic and easily reconfigurable machines to create highly customized goods whose design is fully digitized with significant simplifications in warehouse management.

Due to the spread of IoT devices and the importance of Big Data for a company it becomes essential to have modern IT equipment that, at this point, becomes a critical resource for development. For most companies, the solution is no longer to adopt only internal resources but to rely on commercial cloud services [10]. The term **cloud** refers to a set of remote data storage or processing services based on the pay-as-needed formula managed by professionals: this guarantees companies using it a high level of efficiency, flexibility, and scalability. Another advantage of cloud computing is that it facilitates remote access to corporate resources by improving collaboration and management possibilities [15]. The main problem related to this solution is the security of data that are no longer managed within the company: this can be an issue for critical data [2]. The concept of cloud computing can be extended to that of cloud manufacturing. In this case, the same concepts of cloud (on-demand, distributed services, absence of centralized management) are applied to the world of manufacturing in order to achieve a transition from a production-oriented to a service-oriented system. This application is not limited to the production part but can be extended to the design and testing phases [15].

Since all innovations related to Industry 4.0 rely on network infrastructure, it is imperative to provide for its security. For this reason, **cybersecurity** is of paramount importance for a company looking to make the digital transition, and should always be considered a critical aspect to invest in.

1.2 Additive Manufacturing

Additive Manufacturing (AM) processes are those processes in which the product is built by adding material layer by layer, they are performed by numerically controlled machines commonly called 3D printers. Compared to traditional manufacturing methods (such as turning, milling, and casting) AM systems have the advantage of greatly simplifying production. In fact, the same machine can manufacture parts with a wide variety of shapes and properties starting from the same raw materials, many times these parts do not need further processing and are ready for use just after printing. In addition, compared to traditional subtractive methods, it offers the possibility to make complex shapes in one machining process where with traditional methods many separately produced parts would need to be assembled [16].

Compared to traditional systems, the main defect of AM systems is the slow building process, which makes them unsuitable for mass production. Another problem that has so far limited the use of this technology is the poor repeatability of the results that makes it difficult to validate critical parts made in this way.

The technologies on which many additive manufacturing machines are based (photopolymerization, powder melting [17]) were developed in the 1960s and 1970s but the real breakthrough came only with the development of the chemical and computer industries in the late 20th century [18]. The 1980s and 1990s saw the publication of the first patents and scientific papers on the subject [19, 20]: the term “3D printing” was invented [21], the first prototype of Powder Bed Fusion printer [22], of stereolithography machine [23] and of Fused Deposition Modeling machine [24] were presented. Between the ’90s and 2000s, the technology spread further thanks to the discovery of more efficient processing systems and the development of the first specific software. From 2000 onwards, thanks to the decrease in the price of computers and microcontrollers, additive manufacturing systems became widespread in industries, laboratories, and homes [14].

While at the beginning of their history 3D printers were mainly used for rapid prototyping and soft-tooling, currently their use in the production of highly customized finite products or parts with shapes that cannot be realized traditionally is getting wider and wider [14].

According to the European Association of the Machine Tool Industries and related Manufacturing Technologies [25], the wide use of AM will bring developments in many field [26], such as:

- Innovation, as it encourages the creation of business models based on small production and customization.
- Performance, as it multiplies the freedom of design by allowing the optimization of material distribution.
- Sustainability, as the most used polymers are recyclable and the shortening of supply lines of finished goods is encouraged.
- Competitiveness, as the time between the conception of a product and its realization is reduced.

AM technologies have shown their potential during the recent international crisis due to the COVID-19 pandemic. In many cases, they have been invaluable in making up for supply chain disruption [27] especially in the medical field [28].

Currently, there are several main AM technologies, which in turn, are divided into several sub-genres often related to the patents of the manufacturing companies:

- **Vat Photopolymerization** (VPP) [29]: it uses liquid polymeric material (called photopolymer) in which a polymerization reaction is triggered by exposure to electromagnetic radiation (usually UV). It is used both industrially and in laboratories and is known for its excellent surface finish and geometric accuracy. The main defect of this technology is the limited choice of materials, restricted to photosensitive polymers only. The best known VPP technologies are Stereolithography (SLA or SL) and Digital Light Processing (DLP).
- **Powder Bed Fusion** (PBF) [30]: Powder Bed Fusion printers use one or more thermal sources to fuse a powdered material. They also have systems for defining the melting zone and inserting new layers of powder. Nowadays the most common heat source is a laser beam: in this case, the technology is called Laser Sintered Powder Bed Fusion (LS-PBF). This technology has the advantage of being able to be used with a large number of polymeric, ceramic, and metallic materials. The main drawback is the slow build process of the part. The most popular PBF technologies are Selective Laser Sintering (SLS), Selective Laser Melting (SLM), Electron Beam Melting (EBM).
- **Material Extrusion** (MEX) [31]: in these systems, the construction of the piece is carried out with the extrusion of material from a calibrated nozzle. Normally, this is thermoplastic material that is brought to melt during extrusion and then solidifies by cooling. Different technologies, such as drying or contact polymerization with a reagent, can also be adopted. The machine is fed with solid material in the form of filaments or pellets, then the material is pushed into the nozzle with a pinion, a screw, or a piston. Currently, it is the most widespread technology for 3D printing and is used both at hobbyist and professional level due to the great versatility

and limited cost of the machines. The main advantage of this technology is the possibility to use any material that, once liquefied and extruded, can maintain its shape for a sufficient time to solidify, so it is currently used for printing polymers, metals, and ceramics but also food products [32], medical devices [33] and concrete [34]. The technology's shortcomings are long building time, poor accuracy and surface finish, and anisotropy of the produced part. MEX technology is also known as Fused Deposition Modeling (FDM) or Fused Filament Fabrication (FFF).

- **Material Jetting** (MJT) [35]: in this type of printer, the material is ejected from a moving head by spray or drip in a manner similar to what happens to a 2D inkjet printer. Initially, materials similar to wax were used, but currently the application has been extended to photopolymers, ceramics, and metals. The main advantages are low cost, fast printing speed, and the ability to mix different materials and manage color. The problem with this type of technology is the limited choice of materials, and the poor geometric accuracy.

- **Binder Jetting** (BJT) [36]: printers using this technology have a head that sprays a binder material onto a powder consisting of the main material. In terms of operation and choice of materials, the machines are similar to MJTs, except that in this case only a fraction of the mass of the final part is sprayed by the head as the bulk is powder. This gives faster and stronger prints but deteriorates geometric accuracy and surface finish. BJT printed parts also often need post-processing, such as sintering in a furnace or infiltration of an additional adhesive.

- **Sheet lamination** (SHL) [37]: in these machines, each layer consists of a sheet (usually of paper or cardboard) that has been suitably cut. The raw sheets are laser cut, stacked, and glued together to obtain the desired part. SHL printers were among the first to be built but today they have few industrial applications. The main merits of this technology are the low cost, the simplicity of the raw material required, and the speed of construction of the part. The defects are the inefficiency in the use of the material and the anisotropy of the pieces obtained. In recent years studies are being carried out on the possibility of using this technology for the realization of composite and reinforced materials. The SHL technology currently most widely used is Ultrasonic Additive Manufacturing, which works with metal foils stacked at high temperatures, ultrasonically welded, and milled.

- **Directed Energy Deposition** (DED) [38]: this technology is based on a head that releases powder or a filament of material that is immediately melted (by a laser or electron beam). Although it can be used for many types of material, it is mostly used for metal printing. The main advantage of the technology is the ability to create high-density parts with controlled microstructure while current resolution and surface finish remain limited. An interesting aspect of this technology is that it can be used to perform repairs on existing parts, as well as print parts from scratch.

1.3 Internet of Things

The Internet of Things is a network of physical devices connected to the global network infrastructure through which they can exchange data. The devices can be machines, vehicles, drives, buildings, or plants: the important thing is that they are equipped with sensors to collect data, actuators to execute a command, software, and a connection [3]. An IoT system can be composed of a large number of different devices, the operation of which must be integrated into a single architecture, that is why systems of this type are ideally divided into four layers [39]:

- **Sensing layer**: this is the part that integrates with the physical devices. Its purpose is to collect data and execute actions.
- **Networking layer**: is the layer that manages communication between devices. Its purpose is to collect, group, and send data, connect with new devices, manage network problems and security.
- **Service layer**: manages the services offered by the various devices and associated tasks, such as transmission and storage of data, research, communication...
- **Interface layer**: manages the interactions with other applications, offers a single interface that can be used to interact with the entire network of devices.

IoT systems are being widely adopted in the industrial environment; in fact, it has been noted that this technology is fundamental especially at the management level. A typical IoT system is one in which each product that moves within a plant is equipped with a chip that uniquely identifies it. This allows for real-time visualization of the flow of materials throughout the production chain, greatly improving supply chain integration and efficiency [40]. The importance of this system is also evident from modern attempts at standardization through an Electronic Product Code uniquely associated with the RFID tag present on an individual item [41].

1.3.1 Internet of Things and Additive Manufacturing

In addition to the management point of view, the manufacturing sector can also make use of IoT systems from the perspective of optimizing industrial processes, improving safety and energy impact [42]. Given these considerations, it is not surprising that IoT systems are already widespread in the Additive Manufacturing arena. For example, many control systems are already available that allow users to send and or receive data over the network from a printer or group of printers, upload their designs to the cloud, and observe the status of the print in real-time [43].

An example of applying IoT to additive manufacturing is the cloud manufacturing system developed by Chen and Wang [44] in which workload distribution and delivery logistics are automatically managed. Similarly, Sivabalakrishnan et al. [45] have built a prototype of Smart Factory for Additive Manufacturing System that manages printed part orders and enables remote management of printing, machine offloading, and conformity assessment. Song and Zhang [46] propose the realization of an IoT-based predictive maintenance system that allows the automatic production of 3D printed parts in order to minimize downtime. Instead, Salama et al. [47] built an IoT-based monitoring system for an MEX printer and tested it by tracking nozzle temperature and detecting filament breakage. Barbosa and Aroca [48] finally propose an interface for mobile devices able to recognize through a beacon module a nearby 3D printer, show its process data, and control its operations.

1.4 Cyber-Physical System and Digital Twin

One of the most important features brought by Industry 4.0 is the ability to connect the real world with a digital representation of itself using sensors and high-speed networking, this type of connection is called Cyber-Physical System (CPS). The term CPS was coined by Gill [49] at the National Science Foundation in 2006 to denote new complex systems for which traditional nomenclature could not be used [50]. The CPS consists of three main elements: physical objects appropriately equipped with sensors (smart objects), digital models of the objects in question (digital twins), and services based on available data [51]. In this context, the concept of Digital Twin (DT) is particularly relevant: the term was first presented by Grieves in 2003 [52] then mainly used in the aerospace field [50]. A digital twin is defined as an integrated, multi-physics, multi-scale simulation of a system using physical models, data collected from sensors, and historical data to mirror the state of its real-world equivalent [53]. The digital twin performs three primary functions [11]:

- **Prediction**: the DT can perform simulations to predict how its physical part will perform to certain inputs.
- **Monitoring**: the DT can perform in real-time the same operations as its real counterpart, using values collected from sensors and simulated values derived from its mathematical models. These values can be consulted directly by the operators or by automatic systems to identify any anomalies and remedy them, they can serve as input for optimization algorithms or can be recorded in an orderly manner to be used as a comparison or statistical basis.
- **Diagnosis**: by examining the monitoring data and comparing them with historical information or with preset threshold values the DT is able to locate the source of many anomalies and, in any case, the data collected during operation and the possibilities of simulation are a great help for error tracing.

Depending on the functionality, the digital twin can reach different levels of integration [54]:

- **Digital Model**: is the mathematical model that represents a physical part. It is not directly connected to the real environment but can perform offline simulations (Figure 1.1).

- **Digital Shadow**: the system can process data directly from the real world but cannot act on its physical part. Systems of this type mainly act as support for human operators by acting as an advanced graphical interface with monitoring and diagnostic functions (Figure 1.2).

- **Digital Twin**: to have a true digital twin, the system must not only receive data but also act on the real world. The system can actively participate in the operation of the physical part to optimize its functionality or act only in certain situations (Figure 1.3).

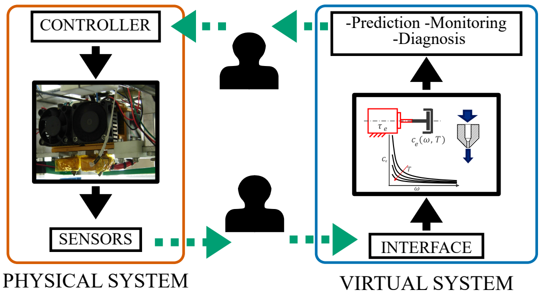


Figure 1.1: Example of CPS with a Digital Model type integration: the physical and virtual systems do not communicate directly.

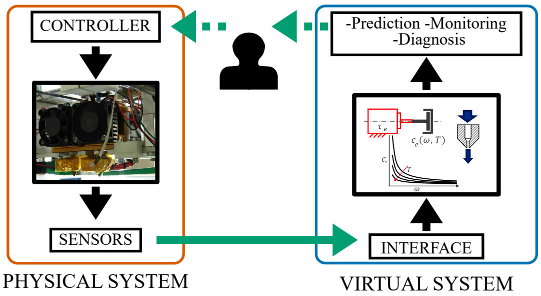


Figure 1.2: Example of CPS with a Digital Shadow type integration: the virtual system automatically receives data from the physical system but cannot operate on it.

The systems based on digital twin are very widespread in the manufacturing sector, so much so that the establishment of a CPS is one of the pillars on which the concept of Smart Manufacturing is based. Smart Manufacturing refers to the application of intelligent technologies for fast and efficient production of new products, dynamic market response, and optimized supply chain management [55]. At smart manufacturing level, several models of implementing a DT with different communication capabilities are possible [56]:

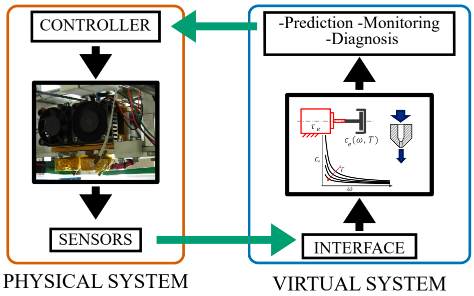


Figure 1.3: Example of CPS with a Digital Twin type integration: the virtual system automatically receives data from the physical system and can act on it.

- DT of a product: a digital model represents the product during the design and production phases as well as, possibly, during use, maintenance, and disposal [57]. Usually, twins of this type have no direct connection to their physical part but rely on external systems to receive data from their physical equivalent. They can be used to correlate how a part is produced with its actual characteristics and performance once in use. [58]

- DT of a part: the individual component of a machine can be represented by a digital model, generally for monitoring and predictive maintenance purposes. This type of twin can be without direct connection (in this case the actual data is inferred from sensors placed in other locations) or it can run on a smart device with integrated measurement and communications systems.

- DT of a production system: this means the digital twin of a machine or a production process, intended as a single part or as a union of the twins of various components. The purpose of this twin is normally the control, optimization, and diagnosis of a process or a system that includes the modeling of many parts. In this type of structure, the DT is often directly integrated into the control system.

- DT of a plant: it is possible to create a model that includes all the DTs of one or more plants and, possibly, services outside the company itself. It can greatly improve the flexibility of an automated environment by updating the workflow according to the contingent needs, even distributing it among different structures (Virtual Manufacturing or Cloud Manufacturing). Also, a system of this type is essential to optimize the management of the supply chain, to evaluate the costs associated with each product and energy consumption.

1.4.1 Digital Twins and Additive Manufacturing

In the field of smart manufacturing, the additive manufacturing can greatly benefit from the use of a digital twin at all levels of implementation due to the intrinsic characteristics of this technology:

- Complex and little-explored physical processes are involved in additive manufacturing, as it is based on rather new and continuously developing technologies. The study of these phenomena requires continuous experimental campaigns and the systematic collection of a large number of data. The use of a DT can reduce or eliminate the "trials and errors" phase which is very costly in terms of time and materials [59].
- In additive manufacturing the cost of the machine is quite high concerning the number of pieces produced [60] so, to keep costs at an acceptable level, the machine must be put in a condition to produce as many pieces as possible. This is feasible by evaluating the health of each component and the impact that part degradation has on performance [61].
- In additive manufacturing the cost of materials is a very significant item in the total (especially metals and engineered polymers) so it can benefit greatly from optimizing printing processes [62].
- In additive manufacturing the product qualification path is long and expensive: using a digital twin can help shorten it [63].
- Each additive manufacturing machine can produce an infinite variety of geometries and often process different materials under different conditions. This makes it very difficult to assess the impact on energy consumption and machine life of individual processes and, consequently, their cost [43]
- Additive manufacturing is an extremely flexible technology: the use of a digital twin can help leverage this feature to improve business management.

For the reasons outlined above, many examples of the application of digital twin technology to the field of additive manufacturing can be found in the scientific literature. In general, in the observed works, digital twins have several application areas [64]:

- **Factory-level**: the twin aims to optimize the manufacturing system in its entirety, therefore, load distribution, supply chain, and in general PLM. This type of system generally includes more than one printer and loading/unloading and handling systems.
- **Machine-level**: the twin controls the manufacturing process of a single part, focusing on the mechanical and thermal operation of the machine.
- **Process-level**: the twin models the printing process to control the microstructure of the material and optimize its mechanical properties and surface finish.

Digital twins of 3D printers have had a great development in the last years; in fact, the first digital models explicitly thought for this purpose are theorized in 2017, and in the space of a few years, they have gone from the first digital shadows with assistance and prediction functions to real digital twins with integrated control systems. In 2017, in fact, DebRoy et al. [63] illustrate possible ways to simulate material and heat flow, solidification, and grain formation with the aim of determining at the microstructural level the properties of metals printed with PBF and DED machines, and the presence of residual stress. This work was validated later by Knapp et al. [65]. Furthermore, the same year, Yang et al. [66] proposed a gray box model that combines the simulation of a physical system with the statistical data analysis to predict processing errors in a PBF metal printer. This works kicks off a series of scientific publications that focus on prediction and control, at the process level, of laser PBF metal printers.

In 2019, Mukherjee and DebRoy [67] theorized about a DT that could reduce the trials-and-errors phase required for PBF printer calibration and the time to qualify printed products. Ko et al. [68], on the other hand, propose a framework, based on digital twin and machine learning, that can perform predictive models using data accumulated during previous printing processes (self-learning functions). Next, Gaikwad et al. [69] present a system based on supervised learning to identify printing defects by comparing a simulated digital model with collected historical data. Gunasegaram et al. [70] propose the use of machine learning models trained using real and simulated data with the aim of obtaining real-time simulations to realize closed-loop controls. Heo et al. [71] study a DT to determine the microstructure and mechanical properties of 3D printed metal and use it to optimize the shape of laser beams to achieve specific final properties. Similar work is done by Yavari et al. [72] who focus on modeling the melting and solidification of the material combined with real-time measured data to find any defects in the parts. Stavropoulos et al. [73] build a robust DT-based control system capable of handling key process parameters (such as peak temperature) of a laser PBF printer in real-time and compare it to a traditional control system demonstrating its merits. Finally, Klingaa et al. [74] realize a digital twin able to model and predict the influence of gas and heat flows on the surface properties of molded parts.

With reference to metal 3D printers, in the same years can also be seen the realization of various factory-level DTs focused on product modeling. For example, Mandolla et al. [75] propose to use digital twin based on blockchain to manage the supply chain, track the entire life of parts produced for AM for the aviation industry with the idea of improving their compliance, traceability, quality, and safety. Liu et al. [76] realize a Cloud-based digital twin capable of managing data during the entire creation phase of a product made by AM (design, planning, printing, post-processing, verification) that synchronizes and controls the local DTs in charge of each particular task. Scime et al. [77] propose a DT and AI-based infrastructure that can manage the qualification of 3D printed parts based on collected data that is designed to be easily scalable. Similarly, Guo et al. [78] propose a blockchain and DT-based model to implement a PLM system encompassing design, manufacturing, and service in which the collected information is used to optimize all phases. The system does not relate specifically to metal printers but is designed more broadly.

The reason why the research on digital twins for metal AM has had this development is that 3D printing is very promising for the aeronautical industry, as it responds to the need to produce small volumes of very complex-shaped parts. In addition, the possibility of freely setting the internal structure of the parts translates into the possibility of greatly reducing their weight, a factor of fundamental importance as it saves fuel during flights. Despite the attractiveness of this technology, so far its wide use in this field has been held back by the poor repeatability of the printing process that causes non-uniform properties and the possible presence of flaws (hence process-level applications), and by the complex qualification process for the printed parts (hence factory-level applications).

Research on MEX printers has also made great strides in recent years and especially applied to the machine level. The reason for this success is essentially due to the wide diffusion of this type of systems and to their low cost, which makes them suitable also to serve as a proof-of-concept to verify the operation of more generic DTs in the manufacturing industry. It should also not be forgotten that these technologies can also be used to fabricate critical parts in engineering plastics and suffer from similar defects to those highlighted for PBF molds. In 2018 Hu et al. [79] propose a cloud-based digital twin to manage a cloud manufacturing system focused on the efficient management of data flows and quality of service, while Shainar et al. [80] propose a cyber-physical manufacturing cloud (CPMC) that allows the simultaneous management of printers placed in different locations focusing on the efficiency of the communication protocol. Both types of research are not directly focused on 3D printing but use a fleet of MEX printers as proof of concept for a factory-level DT.

An example of realizing machine-level DT for MEX printer is the work of Balta et al. [81] who propose the architecture of a DT for performance monitoring and anomaly detection and apply it by realizing a system for energy performance evaluation and extrusion temperature monitoring. Chhetri et al. [82] meanwhile realize DT of a MEX printer that uses only side-channels emissions (sound, vibration, energy absorption, magnetic field...) as input, and is thus able to locate anomalies and evaluate the printing quality in a non-intrusive way. Lai et al. [57] highlight the possibility of realizing predictive maintenance systems based on DT using as an example the extrusion system of a MEX printer. Moretti et al. [83] use a DT to accurately identify the perimeter of layers with the goal of precisely commanding a machine-vision-based geometric accuracy evaluation system. Henson et al. [84] implement a DT capable of detecting failures in part construction by comparing simulated data with data collected by a multi-view vision system. Mourtzis et al. [85] propose a DT capable of collecting data, running simulations, and correcting printing parameters that can be controlled remotely with a special interface. Guo et al. [86] propose a framework for a cloud-based digital twin that relies on a locally executed edge layer for time-sensitive tasks and tests it on a group of MEX printers. Finally, Corradini and Silvestri propose a machine-level twin for print process monitoring, quality assessment [87], and error prediction with self-calibration capabilities [88].

Some relevance is also given to DTs using augmented reality in the AM domain. In this sense the work of Cai et al. [89] can be mentioned: it consists of a printer, composed of several reconfigurable robotic arms with one extruder each, that uses a digital twin to calculate the position of each arm, simulates the printing process and, in collaboration with augmented reality, guides the user in reconfiguring the system. Yi et al. [64] also propose a digital twin that models the creation of the part using augmented reality. The part is generated by modeling the filaments as cylinders and posing volume conservation. In addition, the twin can calculate the cost, energy spent, and greenhouse gas emission associated with each print.

1.5 Objective of the proposed work

The work proposed in this thesis was carried out within the international cooperation project *INCANTO PrINT*. A research project involving the collaboration of seven university in different countries (Universidade Federal de Santa Catarina, Bremer Institut für Produktion und Logistik, Università di Parma, Scuola Universitaria Professionale della Svizzera Italiana, University of Tennessee, Université Laval, Pontificia Universidad Católica de Chile) for the realization of an IoT platform intended for the distributed and automated production of personalized medical devices.

The objective of this work is to detail the realization of a machine-level digital twin for a MEX printer that possesses the main functionalities required by Industry 4.0 paradigms: simulation, monitoring, and diagnostics and that can effectively act in case of anomalies without user intervention.

The features that have been implemented in the twin are:

1. Advanced interface: the graphical interface allows remote real-time visualization and control, of the printing process and critical parameters.

2. Automatic monitoring functions: the twin is able to recognize the most common irregularities, stop printing, and secure the machine.

3. Diagnosis functions: when an irregularity is detected, the system is able to isolate the possible reasons for the failure and guide the user through the recovery process.

4. Simulation functions: it is possible to disconnect the digital twin from the physical printer to run simulations in order to verify the effectiveness of the printing parameters.

5. Data collection and storage: the system automatically collects and sorts printing data so that further analysis with external software is possible.

6. Auto-calibration function: the system can automatically recreate certain critical situations in a controlled manner and calibrate itself to alert the user if they arise during normal operation.

7. Creation of a digital twin of the printed product: the digital twin of the machine is able to recreate in real-time a detailed digital model of the printed piece that

can then be exported for post-processing analysis or possible use in an augmented reality environment.

8. Predictive maintenance guidance: the system is able to assist the user in maintenance interventions as it takes into account the movement and load to which critical parts are subjected during the life of the machine.

9. Conformity evaluation: the system is able to automatically evaluate, during the post-processing phase, the geometric conformity of the printed piece comparing it with an ideal printing model and highlighting the deviations.

10. Modularity: the twin is composed of several independent parts that communicate via the network. This allows the distribution of the computational load on different machines and offers greater security in case of problems with one of the systems.

11. Non-exclusivity of input data: the digital twin system must not exclude other monitoring systems that use the same input data. Different monitoring systems must be able to operate independently at the same time.

All of these features were added without relying on proprietary solutions. The printer has been custom built with generic hardware, the software used is open-source, the printer control system has not been modified from the default version. A print host program with a specially made plugin is used to communicate print data and send commands to the printer.

Chapter 2

Material and Methods

In this chapter, the structure of the proposed digital twin is outlined. Initially, the operation of a typical printing process is explained, then the strategy on which the implementation of the digital twin is based is discussed. Then all parts of the digital twin are explored, including the hardware and software structure.

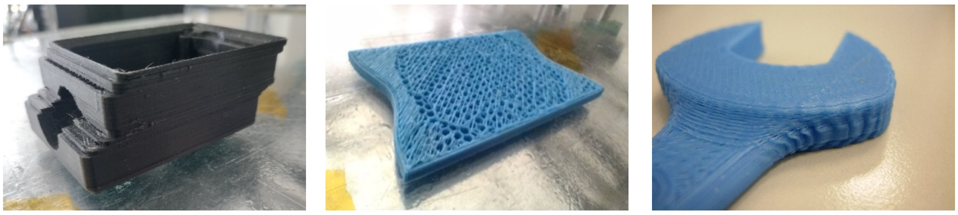
2.1 The digital twin

2.1.1 The typical workflow of a MEX printing process

The MEX 3D printer is a numerically controlled machine in which one or more extruders compose the printed part layer by layer by extruding a filament, usually of thermoplastic material. There are many possible layouts for this type of machine but in most cases, they are Cartesian or parallel kinematic machines with three degrees of freedom movement (X and Y axis for horizontal movement, Z-axis for vertical movement) plus one or more independent axes that regulates the amount of extruded material for each extruder (E-axes).

These printers are controlled by an integrated controller, usually equipped with a physical user interface for basic commands and the setting of working sequences. In almost all cases there is also a serial interface that allows the printer to receive commands from other devices (generally a PC). The main role of the controller is to interpret the commands received and control the temperature and the movement of the machine.

The movement is usually performed by stepper motors controlled in open-loop whose movement is reset by mechanical or magnetic switches. The reason for this choice is due to the low cost of this technology that uses cheap motors and especially allows the use of a simple control strategy. The price that is paid with this choice is certainly the lower precision, but it must be taken into account that, in any case, MEX printers do not reach such precision that this disadvantage is observable. A more evident problem due to this choice is the possibility that a motor misses one or more steps during a machining process if the required torque is too high. This event cannot be detected by the control



(a) Layer mismatch (b) Extruder cogging (c) Excessive acceleration

Figure 2.1: Examples of printing errors not easily preventable in the standard layout. There is a case of shifted layers (a) and nozzle cogging (b) caused by loss of stepper motor steps, and a case of geometric errors (c) due to the effect of elasticity of the drive system combined with excessive acceleration.

system and leads to a layer mismatch (if it occurs on the X, Y, or Z-axis) or to the failure to extrude the filament (if it occurs on the E-axis). A more insidious problem is that, even in the absence of steps missed, the lack of feedback on the position of the end-effector makes the geometric accuracy highly dependent on the kinematic chain used, in particular, the presence of backlash or the elastic component of the drives can alter the geometric conformity of the finished product (Figure 2.1).

The temperature control system mainly uses a heater for the nozzle, then optionally there can be heaters for the plate, the printing chamber, or cooling fans. Unlike motion systems, these always receive feedback via thermometers to ensure a regulation that is effective and, above all, safe.

The first step to take when a 3D printed piece is needed is to obtain the three-dimensional model of the piece itself, usually saved as a triangular mesh in STL format. Then it is necessary to proceed with the division of the part into layers and the definition of the printing parameters. This process is called slicing and is done by specific software.

Typically, this is one of the most critical stages as part orientation, layer direction, and filler geometry affect the printability and mechanical properties of the finished part. In addition, most of the characteristics of the molded part and the printing process are decided at this stage:

- Nozzle and plate temperatures and their variation during the build stages.
- Extrusion width and speeds, depending on the type of path (perimeters, filling, surfaces...).
- The geometry of each layer (what kind of infill to use and with what density, how many perimeters, the directions of the infill...).
- The nozzle movement strategies (the use of filament retraction, lifting during movement...).
- The use of printing auxiliary structures (skirt, brim, support structures...).

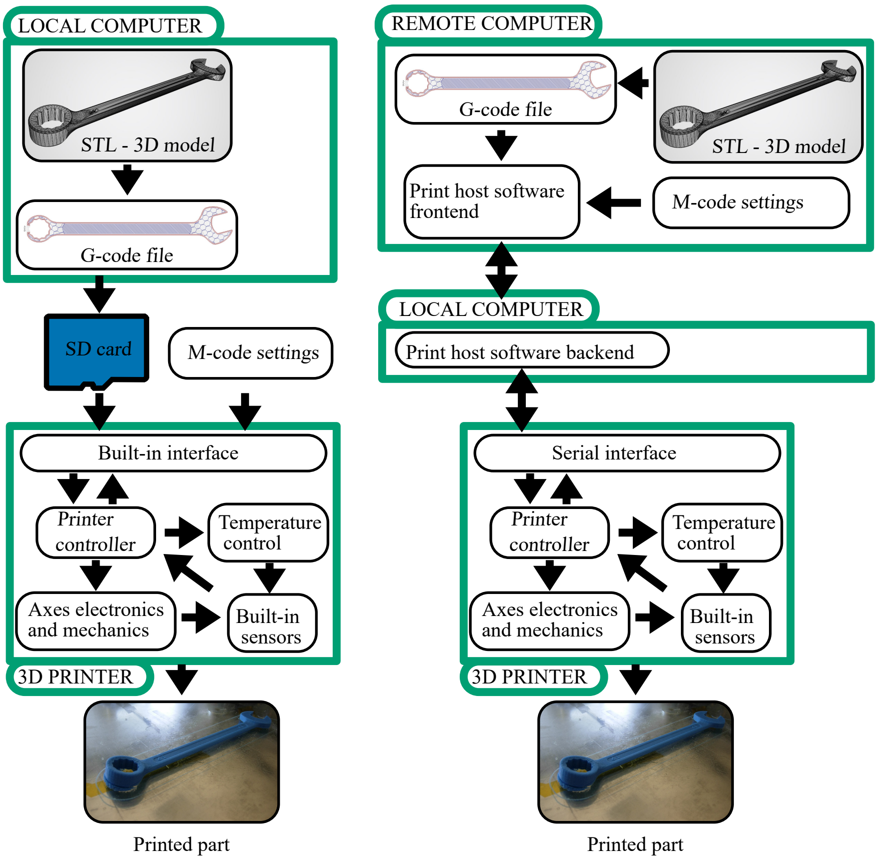


Figure 2.2: The left diagram shows a classic printing layout that uses a removable memory device and a built-in interface to control a 3D printer. The right diagram shows the layout of a printing system that leverages IoT to remotely manage the 3D printer.

Each choice offers potential benefits and criticalities, therefore, several printing trials are often required to find the right configuration. The slicing operation requires a certain computational burden and the use of a complex graphic interface, therefore, it cannot be performed by a system embedded into the 3D printer but it must be done by a separate computer. In addition, the choice of printing parameters at this stage is not strictly related to the type of machine that will be used for printing but more to the properties that the user wants to achieve in the printed part.

From the slicing of the part, the user obtains a file that contains the instructions for printing, the so-called G-code file. The G-code file is nothing more than a text file that contains, line by line, the positions that must be reached by the axes of the machine, the type of interpolation, and the speed of movement written in a language called G-code. This file could also contain commands that affect the operation of the machine in general (called M-codes), these, with a few exceptions, are not used in the printing phases but are set during the commissioning phases of the machine.

Once the G-code file is obtained, it must be executed by the 3D printer controller. In traditional systems, an SD card or a USB key is used as removable memory for the controller, so that the file can be loaded by a computer and then read by the printer. The advantage of this solution is that, after loading the G-code file on the memory, the printer becomes a stand-alone system and can operate independently. The defect is that the user has to rely completely on the interface integrated into the printer that is not necessarily present and is often minimal.

An alternative solution is to take advantage of the USB serial interface present on practically every printer model. Thanks to this interface the printer can receive the G-codes, line by line, and execute them one at a time, there is also a buffer that allows the controller to queue a certain number of commands in order to circumvent the effects of transmission latency. This operation is carried out automatically by a family of software called *print host software* that automatically manages the sending of commands from the serial interface. They also take advantage of serial communication to receive real-time information from the printer and thus offer a user interface from which to monitor the progress of the creation process. The positive side of this solution is that it allows the use of an advanced interface running on a PC to control printing, bypassing the minimalist interface embedded in the devices. The negative side is that it constrains the printer to work only in the presence of a PC physically wired to it, a very limiting factor if a large number of printers is needed. An evolution of this system is the one that, exploiting IoT technologies, allows the user to remotely control the print host software via the internet [43]. This, in addition to offering the ability to control printing remotely, allows smaller and simpler PCs to be associated with the printers, even embedded ones, as they are not required to have a user interface and a screen (Figure 2.2).

2.1.2 The digital twin structure

During the design phase of the digital twin, it was decided to start with the configuration of having a local computer with the backend of a print host application and a remote computer with the frontend and GUI (right diagram of Figure 2.2). This decision was made because it allows the implementation of a remote, two-way communication service with the digital twin without significantly impacting the base system.

The typical 3D printer controller consists of a microcontroller on which firmware operates, which can be open source or proprietary. In general, the control system is neither open nor accessible from the outside; therefore, it is not possible to freely access sensor readings, axis position or know which line of G-code is being processed, nor is it possible to freely alter working parameters to optimize a process or solve a critical situation.

A possible solution to this problem would have been to intervene directly on the controller firmware and create an ad-hoc communication system to interact with the processes of the digital twin. This solution would have certainly made the system layout simpler and more efficient, but it was discarded because it would have been difficult to

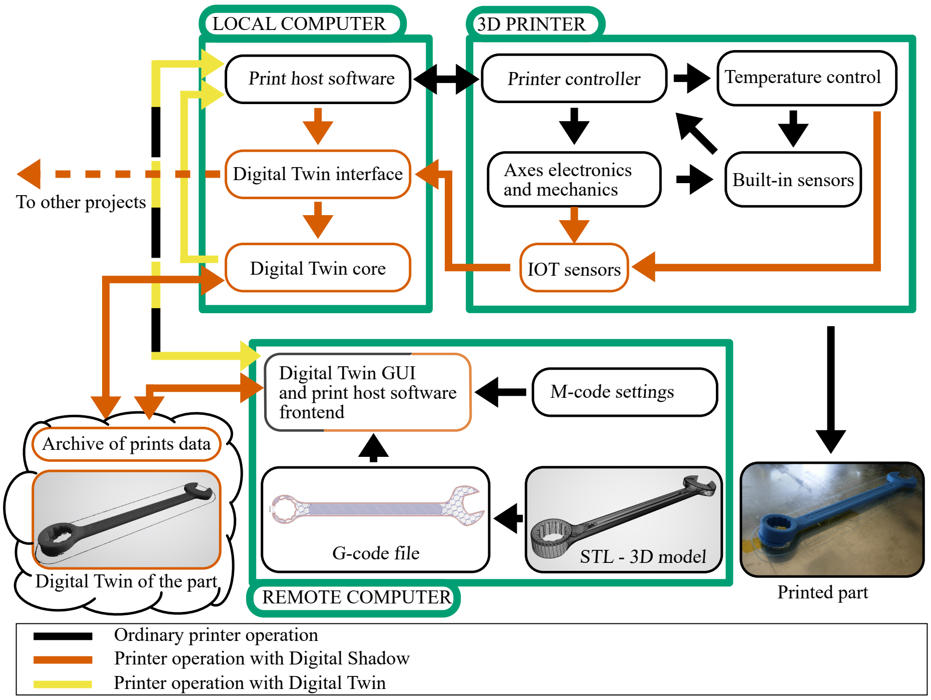


Figure 2.3: The diagram shows the structure of the proposed digital twin.

implement and would have affected the reusability of the system. The firmware of the 3D printer controller, in fact, is always calibrated ad hoc according to the structure of the machine on which it operates, very often the source code is not open and it is not even given the possibility to load a free one; for this reason,in order to export a customized control firmware to other printers, it would have been necessary to physically replace the controller, its recalibration and reprogramming with all the consequent complications.

The chosen solution, shown in Figure 2.3, uses the print host software that communicates with the controller through the serial interface. This interface, while accepting only a limited number of G-code commands and denying the possibility of reading/writing in real-time, has the advantage of being compatible with all the firmware on the market and not interfering with the basic operation of the system. Among the print host software, many open-source versions make it easy to modify.

In order to obtain effective monitoring and to close the control loops, it was decided to insert a series of smart sensors able to detect process data also in a redundant way. For the structure of the digital twin, a block system similar to what has been observed in the literature [81] has been realized: the three main blocks are the Digital Twin Interface (DTI), the Digital Twin Core (DTC), and the Digital Twin Graphic User Interface (DT GUI).

The digital twin interface's main purpose is to collect and sort data. This block receives information both from a set of smart sensors located on the printer and directly from the print host software. Its role is to extract useful information from the incoming data and send it to the digital twin core in order not to overload it and keep the complexity of the real-time simulation at an acceptable level. Another function of the DTI is to virtually untie the sensor network from the digital twin, in this way the collected data can be eventually sent to other destinations such as remote storage systems or other experimental monitoring systems.

The digital twin core is the program that hosts the actual simulation process. It receives in real-time the information collected by the DTI and uses it as input for its physical model, then by comparing the data collected and obtained from the physical model it performs the prediction, monitoring, and diagnosis functions. The results and the collected data are sent in real-time to the graphical interface and to cloud storage to improve accessibility.

The digital twin GUI, on the other hand, handles user interaction and all related functions. It allows the control of the printing process as it includes the functions of print host software frontend, it also allows the user to view the information and messages of the DTC and to set its functionality. In addition, it can be used to display historical data and perform post-processing operations.

The links explained so far (visible in orange in Figure 2.3) are all functional to provide data to the digital twin and to the user, however, to realize a proper digital twin it is necessary to implement also a reverse communication strategy, through which the twin can automatically influence the printer behavior. To do this, two communication channels have been implemented between the DTC and print host software and DT GUI and print host software through which the system can automatically send commands that are executed by the printer (yellow links in Figure 2.3).

2.2 Hardware and sensors

2.2.1 The 3D printer

For this project, a Cartesian MEX printer present in the laboratory of *Automated production systems* of the University of Parma has been used. This printer is the result of previous teaching and research projects and was built for printing large pieces (Figure 2.4). It has a printing volume of 626x355x150 mm.

The structure of the printer is made of cut and assembled aluminum grooved profiles and custom made aluminum parts. The printing surface consists of a large aluminum shelf on which is placed a glass plate that serves as a base for printing. The plate slides vertically along four linear guides and is moved by worm screws driven by NEMA 23 stepper motors forming the Z-axis (orange part in the diagram in Figure 2.5). In consideration of the high weight of the piece, each motor is equipped with a dedicated driver. During the printing process, the plate gradually lowers while the height of the extruder remains fixed, in this way the layers are correctly overlapped.

The homing of the Z-axis position is done by a magnetic sensor placed on the extruder block that also offers the bed leveling function, and a mechanical switch placed at the

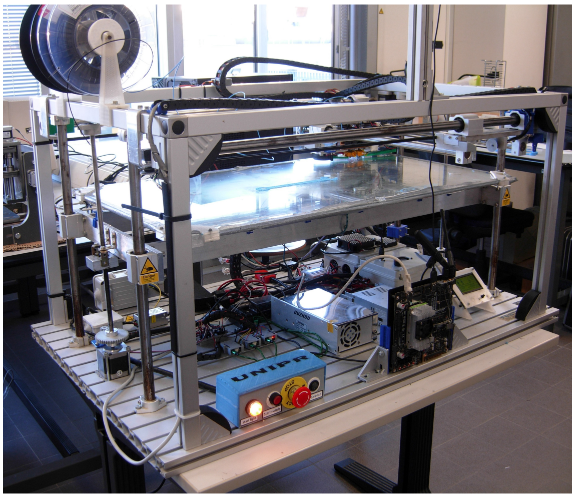


Figure 2.4: The printer used for testing

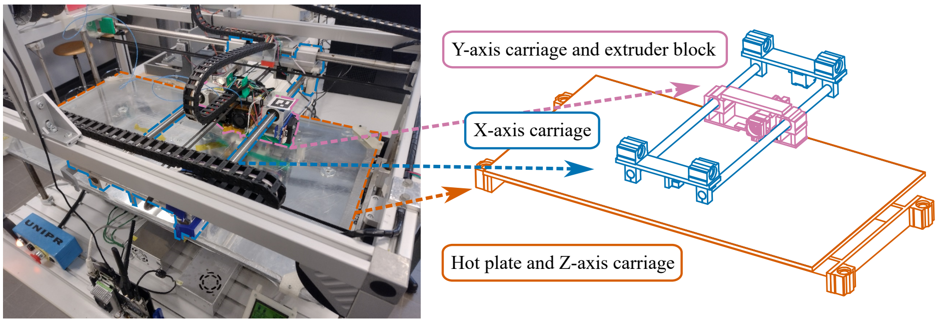

Y-axis carriage and extruder block
X-axis carriage
Hot plate and Z-axis carriage

Figure 2.5: The diagram shows the main moving parts of the printer used for the tests.

lower limit of the movement for safety reasons. In order to use the magnetic sensor, steel washers have been placed between the bed and the glass in all the positions where the probe is used. The table is heated by two 300 W Kapton heating mats that require a separate power supply and the temperature is measured by a thermistor.

The extruder block is made of perforated aluminum plates connected by screws, it is prearranged for the installation of two extruders, one of which has been deactivated for this test phase on digital twins (pink part in the diagram in Figure 2.5). The extruders are of the direct drive type, i.e. they have the motor mounted on the mobile part immediately above the nozzle. This solution, combined with the size of the piece, makes the extruder block particularly heavy. The motors that control the advancement of the filament in the extruders are NEMA 17, one of which is inactive and one equipped with a double shaft so as to move the pinion and encoder at the same time. In this type of extruder, the filament, after being pushed by the pinion, passes through a small Teflon-coated tube and through the heating block where its melting takes place and is finally extruded by a calibrated nozzle. The heating block consists of a cylindrical cartridge containing an electrical resistance and a thermistor for measurements.

In correspondence with the extruder block, there are also fans for cooling the material just extruded and the one engaged in the pinion. In the configuration used for testing the speed of the fans is not controlled but they are always active at maximum speed.

The extruder block moves along the Y-axis sliding on two cylindrical guides constrained to the X-axis carriage, the motion is transmitted by a gt2 synchronous belt driven by a NEMA 17 stepper motor.

The X-axis carriage moves along two cylindrical guides attached to the frame (blue part in the diagram in Figure 2.5). Due to its high mass, the motion is entrusted to two NEMA 17 motors and transmitted by two GT2 synchronous belts. Regarding the X and Y axes, the homing is performed by mechanical switches placed at the end and the beginning of the corresponding axis.

The system is controlled by an Arduino Mega 2560 with a RAMPS 1.4 shield, the firmware used is Marlin firmware version 1.1.9 [90], an open-source control system for 3D printers based on ATmega microcontrollers, compatible with a large number of models and very popular both in professional and hobbyist field [91].

During the use of the printer some problems have emerged, related to its structure and its use:

- Since it is mainly used for printing large parts, the printing jobs are usually very long (even tens of hours) and cannot be constantly supervised by an operator. For this reason, if a problem occurs during the printing of a layer, it can take a long time before someone notices it, with the consequent extension of the defect to the upper layers, waste of time and material, and in addition, the risk of damaging the machine.

- The resources invested in a print job in terms of time and material are considerable and, for this reason, the failure of the construction process is particularly harmful. To reduce the risk it is necessary to use very conservative settings on the printing speed but this extends the duration even more.

- Since the printer is used for educational and research purposes, it is often subject to modifications or revisions that involve the addition and replacement of parts. Each time a change is made, a new calibration process must be performed.

- Because of the long lead times for building parts, it is necessary to pay close attention to how much and how the most worn parts are used. However, this is very difficult as the machine is used by different teams for various projects.

- A fully accessible printer is very useful for collecting data for many purposes, however, each user has to implement their own framework and install their own sensors risking interfering with the work of others

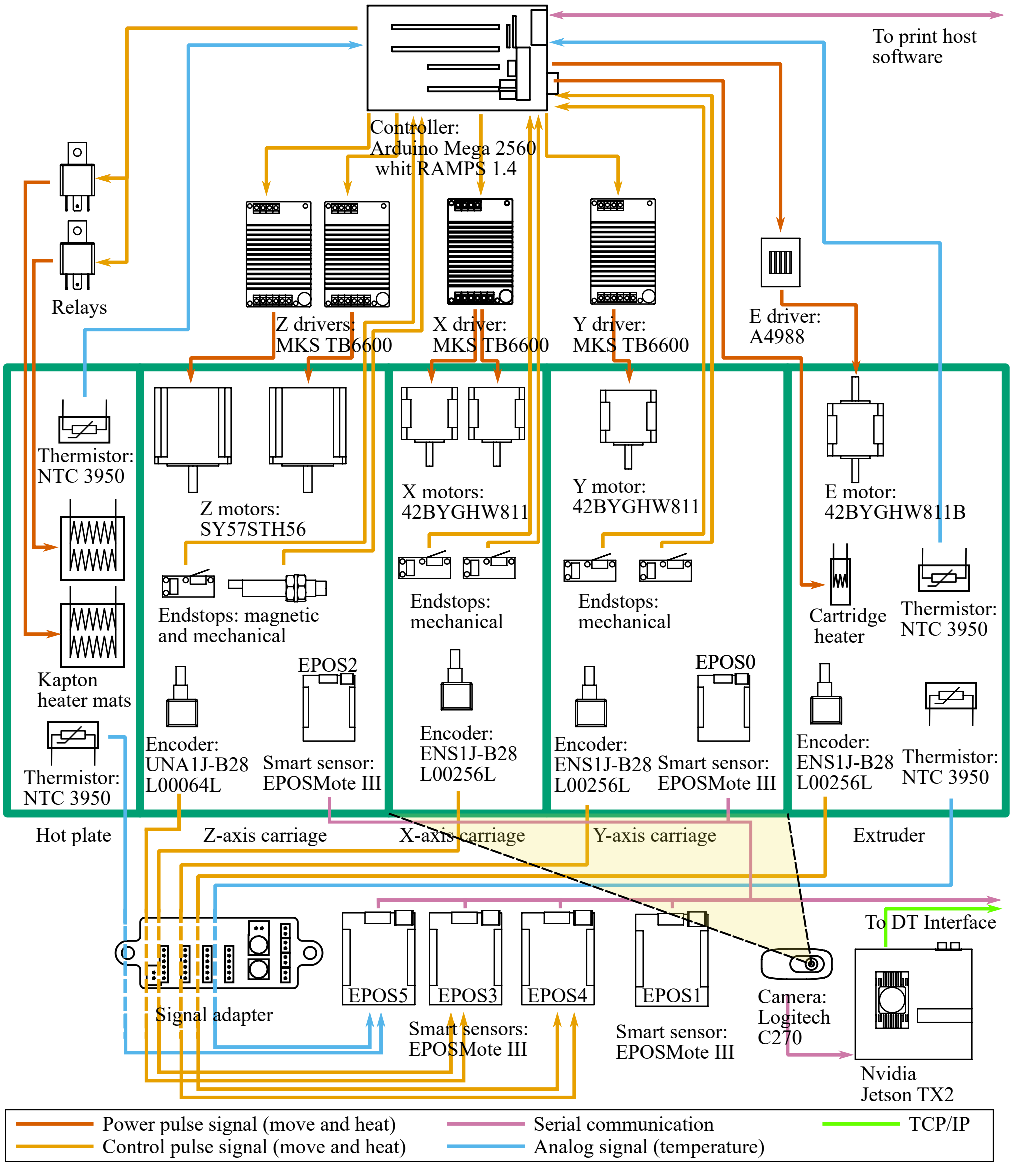


Figure 2.6: This diagram shows the actuators and sensors installed on the 3D printer used for test and how they are connected.

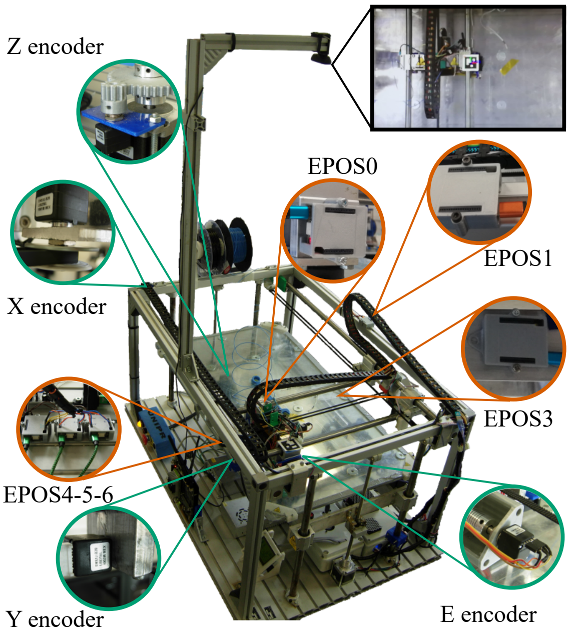


Figure 2.7: The picture shows the position of the main installed sensors.

2.2.2 Digital twin sensors

In the printer, for ordinary operation, the actuators and sensors described in the previous paragraph and shown in Figure 2.6 are installed.

The implementation of on-board smart sensors has been carried out within the *INCANTO PrINT* project with the aim of collecting data for several related works. These include an error detection system proposed by Scheffel et al. [92], the digital twin that is the subject of this thesis, and an ongoing project on print data validation using blockchain.

These smart sensors are placed in critical positions of the printer and collect data from other ordinary sensors. This, in addition to a machine vision system, is able to provide the necessary data and with the right level of redundancy.

As on-board sensors, it was decided to use the EPOSMote III smart sensors, open hardware devices based on the Epos (Embedded Parallel Operating System) project by LISHA [93]. The device is based on the CC2538 of Texas Instruments that ensures good computing power and support for the most popular IoT communication protocols. In fact, it supports both wireless communication, thanks to the IEEE 802.15.4 radio interface, and wired, thanks to the USB, UART, SPI, and I2C interfaces. The board also has analog input ports, digital I/O ports, an LSM330 accelerometer, and a Si7020 temperature and humidity sensor.

On the printer have been installed 6 smart sensors with different functions (Figure 2.7), the workload of reading and sending have been distributed in a balanced way to better exploit the computing power:

- **EPOS0**: the sensor is placed on the extruder block at the Y-axis carriage. It measures the acceleration of the extruder block and therefore of the nozzle during printing with a frequency of 200 Hz.

- **EPOS1**: the sensor is located on the fixed structure of the printer. It measures the acceleration of the structure as a result of vibrations introduced by the movement of the parts or by external factors with a frequency of 200 Hz.

- **EPOS2**: the sensor is placed on the print bed. It measures the acceleration of the plane due to its vertical movement with a frequency of 200 Hz.

- **EPOS3**: receives square wave signals from 2 optical 2-channel encoders. The first one is placed on the X-axis idler pulley and is type ENS1J-B28 L00256L, it has a resolution of 20.477 pulses/mm relative to the axis movement. The second, of the same type, is placed at the rear shaft of the extruder motor and measures the E-axis feed with a resolution of 25.6 pulses/mm. Position data are sent with a frequency of 30 Hz. Due to the different working voltages, the encoder signals must be brought from 5 V to 3.3 V by a signal adaptation module.

- **EPOS4**: Receives square-wave signals from 2 optical 2-channel encoders. The first one is placed on the Y-axis idler pulley and is type ENS1J-B28 L00256L, it has a resolution of 20.477 pulses/mm. The second one is placed on one of the Z-axis worm screws, is type ENS1J-B28 L00064L, and has a resolution of 550.4 pulses/mm. As for the other axes, data are sent with a frequency of 30 Hz and the encoder signal must be brought from 5V to 3.3 V.

- **EPOS5**: receives the analog signals from two NTC 3950 thermistors placed on the printing table and in the heating block of the extruder. These thermometers are placed in correspondence with those integrated into the printer control system and are of the same type but, unlike these, have undergone an ad hoc calibration process to increase their accuracy. Before the acquisition, the signals are also filtered by an adaptation module specially designed to remove noise and increase sensitivity in the useful measurement range.

In order to ensure a stable and fast connection, given the large amount of data exchanged, it was decided to connect the smart sensors to the local computer and then to the DTI via serial connection wired via USB.

In addition to the sensors just described, a Machine Vision Positioning System (MVPS) was inserted to detect the position of the extruder in the X, Y plane. The position measured in this way does not have the resolution and precision of that based on encoders but has the advantage of being absolute (it does not need to be reset if the position of the camera is not changed) and to include a redundancy factor, useful for the diagnostic phase.

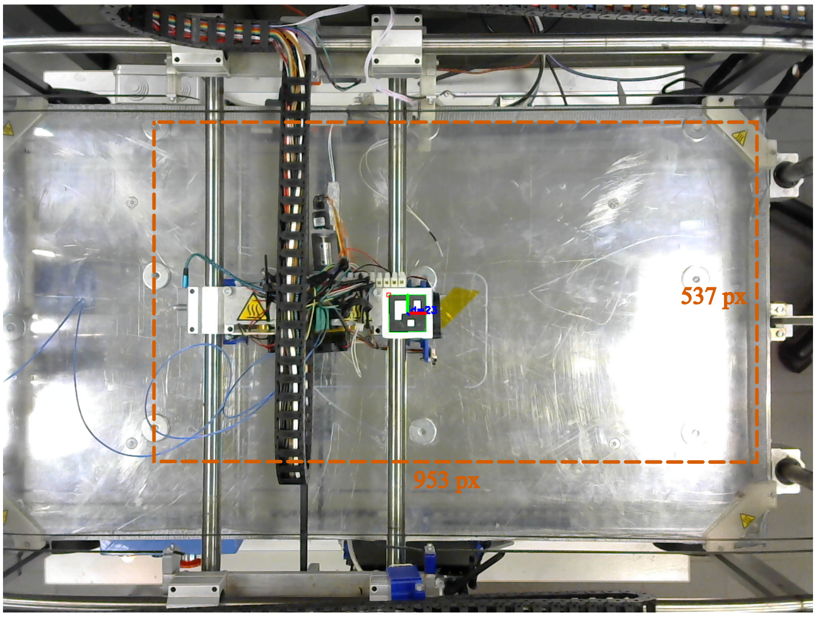


Figure 2.8: In the image it can be seen a frame captured by the machine vision positioning system in which the position of the Aruco marker number 23 is highlighted. The image has been overlaid with a box indicating the actual working area of the system.

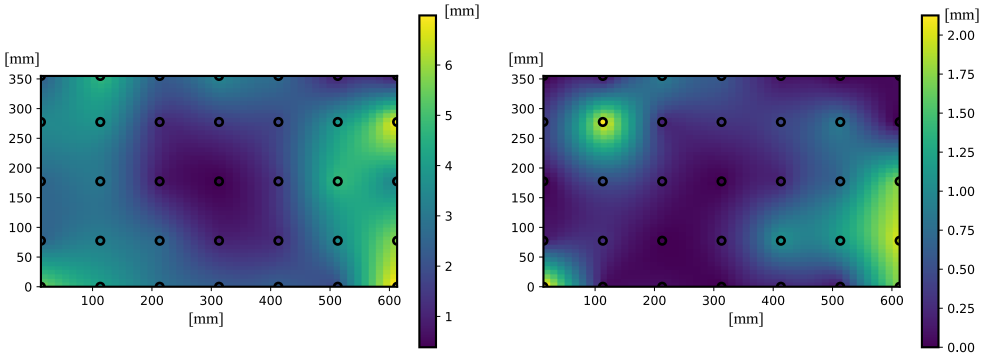
2.2.3 The machine vision positioning system

The Machine Vision Positioning System (MVPS) uses a Logitech C270 camera connected to an NVIDIA Jetson TX2 card on which the position tracking program operates. The position data collected is sent via a wired, dedicated network connection to the local computer where it is captured by the DTI process. The digital twin is designed to use the machine vision data only when diagnosing errors. If the camera is disabled, the system still operates but its diagnostic functions are reduced.

The extruder position tracking program was built in C++ with the OpenCV library [94] and is based on detecting the relative position of an Aruco marker within an input video stream [95]. In addition to its standard functions, the system can also acquire a video stream from a remote network device using the GStreamer library [96], there is also a calibration wizard to be activated if the camera undergoes a slight shift, this aligns the relative positions in pixels of the marker in the acquired image with the real position of the extruder.

In the configuration used for the tests the camera is placed orthogonally to the printing plane and is installed at a height of 1.04 m above it. The camera operates with a video resolution of 1280X960 and a frame rate of 7.5 images per second, it frames a 16 bits Aruco marker of 40X40 mm bound to the extruder block and this allows it to reach a theoretical resolution of 0.66 mm/pixel (see Figure 2.8).

Actually the accuracy and precision of the measurement decrease significantly moving



(a) Distribution of detected position accuracy in the working area

(b) Distribution of detected position precision in the working area

Figure 2.9: The image shows the distribution of the accuracy (a) and precision (b) of the vision system in the working area. For each of the 35 points indicated in the images with circles, 20 position measurements were taken. The standard deviation of these measurements constitutes the value of precision while the absolute value of the difference between the average of the measurements and the position given by the controller constitutes the value of accuracy. For the realization of the graph it has been carried out a linear interpolation from the analyzed points.

away from the center of the image due to the effects of lens distortion, which can not be totally eliminated. In addition, the type of video compression is not optimal and the vision is disturbed by the presence of shadows and reflections. In the graph represented in Figure 2.9(a), it can be seen the trend of accuracy measured from the difference between the position averaged over 20 measurements of the camera and that detected by the printer controller, judged much more accurate and therefore used as a reference. The measurements were made in 35 positions in order to form a grid. For the realization of the graph, the value of the points between the measurement positions was obtained with linear interpolation. In the image, it can be clearly seen, moving away from the center, the worsening of the measurement error that reaches almost 7 mm in the corners.

Figure 2.9(b) shows the trend in measurement precision, calculated using a method similar to that of the previous graph, except that in this case the standard deviation between the 20 measurements taken is shown. A trend similar to that observed for accuracy can be seen, even if less regular. In this case, the decrease in accuracy in some peripheral areas may be due, in addition to image distortion, to the presence of particularly strong reflections on the glossy surface of the printing plate (compare with Figure 2.8).

From the results obtained, it can be seen that the vision system is currently not ready to operate as the sole position tracking system, but it can still provide a valuable aid to the system for error diagnosis. The performance of the system is currently limited by the use of a low-end camera, not suitable for machine vision tasks, and by the presence of unsuitable lighting but could be greatly improved in future applications.

2.3 The print host software

As print host software was chosen Octoprint version 1.4.0 [97], an open-source Python software that, in addition to performing the basic functions of reading and sending the G-code lines has an integrated and complete web interface that can be used as a frontend for remote control. In addition to these features, Octoprint has a practical plugin system that allows the user to add functionalities without changing the code base and thus facilitating the maintenance of compatibility with later versions. In addition, it has a REST API, which is a library that can be used to add the functionality of the Octoprint web interface to any application. This library is of great value as it allows the DTC and DT GUI application easy access to print data and the ability to send commands.

The plugin specifically made for Octoprint has the function to connect with the DTI and send messages when certain events occur:

- When an axis homing is done, the positions that need to be set for the encoders are sent to synchronize them with the actual position of the axes.
- When there is a Z-axis movement during printing a message is sent that a new layer has started.
- When a printing process starts or ends, a message is sent to indicate this.

2.4 The Digital Twin Interface

The digital twin interface is a Python script developed within the *INCANTO PrINT* project and its main purpose is to receive data from the smart sensors, the vision system, and the print host software, group them in packets and send them to the corresponding destinations. The script acts as a hub connecting the incoming data to the various users: currently the collected data can be sent to the DTC or to a remote storage system where they are analyzed by a monitoring system based on deep learning [92]. Table 2.1 shows the data received, frequency, and usage.

In addition to the task of data management, the interface performs the function of the main node for the smart sensor network: it recognizes the connected sensors and initializes the serial communication, synchronizes the start time of the devices, and resets the position in the boards connected to the encoders.

Table 2.1: The table shows the packets that are received by the DTI with the source, size, frequency, and destination to which the contained data is sent. For reasons of space, Machine Vision Positioning System has been abbreviated to MVPS, Print Host Software to PHS, and Remote Storage System to RSS.

Data	Source	Size	Frequency	Destination
X and Z encoder posit.	EPOS3 (serial)	19 bytes	30 Hz	DTC - RSS
Y and E encoder posit.	EPOS4 (serial)	19 bytes	30 Hz	DTC - RSS
Plate and extruder temp.	EPOS5 (serial)	17 bytes	2 Hz	DTC - RSS
Extruder acceleration	EPOS0 (serial)	22 bytes	200 Hz	DTC - RSS
Frame acceleration	EPOS1 (serial)	22 bytes	200 Hz	RSS
Plate acceleration	EPOS2 (serial)	22 bytes	200 Hz	RSS
Position by MVPS	MVPS (TCP/IP)	17 bytes	7.5 Hz	DTC
New layer	PHS plugin (TCP/IP)	5 bytes	<1Hz	DTC
Print start/end	PHS plugin (TCP/IP)	5 bytes	<1Hz	RSS
Axis homing	PHS plugin (TCP/IP)	28 bytes	<1Hz	None

2.5 The Digital Twin Core

The DTC is the part of the digital twin on which the physical model of the printer operates and which offers the services of diagnosis, monitoring, and simulation.

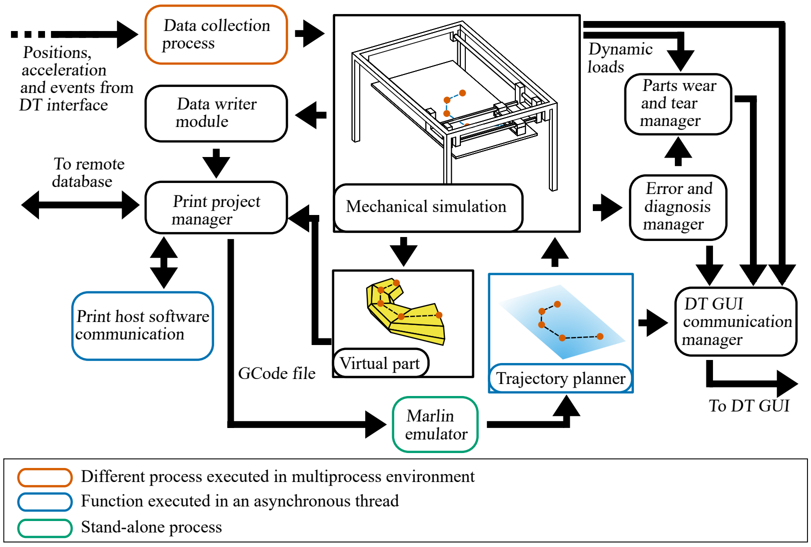

Positions, acceleration and events from DT interface
Data collection process
Data writer module
To remote database
Print project manager
Print host software communication
GCode file
Mechanical simulation
Virtual part
Marlin emulator
Trajectory planner
Dynamic loads
Parts wear and tear manager
Error and diagnosis manager
DT GUI communication manager
To DT GUI
Different process executed in multiprocess environment
Function executed in an asynchronous thread
Stand-alone process

Figure 2.10: Diagram of DTC operation in regular mode with active modules and their interactions outlined.

The DTC can be started in three different modes according to which the modules that are used change. The modes are:

- **Regular operation mode**: in this mode, the DTC acquires data from the DTI and the real printer while the monitoring, diagnosis, fault intervention, data collection, and GUI communication functionalities are active (Figure 2.10).
- **Simulation mode**: in this mode, the DTI is not active and all data are acquired by a process that simulates the mechanics of the printer. In this mode, the functions of diagnosis and intervention in case of anomalies are suspended while monitoring is carried out in partial mode (Figure 2.11).
- **Calibration mode**: this mode is designed to be executed locally during printer setup or after a significant change in the printer hardware. In this phase, the GUI is deactivated and only the modules strictly necessary to send commands and analyze the system response remain active (Figure 2.12).

The DTC can be divided in various modules whose operation changes depending on the configuration used; unless otherwise specified all modules are written in Python:

- **Data collection process**: it is launched as a separate process from the main DTC script. Its purpose is to check the connection with the DTI, read and interpret the incoming data from it and synchronize the position data.

- **Printer simulation process**: it is launched as a separate process instead of the data collection process when the DTC is started in the simulation phase. It uses the Marlin emulator to generate the laws of motion with which it simulates the operation of the printing axes in real-time.

- **Marlin emulator**: it is an executable generated from the C++ script of Marlin. It is used to emulate the planning of the trajectories of the controller of the printer, both for simulation and monitoring purposes.

- **Trajectory planner**: it is a thread with similar behavior to the printer simulator but lacking real-time capabilities. In normal operation mode, it analyzes the G-code file before printing. From this analysis it derives data useful for error detection during the actual printing.

- **Mechanical model**: this is the basic module of the DTC on which all the others are ideally connected, it manages the main loop of the script and the execution of the mechanical model. The mechanical model is always active and provides kinematic and dynamic data on the operation of the axes.

- **Parts wear and tear monitor**: this module takes care of keeping track of the wear and tear of the parts of the printer that are most subject to it, and updating the status so that it is visible in the GUI.

- **Communication thread with the print host software**: An asynchronous thread that is always active, leveraging the Octoprint REST API it polls useful information from the print host software and is used to send commands to the printer.

- **Print Project Manager**: This is the module that takes care of sorting the print data according to projects, manages the saving and loading of files to local or remote memory.

- **Data write manager**: is the module that takes care of data logging. The data log is not activated automatically but must be enabled by the GUI before starting the printing.

- **Communication manager with DT GUI**: the communication manager is in charge of sending data to the GUI with predefined packets at particular events. In addition, it receives messages from the GUI, updates system variables, and executes any associated callbacks.

- **Virtual Part Generator**: During printing, position data is used to generate a virtual model of the printed part which is then saved to the corresponding project.

- **Auto-calibration module**: The system can be started in auto-calibration mode. In this case, communication with the GUI is sacrificed in favor of a more basic command-line interface. In this mode, the auto-calibration module directly controls the printer actuators so that certain error situations occur in a controlled manner. From the system's response to these stresses, it is possible to determine some useful parameters for the digital twin.

Such a system is designed to operate on a computer located near the printer and then communicate with a GUI operating on a remote computer. This communication must take place through a commercial cloud service that offers the possibility to receive data packets, send them back to one or more destinations, and use shared memory between several devices also managing the authentication phase. For the realization of the proof of concept reported in this thesis, it was decided to implement a simplified service of this type, operating on a local network.

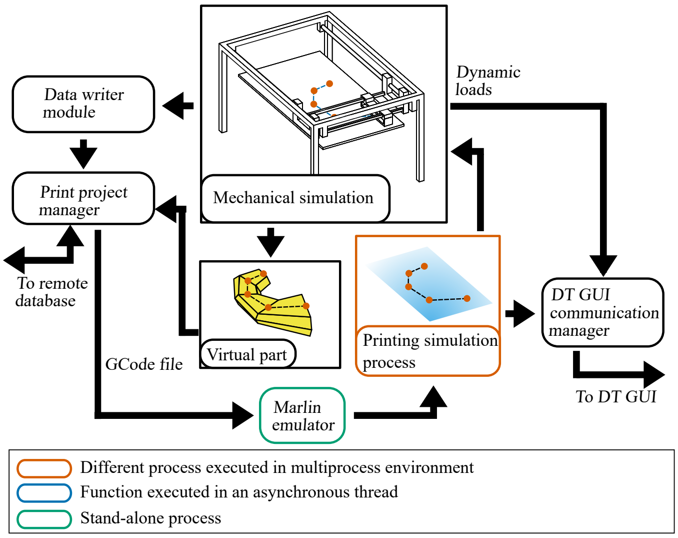


Figure 2.11: Diagram of DTC operation in simulation mode with active modules and their interactions outlined.

2.5.1 Data collection process

The data collection process establishes the connection with the DTI, receives the messages with the data, and parses them. Given the number of incoming messages and the need not to overload the system, this function is launched in a separate process and communicates with the main one with a series of interprocess FIFO queues.

When started, first, this module tries to connect as a client to the DTI. If the connection succeeds the data reading and parsing thread is started, otherwise the system keeps trying to establish the connection. During the operation, the correct functioning of the communication is continuously verified, in case of problems the process tries to re-establish it automatically.

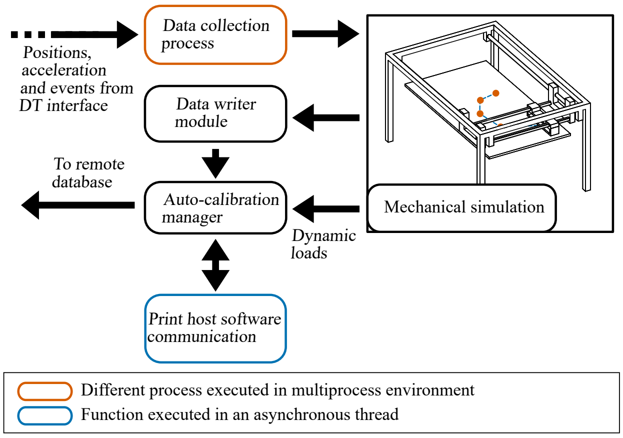

Positions, acceleration and events from DT interface
Data collection process
Data writer module
To remote database
Auto-calibration manager
Dynamic loads
Mechanical simulation
Print host software communication
Different process executed in multiprocess environment
Function executed in an asynchronous thread

Figure 2.12: Diagram of DTC operation in calibration mode with active modules and their interactions outlined.

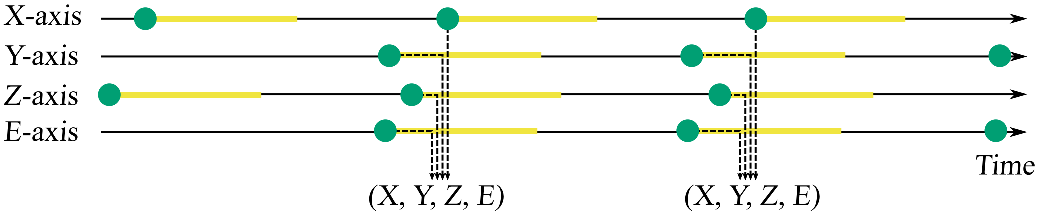


Figure 2.13: In the image can be seen the working scheme of the synchronization algorithm. The green balls placed on the timeline represent the instant in which a data is received, the yellow line indicates the period of validity of the data. It can be seen that the positional data are forwarded only when they are all valid.

The data reading thread acquires the incoming packets, extracts the data, processes them, and directs them to the right queue where they can be used by the main process.

Temperature, start of a new layer, and position data from the vision system are routed to the main output queue and identified with appropriate headers.

Regarding the extruder acceleration data, the vertical acceleration value is discarded as the extruder block can only have a horizontal movement. The acceleration values in X and Y direction are filtered and then sent to the main process via a dedicated queue.

The data coming from the encoder are first converted into millimeters and then synchronized. In fact, since they come from two different cards and pass through an intermediate device (see paragraph 2.4) there is no guarantee on the reception order and timing. However, since the timer of the smart sensors is reset and synchronized at startup, it can be assumed that the positions of the various axes are acquired with a time difference that is much less than the sampling period.

The synchronization algorithm is based on a timestamp that marks the actual time of acquisition and is transmitted together with the data itself. In fact, the position data received is stored but not immediately sent to the main process; it is sent only when the difference between the sampling times of the axes is within a certain threshold, set as a percentage of the sampling time. Once the sending condition is met, the data of each axis are grouped and sent to the main process with a dedicated queue (Figure 2.13).

2.5.2 Marlin emulator

The Marlin emulator is a minimal version of the original C++ script of the printer control system firmware (Marlin version 1.1.9) modified to work on a PC platform and to be compatible with Windows and GNU/Linux.

The original Marlin firmware offers the possibility to control a large number of printers with different geometries and peripherals, however, for the purpose of this work, it was decided to emulate only the part of reading and interpretation of G-code and trajectory planning.

The original program is designed to acquire G-code commands, one line at a time, from a serial connection (which is also used to send log messages) or from an SD card connected to the control device. Regardless of the acquisition source, the commands are sent to the parsing library that executes the functions associated with each command. In case a motion command is received the system forwards it to the trajectory planning module, here a FIFO buffer is populated containing the parameters of the motions to be executed by the motor of each axis (speed, acceleration, number of steps). At this stage, a timed interrupt service routine has the task of reading the instructions and commanding the advancement of the stepper motors while emptying the buffer. The purpose of the buffer is to untie the execution of the movement from the timing of data reception and to allow the execution of optimization algorithms on the movements already programmed.

In the program emulator, the original script has been modified to read data directly from the memory of the computer on which it operates, serial communication has been disabled, and log messages are saved in a special log file. The functions of parsing and trajectory planning have not been touched while the movement parameters are no longer saved on a buffer in RAM but directly on a file on the hard disk (Figure 2.14). The reason for this choice is the fact that it is not possible to emulate the interrupt service routine of the AVR microcontroller effectively without performing hardware emulation. So while in the original version the controller acquires the G-code as it prints, populating a buffer that is then emptied once the movement is executed, in the simulated version the G-code is read all at once and the movement parameters are saved in a binary file, ready to be interpreted later by the digital twin's trajectory planning module or by the printer simulation process.

2.5.3 Printer simulation process

The printer simulation process replaces the data collection process when the DTC is started in simulation mode. Given the high computational load required for simulation,

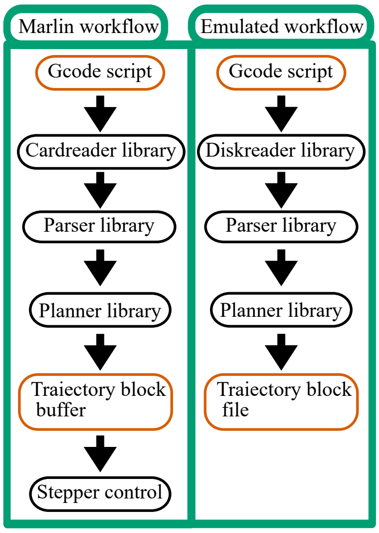


Figure 2.14: The image compares the motion control structure performed by the original printer controller with that of the emulated controller.

this module is executed in a separate process from the main one to which it is connected via interprocess FIFO queues.

The printer simulator is activated when it receives an order from the DTC to start a print job. It first calls the Marlin emulator which processes the desired G-code file and creates the binary file with the motion instructions. Then the binary file is read and the information is used to rasterize the trajectory using the Bresenham line algorithm [98], in this way the feed commands are generated for the four virtual motors that represent the machine axes. The algorithm uses the time provided by the operating system to update the position of the axes as much as possible in real-time, consistent with the accuracy that can be achieved by a non-real-time operating system.

In this way, the simulator is able to send the messages about position, acceleration, and change of layer as if they came from the DTI.

As an alternative mode of operation, the simulator is able to repeat an old print of which the data have been saved. In this mode, the data is simply read from the repository and the corresponding messages are issued with the right timing.

2.5.4 Trajectory planner and error detection

The trajectory planner thread is based on the same principle as the printer simulator but performs its functions during regular mode operation before the actual printing takes place. The simulation is therefore executed all at once, as fast as possible and not in real-time. When the DTC receives the command that marks the start of a print job, the trajectory planner runs the Marlin simulator and analyzes the file with the motion parameters. Unlike what happens in the printer simulator, however, in this case, the movements of the axes are analyzed and stored immediately in order to obtain, for each layer, the information that will be used by the error detection system to identify printing anomalies.

In particular, for each layer the following information is obtained:

- The bounding box: i.e. the space included in a square with sides parallel to the movement of the axes in which the end-effector moves during the execution of the layer.
- The amount of extruded material: that is the length of the extruded filament during the development of the layer.
- The total displacement: the total movement of the axes.
- The height of the layer: i.e. the displacement made by the Z-axis at the beginning of the layer.

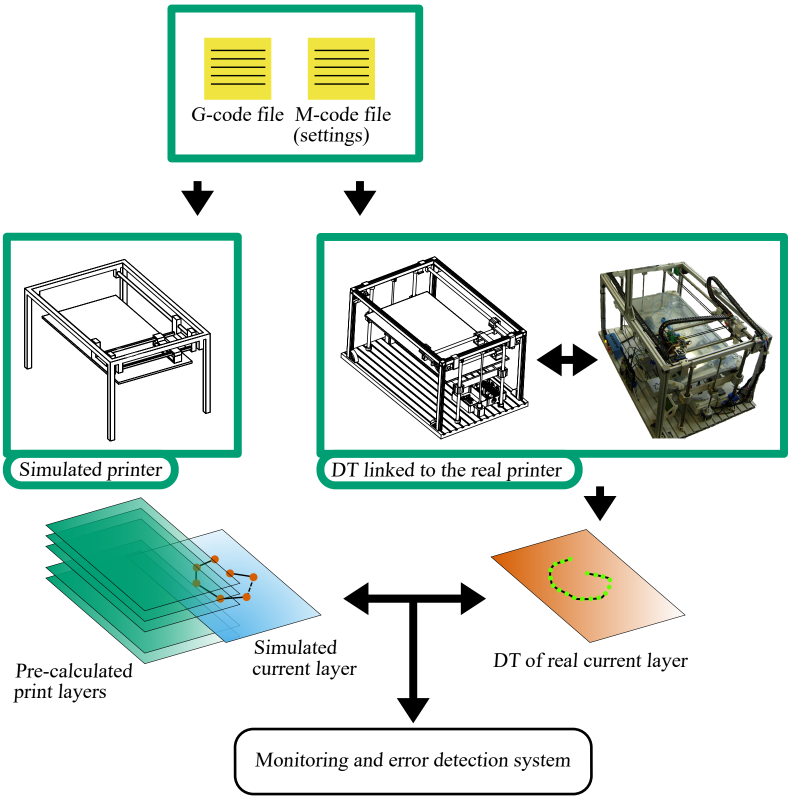


Figure 2.15: In the picture is shown the schematic of the monitoring and error detection system.

During the regular operation of the printer, the information obtained from the simulated axes is constantly compared with those measured on the real axes, in this way it is possible to verify the presence of anomalies in the printing process (Figure 2.15). Currently, the possibility of recognizing the most common anomalies has been implemented:

- Layer mismatch: if a position outside the bounding box is detected for the current layer.
- Abnormal extrusion: if the ratio between the extruded material and the distance traveled for the current layer is different from that predicted in the simulation phase.

When an error is detected, the system reports the event in the log system, moves the extruder to a safe position, and sends a message to the GUI from which a recovery procedure can be performed. The parts wear manager is also notified, which parts might be involved in the malfunction, the status of these parts is changed to NOT WORKING and must then be manually reset by the GUI after a check.

After securing, the system can perform a diagnosis to rule out the failure of some parts in order to reduce the number of checks to be performed. In case of a layer mismatch, the diagnosis is based on the agreement of the position data between the printer controller, the encoders, and the vision system.

For each axis, the first thing to be checked is the correspondence of the position between the controller and the encoders, if it is verified the axis concerned had no problems, the status of its parts is restored and the system can move on to the verification of the other axis. If a discrepancy between the position values is found, it is possible to investigate further only if the vision system is active and if the difference between the value detected by the controller and the encoder is not too small, in fact in this case the low accuracy of the vision system makes its use impossible.

If the above conditions are verified and the position of the camera agrees with that of the encoder, it is deduced that the malfunction is due to one of the components of the motion system; if it agrees with that of the controller, it means that the malfunction is due to a failure of the encoder.

2.5.5 Mechanical model

The mechanical model is the part of the DTC that deals with simulating the physics of the system. The model currently maps the operation of motors and loads and this provides a continuous output of the force delivered by each axis and the maximum force that can be supplied. The modeling methods used are listed below:

- **Stepper motors**: for each stepper motor, the corresponding pull-out curve is interpolated in order to obtain the maximum deliverable force as a function of the operating speed. The curve can be inserted in the configuration files if known or automatically calculated with the auto-calibration procedure (see paragraph 2.5.10).
- **Y-axis carriage and extruder block**: in consideration of the high mass of the carriage, the load necessary to move it has been calculated as purely dynamic and therefore dependent only on the acceleration of the piece.

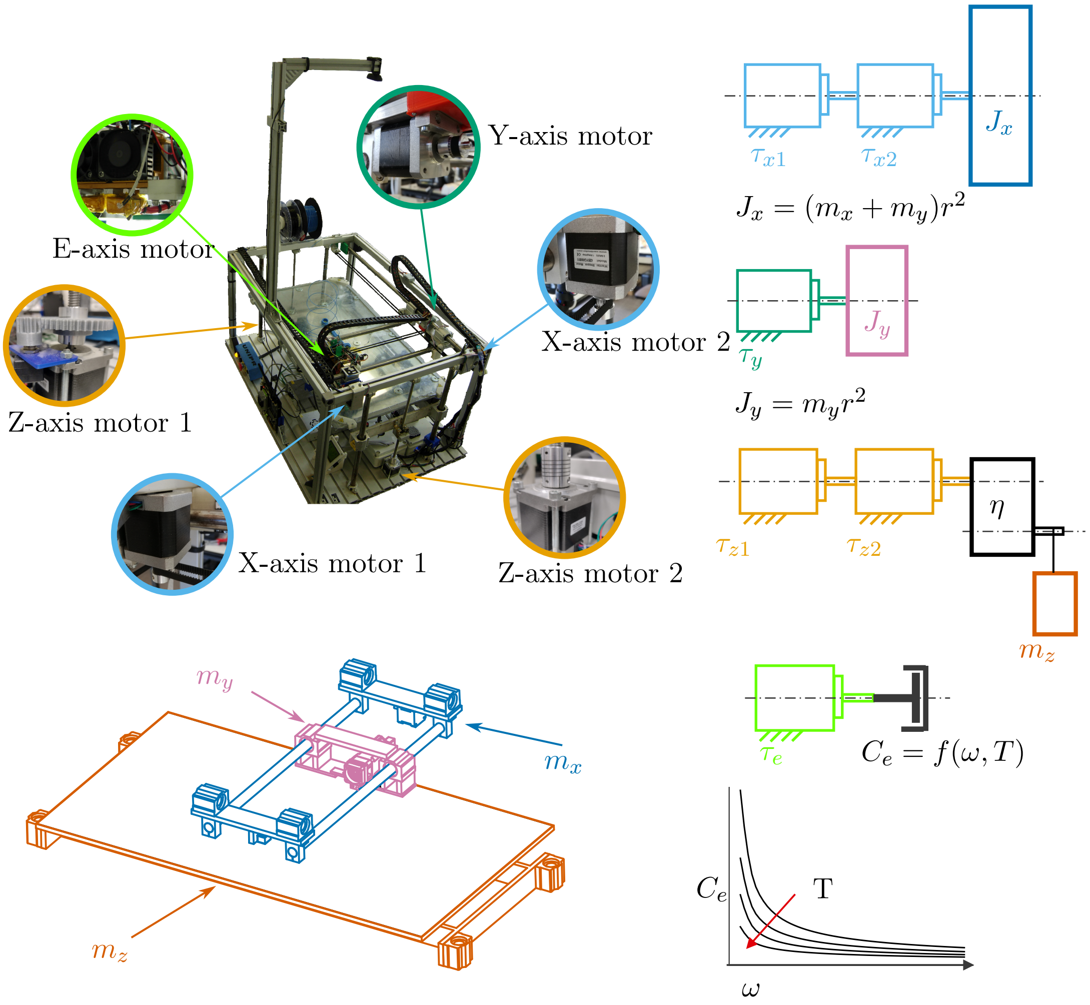


Figure 2.16: Schematic representation of the mechanical model used in the digital twin

- **X-axis carriage**: the same considerations made for the Y-axis are valid, but in this case, it is assumed that the load is divided evenly between the two motors.

- **Z-axis carriage**: in this case, the purely dynamic simulation is not sufficient due to the high friction and the low efficiency of the transmission system. For this reason, an efficiency factor has been introduced for both direct and retrograde motion for a steel screw coupling on lubricated brass. For the calculation of the force of inertia is also taken into account the acceleration of gravity, since the movement is vertical.

- **Extruder**: the load acting on the extruder was modeled as purely viscous, with a viscosity coefficient dependent on both velocity (non-Newtonian fluid) and temperature. To keep computation time short, the viscous coefficient is interpolated from a two-dimensional table generated offline. Values were calculated from a capillary tube model, traversed by non-Newtonian fluid in laminar regime using generalized Reynold number [99] and rheological properties of molten polylactic acid at different temperatures [100].

2.5.6 Parts wear and tear monitor

The parts wear monitor is the system that allows tracking the movement and load acting on the parts, in order to check for wear and tear. During the operation of the printer, this module obtains the rotations of the axes and the sliding of the parts in contact using the position data; moreover, through the load values of the mechanical model, it is able to obtain the energy supplied or dispersed by the various components. The parameters kept under control are:

- The movement carried out by the part, in rotations or millimeters depending on the part analyzed.

- The energy absorbed and delivered or dissipated.

- The actual time of use.

- The time elapsed since the last maintenance, if the component requires it.

- The time elapsed since the part was installed.

The parts monitored with this system are those most susceptible to wear and tear and in need of periodic maintenance/replacement (see Table 2.2 for a complete list) Each part can have three states WORKING, REQUIRES ACTION, NOT WORKING. For each of the monitored wear parameters is set an alert threshold, provided by the manufacturer or dictated by the experience, if one of the parameters exceeds the threshold, the status of the piece goes from WORKING to , REQUIRES ACTION. The status of NOT WORKING instead can be triggered only by the error detection system.

Table 2.2: The table shows the parts and parameters monitored by the wear monitor

		Monitored parameters					
		Number of parts	Movement	Energy	Usage time	Time since maintainance	Time since installation
Monitored Parts	Stepper motors	6	✓	✓	✓		✓
	Motor drivers	5		✓	✓	✓	✓
	Timing belts	3	✓	✓	✓		✓
	Linear bearings	3	✓		✓	✓	✓
	Extruder pipe	1	✓		✓	✓	✓
	Nozzle	1	✓		✓	✓	✓
	Encoders	4	✓		✓		✓

2.5.7 Print Project Manager

The print project manager has the task of initiating print jobs, saving the data of each print in an ordered manner, and managing the machine settings.

Before starting a print job, it is necessary to create a *project*; this includes all the prints made from the same G-code file, regardless of the settings or type of machine used (real or virtual). It is also possible to specify a project *without extrusion* if the G-code file includes only axis movements without the realization of an actual print (projects of this type are used to test the generation of trajectories).

The following data is stored for each project:

- The G-code file of the project.
- The 3D model in STL format that was used to generate the G-code file.
- The ideal virtual model of the printed part. This is the three-dimensional model of the printed part generated directly from the G-code (see paragraph 2.5.9) that represents the ideal result of the printing.

Each project also contains the data of all the prints made using the corresponding G-code. For each printing, the following are stored:

- The volume of extruded material.
- The printer used (real or simulated).

- The maximum ratio between the load delivered by the motors and the load that can be supplied; it indicates how much the axes have been loaded concerning the settings used.

- The average temperature of the extruder and plate during printing, calculated using the data provided by the smart sensors.

- The date the print was made and its duration.

- The evaluation of the adherence of the print made to the ideal printing model calculated with the post-processing tool of the GUI (see paragraph 2.6.4).

- The settings used for printing. In fact, before printing it is possible to define the machine settings to be used, these are then saved in the form of an M-code file.

- The history of position and acceleration data received during printing. The collection of this type of data must be enabled by the GUI before starting the job.

- The virtual model of the printed object, i.e. the digital twin of the part (see paragraph 2.5.9).

- The LOG of the print.

- The status of the actuators at the beginning and end of the print job.

All this information is saved on a remote memory that is accessible to both the DTC and the GUI, so the user can consult it from the interface even if the DTC is not active or is disconnected. This information can be used to correlate the status of the machine and the print settings with the results of the printing itself with a view to optimization and diagnostics.

The settings that are saved and that can be modified before launching the print are those that most influence the printing time, the quality of the piece, and the possibility of error:

- **The step/mm ratio for each axis**: these parameters depend on the motor used, the driver settings, and the transmission ratio of each axis. They are not normally changed, but it is possible to increase or decrease that of the E-axis if an over-extrusion or under-extrusion effect is desired.

- **The maximum acceleration for each axis**: these parameters determine the maximum acceleration achievable by each axis. These values have a great influence on the duration of the prints that include many short lines, but also on the possibility of errors occurring as the force required from the motors increases. It is possible to use the auto-calibration system of the digital twin to find a suitable value for the X, Y, and E axes.

- **The maximum speed for each axis**: these parameters determine the maximum speed that can be reached by each axis. These values have a great impact on the time of prints that include long lines and fast movements, but increase the possibility of errors because as the rotation speed increases, the maximum force that can be delivered by the motors decreases. As for the accelerations, the maximum speeds for the X, Y, and E axes can be determined in the auto-calibration procedure

- **Printing acceleration, filament displacement, and retraction**: these parameters are not related to a single axis but the movement as a whole. These parameters are generally adjusted in consideration of the quality to be obtained in the piece. In fact, the increase of the maximum accelerations causes a worsening in the dimensional accuracy, especially along the curved sections and in the corners as the high forces involved stress the elasticity of the motion transmission system.

- **Jerk for each axis**: the controller of the printer does not interpret the jerk in a physical sense but sees it as the minimum speed variation for which an acceleration phase is not required. A high Jerk can decrease the printing time but increases the possibility that a motor misses steps during a speed change.

- **Percentage speed factor**: this is the option that is changed most often if the printing speed or the quality of the product is to be increased as it acts as a multiplying factor on all the speeds set in the G-code file. In fact, a high printing speed decreases the duration of the job but causes a worsening of the surface finish and dimensional accuracy as it does not allow time for the layers to solidify properly.

The system that collects position and acceleration data during printing can be used to perform post-process analysis or to repeat the process using the simulation function. The collected data is stored in HDF5 files by the PyTables library [101], which specializes in efficiently managing datasets that include large amounts of data.

2.5.8 Communication

DTC communications take place over three distinct channels: communication with the print host process, with the DTI, and with the GUI.

As for the communication with the DTI, it is one-way (receive only) and is explained in section 2.5.1. Communication with the software print host is done via an asynchronous thread that implements Octoprint's REST API library. The process is able to send commands to the printer regarding movement, temperature, start or stop of a print job and the settings to be used. Similarly, it continuously collects data about controller temperatures, target temperatures, and the status of the current print job.

Communication with the GUI is based on a series of coupled variables in the main DTC process and in the DT GUI process. If one of these variables is changed in either environment, the system automatically sends a packet to the other environment to update the value of the coupled one. The communication system works synchronously with the main process and is responsible for detecting changes to the observed variables, creating and sending messages with the updated values, receiving messages updating the corresponding variables, and executing any associated callback functions.

All the main modules of the DTC have a corresponding interface in the GUI to which they are connected in order to show information or receive user input.

2.5.9 Virtual part generator

During printing, the virtual part generator uses axis position data to create a real-time 3D model of the piece being printed. Part of the data obtained is sent in real-time to the GUI so that the user can visually follow the printing process, the model displayed however has a lower density of points than the one generated by the DTC which can be accessed only once the printing is finished.

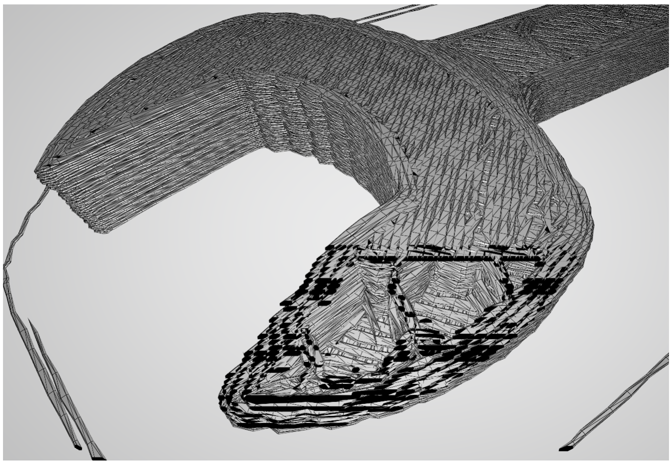


Figure 2.17: View of the 3D model obtained by printing a real piece with cutaway to show the infill.

The idea behind the three-dimensional representation of the part is similar to what has been done by Yi et al. [64] who build the printed piece by modeling extruded filaments with cylindrical bodies. In this work, however, it was decided to use the modeling proper to slicing algorithms [102] that see the extruded filament as a rectangle of height equal to that of the layer with two semicircles applied to the ends (see Figure 2.18). This model, also used by Moretti et al. [83] for the identification of the boundary of the layer edges, has the advantage of reliably determining the thickness of the extruded sections starting from the height of the layer, the space covered, and the amount of extruded material.

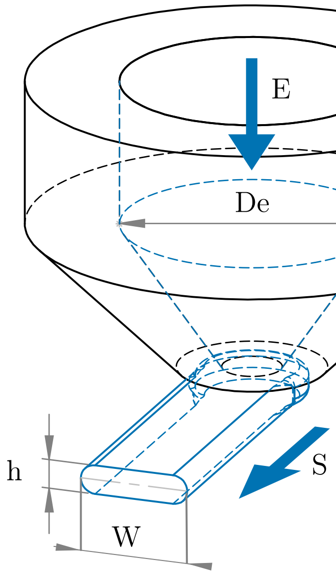


Figure 2.18: Nozzle and extruded filament model

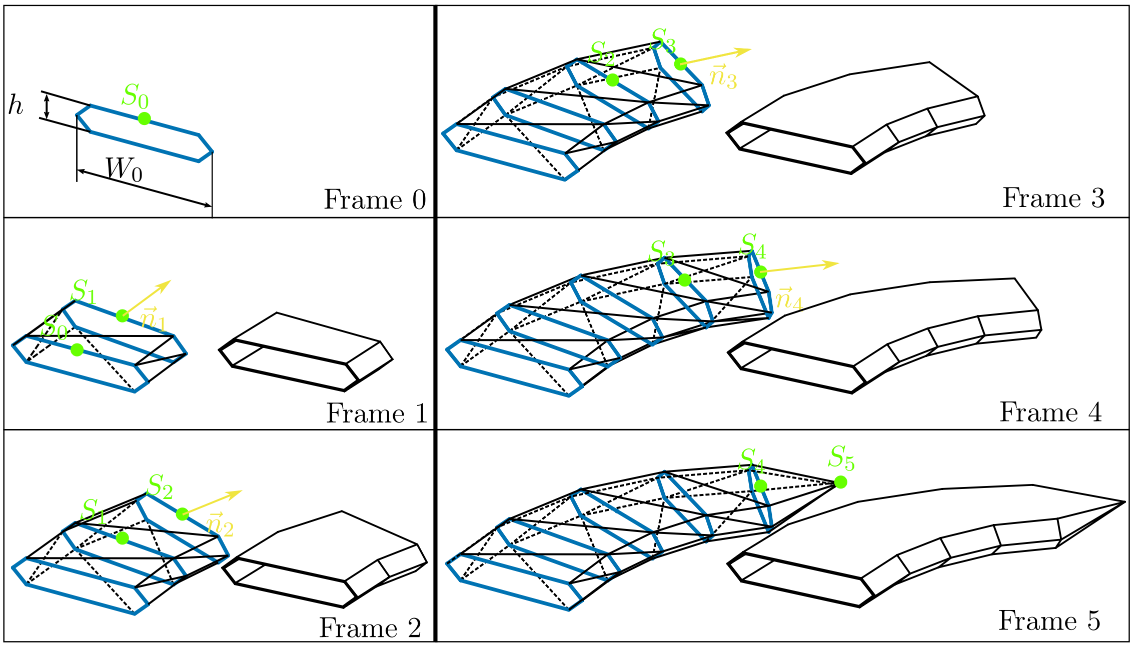


Figure 2.19: In the image is shown, in all its phases, the construction of the model of an extruded filament sampled in 6 points.

By imposing volume conservation between the extruded material and that contained in the filament, the width W of the filament can be calculated using the formula (valid for $W > h$ only):

$$W = h\left(\frac{\pi}{4}\left(\frac{D_e^2}{h^2}\frac{dE}{dS} - 1\right) + 1\right)$$

where D_e is the diameter of the filament, h is the height of the current layer, E is the displacement of the extruded material and S is the movement of the nozzle relative to the plane.

The model of the printed part is represented with a triangular mesh, then, to keep the total number of nodes within an acceptable limit, the 3D model generator approximates the section of the filament by replacing the two semicircles with inscribed right triangles and obtaining a hexagonal section.

During the execution of the program, the system receives at regular time intervals a vector containing the position of axes; when at the instant i a new point (x_i, y_i, z_i, e_i) is received the algorithm calculates the section of the filament that joins it to the previous one, using the width W_i and the versor $\vec{n}_i$ calculated with the following formulas:

$$W_i = h_i\left(\frac{\pi}{4}\left(\frac{D_e^2}{h^2}\frac{e_i - e_{i-1}}{|\vec{S}_i - \vec{S}_{i-1}|} - 1\right) + 1\right)$$

$$\vec{n}_i = \frac{\vec{S}_i - \vec{S}_{i-1}}{|\vec{S}_i - \vec{S}_{i-1}|}$$

where $\vec{S}_i = [x_i, y_i, z_i]$.

The i-th section, corresponding to the instant i, is generated below the point $\vec{S}_i$ (which indicates the position of the nozzle tip) as a hexagon lying on the plane defined by the vector $\vec{n}_i$, with a base $W_i - 2h$ always parallel to the horizontal plane and a height h. Then the hexagon constituting section i is joined to that of section $i - 1$ by 12 triangles that form a sort of tube (see Figure 2.19). These triangles constitute the mesh that models the geometry of the printed piece. In practice, the representation of the filaments is never an open figure (such as in the image) because in the points of zero extrusion (at the beginning and end of each extruded filament) the hexagonal section collapses to a point closing the tube that is formed (see Figure 2.19, frame 5).

In addition to what has been described, some features have been added to the modeling of the extrusion system in order to make the model closer to reality:

- If, for a section i, it is verified that $W_i < h_i$ it is assumed that the amount of extruded material $e_i - e_{i-1}$ is not sufficient to generate an extruded filament for the corresponding section. In this case, is imposed that $W_i = 0$ (the section collapses at one point) and $e_i = e_{i-1}$ because the little material pushed by the extruder has not been used but has accumulated at the nozzle inlet and will cause an increase in the thickness of the next section.

- If, for a section i, it occurs that $|\vec{S}_i - \vec{S}_{i-1}| < 0.1$ mm the iteration is skipped. This prevents nodes from accumulating in sections where the extruder is stationary or proceeding slowly and gives more uniformity to the mesh size.

- To limit the error due to the different resolution of the encoders for each section the thickness W_i found is averaged according to a weighted average over the traveled space with a 2 mm moving window.

The extruder modeling system, in addition to generating the model of the molded part, provides the GUI with information about the extruder status and extrusion width.

2.5.10 Auto-calibration module

The auto-calibration module is automatically activated if the system is started in auto-calibration mode. In this mode, it creates critical situations in a controlled way to check the limits of the actuators and estimate their characterization, then using the mechanical model it is possible to determine safe settings for the axes in order to avoid printing errors [88].

This mode is designed to be performed only in case of substantial changes to the mechanical or control system (change of actuators, transmission mechanisms, electronics, or masses involved), for this reason, it does not communicate with the user through the DT GUI but through a simple command-line interface running directly from the DTC.

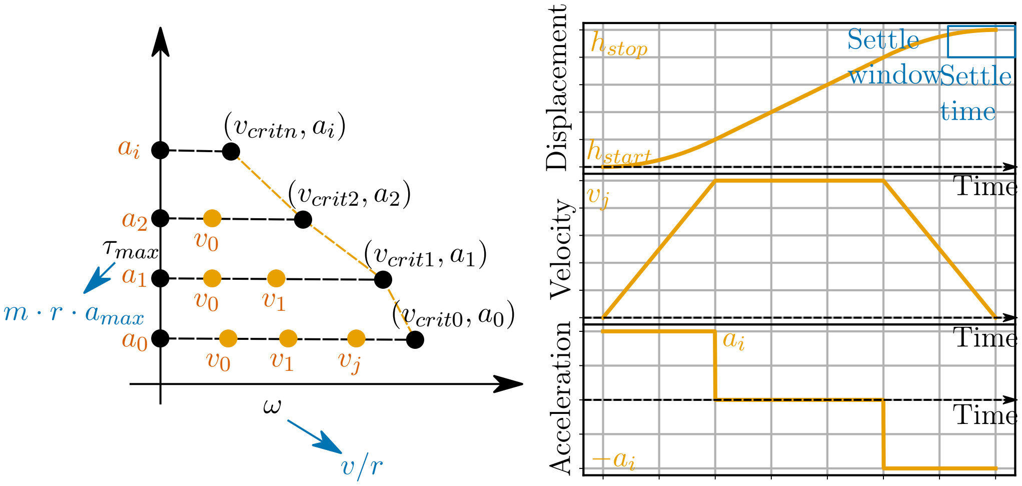


Figure 2.20: The picture shows the procedure adopted by the digital twin for plotting the pull-out curve of a stepper motor with an inertial load. On the left, it can be observed that the curve can be seen as a function only of the velocity and acceleration of the axis and plotted knowing the critical values of velocity at each acceleration. On the right is represented the trajectory used for the tests

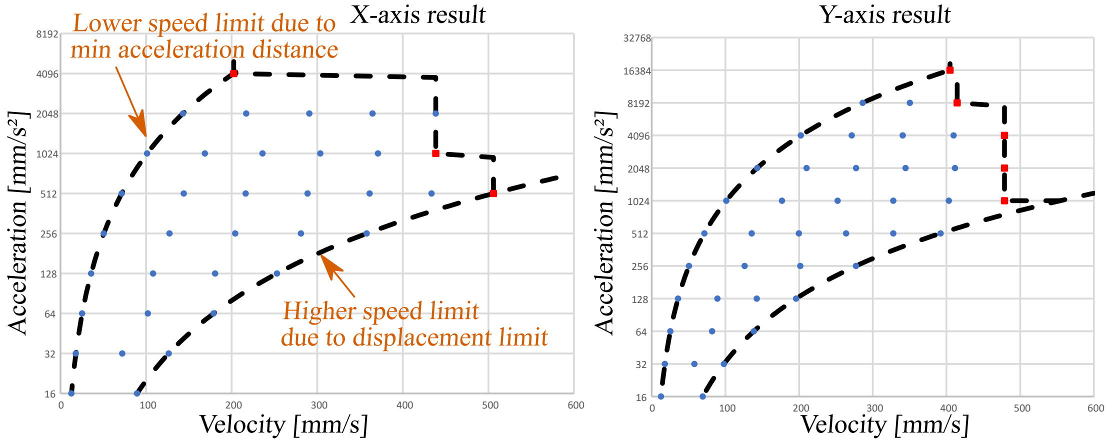


Figure 2.21: The image shows the tests performed for the X and Y axes and lines delineating the area of the graph where tests can be performed. A blue dot indicates successful tests, a red dot indicates failed tests.

In the current configuration, the calibration procedure is used to estimate the pull-out curves of the stepper motors of the X, Y, and E axes and to determine the most convenient kinematic settings. The Z-axis has been excluded from the analysis because its speed is not very relevant to the printing time (it moves a few tenths of a millimeter for each layer) and therefore conservative settings can be used. The procedure is fully automatic but requires user confirmation at the beginning of the calibration phase of each axis as it is necessary that the printing plane is clear and the extruder is free to move.

The characterization feature is very useful because it is not always easy to obtain the pull-out curve of a stepper motor, especially if it is a low-end one. In fact, the curve depends very much on the control system used (type of driver, use or not of microsteps, supply voltage) so, even if it is provided, it is not always applicable to the case in use.

When the calibration of the X-axis or the Y-axis is started, the system sends movement commands for the axis involved, with accelerations and speed peaks that are increasing in order to raise the load acting on the motor. The program, meanwhile, monitors the encoder positions and verifies if the trajectory is executed correctly, in case of failure it is assumed to have reached the speed or acceleration limit of the motor.

The speed profile of the generated trajectories is trapezoidal with a constant acceleration phase, a maximum speed phase, and a deceleration phase. The purpose of the acceleration and deceleration phases is to generate a load on the motor, the purpose of increasing peak speed is to gradually decrease the torque that can be delivered by the motor up to the point where there it misses steps and therefore the operating limit is reached (Figure 2.20).

The limitation of this approach is the fact that acceleration and speed cannot be set arbitrarily but the generated trajectory must be subject to certain constraints:

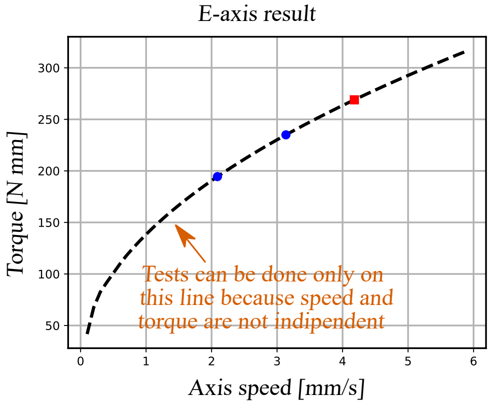


Figure 2.22: The image shows the tests performed for the E-axis. A blue dot indicates successful tests, a red dot indicates failed tests.

- Maximum usable space depends on the length of the axis.
- The acceleration and deceleration phase cannot be longer than 8 s, as this causes problems to the printer controller firmware.
- The acceleration must be maintained for a sufficient distance to overcome the backlash and elasticity of the kinematic chain.
- The maximum speed allowed for a test at a given acceleration must be in any case lower than the one that caused a failure at a lower acceleration.
- After the acceleration ramp, the axis maintains the set speed for a certain time. A maximum value relative to this time is set to reduce the duration of the test.

This limits the motor operation to the area shown in Figure 2.21.

At the start of the procedure, starting from an acceleration a_0, trajectories with increasing peak velocities are generated. The motion is considered correctly completed if a position within a settling window placed at the ideal end position is detected by the encoder and if it is maintained for a settling time (in this way a failed deceleration is also considered a failure).

When a trajectory fails or the maximum possible velocity is reached, the trajectory is changed to a higher acceleration and restarted from a low velocity. It is possible to plot the pull-out curve by points considering the critical speed for each acceleration. If a trajectory fails at the lowest possible speed, it is assumed that the load corresponding to the acceleration used is the static motor load and the test ends.

For E-axis calibration the considerations are different since the load acting on the motor depends mostly on the axis speed. Also, it is not possible to achieve rotational speeds comparable to other motors. For these reasons, it is not possible to plot a true pull-out curve but the results are limited to static torque.

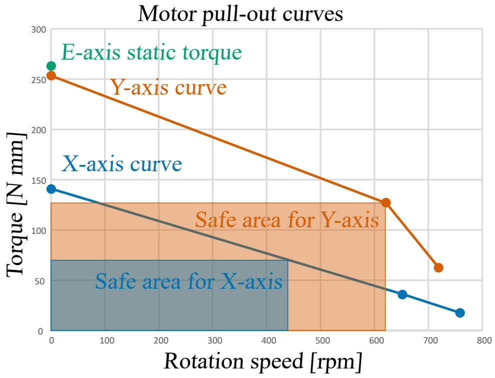


Figure 2.23: The image shows the curves obtained from the calibration procedures and the safe operating zones that are used to define the acceleration and maximum axis speed.

Each E-axis test consists of a long extrusion of molten PLA at 200°C at a constant speed. As with the other axes, the test is considered successful if the final position is reached and held for a certain time. Tests are conducted at increasing speed, in case of failure the same test is repeated by decreasing the acceleration (Figure 2.22).

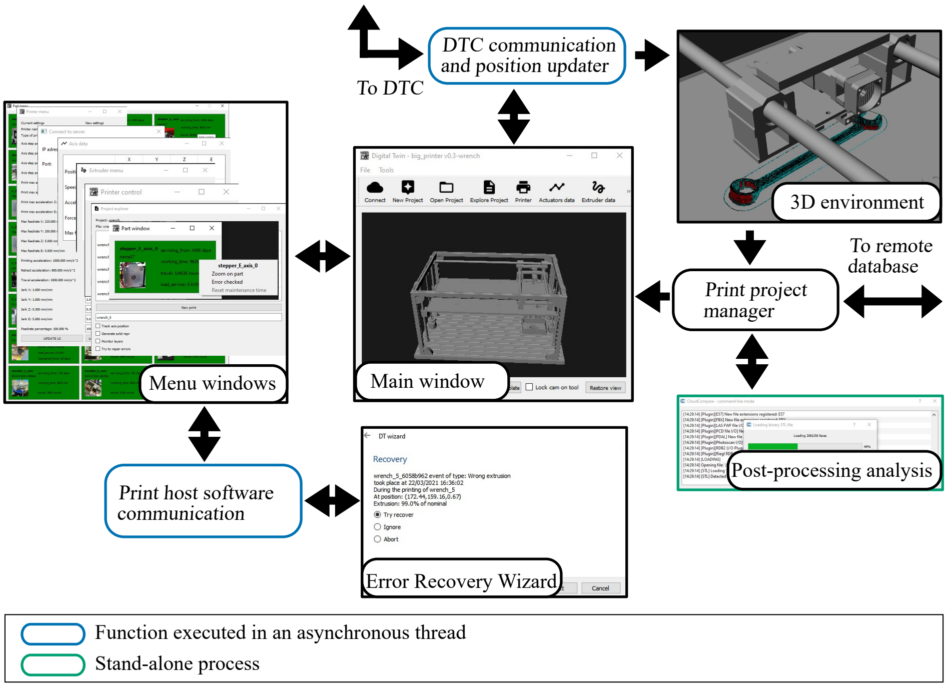
Using the results obtained from the calibration procedure, the system traces the pull-out curve of the motors and obtains the limit values of acceleration and speed to be used as print settings to stay in a safe zone and reduce the risk of overloading the motors (Figure 2.23).

2.6 Digital Twin GUI

The GUI of the digital twin is the program that acts as an intermediary between the user and the core of the digital twin, it also offers some features that are directly related to human-machine interaction such as wizards for project creation and error resolution and, in addition, allows to perform post-processing operations and visualize stored data (Figure 2.24). The GUI is designed to run from a remote computer and thus extend the control capabilities of the digital twin.

The GUI is based on a main thread, on which the user interface operates, a secondary thread that has the task of managing communication with the DTC and updating the value of variables, and a last thread that deals with communication with the print host software (see section 2.5.8).

The user interface has been developed with the PyQt5 library [103]; it includes the menus and the 3D environment window. The menus interact with the DTC modules through the communication system, allowing to read data and send commands (for the communication with the DTC see paragraph 2.5.8). The 3D environment instead allows the user to see the reconstruction of the digital twin of the printer and the printed part in an environment similar to reality, also it is possible to visually check the status of the



DT wizard

Recovery

wrench_5_6058b962 event of type: Wrong extrusion
took place at 22/03/2021 16:36:02
During the printing of wrench_5
At position: (172.44,159.16,0.67)
Extrusion: 99.0% of nominal

● Try recover
○ Ignore
○ Abort

Figure 2.24: Schematic representation of the DT GUI structure.

parts as they flash if they are in a warning state.

2.6.1 The 3D environment

The 3D environment has been built with a Python wrapper of the Panda3D graphics engine [104] integrated to run in a Qt environment [105].

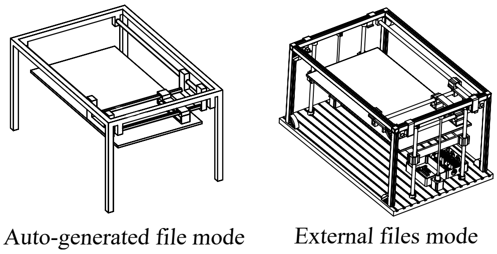
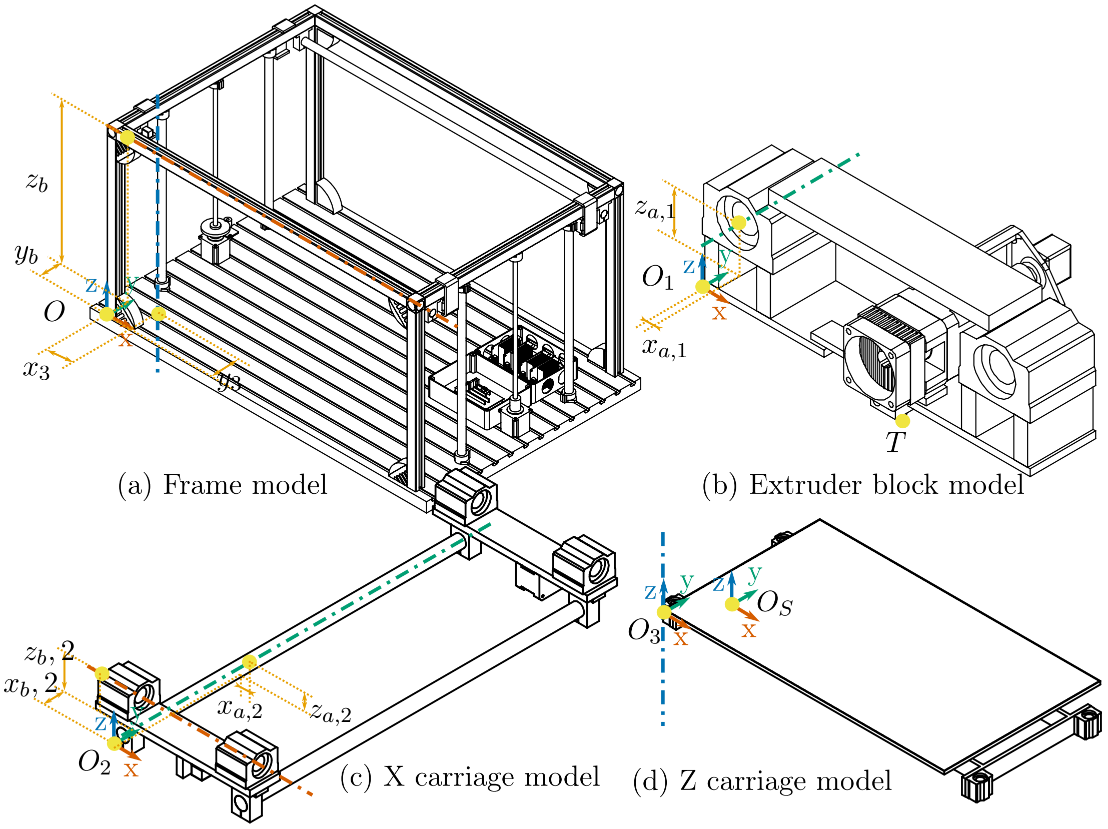


Figure 2.25: Visual comparison between solid object generation modes

The environment can be used in two different configurations (Figure 2.25):

- **External file mode**: in this mode, the 3D models of the visible objects must be supplied to the program in STL format after being created separately with any CAD software.

- **Autogenerated file mode**: In the absence of 3D models of the parts, the system is able to automatically generate an elementary geometry for the objects using OpenScad [106] and the dimensional parameters provided during installation.



(a) Frame model

(b) Extruder block model

(c) X carriage model

(d) Z carriage model

Figure 2.26: In the images are visible the 3D models of the 4 main bodies with marked the origins of the axes and the points that need to be measured to determine their kinematics.

The printer model consists of four main bodies: the frame, the Z carriage, the X carriage, the extruder block (see Figure 2.5). Each of these parts has some spatial points that must be defined (by measuring distances on the real machine) to allow the kinematic to be set. Then there are secondary bodies (such as parts whose state is monitored or the virtual model of the printed part) that move in conjunction with the main ones.

The frame includes all fixed parts of the printer chassis. All parts that do not move, such as the drivers, motors, and encoders of the X and Z axes, are attached to the frame. Its reference system O coincides with the fixed reference system of the graphic environment and is placed in correspondence of the junction between the base plate and one of the vertical supports, the axes are oriented like the printing ones (see Figure 2.26(a)). In the reference system of the frame, it is necessary to define, through the values x_b and y_b, the point through which the axis of the nearest linear guide for the carriage X passes. In addition, the position of the vertical axis through the origin O_3 of the Z carriage reference system (coordinates x_3 and y_3) must be measured.

The extruder block (or Y carriage) represents the parts of the printer that move in the Y direction relative to the X-axis carriage. Attached to the extruder block are the nozzle, Teflon tube, Y-axis bearings, E-axis motor, and encoder. The origin O_1 of the block is in the corner at the lower edge of the support plate, the axes are oriented as the printing axes. In this reference system, it is important to measure the distance $x_{a,1}$ and $z_{a,1}$ to the geometric axis corresponding to the Y-axis bearing and the position of point $\vec{T_1}$ corresponding to the nozzle tip (Figure 2.26(b)).

The X carriage represents the parts of the printer with horizontal movement along the X-axis of printing. Also moving integral to this body are the X-axis bearing, and Y-axis motor and encoder models. The X carriage has an O_2 origin that is located in the corner at the bottom edge of the slide bar bearing, the reference system is aligned with the printing system. The important quantities, in this case, are the position of the axis corresponding to the X-axis slide bearing ($y_{b,2}$, $z_{b,2}$) and the position of the axis corresponding to the Y-axis cylindrical guide ($x_{a,2}$, $z_{a,2}$) (Figure 2.26(c)).

The Z carriage includes all parts of the printer that have vertical motion. In addition, the Z-axis bearings are attached to these parts. The origin O_3 of the carriage is in one of the corners in the lower edge of the printing plane while the axes are oriented as the printing axes (see Figure 2.26(d)). In the carriage reference system, it is necessary to define the position of the origin of the printing plane (as seen by the controller) with the vector $\vec{S_3}$

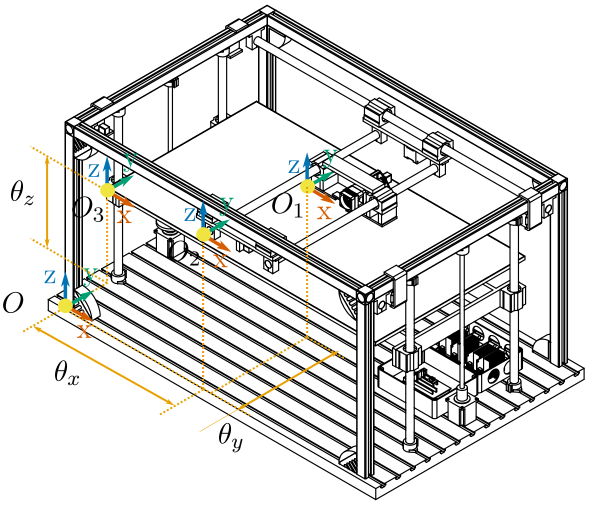


Figure 2.27: The image shows the assembly of the 3D models with the origin of the axes marked and the definition of the components of $\vec{\theta}$.

As visible in Figure 2.27 a $\vec{\theta}$ vector is defined that describes the position of the actuators of the X, Y, Z axes. Starting from $\vec{\theta}$ it is possible to define the position of the origin of all bodies with pure translations:

$$\vec{O_1} = \vec{K_1} + \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \vec{\theta}$$

$$\vec{O_2} = \vec{K_2} + \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \vec{\theta}$$

$$\vec{O_3} = \vec{K_3} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \vec{\theta}$$

With:

$$\vec{\theta} = \begin{pmatrix} \theta_x \\ \theta_y \\ \theta_z \end{pmatrix}, \vec{K_1} = \begin{pmatrix} x_{a,2} - x_{a,1} \\ 0 \\ z_b - z_{b_2} + z_{a_2} - z_{a_1} \end{pmatrix}, \vec{K_2} = \begin{pmatrix} 0 \\ y_b - y_{b,2} \\ z_b - z_{b,2} \end{pmatrix}, \vec{K_3} = \begin{pmatrix} x_3 \\ y_3 \\ 0 \end{pmatrix}$$

During printer operation all positions coming from the DTC are provided in the reference system of the printer controller O_s. It is necessary to define a transformation that, starting from any point $\vec{P_s}$ defined in the printing plane (therefore according to the reference system O_s) allows to obtain the vector $\vec{\theta}$ and from there the positions of all bodies. Given the Cartesian structure of the machine, it is possible to use only translations:

$$\vec{\theta} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \cdot \left(\vec{P_s} + \vec{S_3} + \vec{K_3} - \vec{T_1} - \vec{K_1} \right)$$

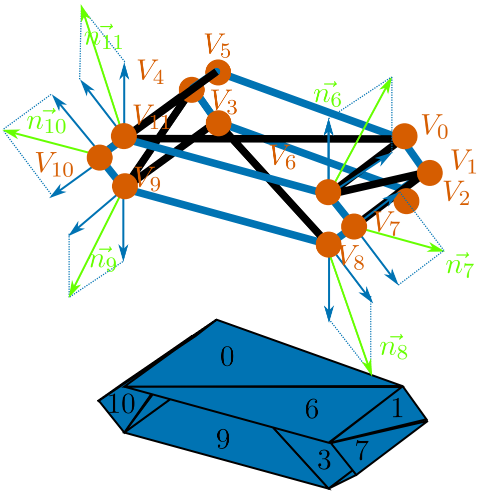
Where $\vec{S_3}$ in the position of O_s in the O_3 coordinates system. During the operation of the system, the position updater calculates $\vec{\theta}$ at regular intervals and, from that, the positions of all main bodies.

2.6.2 The 3D model of the printed piece

The 3D model of the part is generated from the data provided by the DTC with the same algorithm described in paragraph 2.5.9, the data provided allows the recreation of a three-dimensional mesh similar to that generated by the DTC but with a lower number of nodes. Once the printing is complete, it is possible to load the higher density representation for postprocessing operations.

The three-dimensional geometry of the part is generated procedurally by updating the list of vertices and normals and then defining triangular primitives to join them. In each hexagonal section, the vertices' normals are defined by summing the normals of adjacent segments (Figure 2.28).

GeomVertexData		
ID	Vertex	Normal
0	V_0	$\vec{n_0}$
1	V_1	$\vec{n_1}$
2	V_2	$\vec{n_2}$
3	V_3	$\vec{n_3}$
4	V_4	$\vec{n_4}$
5	V_5	$\vec{n_5}$
6	V_6	$\vec{n_6}$
7	V_7	$\vec{n_7}$
8	V_8	$\vec{n_8}$
9	V_9	$\vec{n_9}$
10	V_{10}	$\vec{n_{10}}$
11	V_{11}	$\vec{n_{11}}$



GeomTriangles	
ID	Vertex IDs
0	11, 0, 5
1	6, 1, 0
2	7, 2, 1
3	8, 3, 2
4	9, 4, 3
5	10, 5, 4
6	0, 11, 6
7	1, 6, 7
8	2, 7, 8
9	3, 8, 9
10	4, 9, 10
11	5, 10, 11

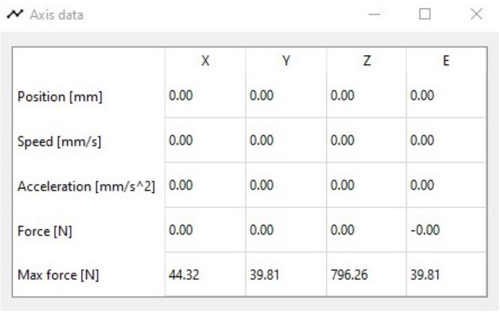
Figure 2.28: The image shows the selection of vertices and normals for procedural creation of solid elements in the graphical environment.

2.6.3 The interfaces

The GUI provides some window interfaces through which it is possible to read the values processed by the modules of the DTC and interact with them. If the DTC is started in simulation mode the windows associated with disabled modules are not available.

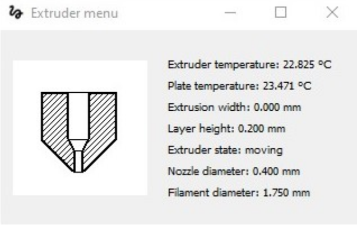
Through the *Actuators data* window it is possible to consult in real-time the data elaborated by the mechanical model (paragraph 2.5.5) of the DTC as position, speed, acceleration, force acting on the axis, maximum force deliverable by the axis (Figure 2.29).

Through the *Extruder data* window it is possible to check the temperature of the extruder and of the plate (measured by smart sensors) and the data obtained from the extruder model that manages the generation of the virtual part such as the width of the



	X	Y	Z	E
Position [mm]	0.00	0.00	0.00	0.00
Speed [mm/s]	0.00	0.00	0.00	0.00
Acceleration [mm/s^2]	0.00	0.00	0.00	0.00
Force [N]	0.00	0.00	0.00	-0.00
Max force [N]	44.32	39.81	796.26	39.81

Figure 2.29: The window that allows to see the status of the actuators



Extruder temperature: 22.825 °C
Plate temperature: 23.471 °C
Extrusion width: 0.000 mm
Layer height: 0.200 mm
Extruder state: moving
Nozzle diameter: 0.400 mm
Filament diameter: 1.750 mm

Figure 2.30: The window that allows to see the status of the extruder

extruded section and the status of the extruder (Figure 2.30).

With the *Parts data* window it is possible to visualize the output of the parts wear monitoring system (paragraph 2.5.6). From this interface it is possible to check the status of the parts and the value of the monitored parameters, moreover, for each part, useful information is shown such as the model name, the picture, visible in the thumbnail, and eventual notes, visible in the tooltip. It is possible to recall the information related to a single piece directly with a double click on its model in the 3D environment. When the status of the part is WORKING its box in the menu is green and the part is gray in the 3D environment, if the status changes to REQUIRE ACTION the box of the part becomes yellow while the 3D representation flashes with the same color, also the parameter that has exceeded the maximum threshold is marked in bold (Figure 2.31). If the status of the piece changes to NOT WORKING as a result of a detected error, the box becomes red while the 3D model flashes with the same color. Clicking with the right button on the box of a part opens a contextual menu that allows the user to perform three operations:

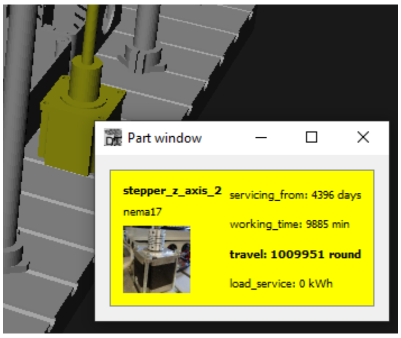
- **Zoom in on the part**: the viewpoint of the 3D environment of the GUI moves to focus on the target part, this can be useful to identify it on the real printer.

- **Confirm check done**: if the part has the status of NOT WORKING after the detection of an error, it is possible, through this command, to confirm that the functionality of the part has been verified and restore the previous status.

- **Confirm maintenance operation**: through this command it is possible, for the parts that support it, to reset the timer that measures the time elapsed since the last maintenance operation.

Through the *Printer Menu interface*, the user can see information about the printer in use such as its name, or its status as a virtual or simulated printer. It is also possible to view the settings and modify them (for the list of implemented settings see paragraph 2.5.7).

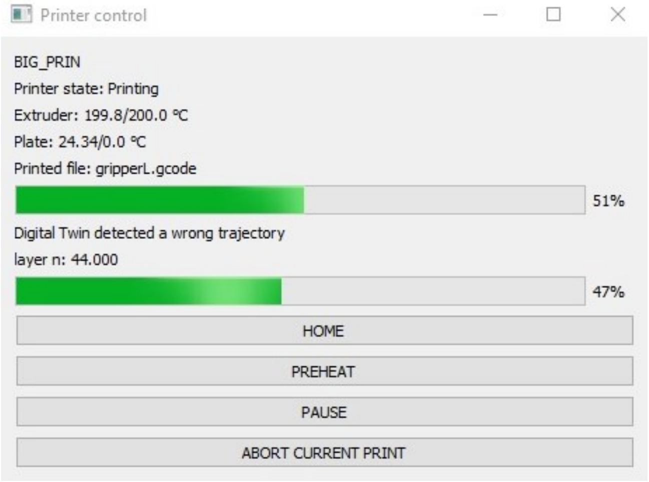
The *Printer control* window changes according to the status of the current print job. In general, it allows the display and sending of data related to the print host software. When there are no prints in progress the user can view the printer status, the current temperatures detected by the controller, and the target temperatures, moreover, it is



stepper_z_axis_2
nema17

servicing_from: 4396 days
working_time: 9885 min
travel: 1009951 round
load_service: 0 kWh

Figure 2.31: The image shows one of the Z-axis motors in a REQUIRE ACTION state for exceeding the maximum number of rotations.



BIG_PRIN
Printer state: Printing
Extruder: 199.8/200.0 °C
Plate: 24.34/0.0 °C
Printed file: gripperL.gcode
Digital Twin detected a wrong trajectory
layer n: 44.000

Figure 2.32: The printer control window during a print job.

possible to start the homing of the axes or start the machine preheating. During a print job the user can see the name of the G-code file that is being printed, the percentage of completion of the print job (calculated by the print host software), the current layer and its percentage of completion (calculated by the path planner), the status of the DTC (if it is loading or has detected an error). It is also possible to pause or stop the print (Figure 2.32).

Through the *New project wizard* it is possible to create the folder and files associated with a new project (see paragraph 2.5.7). During the project creation, the system runs the algorithm for the generation of virtual parts using as input the positions taken directly from the G-code file (without going through the Marlin simulator and trajectory planner). This 3D model is used as an ideal reference for the evaluation of the geometric conformity of the parts.

Through the window of the *Project explorer* it is possible to observe the print jobs carried out within the current project and to start a new one by selecting the features that are to be activated. For each job, the user can see the type of printer used, the printing time, the concordance of the job with the ideal model for the project (average and standard deviation). Right-clicking on the line of one of the printouts opens a contextual menu that allows various operations:

- The visualization of further information such as the start and end date of the print, the extruded volume, the maximum ratio between the load delivered by the axes and the load that can be supplied, the average temperature of the extruder and the plate, the settings used at the time of printing.
- The removal of the print job from the project and the deletion of its data.
- The replay of the print, only if the DTC is started in simulation mode.
- The visualization of the 3D model of the printed part with marked in red the points characterized by a high error (only if the post-processing has already been done)
- The execution of the post-processing operations for the current print.

2.6.4 Post-processing and geometric conformity assessment

Once the printing of a piece has been completed, it is possible to perform an automatic post-processing operation that results in an index of the geometric conformity of the part and allows deviations to be visualized. To perform the analysis, Cloud Compare [107] is used, a software specialized in processing and analyzing clouds of points and 3D meshes.

The analysis program is launched automatically from the GUI using its command-line interface, all options are set automatically, and the results are viewable at the end of the analysis.

The analysis consists of several phases:

- **Import phase**: in this phase, the mesh of the ideal print is imported as well as the mesh of the real part.
- **Alignment phase**: due to inaccuracies in the homing system of the axes, it can happen that the ideal model and the real one are slightly shifted. For this reason, before the real analysis, an automatic alignment is carried out through the iterative closest point registration algorithm with 3 degrees of freedom (translation along the 3 axes). The process stops when the difference in positioning error between two successive iterations is less than a predefined threshold.
- **Analysis phase**: during the actual analysis phase, for each node of the real part, the distance (with sign) to the mesh of the ideal part is calculated. The results are stored in a scalar field associated with the nodes of the real mesh.
- **Post-analysis phase**: this phase is not performed by the CloudCompare instance but directly by the GUI. The data of the scalar field obtained in the analysis

phase are acquired, the sign is removed and the mean and standard deviation are calculated. The mean obtained in this way constitutes an evaluation of the geometric conformity of the part, the standard deviation indicates the dispersion of the results. The absolute data field is now saved in association with the mesh of the virtual representation of the part in PLY format (Polygon File Format).

The PLY file obtained from post-processing operations can be used for further analysis with specific software, or it can be opened directly from the DT GUI from the Project Explorer menu, in this case, the algorithm reconstructs the mesh point by point but setting the red color for the nodes which are associated with a value greater than the average plus standard deviation (Figure 2.33).

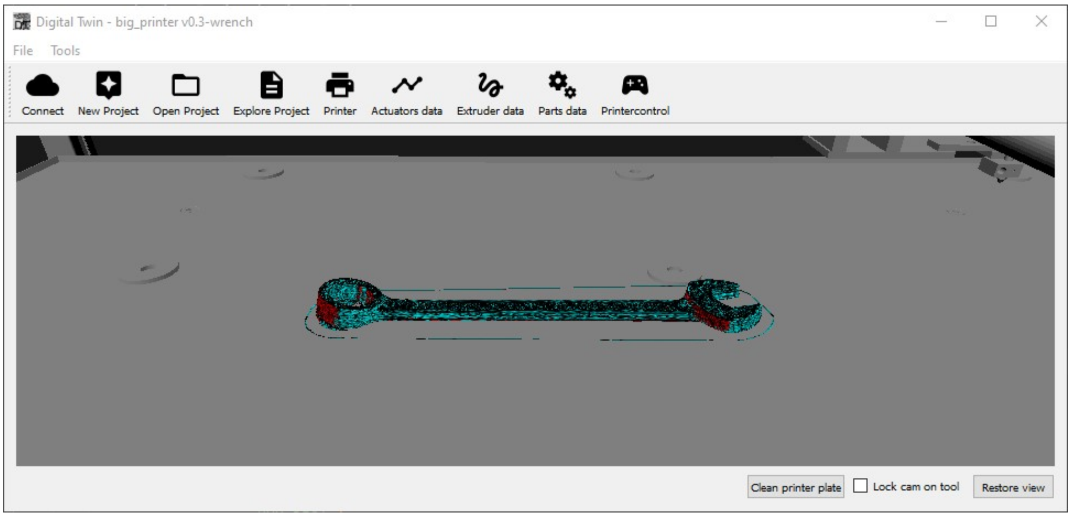


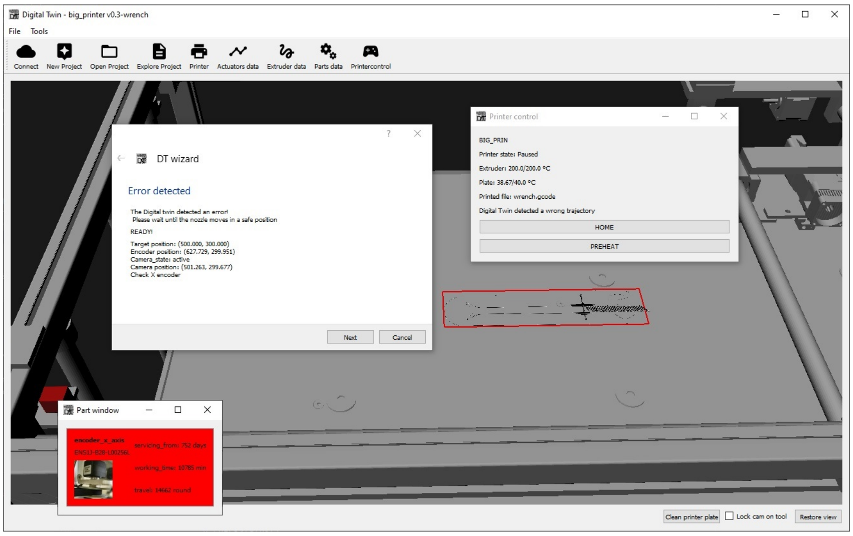
Figure 2.33: In the image is visible the 3D representation of a printed part after post-processing, the parts that deviate most from the ideal model are marked in red.

2.6.5 Error Recovery Wizard

During the printing process, if an error is detected, the GUI automatically starts the Recovery Wizard.

The first window is the same for all error types, an alert warns the user that a printing error has been detected and invites them to wait for the printer to move to a safe position. Next, the DTC diagnosis module is called up and a list of parts that may be involved in the problem is shown. The status of the involved parts changes to NOT OPERATING and can be seen flashing red in the GUI (Figure 2.34).

The next step asks the user whether they wants to continue with the recovery procedure, ignore the error and continue printing or cancel it. If the user decides to continue, the print recovery wizard starts and changes depending on the error detected.



DT wizard

Error detected

The Digital twin detected an error!
 Please wait until the nozzle moves in a safe position

READY!

Target position: (500.000, 300.000)
Encoder position: (627.729, 299.951)
Camera_state: active
Camera position: (501.263, 299.677)
Check X encoder

Next Cancel

Printer control

BIG_PRIN

Printer state: Paused

Extruder: 200.0/200.0 °C

Plate: 38.67/40.0 °C

Printed file: wrench.gcode

Digital Twin detected a wrong trajectory

HOME

PREHEAT

encoder_x_axis
ENS1J-B28-L00256L
servicing_from: 752 days
working_time: 10785 min
travel: 14662 round

Figure 2.34: The image shows the first page of the recovery wizard in case of encoder X malfunction. Note that the encoder model is highlighted in red..

The wizard presents a tree structure (visible in Figure 2.35) that guides the user through the recovery process and identification of the fault.

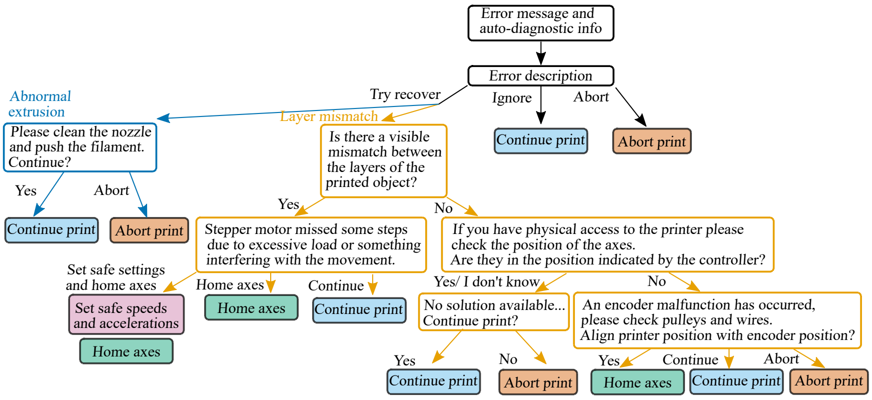

Error message and auto-diagnostic info
Error description
Try recover
Ignore
Abort
Continue print
Abort print
Abnormal extrusion
Please clean the nozzle and push the filament. Continue?
Yes
Abort
Continue print
Abort print
Layer mismatch
Is there a visible mismatch between the layers of the printed object?
Yes
No
Stepper motor missed some steps due to excessive load or something interfering with the movement.
Set safe settings and home axes
Home axes
Continue
Set safe speeds and accelerations
Home axes
Home axes
Continue print
If you have physical access to the printer please check the position of the axes. Are they in the position indicated by the controller?
Yes/ I don't know
No
No solution available... Continue print?
Yes
No
Continue print
Abort print
An encoder malfunction has occurred, please check pulleys and wires. Align printer position with encoder position?
Yes
Continue
Abort
Home axes
Continue print
Abort print

Figure 2.35: Tree structure of the Recovery Wizard

Chapter 3

Result and Discussion

3.1 Test setup

In order to test the functionality of the digital twin, it was decided to print a 19 mm combined wrench whose dimensions are visible in Figure 3.1.

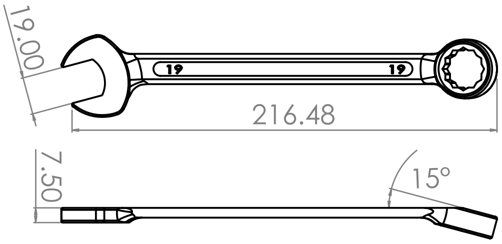


Figure 3.1: Main dimensions [mm] of the object printed in test phase.

First of all, the 3D model was sliced and the G-code file was created. Then, using the wizard, a new project has been created starting from the file just made. In this way it has been generated the ideal model of the print visible in Figure 3.2.

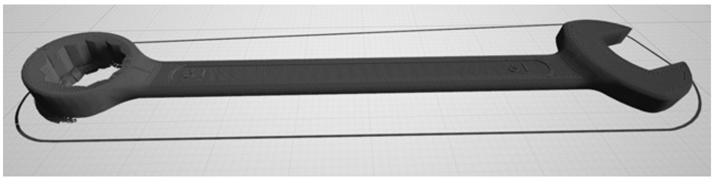


Figure 3.2: 3D rendering of the ideal model of printed part.

Two different profiles of print settings were then implemented: a slower one, called *Wrench_ 100*, from which a better quality and a lower risk of errors is expected and a faster one, called *Wrench_ 200*, that is characterized by a higher risk of failure (Table 3.1).

Table 3.1: Description of the used settings

Print name	Wrench_100	Wrench_200
X-axis max acc. [mm/s^2]	2048	2048
Y-axis max acc. [mm/s^2]	8192	8192
Z-axis max acc. [mm/s^2]	5	5
E-axis max acc. [mm/s^2]	5000	5000
X-axis max f.r. [mm/s]	220	220
Y-axis max f.r. [mm/s]	200	200
Z-axis max f.r. [mm/s]	5	5
E-axis max f.r. [mm/s]	5	20
Print acc. [mm/s^2]	1000	1000
Retract acc. [mm/s^2]	800	800
Travel acc. [mm/s^2]	1000	1000
Feedrate factor [%]	100	200

3.2 Simulation stage

As a first step, it was decided to carry out a printing simulation to verify the characteristics of the G-code, the settings used, and their impact on the machine components.

The simulations performed produced the data shown in Table 3.2.

Table 3.2: Data collected from printing simulation

Print name	Wrench_100	Wrench_200
Extruded volume [mm^3]	14235	14237
Maximum load on X-axis [%]	32	39
Maximum load on Y-axis [%]	6	6
Maximum load on Z-axis [%]	32	32
Maximum load on E-axis [%]	97	128
Printing time	1:47:06	1:21:55
Average extruder temp. [$°C$]	200	200

From the data provided, it can be seen that the use of a profile characterized by higher speeds causes a reduction in printing time of about 25 minutes at the cost of an increase in the load acting on the machine axes.

The increase in load that is visible for the X-axis is not caused by the greater forces involved (in fact the maximum acceleration has not changed) but by the decrease in the maximum force that can be delivered by the motor due to high speeds. It can also be seen that the increase in extrusion speed brings a load on the extruder motor that exceeds its maximum sustainable torque. In order to investigate this phenomenon, the graph visible in Figure 3.3 has been realized which, starting from the data collected during the simulation, shows the trend of the load acting on the E-axis in the layers where it is more critical. From this graph, it can be seen that the straight lines corresponding to the perimeters and the surfaces are the most critical parts, in the real print it is expected that these parts have a lower thickness than normal because of the E-axis motor missing steps.

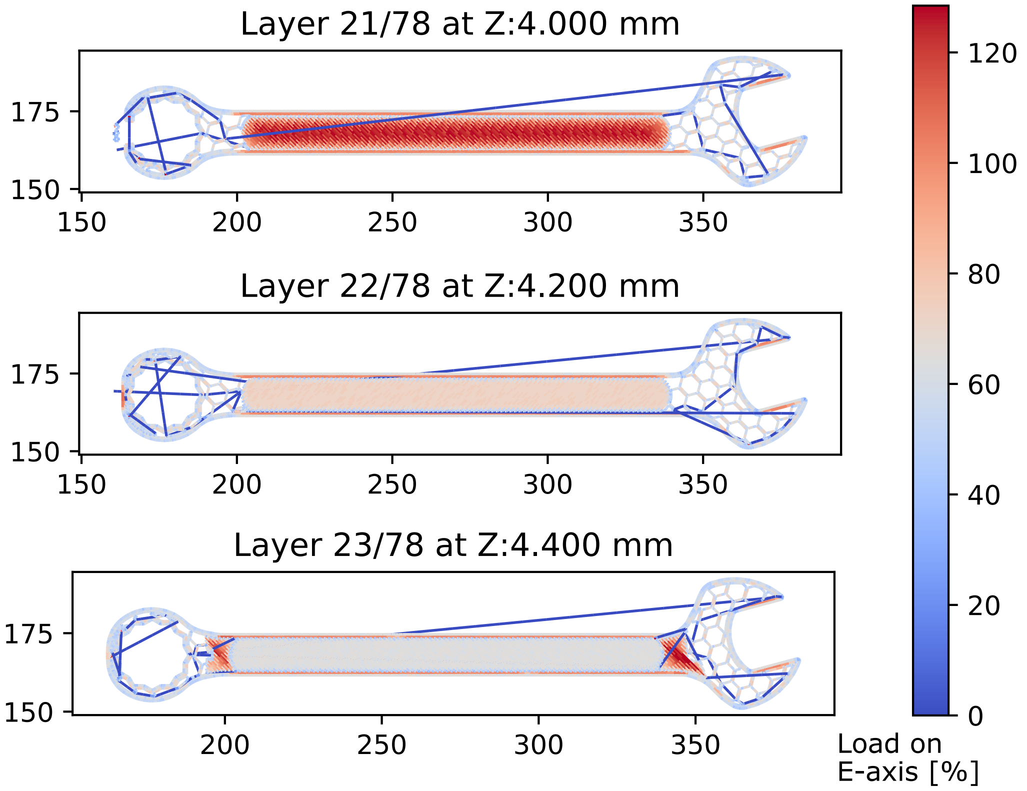


Figure 3.3: The figure shows the representation of layers 21, 22 and 23 in the printout *wrench_ 200*, the lines are colored as a function of the load on the E-axis.

The simulation also predicts an increase in part wear parameters equal to the data shown in Table 3.2. From these parameters it is possible to evaluate the impact that the printing and the settings used have on the life of the machine parts, information that constitutes a valid support to the preventive cost calculation.

Table 3.3: Data obtained from the simulation about the wear of the single parts

Print name	Wrench_100	Wrench_200
X motors travel [$rotation$]	2961	2954
X motors energy [kWh]	0.0586	0.0820
Y motor travel [$rotation$]	1857	1853
Y motor energy [kWh]	0.0186	0.0253
Z motors travel [$rotation$]	13	13
Z motors energy [kWh]	0.0002	0.0002
E motor travel [$rotation$]	173	173
E motor energy [kWh]	0.0400	0.0455
X driver energy [kWh]	0.1172	0.1639
Y driver energy [kWh]	0.0186	0.0253
Z drivers energy [kWh]	0.0002	0.0002
E driver energy [kWh]	0.0400	0.0455
X belts travel [mm]	118435	118175
X belts energy [kWh]	0.0586	0.0820
Y belt travel [mm]	74289	74135
Y belt energy [kWh]	0.0186	0.0253
X bearings travel [mm]	118435	118175
Y bearings travel [mm]	74289	74135
Z bearings travel [mm]	25	25
Extruder pipe filament [mm]	6917	6911
Nozzle filament [mm]	6917	6911
X encoder travel [$rotation$]	2369	2363
Y encoder travel [$rotation$]	1486	1482
Z encoder travel [$rotation$]	54	54
E encoder travel [$rotation$]	173	173

It can be seen that using higher speeds increases the energy required for machining. The total displacement is slightly lower in the second simulation due to the effect of the frame rate at which the wear values are calculated (10 Hz).

Finally, the 3D model of the part, generated during simulation, can be used to determine the actual final dimensions of critical parts of the piece. In this way, if the desired dimensions are not verified within an expected tolerance, the CAD model can be modified before the actual printing.

For example, as seen in Figure 3.4, it is possible to estimate that the opening between the forks of the wrench will be 18.9 mm instead of the nominal 19 mm. This prediction takes into account the approximations of the slicing algorithm, the path planner, and the discretization due to stepper motors and is a minimum limit of error. In real printing,

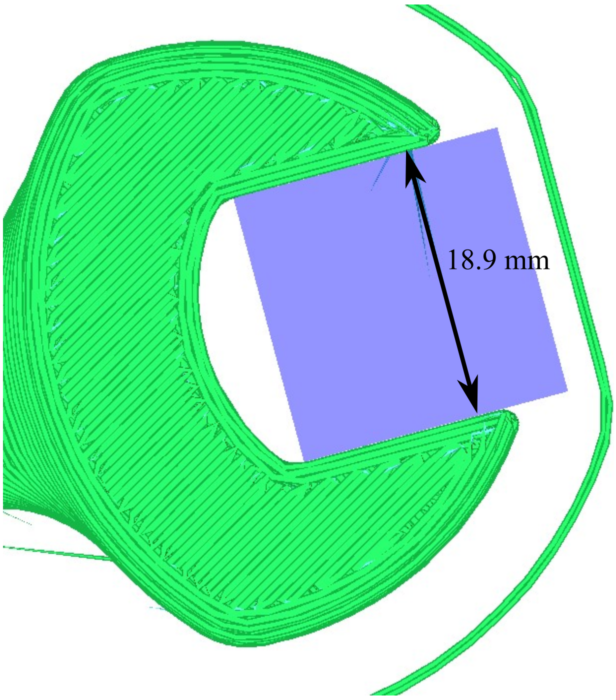
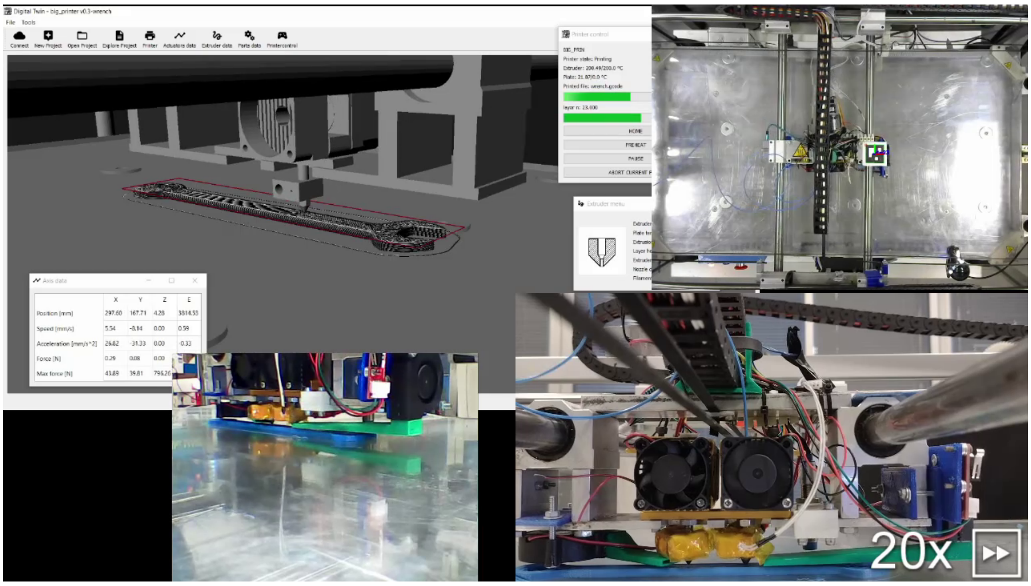


Figure 3.4: Main dimensions [mm] of the object printed in test phase.

other factors such as elasticity and backlash of the transmission system and thermal deformation can increase this value.

3.3 Printing stage

Once the simulation phase was complete, the actual prints were made (Figure 3.5).



BIG_PRIN
Printer state: Printing
Extruder: 200.49/200.0 °C
Plate: 21.87/0.0 °C
Printed file: wrench.gcode
layer n: 23.000

	X	Y	Z	E
Position [mm]	297.60	167.71	4.28	3814.53
Speed [mm/s]	5.54	-8.14	0.00	0.59
Acceleration [mm/s^2]	26.82	-31.33	0.00	-0.33
Force [N]	0.29	0.08	0.00	
Max force [N]	43.89	39.81	796.26	

Figure 3.5: A frame taken from the recording of the printing phase. In the upper left corner the GUI can be seen, in the other panels there are images taken from various angles.

Table 3.4 shows the data obtained from the digital twin at the end of the print jobs, while Table 3.5 shows the data relating to the wear of the parts in the real prints.

It can be seen that the load acting on the X and Y axes is slightly higher than what was estimated in the simulation phase, this is due to the vibrations to which the structure of the prints is subjected during the printing process that cause irregularities in the movement of the extruder block. The load acting on the E-axis instead is lower than expected, in this case, the reason is the increase of the actual temperature of the extruder compared to the one set by the G-code file. The average temperature measured by the calibrated smart sensors is in fact 213°C instead of the expected 200°C, this causes the decrease of the material viscosity and therefore of the force required for the extrusion. The expected printing time is respected, considering that the real process includes the time needed to heat the system, not modeled in the simulation phase.

Regarding the parameters of wear of the single parts, it is noticed that they are, in general, respecting the values previewed in the simulation phase with exception of the displacement of the axes that results greater. This is because the real phase involves also the homing procedure that instead it is not foreseen in the simulation.

Table 3.4: Data collected from real print jobs

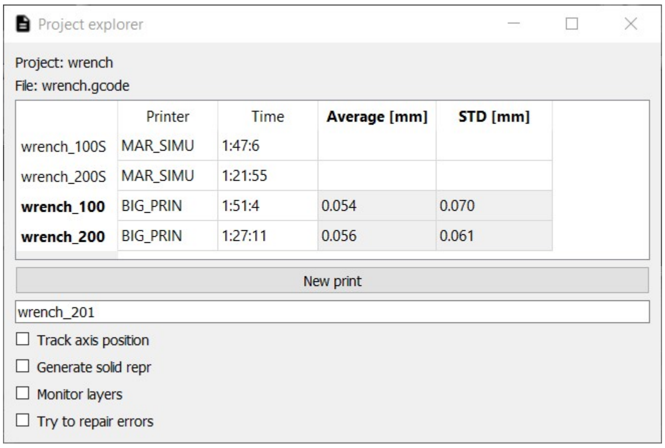
Print name	Wrench_100	Wrench_200
Extruded volume [mm^3]	14235	14237
Maximum load on X-axis [%]	46	50
Maximum load on Y-axis [%]	6	7
Maximum load on Z-axis [%]	32	32
Maximum load on E-axis [%]	69	80
Printing time	1:51:04	1:27:11
Average extruder temp. [$°C$]	213	212

Table 3.5: Data obtained from the real print jobs about the wear of the single parts

Print name	Wrench_100	Wrench_200
X motors travel [$rotation$]	2987	2985
X motors energy [kWh]	0.0541	0.0729
Y motor travel [$rotation$]	1864	1862
Y motor energy [kWh]	0.0121	0.0155
Z motors travel [$rotation$]	27	27
Z motors energy [kWh]	0.0007	0.0007
E motor travel [$rotation$]	171	170
E motor energy [kWh]	0.0253	0.0289
X driver energy [kWh]	0.1081	0.1459
Y driver energy [kWh]	0.0121	0.0155
Z drivers energy [kWh]	0.0007	0.0007
E driver energy [kWh]	0.0253	0.0289
X belts travel [mm]	119473	119406
X belts energy [kWh]	0.0541	0.0729
Y belt travel [mm]	74558	74461
Y belt energy [kWh]	0.0121	0.0155
X bearings travel [mm]	119473	119406
Y bearings travel [mm]	74558	74461
Z bearings travel [mm]	54	54
Extruder pipe filament [mm]	6831	6814
Nozzle filament [mm]	6831	6814
X encoder travel [$rotation$]	2389	2388
Y encoder travel [$rotation$]	1491	1489
Z encoder travel [$rotation$]	116	116
E encoder travel [$rotation$]	171	170

3.4 Post-processing stage

After printing, the post-processing module of the GUI is used to obtain a numerical evaluation of the geometric conformity of the finished part (see section 2.6.4) [87]. Then, it is possible to visualize the printed part with the areas characterized by a greater divergence from the G-code marked in red.



Project: wrench
File: wrench.gcode

	Printer	Time	**Average [mm]**	**STD [mm]**
wrench_100S	MAR_SIMU	1:47:6		
wrench_200S	MAR_SIMU	1:21:55		
wrench_100	BIG_PRIN	1:51:4	0.054	0.070
wrench_200	BIG_PRIN	1:27:11	0.056	0.061

wrench_201

☐ Track axis position
☐ Generate solid repr
☐ Monitor layers
☐ Try to repair errors

Figure 3.6: In the project explorer window the printouts made in the current project and the post-processing results are visible.

From the values obtained, visible in Figure 3.6, and from the 3D reconstructions in Figure 3.7 and 3.8, it can be seen that an increase in printing speed corresponds to a general decrease in the geometric characteristics of the part, especially with regard to curved sections.

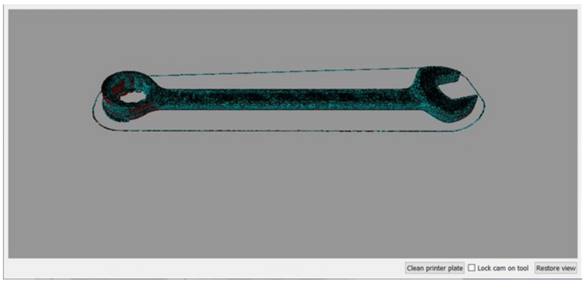


Figure 3.7: Virtual model of the piece resulting from the *wrench_ 100* print with the areas characterized by a high geometric dissimilarity marked in red.

In order to verify the results obtained, it was decided to use the PLY files from the post-processing procedure and analyze them manually with the CloudCompare application. This allows us to verify the results and investigate the causes of the defects found in the geometry of the parts.

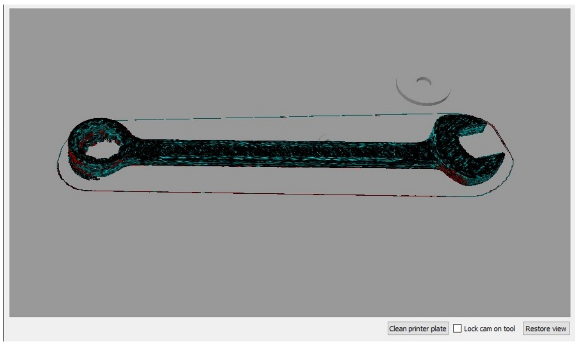


Figure 3.8: Virtual model of the piece resulting from the *wrench_200* print with the areas characterized by a high geometric dissimilarity marked in red..

Figure 3.9 shows the frequency graphs of the dimensional error associated with each print job. Underneath each graph, there is a representation of the virtual part, both whole and in section, in which the areas characterized by different error magnitudes are marked, and finally there is a picture of the real part. CloudCompare allows the user to visualize a scalar field associated with the mesh (in this case the difference compared to the ideal model) as a variation of colors defined by an algorithm, the same algorithm has been used for the realization of the bar graph so that it acts also as a reference. It can be seen that in both graphs there are peaks at the most frequent error values. It is possible to analyze the causes of these peaks by narrowing range and saturation of the values shown in the histograms and observing which areas remain colored in the figure. In any case, the analysis is limited to values greater than 0.05 mm, which is the resolution of the encoders used for the measurement.

By narrowing the range of values between 0.12 and 0.2 (Figure 3.10), it is possible to investigate the causes for the presence of a peak at this interval in both prints. From the screenshots taken from the application, it can be seen that the points associated with this error value are concentrated in the first layer in both cases. For this reason, it is believed that this error in geometric conformity is determined by the limited accuracy of the Z-axis homing system that causes a variation in the actual height of the printed part compared to the ideal model. This type of error similarly affects both print jobs as it is not related to the printing speed.

Next, a peak is analyzed that, in the first print, ranges from 0.05 to 0.10 mm (Figure 3.11). The areas most affected by this type of error are those related to the perimeters from the tenth layer upwards. This is due to the superimposition of two factors:

- Around the tenth layer there was a small Y-axis slippage, which is also clearly visible in the real object. This may have been caused by the backlash of the transmission system or by a motor that has missed a step due to an obstruction of the nozzle movement (it may be that the edge of the printed part rises slightly in the presence of an undercut due to thermal deformation).

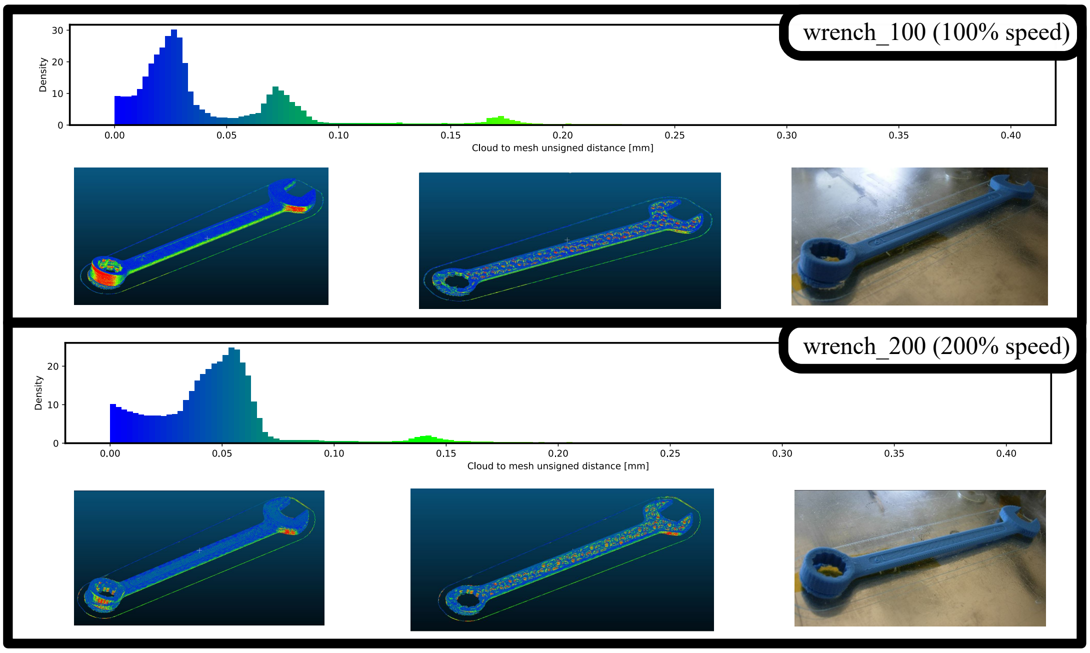


Figure 3.9: For each print is represented the density graph of the dimensional error, the corresponding colored rendering for both whole and section, and a photograph of the real object.

- In the curved sections of the print, irregularities can be observed, presumably due to the elasticity of the belt system which, if stressed with high acceleration values, can cause momentary positioning errors.

Regarding the *wrench_ 200* print, it is assumed that the rather high average error is due precisely to the phenomenon of the elasticity of the drive system. As can be seen in Figure 3.12, these errors are rather high (from 0.2 to 1.0 mm) and present above all in curved sections, i.e. those characterized by higher accelerations. This kind of error is much more evident in the faster printing because the accelerations are maintained longer to reach the correct printing speed, the phenomenon is also evident to the naked eye in the real printed piece.

Comparing the density graphs on the same scale it is possible to see the influence of the detected defects on the final index calculated by the post-processing procedure (Figure 3.13).

From the analysis, it emerges that the predominant problem is the inaccurate homing of the Z-axis, which can be solved only by replacing the installed sensors. Another problem concerns the Y-axis offset in the *wrench_ 100* print probably caused by the extruder being caught in one of the undercuts. The possibility of this event recurring can be avoided by modifying the G-code and setting the nozzle lift in the movement phases without extrusion. Or as an alternative solution, it is possible to reduce the printing temperature

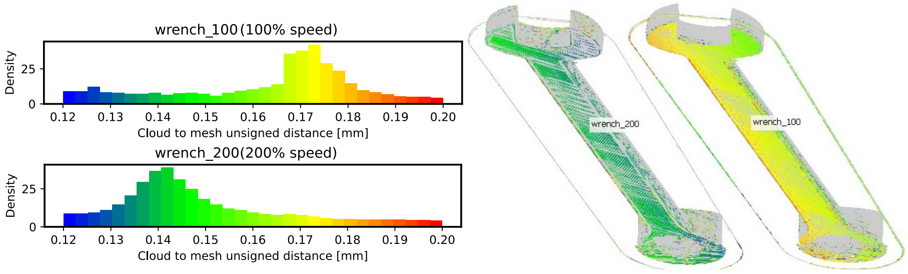


Figure 3.10: In the image the density graph has been limited to the range between 0.12 and 0.2 mm in order to highlight the areas affected by this type of error.

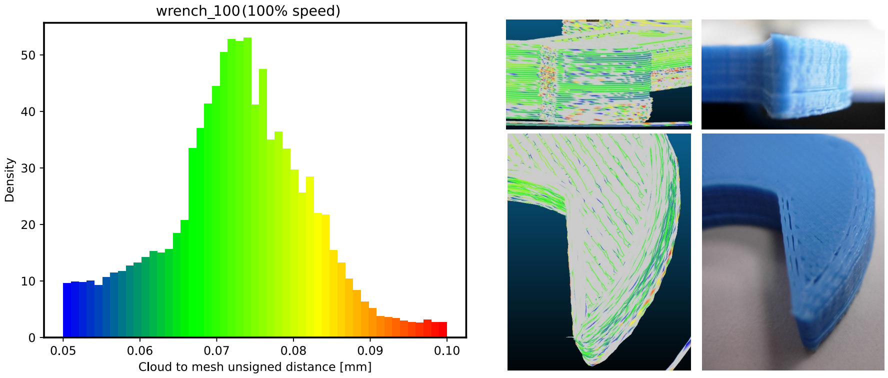


Figure 3.11: In the image the density graph has been limited to the range between 0.02 and 0.1 mm in order to highlight the area responsible for the corresponding peak in print *wrench_ 100*.

and in this way lower the probability that the edge of the print lifts due to thermal deformation.

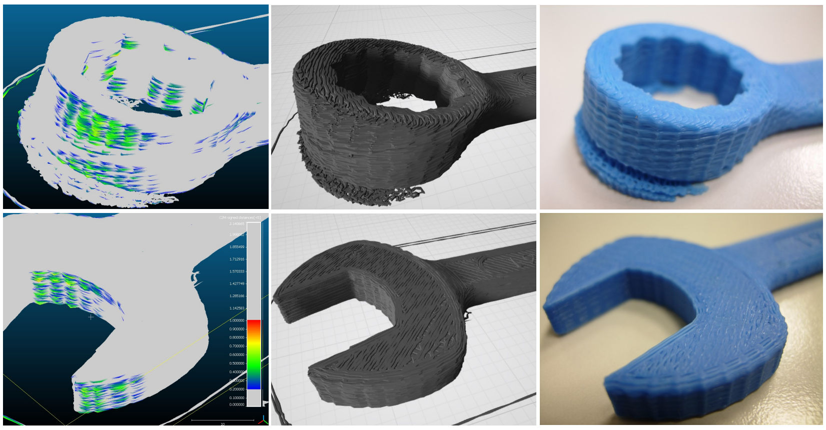


Figure 3.12: The image shows printing defects in the curvilinear segments of the *wrench_200* part. Of the represented parts are shown the graph displaying the value of the errors (from 0.2 to 1.0 mm), the 3D representation with the rendering of the shadows and the photograph of the real part.

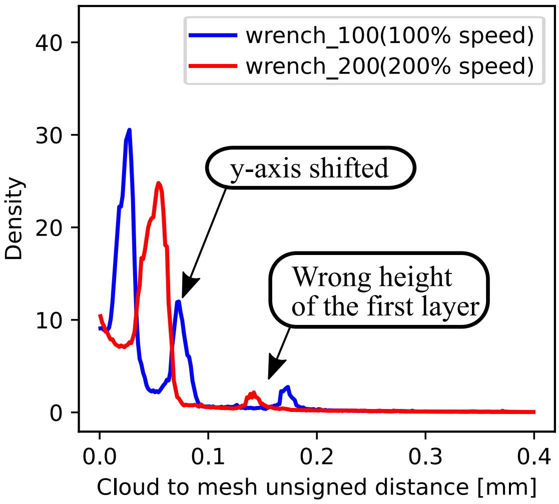


Figure 3.13: Density graph of the error with the probable causes of the peaks indicated.

Conclusions

In this thesis, the structure of a multifunctional digital twin for a material extrusion 3D printer capable of performing simulation, monitoring, and diagnostic functions is illustrated and its capabilities are demonstrated with specific tests.

Before printing, the system can predict the impact of the settings used on actuator loads and part wear, assess the final part size, and perform an auto-calibration to check system limits. During printing, it enables remote monitoring of process parameters, automatically intervenes in case of errors, and guides the user through the recovery process. After printing, it is possible to analyze the collected data obtaining an evaluation of the geometric conformity of the piece, moreover, a digital model of the printed piece is generated, useful for further analysis and for the evaluation of the characteristics of the printed part.

The solution is designed to be flexible and extendable to different systems and is a valuable aid for implementing the paradigms of digital transition and Industry 4.0.

In the future, it is intended to improve the physical model of the twin with a more complete mechanical simulation and the introduction of thermal phenomena. Currently, the proposed digital twin is implemented at the level of a single machine but the ultimate goal is to integrate it into the INCANTO IoT platform together with the other projects developed within the INCANTO PrINT initiative.

Bibliography

[1] H. Kagermann, W.-D. Lukas, and W. Wahlster, "Industrie 4.0: Mit dem Internet der Dinge auf dem Weg zur 4. industriellen Revolution," VDI Nachrichten, no. 13, pp. 3–4, 2011, [Online]. Available: `http://www.vdi-nachrichten.com/Technik-Gesellschaft/Industrie-40-Mit-Internet-Dinge-Weg-4-industriellen-Revolution`.

[2] H. Kagermann, W. Wahlster, and J. Helbig, "Securing the future of German manufacturing industry: Recommendations for implementing the strategic initiative INDUSTRIE 4.0," Final Report of the Industrie 4.0 Working Group, no. April, pp. 1–84, 2013.

[3] E. Oztemel and S. Gursev, "Literature review of Industry 4.0 and related technologies," J. Intell. Manuf., vol. 31, no. 1, pp. 127–182, Jan. 2020, doi: 10.1007/s10845-018-1433-8.

[4] U. A. Pozdnyakova, V. V. Golikov, I. A. Peters, and I. A. Morozova, "Genesis of the Revolutionary Transition to Industry 4.0 in the 21st Century and Overview of Previous Industrial Revolutions," in Studies in Systems, Decision and Control, vol. 169, Springer International Publishing, 2019, pp. 11–19, doi: 10.1007/978-3-319-94310-7_2.

[5] E. G. Popkova, Y. V. Ragulina, and A. V. Bogoviz, "Fundamental Differences of Transition to Industry 4.0 from Previous Industrial Revolutions," in Studies in Systems, Decision and Control, vol. 169, Springer, Cham, 2019, pp. 21–29, doi: 10.1007/978-3-319-94310-7_3.

[6] R. Drath and A. Horch, "Industrie 4.0: Hit or Hype? [Industry Forum]," IEEE Ind. Electron. Mag., vol. 8, no. 2, pp. 56–58, Jun. 2014, doi: 10.1109/MIE.2014.2312079.

[7] Y. Liao, F. Deschamps, E. de F. R. Loures, and L. F. P. Ramos, "Past, present and future of Industry 4.0 - a systematic literature review and research agenda proposal", vol. 55, no. 12, pp. 3609–3629, Jun. 2017, doi: 10.1080/00207543.2017.1308576.

[8] O. Bongomin, E. O. Nganyi, M. R. Abswaidi, E. Hitiyise, and G. Tumusiime, "Sustainable and Dynamic Competitiveness towards Technological Leadership of Industry

4.0: Implications for East African Community," J. Eng., vol. 2020, pp. 1–22, Jun. 2020, doi: 10.1155/2020/8545281.

[9] M. Rüßmann et al., "Industry 4.0: The Future of Productivity and Growth in Manufacturing Industries," Bost. Consult. Gr., no. April, pp. 1–14, 2015, [Online]. Available: https://inovasyon.org/images/Haberler/bcgperspectives_Industry40_2015.pdf.

[10] S. Melnik, M. Magnotti, C. Butts, C. Putman, and F. Aqlan, "A concept relationship map for industry 4.0," in Proceedings of the International Conference on Industrial Engineering and Operations Management, 2020, no. August, [Online]. Available: http://www.ieomsociety.org/detroit2020/papers/684.pdf.

[11] M. Glatt, C. Sinnwell, L. Yi, S. Donohoe, B. Ravani, and J. C. Aurich, "Modeling and implementation of a digital twin of material flows based on physics simulation," J. Manuf. Syst., vol. 58, pp. 231–245, doi: 10.1016/j.jmsy.2020.04.015.

[12] T. Stock and G. Seliger, "Opportunities of Sustainable Manufacturing in Industry 4.0," Procedia CIRP, vol. 40, pp. 536–541, Jan. 2016,doi: 10.1016/J.PROCIR.2016.01.129.

[13] M. Pérez-Lara, J. A. Saucedo-Martínez, J. A. Marmolejo-Saucedo, T. E. Salais-Fierro, and P. Vasant, "Vertical and horizontal integration systems in Industry 4.0," Wirel. Networks, vol. 26, no. 7, pp. 4767–4775, Oct. 2020, doi: 10.1007/s11276-018-1873-2.

[14] M. K. Thompson et al., "Design for Additive Manufacturing: Trends, opportunities, considerations, and constraints," CIRP Ann., vol. 65, no. 2, pp. 737–760, Jan. 2016, doi: 10.1016/j.cirp.2016.05.004.

[15] X. Xu, "From cloud computing to cloud manufacturing," Robot. Comput. Integr. Manuf., vol. 28, no. 1, pp. 75–86, Feb. 2012, doi: 10.1016/j.rcim.2011.07.002.

[16] D. J. Horst, C. Duvoisin, and R. D. A. Vieira, "Additive Manufacturing at Industry 4.0: a Review," Artic. Int. J. Eng. Tech. Res., vol. 8, no. 8, p. 3, 2018, [Online]. Available: https://www.academia.edu/download/57396694/IJETR2643.pdf.

[17] P. A. L. Ciraud, "Method and Device for Manufacturing any Articles from any Meltable Material," DE2263777A1, 1972.

[18] T. Wohlers and T. Gornet, "History of additive manufacturing," wohlersassociates.com, 2016, Accessed: Dec. 02, 2021. [Online]. Available: http://www.wohlersassociates.com/history2016.pdf.

[19] M. Shellabear and O. Nyrhilä, "DMLS – Development History and State of the Art," in LANE 2004, 2004, pp. 1–12.

[20] I. H. Lee and Y. G. Kim, "The Recent Patent Analysis and Industrial Trend of 3D Printing," Indian J. Sci. Technol., vol. 8, no. S8, p. 70, Apr. 2015, doi: 10.17485/ijst/2015/v8iS8/64717.

[21] E. Sachs, M. Cima, P. Williams, D. Brancazio, and J. Cornie, "Three Dimensional Printing: Rapid Tooling and Prototypes Directly from a CAD Model," J. Eng. Ind., vol. 114, no. 4, pp. 481–488, Nov. 1992, doi:10.1115/1.2900701.

[22] R. F. Housholder, "Molding process," US4247508A, 1979.

[23] C. W. Hull, "Apparatus for production of three dimensional objects by stereolithography," US6027324A, 1984.

[24] S. Scott Crump, "Apparatus and method for creating three-dimensional objects," US5121329A, 1989.

[25] CECIMO, "Additive - CECIMO." `https://www.cecimo.eu/machine-tools/additive/` (accessed Nov. 24, 2021).

[26] N. Araújo, V. Pacheco, and L. Costa, "Smart Additive Manufacturing: The Path to the Digital Value Chain," Technologies, vol. 9, no. 4, p. 88, Nov. 2021,doi:10.3390/technologies9040088.

[27] A. Nazir et al., "The rise of 3D Printing entangled with smart computer aided design during COVID-19 era," J. Manuf. Syst., vol. 60, pp. 774–786, Jul. 2021, doi:10.1016/j.jmsy.2020.10.009.

[28] R. Tino et al., "COVID-19 and the role of 3D printing in medicine," 3D Print. Med., vol. 6, no. 1, p. 11, Dec. 2020, doi: 10.1186/s41205-020-00064-7.

[29] I. Gibson, D. Rosen, B. Stucker, and M. Khorasani, "Vat Photopolymerization," in Additive Manufacturing Technologies, Cham: Springer International Publishing, 2021, pp. 77–124.

[30] I. Gibson, D. Rosen, B. Stucker, and M. Khorasani, "Powder Bed Fusion," in Additive Manufacturing Technologies, Cham: Springer International Publishing, 2021, pp. 125–170.

[31] I. Gibson, D. Rosen, B. Stucker, and M. Khorasani, "Material Extrusion," in Additive Manufacturing Technologies, Cham: Springer International Publishing, 2021, pp. 171–201.

[32] G. Avventuroso, R. Foresti, M. Silvestri, and E. M. Frazzon, "Production paradigms for additive manufacturing systems: A simulation-based analysis," in 2017 International Conference on Engineering, Technology and Innovation (ICE/ITMC), Jun. 2017, vol. 2018-Janua, pp. 973–981, doi: 10.1109/ICE.2017.8279987.

[33] G. Avventuroso, M. Silvestri, and E. M. Frazzon, “Additive Manufacturing Plant for Large Scale Production of Medical Devices: A Simulation Study,” IFAC-PapersOnLine, vol. 51, no. 11, pp. 1442–1447, Jan. 2018, doi: 10.1016/J.IFACOL.2018.08.312.

[34] R. Duballet, O. Baverel, and J. Dirrenberger, “Classification of building systems for concrete 3D printing,” Autom. Constr., vol. 83, pp. 247–258, Nov. 2017, doi: 10.1016/j.autcon.2017.08.018.

[35] I. Gibson, D. Rosen, B. Stucker, and M. Khorasani, “Material Jetting,” in Additive Manufacturing Technologies, Cham: Springer International Publishing, 2021, pp. 203–235.

[36] I. Gibson, D. Rosen, B. Stucker, and M. Khorasani, “Binder Jetting,” in Additive Manufacturing Technologies, Cham: Springer International Publishing, 2021, pp. 237–252.

[37] I. Gibson, D. Rosen, B. Stucker, and M. Khorasani, “Sheet Lamination,” in Additive Manufacturing Technologies, Cham: Springer International Publishing, 2021, pp. 253–283.

[38] I. Gibson, D. Rosen, B. Stucker, and M. Khorasani, “Directed Energy Deposition,” in Additive Manufacturing Technologies, Cham: Springer International Publishing, 2021, pp. 285–318.

[39] L. Da Xu, W. He, and S. Li, “Internet of things in industries: A survey,” IEEE Trans. Ind. Informatics, vol. 10, no. 4, pp. 2233–2243, doi: 10.1109/TII.2014.2300753.

[40] M. Tu, M. K. Lim, and M.-F. Yang, “IoT-based production logistics and supply chain system – Part 1,” Ind. Manag. Data Syst., vol. 118, no. 1, pp. 65–95, Feb. 2018, doi: 10.1108/IMDS-11-2016-0503.

[41] X. Liu and Y. Sun, “Information Flow Management of Vendor-Managed Inventory System in Automobile Parts Inbound Logistics Based on Internet of Things.,” J. Softw., vol. 6, no. 7, pp. 1374–1380, 2011, [Online]. Available: `http://www.jsoftware.us/vol6/jsw0607-29.pdf`.

[42] F. Shrouf and G. Miragliotta, “Energy management based on Internet of Things: practices and framework for adoption in production management,” J. Clean. Prod., vol. 100, pp. 235–246, Aug. 2015, doi: 10.1016/j.jclepro.2015.03.055.

[43] R. Ashima, A. Haleem, S. Bahl, M. Javaid, S. Kumar Mahla, and S. Singh, “Automation and manufacturing of smart materials in additive manufacturing technologies using Internet of Things towards the adoption of industry 4.0,” Mater. Today Proc., vol. 45, pp. 5081–5088, Jan. 2021, doi: 10.1016/j.matpr.2021.01.583.

[44] T. Chen and Y.-C. Wang, “An advanced IoT system for assisting ubiquitous manufacturing with 3D printing,” Int. J. Adv. Manuf. Technol., vol. 103, no. 5–8, pp. 1721–1733, 2019, doi: 10.1007/s00170-019-03691-5.

[45] R. Sivabalakrishnan, A. Kalaiarasan, M. S. Ajithvishva, M. Hemsri, G. M. Oorappan, and R. Yasodharan, “IoT visualization of Smart Factory for Additive Manufacturing System (ISFAMS) with visual inspection and material handling processes,” IOP Conf. Ser. Mater. Sci. Eng., vol. 995, no. 1, p. 012027, Dec. 2020, doi: 10.1088/1757-899X/995/1/012027.

[46] J.-S. J. Song and Y. Zhang, “Predictive 3D Printing with IoT,” SSRN Electron. J., pp. 1–65, 2021, doi: 10.2139/ssrn.3895854.

[47] M. Salama, A. Elkaseer, M. Saied, H. Ali, and S. Scholz, “Industrial Internet of Things Solution for Real-Time Monitoring of the Additive Manufacturing Process,” in Advances in Intelligent Systems and Computing, vol. 852, 2019, pp. 355–365, doi: 10.1007/978-3-319-99981-4_33.

[48] G. Barbosa and R. Aroca, “An IoT-Based Solution for Control and Monitoring of Additive Manufacturing Processes,” J. Powder Metall. Min., vol. 06, no. 01, pp. 1–7, Mar. 2017, doi: 10.4172/2168-9806.1000158.

[49] H. Gill, “Systems, NSF perspective and status on cyber-physical systems. In National Workshop on Cyber-physical,” Austin, TX, USA. Alexandria Natl. Sci. Found., 2006.

[50] F. Tao, Q. Qi, L. Wang, and A. Y. C. Nee, “Digital Twins and Cyber–Physical Systems toward Smart Manufacturing and Industry 4.0: Correlation and Comparison,” Engineering, vol. 5, no. 4, pp. 653–661, Aug. 2019, doi: 10.1016/j.eng.2019.01.014.

[51] R. Drath and A. Horch, “Industrie 4.0: Hit or Hype? [Industry Forum],” IEEE Ind. Electron. Mag., vol. 8, no. 2, pp. 56–58, Jun. 2014, doi: 10.1109/MIE.2014.2312079.

[52] M. Grieves, “Digital Twin : Manufacturing Excellence through Virtual Factory Replication,” White Pap., no. March, pp. 1–7, 2014, [Online]. Available: link.

[53] M. Shafto et al., “DRAFT Modelling, simulation, information technology & processing roadmap - technology area 11,” Natl. Aeronaut. Sp. Adm., vol. November, p. 27, 2010, [Online]. Available: `https://www.nasa.gov/pdf/501321main_TA11-MSITP-DRAFT-Nov2010-A1.pdf`.

[54] W. Kritzinger, M. Karner, G. Traar, J. Henjes, and W. Sihn, “Digital Twin in manufacturing: A categorical literature review and classification,” IFAC-PapersOnLine, vol. 51, no. 11, pp. 1016–1022, Jan. 2018, doi: 10.1016/j.ifacol.2018.08.474.

[55] B. Wang, F. Tao, X. Fang, C. Liu, Y. Liu, and T. Freiheit, “Smart Manufacturing and Intelligent Manufacturing: A Comparative Review,” Engineering, vol. 7, no. 6, pp. 738–757, Jun. 2021, doi: 10.1016/J.ENG.2020.07.017.

[56] G. N. Schroeder, C. Steinmetz, R. N. Rodrigues, R. V. B. Henriques, A. Rettberg, and C. E. Pereira, “A Methodology for Digital Twin Modeling and Deployment for Industry 4.0,” Proc. IEEE, vol. 109, no. 4, pp. 556–567, Apr. 2021,doi: 10.1109/JPROC.2020.3032444.

[57] Y. Lai, Y. Wang, R. Ireland, and A. Liu, “Digital twin driven virtual verification,” in Digital Twin Driven Smart Design, Elsevier, 2020, pp. 109–138, doi: 10.1016/B978-0-12-818918-4.00004-X.

[58] F. Tao, J. Cheng, Q. Qi, M. Zhang, H. Zhang, and F. Sui, “Digital twin-driven product design, manufacturing and service with big data,” Int. J. Adv. Manuf. Technol., vol. 94, no. 9–12, pp. 3563–3576, Feb. 2018, doi: 10.1007/s00170-017-0233-1 doi: 10.1007/s00170-017-0233-1.

[59] D. R. Gunasegaram, A. B. Murphy, M. J. Matthews, and T. DebRoy, “The case for digital twins in metal additive manufacturing,” JPhys Mater., vol. 4, no. 4, Oct. 2021, doi: 10.1088/2515-7639/AC09FB.

[60] E. Atzeni and A. Salmi, “Economics of additive manufacturing for end-usable metal parts,” Int. J. Adv. Manuf. Technol., vol. 62, no. 9–12, pp. 1147–1155, 2012, doi: 10.1007/s00170-011-3878-1.

[61] J. Lee, H.-A. Kao, and S. Yang, “Service Innovation and Smart Analytics for Industry 4.0 and Big Data Environment,” Procedia CIRP, vol. 16, pp. 3–8, Jan. 2014, doi: 10.1016/j.procir.2014.02.001.

[62] W. E. Frazier, “Metal Additive Manufacturing: A Review,” J. Mater. Eng. Perform., vol. 23, no. 6, pp. 1917–1928, Jun. 2014, doi: 10.1007/s11665-014-0958-z.

[63] T. DebRoy, W. Zhang, J. Turner, and S. S. Babu, “Building digital twins of 3D printing machines,” Scr. Mater., vol. 135, pp. 119–124, Jul. 2017, doi: 10.1016/j.scriptamat.2016.12.005.

[64] L. Yi, M. Glatt, S. Ehmsen, W. Duan, and J. C. Aurich, “Process monitoring of economic and environmental performance of a material extrusion printer using an augmented reality-based digital twin,” Addit. Manuf., vol. 48, p. 102388, Dec. 2021, doi: 10.1016/j.addma.2021.1023885.

[65] G. L. Knapp et al., “Building blocks for a digital twin of additive manufacturing,” Acta Mater., vol. 135, pp. 390–399, Aug. 2017, doi: 10.1016/j.actamat.2017.06.039.

[66] Z. Yang et al., “Investigating Grey-Box Modeling for Predictive Analytics in Smart Manufacturing,” Aug. 2017, doi: 10.1115/DETC2017-67794.

[67] T. Mukherjee and T. DebRoy, “A digital twin for rapid qualification of 3D printed metallic components,” Appl. Mater. Today, vol. 14, pp. 59–65, Mar. 2019, doi: 10.1016/j.apmt.2018.11.003.

[68] H. Ko, P. Witherell, N. Y. Ndiaye, and Y. Lu, “Machine Learning based Continuous Knowledge Engineering for Additive Manufacturing,” in 2019 IEEE 15th International Conference on Automation Science and Engineering (CASE), Aug. 2019, vol. 2019-Augus, pp. 648–654, doi: 10.1109/COASE.2019.8843316.

[69] 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,” IISE Trans., vol. 52, no. 11, pp. 1204–1217, Nov. 2020, doi: 10.1080/24725854.2019.1701753.

[70] D. R. Gunasegaram et al., “Towards developing multiscale-multiphysics models and their surrogates for digital twins of metal additive manufacturing,” Addit. Manuf., vol. 46, p. 102089, Oct. 2021, doi: 10.1016/J.ADDMA.2021.102089.

[71] T. W. Heo et al., “A mesoscopic digital twin that bridges length and time scales for control of additively manufactured metal microstructures,” J. Phys. Mater., vol. 4, no. 3, p. 034012, Jul. 2021, doi: 10.1088/2515-7639/abeef8.

[72] R. Yavari et al., “Digitally twinned additive manufacturing: Detecting flaws in laser powder bed fusion by combining thermal simulations with in-situ meltpool sensor data,” Mater. Des., vol. 211, p. 110167, Dec. 2021, doi: 10.1016/j.matdes.2021.110167.

[73] P. Stavropoulos, A. Papacharalampopoulos, C. K. Michail, and G. Chryssolouris, “Robust Additive Manufacturing Performance through a Control Oriented Digital Twin,” Metals (Basel)., vol. 11, no. 5, p. 708, Apr. 2021, doi:10.3390/met11050708.

[74] C. G. Klingaa, S. Mohanty, C. V. Funch, A. B. Hjermitslev, L. Haahr-Lillevang, and J. H. Hattel, “Towards a digital twin of laser powder bed fusion with a focus on gas flow variables,” J. Manuf. Process., vol. 65, pp. 312–327, May 2021, doi:10.1016/j.jmapro.2021.03.035.

[75] C. Mandolla, A. M. Petruzzelli, G. Percoco, and A. Urbinati, “Building a digital twin for additive manufacturing through the exploitation of blockchain: A case analysis of the aircraft industry,” Comput. Ind., vol. 109, pp. 134–152, Aug. 2019, doi:10.1016/j.compind.2019.04.011.

[76] C. Liu, L. Le Roux, C. Körner, O. Tabaste, F. Lacan, and S. Bigot, “Digital Twin-enabled Collaborative Data Management for Metal Additive Manufacturing Systems,” J. Manuf. Syst., May 2020, doi:10.1016/j.jmsy.2020.05.010.

[77] L. Scime, A. Singh, and V. Paquit, “A scalable digital platform for the use of digital twins in additive manufacturing,” Manuf. Lett., Jun. 2021, doi:10.1016/j.mfglet.2021.05.007.

[78] D. Guo et al., “A framework for personalized production based on digital twin, blockchain and additive manufacturing in the context of Industry 4.0,” in 2020 IEEE 16th International Conference on Automation Science and Engineering (CASE), Aug. 2020, vol. 2020-Augus, pp. 1181–1186, doi:10.1109/CASE48305.2020.9216732.

[79] L. Hu et al., “Modeling of Cloud-Based Digital Twins for Smart Manufacturing with MT Connect,” Procedia Manuf., vol. 26, pp. 1193–1203, 2018, doi:10.1016/j.promfg.2018.07.155.

[80] M. R. Shahriar, S. M. N. Al Sunny, X. Liu, M. C. Leu, L. Hu, and N.-T. Nguyen, “MTComm Based Virtualization and Integration of Physical Machine Operations with Digital-Twins in Cyber-Physical Manufacturing Cloud,” in 2018 5th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2018 4th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom), Jun. 2018, pp. 46–51, doi:10.1109/CSCloud/EdgeCom.2018.00018.

[81] E. C. Balta, D. M. Tilbury, and K. Barton, “A Digital Twin Framework for Performance Monitoring and Anomaly Detection in Fused Deposition Modeling,” in 2019 IEEE 15th International Conference on Automation Science and Engineering (CASE), Aug. 2019, vol. 2019-Augus, no. 70, pp. 823–829, doi:10.1109/COASE.2019.8843166.

[82] S. R. Chhetri, S. Faezi, A. Canedo, and M. A. Al Faruque, “QUILT,” in Proceedings of the International Conference on Internet of Things Design and Implementation, Apr. 2019, pp. 237–248, doi:10.1145/3302505.3310085.

[83] M. Moretti, A. Rossi, and N. Senin, “In-process monitoring of part geometry in fused filament fabrication using computer vision and digital twins,” Addit. Manuf., vol. 37, no. July 2020, p. 101609, 2021, doi:10.1016/j.addma.2020.101609.

[84] C. M. Henson, N. I. Decker, and Q. Huang, “A digital twin strategy for major failure detection in fused deposition modeling processes,” Procedia Manuf., vol. 53, pp. 359–367, Jan. 2021, doi:10.1016/J.PROMFG.2021.06.039.

[85] D. Mourtzis, T. Togias, J. Angelopoulos, and P. Stavropoulos, “A Digital Twin architecture for monitoring and optimization of Fused Deposition Modeling processes,” Procedia CIRP, vol. 103, pp. 97–102, Jan. 2021, doi:10.1016/j.procir.2021.10.015.

[86] L. Guo, Y. Cheng, Y. Zhang, Y. Liu, C. Wan, and J. Liang, “Development of Cloud-Edge Collaborative Digital Twin System for FDM Additive Manufacturing,” in 2021 IEEE 19th International Conference on Industrial Informatics (INDIN), Jul. 2021, pp. 1–6, doi:10.1109/INDIN45523.2021.9557492.

[87] F. Corradini and M. Silvestri, “Design and testing of a digital twin for monitoring and quality assessment of material extrusion process,” Addit. Manuf., 2022 - accepted for publication.

[88] F. Corradini and M. Silvestri, “A Digital Twin Based Self-Calibration Tool for Fault Prediction of FDM Additive Manufacturing Systems,” in Proceedings of the 32nd DAAAM International Symposium, B. Katalinic, Ed. Vienna, Austria: DAAAM International, 2021, pp. 0607–0616, doi: 10.2507/32nd.daaam.proceedings.086.

[89] Y. Cai, Y. Wang, and M. Burnett, "Using augmented reality to build digital twin for reconfigurable additive manufacturing system," J. Manuf. Syst., vol. 56, pp. 598–604, Jul. 2020, doi:10.1016/j.jmsy.2020.04.005.

[90] "Home | Marlin Firmware." `https://marlinfw.org/` (accessed Apr. 19, 2021).

[91] S. Stanciulescu, S. Schulze, and A. Wasowski, "Forked and integrated variants in an open-source firmware project," in 2015 IEEE International Conference on Software Maintenance and Evolution (ICSME), Sep. 2015, pp. 151–160, doi:10.1109/ICSM.2015.7332461.

[92] R. M. Scheffel, A. A. Fröhlich, and M. Silvestri, "Automated fault detection for additive manufacturing using vibration sensors," Int. J. Comput. Integr. Manuf., vol. 34, no. 5, pp. 500–514, Mar. 2021, doi: 10.1080/0951192X.2021.1901316.

[93] A. A. Frohlich, R. V. Steiner, and L. M. Rufino, "A Trustful Infrastructure for the Internet of Things Based on EPOSMote," in 2011 IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing, Dec. 2011, pp. 63–68, doi: 10.1109/DASC.2011.35.

[94] B. Gary, "The OpenCV Library," Dr. Dobb's J. Softw. Tools Prof. Program., vol. 25, no. 11, pp. 120–123, 2000.

[95] P. Oščádal et al., "Improved Pose Estimation of Aruco Tags Using a Novel 3D Placement Strategy," Sensors, vol. 20, no. 17, p. 4825, Aug. 2020, doi: 10.3390/s20174825.

[96] S. Baker, R. S. Bultje, and S. Kost, "GStreamer Application Development Related papers GStreamer Application Development," 2013, [Online]. Available: link.

[97] Häußge Gina, "OctoPrint.org." `https://octoprint.org/` (accessed Apr. 19, 2021).

[98] J. E. Bresenham, "Algorithm for computer control of a digital plotter," IBM Syst. J., vol. 4, no. 1, pp. 25–30, 1965, doi: 10.1147/sj.41.0025.

[99] K. Madlener, B. Frey, and H. K. Ciezki, "Generalized reynolds number for non-newtonian fluids," Prog. Propuls. Phys., vol. 1, pp. 237–250, 2009, doi: 10.1051/EUCASS/200901237.

[100] Q. Fang and M. A. Hanna, "Rheological properties of amorphous and semicrystalline polylactic acid polymers," Ind. Crops Prod., vol. 10, no. 1, pp. 47–53, Jun. 1999, doi: 10.1016/S0926-6690(99)00009-6.

[101] PyTables Developers Team, "PyTables: Hierarchical Datasets in Python." 2002, [Online]. Available: `http://www.pytables.org/`.

[102] G. Hodgson, A. Ranellucci, and J. Moe, "Slic3r Manual - Flow math", 2016, `https://manual.slic3r.org/advanced/flow-math`.

[103] Riverbank Computing, “PyQt5 · PyPI.” https://pypi.org/project/PyQt5/ (accessed Apr. 25, 2021).

[104] “Panda3D | Open Source Framework for 3D Rendering & Games.” https://www.panda3d.org/ (accessed Apr. 25, 2021).

[105] A. Saifeddine, “QPanda3D” https://pypi.org/project/QPanda3D/ (accessed Apr. 25, 2021).

[106] “OpenSCAD-The Programmers Solid 3D CAD Modeller.” http://openscad.org/ (accessed Apr. 25, 2021).

[107] “CloudCompare - 3D point cloud and mesh processing software.” https://www.cloudcompare.org/ (accessed Apr. 25, 2021).

Acknowledgements

I would like to thank all the collaborators of the *INCANTO PrINT* project with whom I worked for the definition of the objectives of this work and the realization of the measurement and data acquisition system, in particular Prof. Enzo Morosini Frazzon, Prof. Roberto Milton Scheffel, and Prof. Jean Everson Martina.

I thank my tutor, Prof. Marco Silvestri, for the constant support provided during the Ph.D. activity and my colleagues Marco Riboli and Simone Benatti for the continuous exchange of ideas that has greatly enriched this experience.

I thank my family for the support and trust they have always shown me during these years. Finally, I thank my wife Idina for her help, support and because she has always believed in me, even in difficult times. The results I have achieved I owe to her.