

POLITECNICO DI TORINO

Master's Degree Course in Automotive Engineering

**Product And Process Design  
For Additive Manufacturing**



**Politecnico  
di Torino**

**Advisor:**

Prof. Paolo Chiabert

**Candidate:**

Tianhao Liu

March 2023

# ***ABSTRACT***

The objective of this thesis is to propose a feasible solution for meeting the demand for mass customization and flexibility in production industries using additive manufacturing (AM). The thesis firstly introduces and compares several types of AM technology, then proposes a framework that employs a 6-axis cobot as the target positioning mechanism and a filament extrusion system utilizing fused deposition modelling (FDM) technique attached to the end effector. The proposed system uses computer-aided design (CAD) models imported to RoboDK, a robot simulation and programming software that replicates the real operational environment and converts the virtual simulation into reality. The system offers greater flexibility and can operate within complex and changeable environments compared to traditional 3D printers. It extends the working capabilities of collaborative robots and integrates the concept of man-machine collaboration into the development of additive manufacturing.

# ***CONTENTS***

ABSTRACT.....	2
CONTENTS.....	3
1 INTRODUCTION .....	5
2 ADDITIVE MANUFACTURING.....	6
2.1 Classification of Additive Manufacturing .....	7
2.1.1 Binder Jetting (BJT) .....	7
2.1.2 Directed Energy Deposition (DED).....	9
2.1.3 Material Extrusion (MEX).....	10
2.1.4 Material Jetting (MJT).....	11
2.1.5 Powder Bed Fusion (PBF).....	11
2.1.6 Sheet Lamination (SHL).....	13
2.1.7 VAT Photopolymerization (VPP) .....	14
2.2 Applications and Challenges of Additive Manufacturing .....	15
2.2.1 Advantages and disadvantages compared to conventional manufacturing	15
2.2.2 Applications of additive manufacturing .....	16
2.2.3 Challenges of additive manufacturing .....	17
2.3 Type of AM technique chosen for this project .....	18
3 HUMAN COBOT COOPERATION .....	19
3.1 Characteristics of cobots.....	19
3.1.1 Standards and test specifications .....	20
3.1.2 Difference between cobot and industrial robot.....	21
3.2 Universal-Robot UR10e and ON Robot end effectors .....	26
3.2.1 Features of UR10e .....	26
3.2.2 Reason to use a cobot in the AM system .....	27
4 COBOT ADDITIVE MANUFACTURING SYSTEM DESIGN .....	28
4.1 Hardware design .....	28
4.1.1 End effector design.....	28
4.1.2 Printing platform design .....	29
4.1.3 Other devices .....	29
4.2 Program design .....	30
4.2.1 Work cell design for RoboDK .....	30
4.2.2 Extrusion control .....	34
4.2.3 Temperature control.....	35
4.2.4 G-code conversion.....	35

5	TEST RUNNING.....	38
5.1	Importing the model to be made:.....	38
5.2	Print Program Generation.....	40
5.3	Stepper motor speed setting.....	42
5.4	Bed levelling.....	44
5.5	Program running.....	45
5.6	Test result.....	45
6	MULTIPLANAR AM VERIFICATION .....	47
6.1	Object preparation .....	47
6.2	Simulation and printing configuration.....	48
6.3	Result Evaluation.....	50
7	CONCLUSION AND FUTURE WORK.....	53
7.1	Conclusion.....	53
7.2	Future works .....	54
8	ANNEX.....	55
8.1	Stepper motor controlling program: .....	55
9	REFERENCE.....	56

# ***1 INTRODUCTION***

With the vast improving technologies in manufacturing industry and the changing situation of the global market. There are needs of a manufacturing system, being flexible in making different products with less time and cost consuming ability. Moreover, the system should fit as many working environments as possible. With the thought of simplification, the computerized numerical control (CNC) machining is invented to substitute the process with multiple machines which have only one function. Yet the limits of the CNC machines, such as their size, difficulty in producing complex structured objects, and the immobility, are arousing the people's desire of designing new manufacturing systems. The goals are quite clear: less modification of the system for producing more types of products, the adaptability to various working environments, the mobility and less costly if possible.

## 2 ADDITIVE MANUFACTURING

Additive manufacturing (AM) is an automated manufacturing technology that builds components layer by layer. Originally known as rapid prototyping, AM uses slicing software to define the process of forming a scaled 3D object by generating each layer and connecting voxels. Unlike conventional subtractive manufacturing techniques such as milling or drilling, AM does not require process-dependent tools. This gives it an advantage in producing complex structures that are difficult to realize using traditional methods and can simplify the production process while shortening the production cycle.

The entire process chain of additive manufacturing can be divided into four phases: engineering design, pre-processing, additive manufacturing, and post-processing (see Figure 2.1).

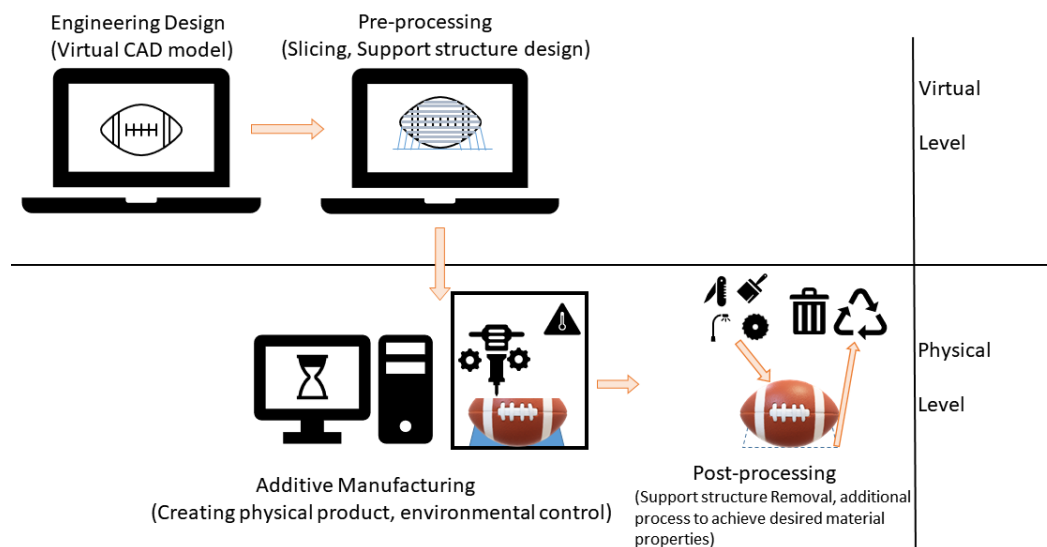


Figure 2.1: Additive manufacturing process chain

In the engineering design step, a 3D CAD file, representing the part to be created must be made. The subsequent process is pre-processing, where it is necessary to generate a manufacturing path file of the sliced model while considering the minimum feature and the support structure. Additionally, the design should conform to the chosen type of AM. Different processes will be described in the following paragraph.

During the manufacturing step, related variables and environmental conditions must be controlled according to the specific AM process. The final part is post-processing, which includes steps such as machining, sintering and polishing to achieve desired material properties.

## **2.1 Classification of Additive Manufacturing**

Defined in ISO 52900, AM processes can be divided into 7 categories, depending on the type of energy used for material physical transformation and the technique used to connect the layers: binder jetting (BJT), directed energy deposition (DED), material extrusion (MEX), material jetting (MJT), powder bed fusion (PBF), sheet lamination (SHL) and vat photopolymerization (VPP) [1]. The following section provides a brief description of each method. This classification is made based on the material used and the final properties of the product in the AM processes.

### **2.1.1 Binder Jetting (BJT)**

Binder jetting (BJT) is an AM process in which liquid binders are selectively injected on top of the powder bed to consolidate powder material into net shapes. As shown in Figure 2.2, a thin layer of granules (c) is distributed on the powder bed (f), and the binder droplets are injected into the surface by a moving head to form a cross-sectional layer. Every time a layer of the product is generated, a new layer of powder will be spread across the surface again. Until being built completely, the part (d) will remain in the unbound powder (e), which works as the support of the part. Finally, some thermal treatment or subsequent sintering is applied to remove the binder in the product to obtain strength and density.

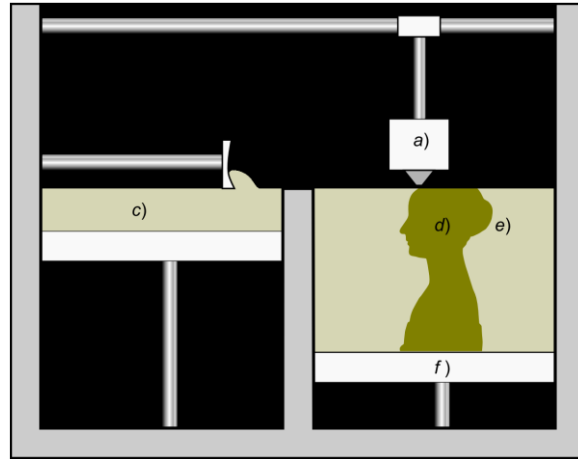


Figure 2.2: Granular Binding. Source: [2]

The powder material varies from plastics, metals, ceramics and others. As suitable binders are to be found for each kind of powder, the range of materials is nearly unlimited, including applications in the fields of food and pharmaceutical products. [3]



### 2.1.2 Directed Energy Deposition (DED)

Directed energy deposition (DED) is a type of AM process that deposits and fuses melted materials by focusing thermal energy. One well-known technique among DED is Direct metal deposition (DMD), which process metal materials and was developed in the University of Michigan [4]. In DMD, a high-power laser beam is used to melt the metal powder fed from side together with the work surface to directly form a new section on the working part as shown in Figure 2.3. It is important to note that the feedback system embedded to the DMD is the key characteristic to distinguish it from other DED processes. It senses the light from the workplace then controls all the related parameters allowing for the deposition thickness to be maintained.

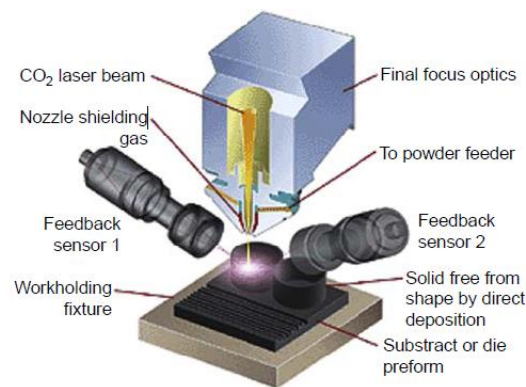


Figure 2.3: Direct metal deposition process with an active height controller. Source:[5]

Another DED process uses wire feeders instead of powder. Regardless of the type of material used, post-machining is usually inevitable due to the poor surface quality compared to other AM processes. However, the process doesn't require support, the product can be made without being sliced into stair-shape by the CAD software in advance.

### 2.1.3 Material Extrusion (MEX)

Often referred to as Fused Filament Fabrication (FFF), Fused Layer Modeling (FLM), or Fused Deposition Modelling (FDM), Material Extrusion (MEX) is an AM process that uses raw material pre-fabricated into thin filament. The material is heated and extruded through a nozzle to create precise deposits on a build platform, as shown in Figure 1.4. The build platform moves in Z-axis to control the layer thickness, while the nozzle moves in X and Y axis to define the extrusion position.

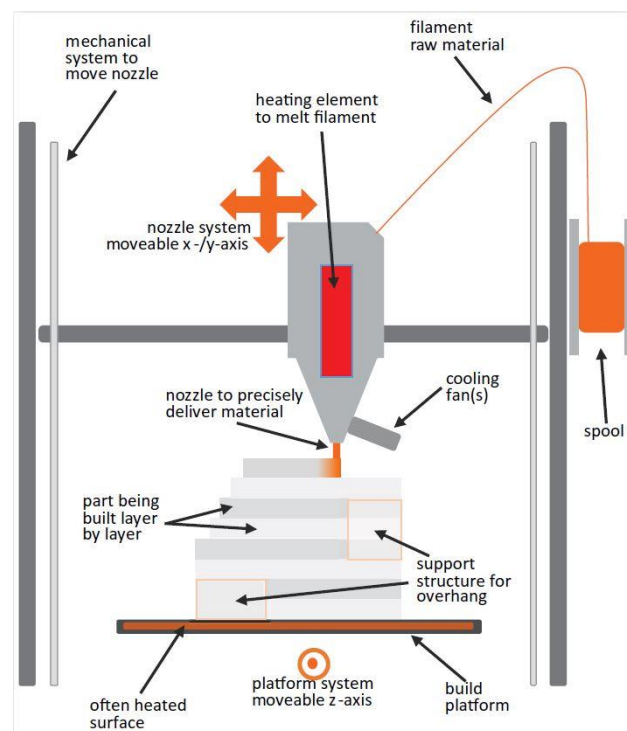


Figure 2.4: FFF/FDM process basics. Source:[6]

A wide variety of thermoplastic materials are available for MEX, including popular options such as Polylactic Acid (PLA) and Acrylonitrile Butadiene Styrene (ABS), as well as less common materials like Nylon, Polyethylene Terephthalate (PET) and Polypropylene (PP). Some specialized materials have been developed by research facilities to meet specific requirements, such as aerospace grade fireproof or medical reproduction.

With respect to target resolution, building volume, and the material, the range of

machines covers from personal 3D printers to industrial production systems. Among all AM processes, MEX is the most widespread, due to its relatively low cost. This characteristic enables applications of MEX to be used in different fields such as industry, research, education, healthcare, and more.

#### 2.1.4 Material Jetting (MJT)

Material jetting (MJT) refers to the AM processes selectively dropping build material to the work platform. A schematic of MJT is shown in Figure 1.5. Resin material with support material is dispensed together by multiple printheads and cured simultaneously by a UV lamp. The formation of the building part represents a photopolymerization process.

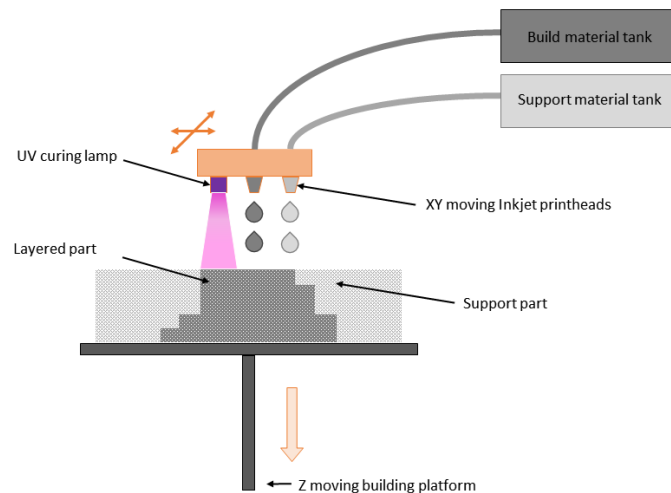


Figure 2.5: MJT process basics

This process allows for the production of products with smooth surface, high precision, and fine details. However, the disadvantage of MJT is its long production time, which could be reduced by adapting extra print heads are adapted due to the large number of layers involved.

#### 2.1.5 Powder Bed Fusion (PBF)

PBF is a family name for several similar processes. A difference between PBF and

DED is that the powder used in DED are often larger in size and require higher energy density than the PBF [7]. The term selective laser sintering (SLS) and laser sintering (LS) are primarily used for machines processing plastics. Whereas selective laser melting (SLM) is referred to as metal processing technique. Electron beam melting (EBM) is another term when an electron beam is used instead of a laser. Despite many different names, the principles are the same. Plastic or metal materials in powder shape are dispersed across the moveable platform of the build space. A power source focusing the energy selectively on the surface to create the contour and cross-section of the part by melting and re-solidification after cooling (fusion). Then the build platform moves downward to leave the space for the next layer of powder material to be dispersed. In Figure 1.6, an SLS process following the principle described before is presented. The recyclable un-melted powder in the building space supports the product from tilting.

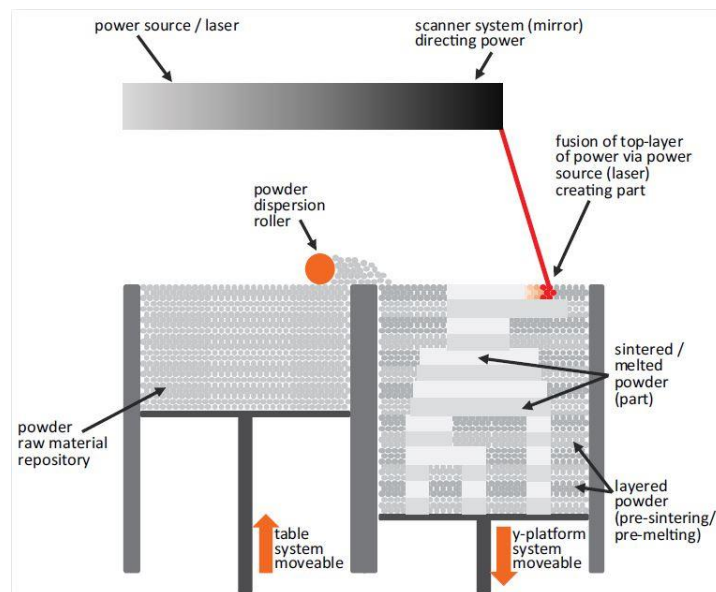


Figure 2.6: SLS process basics. Source:[6]

Allowing to handle various kinds of material, such as metal, ceramics and polymer, meanwhile maintaining good material properties, PBF is suitable for creating functional components for industrial use.

### 2.1.6 Sheet Lamination (SHL)

Cutting contours from constant thickness sheet material and bonding each layer with the surface from the previous one is called sheet lamination (SHL). The most common technique from this family is called Laminated Object Manufacturing (LOM). Figure 2.7 is a schematic of a LOM machine. To produce a part, the material sheet is positioned on the building platform by a heated roller, which activates the bonding substance. The laser beam is reflected by the mirror and guided by an XY-direction moving optic head to cut the contour of each layer of the product from the sheet. An additional tool, usually being a laser cutting system as well, cuts the rest of the entire layer off from the material sheet according to the size of the platform to build the support. After the building procedure is finished, the support can be easily removed for being cut as square blocks previously.

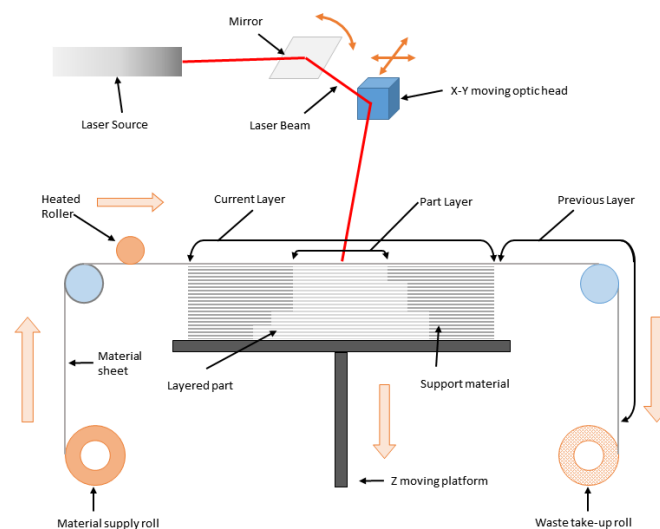


Figure 2.7 SHL process basics

The material sheet for SHL is usually made from plastics, paper, ceramics, or metals. Not only laser beams, blades or even mills can be used as a cutting tool. Glue, soldering, and welding are applied to join each layer in different types of machines. Overall, SHL mostly requires only one manufacturing step to build a product, without a post-processing step following. Thanks to the speed and low cost of this process, SHL

remains in a good position in making conceptual visual prototypes.

### 2.1.7 VAT Photopolymerization (VPP)

Vat photopolymerization is a generic term of process which represents all similar AM processes, “producing parts from photopolymer materials in a liquid state using one or more lasers to selectively cure to a predetermined thickness and harden the material in to shape layer upon layer”, defined from ISO 52900 [1].

Various kinds of AM system use this technique to produce parts, only differs in the method ultraviolet (UV) radiation is generated. For example, Laser stereolithography (SL) uses a mirror to guide the UV laser to create and cure each layer from the liquid photopolymer resin (vat), as well as keep its connection to the previous one as shown in Figure 2.8.

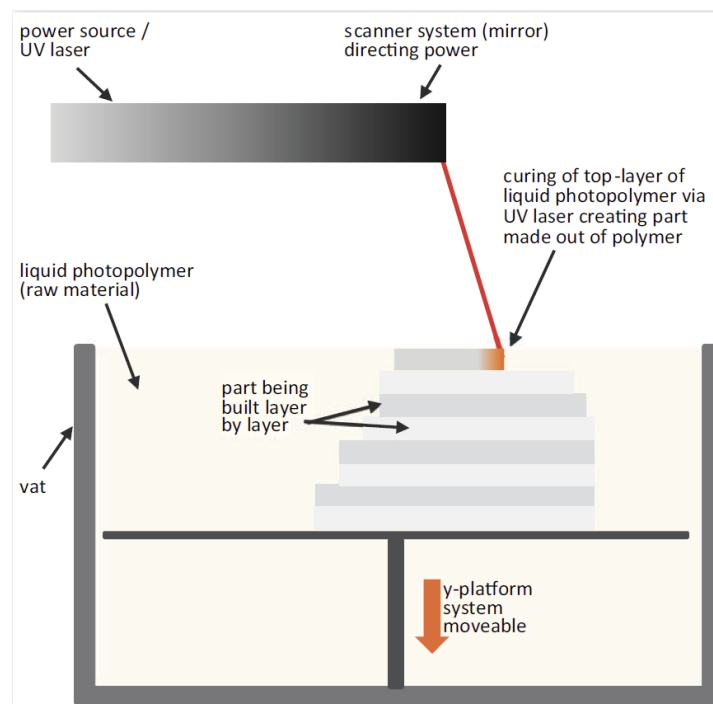


Figure 2.8: SL process basics. Source:[6]

Some other techniques, such as polymer printing/jetting, where liquid monomer material is applied by jetting heads, or digital light processing (DLP), where the entire layer of a product is cured by a single UV light flash projected from the bottom, have a

place in their respective fields because of their characteristics, although their principles are the same.

Resins are the most used material. To resist heat deformation, carbon or ceramic granule materials are added. VPP has been widely used in creating casting molds, non-functional prototype in industry, dentistry because it's suitable for any situation with requirements of high precision and smooth surface.

## 2.2 Applications and Challenges of Additive Manufacturing

Due to the fact that products with different shapes can be easily produced without modification of AM equipment, this technology is gaining increasingly popularity among small scale businesses and multinational enterprises. It's interesting to know how this technology can be used and how it will develop to change our life.

### 2.2.1 Advantages and disadvantages compared to conventional manufacturing

Compared to subtractive manufacturing or molding, AM is still quite a new technology despite it having been researched and developed for decades. Table 2.1 presented below describes the advantages and disadvantages of AM.

Advantages	Disadvantages
Flexibility in production	Low economic benefit for mass production
Ability to make complex structure	Long production time for one batch
Low prototyping cost	High raw material cost
Low initial investment	Restricted mechanical properties
Inventory reduction	
Low waste of material	

Table 2.1: Advantages and disadvantages of AM with respect to conventional manufacturing

The most significant advantage of AM is the flexibility of production. A conventional manufacturing line usually requires long time and high cost to establish, so it takes the decision maker to consider carefully of the benefit. Whereas an AM line is capable of switching manufacturing products immediately without much additional

cost. So, AM machines are relatively suitable for mid-small manufacturing enterprises which lack long-term customers or stable orders.

Besides its flexibility in production, AM has another competitive edge: manufacturing products with complex structures. It's relatively difficult for tools to reach inside and shape a block of raw material into a complex one-piece part. But with AM, products can be redesigned according to function and geometry requirements which can't be made by conventional manufacturing.

When it comes to economic returns, the largest one is that when AM is implemented to a manufacturing process, the cost of equipment for indirect process is much lower than the conventional one. The biggest benefit of implementing AM technology in the investment casting process has been observed in terms of the elimination of the various machine tool facilities required for producing a metallic die as well as sacrificial pattern.[8] Due to the procedures to make a final product is reduced, the duration for production is reduced, too.

Although AM has many advantages, it still cannot completely replace the traditional manufacturing processes. In mass production, cost reduction and takt time are the key elements which matters. It might require a lot of time and money to build a production line but in terms of long-turn return, subtractive manufacturing and injection molding still show the dominance.

### **2.2.2 Applications of additive manufacturing**

#### **a) Direct manufacturing:**

Bugatti AUTOMOBILES S.A.S. announced the use of AM in producing their state-of-art hyper car Bolide. As per Bugatti, thin-walled (up to 0.5 millimeters) titanium alloy functional parts ,as shown in Figure 2.9 originated from AM machines are created to withstand a tensile strength of  $1,250 \text{ N/mm}^2$  [9]. With the AM technique, the company is able to create hollow push rods that only weigh 100 grams.





Figure 2.9: Bugatti Bolide chassis

#### b) Direct tooling:

With the help of GE additive, EROFIO group produced a mold core which allows the cooling circuits following to the surface of the mold inserts, creating conformal cooling effect [10], as shown in figure

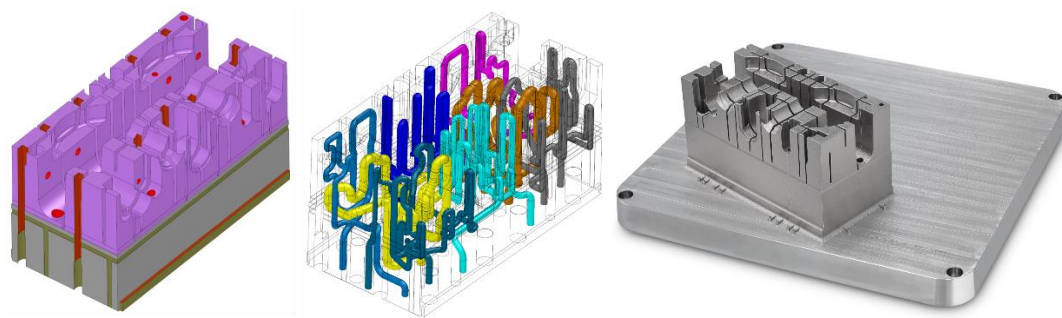


Figure 2.10: EROFIO mold core [10]

The mold contains more than 8 independent cooling channels, which significantly reduces the cycle time for injection molding, while keeping the heat distribution even.

### 2.2.3 Challenges of additive manufacturing

Despite numerous research and examples revealing the potential of AM to resolve the manufacturing industry, there is still a long way to go for both manufacturers and consumers to adopt the concepts. From the examples of the last sub chapter, high end producers are more receptive to technologies that help them to stay ahead of the curve in the industry despite the cost. Whereas manufacturers with high yield focus on process

improvement for cycle time reduction. A case study found that AM adoption happens through successful innovation projects that cover both product and process innovation [11]

## **2.3 Type of AM technique chosen for this project**

For this project, fused deposition modeling (FDM) of MEX was chosen as the implementation of the AM system. The reason is that FDM is considered the most economical and safest method among all AM techniques. After the AM system is built, it will be compared to conventional equipment to verify its performance. Two FDM household 3D printers have been bought. One is scattered to form the end effector of the AM system. The other will be used as comparison.

### ***3 HUMAN COBOT COOPERATION***

Robotics plays an important part in Industry 4.0. Furthermore, connection between machines, from machines to human are the keys to make the next industrial revolution. Some say that robots and artificial intelligence (AI) in the future may completely replace human beings in manufacturing. But based on International Federation of Robotics' research, they share a different view: only 10% of the tasks can be fully automated and the change over time in an unmanned production line is very long thus reduces the capacity.[12]

To increase productivity, but without taking the cost of losing safety and the flexibility of changes in production lines, Collaborative robots (cobots) are created. In this chapter, the characteristics and applications of cobot are presented.

#### **3.1 Characteristics of cobots**

Cobots were invented with the idea of human robot collaboration (HRC). Suggested by Volker et al, HRC was developed with the 2 following aspects: Firstly, an interface should be provided to the user to intuitively translate his idea, executing the robot program correctly. Secondly, feedback should be provided to the user to immediately understand what's happening on the robot side [13]. Hence, in practice, the cobot will require interacting with the human operator frequently. The workplace of a cobot will be shared with humans to complete a task together. Overall, a cobot requires more than the capabilities of an industrial robot, especially about safety regulation and human machine interaction.

In this section, the distinction between collaborative robots and industrial robots will be discussed, and the characteristics of a cobot manufactured by Universal-Robots (UR) will be described in detail.

### 3.1.1 Standards and test specifications

Safety standards regarding cobots are originated from the basic machinery safety standard (ISO-12100:2010). Then in ISO/TR 22100-1:2021, a guideline of how to choose an appropriate standard for designing a machine is provided. Standards on safety of machinery are divided into 3 parts:

- Type-A: Basic standards giving basic principles for design with general concepts about safety of machinery.
- Type-B: Generic standards related to one specific safety aspect for a wide range of machines.
- Type-C: Machine standards specifying the detailed requirement of safety for a particular type of machines [14].

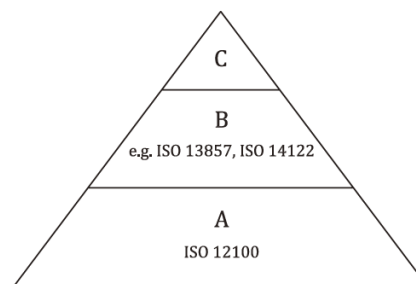


Figure 3.1: General structure of the system of machinery standards [14]

Industrial robot safety design is controlled by ISO-10218-1:2011 (Robots) and ISO-10218-2:2011 (Robot system and integration). Because the working environment of collaborative robots is not consistent with that of industrial robots, the lack of applicable standards leads to research difficulties, coupled with expectation of end users, etc., a standard suitable for cobots ISO/TS-15066:2016 - “Robot and robotic devices- Collaborative robots” was finally published in 2015 after many people called for it.

Not being standalone, ISO/TS 15066 is integrated into ISO 10218-2. Incidental contact with the cobot occurs when operators are staying nearby, how to design the cobot safety systems to avoid injuries to nearby operators is a concern for every robot manufacturer. It is because of the emergence of this series of standards that the

manufacturers and users have obtained sufficient information and recognition on the production and use of cooperative robots.

### **3.1.2 Difference between cobot and industrial robot**

#### **a) Workspace safety**

Both industrial robots and collaborative robots can provide perfect safety protection mechanisms for people. However, industrial robots are equipped with a complex set of guarding system around, while cobots mainly rely on collision detection, force and speed restriction, and when a collision occurs, cobot must make protective feedback actions that meet the standard requirements to avoid injuries.[15]

When sensors detect that a person has entered the operation area, the feedback action of an industrial robot and a collaborative robot is different. Access to the industrial robot will trigger a protective stop, for example, remove objects that hinder the operation. While there are several collaborative operation mode can be selected for a cobot when an operator comes nearby, for instance, safety-rated monitored stop, hand guiding, speed and separation monitoring and etc,[16]. A detailed introduction to the above modes will be described in a later section.

#### **b) Moving speed**

The cobots are intended to be operated at a speed which allows persons to have enough time to retreat from hazardous motion or stop it. As outlined in the standard ISO 10218-1, the speed of tool center point (TCP) shall not exceed 250mm/s when the robot is in a mode when human can intervene [16]. In addition, the standard ISO/TS 15066 specifies the speed limits for other moving parts of the robot and makes more stringent restrictions for different contact positions between the machine parts and human body.[15]

Industrial robots usually work automatically once the program is made and verified. In general, industrial robots run faster than cobots when people are in the safeguarded space.

c) Payload of a robot

There are no strict limits on the maximum payload of a cobot, but its value can affect the transferred energy and robot speed during transient contact with the human body. As per the standard, the transferred energy is limited so that the speed at which the cobot moves through the collaborative workspace, the pressure and force may be applied on human body part can be controlled within the threshold [15].

As is known that the industrial robots are designed for handling heavy material. Fanuc offers an industrial robot which carries 2.3 ton maximum, which is claimed having the world's strongest wrist [17]. According to Cobot Trends, the heaviest payload that a cobot offers is 35 kg [18].



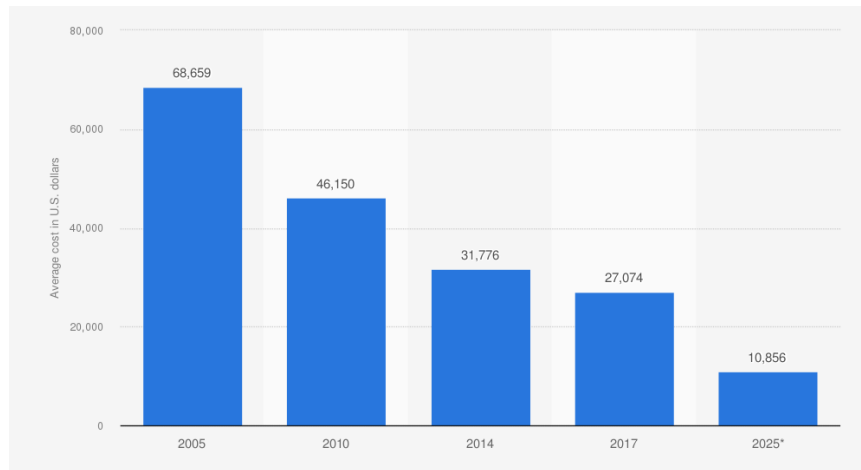
*Figure 3.2: Fanuc industrial robot M-2000iA/2300 [17]*

In general, the maximum payload of an industrial robot is much higher than that of a cobot. When the weight of the object to be handled is small, the workspace safety during the process becomes a crucial factor for choosing which type of robot to use.

d) General cost

The price of industrial robots is lower than that of cobots of the same specification (payload, reachability, etc.). Industrial robots are developed much earlier than collaborative robots, so their manufacturing costs are reduced nowadays.[19] Whereas cobots are new products appearing in recent years. On the other hand, cobots are

equipped with more sensors to meet the requirement for collaborative operations, so that the cost is added.



*Figure 3.3: Average cost of industrial robots in selected years from 2005 to 2017 with a forecast for 2025 [19]*

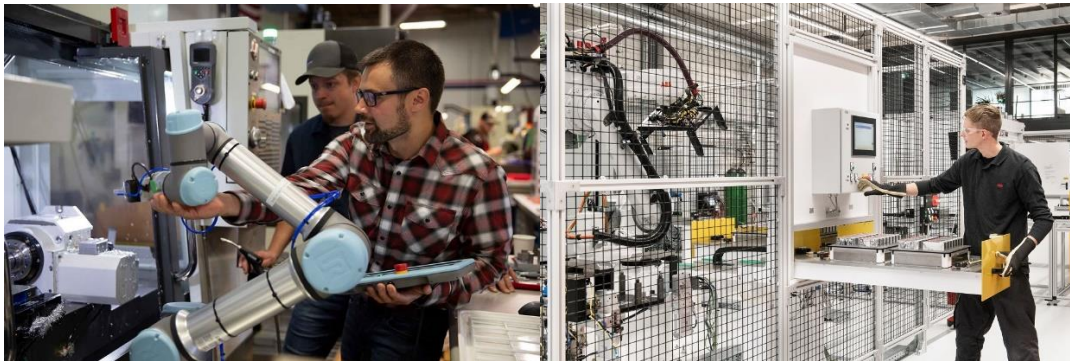
However, choosing to buy a relatively economically priced type of robot doesn't help companies to save money, because these two types of robots are designed for different purposes. Industrial robots are designed to perform one single task with precision at the allowed fastest speed. Other tasks of the workers are adjusted according to the capabilities and workload of the robots. While Cobots are automated solutions made to work with operators interactively. They are designed in lightweight to make sure that they can be easily moved to carry out different works. Unlike industrial robots, the task of collaborative robots is to reduce the workload of employees in a process. Flexibility and adaptability are the priorities to design a cobot.

To sum up, both robots have their own environment to use. The right choice can help the factory reduce the cost and increase the production efficiency, while the wrong choice not only increases the cost but also may cause danger.

e) Ease of programming

The way to program a cobot is more convenient to inexperienced beginners. Operators are allowed to physically guide the cobot by hand to a desired position and

ask the cobot to save the posture. In this way, multiple target locations can be added and linked together to form a sequenced process. In addition, user can select different commands from the menu and place them between the targets in order of operation. All he needs is just holding the teach pendant by one hand and move the cobot by the other hand. For a more sophisticated requirement, as shown in figure 2.4. The manufactures provide other ways for expert level users to program the system, too.

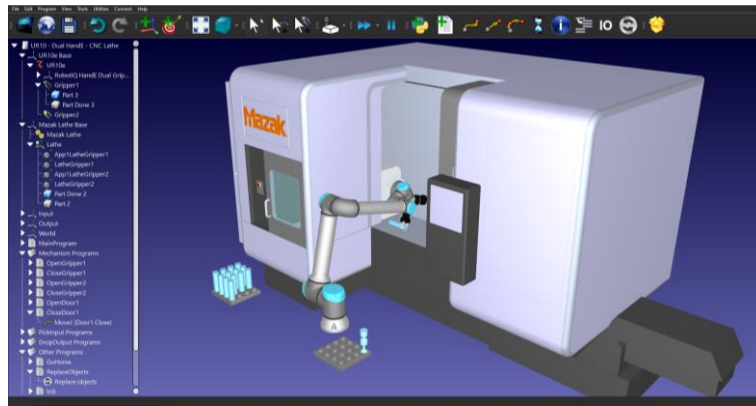


*Figure 3.4: An automation engineer teaching the easy operation of the UR5e cobot that handles the operation of the company's CNC machine (left)[20]. An operator is collecting assembled battery packs from outside the fence and checking the information from the controller (right) [21].*

Almost all industrial robots need to be programmed by proprietary software. The operation of an industrial robot is closely related to the production line process. This requires a host in the network, having access to each part that makes up the production line, to implement monitoring and control. Precision is the key to make a program for industrial robots. Provided with a lot of selections from the PC software or the teach pendant, the operator needs to adjust many details during programming and verification to make the whole process fast and precise.

Thanks to the offline programming and simulation software, creating an application for a robot is becoming easier and easier, even without the actual machine. These simulators help the user to present their ideas in a 3D digital scene and run the program before it is uploaded to the actual robot.





*Figure 3.5: A screenshot of an example provided by RoboDK software shows a cobot performing a pick and place task following the pace of a CNC machine.*

It helps not only to verify the possibilities but also to improve the productivity of an existing robot cell. The biggest contribution of this kind of software is to break down the barriers between different brands of robots, even different kinds of robots and machines. An example is presented in Figure 3.5.

## 3.2 Universal-Robot UR10e and ON Robot end effectors

In this sector, the equipment and tools used in this project will be described. Robot manufacturers usually provide only the robot itself to the customer with its connection, communication and power supply equipment. The end effectors need to be purchased from other manufacturers or customized according to users' requirement. Details about how the manufacturers can integrate their system will be presented.

### 3.2.1 Features of UR10e

Manufactured by Universal-Robot (UR), UR10e is a cobot which belong to e-series collaborative robot system of the company. E-series products are built for OEM automation applications. Being different from their other products, e-series provide the users with a package including various communication port choices, AC or DC power supply options, and settings in accordance with production safety standards.

The cobot is a robot arm with six degrees of freedom. It has built in force sensors which enables the robot to stop motion when excessive external force is detected. The maximum capacity on the end of the robot is 12.5kg which makes it suitable for different applications in packaging, palletizing and machine tending objectives. Detailed specifications are presented in table 2.1.

Robot Technical Specification	
Robot Type	UR10e
Weight	33.5 kg
Reach	1300mm
Maximum Payload	12.5 kg
Speed	Base and Shoulder joints: Max 120 °/s. All other joints: Max 180 °/s Tool: Approx. 1 m/s
System Update Frequency	500 Hz
Force Torque Sensor Accuracy	5.5 N

Pose Repeatability	$\pm 0.05 \text{ mm}$
--------------------	-----------------------

Table 3.1: Cobot UR10e specification

The power of the cobot is given through the control box inside the cabinet. In addition, the control box provides external connection such as USB, Modbus, Ethernet, analog I/O and digital I/O. An operation system called Polyscope is provided to the user through a teach pendant with touch screen connected via cable to the control box. (Figure 3.6)

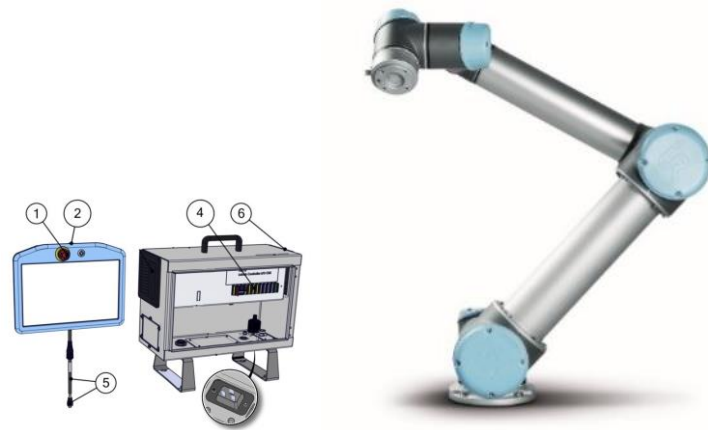


Figure 3.6: UR teach pendant (left), UR control box (middle), UR10e robot arm (right)) [22]

### 3.2.2 Reason to use a cobot in the AM system

An AM system using a cobot is an alternative solution to a conventional machine. In other words, AM could be designed as another function for the cobot. The system has no cost advantage over traditional 3D printers, but it adds convenience to the use of cobots. There may be tasks that require both additive manufacturing and other functions of the cobot, such as repositioning of processing parts during production, or handling objects after production. As can be expected, the possibilities for additive manufacturing using cobots are endless.

# ***4 COBOT ADDITIVE MANUFACTURING SYSTEM DESIGN***

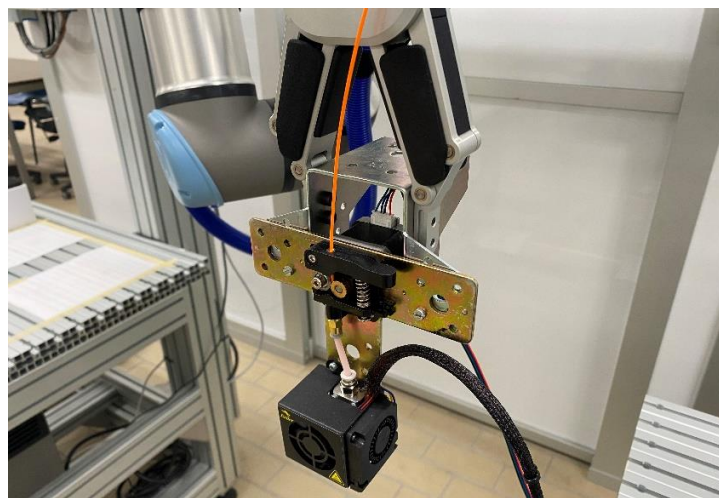
In this chapter, the process from design to implementation of this AM system will be introduced. From end-effector design to cell design, and to control software matching will be elaborated in detail.

## **4.1 Hardware design**

To investigate the possibility of multiplanar object generation, the UR10e cobot is integrated with an extrusion system typically used in MEX type 3D printer.

### **4.1.1 End effector design**

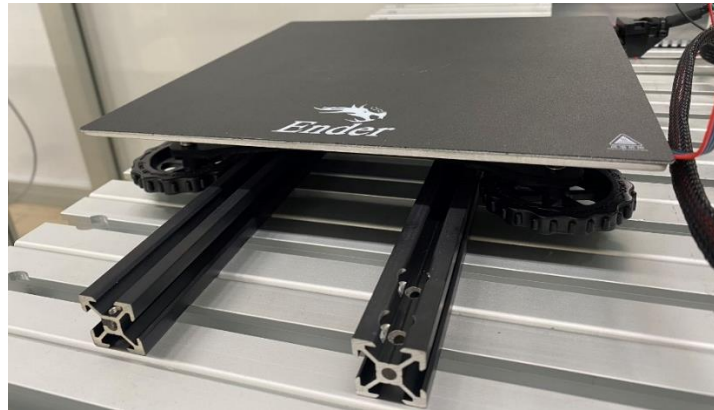
The extrusion system, as shown in Figure 4.1, is composed with two sheet metal brackets and a set of Bowden extruder components (including a NEMA 17 stepper motor, extruder cold side, nozzle hot end) taken from an Creality Ender-3 3D printer. Since the robot may need to install other tools for different tasks, the system is fixed to an OnRobot finger gripper using screws to enable quick changes.



*Figure 4.1: Extrusion system*

### 4.1.2 Printing platform design

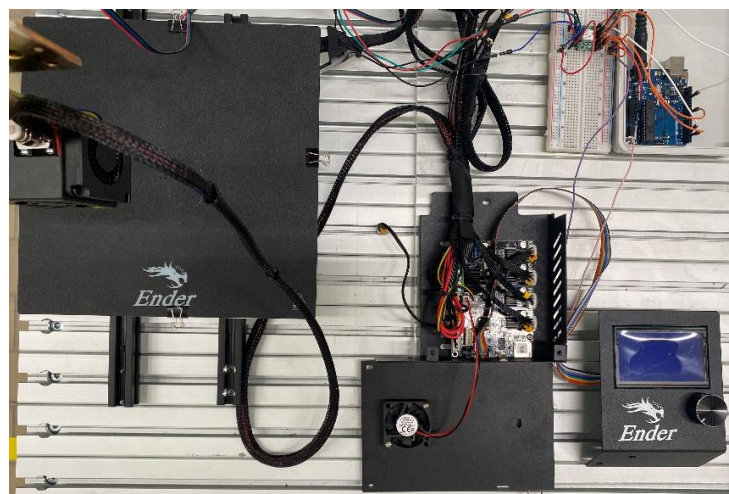
As in Figure 4.2, the printing platform is a heated bed taken from the Ender-3 3D printer. To keep the plane level, it is placed on 2 pieces of 20 x 20 aluminium profile in the same way as it is held on the conventional 3D printer, with all 4 levelling knobs in the bottom adjustable.



*Figure 4.2 Heated bed taken from Ender-3 3D printer.*

### 4.1.3 Other devices

In addition to the devices mentioned above, the entire AM system is built using a control box from UR10e for signal generation, an Arduino Uno and an A4988 driver for controlling the stepper motor, a screen with a knob and the mother board from Ender-3 for controlling the temperature, and a 24V power supply. The collaborative approach of all devices will be explained in more detail in a later section.



*Figure 4.3: The other devices (Temperature and extrusion control)*

## **4.2 Program design**

The conventional 3D printers are typically controlled using G-code generated by slicing software. Conventional 3D printers usually rely on G-code generated by slicing software to control the coordinates of the extruder and the amount of material extruded. To complete the task of additive manufacturing with the cobot, a software is needed to translate the G-code generated by the slicing software into commands that the robot can execute. Therefore, RoboDK is chosen to make the program needed for this thesis because it met exactly this requirement.

### **4.2.1 Work cell design for RoboDK**

RoboDK is a simulation software for robot application and robot programming. One of the advantages of this software is that one doesn't need to operate a real robot to verify the feasibility of his program. It offers APIs supporting various kinds of industrial and collaborative robots, translating programs from software simulation into the language the UR10e cobot can understand.

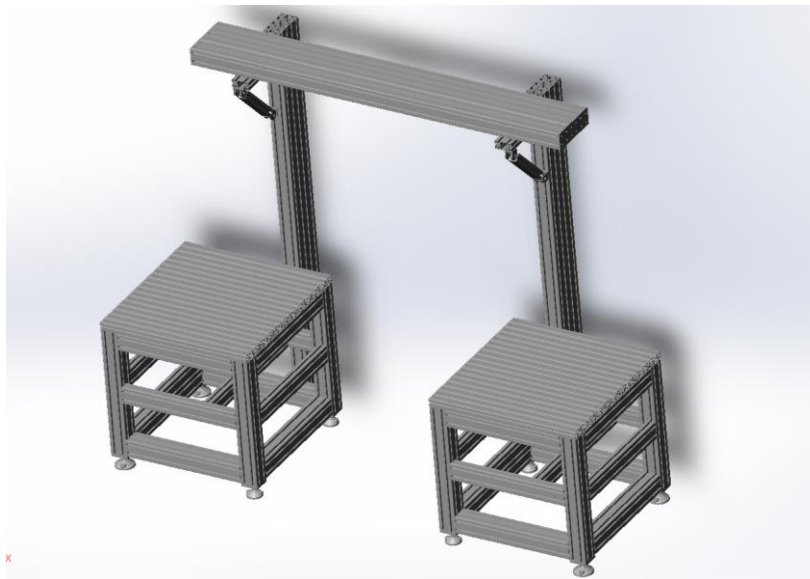
#### **a) work cell digital model replication:**

To make an accurate program, it is required to replicate work cell environment in the RoboDK as precisely as possible. The robot library provided by RoboDK enables the simulated UR10e to precisely imitate the movements of the actual robot, reducing errors and identifying potential collision risks during the printing process.



*Figure 4.4: UR10e work cell*

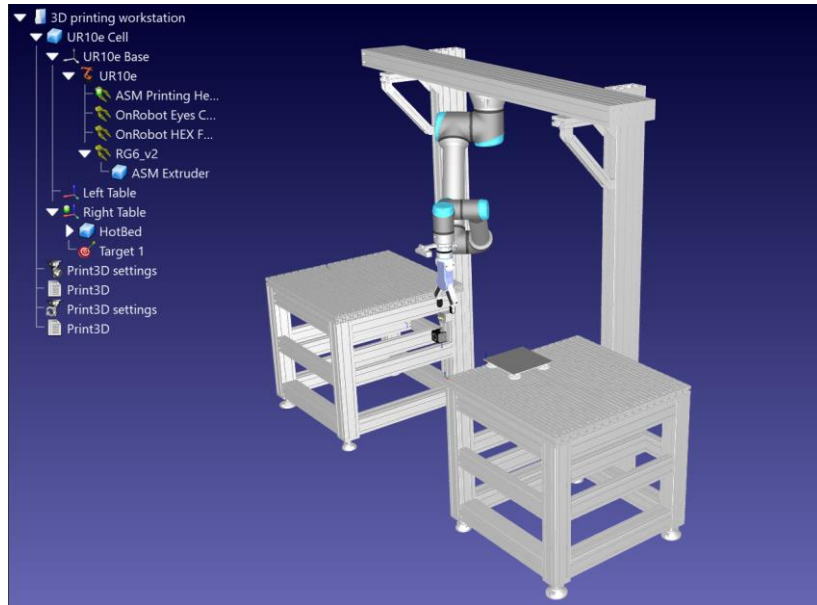
The CAD model of the robotic work cell structure is created in Solidworks according to the parameters measured. Then it is imported into RoboDK as the base where UR10e is fixed.



*Figure 4.5: work cell structure CAD model in Solidworks*

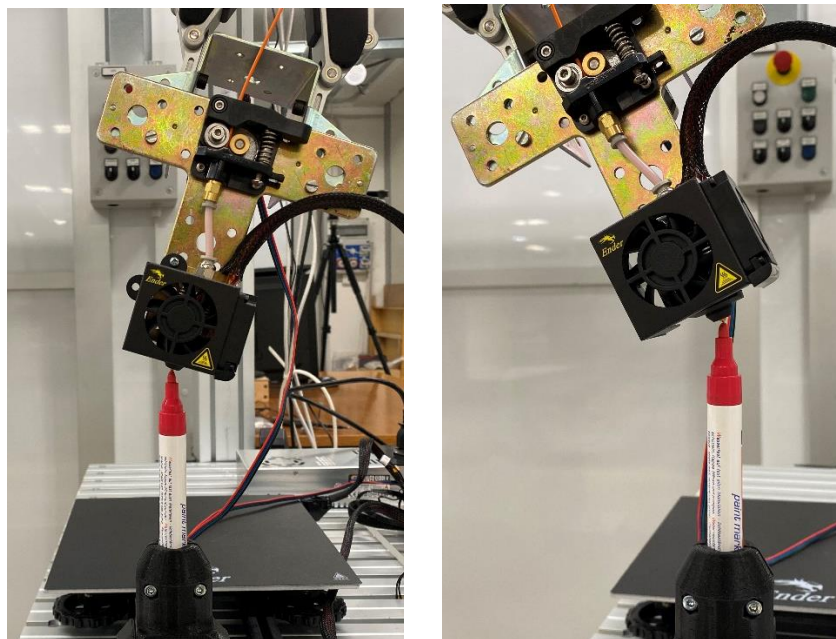
The UR10e digital model and the tools attached on the end of the cobot are imported to the simulation scenario, which referred as a “station”, from the online library. However, non-standard end effectors are not included and require personal 3D models to be imported into the Station, as shown in Figure 4.5. Adjust the geometry of the part with respect to the tool flange and define the tool centre point (TCP) when all necessary components are imported.





*Figure 4.6: work cell digital model in RoboDK*

The TCP position is defined by TCP teaching function supported by UR teach pendant. To set a TCP of a UR cobot, a fixed position represented by the tip of a pen is chosen, move the hot end nozzle from 4 different angles and save the corresponding positions of the end effector. The TCP position of the end effector with respect to the robot base will be calculated automatically.



*Figure 4.7: Teaching TCP by reaching a fixed position in different angles*



In Figure 4.7, to set a TCP of a UR cobot, a fixed position represented by the tip of a pen is chosen. Guide the hot-end nozzle from 4 different angles reaching the tip and save the corresponding positions of the end effector. The TCP position of the end effector with respect to the robot flange will be calculated automatically (Figure 4.8).

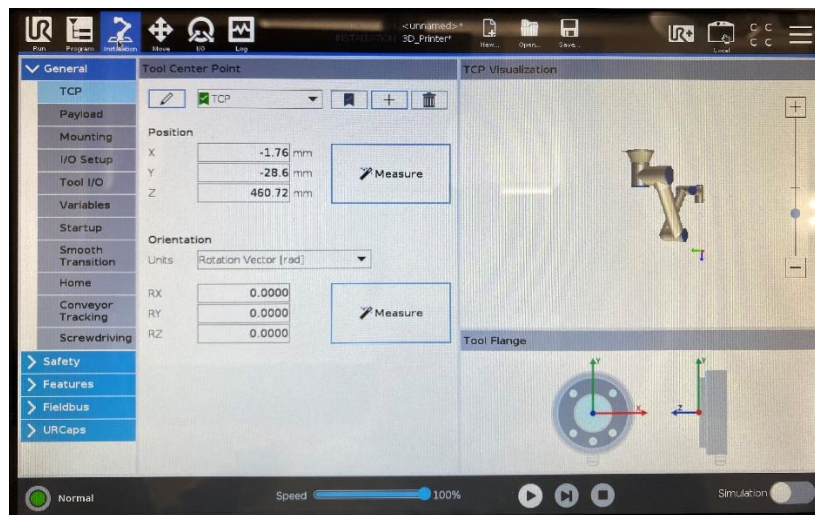


Figure 4.8: TCP position calculated by URsim

The TCP position coordinates obtained by the actual measurement are used to adjust the TCP of the robot digital model in RoboDK.

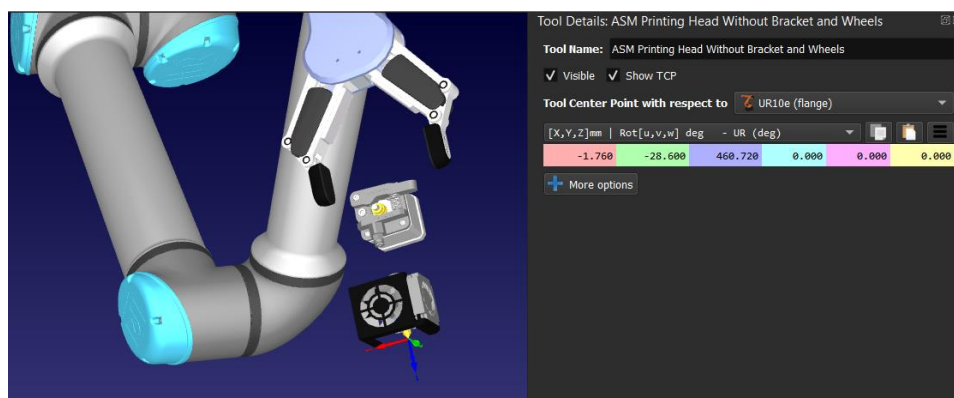


Figure 4.9: TCP configuration in RoboDK

The heated bed is placed in the position as seen from Figure 4.9, and it is elevated from the table by 2 aluminum profiles. The reason for this is to keep the four knobs at the bottom of the heated bed in the air, so that there is space for fine tuning. The next step is to measure the distance from the surface of the heated bed to the tabletop, and

then fill the Z coordinate value of the heated bed position with respect to the table in Robodk as Figure 4.10. Despite fine measurement, the height difference between the bed and robot base is still inaccurate. A bed leveling process should be carried out before running the program on the robot. The method of calibration will be introduced in the test run in Chapter 5.

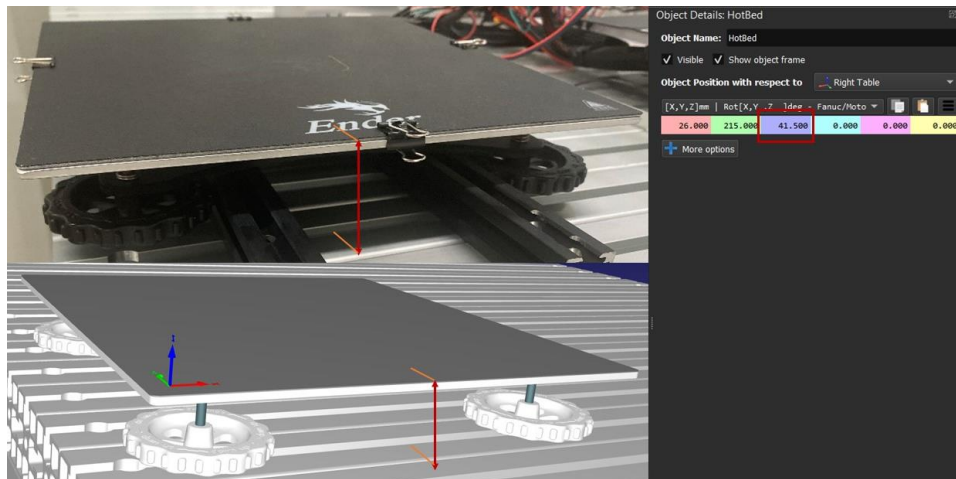


Figure 4.10: Heated bed position configuration according to measurement

## 4.2.2 Extrusion control

The filament is fed by a NEMA 17 stepper motor from the extruder cold end to the nozzle hot end. The A4988 driver, receiving signal from an Arduino microcontroller, controls the stepper motor. The UR control box generates analogue signal to the Arduino to switch on and off the stepper motor (see Figure 4.11).

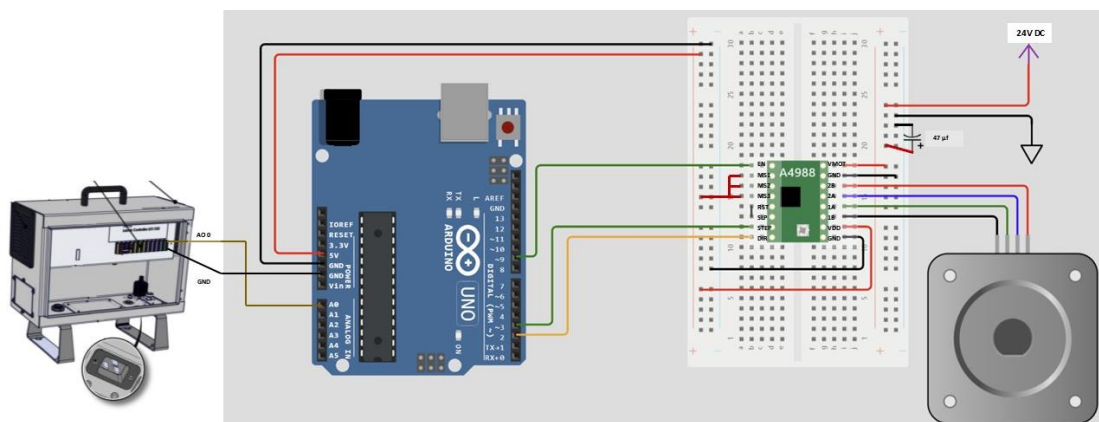


Figure 4.11: Extrusion stepper motor controlling circuit

The speed of the motor is set constant in accordance with the flow rate of the filament and dimension of each layer of material extruded. The program to drive and stop the stepper motor is presented in Annex 1.

While the 24V DC power supply, taken from the Ender-3 printer for the stepper motor is shared with the control board for temperature.

#### 4.2.3 Temperature control

The temperature of the extruder hot end and the heated bed is controlled by the mother board of the Ender-3. They are set manually from the firmware running in the mother board of the Ender-3.

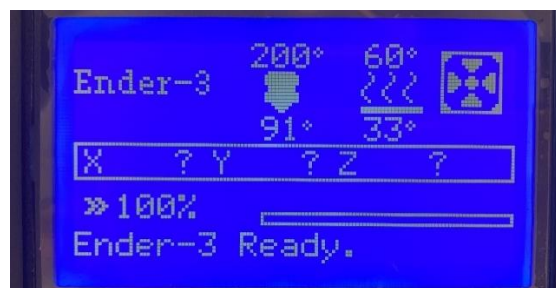


Figure 4.12: Temperature control and monitoring

#### 4.2.4 G-code conversion

Here shows part of a G-code file for 3D printing. The amount of material needed to be extruded before in movement is represented by the E values.

```
G1 X71.668 Y69.274 E2.07096
```

```
G1 X73.820 Y68.481 E2.14192
```

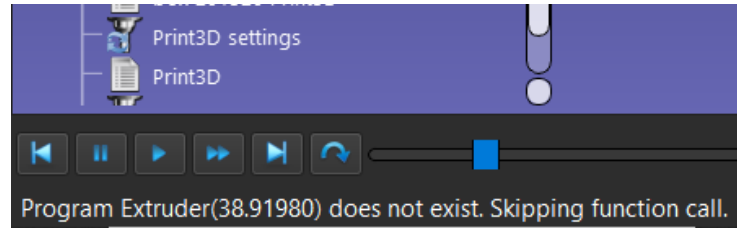
```
G1 X75.000 Y68.375 E2.17858
```

```
G1 X125.000 Y68.375 E3.72564
```

```
G1 X127.259 Y68.772 E3.79660
```

The difference between the values of E in the two commands represents the length of the material extruded as the extruder moves from one position to the other. In RoboDK, the E is translated to a program call named as Extruder and the value as its

parameter, as shown in Figure 4.13. When the 'Extruder()' commands are running in RoboDK, they will be ignored. Despite the simulation can be ran from RoboDK directly on the robot, there will be only robot's movement can be transmitted. To utilize the command for driving the extruder stepper motor, it is needed to use the post processor of RoboDK to convert the value into signal output.



*Figure 4.13 Extruder() value skipped in simulation*

The post processor is a python program built by RoboDK to generate programs for a specific robot controller. A Post Processor is a key component of simulation and offline programming of industrial robots [23]. Each robot vendor has its own programming rule and language. The post processor handles the conversion from simulation to the robot it supports. The software provides a post processor specially designed for using UR robot for 3D printing, yet modification is required according to specific needs in this project.

As discussed before, the hot end movement during the process will be in constant and slow. The first step is to modify the code of the post processor so that the translated program can trigger the start and stop function for the stepper motor, instead of commands with varying value inside. Figure 4.14 indicates the modified code for extrusion signal control. The original codes take care of the Extruder() value, converting the length value in mm from the bracket to a value for the tool controller to adjust the material extrusion speed to match the time for tool movement. While in this project, the tool is moving in 10 mm/s during material dispensing, so the stepper motor for extrusion runs in a constant speed. The modified part simplifies the process by skipping the time value and ask the robot controller to output an analogue value which is high enough for the Arduino board to recognize.

```

if add_material > 0:
    distance_mm = norm(subs3(self.PRINT_POSE_LAST.Pos(), pose2.Pos()))
    # calculate movement time in seconds
    time_s = Calculate_Time(distance_mm, self.SPEED_MMS, self.PRINT_ACCEL_MMSS)
    signal = 5
    self.setA0(self.PRINT_E_A0, signal)
else:
    # DO not add material
    self.setA0(self.PRINT_E_A0, "0")

# Remember the last position
self.PRINT_POSE_LAST = pose2

```

*Figure 4.14 Modified extrusion control codes*

In conclusion, when material needs to be dispensed, a 5V analogue signal is sent from the I/O port 0 on the UR control box. When no material is needed, the signal stops. Then once the printing program is running in the UR10e, the analogue signal will be received by Arduino micro controller at port A0, driving the stepper motor running in a constant speed, while stopping extrusion if no material needs to be dispensed.

## 5 TEST RUNNING

Before the system can be used to produce objects for evaluation, a calibration test run must be performed. This involves printing several sample objects, as mentioned earlier, until the filament extrusion becomes steady, and the layer-to-layer adhesion is satisfactory.

### 5.1 Importing the model to be made:

When used for the first time, the system can be unstable, and its performance may be unknown. Therefore, in the first experiment, a simple-shaped object with a few materials is imported to calibrate the system.

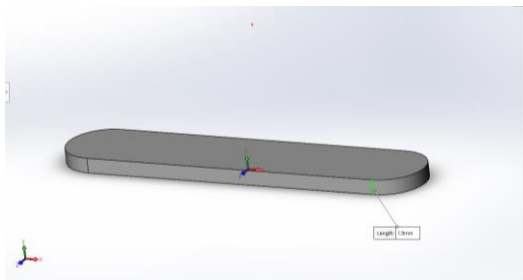


Figure 5.1: test piece in Solidworks

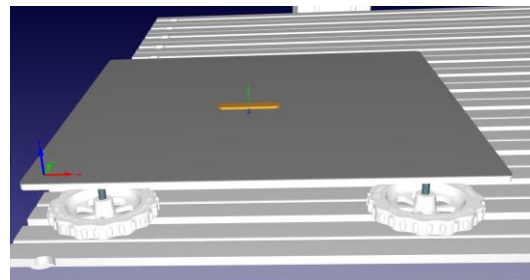


Figure 5.2: test piece imported to RoboDK

After the object is imported, RoboDk's built in 3D printing utility is used to generate the tool path. It's necessary to change the tool orientation and the starting point of the robot, eliminate redundant paths and reduce the possibility of collision during the process. In this experiment, the tool orientation is rotated 180 degrees around the flange Z-axis to avoid excessive pulling forces on the cables around the cobot. In addition, the posture of the cobot at the starting point is adjusted according to the reality. The adjusted parameters have been marked in the red boxes in Figure 5.3.

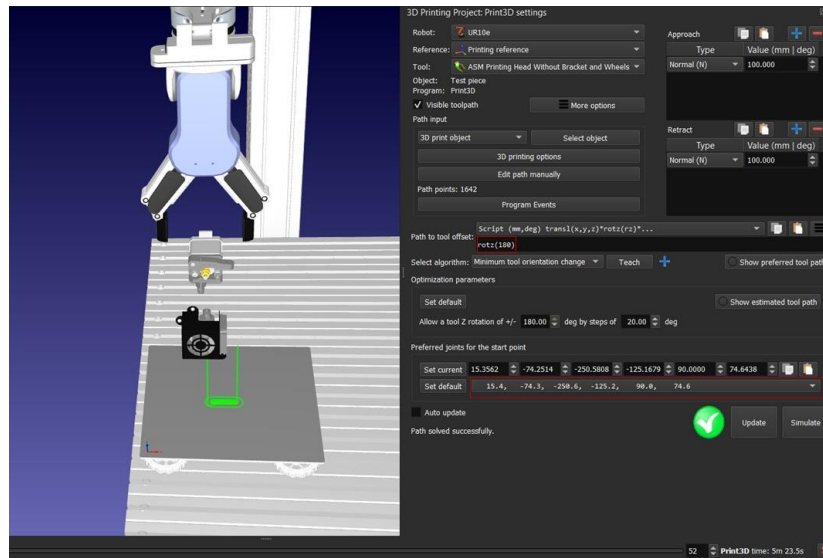


Figure 5.3: RoboDK printing configuration

The next step involves setting the parameters in the slicing software. The default slicing software in RoboDK is Slic3r, which is used for convenience in this project.

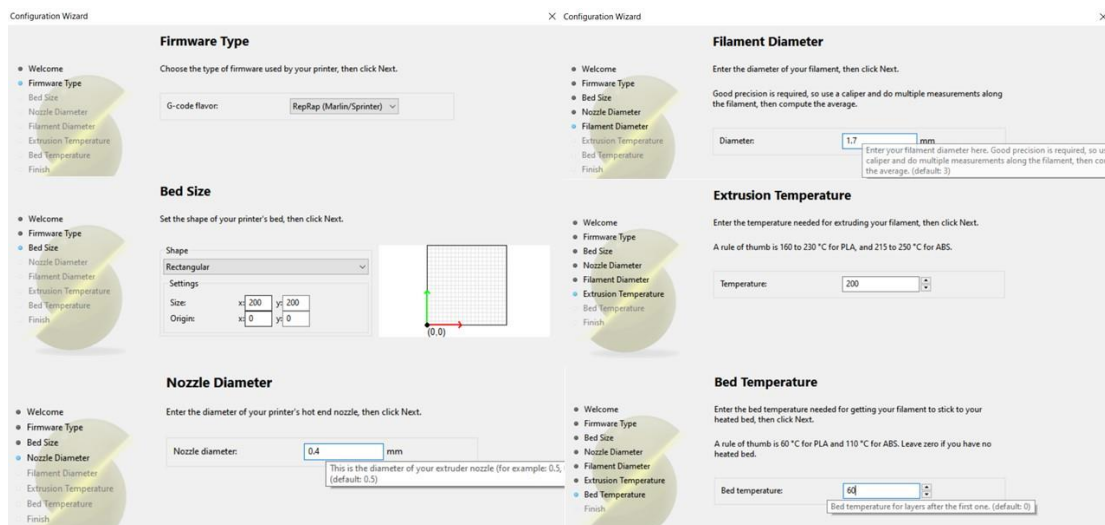


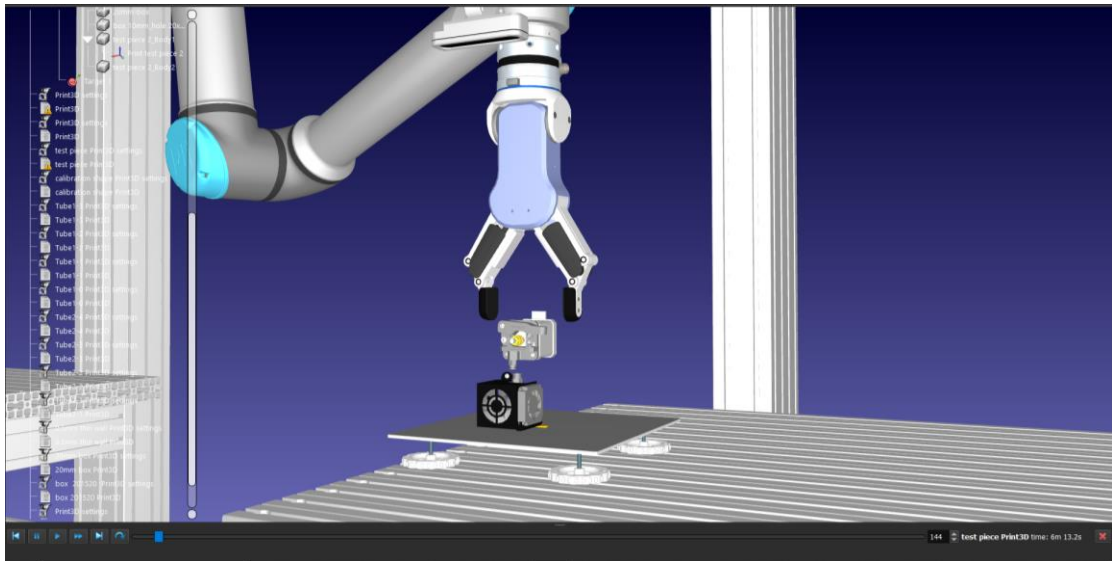
Figure 5.4: Slic3r configuration

In configuration wizard of Slic3r as shown in Figure 5.4, the most important parameters are the bed size, the nozzle diameter, and the filament diameter. Firmware type and bed temperature could be left blank or set as default. These two parameters are useful if the temperature control needs to be integrated with the main system, which in this case is a separated one.



## 5.2 Print Program Generation

Before transferring the program, run the simulation in RoboDK to check if there is collision during the process as shown in Figure 5.5. A printing program written in UR Script will be generated as per simulation when the modified UR 3D printing post processor is selected.



*Figure 5.5 Simulation running in RoboDK for collision detection*

Figure 5.6 shows the beginning part of the printing program. First, it defines the moving speed, blend radius and the TCP. What follows behind are the commands for moving the coordinates, with defined acceleration, speed and whether to activate the analog output during the movement. As can be seen from the program, the analog output commands are telling the analog I/O 0 port in UR control box to output 5V or 0V.



```

# Main program:

# Program generated by RoboDK v5.4.2 for UR10e on 19/11/2022 15:58:58

# Using nominal kinematics.

blend_radius_m = 0.001

speed_ms      = 0.083

speed_ms      = 0.100

# set_reference(p[-0.604289, -0.488399, 1.123500, 0.000000, -3.141593, 0.000000])

set_tcp(p[-0.001760, -0.028600, 0.460720, 0.000000, 0.000000, 0.000000])

# Show ASM Printing Head Without Bracket and Wheels

movej([0.268016, -1.342235, 1.866787, -2.095355, 1.570796, 1.302780], accel_radss, speed_rads, 0, 0)

movel(p[-0.681921, -0.395986, 1.118500, 0.000000, 0.000000, 0.000000], accel_mss, speed_ms, 0, 0)

speed_ms      = 0.020

set_standard_analog_out(0, 0)

movel(p[-0.681921, -0.395986, 1.123100, 0.000000, 0.000000, 0.000000], accel_mss, speed_ms, 0, blend_radius_m)

speed_ms      = 0.010

set_standard_analog_out(0, 5)

movel(p[-0.683226, -0.397027, 1.123100, 0.000000, 0.000000, 0.000000], accel_mss, speed_ms, 0, 0.001)

set_standard_analog_out(0, 5)

movel(p[-0.685918, -0.398323, 1.123100, 0.000000, 0.000000, 0.000000], accel_mss, speed_ms, 0, blend_radius_m)

set_standard_analog_out(0, 5)

movel(p[-0.687545, -0.398695, 1.123100, 0.000000, 0.000000, 0.000000], accel_mss, speed_ms, 0, 0.001)

set_standard_analog_out(0, 5)

movel(p[-0.689039, -0.398863, 1.123100, 0.000000, 0.000000, 0.000000], accel_mss, speed_ms, 0, 0.001)

```

Figure 5.6 Printing program generated by RoboDK written in UR Script

The printing program is loaded from USB drive by using the socket on the UR10e teach pendant as can be seen in Figure 5.7.

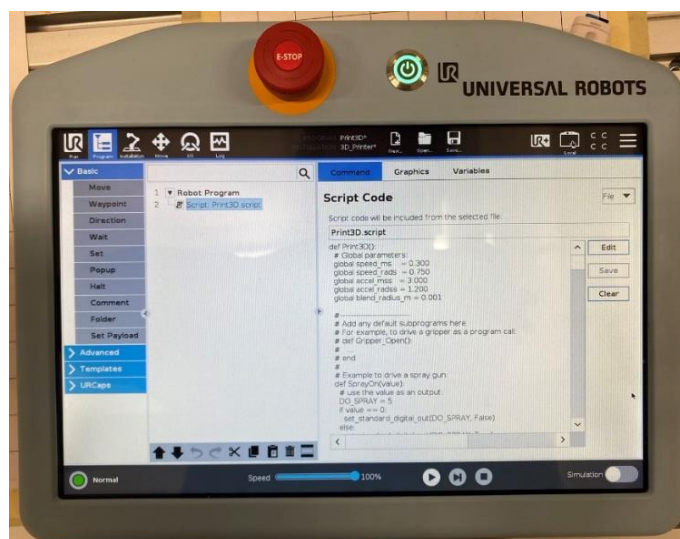


Figure 5.7: Teach pendant loading the program from USB drive on the top

### 5.3 Stepper motor speed setting

The maximum allowed printing speed for PLA material at 60mm/s. For maximum accuracy, the speed is set to 10mm/s. The speed of the stepper motor to be inserted into the Arduino program is got as the following procedure.

First, it is required to calculate the material fed rate in mm/s. Open the Slic3r from RoboDK 3D print configuration. The layer height, filament diameter, and speed for print moves shall be inserted into the Slic3r correctly in the configuration as shown in Figure 5.8. Export the G-code from Slic3r, use another app to check the time and material consumption. In this thesis, an open-source app called gCodeViewer is used [24]. This allows to get the estimated print time and total length of the filament as per Figure 5.9.

The image shows the Slic3r configuration interface with the following settings:

- Layer height:** Layer height: 0.3 mm, First layer height: 0.3 mm or %, Use adaptive slicing: ☐, Adaptive quality: 75 %, Match horizontal surfaces: ☐
- Filament:** Color: [Color swatch], Diameter: 1.75 mm, Extrusion multiplier: 1
- Size:** Nozzle diameter: 0.4 mm
- Speed for print moves:** Perimeters: 10 mm/s, ↳ small: 10 mm/s or %, ↳ external: 10 mm/s or %, Infill: 10 mm/s, ↳ solid: 10 mm/s or %, ↳ top solid: 10 mm/s or %, ↳ gaps: 10 mm/s or %, Bridges: 10 mm/s, Support material: 10 mm/s, ↳ interface: 100% mm/s or %
- Speed for non-print moves:** Travel: 20 mm/s
- Modifiers:** First layer speed: 10 mm/s or %

Figure 5.8 Parameters configuration in Slic3r

Model size is: 50.88x20.57x1.20mm  
 Total filament used: 181.45mm  
 Total filament weight used: 0.45grams  
 Estimated print time: 0:5:35  
 Estimated layer height: 0.30mm  
 Layer count: 5printed, 7visited  
 Time cost: 0.09  
 Filament cost: 0.02

*Figure 5.9 Filament calculation based on 3D model*

Through these two data, the linear speed at the gear of the extruder can be obtained, and finally the rotational angular speed of the stepper motor can be calculated.

The  $v_{extrusion}$  represents the speed which the filament is fed into the extruder. The total length of the filament used and total time for the process are used as the equation below.

$$v_{extrusion} = \frac{L_{filament}}{t_{print}} = \frac{181.45 [mm]}{335 [s]} = 0.54 [mm/s]$$

Using the result given, we calculate the rotational speed of the gear on the extruder. The radius of the gear is half of the diameter measured by the caliper.

$$\omega \cdot \pi \cdot \frac{r_{gear}}{180} = v_{extrusion}$$

$$\omega = \frac{v_{extrusion}}{\pi \cdot \frac{r_{gear}}{180}} = \frac{0.54}{\pi \cdot \frac{5.5}{180}} = 5.63 [^\circ/s]$$

To transfer the rotational speed in  $[^\circ/s]$  to  $[step/s]$ , that is used in Arduino program, an additional calculation should be done. It is known that the stepper motor runs at  $1.8 [^\circ/step]$  when it is running in full-step mode, and the stepper motor is currently running in  $16^{th}$ -step mode. To keep its running speed at  $5.6 [^\circ/s]$ , the following calculations are required.

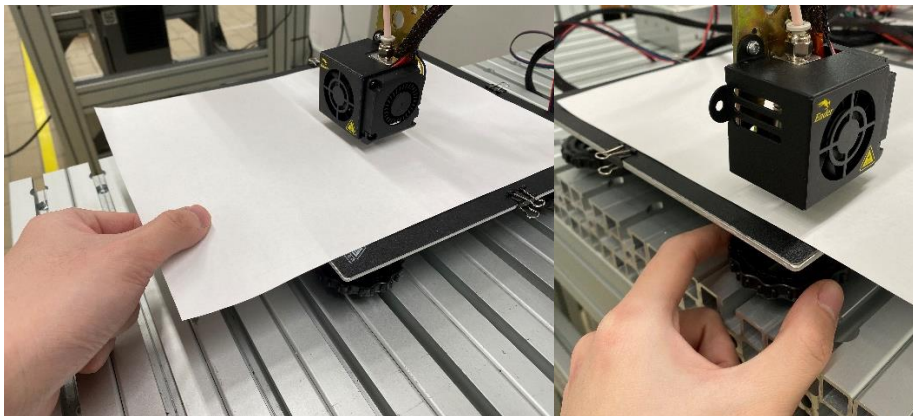
$$\frac{1 \cdot 16 [\text{step/s}]}{1.8 [\text{o/s}]} = \frac{x [\text{step/s}]}{5.63 [\text{o/s}]}$$

$$x = 50 [\text{step/s}]$$

So that 50 is the number to be inserted into the program in the Arduino.

## 5.4 Bed levelling

To ensure high production quality, it is important to align the heated bed with the coordinate system of the cobot. First, all components of the end effector should be assembled on the cobot tool flange. Then, use the automatic align function on the teach pendant to align the Z-axis of the tool flange with the Z-axis of the cobot base. Next, run the program transferred to the teach pendant at a reduced speed. When the cobot starts to move down towards the heated bed, stop it immediately once the end effector is running horizontally or the nozzle tip is touching the bed. Place a piece of paper between the nozzle tip and the heated bed and adjust the height of the heated bed using the knobs until the paper can be pulled out with some resistance but without hindering the process.



*Figure 5.10: Bed leveling with paper*

After that, record the vertical distance between TCP and the base (Z-value) of the cobot from the teach pendant. Then, move and guide the TCP towards remaining corner of the heated bed and set the Z-value to the recorded value while keeping the

Z-axis of the tool vertical towards each corner of the heated bed. Adjust the height of the heated bed by turning the knob until the paper can be pulled out with only a little friction force.

## **5.5 Program running**

A check list should be followed as following: bed levelling every time running a new program, extrusion on/off signal, wire connection, Arduino board functionality and powering. Once the nozzle and heat bed temperatures are reaching to the targets (200°C for nozzle and 60°C for bed), the program is ready to run. Load the program from USB drive and run it from the teach pendant. Add another command in the end, setting the analog signal to 0 volt, to stop the extrusion signal.

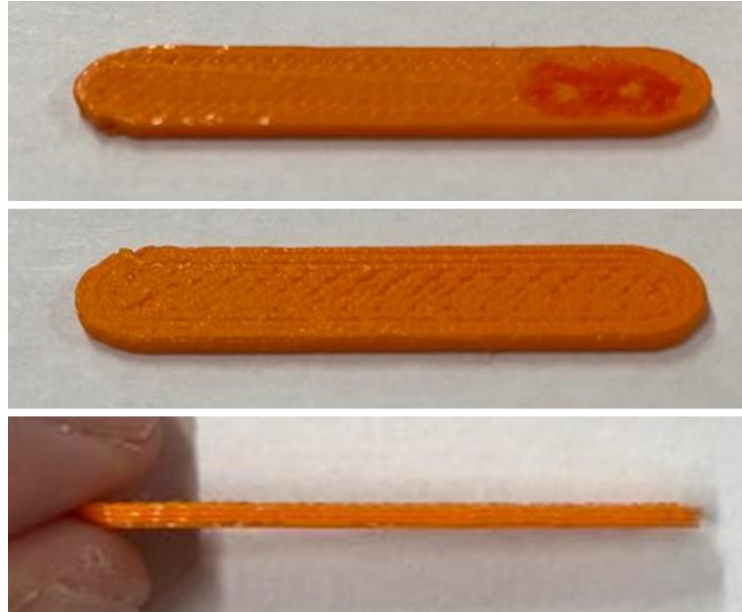
## **5.6 Test result**

To check if the system is working as expected, it is necessary to evaluate the following characters of the generated product.

Layer adherence: Each layer should bond to layer below, so that the final structure will be reliable and strong. If the layers don't bond together, increase the layer thickness.

Surface quality: An even surface could indicate that the material may be extruded too much or too little. It means the extrusion speed or cobot traveling speed is incorrect. Another reason is that the layer thickness is too low that the extruder touched the former layer during extrusion or traveling. In this case, increase the layer thickness setting. The infill density could affect the result as well. If it is set too low, there will be gap between the infill and the object boundary. Add infill density in the setting as necessary.

Layer shifting: If there is layer shifting, it means that the first layer is set too low that the extruder is pressing the heat bed and object. The force pushed the object while printing. Recalibrate the levelling and run the program again.



*Figure 5.11 Printed test piece. Top view (up), bottom view (middle), side view(down)*

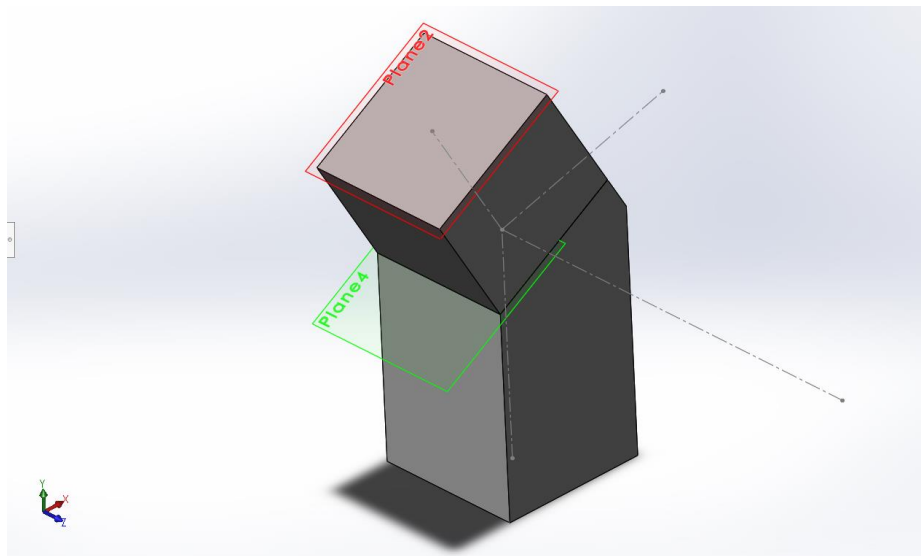
The test sample did not show obvious defects. The AM system is running in an expected condition.

## 6 MULTIPLANAR AM VERIFICATION

Compared to the 3-axis machines, 6-axis robots have more flexible movement space. To find out the possibility of multiplanar AM, the system capability is explored by comparing the dimensions of a generated object with its nominal dimensions defined in Solidworks.

### 6.1 Object preparation

A block-shaped object has been designed in Solidworks to verify multi-planar printing. The object is composed of two cuboids whose centerlines intersect at an angle of  $150^\circ$ . The dimensions of the bottom cuboid are 20 x 20 x 30 mm, while the upper cuboid measures 20 x 20 x 20 mm. The two cuboids are merged at the coinciding part.

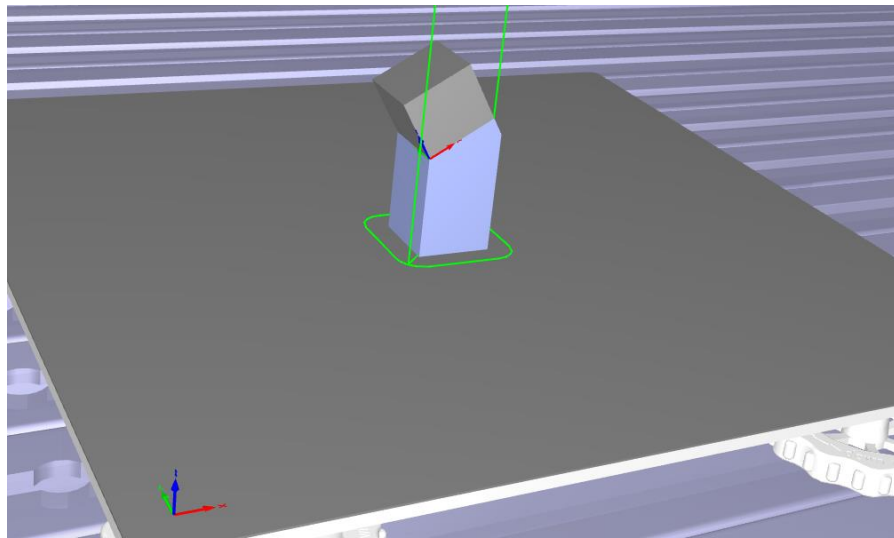


*Figure 6.1 Object 3D model in Solidworks*

From Figure 6.1, it can be observed that the object has been divided into two parts for the purpose of designing two sets of printing paths with different printing reference systems in the subsequent processes. The splitting plane has been selected to be at the intersection of the two cuboids and parallel to the surface of the upper part.

## 6.2 Simulation and printing configuration

The printing object is imported as 2 separate parts into RoboDK simulation scenario. Thanks to the geometry origin predefined in Solidworks, their relative position doesn't need to be adjusted. Then an additional printing reference frame is added at the plane which divides the 2 parts as shown in Figure 6.2.



*Figure 6.2 Object imported to RoboDK AM cell simulation. The green lines are the print path for the first part. Additional printing reference added in the middle*

As same as in the test running section, 3D Print Project utility is used to make the tool path for printing. Different from previous case, a second 3D Print Project is added to generate the upper part of object and the printing reference is the reference frame added at the splitting plane.



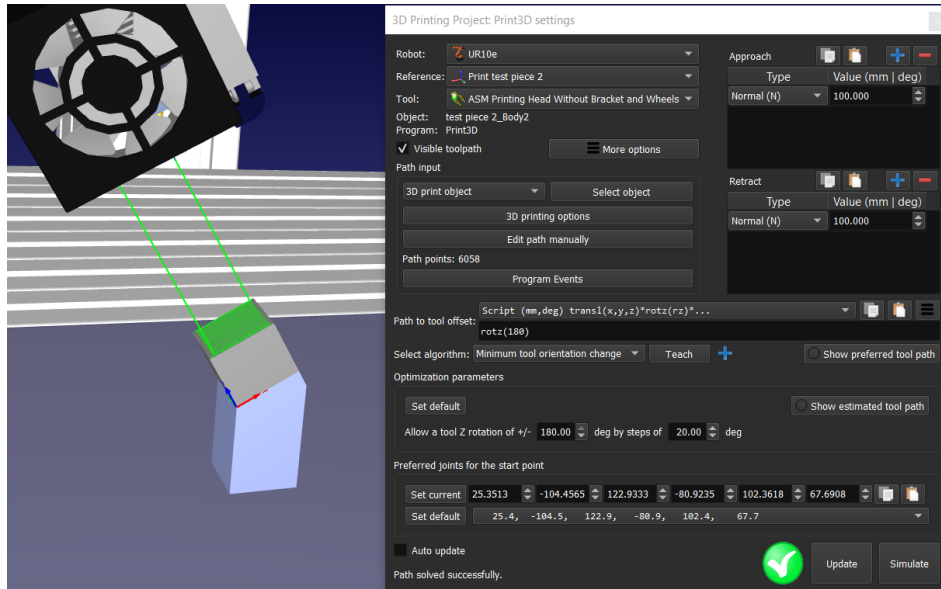


Figure 6.3 Second printing path created based on the additional printing reference

The settings in the Slic3r for these 2 projects are slightly different. The skirt around the second part is eliminated because there is no pre-extrusion needed after finishing the first part as can be seen in Figure 6.4. The remaining parameters are the same as applied in the test running.

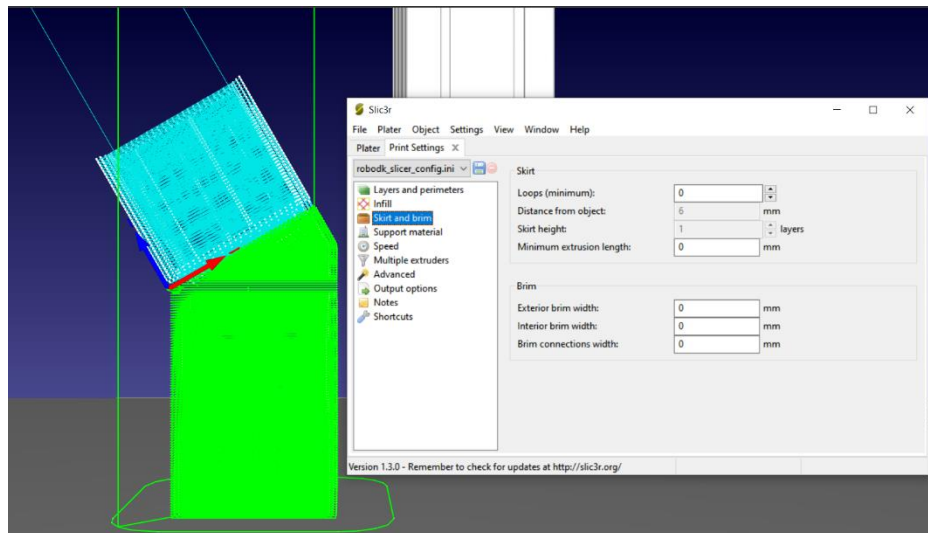


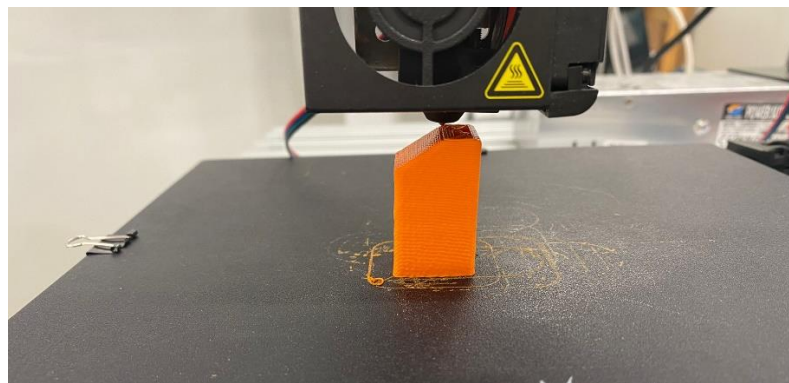
Figure 6.4 Slic3r setting for the second printing paths

Use the edited Universal 3D printing post processor as before to generate two programs for the 2 parts. The generated programs are combined into one and connected together end to end for time saving, so that the filament doesn't cool down when the

cobot is switching the printing reference frame. Check the AM system status and load the printing program. Wait the object until the object is fully generated.

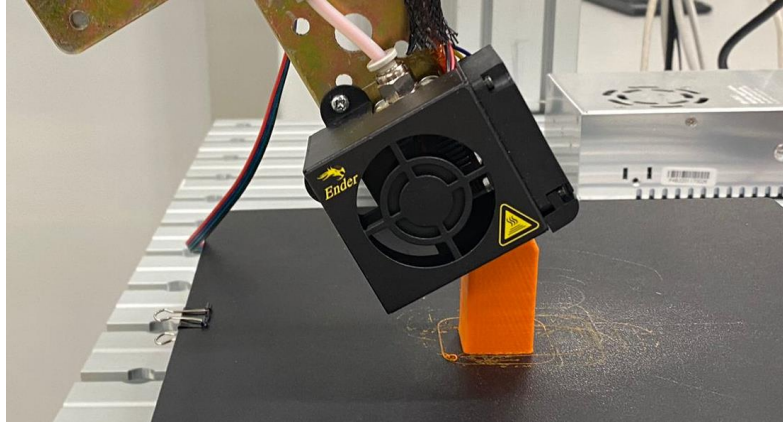
### 6.3 Result Evaluation

An object is printed by the AM system and the program ran smoothly with no malfunction or error during the process. The cobot reached to heated bed vertically and retracted vertically when the first part of the object was finished. In Figure 6.5, it shows the first object is printing in a conventional way.



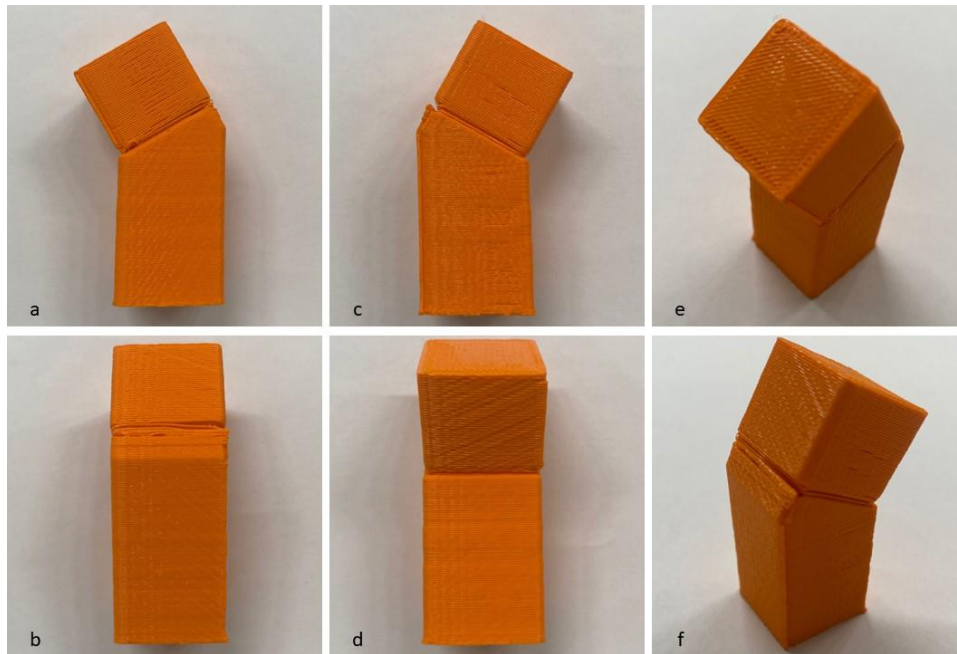
*Figure 6.5 First part of the object in 3D printing process*

Then, the extrusions components tilted in the XZ plane of the heated bed and reached to the surface of the firstly printed object, started to print the second part in an inclined position as shown in Figure 6.6



*Figure 6.6 The second part in printing process*

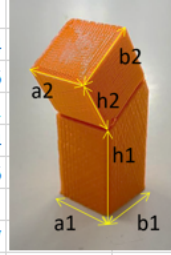
After the whole process is finished, the object is taken away from the heated table for evaluation. In Figure 6.7, the object is presented in different views.



*Figure 6.7 Printed object, (a) and (c) side view, (b) back view, (d) front view, (e) and (f) 3D view*

The printed object is measured and compared with the nominal dimensions. To evaluate its surface quality, a visual inspection is carried out to identify defects. From Table 6.1, it can be seen that despite the average value of the same dimension measured from two sides are different, they remained uniformity. The measured values near the bottom are slightly larger than the average.

	a1 [mm]	b1 [mm]	h1 [mm]		a2 [mm]	b2 [mm]	h2 [mm]
measured	20.29	20.44	27.03		19.9	20.11	16.94
	20.26	20.4	27.11		19.89	20.13	16.96
	20.26	20.42	27.04		19.89	20.21	17.1
	20.23	20.35	27.14		19.94	20.22	17.04
	20.25	20.36	27.13		19.92	20.24	17.06
nominal	20	20	27		20	20	17



*Table 6.1 Object dimension measurement*

An obvious line raised from the surface is observed on the edge, which is believed to be caused by the starting point of each layer. Close to the top edge of the first part, the layers are not well-formed, resulting in a gap between the connecting section of the two parts. The first layer is smaller in size than the rest, and the connection area is the weakest despite acceptable adherence to the stair-shaped surface. However, there is no significant deformation or layer shifting, particularly in the upper part. The object's outer bend has both horizontally and tilted extruder-printed surfaces, but it was mostly kept level during the test, with only minor faults at the connection.

# ***7 CONCLUSION AND FUTURE WORK***

This chapter summarize the results of the present work, the performance of the cobot assisted AM system is examined and the clusions are presented. Furthermore, future work upon on this research is proposed.

## **7.1 Conclusion**

Analysing the performance of the cobot assisted AM system, it is proven that the system would be implemented in the practical cases, expanding the function of the cobot and being integrated into the manufacturing system in a flexible way. Some characteristics of the system are discussed as following to show advantages and inadiquicies:

**Speed and stability:** The production speed of this AM system is limited to 10 mm/s to keep moving stability. While an ordinary desktop FDM printer prints maximum at 60 mm/s using PLA material. When the cobot move more faster, the extruder tip will vibrate when the printing direction is changed, causing a shift of the extrusion path and defects of the object. This phenomenon is more visible at the top of the first printed object in multiplanar printing chapter, when the surface of a layer is getting smaller. The point should be highlighted that the extrusion system is not sturdy enough because of the length and the structure design. The stability is expected to be improved with a redesigned extrusion structure which the components fixed to the cobot flange are reduced. Meanwhile, the moving speed for printing can be increased.

**Path complexity:** The application for path generation in this thesis is originally designed for conventional 3D printers. If the extruder is allowed to rotate or elevate during the process, the generated object would obtain a stronger structure.

The insufficient adherence found in the multiplanar printing is occurred due to the incorrect placement of the first layer of the second part. The starting point failed to be placed on the second printing reference due to an unexpected error of this application. Nevertheless, the issue is solvable if the second reference system is moved certain distance in its Z axis to compensate the error. To generate perfect path for multiplanar 3D AM, a well-designed simulation software is necessary. With low budget and existing resources, it is still possible to produce a quality acceptable product, but more time is needed for parameters adjustment.

**Adaptability in manufacturing process:** The conventional AM machines, both desktop and industrial type, are dedicated to only one process. By using the cobot, the AM process is able to be integrated into a work station for multiple processes. For example, if a cobot is fixed on a mobile robot, the FDM extrusion components can be equipped as an changeable tool. The product generation surface, the hot bed as a reference in this thesis, would be removed to produce an AM product in a non-fixed environment.

## 7.2 Future works

The cobot assisted AM system has more potential to be discovered for future research, especially when the limitations discussed before are overcome. First, a re-designed compatible extrusion system with quick exchange function, thus the reliability and practicability will be strongly improved. The user will be able to select which end effector to use based on his need.

Another interesting field is to design and implement non-planar toolpath for the AM system. Because horizontal layering method is still the most used slicing technique for most of the AM processes. With multiplanar AM process, it's easier to generate complex structured object, but losing the mechanical strength. With non-planar AM, the complexity and strength are believed to be improved, while reducing the staircase effect.

## 8 ANNEX

### 8.1 Stepper motor controlling program:

```
#include <AccelStepper.h>
AccelStepper Extruder(1, 3, 2); // Configuring motor: A stepper motor
controlled by a dedicated driver board, pin3 = step, pin2 = direction.

// Define digital communication pin names.
int AnInPin = A0; // pinA0: receives digital signals from external.
int DigOutPin = 9; // pin9: Sends digital signal to Enable pin on the driver.
int val = 0; // variable to store the read value.

void setup() {
  Serial.begin(9600); // Initialize serial communication at 9600 bits per
second.
  pinMode(AnInPin, INPUT); // Set DigInPin as Input
  pinMode(DigOutPin, OUTPUT); // Set DigOutPin as Output
  Extruder.setMaxSpeed(20000); // Set the maximum speed in steps per second.
  Extruder.setSpeed(50); // Set the initial speed in steps per second.

void loop() {
  val = analogRead(AnInPin); // store the value read by AnInPin
  if(val >= 950) {
    digitalWrite(DigOutPin,LOW); // If high: extruder runs in the speed set
before. Set Enable pin on A4988 driver to low to activate the extruder.
    Extruder.setSpeed(58); // Set the initial speed in steps per second.
    Extruder.runSpeed();
  }
  if(val < 950) {
    Extruder.stop(); // If low: stops the Extruder and sends high signal to
Enable pin on A4988 driver to disable the extruder control.
    digitalWrite(DigOutPin,HIGH);
  }
}
```

## **9 REFERENCE**

- [1] BS EN ISO/ASTM 52900:2021: Additive manufacturing. General principles. Fundamentals and vocabulary, (2022).
- [2] R. Scopigno, P. Cignoni, N. Pietroni, M. Callieri, M. Dellepiane, Digital Fabrication Techniques for Cultural Heritage: A Survey, n.d.
- [3] K. Julia, T. Laura, G. Andreas, Basics of 3D Printing Technology, in: 3D Printing - Understanding Additive Manufacturing, 2nd Edition, Hanser Publishers, 2019: pp. 1–2.
- [4] J. Mazumder, J. Choi, K. Nagarathnam, J.L. Koch, D.W. Hetzner, The direct metal deposition of H13 tool steel for 3-D components, JOM. 49 (1997) 55–60.
- [5] J. Mazumder, Laser-aided direct metal deposition of metals and alloys, in: Laser Additive Manufacturing: Materials, Design, Technologies, and Applications, Elsevier Inc., 2017: pp. 21–53. <https://doi.org/10.1016/B978-0-08-100433-3.00001-4>.
- [6] R. Harik, T. Wuest, Introduction to Advanced Manufacturing, SAE International, Warrendale, 2019.
- [7] S.M. Yusuf, N. Gao, Influence of energy density on metallurgy and properties in metal additive manufacturing, Materials Science and Technology. 33 (2017) 1269–1289. <https://doi.org/10.1080/02670836.2017.1289444>.
- [8] C. Prakash, S. Singh, H. Kopperi, S. Ramakrishna, S.V. Mohan, Comparative job production based life cycle assessment of conventional and additive manufacturing assisted investment casting of aluminium: A case study, J Clean Prod. 289 (2021). <https://doi.org/10.1016/j.jclepro.2020.125164>.
- [9] Molsheim, Bugatti Bolide – what if ..., (2020). <https://www.bugatti.com/media/news/2020/bugatti-bolide-what-if/> (accessed October 12, 2022).
- [10] EROFIO Group prints first part on Concept Laser M Line system | GE Additive, (2021). <https://www.ge.com/additive/story/erofio-group-prints-first-part-concept-laser-m-line-system> (accessed October 12, 2022).
- [11] T. Luomaranta, M. Martinsuo, Additive manufacturing value chain adoption, Journal of Manufacturing Technology Management. 33 (2022) 40–60. <https://doi.org/10.1108/JMTM-07-2021-0250>.
- [12] International Federation of Robotics, (2020). <https://ifr.org/case-studies/complete-human-robot-collaborative-production-line-with-t-skin-in-qisda> (accessed October 12, 2022).
- [13] V. Klingspor, J. Demiris, M. Kaiser, Human-Robot-Communication and Machine Learning Abbr. title: Human-Robot-Communication and ML, n.d.
- [14] BSI Standards Publication Safety of machinery-Relationship with ISO 12100, 2021. [www.gov.uk](http://www.gov.uk).
- [15] BSI Standards Publication Robots and robotic devices-Collaborative robots,



2016.

- [16] British Standards Institution, Robots and robotic devices: safety requirements for industrial robots. Part 1, Robots., n.d.
- [17] M-2000iA/2300 - Fanuc, (n.d.). <https://www.fanuc.eu/it/en/robots/robot-filter-page/m-2000-series/m-2000ia-2300> (accessed October 12, 2022).
- [18] Steve Crowe, Cobot Comparison Tool: Collaborative Robot Buyer's Guide, (n.d.). <https://www.cobottrends.com/cobot-comparison-tool/> (accessed October 12, 2022).
- [19] Industrial robots - average cost | Statista, (2022). <https://www.statista.com/statistics/1120530/average-cost-of-industrial-robots/#statisticContainer> (accessed October 13, 2022).
- [20] National Cobot Awareness Month | Empowering manufacturing, (n.d.). <https://www.universal-robots.com/about-universal-robots/news-centre/universal-robots-makes-january-national-cobot-awareness-month/> (accessed October 13, 2022).
- [21] Robots pave the way for sustainable mobility at ABB, (n.d.). <https://new.abb.com/news/detail/66300/robots-for-sustainable-mobility> (accessed October 13, 2022).
- [22] Collaborative robotic automation | Cobots from Universal Robots, (n.d.). <https://www.universal-robots.com/> (accessed February 22, 2023).
- [23] 3. Post Processors — RoboDK API Documentation, (n.d.). <http://robodk.com//doc/en/PythonAPI/postprocessor.html?highlight=post%20processor> (accessed March 2, 2023).
- [24] GitHub - hudbrog/gCodeViewer: a visual web based gcode analyzer, (n.d.). <https://github.com/hudbrog/gCodeViewer> (accessed March 7, 2023).