Proceedings of the 10th International Conference of Control Systems, and Robotics (CDSR'23)
Ottawa, Canada – June 01 - 03, 2023
Paper No. 216
DOI: 10.11159/cdsr23.216

Development of a Robotic Additive Manufacturing Framework for Fused Deposition Modeling: Technical Considerations

Pooyan Nayyeri[1], Habiba Bougherara[1], Kourosh Zareinia[1]
[1]Department of Mechanical and Industrial Engineering, Toronto Metropolitan University
350 Victoria Street, Toronto, Canada
pnnayyeri@torontomu.ca; habiba.bougherara@torontomu.ca; kourosh.zareinia@torontomu.ca

Abstract - Additive manufacturing, commonly referred to as 3D printing, is a rapidly growing technology that allows for the creation of three-dimensional parts in a fraction of the time required by traditional methods. Conventional 3D printers use either cartesian or delta mechanisms, which are reliable but limited in movement due to the fixed orientation of the tool head. Researchers have been working on using robotic manipulators to create new 3D printing techniques. To accomplish this, they first need to establish a robotic framework for basic 3D printing. This technical brief explains the steps for the implementation of a robotic manipulator for fused deposition modelling (FDM). The proposed approach can help other researchers develop their own robotic 3D printing framework. While many other alternatives can be utilized, the proposed methodology is not intended to be unique or optimized. However, it provides important technical details that can help to expedite the process of establishing new research projects in this field. Additionally, this brief introduces the concept of the "printability index", which can be used to create a map for positioning the build platform in the robot's workspace.

***Keywords*:** additive manufacturing; robotic manufacturing; fused deposition modelling; robotic manipulator; printability

1. Introduction

Additive manufacturing or 3D printing is a relatively new set of techniques for manufacturing. In its early stages, 3D printing was introduced as a rapid prototyping method, but with the improvement of the mechanical properties of the printed parts using different technologies and materials, it is now considered one of the manufacturing methods for end-use parts with practical applications [1]. Fused deposition modeling (FDM) is a material extrusion technique and a subset of additive manufacturing methods that deposits molten polymers to form a 3D part. FDM is the most ubiquitous and prevalent 3D printing method [2].

The components of an FDM 3D printer are the mechanical mechanism, extruder, and build platform. Improving any of these components can result in better quality and properties in the printed part. Conventional mechanisms used in FDM printers include cartesian and delta, which have three degrees of freedom. While these simple mechanisms can maintain positional accuracy throughout their workspace, the end-effector (in this case, the extruder) cannot change its orientation, resulting in planar printing and limiting the part's quality and mechanical properties. To address this issue, mechanisms with higher degrees of freedom, such as industrial robotic arms, can be used to add rotations to the end-effector and produce non-planar printing, which directly improves the surface quality and mechanical properties of the printed part.

The use of a robotic framework for additive manufacturing is known as robotic additive manufacturing (RAM). This offers multi-axis and non-planar printing, which can help build more complex geometries and reduce or eliminate support structures. There have been many innovations in RAM related to material properties and toolpath generation [3]. For example, Bhatt et al. developed a multi-resolution robotic platform using two robotic arms for faster printing with satisfactory surface quality [4]. Similarly, Fry et al. used two robotic arms to increase the total degrees of freedom of the system by holding the build platform with one arm and moving the extruder with another [5]. However, these projects all start with a robotic framework for FDM planar printing, like conventional 3D printers.

A primary challenge in establishing a RAM framework is the location of the build platform relative to the robotic manipulator [6]. The dexterity of robotic manipulators is variable throughout their workspace, in contrast to cartesian mechanisms, so the build platform must be accurately located to maximize robot manipulation. To this end, researchers use the concept of reachability, which can be visualized in a reachability map to determine the placement of tools/equipment in the workspace [7]. There are various approaches to generating a reachability map for robot manipulators, but the fastest and most accurate is the hybrid approach that uses both forward kinematics (FK) and inverse kinematics (IK) of the manipulator [8].

This paper presents the technical considerations of a RAM framework for FDM printing. The authors have developed a robotic framework for 3D printing as a starting point for investigating innovative RAM projects. The paper is structured as follow: Section 2 introduces the components of a RAM framework and their relationship, including the components used in the developed framework and their advantages and disadvantages. Additionally, the concept of "printability" is introduced and used to generate a printability map for the robot. Section 3 explores the software and programming aspects of the framework, from slicing software to custom firmware for synchronizing the robot's movement and extrusion. Section 4 presents the results obtained from the final RAM framework and discusses its achievements. Finally, Section 5 contains the concluding points of the paper.

2. Experimental Setup

The key components of a RAM framework for FDM are a robotic manipulator, an FDM extruder, and a build platform, as depicted in Fig. 1. There are several alternatives available for each component, each with its own advantages and disadvantages. In the subsequent sub-sections, each component will be described and analyzed, along with the specific component utilized in the framework development by the authors.

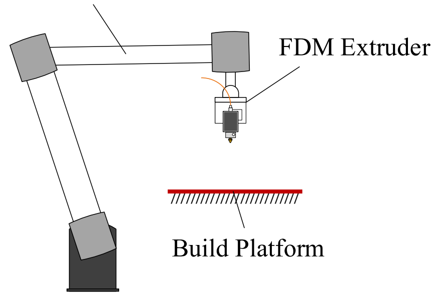

Fig. 1: Schematic of a robotic platform for FDM 3D printing.

2.1. Robotic Arm

The main advantage of using a robotic manipulator for 3D printing is its high manipulability and greater degrees of freedom compared to cartesian or delta mechanisms. Conventional mechanisms, such as cartesian, typically have only three degrees of freedom, limiting their ability to control the orientation of the end-effector. In contrast, industrial manipulators with six or seven degrees of freedom can follow complex trajectories and toolpath orientations, providing more complex 3D layers and toolpaths.

Commercial robotic manipulators, usually serial manipulators, are widely available and come equipped with computers and drivers for controlling the arm, making trajectory tracking of the end-effector easier. Accuracy and repeatability are two key factors that determine the performance of a robotic arm in 3D printing. The reach of the manipulator also plays a crucial role in determining the printing volume of the framework, along with factors such as joint limitations and the installation position of the manipulator relative to the build platform. For this project, the authors used a KUKA KR-10 R1100-2, a 6-axis serial manipulator.

The workspace of the robot refers to the area around the manipulator that can be reached by the end-effector. It is important to understand the workspace to ensure the safety and proper placement of equipment. The workspace can be calculated using Forward Kinematics (FK) of the manipulator. The workspace of the robot is generated by about 1,000,000 random end-effector positions provides an approximation of the robot's workspace.

The reachability of points inside a robot's workspace is an important factor in determining and designing equipment around the manipulator. While all points inside the workspace are reachable by the end-effector, there are different levels of reachability for each point. The reachability of a point inside the workspace refers to the level of control that the manipulator has over that point. Therefore, the workspace of a robot should not be the only consideration when analyzing and placing equipment.

A reachability map visualizes the reachability of the points inside the workspace. There are different methods to generate reachability maps for robotic manipulators, including forward kinematics (FK), inverse kinematics (IK), or a hybrid approach using both FK and IK [9]. In this paper, a hybrid approach inspired by [9] is followed to generate a reachability map for the KUKA robot, which can then be used to determine an appropriate location for the heated bed. The advantage of the hybrid approach is its combination of reduced computation cost and time with maintained accuracy.

The first step in generating the reachability map is to divide the space around the robot into small voxels or bins. This is to identify which voxels are within the robot's workspace. The reachable voxels around the robot, numbering 125,000, are populated with nearly 1 million end-effector positions.

Once all the reachable voxels are calculated, the next step is to implement IK to calculate the reachability index for each voxel. The reachability index is a number that indicates what is the level of reachability for the center point of a voxel. To obtain the reachability index, the IK solver attempts to find a solution for the manipulator to position the end-effector to a pre-defined pose. If there is at least one solution for the pre-defined pose, the reachability index for that voxel is increased by one. This process is repeated for multiple pre-defined orientations for each voxel. After going through all the orientations, the reachability index of the corresponding voxel is obtained, and the algorithm moves on to the next voxel.

The concept of "printability" is introduced in this paper and is similar to reachability. The difference between the two is in the pre-defined desired orientations used in the IK step. Reachability considers all orientations to calculate the reachability index, while printability focuses on the orientations that are more common in the 3D printing process. In the printing process, the tool head is usually located above the build plate or the printed part and pointing downwards. Thus, printability emphasizes orientations with similar configurations. The printability index differs from the reachability index, especially for the points near other objects such as build plate.

In this paper, the printability index is calculated based on 10 orientations of the tool head for each voxel center point. These orientations, which are more commonly used in 3D printing, are visualized on a sphere for better visualization in Fig. 2. Note that there are also 4 additional orientations on the other side of the sphere that are not visible.

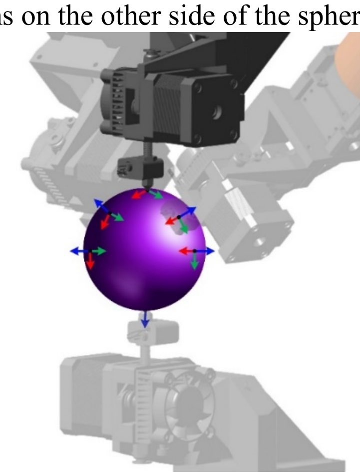

Fig. 2: Pre-defined orientations for printability analysis.

The computed printability map for the KUKA KR10 R1100-2 robot, along with the robot's workspace, is shown in Fig. 3. The highest printability is found at the back of the manipulator. However, regions with a high elevation relative to the robot's base are not optimal as they are prone to collisions with the build plate, which is not considered in the simulations. Hence, regions with a low enough elevation where the robot can approach the build plate from above should be considered. Based on the printability map, the optimal location for the heated bed is in front of the robot and approximately 0.2 m above the table.

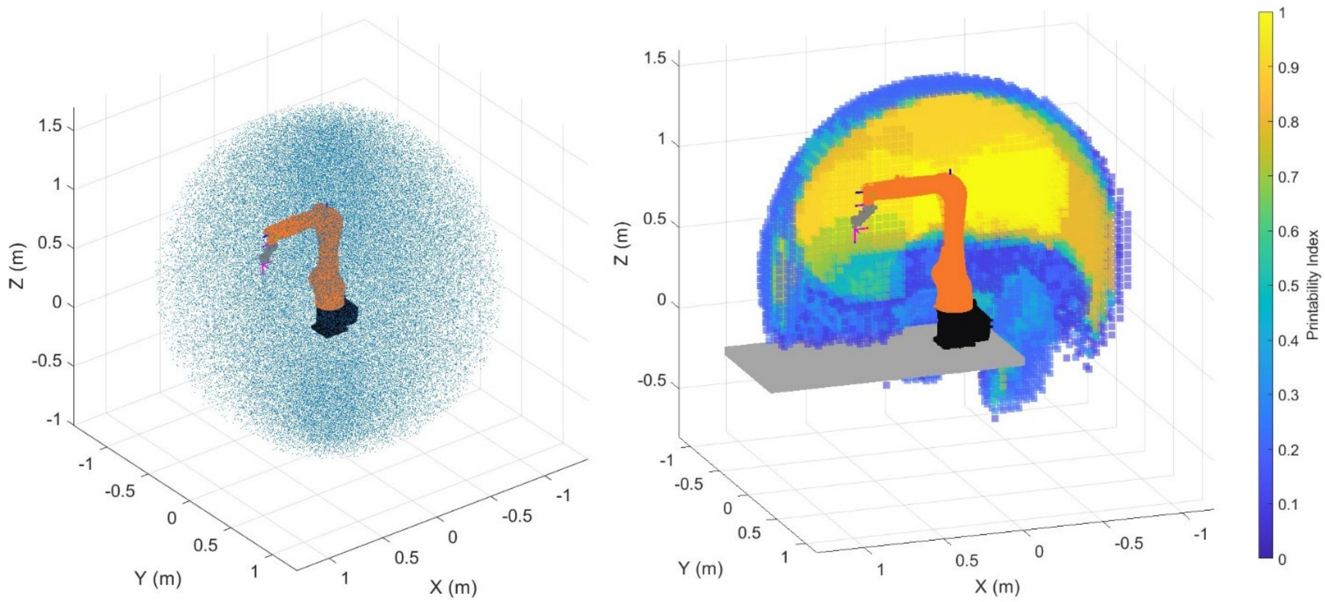

Fig. 3: (Left) Workspace of the robot; (Right) Cross-section of the printability map obtained by the hybrid method along with the robot and the table.

2.2. Extruder

FDM extruders mainly use polymers in filament form as input material to create 3D objects layer by layer. The printing quality and speed mainly depend on the nozzle size, among other effective factors. To develop the RAM framework, a LulzBot single tool head is used. The input filament size is 2.85 mm, and the hardened steel nozzle size is 0.5mm. This extruder can handle flexible filaments as well as tough reinforced polymers such as carbon fiber-reinforced polymer filaments. Also, since the heater is run by a 24VDC power supply, it can reach the desired temperature in a shorter time.

2.3. Heated Bed

A build platform, in simplest form, acts as a substrate to hold 3D-printed parts. Since some polymers need to be deposited on a warm/hot surface to avoid sudden cooling problems, build platforms are often equipped with a heater and a temperature sensor. For the developed framework, a Folgertech heated bed is used. The bed is comprised of a 760 mm by 330 mm aluminum bed covered with a PEI film to increase the adhesion. The bed is equipped with a 700 w 110 VAC silicone heater almost the same size as the aluminum sheet controlled by a solid-state relay and is supported by a spring-suspension system for an easier leveling process.

3. Software and Programming

In conventional 3D printers, a controller board (mainly an Arduino-based board) is responsible for motor control, temperature control, GCode interpretation, and any other tasks related to the printing process. There are many different controller boards for printers such as Arduino Mega (coupled with a RAMPs board) and RAMBo. They have ports for stepper motors, thermistors, fans, heaters, end-stops, memory card readers, LCDs, power, and serial communication (USB). Nevertheless, since a robotic manipulator is used to move the extruder around, the extruder controller board cannot be used to control the robot's axes as the robot has its own controller. In addition, robot controllers are not usually developed to drive stepper motors. Therefore, a controller board is needed to control the extruder and heated bed while a robot controller is needed for robot movements.

In the following subsections, the developed frameworks for the robot controller, extruder controller board, and the synchronization of these two are discussed in detail.

3.1. Robot Control

KUKA uses KUKA Robotic Language (KRL) for its programs. While the smartPAD can be used to develop/modify relatively simple programs, other software should be used for more advanced programs. RoboDK [10] is a powerful software developed for offline programming, online programming, and simulation of industrial robots. The advantage of RoboDK is the built-in functionalities of several manufacturing processes, especially 3D printing. Furthermore, RoboDK utilizes robot drivers to program robots online. Nevertheless, it does not result in smooth and continuous movements of the end-effector since each command is separately sent by a computer to the robot controller via a LAN connection. Thus, online programming is not a practical approach for many projects.

In contrast, offline programming does not require a computer in the loop since all commands are executed by the robot controller. In this approach, a robot-specific program is generated by a computer and transferred to the robot controller. The movement of the manipulator and any peripheral devices (extruder, in this case) are then controlled by the robot controller. The offline programming process is shown in Fig. 4. This approach leads to smooth and continuous movements of the robot's end-effector, which is vital, especially in the 3D printing process.

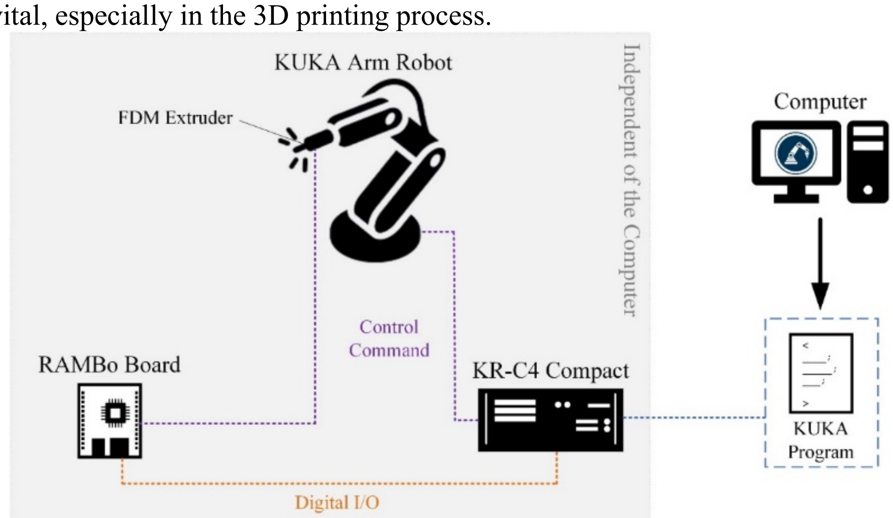

Fig. 4: Offline programming of a robot does not rely on a computer; therefore, leads to continuous and smooth motion of the end-effector.

RoboDK 3D printing function helps the user to configure a 3D printing project from a 3D model, NC file (G-Code), points, or curves. It also can be integrated with Slic3r, an open-source slicing software, to automatically slice 3D models and generate tool paths and instructions. Therefore, RoboDK can play the role of slicing software of a conventional 3D printer for a RAM framework.

3.2. G-Code Generation

G-Code generation can be done using any slicing software as we are interested in planar layers for the generated tool paths. Many commercial and free software programs will create G-Code instructions based on the selected configuration. Since robotic manipulators use their programming languages, a post-processor is needed to translate G-Code instructions to robot-specific programs.

RoboDK has developed post-processor APIs (stands for application programming interface) that convert G-Codes from slicing software to robot programming languages. Using this function, G-Codes can be converted to *.src* files for the KUKA controller. It should be noted that it is possible to run the G-Codes on RoboDK simulations or online programming using robot drivers. Nevertheless, as mentioned before, online programming cannot provide continuous and smooth movement of the extruder and leads to an impractical approach for 3D printing. The difference between offline and online programming is explained in Subsection 3.1.

3.3. Extruder and Heated Bed Control

Generally, 3D printers use firmware to read G-Codes and to enable the user to interact with them. For example, Marlin is a powerful, open-source firmware developed for Arduino-based control boards to enable them with different printing capabilities. The firmware can read G-Codes from its memory and interpret them to 3D-print an object. Alternatively, G-Code commands can be sent to the microcontroller using serial communication.

The problem with developed firmware packages is the inability to run stepper motors with a constant speed and stop it during a pre-defined motion, meaning that the amount of rotation of the extruder motor must be pre-determined when the move command is sent. This will be problematic while sending the extrusion speed from the KUKA controller to the RAMBo board as it cannot send the motion and speed using the provided digital output ports. Therefore, a custom firmware is developed to meet our needs.

The retraction function is another feature added to the firmware and essential for printing a part with discontinuous tool paths. This feature is included in slicing software that automatically adds some instructions to the GCode to retract the filament at the end of a printing path to prevent extrusion during extrusion-less paths.

3.4. Robot-Extruder Synchronization

A satisfactory synchronization between the extrusion of material and the movement of the tool is essential for a successful 3D printing process. Since all the instructions are followed by the robot's controller in the offline programming approach, there is a need to provide a fast and reliable way to synchronize the robot with the extruder.

Synchronization between the robot and the extruder can be done using three different approaches: external axis, analog output, and digital output. The simplest approach is to have an on/off signal to start/stop the extrusion. Using adequate digital output ports, printing speed can be converted into digital bits at the robot controller and then interpreted by the extruder controller. In all cases, since extruders mainly use stepper motors for the filament feed mechanism, there needs to be some intermediate device to interpret the signals from the robot controller and drive the stepper motor accordingly. In addition, the DC voltage of the robot controller needs to be compatible with the extruder control board. This might need an extra component such as a logic level converter to transmit signals.

The developed framework is equipped with a KUKA KR-C4 Compact controller that includes a digital output module (an EL2809 Beckhoff 16-channel 24 VDC digital output module) that offers 16 digital output pins. Based on conventional slicing software programs, the printing speed is usually defined between 15 mm/s to 90 mm/s. Considering the unloading process of the filament which requires the stepper motor to move in the reverse direction, it can be assumed that we need almost 200 states to fully define the speed and direction of the extrusion. As a result, at least 7 bits are needed to encode the printing speed and one bit to define the direction of motion and transmit it from the robot controller to the control board of the extruder. In other words, 8 digital outputs have been used to command the extruder controller using the robot controller.

Since the KUKA program generated by RoboDK needs to contain information on how to enable/disable digital output ports to define the extrusion speed and direction at each moment, a post-processing step is required. For post-processing, a script reads every line of the KUKA program and if the line defines a new printing speed, converts it to proper digital output signals.

3.5. Heated Bed Leveling

Leveling the heated bed is a regular procedure in many additive manufacturing methods. Accurate leveling of the heated bed can result in satisfactory first-layer adhesion and quality, thus a flawless deposition throughout the printing process. In FDM 3D printing using cartesian printers, the heated bed calibration is done to ensure that the bed plane is precisely parallel to the XY plane of the mechanism. However, in robotic printing, we need to make sure that the virtual heated bed in the software environment precisely complies with the one in the actual system. In other words, different points on the virtual bed must precisely represent the same points on the actual bed.

If the virtual and actual heated beds are considered as two planes, the calibration can be defined as moving one of the planes (the actual bed) to coincide with the other one (the virtual bed). This constraint can also be defined as having 3 pairs of coinciding points for the two planes. Therefore, the first goal in leveling the bed is to have 3 pairs of points located at the same locations. Next, off-plane points need to be adjusted to get on-plane since the actual bed is not a perfectly flat surface and has deflections. In the case of the developed framework, the heated bed has 4 screws at each corner and 4 screws at the larger edges of the bed. This means that we need to calibrate the corresponding 8 points on the virtual bed in the RoboDK environment to be located at the same locations in the actual system.

3.6. Extrusion Rate Calibration

Synchronization between the extrusion and robot movements is ensured by developing reliable communication between the KUKA controller and RAMBo board. While the stepper motor library used in the developed custom firmware for the RAMBo board runs the motor with a specified speed in revolution/min or RPM, the printing speed specified in G-Codes is calculated in mm/sec (or mm/min in some slicing software programs). Therefore, a conversion needs to be carried out once the printing speed is read by the RAMBo board through the digital input pins. This conversion maps the printing speed in mm/sec to the proper stepper motor speed in RPM. Therefore, the next step of calibration is to find a proper coefficient (that is called extrusion multiplier constant) to be multiplied by the printing speed in the G-Code to result in a suitable stepper motor RPM.

The extrusion multiplier constant can be calculated using analytical approaches by considering the gear ratio of the extruder and filament diameter. Another method is a trial-and-error approach to find the right value for the constant. Regardless of the approach, practical fine-tuning might be needed for the best results.

By printing and analyzing a sample part, the extrusion multiplier constant can be tuned. Since under- and over-extrusion can be directly detected by visual inspection of the printed part, the right value can be obtained by changing the extrusion multiplier constant during the print.

4. Results and Discussion

After completing all the steps explained in previous sections, the framework is ready to 3D print similar to conventional 3D printers. Several parts with different characteristics, sizes, and geometries are printed by the robotic framework and are shown in Fig. 5. Subfigures (a) and (b) show the ability of the framework to manufacture large objects in vase (shell) mode with intricate details. Subfigure (c) shows a printed part with overhang and bridge features.

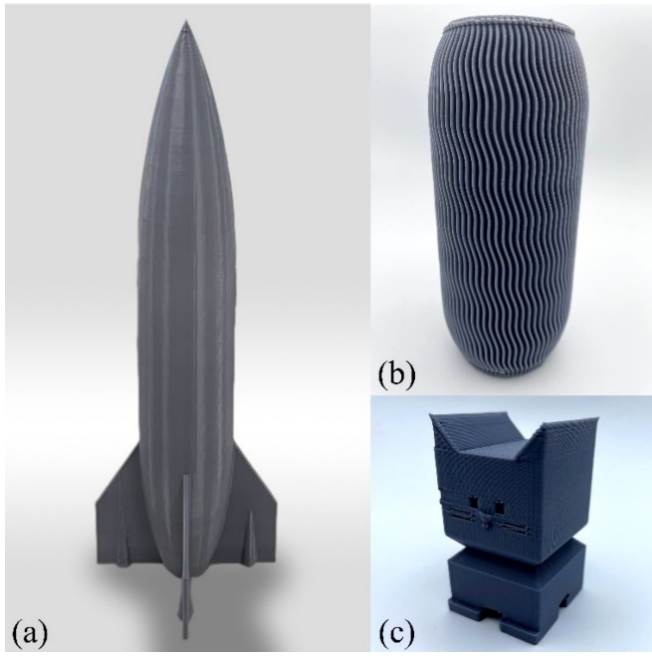

Fig. 5: 3D printed parts with the developed robotic platform to showcase its capability of printing shell (vase) structures (a) and (b) and overhang structures (c).

5. Conclusion

This paper presents a robotic framework for 3D printing, focusing on fused deposition modeling (FDM), which is the most widely used 3D printing technology. The authors describe the components of a RAM framework, including the mechanical mechanism, extruder, and build platform, and discuss their relationship. The authors also introduce the concept of "printability" and use it to generate a printability map to determine the best placement of tools and equipment in the workspace. The paper explores the software and programming aspects of the framework, including slicing software and custom firmware for synchronizing the robot's movement and extrusion. The results of the final RAM framework are presented and discussed, including the advantages and limitations of the framework.

References

[1] S. Wickramasinghe, T. Do, and P. Tran, "FDM-Based 3D Printing of Polymer and Associated Composite: A Review on Mechanical Properties, Defects and Treatments," *Polymers 2020, Vol. 12, Page 1529*, vol. 12, no. 7, p. 1529, Jul. 2020, doi: 10.3390/POLYM12071529.

[2] I. Karakurt and L. Lin, "3D printing technologies: techniques, materials, and post-processing," *Curr Opin Chem Eng*, vol. 28, pp. 134–143, 2020, doi: 10.1016/j.coche.2020.04.001.

[3] P. Nayyeri, K. Zareinia, and H. Bougherara, "Planar and nonplanar slicing algorithms for fused deposition modeling technology: a critical review," *The International Journal of Advanced Manufacturing Technology 2021 119:5*, vol. 119, no. 5, pp. 2785–2810, Jan. 2022, doi: 10.1007/S00170-021-08347-X.

[4] P. M. Bhatt *et al.*, "A robotic cell for multi-resolution additive manufacturing," *Proc IEEE Int Conf Robot Autom*, vol. 2019-May, pp. 2800–2807, 2019, doi: 10.1109/ICRA.2019.8793730.

[5] N. R. Fry, R. C. Richardson, and J. H. Boyle, "Robotic additive manufacturing system for dynamic build orientations," *Rapid Prototyp J*, vol. 26, no. 4, pp. 659–667, 2020, doi: 10.1108/RPJ-09-2019-0243.

[6] J. (James) Yang, W. Yu, J. Kim, and K. Abdel-Malek, “On the placement of open-loop robotic manipulators for reachability,” *Mech Mach Theory*, vol. 44, no. 4, pp. 671–684, Apr. 2009, doi: 10.1016/J.MECHMACHTHEORY.2008.05.004.

[7] J. Dong and J. C. Trinkle, “Orientation-based reachability map for robot base placement,” *IEEE International Conference on Intelligent Robots and Systems*, vol. 2015-December, pp. 1488–1493, Dec. 2015, doi: 10.1109/IROS.2015.7353564.

[8] O. Porges, R. Lampariello, J. Artigas, A. Wedler, C. Borst, and M. Roa, “Reachability and Dexterity: Analysis and Applications for Space Robotics,” *undefined*, 2015.

[9] A. Makhal and A. K. Goins, “Reuleaux: Robot Base Placement by Reachability Analysis,” *Proceedings - 2nd IEEE International Conference on Robotic Computing, IRC 2018*, vol. 2018-January, pp. 137–142, Oct. 2017, doi: 10.48550/arxiv.1710.01328.

[10] “Simulator for industrial robots and offline programming - RoboDK.” https://robodk.com/index (accessed Mar. 09, 2023).