

A 3-Dimensional Printing System Using an Industrial Robotic Arm

Quan Khanh Luu*¹, Hung Manh La², and Van Anh Ho¹

Abstract—This paper describes the development of a three-dimensional (3D) printing system that integrates a six-degree-of-freedom industrial robot into a fused deposition modeling process. By using the robot-based 3D printing system, printing on inclined planes became possible, which cannot be achieved by a conventional 3D printer. Moreover, the robotic 3D printing is supposed to achieve faster and smoother motion compared to its counterpart under the same temporal settings, thanks to a knowledge-based strategy to re-plan printing trajectories from a set of G-commands. The accurate execution of the printing trajectories and other necessary components for the printing process (for example, an extruder) are regulated by the robot operating system (ROS). The efficiency of the printing system was evaluated by 3D printing a couple of simple 3D models using a six-axis Denso robot. The preliminary results revealed great potential for rapid prototyping and printing in close contact with humans, especially in the field of interactive manufacturing, or human-robot collaboration.

I. INTRODUCTION

A. Fused Deposition Modeling

Fused Deposition Modeling (FDM) is the most widely used process in 3D (three dimensional) printing, by which 3D objects of interest are fabricated by depositing melted materials layer-by-layer through a hot extruder. The popular 3D printers that run on the FDM process often have a gantry-style design, in which the extruder is controlled to move linearly on a horizontal plane in two-dimensional space while a print bed is displaced vertically to adjust the height of layer. Nowadays, this kind of 3D printers has been adopted in a variety of applications, ranging from aerospace, construction even to medicine and biomolecules [1]. Despite being seen many benefits from the 3D printers with FDM, there are remaining drawbacks. The most significant limitation of conventional 3D printers is that their movements are restricted on the 2D horizontal plane, consequently leading to the necessity of supporting materials as printing objects composed of overhanging elements [2]. Also, there have been many developments in 3D printers embedded with multiple nozzles that can extrude a range of different materials, either to speed up the process or to form desired physical properties of printed objects [3]. However, scalability is still a noticeable problem of the conventional 3D printer, which means a group of 3D printers is probably impossible to work collaboratively.

¹Luu and Ho are with the School of Materials Science, Japan Advanced Institute of Science and Technology (JAIST), 1-1 Asahidai, Nomi, Ishikawa, 923-1211 Japan. Email: quan-luu@jaist.ac.jp, van-ho@jaist.ac.jp. This research was partly supported by Japan Society for the Promotion of Science KAKENHI under Grant 18H01406.

²La is with the Advanced Robotics and Automation (ARA) Laboratory, University of Nevada, Reno, USA hla@unr.edu

*Corresponding author.

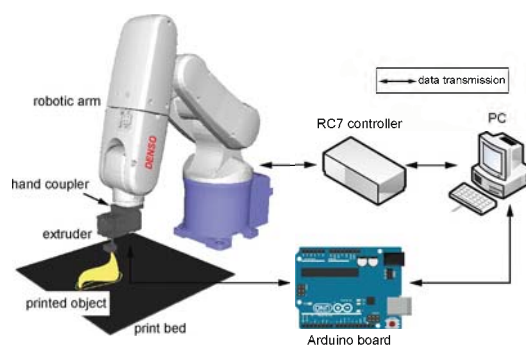


Fig. 1. The overview of main hardware components making up the robot-based 3D printing system.

B. Integration of robotic arm and FDM

To transcend the technical limitations of the conventional 3D printers, research in recent years has been focused on the development of the 3D printing system that integrates the FDM process into a robotic arm [4]. In this proposed system, the movements of the extruder are regulated by the joint motion of the robot arm, and thus could freely deposit materials in an open three-dimensional space [5]. The first notable point of the robot-based 3D printing is that the decrease in fabrication time of large and complex objects has been achieved by employing a team of printing robots performing the task collaboratively [6], or more recently, along with a moving base, the robot arm could conduct the printing task while moving, which enables a large-scale 3D printing system [7]. As a result, the problem of the scalability present in the conventional 3D printers has been partly resolved. Another impressive performance of the robot-based printing system is that supportless fabrication has been realized by adopting a relatively complex framework consisting of two industrial robots, in which one robot performs printing as usual while its collaborator is intermittently or continuously changing the build plate orientation to eliminate or diminish the usage of support structures [8], [9], [10]. Furthermore, in order to minimize the need for support, multi-directional 3D printing is also a new practical method of extrusion-based printing enabled by the use of robot arms [11], [12], in this system algorithms were developed to generate tool paths that regulate the robot to deposit materials on the building objects from multiple directions.

C. Contribution

Following the light of the previous works, this research introduces a 3D printing system using a 6-DoF (degrees of freedom) robotic arm, which could complete the task in a

short time, but still retains the quality of printed objects at an acceptable level. In addition, thanks to the high orientation of the robotic arm, 3D printing on *inclined planes* is entirely possible. With the above key ideas in mind, our study made contributions to the following points:

- 1) The knowledge-based strategy for generation of smooth printing trajectory, and generated trajectory as a function of time is validated based on a robot kinematics simulation.
- 2) The ability of 3D printing on various planes at different angles from the horizontal surface was thoroughly evaluated. This examination are expected to lay a groundwork for *multi-plane* 3D printing, leaning towards forming complex objects without the need of support structures, or printing on rough surfaces in building and construction.

II. REVIEW OF ROBOT KINEMATICS

For ease of following, in this section, necessary equations for generation of velocity and trajectory of a typical 6-DoF industrial robot are briefly presented.

A. Inverse Velocity Kinematics

Let $\mathbf{q} \in \mathbb{R}^6$ be the six joint coordinates of the 6-DoF robot arm, and $\mathbf{x} \in \mathbb{R}^m$ is a minimal representation of the end-effector configuration in m -dimensional task space. In this case, the forward kinematics is a direct map of $\mathbf{x}(t) = f(\mathbf{q}(t))$. Consequently, the velocity kinematics can be derived as

$$\dot{\mathbf{x}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}, \quad (1)$$

where $\mathbf{J}(\mathbf{q}) = \partial f(\mathbf{q})/\partial \mathbf{q} \in \mathbb{R}^{m \times 6}$ is the so-called Jacobian matrix, while $\dot{\mathbf{q}}$ and $\dot{\mathbf{x}}$ are the joint velocity and the corresponding end-effector velocity of the robot, respectively. For a redundant task like 3D printing, *i.e.*, $m < 6$, the robot has more joints than the minimum actuators required to realize the end-effector configurations needed for completing a specific task. In the redundant case the Jacobian matrix is definitely not square, thus the solution of joint velocity can be obtained as

$$\dot{\mathbf{q}}^* = \mathbf{J}^\dagger(\mathbf{q})\dot{\mathbf{x}}, \quad (2)$$

where $\mathbf{J}^\dagger(\mathbf{q}) \in \mathbb{R}^{6 \times m}$ ($6 > m$) is the Moore-Penrose pseudoinverse of the task Jacobian [13]. Solving for the joint velocity in Eq. (1) using the pseudoinverse, we can find out the $\dot{\mathbf{q}}^*$ solution (Eq. (2)) such that $\|\dot{\mathbf{q}}^*\| \leq \|\dot{\mathbf{q}}\|$, where $\dot{\mathbf{q}}$ is any solution satisfying $\mathbf{J}\dot{\mathbf{q}} = \dot{\mathbf{x}}$. In other words, the $\dot{\mathbf{q}}^*$ solution minimizes the joint velocity that fulfills the desired end-effector velocity of the robot, at a given \mathbf{q} configuration.

B. Trajectory Planning with Trapezoidal Time Scaling

The problem of trajectory generation can be decoupled into one issue of finding a pure geometric path achieved by a sequence of robot configurations, and another issue is to specify how those configurations should track the path in the course of time (*i.e.*, time scaling) [14]. In 3D printing, while the geometric path followed by the robot configurations is strictly conformed to the shape of the printed objects, we

have freedom to define a smooth function of time that the robot could efficiently move along the specified path, and still respects to its kinematic limits on joint velocities, or accelerations. In this section, we briefly review the trapezoidal time scaling that has mainly been adopted to generation of a smooth printing trajectory for our 3D printing robot.

A typical trapezoidal time scaling $s(t)$ for point-to-point motion consists of three phases: a constant acceleration $\ddot{s} = a$ over time t_a , followed by a constant velocity $\dot{s} = v$ over time $T - 2t_a$ (*i.e.*, coast phase), and finally coming to rest through a constant deceleration $\ddot{s} = -a$ over time t_a . If $v^2/a \leq 1$, the robot would follow precisely the motion profile of $s(t), \dot{s}(t), \ddot{s}(t)$ [Fig. 2-condition (1)] characterized by the trapezoidal time scaling. In this specific case, as the particular motion is specified by v and a , the total time to complete the typical three-stage trapezoidal profile is computed as [14]

$$T = \frac{a + v^2}{va}. \quad (3)$$

Also, the motion profile as a function of time t during the three stages can be described as follows [14]

Phase 1: $0 \leq t \leq t_a$ (constant acceleration)

$$\ddot{s}(t) = a, \quad \dot{s}(t) = at, \quad s(t) = \frac{1}{2}at^2. \quad (4)$$

Phase 2: $t_a < t \leq T - t_a$ (constant velocity)

$$\ddot{s}(t) = 0, \quad \dot{s}(t) = v, \quad s(t) = vt - \frac{v^2}{2a}. \quad (5)$$

Phase 3: $T - t_a < t \leq T$ (constant deceleration)

$$\ddot{s}(t) = -a, \quad \dot{s}(t) = a(T - t), \quad s(t) = \frac{2avT - 2v^2 - a^2(t - T)^2}{2a}. \quad (6)$$

In another notable case of $v^2/a > 1$, the robot would never be accelerated to the velocity v during the entire motion [14]. That is, the three-stage accelerate-coast-decelerate motion becomes a "bang-bang" motion which characterized by accelerate-decelerate phases [Fig. 2-condition (2)].

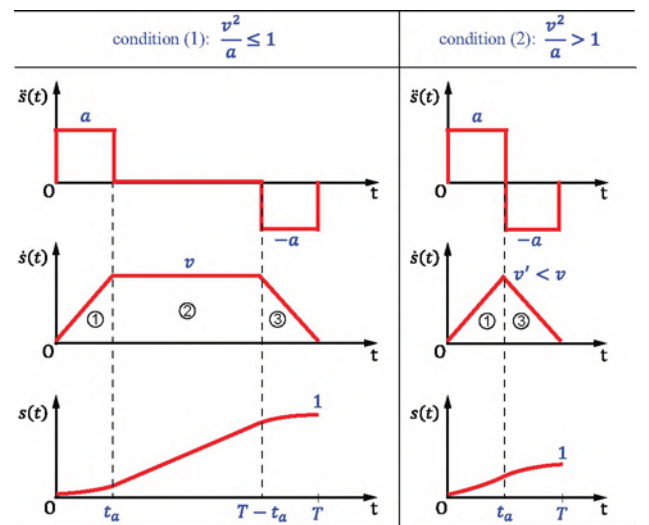


Fig. 2. The scaling time of the trapezoidal and triangular motion profiles, respectively corresponding to the condition (1) and (2).

III. 3D PRINTING BY AN INDUSTRIAL ROBOT

This section firstly presents the whole process of integration for the robot-based 3D printing system. After that, we discuss the underlying planning scheme for printing trajectory based on a set of rules, which is then verified by kinematics simulation. At last in this section, a methodology is proposed for evaluating the capability of 3D printing on inclined planes by using the robot system.

A. System Integration

1) *Hardware*: Fig. 1 depicts the key hardware devices that constitutes the 3D printing system employing the 6-DoF robotic arm (VP6242 Denso robot). In this robot system, the printing extruder (Geeetech MK8 extruder) plays a role of the robot end-effector, and the tip of nozzle depositing polylactic acid (PLA) material on a fixed print bed (Geeetech superplate), is actuated by the motion of robot joints to follow the printing path in the task space. The extruder consisting of one stepper motor, heating element, and temperature sensor, is attached to the tool mounting face of the robot through a 3D-printed coupler. Overall, the whole system is controlled by a personal desktop (PC) running the robot operating system (ROS). Specifically, the control/feedback signals communicating between the robot and ROS are handled through the Denso RC7 controller. On the other side, ROS regulates the speed of filament extrusion and nozzle temperature through an Arduino board.

2) *Software*: The pipeline of information in the robot-based 3D printing system is presented in the Fig. 3. At first, a 3D digital model is imported into a slicer software (open source Slic3r) in which parameters for 3D printing (printing speed, layer height, infill percent, and so on) can be set and then the corresponding G-code file is exported. The G-code command at the line i contains a destination point $p_i = [x, y, z]^T$, the speed v_i , and the amount of extruded filament e_i . After passing the G-code file to a parsing module, all sets of the destination points \mathbf{p} (*i.e.*, print points), speeds \mathbf{v} , and extrusion amount \mathbf{e} are extracted. Next, the inclined angle is specified at $0 < \alpha < 90^\circ$ if printing on an inclined surface are intended (see Section III-D). The output of this module includes the transformed print points and the extruder orientation, which will be explained in Section III-D. The two kinds of data are taken as input for the module of knowledge-based planning (Section III-B), following which the generated waypoints (*i.e.*, a sequence of joint positions and joint velocities) in terms of time would be verified by the kinematics simulation (Section III-C). The waypoints that do not conflict over the robot kinematic constraints would be the referenced trajectory as a function of time for the ROS controller, based on which and the joint state feedback, the joint commands are produced to precisely regulate the robot joint motion so that the nozzle could trace accurately the printing trajectory. Besides that, the action of extruding material by the extruder can be exactly synchronized with the robot motion by controlling the speed of filament extrusion based on the extrusion amounts \mathbf{e} and the time series of the referenced robot waypoints.

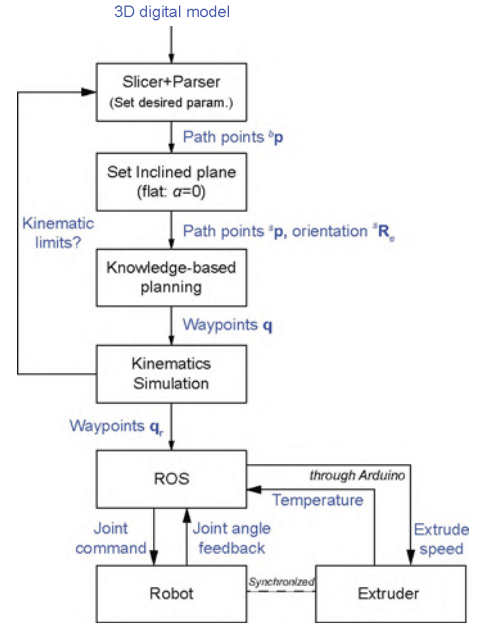


Fig. 3. The pipeline of the robot-based 3D printing system.

B. Knowledge-based Trajectory Planning

The smooth trajectory generation is basically a refinement from a set of point-to-point motion commands (G-code) exported from a slicer software. Considering the two conditions of trapezoidal time scaling mentioned in section II-C, the set of G-commands can be divided into two kinds of paths, or segments. The first kind is long straight segments that satisfy $v^2/a \leq 1$ [see condition (1)], and the other is rounded curves made up of a series of short straight segments, each of which meets the condition of $v^2/a > 1$ [see condition (2)], where the velocity v is specified in the G-command lines, while the acceleration a is a user-defined parameter. By following strictly the G-commands line-by-line, printing over the rounded curves would lead to jerky movements, which results from executing repeatedly the accelerate-decelerate motion profile.

For smoothing the jerky motion along the round-turning path, the knowledge-based trajectory planning consecutively parses over the lines of G-commands to predict the set of curved segments C and then make planning over each of the constituent curve $c_j \in C$. The prediction could be realized by comparing the physical attributes of desired motion at one instance i to the condition (2). In addition, the directional relation between the current segment s_i and the next segment s_{i+1} serves as a strong indication of a curve path. Given the directional unit vectors of s_i and s_{i+1} as u_i and u_{i+1} , respectively, we define the cosines of the angle between the two unit vectors as the directional relation of the two segments, $\cos(\phi) = u_i \cdot u_{i+1}$. Therefore, the two adjacent segments are said to be the element of one specific curve (*i.e.*, $s_i, s_{i+1} \in c_j$), if $\cos(\phi) \geq \epsilon$, where ϵ is a predefined threshold to assure that the curve is not too sharp (*e.g.*, $\cos(\phi) \geq 0.5$, then $\phi \leq 60^\circ$). Regarding the time scalings for a curved segment c_j , there are also three stages similar to the typical

trapezoidal motion profile, including acceleration, constant velocity, and deceleration phases, called phase (1), phase (2), phase (3), respectively. Assuming that the sequence of segments in one specific curve c_j starts accelerating at s_0 and then comes to rest at s_n , then $c_j = \{s_0, s_1, \dots, s_{n-1}, s_n\}$. The initial segment s_0 could be parameterized with the time scaling function of phase (1), followed by the phase (2) applying to the sequence of paths from s_1 to s_{n-1} , and finally the s_n segment can be described as a function of time in phase (3). The functions concerning to the three-phase time scalings can be found in Eqs. (4)-(6), in turn. Table I describes concisely the set of rules that define the time-scaling phases for the knowledge-based trajectory planning, based on which the knowledge-based planning requires only the information at three consecutive segments, that is, s_{i-1} , s_i and s_{i+1} . With reference to the variables in the table, we define $\zeta_k = v_k^2/a$, where v_k is the target velocity over the segment s_k with $k = \{i-1, i, i+1\}$, and a is the user-specified acceleration on every single path.

TABLE I
SET OF RULES FOR KNOWLEDGE-BASED PLANNING

Rule	Time Scaling
(1) if: $\zeta_{i-1} < 1 \& \zeta_i > 1 \& \zeta_{i+1} > 1$ if: $\cos(\phi) < \epsilon$ else: $\cos(\phi) > \epsilon$	Phase (1)+(3) Phase (1)
(2) if: $\zeta_{i-1} > 1 \& \zeta_i > 1 \& \zeta_{i+1} > 1$ if: $\cos(\phi) > \epsilon$ else: $\cos(\phi) < \epsilon$	Phase (2) Phase (3)
(3) if: $\zeta_{i-1} > 1 \& \zeta_i > 1 \& \zeta_{i+1} < 1$	Phase (3)

C. Kinematics Simulation

The kinematics simulation validates the knowledge-based planning strategy discussed in the previous section by allowing the robot to virtually print a hollow square box (Fig. 4a). The nozzle of the robot is controlled to trace along two kinds of trajectory. The first one is the trajectory that strictly follows every G-command line, and the second is the one generated by adopting the knowledge-based planning. To clearly recognize the difference between the two planning schemes, their velocity profiles are analyzed as the robot is moving over the path consisting of four round-turning segments which is the first outer layer of the printing model (Fig. 4b). The effect of the knowledge-based planning on the printing trajectory is notably presented with the smooth task-space velocity shown in Fig. 5b-1 in comparison with the jerky motion profile produced by the normal planning method (Fig. 5a-1). As a result, by adopting the novel planning method the joint motion of the robot (Fig. 5b-2) would become smoother than that generated by its counterpart (Fig. 5a-2). In addition to the improvement of the motion, there is also a reduction in printing time. As clearly shown in Figs. 5a-5b, the printing time in the smooth motion (approximately 6 s) is almost half of that in the jerky motion (about 11 s).

Moreover, despite having the same maximum end-effector velocity inputted, which reaches at 0.03 m/s with regard to \hat{x} and \hat{y} axes in the task space (see Figs. 5a-1, 5b-1), the maximum speed of the robot joint produced by the knowledge-based planning is lower than that generated by the normal planning scheme, respectively at around 0.1 and 0.2 rad/s (see Figs. 5a-2, 5b-2). This is achieved by applying the concept of pseudoinverse of the task Jacobian (Section II-A) for this particular redundant 3D printing task. By using this method, moreover, in some cases the singularities during the printing task could be resolved.

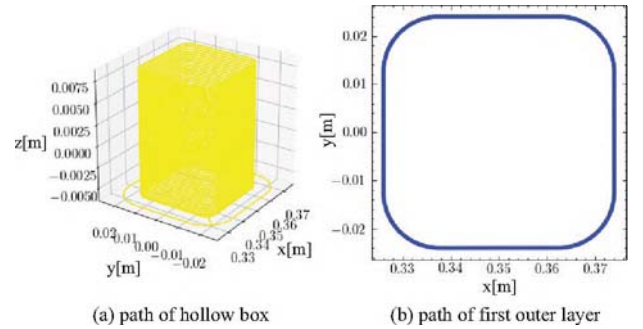


Fig. 4. Figure (a) presents 3-coordinated path of the hollow box. While (b) displays the path of the first outer layer, projected on a 2D plane.

D. 3D Printing on Inclined Planes

This section presents a methodology to investigate the capability of 3D printing on an inclined surface forming an angle α with the horizontal plane. The evaluation of printing on inclined planes at a range of different angles could be realized based on the set-up depicted in Fig. 6a. Specifically, the relative height h could be tuned by the adjustable bolts to meet the desired printing angle α which is then computed as

$$\alpha = \arctan \frac{h}{d}, \quad (7)$$

where d is the designated distance between the fixed and adjustable bolts (see Fig. 6a). The printing path ${}^s\mathbf{p}$ on the corresponding inclined surface could be obtained by the following equations. Let firstly define the coordinate frames assigned to the robot base, nozzle tip, and print bed as $\{s\}$, $\{e\}$, and $\{b\}$ (Fig. 6a), respectively, and the transformation matrix of $\{b\}$ relative to $\{s\}$ as ${}^s\mathbf{T}_b$. Also, a transformation operation \mathbf{T}_x is defined as

$$\mathbf{T}_x = \left[\begin{array}{c|c} \mathbf{R}(\hat{x}, \alpha) & 0 \\ \hline 0 & 1 \end{array} \right], \quad (8)$$

where $\mathbf{R}(\hat{x}, \alpha)$ is the rotation operation revolving around the \hat{x} axis of the base frame at the angle of α (see [14]). Thus, tilting the print bed with an angle of α leads to a new transformation matrix of $\{b\}$ relative to $\{s\}$, ${}^s\mathbf{T}'_b = {}^s\mathbf{T}_b \cdot \mathbf{T}_x$. Consequently, a set of path points measured in the bed frame ${}^b\mathbf{p}$, after being moved to the inclined surface can be described with regard to base frame $\{s\}$ as

$${}^s\mathbf{p} = {}^s\mathbf{T}'_b \cdot {}^b\mathbf{p}. \quad (9)$$

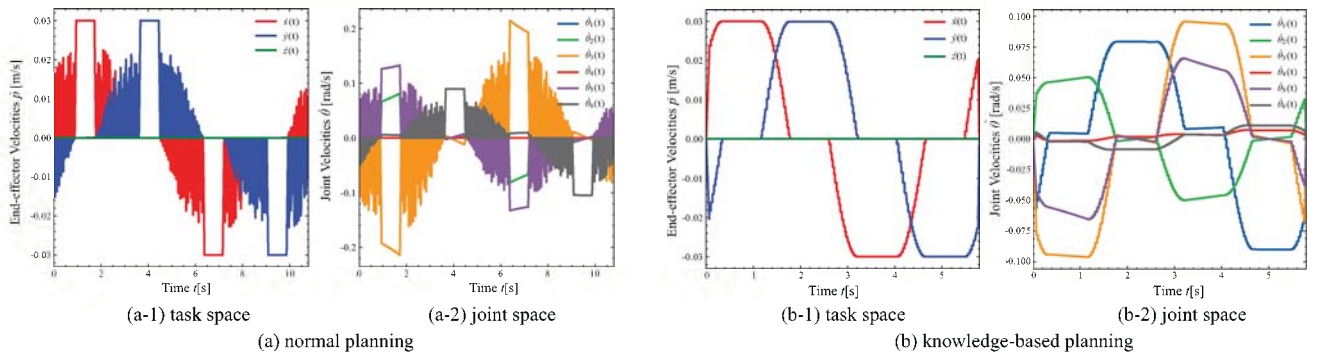


Fig. 5. The robot velocities as a function of time are planned by the normal (a) and knowledge-based (b) planning schemes for moving on the first outer layer. Figures (a-1), (b-1) and (a-2), (b-2) are the velocities in the task space and joint space, respectively.

In terms of the critical issue of orienting the extruder to follow the printing path on the inclined surface. The orientation of the nozzle tip relative to the base frame can be defined by the rotation matrix

$${}^s\mathbf{R}_e = \mathbf{R}(\hat{y}, \pi) \cdot \mathbf{R}(\hat{x}, \alpha), \quad (10)$$

where $\mathbf{R}(\hat{y}, \pi)$ is the rotation operation rotating around the \hat{y} axis of the base frame through 180 degrees [14].

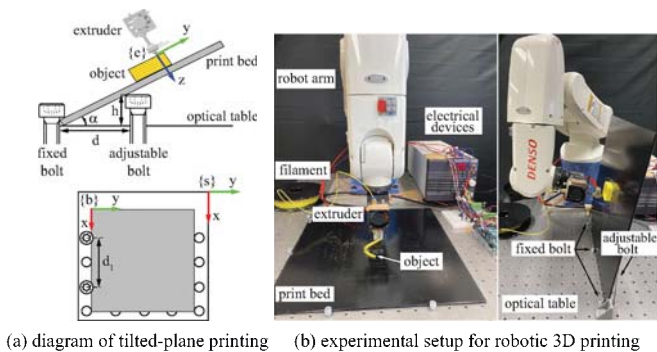


Fig. 6. (a) depicts schematic diagram of 3D printing on an inclined plane. And (b) shows the working environment of the robot-based 3D printing system, printing on a flat surface (b-left), and an inclined plane (b-right).

IV. EXPERIMENT AND RESULTS

A. Experiment

The experimental setup for evaluating the technical capabilities of the robot-based 3D printing system is displayed in Fig. 6b. Overall, the key components of the system have been listed in Section III-A, except for a filament roller (Zotrax Z-PLA filament), and additional electrical devices for powering the extruder. The temperature of the extruder was regulated by an on-off controller within a range of 230°C to 235°C. And for enhancing the adhesion of the printing objects to the print plate, a glue stick was used instead of a heated bed. The abilities of the system were showcased by two experiments: (1) printing a rather complex wave-shaped spoon holder on a flat surface (Fig. 7a-1), and (2) printing a simple hollow square box either on inclined planes with a variety of different angles (10°, 45°, 60°, and 80°), or on a flat surface (Fig. 7a-2). In terms of the former case, the effect

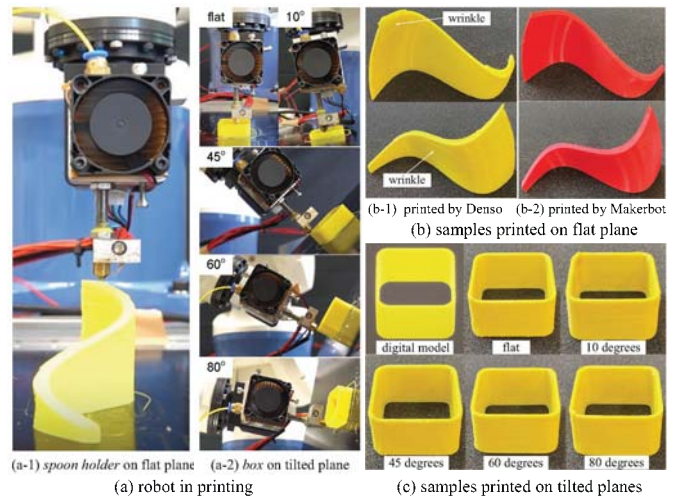


Fig. 7. Fig. (a) captures the robot while printing on a flat surface (a-1) and a variety of tilted planes (a-2). Fig. (b) presents the external appearance of samples printed by the Denso robot (b-1) and Makerbot printer (b-2). While Fig. (c) shows up the digital model and samples printed by the Denso robot on a wide range of tilted planes.

of a smooth trajectory on reduction in printing time was verified by comparing that to a commercial 3D printer, and the trade-off between the surface quality of printed objects and the printing speed was considered as well. The latter case demonstrates the ability of printing on inclined planes, and how the quality of the object surfaces exhibits, afterwards.

B. Results

1) *Result of experiment (1)*: The measurement of printing time was manually recorded using the Stopwatch application of Windows OS. As for fabricating the spoon holder without infill, the time for printing by the robot was around 33 minutes, which was shortened nearly 1.5 times when printing by a Makerbot 3D printer (Replicator Z18) under the same printing speeds and layer height (0.3mm) settings. The specified printing speeds and the comparison of the printing time between the two machines are summarized in Table II. However, the surface quality of the object printed by the Denso robot (yellow sample, Fig. 7b-1) appeared to be not as smooth as that fabricated by the Makerbot 3D printer (red

sample, Fig. 7b-2), but it remains acceptable. Specifically, there are some wrinkled areas on the surface of the yellow sample, while the red one shows a better appearance. The differences in the quality of the samples might be partly accounted for by a more sophisticated heating and cooling process regulated inside the Makerbot 3D printer, which has not been adopted for the robotic 3D printing system.

TABLE II
THE COMPARISON OF PRINTING TIME

Machine	Object Description	Printing Speed (mm/s)	Printing Time (minutes)
Denso Robot	Hollow box $35 \times 35 \times 20\text{mm}^3$	20 (Max. 40)	~ 15
	Spoon holder (No infill)	40 (Max. 100)	~ 33
3D Printer	Spoon holder (No infill)	40 (Max. 100)	~ 45

2) *Result of experiment (2)*: 3D printing on the inclined planes, in general, yielded favorable results. In terms of time, regardless of printing angles, nearly 15 minutes was the total time taken to finish a hollow square box with the dimensions of $35 \times 35 \times 20\text{mm}^3$ (Table II). When it comes to the quality of printed objects, the overall appearances of the samples printed on either the flat or inclined planes with the angles of 10° , 45° , 60° , and 80° , appeared to be satisfactory (see Fig. 7c). Moreover, for quantitatively evaluating how the printing angles affect the surface quality of the resulted objects, we measured the averaged roughness R_z of the two most wrinkled surfaces of each sample using a laser scanning microscope (Keyence VK-9710). The representative images for the 3D surface profile of those samples are displayed in Fig. 8a and Fig. 8b reports the averaged surface roughness R_z of the corresponding samples. Obviously, there is not much difference among the value of the surface roughness, all at around $900R_z$. Therefore, generally speaking, no matter how much the printing angle is from 0 to 80 degrees, the printed objects always attain acceptable or even high quality.

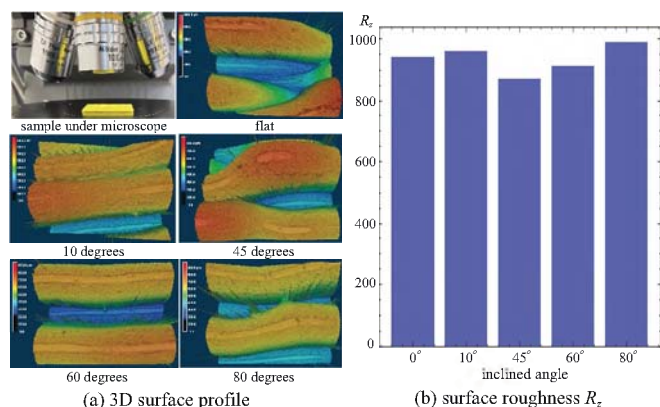


Fig. 8. (a) displays images for the 3D surface profile of thin walls. And the bar chart presents the averaged surface roughness R_z of the samples (b).

V. CONCLUSION

Towards advancements in fast prototyping and human-robot collaboration, adopting a 6-DoF industrial robot accompanied by a suitable trajectory planning scheme, like the proposed strategy of knowledge-based planning, could further consolidate their potential for the additive manufacturing. Moreover, 3D printing on inclined planes up to 80 degrees, by the robot arm is entirely feasible and has been verified in this research, whose results are expected to contribute to applications beneficial from this robot-based system, especially in the field of building and construction.

ACKNOWLEDGMENT

The authors would like to express our gratitude to colleagues for their support in this research, especially Mr. Dinh Quang Nguyen with the help of experimental setup for inclined-plane printing, and Mr. Pho Van Nguyen for the support in the measurement of surface roughness.

REFERENCES

- [1] Tuan D. Ngo, Alireza Kashani, Gabriele Imbalzano, Kate T.Q. Nguyen, David Hui, "Additive manufacturing (3D printing): A review of materials, methods, applications and challenges", *Composites Part B: Engineering* 143, pp. 172-196, 2018.
- [2] Vanek, J., Galicia, J.A.G. and Benes, B., "Clever Support: Efficient Support Structure Generation for Digital Fabrication", *Computer Graphics Forum* 575, pp. 330-335, 2014.
- [3] Skylar-Scott, M.A., Mueller, J., Visser, C.W. et al., "Voxelated soft matter via multimaterial multinozzle 3D printing", *Nature* 575, pp. 330-335, 2019.
- [4] Bhatt, Prahar M. et al., "Expanding capabilities of additive manufacturing through use of robotics technologies: A survey", *Additive manufacturing* 31, pp. 100933, 2020.
- [5] Pinar Urhal, Andrew Weightman, Carl Diver, Paulo Bartolo, "Robot assisted additive manufacturing: A review", *Robotics and Computer-Integrated Manufacturing* 59, pp. 335-345, 2019.
- [6] Xu Zhang, Mingyang Li, Jian Hui Lim, Yiwei Weng, Yi Wei Daniel Tay, Hung Pham, Quang-Cuong Pham, "Large-scale 3D printing by a team of mobile robots", *Automation in Construction* 95, pp. 98-106, 2018.
- [7] Mehmet Efe Tiriyaki, Xu Zhang, Quang-Cuong Pham, "Printing-while-moving: a new paradigm for large-scale robotic 3D Printing", 2018, arXiv:1809.07940 [cs.RO].
- [8] W.S. Yezzunis, J.C. Barnwell III, D.N. Nikovski, "Strengthening abs, nylon, and polyester 3d printed parts by stress tensor aligned deposition paths and five-axis printing", *Proceedings of the Solid Freeform Fabrication Symposium, ATT Conference Center* 10, Austin, TX, 2016.
- [9] P.M. Bhatt, R.K. Malhan, S.K. Gupta, "Computational foundations for using three degrees of freedom build platforms to enable supportless extrusion-based additive manufacturing", *ASME 14th Manufacturing Science and Engineering Conference*, Erie, PA, 2019.
- [10] C. Wu, C. Dai, G. Fang, Y.-J. Liu, C.C. Wang, "RoboFDM: a robotic system for support-free fabrication using FDM", *International Conference on Robotics and Automation (ICRA)*, IEEE, Singapore, pp. 1175-1180, 2017.
- [11] I.B. Ishak, J. Fisher, P. Larochelle, "Robot arm platform for additive manufacturing using multi-plane toolpaths", *ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* 5A, American Society of Mechanical Engineers, Charlotte, NC, 2016.
- [12] I. Ishak, P. Larochelle, "Robot arm platform for additive manufacturing: 3D lattice structures", *Florida Conference on Recent Advances in Robotics (FCRAR)*, Boca Raton, FL, pp. 11-12, 2017.
- [13] T. L. Boullion and P. L. Odell, *Generalized Inverse Matrices*, Wiley-Interscience, 1971.
- [14] Lynch, Kevin M. and Park, Frank C., *Modern Robotics: Mechanics, Planning, and Control*, Cambridge University Press, USA, 2017.