Artificial Intelligence Planning for Robotic Construction 3D Printing Applications: Printability Checking and Prefabrication

Erfan Shojaei Barjuei

Department of Computer Science, Bioengineering, Robotics and System Engineering (DIBRIS)

University of Genova

Supervisor

Prof. Marco Maratea, Prof. Fulvio Mastrogiovanni

Co-supervisors

Dr. Alessio Capitanelli

Dr. Riccardo Bertolucci

In partial fulfillment of the requirements for the degree of

Master of Science in Computer Engineering

March 29, 2022

Acknowledgements

I would like to thank the people at the department of computer science, bioengineering, robotics and system engineering (DIBRIS) of the university of Genova especially my supervisor, Prof. Marco Maratea for his help, support and interest in my works.

It is difficult to overstate my gratitude to Prof. Fulvio Mastrogiovanni, Dr. Alessio Capitanelli and Dr. Riccardo Bertolucci. With their enthusiasm, their inspiration, and their great efforts to explain things clearly and simply, throughout my thesis-writing period, they provided encouragement, sound advice, good teaching, good company, and lots of good ideas. I would have been lost without them. Lastly, and most importantly, I wish to thank my parents and my brother. They bore me, raised me, supported me, taught me, and loved me. To them I dedicate this thesis.

I dedicate my dissertation work to my family and many friends. A special feeling of gratitude to my loving parents, who have always loved me unconditionally and whose good examples have taught me to work hard for the things that I aspire to achieve.

I also dedicate this dissertation to my uncle "*Ebrahim Taghinejat*". Although he is no longer of this world, his memories continue to regulate my life.

Abstract

Construction 3D printing is a form of additive manufacturing which represents a process of fabricating buildings or construction components from a digital file by depositing a building material layer by layer without any formwork support. In this research work, an application of the automated planning, which is an artificial intelligence (AI) technique, to construction 3D printers is presented. On this basis, AI planners, expressed in Planning Domain Definition Language (PDDL 2.1), are developed and employed to generate a sequence of operations comprehensible to the control system of a robotic manipulator system which is to perform specific concrete 3D printing tasks to produce two spatial objects with different geometry specifications. Accordingly, AI planners are executed based on requirements of printability checking and prefabrication in robotic construction 3D printers. The planned sequences will then be input to a robotic simulator framework that will allow the user to monitor the whole 3D printing process. Moreover, the performance of the approach has been examined and analyzed through scalability tests and the obtained results demonstrated that incrementing edges and layers of an object causes an increase in the planner runtime. The work described in this paper addresses a new application of AI concepts to the robotic additive manufacturing domain so far lacking in the scientific literature.

Contents

List of Figures

Chapter 1

Introduction

1.1 Robotic Construction 3D Printing

Industrial automation and robotics are about the application of computers, control systems and information technology to drive industrial operations such as and construction, manufacturing and machinery [2]. In industrial automation, automatising of a process has been almost understood as a faster, cheaper and more accurate and precise way of production. Nevertheless, the sector of automation in construction has not been been developed and automated in the same way as other industrial automation sectors [3].

3D printing, also known as additive manufacturing technology, was first introduced in 1987. Today there are many different domains and branches within this technology but the basic principle is the same. 3D printing is an automated process for creating 3D objects from computer-aided design (CAD) models. In this technology, 3D models are subdivided into several layers which are used to be deposited layer by layer in order to construct the designed object [4]. Manufacturing objects with complex geometry specifications in a fast and optimized way is the main benefit of this technology [5]. Furthermore, 3D printing approaches reduce material waste and labor cost [6]. The key to a successful 3D printing process highly depends on the printing system components, building material properties and skills of users [7]. Nowadays, applications of 3D printers can be found in a wide range of domains such as food processing, aerospace, bio-engineering and constructions [8; 9; 10; 11].

3D printing technologies have been adopted to produce architectural and building models since the early 2000s. In fact, construction 3D printing deals with various technologies that employ 3D printing principals as the main core for producing buildings or construction components [12]. One of technologies that are widely used in Construction 3D Printing is Robotics. There are several types of robotic construction 3D printing platforms in available. Cartesian robots [13], robotic arms (manipulators) [14] and cable driven robots [15] are some examples of robotic applications in construction 3D printers. An example of a robotic manipulator construction 3D printer is illustrated in Figure 1.1. The application of the robotic science in Construction 3D Printing technology shows an autonomous, simple, flexible and adaptive approach for construction purposes [16].

Another type of concrete 3D printers, used in the industry, is a framed printer. This kind of construction 3D printers would suit only in factories as the transporting and assembling of this kind of printers are very difficult. Larger frame of the printer respect to the object to be printed is the main disadvantage of this kind of 3D printers. An example of a framed 3D printer is shown in Figure 1.3 [1].

The various stages related to the traditional construction techniques of a structure is depicted in Figure 1.2. The various stages involve human resources in different locations of the structure. While the traditional method involves human resources in different locations of the structure and it is time consuming and expensive, The 3D printing process requires less involvements of the user [17].

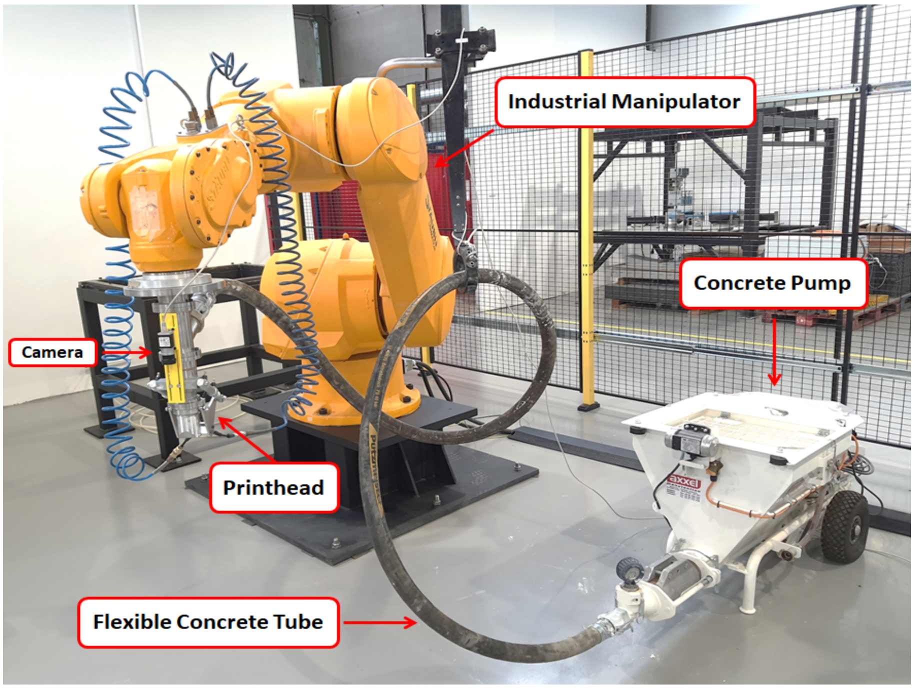


Figure 1.1: An example of a robotic arm construction 3D printer with indications to some main components, photo courtesy of INSA Rennes - LGCGM.

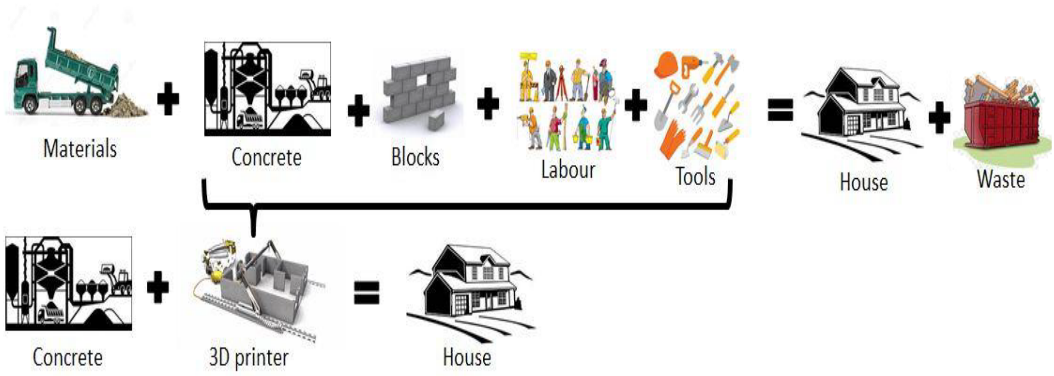


Figure 1.2: Traditional production process vs 3D printing.

In the past few years, many robotic construction 3D printing (RC3DP) approaches have been developed and proposed to enhance the application of automation in construction industry. However, most of these methods are based two principles i.e., extrusion-based and powder-based. The extrusion-based printing technique is done layer by layer deposition of the building material, whereas powder based printing is implemented by spreading the dry base materials in the fist step and covering it selectively by cement-based material [18]. While he first cement based additive operation was proposed by J. Pegna in 1997 [19] through a free form construction method, the first popular 3D concrete printing technique was developed and presented by Prof. Khoshnevis and his research group within a series of works at the university of southern California and has been named as ‘Contour Crafting method’ [20]. Contour Crafting method is, principally, a layered-based printing method which uses different materials such as cement, ceramic paste and polymer etc. to print and fabricate a structure with a smooth surface.

Using of cement-base materials for 3D printing needs a special type of these building materials. In fact, the cement-base materials should be flow-able like a paste and should be harden once it has been extruded. The building materials should not be harden fast while it is being used in the printing process. For the building materials used in the 3D printing process, there are certain properties which can be modified and adapted based on the application requirements [21].

Printability in additive manufacturing industry is defined as the ability to reproduce closely a 3D model through a 3D printer [22]. Although 3D printers are excepted to produce any 3D object, applications of 3D printers are still limited because of geometrical specifications, time consumption and building material properties.

In the present study, prefabrication also known as process planning is defined

Figure 1.3: Framed 3D printer used by TU Eindhoven [1].

as the relation between design and manufacturing process [23]. In fact, process planning defines a sequence of steps that should be implemented to create a product. Evaluation of time consumption close to a real practical scenario and consolidation of threshold control and print segments are the main problems in prefabrication of a 3D model [24].

1.2 Artificial Intelligence Planning

Artificial intelligence (AI) is a formation of intelligence demonstrated by machines, also known as machine intelligence. AI gives the capability of learning, acquiring information, planning and creativity to machines. On this basis, these devices can perform and carry out advanced and complex tasks [25].

AI systems are able to be adapted to a certain degree by examining the effects of past actions and processing autonomously. AI has been developed and applied to a wide range of domains, including robotics [26], computer science [27], transportation [28], navigation [29], marketing [30], medicine [31], industrial manufacturing [32] and so on.

Making plans is recognized an evidence of intelligence, on this basis, automated planning has been one of the goals of research in AI since its beginning. Accordingly, AI planning can be specified as the study of computational models and methods of producing, analysing, managing, and executing plans [33].

AI planning also known as planner is a branch of Artificial Intelligence which investigate the process of using autonomous methods to solve planning and scheduling problems in an efficient way. An AI planning system takes either the problem formalisation or model as input and implement some problem solving methods, such as heuristic search, propositional satisfability, etc, to obtain an efficient solution [34].

The Planning Domain Definition Language (PDDL), is a standard language for describing and expressing planning problems and domains. PDDL provides the possibility to select a suited planner for describing the technical and epistemological prerequisite of planning domains and the abilities of planners in a uniform and easy way. The model development of for PDDL requires the analysis of PDDL domain and problem files structures. The domain file consists of the domain predicates and operators (also recognized as actions in PDDL). The problem fie specifies an initialisation and a goal to be achieved [35].

There are different versions of PDDL and their extensions. Each version of PDDL describes new prerequisites that should be supported and enhances the functionality of PDDL language. PDDL 2.1 describes the possibility to implement numerical variables, plan-metrics, and durative actions which make the conditional AI planning such as fuel consumption, time constraints and capacity constraints possible [36].

1.2.1 Numerical expressions

Numeric expressions are formulated, using arithmetic operators, from primitive numeric descriptions. Numeric expressions can be represented as non-linear expressions and they can be applied to both in action preconditions, defining numeric constraints as a comparison between pairs of numeric expressions, and in action effects, renovating a numeric fluent value allocating it an updated value or increasing/decreasing it. Numeric expressions allow increasing the expressiveness of PDDL and permit a better definition of domains. They also provide the possibilities of planners which can keep track of numerical information and utilize them during the AI planning process.

1.2.2 Plan metrics

Plan metrics specify the basis on which a plan has to be assessed through numeric descriptions. Metrics can be represented in the problem description with any arithmetic formulation with no requirements of linearity and they can need to minimize or maximize the given definition.

1.2.3 Durative actions

This kind of action allows to define temporally annotated tasks. A durative action is represented by duration constraints, and temporally annotated conditions and effects.

1.3 Motivations

Additive manufacturing is a complicated process. For a high efficient 3D printing process, numerous process parameters and variables should be pre-tuned and monitored before and during the process which is almost impossible in practice. On this basis, AI can improve remarkably the efficiency of 3D printing technology [37].

As mentioned in Section 1.1, one of the important advantages of 3D printing is the capability to produce customized components and objects. Principally every 3D model is unique by design therefore it is very important to evaluate the printability of the profile of the product to be printed to ensure that it can be successfully fabricated with high quality. Furthermore, the quality of the final structure can be predicted and processing errors be avoided that could also result in time savings. Unfortunately, the methods, proposed so far for the printability checking issue in additive manufacturing domains, have not yet proven to be robust enough and cannot be considered as a conclusive solution. Hence, a new approach is therefore needed such that users do not require dealing with high complex tasks and data. Accordingly, to be able to feasibly solve the printability checking issue in robotic construction 3D printing industry, it will be demonstrated in next sections that AI planning methods can be considered as a promising solution.

By increasing the complexity of the design, a challenging problem which arises in this printing domains is the optimisation of prefabrication or process planning [38]. The importance of the optimisation of prefabrication can be clearly seen in shortening the process planning time and consolidation of the threshold problem of print segment. In an ideal prefabrication process, the time consumption should closely match to a practical scenario, which can provide high quality monitoring services and optimal solutions for users in a real manufacturing process. On the other hand, the high computational complexity is one major drawback of the approaches proposed and developed so far for the prefabrication issue in additive manufacturing domains. In order to be able to feasibly solve the prefabrication (process planning) problem in robotic construction 3D printing systems, a method to model the tasks necessary to enable the robot to carry out an concrete 3D printing process is needed. Thus, it requires planning techniques that includes both robotic construction 3D printing task planning and motion planning. In this research work, it will be demonstrated that applying AI planning techniques not only can be understood as a promising solution for the printability checking issue but also can be considered as an appropriate and effective approach for the prefabrication problem in construction robotic technologies.

Furthermore, an integrated parallel computing of printability checking and process planning may accelerate computing time and reduce the time consumption of the manufacturing process.

1.4 Context of the Study

AI is mostly defined and specified as machine capability to solve given problems by itself in the absence of human intervention based on available data and experiences of the past. Hence, applications of AI to additive manufacturing industry could enhance the performance of 3D printer machines by decreasing the possible printing errors and improving the automatic process. On this basis in this thesis, a methodology to apply some AI techniques to task of performing automatic robotic concrete 3D printing operation is described. Thus, the task of AI is to create a bridge between the physical and digital worlds.

In fact, the main focus of this study is on the planning algorithm expressed in PDDL 2.1 to provide a intelligent module that can generate tasks required by a robotic construction 3D printer to follow in order to perform a 3D printing process. It will be demonstrated in next sections that the proposed AI module can be understood as a promising approach for printability checking and process planning issues in robotic construction 3D printers.

1.5 Objectives and Contributions

In this thesis, an AI planning framework is developed and proposed to generate the commands that a robotic construction 3D printer platform will have to follow in order to carry out an automatic operation. The commands by AI planning system will then be input to a robotic simulator, that will provide the ability of monitoring the whole 3D printing process for the user. Finally, the commands will be executed by the robotic platform thus performing the additive manufacturing tasks, while meeting all the principal requirements concerning printability checking and prefabrication of a concrete 3D printing operation. It should be noted that the AI system proposed in this study is not intended to substitute the existent robotic contraction 3D printer systems, but rather to integrate and assist such systems. The work, described in this thesis, reports a high feature of novelty, as no similar implementations of AI planning frameworks to a robotic concrete 3d printer platform could be observed in the scientific literature.

1.6 Overview of the Thesis

This thesis consists of the following parts as described below:

1. Introduction: This section gives a brief introduction about the background of additive manufacturing, robotic construction 3D printing, AI planning, research aims, scope and contribution of this thesis.

2. State of the Art: This section gives a detailed view on previous research on robotic construction 3D printing technologies, portability checking and prefabrication in the additive manufacturing industry.

3. Related Works: This section overlooks the various the works carried out on applications of AI to 3D Printing and planning techniques in additive manufacturing industry.

4. An Intelligent System for Planning RC3DP Operations: This section describes rationale, assumptions, problem statement, reference scenario and the formulation of the AI planning system.

5. Experimental Analysis: This section explains the procedures of simulation scenario and scalability tests and reports the obtained results from these procedures as well.

6. Conclusions: This chapter concludes the findings and analyses of development of the proposed AI planner for robotic construction 3D Printers in the present and future.

Chapter 2

State of the Art

Summary

Although the robotic technology is developing fast in the past decades, application of robots in additive manufacturing industry is almost a new research topic [39]. Given this, the presented literature covers various parts of the research including two main issues in additive manufacturing domains, namely "printability checking" and "prefabrication".

2.1 Prior Work on RC3DP Technologies

Among the developed and presented methods and techniques for construction 3D printers, the extrusion-based additive manufacturing approaches have been the most investigated and studied [40]. In this technique, the cement-based material is dispensed precisely at predefined locations by means of a printhead (extruder). Accordingly, the printhead moves in a 3D space based on a predetermined path to build up the predesigned object in a layer-by-layer way [41]. It is worth to mention that the success of this process highly depends on the material structural build-up rate and the construction rate [42]. Nematollahi et al. in [43] report some remarkable benefits of using extrusion-based additive manufacturing methods in respect of reducing the construction cost and time, reducing the injury rates and increasing architectural freedom. Several types of construction 3D printers such as Cartesian robots [44], robotic arms (manipulators) [45], cable driven parallel robots [46] and Delta robots [47] are currently in used and various objects [41] have been produced by these robotic platforms. In a concrete 3D printing process, printing speed, printhead shut-off system, stand-off distance (SOD) and temperature are parameters which are needed to be controlled and predefined as they affect the shape, quality and behavior of printed concrete objects [48; 49]

2.2 Prior Work on Printability Checking

In additive manufacturing domains, the term printability basically refers to the ability to closely reproduce a 3D model by means of a 3D printer device [50]. The printability of a 3D object depends on several factors such as consumption time, manufacturing cost, building material, geometry specifications of the object, etc [51; 52; 53; 54]. Fudos in [55] developed and proposed an approach to analyze and evaluate the complexity of the geometrical features (vertices in polynomial meshes, edges, faces, etc.) of the object to be printed. These characterization are directly extracted from a computer-aided design (CAD) file. Nevertheless, in a real practical 3D printing operation by dealing with a numerous features and a high amount of data, there is a very high risk of failure for this technique. On the other hand, various studies regrading to printability checking have been carried out based on machine learning (ML) techniques [56; 57; 58; 59]. A challenging problem which arises with such approaches is that current ML methods need a large amounts of data for training models but in robotic construction 3D printing machines, obtaining high-fidelity data is a critical problem [60].

2.3 Prior Work on Prefabrication

Apart from printability, prefabrication or process planning is another issue in additive manufacturing domains. In fact, geometry specifications of the 3D model and time consumption time are the most important factors in a process planning for a robotic construction 3D printer. While geometry specifications of the 3D model can effect path planning and control approaches, consumption time should closely match up to a real practical scenario [40; 61]. So far, various approaches have been developed to solve the prefabrication problem in construction 3D printing industry [62]. The proposed approaches are mostly based on converting 3D model into a set of slicing planes, Graphic Processing Unit (GPU) schemes and search algorithms for the optimal printing trajectory [63; 64; 65; 66]. However, one major drawback of these methods is the The high computational complexity.

It is worth noticing that a few works can be found on the integration of printability checking and process planning in the field of additive manufacturing in the literature [67].

Chapter 3

Related Works

Summary

With the continuous development of AI and construction 3D printing at the same time, how to integrate AI techniques into additive manufacturing industry in order to drive the transformation of traditional construction and improve efficiency of operation has become a hot research topic. Currently, robotic construction 3D printing (RC3DP) is still immature, there are many issues in theory,approaches and applications. AI provides new solutions for these problems by applying a number of methods, such as machine learning and computer vision, to construction 3D printing.

3.1 Research on Applications of AI to 3D Printing

In the last few decades, researchers and scientists have worked and published articles as well on the application of AI and its domains to tackle construction specific issues and challenges. For example, machine learning has been applied and used for health and safety monitoring, cost estimation, supply chain and logistics process improvements, possible risk prediction amongst others [68; 69; 70; 71]. Intelligent robots has been used and applied in site monitoring and performance evaluation, virtual assembly, and the management of construction buildings, plant and equipment [72; 73; 74]. Furthermore, knowledge-based systems have also been used and applied for tender evaluation, conflict resolution, risk analysis, waste management, sustainability evaluations, etc [75; 76]. Despite recent and remarkable developments in automated and semi-automated AI-based optimisation of 3D printing operations, researchers have addressed many issues to solve [77]. The main applications focused on parameter optimisation, and anomaly detection, and may be classified into several kinds of machine learning techniques, including regression, classification, and clustering [78].

3.2 Research on Planning in Additive Manufacturing

There have been a large mounts of research works on trajectory planning and optimization for use on industrial robots in various fields such as welding [79], painting [80], cleaning [81] and finishing [82]. Regarding to robotic additive manufacturing operations, Ding et al. [83] proposed an adaptive path planning for wire-feed additive manufacturing by using medial axis transformation. The proposed algorithm was applied by a robotic wire platform and arc additive manufacturing system and casused material savings, voidfree deposition, and improved accuracy at the boundary. This research presented an advanced additive manufacturing algorithm, however, not many works have been carried out on robotic trajectory planning. Another research project was developed on a robotic additive manufacturing operation by Zhang et al through the direction of curved surfaces [84].

In [85], a robotic additive manufacturing process is simulated by means of the RobotStudio software and the fabrication process is performed accordingly.

Chapter 4

An Intelligent System for Planning RC3DP Operations

Summary

Typically, AI planning systems are faced with following steps: given, i) a complete description and specification of the initial state of the world (an application domain), ii) a set of action schemas that identify how the world may change, and iii) a goal condition, the system has to find a sequence of actions in such manner when applied one after the other in the initial state, it transforms the state into one which satisfies the goal condition [86]. In this study, the AI planning system allows planning and execution of different tasks, assigned to a robotic construction 3D printing process, based on orchestrating the behavior of different modules. The data, structure and specifications formats of the intelligent system, developed and proposed in this work, is described in this section.

4.1 AI System Principals

Generally, AI planing systems are problem-solving algorithms that operate on explicit propositional or relational representations of states and actions. These definitions provide the possibility to obtain effective heuristics and the development of powerful and flexible algorithms for solving problems [87]. As was pointed out in Section. 1, the main goal of the work presented here is to develop a planner allocated to automatic robotic construction 3D printing operations in a "virtual reality" environment to conform the requirements of printability and prefabrication in such systems. Hence, the task of the AI system is to make a bridge between the physical and digital worlds (environments), namely the "real" world of the additive manufacturing process and the simulation environment. Accordingly, a robotic arm has to follow the commands generated by the AI planner in order to perform the 3D printing tasks based on printing requirements and geometry specifications of the object to be printed. Fig. 4.1 represents the phases of the proposed AI planning system. Accordingly, the problem considered in this work can be defined as follows: given the geometry specification of a 3D object to be printed and printing process requirements, determining a plan as an ordered set of actions:

$$\mathcal{A} = \{a_1, \ldots, a_k, \ldots, a_n; \prec\} \tag{4.1}$$

where every action a_k involves construction 3D printing operations to be executed by a robotic arm platform. Hence, the 3D printing process consists of a sequence steps that should satisfy the following conditions:

- The extrusion approach of the building material is based on the infinite brick strategy [88];
- Each object to be printed consists of either one layer or a number of layers;

- Each layer of the object to be printed is subdivided into edges (links) and vertexes;
- The object to be printed contains only direct bar-shaped edges;
- The cross-section size of each printed layer is equal to the outlet dimension of the printhead nozzle;
- Printing of layers should be done in a sequence way;
- Printing of each edge should be done with a specific constant velocity;
- The object to be printed has a uniform layer shape;
- The building material is cement-based;
- There is a resting time between printing two stacked layers;
- The height of the printhead above the layer to be printed during the printing process is almost equal to the printhead nozzle outside width.

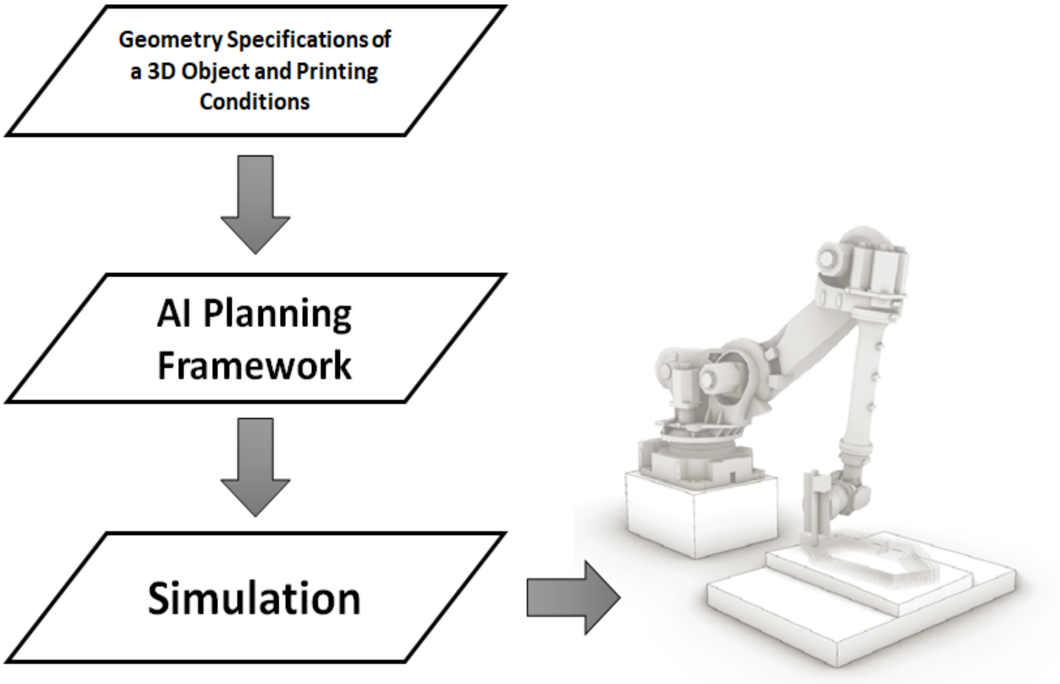


Figure 4.1: Phases in the intelligence system which starts from geometry specifications and printing requirements of a 3D object, through an AI planning framework, carries out a simulation.

Based on above conditions, the geometry configuration of the 3D object to be printed can be defined as a 2-ple $\alpha = (\Upsilon, \Gamma)$, where Υ is the ordered set of the object vertices (v) and Γ represents a set of function which refer to the angle (θ) and length (ζ) of the object edges, i.e.,

$$\begin{cases} \Upsilon = \{v_{00}, \ldots, v_{0j}, v_{i0}, \ldots, v_{ij}\} \\ \Gamma = \{f_l(v_{00}, v_{01}), \ldots, f_l(v_{ij-1}, v_{ij})\} \end{cases} \Longleftrightarrow \begin{cases} i = 0, 1, \ldots, N_\beta - 1 \\ j = 0, 1, \ldots, N_\gamma - 1 \end{cases} \tag{4.2}$$

where each $v_{ij} \in \Upsilon$ is characterized by its position in the $x - y$ plane i.e., $v_{ij} = (x_{ij}, y_{ij})$. While N_β and N_γ represent the number of layers and vertices respectively, f_l is a function referring to the angle (θ_{ij}^{ij-1}) and length (ζ_{ij}^{ij-1}) between two connected vertices (a link) in the $x - y$ plane respect to the global

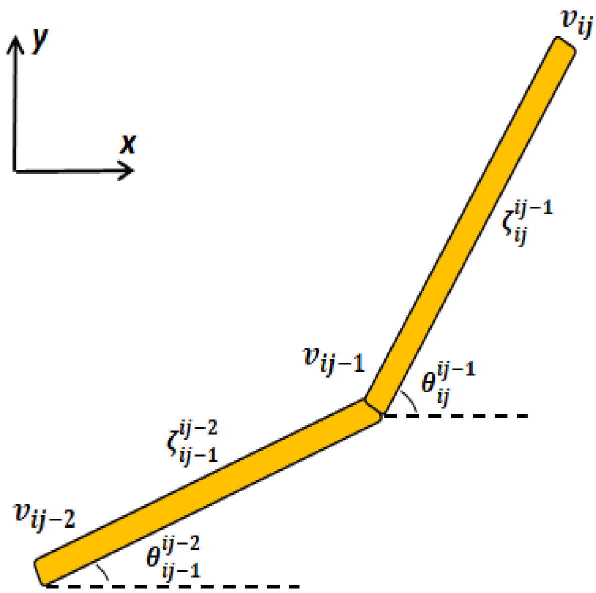


Figure 4.2: A graphical representation of the layer of an object with two edges (links) from top view.

reference frame as:

$$f_l(v_{ij-1}, v_{ij}) = (\theta_{ij}^{ij-1}, \zeta_{ij}^{ij-1}) \tag{4.3}$$

To exemplify the mathematical formulations used to describe a 3D object in this work, a top view of a graphical representation of an object with two edges (links) is shown in Fig. 4.2. By knowing that the printhead has a cartesian motion above and aligned with the layer to be printed during the printing process, the robot configuration ($\mathcal{C}_{robot}$) can be defined through the end-effector path in the operative space ($\mathcal{C}_{oper}$) from a given start to a given goal state. For example, the robot configuration space, $\mathcal{C}_{robot} \subseteq \mathcal{C}_{oper}$, is the space where the robot does not collide with the printed structure or itself and it can be mathematically expressed as:

$$\mathcal{C}_{robot} = \mathcal{B}\left(u(t)\right) \tag{4.4}$$

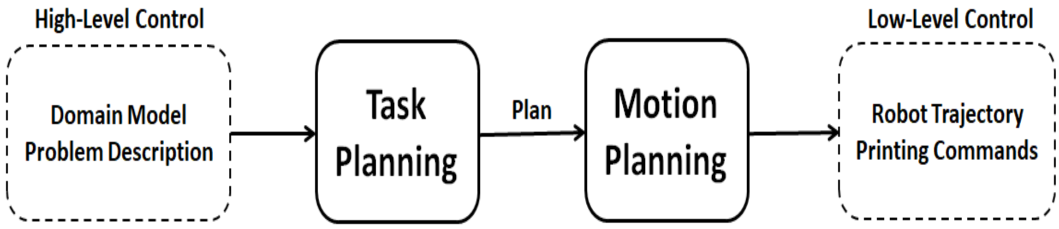


Figure 4.3: An overview of the AI planning framework for robotic construction 3D printing applications.

in which u and t represent the motion law and instantaneous time respectively. $\mathcal{B}$ can be any convenient parametrized form of the robotic arm based on the printhead (end-effector) path. Accordingly, the time interval between two consecutive vertices for the robot end-effector can be computed as:

$$\Delta t_{ij}^{ij-1} = \frac{\left|v_{ij} - v_{ij-1}\right|}{\psi_{ij}^{ij-1}} = \frac{\zeta_{ij}^{ij-1}}{\psi_{ij}^{ij-1}} \tag{4.5}$$

where ψ_{ij}^{ij-1} is a constant linear velocity of the printhead during the printing operation of an edge. It should be pointed out that one of the important criteria to be considered for choosing a robotic manipulator for construction 3D printing applications is the robot's workspace (envelope). That is, the workspace structure is important for assuring kinematic characteristics of the robot during the 3D printing operation in a way that the robotic arm can reach any given point in space, with any given orientation.

4.2 AI Planning Approach

A robotic construction 3D printing platform needs logical reasoning to determine which actions (tasks) are required to achieve a given goal, and it also needs geometric reasoning to know if and how these actions can be physically implemented. On this basis, the main objective of the AI planning approach is to combine logical and geometric reasoning for carrying out an automatic robotic construction 3D printing process. An overview of the AI planning framework, used in this work, is shown in Fig. 4.3. As it can be seen, the framework mainly consists of two elements, namely task planning and motion planning modules. While at a high-level control perspective, the task planner generates a sequence of discrete or symbolic actions (tasks) from an initial state to a predefined goal state, the motion planner determines corresponding paths through the robotic arm configuration space characterized by numerical robot motion commands (low-level control).

A wide range of problems such as 3D objects with different geometry specifications and printing requirements can be covered by the task planning to generates AI plans. An AI plan, defined formally in Ex. 4.1, is as an order sequence of n actions whose execution changes the initial state from s_I to s_G (goal state) through a finite set of states:

$$\mathcal{S} = \{s_I, \ldots, s_e, \ldots, s_G; \prec\} \tag{4.6}$$

In a plan, each action may correspond to one or more scripted robot behaviors; in other words, each action a_k in an AI plan is assumed to transform a state s_e into a state s_{e+1}. In accordance with the definition of AI plan and problem description file, a task domain model can be defined as a 5-ple $\Lambda = (\mathcal{S}, \mathcal{A}, \gamma, s_I, s_G)$ where $\gamma : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ is a deterministic state-transition function which gives one state when applicable. Thus, the AI plan generated by the task planning is in the string format (language) of Λ. As it is computationally difficult and infeasible to represent all the states ($\mathcal{S}$) in an explicit manner even for small task planning problems, the Planning Domain Definition Language (PDDL), which benefits from various compact and symbolic representations, has been used to formulate the domain model in this work.

Assuming that, a 3D object (α) is aimed to be printed based on a plan (actions) by the task planning module. A motion planing module for the robotic arm is needed to find valid paths through $\mathcal{C}_{oper}$ from an initial configuration ($q_I \in \mathcal{C}_{oper}$) to a goal configuration ($q_G \in \mathcal{C}_{oper}$). Hence, the motion plan can be defined by restating Eq. 4.4 as an ordered set of robot configurations:

$$\mathcal{C}_{robot} = \{q_I, \dots, q_r, \dots, q_G; \prec\}, \quad \|q_{r+1} - q_r\| \leq \epsilon \tag{4.7}$$

in which every $q_r \in C_{oper}$ corresponds to joints position vector for the robot. Consequently, Eq. 4.7 can also be inferred as a continues trajectory $\tau : [0,1] \rightarrow \mathcal{C}_{oper}$ for the manipulator during 3D printing operation such that $\tau(0) = q_I$ and $\tau(1) = q_G$.

As it was mentioned in Section.4, the robot moves in a planar dimensional worlds ($\mathcal{W} = \mathbb{R}^2$) during the printing edges of a 3D object. It should be noted that the MoveIt software [89], which is a widely-used motion planning package, has been used in this work to model the configuration space of the robot manipulator. Furthermore, robotic construction 3D printing commands such as switching on/off the printhead (opening and closing the nozzle) have been considered as independent robot configurations ($q_r \in C_{oper}$).

4.3 AI Planning Formulation

In order to address the issue introduced in this work, Planning Domain Definition Language (PDDL) has been exploited to formulate the AI task planning explained in Section. 4.2. PDDL was introduced by Drew McDermott in 1988 to represent planning problems. PDDL is a language which is influenced by the STRIPS formulations of AI planning problems and it is widely used in the academic planning community. In PDDL, the description of the predicates, functions and action schemas is separated from the definition of the initial state and the goal condition. While the first part is principally referred to the planning domain model (planning domain), the second part is involved in the problem description (planning problem) of the AI planning system. Such type of structuring that not only provides the use of variables to parameterize actions but also to reclaim and reuse a single domain file with various problem description files [34]. In particular for a robotic additive manufacturing application, functionalities of a robotic platform that may be available at different times are specified in a planning domain. By using this planning domain, different geometry specifications and printing requirements for 3D objects to be printed as well as goal conditions for the corresponding robotic additive manufacturing tasks can be represented in planning problems.

Several extensions of PDDL have been introduced so far. Every version of PDDL defines new necessities that should be supported and enhances the flexibility and functionality of PDDL language. One of the most practical PDDL versions is PDDL 2.1, which is used in this work, specifies the possibility to use numerical variables, plan-metrics, and durative actions which allow planning with conditions [**?**]. The syntax of PDDL 2.1 is formed based the logical representation of literals with the use of a prefix notation. As an instance, the positive literal $initial\_vertex(v_{ij})$ may be represented as (`initial_vertex ?vtx`) where the variable is denoted with a preceding question mark (?). Correspondingly, the *initial state* is described as a set of positive ground literals. This provides an entire identification of the state based on a closed-world assumption, i.e., for all the ground literals not included in the set, the negative version of the literal is supposed to hold. Regarding to *action schemas*, in the case of preconditions and conditions, a set of positive literals describe what requires to be specified in the state representation of an action to be executable, and for the effects of an action, a set of positive and negative literals describe how the states should be changed and transformed after the action execution: all the positive literals in the set of effects are added in the set specifying the state, and all negative literals are removed. A *goal condition* in PDDL also includes a set of positive ground literals. Basically once all the literals listed in the goal condition are involved in the set that defines the state, the goal will be satisfied. In PDDL 2.1, the special prefixes like `(:predicates ...)`, `(:functions ...)`, `(:action ...)` and `(:durative-action ...)` are used to specify of the planning domain. Accordingly, a planning problem components are specified through the special prefixes `(:objects ...)`, `(:init ...)` and `(:goal ...)`. This notation will become more clear in this section by explaining the basics of the planning domain and problem files for a robotic construction 3D printing process in the following.

The predicates of the AI planning domain file essentially represent properties of the state of the robotic construction 3D printing process at any time. In particular, the following predicates describe information regarding to the status, motion paths, position and rotation of the available robotic manipulator end-effector (printhead nozzle):

- `(valid_move ?vtx_from - vertex ?vtx_to - vertex)`
- `(visited ?vtx - vertex)`
- `(nozzle_at ?vtx - vertex)`
- `(nozzle_is_open)`
- `(nozzle_is_close)`
- `(on_same_way ?vtx_from - vertex ?vtx_to - vertex)`
- `(not_on_same_way ?vtx_from - vertex ?vtx_to - vertex)`

By way of illustration, the predicate `(visited ?vtx - vertex)` specifies that the printhead has been already located on the top of the vertex v_{ij}. The following predicates specify the 3D geometry of the object to be printed:

- `(printed_link ?vtx_from - vertex ?vtx_to - vertex)`
- `(not_printed ?vtx_from - vertex ?vtx_to - vertex)`
- `(above_on ?vtx_from - vertex ?vtx_up - vertex)`
- `(initial_vertex ?vtx - vertex)`

As it was mentioned in the former sections, both the robot configuration and geometry specifications of the 3D object are characterized based on proprieties of the object vertices and the relationships between them. For example, the predicate `(nozzle_at ?vtx - vertex)` indicated that the robot end-effector (printhead) is located on the top of the vertex v_{ij} with a dedicated distance. The printhead path during the printing process as well as object links (edges) are specified via `(valid_move ?vtx_from - vertex ?vtx_to - vertex)`. It should be noted that some literals such as `(nozzle_is_open)` are served as book-keeping information for modeling the robotic platform actions and their corresponding effects.

Numeric expressions (numeric fluents) are defined through functions in the domain description file. The functions are declared similar to predicates and their values may vary when an action is executed. A list of functions used in this study is given as:

- `(layer_tickness)`
- `(nozzle_angle)`
- `(nozzle_height)`

- `(rest_time)`
- `(relocate_time)`
- `(change_layer)`
- `(link_length ?vtx_from - vertex ?vtx_to - vertex)`
- `(printing_velocity ?vtx_from - vertex ?vtx_to - vertex)`
- `(motion_angle ?vtx_from - vertex ?vtx_to - vertex)`
- `(link_height ?vtx_from - vertex ?vtx_to - vertex)`

The functions as same as predicates are used to encode state variables of the robotic additive manufacturing operations. For instance, `(link_length ?vtx_from - vertex ?vtx_to - vertex)` indicates to the length of the object edges (ζ_{ij}^{ij-1}).

An action schema in a domain file describes an approach that can effect the state of the world. PDDL 2.1 supports both simple and durative action schemes which are defined in terms of their parameters, duration (for `durative-action` schemata), preconditions (for `action` schemata), conditions (for `durative-action` schemata) and effects which may include either positive or negative predicates and also numeric and conditional expressions. The action schemes, developed and used in this research, are as follows:

- `print (?from - vertex ?to - vertex)`
- `switch_off (?from - vertex ?to - vertex)`
- `switch_on (?from - vertex ?to - vertex)`
- `rotate_nozzle (?from - vertex ?to - vertex)`
- `go_to_init (?from - vertex ?to - vertex)`

- `waiting (?from - vertex ?to - vertex)`
- `increase_nozzle_height (?from - vertex ?to - vertex ?up - vertex)`

Each aforementioned actions schemes has an intuition meaning. For example, the following statement is a classical instantaneous action schema for switching on the printhead (opening the nozzle) during a 3D printing process:

```
(:action switch_on
:parameters (?from - vertex ?to - vertex)
:precondition (and
    (nozzle_at ?from)
    (valid_move ?from ?to)
    (visited ?from)
    (not_printed ?from ?to)
    (on_same_way ?from ?to)
    (nozzle_is_close))
:effect (and
    (not (nozzle_is_close))
    (nozzle_is_open))
)
```

The intuition is that a ground instance of the action schema `switch_on (?from - vertex ?to - vertex)` is performed once the switched-off printhead (closed nozzle) (`nozzle_is_close`) is located at the beginning of an edge, i.e., the robot end-effector is on the top of v_{ij-1}, (`nozzle_at`, `visited` and `valid_move`). The edge has not been printed yet (`not_printed`) and the printhead is properly aligned with the direction (θ_{ij}^{ij-1}) of this target edge (`on_same_way`). As the result of performing the action is that the prinhead receives a signal to be switched-on (`nozzle_is_open` and `not (nozzle_is_close)`).

The main difference between actions in classical planning and temporal planning is that in the latter it must be specified if an effect takes place at the start or end of an action as well as if a condition is for the start, end, or entire duration. From the modelling perspective, a durative action is distinguished from a classical instantaneous action by using the prefix `(:durative-action ...)` to describe it in the domain description file. An example for a durative action in this work is the one for concrete 3D printing of an edge which is defined as:

```
(:durative-action print
:parameters (?from - vertex ?to - vertex)
:duration (= ?duration (/ (link_length ?from ?to) (
    printing_velocity ?from ?to) ))
:condition (and
    (at start (nozzle_at ?from))
    (at start (valid_move ?from  ?to))
    (at start (nozzle_is_open))
    (at start (not_printed ?from ?to))
    (at start (> (nozzle_height) (link_height ?from ?to)))
    (at start (on_same_way ?from ?to)))
:effect (and
    (at end (not (nozzle_at ?from)))
    (at end (not (not_printed ?from ?to)))
    (at end (nozzle_at ?to))
    (at end (visited ?to))
    (at end (printed_link ?from ?to)))
)
```

The intuition is that the duration of a ground instance of this durative action schema is the printing time between the initial vertex v_{ij-1} and final vertex v_{ij} of an edge which is obtained from Eq. 4.5. This action schema can be performed after locating the switched-on nozzle (`nozzle_is_open`) in correct orientation (θ_{ij}^{ij-1}) (`on_same_way`) and position (`nozzle_at` and `(> (nozzle_height) (link_length))`) respect to the first vertex of the edge to be printed (`valid_move` and `not_printed`). Conversely, at the end of the action, the target edge will be printed (`printed_link` and `not (not_printed)`) and the nozzle will be located in the top of the second vertex (`nozzle_at`, `not (nozzle_at)` and `visited`) of the printed link.

Note that the difference from a classical instantaneous action's `:precondition` section, there is the `:condition` section in a durative action schema. Furthermore, predicates or expressions with `at start` prefixed to them, imply that the condition must be true at the start of the action in order for the action to be applied. In like manner, predicates and expressions with `at end` prefixed to them, mean that the expected effects must be true at the end of the action execution. In this work in order to allow maximum compatibility with available planners, only positive literals in the preconditions of actions have been used.

A problem file is what an AI planning system tries to solve. A problem file is formalized with the `:objects` section , specifying all the available objects (e.g., here an object of `vertex` is specified), `:init` and `:goal` sections to describe the initial state and the desired final state, using positive ground literals. While the initial state is defined as a list of literals, the goal condition is formalized as a logical sentence.

The complete domain model file as well as problem description files, developed and used in this work, are reported in Appendix. A.

Chapter 5

Experimental Analysis

Summary

In this section, the robotic simulator framework, developed in this work, for converting action sequences by the AI planner to numerical motion control commands of a manipulator (from high- level to low-level control [90]) as well as the 3D printing results of two objects are described in details. Moreover, in order to evaluate the performance of the AI algorithm, scalability tests have been carried out and the obtained results have been discussed accordingly.

5.1 Simulations

The target 3D objects to be printed in this work are depicted in Fig. 5.1 and Fig.5.2 where every vertex is represented with vtx_{ij}. Subsequently, the problem description files have been developed based on geometry specifications and 3D printing necessities of the target objects.

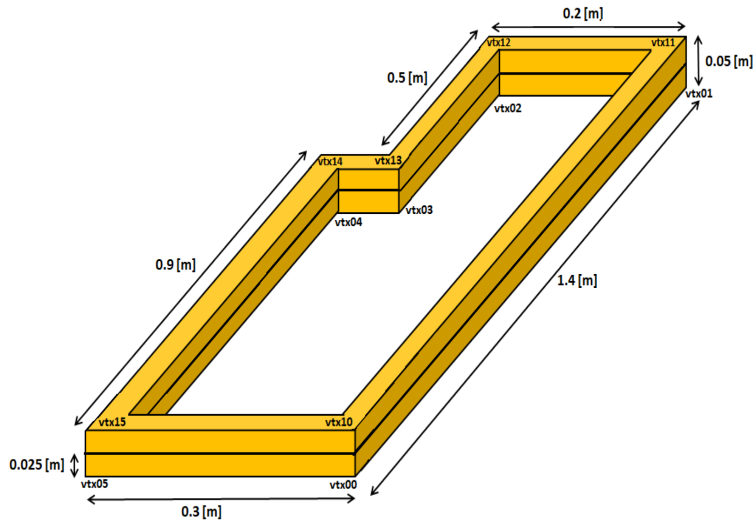


Figure 5.1: The closed-shape 3D object to be printed. The object consists of two layers and direct edges.

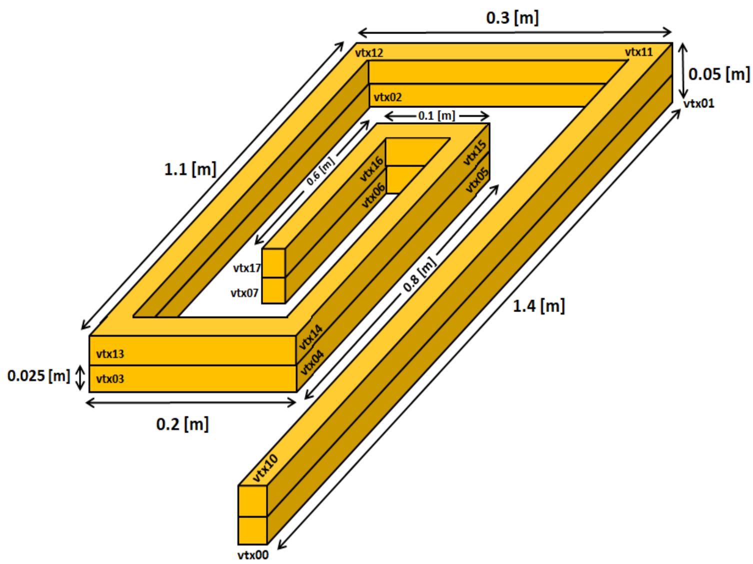


Figure 5.2: The open-shape 3D object to be printed. The object consists of two layers and direct edges.

Figure 5.3: The robotic simulator framework for converting AI plan tasks to robotic numerical motion control commands and 3D printing operations.

In order to obtain the plan for a robotic additive manufacturing process, the ROS module ROSPlan [91] has been used. ROSPlan solves the resulting planning problem using heuristic forward search planners such as the Partial Order Planning Forwards (POPF) planner [92] which is a temporal planner based on PDDL 2.1. Accordingly, the initial conditions as the preliminary data must be taken into account as follows:

- The coordination of all vertexes of the target object and the relation between them are known;
- The edges and links between vertexes that printhead can move toward are known;
- The printing velocity for each edge is known;
- The thickness of each layer, as well as the dimension of the nozzle cross-section, is known;
- The resting time between stacked layers and the time that the printhead requires to be relocated in the start point of each printed surface (layer) are known;
- The initial position, orientation and on-off status of the printhead are known.

The actual concrete 3D printing operation is then carried out in a step-by-step manner as follows:

1. While the printhead is switched-off, it is located at the start point (above the first vertex of the first layer) and its direction is in the same direction of the first edge;

2. The printhead receives a signal (command) to be switched-on (`switch_on`);

3. While the printhead is switched-on, it starts to move along the predefined trajectory with a dedicated constant velocity and during the same period of time the first edge is being printed (`print`);

4. While the first edge is printed and printhead is located above the second vertex, the printhead receives a command to be switched-off (`switch_off`);

5. While the printhead is switched-off, the manipulator end-effector starts to rotate in a way that the printhead direction be in the same direction of the second edge (`rotate_nozzle`).

6. The steps 2, 3, 4 and 5 will be repeated for non-printed vertexes and edges until the printing termination of the first layer;

7. If the object to be printed has a closed geometric shape, the printhead is located at the start point and it remains fix until satisfying the resting time condition between two stacked layers. Otherwise, the object has an open geometric shape and the switched-off printhead first moves from the above of the last vertex to the start point (`go_to_init`) and then it remains constant based on the resting time condition between two stacked layers (`waiting`);

8. With the passing of the resting time, the robot end-effector goes up vertically in a way that the printhead be located in the start point of the second layer (`increase_nozzle_height`);

9. While the printhead is located in the start point of the second layer and it is in the switched-off mode, the robot end-effector starts to rotate in a way that the printhead direction be in the same direction of the second edge of the second layer (`rotate_nozzle`);

10. The rest of the printing process contains the repetition of the presented procedure for non-printed layers from step 2 until the printing termination of the 3D object.

The robotic manipulator chosen to perform the 3D printing operation undergoing the simulation test phase is a commercial Universal Robot (UR10), whose kinematic features are well known; however, the simulation can be implemented with other standard robotic manipulator based on the user's application and dimension of the 3D object to be printed. Figure. 5.3 shows the overview of the simulation procedure built in the Robot Operating System (ROS - Noetic) environment [93]. As it can be seen and previously mentioned, the AI plan is generated and executed with the ROSPlan planning framework tool based on the defined domain and problem files. Then, the corresponding actions (tasks) by the AI plan are dispatched to the MoveIt software through a ROS node. Accordingly, the angle and speed information of the robot joints for creating the motion plan are calculated by the MoveIt. Figure. 5.4 shows a screenshot of the MoveIt graphical interface panel for the simulation operation. It is worth to mention that all programs related to the MoveIt software are developed with C++ programming language.

In the next step, the information generated by the MoveIt including the ma-

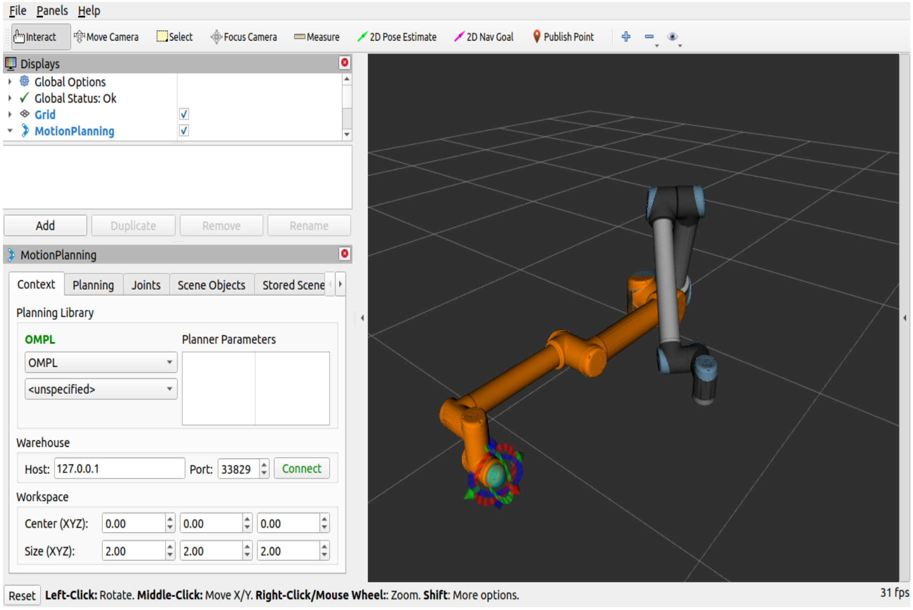


Figure 5.4: A screenshot of the motion planning configuration developed and displayed in the MoveIt control panel.

nipulator kinematics data and switch on/off signals of the printhead are transferred to the CoppeliaSim robotics simulator [94] through ROS nodes for visualizing robotic construction 3D printing operations. Example frames from the simulation scenario for the closed-shape and open-shape 3D objects in the CoppeliaSim simulator are shown in Figure. 5.5 and Figure. 5.6 respectively. On this basis, the operator is able to check the manipulator motion in a virtual environment prior to the execution of additive manufacturing operations in the real world in terms of robot motions and corresponding 3D printing actions.

Overall, the results of the simulation found clear support for the printability checking and prefabrication in robotic construction 3D printing applications. In fact, important and explicit information can be extracted from the obtained results regarding to the robot motion, trajectory constraints, process time, manufacturing cost, actions executed by the robotic platform and possible collisions between the robot and printed structure.

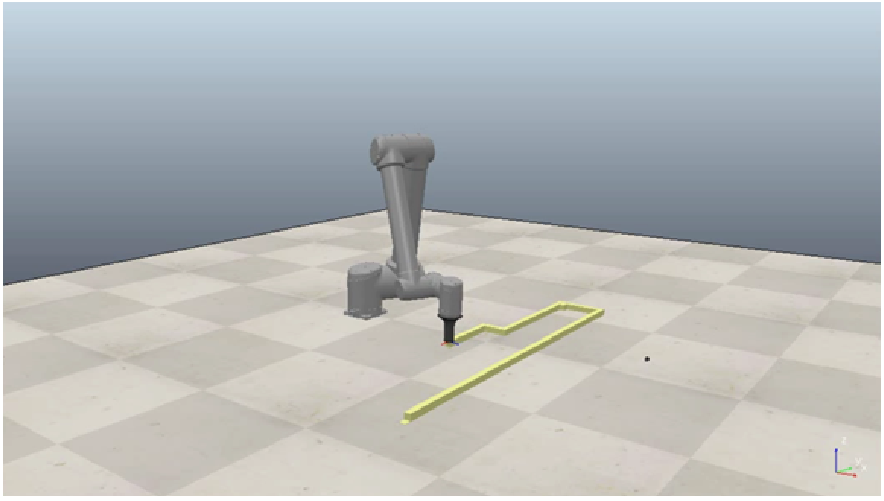
It is worth to point out that the real operation will have to be executed by using standard devices based on safety protocols, so as to have a secure and safe operation.

The generated plan for the 3D printing of the closed-shape and open-shape 3D objects are given in Appendix. B.

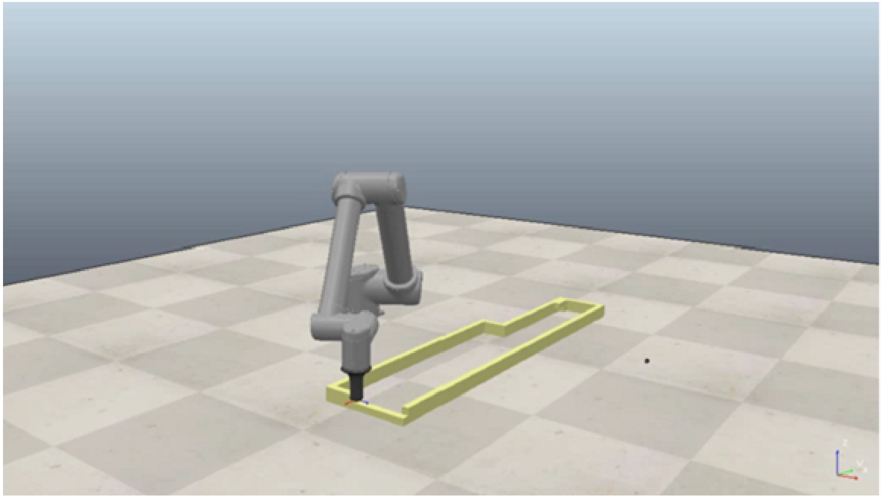
5.2 Evaluations

In order to evaluate more task-level actions and escalating the task level complexity of the proposed approach, the scalability tests [95; 96] have been carried out. The tests are implemented on fifteen 3D objects with different geometry specifications (printing scenario). In fact, we intend to examine the impact of increasing the number of the object edges and the complexity of the 3D printing process on the proposed AI planning system. A top view of objects used in the tests are illustrated in Figure 5.7 where each shape represents 3D objects with one, two and three layers.

All the tests are conducted on a computer with Intel i7-7500U CPU, 8 *GB* of RAM and Linux operating system (Ubuntu 20.04 LTS) by means of POPF (via ROSPlan) and Local search for Planning Graphs (LPG) [97; 98] planners. Correspondingly, the planners have been run 10 times to take into account the randomness associated with the employed heuristics. Subsequently, planners generate plans for the robotic concrete 3D printing operation of each target object and the execution time of each iteration has been recorded. The obtained results from these tests are shown graphically in Figure 5.8 where mean, minimum and maximum of solution times for different problem instances are shown through box plots. It can be clearly seen that when the number of links increase, planning time significantly increases as well, and thus the means. Furthermore, it can

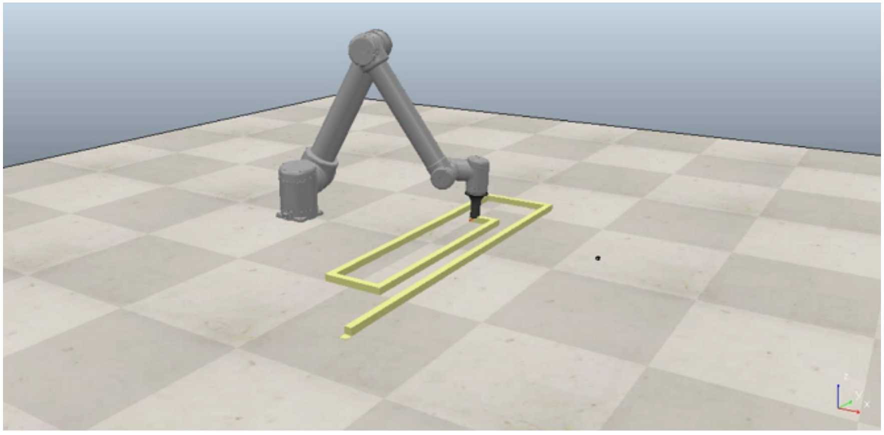


(a) Printing process of the first layer.

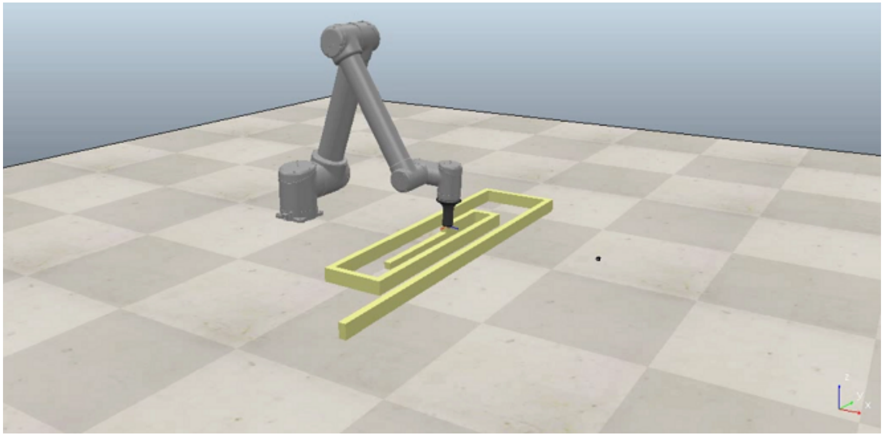


(b) Printing process of the second layer.

Figure 5.5: The simulation scenario of the robotic concrete 3D printing operation for the closed-shape 3D object in CoppeliaSim.



(a) Printing process of the first layer.



(b) Printing process of the second layer.

Figure 5.6: The simulation scenario of the robotic concrete 3D printing operation for the open-shape 3D object in CoppeliaSim.

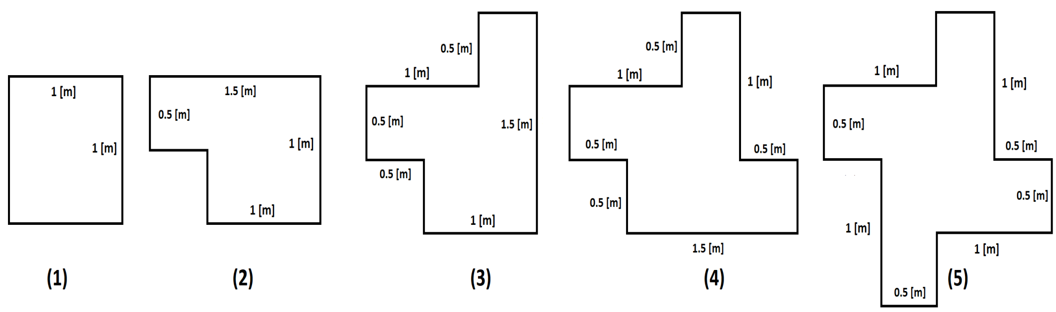


Figure 5.7: A top view of objects used in scalability tests. Each depicted shape represents 3D objects with one, two and three layers.

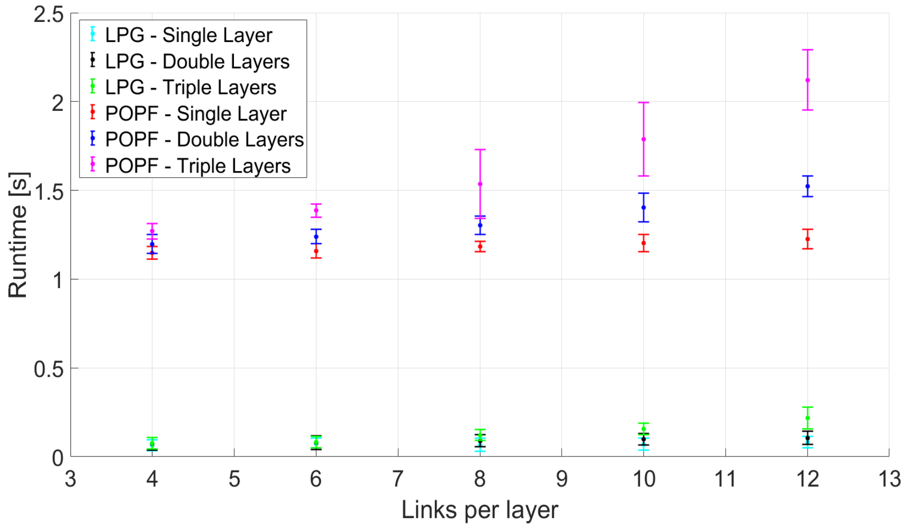


Figure 5.8: Execution times (runtime) for robotic concrete 3D printing scenarios of target objects including mean, minimum and maximum of solution times.

also be inferred that the LPG planner is rather faster than the POPF planner. It is worth noting that the AI plans are also evaluated and generated with the Temporal Fast Downward (TFD) planner [99], but the results are not reported in this manuscript due to the high calculation time by this planner.

Chapter 6

Conclusions

In this thesis, an application of the AI to robotic construction 3D printing systems is presented and described. Namely, AI planners expressed in PDDL 2.1 have been developed to obtain sequences of operations to be input to the control system of a robotic manipulator in order to perform specific tasks for creating spatial objects with different geometry specifications according to the requirements of printability checking and prefabrication in robotic construction 3D printing applications.

The AI techniques, including domain and problem files and generated plans, employed in this work have been described both in their theoretical approach and in their implementation. In continue, a robotic simulator framework has been built and explained so as to be able to test and evaluate the developed approach in a virtual environment by monitoring two concrete 3D printing processes within trajectories generated for the robot and 3D printing actions by the intelligent planners. The results of the simulation scenarios have been satisfying, in the sense that the concrete 3D printing operation can be effectively reproduced in a virtual environment by meeting all the essential requirements concerning printability checking and prefabrication. Moreover, the performance of the approach has been examined through scalability tests which show that incrementing edges and layers of a 3D object increases the execution time of the AI planner.

The present findings confirm the combination of 3D printing with AI can increase the performance of 3D printers through the reduction of errors and automation of production process by reproducing efficiently the robotic commands in a virtual environment. Furthermore, AI planning techniques can be understood as a promising solution for problems of prefabrication and printability checking in construction robotic technologies.

Further developments of the work will be aimed at implementing the concrete 3D printing operation in a real operating environment and using an existent industrial robotic manipulators, so as to get a working prototype of the whole additive manufacturing system, that will then be thoroughly tested.

Appendix A

Domain Model and Problem Description Files

An excerpt of the PDDL 2.1 domain file for a robotic concrete 3D printing processes, used in this thesis, is given as follows:

```
(define (domain concrete_printer)
(:requirements :typing :durative-actions :fluents)
(:types
    vertex
)
(:predicates
    (valid_move ?vtx_from - vertex ?vtx_to - vertex)
    (printed_link ?vtx_from - vertex ?vtx_to - vertex)
    (above_on ?vtx_from - vertex ?vtx_up - vertex)
    (visited ?vtx - vertex)
    (initial_vertex ?vtx - vertex)
    (nozzle_at ?vtx - vertex)
    (nozzle_is_open)
    (nozzle_is_close)
    (not_printed ?vtx_from - vertex ?vtx_to - vertex)
    (on_same_way ?vtx_from - vertex ?vtx_to - vertex)
    (not_on_same_way ?vtx_from - vertex ?vtx_to - vertex)
)
(:functions
    (layer_tickness)
    (nozzle_angle)
    (nozzle_height)
    (rest_time)
    (relocate_time)
    (change_layer)
```

```
    (link_length ?vtx_from - vertex ?vtx_to - vertex)
    (printing_velocity ?vtx_from - vertex ?vtx_to - vertex)
    (motion_angle ?vtx_from - vertex ?vtx_to - vertex)
    (link_height ?vtx_from - vertex ?vtx_to - vertex)
)
(:durative-action print
    :parameters (?from - vertex ?to - vertex)
    :duration (= ?duration (/ (link_length ?from ?to) (printing_velocity ?from ?to) ))
    :condition (and
        (at start (nozzle_at ?from))
        (at start (valid_move ?from  ?to))
        (at start (nozzle_is_open))
        (at start (not_printed ?from ?to))
        (at start (> (nozzle_height) (link_height ?from ?to)))
        (at start (on_same_way ?from ?to)))
    :effect (and
        (at end (not (nozzle_at ?from)))
        (at end (not (not_printed ?from ?to)))
        (at end (nozzle_at ?to))
        (at end (visited ?to))
        (at end (printed_link ?from ?to)))
)
(:action switch_off
    :parameters (?from - vertex ?to - vertex)
    :precondition (and
         (nozzle_at ?to)
         (valid_move ?from ?to)
         (visited ?from)
         (visited ?to)
         (printed_link ?from ?to)
         (on_same_way ?from ?to)
         (nozzle_is_open))
    :effect (and
         (not (nozzle_is_open))
         (nozzle_is_close))
)
(:action switch_on
    :parameters (?from - vertex ?to - vertex)
    :precondition (and
        (nozzle_at ?from)
        (valid_move ?from ?to)
        (visited ?from)
        (not_printed ?from ?to)
        (on_same_way ?from ?to)
        (nozzle_is_close))
    :effect (and
        (not (nozzle_is_close))
        (nozzle_is_open))
```

```
)
(:durative-action rotate_nozzle
    :parameters (?from - vertex ?to - vertex)
    :duration (= ?duration 1)
    :condition (and
        (at start (valid_move ?from ?to))
        (at start (nozzle_at ?from))
        (at start (visited ?from))
        (at start (> (nozzle_height) (link_height ?from ?to)))
        (at start (nozzle_is_close)))
    :effect (and
        (at end (on_same_way ?from ?to))
        (at end (assign (nozzle_angle) (motion_angle ?from ?to))))
)
(:durative-action go_to_init
    :parameters (?from - vertex ?to - vertex)
    :duration (= ?duration relocate_time)
    :condition (and
        (at start (nozzle_at ?from))
        (at start (valid_move ?from  ?to))
        (at start (nozzle_is_close))
        (at start (visited ?from))
        (at start (visited ?to))
        (at start (on_same_way ?from ?to)))
    :effect (and
        (at end (not (nozzle_at ?from)))
        (at end (nozzle_at ?to)))
)
(:durative-action waiting
    :parameters (?from - vertex ?to - vertex)
    :duration (= ?duration rest_time)
    :condition (and
        (at start (nozzle_at ?from))
        (at start (printed_link ?from ?to))
        (at start (initial_vertex ?from)))
    :effect (and
        (at end (assign (change_layer) 1)))
)
(:durative-action increase_nozzle_height
    :parameters (?from - vertex ?to - vertex ?up - vertex)
    :duration (= ?duration 1)
    :condition (and
        (at start (nozzle_at ?from))
        (at start (nozzle_is_close))
        (at start (above_on ?from ?up))
        (at start (> (change_layer) 0))
        (at start (initial_vertex ?from))
        (at start (printed_link ?from ?to))
```

```
)
    :effect (and
        (at end (increase (nozzle_height) (layer_tickness)))
        (at end (not (nozzle_at ?from)))
        (at end (nozzle_at ?up))
        (at end (assign (change_layer) 0))
        (at end (visited ?up)))
)
)
```

The PDDL 2.1 problem file created for the 3D printing of the closed-shape 3D object based on its geometry specifications and 3D printing necessities is given as follows:

```
(define (problem concrete_printer_task)
(:domain concrete_printer)
(:objects
    vtx00 - vertex
    vtx01 - vertex
    vtx02 - vertex
    vtx03 - vertex
    vtx04 - vertex
    vtx05 - vertex
    vtx10 - vertex
    vtx11 - vertex
    vtx12 - vertex
    vtx13 - vertex
    vtx14 - vertex
    vtx15 - vertex
)
(:init
    (valid_move vtx00 vtx01)
    (valid_move vtx01 vtx02)
    (valid_move vtx02 vtx03)
    (valid_move vtx03 vtx04)
    (valid_move vtx04 vtx05)
    (valid_move vtx05 vtx00)
    (valid_move vtx10 vtx11)
    (valid_move vtx11 vtx12)
    (valid_move vtx12 vtx13)
    (valid_move vtx13 vtx14)
    (valid_move vtx14 vtx15)
    (valid_move vtx15 vtx10)
    (not_printed vtx00 vtx01)
```

```
(not_printed vtx01 vtx02)
(not_printed vtx02 vtx03)
(not_printed vtx03 vtx04)
(not_printed vtx04 vtx05)
(not_printed vtx05 vtx00)
(not_printed vtx10 vtx11)
(not_printed vtx11 vtx12)
(not_printed vtx12 vtx13)
(not_printed vtx13 vtx14)
(not_printed vtx14 vtx15)
(not_printed vtx15 vtx10)
(on_same_way vtx00 vtx01)
(above_on vtx00 vtx10)
(visited vtx00)
(nozzle_at vtx00)
(nozzle_is_close)
(initial_vertex vtx00)
(initial_vertex vtx01)
(= (printing_velocity vtx00 vtx01) 0.08)
(= (printing_velocity vtx01 vtx02) 0.06)
(= (printing_velocity vtx02 vtx03) 0.08)
(= (printing_velocity vtx03 vtx04) 0.06)
(= (printing_velocity vtx04 vtx05) 0.08)
(= (printing_velocity vtx05 vtx00) 0.06)
(= (printing_velocity vtx10 vtx11) 0.08)
(= (printing_velocity vtx11 vtx12) 0.06)
(= (printing_velocity vtx12 vtx13) 0.08)
(= (printing_velocity vtx13 vtx14) 0.06)
(= (printing_velocity vtx14 vtx15) 0.08)
(= (printing_velocity vtx15 vtx10) 0.06)
(= (link_length vtx00 vtx01) 1.4)
(= (link_length vtx01 vtx02) 0.2)
(= (link_length vtx02 vtx03) 0.5)
(= (link_length vtx03 vtx04) 0.1)
(= (link_length vtx04 vtx05) 0.9)
(= (link_length vtx05 vtx00) 0.3)
(= (link_length vtx10 vtx11) 1.4)
(= (link_length vtx11 vtx12) 0.2)
(= (link_length vtx12 vtx13) 0.5)
(= (link_length vtx13 vtx14) 0.1)
(= (link_length vtx14 vtx15) 0.9)
(= (link_length vtx15 vtx10) 0.3)
(= (motion_angle vtx00 vtx01) 0)
(= (motion_angle vtx01 vtx02) 90)
(= (motion_angle vtx02 vtx03) 0)
(= (motion_angle vtx03 vtx04) 90)
(= (motion_angle vtx04 vtx05) 0)
(= (motion_angle vtx05 vtx00) 90)
(= (motion_angle vtx10 vtx11) 0)
```

```
    (= (motion_angle vtx11 vtx12) 90)
    (= (motion_angle vtx12 vtx13) 180)
    (= (motion_angle vtx13 vtx14) 90)
    (= (motion_angle vtx14 vtx15) 0)
    (= (motion_angle vtx15 vtx10) 90)
    (= (link_height vtx00 vtx01) 0.025)
    (= (link_height vtx01 vtx02) 0.025)
    (= (link_height vtx02 vtx03) 0.025)
    (= (link_height vtx03 vtx04) 0.025)
    (= (link_height vtx04 vtx05) 0.025)
    (= (link_height vtx05 vtx00) 0.025)
    (= (link_height vtx10 vtx11) 0.05)
    (= (link_height vtx11 vtx12) 0.05)
    (= (link_height vtx12 vtx13) 0.05)
    (= (link_height vtx13 vtx14) 0.05)
    (= (link_height vtx14 vtx15) 0.05)
    (= (link_height vtx15 vtx10) 0.05)
    (= (nozzle_height) 0.05)
    (= (nozzle_angle) 0)
    (= (layer_tickness) 0.025)
    (= (relocate_time) 0)
    (= (rest_time) 35)
    (= (change_layer) 0)
)
(:goal (and
    (nozzle_at vtx10)
    (nozzle_is_close)
    (printed_link vtx00 vtx01)
    (printed_link vtx01 vtx02)
    (printed_link vtx02 vtx03)
    (printed_link vtx03 vtx04)
    (printed_link vtx04 vtx05)
    (printed_link vtx05 vtx00)
    (printed_link vtx10 vtx11)
    (printed_link vtx11 vtx12)
    (printed_link vtx12 vtx13)
    (printed_link vtx13 vtx14)
    (printed_link vtx14 vtx15)
    (printed_link vtx15 vtx10)
))
)
```

In the same manner, the PDDL 2.1 problem file corresponding to the geometry specifications and printing conditions of the 3D open-shape object can be written as follows:

```
(define (problem concrete_printer_task)
(:domain concrete_printer)
(:objects
    vtx00 - vertex
    vtx01 - vertex
    vtx02 - vertex
    vtx03 - vertex
    vtx04 - vertex
    vtx05 - vertex
    vtx06 - vertex
    vtx07 - vertex
    vtx10 - vertex
    vtx11 - vertex
    vtx12 - vertex
    vtx13 - vertex
    vtx14 - vertex
    vtx15 - vertex
    vtx16 - vertex
    vtx17 - vertex
)
(:init
    (valid_move vtx00 vtx01)
    (valid_move vtx01 vtx02)
    (valid_move vtx02 vtx03)
    (valid_move vtx03 vtx04)
    (valid_move vtx04 vtx05)
    (valid_move vtx05 vtx06)
    (valid_move vtx06 vtx07)
    (valid_move vtx07 vtx00)
    (valid_move vtx10 vtx11)
    (valid_move vtx11 vtx12)
    (valid_move vtx12 vtx13)
    (valid_move vtx13 vtx14)
    (valid_move vtx14 vtx15)
    (valid_move vtx15 vtx16)
    (valid_move vtx16 vtx17)
    (valid_move vtx17 vtx10)
    (not_printed vtx00 vtx01)
    (not_printed vtx01 vtx02)
    (not_printed vtx02 vtx03)
    (not_printed vtx03 vtx04)
    (not_printed vtx04 vtx05)
    (not_printed vtx05 vtx06)
    (not_printed vtx06 vtx07)
    (not_printed vtx10 vtx11)
    (not_printed vtx11 vtx12)
    (not_printed vtx12 vtx13)
    (not_printed vtx13 vtx14)
    (not_printed vtx14 vtx15)
```

```
(not_printed vtx15 vtx16)
(not_printed vtx16 vtx17)
(on_same_way vtx00 vtx01)
(above_on vtx00 vtx10)
(visited vtx00)
(nozzle_at vtx00)
(nozzle_is_close)
(initial_vertex vtx00)
(initial_vertex vtx10)
(= (printing_velocity vtx00 vtx01) 0.05)
(= (printing_velocity vtx01 vtx02) 0.06)
(= (printing_velocity vtx02 vtx03) 0.05)
(= (printing_velocity vtx03 vtx04) 0.06)
(= (printing_velocity vtx04 vtx05) 0.05)
(= (printing_velocity vtx05 vtx06) 0.06)
(= (printing_velocity vtx06 vtx07) 0.05)
(= (printing_velocity vtx10 vtx11) 0.05)
(= (printing_velocity vtx11 vtx12) 0.05)
(= (printing_velocity vtx12 vtx13) 0.06)
(= (printing_velocity vtx13 vtx14) 0.05)
(= (printing_velocity vtx14 vtx15) 0.06)
(= (printing_velocity vtx15 vtx16) 0.05)
(= (printing_velocity vtx16 vtx17) 0.06)
(= (link_length vtx00 vtx01) 1.4)
(= (link_length vtx01 vtx02) 0.3)
(= (link_length vtx02 vtx03) 1.1)
(= (link_length vtx03 vtx04) 0.2)
(= (link_length vtx04 vtx05) 0.8)
(= (link_length vtx05 vtx06) 0.1)
(= (link_length vtx06 vtx07) 0.6)
(= (link_length vtx10 vtx11) 1.4)
(= (link_length vtx11 vtx12) 0.3)
(= (link_length vtx12 vtx13) 1.1)
(= (link_length vtx13 vtx14) 0.2)
(= (link_length vtx14 vtx15) 0.8)
(= (link_length vtx15 vtx16) 0.1)
(= (link_length vtx16 vtx17) 0.6)
(= (motion_angle vtx00 vtx01) 0)
(= (motion_angle vtx01 vtx02) 90)
(= (motion_angle vtx02 vtx03) 0)
(= (motion_angle vtx03 vtx04) 90)
(= (motion_angle vtx04 vtx05) 0)
(= (motion_angle vtx05 vtx06) 90)
(= (motion_angle vtx06 vtx07) 0)
(= (motion_angle vtx10 vtx11) 0)
(= (motion_angle vtx11 vtx12) 90)
(= (motion_angle vtx12 vtx13) 0)
(= (motion_angle vtx13 vtx14) 90)
(= (motion_angle vtx14 vtx15) 0)
```

```
    (= (motion_angle vtx15 vtx16) 90)
    (= (motion_angle vtx16 vtx17) 0)
    (= (link_height vtx00 vtx01) 0.025)
    (= (link_height vtx01 vtx02) 0.025)
    (= (link_height vtx02 vtx03) 0.025)
    (= (link_height vtx03 vtx04) 0.025)
    (= (link_height vtx04 vtx05) 0.025)
    (= (link_height vtx05 vtx06) 0.025)
    (= (link_height vtx06 vtx07) 0.025)
    (= (link_height vtx10 vtx11) 0.05)
    (= (link_height vtx11 vtx12) 0.05)
    (= (link_height vtx12 vtx13) 0.05)
    (= (link_height vtx13 vtx14) 0.05)
    (= (link_height vtx14 vtx15) 0.05)
    (= (link_height vtx15 vtx16) 0.05)
    (= (link_height vtx16 vtx17) 0.05)
    (= (nozzle_height) 0.05)
    (= (nozzle_angle) 0)
    (= (layer_tickness) 0.025)
    (= (relocate_time) 10)
    (= (rest_time) 30)
    (= (change_layer) 0)
)
(:goal (and
    (nozzle_at vtx10)
    (nozzle_is_close)
    (printed_link vtx00 vtx01)
    (printed_link vtx01 vtx02)
    (printed_link vtx02 vtx03)
    (printed_link vtx03 vtx04)
    (printed_link vtx04 vtx05)
    (printed_link vtx05 vtx06)
    (printed_link vtx06 vtx07)
    (printed_link vtx10 vtx11)
    (printed_link vtx11 vtx12)
    (printed_link vtx12 vtx13)
    (printed_link vtx13 vtx14)
    (printed_link vtx14 vtx15)
    (printed_link vtx15 vtx16)
    (printed_link vtx16 vtx17)
))
)
```

Appendix B

Generated Plan Files

The generated plan for the 3D printing of the closed-shape 3D object is given as:

```
0.000: (switch_on vtx00 vtx01) [0.001]
0.001: (print vtx00 vtx01) [17.500]
17.502: (switch_off vtx00 vtx01) [0.001]
17.503: (rotate_nozzle vtx01 vtx02) [0.001]
17.504: (switch_on vtx01 vtx02) [0.001]
17.505: (print vtx01 vtx02) [3.333]
20.839: (switch_off vtx01 vtx02) [0.001]
20.840: (rotate_nozzle vtx02 vtx03) [0.001]
20.841: (switch_on vtx02 vtx03) [0.001]
20.842: (print vtx02 vtx03) [6.250]
27.093: (switch_off vtx02 vtx03) [0.001]
27.094: (rotate_nozzle vtx03 vtx04) [0.001]
27.095: (switch_on vtx03 vtx04) [0.001]
27.096: (print vtx03 vtx04) [1.667]
28.764: (switch_off vtx03 vtx04) [0.001]
28.765: (rotate_nozzle vtx04 vtx05) [0.001]
28.766: (switch_on vtx04 vtx05) [0.001]
28.767: (print vtx04 vtx05) [11.250]
40.018: (switch_off vtx04 vtx05) [0.001]
40.019: (rotate_nozzle vtx05 vtx00) [0.001]
40.020: (switch_on vtx05 vtx00) [0.001]
40.021: (print vtx05 vtx00) [5.000]
45.022: (switch_off vtx05 vtx00) [0.001]
45.022: (waiting vtx00 vtx01) [35.000]
80.023: (increase_nozzle_height vtx00 vtx01 vtx10) [0.001]
80.024: (rotate_nozzle vtx10 vtx11) [0.001]
80.025: (switch_on vtx10 vtx11) [0.001]
80.026: (print vtx10 vtx11) [17.500]
97.527: (switch_off vtx10 vtx11) [0.001]
97.528: (rotate_nozzle vtx11 vtx12) [0.001]
```

```
97.529: (switch_on vtx11 vtx12) [0.001]
97.530: (print vtx11 vtx12) [3.333]
100.864: (switch_off vtx11 vtx12) [0.001]
100.865: (rotate_nozzle vtx12 vtx13) [0.001]
100.866: (switch_on vtx12 vtx13) [0.001]
100.867: (print vtx12 vtx13) [6.250]
107.118: (switch_off vtx12 vtx13) [0.001]
107.119: (rotate_nozzle vtx13 vtx14) [0.001]
107.120: (switch_on vtx13 vtx14) [0.001]
107.121: (print vtx13 vtx14) [1.667]
108.789: (switch_off vtx13 vtx14) [0.001]
108.790: (rotate_nozzle vtx14 vtx15) [0.001]
108.791: (switch_on vtx14 vtx15) [0.001]
108.792: (print vtx14 vtx15) [11.250]
120.043: (switch_off vtx14 vtx15) [0.001]
120.044: (rotate_nozzle vtx15 vtx10) [0.001]
120.045: (switch_on vtx15 vtx10) [0.001]
120.046: (print vtx15 vtx10) [5.000]
125.047: (switch_off vtx15 vtx10) [0.001]
```

The AI plan for the robotic concrete 3D printing of the open-shape object can be found as follows:

```
0.000: (switch_on vtx00 vtx01) [0.001]
0.001: (print vtx00 vtx01) [28.000]
28.002: (switch_off vtx00 vtx01) [0.001]
28.003: (rotate_nozzle vtx01 vtx02) [0.001]
28.004: (switch_on vtx01 vtx02) [0.001]
28.005: (print vtx01 vtx02) [5.000]
33.006: (switch_off vtx01 vtx02) [0.001]
33.007: (rotate_nozzle vtx02 vtx03) [0.001]
33.008: (switch_on vtx02 vtx03) [0.001]
33.009: (print vtx02 vtx03) [22.000]
55.010: (switch_off vtx02 vtx03) [0.001]
55.011: (rotate_nozzle vtx03 vtx04) [0.001]
55.012: (switch_on vtx03 vtx04) [0.001]
55.013: (print vtx03 vtx04) [3.333]
58.347: (switch_off vtx03 vtx04) [0.001]
58.348: (rotate_nozzle vtx04 vtx05) [0.001]
58.349: (switch_on vtx04 vtx05) [0.001]
58.350: (print vtx04 vtx05) [16.000]
74.351: (switch_off vtx04 vtx05) [0.001]
74.352: (rotate_nozzle vtx05 vtx06) [0.001]
74.353: (switch_on vtx05 vtx06) [0.001]
74.354: (print vtx05 vtx06) [1.667]
76.022: (switch_off vtx05 vtx06) [0.001]
76.023: (rotate_nozzle vtx06 vtx07) [0.001]
76.024: (switch_on vtx06 vtx07) [0.001]
76.025: (print vtx06 vtx07) [12.000]
```

```
88.026: (switch_off vtx06 vtx07) [0.001]
88.027: (go_to_init vtx07 vtx00) [10.000]
98.028: (waiting vtx00 vtx01) [30.000]
128.029: (increase_nozzle_height vtx00 vtx01 vtx10) [0.001]
128.030: (switch_on vtx10 vtx11) [0.001]
128.031: (print vtx10 vtx11) [28.000]
156.032: (switch_off vtx10 vtx11) [0.001]
156.033: (rotate_nozzle vtx11 vtx12) [0.001]
156.034: (switch_on vtx11 vtx12) [0.001]
156.035: (print vtx11 vtx12) [6.000]
162.036: (switch_off vtx11 vtx12) [0.001]
162.037: (rotate_nozzle vtx12 vtx13) [0.001]
162.038: (switch_on vtx12 vtx13) [0.001]
162.039: (print vtx12 vtx13) [18.333]
180.373: (switch_off vtx12 vtx13) [0.001]
180.374: (rotate_nozzle vtx13 vtx14) [0.001]
180.375: (switch_on vtx13 vtx14) [0.001]
180.376: (print vtx13 vtx14) [4.000]
184.377: (switch_off vtx13 vtx14) [0.001]
184.378: (rotate_nozzle vtx14 vtx15) [0.001]
184.379: (switch_on vtx14 vtx15) [0.001]
184.380: (print vtx14 vtx15) [13.333]
197.715: (switch_off vtx14 vtx15) [0.001]
197.716: (rotate_nozzle vtx15 vtx16) [0.001]
197.717: (switch_on vtx15 vtx16) [0.001]
197.718: (print vtx15 vtx16) [2.000]
199.719: (switch_off vtx15 vtx16) [0.001]
199.720: (rotate_nozzle vtx16 vtx17) [0.001]
199.721: (switch_on vtx16 vtx17) [0.001]
199.722: (print vtx16 vtx17) [10.000]
209.723: (switch_off vtx16 vtx17) [0.001]
```

References

[1] F. P. Bos, Z. Y. Ahmed, E. R. Jutinov, and T. A. Salet, "Experimental exploration of metal cable as reinforcement in 3d printed concrete," *Materials*, vol. 10, no. 11, p. 1314, 2017. vi, 2, 5

[2] E. S. Barjuei, P. Boscariol, R. Vidoni, and A. Gasparetto, "Robust control of three-dimensional compliant mechanisms," *Journal of Dynamic Systems, Measurement, and Control*, vol. 138, no. 10, p. 101009, 2016. 1

[3] H. Kwon, *Experimentation and analysis of contour crafting (CC) process using uncured ceramic materials*. University of Southern California, 2002. 1

[4] A. Siddika, M. A. A. Mamun, W. Ferdous, A. K. Saha, and R. Alyousef, "3d-printed concrete: Applications, performance, and challenges," *Journal of Sustainable Cement-Based Materials*, vol. 9, no. 3, pp. 127–164, 2020. 1

[5] F. Pieterse and A. L. Nel, "The advantages of 3d printing in undergraduate mechanical engineering research," in *2016 IEEE Global Engineering Education Conference (EDUCON)*, pp. 25–31, IEEE, 2016. 1

[6] D. D. Camacho, P. Clayton, W. J. O'Brien, C. Seepersad, M. Juenger, R. Ferron, and S. Salamone, "Applications of additive manufacturing in the construction industry–a forward-looking review," *Automation in construction*, vol. 89, pp. 110–119, 2018. 2

[7] S. Mishra, J. Narayan, K. Sandhu, and S. K. Dwivedy, "Successful stories of 3d printing in healthcare applications: A brief review," *Applications of 3D printing in Biomedical Engineering*, pp. 199–213, 2021. 2

[8] P. Rando and M. Ramaioli, "Food 3d printing: Effect of heat transfer on print stability of chocolate," *Journal of Food Engineering*, vol. 294, p. 110415, 2021. 2

[9] A. Madan, K. Yadav, H. Jain, H. Bhatt, and H. Bist, "Additive manufacturing in aerospace and defense industries: A review," *Additive Manufacturing in Aerospace and Defense Industries: A Review. Journal of Production Research & Management*, vol. 11, no. 1, pp. 1–5p, 2021. 2

[10] V. Serpooshan, M. Mahmoudi, D. A. Hu, J. B. Hu, and S. M. Wu, "Bioengineering cardiac constructs using 3d printing," *Journal of 3D printing in medicine*, vol. 1, no. 2, pp. 123–139, 2017. 2

[11] F. Bos, R. Wolfs, Z. Ahmed, and T. Salet, "Additive manufacturing of concrete in construction: potentials and challenges of 3d concrete printing," *Virtual and Physical Prototyping*, vol. 11, no. 3, pp. 209–225, 2016. 2

[12] P. Wu, J. Wang, and X. Wang, "A critical review of the use of 3-d printing in the construction industry," *Automation in Construction*, vol. 68, pp. 21–31, 2016. 2

[13] M. Ratiu, M. Prichici, D. Anton, and D. Negrau, "Compression testing of samples printed on delta and cartesian 3d printer," in *IOP Conference Series: Materials Science and Engineering*, vol. 1169, p. 012008, IOP Publishing, 2021. 2

[14] M. H. Ali, Y. Kuralbay, A. Aitmaganbet, and M. Kamal, "Design of a 6-dof

robot manipulator for 3d printed construction," *Materials Today: Proceedings*, 2021. 2

[15] D. Zhang, D. Zhou, G. Zhang, G. Shao, and L. Li, "3d printing lunar architecture with a novel cable-driven printer," *Acta Astronautica*, 2021. 2

[16] P. Pradhananga, M. ElZomor, and G. Santi Kasabdji, "Identifying the challenges to adopting robotics in the us construction industry," *Journal of Construction Engineering and Management*, vol. 147, no. 5, p. 05021003, 2021. 2

[17] C. Schubert, M. C. Van Langeveld, and L. A. Donoso, "Innovations in 3d printing: a 3d overview from optics to organs," *British Journal of Ophthalmology*, vol. 98, no. 2, pp. 159–161, 2014. 2

[18] R. Shaw, K. K. Maurya, and D. Maity, "3d concrete printing: A road map for future of automated construction in india," in *PREPARE@ u®— IEI Conferences*, 2021. 4

[19] J. Pegna, "Exploratory investigation of solid freeform construction," *Automation in construction*, vol. 5, no. 5, pp. 427–437, 1997. 4

[20] B. Khoshnevis, D. Hwang, K.-T. Yao, and Z. Yeh, "Mega-scale fabrication by contour crafting," *International Journal of Industrial and Systems Engineering*, vol. 1, no. 3, pp. 301–320, 2006. 4

[21] J. Xiao, G. Ji, Y. Zhang, G. Ma, V. Mechtcherine, J. Pan, L. Wang, T. Ding, Z. Duan, and S. Du, "Large-scale 3d printing concrete technology: Current status and future opportunities," *Cement and Concrete Composites*, vol. 122, p. 104115, 2021. 4

[22] A. Telea and A. Jalba, "Voxel-based assessment of printability of 3d shapes," in *International symposium on mathematical morphology and its applications to signal and image processing*, pp. 393–404, Springer, 2011. 4

[23] J. Yang, Y. Chen, W. Huang, and Y. Li, "Survey on artificial intelligence for additive manufacturing," in *2017 23rd International Conference on Automation and Computing (ICAC)*, pp. 1–6, IEEE, 2017. 5

[24] P. Kulkarni, A. Marsan, and D. Dutta, "A review of process planning techniques in layered manufacturing," *Rapid prototyping journal*, 2000. 5

[25] A. Barr and E. A. Feigenbaum, *The Handbook of Artificial Intelligence: Volume 2*, vol. 2. Butterworth-Heinemann, 2014. 5

[26] D. Vrontis, M. Christofi, V. Pereira, S. Tarba, A. Makrides, and E. Trichina, "Artificial intelligence, robotics, advanced technologies and human resource management: a systematic review," *The International Journal of Human Resource Management*, pp. 1–30, 2021. 5

[27] V. Vimal, T. Singh, S. Qamar, B. Nautiyal, K. Udham Singh, and A. Kumar, "Artificial intelligence-based novel scheme for location area planning in cellular networks," *Computational Intelligence*, vol. 37, no. 3, pp. 1338–1354, 2021. 5

[28] H. Fatemidokht, M. K. Rafsanjani, B. B. Gupta, and C.-H. Hsu, "Efficient and secure routing protocol based on artificial intelligence algorithms with uav-assisted for vehicular ad hoc networks in intelligent transportation systems," *IEEE Transactions on Intelligent Transportation Systems*, 2021. 5

[29] F. Zeng, C. Wang, and S. S. Ge, "A survey on visual navigation for artificial agents with deep reinforcement learning," *IEEE Access*, vol. 8, pp. 135426–135442, 2020. 5

[30] M.-H. Huang and R. T. Rust, "A strategic framework for artificial intelligence in marketing," *Journal of the Academy of Marketing Science*, vol. 49, no. 1, pp. 30–50, 2021. 5

[31] G. Briganti and O. Le Moine, "Artificial intelligence in medicine: today and tomorrow," *Frontiers in medicine*, vol. 7, p. 27, 2020. 5

[32] G. Zeba, M. Dabić, M. Čičak, T. Daim, and H. Yalcin, "Technology mining: Artificial intelligence in manufacturing," *Technological Forecasting and Social Change*, vol. 171, p. 120971, 2021. 5

[33] C. Aeronautiques, A. Howe, C. Knoblock, I. D. McDermott, A. Ram, M. Veloso, D. Weld, D. W. SRI, A. Barrett, D. Christianson, *et al.*, "Pddl—the planning domain definition language," *Technical Report, Tech. Rep.*, 1998. 6

[34] P. Haslum, N. Lipovetzky, D. Magazzeni, and C. Muise, "An introduction to the planning domain definition language," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 13, no. 2, pp. 1–187, 2019. 6, 26

[35] P. Gregory, "Pddl templating and custom reporting: Generating problems and processing plans," *Proceedings of the 30th ICAPS, Nancy, France*, pp. 14–19, 2020. 6

[36] W. Cheng and Y. Gao, "Using pddl to solve vehicle routing problems," in *International Conference on Intelligent Information Processing*, pp. 207–215, Springer, 2014. 6

[37] K. Tan, "The framework of combining artificial intelligence and construction 3d printing in civil engineering," in *MATEC web of conferences*, vol. 206, p. 01008, EDP Sciences, 2018. 8

[38] C. P. Chea, Y. Bai, X. Pan, M. Arashpour, and Y. Xie, "An integrated review of automation and robotic technologies for structural prefabrication and construction," *Transportation Safety and Environment*, vol. 2, no. 2, pp. 81–96, 2020. 8

[39] G. Carra, A. Argiolas, A. Bellissima, M. Niccolini, and M. Ragaglia, "Robotics in the construction industry: state of the art and future opportunities," in *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*, vol. 35, pp. 1–8, IAARC Publications, 2018. 12

[40] A. Perrot, D. Rangeard, and E. Courteille, "3d printing of earth-based materials: Processing aspects," *Construction and Building Materials*, vol. 172, pp. 670–676, 2018. 12, 14

[41] M. Hoffmann, S. Skibicki, P. Pankratow, A. Zieliński, M. Pajor, and M. Techman, "Automation in the construction of a 3d-printed concrete wall with the use of a lintel gripper," *Materials*, vol. 13, no. 8, p. 1800, 2020. 12, 13

[42] A. Perrot, D. Rangeard, and A. Pierre, "Structural built-up of cement-based materials used for 3d-printing extrusion techniques," *Materials and Structures*, vol. 49, no. 4, pp. 1213–1220, 2016. 13

[43] B. Nematollahi, M. Xia, and J. Sanjayan, "Current progress of 3d concrete printing technologies," in *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*, vol. 34, IAARC Publications, 2017. 13

[44] C. B. Costanzi, Z. Ahmed, H. R. Schipper, F. Bos, U. Knaack, and R. Wolfs, "3d printing concrete on temporary surfaces: The design and fabrication of

a concrete shell structure," *Automation in Construction*, vol. 94, pp. 395–404, 2018. 13

[45] J. H. Lim, Y. Weng, and Q.-C. Pham, "3d printing of curved concrete surfaces using adaptable membrane formwork," *Construction and Building Materials*, vol. 232, p. 117075, 2020. 13

[46] T. P. Tho and N. T. Thinh, "Using a cable-driven parallel robot with applications in 3d concrete printing," *Applied Sciences*, vol. 11, no. 2, p. 563, 2021. 13

[47] D. Asprone, F. Auricchio, C. Menna, and V. Mercuri, "3d printing of reinforced concrete elements: Technology and design approach," *Construction and Building Materials*, vol. 165, pp. 218–231, 2018. 13

[48] B. Panda, N. A. Noor Mohamed, S. C. Paul, G. Bhagath Singh, M. J. Tan, and B. Šavija, "The effect of material fresh properties and process parameters on buildability and interlayer adhesion of 3d printed concrete," *Materials*, vol. 12, no. 13, p. 2149, 2019. 13

[49] T. S. Rushing, P. B. Stynoski, L. A. Barna, G. K. Al-Chaar, J. F. Burroughs, J. D. Shannon, M. A. Kreiger, and M. P. Case, "Investigation of concrete mixtures for additive construction," in *3D Concrete Printing Technology*, pp. 137–160, Elsevier, 2019. 13

[50] C. T. Smyth, *Functional Design for 3D Printing: Designing 3d printed things for everyday use.* Clifford Smyth, 2015. 13

[51] K. D. Roehm and S. V. Madihally, "Bioprinted chitosan-gelatin thermosensitive hydrogels using an inexpensive 3d printer," *Biofabrication*, vol. 10, no. 1, p. 015002, 2017. 13

[52] Y. Cheng, X. Shi, X. Jiang, X. Wang, and H. Qin, "Printability of a cellulose derivative for extrusion-based 3d printing: the application on a biodegradable support material," *Frontiers in Materials*, vol. 7, p. 86, 2020. 13

[53] Z. Liu, M. Zhang, and Y. Ye, "Indirect prediction of 3d printability of mashed potatoes based on lf-nmr measurements," *Journal of Food Engineering*, vol. 287, p. 110137, 2020. 13

[54] S. Kyle, Z. M. Jessop, A. Al-Sabah, and I. S. Whitaker, "'printability'of candidate biomaterials for extrusion based 3d printing: state-of-the-art," *Advanced healthcare materials*, vol. 6, no. 16, p. 1700264, 2017. 13

[55] I. Fudos, M. Ntousia, V. Stamati, P. Charalampous, T. Kontodina, I. Kostavelis, D. Tzovaras, and L. Bilalis, "A characterization of 3d printability," *arXiv preprint arXiv:2010.12930*, 2020. 13

[56] F. W. Baumann, A. Sekulla, M. Hassler, B. Himpel, and M. Pfeil, "Trends of machine learning in additive manufacturing," *International Journal of Rapid Manufacturing*, vol. 7, no. 4, pp. 310–336, 2018. 13

[57] T. Lu, "Towards a fully automated 3d printability checker," in *2016 IEEE International Conference on Industrial Technology (ICIT)*, pp. 922–927, IEEE, 2016. 13

[58] B. M. Castro, M. Elbadawi, J. J. Ong, T. Pollard, Z. Song, S. Gaisford, G. Pérez, A. W. Basit, P. Cabalar, and A. Goyanes, "Machine learning predicts 3d printing performance of over 900 drug delivery systems," *Journal of Controlled Release*, vol. 337, pp. 530–545, 2021. 13

[59] J. Lee, S. J. Oh, S. H. An, W.-D. Kim, and S.-H. Kim, "Machine learning-based design strategy for 3d printable bioink: elastic modulus and yield

stress determine printability," *Biofabrication*, vol. 12, no. 3, p. 035018, 2020. 13

[60] Q. Zhu and J. Yan, "A mixed interface-capturing/interface-tracking formulation for thermal multi-phase flows with emphasis on metal additive manufacturing processes," *Computer Methods in Applied Mechanics and Engineering*, vol. 383, p. 113910, 2021. 14

[61] N. Roussel, "Rheological requirements for printable concretes," *Cement and Concrete Research*, vol. 112, pp. 76–85, 2018. 14

[62] S. Singh, S. kumar Sharma, and D. W. Rathod, "A review on process planning strategies and challenges of waam," *Materials Today: Proceedings*, 2021. 14

[63] J. Jiang, "A novel fabrication strategy for additive manufacturing processes," *Journal of Cleaner Production*, vol. 272, p. 122916, 2020. 14

[64] Z. Wu, T. M. Tucker, C. Nath, T. R. Kurfess, and R. W. Vuduc, "Step ring based 3d path planning via gpu simulation for subtractive 3d printing," in *International Manufacturing Science and Engineering Conference*, vol. 49903, p. V002T04A006, American Society of Mechanical Engineers, 2016. 14

[65] X. Lai and Z. Wei, "Slicing algorithm and partition scanning strategy for 3d printing based on gpu parallel computing," *Materials*, vol. 14, no. 15, p. 4297, 2021. 14

[66] K.-Y. Fok, C.-T. Cheng, and K. T. Chi, "A refinement process for nozzle path planning in 3d printing," in *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–4, IEEE, 2017. 14

[67] J. F. Arinez, Q. Chang, R. X. Gao, C. Xu, and J. Zhang, "Artificial intelligence in advanced manufacturing: Current status and future outlook," *Journal of Manufacturing Science and Engineering*, vol. 142, no. 11, p. 110804, 2020. 14

[68] A. N. Khobragade, N. Maheswari, and M. Sivagami, "Analyzing the housing rate in a real estate informative system: A prediction analysis," *Int. J. Civil Engine. Technol*, vol. 9, no. 5, pp. 1156–1164, 2018. 16

[69] C. Q. Poh, C. U. Ubeynarayana, and Y. M. Goh, "Safety leading indicators for construction sites: A machine learning approach," *Automation in construction*, vol. 93, pp. 375–386, 2018. 16

[70] C. Chen, D. T. Huy, L. K. Tiong, I.-M. Chen, and Y. Cai, "Optimal facility layout planning for agv-based modular prefabricated manufacturing system," *Automation in Construction*, vol. 98, pp. 310–321, 2019. 16

[71] A. Ajayi, L. Oyedele, H. Owolabi, O. Akinade, M. Bilal, J. M. Davila Delgado, and L. Akanbi, "Deep learning models for health and safety risk prediction in power infrastructure projects," *Risk Analysis*, vol. 40, no. 10, pp. 2019–2039, 2020. 16

[72] B. Chu, K. Jung, C.-S. Han, and D. Hong, "A survey of climbing robots: Locomotion and adhesion," *International journal of precision engineering and manufacturing*, vol. 11, no. 4, pp. 633–647, 2010. 16

[73] S. M. Sepasgozar, S. R. Davis, and M. Loosemore, "Dissemination practices of construction sites' technology vendors in technology exhibitions," *Journal of Management in Engineering*, vol. 34, no. 6, p. 04018038, 2018. 16

[74] J. M. D. Delgado, L. Oyedele, A. Ajayi, L. Akanbi, O. Akinade, M. Bilal, and H. Owolabi, "Robotics and automated systems in construction: Un-

derstanding industry-specific challenges for adoption," *Journal of Building Engineering*, vol. 26, p. 100868, 2019. 16

[75] E. Shahvand, M. H. Sebt, and M. T. Banki, "Developing fuzzy expert system for supplier and subcontractor evaluation in construction industry," *Scientia Iranica. Transaction A, Civil Engineering*, vol. 23, no. 3, p. 842, 2016. 16

[76] L. A. Akanbi, A. O. Oyedele, L. O. Oyedele, and R. O. Salami, "Deep learning model for demolition waste prediction in a circular economy," *Journal of Cleaner Production*, vol. 274, p. 122843, 2020. 16

[77] M. A. Kaleem and M. Khan, "Significance of additive manufacturingfor industry 4.0 with introduction of artificial intelligence in additive manufacturing regimes," in *2020 17th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, pp. 152–156, IEEE, 2020. 16

[78] L. Meng, B. McWilliams, W. Jarosinski, H.-Y. Park, Y.-G. Jung, J. Lee, and J. Zhang, "Machine learning in additive manufacturing: a review," *Jom*, vol. 72, no. 6, pp. 2363–2377, 2020. 16

[79] L. Huo and L. Baron, "The joint-limits and singularity avoidance in robotic welding," *Industrial Robot: An International Journal*, 2008. 16

[80] K. Zbiss, A. Kacem, M. Santillo, and A. Mohammadi, "Automatic collision-free trajectory generation for collaborative robotic car-painting," *IEEE Access*, 2022. 16

[81] A. M. Kabir, K. N. Kaipa, J. Marvel, and S. K. Gupta, "Automated planning for robotic cleaning using multiple setups and oscillatory tool motions," *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 3, pp. 1364–1377, 2017. 16

[82] A. M. Kabir, B. C. Shah, and S. K. Gupta, "Trajectory planning for manipulators operating in confined workspaces," in *2018 IEEE 14th International Conference on Automation Science and Engineering (CASE)*, pp. 84–91, IEEE, 2018. 16

[83] D. Ding, Z. Pan, D. Cuiuri, H. Li, and N. Larkin, "Adaptive path planning for wire-feed additive manufacturing using medial axis transformation," *Journal of Cleaner Production*, vol. 133, pp. 942–952, 2016. 16

[84] G. Q. Zhang, W. Mondesir, C. Martinez, X. Li, T. A. Fuhlbrigge, and H. Bheda, "Robotic additive manufacturing along curved surface—a step towards free-form fabrication," in *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 721–726, IEEE, 2015. 16

[85] G. Q. Zhang, A. Spaak, C. Martinez, D. T. Lasko, B. Zhang, and T. A. Fuhlbrigge, "Robotic additive manufacturing process simulation-towards design and analysis with building parameter in consideration," in *2016 IEEE International Conference on Automation Science and Engineering (CASE)*, pp. 609–613, IEEE, 2016. 17

[86] J. Huckaby, S. Vassos, and H. I. Christensen, "Planning with a task modeling framework in manufacturing robotics," in *2013 ieee/rsj international conference on intelligent robots and systems*, pp. 5787–5794, IEEE, 2013. 18

[87] D. Long and M. Fox, "Progress in ai planning research and applications," *UPGRADE: The European Journal for the Informatics Professional*, vol. 3, no. 5, pp. 10–25, 2002. 19

[88] V. Mechtcherine, F. P. Bos, A. Perrot, W. L. da Silva, V. Nerella, S. Fataei, R. J. Wolfs, M. Sonebi, and N. Roussel, "Extrusion-based additive manufac-

turing with cement-based materials–production steps, processes, and their underlying physics: A review," *Cement and Concrete Research*, vol. 132, p. 106037, 2020. 19

[89] S. Chitta, I. Sucan, and S. Cousins, "Moveit![ros topics]," *IEEE Robotics & Automation Magazine*, vol. 19, no. 1, pp. 18–19, 2012. 25

[90] E. Shojaei Barjuei, D. G. Caldwell, and J. Ortiz, "Bond graph modeling and kalman filter observer design for an industrial back-support exoskeleton," *Designs*, vol. 4, no. 4, p. 53, 2020. 33

[91] M. Cashmore, M. Fox, D. Long, D. Magazzeni, B. Ridder, A. Carrera, N. Palomeras, N. Hurtos, and M. Carreras, "Rosplan: Planning in the robot operating system," in *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 25, 2015. 35

[92] A. Coles, A. Coles, M. Fox, and D. Long, "Forward-chaining partial-order planning," in *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 20, 2010. 35

[93] S. Canbaz and G. Erdemir, "Performance analysis of real-time and general-purpose operating systems for path planning of the multi-robot systems," *International Journal of Electrical and Computer Engineering*, vol. 12, no. 1, p. 285, 2022. 37

[94] S. Rooban, S. D. Suraj, S. B. Vali, and N. Dhanush, "Coppeliasim: Adaptable modular robot and its different locomotions simulation framework," *Materials Today: Proceedings*, 2021. 38

[95] M. Cardellini, M. Maratea, M. Vallati, G. Boleto, and L. Oneto, "In-station train dispatching: a pddl+ planning approach," in *Proceedings of the In-*

ternational Conference on Automated Planning and Scheduling, vol. 31, pp. 450–458, 2021. 39

[96] R. Bertolucci, A. Capitanelli, M. Maratea, F. Mastrogiovanni, and M. Vallati, "Collaborative robotic manipulation: A use case of articulated objects in three-dimensions with gravity," in *2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 1167–1174, IEEE, 2020. 39

[97] F. Benzi, A. E. Gerevini, A. Saetti, and I. Serina, "On the use of landmarks in planner lpg," *Intelligenza Artificiale*, vol. 10, no. 2, pp. 97–111, 2016. 39

[98] M. Balduccini, D. Magazzeni, and M. Maratea, "Pddl+ planning via constraint answer set programming," *arXiv preprint arXiv:1609.00030*, 2016. 39

[99] P. Eyerich, R. Mattmüller, and G. Röger, "Using the context-enhanced additive heuristic for temporal and numeric planning," in *Nineteenth International Conference on Automated Planning and Scheduling*, 2009. 42

[100] P. C. Jackson, *Introduction to artificial intelligence.* Courier Dover Publications, 2019.

[101] E. Shojaei Barjuei and J. Ortiz, "A comprehensive performance comparison of linear quadratic regulator (lqr) controller, model predictive controller (mpc), h_∞ loop shaping and μ-synthesis on spatial compliant link-manipulators," *International Journal of Dynamics and Control*, vol. 9, pp. 121–140, 2021.

[102] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

[103] M. De Graaf, R. Aarts, B. Jonker, and J. Meijer, "Real-time seam tracking for robotic laser welding using trajectory-based control," *Control engineering practice*, vol. 18, no. 8, pp. 944–953, 2010.

[104] E. S. Barjuei, M. M. G. Ardakani, D. G. Caldwell, M. Sanguineti, and J. Ortiz, "Optimal selection of motors and transmissions in back-support exoskeleton applications," *IEEE Transactions on Medical Robotics and Bionics*, vol. 2, no. 3, pp. 320–330, 2020.

[105] T. Klinger, *Image processing with LabVIEW and IMAQ Vision*. Prentice Hall Professional, 2003.

[106] K.-S. Kwon and S. Ready, *Practical guide to machine vision software: an introduction with LabVIEW*. John Wiley & Sons, 2014.