Tampere University of Technology

Report on 3-month Internship

Implementing 3D printing concept in Construction Technology

Palash Halder | 267962

December 17, 2017

Abstract

3D Printing is one of the most promising developments of today. It has shown its potential in a wide range of disciplines, varying from the medical world to the food industry and from aerospace engineering to household uses. Unsurprisingly, the building industry has adopted this technique and aims to apply it on a larger scale. 3D Concrete printing results in a low cost and high speed construction method, which allows for a greater freedom in both architectural and structural design. Despite these clear benefits shown by a handful of pioneering companies and institutes spread around the world, the building industry is still behind in the development of 3D printing. This may be attributed to the unavailability of machines suited for printing in construction and the lack of knowledge in the field of structural behavior of the to-be-printed shapes and materials.

This internship report aims to find a road-map to incorporate the concept of 3D printing in construction using the readily available industrial robots. The internship was performed at **"Cazza"** a startup in 3D-printing construction industry. During the **3-month internship** phase research was done on subjects like robotics, civil and construction material, electrical and electronics and design.

As a robotics developer I was responsible for studying the capability of **ABB IRB 6650s** robot, initiating the installation and setup, troubleshooting communication problem between the robot and the **controller (ABB S4C plus)**. The next phase of my work included programming the robot in **ABB RobotStudio** and create simple trajectory. After that **ROS** was used to control the robot and perform simulation. During the end of internship, a simple algorithm was developed so that the robot can follow a trajectory that can build a structure as defined by a 3D-model.

Contents

List of Figures

1 Introduction

In almost every industrial sector, automation of processes has contributed to a faster and cheaper way of production. The sector of construction has not been automated in the same manner as other industrial sectors. Therefore large benefits could be obtained by further automating this process. On comparison with conventional manufacturing a automated process has the advantage of lesser production time, lesser cost and precise finishing.

3d printing, or rapid prototyping, was first introduced in 1987. Today there are many different techniques within this area but the basic principle is the same. It is an additive manufacturing technology, meaning that it adds on material layer by layer. The 3d printer prints thin cross sectional areas of the detail, one at the time, on top of each other and creates by this way the whole detail. The main advantages with this technology are that it can manufacture complex unstandardized details rapidly.

2 Earlier methods and Research

This section discusses some of the earlier methods used in construction 3D printing.

2.1 Contour Crafting

The origins of 3D concrete printing lie far back, as shown by Contour Crafting (CC), one of the oldest still existing techniques. The first publications on the technique by Khoshnevis (University of Southern California) can be found in 1998 and much progress has been made since. Contour Crafting is a method of layered manufacturing, using polymer, ceramic slurry, cement, and a variety of other materials and mixes to build large scale objects with smooth surface finish. The smoothness of the extrusion is achieved by constraining the extruded flow in the vertical and horizontal direction to trowel surfaces.

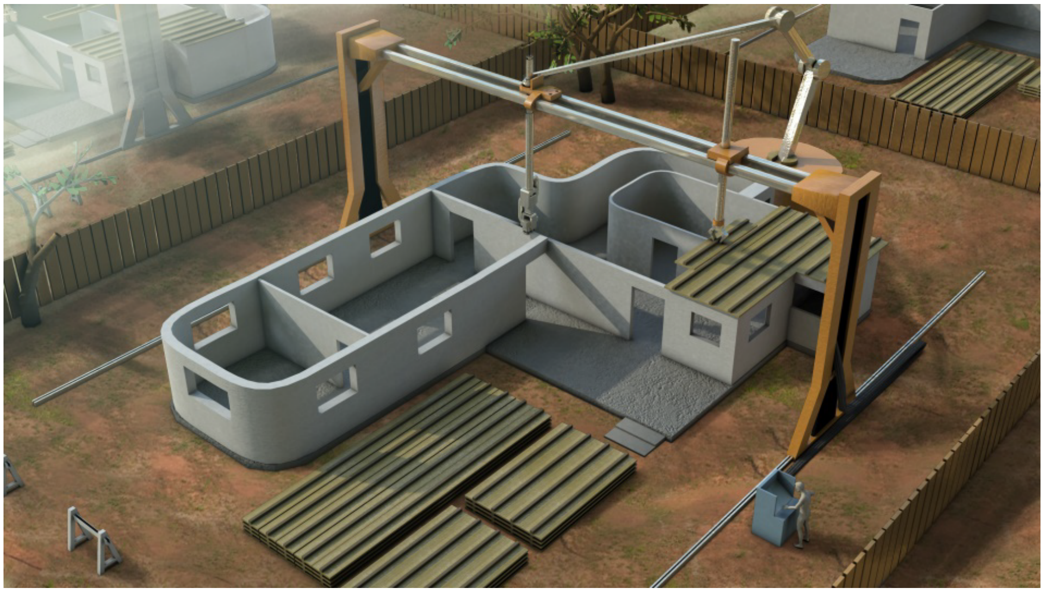


Figure 1: Contour Crafting by University of Southern California (contour-crafting.org, 2014

The American contractor Andrey Rudenko has founded TotalKustom (TK), a company that is able to print structures comparable to CC, but with a much smaller layer height (Figure 2). Similar developments are seen in China, where Yingchuang has constructed large parts of 3D printed buildings and assembles them on site (Figure 3). Their huge concrete printer (150 x 10 x 6.6 m) uses construction waste materials mixed with cement and glass fibre reinforcement (3Ders.org, 2014).

Figure 2: 3D printed walls by TotalKustom.

Figure 3: 3D printed walls by Yingchuang

2.2 Concrete Printing

Concrete Printing (CP), a technique developed at the Department of Civil and Building Engineering of the Loughborough University, is similar to CC in their extrusion based construction process. This technique however, has a smaller resolution of deposition to achieve higher 3-dimensional freedom, as it allows greater control of internal and external geometries. One of the by-products of this process is the ribbed surface finish, as the resulting surface is heavily dependent on the layer thickness.

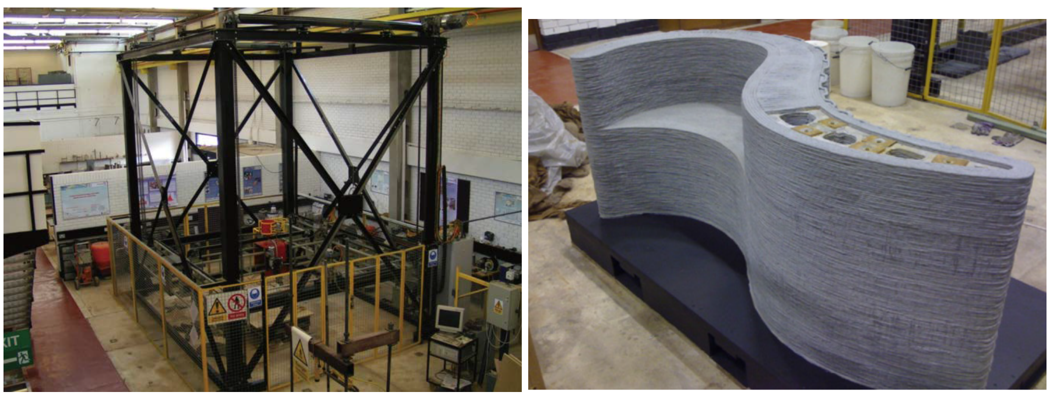


Figure 4: Concrete Printing by the University of Loughborough

3 Research Goal and Methodology

3.1 Research

The internship was based on finding ways to use a industrial robot in the field of construction technology. The readily available industrial robots helps in faster setup and reduces the research time. The readily available software packages like **RobotStudio** made writing simple programs easy and provided easier environment for simulation and testing.

With readily available **ROS** Packages namely **ROS industrial** and other packages like **ROS-moveit** and **ROS-Rviz** made it possible to use open source softwares in the development of this project. The internship succeeded in creating simulation of the industrial robot using **ABB RobotStudio virtual controller (IRC5)** and controlling the robot using **ROS**.

During the internship the following were developed:

1. Python code for creating robot trajectory from **"G-code"**.
2. Creating packages for **ABB IRB 6650s**.
3. Simulating trajectory followed by the robot in **ABB RobotStudio** and **RViz**.
4. Installing basic program in **ABB S4C plus** controller and performing basic motion of the manipulator.

3.2 Methodology

The methodology implemented in this project is shown below.

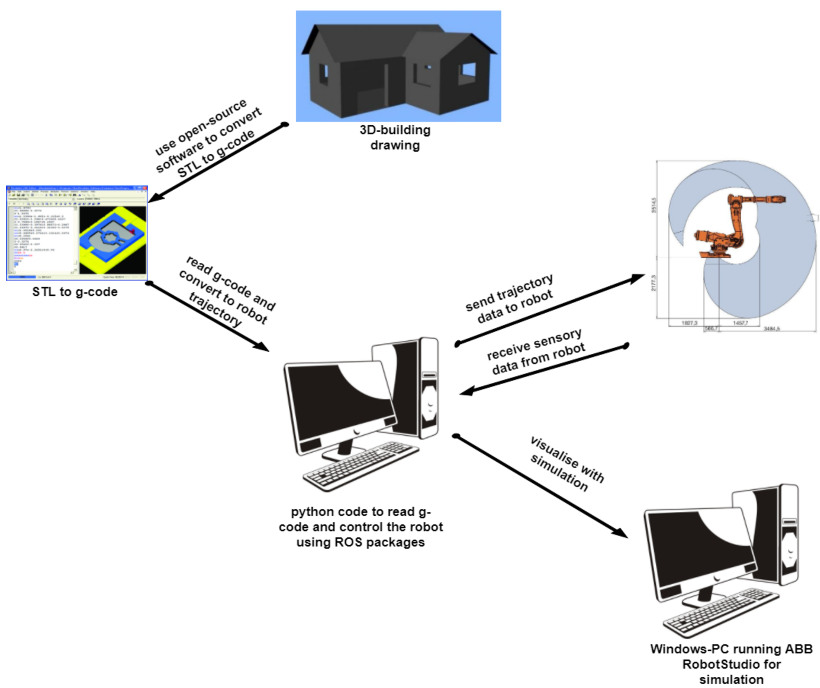


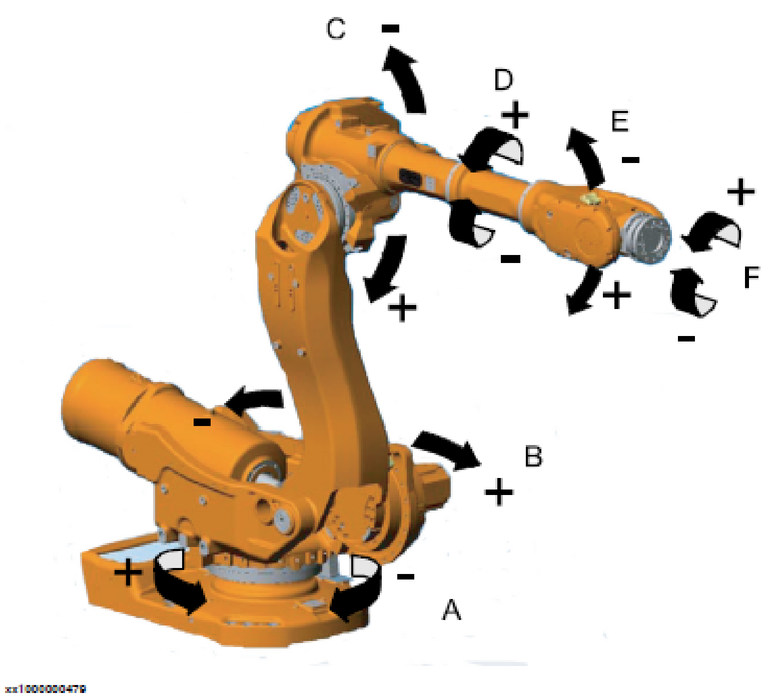
Figure 5: Methodology

4 Equipment and Software used

4.1 Equipment

4.1.1 Robot: ABB IRB 6650s

An industrial robot from ABB. It is available in three different types. During the internship the type with maximum handling capacity of 90kg and reach of 3.9m was used.



Pos	Description	Pos	Description
A	Axis 1	B	Axis 2
C	Axis 3	D	Axis 4
E	Axis 5	F	Axis 6

Figure 6: ABB IRB6650s with axis configuration

4.1.2 Controller: ABB S4C plus

The **ABB S4Cplu**s was used to control the robot using installed programs or by connecting with a PC providing the trajectory commands. It offers high performance and reliability, superior motion control, and extension communication possibilities. This controller has two built-in Ethernet channels for easier service and factory networking. It also presents shorter cycle times with QuickMove and greater precision and path following with TrueMove and utilizes the powerful **RAPID** programming language.

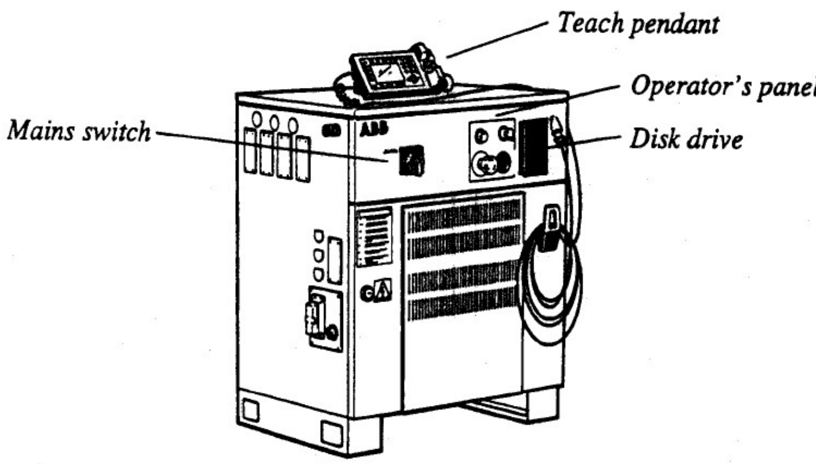


Figure 7: ABB S4C plus controller

4.2 Software

4.2.1 ABB RobotStudio

RobotStudio is built on the **ABB VirtualController**, an exact copy of the real software that runs ABB robots in production. This allows very realistic simulations to be performed, using real robot programs and configuration files identical to those used on the robots.

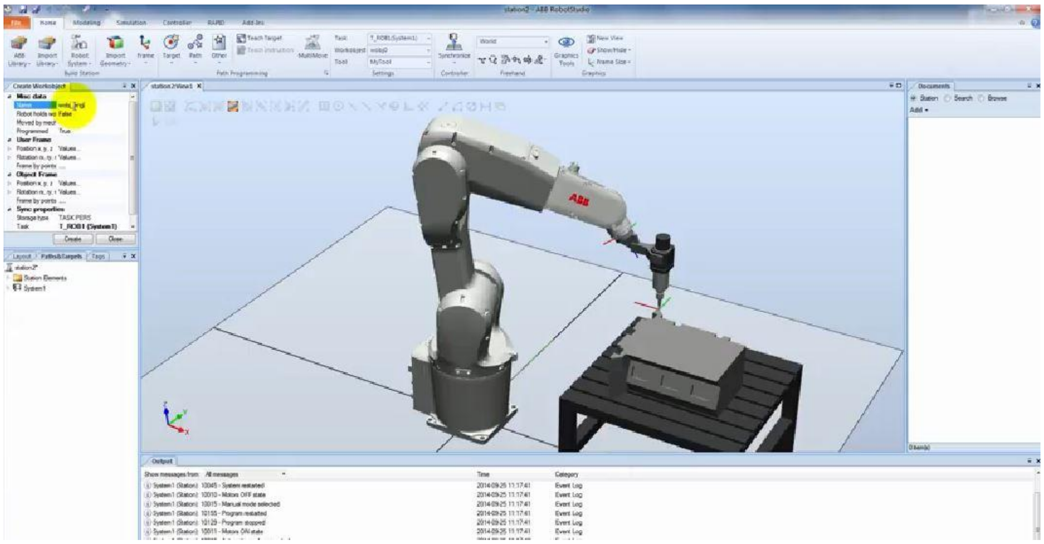


Figure 8: ABB RobotStudio

4.2.2 ROS

The **Robot Operating System (ROS)** is a set of software libraries and tools that helps to build robot applications. It is open source and includes drivers to state-of-the-art algorithms, and powerful developer tools.

4.2.3 ROS-industrial

ROS-Industrial is an Open Source robotics software project that extends the advanced capabilities of ROS to new manufacturing applications. It includes different packages for Robots from different manufactures. Also it includes other general packages like **industrial core**. The **ABB driver** and **ABB experimental** package was used to build various files required in this internship to be used with **ABB IRB6650s** robot.

4.2.4 ROS-moveit

MoveIt! is a state of the art software for mobile manipulation, incorporating the latest advances in motion planning, manipulation, 3D perception, kinematics, control and navigation. It provides an easy-to-use platform for developing advanced robotics applications, evaluating new robot designs and building integrated robotics product for industrial, commercial and Research and development.

4.2.5 ROS-rviz

rviz is a 3D visualizer for displaying sensor data and state information from **ROS**.

Python and **C++** was used as the language for the code.

5 Results

During the internship packages were developed which can be used for controlling the robot in hand. There is no separate package for ABB IRB 6650s and as such the following were developed taking idea from already developed packages in **ABB drivers**.

5.1 my_abb_irb6650s_support

This package was developed to include the support files for defining the robot in use ABB IRB 6650s. Some of the important files included in this package are:

- **joint_names_irb6650s_90_390.yaml** in the **"config"** folder, this file defines the name of the joints present in the robot. the jointlist should match the joint names specified in the robot's **URDF** file (described below). Order is not important. Here the standard joint_names were used.
- **irb6650s_90_390.xacro** in the **"urdf"** folder, this file calls for another file namely **"irb6650s_90_390_macro.xacro"** in the same folder. The **"irb6650s_90_390_macro.xacro"** defines the **links** and the **joints** of the robot.

The xacro files described above are used to create **"URDF"** file by using the command **"rosrun xacro xacro *model*.xacro"** where ***model*** is the name of the xacro file, here **irb6650s_90_390.xacro**.

Unified Robot Description Format (URDF), is the file for representing a robot model in **XML**. It includes various parameters like inertia, material as elements. **xacro** is used to construct shorter and more readable **XML** files by using macros that expand to larger **XML** expressions.

5.2 my_abb_irb6650s_moveit_config

This package was developed using **ROS-moveit** package and including the **URDF** files developed above. The main files in this folder are:

- **moveit_planning_execution.launch**, this is the main file to launch the robot system with **moveit**.

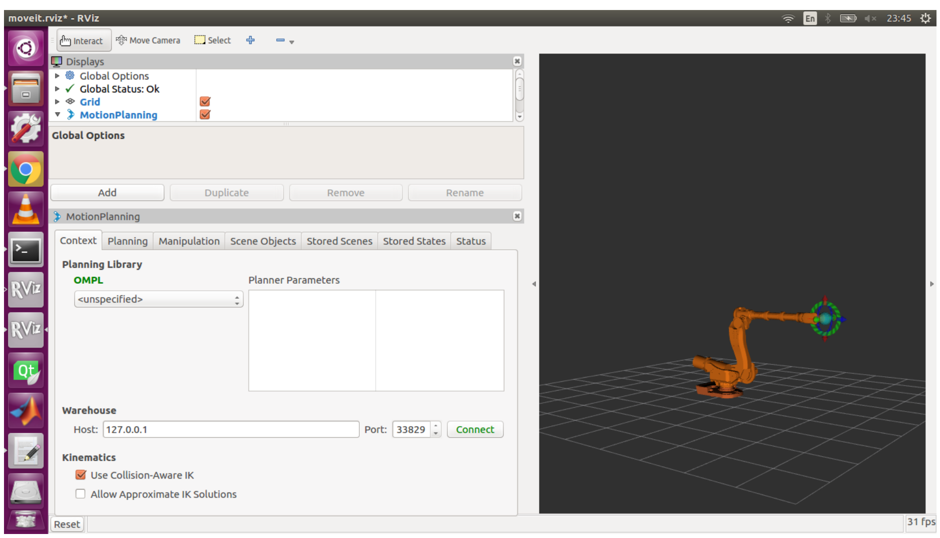


Figure 9: ABB IRB 6650s robot in rviz with moveit

On launching this file **"rviz"** is launched with additional **"MotionPlanning"** option as shown above. **moveit** takes care of motion planning and defines the trajectory of the robot tool-point. In the Planning tab **"Plan and execution"** can be visualized by setting valid goal points. There are various planner libraries from **moveit**. The **moveit** interface uses the planner and generates a trajectory for the robot to follow.

5.3 my_abb_irb6650s_python

As the name suggest this package contains the code for controlling the robot using **"Python"**. The main files in this folder are:

- **G-code files**. **"G-code"** is a language in which people tell computerized machine tools how to make something. G-code is used in CNC and 3D printing. In 3D-printing the 3D-drawing in STL format is converted to **"G-code"** and then sent to the printer. This folder contains the **"G-code"** of the 3D-drawing the robot has to print.

- **gcode_reader.py**, this file reads the **"G-code"** defined by the user and creates an array of way-points defining the trajectory of robot tool-point. The array returned by this function is in the format:

  [[Z1,[X1,Y1],[X2,Y2],[X3,Y3],[X4,Y4]], [Z2,[X1,Y1],[X2,Y2],[X3,Y3],...]]. As such it reads the **"G-code"** file and finds the coordinates to reach at a particular height i.e. **Z-axis**.

- **myABBrobot_moveit.py**, this file initiates ROS connection, performs motion planning, sends trajectory command to the robot. It converts the way-points received from **gcode_reader.py** and defines set of coordinates in the format **(posX, posY, posZ, orientX, orientY, orientZ, orientW)**.

  This code uses **moveit_Commander** which computes the trajectory of robot tool-point based on the way-points given. **moveit_commander** is capable of computing trajectory in both Cartesian space and joint space. **moveit_msgs** is used for sending and receiving **ROS-messages**.

The video link https://www.youtube.com/watch?v=OwALl9o2EOk&t=206s shows simulation of ABB IRB6640 robot. The packages generated during the internship are present in the link: https://github.com/palashTUT/cazza.git

6 Discussion

This internship provided an insight on how to implement **ROS-I** architecture into our industrial robot environment (with ABB IRB6650s robot). The **ROS** packages developed related to the robot modelling and configuring our robot with the **ROS-I** framework were **my_abb_irb6650s_support** and **my_abb_irb6650s_moveit_config**.These packages integrated our robot with the **ROS-I** architecture. These packages are developed by following the same standard structure that is followed in other packages of ABB robot models, present in **ROS-I** repository.

The other important idea was to use the available **"G-code"** generated for CNC or 3D printing and use it for generating robot trajectory. The package **my_abb_irb6650s_moveit_python** was developed to read **"G-code"** and give appropriate trajectory command to the robot.

At present the simulation was performed using **ABB RobotStudio** using **ABB IRC5** virtual controller. Due to unavailability of **IRC5** controller the ideas couldn't be implemented in the real robot system during the tenure of the internship. After getting successful result from simulation it is anticipated that the results will be similar in the real robot system.

7 Conclusion

The goal of creating a building by using an industrial robot was fulfilled via simulation. The accomplishment of the internship is to implement **ROS-I** architecture into our industrial robot environment (ABB IRB6650s) and bring together the concepts of 3D printing into the world of construction.

However, although this system exhibits the general idea to take industrial robots out of the cages and enable them to work in a construction environment. The adaptation and reaction feature of this system to read from **"G-code"** and create a basic 3D model will be accomplished with the development of other aspects of this project where the robot will be able to follow trajectory and deposit concrete to build a real structure.

It is anticipated with the development of other aspects of engineering this project will accomplish the task of printing a building.

8 References

- http://www.ros.org/
- http://wiki.ros.org/
- http://ros-industrial.github.io/industrial_training/
- http://moveit.ros.org/
- https://groups.google.com/forum/#!topic/swri-ros-pkg-dev/liFni3wMTCU
- http://new.abb.com/products/robotics/robotstudio
- http://new.abb.com/products/robotics/industrial-robots/irb-6650s
- https://library.e.abb.com/public/b0cf5a85b4b7e911c1257b4b00521469/EngS4Cplus_spec_Re
- https://www.robot-forum.com/robotforum/
- https://connect.bim360.autodesk.com/3d-printing-in-construction
- https://pure.tue.nl/ws/files/47026521/795000-1.pdf
- http://publications.lib.chalmers.se/records/fulltext/218418/218418.pdf
- https://dspace.cc.tut.fi/dpub/bitstream/handle/123456789/25016/Muneeb-Ul-Haq.pdf