University of Arkansas – CSCE Department
Capstone I – Final Proposal – Fall 2022

AMBOTS

Stanley Van, Michael Darden, Cassandra Nelson, Alvaro Becares Fernandez

Abstract

The manufacturing industry currently uses specialized machines and factories to produce products. This limits the flexibility of what can be produced out of one factory. New machines are often required for new products. AMBOTS is a company using existing or developing new technologies to break down manufacturing tasks into smaller and simpler ones. According to the AMBOTS website [1], "AMBOTS is an advanced manufacturing company with a focus on swarm 3D printing and assembly." These can then be automated and coordinated by robots. However, these robots need a way to communicate with one another. Our goal is to design and implement a communication protocol for different third-party robotic arms to communicate with each other to cooperatively complete a task.

We will solve this problem by breaking it down into manageable steps. First, we will select a group of robotic arms to start with and learn how to operate these arms in ROS, Moveit, and Gazebo. After this, we will create a program that can translate G-Code files into the robotic arms' native languages. Along with the translation program, another will be needed to allow the separate robots to communicate with each other. Finally, we will use both of the previous programs to perform a cooperative print using two separate robotic arms. When this goal is. achieved, the third-party robots will be able to communicate with each other, perform a task cooperatively, and complete a 3D print of a file they are given.

1.0 Problem

When a new product is designed, there is often a long and expensive production process. If one were to create a prototype of a car and work out all of the design flaws, one then must deal with the cost and issues of finding a way to produce the car on a massive scale. The answer to that problem in today's world is to design a factory around making that car. For several components of the car, specific machines have to be designed and built solely for that one car. This is an expensive and non-reusable solution that has become a massive challenge for almost all mass-produced products today. This also causes issues for some companies in the supply chain. It discriminates against the small guys. Because companies are so specialized on specific things, they have to rely on the services of others who may not want to provide their services as it may seem unprofitable to serve them. As co-founder and CTO of AMBOTS, Dr. Zhou put it, our civilization is built on manufacturing. Any production capability we lost is bad for our civilization. We cannot even reproduce the pyramids. We went to the moon and still have not been back.

Factories are built for the product. When demand for a product decreases, we lose the factory and the capability to produce the product. We cannot reproduce the sophisticated production process we have here on Earth on Mars. These problems lead to the overarching goal of AMBOTS being to create a general-purpose factory. A large obstacle to this goal is stationery and specific machines, and AMBOTS is the pioneer of the answer. In order to rid a factory of its need for stationery and specific machinery, the production process is broken down to the assignments of specific tasks to different robots. When production changes are needed, the robots can be assigned different tasks and can move around accordingly to their new tasks. This means that there is a need for an open ecosystem software package that can give instructions to these kinds of robots and also support a wide variety of third-party robots. It would be impractical to create a whole new network of robots for this, so instead, accommodating existing robots in the industry is a more feasible and economical goal.

2.0 Objective

The objective of this project is to perform cooperative 3D printing with Industrial Robotic Arms with ROS. This will be achieved by developing a universal printing interface such that it is possible to control different third-party robots by integrating them into our sponsor's platform and cooperate with other robots for manufacturing. We will do this by designing and implementing a communication protocol such that other third-party ROS-compatible robots can effectively talk with our robots over a local wireless network.

3.0 Background

3.1 Key Concepts

Swarm Manufacturing [2] is a new form of manufacturing developed for future factories. It is the employment of a swarm of different robots to manufacture products cooperatively on an open factory floor.

A 3D printer [3] is a machine where it constructs 3D models by added material together, typically layer by layer.

The Robot Operating System (ROS) [4] is a set of software libraries and tools that help. researchers and developers build and reuse code between robotic applications. ROS 2 has support for real time code and embedded systems. Any code file that utilizes ROS is called a node. Nodes have three ways of communicating. The first way is the publisher subscriber method. The second way is through services. The third way is through actions.

MoveIt [5] is an open-source ROS package. Its basic task is to provide the necessary trajectories for our robotic arms. This allows the robotic arms to move to the right locations.

There are two main functions which are creating a plan and sending a plan. Known obstacles can be added so that they can be avoided.

Gazebo [6] is an open-source 3D robotics simulator. It helps developers rapidly test algorithms and design robots in digital environments. This reduces the costs and safety risks of testing significantly. It can also simulate sensors. Sensors can use noise models for noise properties in the data the sensors return.

Degrees of Freedom (DOF) [7] is the number of independent variables that define the possible positions or motions of a mechanical system in space. The number of degrees of freedom is equal to the total number of independent displacements or aspects of motion (translational or rotational).

Linux [8] is an open-source operating system. It comes in many different distributions. The specific distribution we will be using Ubuntu 20.04. This is to allow the use of ROS 1 in case it is needed.

Computer-aided design (CAD) [9] is a design of real-world objects where computers were used in their creation. This type of design lets engineers create precise and quality models and prototypes. There is a system of software that allows designers to simulate, analyze, and optimize their designs.

A G-Code [10] file is a file that contains a series of instructions that 3D printers use to create a model in the real world. These instructions tell the printer where to move, how fast to move, and what path to follow.

A slicer [11] is software that takes a 3D object model and converts it to specific instructions for a 3D printer. The output of a slicer is a G-Code file. The slicer interprets the 3D models and finds a path for the 3D printer so that it can put down layers of material that will be in the shape of the model.

Python [12] is an interpreted, object-oriented, and high-level programming language. Python files end with the “py” file extension. It is simplistic in its syntax and comes with a standard library of many useful functions and data structures.

The Universal Robots UR10 [13] is the largest robot in the Universal Robots collaborative series. It has a payload up to 10 kg. It is very easy to set up and provides an accuracy of about 0.1mm. It is ROS compatible and can be simulated in both MoveIt and Gazebo.

The Kinova Gen3 Six Degrees of Freedom (6DOF) robotic arm [14] is a 6-axis robotic arm. It is ROS compatible and can be simulated in both MoveIt and Gazebo.

3.2 Related Work

The idea of swarm manufacturing is relatively new, and according to Additive Manufacturing [15], AMBOTS achieved the first end-to-end solution for cooperative 3d printing. Because of this there are not many other equivalent accomplishments. Something similar that can be compared is in the field of Additive Manufacturing and in the field of Swarm Robotics. We can look at these two concepts because when used together they create swarm 3d printing.

First, we will look at Additive Manufacturing. According to General Electric [16], Additive Manufacturing is the use of CAD software or 3D scanners to deposit material in precise shapes layer by layer. One company that utilizes this concept is 3D Systems [17]. The SLA 750 is one of their most recent developments. It is a "High-speed stereolithography solution for production manufacturing." What this means is it is a large end-to-end manufacturing machine. It uses dual lasers and photopolymer materials. Even with these improvements to a typical 3D printer, there is still the limitation of size. Because it is a single machine, the product produced must still fit inside the machine is the designated area. This defers from AMBOTS as having mobile robotic arms in swarm printing, size is not limited to inside one specific machine.

Next, we can look at Swarm Robotics. According to Scholarpedia [18], swarm robotics is the design of groups of robotics that operate without any interference of external infrastructure or centralized control. Robot swarms are self-organizing. One company that utilizes this concept is Unbox Robotics [19]. Unbox Robotics uses swarm intelligence to improve the throughput and sorting process. Although this is helpful in the manufacturing industry, it is not manufacturing the products themselves. It is just organizing them after the fact.

Finally, we will look at Rosotics [20]. Rosotics is the closest comparison to what AMBOTS is working on. While AMBOTS is working on swarm 3d printing for end-to-end manufacturing of a product, Rosotics is working on Rapid Induction Printing for metal additive manufacturing. According to an article from NASA's startup series [21], Rosotics is "a pioneer of swarm robotics". Although most of their work is hard to find specifics on, from this article we can assume that they use swarm intelligence in their approach. Even knowing this, they are mostly focusing on Rapid Induction Printing, rather than the standardization of communication between third party robotic arms as our project will be focusing on.

4.0 Approach/Design

4.1 Requirements and Use Cases

The requirements for this project will include the ability to input a G-Code file to the machine and have two different robots simulate the 3D printing of the object in the G-Code file. This code should be general and be able to work on multiple different brands of robotic arm. We are looking for universality. There is the potential to add a slicer so that an object from CAD software can be directly inputted without the prior translation to G-Code. We will begin by using an open-source slicer to do this but have the stretch goal of designing a slicer ourselves.

The main usage of this end goal will be mechanical engineers at AMBOTS. It will be controlled via command line interface and will only be able to be ran through Ubuntu. Because of this, it will require the user to have some basic knowledge of Linux along with using a command line. There is also the potential to transition this to use a Graphical User Interface (GUI). This could allow for a wider range of users as they wouldn't have to have as much background knowledge to use the application. The main limitation of this currently is the required process of setting up the machine beforehand. Beyond needing a Linux operating system, ROS, MoveIt, the robotic arms you want to use, and Gazebo all need to be installed and setup. We hope that our documentation (and potential bash files) can ease this process, but it all requires understanding how to download things via a command line.

4.2 High Level Architecture

The basic high-level architecture will take in an input of a collection of CAD files. The output will be a 3D printed version of the object that the CAD files represent.

There will be a command line interface to interact with the arms. This interface will receive an input directory of where the CAD files are located at. This will allow us or a future team of developers to easily implement a graphical user interface. The interface will run the slicer on all of the files to produce a collection of G-Code files.

There will be multiple ROS arm interfaces. They will receive an input of an array of G-Codes. These arrays will be produced from a G-Code file reader. The interfaces will have ROS nodes that communicate with each other to communicate information about their own printing tasks so that other arms can or cannot begin their own printing. The ROS interfaces will interpret the G-Code files and send the corresponding arm movement commands to their specific arms.

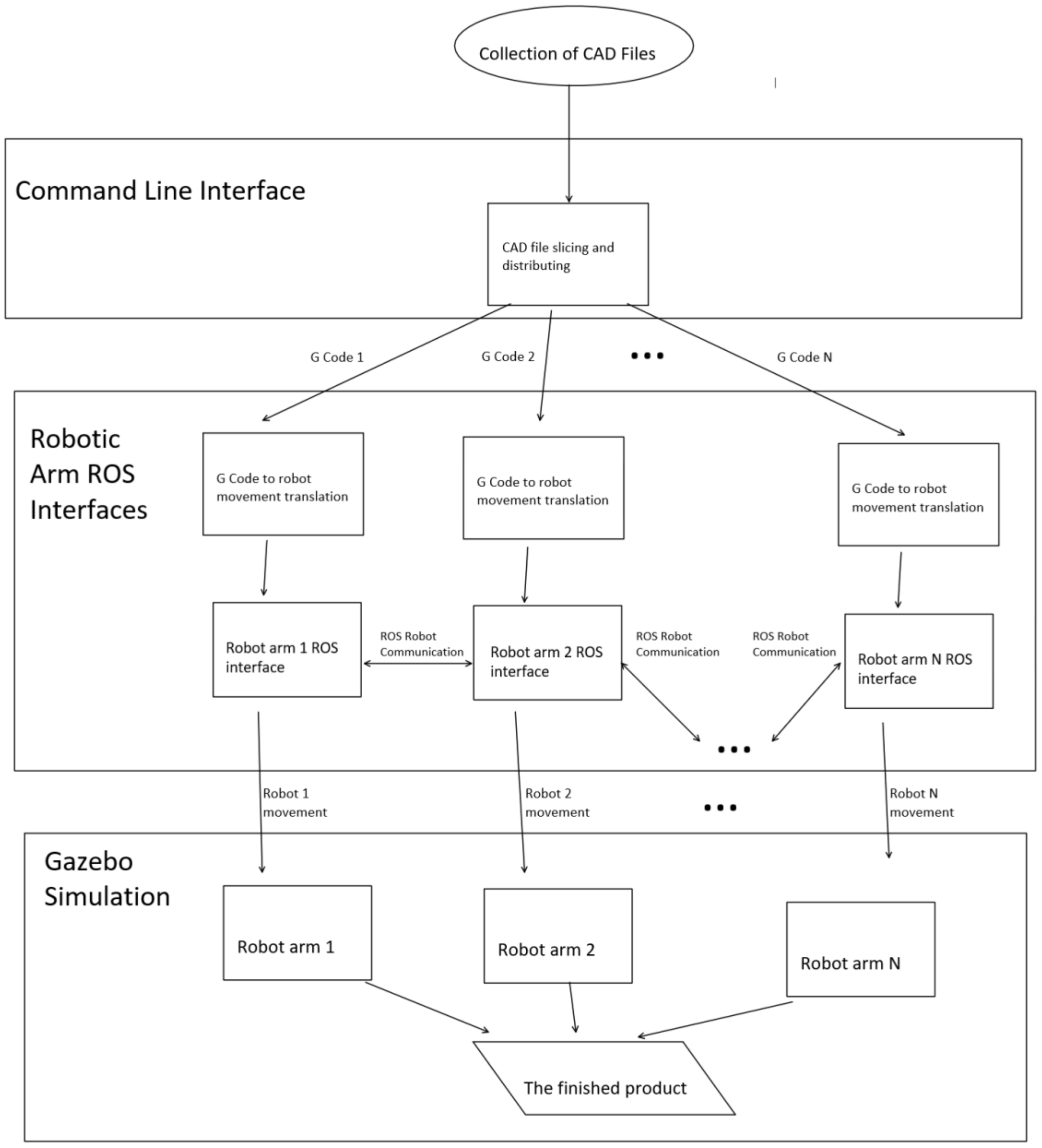
The ROS Communication interface will have two parameters. The first parameter will be the ID of the robot arm. The second parameter will be the communication command that will be sent. There will be 4 commands: begin, stop, continue, and home. The begin command will tell the robotic arm to start printing its received G-Code. The stop command will tell the robotic arm to stop its current printing assignment. The continue command will tell the robotic arm to continue its current printing assignment. The home command will tell the robotic arm to return to its home position.

The ROS interfaces will then communicate with the Gazebo simulation. This will display and simulate the robotic arm movements and 3D printing.

Proposed command interface:

```
python executeArmPrinting ./gCodeFolder
```

High Level Design Diagram:



4.3 Risks

Risk	Risk Reduction
Network Security	Using an isolated network that is not connected to the internet along with utilizing encryption and/or a firewall.
Physical Safety	Practice general lab safety and initially everything will be done via simulation.

4.4 Tasks

1. Research ROS, MoveIt, and Gazebo. When researching, a decision should be made on which version of ROS will work best with this specific project. This will help in the decision made in the second task.

2. Research Robotic arm options and select two arms of different brands. This decision should consider the version of ROS chosen. The robotic arm should support the utilization of gazebo and should preferably have vast documentation to make the learning process smooth.

3. Document set up of ROS, MoveIt, the robotic arms, and the Ubuntu environment. The version of Ubuntu used will depend on the version of ROS. This setup will take some time as each robotic arm has a different launch process. Each member of the team should have both robotic arms simulated on their machine once this step is done. The documentation will be organized in a GitLab for future usage.

4. Research how each of the chosen robotic arms receive commands. In order to successfully implement communication and printing, the robotic arms need to be better understood. Each brand of robotic arm has its own native language. We will research how the individual arms receive movement commands and how they utilize MoveIt to calculate the trajectories necessary to end at given positions.

5. Simulate basic movement of robotic arm choices in gazebo. This will include looking into the launch file (typically launch.py) of each arm. To verify that we have setup the robotic arms correctly we will simulate them in gazebo. Initially in step three they will be simulated in RViz [22] as that is what the setup instructions have examples for when using ROS and Moveit.

6. Research G-Code. An object that is to be printed will be provided via a .stl file. This .stl file will be imported into a slicer and a G-Code file will be exported.

7. Download and use slicer to create G-Code file. We will use PrusaSlicer [23], which is an open-source slicer that can be run via command line. A batch file that takes in an .stl file will then be created for easier use.

8. Pipe G-Code files to the robotic arms by creating a program to translate from G-Code file to robotic arms' native language. The G-Code file will be parsed, and the coordinates will be

extracted. These coordinates will then be translated into the robotics arms' native language for each robotic arm respectively.

9. Simulate the G-Code movement of robotic arm choices in gazebo using basic G-Code files. This is the process to make sure the program created in step 8 works. The program can then be debugged and improved upon.

10. Find and set up a concrete or material extruder for both robotic arms. Preferably a concrete extruder will be found, but because this is a simulation and not an actualized project, a generic material extruder is acceptable replacement.

11. Simulate printing using the extruder in gazebo. While the robotic arm is moving using the coordinates from the G-Code file, the extruder should begin simulating extruding. This is to prepare for the collaborative printing process. This should be done with both robotic arms.

12. Run both robotic arms in the same environment. Up until this point the robotic arms will be run in separate environments. Although on the same machine they will not be run in the same instance. This will be a process of figuring out how to have the robots running simultaneously and will allow for communication to happen in future steps.

13. Use ROS to communicate between two arms (One moves then tells the other to begin moving). In order to all for collaboration between the robotic arms, ROS will be used.

14. Simulate simultaneous printing with the robotic arm choices in gazebo using ROS communication. This last task is the culmination of all the previous tasks. The extruders simulated in task 11 will be used to simulate the printing process and the communication in tasks 13 will be used in the collaboration between robotic arms.

4.5 Schedule

Tasks	Dates
1. Research ROS, MoveIt, and Gazebo	11/07 - 11/14
2. Research Robotic arm options and select two arms of different brands.	11/14 - 11/21
3. Document set up of ROS, MoveIt, the robotic arms, and the Ubuntu environment	11/21 - 12/05
4. Research how each arm receives commands.	12/05 - 12/12
5. Simulate basic movement of robotic arm choices in gazebo	12/12 - 12/26
6. Research G-Code	12/26 - 01/02
7. Download and use slicer to create G-Code file.	01/02 - 01/09

8. Pipe G-Code files to the first robotic arm by creating a program to translate from a G-Code file to robotic arms' native language	01/09 - 01/23
9. Pipe G-Code files to the second robotic arm by creating a program to translate from a G-Code file to robotic arms' native language	01/23 - 02/06
10. Simulate the G-Code movement of robotic arm choices in gazebo using basic G-Code files.	02/06 - 02/13
11. Find and setup a concrete or material extruder for each robotic arm.	02/13 - 02/20
12. Simulate printing using the extruder in gazebo.	02/20 - 03/06
13. Run both robotic arms in the same environment.	03/06 - 03/20
14. Use ROS to communicate between the two robotic arms.	03/20 - 04/03
15. Simulate simultaneous printing with the robotic arm choices in gazebo using ROS communication.	04/03 - 04/24

4.6 Deliverables

- Research: Throughout the development process, research is done on the different software that is to be used. The research will be collected into one coherent document for. better understanding of what was learned.
- Documentation: With all the code written, there will be documentation explaining how it works, and why choices were made.
- Software package for ROS2 robotic arm control: This is the collection of software that is used and/or created to complete the goal of simultaneous robotic arm 3D printing.
- Tutorial for setting up and working with the software package: To simplify the process, a complete tutorial will be created for not only the setup of the machine to run the necessary programs, but also the process of running the software produced through this project.
- Final zip file: The final zip file required for the class including a report summarizing the process and implementation of the project along with all of the code written.

5.0 Key Personnel

Stanley Van – Van is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has completed the relevant courses Software Engineering, Database Management Systems, Programming Paradigms, Operating Systems, and Computer Networks. This student will be responsible for piping the G-Code files to the robots along with any other tasks that need extra assistance.

Cassandra Nelson– Nelson is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. She has completed relevant courses up to and including Database Management, Programming Paradigms, and Software Engineering. She held a C-Unix Programmer Analyst Intern position, and currently works as a Computer Support Assistant at the CORD on the University of Arkansas campus. Nelson will be responsible for documentation and any scripting required for setting up environments along with any other tasks that need extra assistance.

Michael Darden– Darden is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has completed relevant courses such as Programming Foundations one and two, Programming Paradigms, Computer Organization, Software Engineering, Computer Networks, and Discrete Mathematics. He has interned at Clear C2 for the summers of 2020, 2021, and 2022. Clear C2 is a web development company that creates costumer recourse management tools for other businesses. This student will be responsible for setting up the environments and simulations of the robots. This includes development in ROS to adapt the environments to support the different types of robots.

Alvaro Becares Fernandez– Becares is a senior Telematics Engineering major in the Telematics Engineering Department at the University Carlos III of Madrid, currently an exchange student at the University of Arkansas. He has completed relevant courses such as Artificial Intelligence, Big Data Analytics and Management. Becares will be responsible for compiling research done along with any other tasks that need extra assistance.

Dr. Wenchao Zhou– Dr. Zhou is an associate professor in the Mechanical Engineering department at the University of Arkansas. He is the co-founder and Chief Technology Officer of AMBOTS. After receiving his PhD in Mechanical Engineering from the Georgia Institute of Technology, he has participated in research that led to the publication on swarm manufacturing cited previously in this proposal [2].

Zachary Hyden – Hyden received a bachelor's degree in mechanical engineering from the University of Arkansas in 2019. He is now the Chief Mechanical Engineer at AMBOTS. During his undergraduate education he was an Additive Manufacturing Researcher in the AM^3 lab at the University of Arkansas.

6.0 Facilities and Equipment

Because this project will be developed using simulation software, there will not be any facility usage required. The funding required to acquire the robotic arms is estimated to not arrive in time within the time period of the development and completion of the project which is another reason that facility usage is not required. As for equipment, each member will be using a computer with Ubuntu installed and running in a virtual machine or natively as the operating system. Although there is a potential for this project to move to the physical lab, initially everything will be developed through simulation. It is then up to the sponsor whether or not to actualize our solution in their lab.

7.0 References

[1] AMBOTS, https://www.ambots.net/

[2] Poudel, L., Marques, L. G., Williams, R. A., Hyden, Z., Guerra, P., Fowler, O.L., Sha, Z., and Zhou, W. (February 16, 2022). "Toward Swarm Manufacturing: Architecting a Cooperative 3D Printing System." ASME. J. Manuf. Sci. Eng. August 2022; 144(8): 081004. https://doi.org/10.1115/1.4053681

[3] 3D printing, https://en.wikipedia.org/wiki/3D_printing

[4] ROS, https://www.ros.org/

[5] MoveIt, https://moveit.ros.org/

[6] Gazebo, https://gazebosim.org/home

[7] Investopedia, https://www.investopedia.com/terms/d/degrees-of-freedom.asp

[8] Linux, https://www.linux.com/what-is-linux/

[9] TechTarget, https://www.techtarget.com/whatis/definition/CAD-computer-aided-design

[10] G-code, https://en.wikipedia.org/wiki/G-code

[11] Slicer (3D printing), https://en.wikipedia.org/wiki/Slicer_(3D_printing)

[12] What is Python? Executive Summary, https://www.python.org/doc/essays/blurb/

[13] UR10, https://wiredworkers.io/product/ur10/

[14] Kinova Robotics, https://www.kinovarobotics.com/product/gen3-robots

[15] Additive Manufacturing, https://www.additivemanufacturing.media/articles/robots-assemble-a-new-path-to-autonomous-mobile-3d-printing

[16] General Electric, https://www.ge.com/additive/additive-manufacturing

[17] 3D Systems, https://www.3dsystems.com/3d-printers/sla-750

[18] Scholarpedia, http://www.scholarpedia.org/article/Swarm_robotics

[19] Unbox Robotics, https://unboxrobotics.com/

[20] Rosotics, https://www.rosotics.com/

[21] NASA, https://technology.nasa.gov/virtual-event/startup-nasa-series-rosotics-inc

[22] RViz, http://wiki.ros.org/rviz

[23] PrusaSlicer, https://www.prusa3d.com/page/prusaslicer_424/