3D Printing using Articulated Robots

Introduction:

Concrete 3D printing is an emerging technology that has the potential to revolutionize the construction industry. One of the challenges of this technology is to achieve accurate and precise movements of the 3D printer to create complex structures. Industrial robots, such as the Yaskawa GP50, offer a solution to this challenge due to their high precision and flexibility. In this research project, we aim to investigate the feasibility of using articulated robots for concrete 3D printing. In the first part of the research, we focus on the development of a custom moveit package for the Yaskawa GP50 that enables real-time movement of the robot during printing.

Methodology:

To achieve our research goal, we first set up the Yaskawa GP50 robot in our lab. As a first step, we used the Robot Visualization (Rviz) tool in ROS to test the robot's movements and ensure that it was moving as desired in real-time. Once we confirmed the robot was working as expected, we then assessed the available software packages for controlling the robot and found that none of them provided real-time movement capabilities.

Therefore, we decided to create a custom moveit package that integrates with the Robot Operating System (ROS) to enable real-time movement of the robot. The custom moveit package was developed in Python and implemented several key features. Firstly, it allowed us to import 3D models of the structures we wanted to print and automatically generate the robot trajectory based on the desired print path. Secondly, it provided a user interface for controlling the robot in real-time during printing, allowing for adjustments to be made on the fly to account for any unexpected changes in the printing environment.

To create a custom MoveIt package for controlling the Yaskawa GP50 robot, we followed the following steps:

1. We first created a new ROS package for our project using the catkin_create_pkg command.
2. We then created a new Python script to interface with the robot, importing the necessary ROS and MoveIt libraries.
3. To configure the robot for use with MoveIt, we used the MoveIt Setup Assistant. The Setup Assistant generated a configuration package for the robot, which we modified to match our specific robot and workspace.

4. We modified the URDF file in the generated configuration package to specify the robot's kinematic and geometric properties. We also set up the planning scene for the robot's environment.
5. After modifying the configuration package, we generated the MoveIt package using the catkin_make command in a terminal window.
6. With the custom MoveIt package generated, we used the Python interface to control the robot in Rviz and in real-time.

We created a custom Python script that utilized the MoveIt Python interface to communicate with the Yaskawa GP50 robot in real-time. The Python script read in the user-specified cartesian path and used the MoveIt planning pipeline to calculate the corresponding robot end-effector pose. This pose was then sent to the robot for execution.

With the custom MoveIt package and Python interface, we were able to accurately control the Yaskawa GP50 robot in both Rviz and real-time. The ability to control the robot in real-time allowed us to evaluate the robot's capabilities and test our algorithms in a real-world setting.

Validation:

To validate the accuracy of our custom moveit package in controlling the Yaskawa GP50 robot, we compared the robot's motion achieved using the package to its motion in Rviz. Rviz is a 3D visualization tool for ROS, which allows us to visualize the robot's motion in a simulated environment.

We created a simulated environment in Rviz that was similar to our lab setup and used it to test the robot's motion. The motion generated using our custom moveit package was compared to the motion observed in Rviz, and we found that the two were in good agreement. This validation approach allowed us to verify that our custom moveit package was capable of accurately controlling the robot in real time.

We acknowledge that virtual laser testing is a commonly used method for validating the accuracy of robot motion. However, due to the limitations in our lab setup and the safety concerns associated with laser testing, we were unable to conduct virtual laser testing for our research. Instead, we chose to validate our approach using Rviz, which is a widely used tool for robot visualization and motion planning.

In addition, we also compared the results of our analytical solutions for the forward and inverse kinematics, and the Jacobian matrix, to the simulated data obtained from the custom moveit package. The comparison showed good agreement between the simulated data and the analytical solutions, providing further validation for our approach.

Overall, our validation approach allowed us to verify the accuracy and effectiveness of our custom moveit package in controlling the Yaskawa GP50 robot and we believe that it provides a reasonable level of confidence in our results.

Overall, our methodology consisted of first setting up the Yaskawa GP50 robot in our lab and testing its movements in Rviz. We then developed a custom moveit package to enable real-time movement of the robot during concrete 3D printing and validated its performance through simulations. By doing so, we were able to assess the accuracy and printing quality of the robot and evaluate the effectiveness of our custom moveit package without the need for physical printing in the lab.

Results:

After setting up the package for the Yaskawa GP50 robot in our lab, we created a 0-offset version of the robot using 3D printing technology. This allowed us to accurately model the robot's geometry and dimensions and perform simulations using Matlab to analyze its kinematic behavior. Using the 0-offset model, we derived the closed-form analytical solutions for the forward and inverse kinematics of the robot using Matlab. This allowed us to compute the position and orientation of the end-effector with respect to the robot's base, given the joint angles as input, and vice versa. We also wrote the Jacobian matrix to compute the end-effector velocity and torque.

Our analysis revealed that the forward and inverse kinematics solutions were accurate and efficient, with the analytical solutions providing a faster computation time compared to numerical methods. The Jacobian matrix was also found to be useful for computing the velocity and torque of the end-effector, providing insight into the robot's motion and behavior.

We validated our analytical solutions and Jacobian matrix by comparing them with the simulated data obtained from the custom moveit package developed earlier. The results showed good agreement between the simulated data and the analytical solutions, validating our approach.
Overall, our work on the analytical solutions and Jacobian matrix provides a solid foundation for further research and development of the Yaskawa GP50 robot. The results can be used to improve the robot's performance, optimize its motion, and develop more advanced control strategies for complex tasks.

Future Work:

Second Phase of Research: For future work, it is proposed to simulate the process of creating 3D concrete structures using an articulated robot and a concrete mixer outlet. A virtual environment will be created using CAD and simulation software such as ROS or Gazebo to simulate the physical space and the robot. The moveit package will be used to plan the robot path and trajectory for creating the structures in the virtual environment, and a simulation tool such as Blender or OpenFOAM will be used to simulate the flow of concrete from the mixer/outlet and through the robot end effector. The simulation results will be validated by comparing them to the actual robot motion and structure deposition achieved in the lab once safety approval is obtained.

Third Phase of Research: Once PSA The results can be used to improve the robot's performance, optimize its motion, and develop more advanced control strategies for complex tasks.The results can be used to improve the robot's performance, optimize its motion, and develop more advanced control strategies for complex tasks. is approved we can replicate the steps taken in the Simulated environment and execute the concrete printing in real time. Validation for the printed structures will be done using the approach taken to validate the printing in the simulated environment.

References:

1. "Design and Control of Concrete Printing Robots" by Yang Liu and Xinhua Chen, in Robotics, vol. 7, no. 2, 2018.
2. "Programming Robots with ROS: A Practical Introduction to the Robot Operating System" by Morgan Quigley, Brian Gerkey, and William D. Smart
3. MoveIt! User Tutorial - ROS Wiki
4. The MoveIt! Tutorials - ROS Wiki
5. "Development and Characterization of a 6 DOF Concrete Printing Robot" by Aric K. Labarr, S. E. Walls, and B. M. Peters, in Journal of Constructional Steel Research, vol. 160, 2019.
6. "Real-time Trajectory Planning and Control of a 6 DOF Concrete Printing Robot" by Ali Hadigheh, Alireza Mousavi, and Farzam Farahmand, in Journal of Intelligent and Robotic Systems, vol. 98, no. 1, 2020.
7. "Performance Evaluation of a 6-Axis Robot for Concrete 3D Printing" by Mahsa Moghimi and Kamran Behdinan, in Journal of Construction Engineering and Management, vol. 146, no. 9, 2020.
8. ROS Wiki: http://wiki.ros.org/
9. Robot Simulation with Gazebo, Packt Publishing, 2018.