SCRIPTA
Volume 1, Number 2, Fall 2022
ISSN: 2769-9242

Automating Robotic Arm to Move Stem Cell Well Plates

Pranavkrishna Suresh*

Abstract

The paper introduces an automated robot and gripper mechanism to transfer stem cell well plates between analytical machines in a clean room setting. The software utilizes an ROS finite state machine architecture to autonomously decide the course of action to safely transport a well plate given a final location. Contrary to most previous systems, this model is specialized for use by scientists with no computer science background to increase its usability. Therefore, it does not require any coding to operate beyond the one-time use of a Write Service function to generate a JSON file indicating destination position.

1 Introduction

Stem cell research has long been lauded for its extensive potential in the laboratory-facilitated growth of human tissue. Facing decades of court battles on the ethicality of collecting stem cells and insufficient funding for research, scientists' general understanding of embryonic stem cells has stagnated. Embryonic stem cells are the first cells of the human fetus that differentiate and specialize to serve various functions. Such differentiation allows various organs of the body to grow and function together. Harnessing this ability in a laboratory setting is essential for various applications, one of which is supporting the body's immune system in fighting certain disorders such as cancers and Crohn's disease [1]. This project is primarily intended to work with Mesenchymal stem cells and CAR-T cells, which have potential to augment immune responses against pathogenic invaders [2] [3]. Despite the immense ability of this research to restructure the

*Senior Student, Fulton Science Academy.
Contact: psuresh@fultonscienceacademy.org

medical industry, it has been delayed indefinitely due to the safety conditions and ethical considerations that must be adhered to while handling physical samples to maintain the precision of experiments [4]. The primary cause of ethical concerns is the source of embryonic stem cells: the human fetus [5]. With regulations varying across states and interference from religious groups, securing samples is an intensive process that requires years of planning. Furthermore, maintaining high levels of consistency in the results of stem cell treatments has proven to be a difficult task due to variation in human genomes [6]. Therefore, the limited samples that are approved must be handled with immense caution as the influx of further supply could be interrupted at any moment.

The process of preparing a stem sample is extremely meticulous and involves transporting the sample to various machines that analyze different characteristics of the sample at progressing stages of growth. When moving the well plates containing the samples, they must be held perpendicular to the ground and moved at a constant velocity so that the contents will not be tilted or warped. Precise measurements allow scientists to understand the rate at which stem cells grow and when intervention would allow scientists to control the trajectory of the cells and their functions.

The unique requirements of the situation inspired our team to find an effective solution that utilized automation. We decided to program an industrial-style robot (UR3e) to automate the procedure using the Python-based Robotic Operating System with a specially designed 3D-printed gripper-mechanism that would pick up and place the well plate at requested locations. This apparatus would remove the need for trained scientists to carefully maneuver well plates in dense clean-room settings, increasing the rate at which experiments can be conducted. Furthermore, the possibility of human contamination in the process is eliminated, making procedures more reliable [7]. We identified the clean room at Marcus Nanotechnology Research Laboratory at The Georgia Institute of Technology as a potential client to assess the utility of our technology

2 Materials and Methods

2.1 Hardware: Gripper Mechanism

The gripper-mechanism was designed with the motive of making it universally applicable to standard-sized well plates. For the 3D model, we used the SolidWorks software. Various iterations were 3D printed with 100% infill to maximize the product's durability, an industry requirement in the potentially corrosive conditions the gripper could encounter in a non-controlled laboratory environment [8]. The gripper arms span the length of the well plate and incorporate two limit switches in their architecture to measure the pressure that the well plate exerts onto the gripper arm. The switches are wired to an Arduino Mega and are essential for automation as they transmit binary data detailing the gripper's status in grabbing or detaching from the well plate to the finite-state machine's algorithmic architecture. The Arduino is further configured to operate a servo motor, the driver behind the movement of the gripper arms. In order to maximize the grip of the arms, they are lined with industrial rubber tape to act as a frictional force against the well plate. View Figure 1 for the consolidated gripper mechanism design detailing the various components. View Figure 2 for the Arduino configuration. View Figure 3 for an image of the robot and gripper-mechanism in operation.

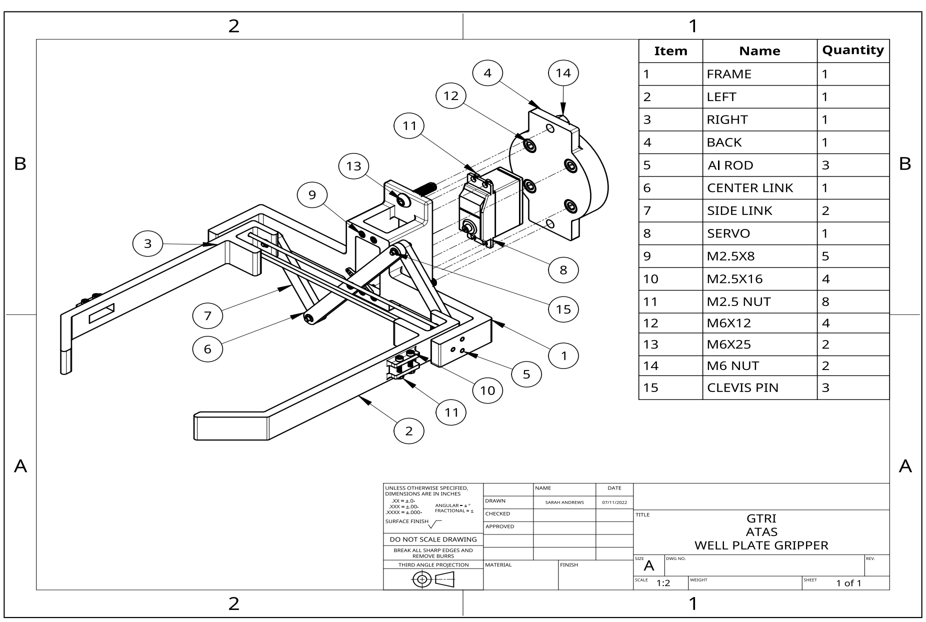

Figure 1: Gripper Mechanism Component Diagram

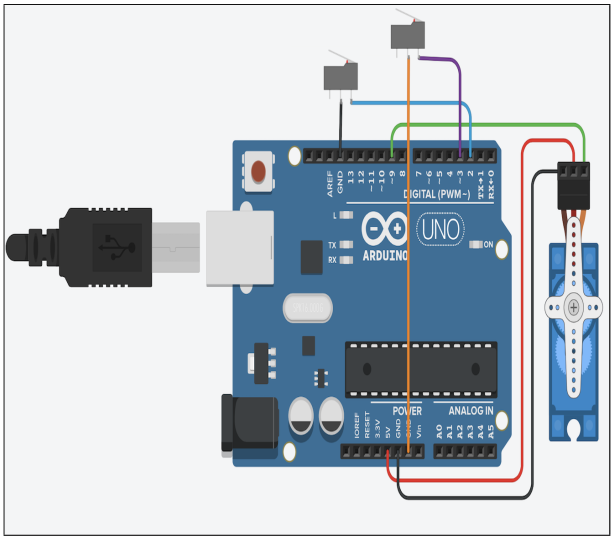

Figure 2: Arduino Diagram

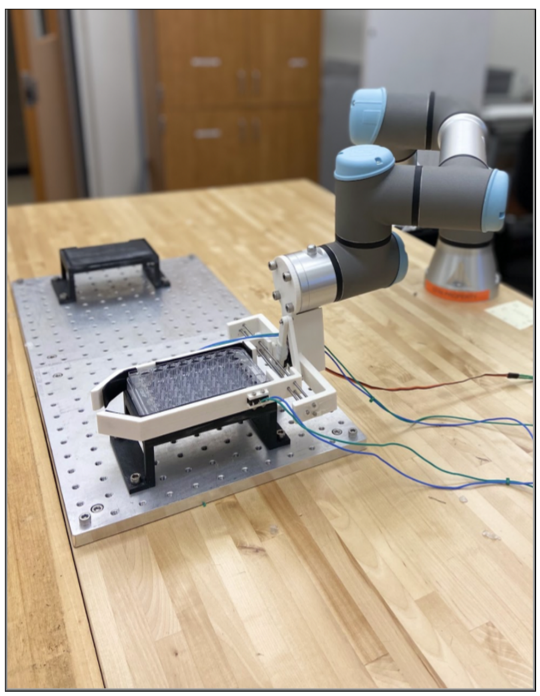

Figure 3: Robot Arm in Operation

2.2 Software

2.2.1 ROS and Related Softwares

The UR3e robot utilizes the Robotic Operating System (ROS) as the main method of operation, a C-based open-source framework. We opted to use the python interface for the purposes of accessible core functionality. The ROS MoveIt software incorporates Stochastic Trajectory Optimization for Motion Planning (STOMP) to automatically plan the robot's movement given any two locations and indicated constraints. This significantly facilitates piloting the robot as complex matrix operations for the movement of each joint of the arm are computed locally and instantly executed. The robot arm was constrained to movement at a constant velocity, remain at or above the height of the well plate at all times, and never directly rotate the end effector as it is attached to the gripper mechanism. These constraints ensure that the contents of the well plate will not be warped during the transition from the starting to the ending position. View Figure 4 for the predefined homogeneous transformation matrix calculated using STOMP.

$$
{}^{n-1}T_n = \left[\begin{array}{ccc|c} \cos\theta_n & -\sin\theta_n\cos\alpha_n & \sin\theta_n\sin\alpha_n & r_n\cos\theta_n \\ \sin\theta_n & \cos\theta_n\cos\alpha_n & -\cos\theta_n\sin\alpha_n & r_n\sin\theta_n \\ 0 & \sin\alpha_n & \cos\alpha_n & d_n \\ \hline 0 & 0 & 0 & 1 \end{array}\right] = \left[\begin{array}{ccc|c} & & & \\ & R & & T \\ & & & \\ \hline 0 & 0 & 0 & 1 \end{array}\right]
$$

Figure 4: STOMP Matrix Computation

Due to a finite number of joints on the robot arm, there are certain movements that it will not be able to execute without risking self-damage. In a laboratory setting, this would not only risk the structural integrity of a multi-thousand dollar robot arm but also the viability of the samples in the well plate. The Rviz software enables risk-free simulations of the robot environment and works with STOMP to create usable solutions rather than unrealistic optimized solutions that are mathematically preferable. View Figure 5 for Rviz simulating the movement of the robot arm in its environment.

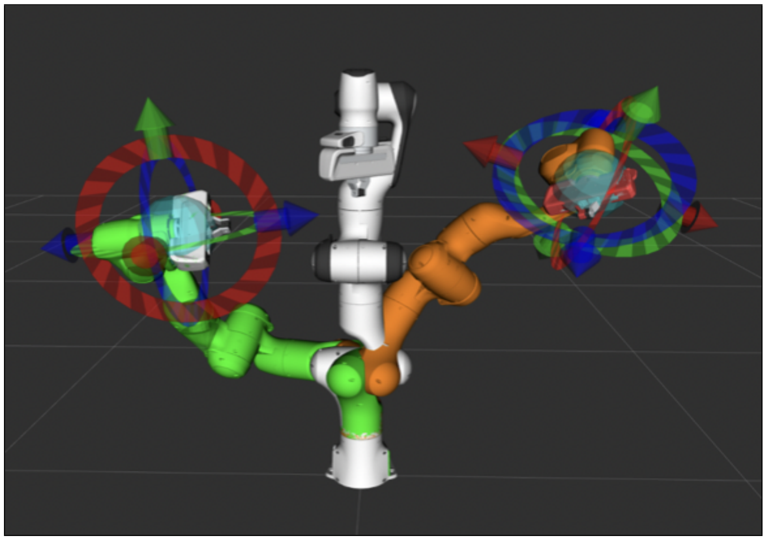

Figure 5: ROS MoveIt Visualization

2.2.2 Finite-State Machine Framework

In order to automate the robot, we employed the finite-state machine architecture, which infinitely loops the algorithm to decide the next action for the robot to execute based on the present conditions. The architecture has seven preprogramed states (State_command_received, state_no_command, state_at_wellplate, state_grabbed_wellplate, state_at_destination, state_completed_drop, state_error) that describe the current conditions under which the robot operates. While the first six states address various stages of the robot transferring the well plate, state_error is an open ended state; it attempts to solve functional issues in real-time or freezes the robot at a stationary position to prevent potential damage. States such as state_grabbed_wellplate and state_completed_drop are dependent on data transferred from the arduino node indicating the movement of the gripper-mechanism. View Figure 6 for a diagram of the employed finite state machine.

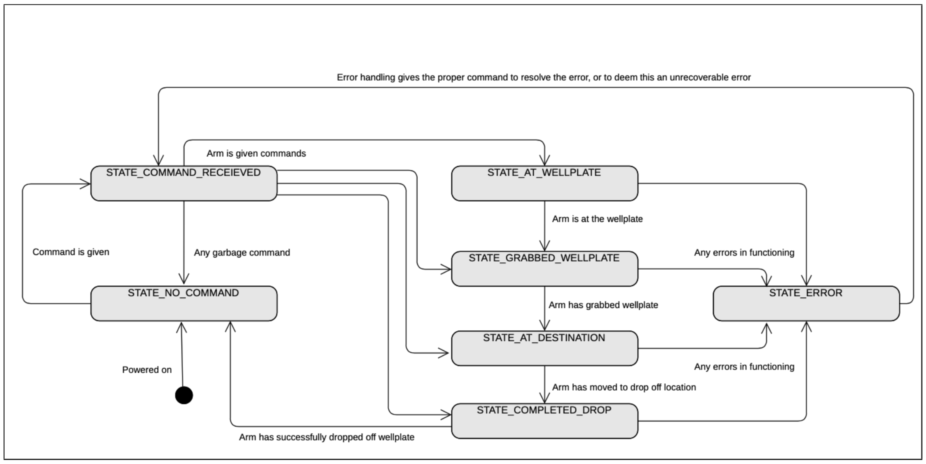

Figure 6: Finite State Machine Diagram

To initialize the finite state machine, a robot pose indicating the final location of the robot must be input as a JSON file. This JSON file can be collected via two methods: the write and read service. The Write Service is designed to automatically generate the JSON file based on the current position of the robot arm; this enables scientists to free-handle the robot arm and generate the file for a one-time initial setup, easing the procedure and preventing the traditional requirement of coding experience to operate the arm. Then, the Read Service fetches a position that is either manually requested or generated by the Write Service. Upon preprocessing, the finite state machine activates and begins to transfer the well plate.

2.2.3 Arduino Node

The Arduino Service is configured as a node to the finite state machine and engages in one-way binary data transfer. The primary purpose of the Arduino Service is to serve as the operative link between the gripper-mechanism and the rest of the automation framework. When the limit switches return 1 to the arduino, meaning that the well plate is exerting pressure on them, a confirmation that the gripper has attached to the well plate is transmitted to the finite state machine. If only one switch returns a 1 or both return 0, then a negative confirmation is transmitted. This enables the robot to pursue further actions only when the gripper-mechanism has successfully completed attachment and STATE_GRABBED_WELLPLATE reads true. View Figure 7 for a component overview that features the pathways of communication in the finite state machine.

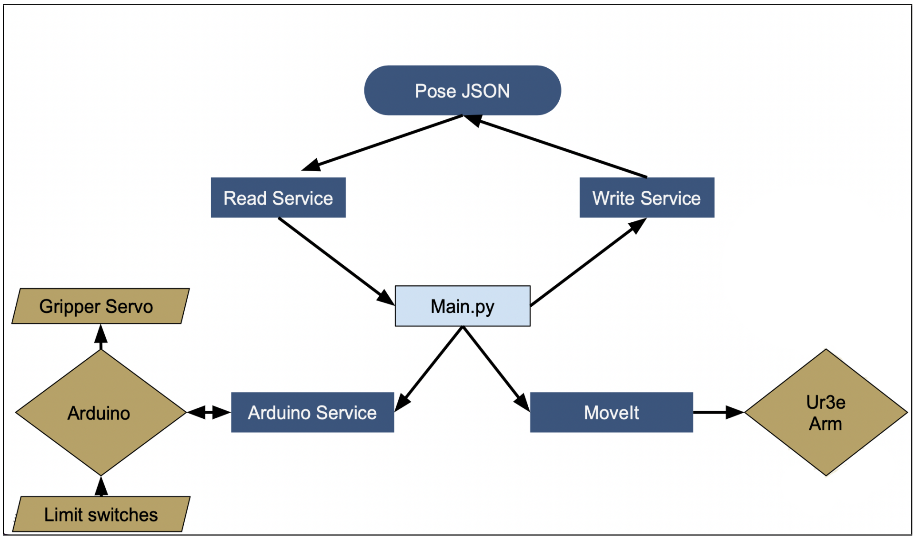

Figure 7: Component Overview

Conclusions

The gripper-mechanism design and automated finite state machine architecture discussed in this paper enable substantial efficiency gains in laboratory settings. We expect that future designs will increase the gripper-mechanism's versatility, allowing it to remove and attach the seal placed on well plates. Despite these advances, it will be difficult to approach error-free performance. The limitation is due to complexities in the current software framework, which must be mitigated to ensure ease of use among scientists. We hypothesize that using a more malleable gripper will be suitable for this task and enable the mechanism to operate with higher precision.

3 Acknowledgments

This author would like to specially thank Dr. Ahmadoghlu for his guidance in writing this paper.

This author would like to thank his fellow team members on the software team for this project: Anirudh Malli and Marc Van Zyl.

This author would like to thank his fellow team members on the hardware team for this project: Colin Arscott, Akshaj Evuru, and Sarah Andrews.

This author would like to thank his mentors and supervisors for this project: Mr. Walker Byrnes, Mrs. Shelley Carter, and Mr. Avi Balakirsky.

This author would like to thank the program organizers who organized this internship opportunity: Dr. Erick Maxwell, Mrs. Therese Boston, Dr. Leigh McCook, and Dr. Jim Hudgens.

This author would like to thank the following researchers at Georgia Tech who provided insight into their respective projects: James Childers, Codey Fernandez, Dante Orlando, Alan Freeland, Quinn Spadola, Spencer Fallek, David Reid, Daniel Jones, Jack Wood, Dr. Stephen Balakirsky, Sean T. Wilson, Lee Whitcher, and Dr. Kelly Griendling.
Conflict of Interest: None declared.

References

[1] Nayoun Kim and Seok-Goo Cho. Clinical applications of mesenchymal stem cells. *The Korean Journal of Internal Medicine, National Library of Medicine*, 28(4):387, 2013.

[2] American Cancer Society. https://www.cancer.org/treatment/treatments-and-side-effects/treatment-types/immunotherapy/car-t-cell1.html.

[3] Yang Wang, Hanxiao Yi, and Yancheng Song. The safety of msc therapy over the past 15 years: A meta-analysis. *Stem cell research & therapy*, 12(1):1–15, 2021.

[4] Fernando et al. Cobo. Microbiological control in stem cell banks: approaches to standardisation. *Applied microbiology and biotechnology*, 68(4):456–466, 2005.

[5] NIH. https://stemcells.nih.gov/research-policy/guidelines-for-human-stem-cell-research.

[6] Bernard Lo and Lindsay Parham. Ethical issues in stem cell research. *Endocrine reviews*, 30(3):204–213, 2009.

[7] Susumu Ikehara. Grand challenges in stem cell treatments., 10 Oct 2013.

[8] Robin Wesselschmidt and Philip Schwartz. The stem cell laboratory: Design, equipment, and oversight. In *Human pluripotent stem cells*, pages 3–13. Springer, 2011.