**Chapter 1**

**Background**

In the last few decades, the world has seen a radical shift in the way that autonomous systems are utilised in the manufacturing industry. As factories and warehouses begin to become more reliant on robots performing repetitive tasks, there has been an upsurge in the number of other industries looking to enhance their everyday operations through the implementation of autonomous systems. Thanks to recent robotic developments in the areas of portability, safety, and ease of use, this new generation of collaborative robots has been designed to work alongside humans in the workplace. Collaborative robots have continued to push the bounds of what was previously seen as work that could only be performed by a human worker.

The use of robots in a lab environment is not a foreign concept, although now they are being considered for use in liquid handling and data collection roles. It is believed that collaborative robots have reached a level of technological advancement that they are able to overcome the challenges faced by typical robots of rigid movements and being less adaptable to changing conditions.

Likewise, the often-random nature of liquid makes it a difficult subject to integrate with autonomous data acquisition systems. There is often a large variability between different kinds of liquid each with their own mass, density, volume and viscosity, meaning that a system would have to account for all of these factors to accurate estimate its physical properties. Any system that can accomplish this task and be added to a lab would effectively help automate research and provide a cost-benefit to the organisations that utilise it.

In recent years there have been many attempts at developing a system that could perform these tasks to an acceptable degree. Methods such as the use of a camera to measure the changing height of a poured liquid, as presented in (Do & Burgard, 2018) and (Schenck & Fox, 2016), found that there were shortcomings with the use of transparent liquids such as oil. These methods were also susceptible to inaccurate data from their video data. Another intuitive method found was presented in (Liang, et al., 2020) and featured acquisition of liquid data based upon audio feedback of the pouring liquid. Like the previous methods this too was considered unsuitable due to erroneous data from the audio sensors and its incompatibility with highly viscous fluid.

The inspiration for this project was based upon the work presented in (Matl, et al., 2019). The method used in this paper was the use of a robotic arm with an accurate force/torque sensor. Based upon mechanical movements made by the arm and data readings for the mass and torque, it would be possible to accurately measure the properties of a liquid within a container such as mass, volume and viscosity. This method contained significant advantages over other papers such as having a wide range of liquids that could be accurately measured, regardless of viscosities or volume. Whereas other methods were centred around pouring of a liquid, this method was capable of measuring a liquid within an enclosed container. This project explores the use of this system design for the goal of autonomously measuring the physical properties of a liquid.

**Research Objectives**

The aim of this project is to develop a system that is capable of measuring the properties of a liquid through mechanical manipulation and data gathering via a robotic platform and mathematical models.

Specific objectives:

* Utilise the approach presented in (Matl, et al., 2019).
* Combine the above approach with a real autonomous system.
* Demonstrate the validity of this system as a solution to collecting physical properties of liquid samples.

**Report Organisation**

This report is organised as follows:

* Chapter 2 explores previous work that has been used to influence the project.
* Chapter 3 details the project approach and discusses the implementation of the chosen method.
* Chapter 4 presents the experimentation process and the project results gathered.
* Chapter 5 discusses the results and suggests project changes and potential future work that can be performed.
* Chapter 6 is the conclusions that were drawn from this research project.

**Chapter 2**

**Related Work**

**Liquid Mass Measurement**

Knowing the mass of a sample being manipulated by a collaborative robot has been a widely studied aspect of autonomous liquid handling systems. Due to the relative inaccuracy of visual data, liquid mass measurements are the prime method of calculating further physical properties of liquids such as volume from a known density, or density from a known volume. Solutions to the problem of collecting liquid mass data by autonomous systems has often been dependant on its final application. In the research papers (model based flow rate control) and (Outflow Liquid Falling Position) the solution to the problem of liquid mass measurement came from load-cell that were utilised during liquid pouring operations. Whereas this solution was adequate for the application of the systems described in the papers, it would be unsuitable for system applications involving careful handling of liquids.

This is in contrast to the method presented in (Matl, et al., 2019) where the liquid mass is based off of sensors readings from a force torque sensor. This is due to the project scope in (Matl, et al., 2019) being to design a system capable of measuring the physical properties of a liquid from within an enclosed container.

**Liquid Volume Measurement**

Correctly estimating the mass of a liquid within a container is a fundamental aspect of the goal set out to be achieved by this project. This topic has many extensively studied methods, with a majority of them involving the use of image data systems that often require further image processing and neural networks to adequate generate valid data (Schenck & Fox, 2016). Others have attempted to use audio feedback as a method of measuring volume, however this too incorporated a multitude of sensors and neural networks (Liang, et al., 2020). The method presented by (Matl, et al., 2019) showcases a way of using a physics-based model to generate equations for the volume of a liquid within an enclosed container based upon haptic feedback from a force torque sensor.

**Liquid Viscosity Measurement**

Due to the challenging nature of measuring liquid in motion, there have been many attempts to find a solution to this problem. Methods such as those presented in (Particle-Based Fluid Simulation for Interactive Applications) showcase a method of simulating liquid motion in a container. Other papers draw conclusions that such models can be equated to a much simpler multi-mass-spring-damper system (Point-to-point liquid container transfer via a PPR robot with sloshing suppression). Ultimately, the paper (Matl, et al., 2019) bases much of their mathematical models for calculation of liquid viscosity from (the new dynamic behavior of liquids in moving containers dodge) and they will be the methods I attempt to utilise in this project.

**Chapter 3**

**Overall Approach**

The goal of this capstone project was to investigate methods that could be incorporated with an autonomous system to measure the physical parameters of a liquid within an enclosed container. These methods were drawn from the research paper (Matl, et al., 2019) and were implemented based upon the resources that were available to conduct this project. The basis of this approach is the physical manipulation of a liquid within an enclosed container and gathering data via haptic feedback of internal fluid position and motion. The mathematical models used by (Matl, et al., 2019) to analyse this data into meaningful information provides equations to calculate the height and volume of a liquid within a container based upon its changing centre of mass. By analysing this data at discrete angular rotations an approximation for the internal volume of the liquid can be found. Likewise, the calculation of the liquid viscosity could be found through analysing the decaying oscillations of a sloshing liquid within a closed cylinder.

Implementation of these mathematical models required the development of an autonomous system that could collect and processing data effectively in a reasonable amount of time. This included managing the robotic hardware and software needed to adequately achieve the project objectives.

At the beginning of operation, the autonomous control code would zero the sensor readings coming from the force/torque sensor on the robot end effector. It would then allow a bottle to be placed in its gripper arm before recording the mass of the bottle and liquid and starting its pre-planned movements. These movements would take it through several discrete angular rotations, at each stage stopping to record torque readings generated by the liquid centre of mass. Once an adequate amount of data has been recorded the bottle is then rotated onto its side and rotated quickly back to an upright vertical position while torque data is continuously recorded as the liquid settles within the container. The data gathered by the system is then saved into files and analysed by the mathematical models discussed above.

The individual approaches for each of these methods will be detailed below.

**Robot Details and Control Method**

The robotic system that was selected for use with this project was the UR10e collaborative robot made by Universal Robotics. Attached to the end effector of this system was a RG2 gripper made by OnRobot which allowed the handling of a bottle containing liquid. The end effector of the UR10e has an inbuilt force/torque sensor that was used to collect data during this project. Installed on this UR10e was the appropriate URCap that allowed interfacing with the RG2 through Polyscope.

Control of this system was achieved through a sequential program constructed in UR Polyscope that ran a predefined set of instructions that took the UR10e through all movements it must undertake. These instructions dealt with the sequencing of all tasks such as control over when to open and close the end effector gripper arm and the transition from one angled position to the next. Sequencing with externally running ROS nodes was accomplished with the UR10e external I/O. It was possible to set these digital outputs high and low using Polyscope commands and their current states were able to be read via a ROS topic. Sequencing with an external ROS node allowed the correct acquisition of data to take place.

**Data Acquisition**

Data from the UR10e force/torque sensor was achieved through a ROS node running with C++. Upon start-up of the robot a ROS package called ur\_robot\_driver would be launched which would interface with the UR10e and allow ROS topics to be published from the robot. These topics would include joint data, IO states, and force/torque data. These topics were subscribed to from the ROS node, and via sequencing from reading the IO states topic the node could collect data when it was specified by the system to take through callbacks. These IO states are read to indicate to the ROS node about the current state of the UR10e Polyscope script running such as when the program is running, when the script indicates the correct time to record data, and when the data can be saved to the external files.

When the UR10e end effector moves to an angle specified by its control program, it will set high one of the internal digital IO ports. The status of this port is read by the ROS node and when it reads high the node immediately starts recording data from the force/torque sensor. This signal will remain high for 5 seconds and while it remains high data will continue to be read from the sensor. When this signal goes low the data vector of torque readings is saved to an internal variable of the ROS node for later saving. Upon conclusion of the program, the data within these internal variables are written to text files for later processing.

**Data Processing**

Much of the data processing for this project is done through MATLAB. All data to be analysed appears in the form of text files generated from the ROS node that acquires the data. This includes 10 text files containing data related to the centre of mass measurements, 1 text file containing continuous data readings for viscosity calculations, and a text file containing the initial zeroed value of the force/torque sensor.

Within MATLAB each of these text files are read and interpreted as a matrix and saved as the raw data variables. These matrices are then processed with a function to remove potential outliers from the data before the mean value of the centre of mass readings and zero data is calculated. The equations formulated by (Matl, et al., 2019) detail a way in which the volume of a liquid within a container can be calculated from the known properties of bottle radius, angle, bottle length, length from bottle base to gripper, liquid mass, gripper mass and container mass. This equation is used to calculate the perceived torque experienced by the force/torque sensor at a particular angle dependant on the amount of liquid in the container. This equation contains the unknown variable of the liquid height within the container. By comparing the measured torque against the calculated torque using the method of non-linear least squares a value for the height can be found that minimises the error between the measured and calculated value and therefore finds a value for height that satisfies the real-world experiment. This value of height is then used in an equation with the bottle angle to finally compute a final value for the liquid volume.

**Approach justification**

This capstone project approach was centred on the basis of using the UR10e robot for all autonomous movements. This robot was chosen due to its availability and access to the required resources to control its movement. Much of this project was a learning experience particularly with implementing a real-world robotic system with ROS, and time was taken to learn which methods of control would be suitable and could be achieved in a reasonable timeframe. The selected method of control of the UR10e through a separate program running within Polyscope was chosen due to the time taken to implement other methods such as the ROS Moveit toolbox. This method also allowed much of the sequencing and ordering of commands to take place outside of the ROS node in a programming environment that could be easily altered.

The use of a ROS node interfacing with the UR10e through the ur\_robot\_driver package allowed subscribing to the appropriate topics published by the UR10e such as the joint states, IO states, and force/torque sensor data. It was initially unknown if the UR10e would allow a Polyscope control program to run while publishing topics via ROS. Other methods of acquiring data such as through reading network packages was discussed but never implemented. Ultimately a ROS node coded in C++ was chosen due to ease of gathering data via the published topics and the ability to export collected data in the form of text files with relative ease.

For the processing of the data the selected program that was used was Matlab. Matlab was chosen due to its nature of being a useful tool for creating and implementing mathematical models, as well as the data processing tools it provided that could be used to analyse the results gathered by the system. Matlab allowed processing of each data file and allowed mathematical functions to be used to remove outliers from the data before the mean was calculated.