**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.

**Experimental Setup**

The bottle that was chosen to use for this experiment was a glass drinking bottle with a screw lid. This bottle had an internal radius of 30.15mm and was approximately 260mm tall. When weighed on a set of scales the bottle had a mass of approximately 0.339kg. This bottle was used for all experimentation with liquid and was always inserted into the gripper arm at distance of 160mm from the base. During experimentation, the bottle was filled with approximately 188mL of both water and oil.

**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, 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.

This function is plotted as the value for height is increased from 0 to 200mm. The output from this function would be the error between the calculated torque and the measured torque. Using the value for height 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 compute a final value for the liquid volume.

EQUATIONS WILL BE INCLUDED IN APPENDIX

**Approach justification**

Glass bottle was selected for use in this project because the internal base of the bottle contained a small amount of irregular geometry compared to other similarly sized bottles. If the base of the bottle was an irregular shape this would affect the results as the equations utilised in (Matl, et al., 2019) were based upon the assumption that the volume of the bottle was a perfect cylinder. The outside diameter of this bottle also ensured it could easily fit within the RG2 gripper arm for the UR10e without need for further modifications. The large ratio between the height and width of the bottle was also desirable as this would allow for a larger range of movement for the liquid centre of mass as it was moved between angular positions, as this was believed to help with the sensitivity of data gathered.

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.

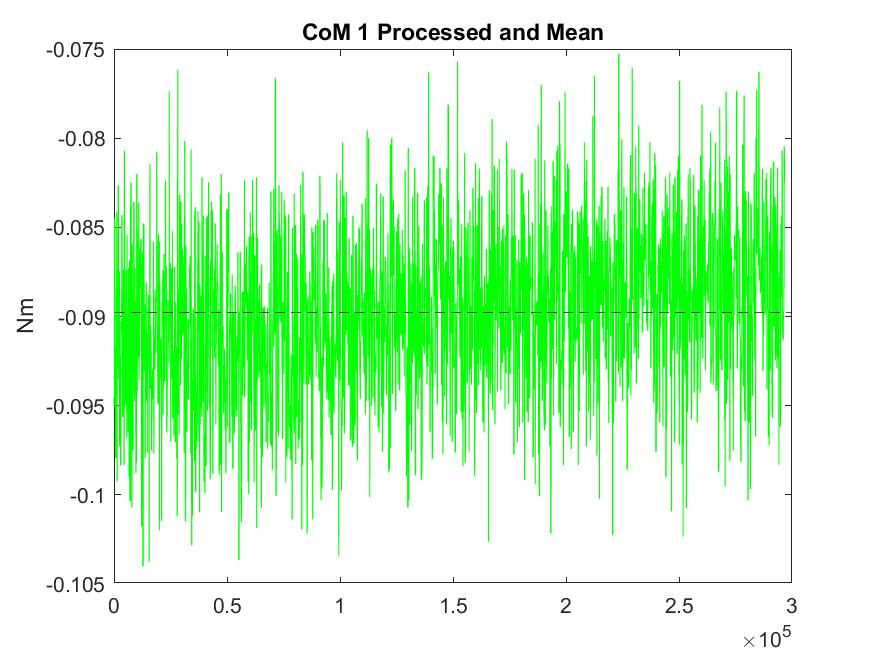
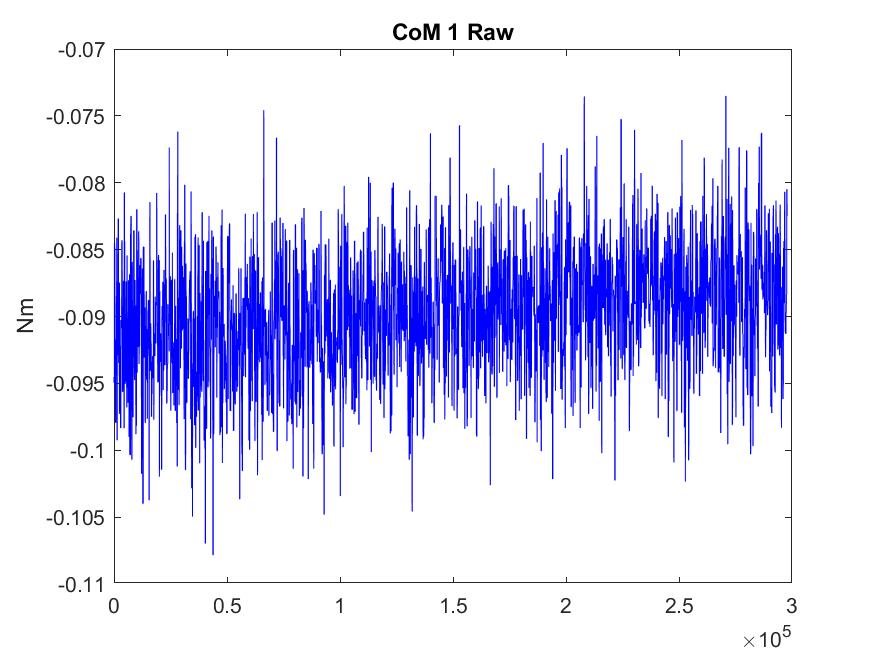
For the processing of the data the selected program that was used was Matlab. Matlab was chosen due to its nature of being a useful tool for creating and implementing mathematical models, as well as the data processing tools it provided that could be used to analyse the results gathered by the system.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

**Chapter 4**

**Results and Evaluation**

The results generated by the ROS node come in the form of ten text files for each of the centre of mass readings, a text file containing zeroed force/torque data, and a text file containing viscosity data readings. These files are processed through Matlab to remove outliers from the data and calculate the mean values for the centre of mass and zero data files.

The program was run with different testing conditions to establish a well-rounded basis for how external factors affected the torque measured by the sensor from other sources apart from the liquid sample. These included without a bottle and with just the bottle. Once these results were gathered the test was run again with water and sunflower oil. Each of these experiments was performed twice without changing the experimental setup to compare results and check the accuracy of the system.



Shown above is the data contained in one of the text files for the experiment run with no bottle. On the left is the raw data contained within the file which was processed in Matlab. This data was processed to remove the outliers and from this the mean was calculated, both of which are presented in the graph on the right. This same process was performed for all runs of the experiment with each point where the centre of mass was read, and the recorded zero data of the experiment. The values for the end effector angle and mean torque at each of these points were recorded in the following tables.

|  |  |  |
| --- | --- | --- |
| **No Bottle** | | |
| **Reading Type** | **Angle (θ)** | **Mean Torque (Nm)** |
| Zero | 0 | 0.000469984 |
| CoM 1 | 9.99942 | -0.089765496 |
| CoM 2 | 19.7147 | -0.109094581 |
| CoM 3 | 29.7114 | -0.112744514 |
| CoM 4 | 39.7143 | -0.14037828 |
| CoM 5 | 49.7119 | -0.154494568 |
| CoM 6 | -10.2859 | 0.010135029 |
| CoM 7 | -20.2854 | -0.001847533 |
| CoM 8 | -30.2836 | -0.006111559 |
| CoM 9 | -40.2876 | -0.02186066 |
| CoM 10 | -50.2882 | -0.040660176 |

|  |  |  |
| --- | --- | --- |
| **Only Bottle** | | |
| **Reading Type** | **Angle (θ)** | **Mean Torque (Nm)** |
| Zero | 0 | 0.001129252 |
| CoM 1 | 10.0022 | -0.067898762 |
| CoM 2 | 19.7147 | -0.096827512 |
| CoM 3 | 29.7135 | -0.108395671 |
| CoM 4 | 39.7129 | -0.14474166 |
| CoM 5 | 49.7119 | -0.164229923 |
| CoM 6 | -10.2859 | 0.101588789 |
| CoM 7 | -20.2868 | 0.099088678 |
| CoM 8 | -30.2836 | 0.099932153 |
| CoM 9 | -40.2862 | 0.094566729 |
| CoM 10 | -50.2861 | 0.085201848 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bottle with Water** | | | | |
| **Reading Type** | **Angle 1 (θ)** | **Mean Torque 1 (Nm)** | **Angle 2 (θ)** | **Mean Torque 2 (Nm)** |
| Zero | 0 | 0.005880078 | 0 | -0.007490919 |
| CoM 1 | 10.0022 | -0.091731896 | 9.99942 | -0.082627783 |
| CoM 2 | 19.7154 | -0.152815257 | 19.7119 | -0.155503304 |
| CoM 3 | 29.7114 | -0.199845926 | 29.7141 | -0.199768551 |
| CoM 4 | 39.7122 | -0.268075221 | 39.7122 | -0.270639101 |
| CoM 5 | 49.7147 | -0.309623959 | 49.714 | -0.319107842 |
| CoM 6 | -10.2859 | 0.220569689 | -10.2831 | 0.229471511 |
| CoM 7 | -20.2854 | 0.252032275 | -20.2868 | 0.261037401 |
| CoM 8 | -30.2878 | 0.280528957 | -30.2843 | 0.290620944 |
| CoM 9 | -40.2869 | 0.308181553 | -40.2862 | 0.323785441 |
| CoM 10 | -50.2875 | 0.334279357 | -50.2855 | 0.347097379 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bottle with Sunflower Oil** | | | | |
| **Reading Type** | **Angle 1 (θ)** | **Mean Torque 1** | **Angle 2 (θ)** | **Mean Torque 2** |
| Zero | 0 | -0.005949557 | 0 | 0.009307426 |
| CoM 1 | 10.0001 | 0.016321445 | 10.0022 | -0.113127257 |
| CoM 2 | 19.714 | -0.048950251 | 19.7147 | -0.17481718 |
| CoM 3 | 29.7169 | -0.094856618 | 29.7135 | -0.218635946 |
| CoM 4 | 39.7129 | -0.166749024 | 39.7143 | -0.284421621 |
| CoM 5 | 49.714 | -0.21446498 | 49.7119 | -0.329980715 |
| CoM 6 | -10.2859 | 0.378029927 | -10.2866 | 0.240008156 |
| CoM 7 | -20.2861 | 0.414313405 | -20.2868 | 0.271768901 |
| CoM 8 | -30.2843 | 0.453699982 | -30.285 | 0.301552432 |
| CoM 9 | -40.2869 | 0.483250221 | -40.2849 | 0.331610118 |
| CoM 10 | -50.2848 | 0.509296794 | -50.2868 | 0.353118456 |

The results gathered by the experiments involving liquid were then analysed with the Matlab model developed to find a final value for the liquid volume at each reading of the centre of mass. Each point was analysed comparing the change in height of the liquid at a specific angle to the change in the error between the measured and calculated values for the torque generated. At each of these points the calculated height was used again to calculate the volume of the liquid.

**Chart, line chart

Description automatically generatedBottle containing 188mL Water**

Chart, line chart

Description automatically generated

Chart, line chart

Description automatically generated

Chart, line chart

Description automatically generated

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated

Chart, line chart

Description automatically generated

Chart, line chart

Description automatically generated

The height of the liquid where the error is minimised and closest to zero is used to calculate the final volume of the liquid.

|  |  |  |
| --- | --- | --- |
| **Reading Type** | **Water Height (m)** | **Water Volume (mL)** |
| CoM 1 | 0.025 | 57.3109 |
| CoM 2 | 0.031 | 63.1869 |
| CoM 3 | 0.035 | 65.9474 |
| CoM 4 | 0.039 | 73.2678 |
| CoM 5 | 0.041 | 79.5021 |
| CoM 6 | 0.015 | 59.1618 |
| CoM 7 | 0.010 | 62.2712 |
| CoM 8 | 0.005 | 66.8252 |
| CoM 9 | 0 | 72.9857 |
| CoM 10 | 0 | 103.6640 |

**Chart, line chart

Description automatically generatedBottle containing 188mL Sunflower Oil**

Chart, line chart

Description automatically generated

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated

Chart, line chart

Description automatically generated

Chart, line chart

Description automatically generated

The height of the liquid where the error is minimised and closest to zero is used to calculate the final volume of the liquid.

|  |  |  |
| --- | --- | --- |
| **Reading Type** | **Oil Height (m)** | **Oil Volume (mL)** |
| CoM 1 | 0.026 | 60.2135 |
| CoM 2 | 0.028 | 54.0877 |
| CoM 3 | 0.033 | 59.3663 |
| CoM 4 | 0.039 | 73.2675 |
| CoM 5 | 0.041 | 79.5020 |
| CoM 6 | 0.015 | 59.1618 |
| CoM 7 | 0.010 | 62.2725 |
| CoM 8 | 0.004 | 63.5105 |
| CoM 9 | 0 | 72.9857 |
| CoM 10 | 0 | 103.6541 |

As can be seen from the results the Matlab model does not generate accurate values for the final volume of the liquid within the container. The expected result was that any single point could give an approximation for the liquid volume however there is a large discrepancy between all values of the results. Even when comparing the data gathered between tests of the same liquid with the same experimental conditions, discrepancies can be seen in the tables below.

|  |  |  |  |
| --- | --- | --- | --- |
| **Bottle with Water - Torque Data Comparison** | | | |
| **Reading Type** | **Mean Torque 1 Zero Adjusted (Nm)** | **Mean Torque 2 Zero Adjusted (Nm)** | **Difference between readings 1-2 (Nm)** |
| CoM 1 | -0.097611974 | -0.075136864 | -0.02247511 |
| CoM 2 | -0.158695335 | -0.148012385 | -0.010682951 |
| CoM 3 | -0.205726004 | -0.192277632 | -0.013448372 |
| CoM 4 | -0.273955299 | -0.263148182 | -0.010807116 |
| CoM 5 | -0.315504037 | -0.311616923 | -0.003887113 |
| CoM 6 | 0.214689611 | 0.236962431 | -0.022272819 |
| CoM 7 | 0.246152197 | 0.26852832 | -0.022376123 |
| CoM 8 | 0.274648879 | 0.298111864 | -0.023462984 |
| CoM 9 | 0.302301475 | 0.33127636 | -0.028974885 |
| CoM 10 | 0.328399279 | 0.354588299 | -0.026189019 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Bottle with Oil - Torque Data Comparison** | | | |
| **Reading Type** | **Mean Torque 1 Zero Adjusted (Nm)** | **Mean Torque 2 Zero Adjusted (Nm)** | **Difference between readings 1-2 (Nm)** |
| CoM 1 | 0.022271002 | -0.122434684 | 0.144705685 |
| CoM 2 | -0.043000695 | -0.184124607 | 0.141123912 |
| CoM 3 | -0.088907061 | -0.227943372 | 0.139036311 |
| CoM 4 | -0.160799467 | -0.293729047 | 0.13292958 |
| CoM 5 | -0.208515423 | -0.339288141 | 0.130772718 |
| CoM 6 | 0.383979484 | 0.230700729 | 0.153278754 |
| CoM 7 | 0.420262962 | 0.262461474 | 0.157801487 |
| CoM 8 | 0.459649539 | 0.292245005 | 0.167404534 |
| CoM 9 | 0.489199778 | 0.322302691 | 0.166897087 |
| CoM 10 | 0.515246351 | 0.343811029 | 0.171435322 |

**Chapter 5**

**Discussion**

Comparing the results of this project with the expected results of the actual liquid measurement it becomes clear that there is large error. This error could be attributed to misinterpretations with implementing the model used in (Matl, et al., 2019), as this model was used without any prior knowledge of liquid dynamics equations or experience with data processing. Errors with the data could also be attributed to factors such as inaccurate sensor data due to excessive sensor noise, lack of sensor accuracy, or improper data processing after collecting.

As stated in (Matl, et al., 2019), the robotic system that was utilised by them was a UR5, also made by Universal Robotics. Joint angle accuracy was ruled out as a possible cause of invalid data because the UR5 has a pose repeatability of ±0.1mm and this was sufficient for them to achieve reasonably accurate results. The UR10e has a pose repeatability of ±0.05mm, double the capabilities of the UR5 and therefore did not contribute to data error. This point is further reinforced by the angles reached through experimental operation of the UR10e. All positions reached were within ±0.3˚ of their intended position, however this slight difference would already have been taken into consideration by the system as the angle values were directly used in the calculations.

A far more likely reason for the inaccuracy of the model would be the inaccuracy of the force/torque sensor used during experimentation. The sensor used was mounted to the UR10e end effector and was the sensor that comes with the cobot by default. This sensor has a force accuracy of 5.5N, far lower than the accuracy of the force/torque sensor used in (Matl, et al., 2019). The sensor that was used was an ATI Axia80 EtherNet Force/Torque, which boasts a force accuracy of 0.04N and a torque accuracy of 0.002Nm. This sensor accuracy is far greater than the calculated difference between experiments shown in the table above, hence it is a reasonable assessment that if a sensor of this accuracy was used instead, the results of the experiment would have been more reliable and consistent. It was due to this low accuracy of the UR10e sensor that the mass of the bottle and liquid was gathered independently of the experiment.

The possibility of further improvements to the data processing could have positive effects on the readings coming from the force/torque sensor and aid with noise removal. The signal processing techniques used in (Matl, et al., 2019) made use of a one-dimensional Gaussian filter with a standard deviation of 5. In comparison, the technique used in this project was to format the raw data with a Matlab function that automatically removes outliers from the data that are more than 3 standard deviations away from the mean. The mean of this new data set was calculated and used as the calculated value for the torque at that reading instance.

As a result of these erroneous data readings as well as issues with the mathematical models implemented, analysis of the viscosity data collected was not performed. Future iterations of this experiment, provided they use a force/torque sensor of appropriate accuracy, would be suitable to collect this data in a meaningful capacity. The difficulties of measuring the viscosity of a liquid autonomously means that no meaningful insights could be gained during this project on other methods to investigate to provide another way to calculate this parameter.

In the case of measuring values such as liquid mass and volume, a method worthy of investigation would be measuring the liquid electrical resistance via a plunger type mechanism mounted to a robotic end effector, similar to automatic bed-leveling technology used in 3D printing. Using this method to measure liquid height within a container would make the accuracy of the system dependant on positional accuracy of the robot arm and not on sensor readings requiring further processing and denoising. It is also possible that liquid mass could be inferred from the density of a liquid with known volume. This density could be measured through the force on an object moving through the liquid being analysed through hydrodynamic equations.

**Chapter 6**

**Conclusion**

The goal of this capstone project was to develop a system that could autonomously measure the physical properties of a liquid through haptic feedback. Based upon the mathematical models and experimental methods introduced by the research paper (Matl, et al., 2019), this project attempted to implement these same methods to produce this system. System development using ROS, C++, Matlab and Polyscope took a majority of the time dedicated to this project due to having to not only learn how these systems worked themselves, but also how to make these different systems work together in a way to generate and process data from experimentation. Based upon the results gathered, the models developed in this project were not sufficient to achieve the goal of the project. Because of an insufficient sensor for the task and an incorrect implementation of the reference method, this project will require further work to achieve the correct results within a reasonable degree of accuracy. It is hoped that this report can act as a starting point for further research on this project and hopefully produce a system capable of achieving all project goals.

**References**

**Appendix**