Small-Scale Drinking Water Distribution Project for a Remote Community

By:

Rundong Liu /1072537

Battulga Batkhurel / 1074076

Haiwei Wang / 1084200

Tuan Khoi Nguyen / 1025294

Team Name: Good Good Study Day Day Up Team
Workshop 11 Group 15

ENGR10004 / Engineering / University of Melbourne

Date of Submission: 11/06/19

Abstract

This report covers the design of a water pumping, treatment and distribution system which provides clean drinking water for a small rural community. Our goal is to meet all the required specifications while ensuring the design to be practical and cost-effective in the real world. In order to do this, the iterative process is applied throughout the design to provide a variety of solutions to choose from. It outlines the experimental and design procedures used and presents the results of the experiments and finally proposes the solution. In our discussion of experimental technique and results, we highlight the discrepancies between specification values and experimental values, which are important to take into account during the design. In the design process, there are infinitely many solutions that are found to be viable. However, due to the shortage of time and resources, only a limited number of designs are taken into consideration. The final design is suitable for all the requirements, while still being practical and cost-effective. In the future, more divergent options on the research are hoped to be carried out to utilize effectiveness further.

Summary of Experiment and Design parameters

	Value
Experiment:	
Workshop 2 - Full open ball valve K value	12.67
Workshop 4 - Custom impeller max efficiency (%)	23
Workshop 8 - Average particle size, diameter (μm)	54.10
Project Design:	
Section 1 Pipe Network	
Community water requirement (L/day)	5955
Water treatment time (hr)	0.18
Operating flow rate, Q_{op} (L/s)	0.61
Section 1 pipe before reducer:	
Size	DN100
Material (e.g. Stainless Steel Schedule40)	PTFE
Absolute roughness (mm)	0.0015
Section 1 pipe after reducer:	
Size (e.g. DN80)	DN100
Material (e.g. Stainless Steel Schedule40)	PVC
Absolute roughness (mm)	0.0015
Elevation (m)	21.5
Treatment Length, L (m)	25
System head @ Q_{op} (m)	26.10
Pump Power (W)	674.42
Membrane	
Quantity of Gad-Cell 2x13	1
Quantity of Gad-Cell 4x13	1
Quantity of Gad-Cell 4x28	1
Quantity of Gad-Cell 6x28	0
Quantity of Gad-Cell 8x20	0
Quantity of Gad-Cell 8x40	0
Total membrane loss (m)	26.72
Inline Image Monitoring	
Particle concentration threshold (particles/ml)	5.84
Particle filter lifetime (months)	10.5

Project Design:		**Value**		
Section 2 Pipe Network				
		Tap 1	**Tap 2**	**Tap 3**
Elevation (m)		**10.5**	**12.5**	**12.5**
Section 2 main pipe:				
Size		DN20	DN20	DN20
Material		PVC Schedule 40	PVC Schedule 40	PVC Schedule 40
Absolute roughness (mm)		0.015	0.015	0.015
Total pipe length (m)		20.15	26.25	26.25
K-Value	*Fittings*	*Quantity*	*Quantity*	*Quantity*
0.75	Entrance	1	1	1
0.35	45 elbow	0	0	0
0.75	90 elbow	6	4	5
0.4	Through tee	0	1	2
1	Branch tee	1	0	0
1	Exit	1	0	1
Reducer K-value		N/A	0.807	0.807
Section 2 secondary pipe, after reducer:				
Size (e.g DN20)		N/A	DN15	DN15
Material (e.g. Stainless Steel Schedule40)		N/A	Schedule 40 PTFE	Schedule 40PTFE
Absolute roughness (mm)		N/A	0.015	0.015
Total pipe length (m)		N/A	41.5	61.5
K-Value	*Fittings*	*Quantity*	*Quantity*	*Quantity*
0.75	Entrance	N/A	0	0
0.35	45 elbow	N/A	0	0
0.75	90 elbow	N/A	2	3
0.4	Through tee	N/A	0	1
1	Branch tee	N/A	1	0
1	Exit	N/A	1	1
Absolute pressure at tap (kPa)		200.3kPa	209kPa	207kPa
Gauge pressure at tap (kPa)		99.3kPa	108kPa	106kPa

Table of Contents

1. Introduction

This report proposes a detailed design solution to the problem of supplying water to a small remote community, adapting the framework from previous designs to similar problems.

Currently, the inhabitants are using a system of buckets and ropes to retrieve drinking water from an underground well. However, this system is failing to meet the water requirements of a growing population and has started to cause safety issues in terms of water quality.

As such, this project will devise a new solution in order to rectify those problems, using the framework below:

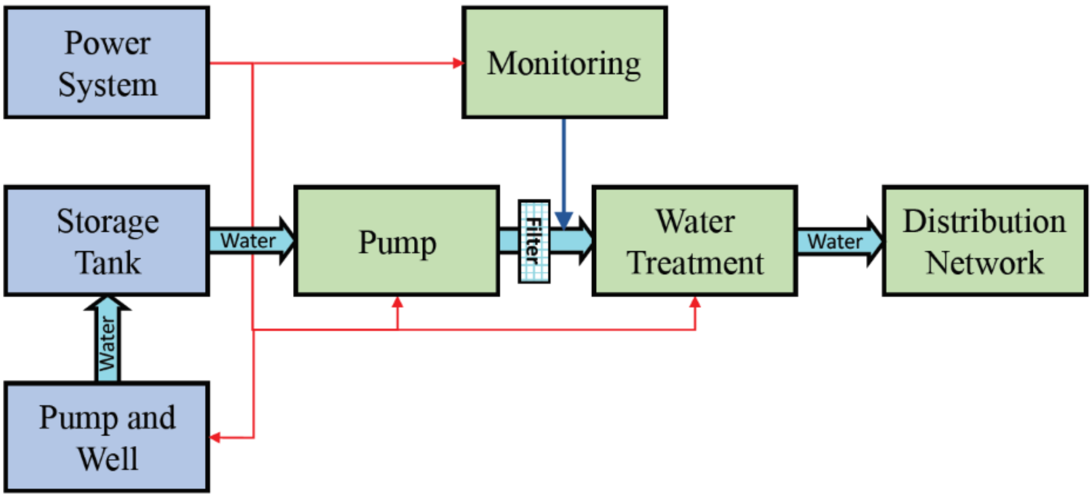

Figure 1: Diagram of various subsystems in the design and how they interact.

The aim is to automate the delivery of water from the well to the consumers and to filter and disinfect the water, with a design that will last for at least 10 years.

This report will cover sequentially:

- The experimental and design method used.
- The results from experiment.
- The design solution based on the experiments.

2. Method

2.1. Management of Project

In the formation of the team, the goals and standards to achieve are identified using the SMART criteria for goal setting. Using this, a time-constrained strategy is developed.

In the experimental phase, the team was split into the roles of technician and data analyst. The technicians are responsible for the initialisation and safe operation of the experiment, while the data analysts process the data by creating MATLAB scripts and storing the outputs of the data to be used in the future.

The structure of the design process focuses on specialisation. Each member was responsible for the design of a subsection, becoming an expert in that area, then deliver that knowledge and information to other team members during team meetings for evaluation. The meeting times are presented in a table in Appendix A.

In the drafting of the report, each member contributes by inputting their parts. However, to maintain consistency, a head editor is appointed to do the final edit of the report.

2.2. Experimental Methods

2.2.1. Fluid Rig, Measuring the Pressure Drop from a Ball Valve

An experiment was done to determine the K-value (Resistance coefficient) of a ball valve of the Fluid Rig.

After preparing the water system used for the experiment, the ball valve fracture opening was varied to obtain different flow rates across the system and different pressure drops across the ball valve.

The ball valve was fully opened at first and was reduced by approximately 5 degrees each time until the flow rate dropped down to 0.

General steps of measuring and recording:

1. Leave the ball valve fully open
2. Once the flow rate of the system is stabilised, record the flow rate across the system and the pressure drop across the ball valve
3. Reduce the ball valve opening by 5 degrees
4. Repeat step 2
5. Repeat step 3 and 4 until the flow rate drops to 0

In some cases, no matter of the time period passed to allow the system flow rate to stabilise; the flow rate kept fluctuating between two values with a small difference. The frictional effects of the pipes could have caused flow rate instability. To reduce random errors, the mean of two fluctuating values were recorded.

In order to determine the Resistance coefficient of the fully open valve, the recorded values of the flow rates and the pressure drops at different ball valve openings were put into the MATLAB script (Appendix B1).

2.2.2. Methods of drawing/designing a custom impeller

For the water system design, a custom impeller was designed using 3D CAD software Inventor. Prior to designing, a .ipt file, called impeller_hub.ipt was provided.

After opening the impeller_hub.ipt file in Inventor, the following general steps were followed:

1. Use an Offset Plane
2. Using 2D Sketch tools, draw a blade cross section
3. Set dimensions and constraints

Constraints according to the pump specification:

- Each blade has a thickness of 2mm
- Blades have an entrance and exit angles of 41^0 and 36, which were obtained by using the impellerangles.m (see Appendix B9)
- Blades must be 4mm above from the base of the hub
- Blades must stop at the top of the hub
- Blades cannot block inside of the hub

4. Then use Extrude tool to make a single blade with a length of 19mm
5. Locate the blade to the proper place in Browser, which is right below "Put vanes between these".
6. Add fillets with a radius of 1mm on the outside edges of the blade for smooth edges
7. Locate the Fillet to the proper place in the Browser, which is right below the Blade
8. Use Circular pattern to automatically populate the impeller hub with the 6 evenly spaces blades required

9. Locate the Circular pattern to the proper place in Browser, which is right below the Fillet

Images (Taken from Autodesk Inventor):

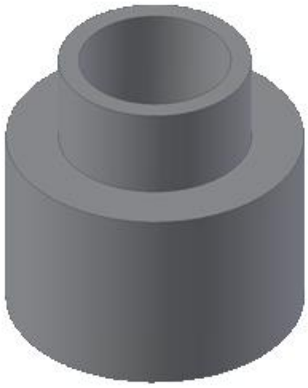

Figure 2: The hub base template provided by the test pump manufacturer, upon which the impeller blades were added.

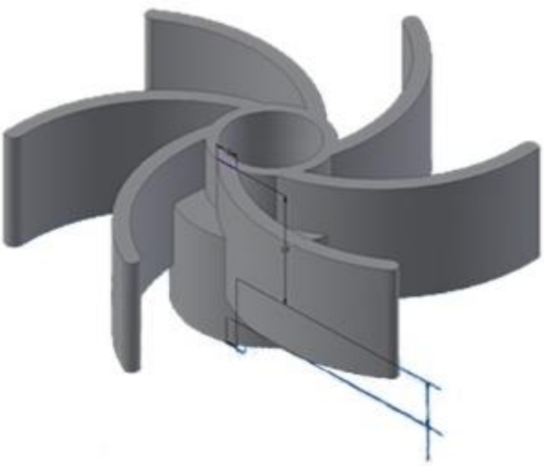

Figure 3: The completed impeller.

2.2.3. Fluid Rig, Method for 3D printing an impeller

MakerBot Replicator+ 3D printer was used to 3D print copies of the custom impeller.

General steps:

1. Open the MakerBot Print software
2. Insert and load converted STL version of the impeller design onto the print area
3. Adjust the orientation of the model by using the "Orient" function
4. Duplicate the impeller to a total of 4 by right-clicking and selecting Duplicate
5. Rearrange all impellers to fit into the print area

After setting up the impellers for printing:

6. Select Replicator+ as the printer
7. Use "Print Settings" from the right panel to set up the following:

- Infill Density = 10%
- Layer Height = 0.2mm

Once all the settings are finalised, save the project as a .print file.

8. Submit the .print file for printing

2.2.4. Fluid Rig, Measuring the Pump Curve and Efficiency for an Impeller

An experiment was done to measure the efficiency of the custom impeller.

Before the experiment, the stock impeller in the pump was replaced with the custom impeller. After preparing the fluid rig system used for our experiment, the pressure drop across the pump and flow rate across the system were measured at different fracture openings of the ball valve.

The ball valve was fully open at first and was reduced by approximately 5 degrees each time until the flow rate dropped down to 0.

General steps of measuring and recording the pump curve:

1. Leave the ball valve fully open
2. Once the flow rate of the system is stabilised, record the flow rate and the pressure drop across the pump
3. Reduce the ball valve opening by 5 degrees

4. Repeat step 2
5. Repeat step 3 and 4 until the flow rate drops to 0

Just like section 2.2.1, in some cases, flow rate instability could have been caused by the frictional effects of the pipes. To reduce random errors, the mean of the 2 values were recorded.

In order to determine the maximum efficiency of the custom impeller, the recorded values of the flow rates and the pressure drops at different ball valve openings were put into the MATLAB script (see Appendix B9).

2.2.5. Image Monitoring Rig, Calibration and Particle Detection

Distance Calibration:

An experiment was done to determine the image resolution of the camera & lens system, which will be used to detect particles in fluids in the design.

MATLAB 2016b Image Acquisition tool was used to capture images of a scale bar on a calibration strip on a petri dish. After adjusting the camera to ensure the view of the scale bar is as clear as possible, an image was captured and exported for analysation.

On the exported image of the bar scale, the 'getpts' MATLAB function was used to find the number of pixels between adjacent scale bars. The measurements from multiple scale bars were averaged to reduce reading errors.

Using this average value, the distance per pixel can be calculated by:

$$Resolution = \frac{Distance}{N_{pixels}}$$

Equation 1: Calculation of the resolution for distance calibration.

Particle size detection:

A binary mask is created by specifying an appropriate threshold value.

The binary mask shows white, where there are supposed to be particles, and black, where there is not. This can be achieved by a Boolean expression. For example, if the particle is darker than the background, then the binary mask is set to be 1 when the value is less than the threshold and vice versa for particles lighter than the background.

The MATLAB functions ‘bwconncomp’ and ‘regionprops’ determines the area and length of the particles as the number of pixels. By multiplying the length in pixels with the resolution and the area in pixels squared with the resolution squared, the physical lengths and areas of the particles are found.

However, the binary mask may contain false particles which are either very large or very small; therefore, they are removed by another Boolean expression with a lower limit of 100 um and upper limit of 10000um.

The length of a particle is given by the average of the major axis length and the minor axis length. The average of lengths and areas are determined by using the ‘mean’ function in MATLAB.

2.3. Project Design/Modelling Methods

2.3.1. EBE analysis

This equation allows the pressure at a point of interest in the pipe system to be calculated, given all other variables:

$$\alpha \frac{{v_1}^2}{2g} + h_1 + \frac{p_1}{\rho g} = \alpha \frac{{v_2}^2}{2g} + h_2 + \frac{p_2}{\rho g} - \frac{W_s}{g} + \frac{l_v}{g}$$

Equation 2: The general form of the engineering Bernoulli equation.

The loss term is given by:

$$l_v = \sum_{pipes} \frac{fLv^2}{2d} + \sum_{fittings} \frac{1}{2} K v^2$$

Equation 3: Expression for calculating the frictional loss term in equation 2.

The calculation of f is in presented in Appendix C4.

A list of K-values for different fittings is found in Appendix C3.

2.3.2. Membrane Process Simulation

The membrane simulator is used to determine the area required in order to obtain the requisite ozone concentration in the water for the specified conditions. Specific details of the variables involved are in appendix C6.

2.3.3. Mass Balance Analysis on Hollow Fibre Membrane Modules

The mass flow rate of ozone into the water is be obtained by:

$$\dot{m} = Q_w(C_{out} - C_{in})$$

Equation 4: Mass-flow from the gas phase into water.

There is no accumulation of ozone in the membrane, so the mass flow into the water is equal to the mass flow out of the gas, using Henry's law:

$$\dot{m} = Q_g(\frac{P_{in}}{H} - \frac{P_{out}}{H})$$

Equation 5: Mass-flow expression for the gas phase, here Henry's law is used to express concentration in terms of partial pressures of ozone.

Thus, the partial pressure of ozone at the gas outlet can be determined by equating these two expressions.

2.3.4. Contactor Section of Water Disinfection Design

Based on a fixed pipe diameter and a flowrate range dependent on the operation time of water treatment, the velocity of water through the pipe was calculated. With those velocities, the length of this section was varied to obtain the time the water spends in this section. Using the Chick-Watson law (see Appendix C1), a table of log inactivation credits is determined.

2.3.5. Image Analysis Methods

In this section, the particle's velocity is inferred from the lengths of the streaks. The lengths of the streaks are determined using the same method as in section 2.2.5, i.e., creating the binary mask and then using the in-built MATLAB functions.

The camera's exposure time determines the length of the streaks, and the length of the streak must not exceed the width of the torch beam, else the velocity reading will be incorrect as the actual length of the streaks would not be captured.

Appropriate exposure time in this experiment was found to be 500ms.

Assuming that the particle is a point-liked object, the length divided by the exposure time will give the particle velocity.

The mean particle velocity is then found by averaging the particle velocities found for all particles in the frame.

3. Results

3.1. Pump and Water Distribution Network

3.1.1. Determining the K values for a Ball Valve

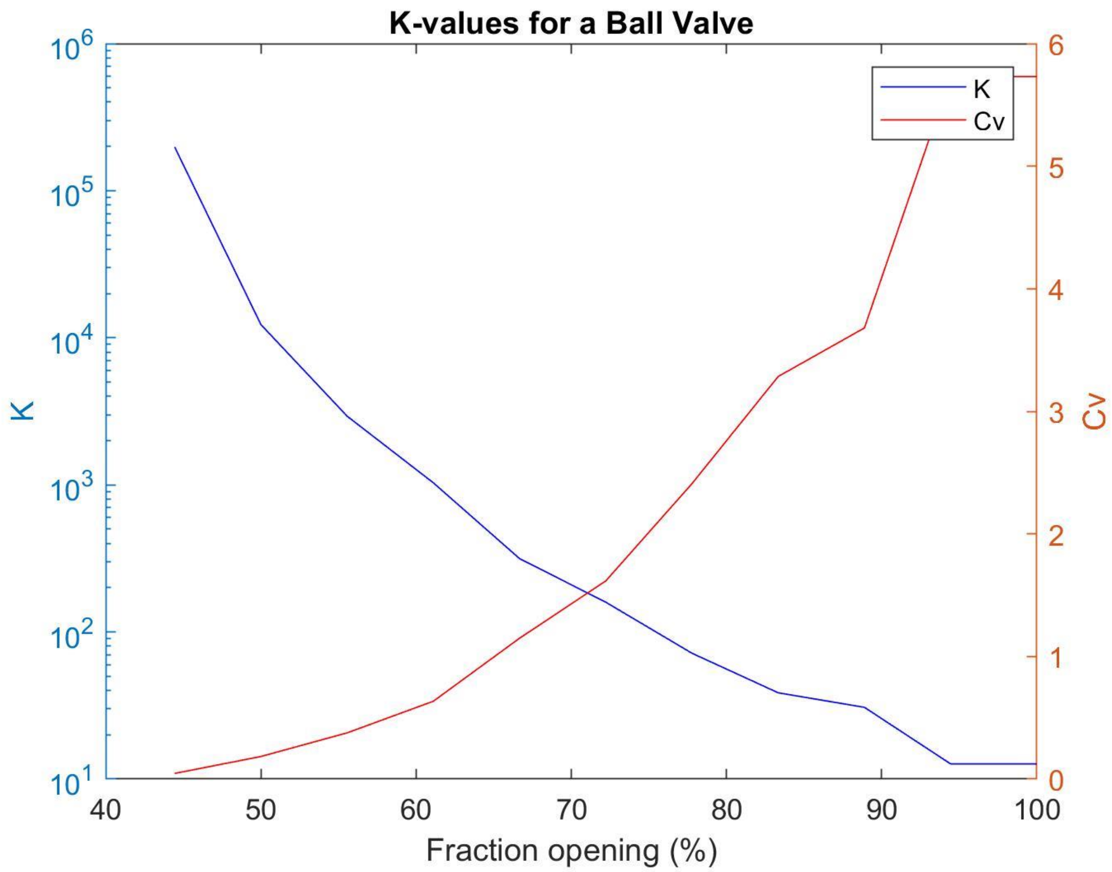

Figure 4: Plot of K-values and Cv values for a ball valve at different fractions of opening. The K-value for maximum opening is 12.67.

The K-value was determined to be 12.67, comparing this to the K-value of a fully open globe valve which is 6.0, it is almost twice as much. The ball valve uses a perforated ball to control the flow (Aikuele 2015), this structure could give rise to increased resistance as the flow goes through expansion then compression as it passes across the ball.

Another source of error which could have increased the resistance is the extra pipe friction loss, which was not accounted for in the pressure measurement.

The data used in the calculation and the MATLAB script for the calculation is found in Appendix B1.

3.1.2. Performance Pump Curve and Efficiency for a stock impeller

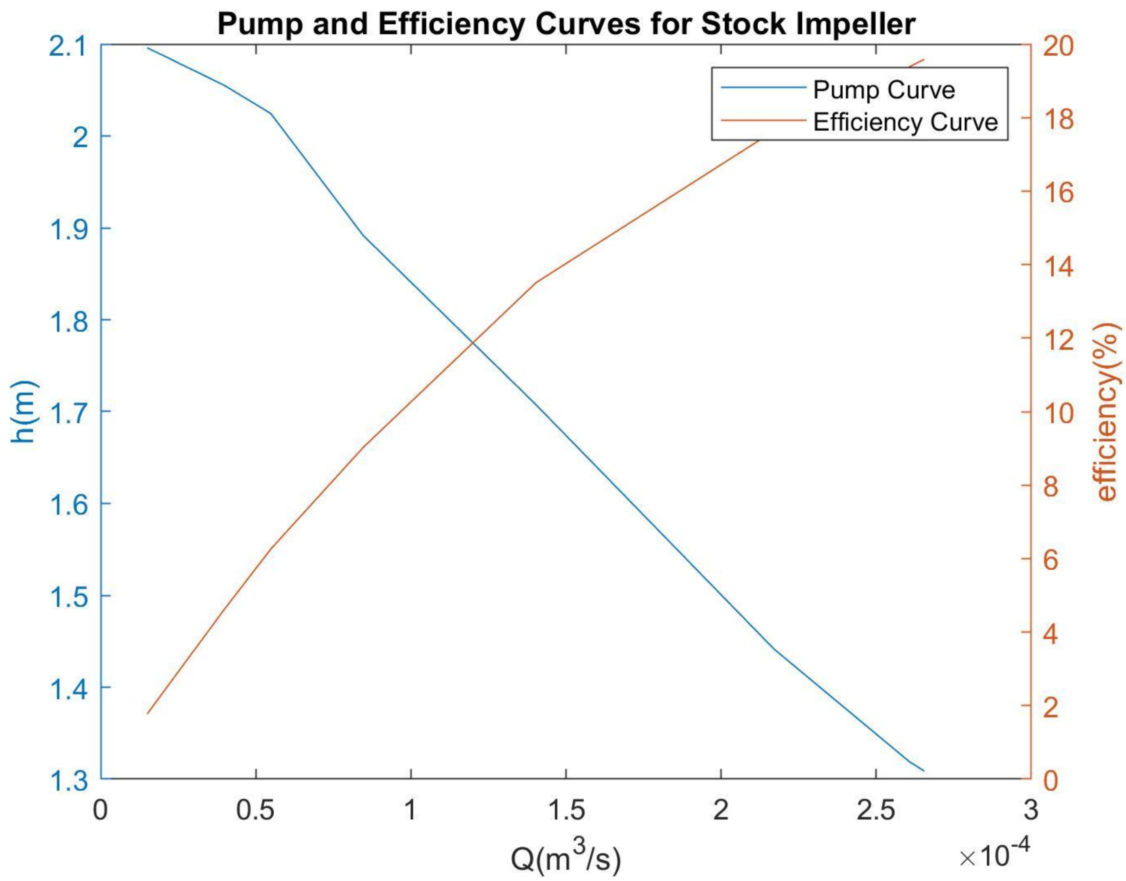

Figure 5: The pump and efficiency curve for the stock impeller.

The pump pressure gauge had an uncertainty of ±5mbar, which equates to ±0.051m pump head. The uncertainty for pump head from 2.35m to 1.50m in percentage is between 2.2% and 3.4%.

Maximum efficiency of the pump for the custom impeller was found to be around 23%±0.506%, whereas maximum efficiency of the pump for the stock impeller was 19%±0.352%.

Therefore, the custom impeller is more efficient than the stock impeller by around 4%, which may have been the result of the methods stated in 3.1.3.

The MATLAB script used to generate these curves are displayed in Appendix B2.

3.1.3. Geometry for a 3D Printed Impeller

As open impellers are subject to continual end thrust, mathematical formulae for Slip factor (ScienceDirect) was used to obtain the best entrance and exit angles based on the impeller diameter. So that the water can be propelled out of the pump without breaking the impellers while ensuring maximum efficiency.
The calculation was done in a MATLAB script which can be found in Appendix B9.

Impeller parameter	
Vane thickness	2mm
Vane height	10mm
B1	41^0
B2	36^0
Diameter	38mm

Table 1: Custom impeller dimensions and geometry.

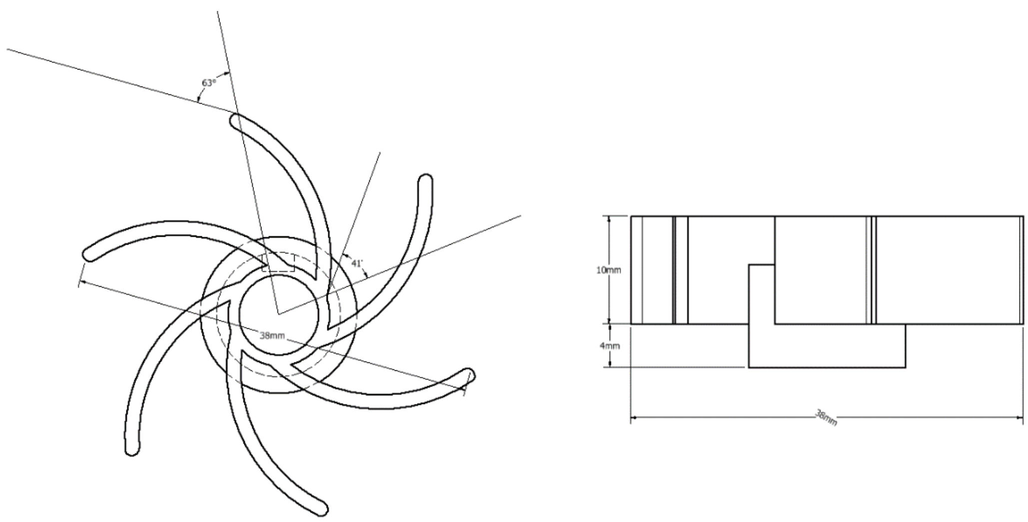

Figure 6: Drawings of the final custom impeller design.

The largest possible diameter and vane height were selected, as it increases the surface area to transfer momentum to the water, allowing the shaft power to transform kinetic energy into hydraulic pressure more efficiently.

The largest possible diameter and vane height are used because they increase the surface area to transfer momentum to the water, allowing the shaft power to transform into hydraulic pressure more efficiently.

3.1.4. Performance Pump Curve and Efficiency for the 3D printed Impeller

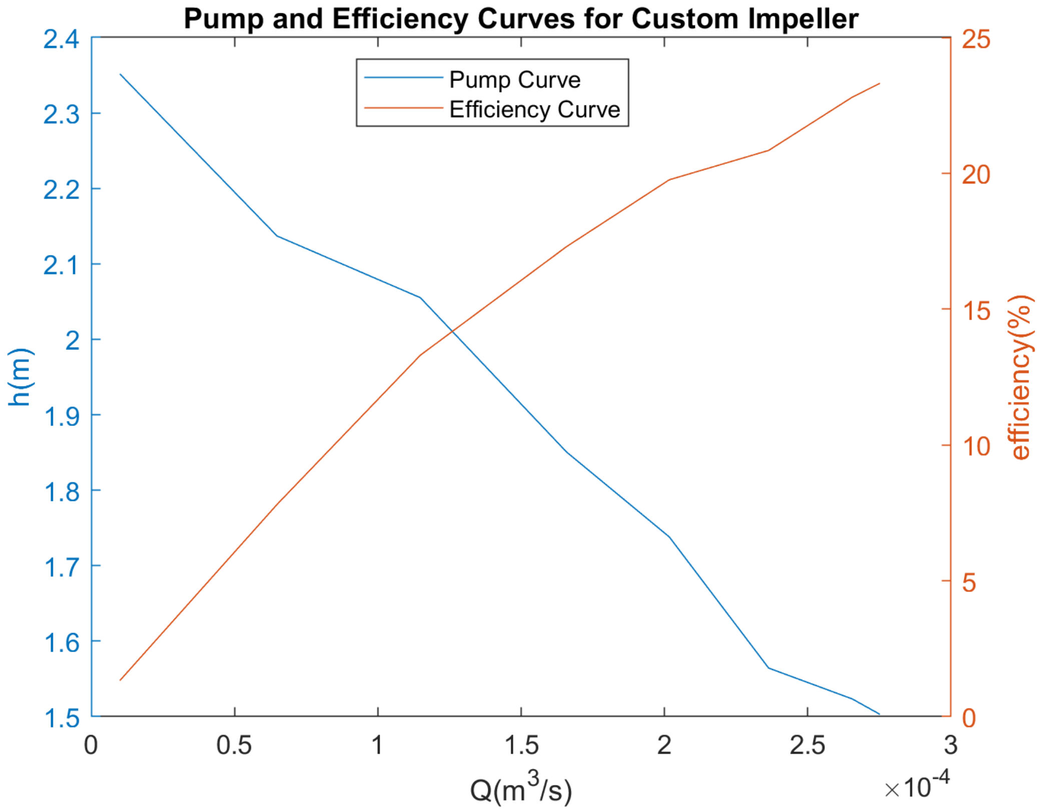

Figure 7: The pump and efficiency curve for the custom impeller.

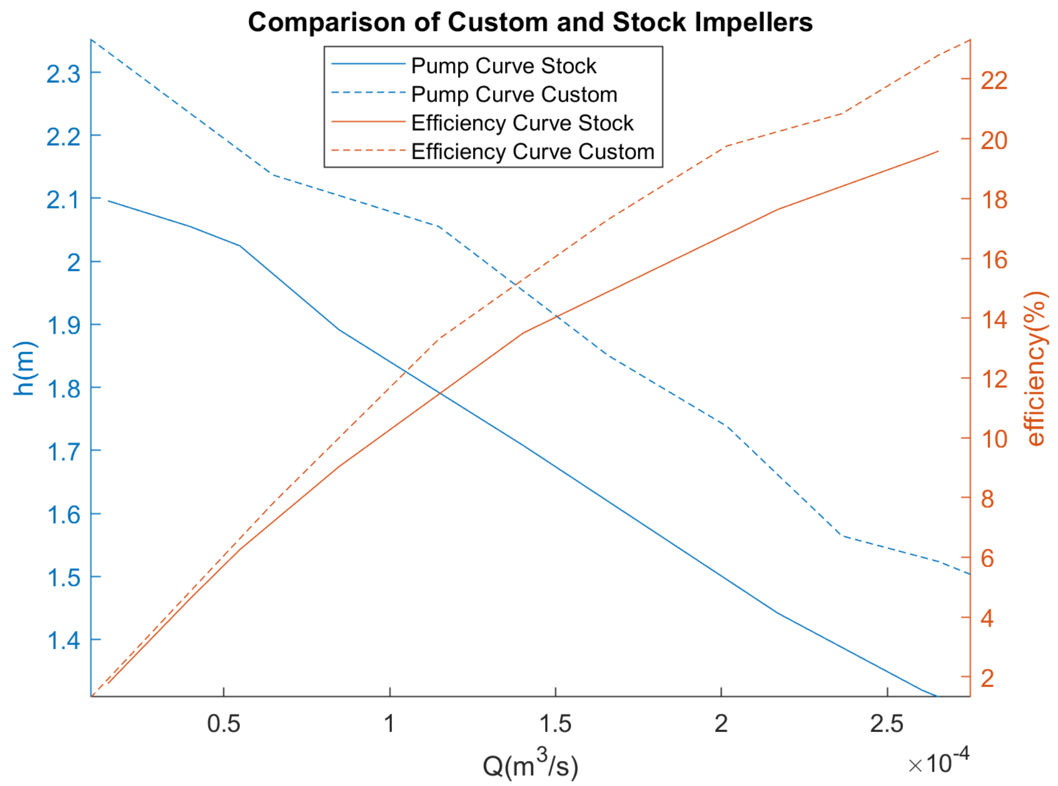

Figure 8: Graph showing pump and efficiency curves for both the stock and custom impeller.

The pump pressure gauge has an uncertainty of ±5mbar, which equates to ±0.051m pump head. Uncertainty in percentage is between 2.2% and 3.4%, for pump head from 2.35m to 1.50m.

Maximum efficiency of the pump for the custom impeller is 23%±0.506%, whereas the maximum efficiency of the pump for the stock impeller is 19%±0.352%.

The custom impeller is more efficient than the stock impeller by around 4%, which could have been the result of the methods stated in 3.1.3.

The increased efficiency could be due to the increased size of the impeller blades, allowing more water to be pumped, or it could be due to the backward-facing curvature of the blades, which could increase the tangential velocity of the water around the pump, thus increasing the pressure.

3.2. Membrane Process Simulation

3.2.1. Ozone Feed Volume Fraction

The volume fraction in the ozone feed is 0.25% v/v.

3.2.2. Changing the Water Volumetric Flow Rate

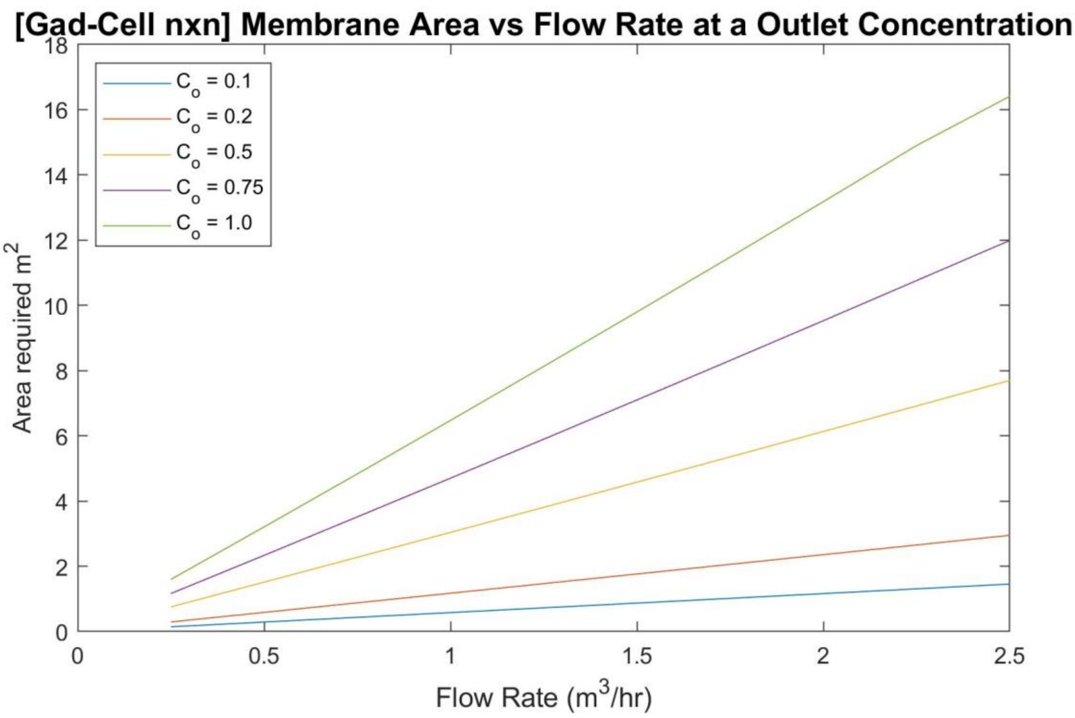

Figure 9: Plot of membrane area vs outlet concentration at fixed flow rates.

Increasing the flow rate increases the velocity of water through the membrane and thus decreases the time of contact. The results show as expected; that more area of contact is required to achieve the same outlet concentration for higher flow rates.

3.2.3. Changing the Outlet Concentration

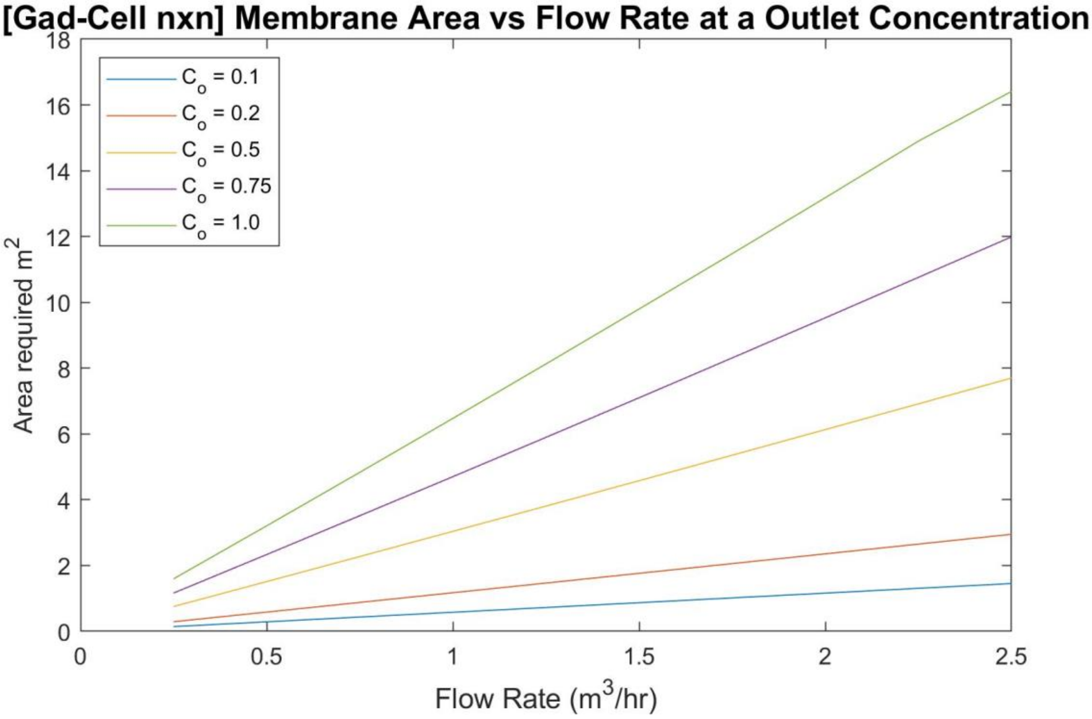

Figure 10: Plot of membrane area vs flowrate at fixed outlet concentrations.

As the outlet concentration required, the area of contact required also increases.

3.2.4. Changing the Water Volumetric Flow Rate and Outlet Concentration

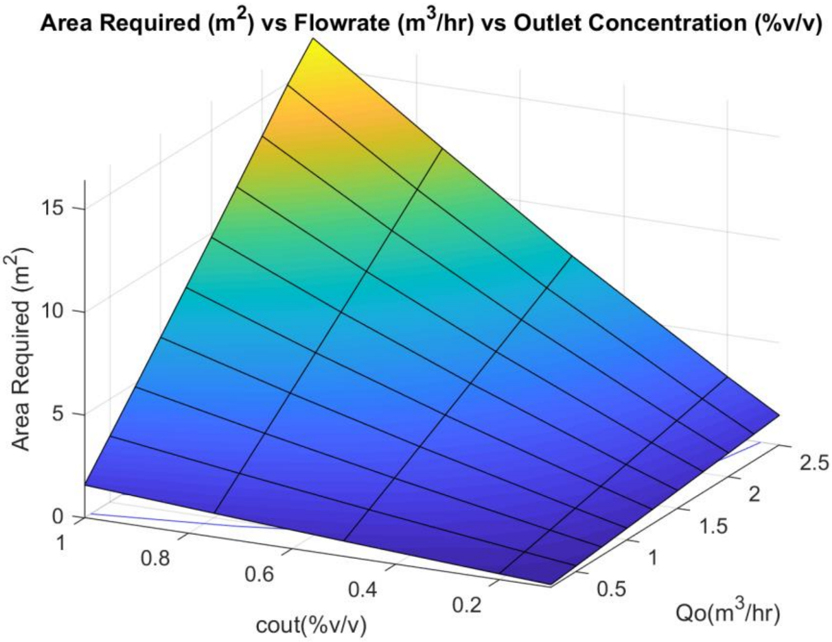

Figure 11: Surface plot of the area of membrane required vs both flow rate and outlet concentration.

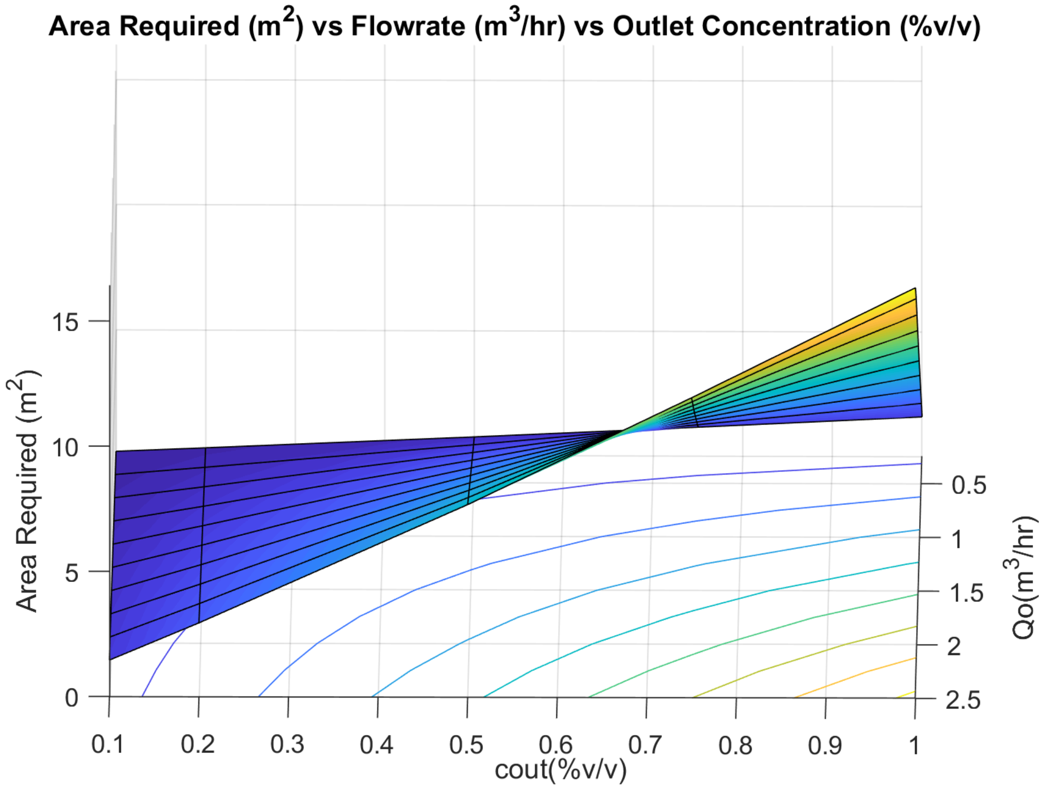

Figure 12: Rotated view of figure 9, showing the projected contours of the lines of equal area required.

Figure 10 shows the contours of equal areas lie along curves; this shows the fact that flow rate and required outlet concentration have a different impact on the area (else it would look more like straight lines).

These plots give a qualitative understanding of the way that certain constraints change the amount of membranes needed. In particular, they show that increasing flow rate will drastically increase the membrane area required, which is a piece of critical information for the selection of operation time in the design.

The MATLAB script for generating these results can be found in Appendix B3.

3.3. Image Monitoring

3.3.1. Camera Calibration

Camera Resolution	$\mathbf{20.2\ \mu m\ /\ pixel}$
Microscope Resolution	$\mathbf{8.33\ \mu m\ /\ pixel}$

Table 2: Results obtained from imaging scales, used in the distance calibration. These results are used in the subsequent sections for the size detection algorithm.

3.3.2. Particle sizing

Microscope:

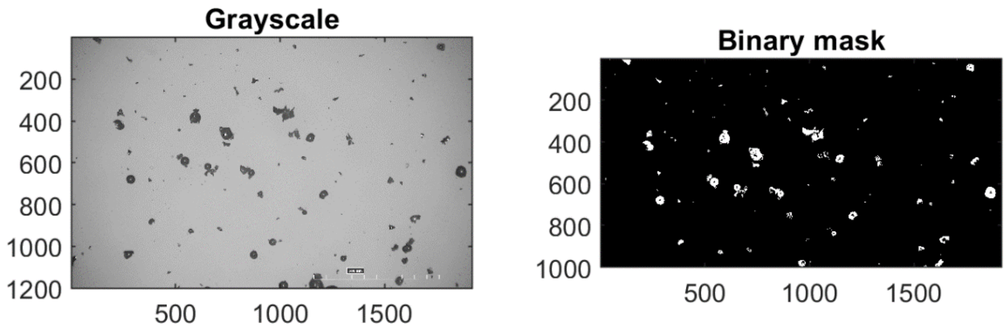

Figure 13: Grayscale image of silver particles under the microscope (left). Binary mask applied to the image with a threshold intensity of 100 (right).

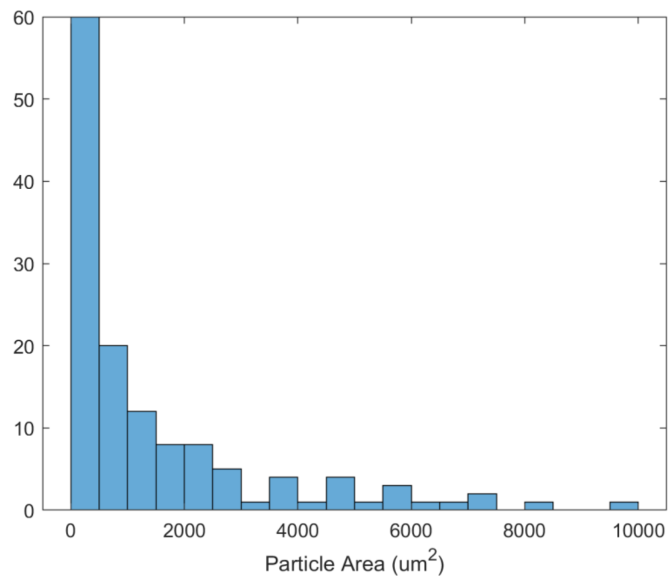

Figure 14: Histogram of the area distribution using 20 bins from image processing of the microscopy image.

Average Length	45.5 μm
Average Area	$1.49 \times 10^3 \mu m^2$

Table 3: Average sizes from the microscopy image.

Camera:

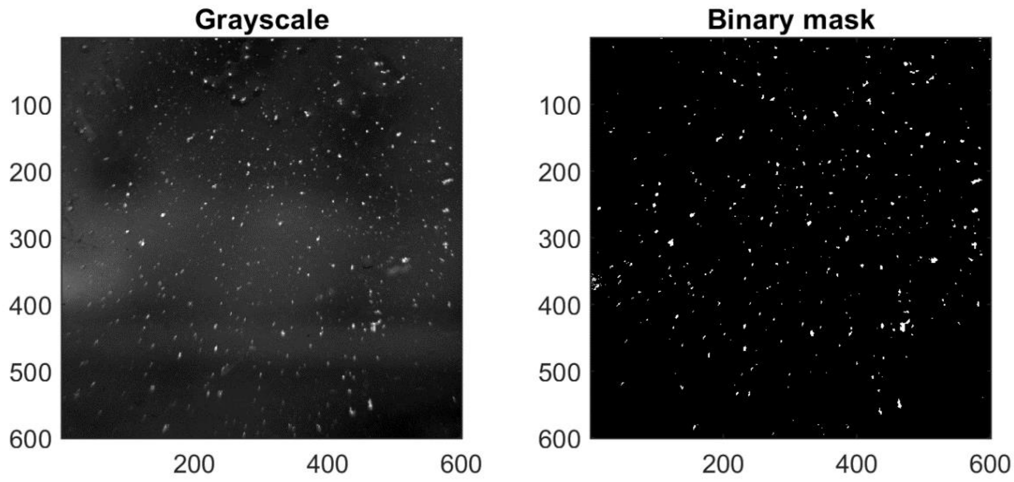

Figure 15: Grayscale image of silver particles under the camera (left). Binary mask applied to the image with a threshold intensity of 120 (right).

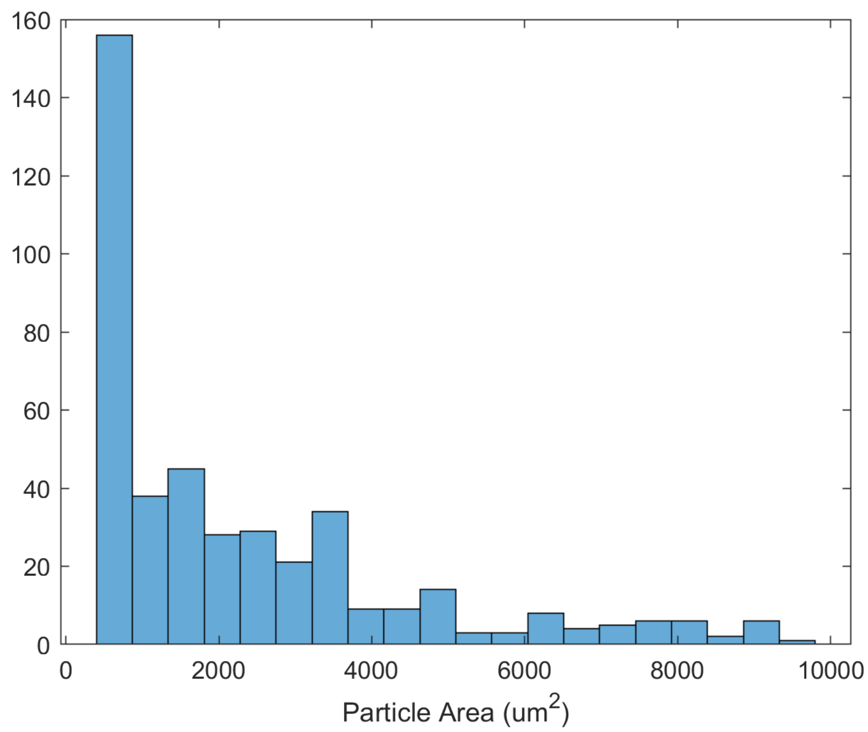

Figure 16: Histogram of the area distribution using 20 bins from image processing of the camera image.

Average Length	54.1 μm
Average Area	$2.27 \times 10^3 \mu m^2$

Table 4: Average sizes from the camera image.

3.3.3. Velocity and density calculations

The torch beam volume was estimated to be $8.94 cm^3$; this value is used in the density calculations. The subsequent processing was completed in MATLAB, see appendix B11.

Single Frame:

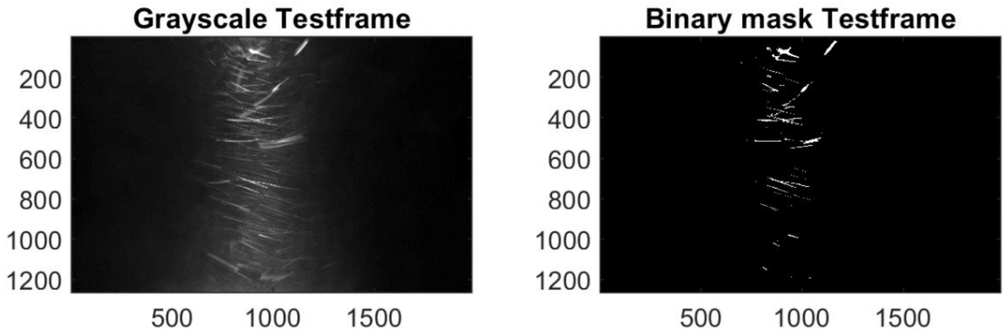

Figure 17: Image of moving particles taken with the camera (left). Binary mask with a threshold of 120 (right).

Average velocity	$0.003\ ms^{-1}$
Average density	$23.36\ particles/cm^3$

Table 5: Average velocity and particle density detected for a single frame.

Ten frames:

Frame	Mean Velocity (ms^{-1})	Mean Particle Density ($particles/cm^3$)
1	0.003	20.91
2	0.003	19.25
3	0.004	24.89
4	0.004	23.49
5	0.003	21.27
6	0.004	25.38
7	0.003	19.57
8	0.004	18.23
9	0.003	24.76
10	0.003	25.83

Table 6: Average velocity and particle density detected for ten different frames.

3.3.4. Particle density detection loop

The threshold for stopping was 25 $particles/cm^3$.

Frame	Particle Density
1	23.2
2	20.9
3	17.3
4	23.5
5	18.1
6	21.3
7	25.3

Table 7: The results for running the particle detection code (Appendix B12), the code terminated as expected when the particle concentration exceeded 25 in frame 7.

The while loop terminated at frame 7; the run time was 7.14 seconds as reported by MATLAB.

4. Design Problem Solution

4.1. Community Water Consumption

The water requirement is calculated by accounting for the maximum water required for the population over 10 years, the clinic bed and clinic area water requirements. Research shows that on average, people drink 2 litres of water per day (Gunnars, 2018).

The calculation can be found in Appendix C1. The requirement per day is 5955L.

4.2. Pump System – Section 1

4.2.1. System Curve

4.2.1.1. Pipe Material, Schedule and Diameters

The pipes used in the design are all schedule 40 pipes, this was decided due to the fact that the pipes do not need to withstand high pressures or excessive stresses as the system is built in a stable, temperate environment. Increasing to schedule 80 thickness would have higher cost implications, which are unnecessary.

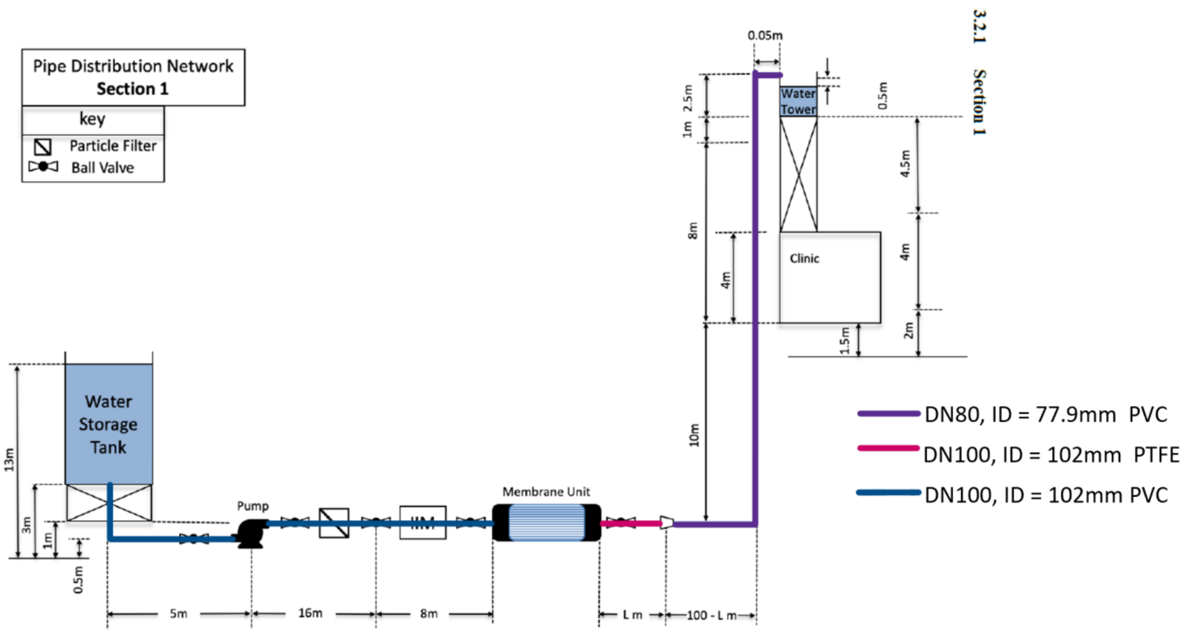

Figure 18: Diagram of the pipe material selections for section 1.

DN100 PVC pipe is used from water tank to membrane, as this material is durable, long-lasting and widely used commercially.

Ozone has a strong corrosive ability, so DN100 PTFE pipe is used in the contactor section as PTFE has strong ozone resistance and is relatively durable.

DN80 PVC pipe is used to transfer the water up to the water tower from the contactor section. In this section, there is a small amount of ozone still present in the water; however, PVC also has satisfactory ozone resistance, thus it will withstand corrosion, simultaneously having the rigidity to be placed vertically.

4.2.1.2. EBE analysis

$$SystemHead = h_2 - h_1 + \alpha \frac{{v_2}^2}{2g} + h_{pipe} + h_{fittings} + h_{membrane}$$

Equation 6: The form of the EBE used in calculating the system curve.

The EBE analysis is performed for different system flowrates in MATLAB; the calculations are found in the script in Appendix B6.

The various parameters used in calculating loss contributions are outlined in the next two sections. The data is obtained from figure 14. Note that the total lengths and K-values will be split as the friction factor and velocities are different for different pipe diameters.

4.2.1.2.1. Pipe losses

Section A (DN100):

Section A (DN100)	
Inner Diameter	102mm
Length	56.5m
Section B (DN100)	
Inner Diameter	77.9mm
Length	96.55m

Table 8: Values used in calculating pipe frictional loss for section 1.

4.2.1.2.2. Fittings

Fitting Type	Number	K-values
Section A (DN100)		
90-degree elbow	1	0.75
Entrance	1	0.75
Ball valve (fully open)	5	63.35
Reducer (102mm ID -> 77.9mm ID)	1	*Calculated in code
TOTAL		$65.6 + K_{red}$
Section B (DN80)		
90-degree elbow	2	1.5
Exit	1	1
TOTAL		2.5

Table 9: Values used in calculating fitting losses for section 1.

4.2.1.2.3. Height Change

The height increased by 19.5m. This value is added to the system head value.

4.2.1.2.4. Membranes

*The pressure drop presented here are for the operating flow rate of 2.2283m^3/hr, it will vary for different flow rates.

Membrane	Re	Pressure drop (kPa)
1 Gad-cell 2x13	2521	121
1 Gad-cell 4x13	1585	48.4
1 Gad-cell 4x28	1585	92.6
Combined		262

Table 10: Membrane pressure drop for section 1, the choice of membranes is presented in section 4.4.2.3.

To find the head loss due to the membranes, divide the pressure by the density of water and the acceleration due to gravity. This is done in the code in Appendix B5.

4.2.1.3. System Curve Plot

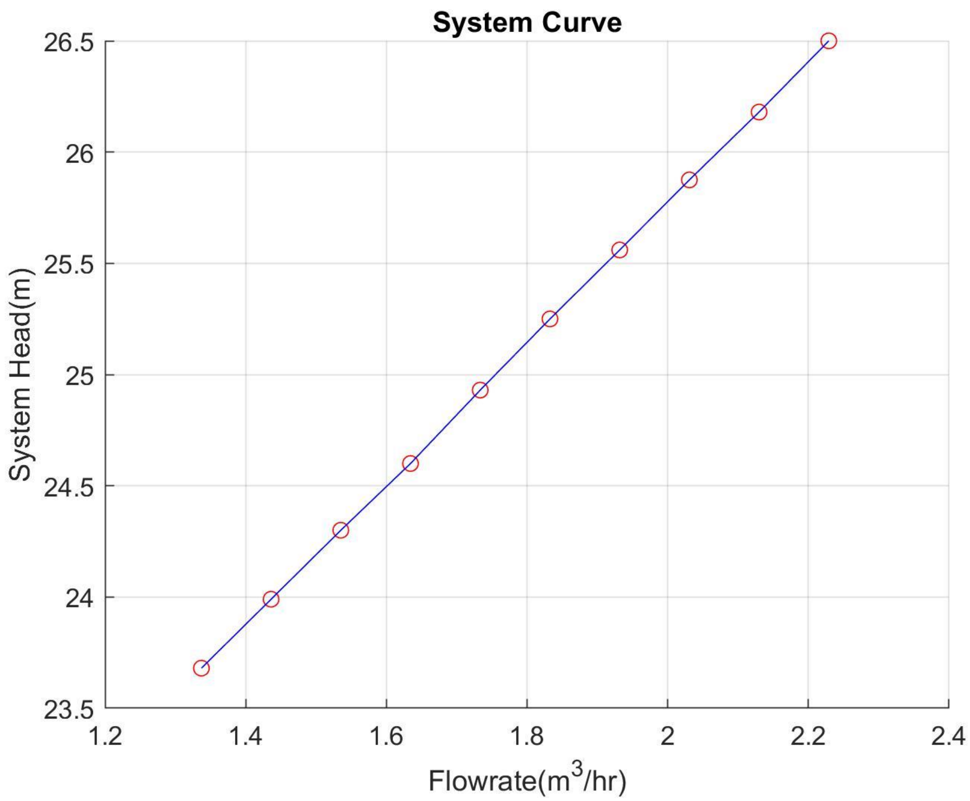

Figure 19: Final system curve calculated using the parameters given above. The calculations are found in Appendix B6.

The most significant contributor to system head is the extra height of the endpoint, however, it is constant regardless of the flow rate. The frictional and velocity heads are dependent on the square of the velocity, thus also on the square of the flow rate. Here the quadratic behaviour is not clearly seen and the graph is roughly linear because the flow rate is relatively low.

4.2.2. Impeller Scale up and Operational Flow Rate

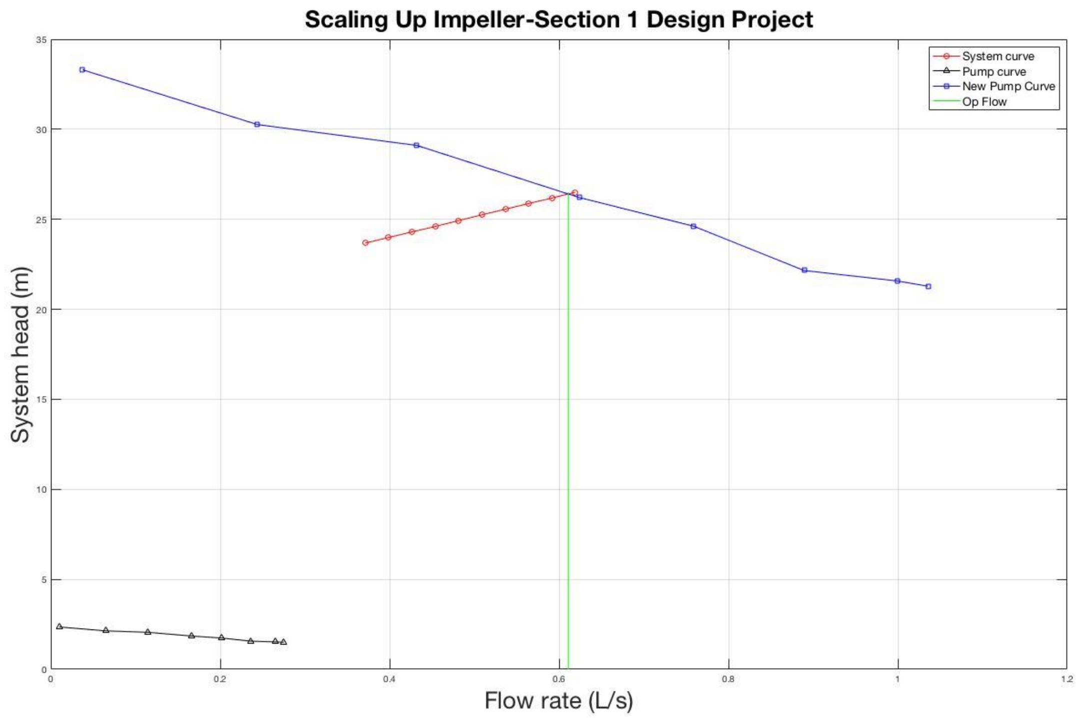

Figure 20: Graph showing the experimental pump curve, scaled up pump curve, the system curve and the operating point.

In order to meet the operational flow rate of 0.61 L/s, the custom impeller was scaled up to a diameter of 143mm, which has a maximum efficiency of 23%.

Using the equation Power = $\frac{p\dot{V}gh_p}{\eta}$, the power required for the pump at the operational flow rate was found to be 674.42W.

MATLAB script used to obtain the system and pump curves is in Appendix B14.

Scaled up impeller has a diameter of 143mm, which is much larger than some centrifugal pumps used for flow rates between 1.4 - 2.3 m^3/hr. The impeller could be larger, as the custom impeller is an open type impeller, which usually requires higher NPSHR values in practice (PetroWiki).

4.3. Water Distribution Network – Section 2

The height of the water tower provides the necessary pressure to drive flow. In this module of the project, the interplay between the pressure in the pipes from pumps and the losses that occur from the friction of the water with the pipe drives critical design decisions.

4.3.1. Pipe Distribution network

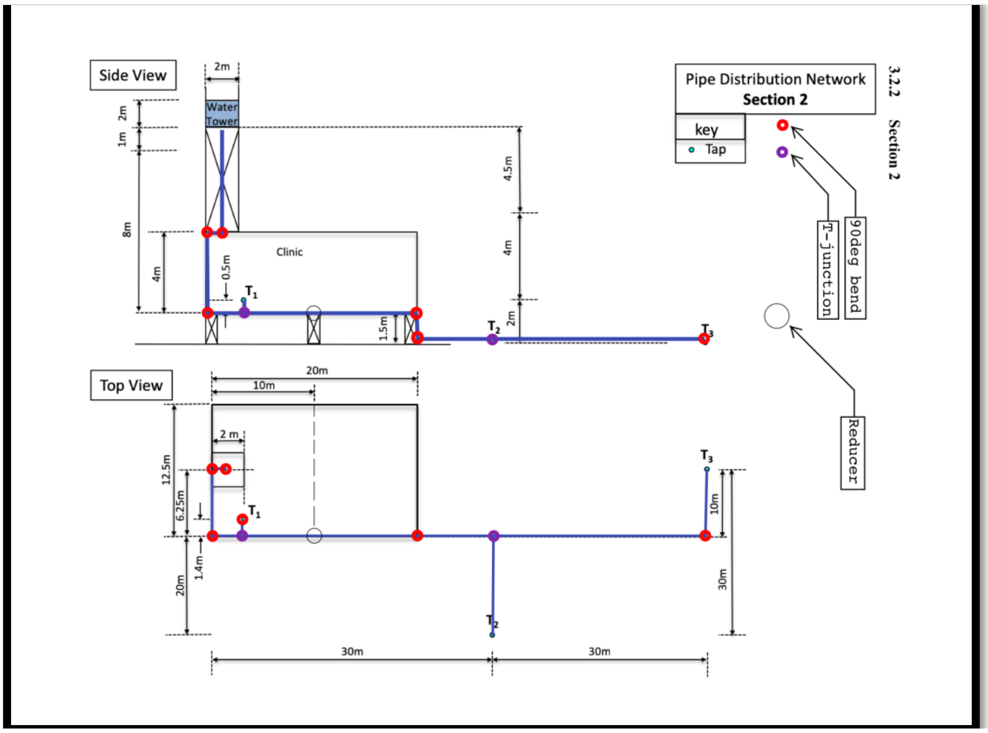

Figure 21: Pipe distribution network, 2D plot

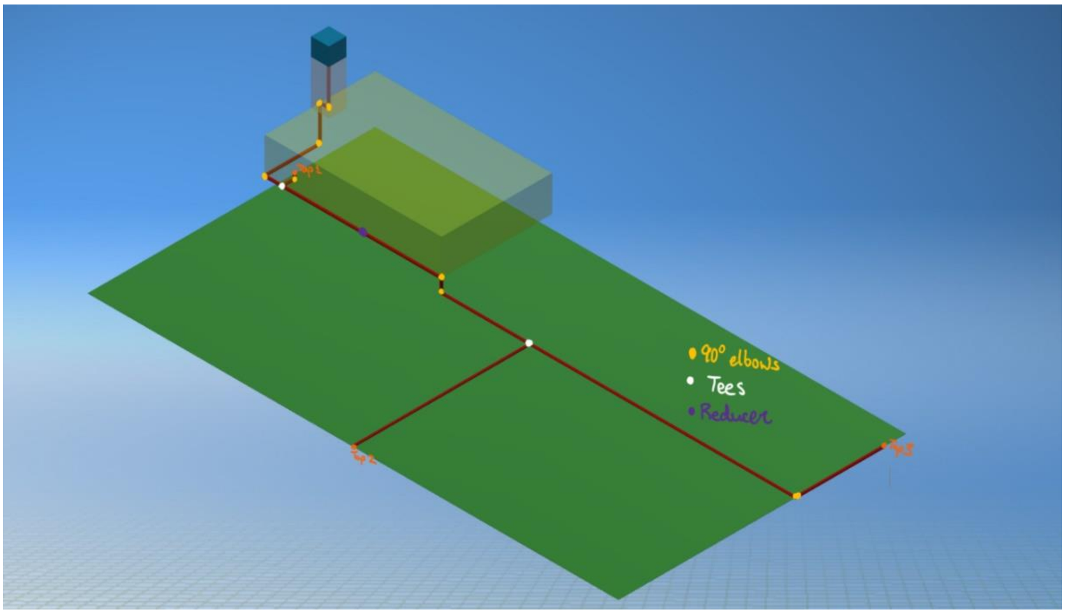

Figure 22: Pipe distribution network, 3D plot.

The system design is based on three priorities:

Inside the clinic:

(1) Avoid sunlight contact. UV in sunlight is harmful to pipes. Therefore, the pipes are run underneath the building instead of across the roof.

(2) Pipes will run along with the clinic's building structure. This is to minimize the physical collisions such as being run over by cars, stepped on, etc.

Outside the clinic:

(3) The pipes are designed to minimize the length, as well as the number of fittings.

(3.5) We have also taken another method into consideration: putting the pipes underground. Further details and comparison will be evaluated in the following section.

Material choices:

Pipes:

According to PVC & PTFE Material Comparison, PTFE has very high resistance to UV exposure and compression while PVC is at a moderate level when it comes to these factors. However, the price of PTFE is increasingly more expensive than that of PVC ("Difference between PVC" 2011). This is why (1) and (2) are prioritized for pipes within the building structure. This will help to reduce the cost of repair, while maintaining the system's healthiness enough for ten years ahead.

Outside the clinic, (3.5) is taken into consideration to make installing PVC all the way long more possible. However, a research by Zhao and Jahani (2002) shows that doing this will highly increase the repair cost as this will involve various costly procedures such as underground digging, labour, … Therefore, we believe choosing (3) and putting PTFE on the surface is the most efficient way to maintain a reasonable setting up cost.

Water tank:

We decided to choose stainless steel as the material for the tank. A comparison blog post by Bradley (2011) has shown that stainless steel is the most efficient material for water tanks, as it can resist corrosion and water contamination. Furthermore, this material is excellent in mobility and durability. The only drawback of stainless steel is that it costs slightly higher than other materials (Bradley 2011). But this is overall considerable, as we believed that choosing this type of material is beneficial in the long run.

Pipe diameter:

PVC (Polyvinyl Chloride) Pipe Dimensions Pressure Ratings

Generally, schedule 40 pipe is white in colour, while schedule 80 is often grey to distinguish it from 40.

Schedule 40 Pipe

Nominal Size DN	Outside Diameter (mm)	Minimum Wall Thickness (mm)	Inside Diameter (mm)*	Max Pres (kPa)
15	21.3	2.77	15.8	4136
20	26.7	2.87	20.9	3309
25	33.4	3.38	26.6	3102
32	42.2	3.56	35.1	2550
40	48.3	3.68	40.9	2275
50	60.3	3.91	52.5	1930
65	73	5.16	62.7	2068
80	88.9	5.49	77.9	1792
100	114	6.02	102	1516
125	141	6.55	128	1310
150	168	7.11	154	1241
200	219	8.18	203	1103
250	273	9.27	255	965
300	324	10.3	303	896
350	356	11.1	333	896
400	406	12.7	381	896

*Inside Diameter = Outside Diameter - 2 x Minimum Wall Thickness

Schedule 80 Pipe

Nominal Size DN	Outside Diameter (mm)	Minimum Wall Thickness (mm)	Inside Diameter (mm)*	Max Pres (kPa)
15	21.3	3.73	13.9	5859
20	26.7	3.91	18.8	4756
25	33.4	4.55	24.3	4343
32	42.2	4.85	32.5	3584
40	48.3	5.08	38.1	3240
50	60.3	5.54	49.3	2757
65	73	7.01	59	2895
80	88.9	7.62	73.7	2550
100	114	8.56	97.2	2206
125	141	9.52	122	1999
150	168	11	146	1930
200	219	12.7	194	1723
250	273	15.1	243	1585
300	324	17.4	289	1585
350	356	19	318	1516
400	406	21.4	364	1516

*Inside Diameter = Outside Diameter - 2 x Minimum Wall Thickness

Figure 23: List of available pipe types. Retrieved from "pump and water distribution.pdf" in April 2019

The final choice is an inside diameter of 0.0209 meters, as it is the smallest diameter in the pipe list that can meet the minimum pressure requirements, and therefore will have the lowest cost. Moreover, to further minimize the cost, a reducer is added to reduce the pipe section to 0.0158 meters of inside diameter. To find the most suitable position for the reducer, measurements are taken starting from the intersection of Tap 2 and Tap 3 in steps of 1 meter towards the entrance. A MATLAB function is written to calculate this (see Appendix B7). As a result, the pipe cost will be at its lowest when the reducer is put in the position as drawn in the distribution network.

4.3.2. Design for Tap 1, EBE analysis

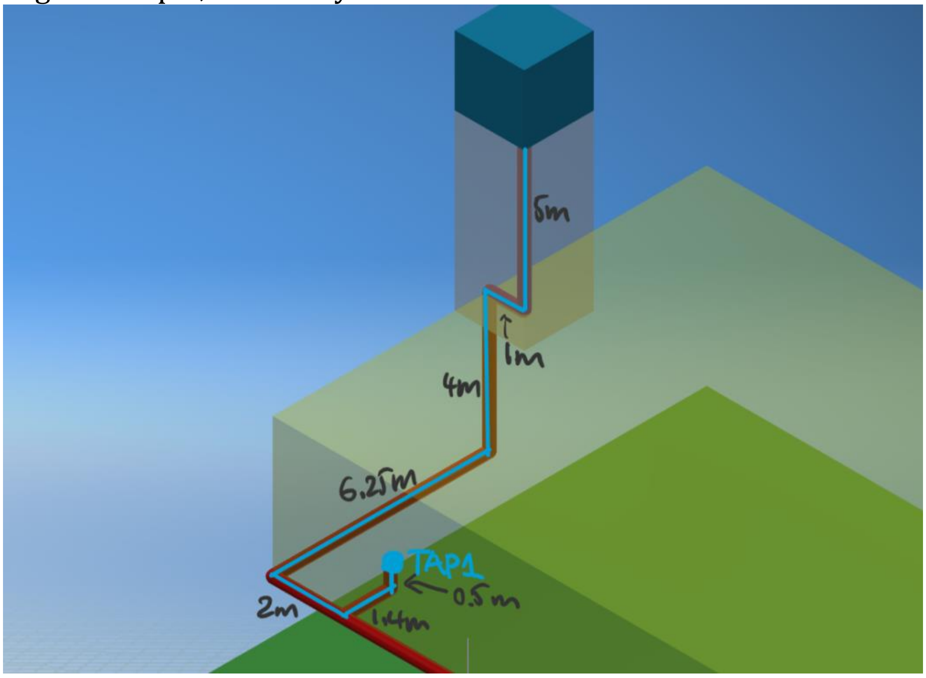

Figure 24: 3D plot of the system showing the path to tap 1.

Using EBE, we have:

$$P_{out} = \rho g(h_1 - h_2) - \rho l_v - \frac{1}{2}\alpha {v_{out}}^2$$

Equation 7: EBE used for tap 1.

$$l_v = \frac{1}{g}(\frac{fLv^2}{2d} + \frac{1}{2}K_{tot}v^2)$$

Equation 8: Loss term calculation for tap 1.

Parameters	Value
Flowrate	9 L/min
$h_1 - h_2$	10.5 m
L	19.15 m
d	0.0209m
K_{tot}	6.5
Result	
$P_{out,gauge}$	99.3kPa
P_{out}	200.3kPa

Table 11: The parameters used in the calculation and the resulting tap pressure for tap 1. The tap pressure is greater than 80kPa, which meets the design requirements.

4.3.3. Design for Tap 2, EBE analysis

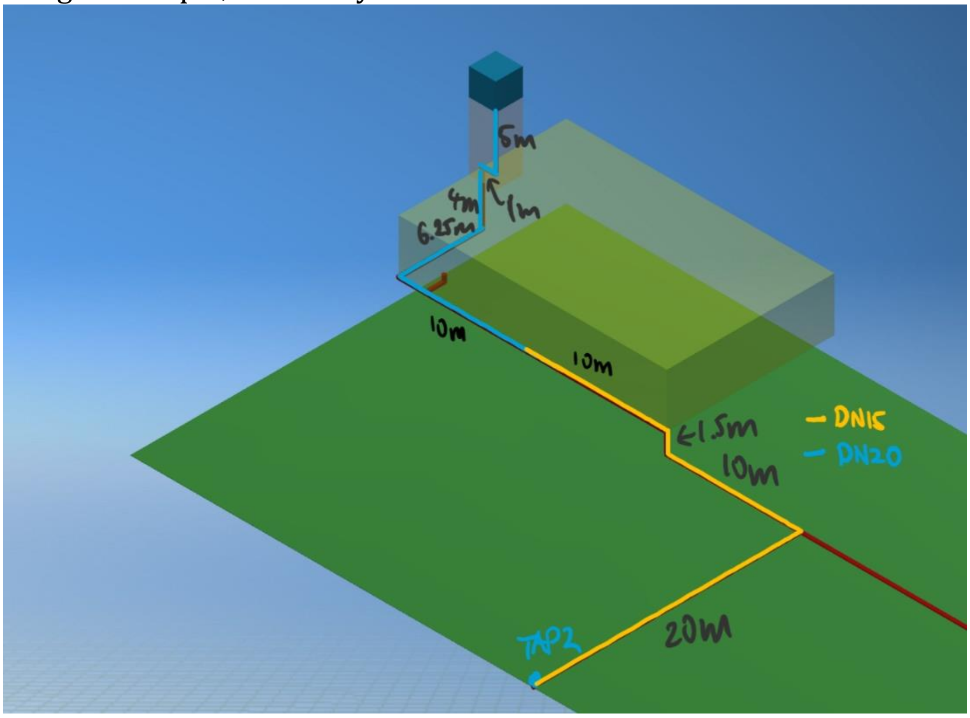

Figure 25: 3D plot of the system showing the path to tap 2.

Using EBE, we have:

$$P_{out} = \rho g(h_1 - h_2) - \rho l_v - \frac{1}{2}\alpha v_{out}^2$$

Equation 9: EBE used for tap 2.

$$l_v = \frac{1}{g}\left(\frac{f_1 L_1 v_1^2}{2d_1} + \frac{f_2 L_2 v_2^2}{2d_2} + \frac{1}{2}K_1 v_1^2 + \frac{1}{2}(K_2 + K_{red})v_2^2\right)$$

Equation 10: Loss term calculation for tap 2.

Parameters	Value
Flowrate	9 L/min
$h_1 - h_2$	12.5 m
L_1	25.75 m
d_1	0.0209m
K_1	4.4
L_2	51.5 m
d_2	0.0158m
K_2	3.25
K_{red}	0.807
Result	
$P_{out,gauge}$	108kPa
P_{out}	209kPa

Table 12: The parameters used in calculation and the resulting tap pressure for tap 2. The tap pressure is higher than 80kPa, which meets the design requirements.

4.3.4. Design for Tap 3, EBE analysis

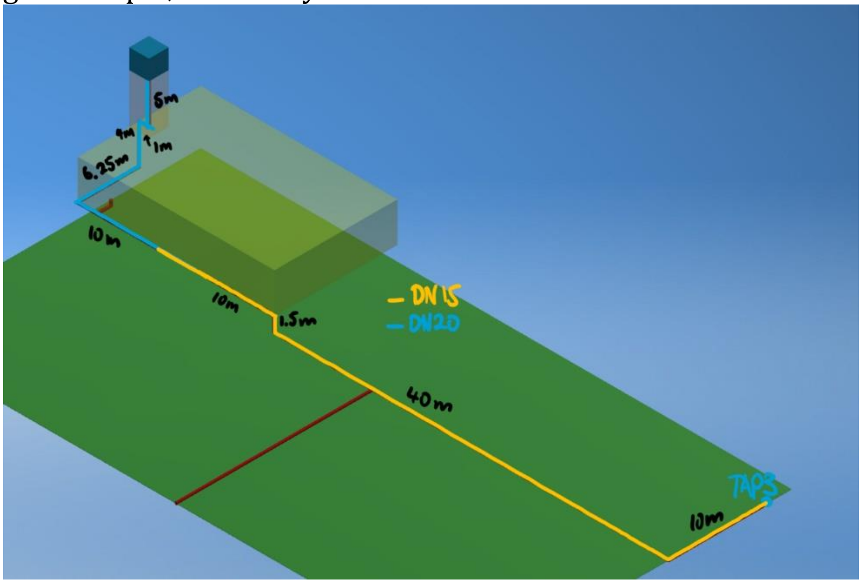

Figure 26: 3D plot of the system showing the path to tap 3.

Using EBE, we have:

$$P_{out} = \rho g(h_1 - h_2) - \rho l_v - \frac{1}{2}\alpha {v_{out}}^2$$

Equation 11: EBE used for tap 3.

$$l_v = \frac{1}{g}\left(\frac{f_1 L_1 {v_1}^2}{2d_1} + \frac{f_2 L_2 {v_2}^2}{2d_2} + \frac{1}{2}K_1 {v_1}^2 + \frac{1}{2}(K_2 + K_{red}){v_2}^2\right)$$

Equation 12: Loss term calculation for tap 3.

Parameters	Value
Flowrate	9 L/min
$h_1 - h_2$	12.5 m
L_1	25.75 m
d_1	0.0209m
K_1	4.4
L_2	61.5 m
d_2	0.0158m
K_2	3.4
K_{red}	0.807
Result	
$P_{out,gauge}$	106kPa
P_{out}	207kPa

Table 13: The parameters used in the calculation and the resulting tap pressure for tap 3. The tap pressure is greater than 80kPa, which meets the design requirements.

4.3.5. Design for Tap 2 & 3, EBE analysis

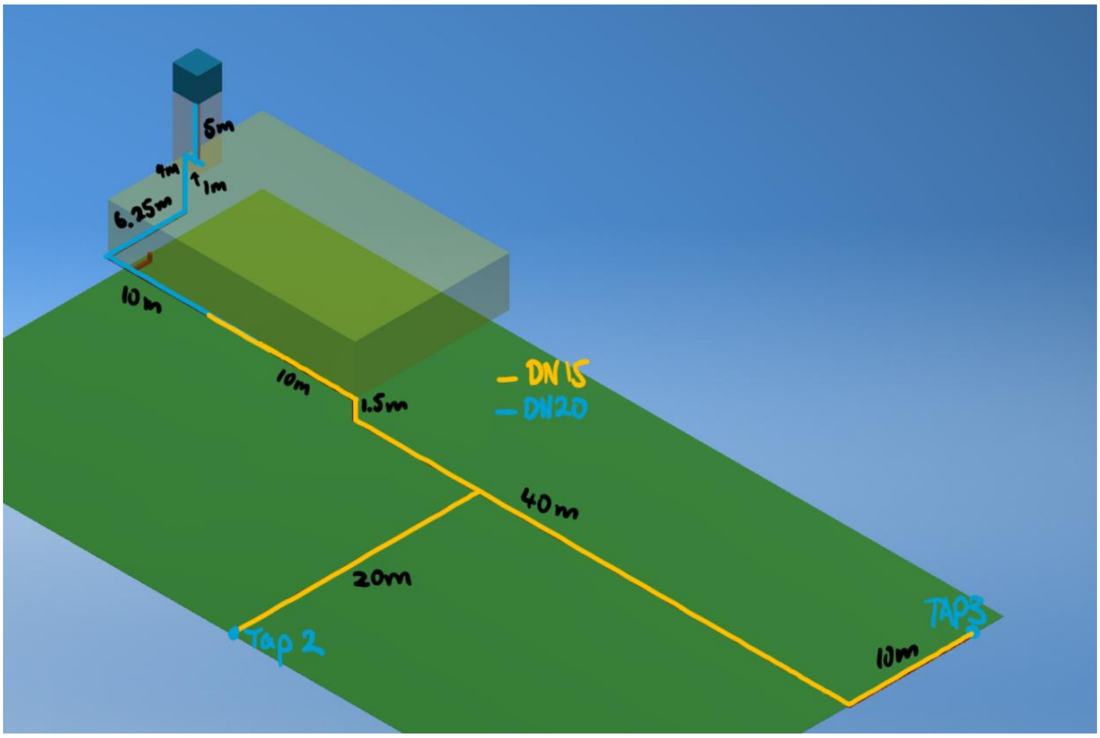

Figure 27: 3D plot showing the path to tap 2 and tap 3.

<u>Theory:</u>

The tee that separates the path to Tap 2 and Tap 3 has dividing flow. According to the thesis of Vasava (2007), for cases of combined or divided flow through tee, the tee can be divided as separate fittings, and the pressure loss, instead of using that of the tee, will be calculated based on the K-value of these fittings. Specifically, a step-by-step formula can be calculated as shown:

$$K_{23} = 0.61\left(\frac{V_2}{V_3}\right)^2 + 1 - 2\left(\frac{V_2}{V_3}\right)\left(\frac{Q_2}{Q_3}\right)\cos\alpha'$$

Equation 13: Calculation of K-value for the separated 90-degree elbow, with α as the bending angle of the tee (Vasava 2007, p.34)

$$K_{13} = 1 - \frac{x}{A}$$

Equation 14: Calculation of K-value for the reducer, with x as the exit area. In case of an expander, it would be A divided by x (Vasava 2007, p.34)

Applying to Tap 2 and 3:

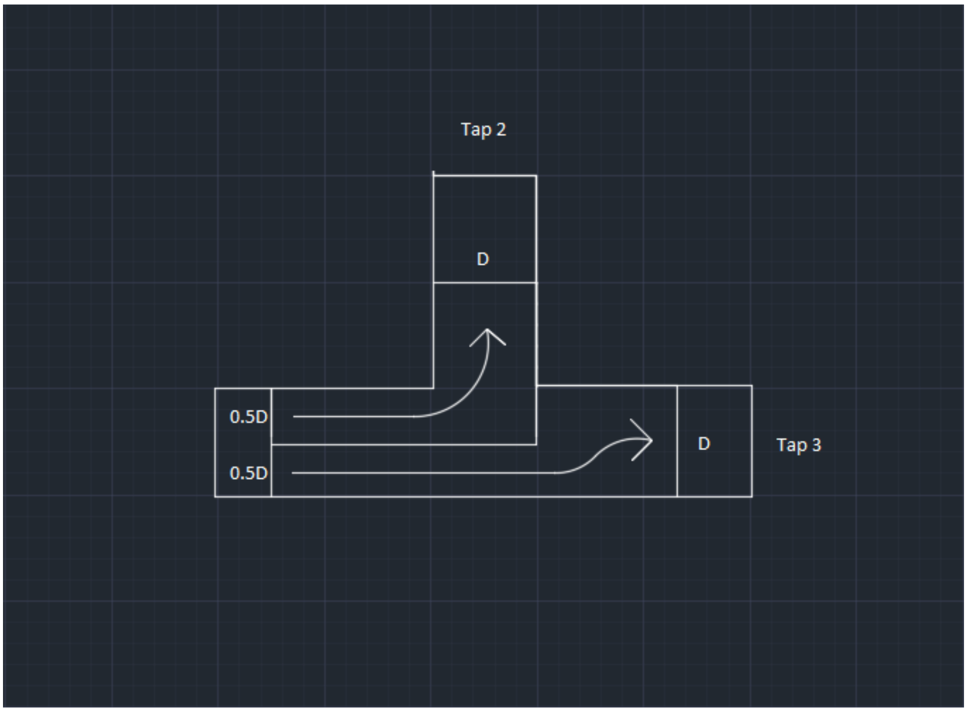

Figure 28: Plot of flow through tee to tap 2 and 3, illustrated using AutoCAD

For the case of the project, there are few specifications:

- The tee is divided into an elbow and an expander.
- The flow rate is constant.
- The entrance diameter of the 2 fittings are equally divided. Therefore, the entrance area of these two are equal, with each equal to a quarter of the exit area (as the area formula: $\frac{\pi D^2}{4}$)

Applying to equation 1:

$$K_2 = 0.61\left(\frac{v_{tap2}}{v_{entrance}}\right)^2 + 1 - 2\left(\frac{V_{tap2}}{V_{entrance}}\right)$$

Equation 15: Here, the flow rate is the same across 2 and 3; thus, the ratio is 1, and the angle is 90 degrees; thus, the cosine factor is also 1.

Substituting into the equations:

$$K_2 = 0.61\left(\frac{1}{4}\right)^2 + 1 - 2\left(\frac{1}{4}\right) = 0.54$$

Equation 16: The numerical value for the K-value of branch 2.

Applying to equation 2:

$$K_3 = 1 - \frac{A_{entrance}}{A} = 1 - 0.25 = 0.75$$

Equation 17: The numerical value for the K-value of branch 3.

Calculation of pressure drop:

$$\Delta P_2 = \frac{1}{2}\rho K_2 V_{entrance}^2$$

$$\Delta P_3 = \frac{1}{2}\rho K_3 V_{entrance}^2$$

Equation 18: Expressions for the pressure drop across the T-junction for the two paths.

These values will be used in the total pressure drop instead of the tee pressure drop. In comparison to the pressure of the separated tap, the new pressure can be calculated with this replacement:

$$P_{2,gauge} = P_{tap2} + P_{Tee} - \Delta P_2$$

$$P_{3,gauge} = P_{tap3} + P_{Tee} - \Delta P_3$$

Equation 19: Calculation of tap pressures when both tap 2 and 3 are open.

Results

Flow rate: 9L/min

Pressure at tap 2: 109 kPa

Pressure at tap 3: 103 kPa

MATLAB calculations: Appendix B8

4.4. Water Disinfection

4.4.1. Contactor Section

4.4.1.1. Pathogen Inactivation Rate Constants

The temperature is assumed to be 25 C. Using this temperature, the inactivation rates are as follows:

$$k_{giardia} = 1.0380 \times (1.0741)^{25} = 6.19891$$

Equation 20: Calculation of giardia inactivation constant.

$$k_{virus} = 2.1744 \times (1.0726)^{25} = 12.5396$$

Equation 21: Calculation of virus inactivation constant.

These are the values inputted into the MATLAB script in Appendix B4.

4.4.1.2. Ozone Decay Constant

$$k_{O3} = 0.0462\ min^{-1}$$

Equation 22: The value of the ozone decay constant.

This value is based on experimental values given in Appendix C2.

4.4.1.3. Contactor Design

Ozone concentration in water	0.13mg/L
Time in contactor	2.75 min

Table 14: Parameters used in Chick-Watson's law calculations.

Pathogen	Log Inactivation Credit
Giardia	3.91
Viruses	7.92

Table 15: Log-inactivation credits achieved as calculated by the Chick-Watson law using the parameters given in table 14 and the values of equation 13, 14 and 15.

The full table from which this data was copied from is in Appendix C7.

The contactor section uses DN100 schedule 40 PTFE piping. The length of the contactor section is 25m.

With a flow rate of 2.229m^3/hr and ozone concentration in the water of 0.13mg/L, we achieved an inactivation credit of 3.91 for giardia and 7.92 for viruses. This satisfied the requirement of 3 for giardia and 4 for viruses (see Appendix B4).

A shorter contactor length would also satisfy the requirements for both pathogens; however, the team picked the longest length because it produced a log inactivation credit for giardia 1 higher than the requirement. This ensures that in case pathogen levels are abnormally high in case of contamination; the water will still be considered to meet the safety requirements.

4.4.1.4. Water travel time from the reducer to the water tower

The piping diameter used is schedule 40, DN80, and the travel time of water was 11.08 minutes. This is above 10 minutes and below 20 minutes as per the requirements.

4.4.2. Membrane Modules

4.4.2.1. Membrane Stream Concentrations and Flow Rates

Qw	2.2283m^3/hr
Qg	5.6m^3/hr
Gas ozone concentration at inlet	0.25% v/v
Gas ozone concentration at outlet	0.24% v/v
System pressure	1 atm
Partial pressure of ozone at inlet	253.3Pa
Partial pressure of ozone at outlet	243.4Pa
Water ozone concentration at outlet	0.13 mg/L
Water ozone concentration at inlet	0 mg/L
Mass flowrate of ozone from gas into water	8.049e-6 kg/s

Table 16: Table reporting the parameters inputted into the membrane simulator which imitates conditions of the membrane design in the real world.

These calculations are performed based on the theory set out in section 2.3.3.

4.4.2.2. Total Required Area

Using the parameters from the previous section, we get:

$$A_{req} = 7.141m^2$$

In this calculation, we assumed that the process simulator's results are verified against experimental data, which would ensure our value is correct.

4.4.2.3. Membrane Module Selection, Type and Number, and Pressure Drop

These are the possible membrane selection solutions:

Viable Combinations	**Total Area (m^2)**	**Presure Drop Over Membranes in Series (kPa)**
7 2x13	7.7	849
3 4x13 + 1 2x13	7.61	267
1 4x28 + 3 2x13	7.445	466
1 2x13 + 1 4x13 + 1 4x28	7.415	262

Table 17: Shows the possible membrane combinations, their total combined surface area, and the pressure drop for the operational flowrate.

The choice made was 1 2x13, 1 4x13 and 1 4x28.

The total area is 7.415m^2, this is 0.274m^2 above the requirement calculated in section 4.4.2.2. This have met the requirement that the actual area is less than 0.6m^2 above the required area.

This was chosen because it had the least pressure drop and only involves 3 membranes as opposed to more in the other combinations. Having only 3 membranes may reduce cost.

The MATLAB script used in the pressure drop calculation can be found in Appendix B5.

4.5. **Inline Image Monitoring System**

4.5.1. Particle concentration threshold

T_part (months)	C_part (particles/ml)	No of filter changes in 5 years	T_filters (hours)	T_foul (hours)
8.0	0.10	7	21	5305.3
8.5	0.23	7	21	2306.7
9.0	0.51	6	18	1040.3
9.5	1.15	6	18	461.3
10.0	2.59	6	18	204.8
10.5	**5.84**	**5**	**15**	**90.8**
11.0	13.16	5	15	40.3
11.5	29.69	5	15	17.9
12.0	66.96	5	15	7.9

Table 18: Table for the selection of a particle concentration threshold for the design. The row highlighted in green is the selected condition.

The calculations performed to generate this table is found in Appendix B13.

The filter lifetime dependent on particle concentration is given in Appendix C8.

Our choice for the threshold value of 5.84 ensures that less than 7 filter changes occur over 5 years. Moreover, the total contaminated run time to foul the membrane is 90.8 hours for this concentration, over 5 years, only 15 hours of contaminated run time occurs, which means the membranes will not be fouled for 30 years. This ensures that membranes will not be fouled at all due to time in filter changing over the 10 years the design lasts for.

4.5.2. Detection algorithm scale up

To adapt the detection algorithm used in the experiment to the design, simply change the parameters which are different in the image processing function.

The torch beam volume changes because the design uses a different viewing window and the particle concentration also changes.

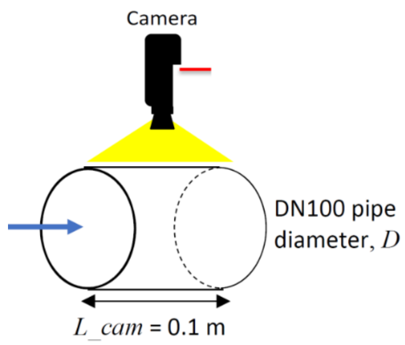

Figure 29: The new dimensions for the volume of a torch beam is calculated using this design information:

To calculate the new volume, use the formula for the volume of a cylinder using the dimensions given above.

Parameter	Old (Used in the experiment)	New (Design)
Torch Beam Volume(cm^3)	8.94	8.17
Particle Concentration Threshold (per mL)	25	5.84

Table 19: The changes in parameters required.

5. Discussion

The design proposed in this report is an example of a water distribution used in large scales, such as those for large towns or cities. The structure and subsystems used in this water system closely resembled that outlined in (Ian L. Peppers 2009), where water is pumped from an initial reservoir to treatment and then stored for distribution in water towers such that continuous pumping is not required.

The experimentation has obtained pipe flow system parameters which are subsequently scaled up through schemes such as the pump affinity law. The research based on water consumption allowed us to construct the daily water requirement through modelling an exponential population growth. However, in the design, the overflow that could occur when the population is small and does not consume all the water stored in the tank daily was not taken into consideration, which could be problematic in real life. This could be remedied through an extra monitoring system which controls the pumping.

The design was focused on a cost-efficiency basis, in order that this system is viable for multiple businesses or organisations. This could be beneficial as the solution can be implemented prolifically, helping more communities. An optimisation is also critical for being climate-friendly, which could carry in future legal implications.

The inline monitoring system has not been implemented in larger scale systems; however, it is critical here as it ensures the safety of the system and decreases the maintenance required.

While the design proposal focuses on efficiency, it should be very effective over the ten years as per the requirements, this is ensured by the sophisticated particle detection algorithm and detailed simulations to ensure the accuracy of the design in meeting safety standards and ensuring the longevity of the subsystems. However, there should be stationed in the community a technician as there is the possibility of damage during its operation by natural disaster or accidents.

6. Conclusions

This report has outlined all the experiments conducted in support for an effective design which met the requirements. The results have been presented along with a discussion of trends and limitations.

The design of the subsystems (pump, distribution network, treatment and inline monitoring) have been presented in detailed drawings and parameters laid out in a tabulated form.

The design satisfies all of the requirements while emphasising the importance of cost-efficiency, which meets the goals of the project.

7. Recommendations

Ways of improving the current system:

- The pipes leading to the consumers should be protected by copper or plastic tubing, so that wear down or breakage due to traffic or the environment can be avoided or reduced.
- Water meters and water measurement system could be added in the water tower, so only the amount of water needed would be pumped from the water source.
- Fittings such as valves and membranes should be protected with boxes that are easily accessible and fixable.
- Protective measures should be added in section 2 water distribution, as contamination during the distribution process is possible. Possible contamination cause includes water leaks.

Future system:

After the ten-year operation period of this system, a scaled-up system is going to be required in order to meet new requirements. To meet these requirements:

- A larger pump is going to have to be installed and the pipe sizes increased.
- The treatment must also be scaled-up to meet the disinfection requirement.
- More massive and possibly higher placed water towers might have to be installed to increase the number of taps supplied to.

The system framework can be adapted to provide general-use water; however, the volume of water to be supplied will increase markedly and significant scale-up is required. This may be an option for the community if there is a need to do so.

8. References

Aikhuele, D. (2015). Ball Valves, Issues and Material Selections for Manufacturing. *Academia.*https://www.academia.edu/8833301/Ball_Valves_Issues_and_Material_Selections_for_Manufacturing

Bradley, K. (2011, February 14), Water Tank comparisons for drinking water: defining clean and green [Blog post]. Retrieved from https://www.milkwood.net/2011/02/14/water-tank-comparisons-for-drinking-water-defining-clean-and-green/

Difference Between PVC and PTFE Cables (2011, November). Retrieved from https://www.wesbellwireandcable.com/blog/difference-between-pvc-and-ptfe-cables/

Gunnars, K. (2018). How Much Water Should You Drink Per Day? Retrieved from https://www.healthline.com/nutrition/how-much-water-should-you-drink-per-day

Ian L. Pepper, ... Charles P. Gerba, in Environmental Microbiology (Second Edition), 2009, https://www.sciencedirect.com/topics/earth-and-planetary-sciences/water-distribution-system

Montazerin, N. (2015). Slip Factor. *Developments in Turbomachinery Flow.* Retrieved from: https://www.sciencedirect.com/topics/engineering/slip-factor

PetroWiki. Centrifugal Pumps. Retrieved from: https://petrowiki.org/Centrifugal_pumps#Open_impellers

PVC & PTFE Material Comparison. Retrieved from http://ptfe-machinery.com/pvc-ptfe-material-comparison/

Vasava, P. (2007). *Fluid Flow in T-Junction of Pipes*. (Published Master's thesis). Lappeenranta University of Technology, Lappeenranta, Finland.

Zhao, Q., Rajani, B. (2002). Construction and Rehabilitation Costs for Buried Pipe with a Focus on Trenchless Technologies, *Research Report (National Research Council Canada. Institute for Research in Construction).*

9. Appendices

Appendix A: Minutes of Team Meetings

Date	Discussion	Meeting duration
March 16	What each member knows about the project and assignment of roles	1 hour
March 23	Task allocation and deadlines discussion	1 hour 5 min
April 2	Water requirements	30 min
April 9	Distribution/treatment checklist	1 hour
April 17	Membrane selection	50 min
April 22	Water distribution and tap pressures (Matlab) and presentation part allocation	2 hours
April 30	Presentation development	1 hour
May 8	Presentation preparation	2 hours
May 27	Draft report feedback discussion.	2 hours
June 3	Finalising the inline monitoring, allocation of editors for the final report	4 hours

Table 20: Table of meeting times and tasks completed in the meeting.

Appendix B, MATLAB Script Appendix

B1: Ball-valve Calculations

```
%constants
d = 21*10^-3;
A = pi*(d/2)^2;

%Independent
degrees_closed = [0 5 10 15 20 25 30 35 40 45 50];
F_opening = (1-degrees_closed/90)*100;

%Measured

%mbar
DP_mbar = [35 35 65 75 100 140 165 190 205 215 220];

%L/min
Q_Lmin = [15.46 15.46 13.53 12.97 10.99 8.72 6.73 3.99 2.46 1.23 0.31];

%Conversion

%Pa
DP = DP_mbar*100;

%m^3/s
Q = Q_Lmin*(1/1000)*(1/60);

%velocity calculation
v = Q/A;

%K-value

K = (2*DP)./(998*v.^2)
```

```
%plotting K-values

yyaxis left

semilogy(F_opening,K,'b-')

title("K-values for a Ball Valve")

xlabel("Fraction opening (%)")
ylabel("K")

%Cv calculation

DPbar = DP_mbar/1000;
Cv = Q_Lmin./(sqrt(DPbar)*14.42)

yyaxis right

plot(F_opening,Cv,'r-')

ylabel("Cv")

legend("K","Cv")
```

B2: Generation of pump curves comparing stock and custom impeller.

```
% clear workspace variables, clear command window and close all
 figures at the start
clear all
clc
close all

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  Plot pump and efficiency curve for Workshop 4 custom impeller %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% measured values from Workshop 4 experiment using custom impellers
%-----------------------------------------------------------------
Q_L_per_min_W4 = [16.51 15.93 14.18 12.10 9.95 6.89 3.89 0.6]      %
 enter values for flowrate from Workshop 4 in L/min
DP_mbar_W4 = [147 149 153 170 181 201 209 230];         % enter values
 for pressur drop from Workshop 4 in mbar

% calculate pump head and efficiency for Workshop 4 custom impeller
% ----------------------------------------------------------------
% specify contants
rho = 998;    % in kg/m3
D = 2.1/100;       % in m

A = pi*(D/2)^2;                    % calculate cross section area of
 pipe (in m2)
Q_m3_per_s_W4 = Q_L_per_min_W4*0.001/60;      % convert Q_L_per_min_W4
 from L/min to m3/s
DP_Pa_W4 = DP_mbar_W4*100;             % convert DP_mbar_W4 from mbar
 to Pa
h_W4 = DP_Pa_W4/(rho*9.8);               % calculate pump head for
 custom impeller in m

EP_W4 =
 8.075e-3.*(Q_m3_per_s_W4).^2+5.616e-2.*(Q_m3_per_s_W4).^2+17.35;
        % calculate electrical power from expirical formula in W
efficiency_W4 = (Q_m3_per_s_W4.*DP_Pa_W4)./EP_W4;       % calculate
 efficiency for custom impeller in %

figure   % open a new figure for plotting
% plot pump curve, [h(m) vs Q(L/min)] of custom impeller on the left
 axis
% using the 'yyaxis'function. Label ths left y-axis
yyaxis left;
plot(Q_m3_per_s_W4,h_W4);
xlabel('Q(m^3/s)');
ylabel('h(m)')

% plot efficiency curve, [n(%) vs Q(L/min)] of custom impeller on the
 right
% axis using the 'yyaxis' function. Label the right y-axis
hold on;
yyaxis right;
```

```
plot(Q_m3_per_s_W4,efficiency_W4*100);
xlabel('Q(m^3/s)');
ylabel('efficiency(%)');
legend('Pump Curve','Efficiency Curve')
title('Pump and Efficiency Curves for Custom Impeller')


% plot formatting (title, a-axss label, legend and grid)



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%
%  Plot pump curves from Workshop 3 (supplied impeller) and Workshop 4
 (custom impeller) %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%

% enter your experimental values from Workshop 3 (supplied impeller)
Q_L_per_min_W3 = [15.93 15.64 13.02 8.42 5.08 3.29 2.39 0.90];      %
 enter values for flowrate from Workshop 3 in L/min
DP_mbar_W3 = [128 129 141 167 185 198 201 205];              % enter
 values for pressur drop from Workshop 3 in mbar

% calculate pump head for Workshop 3 supplied impeller
Q_m3_per_s_W3 = Q_L_per_min_W3*0.001/60;       % convert
 Q_L_per_min_W3 from L/min to m3/s
DP_Pa_W3 = DP_mbar_W3*100;                  % convert DP_mbar_W3 from
 mbar to Pa
h_W3 = DP_Pa_W3/(rho*9.8);                 % calculate pump head for
 supplied impeller in m

EP_W3 =
 8.075e-3.*(Q_m3_per_s_W3).^2+5.616e-2.*(Q_m3_per_s_W3).^2+17.35;
         % calculate electrical power from expirical formula in W
efficiency_W3 = (Q_m3_per_s_W3.*DP_Pa_W3)./EP_W3;

figure  % open new figure for plotting
% Superimpose the pump curves [h(m) vs Q(L/min)] from both Workshop 3
 and Workshop 4
% onto the same graph. Hint: Use the hold ON and hold OFF functions
hold on
yyaxis left;
plot(Q_m3_per_s_W3,h_W3);
plot(Q_m3_per_s_W4,h_W4);
xlabel('Q(m^3/s)');
ylabel('h(m)')

% plot formatting (title, axes labels, legend and grid)

yyaxis right;
plot(Q_m3_per_s_W3,efficiency_W3*100);
plot(Q_m3_per_s_W4,efficiency_W4*100);
xlabel('Q(m^3/s)');
```

```
ylabel('efficiency(%)');
legend('Pump Curve Stock', 'Pump Curve Custom','Efficiency Curve
 Stock','Efficiency Curve Custom')
title('Comparison of Custom and Stock Impellers')

figure

yyaxis left;
plot(Q_m3_per_s_W3,h_W3)
xlabel('Q(m^3/s)');
ylabel('h(m)')

yyaxis right;
plot(Q_m3_per_s_W3,efficiency_W3*100)
ylabel('efficiency(%)')
legend('Pump Curve','Efficiency Curve')
title('Pump and Efficiency Curves for Stock Impeller')
```

B3: Generating plots of membrane characteristics.

```
% clear workspace variables, clear command window and close all
 figures at the start
clear all
clc
close all

% GAD-Cell ????
% Define a Q vector in m3/hr from 0.25 to 2.5 with at most 10 points.
Qo=linspace(0.25,2.5,10);

%At conditions of Ozone (O3) v/v% of ???? at the gas side inlet and
 C_in = 0.
%Set the outlet water
%concentration to 1.0 mg/L and find the area at the flowrates above:
A_Cout1_0 =[1.597 3.208 4.833 6.473 8.128 9.798 11.48 13.18 14.9
 16.41] ;

%At conditions of Ozone (O3) v/v% of ???? at the gas side inlet and
 C_in = 0.
%Set the outlet water
%concentration to 0.75 mg/L and find the area at the flowrates above:
A_Cout0_75 = [1.164 2.335 3.515 4.702 5.897 7.1 8.311 9.53 10.76
 11.99] ;

%At conditions of Ozone (O3) v/v% of ???? at the gas side inlet and
 C_in = 0.
%Set the outlet water
%concentration to 0.5 mg/L and find the area at the flowrates above:
A_Cout0_5 = [0.755 1.513 2.275 3.04 3.808 4.579 5.354 6.133 6.915
 7.7];

%At conditions of Ozone (O3) v/v% of ???? at the gas side inlet and
 C_in = 0.
%Set the outlet water
%concentration to 0.2 mg/L and find the area at the flowrates above:
A_Cout0_2 = [0.2927 0.5859 0.8796 1.174 1.468 1.764 2.059 2.356 2.652
 2.95] ;

%At conditions of Ozone (O3) v/v% of ???? at the gas side inlet and
 C_in = 0.
%Set the outlet water
%concentration to 0.1 mg/L and find the area at the flowrates above:
A_Cout0_1 = [0.1449 0.2899 0.435 0.5802 0.7256 0.8711 1.017 1.162
 1.308 1.454];


% Define a cout vector in mg/L f at 0.1, 0.2, 0.5 0.75 and 1, ie. the
% concentrations used above
c_out_list=[0.1 0.2 0.5 0.75 1];

%Now form a new set of vectors using the existing date.
```

```
%At conditions of Ozone (O3) v/v% of ???? at the flow rate of 0.5 m3/
hr.
%Find area area at the range of outlet concentrations above
% This does not require more simulations runs, simple asscess the area
 as a
% fucntion of flow rate vectors above at the the appropiate flow rates

AREAQ0_25 = [A_Cout0_1(1) A_Cout0_2(1) A_Cout0_5(1) A_Cout0_75(1)
 A_Cout1_0(1)];
AREAQ0_5 = [A_Cout0_1(2) A_Cout0_2(2) A_Cout0_5(2) A_Cout0_75(2)
 A_Cout1_0(2)];
AREAQ0_75 = [A_Cout0_1(3) A_Cout0_2(3) A_Cout0_5(3) A_Cout0_75(3)
 A_Cout1_0(3)];
AREAQ1_0 = [A_Cout0_1(4) A_Cout0_2(4) A_Cout0_5(4) A_Cout0_75(4)
 A_Cout1_0(4)];
AREAQ1_25 = [A_Cout0_1(5) A_Cout0_2(5) A_Cout0_5(5) A_Cout0_75(5)
 A_Cout1_0(5)];
AREAQ1_5 = [A_Cout0_1(6) A_Cout0_2(6) A_Cout0_5(6) A_Cout0_75(6)
 A_Cout1_0(6)];
AREAQ1_75 = [A_Cout0_1(7) A_Cout0_2(7) A_Cout0_5(7) A_Cout0_75(7)
 A_Cout1_0(7)];
AREAQ2_0 = [A_Cout0_1(8) A_Cout0_2(8) A_Cout0_5(8) A_Cout0_75(8)
 A_Cout1_0(8)];
AREAQ2_25 = [A_Cout0_1(9) A_Cout0_2(9) A_Cout0_5(9) A_Cout0_75(9)
 A_Cout1_0(9)];
AREAQ2_5 = [A_Cout0_1(10) A_Cout0_2(10) A_Cout0_5(10) A_Cout0_75(10)
 A_Cout1_0(10)];



% Create two plots
% One of a series of curves of Area as a function of flowrate at a set
% concentrations.
% The second as a series of curves of Area as a function of
 concentration
% at a set flow rate.

% Area as a function of flowrate at a set
% concentrations.
figure
grid
plot(Qo,A_Cout0_1)
hold on
plot(Qo,A_Cout0_2)

plot(Qo,A_Cout0_5)

plot(Qo,A_Cout0_75)

plot(Qo,A_Cout1_0)
```

```
title("[Gad-Cell nxn] Membrane Area vs Flow Rate at a Outlet
 Concentration","FontSize",14)
xlabel("Flow Rate (m^3/hr)","FontSize",12)
ylabel('Area required m^2')
legend("C_o = 0.1","C_o = 0.2","C_o = 0.5","C_o = 0.75","C_o =
 1.0","location","northwest")
hold off

%Area as a function of concentration
% at a set flow rate.
figure
grid
plot(c_out_list,AREAQ0_25)
hold on
plot(c_out_list,AREAQ0_5)

plot(c_out_list,AREAQ0_75)

plot(c_out_list,AREAQ1_0)

plot(c_out_list,AREAQ1_25)

plot(c_out_list,AREAQ1_5)

plot(c_out_list,AREAQ1_75)

plot(c_out_list,AREAQ2_0)

plot(c_out_list,AREAQ2_25)

plot(c_out_list,AREAQ2_5)

title("[Gad-Cell nxn] Membrane Area vs Flow Rate at a Outlet
 Concentration","FontSize",14)
xlabel("Concentration (kg/m^3)","FontSize",12)
ylabel('Area required m^2')
legend("Qo = 0.25","Qo = 0.5","Qo = 0.75","Qo = 1.0","Qo =
 1.25","Qo = 1.5","Qo = 1.75","Qo = 2.0","Qo = 2.25","Qo =
 2.5","location","northwest")
hold off




% Create a matrix of area as a function of both outlet concentration
 and
% flow rate.  Make a matrix out of the vectors of flow rate versus
 area for
% the five different outlet concetrations
AreaMat= [A_Cout0_1;A_Cout0_2;A_Cout0_5;A_Cout0_75;A_Cout1_0];

%Make a 3D plot of these data a surface using surfc.  surfc also,
%automatically plots a contour plot on the bottom x-y plane.
```

```
% To use surfc and give the correct limits of the surf plot, one need
 to
% tell surc the points and range of the concentrations and flow rate
 that
% correspond to the area data.
%So the surfc command requires two vectors Qo and cout, before the
 matrix
%data
figure;

surfc(Qo,c_out_list,AreaMat)
% This makes the shading an interpolation, it looks better
shading interp
hold on

% labels
title("Area Required (m^2) vs Flowrate (m^3/hr) vs Outlet
 Concentration (%v/v)")
xlabel("Qo(m^3/hr)")
ylabel("cout(%v/v)")
zlabel("Area Required (m^2)")

% This overlays a mesh of the same data, not required but looks nice
mesh(Qo,c_out_list,AreaMat,"FaceAlpha",0,"EdgeColor",
[0,0,0],'LineWidth',0.5)
hold off
```

B4: Contactor section log-inactivation credit calculations.

Table of Contents

This is an m file to design a contact section of water disinfection pipe network

```
%Mass of water required in one day, from Pump and Pipe Network seciton
 1
% Determined previously.
m3day=   6.685 ;
```

Treatment Flow Rates*

Set process treatment time between 7-9 hours

```
%treatment time (hr)
TRtimeHr= [3 4 5] ;
%treatment Time (sec)
TRtimeSec=TRtimeHr./3600;
%flowrate (m3/hr)
Qm3hr= m3day./TRtimeHr;
%flowrate (m3/s)
Q= Qm3hr./3600;  %Initial flow rate (m^3/s)

%constants, assuming water at 25 Deg C

rho = 998;           %fluid density (kg/m3)
mu = 0.89e-3;        %fluid viscosity (Pa*s)
g = 9.81             % gravitational constant


g =

    9.8100
```

Contactor section

Contactor section, L from membrane to reducer

```
% Use DN150, required for this section
ID = 0.102;
A = pi*(ID/2)^2;
```

```
% Set contactor length from 15 to 25 m

%calculate velicty in pipe, v
v = Q./A;  % m/s

% caclulate contact time, convert to seconds to in minutes
con_time_L15= 15./v./60;
con_time_L20= 20./v./60;
con_time_L25= 25./v./60;
```

Calculate the login activation credits

```
% Calc kp for giardia and viruses, set temp = 25C
k_p_giardia =  1.0380*(1.0741)^25 ;     %k_p for Giardia as a function
 of T(C); s.L/mg
k_p_virus   =   2.1744*(1.0726)^25 ;    %k_p for viruses as a function
 of T(C); s.L/mg
% Calc. ozone decay contstant , half life = 15 min.
k_O3 =   log(2)/15;                %k_O3 at 25C, min^-1
% Concentration of ozone in water, design specification
Conc_ozone =  0.13;           %Onzone Conc in mg/L

% Calc log inactivation credits, use contact time, concentration and
 rate
% costants.
cgiardia_L15 = (k_p_giardia*Conc_ozone*(1-exp(-k_O3*con_time_L15)))/
(k_O3);
cgiardia_L20 = (k_p_giardia*Conc_ozone*(1-exp(-k_O3*con_time_L20)))/
(k_O3);
cgiardia_L25 = (k_p_giardia*Conc_ozone*(1-exp(-k_O3*con_time_L25)))/
(k_O3);

cvirus_L15= (k_p_virus*Conc_ozone*(1-exp(-k_O3*con_time_L15)))/(k_O3);
cvirus_L20= (k_p_virus*Conc_ozone*(1-exp(-k_O3*con_time_L20)))/(k_O3);
cvirus_L25= (k_p_virus*Conc_ozone*(1-exp(-k_O3*con_time_L25)))/(k_O3);

GMatrix = [cgiardia_L15;cgiardia_L20;cgiardia_L25];
VMatrix = [cvirus_L15;cvirus_L20;cvirus_L25];
```

Display results

```
clc
disp('Matrix display of log inactivation credits, vertical is section
 lengths 15, 20 and 25 meters, horizontal is the running time 3, 4 and
 5 hours.')
disp('Giardia')
disp(GMatrix)
disp('Viruses')
disp(VMatrix)
```

```
Matrix display of log inactivation credits, vertical is section
 lengths 15, 20 and 25 meters, horizontal is the running time 3, 4 and
 5 hours.
```

B5: Membrane pressure drop calculations.

```
%constants

Qw = 6.685/3600;
mu = 0.00089;
rho = 998;
epsilon = 0.0025e-3;

%membrane combination
n2x13 = 0
n4x13 = 0
n4x28 = 1

%membrane lengths
L2x13 = 0.333;
L4x13 = 0.413;
L4x28 = 0.789;

%Flowrates through each fibre in different membranes
Qfib2x13 = Qw/1502
Qfib4x13 = Qw/2389
Qfib4x28 = Qw/2389

%Fibre diameters
dfib = 700*(1e-6);

Afib = pi*(dfib/2)^2

v2x13 = Qfib2x13/Afib;
v4x13 = Qfib4x13/Afib;
v4x28 = Qfib4x28/Afib;

%Reynolds number and friction factor for each membrane
Re2x13 = (rho*v2x13*dfib)/mu;
Re4x13 = (rho*v4x13*dfib)/mu;
Re4x28 = (rho*v4x28*dfib)/mu;

if Re2x13 < 2000
    f2x13 = 64/Re2x13
else
   f2x13 = (1/(-1.8*log10(6.9/Re2x13+((epsilon/dfib)/3.7)^1.11)))^2
end

if Re4x13 < 2000
    f4x13 = 64/Re4x13
else
   f4x13 = (1/(-1.8*log10(6.9/Re4x13+((epsilon/dfib)/3.7)^1.11)))^2
end

if Re4x28 < 2000
    f4x28 = 64/Re4x28
else
   f4x28 = (1/(-1.8*log10(6.9/Re4x28+((epsilon/dfib)/3.7)^1.11)))^2


   end

   DP = (rho/2)*(f2x13*L2x13*n2x13*v2x13^2 + f4x13*L4x13*n4x13*v4x13^2 +
    f4x28*L4x28*n4x28*v4x28^2)/dfib
```

Treatment Flow Rates*

```
%treatment time (hr)
%TRtimeHr=
%treatment Time (sec)
%TRtimeSec=

%flowrate (m3/hr)
Qm3hr= 1.9317;
%flowrate (m3/s)
Qm3s= Qm3hr/3600;

Qo = Qm3s;    %Initial flow rate (m^3/s)

%constants, assuming water at 25 Deg C

rho = 998;          %fluid density (kg/m3)
mu = 0.89e-3;       %fluid viscosity (Pa*s)
g = 9.81            % gravitational constant


g =

    9.8100
```

DN100 section (tank outlet to reducer, minus membrane)

```
%This accounts for frictional losses (not heigh changes)
%for the section of pipes and valves from the water storage tank to
 the
%reducer
%NOTE: The membrane is calculated in a separate secion.

%Set flow rate from section above
Q = Qo ;   % m3/s

% Use DN100, recomended for pipe upto Membrane, required for section
% just after membrane, before reducer.
% Use DN100,
% Choose pipe schedule
% Choose pipe material, epsilon = ?? mm
% DN100, get OD and wall thickness from pipe schedule
% OD - 2*wall thickness = ID
```

```
ID =0.102 ;
A = pi*ID^2/4;
AR = 0.0015e-3 %absolute roughness (m), i.e. epsilon in m

%v, Re and f

v = Q/A;  % v for v large, this v is not either point in the EBE

Re = rho*v*ID/mu;


if Re < 2000
    f = 64/Re;
else
 f = (1/(-1.8*log10(6.9/Re+((AR/ID)/3.7)^1.11)))^2; % Haaland
 correlation turbulent
end

%fitting losses
%K-value for reducer calculation
% Need D2
% As a start, use the same material as above
% Use either DN 65 or DN80 for now. Change later if you want to
% Remember to get OD and wall thickness to calc D2
D2 = 0.0779

% Calc K value for reducer
if Re < 2500
    Kred = (1.2+160/Re)*((ID/D2)^4-1)
else
    Kred = (0.6+0.48*f)*((ID/D2)^2-1)*(ID/D2)^2;
end

% Add up fitting losses
% ent???
% el????
% Ball Valve, open  how many,    note: use value from workshop!!
% Kred
Ktot = Kred+(0.75+0.75)+12.67*5
h_f_100 = Ktot*v^2/(2*9.81);

%head loss due to pipe flow
% ???(vertical pipes)
% ???
% ???
% L from membrane to reducer, set to Lcontact = 20 for now.
Lcontact = 25
Ltot =  31.5 +  Lcontact;
h_pipe_100 = f*Ltot/ID*v^2/(2*9.81);


AR =

   1.5000e-06
```

```
D2 =

    0.0779


Ktot =

   65.6046


Lcontact =

    25
```

DN80 or DN65 mm section (reducer outlet to water tower)

after water treatment you can refine and change D2 as needed

```
A2 = pi*D2^2/4;
v2 = Q/A2;  % this is the v in the EBE at exit, v2

Re2 = rho*v2*D2/mu;   %Calc Re2 using A2 and v2 and D2


if Re2 < 2000
    f80 = 64/Re2;
else
    f80 = (1/(-1.8*log10(6.9/Re2+((AR/D2)/3.7)^1.11)))^2; % Haaland
 correlation turbulent
end


% Add up fitting losses
% ex
% elb
Ktot = 2.5
h_f_80 = Ktot*v2^2/(2*9.81);

%head loss due to pipe flow
% portion of 100 m not set by Lcontact  100 - Lcontact
% vertical ?????
% horizontal ??????
Ltot80= (100 - 25)+ 21.5 + 0.05;
h_pipe_80 =  f80*Ltot80/D2*v2^2/(2*9.81)     % similar to h_pipe_100,
 but different variables


Ktot =
```

```
    2.5000


h_pipe_80 =

    0.0249
```

velocity heads

```
v1 = 0   %assume velocity at surface of tank ????
v2 =  v2  %velocity at end of pipe ??????
h_vel = (v2^2-v1^2)/(2*9.81); %alpha = 1, turbulent flow


v1 =

     0


v2 =

    0.1126
```

elevation head

assume lowest point, h = 0 is the pipe into the pump h1 = water storage tank height h2 = Exit into water tower

```
h_z = 31.5-12;
```

Membrane Pressure Drop

Calcualtes the pressure drop using the flow rate Q for a flow through a membrane or sereis of membranes based on the solution to the treatment requirements.

```
%Until the water treatment is finalised
%use the following membrane properties

epsilon = 0.0025e-3;

%membrane combination
n2x13 = 1
n4x13 = 1
n4x28 = 1

%membrane lengths
L2x13 = 0.333;
L4x13 = 0.413;
L4x28 = 0.789;
```

```
%Flowrates through each fibre in different membranes
Qfib2x13 = Q/1502
Qfib4x13 = Q/2389
Qfib4x28 = Q/2389

%Fibre diameters
dfib = 700*(1e-6);

Afib = pi*(dfib/2)^2

v2x13 = Qfib2x13/Afib;
v4x13 = Qfib4x13/Afib;
v4x28 = Qfib4x28/Afib;

%Reynolds number and friction factor for each membrane
Re2x13 = (rho*v2x13*dfib)/mu;
Re4x13 = (rho*v4x13*dfib)/mu;
Re4x28 = (rho*v4x28*dfib)/mu;

if Re2x13 < 2000
    f2x13 = 64/Re2x13
else
   f2x13 = (1/(-1.8*log10(6.9/Re2x13+((epsilon/dfib)/3.7)^1.11)))^2
end

if Re4x13 < 2000
    f4x13 = 64/Re4x13
else
   f4x13 = (1/(-1.8*log10(6.9/Re4x13+((epsilon/dfib)/3.7)^1.11)))^2
end

if Re4x28 < 2000
    f4x28 = 64/Re4x28
else
   f4x28 = (1/(-1.8*log10(6.9/Re4x28+((epsilon/dfib)/3.7)^1.11)))^2
end



% frictional loss
h_mem=(1/(2*9.8))*(f2x13*L2x13*n2x13*v2x13^2 +
 f4x13*L4x13*n4x13*v4x13^2 + f4x28*L4x28*n4x28*v4x28^2)/dfib


n2x13 =

     1


n4x13 =

     1
```

```
n4x28 =

     1


Qfib2x13 =

   3.5725e-07


Qfib4x13 =

   2.2461e-07


Qfib4x28 =

   2.2461e-07


Afib =

   3.8485e-07


f2x13 =

    0.0878


f4x13 =

    0.1397


f4x28 =

    0.1397


h_mem =

    6.0060
```

determine h (note, -h_z because change in elevation is going down, h1 = 0)

```
h_fittings = h_f_80 + h_f_100;
h_pipe = h_pipe_80 + h_f_100;
h = h_z + h_vel + h_fittings + h_pipe+h_mem; %head at pipe outlet
```

```
DP=h*rho*9.81
```

```
DP =

   2.5026e+05
```

Calc Pump Requirements

Calc shaft work and assume an efficeincy and calc. required power Once impeller design finalised with a set value, use eff ~ 20 %. You should use the maxium eff you measured on your choosen impeller once you finalise your design. It's probably of the order of 20%-ish

```
%Ws = ;
%Pumpeff= 0.20;
%Power =
```

Display results

```
clc
disp('For a flow rate of: (m3/s),(m3/hr), (L/s)')
disp([Q, Qm3hr, Q*1000])
disp('')
disp('Total system head: (m)')
disp(h)
disp('')
disp('Difference in Pressure: (Pa)')
disp(DP)
%disp('')
%disp('Shaft work required: (W)')
%disp(Ws)
%disp('')
%disp('Power Required: (W)')
%disp(Power)
%disp('')
```

```
For a flow rate of: (m3/s),(m3/hr), (L/s)
    0.0005    1.9317    0.5366

Total system head: (m)
   25.5619

Difference in Pressure: (Pa)
   2.5026e+05
```

Published with MATLAB® R2018b

B7: Function for calculating tap pressures in section 2.

```
%How to use:
%Step1: Run. Don't care about the comments
%Step2: Type p_tap([Q,dh], [ID values], [L values], [K values incl.
 entrance & exit]) (make sure they are in SI units)
%note: take the values of each pipe section. Section head points are
 entrance, exit and reducers
function ptap1=p_tap(x,D,L,K_val)
e=0.0015e-3;  %epsilon
r=998;   %rho
g=9.81;
u=8.9e-4; %mu

Q=x(1); %reading values
dh=x(2);

K=[]; %initial value settings
Kval=[];
p_reducer=0;
ptap=0;

if length(D)==1 %if no reducers
    A=0.25.*D(1).^2.*pi;
    v=Q./A;
    Re=r*v*D(1)/u;

    if Re<2000
    f=64./Re;
    else
        f=1./(-1.8.*log10(6.9./Re+(e./3.7./D(1)).^1.11)).^2;
    end

    %alpha
        if Re < 2000
            alpha = 2;
        else
          alpha = 1;
        end

    ptap=ptap + r.*g.*dh - alpha.*r.*v.^2 - (f.*r.*L(1).*v.^2)./
(2.*D(1)) - 0.5.*K_val(1).*r.*v.^2;

else
    for i=1:length(D)-1
    %basic pipe calculations
    A=0.25.*D(i).^2.*pi;
    A1=0.25.*D(i+1).^2.*pi;
    v=Q./A;
    v1=Q/A1;
    Re=r*v*D(i)/u;
    Re1=r*v1*D(i+1)/u;

    if Re<2000
```

```
    f=64./Re;
    else
        f=1./(-1.8.*log10(6.9./Re+(e./3.7./D(i)).^1.11)).^2;
    end

        if Re1<2000
        f1=64./Re1;
        else
            f1=1./(-1.8.*log10(6.9./Re1+(e./3.7./D(i+1)).^1.11)).^2;
        end

        %K val of reducer
        if Re<2500
            K(i)=(1.2+160/Re)*((D(i)./D(i+1)).^4-1)
        else
            K(i)=(0.6+0.48*f)*(D(i)./D(i+1)).^2*((D(i)/D(i+1))^2-1)
        end

        %alpha
        if Re < 2000
            alpha = 2;
        else
          alpha = 1;
        end


        %loop goes from 1 to [# of sections -1]. So I use this way to
 add the missing opening section
        if i==1
            ptap=r.*g.*dh - alpha.*r.*v.^2 - (f.*r.*L(i).*v.^2)./
(2.*D(i)) - 0.5.*K_val(i).*r.*v.^2;
        else
            ptap=ptap;
        end

        %total reducer pressure
        p_reducer = p_reducer + 0.5.*r.*v.^2.*K(i);
        %total pipe pressure
        ptap = ptap - alpha.*r.*v.^2 - (f.*r.*L(i+1).*v.^2)./(2.*D(i
+1)) - 0.5.*K_val(i+1).*r.*v.^2;
    end
end
ptap1=-p_reducer+ptap;
end
```

B8: Calculation of tap pressures using the function.

```
%This is the script for using the function to calculate tap pressures.
%Reminder: p_tap([Q(m^3/s),dh(m)], [ID values(m)], [L values(m)], [K
 values incl. entrance & exit])

Q = 9/1000 /60; %Tap flowrate requirement for all taps
e=0.0015e-3;  %epsilon
r=998;   %rho
g=9.81;
u=8.9e-4; %mu

%Tap 1 path parameters:

dh = 8.5+2;

p_tap1 = p_tap([Q,dh],[0.0209],[19.15],[4.75+1+0.75])


p_tap1 =

   9.9300e+04


%Tap 2 path parameters:

dh = 10.5+2;
p_tap2 = p_tap([Q,dh],[0.0209,0.0158],[47.25,20],[4.9+1,1+0.75])

A=0.25*0.0158^2*pi;
v=Q/A;
DeltaP_tap2 = 0.5*r*0.54*(4*v)^2;
p_tap2_challenge = p_tap2 + 0.5*r*1*v^2 - DeltaP_tap2


K =

    0.8065


p_tap2 =

   1.1085e+05


p_tap2_challenge =

   1.0861e+05


%Tap 3 path parameters:
dh = 10.5+2;

p_tap3 = p_tap([Q,dh],[0.0209,0.0158],[47.25,40],[4.9+1,1.15+0.75])
```

B9: Calculation of entrance and exit angles for the custom impeller.

```
% calculate beta's for impeller

% inital data based on max flowrate and pressure head
%YOU CAN CHANGE THIS
qo = 15; %L/min
q= qo/1000/60;
hp= 1.5;  % m

% pump rpm, constant
rpm = 2000;
omega = rpm/60*2*pi()

% impeller inner and outer radius
%YOU SHOUDL CHANGE R2, max = 19 mm
r1o = 9.4;
r2o = 19;
r1= r1o/1000;
r2 = r2o/1000;

%impeller thickness/width 10 mm
b1 = 0.01;
b2=b1;

%blade angle at r1, convert back to degrees from radians
beta1= atan(q/(2*pi()*b1*omega*r1^2))*180/pi();

%blade angle at r2, convert back to degrees from radians
% calc. tangential velocity at edge of impeller
V2t = 9.81*hp/(omega*r2);
% Calc. normal velocity at edge of impeller
V2n = q/(2*pi()*r2*b2);

beta2= atan(V2n/(omega*r2-V2t))*180/pi();
r2l=[14:0.001:25]/1000;

V2tlist = 9.81.*hp./(omega.*r2l);
V2nlist = q./(2.*pi().*r2l.*b2);
betafun=atan(V2nlist./(omega.*r2l-V2tlist)).*180./pi();
```

B10: Single frame particle size detection

```
% Code to detect particle size from microscope image

% Read microscope image into workspace
I1 = imread('Particles_Microscope.tif');

% Convert image into single layer grayscale image
I = mean(I1, 3);
% display grayscale image in a subplot
subplot(1,2,1);
imagesc(I);
axis image;
colormap gray;
title('Grayscale')

% Use the micropscope_calibration file to obtain the microscope
 resolution and record result here
M_um_per_pixel = 8.33;      % image resolution in µm/pixel

% Select top 1000 pixels
I2 = I(1:1000,:);

% Create binary particle mask
M_ParticleMask = (I2 <= 100);  %threshold of intensity of 100 used
% display biary mask image in subplot
subplot(1,2,2);
imagesc(M_ParticleMask);
axis image;
colormap gray;
title('Binary mask')

% extract properties from mask
CC = bwconncomp(M_ParticleMask);

RP = regionprops(CC,'Area','MajorAxisLength','MinorAxisLength');

Area = [RP.Area];
MajorLen = [RP.MajorAxisLength];
MinorLen = [RP.MinorAxisLength];

% average MajorLen and MinorLen
M_Lengths = (MajorLen+MinorLen)/2;

%calculate lengths and areas in um
M_Lengths_um = M_Lengths * M_um_per_pixel;
M_Areas_um2 = Area * M_um_per_pixel^2;

%remove the extra large and extra small particles (keep only particles
%between 100 and 10000 um2squared)
M_NewLengths = M_Lengths_um(M_Areas_um2 > 100 & M_Areas_um2 < 10000);
M_NewAreas_um2 = M_Areas_um2(M_Areas_um2 > 100 & M_Areas_um2 < 10000);

% display a historgram using 20 bins and proper axes labels
```

```
figure;
histogram(M_NewAreas_um2,20)
xlabel('Particle Area (um^2)')


% calculate mean of the lengths in um
M_NewAvLen = mean(M_NewLengths)
M_NewAvArea_um2 = mean(M_NewAreas_um2)
```

B11: Single frame velocity and density detection function (Process_Image)

```
% Code to calculate mean velocity for a single image frame

% Load the mock image file (if you don't have images acquired from
 camera.
% Comment out this line if you have images acquired from the camera in
 your
% workspace

% Extract a single frame from the 4-D image matrix
TestFrame = M(:,:,1,1);
% Display the single image frame in a (1,2,1)subplot
figure;
subplot(1,2,1)
imagesc(TestFrame);
axis image;
colormap gray;
title('Grayscale Testframe')

% Create binary mask of particle traces
TraceMask = (TestFrame >= 130);    %try different threshold values for
 your image to obtain masked streaks
% Display the single image frame in a (1,2,2)subplot
subplot(1,2,2)
imagesc(TraceMask);
axis image;
colormap gray;
title('Binary mask Testframe')

% Extract properties from mask
CC = bwconncomp(TraceMask);

RP = regionprops(CC, 'MajorAxisLength');
TraceLen = [RP.MajorAxisLength];

% Remove short trace lengths, (< 20 pixels)
TraceLen_1 = TraceLen(TraceLen > 20);
```

```
Undefined function or variable 'M'.

Error in velocity_and_density_single_frame_skeleton (line 9)
TestFrame = M(:,:,1,1);
```

Estimate mean particle velocity for a single frame

1. Camera resolution from calibration (from Workshop 8)and exposure time

```
C_per_pixel_m = 20.2197e-6;          % image resolution in m/pixel
exposure_time_s = 500e-3;        % exposure time in s
% 2. Convert TraceLen_1 from pixels to m to obtain the vector of trace
 lengths
```

```
TraceLen_m = TraceLen_1 * C_per_pixel_m;
% 3. Calculate velocities of the trace lengths, in m/s
Velocities_m_per_s = TraceLen_m / exposure_time_s;
% 4. Calculate mean velocity of the trace lengths in the single image
 frame, in m/s
mean_velocity_single_frame_m_per_sec = mean(Velocities_m_per_s)
```

Estimate particle density for a single frame

1. Calculate volume of torch beam, in cm3 Note: You will first need to obtain the width and length of torch beam, in pixels, using getpts (use the torch_width_length file)

```
width_of_torch_beam_cm = 0.5760e3 * C_per_pixel_m * 100;   % width of
 torch beam in cm
length_of_torch_beam_cm = 1266.2 * C_per_pixel_m * 100;    % length of
 torch beam in cm
depth_of_torch_beam_cm = 3;    % depth of torch beam in cm
volume_of_torch_beam_cm3 = width_of_torch_beam_cm *
 length_of_torch_beam_cm * depth_of_torch_beam_cm   % volume of torch
 beam in cm^3

% 2. Calculate particle density of the single image frame, in
 patricles/cm3.
% [Hint: use total number of traces in TraceLen]
particle_density_single_frame_per_cm3 = length(TraceLen) /
 volume_of_torch_beam_cm3
```

B12: Inline monitoring code.

```
% close all figures and clear the workspace
close all
clear all

   % Place 'tic' to time the code.  One of the last steps in this
 workshop.
tic
```

First, acquire and process 1 image before beginning the while loop

initialise video object

```
vid = videoinput('gentl', 1, 'Mono8');
src = getselectedsource(vid);

% set Exposure Time
src.ExposureTime = 10000;

% acquire single video frame
SingleFrame = getsnapshot(vid);
% process the single frame by using the Process_Image function
[Particle_mean_velocity, Particle_density] =
 Process_Image(SingleFrame)

% display the first acquired frame
figure;
imagesc(SingleFrame);
axis image;
colormap gray;
title('Image frame 1')
fprintf('Frame 1 completed.\n\n');

Error using videoinput (line 219)
Invalid ADAPTORNAME specified. Type 'imaqhwinfo' for a list
 of available ADAPTORNAMEs. Image acquisition adaptors may be
 available as downloadable support packages. Open <a href="matlab:
 imaq.internal.Utility.supportPackageInstaller">Add-On Explorer</a> to
 install additional adaptors.

Error in Run_Loop_ParticleDetection_skeleton (line 9)
vid = videoinput('gentl', 1, 'Mono8');
```

Running the while loop

```
% 7.a. set a threshold value for particle density
Threshold = 22;

% 7.d. initialise counter variable, framenum, to 1
framenum = 1;
```

```
figure;
while Particle_density <= Threshold % insert your condition here-while
 particle density is less than threshold

    % acquire new single frame into variable SingleFrame
    SingleFrame = getsnapshot(vid);
    % process the SingleFrame using the Process_Image function
    [Particle_mean_velocity, Particle_density] =
 Process_Image(SingleFrame)

    % increment framenum by 1 for each iteration
    framenum=framenum+1;

    % display frame

    imagesc(SingleFrame);
    axis image;
    colormap gray;
    title(['Image frame ', num2str(framenum)]);

    % display number of frames completed
    fprintf('Particle density = %d \n', floor(Particle_density));
    fprintf('Frame %d completed. \n\n', framenum)

end

ElapsedTime = toc;   % 'toc' to time the code

% Display message as per the workshop handout:
fprintf('In total, %d frames completed in %f seconds.
\n',framenum,ElapsedTime);
fprintf(2, 'WARNING! Particle threshold exceeded. \n\n');
```

B13: Calculation of particle concentration threshold.

```
%Script for calculating the threshold particle concentration value
 used in
%the IIMS algorithm.

%System Parameters

A_m = 7.141; %m^2
A_particle = 1.49e-9; %m^2
t_op = 3; %hr
Q_water = 2.2283e6; %mL/hr

%Failure characteristics of particle filter in vector form.

%Time in operation (months):
T_part = [8 8.5 9 9.5 10 10.5 11 11.5 12];

%Particle concentration (particles/mL):
C_part = [0.1 0.23 0.51 1.15 2.59 5.84 13.16 29.69 66.96];

%Membrane fouling

A_failure = 0.74*A_m/3; %m^2
n_foul = A_failure./A_particle; %number of particles required to foul
 membrane
n_1day = t_op*Q_water*C_part; %number of particles which contact the
 membrane in one day.

d = n_foul./n_1day; %days until membrane is fouled

t_foul = d.*t_op %Hours of continuous operation until membrane is
 fouled

n_filters = floor(5*12./T_part) %number of filter changes in 5 years

t_filters = t_op.*n_filters %total number of hours of contaminated run
 time
```

B14: Impeller Scale-Up

```
%This is the m-file for ESD1, Salce Up for Impeller in Design Section
 1
%Scaling up an impeller to meet a flow specification

%system curve data flow rate (L/S), Head loss (m)
flowrate = [1.337 1.4361 1.5352 1.6343 1.7334 1.8326 1.9317 2.0308
 2.1299 2.229]*1000/3600; %L^3/s
headloss = [23.68 23.99 24.3 24.6 24.93 25.25 25.56 25.875 26.18
 26.5];


%pump curve data pump flow rate (L/s), performance (m),

%Student put in their best pump curve
pumpflow = [16.51 15.93 14.18 12.10 9.95 6.89 3.89 0.6]/60;
perform = [1.5030    1.5235    1.5644    1.7382    1.8506    2.0551
 2.1369    2.3516];

%Operating Fluid point in  L/s
opflowpoint=0.6109825
opflow=[opflowpoint opflowpoint];
opheight=[0 max(headloss)];

%Current Impeller Diameter in mm from student design
D1= 38

% New impeller diam in mm . will change iteratively
D2 = 143

%Impeller Ratio, d2/d1
dratio= D2/D1

%New Pump Curve . scale old pump data with ratio, remember to use .*
%operator,   scale according to Affinity laws
%(one is dratio, one is dratio^2, which one?
pumpflownew = pumpflow.*dratio
performnew = perform.*(dratio.^2)

%plot curves
plot(flowrate,headloss,'ro-')
grid on
hold on
plot(pumpflow,perform,'k^-')
plot(pumpflownew,performnew,'bs-')
plot(opflow,opheight,'g-')
xlabel('Flow rate (L/s)',"Fontsize",26)
ylabel('System head (m)',"Fontsize",26)
title('Scaling Up Impeller-Section 1 Design Project',"Fontsize",26)
%Decide legend location after looking at plot
legend('System curve','Pump curve','New Pump Curve','Op
 Flow','location','northeast', "Fontsize",13)


%legend('System curve','Pump curve','New Pump
 Curve','location','northeast', "Fontsize",12)
hold off
```

Appendix C: Equations, Data and Calculations

C1: Chick-Watson's Law

$$log_{10}\left(\frac{N_o}{N}\right) = \frac{k_p C_o}{k'}\left(1 - e^{-k't}\right)$$

Equation 23: Chick-Watson's law of disinfection, the quantity on the left is the log-inactivation credit.

C2: Ozone Decay Constant

Based on the tabulated data of half-lives for ozone:

Temperature (°C)	O_3 Half-Life (min)
15	30
20	20
25	15
30	12
35	8

Table 21: Half-lives of ozone under different temperatures.

The ozone decay constant can be calculated by rearranging:

$$ln(0.5) = -k't$$

Equation 24: Calculating ozone decay constant, t is the minutes of a half-life from table 20.

C3: K-Values for Fittings

Below is the table of K-values for different fittings:

Fitting	K
45 ° elbow	0.35
90 ° elbow	0.75
180 ° bend	1.5
Tee – run through – branch blocked	0.4
Tee – all other flow patterns	1
Coupling	0.04
Union	0.04
Pipe exit	1
Pipe entrance	0.75
Gate valve – open	0.17
Gate valve – ¾ open	0.9
Gate valve – ½ open	4.5
Gate valve – ¼ open	24

Table 22: K-values for fittings.

C4: Calculating Friction Factor.

$$Re = \frac{\rho v d}{\mu}$$

Equation 25: Expression for the Reynold's number, used in calculating the friction factor.

$$f = \frac{64}{Re}$$

Equation 26: Expression for the friction factor when Re < 2000.

$$f = \left(\frac{1}{-1.8 log_{10} \left[\frac{6.9}{Re} + \left(\frac{\epsilon / D}{3.7} \right)^{1.11} \right]} \right)^2$$

Equation 27: Expression for the friction factor when Re > 2000.

C5: Calculating Total Water Requirement.

The clinic beds each require 271L of water per day and there are 5 clinic beds.

The clinic has an area of 250 square meters, and each square meter requires 2.48L per day.

Each person requires 2L of water per day, however, the population accounted for is the population after 10 years, this is given by:

$$P = 2000(1.012)^{10}$$

Equation 28: The projected population after 10 years with an initial population of 2000 and an annual growth of 1.2%

To account for unexpected increases in growth rate, 5% of the final population is added.

Summing each contribution:

$$V_{perday} = (250 * 2.48 + 2000(1.012)^{10} * 1.05 + 271 * 5)L = 5955L$$

Equation 29: The total water requirement per day for the community used in the design.

C6: Membrane Simulator

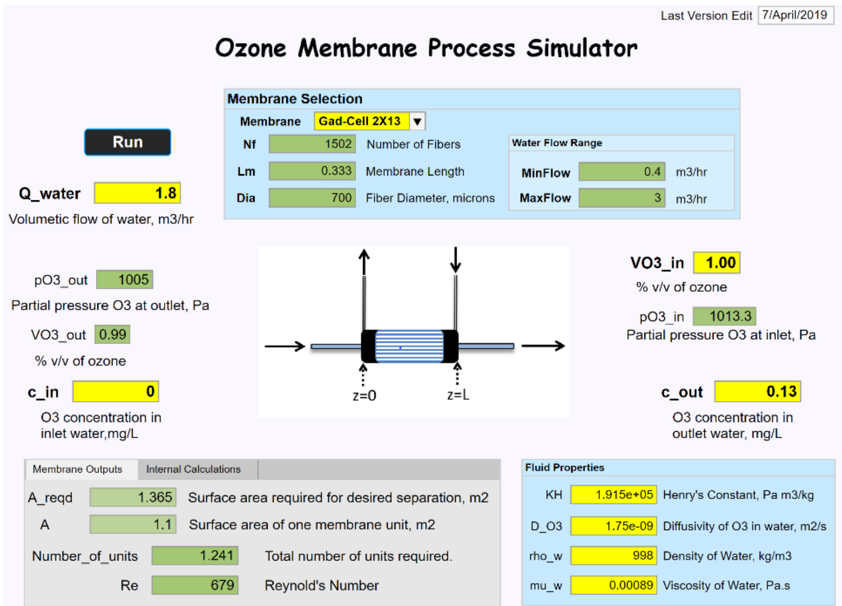

Figure 30: The membrane process simulator interface. The yellow boxes are variables depending on specifications and constants, while the yellow boxes contain the outputs of the experiment.

C7: Log-Inactivation Credits

	3 hours	4 hours	5 hours
15m	2.4667	3.2088	3.9141
20m	3.2088	4.1414	5.0128
25m	3.9141	5.0128	6.0223

Table 23: The log-inactivation credits for giardia over a DN100 contactor section with varying operational times for the treatment process and contactor length.

	3 hours	4 hours	5 hours
15m	4.9898	6.4910	7.9178
20m	6.4910	8.3775	10.1403
25m	7.9178	10.1403	12.1823

Table 24: Log-inactivation credits for viruses with same variables as the calculation used in table 22.

C8: Filter Lifespan

Time in Operation, T_{part} (months)	Particle Concentration, C_{part} (particles/ml)
8.00	0.10
8.50	0.23
9.00	0.51
9.50	1.15
10.00	2.59
10.50	5.84
11.00	13.16
11.50	29.69
12.00	66.96

Table 25: Failure characteristics of the particle filter.