MULTI-DISCIPLINARY OPTIMIZATION OF A MEDICALS SIGNALS SIMULATOR

Semester: W2021

Subject: AE 8139 – Multidisciplinary Design Optimization of Aerospace Systems

Professor: Dr. Joon Chung PhD

Students: Oleg Prykladovskyi C.E.T., Ezeorah .U. Godswill & Dustin Bruce

Table of Contents

Table of Figures

No table of figures entries found.

Abstract (D)

In the diagnosis and treatment of medical conditions dealing with the urinary system, accurate measuring devices are necessary to measure and assess the urinary voiding process. A Urocap device, designed and manufactured by Laborie Medical Corporation, is a wireless Bluetooth device that measures urinary flow rate and volume. As the accuracy of this device is paramount, a medical signals simulator is required to verify if the system is properly calibrated. The current simulator being used by Laborie Medical is overly large and unreliable, presenting a distinct need for optimization. To improve upon this, an attempt has been made to optimize the system using a multi-disciplinary approach. Being that the size of the system was the biggest problem, the focus was to rebuild the simulator to reduce the size while still maintaining the required flow rates. Accomplishing this required a singular payout function in conjunction with the Jacobi method, to optimize across the three different disciplines involved. The optimization process was successful in that the simulator was reduced in size to a total volume of 1449 cm^3 from 180,000 cm^3 while still maintaining its function and accuracy.

1.0 Introduction (Sorry Dustin, didn't see the D letter, so filled out this section too - O)

Human body is the most complicated mechanism created by mother-nature that exists in modern world. It consists of 12 subsystems which interact with each other very closely 24 hours a day, 365 days a year. Every minor disturbance of their functionality can cause the severe consequences and even death, unless there is an appropriate medical treatment provided by healthcare professionals. Usually, specialized medical equipment is used for that purpose, therefore medical research and development is an important part of modern health system engineering in Canada. One of Canadian leaders in medical products development is Laborie Medical.

Laborie Medical Technologies Corporation is a leading global developer, manufacturer and marketer of innovative medical technology and consumables used for the diagnostic and treatment of pelvic health in the Urology, Gynecology, and Colorectal fields as well as gastrointestinal procedures. Their products are well known for the high quality, affordability and precision. They are also popular for their easy mechanical and calibration maintenance due to the fact that each system consists of multiple subsystems, joined together through Bluetooth connection and they simply can be detached from the main tower . Performance stability is a key factor too, and is achieved by constant improvement, verification and validation of products in-house. Multiple testing systems are used for that purpose.

One of the most critical parts of Laborie equipment is Urocap. Urocap is a portable, wireless device that quickly and accurately measures urinary flow and volume. This test is an initial test performed by doctors and nurses on patients to understand if there is a problem with bladder and/or urethra muscle and develop a further plan of treatment.

Urocap performance correctness is checked by using the Medical Signal Simulator system as shown on the Figure 1 below:

Picture 1. Medical Signal Simulator

System contains the reservoir with water on top of the assembly. Upon the command, received from the terminal controlled with CX-Programmer software, valves at the bottom switch to "open state" and Urocap receives 500ml of water. Three main flow tests must be performed – for 10 ml/s (pediatric patient), 30ml/s (adult patient) and 50ml/s (request from Clinical Department, based on received feedback from doctors across Canada)[1]. Urocap itself consists of two parts – top and bottom halves. Top half contains a piezo-electric sensor that reads the pressure from the top of the Urocap, transforms it into an electrical signal and sends it to the bottom half with the PCB board. PCB board performs the ADC conversion, and after further calculations done by the firmware, transmits the signal to the PC station with Urodynamics Software, created by Laborie company in the past, where it is graphically presented for further analysis. The simulation system was created in 2008 and upon day-to-day use, accumulated a number of problems since then which became the primary reason for the optimization. Problems are as follows:

- Electronics – consists of variable PCB boards, which contain expensive components (custom-made resistors and capacitors, old model of processor) which will be almost impossible to replace in case of the breakage.

- Tubes – used to transfer water from reservoir to valves. Because they accumulate algae within two-three weeks of operation and also are subjects to stress at the points of connection to the valves, they become unreliable and must be changed every month.

- Valves – contain metal noses water goes through in case of "open state". Therefore, noses get rusty and overall measurements are distorted as pressure becomes unstable.

- Water reservoir – is not sealed properly. There is no pump or other system that may push the water out of the tank with constant pressure. It is assumed that pressure is stable due to the amount of water that the tank contains (~ 20 liters), however the pressure becomes unreliable when the amount of liquid in the tank gets to the low value (~2-5 liters).

- Dimensions – station is too big (3.5m high, 2m wide and 2m long). Due the reasons described previously, the simulator requires constant maintenance, and it is not convenient to perform it.

- PLC Software – requires manual setup each time the measurements become smaller or higher of a defined value, due to the reason that time required for valves to be in "open state" is maintained in the PLC software program.

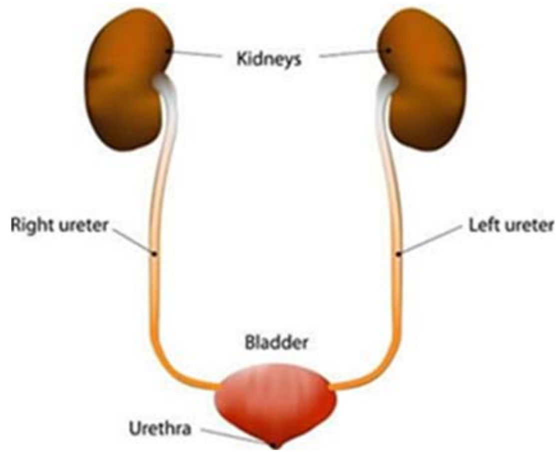
Overall, properly performed MDO may improve the simulator issues described above and have a positive impact on Production performance results, customer's satisfaction and patient health.

2.0 Theory Review (O)

2. THEORY REVIEW

Overall, the human's urinary system consists of kidneys connected to the bladder through the ureters, and urethra **1**.

Picture 2. Urinary system sketch



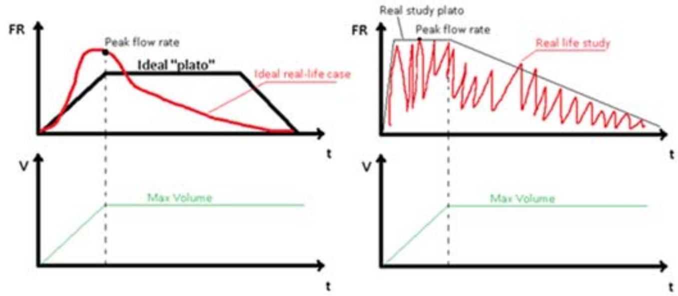
By conducting clinical research, it was defined that bladder refilling speed is around 2 ml/minute, which leads to approximately 500 ml of liquid every 5 hours. As soon as pressure, created by the liquid inside the bladder, becomes greater than the sum of pressures of bladder walls and environment around them, human's brain receives a strong impulse which stimulates the desire to void.

Urethra is one of the important muscles in the urinary system that is responsible for the bladder's deflation. However, if the specified muscle does not work properly, the urology disorder appears. Length of urethra in male's body is approximately 5 cm, while in female's body is 2 cm. Therefore, the two most common disorders in urology field are urethra's constriction (mostly, male patients) and leakage (mostly, female patients). Due to the nature of the specified disorders, Urocap is designed with intention to capture the constriction issue, as leakage issue can be clearly observed.

Urocap product consists of 2 parts – sensoric top part and processing bottom part. Bottom part also uses Bluetooth dongle to transmit the received measurements onto the computer which is capable of visualizing the study measurements in the form of 2 graphs with the help of developed UDS Software. Upper side of the graph demonstrates the Flow Rate – Time relation, while the lower side of the graph shows the Volume – Time relation.

Ideally, the graph shall demonstrate the "Plato" area, where patient's pressure is constant and in reality, study looks as a normal distribution graph shifted to the left of the picture. But if there is an issue, graph becomes distorted, the disorder is clearly observed, and doctors must make assumptions regarding peak values and "Plato" area. Picture 3 below shows the related cases:

Picture 3. Ideal uroflowmetry study vs. Real one.



In terms of signal simulator, "Plato" graph is targeted to achieve the highest level of measuring accuracy. Therefore, such disciplines as structural design (geometry), fluid dynamics, physics (in relation to fluid dynamics) and control system are very important to consider for optimization.

3.0 Problem Statement (D)

To fully understand the scope of the optimization, a problem statement was needed to encompass the objective. At first, the optimization process was going to revolve around the dimensions of the station reservoir as well as the flow rates from the different valves. This, however, presented the need for multiple objective functions, one for the flow rates of the system, and one for the dimensions of the station. It became apparent that a single objective function/payout function was necessary that encompassed all the variables of the optimization problem. Because of this, the scope of optimization was altered to put emphasis on just the dimensions of the system while maintaining the required flow rates. This in turn, yielded the following statement:

Create a new medical signals simulator that decreases the device dimensions by changing the volume of the water reservoir while maintaining stable flow rate and test volume. This will be accomplished by changing the physical design of the tubing, valving, and reservoir while still maintaining the testing rate of the previous station.

4.0 Disciplines Involved (O)

Because the objective function is concentrated on reducing the dimensions while maintaining required test levels of volume and flow rates, there are 3 disciplines totally involved:

- Structural design (geometry)
- Fluid dynamics
    - Physics

Each of the disciplines is described in the following subsections.

4.1 Structural Design (geometry)

Geometry is involved due to the reason that in order to reduce the dimensions of the station, the occupied space (volume) of the station shall be reduced. Volume of the station is completely related on the volume of it's water reservoir, calculated by the following formula:

$$f1 = L \cdot W \cdot H = x_1 \cdot x_2 \cdot x_3$$

where, L - length of the reservoir, W – width of the reservoir and H - height of the reservoir.

4.2 Fluid Dynamics

Fluid dynamics is another important discipline to consider, as it provides the formula for core parameter – flow coefficient **2** or the valve's capacity for a liquid or gas to flow through it, - based on which the correct solenoid valve can be chosen:

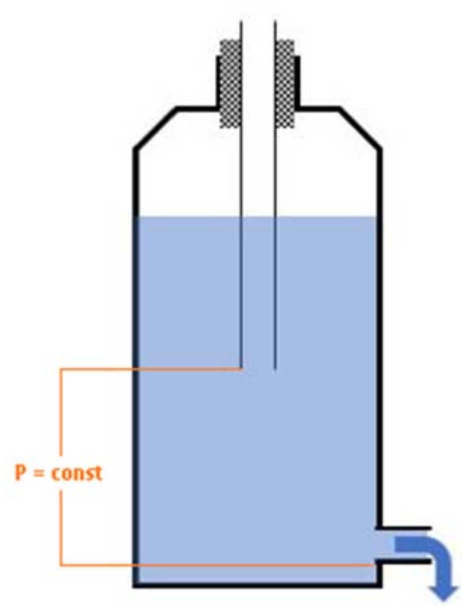
$$C_v = 10\sqrt{\frac{1}{(9.8x_3 - x_5)}}$$

where, C_v – given flow rate, 10 - flow rate of the valve, 1 – specific gravity for water, x_3- height of the reservoir derived from the hydrostatic pressure (please, see section 4.3 of this report) and x_5 – is the outlet pressure of the valve.

4.3 Physics

In this project, process physics is not the discipline to optimize, however the physical phenomena of Mariotte's bottle is used to perform the optimization.

Picture 4. Mariotte's bottle principle



Mariotte's bottle was invented in 17th century by French physicist Edme Mariotte **3**. It is a device that delivers constant flow rate from sealed bottles or containers. As long as the liquid remains above the bottom of the tube that determines the exit pressure, the exit pressure stays constant regardless of the level of the liquid in the container. In modern world, it is widely used in IV bags in medical industry and in irrigation systems related to agriculture or biological laboratories.

To describe the pressure of the hydrostatic pressure inside the bottle, the following expression is used **4**:

$$P=\rho\cdot g\cdot h$$

where ρ – water density (1 kg/m^3), g – standard gravity acceleration (9.8 m/s^2) and h – the depth of the tube injected into the container which is proportional to the height of the container x_3 and is an input to this discipline. Therefore, the valve's inlet pressure is $9.8 \cdot x_3$ – the pressure at the valve's inlet.

5.0 Objective Function (D)

The three disciplines used in this optimization process each of a unique equation or set of equations that need to be considered when formulating the final payout function. As described in section 3.0 above, the two main equations to consider are the volume and flow coefficient of the station. Since the varying flow rates required are generated from 3 valves of equal flow rates, the optimization of the reservoir must take place while ensuring sufficient pressure is maintained through the system. Because of this, both the flow coefficient formula and geometric formula were combined to create the final payout function below

$$f1 = x_{12} \bullet \frac{\frac{100}{x_4} + x_5}{9.8}$$

Where x4 is the flow coefficient, x5 height of the reservoir and x12 the length/width of the reservoir. The above formula can be broken down into its original components through examining the three disciplines involved:

1. Physics
2. Fluid Dynamics
3. Structural Design

1. Physics - The hydrostatic pressure formula:

P1 being the inlet pressure, 9.8 the force of gravity, 1 the density of water and X3 the height of the reservoir itself.

2. Fluid dynamics – The flow coefficient equation:

Where x3 is the height of the reservoir as described above, and X5 is the outlet pressure. Since the dimensions of the reservoir are what is being optimized, the above equation is rearranged to express x3:

3. Structural design – Volume of the reservoir.

Where x1, x2 and x3 are the length width and height of the reservoir. x1 and x2 are variables and therefore are then coupled into x12. The volume of the reservoir must be 1150ml yielding the following formula:

Now, substituting the formulas of X1 and X3 yields the final payout function:

6.0 Constraints(O)

Based on the optimization approach, hardware limitations and tight project deadlines, there are total of 5 constraints considered for this optimization:

- Each valve must provide the flow rate of 10ml/s
- 5 valves must be used due to the 3 flow rates required (10 ml/s, 30 ml/s and 50 ml/s) **5**
- Volume of the tank must be > 1150 ml (500ml for 30 ml/s and 50 ml/s studies, and 150 ml/s for 10 ml/s study)
- Pressure of the valve's outlet < Pressure of valve's inlet (i.e., P2< 9.8·H3)
- The height between the valves and table where the station is placed must be > 24cm

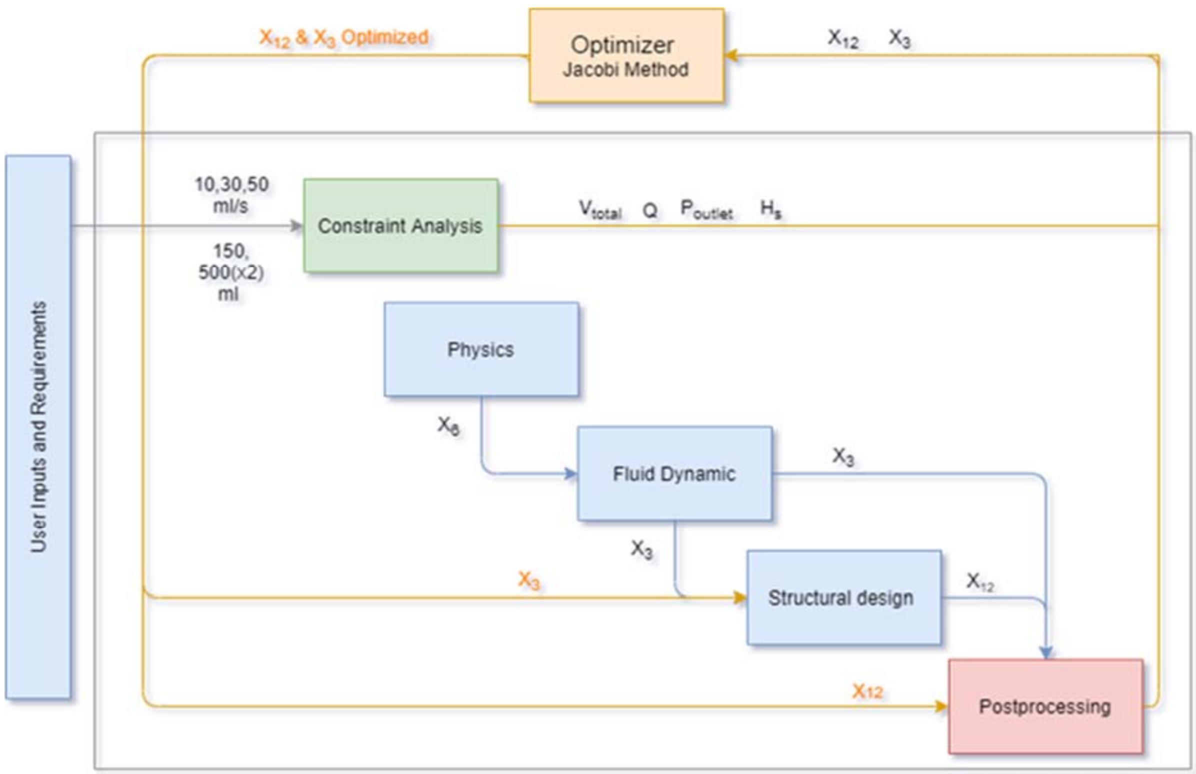
7.0 Variables

Important design variables to be considered are:

- Volume of water reservoir, Vw.r., has 3 variables: length L3, width W3 and height H3

  V = L*W*H = x1*x2*x3

8.0 Optimization Architecture (D)



9.0 Optimization Process

9.1 Optimizer/Jacobi Method

We utilize the Jacobi's method, which is an iterative method of the form 6:

For each $k \geq 1$, generate the components $x_1{}^{(k)}$ of $x^{(k)}$ from $x^{(k-1)}$ by

$x_i{}^{(k)} = \frac{1}{a_{ii}}\left[\sum_{j=1,j\neq i}^{n} \quad (-a_{ij}x_j{}^{(k-1)}) + b_i\right]$, for $i = 1,2,,\ldots n$

For example if we were given the following problem:

$$5x_1 - 2x_2 + 3x_n = -1$$

$$-3x_1 + 9x_2 + x_n = 2$$

$$2x_1 - x_2 - 7x_n = 3$$

we can solve this iteratively by make the variables subjects of the given formula as,

$$x_1 = \frac{-1}{5} + \frac{2}{5}x_2 - \frac{3}{5}x_3$$

$$x_2 = \frac{2}{9} + \frac{3}{9}x_1 - \frac{1}{9}x_3$$

$$x_3 = \frac{-3}{7} + \frac{2}{7}x_1 - \frac{1}{7}x_2$$

and so on, until the n^{th} term. So that each new found variable is used to solve for the next variables until the entire system converges.

9.2 Optimization Process (Algorithm)

Input: Initial design variables x_2, x_3

Output: Optimal Variables & Parameters $x_3{}^*, x_1{}^*, P_2$

0: Initiate System Optimization, $x_3{}^0$

Repeat:

1: Initiate MDA Optimiser, $P_2{}^0$, $x_1{}^1$

Function Call: Structural Discipline Constraint Model (Input: $x_1{}^1$)

2: Compute Discipline Constraints, using $x_2{}^1$

3: Filters Discipline Physics (Optimize) for $x_3{}^1$

4: Returns $x_3{}^1$

5: Updates Local design Variables $x_1{}^2$

6: Compute Fluid Discipline Physics, $P_2{}^1$

7: Compute Fluid Discipline Physics, $x_3{}^2$

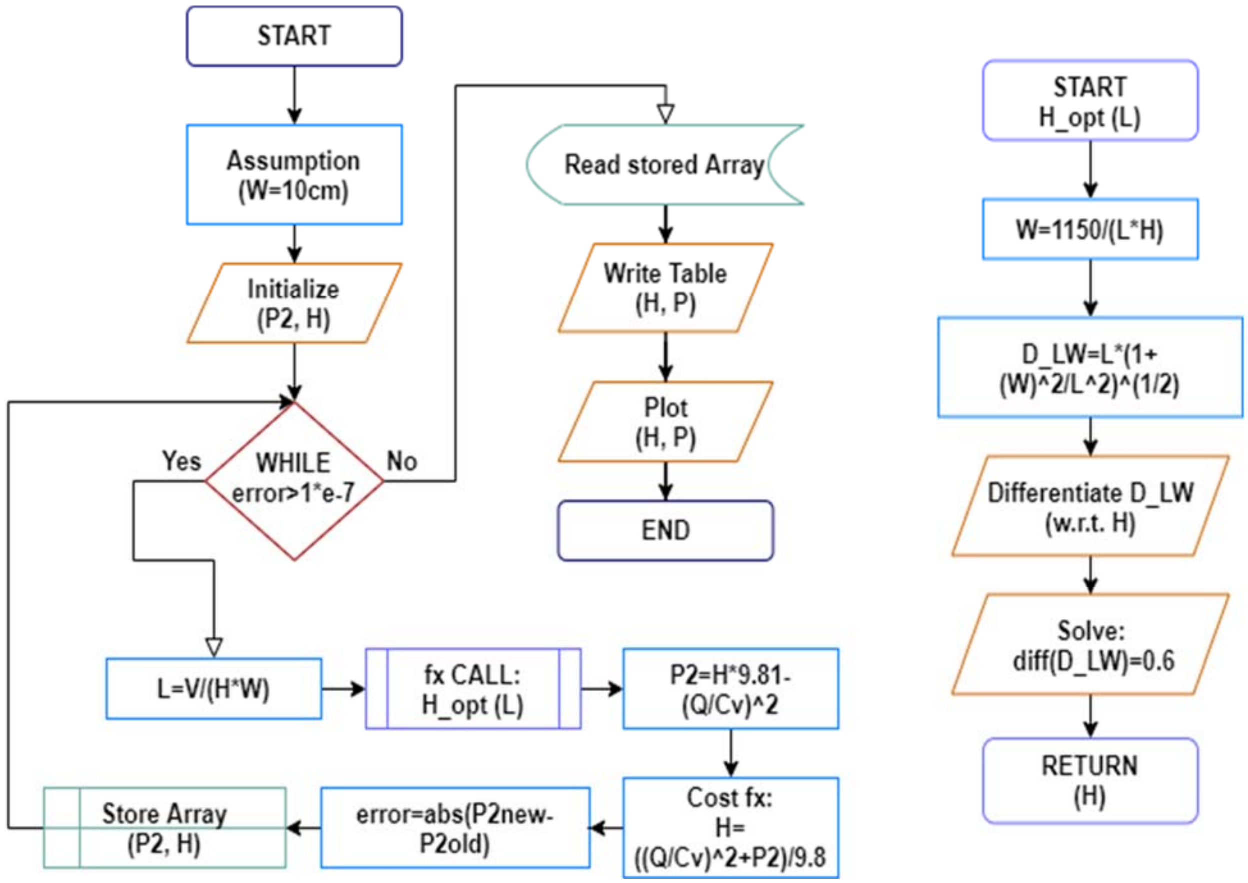
8: Evaluate Error between $P_2{}^0$ and $P_2{}^1$

Until: 8 < tolerance: MDA has Converged

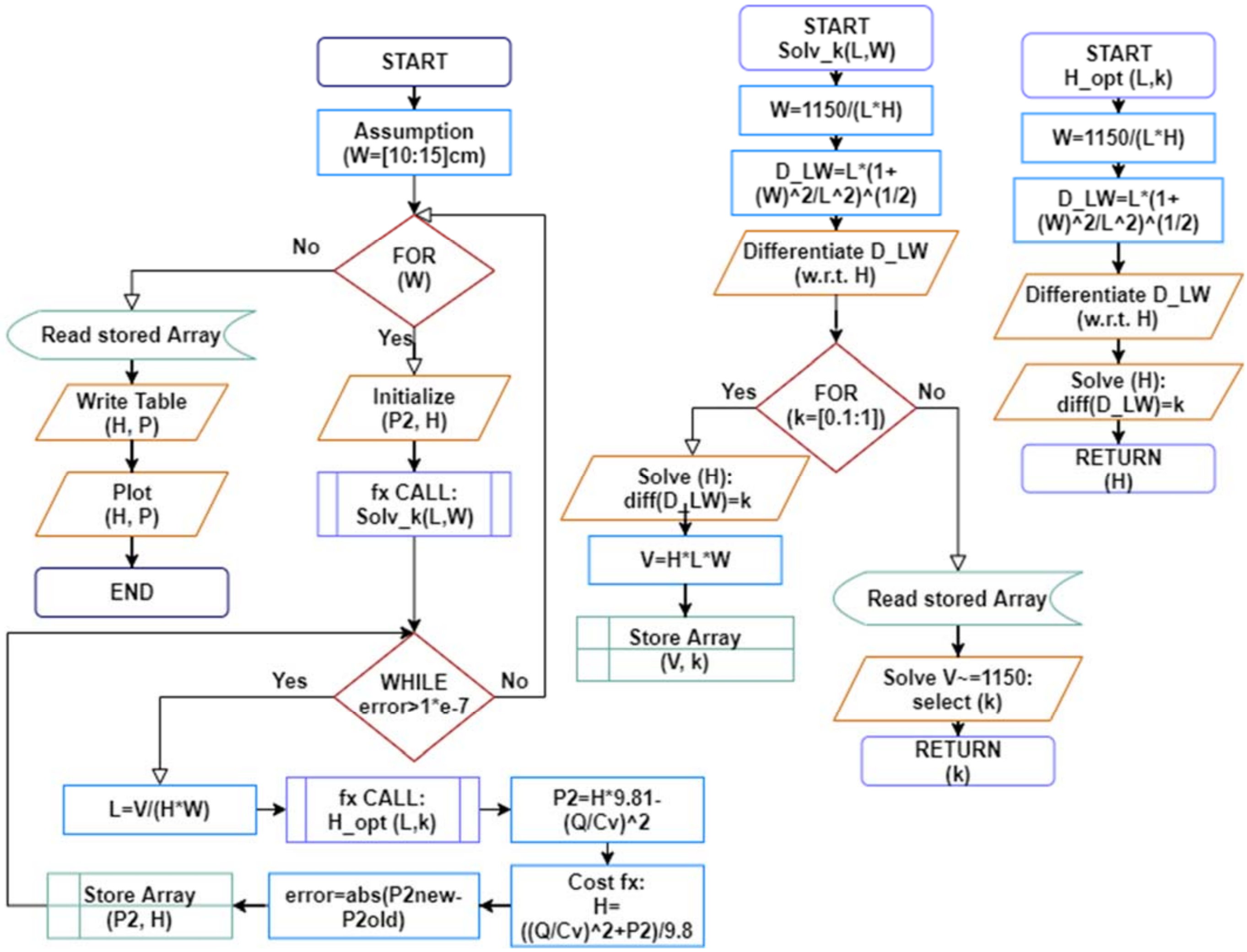
9.3 MATLAB Code FlowChart

The below flowchart shows the computational process in which the optimization simulation is performed.

First MATLAB Script: This consist of one dependent function

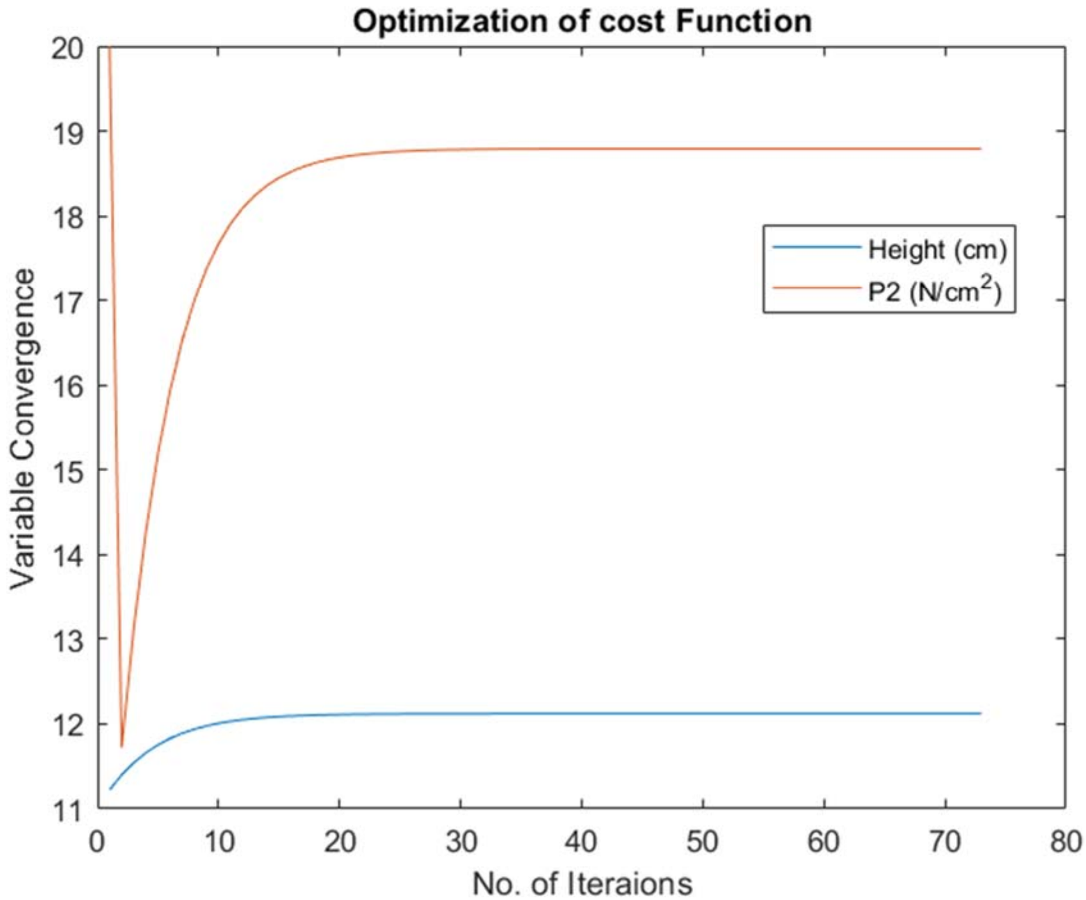


Second MATLAB Script: This consists of two dependent functions with more robust design variable inputs.

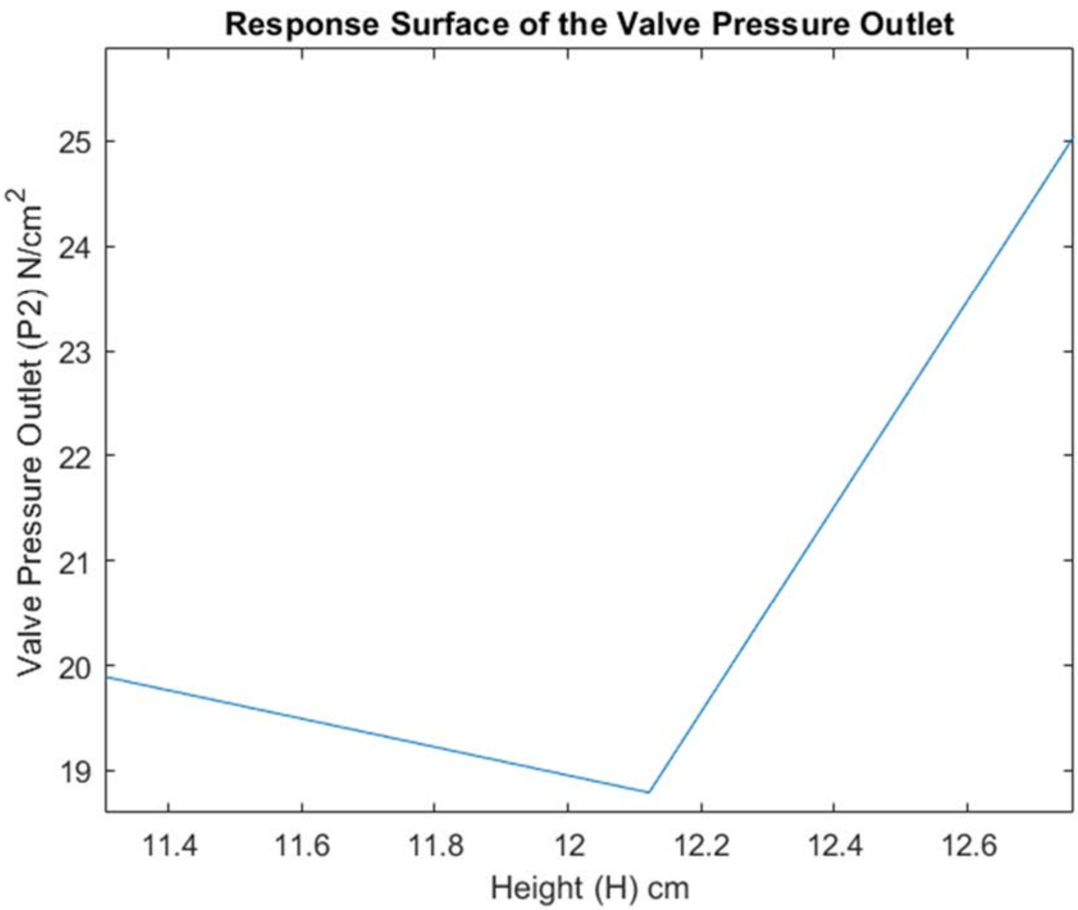


9.4 Optimization Results

The optimization was done in MATLAB, using a unique derivative of the planar diagonal of our reservoir to correct and constrain the analysis. The results are shown in the below plot.



We can see that our optimized reservoir Height,H=12.1216 cm, for our initial condition of reservoir Width, W=10 cm. According to our objective statement we want to reduce the size of the station while maintaining the stability of our system. Hence, we expanded the optimizer to perform analysis for the reservoir range of the Width variable, W=[10,11,12,13,14,15] cm, we got the response surface plot of our Valve Pressure outlet to the reservoir height.



We know that the stability of our system, which is dependent on the Pressure Differential (ΔP) between Valve Pressure inlet (P1) and Valve Pressure Outlet (P2). Now, from the above plot we can see that P2 gives the least responds when, H=12.1216 cm. And the corresponding Valve Pressure Outlet value is P2=18.7915 N/〖cm〗^2.

The resulting optimized value can be validated in conformity with our constraints. For instance, we have that P2< 9.8·H3, substituting H3=12.1216 cm, we will get P2<118.7917 N/〖cm〗^2, which is true as previously described.

10.0 Optimized Station Prototype (O)

Optimized prototype consists of 3 major components:

- Stand - involves mechanical design and further assembly
- Control unit - consists of hardware aspect and software aspect of labor
- Laptop and Laborie GUI software - provided by Laborie company, laptop and installed software are responsible for reading the signal and interpret it through the graphs and nomograms

Each section above is going to be discussed separately.

10.1 Stand

Stand consists of actual carcass and mechanical parts. It is assumed that carcass material does not affect the outcome of the optimization, therefore wood was chosen as a prime building material. Other parts are valves and the water reservoir.

Each valve parameters are as follow **7**:

- Brand: HAK Solenoid Valve
- Inlet connection type: G-1/4"; Outlet connection type: G-1/4"
- Input voltage: 12-24V
- MAWP: 10 $\frac{kgf}{cm^2}$ or 980 kPA

It was decided to build water reservoir based of a food container due to the following reasons:

- The lid of the container can be detached from the body of the container, which guarantees the easiness of maintenance of the upper part of the station (i.e. cleaning of the reservoir from algae, installation of additional components)
- The rubber part of the lid guarantees the complete seal of the container and as a result – vacuum inside of the reservoir. This is required to satisfy the requirement of Mariotte's bottle principle

Also, to get the water from the tank and distribute it to the valves in a proper manner, various push-on hose fittings were used together with a 1-to-3 manifold unit and tubes of appropriate size.
Based on the optimization script, provided by Godswill, provided in section 9.3 Optimization Results, the dimensions of the food container are as follows:

- Height = 12.12 cm
- Length = 9.48 cm
- Width = 10.0 cm

Assembly process was conducted as follows:

The process happened in several stages as follows:

- Carcass parts were cut from the sheet of wood according to the required dimensions
- Carcass parts were then attached together, using the glue gun (parts were too thin to use screws)
- Container was processed to attach the hose fitting at the bottom and pipe on top of the lid
- Container was placed on top of the stand and manifold was attached through the tube
- Valves were then attached to the manifold and placed onto the bar in the middle of the stand, using the Velcro strap
- Tubes were heated with a heat gun to create the required shape and reduce the stress to Velcro straps
- Output hose fittings were adjusted to set the required flow rate
- Valves were connected to the power supply, using the wires

The final assembly can be observed on Picture 5 below:

Picture 5. Prototype Final Mechanical Assembly

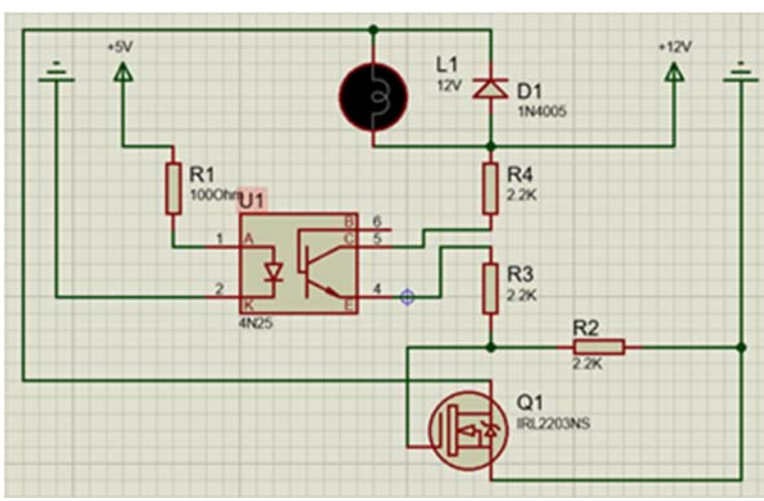
10.2 Control Unit

Control unit is required to manage the choice of flow tests and their sequence. It is also required to drain the flow station after the work hours to decrease the chances of algae appearance in the water reservoir. It consists of 2 parts - hardware and software.

10.2.1 Control Unit Hardware

It was decided to create the hardware part of a Control Unit, utilizing the Arduino Mega 2560 product, which is widely used in the world for both hobby and industrial applications. Additionally, the Arduino Mega shield board was used on top of the microcontroller for splicing the custom circuit which controls the valves. The example of the circuit board for 1 valve can be seen on the Picture 6 below:

Picture 6. Control Circuit for 1 Valve Control



Simulation was conducted in the Simulink environment. On the picture above, 2 voltage sources are present:

- 5 V for powering the Arduino controller
- 12 V for powering the valves

5V source is connected to the U1 chip - 4n25 optocoupler, which controls the 12V voltage supply to the valves. Resistor R1 is used for a safety purpose.

12V pin is connected to the valve and to the emitter pin of the optocoupler. Resistor R4 is used for safety purpose as well. Ground pin of 12V source is connected to the source of the Q1 MOSFET transistor (2203 model).

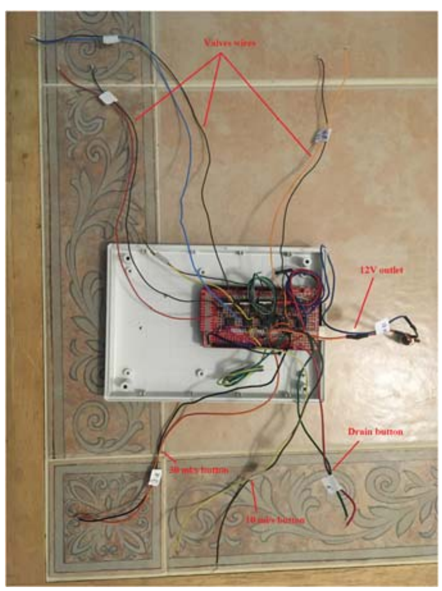
As soon as the control button pressed, 5V power source provides a voltage to the input of the LED (pin 1 on 4N25 diagram) of the optocoupler for a corresponding time, which triggers the conduction on the optocoupler's phototransistor base side (pin 6 on 4N25 diagram). As soon as conduction is present, the 12V signal transfers from the emitter (pin 5 on 4N25 diagram) to the collector (pin 4 on 4N25 diagram) and appears on the voltage divider (R3 and R2) which divides 12V into 6V by the following formula:

$$V_{out} = V_{in}\frac{R2}{R2 + R3} = 12V\frac{2.2}{4.4} = 6V$$

6V then gets to the Gate pin of the MOSFET and enables the transition of the 12V signal ground from source to drain pins of the MOSFET. Ground signal appears at the other side of the valve and closes the loop. Valve becomes and stays "open" until the timer on 5V source ends.

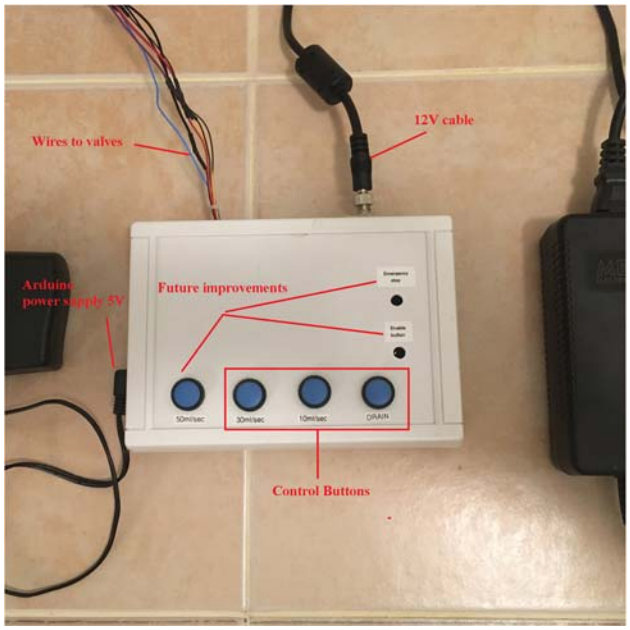
There are a total of 5 circuits, similar to the one presented on picture 6, that are connected in parallel to control the whole stand. The result of splicing activity can be seen on picture 7 below:

Picture 7. Complete Control Circuit



And the overall control unit looks as show on picture 8:

Picture 8. Assembled Control Unit

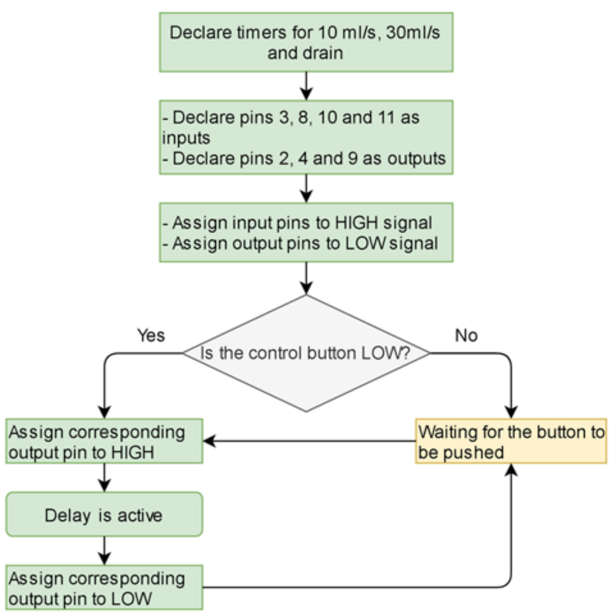


Due to the shortage on the supplier's side, it was decided to assemble the control unit with counting the future improvements in. Therefore, Enable and Emergency Stop buttons, as well as the 50ml/s button, are present on the control panel, but are not working yet.

10.2.2 Control Unit Software

Arduino Controller requires to be programmed by Arduino IDE scripts, which are written in C++ language. The script used for current application, can be seen in Appendix 3 of this document, however block diagram can be observed on picture 9 below:

Picture 9. Arduino Control Script



As per constraints, described in section 6.0 of this report, timers for each flow rate are calculated by the following formula:

$$t = \frac{V}{FR}$$

where V - required volume, FR - given flow rate.

For 10 ml/s study, the timer has to be set to:

$$t = \frac{150}{10} = 15\ seconds$$

For 30 ml/s study, the timer had to be set to:

$$t = \frac{150}{30} = 17\ seconds$$

For Drain (in case if operator finished testing and need to dry the water tank to prevent algae appearance) the timer has to be set to:

$$t = \frac{1500}{30} = 50\ seconds$$

Lastyl, for a future improvement, 50ml/s timer has to be set to:

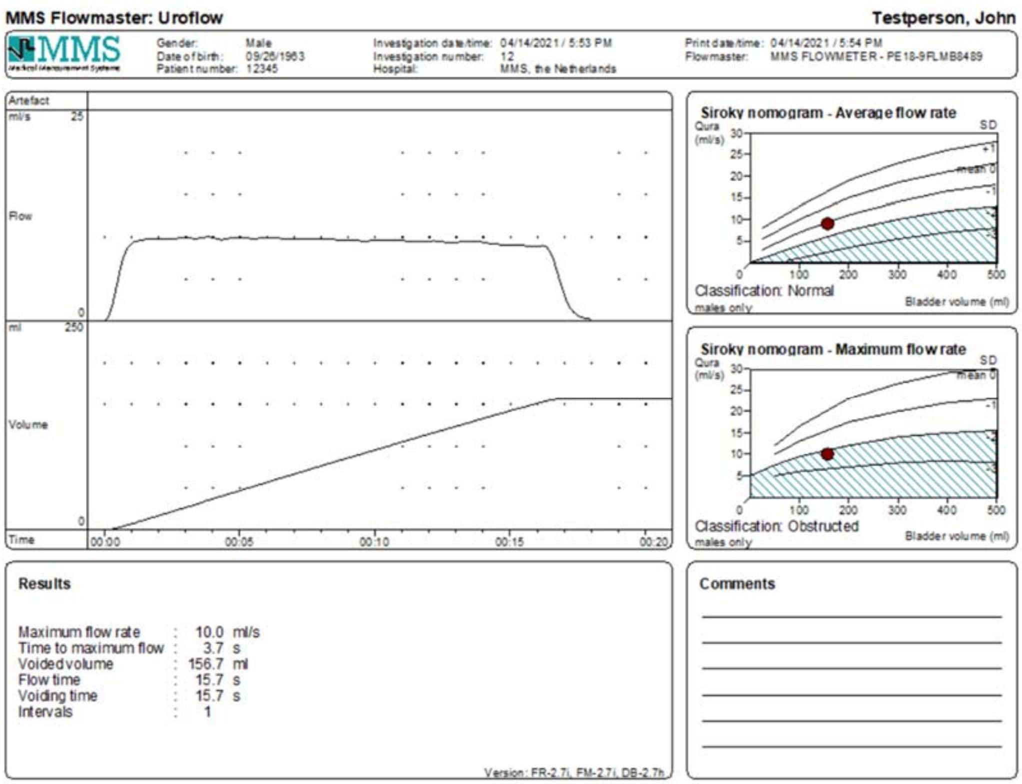
$$t = \frac{500}{50} = 10\ seconds$$

Also, in case of this application, Arduino button pins are set to 5V at the beginning and valve pins are set to 0V. As soon as the control button is pressed, the voltage at the button pin becomes 0V and the valve pin receives a 5V signal which enables the circuit further.

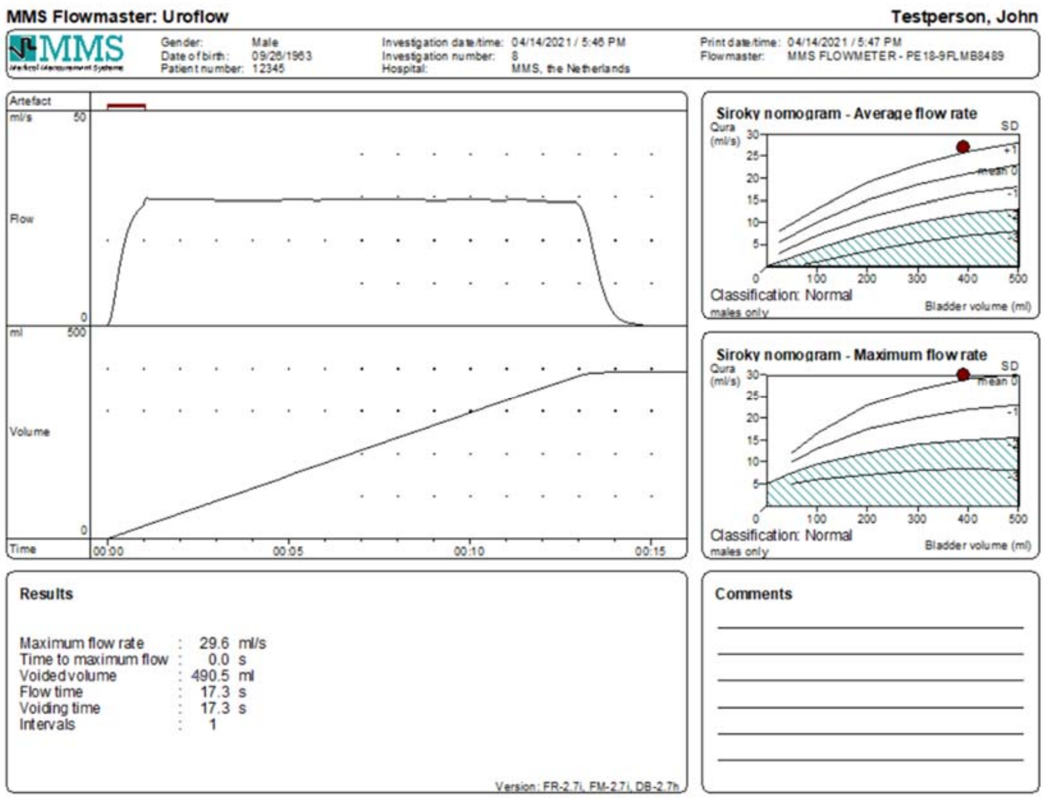
10.3 Software Response and Investigation Protocols

Graphical User Interface is represented by Laborie Urocap software, created at Product Development stage back in 2013. As soon as the patient stops voiding and Urocap's load cell does not get a signal anymore, software automatically creates the report with initial analysis for doctor's consideration. For this project, 10 ml/s and 30 ml/s investigations are presented and their graphs can be seen on picture 10 and 11 below:

Picture 10. 10ml/s Investigation Report



Picture 11. 30ml/s Investigation Report



As per images above, 10 ml/s flow rate, simulated by the prototype, is very accurate and does not have error.

For the 30 ml/s, the minor error is present (~0.4 ml/s), however it is within the tolerance range of $\pm$2%.

Volume for 10ml/s flow rate is 156.7 ml, which identifies a minor error, however it is within the tolerance rage of $\pm$10 ml, set by clinical research.

Volume for 30 ml/s flow rate is 490. 5 ml, which also identifies a minor error, but the tolerance range for 500ml is $\pm$25 ml, also set by clinical research.

Overall, the results are acceptable and proves the correctness of simulator performance.

10.4 Optimized Outputs

Outputs for MDO are shown in the picture 12 below:

Picture 12. Multidisciplinary Design Optimization Outputs

Variable	Old station	Prototype
Height (cm)	60	12.12
Width (cm)	50	9.48
Length (cm)	60	10
Flow rate 10 ml/s	yes	yes
Flow rate 30 ml/s	yes	yes
Flow rate 50 ml/s	yes	in progress
Volume for 10 ml/s	yes	yes
Volume for 30 ml/s	yes	yes
Volume for 50 ml/s	yes	in progress
Total volume (ml)	>1150	>1150

As it can be observed from the picture above, utilization of optimization techniques allowed us to reduce the dimensions of the station dramatically. For the height - almost in 6 times; for the width - almost in 5 times; for the length - in 6 times.

This means that the volume of the water tank was also reduced almost in ~120 times. Such dramatic dimensions reduction will allow to install the signal simulator not only in the allocated spot at Production floor, but on any table at any department (i.e. R&D, Service, Nonconforming), which will save the time and money for the company.

11.0 Conclusion (D),

12.0 Appendix 1 - MATLAB Script 1

```
clear

% Initial Condition And Assumption
P2old=10; P2new=20;
W=10;
L=1;
error=P2new-P2old;
% Constraints
V=1150;
Q=10;
Cv=1;
% Cost Function
H=((Q/Cv)^2+P2old)/9.8;

% Jacobi (initialization of variables)
i=2;
a=[];
```

```
b=[];
c=[];
d=[];
e=[];
a(1)=H;
b(1)=L;
c(1)=P2new;
d(1)=1;
e(1)=1;
while error>1*10^-7
        L=V/(H*W);
        H=Hopt(L);% Function call for parametric optimization of H, using a linear differential corrector eqn.
        P2new=H*9.81-(Q/Cv)^2;
        H=((Q/Cv)^2+P2new)/9.8;
        a(i)=H;
        b(i)=L;
        c(i)=P2new;
        d(i)=i;
        error=abs(c(i-1)-c(i))  % Absolute difference between new and previous pressure values
        e(i)=error;
        i=i+1
end
```

```
% Plotting Variables
figure(1);
plot(d,a)
hold on
plot(d,c)
title('Plot of the Out Pressure Optimization')
xlabel('No. of Iterations')
ylabel('Variable Convergence')
title('Optimization of cost Function')
legend('Height (cm)','P2 (N/cm^2)')

%Export to Excel
R1(1:i-1,1)=a;
R2(1:i-1,1)=b;
R3(1:i-1,1)=c;
R4(1:i-1,1)=d;
R5(1:i-1,1)=e;
columnname={'Iteration','Height','Length','Outlet_Pressure','Pressure_Error'}
R=table(R4,R1,R2,R3,R5,'VariableNames',columnname)
writetable(R,'optimization.xlsx');
type 'optimization.xlsx';
```

```
function H1=Hopt(L) %This function takes the length, L as an input variable
syms H
```

```
W=1150/(L*H);
D_LW=L*(1+(W)^2/L^2)^(1/2); % Diagonal distance along LxW plane
Df=diff(D_LW,H)==0.6; % For our assumption of W=10. setting Df=0.6 will yield optimum value of V>1150
sol=vpa(solve(Df,H));
sol1=sol(imag(sol)==0);
H1=sol1(sol1>=0);
end
```

12.1 Appendix 2 - MATLAB Script 2

```
clear
% Initial Condition And Assumption
P2old=10; P2new=20;
%W=(3:1:30);
L=1;
error=P2new-P2old;
% Constraints
V=1150;
Q=10;
Cv=1;
% Cost Function
H=((Q/Cv)^2+P2old)/9.8;
% Jacobi method (initialization of variables)
i=1;
ai=[];
bi=[];
ci=[];
di=[];
ei=[];
ai(1)=H;
bi(1)=L;
ci(1)=P2new;
di(1)=1;
ei(1)=1;
aj=[];
bj=[];
cj=[];
dj=[];
ej=[];
aj(1)=H;
bj(1)=L;
cj(1)=P2new;
dj(1)=1;
ej(1)=1;
j=2;
im=1;
am=[];
cm=[];
em=[];
am(1)=H;
cm(1)=P2new;
em(1)=error;
for W=(10:1:15)
        am(im)=H;
        cm(im)=P2new;
        em(im)=error;
        H=((Q/Cv)^2+P2old)/9.8;
        L=V/(H*W);
```

```
        k=solv_k(L,W) % a function to compute the gradient for the differential equation.
        i=2;
        error=1;

        while error>1*10^-5
        P2old=H*9.81-(Q/Cv)^2;
        L=V/(H*W);
        H=Hopt(L,k);
        P2new=H*9.81-(Q/Cv)^2;
        P2new=abs(P2new)
        H=((Q/Cv)^2+P2new)/9.8;
        ai(i)=H;
        bi(i)=L;
        ci(i)=P2new;
           di(i)=i;
        aj(j)=H;
        bj(j)=L;
        cj(j)=P2new;
        dj(j)=j;
        error=abs(cj(j-1)-cj(j)); % Absolute difference between new and previous pressure values
        ei(i)=error;
        ej(j)=error;
        ik=i-1;
        i=i+1;
        j=1+j;
        end
        im=im+1;
end
```

```
%% Plotting Variables
figure(1);
scatter(di,ai)
hold on
scatter(di,ci)
title('Plot of the Out Pressure Optimization')
xlabel('No. of Iterations')
ylabel('convergence')
title('Optimization of cost Function')
legend('Height (cm)','Valve Pressure Outlet (f/Kg^2)')
figure(2);
plot(dj,aj)
xlabel('No. of Iterations')
ylabel('Length (cm)')
title('Plot of the Length Optimization')
figure(3);
plot(cm,em)
title('Plot of the Out Pressure Optimization')
xlabel('P2')
ylabel('error')
```

```
figure(4);
plot(am,cm)
title('Response Surface of the Valve Pressure Outlet')
xlabel('Height (H) cm')
ylabel('Valve Pressure Outlet (P2) N/cm^2')
figure(5);
plot(am,em)
title('Plot of the Out Pressure Optimization')
xlabel('Height')
ylabel('error')
```

```
% Export to Excel
R1(1:j-1,1)=aj;
R2(1:j-1,1)=bj;
R3(1:j-1,1)=cj;
R4(1:j-1,1)=dj;
R5(1:j-1,1)=ej;
columnname={'Iteration','Height','Length','Outlet_Pressure','Pressure_Error'};
R=table(R4,R1,R2,R3,R5,'VariableNames',columnname);
writetable(R,'optimization2.xlsx');
type 'optimization2.xlsx';
```

```
function H=Hopt(L,k)
syms H
W1=1150/(L*H);
D_LW=L*(1+(W1)^2/L^2)^(1/2); % Diagonal distance along LxW plane
D_H=diff(D_LW,H);
Df=D_H==k;
sol=vpa(solve(Df,H));
sol=sol(imag(sol)==0);
H=sol(sol>=0);
end


function ksol=solv_k(L,W)
syms H
W1=1150/(L*H);
D_LW=L*(1+(W1)^2/L^2)^(1/2); % Diagonal distance along LxW plane
D_H=diff(D_LW,H);
k=[0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0]; % the slope must range from zero to one
i=0;
V1=1;
V=[];
V(1)=V1;
for i=1:1:10
```

```
        syms H
        Dfk=D_H==k(i);
        sol=vpa(solve(Dfk,H));
        sol=sol(imag(sol)==0);
        H=sol(sol>=0);
        V1=W*L*H;
        V(i)=V1;
end
Vd=V-1150;
[~, I]=min(abs(Vd))
ksol=k(I);
V=V(I)
end
```

12.2 Appendix 3 - C++ Code for Arduino Mega 2560

```
long timer1=50000; //timer for drain, equal to 50s
long timer2=15000; //timer for 10ml/s, equal to 15s
int timer3=17000; //timer for 30ml/s valve, equal to 17s
void setup() {
  // put your setup code here, to run once:
pinMode(8, INPUT);
pinMode(3, INPUT);
pinMode(10, INPUT);
pinMode(11, INPUT);

pinMode(9, OUTPUT);
pinMode(2, OUTPUT);
pinMode(4, OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  // setting up the button pins to HIGH level (5V), so the output pins get triggered when pin state changes to LOW (0V)
 digitalWrite(3, HIGH);
 digitalWrite(8, HIGH);
 digitalWrite(10, HIGH);

// setting up the 4n25 pins LOW (0V), so they can trigger the MOSFETS transition, when changing their state to HIGH (5V)
 digitalWrite(2,LOW);
 digitalWrite(9,LOW);
 digitalWrite(4, LOW);

 if (digitalRead(10) ==  LOW) // drain, set to 50 sec with 3 valves open
{
 digitalWrite(2,HIGH);
 digitalWrite(4,HIGH);
 digitalWrite(9,HIGH);
 delay(timer1);
 digitalWrite(2,LOW);
 digitalWrite(4,LOW);
 digitalWrite(9,LOW);
}

if (digitalRead(8) ==  LOW) // 10 ml/s flow rate (fifth 4n25)
{
 digitalWrite(9,HIGH);
 delay(timer2);
 digitalWrite(9,LOW);
}


if (digitalRead(3) ==  LOW) // 30ml/s flow rate, set to17s with 3 valves open

{

 digitalWrite(2,HIGH);
 digitalWrite(4,HIGH);
 digitalWrite(9,HIGH);
 delay(timer3);
 digitalWrite(2,LOW);
 digitalWrite(4,LOW);
 digitalWrite(9,LOW);
}

 }
```

13.0 Works cited (O,D)