Robotic Mechanisms – design analysis and applications; Soft Robotics - human robot cooperation in industry

Muhie Al Haimus, 230388549

April 2024

Abstract. From soft robotic fish to prosthetic arms: soft robotics covers a wide range of research areas and is an extremely promising and cutting-edge field. This is why it is critical to have a strong foundation on the fundamentals of developing, manufacturing, and understanding rigid mechanisms first. Then how these key principles can be applied to soft robotic actuators both when fabricating them and operating them. The results show that rigid mechanisms can be used to augment the functionality of soft robotic actuators for more specialised scenarios but would be unsuitable for the more general cases in industry today. The results also showed that the performance of soft robotic actuator produced in this study had an extremely high force-to-weight ratio.

1 Introduction

The purpose of case study 3 (comprised of two components) is to demonstrate how one can create and control a fully-fledged soft robotic actuator. The advantages of soft robots are clear; they have a very good force-to-weight and power-to-weight ratio, compared traditional ridged equivalents [6]. However, as soft robotics is such a new and quickly evolving field, there are no standards for design of soft robots [5]. Which makes it hard to integrate these products into industry.
The first component was to investigate how a Grashof system (specifically the four-bar linkage) affected the flow rate of a kink valve by varying the input angle of the driver, and hence subjecting the kink valve to axial compression (made out of an elastomer tube). Using the dimensions used in the real-world example a "Digital twin" was created using Computer Aided Design software (CAD) specifically Autodesk Fusion 360, this was used to verify the results from the real-world experiment through the use of a motion study that these CAD programs provide, in their toolkits.
In the next component of the case study, another experiment was conducted. A fibre augmented soft robotic actuator was fabricated and tested. The pneumatic actuator was driven with positive pressure's inside of the mechanism (which allowed the mechanism to apply more or less force on an object depending on the pressures applied.)

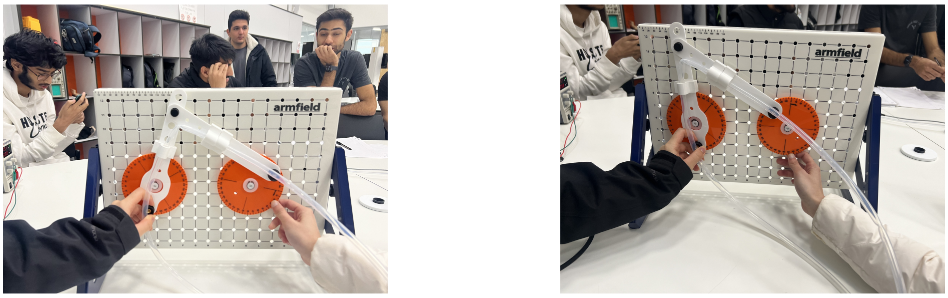
Keywords— soft robotics, fibre augmented soft robotic actuator, axial compression, elastomer tube, Grashof system.

2 Materials and Methods

2.1 Component 1: design analysis and application of a four bar linkage

A mechanism is a set of components connected together in such a way as to produce a desired motion[8]. A linkage with four bars (and four links) is the simplest possible mechanism [1]. A structure consists of three or fewer fully-connected links. A structure cannot move. [1] The classification for a Grashof linkage, is governed by: if at-least one link (usually the shortest link) can complete one full revolution. A non-Grashof four-bar or mechanism is one at which a system cannot complete one full revolution. From the results of the "digital twin" it confirmed that a Grashof linkage was used in our experiments as the kink valve set-up restricted the movement of the mechanism. The equations that govern this and a further explanation of why the mechanism is Gashof can be found in the appendix.
The kink valve is made out of a highly-compressible material. Critical compression, is where the tube snaps into a kink and blocks the flow of the air into tube [4] as seen in figure 1 (b). One can leverage this as a digital on-and-off valve, this is particularly useful with a soft robotic gripper that has a known actuation pressure. A kinked tube blocks fluid flow in the tube up to a certain pressure [4] (breakthrough pressure.) Since the elastomer readily undergoes large and reversible deformation, the kink valve can close and open repeatedly without the tube plastically deforming [4].
Characterising the performance of a kink valve can be broken down into four key factors: 1) Compression at which the tube kinks 2) Compression at which the kink opens 3) Breakthrough pressure at which air starts to leak 4) Pressure at which air leaks at a constant rate.

The Armfield kit (used in the first case study) includes the parts and defines the specification (in the reference manual (Figure 6 (a)) [3]) for creating the four-bar linkage set-up as seen in figure 1. The Armfield kit does not, however, include an additional kink valve set-up. This functionality was created by using the following: an elastomeric tube, air compressor, flow rate sensor, DC power supply and 3d printed clips to mount the tube to the four-bar a detailed diagram for this can be found in the appendix.



(a) (b)

Figure 1: Shows the four-bar linkage kink valve experimental setup, (a) unkinked where a maximum positive air pressure is measured on pressure sensor. (b) where kink valve is fully kinked and zero air pressure is measured on pressure sensor.

2.2 Component 2: Fabricating the Soft Robotic Actuator prototype

The prototype design consists of a silicone bladder wrapped with inextensible reinforcements. The inner bladder acts like any typical balloon [7]; when pneumatic positive-pressure is induced it tries to expand in all directions. Augmenting the bladder with inextensible fibers constrains it from expanding radially; instead, it can only expand in the axial direction. A sheet of inextensible material was also added so that the actuator bends when inflated [7].

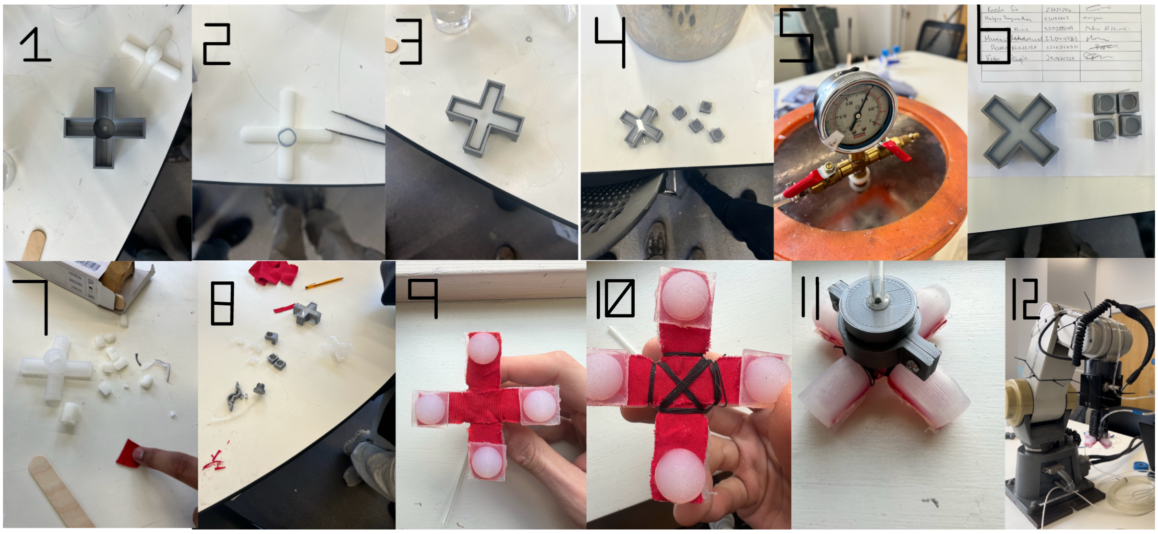


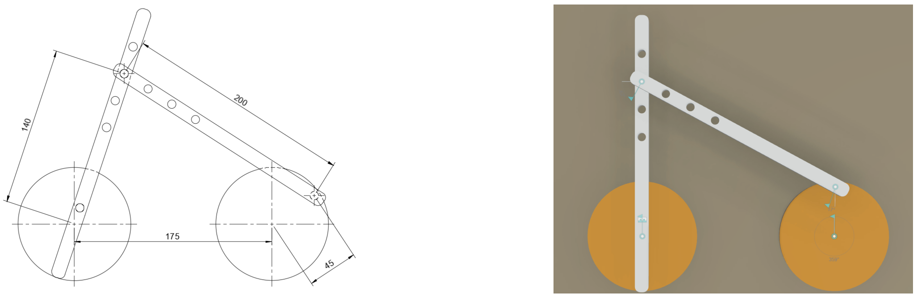
Figure 2: The actuator fabrication stages. 1) 3D printing the Polylactic acid (PLA) moulds which will be used to create the bladder of the actuator and produce the wax insert. In this case the wax insert was already pre-fabricated. This meant that only a small amount of a torus shaped piece of blu-tac 2) was needed to act as a gasket, and was placed on the top of the insert. This insert was then placed into the PLA mould: the wax insert supports the inner bladder, keeping it rigid when curing. The material of the bladder was produced using two parts of DragonSkin 10 (A silicone rubber): A and B. These two parts had to be mixed in a ratio of exactly one-to-one. To produce one actuator, sixteen grams of each part were used in this case to produce a gripper comprised of four bladders, where one bladder is 36mm x 16mm x 16mm (a more detailed diagram of this can be found in the appendix). An additional three grams of thinner was used to speed the vacuuming process and make the solution more viscous. Before pouring the mixture into the mould all the air bubbles need to be removed, so a vacuum chamber is required. Place the mixture into the vacuum chamber and set it to -0.8 atmospheres. When all of the air bubbles are removed from the mixture, (you can see this by shining light on the vacuum chamber) depressurise the chamber. Pour the mixture into the mould 4) and then place the mould back into the 5) vacuum chamber once more. Again, once all of the bubbles are removed, depressurise the chamber 6) and leave the mould to harden for 4-5 hours. 7) Once silicone has cured remove the wax insert from bladder. 8)/9) Cut out two fabric strips (inextensible material) which allow actuator to be bend; using a silicone glue, stick these onto the bladder alongside the four grips on the edges of the gripper 10) Wrap fibres around the gripper in the pattern shown in 11). Add fittings to allow the gripper to be connected to the arm and pressure regulator 12). Attach gripper to arm.

The gripper was tested using a Mitsubishi RV-M1 (Movemaster EX) as seen in figure 2 step 12, this is an industrial grade five axis robotic arm. The gripper itself was connected to a pressure regulator specifically the ITV2050-212BL4, and a pressure sensor was then used (NPA500B005D) to monitor the pressure electronically using an Arduino UNO. The robotic arm was controlled over Serial configured with a baud rate of 9600, 7 data bits, 2 stop bits, even parity bits and RTS/CTS flow control. A modified scale was then used to measure the amount of force applied to the scales.

The performance of the fabricated gripper was surprising. The whole procuress took around 15 hours from start to finish. It has a very low actuation pressure (10kPa) which meant that even a person's lungs could actuate the gripper. The actuator could hold up to 643g (6.31N) as seen in figure 5 before failure this was at a pressure of 13.7kPa which again is not a very high pressure. These results agree with the claims referenced in the introduction, as this soft robotic actuator had both a high force-to-weight ratio of 20 times considering the fabricated actuator only weighed 31g.

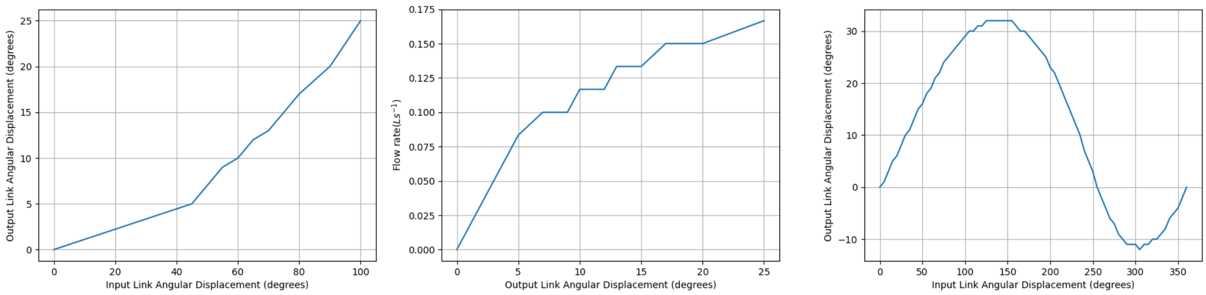
3 Results

3.1 Component 1: design analysis and application of a four bar linkage



(a) (b)

Figure 3: (a) Shows the dimensions in millimeters (mm) for the experimental setup in Figure 1 and "digital twin". (b) Shows the "digital twin" created using Fusion 360 and has been used to conduct the motion study.



(a) (b) (c)

Figure 4: The results: (a) The graph shows the relationship between the driver input angular displacement and how this affects the displacement of the output link on the four bar linkage both in degrees. (b) Shows how the variation of the output angle in degrees against the flow rate of pressure sensor in litres per second (c) Shows the displacement output driver in degrees against the input driver in degrees of the "digital twin" through one full rotation of 360 degrees (motion analysis results.)

3.2 Component 2: Fabricating Robotic Actuator prototype

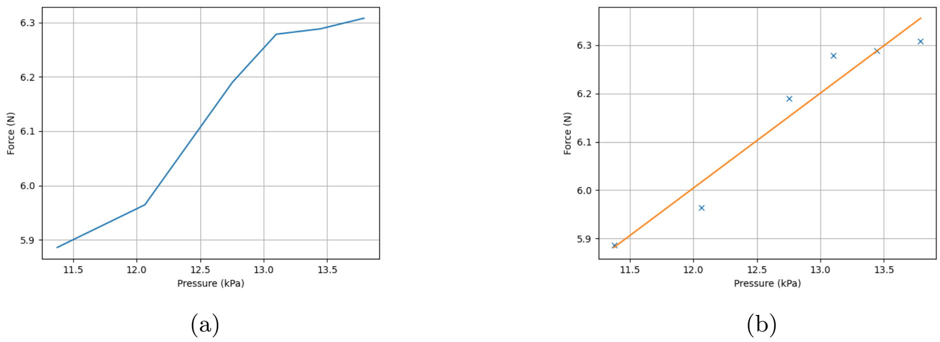


Figure 5: The experimental results of the soft robotic gripper, (a) Shows force applied to the scales in newtons against the pressure flowing through the bladder of the gripper in kilo pascal (kPa). 13.7 kPa shows the breaking pressure of the actuator. (b) Shows a linear regression of the graph in (a).

[1]

4 Discussion

In the first component for the study I was unable to test all of the features of the kink valve including the breakthrough pressure and the pressure at which the air leaks. furthermore, experimentally, I was unable to verify if the four bar linkage was indeed Grashof with the physical mechanism due to the movement being restricted by the kink valve as shown in figure 4 (a) which limited the driver to just 100 degrees. So the "digital twin" was necessary to classify this linkage. From the results in figure 4 (c) a sinusoidal graph was produced but with some sort of "forward bias", that is somewhat similar to the class 2 Grashof linkage in the companion website [8] which had a fully symmetrical sinusoidal shaped graph.

The kink valve can be used to actuate the soft robotic gripper. However the main limitation is that it can only really be used as a digital on-off shown in 4 switch rather than being able to fine tune the amounts of force applied to objects. In an industrial setting this would not be adequate for example Ocado technology is a automation company that deals with automating the packing of groceries, here they have to handle tens of thousands of products each with varying weights and sizes, thus it is essential to be able to have fine control of the gripper.

In general, the amount of pressure applied to the soft robotic actuator did increase the amount force applied to the scales. The problem lies within the amount of manual fabrication required to produce this gripper. When augmenting the bladder with fibres it was hard to quantify how much fibre was added to each side of each chamber. This meant that at higher pressure's each chamber would not expand proportionally to each other and hence not evenly distribute the force along the object. This effect could be mitigated by using circular geometry for the pressure chambers and applying a dense, fibre reinforcement [2] rather than just augmenting the outer chamber in this example.

From our flow experiment results (found in the appendix) they clearly show that there was a large discrepancy (at most 36%) between the analogue and digital measurements. Due to the nature of analogue equipment, they are subjected to lots of parallax errors so I believe that the inclined manometer would be unsuitable for future experiments with the gripper. I believe that the digital pressure transducer would have to be used.

5 Conclusions

In this paper, I explored both soft robotics and rigid mechanisms. From this I have discovered the following:

1. Soft robotic actuators can be rapidly prototyped.
2. Soft robots have a high force-to-weight ratio and power-to-weight ratio.
3. A four bar linkage can be implemented to get a desired motion.
4. Kink valves can only be used as digital on-off switch. This means it can be unsuitable for some general automation purposes.

[1]Due to time restraints multiple repeats of both experiments could not be completed to calculate averages and standard deviation.

Bibliography

[1] Eric Constans. *Introduction to mechanism design: with computer applications*. CRC Press, Jul 2018.

[2] Fras Jan and Kaspar Althoefer. *Soft Fiber-Reinforced Pneumatic Actuator Design and Fabrication: Towards Robust, Soft Robotic Systems*, pages 103–114. 06 2019.

[3] Armfield Limited. Experiment manual. https://armfield.co.uk/category/educational-products/ef-series-engineering-fundamentals/ef-mechanisms/, 2015. Accessed: 19/4/2024.

[4] Kai Luo, Philipp Rothemund, George M. Whitesides, and Zhigang Suo. Soft kink valves. *Journal of the Mechanics and Physics of Solids*, 131:230–239, 2019.

[5] François Schmitt, Olivier Piccin, Laurent Barbé, and Bernard Bayle. Soft robots manufacturing: A review. *Front. Robot. AI*, 5:84, July 2018.

[6] Yoel Shapiro, Alon Wolf, and Kosa Gabor. Bi-bellows: Pneumatic bending actuator. *Sensors and Actuators A: Physical*, 167(2):484–494, 2011. Solid-State Sensors, Actuators and Microsystems Workshop.

[7] Soft Robotics Toolkit. Fiber-reinforced actuators. https://softroboticstoolkit.com/book/fiber-reinforced-bending-actuators Accessed: 23/4/2024.

[8] Eric Constans. *Introduction to mechanism design: with computer applications.* https://softroboticstoolkit.com/book/fiber-reinforced-bending-actuators Accessed: 23/4/2024.

6 Appendix

Mathematical explanation of why the four bar in the study meets the Gashof conditions:

$$S + L < P + Q \text{ (1) Grashof}$$

$$S + L > P + Q \text{ (2) Non - Gashof}$$

$$S + L = P + Q \text{ (3) Gashof Special case}$$

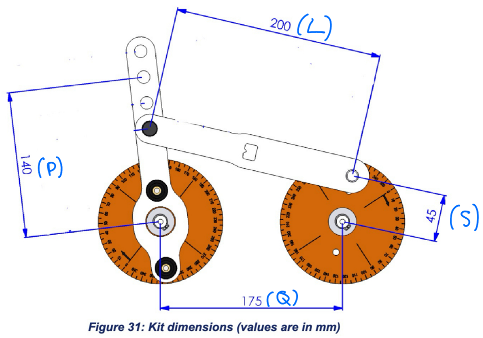


Figure 6: Shows a labelled diagram of what each link represents in the Grashof equation [3].

$$G = 140 + 175 - 45 - 200 (4)$$

$$G = 70, \text{ hence Gashof}$$

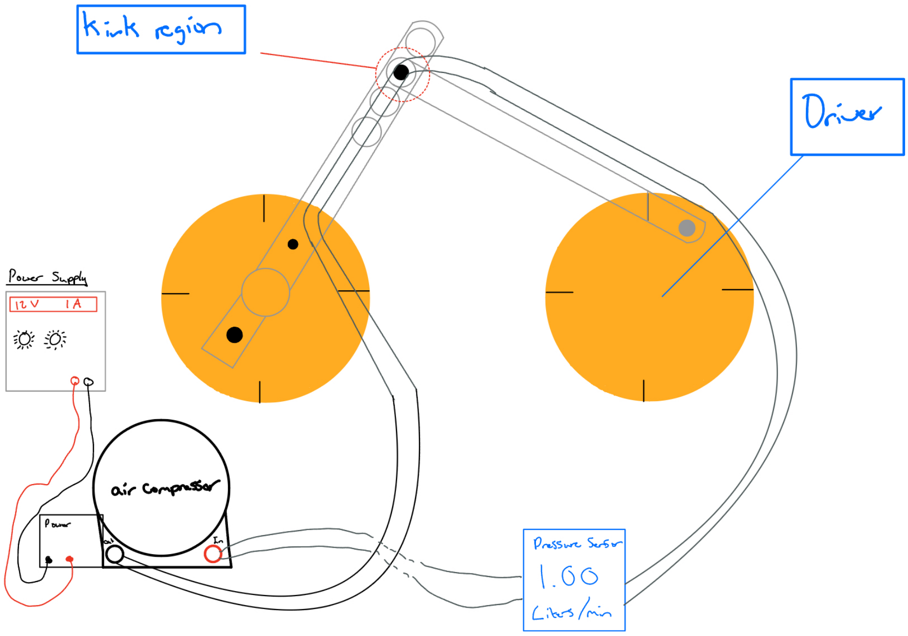


Figure 7: Shows a labelled experimental diagram for case study 3: part one.

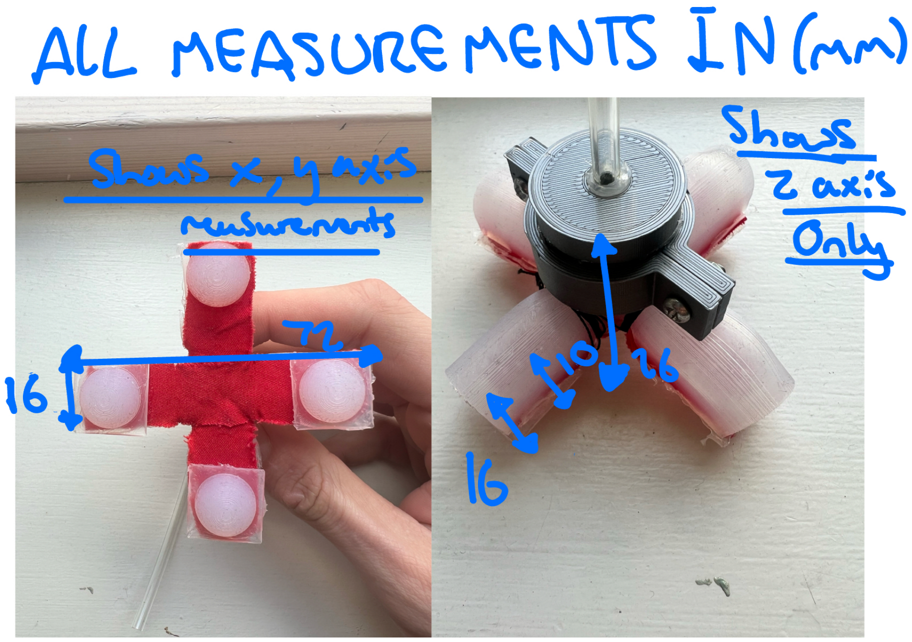


Figure 8: Shows detailed dimensions of gripper fabricated in mm.

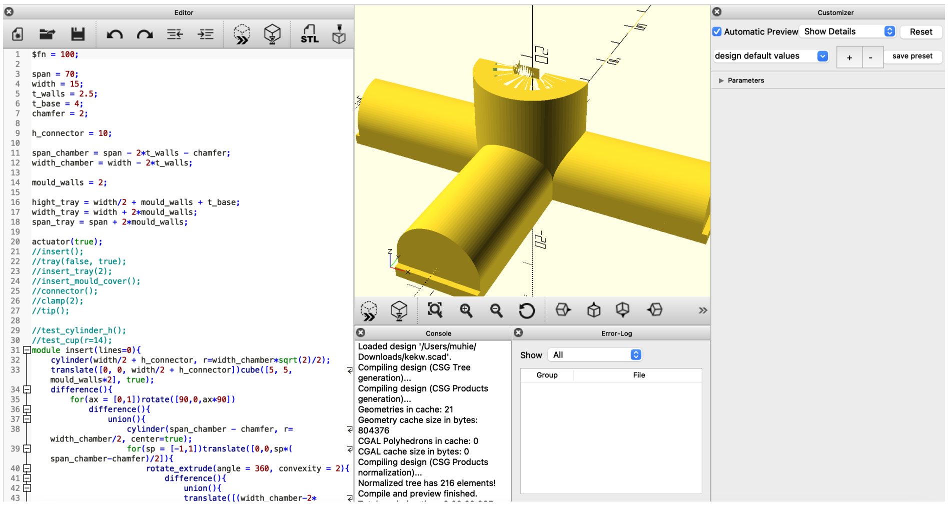


Figure 9: Shows the cad model for the wax insert

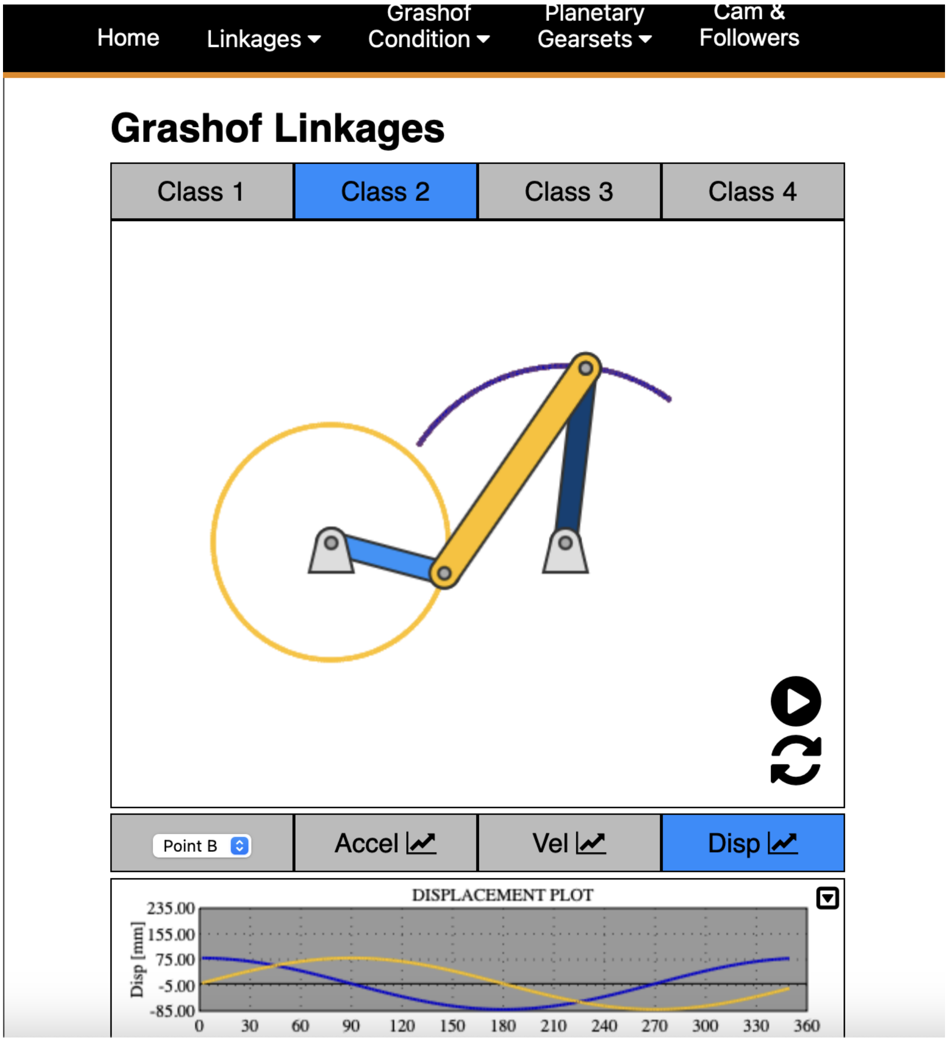


Figure 10: Shows the class two linkage from the companion website

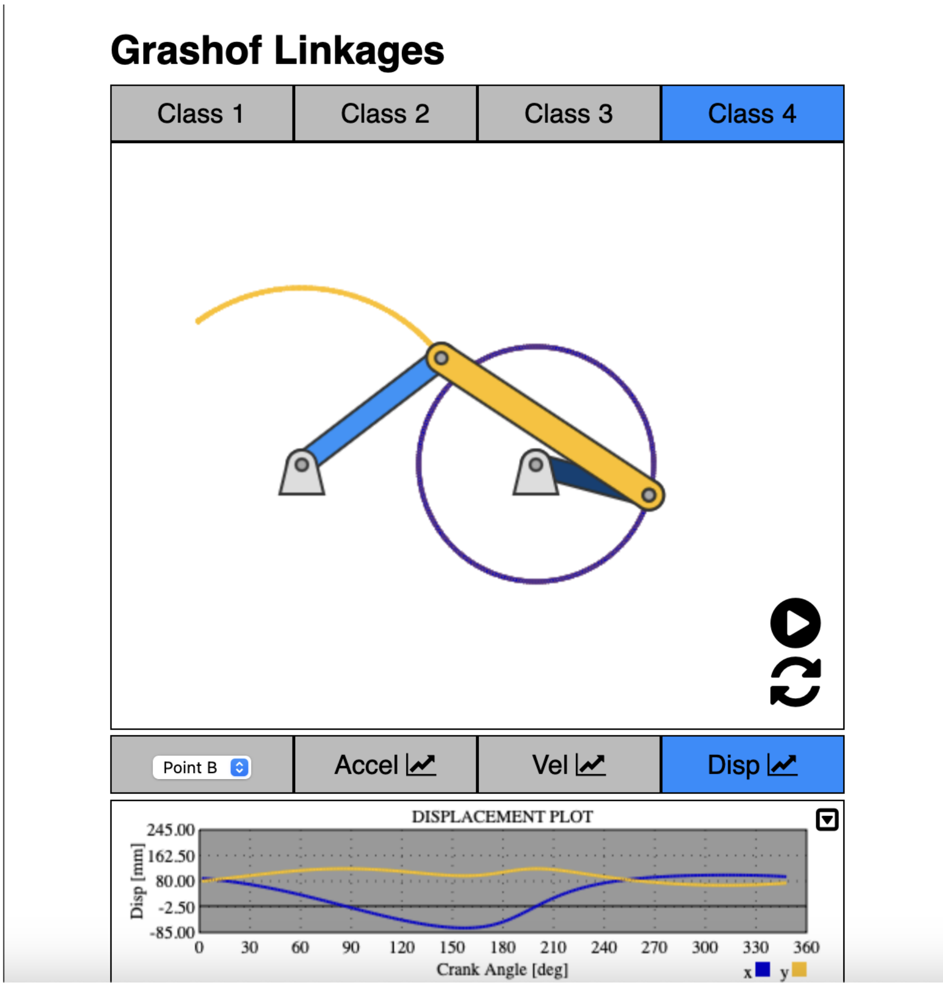


Figure 11: Shows the class four linkage from the companion website

```python
""""Get Data from file"""


import pandas as pd
import matplotlib.pyplot as plt


class Data:
    def __init__(self, file, name, x, y, title, pressure):
        self.file = file
        self.name = name
        self.df = pd.read_csv(file)
        print(self.df)
        self.input_angles = self.df[x]
        self.output_angles = self.df[y]
        self.one_graph = pressure
        if pressure == True:
            self.pressure = self.df["Pressure"] / 60
        self.title = title
        #self.pressures = self.df['Pressure']


    def get_data(self):
        return self.input_angles, self.output_angles,self.pressures

    def get_data_by_column(self, column):
        return self.df[column]

    def get_data_by_index(self, index):
        return self.df.iloc[index]

    def get_data_by_index_column(self, index, column):
        return self.df.iloc[index][column]
    def plot_pressure(self):
            fig, ax = plt.subplots()
            ax.plot(self.input_angles, self.pressure)
            ax.set_xlabel("Output Link Angular Displacement (degrees)")
            ax.set_ylabel("Flow rate($Ls^{-1}$)")
            ax.grid()
            ax.set_title(self.title)
            plt.savefig(self.name)
    def plot_line_chart(self):
        if self.one_graph == True:
            fig, ax = plt.subplots()
            ax.plot(self.output_angles, self.input_angles)
            ax.set_ylabel("Output Link Angular Displacement (degrees)")
            ax.set_xlabel("Input Link Angular Displacement (degrees)")
            ax.grid()
            ax.set_title(self.title)
            plt.savefig(self.name)
        else:
            fig, ax = plt.subplots()
            ax.plot(self.output_angles, self.input_angles)
            ax.set_ylabel("Output Link Angular Displacement (degrees)")
            ax.set_xlabel("Input Link Angular Displacement (degrees)")
            ax.set_title(self.title)
            ax.grid()
            plt.savefig(self.name)



d = Data('real_robo.csv', "four-bar-real", "Left", "Right", "", True)
# d.plot_line_chart()
d.plot_pressure()
```

Figure 12: Shows python code to obtain numerical plots for the case study.

```python
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

class Data:
    def __init__(self, file, name):
        self.file = file
        self.name = name
        self.df = pd.read_csv(file)
        self.pressures = self.df['Pressure']
        self.force = self.df['Force']
    def convert_pressures_to_pascal(self):
        self.pressures = (self.pressures * 6894.76)/1000
    def plot_line_chart(self):
        fig, ax = plt.subplots()
        n, b = np.polyfit(self.pressures, self.force, 1)
        ax.plot(self.pressures, self.force)
        # ax.plot(self.pressures, (self.pressures*n)+b)
        ax.set_ylabel("Force (N)")
        ax.set_xlabel("Pressure (kPa)")
        ax.set_title("") # caption Force Applied to Scales vs Positve Pressure Applied to Mechanism"
        ax.grid()

        plt.savefig(self.name)


d = Data('grip.csv', "gripper-results")
d.convert_pressures_to_pascal()
d.plot_line_chart()
```

Figure 13: Shows python code to obtain numerical plots for the case study.

Student ID	Name	Individual contribution	Comments
230388549	Al Haimus, Muhie	Results, references, and discussion	
230577165	Abdirahman, Ali	Discussion	
220457217	ALHAMDAN, MOHAMMED		Did not make any meaningful contribution/ not present
230870941	ALOTAIBI, ZIAD JM		Did not make any meaningful contribution/ not present
230870631	ALWAJEEH, ABDULAZIZ AH		Did not make any meaningful contribution/ not present

4

Figure 14: Shows individual contributions of the flow experiment attachment (page 4) (PLEASE MAKE NOTE OF THIS FIRST) Attachment below

1 Results

Converting to SI units: All the pressure readings were converted from either mmH2O or mmWG to Pa, using equation (1).

$$p = \rho g h \ (1)$$

Where h is the value from measured from the device in Pa, ρ is the density of the fluid, which was 1000 kg/m^3 for the pressure transducer (mmH2O) and 1880 kg/m^3 for inclined manometer (mmWG).

$$\rho = \frac{p}{RT} \ (2)$$

The atmospheric pressure was 103 kPa, and the atmospheric temperature was 20°C (291K). In equation (2), p is the pressure in Pa, T is the temperature in K and R is the specific gas constant, which for air is 287.05 J/kg K [1], substituting these values into (2) gives a density of 1.233 kg/m^3.

1.1 Flow rate calculations for Venturi tube

Using the conservation of mass (Venturi flow equation) and using Bernoulli's find the velocity at a v_2 and then one can find the volumetric flowrate Q in m^3/s

Using the conservation of mass:

$$A_1 v_1 = A_2 v_2 \ (3)$$

Which can be written as:

$$v_2 = \frac{A_1 v_1}{A_2} (4)$$

Now from Bernoulli's equation rearrange for v_1:

$$P_1 + \frac{1}{2}\rho v_1^2 = P_2 + \frac{1}{2}\rho v_2^2 (5)$$

note: as we taking pressure readings from the same heights (this is how the Venturi operates) the $\rho g z_{1/2}$ terms go to zero

Then subsitute in v_2 into (5):

$$P_1 + \frac{1}{2}\rho v_1^2 = P_2 + \frac{1}{2}\rho \left(\frac{A_1}{A_2} v_1\right)^2 \ (6)$$

Which then can be written as when solving for v_1:

$$v_1 = \sqrt{\frac{2(P_2 - P_1)}{\rho\left(\frac{A_1}{A_2}\right)^2 - 1}} (7)$$

Since $Q = A_1 v_1$ this leads to:

$$Q = A_1 \sqrt{\frac{2}{\rho} \cdot \frac{p_1 - p_2}{(\frac{A_1}{A_2})^2 - 1}} (8)$$

1.2 Flow rate calculations for Pitot tube

First, a velocity profile for each specific speed setting (high, medium, low) was required to find the flow rate reading for the pitot tube. The velocity of the fluid varies in the tube from minum at the pipe wall to a maximum at the centreline of the tube, so the pressure difference is was measured at 2mm increments along the radius of the tube, hence why a profile is required.

Velocity at a given increment (v(r)) can be found using:

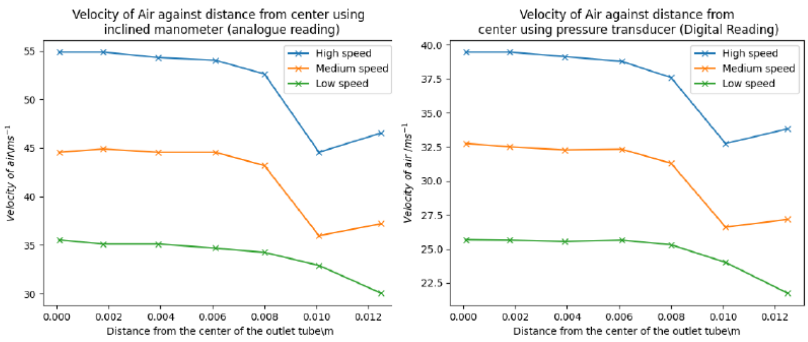
$$v(r) = \sqrt{\frac{2(p_2 - p_1)}{\rho}} (9) \ [2]$$

where $p_2 - p_1$ is the pressure difference measured by the pitot tube and ρ is the density of fluid (1.233) Next the flow rate through the tube needed to be calculated using equation (3) (from the lab flow experiment lab sheet.) A graph of $V(r)r$ against r can be plot; where r is the distance from the center line of the tube at.

1

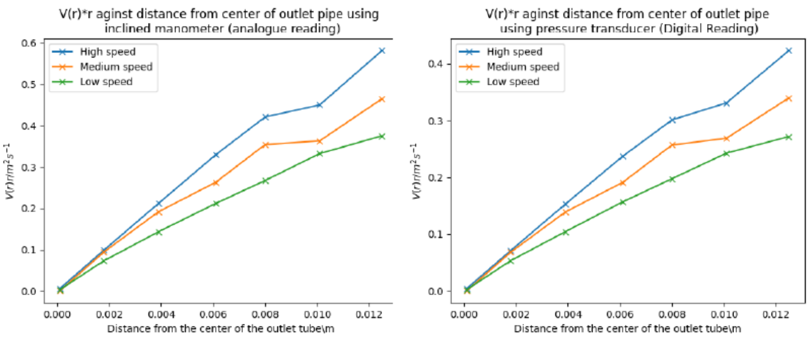
Figure 15: Shows the flow experiment attachment page 1

Finally this graph can be numerically integrated using trapezium rule to aproximate the flow rate, this is again given in the lab sheet with equation (5) this gives you the flow rate for the outlet tube using the pitot probe.



(a) (b)

Figure 1: The graph (a) shows the velocity in ms^{-1} at a given point along the tube, starting from the center of the tube for the inclined manometer readings. The graph (b) shows the velocity in ms^{-1} at a given point along the tube, starting from the center of the tube for the presssure transducer readings



(a) (b)

Figure 2: The graph (a) shows the velocity at a given point V(r) multiplyed by r in m^2s^{-1} at a given point along the tube, starting from the center of the tube for the inclined manometer readings. The graph (b) shows the velocity a given point V(r) multiplyed by r in m^2s^{-1} at a given point along the tube, starting from the center of the tube for the presssure transducer readings

2

Figure 16: Shows the flow experiment attachment page 2

	Venturi Flow Rate/ m^3s^{-1}		Pitot Probe Flow Rate/m^3s^{-1}		Percentage Difference/%	
	IM	PT	IM	PT	IM	PT
High	0.0351	0.0243	0.0244	0.0176	35.9663866	31.9809069
Medium	0.0261	0.0172	0.0199	0.0144	26.9565217	17.721519
Low	0.0194	0.0116	0.0163	0.0119	17.3669468	2.55319149

Table 1: Show the final flow rates calculated using the steps discussed above and shows percentage differences between the IM (Venturi and Pitot) flow rates and the PT (Venturi and Pitot) flow rates

2 Discussion

The results suggest that the flow experiment was flawed: considering the fact that at all times the conservation of mass (A fundemental physical law) must conserved between the inclined manometer (IM) and pressure transducer (PT) through both the Ventri pipe and Pitot tube. The results suggest an almost (on average) 42% difference between the IM and PT readings and an average of a 22% difference between the Ventri and Pitot tube readings between the exact same measuring tool. All of these readings should be the same or atleast very close (less than 5%), however, to prove this fundemental law.

One main source of error could be a parallax error. The manometer may not have been observed perfectly from eye-level. Furthermore, when reading the anolouge, inclined manometer, since it contains a minescus this may not have been observed correctly between readings, causing a large random error.

The lack of higher resolution in apparatus such as the inclined manometer left the results subject to a higher value of uncertainty ± 0.1mm.

Another factor which could have caused these fluctuations in the results obtained could have been the high turbulence. As the flow of air was at a high velocity, it created high turbulence and small vibrations in apparatus such as the pitot tube. This then meant that the results from the inclined manometer would have been skewed. Other areas where flaws could have occurred is the use of the trapezium rule, which is an estimation rather than an exact result. However, this could have been made more accurate if more readings were taken with smaller increasing increments.

Lastly, we did not take repeat readings at different speeds to calculate an average, which then would decrease the percentage uncertainty of the overall results.

3 Conclusion

This experiment was conducted to understand the different methods of measuring the flow rate of air through a system, and then if these methodologies produced the same flow rate due to the use of the fundemental conserveration of mass law. However, as Table 1 shows, these methods provided flow rates which were no where near the same, this means that the way in which we measured the flow rates in the system must have been truely flawed, otherwise, that would mean that the law of conservation of mass no longer holds true, which is impossible.

3

Figure 17: Shows the flow experiment attachment page 3

Bibliography

[1] The Engineering ToolBox (2004). Universal and Individual Gas Constants. https://www.engineeringtoolbox.com/individual-universal-gas-constant-d$_5$88.*html* *Accessed* : 23/4/2024.

[2] Department of Energy Sciences, LTH. MMV211 Fluid Mechanics https://www.lth.se/fileadmin/ht/Kurser/MMV211/Lab2a-PM-English.pdf Accessed: 23/4/2024.

5

Figure 18: Shows the flow experiment attachment page 5

```python
"""Converting pilot tube values into velocities """
""" We need to get the reference pressure """

import pandas as pd
import math
import matplotlib.pyplot as plt

def get_velocity_profile(data, heights, density_of_liquid,  pt):
    # reference_pressure = reference_pressure * 10**3 # convert to Pa
    velocity_profile_array = []
    print(data)
    for i in range(0, len(data)):
        if pt == True:
            pa = data[i] * 9.80665
        if pt == False:
            pa = data[i] * 9.80665 * 1.88
        # print(pa)
        v_1 = math.sqrt((2*(abs(pa))/density_of_liquid))
        print(v_1)
        velocity_profile_array.append(v_1)
    high_speed =  velocity_profile_array[0:7]
    medium_speed = velocity_profile_array[7:14]
    low_speed = velocity_profile_array[14:21]
    return high_speed, medium_speed, low_speed

def create_line_chart(heights, speeds, label, title):
        fig, ax = plt.subplots()
        speed_settings = ["High speed", "Medium speed", "Low speed"]
        number_of_speeds = int(len(heights)/7)
        for i in range(0, number_of_speeds):
            ax.plot(height[0:7], speeds[0][i],label=speed_settings[i], marker = 'x')
        ax.set_ylabel(label)
        ax.set_xlabel("Distance from the center of the outlet tube\m")
        ax.set_title(title)
        plt.legend()
        plt.savefig(title)

def v_r_times_r(heights, speeds):
    int_array = []
    print(speeds)
    for i in range(0, len(height)):
         int_array.append(height[i] * speeds[i])

    high_speed =  int_array[0:7]
    medium_speed = int_array[7:14]
    low_speed = int_array[14:21]
    return high_speed, medium_speed, low_speed


"""A method that gets the radial profile of the pilot tube at different heights"""
def get_radial_profile(number_of_data_points, high_speed, high_speed_heights, medium_speed, medium_speed_heights, low_speed, low_speed_heights):
    high_profile = 0
    medium_profile = 0
    low_profile = 0
    print(high_speed)
    for i in range(0,6):
        diff_height_high = high_speed_heights[i] - high_speed_heights[i+1]
        diff_height_med = medium_speed_heights[i] - medium_speed_heights[i+1]
        diff_height_low = low_speed_heights[i] - low_speed_heights[i+1]
        high_profile = high_profile + 0.5*(high_speed[i+1] + high_speed[i]) * diff_height_high
        medium_profile = medium_profile + 0.5*(medium_speed[i+1] + medium_speed[i]) * diff_height_med
        low_profile = low_profile + 0.5*(low_speed[i+1] + low_speed[i]) * diff_height_low

    print(2*math.pi * high_profile)
    print(2*math.pi * medium_profile)
    print(2*math.pi * low_profile)


df = pd.read_csv('data.csv')

print(df.columns)
height = df["Pitot Probe Pressure difference"]
manometer_pilot_mmWG = df["Unnamed: 4"]
tranducer_pilot_mmh2o = df["Unnamed: 5"]
reference_pressure = 103 # pressure in kPa
density_of_liquid = 1.225

r = get_velocity_profile(tranducer_pilot_mmh2o, height, density_of_liquid, True)
print(r)
for i in range(0, len(height)):
    height[i] = abs(height[i]*10**-3 - 251.4*10**-3)

create_line_chart(height, [r], "$Velocity\ of \ air\ /ms^{-1}$", "Velocity of Air against distance from\n center using pressure transducer (Digital Reading)")

q = v_r_times_r(height, (r[0] + r[1]+ r[2]))

create_line_chart(height, [q], "$V(r)r/m^{2}s^{-1}$", " V(r)*r aginst distance from center of outlet pipe \n using pressure transducer (Digital Reading)")

get_radial_profile(7, q[0], height[0:7].tolist(), q[1], height[7:14].tolist(), q[2], height[14:21].tolist())



# just checking the data, but seems good!
```

Figure 19: Shows python code to obtain numerical plots for the flow experiment.

```python
import pandas as pd
import math
import statistics


def get_flow_rate(data, pt):
    pressure_diff = []
    for i in range(0,len(data)):
        if pt == True:
            pressure_diff.append((abs(data[i]* 9.80665)))
        if pt == False:
            pressure_diff.append((abs(data[i]* 9.80665 * 1.88)))
        """ conversion value from
        https://www.sensorsone.com/mmh2o-to-pa-conversion-table/
        , date accessed 30/3/24 """


    A_1 = math.pi*((diameter_1*10**-3)/2)**2
    A_2 = math.pi*((diameter_2*10**-3)/2)**2
    ratio_A_1_A_2 = (A_1/A_2)**2

    Q = []
    for i in range(0,len(pressure_diff)):
        pa = pressure_diff[i]
        Q_to_be_appended = A_1*math.sqrt((abs((2/density_of_liquid)*(pa/ratio_A_1_A_2-1))))
        Q.append(Q_to_be_appended)

    return(Q)

def get_avg(array):
    total = 0
    # print(statistics.stdev(array))
    for i in range(0, len(array)):
        total = total + array[i]

    return total/len(array)


df = pd.read_csv('data.csv')
diameter_1 = 108.1 # in mm
diameter_2 = 29.6 # in mm
density_of_liquid = 1.225  # in this case air from same source as Q image
print(df.columns)
manometer_venturi_mmWG = df["Venturi Pressure Difference"]
tranducer_venturi_mmh2o = df["Unnamed: 2"]
# print(manometer_venturi_mmWG)

z = get_flow_rate(tranducer_venturi_mmh2o, True)
f = get_flow_rate(manometer_venturi_mmWG, False)
print("manometer ventri: ", get_avg(f[0:7]), get_avg(f[7:14]), get_avg(f[14:21]))
print("pt ventri: ", get_avg(z[0:7]), get_avg(z[7:14]), get_avg(z[14:21]))
```

Figure 20: Shows python code to obtain numerical plots for the flow experiment.