NTNU
Norwegian University of
Science and Technology

Artificial Lateral Line Sensors

Developing Surface Sensors for Flow
Detection around Adaptive Hydrofoils

Håvard Nitter Vestad

Master of Science in Mechanical Engineering
Submission date: June 2018
Supervisor: Martin Steinert, MTP

Norwegian University of Science and Technology
Department of Mechanical and Industrial Engineering

Abstract

In search for more efficient propulsion through water, many methods for improving on the existing hydrofoil and propeller designs we use today have been suggested, the topics of adaptive hydrofoils and propulsion systems is increasingly popular. A large portion of all world trade happens by sea, and even incremental improvements in propulsive efficiencies can have big economic and environmental impacts. Many researchers have looked to nature to find answers as to how to more efficiently use and manipulate water for propulsion and maneuvering.

In this master's thesis I investigate the mechanisms that enables fish to use flows actively, and efficiently, to improve the capabilities of adaptive hydrofoils through sensory inputs. The thesis is based on the work done in my pre-master's thesis into the effects of Kármán gaiting, where fish use alternating vortices shed from obstructions in flows to generate forward momentum and is a response to the limited capabilities I found in the field of high density, non-intrusive/flow-obstructive, flow sensing at the surface of bodies and hydrofoils, such that correct adaptive movements can be made.

I investigate different ways to sense flows along the skin of hydrofoils suspended in flowing water through prototyping and comparing solutions. Through comparison the piezo resistive effect of carbon fibers mixed with rubber silicone showed superior capabilities in detecting pressure changes as well as its ability to be easily imbedded and cast into flexible hydrofoils. The composite was further investigated in its capabilities in flow detection, through expanding the amount of datapoints with multiplexers to generate bigger pictures of the flow and enabling tendencies and pressure gradients to be seen. A water tunnel was run at different flow velocities of 0.04m/s and 0.1m/s, both free flowing and altered to generate alternating Kármán vortex streets. Two silicone hydrofoils were fitted with 64 sensors each. At 0.1m/s the Kármán flow generated cyclic behaviors in the forward-facing sensors in one of the hydrofoils, that directly corresponded with the observed and calculated shedding frequency. A dynamic neural network was further able to distinguish differences between all flow scenarios in separate data from the same calibrations and setups that were used in the training sets.

The carbon fiber rubber silicone sensor shows similar piezoresistive properties as the popular carbon nanocomposite-silicone sensors, but the ease of manufacturing and low cost of the carbon fiber sensor could open up new possibilities and implementation in areas where such carbon nanocomposite sensors have not yet been deployed.

Sammendrag

I en stadig søken etter forbedringspotensialer og besparelser i den maritime industrien, har mange endringer og innovasjoner innen hydrofoil- og propelldesign blitt foreslått. Noen viktige tema som går igjen er adaptive hydrofoiler og fremdriftssystemer som kan tilpasses strømminger og arbeidshastigheter. Med brorparten av internasjonal transport til sjøs vil selv inkrementelle forbedringer kunne gi store økonomiske og miljømessige vinninger. Mange forskere har begynt å se mot naturen for å finne inspirasjon til måter å bedre utnytte vann og strømninger i maritime fremdrifts- og manøvreringssystemer.

I denne masteroppgaven undersøker jeg mekanismene som gjør det mulig for fisk å bruke strømmer aktivt og effektivt, for å forbedre de adaptive evnene til adaptive hydrofoiler gjennom sensorikk og strømmingsfølere. Avhandlingen er basert på arbeidet i min prosjektoppgave, hvor jeg undersøkte *Kármán gaiting,* effekten der fisk bruker vekslende hvirvler fra hindringer i strømninger for å generere fremdriftskraft, og er ment som en videre undersøkelse av de begrensningene jeg oppdaget da jeg prøvde å gjenskape effekten mekanisk: Mangelen i metoder for effektiv implementering av ikke-obstruerende, strømningsfølende sensorer med høy datatetthet på overflaten av legemer og hydrofoiler, slik at et godt grunnlag for adaptive aktueringer og handlinger kan legges.

Gjennom prosjektet blir ulike sensoriske måter å hente data fra strømninger undersøkt, prototypet og sammenlignet. Gjennom sammenligning og prototyper viste de piezoresistive egenskapene til karbonfiber blandet med silikon seg å være egnet for formålet; både den ekstreme sensitiviteten og muligheten til å støpe den elastiske og fleksible sensoren inn i hydrofoiler av silikon gjorde løsningen attraktiv for videre undersøkelser. For å teste materialet, utvidet jeg mengden mulige datapunkt i matrisen med multiplexere for å danne større trykkbilder så tendenser og trykkgradienter ble mer visuelle. En vanntunnel ble kjørt ved forskjellige strømningshastigheter på 0,04m/s og 0,1m/s, både frittflytende og obstruert for å generere vekslende Kármán-strøminger. To hydrofoiler av silikon ble utstyrt med 64 sensorer hver. Ved 0.1m/s genererte Kármán-strømmen sykliske data i de fremre sensorene i et av hydrofoilene, som direkte korresponderte med den observerte og beregnede frekvensen til strømningen. Et dynamisk kunstig nevralt nettverk kunne videre skille forskjeller mellom resterende strømningsscenarier i separat data fra de samme kalibreringene og oppsettene som ble brukt under opptrening av nettverket.

Karbonfiber-gummisilikon-sensoren viser piezoresistive egenskaper som kan ligne på egenskapene til de populære karbon-nanokompositt-silikonsensorene, men den enkle produksjonen og de lave kostnadene for karbonfibersensoren kan åpne opp ytterligere muligheter i områder hvor nanokompositt sensorer ennå ikke blir brukt.

Acknowledgements

This master’s thesis is the result of a challenge given by ProtoMore. Thanks to the research community at TrollLabs NTNU, Achim Gerstenberg and Martin Steinert for providing guidance and an environment in which ideas grow. A big thanks to Tanguy Simon for inspiration and insightful tips and discussions on sensor arrays. Finally; thanks to family and friends for support through this labor-intensive adventure.

Table of contents

LIST OF FIGURES

LIST OF ABREVIATONS

AoA	Angle of Attack
CFRSS	Carbon Fiber Rubber Silicone Sensor
CAD	Computer-Aided Design
CNC	Computer Numerical Control
DIY	Do It Yourself
EM	Electro Magnetic
FDM	Fused Deposition Modeling
FFT	Fast Fourier Transformation
IDE	Integrated Development Environment
LED	Light Emitting Diode
MEMS	Microelectromechanical Systems
OFS	Optical Flow Sensors
PCB	Printed Circuit Board
PSD	Power Spectral Density
PVC	Polyvinyl Chloride
PWM	Pulse Witdh Modulation
RF	Radio Frequency
RMS	Root Mean Square
Vcc	IC Power-supply pin
Vol.%	Volumetric Percentage
Wt.%	Percentage by Weight

1 Introduction

Water is one of the most important resources we have, not only is it vital to human life as we know it, but we have become dependent on water also in other aspects of our existence; we use water as a mode of transport, as well as source of energy. Improving our capabilities in utilizing this resource to its fullest through better propellers, hydrofoils or ship hull designs can show great impact. In an everlasting search for better, more sustainable or more energy efficient solutions, nature has become an increasingly popular source of inspiration.

1.1 Life in fluids and efficient swimmers

In his book Life in moving fluids, Vogel (1983) gives an introduction to how lifeforms have adapted to life in fluids in motion and use the laws that govern their existence to their advantage, from a technical flow perspective. How, through evolution and millions of years of development, organisms have come to overcome fluid dynamic challenges such as drag, lift, and thrust that from a mathematical perspective is often considered advanced.

Fish and aquatic animals have evolved into efficient swimming machines far superior in energy efficiency than any water propulsion alternative we have today. With 90% of all world trade by sea ("Shipping and World Trade," 2018) the economic impact of increased efficiency in water propulsion could be enormous, yet very little research has been done to adapt principals of fishlike propulsion into technical solutions (Triantafyllou & Triantafyllou, 1995).

In the aquatic swimming domain, a vast variation of solutions has emerged to counter different problems faced in different environments. While most long distance swimmers have adapted the *thunniform swimming mode* (Sfakiotakis, Lane, & Davies, 1999) other water based animals have found other ways that they can use fluid behaviors to their advantage for energy preservation. A notable effect that can often be seen in fish is their swarm behavior. Fish swim in formation. Formation swimming, as well as a swarm-decision making tool, can aid in energy preservation, and can also be adapted by mechanical fish swimming with live fish (Marras & Porfiri, 2012). In my pre-master's project thesis I focused on another energy preservation effect, called *Kármán Gaiting*, where fish use flows to stay stationary with minimal muscular input (Liao, Beal, Lauder, & Triantafyllou, 2003a, 2003b).

Although the effects might not seem directly applicable to maritime solutions, the lessons of the functionalities of the solutions might aid in making better and more efficient propulsion solutions in the future. Through this master's project I will use the principals of biomimicry as a foundation to gain new information about methods for improved propulsion in water.

1.2 Biomimicry and bio-inspired innovation

Innovating in *new product development* (NPD) can be difficult. Finding solutions that work is not necessarily hard, with hundreds of years of modern technology we as humans have already developed a quite impressive portfolio of solutions to most of the problems that we meet daily. With so many established solutions, seeing beyond them and solve problems in new ways can often feel blinding. How can we ignore the easy way out and find something new from nothing? Falling back into the same old track with existing solutions and methods is a solution bias most product developers know all too well, and multiple methods have been introduced to overcome it. One of these methods is called *Biomimicry*.

Even with extensive research and development, many of nature's solutions to problems still vastly surpass our current technological capabilities. Nature utilizes 3.5-3.8billion years of evolutionary problem solving, testing, and verification (Doolittle, 2000). Biomimicry is the art of using this vast problem-solving period to human advantage, taking solutions from nature and bringing the functionality of the solution into development of new products.

Biomimicry, like most buzzwords is often over-used and misunderstood. Perhaps some of the confusions origin form the many similar methodologies and name-given fields out there where biology plays the role of inspiration for design. Biomimetics, bionics, biogenesis and biomimicry are just some of the names that are thrown about (Hwang et al., 2015; Vogel, 2000). When the term biomimicry was fist coined by Benyus in 1997 the definition that came with it was that biomimicry is the act of taking a biological model and applying it to a human problem (Benyus, 2008). The notion is not that inspiration is gained from observation or that a copy is made, but that a problem-solving model is taken and applied to another, human, problem. The book, originally released in 1997 talks of biomimicry as a "*new technology",* suggesting that the act of biomimicry has only existed for 20 years. However, we also recognize that mimicking nature has taken place even before it got a name, and that the "new technology" might just be a classification or trial of methodological approach to something that has already been going on for a much longer period of time. If we go as far back as to the *primitive man*, the basis of technological improvement was often imitation of nature's solutions, clothes were made from fur and they learned to water crops to simulate rain, but even in modern history there are countless examples of innovation that fall well within the definition of biomimicry before it was defined. A popular example is the innovation of *Velcro*® (Gonzales, 2011) and how George de Mestrals was able to investigate the bur seeds, and through observation understand that the seeds use tiny hooks to attach to fur.

The increased ability to observe nature through microscopes and test equipment is often credited as one of the reasons for the increased interest in biomimicry these recent years (Gonzales, 2011). Some famous examples that are directly linked to our ability to observe how nature solved the problem is the *Lotus's hydrophobic self-cleansing abilities* (Barthlott & Neinhuis, 1997) and the geckos wall-sticking abilities (Kim et al., 2008) that have both been mimicked and used in products that benefited from the observed effects. A more detailed explanation of biomimicry and examples can be found in my project thesis in **appendix D (chapter 1).**

1.3 Working with biomimicry as an engineer

Biomimicry is the bridge between biology and engineering, two fields that seemingly don't have too much in common: Biologists observer something that exists, while the engineer by contrast creates what is to come.

Working with biomimicry as an engineer can be challenging, the inherit knowledge about biological systems and organisms might not be there. But biomimicry is not really about copying nature to an exact. Rather it is about observing an effect, recreating the effect and then specialize the effect so that it can be applied to a use case. Although the *lotus effect* and *gecko feet* discoveries highly rely on a massive foundation of research and observation into the solution models, there are also examples of biomimicry where approaches closer to pure engineering has been taken and beneficial effects from nature was found through prototyping solutions seen in nature to confirm their effect. One such example is *Mercedes-Benzes* concept car "*Bionic*" from 2005 (Buehler & Patel, 2015), which was able to get a drag coefficient of only 0.19 (Phenix, 2005) through imitating the body shape of box fish. It was later proved that the basis on which they made their aerodynamic assumptions was somewhat flawed (Farina & Summers, 2015), yet they were able to get good results through just pure imitation of a biological solution model. Another example is the research gone into the leading edges of humpback whale flippers (Fish, Weber, Murray, & Howle, 2011). The flippers have sinusoidal shape along the leading edge, and applying similar shapes to wings have shown an increase in lift of 8% ("Humpback whales inspire next generation wind turbine technology," 2009).

These projects show that working with biomimicry can be a work intensive process. To reach a final solution you fist need to find a solution in nature, understand it, recreate it and finally specialise the solution to fit your new use case. Though the two latter projects show that also without too much understanding success can be had through prototyping of solutions. As an engineering student, my energy is probably better spent prototyping and exploring solutions than through theoretical observation in a field in which I have little prior knowledge, and throughout this project I will use existing biological knowledge and try to adapt, and prototype solutions based on this.

1.4 The challenge from ProtoMore

The challenge and original inspiration for both my project thesis and this master's project came from *ProtoMore, Kunskapsparken AS*, a start-up incubator and workshop lab located in Molde at the western coast of Norway. ProtoMore asked for a project showcasing how biomimicry can be used efficiently as a tool for new idea generation in product development and prototyping. Although not stated explicitly I felt the need to form the project in a way that would fit with ProtoMore's demographic, users and owners. ProtoMore is owned by local industries, a total of 36, which are mostly maritime. Using a biomimetic approach to investigate how aquatic animals and organisms use fluid dynamics to become efficient swimming machines, seemed like an open, yet relevant challenge that would reflect ProtoMore's missions. Through the project thesis I worked with this open problem, first in a divergent manner investigating the solution space, from sensors and actuations to surface materials and skin, and in the end, I settled on the concept of *Kármán Gaiting*.

Kármán gaiting is an effect often seen in fish species swimming up rivers. The fish will sometimes, depending on the rivers flow, stay seemingly stationary behind rocks and other obstacles in the flow. To the untrained eye one might think that the fish rests in a low flow zone generated in the wake of the object, however the truth is somewhat more complex.

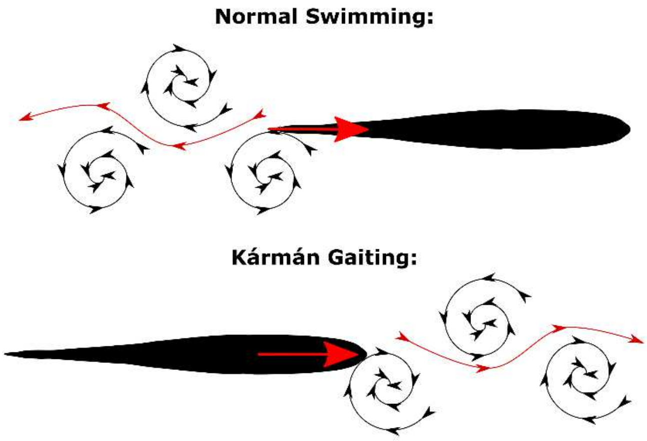


Figure 1.1 - A sketch of the vortices in normal swimming mode and Kármán streets

Behind objects in streams with the correct *Reynolds number*, a pattern of vortex shedding occurs. This alternating shedding of vortices is called a *Kármán Street.* When the fish synchronies its motion with this shedding pattern, much like a sailboat can sail towards the wind at an angle, the fish is able to generate forward motion even when the sum of flow around the fish is going backwards. To visualize the effect, one can think of it as the opposite of swimming. When swimming or flapping a foil in water, a vortex street is generated behind the object, a Kármán street looks very similar, but the rotation of the vortices according to their placement (right/left) is in the opposite direction. Somehow fish can use this to gain forward momentum.

The theories as to how this effect actually works are not very conclusive. Some papers suggest that the vortices generated by the object in the stream actuate the fish so that with low muscular input a positive force forward can still be maintained (Liao et al., 2003a). While in another paper it is suggested that the fish itself changes its camber and *angle of attack* (AoA) back and forth to get a foil shape that would generate lift in its relative flow field and thus sail forward (Liao et al., 2003b).

My original fascination with this effect came from the fact that not only can positive forward momentum be generated in live fish with almost no muscular input, but as shown by James Liao (LiaoLab, 2012) even deceased fish in the right flow conditions can gain a positive momentum forward through Kármán gaiting. Even though the effect has undergone a lot of research and the effect has been reproduced mechanically (Gopalkrishnan, Triantafyllou, Triantafyllou, & Barrett, 1994; Salumäe & Kruusmaa, 2013; Streitlien, Triantafyllou, & Triantafyllou, 1996) there seems to be few conclusions as to what is needed to use Kármán gait in a human perspective, why it actually works and how to implement it into a technology case. For the cases listed, controlled flows and actuations were used to get some energy gain from a controlled vortex street source, but could one also do it simpler?

In my project thesis I tried to tackle the Kármán gaiting effect from a prototyping perspective, making different models and testing them in test environment to see how well they were able to show the effect. My models were both passive and active, yet none of them were able to conclusively show a proper gain from being suspended in a vortex street.

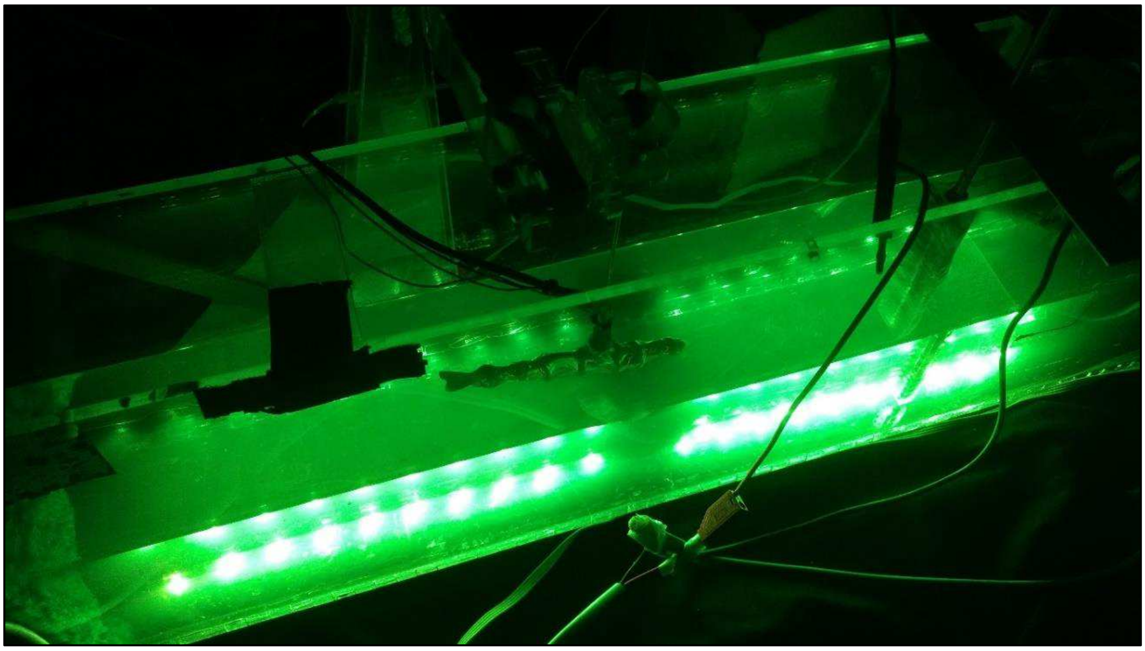


Figure 1.2 - One of my robot fish suspended in a Kármán street trying to actively adapt

1.5 Pre-master's project results and new problem definition

When moving into the domain of actively trying to swim and actuate Kármán-gaiting behavior it became obvious that the problem was not actuating the movement itself or constructing a foil that can adapt to swimming motions. Solutions exist that solve these problems elegantly (Salumäe & Kruusmaa, 2013; Streitlien et al., 1996). The true problem when trying to actively Kármán gait was the synchronization to the flow. With no sensory input of how the flow changes around the foil, determining the shedding frequency and the current flow scenario was extremely difficult. To counter this in my project thesis, found in **appendix D**, I used a manual controller to change the swimming frequency and off-set on the go, and hydrogen bubble visualization of the flow to visually inspect the frequency and state of the flow to determining whether an increase or decrease in speed was necessary for synchronization to happen. Keeping a swimming pattern that correctly corresponded with the Kármán streets shedding frequency over time proved difficult through this method. Even when the frequency was near perfect, over time small errors in the frequency estimate would offset the swimming pattern from the Kármán street rapidly. Another problem was that even in the relatively controlled environment for flow generation, the flow would sometimes behave unpredictable and get offsets in the shedding frequency on its own. Not knowing and feeling the flow around the foil seemed to be the bottleneck, as estimating and calculating the flow was not adequate even in a low speed, controlled environment.

Fish sense flows with *lateral lines* of *hair cells* (Flock & Wersäll, 1962). Many of the research projects that work with Kármán gaiting have focused on sensory input from the flow, and being able to correctly estimate the flow for correct swimming patterns and behaviors (Chambers et al., 2014; Salumäe & Kruusmaa, 2013). Although, some of the solutions work great in the test environments, the solutions are often bulky, give little actual feedback or require expensive specialized equipment to be produced, making the scalability of the solutions minimal. Yet being able to feel and adapt to flow is an essential part of propulsive behaviors in fish and aquatic animals.

Through this project I will try to tackle flow sensing with focus on flexibility, cost and ease of manufacturing to reach a solution for artificial lateral lines, with greater scalability and ability to be implemented in different projects. For a solution to be viable it needs to be able to sense changes in flows around the foil without noticeably changing the flow themselves.

1.5.1 Adaptive wings and hydrofoils

By adaptive hydrofoils and wings it is meant foil profiles that can alter their shape during use such as the chamber and preferred angle of attack. For the sake of this project, I will consider the end adaptive hydrofoil to be one along the lines of the adaptive fish made towards the end of my project thesis; with a flexing skeleton inside a flexible and stretchable skin. The main intention of such a foil would be to recreate thunniform swimming motions to generate thrust and Kármán gaiting as well as other swimming maneuvers. This means that the artificial lateral line needs to be able to be placed on a soft and stretchable surface.

1.6 Problem description

Through using biomimicry from a new product development perspective, I will investigate the possibilities of improving the basis for decision-making for adaptive hydrofoils. The way fish interact with flows and water through lateral lines will form the basis for mimicry and used to investigate sensory input for better understanding of local flows around bodies in flowing water.

The end goal is to produce or alter sensors in such a way, that they can fit into an adaptive hydrofoil that can move and adapt in swimming-like motions. The solution should not significantly alter the flow of water that it is measuring and should not significantly constrain the movement of the foil. The solutions should be able to fit in a streamlined hydrofoil profile, and not hinder the installation of actuation on the inside of the wing. The sensor should be able to differentiate vortices and changing flows around the hydrofoil, so that an adaptive foil can use the observed flow data to make correct movements in accordance with the observed flows for increased propulsive efficiency.

2 Method

Biomimicry as stated earlier is perhaps best described as a combination of biology and engineering. Through this project I will explore how to use this connection from an engineering perspective, to further build knowledge and learnings by investigating existing information and understandings through product developments tools. One of the most powerful learning tools in product development is prototyping. In this project, I will actively use prototyping and prototyping techniques to verify ideas and compare them as I explore the solutions space to reach a good solution to the flow sensing problem.

2.1 Prototyping

Prototyping is used in wide variety of fields and for different purposes, when and why to prototype is up to the individual. Yet using prototypes actively can be important to unearth unknown problems and features of your project.

According to Ulrich and Eppinger (2012) prototypes are tools for: communication, integration, milestones and **learning**. As stated, I will mainly focus on the learnings that can be gained through prototyping and use prototypes to explore possibilities and problems in a practical manner. Humans are excellent problem solvers, yet we are often not able to fully comprehend problems. By prototyping we can put our product or aspects of a solution into perspective and unearth unknown problems in ways that enables us to solve them. Likewise, for this specific project, I already know that there are a multitude of solutions that can be deployed for flow sensing, but by prototyping the solutions I will be able to make a better assessment of which aspects that are actually important in a final solution to the problem, and find problems with the existing solutions that my solution needs to overcome.

With a wide solution space to explore the most fitting innovation model for this project might be the wayfaring/hunter-gatherer model as explained by Steinert and Leifer (2012). Where probing with multiple prototypes is done to find and learn the direction that is best to move forward. To gain a good decision basis, a high degree of probing should be done early in the initial phases so that a more specific direction for the project can be found.

2.2 Front loading and set based designs

When a process *is Front loaded* a higher amount of resources and energy is invested early in the process. This is typically deployed in lean product development, and is said to mitigate risk by gathering more information and knowledge early in the process so that right decisions can be made later (Thomke & Fujimoto, 2000). A way to do this is through *set based design*. A set based design model focuses on splitting tasks into subsystems of problems and generating multiple alternative solutions for the identified subsystems (Kennedy, Sobek, & Kennedy, 2014). As alternatives are proven to not work satisfactory the set converges to one single solution. A pure set based design approach would of course require tremendous amounts of time and resources, but an adaptation of the principles through high amounts of early simple prototypes of potential solutions and narrowing down to what actually works is a good way to ensure a solution that will work for the specific problem.

When developing the first airplane, the Wright brothers deployed what could be considered a set-based approach to their prototyping. They were able to do this by making a good test environment in which multiple low-investment solutions could be tested and compared quickly. And they have later credited a lot of their success to this wind tunnel setup that allowed for rapid prototyping of solutions (American Institute of Aeronautics & Astronautics (AIAA), 2017). Like in my project thesis I will use my own test equipment so that I am not limited by other testing facilities and a more rapid approach can be taken for testing of multiple prototypes.

2.3 Creating a test environment for rapid testing and adjustments

One of my main learnings from the project thesis was the importance of having a good testing environment when prototyping. Only when you are able to rapidly test and evaluate your prototypes and concepts against one another are you able to determine what is worth pursuing or not. As my master's project and project thesis investigates phenomena in the same aquatic environment, much of the testing equipment is reusable, although the problem scope has slightly changed, and some modifications will be made to accommodate this. In my project thesis I argued, that making your own test equipment is a good way to provide the flexibility needed in a wide solutions space where you don't quite know where you will end up. This flexibility enables me to re-use most of the test equipment and make the modifications that I need to fit the new project, quickly and with low risk.

2.4 Flow and turbulence control

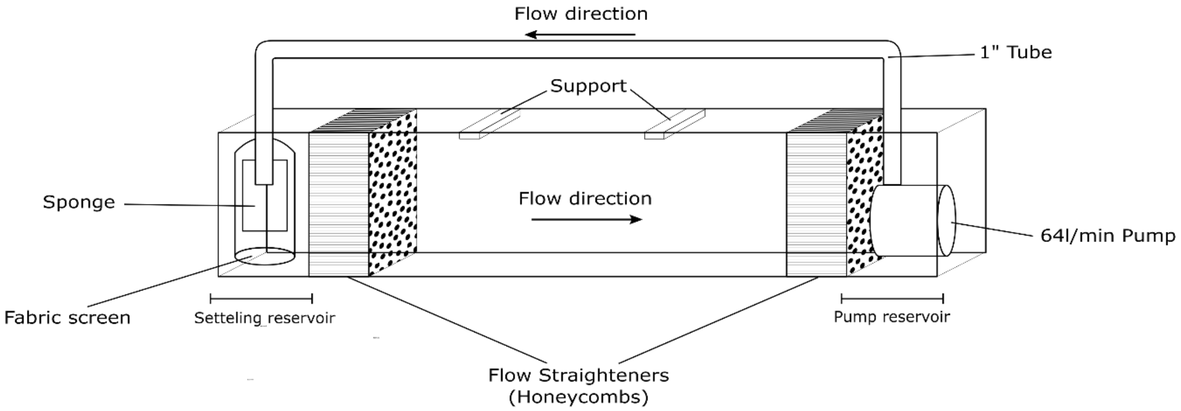


Figure 2.1 - Sketch of water tunnel

In my project thesis, **appendix D (chapter 2),** I went through some extensive steps to create a water tunnel, capable of creating laminar flows and controlled flow conditions. The simple water tunnel consists of a 12*volt* (v) submersible bilge pump, that pumps water from one end of the tank to the other through a 1" tube. The flow from the pump is highly turbulent, so the water tunnel is dependent on flow conditioning to create laminar and transitional flows. Before the water exits the pump-hose and into the settling reservoir, the water passes through two screens. First a sponge and then a cotton screen, removing a lot of the rotational turbulences (Scheiman & Brooks, 1981). A *honeycomb* structure forms the end and beginning of both sides of the tunnel, removing most of the lateral turbulences. The honeycomb structure was made from drinking straws to save time as suggested by Vogel (1983). This combination of screens and honeycombs enables the tunnel to create surprisingly controlled flows as compared to its simplicity. As I had few problems with the tunnel, other than its limited size, I will use the same tunnel in this project. A more detailed description into the making of the tunnel can be found in the project thesis, **appendix D (Chapter 2).**

2.5 Visualization

For visualization of flows I will continue to use the hydrogen bubble generator as described in **appendix D (chapter 2.4)** as it generates the best visualization over time. The hydrogen bubble visualization technique functions through electrolysis of the water. The anode is placed before the flow conditioning as to not influence the flow severely, while the cathode consists of a thin (50um) aluminum wire, so that the bubbles created are small enough to not greatly be influenced by buoyancy through their life-time. In contrast to dye-based indication methods this will not pollute the water over time, and requires less frequent change of the water, while being able to continuously visualize the flow, and not just while dye is being injected. I also found that this method creates a much clearer contrast in pictures and video. The method can’t be used when live wires or electronics are suspended in the water, and the high amounts of electrolyte needed in the water aids in quick corrosion of metallic parts left in the water over time.

2.6 Vibration isolation and dampening

The main concern for this project is to get good readings of the flow activities around bodies in the water. In low speed, controlled flow cases, the forces acting at a single point are very small in amplitude. Trying to sense the small changes sensitively enough, opens up to a lot of problems concerning noise. The most noticeable noise that the different sensors picked up were unrelated vibrations. To reduce the impact of the noise I made changes to the test environment.

2.6.1 Environment

The original placement of the water tunnel was in a room in close proximity to heavy machinery emitting low frequency vibrations. When testing flows with sensitive sensors in this room, activity on the sensors would be very apparent even when no flow was run across them. To reduce the environmental noise, I moved the water tunnel to another room in the workshop. The new room was closer to the workshop machines, lathes, mills, etc., but these emitted much higher frequency noise with lower amplitudes that were less noticeable in the sensors.

2.6.2 Surface contact and material

To further dampen vibrations in the tank and decouple my water tank from the table and room, I cut strips of natural rubber to act as dampening feet between the tank and the table. I also added rubber feet to all structures intended to suspend hydrofoils and specimens into the water to absorb some of the vibrations from the surrounding environment and building (Heckl, 1982). High polymer materials are often used for dampening treatments, and in Cremer, Heckel and Peterssons *“Structure-Borne Sound”* (2005, Chapter 4.5.2) plastics with their loss factors and effective frequency ranges are presented. While pure *Polyvinyl Chloride* (PVC) show superior loss factor for low frequency vibrations, PVC with plasticizer added, as it is normally found, preform much poorer and rubber feet seems like the best and most easily obtainable option for dampening in the low frequency range (Snowdon, 1965). While small surface contact points such as spikes used for speaker systems can be used both to decouple and couple the speakers depending on how they are installed, the effects of reduced or increased surface contact are not as dramatic as material changes (Cremer et al., 2005, Chapter 6), and were not taken into too much consideration, the feet were however kept small to reduce material use and let the equipment sit flat with no wobble.

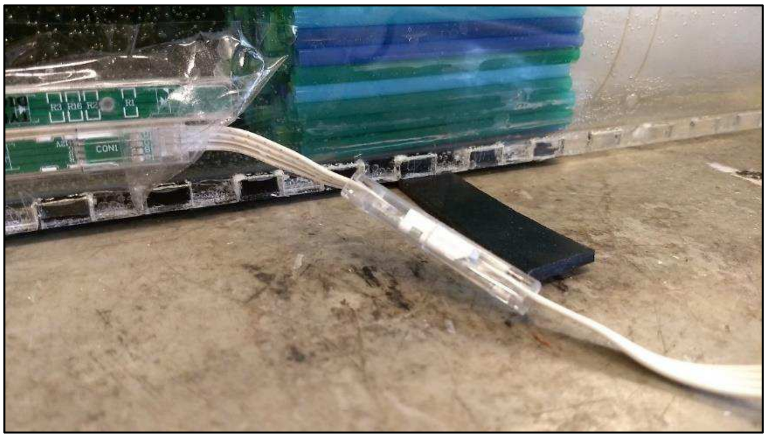


Figure 2.2 - Rubber strip under tank

2.7 Suspensions

Although not strictly necessary for testing all of my prototypes, as in the project thesis having solid rigs to suspend test-samples into the water tunnel would greatly aid in getting more trustworthy readings from the sensors. In the project thesis I made a rigid setup that measured the lateral forces that were applied to it to see the effect flows had on my prototypes. With sensitive sensors a less rigid approach should be taken to further aid dampening of the prototypes. I made two different suspension rigs for two of the foils that I produced, one used rubber bands attached to adjustable nuts to dampen vibrations and give adjustability in roll and yaw direction. The other slightly more elegant solution was constructed as an air-hockey table to make an *air-bearing.* That way the suspension floated on a pocket of air to dampen the rig. This also gave the ability for the rig to move freely back and forth, should any future energy measurements need to be taken. To adjust yaw and roll, spring-loaded screws were used. The pitch and placement of specimens can be controlled by moving the suspension rigs themselves, yaw and roll need to be adjusted in the rigs. The rubber band suspension rig ended up being the rig that I mostly used, due to its simplicity and the fact that the foil attached to this rig gave the best readings.

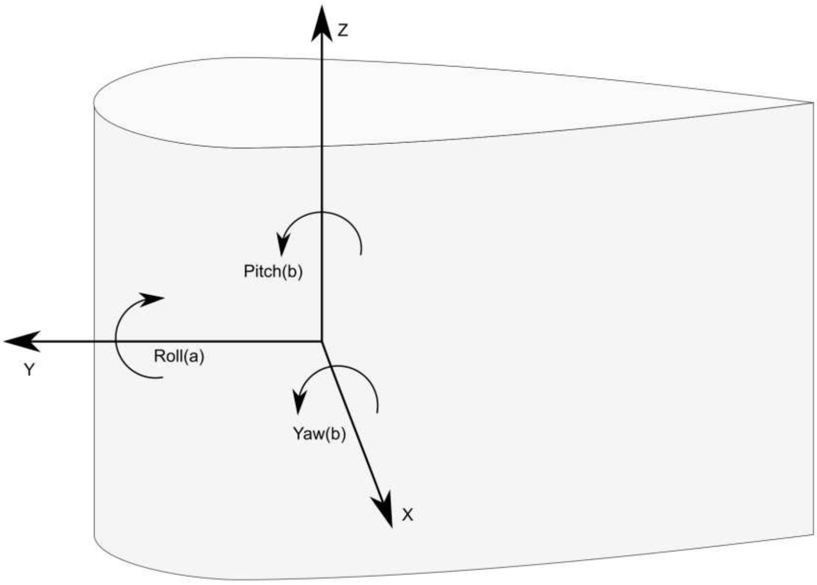


Figure 2.3 - Rotational directions of hydrofoil as seen from the side

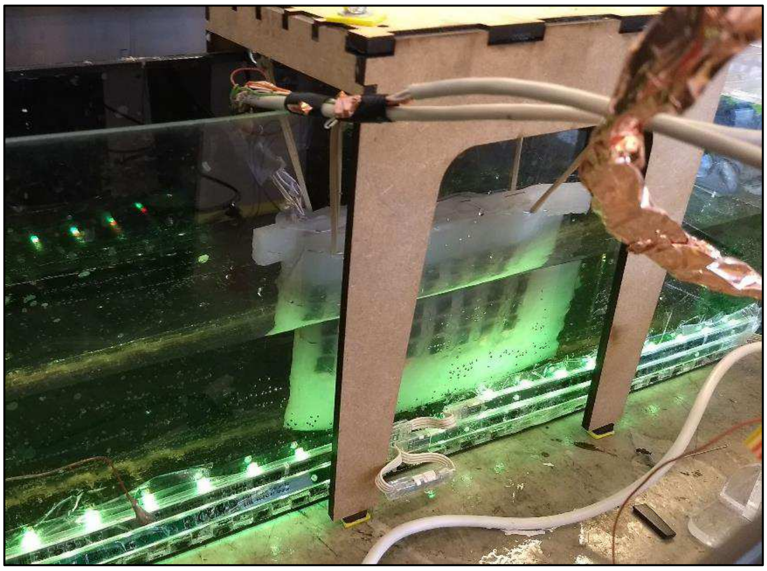


Figure 2.4 - Rubber band suspension rig for sensor foils

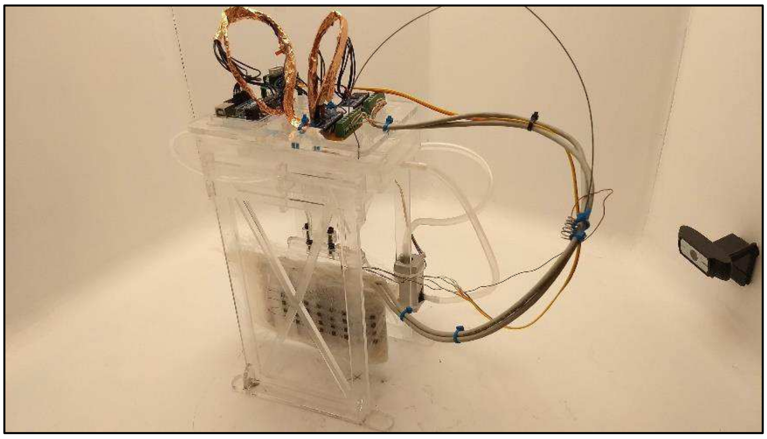


Figure 2.5 - Air bearing suspension rig for sensor foils

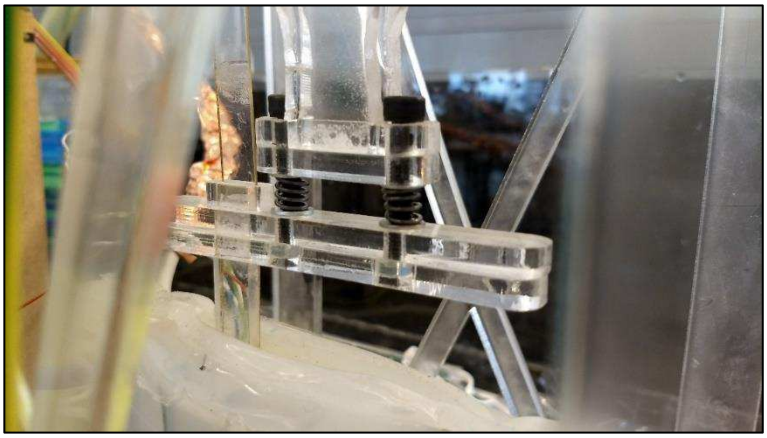


Figure 2.6 - Spring-loaded screws for yaw adjustments

2.8 Electromagnetic noise, capacitance and shielding

Another problem when dealing with analog sensor readings of varying quality is their susceptibility to *electromagnetic* (EM), *radio frequency* (RF) and capacitive noise. Normal procedures to reduce noise on the signal wires is either by using Bells (US244426A, 1881) patented twisting of the wires, or by running a grounded metal foil jacket around the cables to create a *Faraday cage*. To keep noise to a minimum I used grounded copper tape around any signal bearing wire, as seen in Figure 2.7.

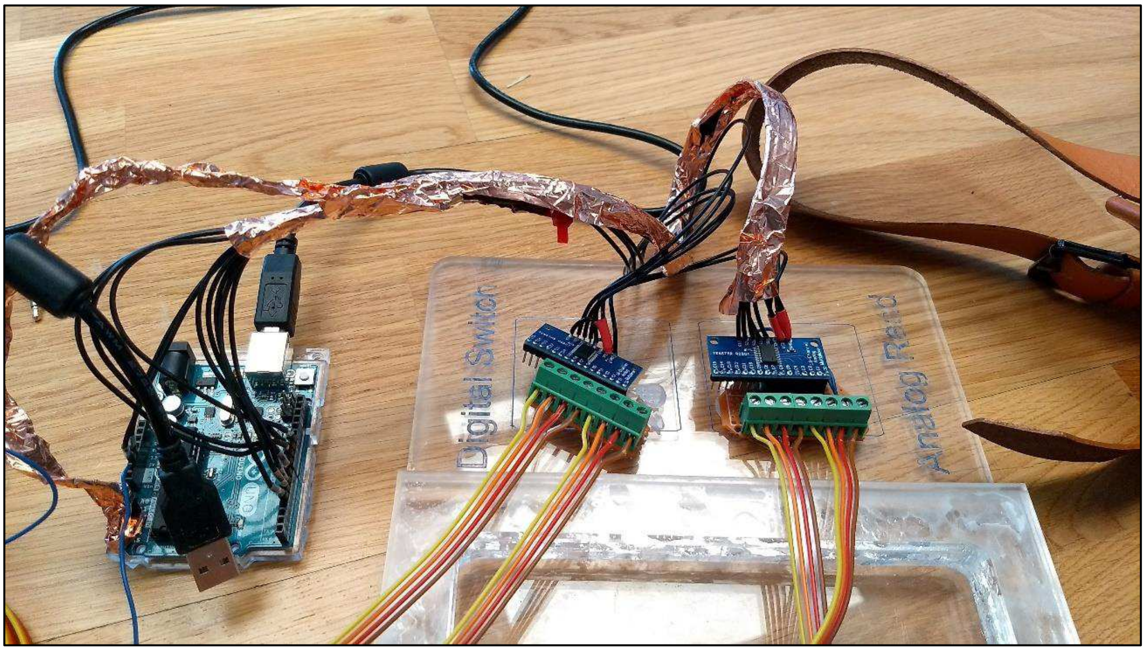


Figure 2.7 - Sensor wires shielded with grounded copper tape

2.9 Flows and benchmarking

In my project thesis I investigated the ability to artificially Kármán gait; to use the alternating flows generated from objects suspended in a steady stream with Reynolds number between 50-1000. To investigate this, I purpose built the water tunnel presented in chapter 2.3, and suspended a half, D-shaped, cylinder with the correct diamete (30mm) in the stream. The flow speed also needs to be kept relatively slow to achieve the wanted Reynolds numbers. Running the tunnel at 5v moves the water at a seemingly laminar at approximately 0.04m/s. The flow is slow enough so that no movement is felt it you stick your finger in the water, but it produces a nice laminar flow that can be predictably manipulated into a Kármán street. The slow velocity of the water means that to be able to feel any flow changes requires an extremely sensitive sensor. The tunnel can also be run at a higher speed. At 12v input to the pump the water speed is about 0.1m/s. The flow at this point is no longer laminar, but rather transitional. An alternating vortex street is still generated when running the tunnel at this speed.

The point of this project is not to purpose build a sensor for these specific flow speeds. However, being able to use known flows that I am able to generate in the water tank would aid in classifying the functionality of sensors even if it is not in itself the hypothesis of success or failure; The water tunnel dimensions and flow speeds were chosen arbitrarily, for ease of manufacturing.

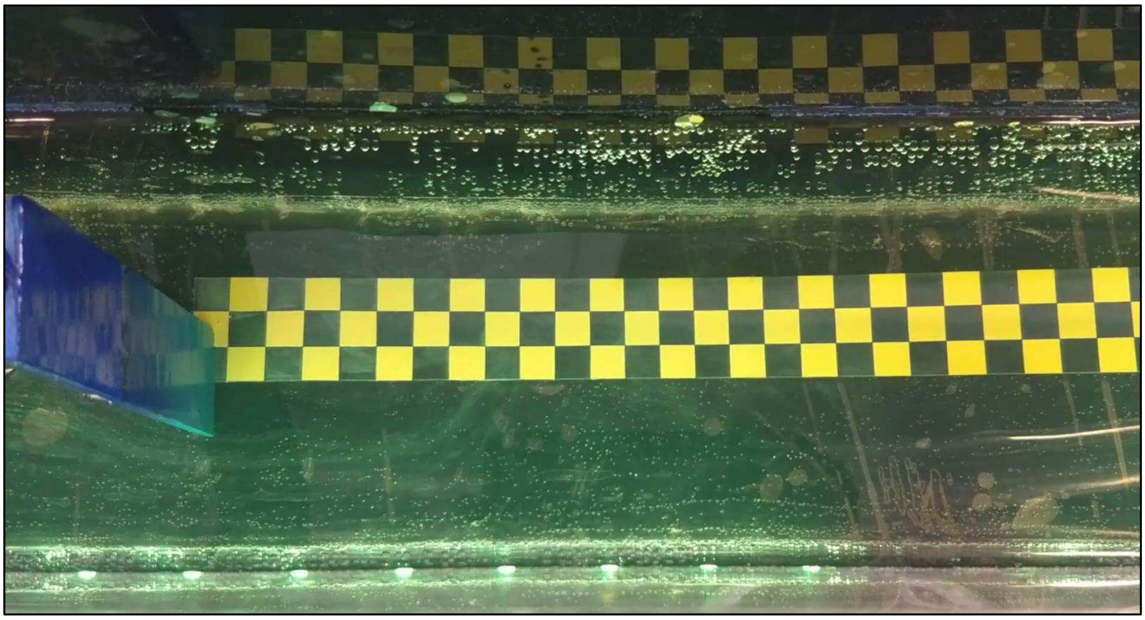


Figure 2.8 – Setup for determining flow speed and shedding frequencies.

Although the tunnel is built with a variable power supply to be able to change the speed of the tunnel, the most reliable speeds are held at the power supplies 12 and 5volt lines, that don't go through the linear voltage regulator, an explanation of the power supply and construction can be found in **appendix D-1 (chapter 2.3.3)**. I therefore ran the tunnel at these voltages to determine the flow speed and shedding frequencies achieved to use later when comparing and discussing the results. I made a strip from sign-material and laser-etched squares of 10x10mm. The flows were run and videoed from the top down so that the distance traveled by the water and speed could be estimated by counting the frames and squares traveled, videos can be found following the links in Figure 2.9 and Figure 2.10. At 5v the water speed in the tunnel is approximately 0.04m/s and at 12v the speed is approximately 0.1m/s. However, as the water speed after and before the obstacles, as well as the effective water speed in the Kármán street of course are not the same, it doesn't make too much sense to talk about the steady water flow speed in the tunnel. So, for future referencing I will use the voltage the tunnel is run at, to indicate the relative speed of the tests being carried out.

Figure 2.9 - QR-Video link to video of speed and shedding estimation of 5v flow . https://www.youtube.com/watch?v=lhcSn5E7YRk

Figure 2.10 - QR-Video link to video of speed and shedding estimation of 12v flow. https://www.youtube.com/watch?v=mb6nnlTq3DA

In the project thesis, **appendix D (chapter 4.1),** I showed how the shedding frequency relates to the flow and size of cylinder. I used the estimated *Strouhal number* for D-shaped cylinders of 0.2 (Gopalkrishnan et al., 1994). By using the characteristic formulas for the shedding frequency, I estimated the single side shedding frequency at full 12v, 0.1m/s, flow to be around 0.74Hz.

$$U_c=U(W/W-D)$$

Equation 2.1 - Effective flow speed

$$F_c= (S_t*U_c)/ D$$

Equation 2.2 - Vortex shedding frequency

With the video I am able to determine the actual shedding frequency by counting the number of sheds in the half minute video clips and then divide it by the time. For the 5v Kármán street

I find it to have a two-sided shedding frequency of 0.58Hz, or single sided frequency of 0.28Hz. The 12v Kármán street has a two-sided shedding frequency of 1.52Hz and a single sided frequency of 0.76Hz, which is very close to what I calculated earlier. If we consider where the water flow hits the front of the foil at flow speed *v* equal to 0.1m/s as the potentially highest experienced force acting in these flow scenarios. I simplify the density of the water to be 1000kg/m^3 and ignore static pressure of the water, p_1. We can quickly use the Bernoulli equation to derive the dimensions of the force, p_2, at this point of stagnation to be about 5Pa or 5N/m^2, which is relatively low. Any sensor I make needs to be extremely sensitive if it should be able to pick up these types of forces.

$$p_2 = p_1 + 1/2\, \rho\, v_1^2$$

Equation 2.3 - Pressure at point of stagnation

3 Divergent testing and probing of sensory inputs

During the initial phase of a product development project, exploring the solution space is important. This can be done either through literature research or physical testing, but a combination of both is probably the most fruitful approach. The goal of the project is to find a way to imitate the way fish are able to use flows, and I want to do this through investigating their ability to sense flows and water activity around their bodies. Imitating lateral lines is not unheard of. Using existing literature, it is possible to find solutions that have worked for others and their specific problems but there is no assurance that it would apply to my specific problem. There is also no guarantee that the solution made by others is the best way to solve the problem, and biomimicry along with a wayfaring and set-based approach to prototyping are ways to free oneself from solution biases and come up with original solutions. Likewise, there is a lot of information that does not transfer through writing, and experiencing yourself what works for you is often beneficial either way.

With a good testing environment, I can make multiple prototypes of potential solutions and test and evaluate them fast, both by investigating new solutions and existing solutions suggested by other research. Working on and testing multiple solutions in parallel has is shown to often produce a better end result as the solution bias is minimized (Dow et al., 2012).

When multiple solutions have been tested I will have a better overview of the solution space and be able to make a better and more informed decision about which solution fits the best to my problem. When a promising solution is found, work can be done to specialize and tweak the solution to my problem, through convergent development and prototyping.

3.1 Sensory systems of fish

Fish are often thought of in their entirety as a sensory organ. Simplified, the way fish feel and detect water flow is through hair cells (Flock & Wersäll, 1962). All along the fish body we find hair cells that are imbedded in cupula, both on the surface but also internally in lateral line canals. The highest concentration of hair cells is along the lateral line, and the head of the fish, where the most information about the flows can be gathered. The hair cells, like the ones found in human ears, detect motion of the hairs connected to nerves. When the hair is moved signals are felt. These signals can then be used to deduce pressure, flow, vibrations and movement. These relatively simple receptors can in great numbers create a good picture of the hydrodynamic situation around the fish, which is often more complicated than one would have thought such as with turbulent boundary layers in dolphins (Fish, 2006). The sensors we use for flow sensing today are often bulkier and require more space and installation time, thus making a grid of thousands of sensors, like hair cells in fish are deployed, impractical.

3.2 Flow detection and visualization

Observing and visualizing flows and aerodynamic properties can be challenging. Many of the techniques used today base themselves on analysis being carried out in a controlled test environment and are perhaps not suited to be carried internally in a smaller moving foil. Yet inspiration can be gathered from the methods to come up with a new sensor technique.

A commonly used method to detect flows in experimental setups is by using heated wires. By measuring the amount the heated wire is shortened or lengthened one can estimate the cooling effect of the water and thus the flow speed around the wire at the given moment (Vogel, 1983). Running heating elements in water is not a very energy efficient way to measure flow, but the method could be scaled to multiple datapoints, measuring the true temperature of heated pads along a foil skin. It would however require accurate measurement equipment for the small temperature changes and really just shifts the problem rather than solve it.

Hair cells are fascinating in their simplicity yet advanced ability to accurately feel flow scenarios. In attempts to mimic the functionality of hair cells, artificial hair cells have been made in numerous different ways (Asadnia, Kottapalli, Miao, Warkiani, & Triantafyllou, 2015; Chen, 2007). The solutions often involve small flaps or hairs that move a piece of piezo resistive or electric material so that miniature movements can be measured. Similar solutions could be tested to see if they can be used in my foils.

Observing flows optically could also be a way to measure flows. In my project thesis I investigated the use of particles to visualize flow fields. By using cameras and software a particle image velocimetry analysis can be made of flows with uniform particles in them. This again is a solution that is probably better in experimental setups as it requires modifications of the test environment. Laser Doppler velocimetry does not require such particles to function, but rather measures the doppler shift in a laser beam to determine the flow in transparent fluids, but due to size price and sensitivity it also is probably best suited for experimental setups.

Another way to measure flow is to not think of it as the movement of water but the forces the movement of the water exert. A flow change should be accompanied by pressure changes. Measuring pressure is the normal approach to finding air speed in airplanes with pitot pipes, but can also be used to determine flow scenarios in water (Salumäe & Kruusmaa, 2013), by comparing the pressure reading from the front and the sides of the body. The ability to use pressure sensors to measure flow opens up a big world of sensors that can be tested and modified.

3.3 Modifying general purpose pressure sensors

A common way to detect flow speeds both in water application and in avionics, is by using pressure sensors. A similar project in which a robot fish is made to recognize different flow scenarios uses this approach, by adding two measurement points along both sides of the fish and one on the front with tubing to internal pressure sensor modules (Salumäe & Kruusmaa, 2013). This way they were able to, under controlled conditions estimate its position behind a cylinder in a flow and make counter measurements to keep the correct position for Kármán gaiting behind the cylinder. The pressure sensors used in this experiment, *MS5407-AM*, and similar modules are reasonably affordable, accurate enough, and widely used for this type of application. They do however add a large footprint on the design of the final part, both due to their dependency of tubing to stop water from entering the sensor, and the module size itself. If more sensors were to be added to a robot fish or hydrofoil to more accurately predict flow around the foil with little prior knowledge of the flow scenario, this type of setup, although proven to work, could quickly become impractical.

A possible solution to make the use of pressure sensor modules more streamline, is applying them along the skin of the foil. Pressure modules in water use air pockets in thin tubes to prevent water from entering the module and measures the pressure of the residual air pocket in the tube between the water and sensor. A potential way to remove the dependency to tubing, is by isolating the pressure module so that it is still able to pick up the pressure from the water yet, not be in direct contact with the water.

3.3.1 Pressure sensor in mineral oil

I used *BMP280* air-pressure sensors and tried to modify them to work under water. The sensors original purpose is to measure barometric pressure and derive height above sea level, but the low size and prize of the module makes it a good candidate for modifications. It also has a high sensitivity to pressure changes, the *root mean square* (RMS) noise is reported to only be ±0.2Pa, as it is meant to measure small pressure changes in air, which means that it might be sensitive enough to deal with the low forces acting in the water flow. While making a waterproof air-pocket around the sensor might be good enough to way to keep the sensor dry while also transfer some of the pressure, the compressive characteristics of air might give a foil unwanted dampening effects. A better solution would be to submerge the sensor in a non-compressive and non-conductive fluid. To solve this, I submerged the *BMP280* pressure sensor in a vegetable oil filled rubber glove. I then submerged the prototype in water to see if pressure differences would be easily picked up. Moving the water around and against the flow to see if any pressure drops and peaks were picked up.

3.3.2 Results

Using the BMP-library and example sketch I was quickly able to read out data from the pressure sensor. The example code and library interpret the digital signal from the sensor and displays both its pressure, estimated height above sea level, humidity, and temperature. I kept an eye on the pressure data and lightly stirred the water around the sensor. Although not significantly, there were definitive pressure changes when the water was stirred.

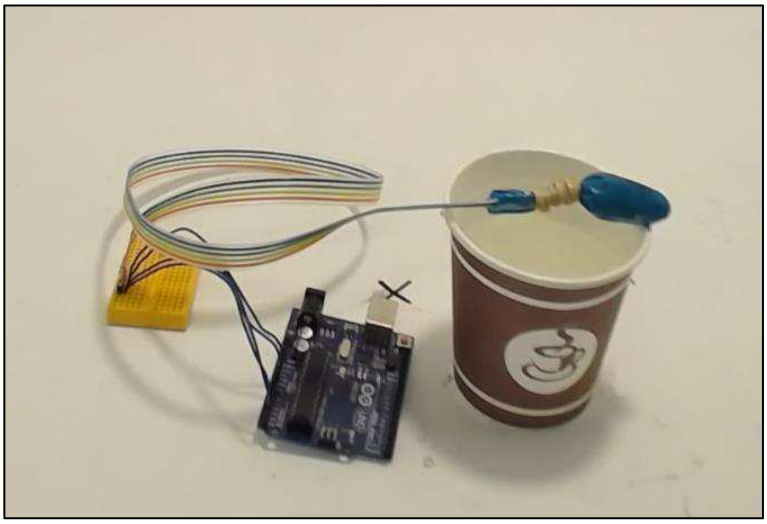


Figure 3.1 - BMP280 sensor inside an oil filled rubber pocket. The pocket was submerged in a cup of water.

3.3.3 Attempt at larger scale

Applying this to a foil would mean finding a way to implement multiple sensors into the foil as more data points are needed. In an attempt to scale up the production I made a sheet with hollowed out pockets for sensors in silicone rubber. The *Ecoflex® 00-30 liquid rubber* was mixed 1:1 by weight and poured into a mold made from laser cut acrylics. The sheet was cast stage-wise to create the void pockets filled with air. Note that casting oil directly into the silicone rubber would be ill-advised as many materials and substances cause inhibition for the *platinum cure rubber silicone* used. Especially fat, oils and hydrophobic liquids seem to affect the silicones curing abilities. When the cavities were cured I cut a small slit along the edge, just big enough so that a pressure sensor module could be placed inside when fully stretched. Wires fit through the hole. The cavity was then filled with oil and silicone applied along the slit to seal it.

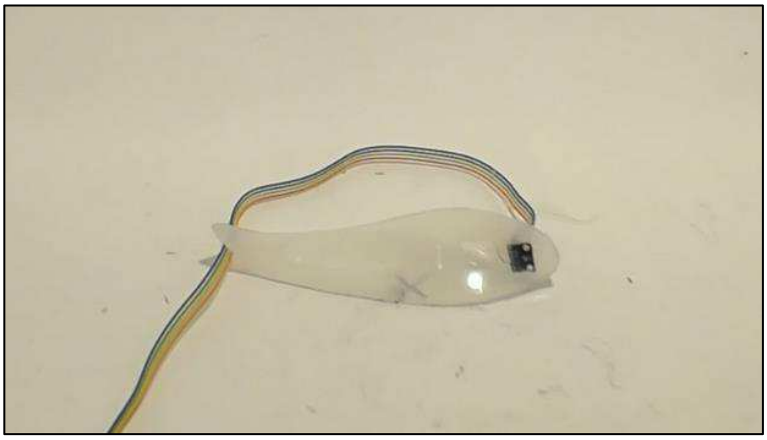


Figure 3.2 - Pressure sensor inside silicone sheet

An immediate problem was that the silicone would not cure along the entrance wound, as it had been contaminated with oil. Using one of the un-opened cells I attempted to fill it with oil post curing of the silicone using a thin syringe and entering through a thick part of the silicone so that it would close in on itself and seal the entrance point post oil-filling. The viscosity of the oil was however too high for any syringes available that were also small enough to enter the silicone and not cause too much damage. Creating sensor pockets this way, would mean finding a better way to construct the pockets sequentially. Another weakness with this concept was the resulting low flexibility and organic feel of the skin. The module felt bulky in the skin, being relatively large rigid part, and the wires coming from each module cause the area around the sensor to feel much less free to move and flex. Although using of the shelf pressure sensors and modify them to fit the purpose would be a safe approach it did not feel like an elegant solution to the problem and was not pursued any further.

3.4 Optical sensor

Another way to think of the water moving around the foil is not through the pressure it exerts on the foil but the visual impact of the flow itself. A way to measure the way visual features move in accordance with a reference point is through *optical flow sensors* (OFS). Measuring the visual impact of the flow would not alter the flow in any meaningful way and would be a non-intrusive approach to flow measurement. Most of the techniques for optical flow measurement use visual features to estimate movements speed (Barron, Fleet, & Beauchemin, 1994), and are usually used to measure an objects speed through a visually changing environment, such as in aircrafts to determine the ground-velocity of the aircraft. Although there are some studies that use optical sensors to calculate wind speeds, these often rely on other sensors as well (Rodriguez, Andersen, Bradley, & Taylor, 2007). More familiar perhaps is the use of similar sensors in computer mice.

To make an optical sensor applicable in water, this means that a high enough concentration of impurities with zero buoyancy must be present in the water so that flow changes can be seen and picked up by the sensor. To quickly determine if this could be a realistic approach, I used an old infra-red computer mouse and water proofed it with a zip-lock bag. I then put it in a bowl filled with water. I gradually added silver coated glass spheres made for laser-sheet flow visualization into the water until the mouse was able to move the cursor on the computer.

3.4.1 Results

The resulting water had an extremely high concentration of glass spheres, as seen in Figure 3.3, and close to no transparency, making it a rather impractical method of flow detection for most real-world scenarios. Using more advanced cameras and software one might able a lower concentration of impurities, but with relatively large sensor module foot print and need for manipulation of the environment it is applied to it did not feel like a good or robust solution in this case.

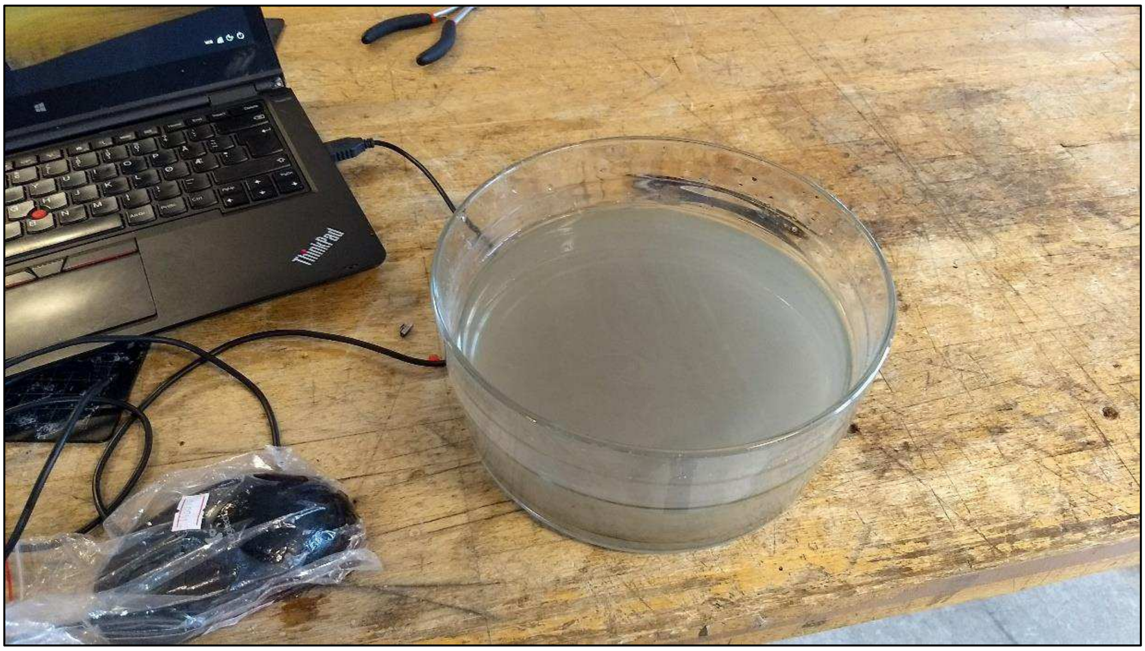


Figure 3.3 - Contaminated water at point where optical mouse can sense movement.

3.5 Analog resistance changing hair cells

Taking the concept of biomimicry even further, we could also consider not only how fish are able to adapt to flows; through sensing water movement, but also observe the functionality-model of the sensory organs the fish use to sense said flows; hair cells (Flock & Wersäll, 1962).

Most of the existing solutions and methods which imitate hair cells base themselves on attaching tiny hairs to piezo-electric or resistive films and materials to measure any force exerted on the hair (Asadnia et al., 2015; Chen, 2007; Zeng & Zhao, 2011). The solutions listed are naturally very small in footprint, and range in size between 1.4-0.05mm in total footprint. The small size creates a need for special equipment and machinery for them to be produced such as *Microelectromechanical systems* (MEMS) micromachining and *sol–gel method* for growing thin-filmed metals. As a result of the more advance manufacturing process to create these sensors the obtainability of the sensors is naturally low.

In an attempt to make an artificial hair-cell design that is more easily produced I initially experimented with *piezo electric films.* But I quickly realized that they were too flimsy to solder and modify without proper equipment. I therefore decided to instead attempt to use the conductivity of the water itself as the piezo-resistive material, letting an electrically conductive and flexible grounded hair move about measuring pads pulled high through a known resistor, and measure the resulting voltage on the pads. To demonstrate that this would work I first made a large-scale prototype with multiple grounded carbon fibers. These were glued in the center of four copper taped pads, which were connected to a power source through individual *light emitting diodes* (LED). The LEDs would then give an indication as to which pad experienced the best connection to ground, aka. being the closest to or forming the most connections with the carbon fiber hairs, the model being triggered can be seen in Figure 3.4. Scaling down the sensor, would enable me to install multiple sensors along the surface of a foil, getting meaningful data about both the direction and the amplitude of flows that move the hair between the pads. However, as each sensor unit gives four individual analog signals, an expansion of the analog pins of a normal Arduino UNO would be needed to be able to read more than one sensor unit. To solve this, I investigated using analog multiplexers along with the Arduino to expand its capabilities.

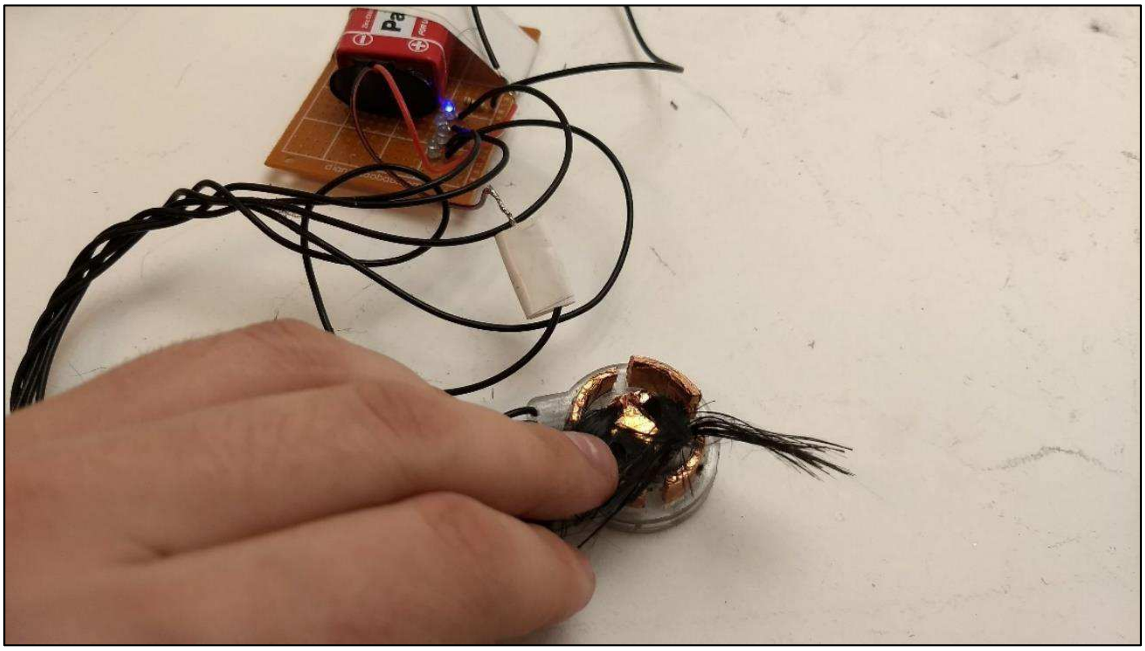


Figure 3.4 - Hairs are pushed to one side triggering a single LED

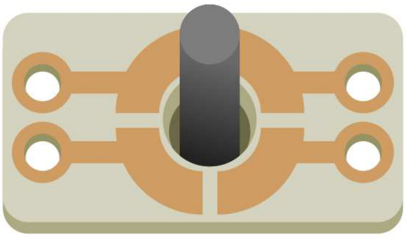


Figure 3.5 - Sketch of piezo resistive hair cell

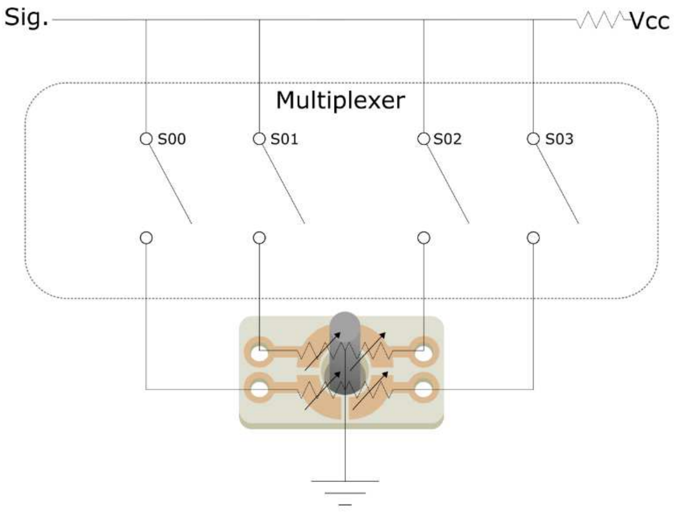


Figure 3.6 - Schematic sketch of hair cell connected to multiplexer

3.5.1 Sizing down

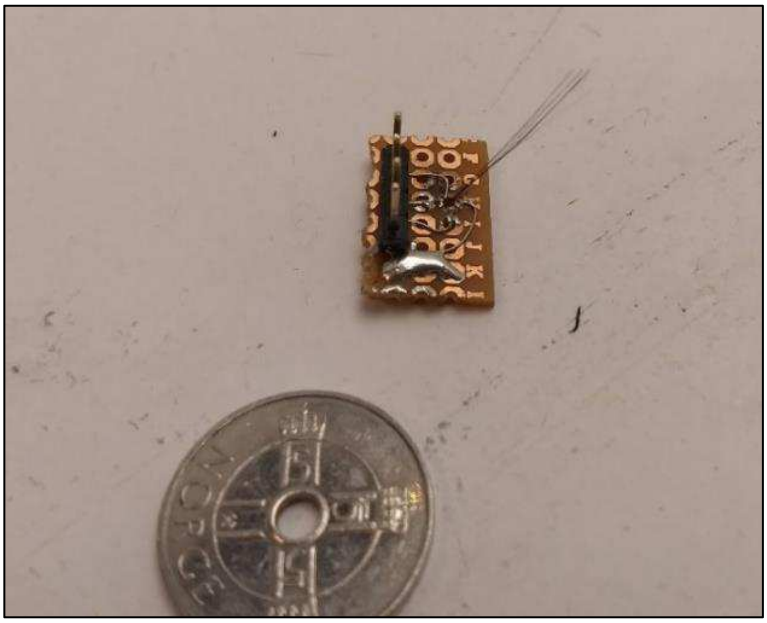


Figure 3.7 - Single hair cell made from prototype board

The seemingly simple design of the analog variable resistor hair cell made me think that it should be easy to create also in a miniature size with *printed circuit board* (PCB) prototype boards. Using a single connection pad, I soldered thin single core wires to four sides of the pad and lightly scored the pad between each quarter of the pad to separate the connection points. From the other side I added another prototype board with carbon fiber strains soldered to one pad. With a single sensor this was unproblematic. I was able to use a multimeter to verify that there was indeed a change in resistance on the pads and the carbon fiber hair when the hair was moved around in water, but when creating more sensor units, I found that the production method was too sloppy and unreliable, and the pads would easily connect to each other or lift from the prototype board, making a large scale production this way unrealistic.

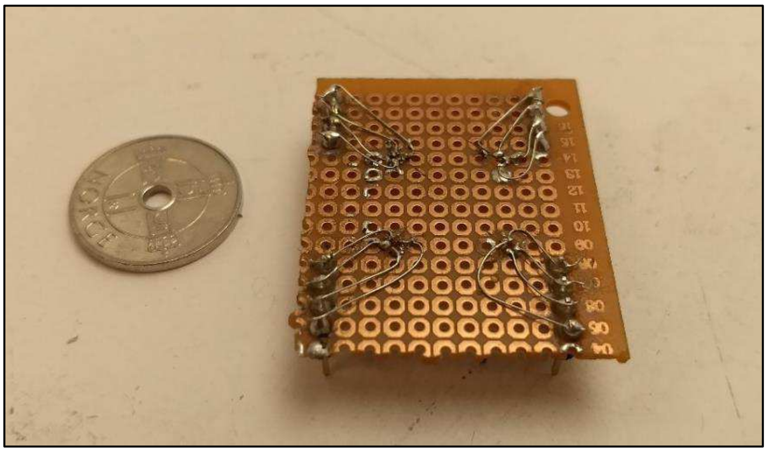


Figure 3.8 - Four carbon fiber hair cells fit on prototype board

3.5.2 CNC'ed pcbs

Having tried and failed to achieve multiple good reliable hair cells through soldering and hand carving prototype boards, I realized that this was perhaps not the best way to produce small electrical components like these. Although etching might be a gentler way to produce fine details in PCBs, having direct access to a *computer numerical control* (CNC) mill with tools and drill bits for PCB production I felt that this was the best solution to try first. I used the smallest drill-bit of 0.8mm to decide the scale of the sensor itself, where the hair enters between the pads. I added two paths to split the pad around the hole into four sections. Each pad section got a connection path leading away from the sensor to separate pads so that wires could be connected for sensor readout. I mirrored the layout of the tracks and connection pads so that wires could be connected on either side leading to the pads. Designing the PCBs like this could enable a matrix with multiple multiplexers along both the sensor and the ground channels, as described in chapter 4.2.1, in the future to vastly increase the available analog channels.

I initially sized and milled the PCB consisting of four sensors to be only 37mm long, and although all tracks and sensor pads seemed fine, the small scale of the PCB made it very hard to attach wires to the output pads without breaking them or connecting to the neighboring pad, the failed sensor can be seen in Figure 3.9. To improve on this, I increased the size of the PCB with 50% and re-milled it. After milling I used wet sandpaper to remove any burr buildup and fluxed all soldering points.

The PCB board was two sided, enabling carbon fiber hairs to be soldered directly on the back through the milled holes. I used an *x-acto knife* to cut back the copper plating on the back along the edges to reduce the potential contact between the grounded back and the analog read-pads on the front through the surrounding water along the edges. A single wire for ground was soldered to the back, while each sensor-point on the front got its own wire. I shielded all the sensor wires with copper tape which was also wired to the ground plate on the back of the sensors to reduce any potential electrical, capacitive, RF, and EM interference. The sensor wires were then directly soldered to a 16-channel analog multiplexer so that all sensor could be read through a single analog pin on an Arduino. I coded the Arduino, with a matrix for the binary address values for the 16 sensor inputs and switched between which channel was read with a simple for-loop, a better explanation of how to control and use multiplexers can be found in chapter 4.2.1. Each sensor value was printed as a comma-separated value with line break to indicate the end of each recording so that they could be plotted directly with Arduinos serial plotter.

I submerged the sensor unit into my water tunnel which was filled with a salt water mixture to increase the conductivity of the water. I moved the unit manually to simulate movement about the hairs. And also let the pump run to show actual flow about the hairs.

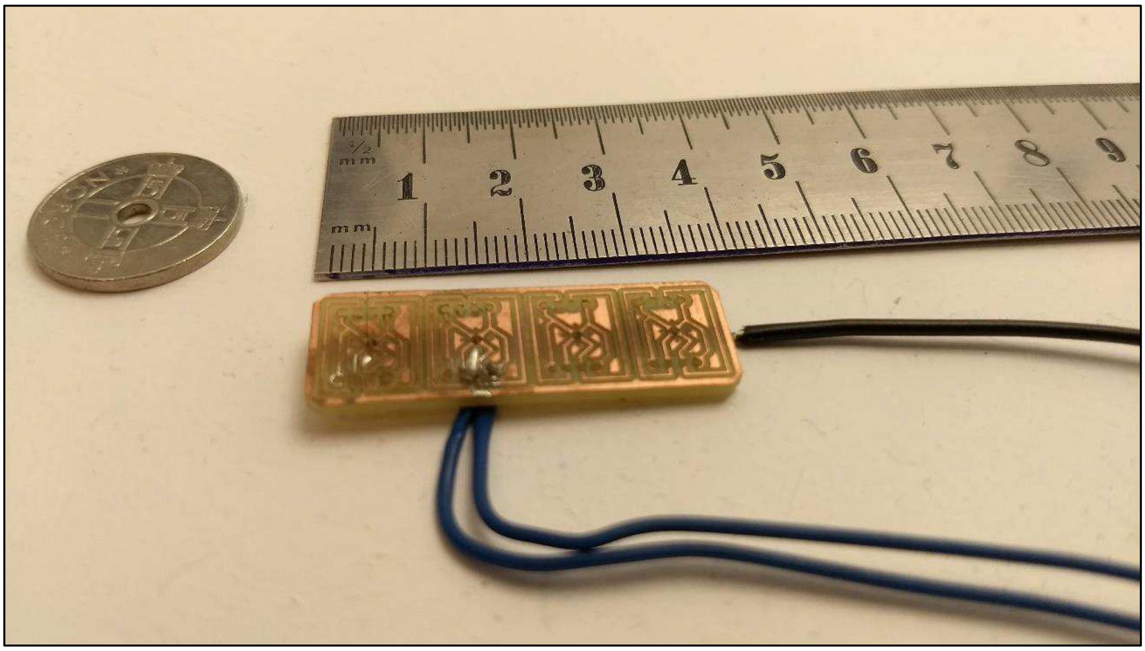


Figure 3.9 - Smallest CNC'ed PCB

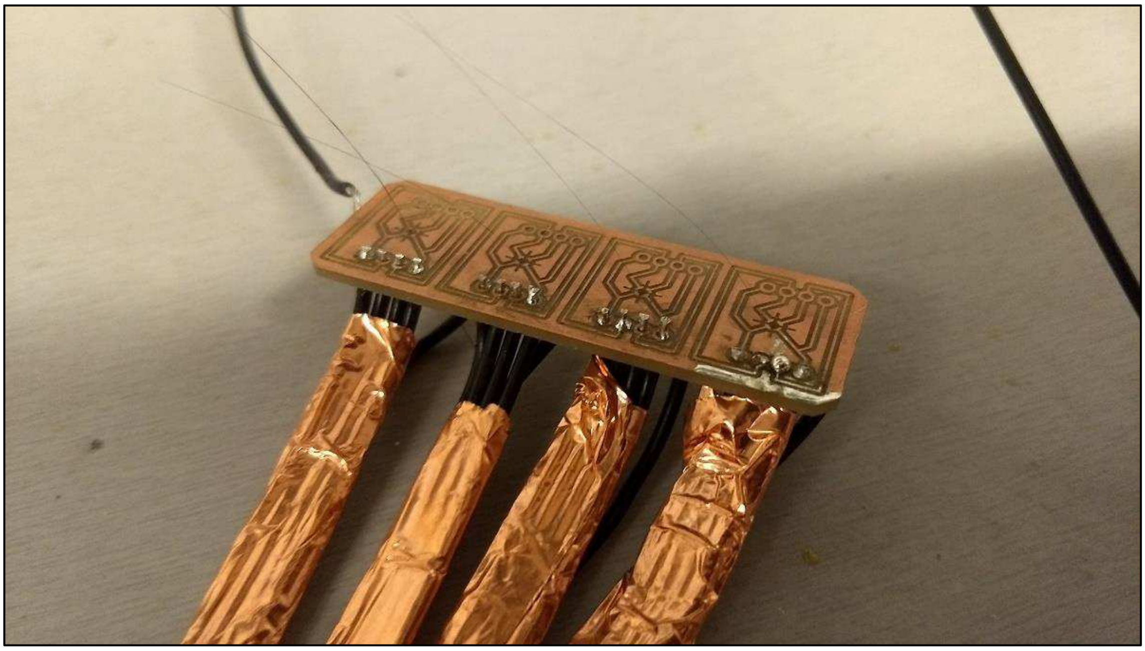


Figure 3.10 - Larger CNC'ed PCB

3.5.3 Results

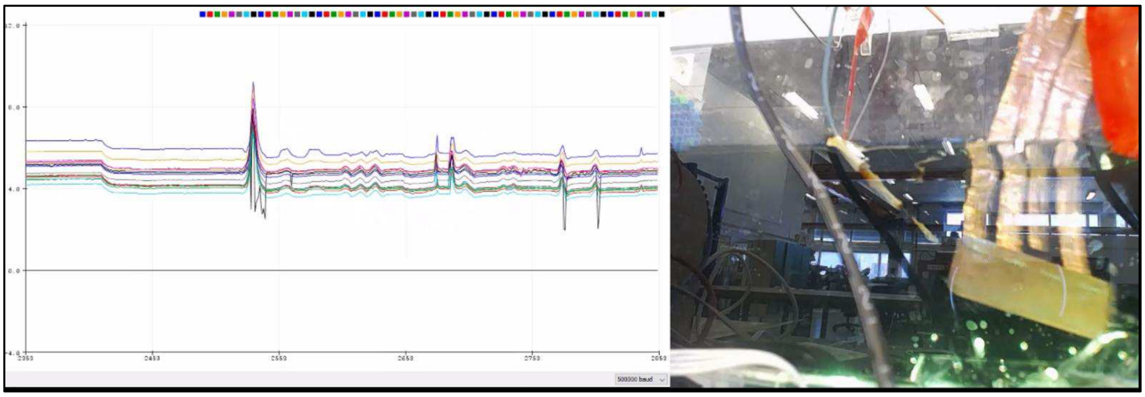


Figure 3.11 - Still image of sampled hair cell data and test environment.. Y axis amplitude is relative to the calibration values and is set to range from 0-10. X-axis show sample number.

Moving the plate with hairs in conductive water definitely affects the resistance between hair and pads. A video showing the effect can be seen in Figure 3.12 and the screen cap in Figure 3.11 shows some of the peaks of the movement. I would have expected one resistance to be reduced while the resistance on the opposite side would increase similarly. With the short burst in movement it is hard to tell whether this is the case or not. I can see that some values do fall while others rise, but the rise is much more dramatic. It might be that when the hair is forced flat, the hair curls causing it to have points of closer proximity to more of the pads, or that due to the long length of the hairs used, the hairs fall flat enough to interfere with the tracks of the PCB. This could also explain why the hairs seem more sensitive in the upwards and downwards directions in the video, as this causes the hair to get closer to the connection pads. Even with somewhat strange behavior this could be a viable way to show the current direction and amplitude of a flow.

Figure 3.12 - QR-Video link to video of har cell being moved in water with live plot of analog pad signals https://www.youtube.com/watch?v=Sf77WAO7VDU

As can be seen in the video, the hairs react very poorly to rapidly changing flows. Using stiffer or shorter hairs should make the system respond faster to changes but when I cut the hairs of the sensors shorter I also observed a significant reduction in their sensitivity. As I mainly want to be able to sense flows and vortices that are not unidirectional this might not be the best method to measure them.

Another thing I found using carbon fiber hairs is that they wear down very rapidly. I suspect that this is mainly due to my manufacturing process, where I mechanically lock the hairs in place with solder. This subjects the fibers to high temperatures which might damage the fibers. It might also not be a tight enough lock to hold the fibers in place over time. Even with few and short testing sessions I've had to replace the hairs on two of the pads. Making the sensor somewhat unsustainable.

As the hair cells are electrically connected to the water-environment, are also very sensitive to the conditions of the water. The conductivity of the water will greatly affect the sensor readings, and any dirt and algae buildup could quickly damage the sensors.

I also discovered that the electric potential of the water of course will affect the sensor values, and any live wires in proximity to the sensor will affect the sensors read-out. This error can be seen in the video in Figure 3.13, when the hydrogen bubble generator is turned on and off repeatedly. The sensor also produces an electrochemical cell. And even though the current through the sensor should be very low (max. 0.005mA) over time the cell reactions might aid in buildup of corrosion and degradation of the metallic parts of the sensor. This effect could be used to benefit the sensor by letting the carbon hair be the anode and thus giving the copper a cathodic protection against corrosion (Baeckmann, Schwenk, & Prinz, 1997), but over time direct contact between metal and an electrolyte might not be the characteristics of a sustainable solution.

Figure 3.13 - QR-video link to hair cells showing electrical capacitance and charge of water . https://www.youtube.com/watch?v=GrK8iFV5T3A

3.6 Carbon compounds in matrices as sensors

When it comes to smart-skin research and pressure sensing, using carbon-based nanomaterials such as graphene and carbon nanotubes has been a popularly investigated idea in recent years (Dharap, Li, Nagarajaiah, & Barrera, 2004; Kang, Schulz, Kim, Shanov, & Shi, 2006; Li et al., 2012; Loh, Lynch, Shim, & Kotov, 2008; Yan et al., 2014). These types of sensors base themselves on the remarkable properties of the compounds such as their conductivity and piezo-resistive behavior. Being small compounds of the Nano scale, means that they are easily mixed with other matrix forming materials, and can be cast directly into the material that you want to measure the strain of, such as silicones and concrete (Yu & Kwon, 2009). Although the matrix in which you infuse the compounds certainly effect the end-characteristics of the sensor, it would seem that piezo resistive behavior of the Nano materials is retained in a high variation of matrices that they are cast into. One such variation, that shows the flexibility of graphene-based sensors is the so called *"G-putty"*. As shown in "*Sensitive electromechanical sensors using viscoelastic graphene-polymer nanocomposites»* (Boland et al., 2016), G-putty can be formed by mixing graphene and *silly-putty*.

Although carbon-based nanocomposites have been a popular subject in the research community, in the real world makers and hobbyists lag behind. Whether it is due to the lower accessibility to the compounds in the private marked or the toxicity of the compounds (Ou et al., 2016) which means that extra care and precautions needs to be taken when working with it, either way it seems that using piezo resistive carbons has not yet reached a stage where it is actively used in private projects and prototypes. Instead other cheaper solutions are used, such as graphite and anti-static foam, to create piezo-resistive materials that can change shape (Gawron, 2011). Although it is hard to argue with the promising results shown by research into carbon-nanotube- and graphene-based sensors; obtainability, workability and flexibility are important aspects in projects that I feel these sensors don't fulfill yet. In an attempt to re-claim the method, but make it fit a prototyping and maker environment I aimed to investigate other carbons for their suitability as sensors.

3.6.1 Modeling clay

The G-putty (Boland et al., 2016), show such piezo resistive properties, that it is able to detect the individual footsteps of spiders. While silly putty, like most other carbon sensor matrices is silicone based, it shows that not only typically solid materials need to be used as a matrix, but also fluid-like materials can be used in the right circumstances to make pressure sensors from carbon nanocomposites.

Graphite can be used to make conductive materials such as glue, ink, and 3d printer filament, as it is conductive and fine enough to be mixed with other materials. Conductive materials made with graphite often have a high resistance through them as compared to metallic wires, by mixing graphite with flexible materials, this resistance can be used to make pressure sensors. When compressing or stretching the graphite infused material, change in distance from one measurement point to another will correspond to change in the resistance between the points (Gawron, 2011).

To test the sensitivity of these types of homemade sensors, I made my own. Using modeling clay and fine graphite powder, I made a quick conductive paste by mixing the graphite powder into a small lump of clay, half a teaspoon at a time. With limited knowledge as to the correct mixing ratio, I went by consistency of the dough and conductivity. I kept adding powder and kneading the dough until I reached a point where the clay lost its flexibility and formability after each addition of graphite, and only re-gained the flexibility after thorough mixing and kneading. I used a multimeter to check that a resistance could be measured across the dough, and then deformed the dough while it was still connected to the multimeter. With increase in the distance between the measurement-points the resistance increased and narrowing the dough between the electrodes would also increase resistance.

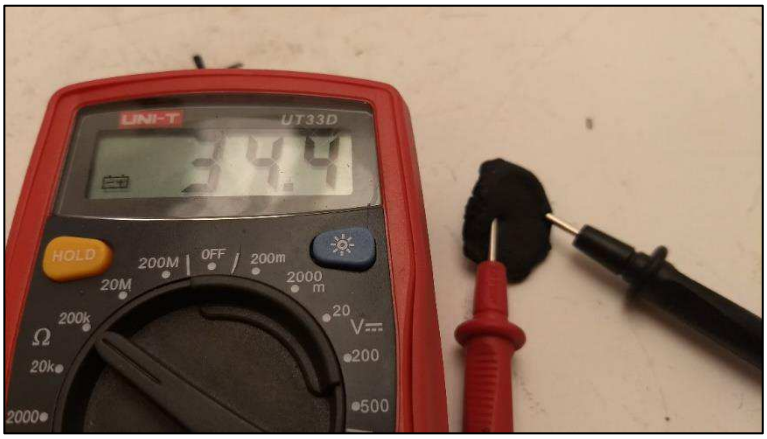


Figure 3.14 - Graphite infused clay before stretching.

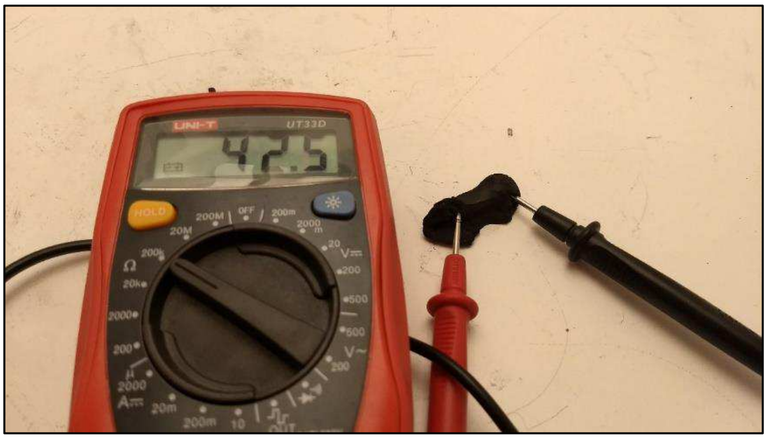


Figure 3.15 - Graphite infused clay after stretching

To see how suitable this type of sensor would be in a silicone foil, I did as in the previous experiment in chapter 3.3 and fit pieces of the dough into hollow pockets in a cast silicone body. I added wires as individual electrodes for each piece of dough and a common copper-tape piece along the back of the dough pieces as a ground electrode.

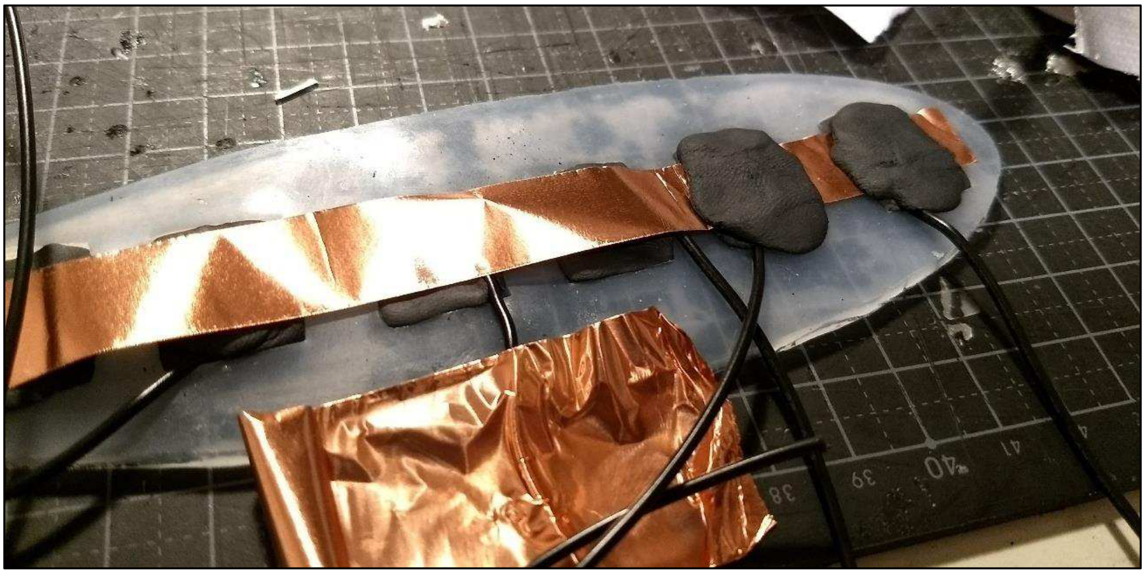


Figure 3.16 - Play dough sensors being placed into silicone foil

When measuring the resistance of the confined dough pieces it became apparent that when confined it was extremely hard to manipulate the doughs in ways that would greatly influence its resistance. Using my thumb to apply generous amounts of pressure on a single sensor, I was only able to change the resistance by ~13% from 24kOhm to 21kOhm without completely flattening it and shorting the two leads. It would seem that this type of graphite-based sensor is much more dependent on the freedom to deform than actual experienced pressure, and would not be suitable for installation along a foil surface.

3.6.2 Rubber Silicone as matrix

One of the bigger challenges in my project thesis was adding water proof outer layers to protect any internal electronics of electric fish suspended in the water. The solution was constructing a skin of cling foil and vacuum tape and brush rubber silicone on the surface to ensure that it was water proof. This worked as waterproofing but offered little flexibility in terms of later addition or modification of the electronics inside the foils and robot fish. Constructing the shape of the foil, with most of the sensor-electronics already imbedded in the silicone would waterproof the sensor-electronics while allow for more advanced geometry and opening/closing mechanism to be implemented for access to internal electronics in a future hydrofoil. I therefore set out to investigate whether rubber silicone would be a suitable matrix for graphite-based sensors.

Not knowing an appropriate mixing ratio of graphite to silicone, I opted to create two different samples with different ratios. I used the ecoflex 00-30 silicone for all experiments. The samples were cast into small discs, 20mm in diameter and 3mm in height. One was mixed to a completely black color while still kept at a liquid consistency. The other was mixed to a point where the silicone was no longer pourable. When testing the samples, I found that neither of the samples were able to make good surface contact with a multimeter. And only when a large surface electrode covered the entire surface, was I able to measure some sort of resistance through the sample, although the resistance was extremely high (>100kOhm) and due to the proximity of the electrodes it might as well have been measuring resistance through other mediums such as fingers in contact with the electrodes or the table. I was not able to determine whether the material showed any proper conductivity let alone piezo resistivity.

3.6.3 Carbon fiber infused silicone

In a side leap, I was searching for ways to make conductive glue to fasten the carbon fiber hairs in chapter 3.5. Knowing that carbon fiber-epoxy composites are electrically conductive, I was searching for ways to use this in circuits when I came across the guide *"Silc Circuits: High Performance Conductive Silicone"* by the *instructables.com* user Blorgggg (2015), where he shows that it is possible to make flexible and soft wires by mixing carbon fiber and silicone. Additionally, he shows that when stretched enough, the resistance of the carbon fiber silicone increases. To test the applicability of this piezo resistivity, I made two samples, as with the graphite, of silicone mixed with carbon fibers cut to lengths of <10mm, one was mixed with fibers until it reached a high viscosity and one to a relatively low viscosity mixture. The fibers were cut from a sheet of *Easy Composites Black Stuff Carbon Fibre 2/2 Twill 3k.*

When measuring the resistance over these samples I was able to make much better connections than with the graphite-based samples and I had no problem getting resistances in the 1-6kOhm range. A weird property observed with these samples however where that the resistance in the samples fluctuated greatly. And would show high instability when moderate amount of force was applied to them. The samples showed a much higher sensitivity to pressure than any of the other materials and methods tested earlier, but distinguishing differences between pressure and noise was very difficult with only a multimeter.

3.6.4 Results

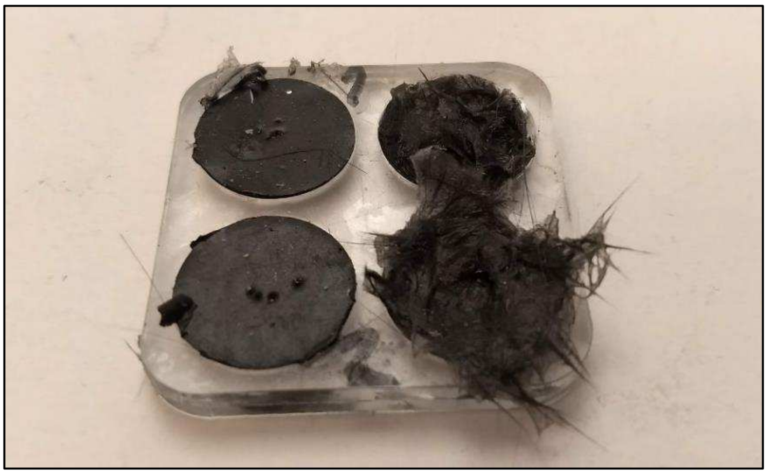


Figure 3.17 - Silicone samples . Samples mixed with graphite to the left and carbon fibers on the right.

Mixing graphite with less conductive materials, makes electrically conductive materials with a high enough resistance, such that change in distance, concentration, or cross-sectional area from one measurement-point to the other is measurable through resistance change. Although dependent on the deformability of the matrix used, this seems more applicable for higher forces than those I want to measure. The materials dependency on deformations, would not be suitable in a foil, where the goal itself is to control the shape and form of the foil.

The shown piezo resistivity of carbon fibers shows great promise in measuring small pressure changes. With the high sensitivity and bad setup for measuring it is hard to determine whether the extremely fluctuating readings are due to environmental noise such as RF- and EM interference or actual pressure, but small applied forces to the pieces gave consistent reduction in resistance and investigating this concept further would be interesting.

3.7 Converging ideas and redefining the problem

Through prototyping you explore your solution space. I set out to explore the solution space, thinking that the problem would be to find a way to sense flows non-intrusively, but as I worked with ideas I found that observing flows is not necessarily the biggest problem, multiple of the methods I tested, showed promise, and multiple more solutions have been used by others that don’t noticeably change the flows they are observing. The bigger problem was making something that I would be able to install into a flexible and moveable hydrofoil. The sensor would have to be small enough, flexible enough and producible in such a way that it would be possible to implement into foils without inducing too many design restrictions. It would need to do all of this while also get enough data points to make good estimations of the flows around the foil.

Being able to cast my sensors directly into the same matrix as the material I want to install them in or cast them into a specific shape that would fit the product would meet these criteria in ways that the other tested methods wouldn’t. To see if the carbon fiber and rubber silicone mixture would be able to produce good sensors for flow observation, I decided to continue to prototype and refine this idea.

4 Development and use of piezo resistant carbon fiber sensors

When trying to research the piezo resistive effect I observed in the carbon fibers when mixed into a rubber silicone matrix I found very little documented work that had used the effect as the basis for sensors.

Exactly why carbon fibers show piezo resistive properties doesn't seem to be fully understood, Blazewicz, Patalita, and Touzain (1997) gives a brief summarization of some of the theories as to the piezo resistive properties of the fibers, ranging from contact resistance of grain boundaries, stress effects on conduction in graphite planes or that the electrical carrier density and mobility is increased with increased stress, but offers no conclusive statements as to why the fibers show these properties.

Wang & Chung (1995) have made sensors using carbon fiber in an epoxy matrix. In carbon fiber composite parts an epoxy matrix is often used, and this type of sensor would be ideal for these composite parts as the sensor would show the same properties as the composite it is measuring. Not only when infused in matrices does carbon fiber generate piezo resistive sensors. In research by Mäder, Nestler, and Wielage (2011) the piezo resistive properties of single carbon fibers is investigated. And it is shown how carbon fibers on their own, without dependencies to their matrix can be used as effect strain sensors. Although Blazewicz, Patalita, and Touzain (1997) show that almost all carbon fibers show piezo resistive properties, some types of fibers are certainly more sensitive than others. Another interesting quality is that some fibers show an increase in resistance when forces are applied while others show a reduction. The fibers I have available are *Formosa TC35 (3K)* and being a newer fiber, it is not listed in this comparative analysis, but from previous test I observed that the fibers form sensitive piezoresistive materials that decrease in resistance when pressure is applied.

In *"Electrical conductivity of carbon black and carbon fiber filled silicone rubber composites"* Sau, Khastgir, and Chaki (1998) show how mixing carbon fibers with silicone creates a piezo resistive material that is highly sensitive to low pressures. They also show that mixing ratio of carbon fibers to silicone has little influence on the sensitivity region of the sensor, but only aids in changing the resistance which the sample works between. Additionally, they suggest that the material has a critical force, after which the change in resistance is only minimal, for their carbon fiber mixtures this critical force was at 120grams/cm^2 which should be well within the small forces that I am operating with in my water tank but could be of consideration should one make sensors of carbon fiber and rubber silicone at a larger scale, and in cases where larger forces are applied.

With limited research in the field of constructing carbon fiber sensors in silicone, the best approach might be through prototyping and trial and error. From my previous experiments I know that the fibers I have available show high sensitivity when mixed with silicone and can be further used as means for investigation.

4.1 Multiple datapoints in one matrix

To investigate how much useful data I would be able to get out of the *carbon fiber rubber silicone sensor (*CFRSS) I wanted to create a larger sample of the material, and attach multiple electrodes to measure the resistance between them, giving more datapoints and covering a larger area.

I cast a strip of 200x6x20mm carbon fiber reinforced silicone in the same manner as before. When cast, I gently sanded all the sides of the part to remove any silicone-film covering the top fibers and ensure good electrical contact along the surface of the part. I then grounded one side of the silicone-piece with wire soldered to coper tape and added datapoints on the other by adding pads of wire and copper tape. The pads were held in place with electrical tape as seen in Figure 4.2. The whole piece was then water-proofed by threading a condom over it with zip-ties. The electrical leads of the measurement points were then connected to 5v as shown in Figure 4.1, on the Arduino through resistors R_1, R_2, R_3, R_4, R_5, and R_6 of 4kOhms as it was close to the resistance from pad to pad through the carbon fiber-silicone piece (R_8,R_9,R_{10},R_{11},R_{12},R_{13}), ensuring that the change in resistance in the carbon fiber-silicone could be read as analog change in voltage at the data-pins. Each data pin was connected to a separate analog input pin on the Arduino, *A0, A1, A2, A3, A4* and *A5*, while the ground wire was connected to ground on the Arduino. A small piece of code was loaded onto the Arduino, which read the raw analog values of the pins and printed the values as comma separated values. The piece was then submerged in water which I gently stirred to see the effect on the different reading points in the matrix.

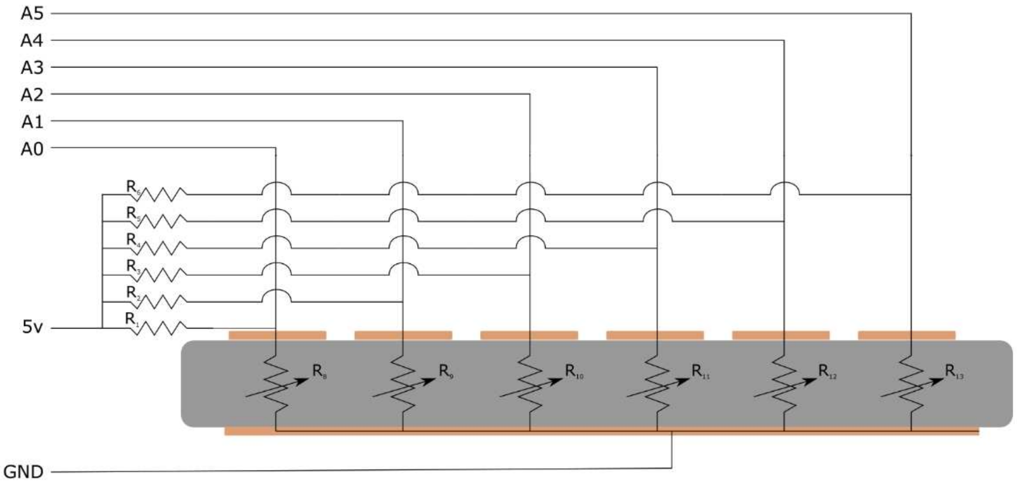


Figure 4.1- Schematic of multiple measurement points in single matrix

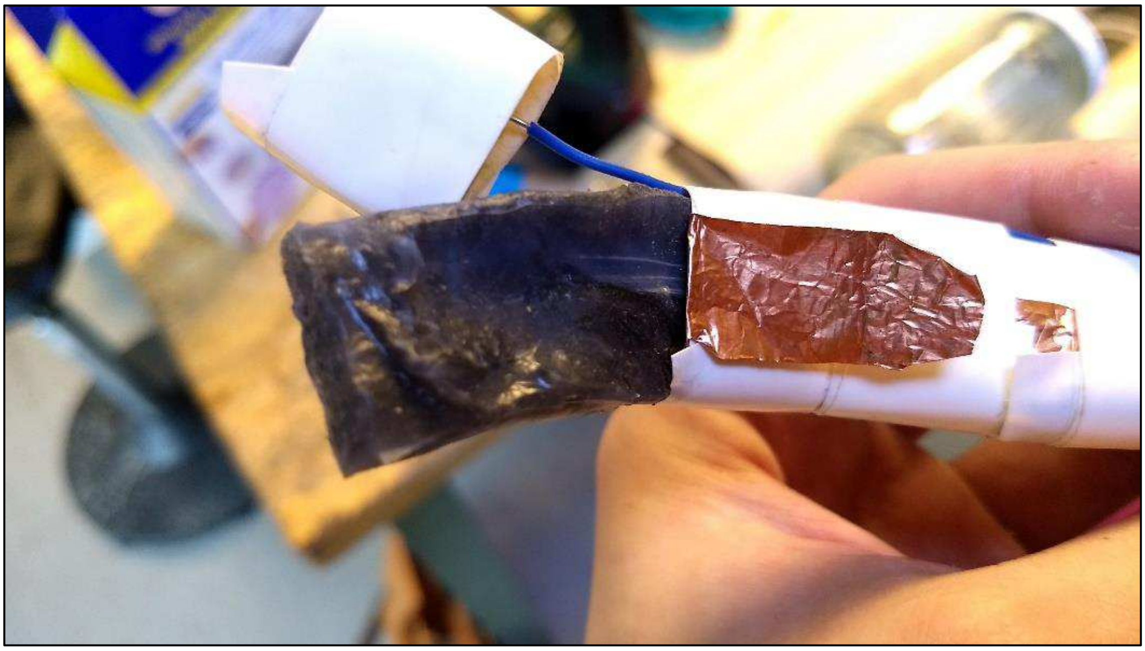


Figure 4.2 - sensor matrix with electrodes peeled away

4.1.1 Results

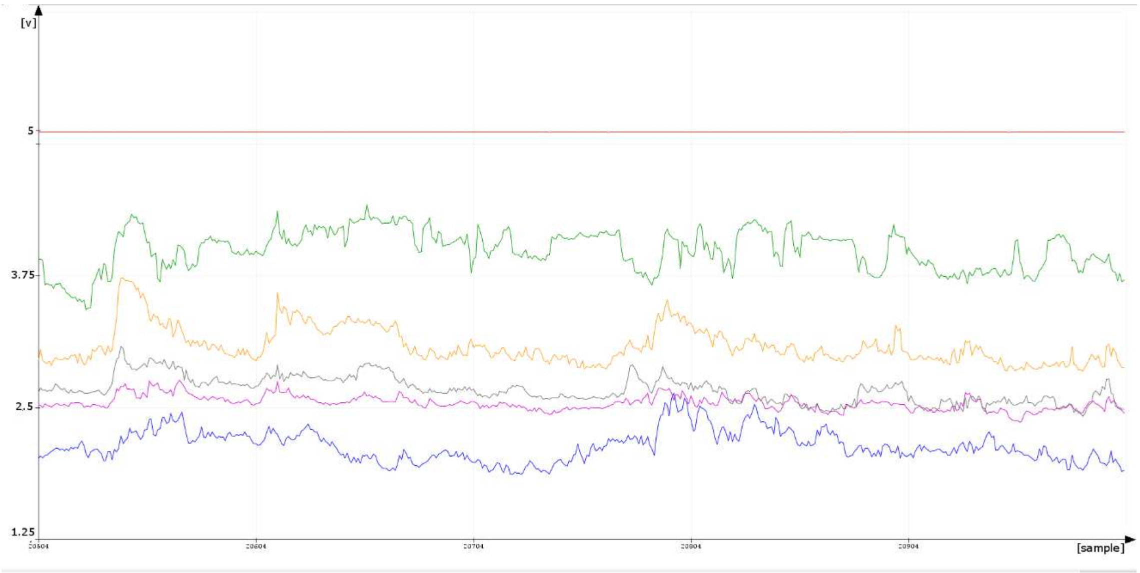


Figure 4.3 - Serial plot of analog voltage read from the data points. Note that the red sensor appears to be shorted to Vcc.

I used serial plotter in Arduinos *Integrated development environment* (IDE) to display the sensor-information as seen in Figure 4.3. The plot might not tell much of a story on its own. The sensors seemed extremely sensitive to pressure and vibrations, to a point where classifying which sensor peak corresponds to where was difficult. Not only would the readings fluctuate greatly when roadwork was conducted outside of the workshop. But also footsteps in the lab, around the table where I conducted the testing, could be linked with peaks in the readings in real time. The pressure change caused by submerging my hand into the bowl was also easy to see, and moving the water lightly made large changes in the measured voltage on the data points.

One of the biggest problems I found with the sensors was differentiating between the individual sensors. Touching the sensor strip on one side, seemed to affect all of the data points, and the rate of change in the sensors didn't seem to change much when I touched the other side instead. There seemed to be high crosstalk between the sensors. This could be due to the sensors all being implemented in the same matrix, or the way I read the data. Either way, the sensors seems very capable of picking up pressure changes due to small flow changes.

4.2 Separating data points

With the high crosstalk experienced when the datapoints were suspended in the same matrix, I decided to take measurements to reduce this crosstalk and able more local data to be read. Even though the sensors seem very capable of sensing the pressure of the flows, their high sensitivity also causes a high noise to signal ratio. With handmade sensors it would be hard to improve the sensors to a point where a higher quality sensor reading could be achieved, but by instead accepting the low quality of the sensor reading and increase the number of sensors, the story that the sensors tell together could compensate for their individual errors. I therefore wanted to increase the number of datapoints, while reduce the crosstalk between the sensors, as compared to the results in chapter 4.1. To do this I planned to separate the CFRSS pads for each datapoint and sample the pads individually. The Arduino UNO is limited to six analog input signals. If I want to increase the number of datapoints any further, I need to make additional modifications.

To generate a good picture of the outside flow from an internal/surface view of the foil, it would not be sufficient to sample only a small area of the flow. Like with fish the more reasonable approach would to give the entirety of the foil ability to sense local flow-conditions (Flock & Wersäll, 1962). To counter the crosstalk errors I discovered in chapter 4.1, I decided to cast all CFRSS data-collection points individually and separate the sensors with non-conductive silicone between them to dampen any vibrations and stop any electrical contact between the sensors in the casting process. To achieve this, I planned on using a stage-wise layer-casting; creating the skin, installing back-leads, casting separation walls over leads, casting CFRSS, installing top-leads and casting the top of the skin to seal it all off. Similar to the way IC's are made, this gives good control over electrical connections not only in two dimensions, but in all three. It is also a time-consuming process, as each layer requires at least 12 hours of curing-time in addition to work related time consumption. Casting the sensors directly into the skin with leads gives great freedom to increase the number of CFRSS data points, to handle the potential increase in sensors I investigated the possibility of using multiplexers.

4.2.1 Multiplexing and crosstalk

As with the carbon fiber artificial hair-cell sensor in chapter 3.5 a multiplexer may be used to increase the number of analog sensor inputs of the Arduino UNO to 16. A multiplexer works as a data selector, with one output line. By sending an address to the multiplexer, the multiplexer selects which of its signal input pins is written to the single signal output pin. The multiplexers I used are *CD74HC4067*, a 16 Channel Analog multiplexer. This means that up to 16 analog signals can be handled by a single analog pin on the Arduino. The multiplexer needs a binary address to select which input signal to read and when, which means that it additionally needs four digital pins to send an address. With a total of six analog pins on an Arduino UNO, we can increase the number of analog pins to 96 (6x16) with the 16-channel analog multiplexers. The trade-of will be the possible sampling rate. As the multiplexer will switch between which of the 16 incoming signal pins it reads to the single out-going signal pin the theoretical sampling-rate would then be 1/16 of the original sampling-rate of the Arduino. The normal read time of a single analog pin for the UNO is about 0.1milliseconds (“analogRead(),” 2017). That being said, when it comes to reading different values at a rapid pace to the same variable, a normal approach is to read each pin value twice, to make sure that the previous value held in the analog processor of the Arduino is purged, as the analog values are converted through a single 6 channel 10-bit analog to digital converter, and only treats one analog value at a time, much in the same way as a multiplexer. If all of these delays in addition to a 1ms delay for the multiplexers to switch gate and any residual capacitance in the sensor leads to settle, we see that for a circuit where we read 64 sensors individually one at a time, the practical read speed would be limited to ~13Hz, although in practice I found it to be close to 10Hz. Since any lower sampling frequency might reduce the accuracy of the data, I will limit my sensor arrays to 64 sensors, if I should need more sensors I would need to change the setup so that multiple datapoints can be read simultaneously.

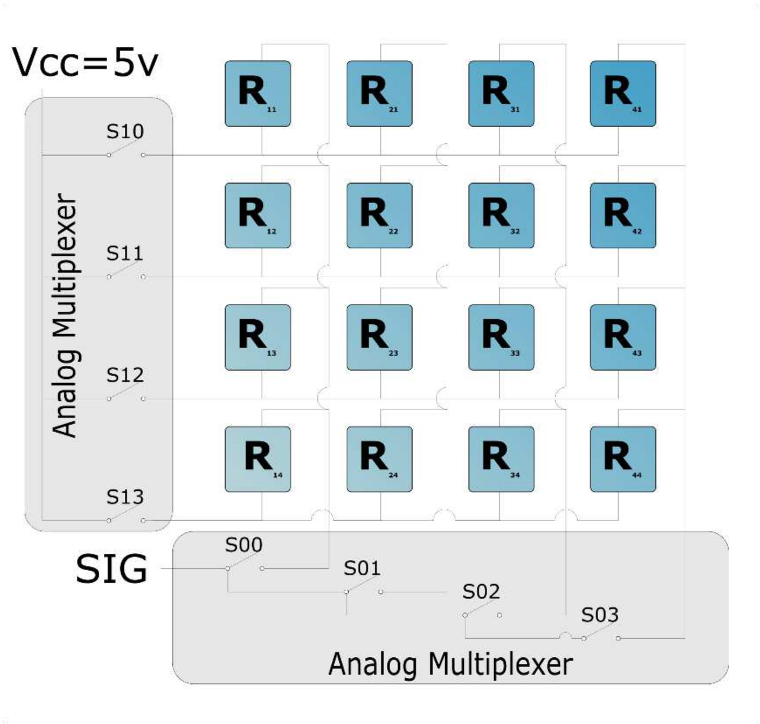


Figure 4.4 - Variable resistances in a 4x4 matrix multiplexing both columns and rows

Using two multiplexers in a line-wise grid as shown in Figure 4.4 is commonly used to make LED displays, as these often use *pulse width modulatio (*PWM) signals to control brightness and color mixing. The rapidly changing signals caused by the multiplexing doesn’t really affect the visual effect of the LEDs. Multiplexing on both the rows and the columns can also be used to create large arrays of digital on/off buttons to increase input. It is not that common to use a dual multiplexer setup for analog input however. The reason for this is the high chance of crosstalk between the analog inputs if the proper precautions are not taken. Deciding how to do the multiplexing I considered a couple of layouts and used them at different times in the process.

4.2.2 Column-wise separate pull-down resistors

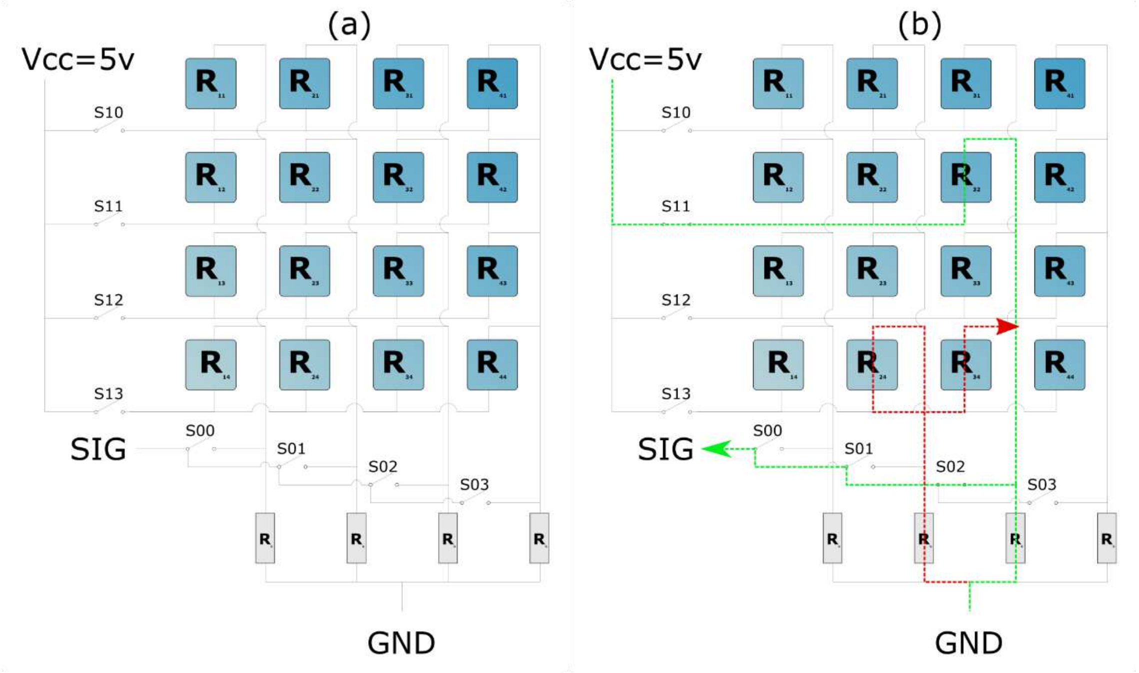


Figure 4.5 – 4x4 Matrix sketch with multiple pull-down resistors. (a) Multiplexer matrix with pull down resistors on each column (b)The desired measurement path is marked in green and a possible error path in red

Originally, I used the multiplexer setup in Figure 4.5a to read the data from the different sensors. Each column is pulled down with equal resistors to create a voltage divider so that the voltage can be read on the signal pin as output. Imagine that a signal gate on each multiplexer is opened, *S11* and *S02*, this closes a circuit with one single variable resistor, R_{32}, to the signals voltage divider as indicated by the green arrow in Figure 4.5b. This is the circuit we want to measure. However, we also get closed circuits going from the other pull-down resistors trough two of the remaining variable resistors, as indicated by the red arrow in Figure 4.5b. In fact, it is possible to create closed circuits like these with every resistor in the skin, as long as it connects to the column of *S02*, meaning that the resistance of any sensor would affect the measured resistance of the current sensor. It is also possible to make closed circuits going through 3,4,5,…,n resistors but the lover the number of resistors it goes through the less impact it will have on the measured results.

For the case of the error signals going through two resistors they act as pull downs on the signal. They will act in parallel with the existing pull-down resistor for the signal pin. We want to measure R_{32}, however the resistance governing the analog read voltage at signal pin would be dependent on all the analog resistances. From *Ohms law* we can derive Equation 4.1 for voltage dividers to find the analog voltage, *Vout*, at a measurement point between two resistors, *R1* and *R2*, going from *Vin* to ground.

$$V_{read} = V_{in} * (\frac{R1}{R1 + R2})$$

Equation 4.1

We see that the input voltage is multiplied by the relationship of the resistors in the voltage dividers. For the sake of future explaining I'll call this relationship *D*, so that:

$$V_{read} = V_{in} * D, \quad D = (\frac{R1}{R1 + R2})$$

Equation 4.2

To find the relationship between the resistors in the first multiplexer case, I simplify the case and only consider the signals going through two and one resistors to find the D in Equation 4.3 that would be used to describe the voltage change at the measurement points in the multiplexer matrix with separate pull-down resistors for each column. Although not strictly correct it highlights the governing error with this sort of setup.

$$D = R_{32} \div \left(R_{32} + \left(\frac{1}{R_G} + \frac{1}{R_{11} + R_{31}} + \frac{1}{R_{12} + R_{32}} + \ldots + \frac{1}{R_{44} + R_{34}} \right)^{-1} \right)$$

Equation 4.3

We see that although change in the interesting resistance R_{32} certainly would cause the biggest impact for the measured voltage, the other resistor values could potentially cause big shifts in the voltage read.

4.2.3 Single signal pull-down resistor

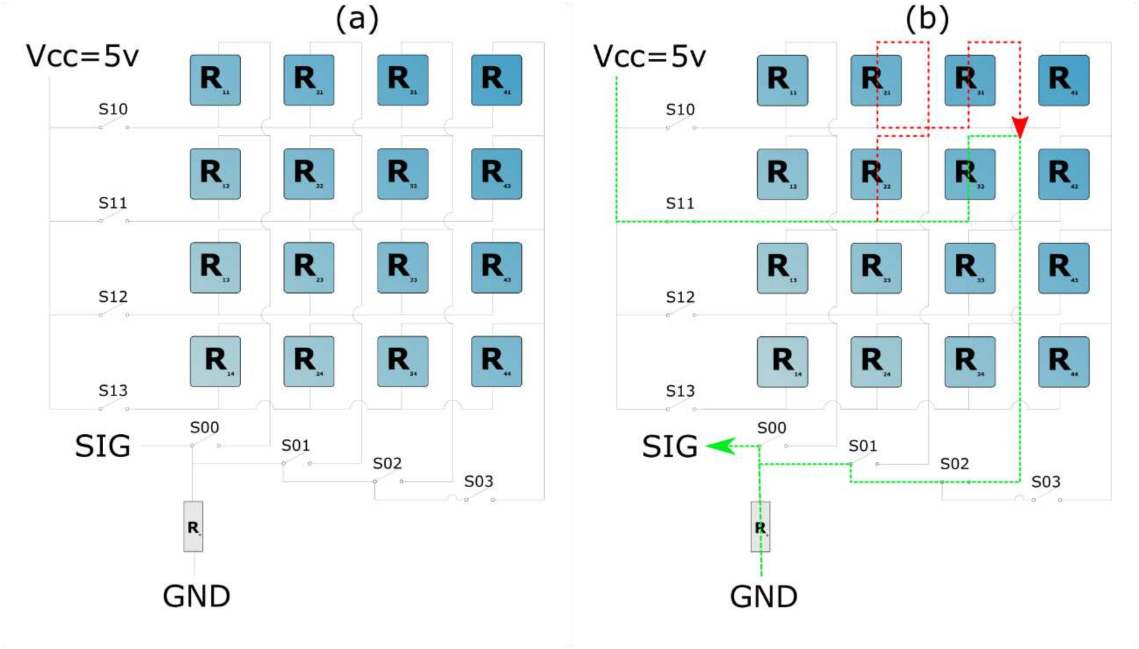


Figure 4.6 – 4x4 Matrix sketch with single pull-down resistor. (a)Multiplexer matrix with single pull-down resistors on signal pin (b)The desired measurement path and a possible error path

Another possibility as I am using the same value of resistors for the voltage divider of all columns, is to leave the resistor behind the multiplexer gates at the signal output wire of the multiplexer. Since none of the columns are pulled down to ground, when their gate is closed we no longer have the problem of being pulled down through two resistor pairs. The next lowest order of resistors we can go through to get unwanted readings is three, but now we are pulling the signal up and not down. To describe the dimensions of the error we can write out the error paths and find a relationship, *D,* as before:

$$D = \frac{\left(\frac{1}{R_{32}} + \frac{1}{R_{12}+R_{11}+R_{31}} + \frac{1}{R_{12}+R_{13}+R_{31}} + \frac{1}{R_{12}+R_{14}+R_{34}} + \frac{1}{R_{22}+R_{21}+R_{31}} + \ldots + \frac{1}{R_{42}+R_{44}+R_{34}}\right)^{-1}}{\left(\frac{1}{R_{32}} + \frac{1}{R_{12}+R_{11}+R_{31}} + \frac{1}{R_{12}+R_{13}+R_{31}} + \frac{1}{R_{12}+R_{14}+R_{34}} + \frac{1}{R_{22}+R_{21}+R_{31}} + \ldots + \frac{1}{R_{42}+R_{44}+R_{34}}\right)^{-1} + R_G}$$

Equation 4.4

This simplifies to:

$$D = 1 - \frac{R_G}{\left(\frac{1}{R_{32}} + \frac{1}{R_{12}+R_{11}+R_{31}} + \frac{1}{R_{12}+R_{13}+R_{31}} + \frac{1}{R_{12}+R_{14}+R_{34}} + \frac{1}{R_{22}+R_{21}+R_{31}} + \ldots + \frac{1}{R_{42}+R_{44}+R_{34}}\right)^{-1} + R_G}$$

Equation 4.5

As the variable resistances are of similar size, we see that for a given sensors, such as R_{32} in this example, the sensors value would affect the total reading much greater than other resistor values. The influence of the neighboring resistors has decreased, as compared to chapter 4.2.2, and the error caused by neighboring sensor-readings would be much more similar for every sensor at this point and could be thought of as more of a globally effecting mean over the sensors. As we actually possess the values that govern this error it would be fully possible in the future to remove the error in the processing of the data, making it more of a calibration issue. However due to the low processing power of the Arduino this is not something I will attempt to tackle at this point and will instead accept that there is some crosstalk between the sensors, but that the main sensors in question will be the most influential for every reading.

4.2.4 Diodes

Although there might be other circuit designs that would eliminate errors from the neighboring sensors completely, the easiest solution I was able to come up with was using diodes to govern the direction of the current through the sensors. All "wrong" sensor paths would in this case go against the diodes, making the desired measurement path the only closed circuit. Adding diodes to my flexible sensor skins was not really an option. It could potential effect flexibility and connections. It would also increase the over-all complexity of the structure and the manufacturing process would become much more complex.

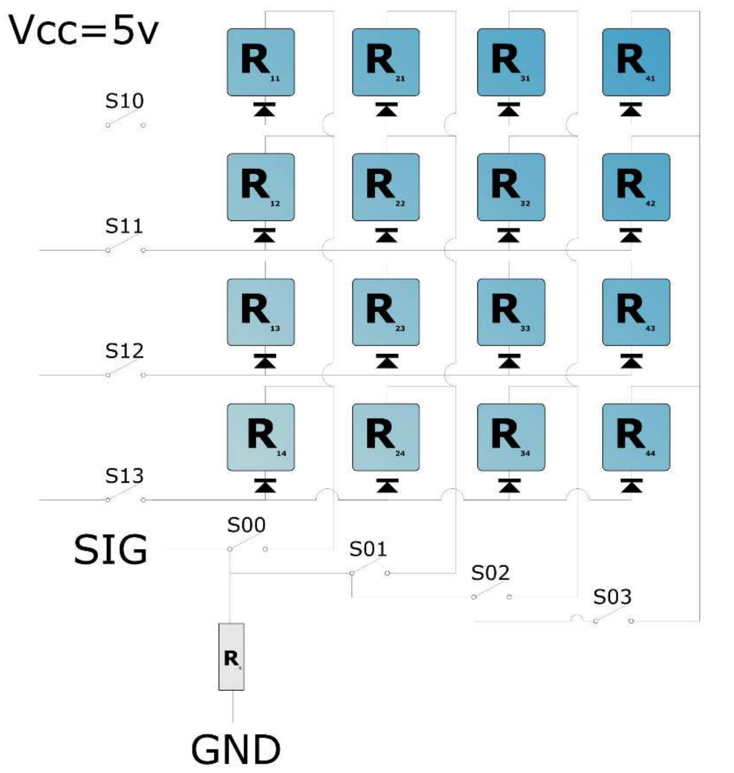


Figure 4.7 - Variable resistor 4x4 matrix with diodes

4.2.5 Skin blank

To make the skin into which I will later cast the sensors I first cast the skin blank with hollow pockets to fit both copper-tape leads at the back of the sensor as well as the sensors themselves, while still covering the back of the skin so that the sensors and leads are insulated. This was done by making a negative of the shape in acrylic sheets in the laser cutter. The molds total height was 3mm while the regions to be hollowed were 2mm so that the back got sealed with about 1mm of silicone when the mold was filled completely.

4.2.6 Copper-lead and separator

To get a back connection on all the sensors, I laid down a piece of thin 5mm thick copper tape on the back of the hollowed skin-pockets, covering a large area of the back should ensure even contact, even with large amounts of movement. With the lead place, I inserted pieces of 10x10x3mm acrylic blocks as place-holders for the sensors pads, as seen in Figure 4.8. I then filled the channels between each sensor pocket with silicone rubber to form separators between each sensor. These aid both in dampening between the slightly stiffer carbon reinforced silicones as well as ensure no electrical contact between the sensor-pads, while also holding the copper-tape leads in place during production.

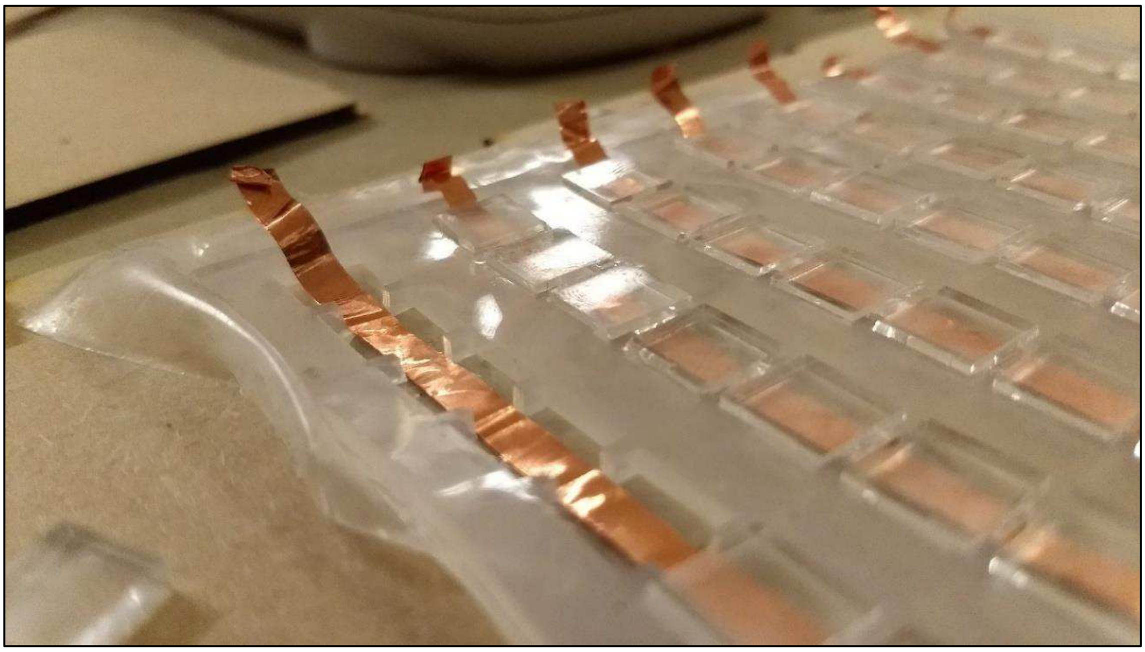


Figure 4.8 - Copper tape lead in hollow pocket of skin blank with acrylic sensor place-holders

4.2.7 Sensors

Following the production methods from previous experiments, I cut woven carbon fiber fabric into short strains of carbon fiber, about 2-10mm in length, trying to keep them as short as possible. I mixed this with the rubber silicone to a thick yet liquid consistency, about 4*percentage by weigh (*wt.%). The skin casting was made with pockets designed to be 10x10x3mm. I cast the silicone rubber-fiber into long rows to get as even distribution of fibers as possible. Note that if some of the fibers cut are considerably longer it will cause the fibers to clump together very easily. I had some clumps forming due to longer fibers and tried to remove these to the best of my capabilities, although some uneven distribution of the fibers was imminent. I cast 8 pieces of sensors, 100mm long so that I would be able to pick out the most suitable areas and pieces to make the sensors from.

When the pieces were cured I cut them to size, with scissors every 10mm. I avoided areas with low infill of carbon fibers and areas with a lot of air bubbles. When test-fitting the sensors into the sockets in the skin-casting I came to realize that the sensors were slightly over sized as the silicone slightly swell after being removed from its mold, and the laser-cut plug to create the hollowed pockets in the casting must have been cut slightly under-sized due to the lasers unfocused radius. Although inserting the oversized sensors into the skin ensured good mechanical bonding between sensor and skin, it also caused bulging of the skin, as seen in Figure 4.9, as more material was added to one side. I used scissors to cut each sensor down 1-2mm in size as a quick fix, getting straight and uniform cuts with scissors was hard causing the sensor-pads to slightly differ in size and shape.

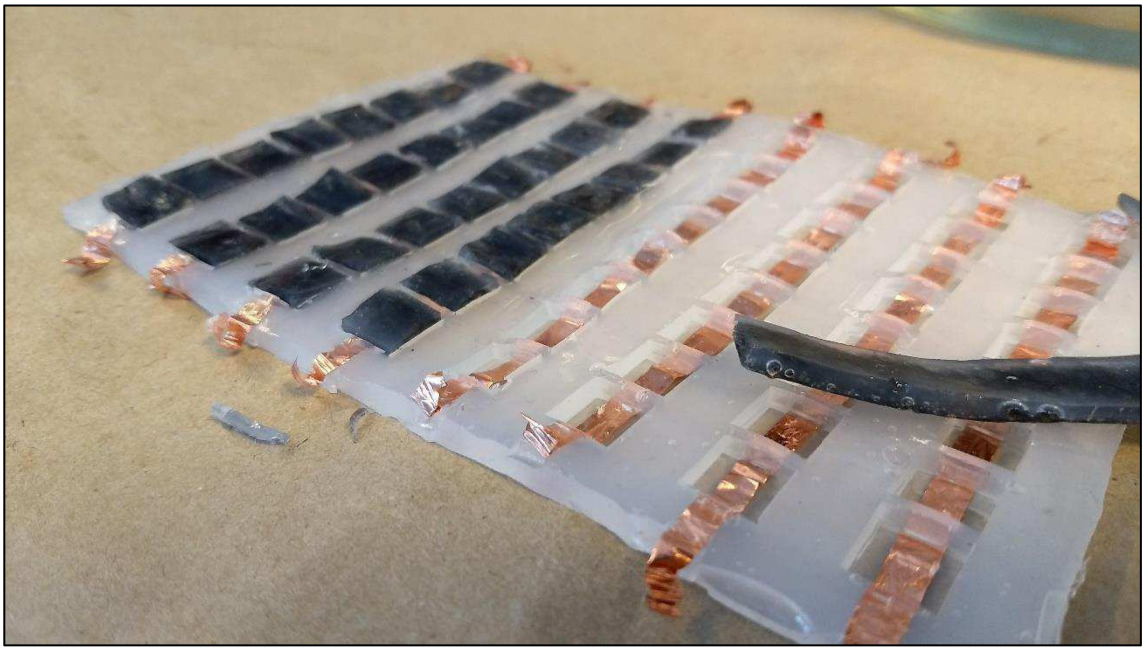


Figure 4.9 - Over sized sensor pads in skin

4.2.8 Flexible lead sewn in

With the sensors in place in their sockets I added the top lead to each sensor in lateral direction. While copper tape is able to bend, it is not able to flex. If the skin were to be attached to a foil, changes in leading edge, shape, AoA and *flaps* would all potentially stretch and compress the skin in lateral direction, posing a need for stretch ability in the leads going laterally. To overcome this, I used *Adafruits 3ply stainless steel conductive thread*, meant for smart clothing. The thread offers high flexibility and repeated bending with low risk of work hardening the thread and cause breaking, which would be an issue with normal wires. To enable stretching between the sensors I made a loop, leaving excess thread between the sensors. The threads were attached to the top of the sensor pads by threading the thread through a small portion of the top of the sensors. To get the threads to lay flat, I used small daps of superglue on the thread, making sure to curl the threads so that they would not touch neighboring threads.

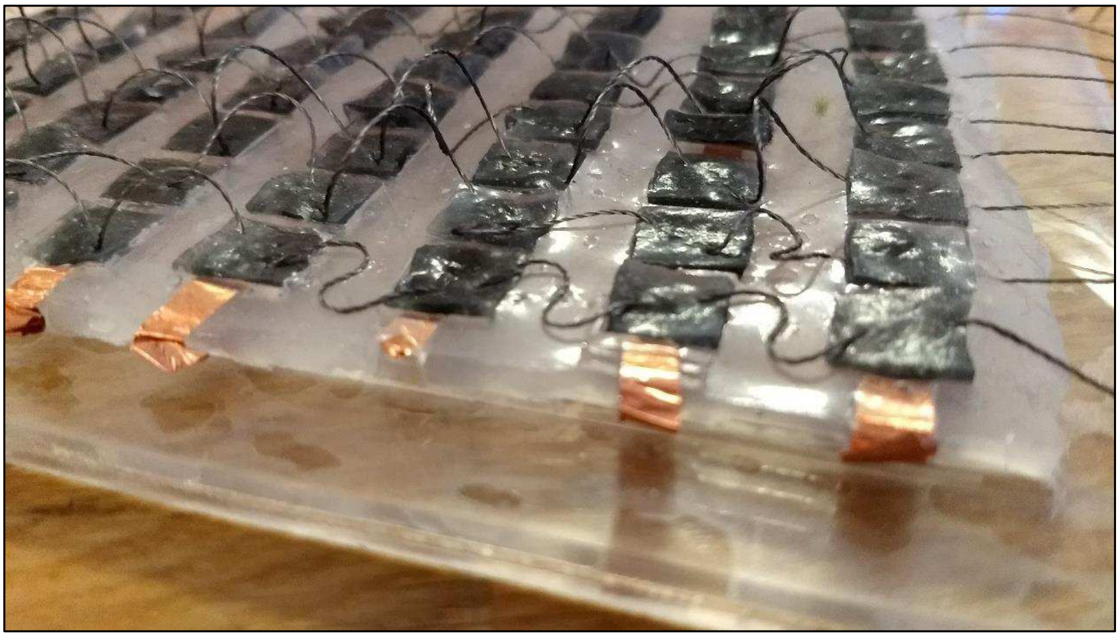


Figure 4.10 - Sewing in flexible leads and gluing them down

4.2.9 Enclosing, wiring and multiplexers

When all connections were sewn correct, I cast an additional millimeter of rubber silicone on the top of the skin to enclose all sensors and wires. The copper tape leads could be soldered directly to wires using generous amounts of flux. The conductive threads however were not solderable, so to connect them to wires I twisted them together with the wire and made a small knot. I then used heat shrink tubing with internal glue, to fix everything in place. This seemed to hold up nicely. The wires were then attached to two multiplexers that I had soldered to screw connectors. One of the multiplexers had its signal wire connected directly to the 5v Vcc, while the other had its signal pin connected to the analog signal cable through voltage dividers as described in the previous sub-chapter 4.2.2. For the flat skin I used separate resistors for each column when testing, aka. one voltage divider for each multiplexer signal gate.

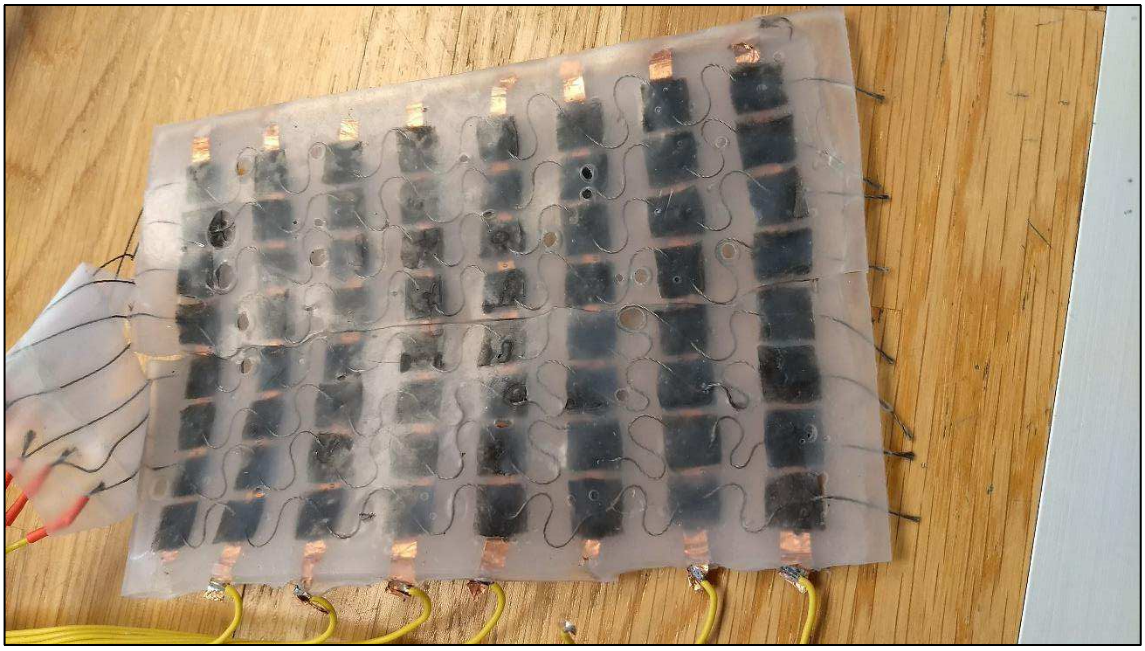


Figure 4.11 - Finished sensor skin

4.2.10 Heatmap

Like earlier, making simple codes on an Arduino with two 8x8 for-loops it is possible to switch between the sensors and read their values one by one and print this as comma separated values that can be displayed in Arduinos built in serial plotter. Arduino codes for writing and calibrating sensor data from the skins can be found in **appendix A**. This offers a good visualization to verify that the sensors work. It is also very visually helpful to be able to see the changes in the sensor data over a small period in time, and I used it frequently during testing of the sensor skin. But when we have 64 sensors it is no longer practically possible to distinguish the different sensor values from one another, which means that even though general trends can be seen in the serial plotter very clearly, it's hard to make any sense of what is actually going on locally for each sensor and areas on the skin. Perhaps a better way to display the information is through a live heatmap, highlighting areas with high and low sensor-readings.

I initial made a script in MATLAB, to read the sensor information of the Arduino and plot it as heatmaps. It however turned out that both the built-in reading and plotting functions of MATLAB were far slower than expected and was only able to plot results every 5-10 seconds. Although tweaking the way the data was sent and processed certainly might have yielded more usable plotting times I instead opted to use a more purpose-built programming ide to plot the Arduino data; *Processing*. Processing is java based and its IDE feels very similar to that of the Arduino programming environment and is commonly used to make *graphical user interfaces* (GUI) on the computer for Arduino projects. I modified codes by processing.org users quark

(2016) and MitchR (2018) to fit and be able to read my comma separated values and plot them as a 8x8 color matrix. To make a smoother plot I also experimented with some different interpolation algorithms, the processing code can be found in **appendix B-1**. This greatly increased the plotting frequency. With interpolation it can be easier to see the entirety of the flow picture, but some of the finer details of the individual sensor data becomes less apparent.

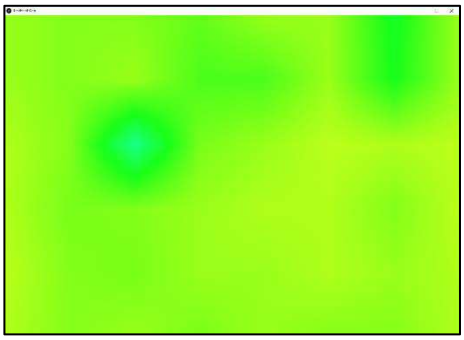


Figure 4.12 - Heatmap plot with interpolation between datapoints

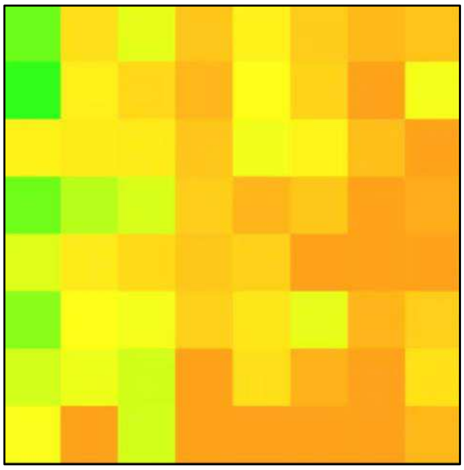


Figure 4.13 - Heatmap plot with no interpolation between data points

4.2.11 Testing and results

To verify that the sensors and codes were all working properly I initially submerged the skin in a shallow pool of water that I stirred with my fingers. I tweaked the draw values in the processing code until I got decent color-variations in the heatmap for the stirred flows. I then submerged the skin in my water tunnel and ran it through the different flow scenarios that I am able to create to see if I was able to spot any patterns in the real time heatmap. I alternated between laying the skin flat at the bottom of the tunnel and holding the skin vertically with the length of the skin going in the direction of the flow. To see if the pressure patterns differed significantly.

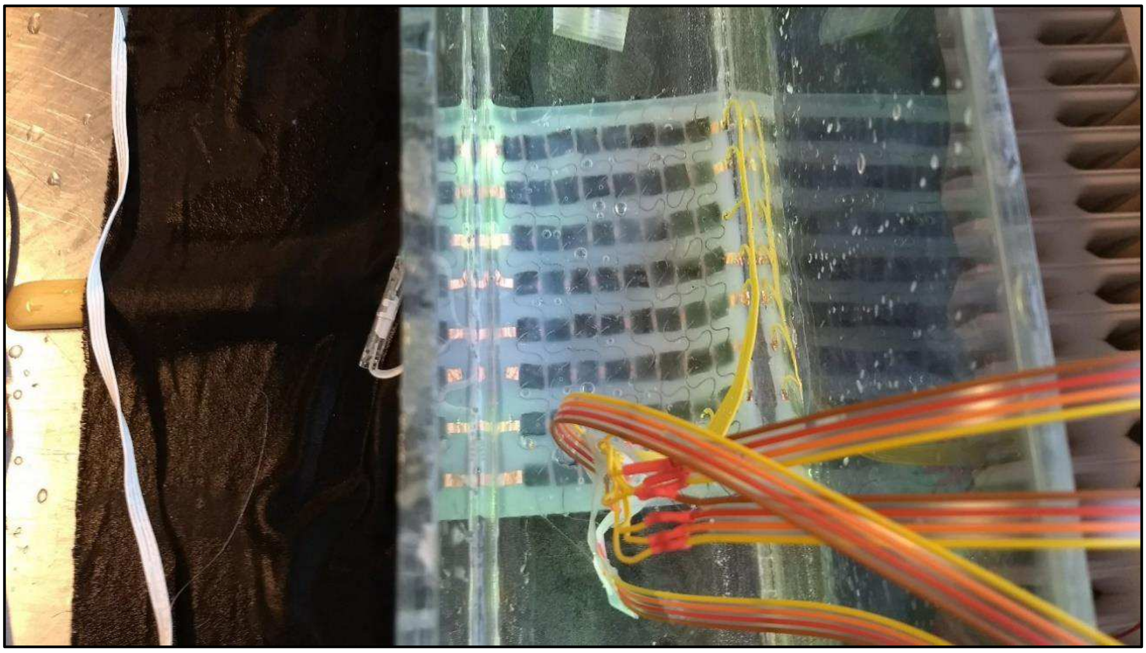


Figure 4.14 - Sensor skin laying flat in vortex street

The exerted forces in the water tunnel were far lower than those that I had tweaked the draw function with initially, causing very low contrasts in the live plots. Visually it could look like the movement of the high and low-pressure zones when the skin was laid flat follows some sinus-like wave pattern from right to left, as seen for a 12v Kármán street in Figure 4.16. When suspending the skin vertically the pressure zones looks as if they move more line-wise, yet these results are very unclear, as trying to keep the skin straight by hand induced high pressure readings of its own. The vertical Kármán street results can be seen in Figure 4.17 while a control with free flow at 12v and the skin laid flat can be seen in Figure 4.15. Although my result bias might be helping me see what I want to see, there seems to be very real pressure readings from the water that moves across the skin, as can be seen in the difference from the readings of when the water is still and moving. Recognizing flow changes should be possible by implementing the skin into a foil.

Figure 4.15 - QR-Video link showing non-altered flow at 12v. With sensor sheet laying flat at bottom. https://youtu.be/YX6xIjkhgeg

Figure 4.16 - QR-Video link showing Kármán street flow at 12v. With sensor sheet lying flat at bottom. https://youtu.be/nUwAlPspM_Q

Figure 4.17 - QR-Video link showing Kármán flow at 12v. With sensor sheet held vertically in direction of flow. https://youtu.be/SLWck_gpmgw

4.3 Skin to hydrofoil

Having showed promise in distinguishing different flows as a separate sheet, I wanted to create foils with the same sensors. With more interesting surfaces for the vortices to hit and interact with, the pressure readings over a foil might prove more interesting than for a flat sheet. As the sensor-skin-making process was extremely time consuming with multiple casting steps, I wanted to re-use the skin and re-cast it into a foil. I made molds to cast the foil shape, with a hollow core, to accommodate an internal, actuated skeleton like the one used to make my animatronic trout in my project thesis. I decided on a rough skeleton shape so that I could make attachment slots in the skin for the skeleton. An area in the middle of the foil was left without slots, so that the sensor skin could sit flat. I designed the thickness of the skin to equal that of the sensors skin at 4mm. I cut the sensor skin in two and attached new wires to the separate parts. I used small dabs of cyanoacrylate glue to hold the sensor skin to the inner core of the mold. The outer foil mold was designed from a top down trout drawing. And made wide enough to fit electronics on the inside. I cut the outer mold in two pieces so that it could attach it from both sides of the inner core. I used vacuum tape to seal the molds together.

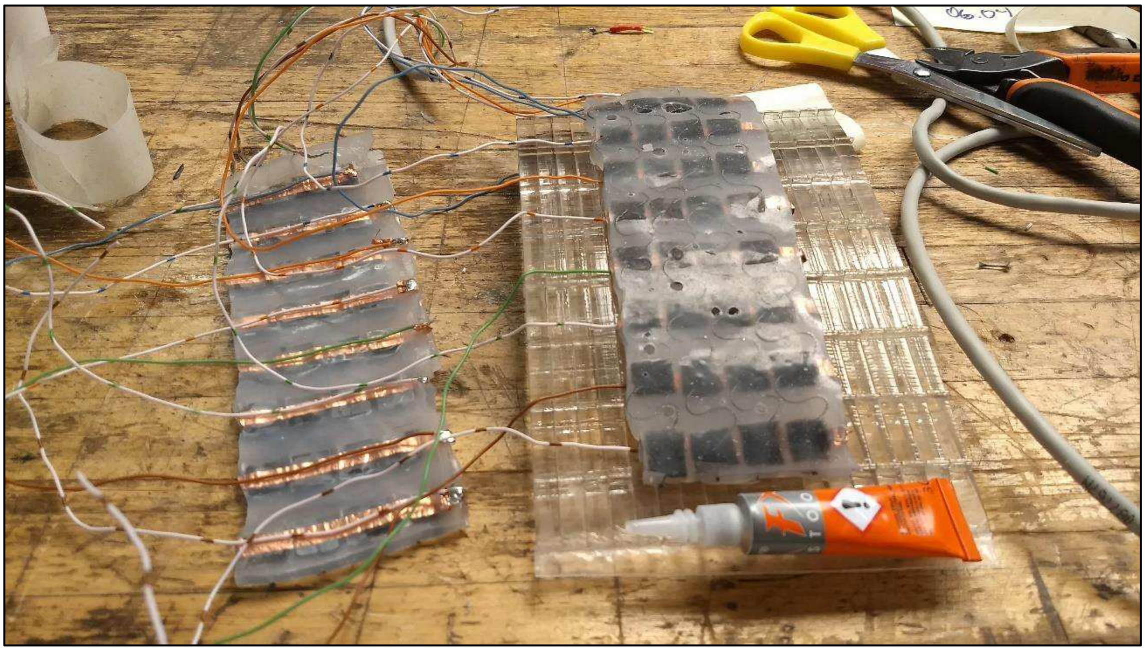


Figure 4.18 - Pieces of sensor skin attached to inner core of mold

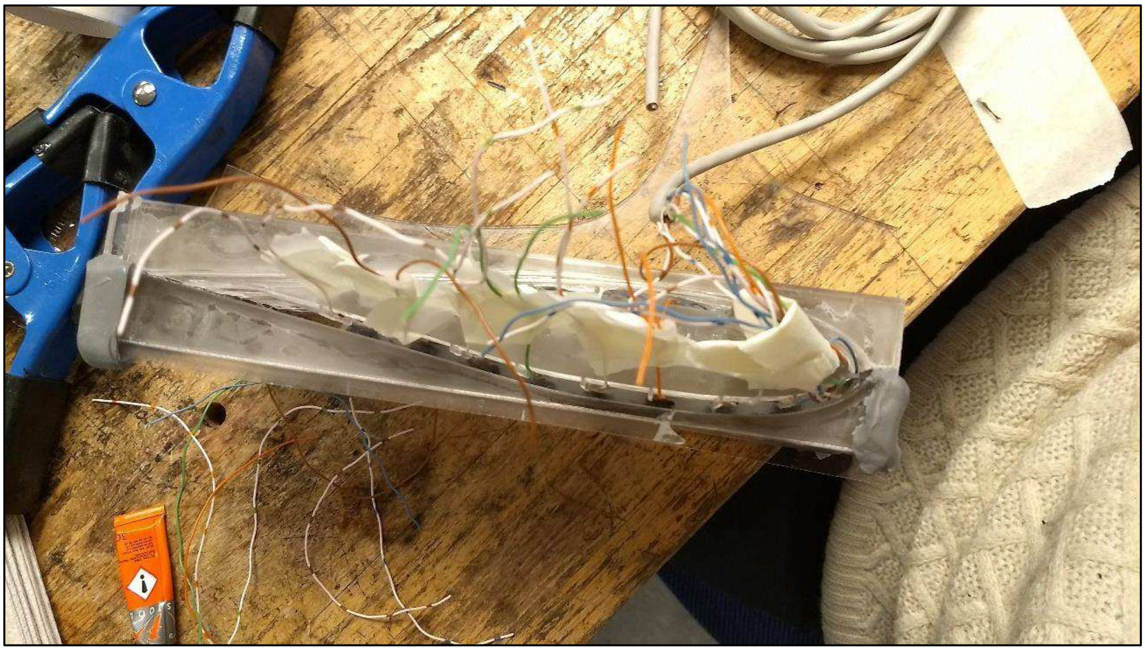


Figure 4.19 - Fully assembled mold ready for casting

As earlier I used the *ecoflex 00-30 rubber silicone* so that it would adhere better to the existing sensor skin and form a homogenous matrix. To remove air bubbles, I used an off-set weight on a drill to vibrate the mold.

4.3.1 Inhibitors

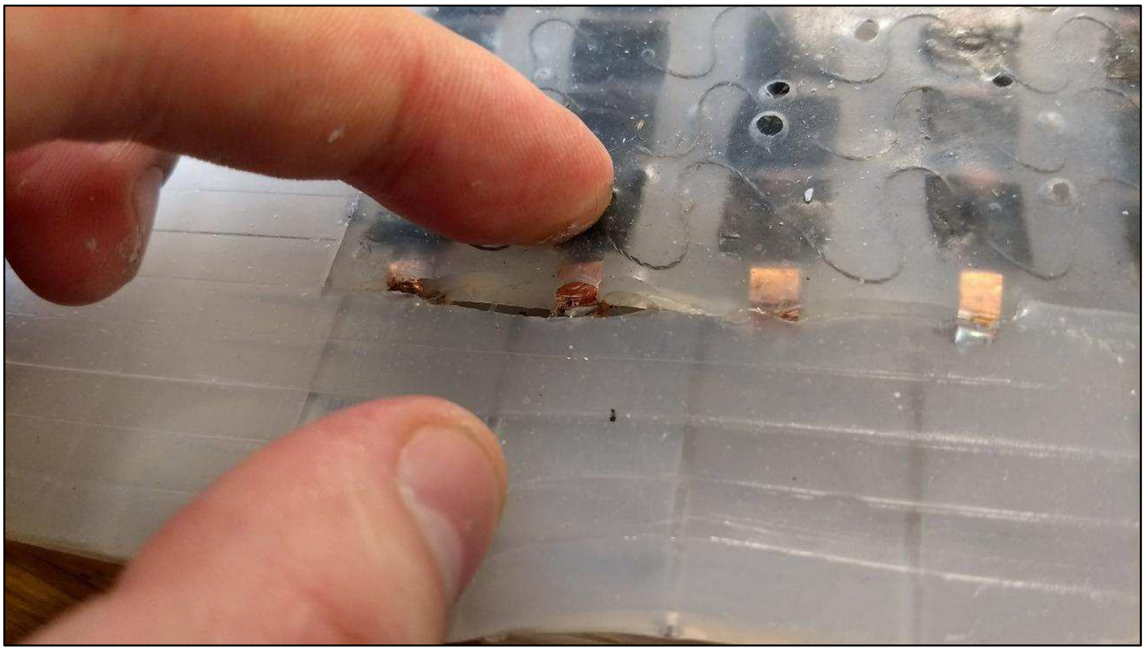


Figure 4.20 - Gap between sensor skin and foil body where silicone didn't cure

When casting rubber silicone, a big problem is inhibitors. Inhibitors are materials that prohibit the curing process of the rubber silicone. When casting around my old sensor skin I ran into big problems due to inhibition especially around the areas where I had previously soldered. The manufacturer does not list all inhibitors for this specific rubber silicone, and rater a small-scale trial and error approach through testing is recommended. From experience, hydrophobic release agents such as grease and petroleum gel will often cause cure inhibition in silicones and is probably what caused the biggest problems with my sensors skin, as I had used soldering fat on the copper tape to solder it. Vaporized fat and flux from the soldering wire could have contaminated the silicone along the edges and be a possible cause for the poor curing and adhesion between the new foil and the old skin. To try to fix the big gaps in the skin, I used generous amounts of degreaser and acetone to clean all areas where the silicone had not set and applied more silicone with syringes. Some areas were able to adhere and cure, but many would still struggle, and only small bridges of cured rubber silicone would form. This made the skin very flimsy, and not very stretchable as it would easily tear. The skin would also not stay water-proof which means that muscle wires could not be used on the inside, as I did in my project thesis.

4.3.2 Results with broken foil

With a highly reduced foil skin, I decided to create a new and more correctly produced skin, but with long curing processes in the production this left time to simultaneously test and play with the existing foil in the water tunnel. To submerge the foil into water, I cast a thicker and longer lip on top of the foil, so that I could cut a larger hole to insert an inner structure into the hydrofoil. The construction of the inner structure is described in chapter 4.7. The foil however preformed extremely poorly. Although the sensors were sensitive enough to feel water pressure, the skin was not stretchable enough for the inner structure to be correctly inserted. Multiple connections in the skin were broken in the copper tape. And with high buoyancy and no structure to hold the foil down, the plotted data was highly influenced by the physical holding down of the foil.

These tests, although poorly executed, helped me realize some areas to improve in the new foil; the foil needs a good mechanism to hold it down, while being able to change the yaw of the foil, either physically holding it down or weighing it down. The inner structure needs a better plan for insertion and copper tape should not be used as leads as it easily breaks.

4.4 Direct sensor production into foil body

After spending unreasonable amounts of time trying to fix and correct the previous foil through re-casting degreasing and cleaning I decided that the more rational approach would be to redo the process from the beginning with the end foil-shape in mind. That way I could reduce the number of steps needed, keep everything clean until the end and hopefully get much better and consistent casting-results, as well as more consistent sensors.

4.4.1 Molds

To keep the re-work to a minimum I kept the basic concept of the molds from my previous attempt, with inner structure for a moving skeleton and a 4x8 array as lateral lines along each side. The sensors were positioned from the front of the fish and distributed evenly towards the tail, ending 2cm before the two sides join at the tail, so that space was left for any wires the sensors might need later. Due to the copper-tapes tendency to break I opted to remove it completely. The conductive flexible threads had given me very little problems and seemed like the best replacement choice. That way I could reduce the number of casting steps as well, as the leads could be sewn in post-casting, rather than being cast into the structure. I decided to make the new sensors 6x6x3mm. That way the rough mold of the foil could be constructed simply with layered 6mm acrylic sheets in the laser cutter. I designed the mold as before, but with a split about the middle. I also added the lip on top of the hydrofoil in the first mold, for easier installation of the inner structure. I made the mold with 2mm holes through them so that I could insert locating pins/dowels through all layers and ensure alignment. I also made an outer rim/skirt with holes for screws and nuts, so that pressure could be applied evenly to avoid leaks. I used 1mm rubber to cut a gasket to size to fit the skirt. After having glued the whole mold together with cyanoacrylate glue, I filled it with water to look for leaks. I found that the mold was not nearly as waterproof as I had hoped for. With leaks around all sides, I decided that it was better to seal the whole mold than just small areas. To achieve this, I roughed up the outer surface of the mold and added two layers of glass fiber and epoxy resin to both sides of the mold. I cast the glass fiber without vacuum and used metal weights and release-film to hold the shape during casting.

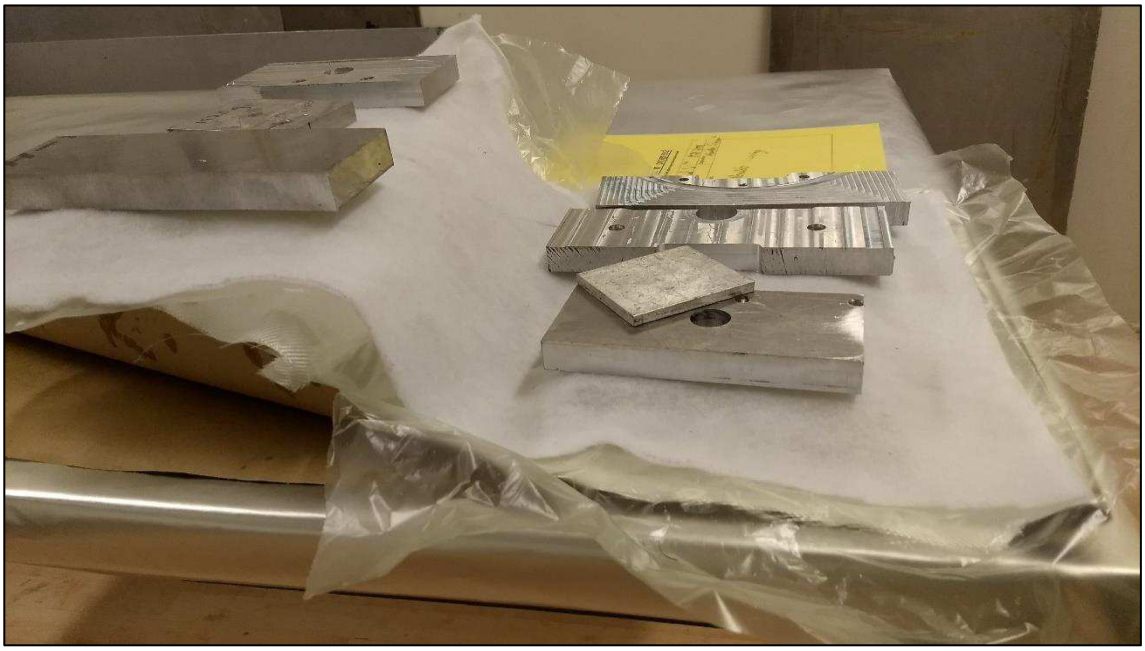


Figure 4.21 - Casting outer glass fiber-epoxy shell to mold. Fiber is covered with release film and breather material before weights were added.

I then tested the mold again with water. I sealed of any small areas where water would leak with cyanoacrylate glue, and a couple of larger areas were sealed making a composite of cyanoacrylate glue and *baking powder*.

Once the mold was confirmed to be waterproof, I dried it out and mixed up ecoflex 00-30 silicone as before. Before pouring into the mold I degassed the silicone in the small jar-vacuum chamber to ensure good flow into the mold and less air bubbles in the casting. I did this in three sessions to make the degassing faster. I also vibrated the mold for 2-3 minutes between each pouring with a drill with an offset weight to help any remaining air-bubbles surface.

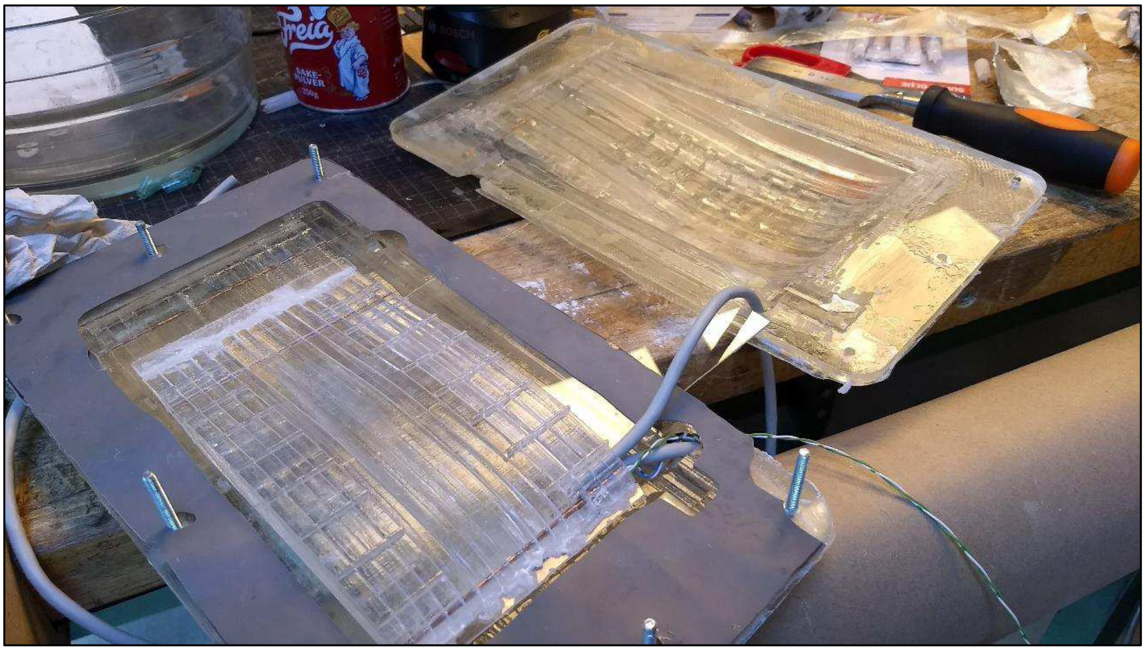


Figure 4.22 - Mold, inner core, gasket and locking screws before assembly and casting.

The quality of this casting was far superior to the previous castings, where less care had been taken. I covered the foil with cling-foil to ensure that it would not be contaminated and left the inner core in the mold so that the shape would be kept for the remaining steps.

4.4.2 Sensors

The hollowed pockets to hold the sensors were designed to be 6x6mm and 3mm in dept. However, taking swelling of the silicone into account and the fact that the laser cutter also burns some extra material, I made the sensors 5x5x3mm to more easily fit into the sockets. As earlier I fist attempted to cast the sensor-pads individually with small cut 5x5mm squares in 3mm acrylics. Previously I had not been able to pour the silicone-fiber mixture into the individual molds, so to counter this I mixed the mixture to a much lower viscosity than I had previously tried, about 2-3wt.% carbon fires. To ensure that the silicone mixture would fill the smaller molds, I cast it inside a small vacuum chamber and hand pumped the vacuum. Once cured, I tested the sensors for consistency both in shape and conductivity. I found that the vacuum chamber had not been sufficient to remove air pockets from the sensors. And the 2-3wt.% was not enough to get good conductivity over the sensors. I was not able to measure a low resistance over any of the sensor pads and visually it would seem that the fibers had not been able to get to the bottom of the small pads, and that the sensors were completely insulated on one side. Measuring on the upper side of the pads, I was able to get connections on and off, but the sensors were much less consistent than any of the previously tried mixtures. A higher weight percentage should probably be used for future sensors.

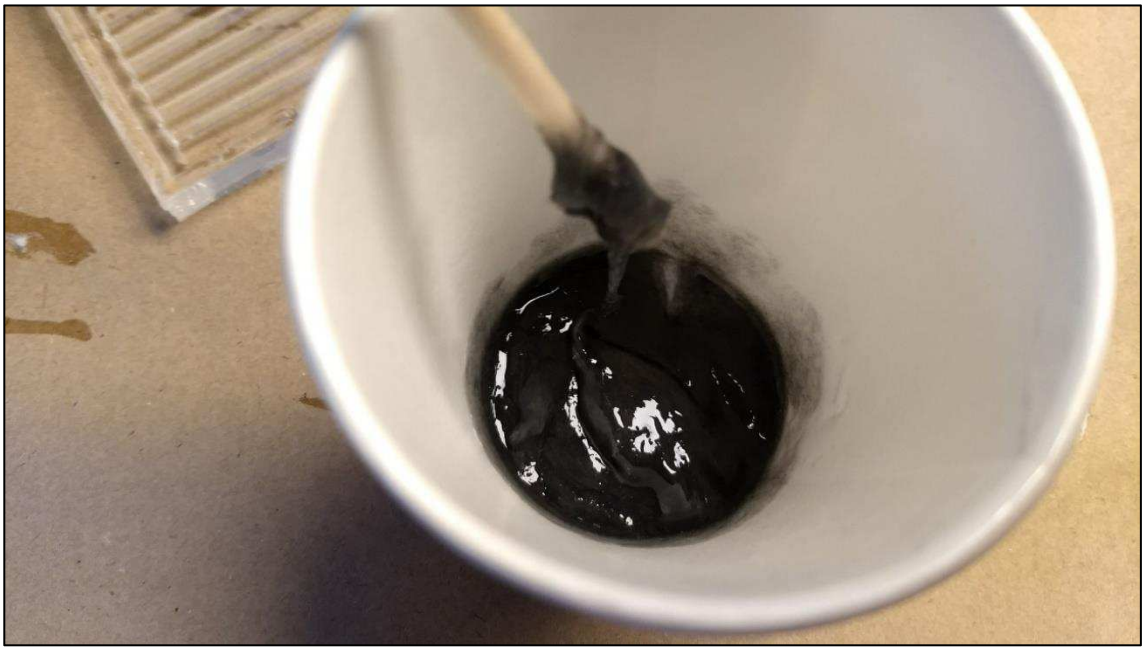


Figure 4.23 - Consistency pre-casting of the failed sensor silicone-fiber mix

Wanting consistent size of the sensors I attempted other production methods. Previously I cast longer lines of silicone-fiber, which I then cut to the correct lengths. But as silicone is not the most forgiving material to cut straight, it was hard to get consistently big sensors. The concentration of carbon fibers in the strips would also wary a lot, as getting consistent concentrations of the fibers along edges was difficult. To circumvent this, I decided to cast the sensor material as one big piece, so that the middle of the piece could be cut to correct sizes and used as sensors, hopefully making them much more consistent in composition. I carefully followed the previous casting procedures, making sure to degas the silicone before mixing in the carbon fibers. I increased the amount of carbon fiber to 4-6wt.% to a paste-like consistency. I made a 100x100mm open square in 3mmacrylics with sheets on both sides to act as the mold, with no open top, the sensor sheet would get a more consistent height along the whole surface. I used a thin rod to force the paste into the mold and force out any obvious air bubbles. The viscosity of the silicon-fiber paste at this point was way too high for vacuum-degassing. The resulting casting was a nice even sheet with a flat, reflective surface. The carbon content appeared to be consistent throughout the sheet, and I was able to measure resistance in the 200-1000Ohm range between points along the whole sheet.

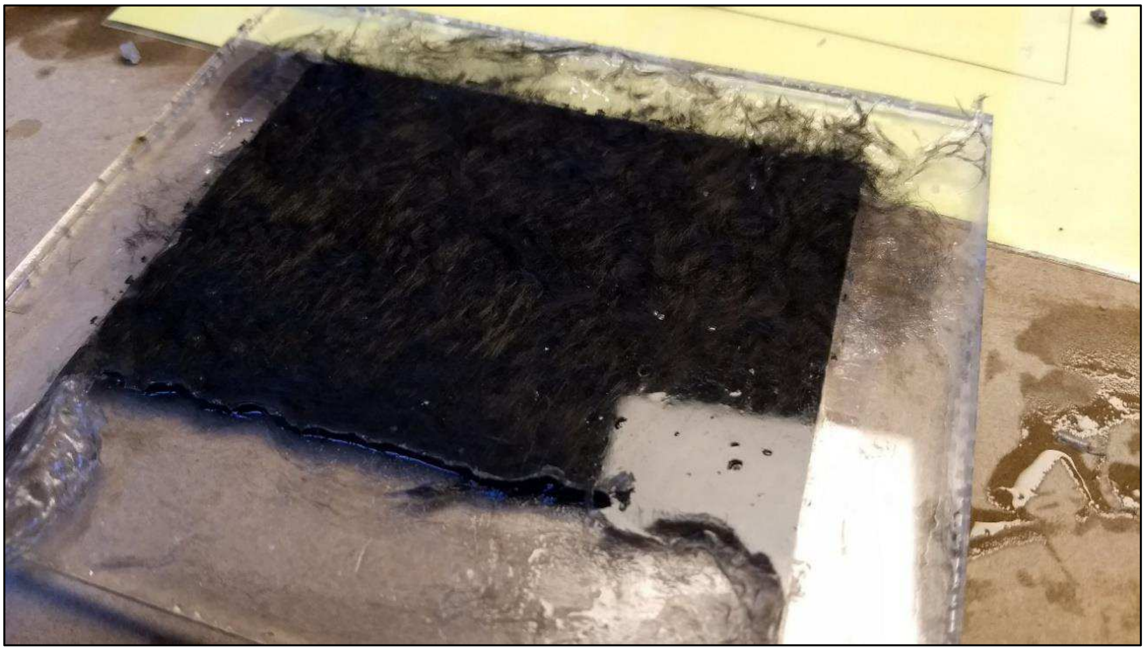


Figure 4.24 - Sensor sheet during casting

Due to the small size, and the difficulty of hand cutting silicone straight, it would be beneficial to let a machine cut the sensors to size. I tested a small piece of carbon fiber-silicone in the laser cutter and found that it was able to easily cut through both the silicone and fibers 1-2mm deep, giving a good cutting guide with razor blade for the remainder. I cut the first 35mm of the sheet into 5x5mm squares with the laser cutter. I then used sharp razor blades pressed straight down into the track left by the laser to cut the remaining material. I cleaned up the sensor-pads in isopropyl alcohol to remove most of the burned and charred materials from the sensors edges.

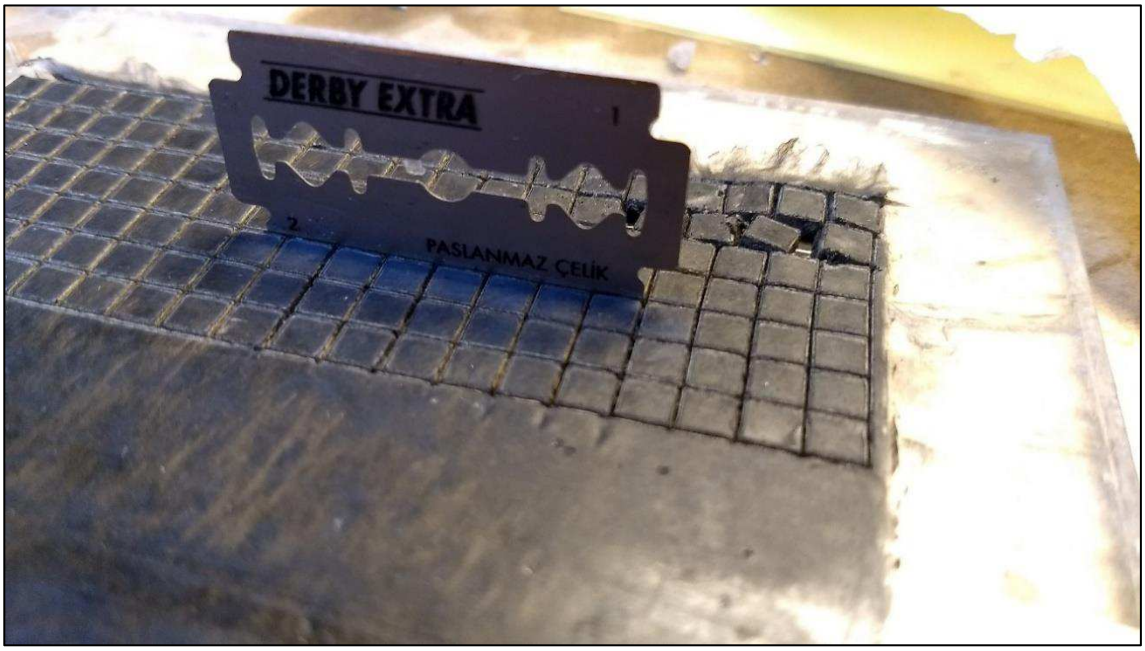


Figure 4.25 - Cutting sensors by following the laser cut path

4.4.3 Wire sewing

As earlier, the wire leads were installed in a 8x8 pattern for multiplexing. To allow the material along the lateral axis to stretch, some extra lengths of wire was needed between each sensor. I found that using a needle and pliers to press down the silicone while the needle went through, was an efficient way to manipulate the path of the needle also depth-wise. To ensure minimum amounts of pollution to the silicone, I left the sensors out of the skin until I reached their socket. I also kept the silicone covered with cling foil in the areas where I was not currently working.

Sewing in the crossing lines for the back lead, was much less time consuming as this could be done in a straight line, because no stretching of the material is expected in this direction. To let the lead run deeper inside the silicone I sewed it in shorter segments coming out on the front and then re-entering the exit hole. That way the thread cut itself deeper into the silicone and became unnoticeable from the outside. To ensure back-contact with the sensor pads, I sewed the thread into the edges of the sensors, going from the side and down towards the bottom. I collected all the lines towards the front, by spiraling the wires as they were moving along the lateral line.

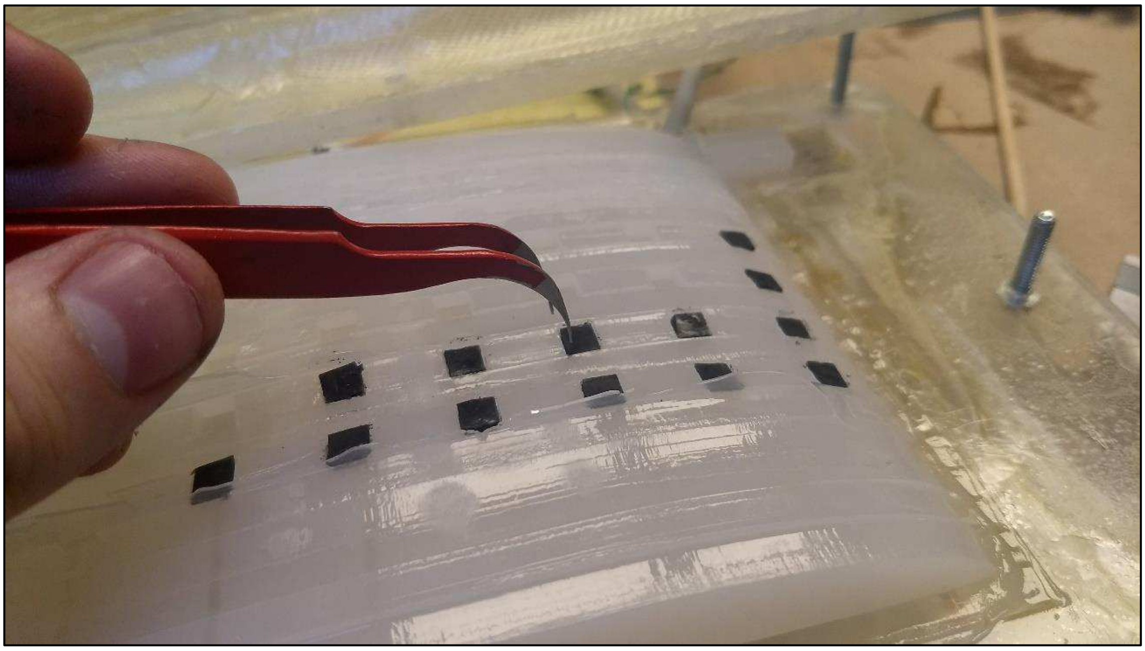


Figure 4.26 - Placing sensors into the sockets

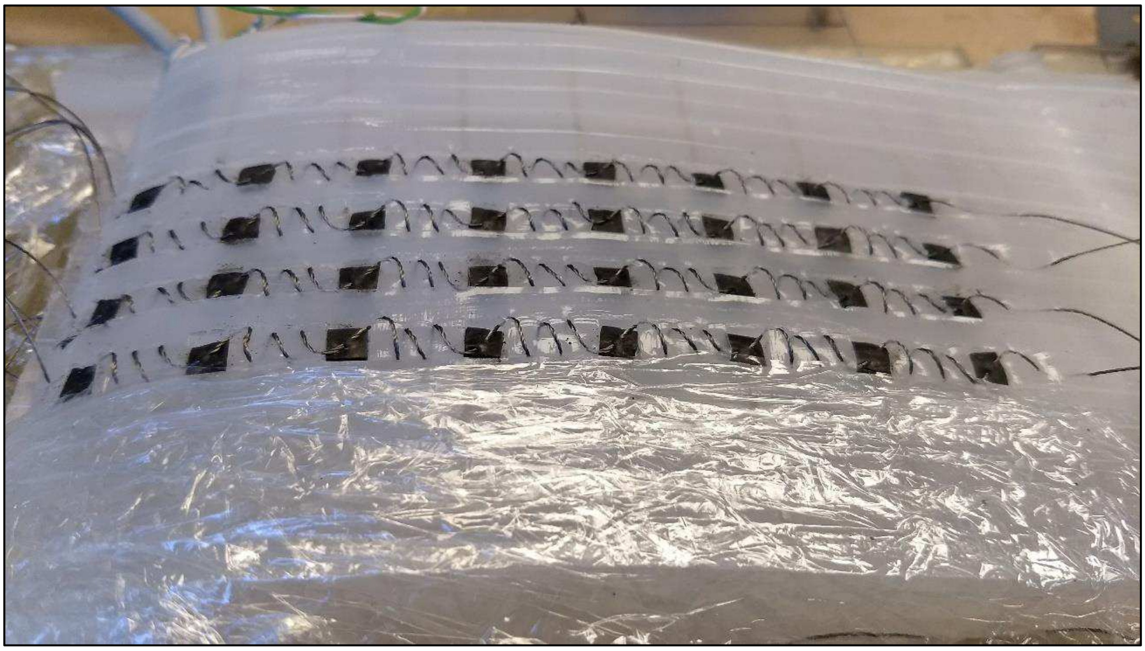


Figure 4.27 - Lateral leads sewn in

4.4.4 Outer mold

One of the weaknesses of the previous foil was that the end shape ended up being extremely lumpy, after several patching attempts. The lumpy surface would greatly affect the flows across the foil, making self-induced vortices and turbulences, breaking up the vortices and flows that we actually want to measure and making the output data less predictable. With the wires stitched into the skin and sensors along the surface, the surface skin of the foil is not exactly smooth.

The lines from misalignments during the acrylic casting is also very visible. The foil needs an outer layer to seal of the electronics from the water. Normally I've used layered molds made in the laser cutter, but to get a smoother result I decided to instead use a CNC mill to mill out the exact shape. I found some old piece of *polyurethane modeling board*, which machines very easily and leaves a smooth finish. Having previously worked with molds of polyurethane boards, they have proven to be very well suited also for silicone molds and should not cause any inhibition in the silicone curing, the only downside being the relatively high price of the material. I salvaged material from an old project, which had started to warp. As each side of the mold I was making was relatively thin the warp of the board would not matter as the whole piece could be milled flat. I made a fast *computer-aided design* (CAD) models of the shape of the mold I wanted to mill, using the dimensions form the previous mold, and giving them an off-set of 1mm, so that extra silicone could be added on the outside of the existing casting. I added holes for locating pins and bolts, so that I could ensure alignment between the two molds. As the mold only have curvatures in one direction I only did the finishing cuts along the curved direction of the mold and only along the curved surfaces. All straight surfaces were cut using larger steps to save time. The finished mold was smooth enough that very little extra work was needed. I lightly sanded the curves with 800 grit sandpaper, to further smooth any lines left from the milling process. I then used pressurized air to clean up the parts, before treating them with *Easy Composites s120 mold sealer,* to give the mold an even smoother surface, as well as blocking any pores which the silicone might get stuck to.

Test fitting the mold with the existing foil shoved that the 1mm off-set which I had used from the original mold was not enough to give clearance to the part, due to swelling. I therefore added a 2mm rubber-gasket between the two molds to achieve enough spacing for silicone to be able to flow into the mold.

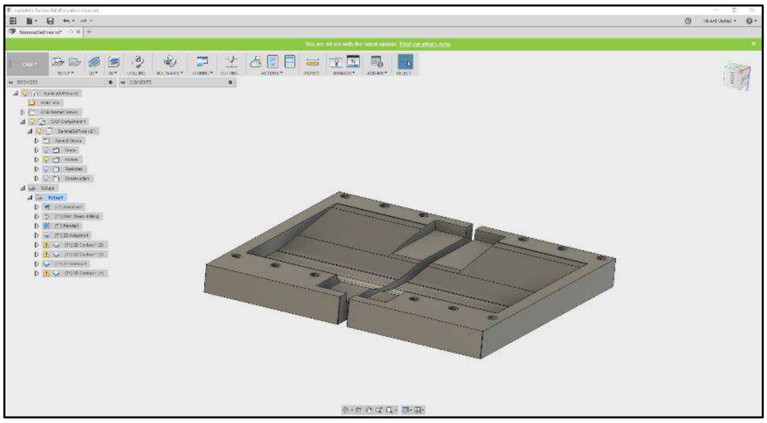


Figure 4.28 - CAD model of mold

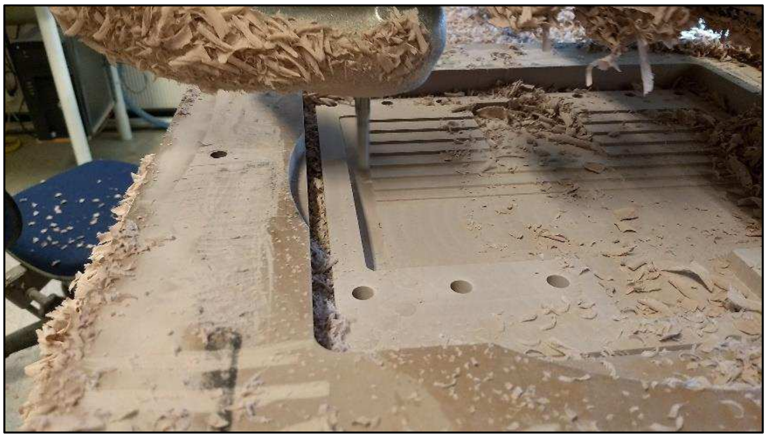


Figure 4.29 - Mold being cut

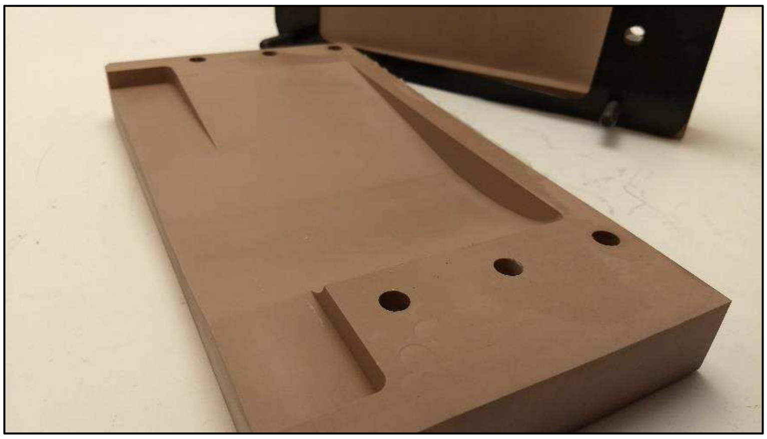


Figure 4.30 - Finished mold and mold surface.

4.4.5 Vacuum and degassing

From previous experience, the only way to achieve good casting results is through proper degassing and taking care that the molds are not contaminated. With the thin outer skin, achieving perfect results was especially critical and although degassing the silicone prior to pouring probably would be sufficient, I decided that the mold should probably be subjected to vacuum after pouring as well to ensure that the thin surface would be smooth with no air bubbles. To be able to do this I had to make a vacuum chamber that could accommodate the big mold of the foil. I used an old welded steel tank that had previously been used as a resin trap for vacuum infusion. The tank had some leaks in it, so to get a good vacuum I filled the bottom of the tank with epoxy resin and hardener to form a thin layer and seal any leaks in the weld from the inside. To get a proper seal along the top I used wet sandpaper to sand the top flush. I then used rubber gasket material to form a fitting gasket around the edge. I made the lid from laser cut acrylic plates of 6mm. I did a fast simulation of the static pressure acting on the lid in *Autodesk Fusion 360* finding the lowest expected safety factor across the lid to be 5.9 at complete vacuum for a 6mm clear acrylic lid. With cyclic pressurization and depressurization of the tank, I was not comfortable with only a 6mm thick lid and used *Acrifix* to glue an additional 6mm to the lid. I used a compressor from an old fridge to make the vacuum pump. The rubber silicone manufacturer recommends 29 inches of mercury for the ecoflex 00-30 rubber silicone, and with the fridge pump I was able to reach this in about 10 minutes.

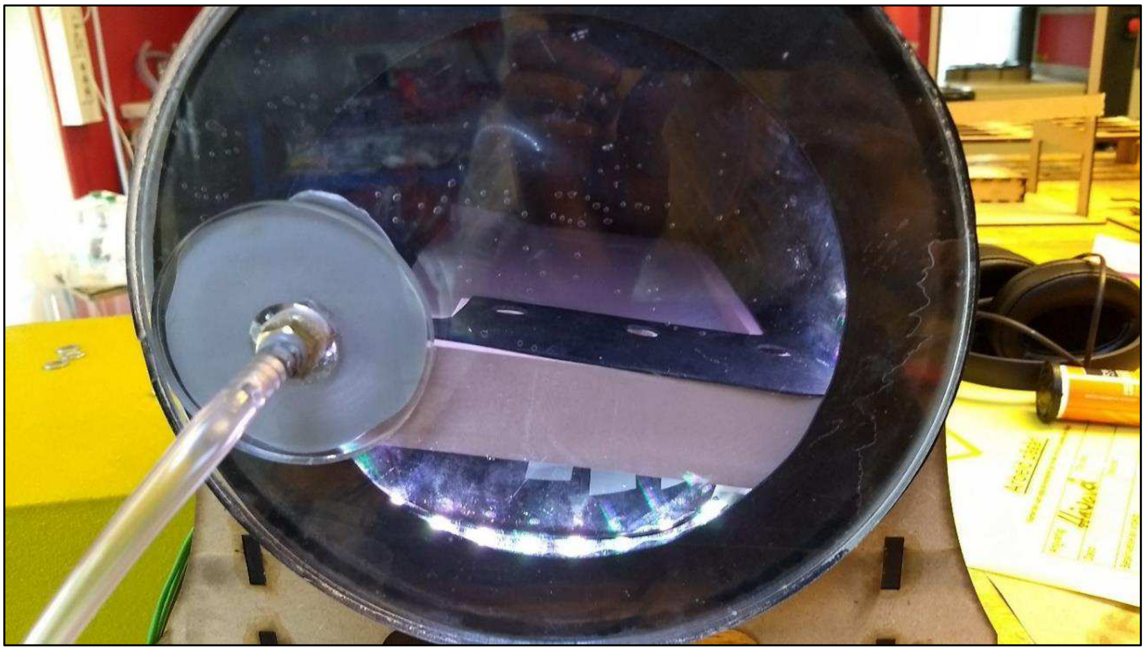


Figure 4.31 - Mold with silicone being degassed before assembly.

4.4.6 Final Wiring

To ensure that the foil remained water proof, all wiring was run through sleeved ethernet cables and cast into the foil during the casting process. These wires were now connected to the flexible threads with twisting and glued heat shrink like earlier, keeping track of the wire order. As the foil skin was now cast as a whole, with only flexible conductive threads as leads it was no problem turning the skin inside out giving good room to work on the wiring from the outside.

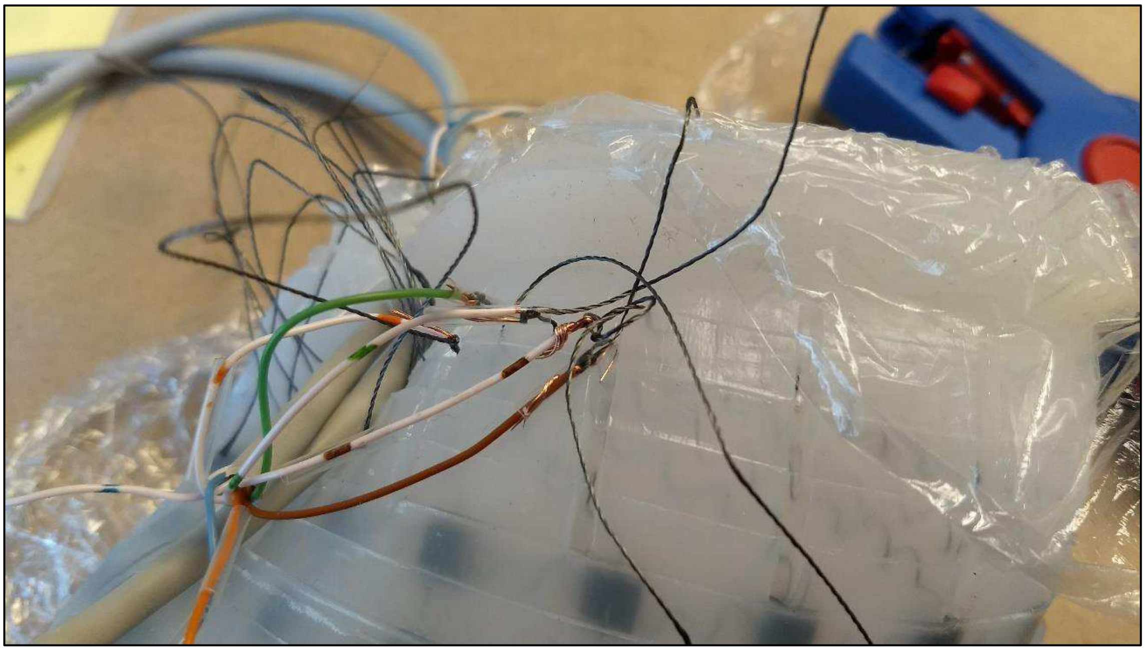


Figure 4.32 - Hydrofoil skin turned inside out for easy access

4.4.7 Results

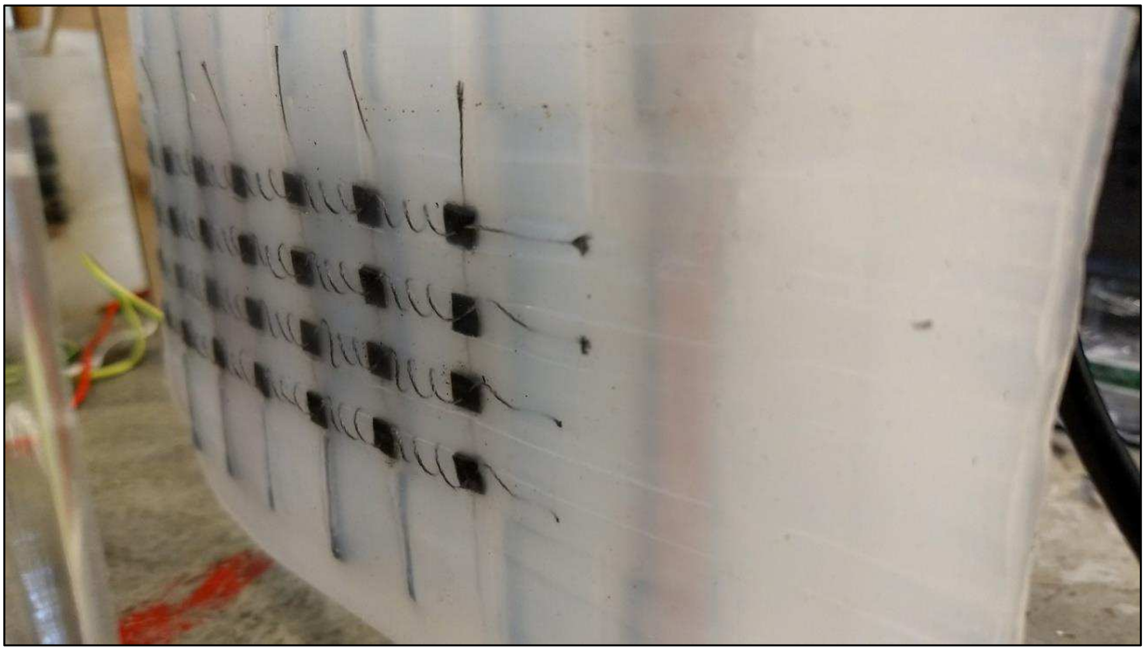


Figure 4.33 - Finished casting of hydrofoil with imbedded sensors

The skin came out looking very nice and even, the surface left from the CNCed molds were much smoother than any of my previous castings and the skin showed no signs of weakness when being stretched and deformed. When wiring the skin up to the multiplexers I made sure to reduce the resistor of the voltage divider to better accommodate the new smaller sensor pads. I lightly touched the foil to determine that it was indeed able to measure pressure along it's the skin surface. The skin gave much more predictable and correct looking pressure readings in the sense that applying pressure locally to only one sensor it was possible to pinpoint which area of the skin had experienced the pressure change. In Figure 4.34 one can see how there is almost no crosstalk between the left and right side of the foil, when the right side of the foil is stimulated.

Another thing that became apparent was that the sensitivity of the sensors decreased the further away from wire attachment they were, especially in lateral direction. The cause for this is probably the low resistance of the sensors and the high resistance of the conductive thread used as leads for the sensors. An additional high resistance is added for the sensors far to the back of the foil, which in turn reduces the relative voltage change these resistors can exert on the voltage divider.

Although even light touches along the skin surface gave high changes in the analog voltage read, the skin was noticeably less sensitive than the previous skins and sensors made. I tested the skin by putting it in the water tunnel, but I was not able to recognize any meaningful pressure changes in the sensors, other than a slight increase in voltage when the water tunnel was abruptly turned on from 0 to 12v. The new skin, although with much more even sensing capabilities, was simply not sensitive enough to measure the waterflows that I was able to make.

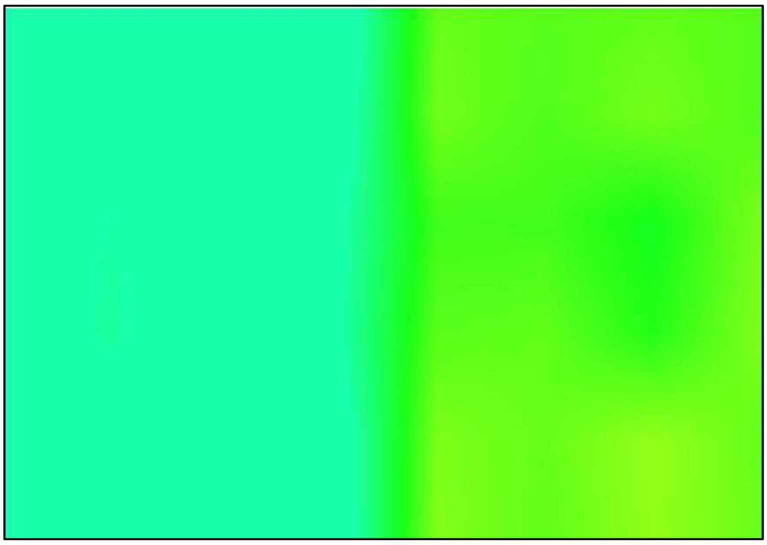


Figure 4.34 - Skin stimulated on one side showing low degree of crosstalk. Right side in the plot corresponds to the port side of the foil.

4.5 Foil end-data

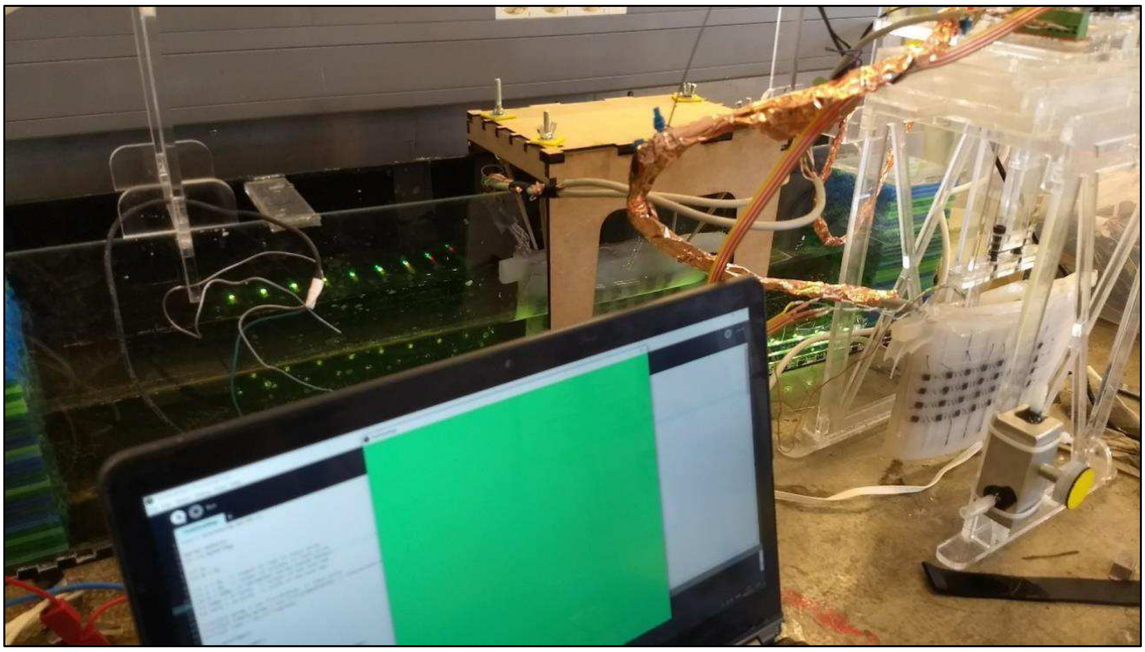


Figure 4.35 - Setup of data collection from hydrofoil

Since the new skin proved to not be sensitive enough to give good readings of the water flow, I went back to the previous foil. I patched up some of the holes in the foil with cyanoacrylate glue and re-soldered the broken leads. Instead of using the inner skeleton structure I inserted layers of 6mm acrylic palates cut to the core-shape to help the wing keep its shape and structure. I then added about 500grams of steel bars cut to size between the acrylic sheets to weigh the foil down and compensate for its buoyancy. I made a rig to hold the foil as described in chapter 2.7 with rubber bands to reduce any vibrational noise, and eye-bolts and wing nuts to enable adjustments of roll and yaw of the foil. I used needles to attach the rubber bands directly to the rubber silicone. The foil was then suspended at three diameters distance behind the D-cylinder. I ran the water tunnel at 5volts with the D-cylinder and ran a calibration software on the Arduino which I scripted to report the maximum and minimum values of each sensor and write it in the format of an Arduino matrix, so that it could be directly inserted into other programs. I ran this calibration to find the calibration values before every test. I used a web-camera and the hydrogen bubble generator to simultaneously record the real time footage that corresponds to the heatmap plots to better illustrate the connection.

4.5.1 Live plots

Running live plotting of the heatmap of the skin with simultaneous video capturing makes it easier to see patterns in the reaction of the sensors. While there is definitely a high amount of noise in the signals, some tendencies can be deduced. The frequency at which sensor values change and the amplitude of these changes seems much greater for the higher velocity flows.

For the free flows where the flow has not been altered with the D-cylinder to create a vortex street, there seems to be a more uniform distribution of pressure in the heatmap and they bear closer resemblance to data shown when the tunnel is not running than to its vortex-street counter scenarios. A clear difference between the right and left side can be seen in the foils, especially at high flows where the foil often would shift its pitch and turn one of its sides slightly towards the flow.

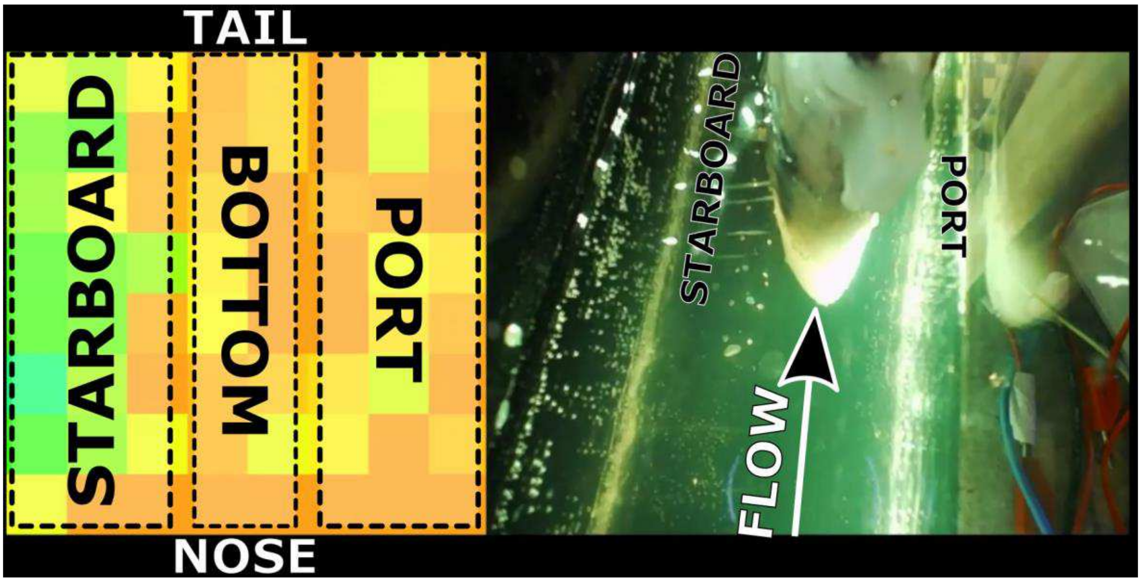


Figure 4.36 - The division and orientation of the heatmap and translation in video

The videos of the live plotted data are formatted as shown in Figure 4.36. The color of the heatmap ranges from cyan for serial value 0 and red for 10. The serial values are read from the Arduino, which has been calibrated for the flow region, so that 0 equals the lowest value experienced in the calibration period for the specific sensors and 10 equals the maximum value multiplied by 1.3.

Figure 4.37 - QR-video link. Hydrofoil in 5v Karman street https://youtu.be/qAElc60T-Rg

Figure 4.38 - QR-Video link. Hydrofoil in 12v Karman street https://youtu.be/ypdiFEIVcEg

By following the link in Figure 4.37, a video is shown of how a Kármán street looks like when interacting with the sensor foil at 5v. The heatmap of the data is plotted live and we see that there is some activity. The changes in pressure are slow, like the flow and seem to move in the direction of the flow. Figure 4.38 shows a video of how a Kármán street looks like when interacting with the sensor foil at 12v. The heatmap of the data is plotted live and we see that there are high amounts of activity, and the readings change rapidly.

Figure 4.39 - QR-Video link. Hydrofoil in still water - 5v turned on – laminar 5v flow https://youtu.be/l64YKEPvl0U

Figure 4.40 - QR-Video link. Hydrofoil in still water - 12v turned on - transitional 12v flow https://youtu.be/MKmJFZTURVg

Figure 4.39 and Figure 4.40 show the sensor foil in no flow before the flows are started at 5 and 12volts. The change in sensor readings is shown as the flow stabilizes and becomes laminar and transitional.

Although visually determining the functionality of the sensors has been useful throughout the prototyping process, it is not a very conclusive verification of the sensors performances. To further verify that there are patterns occurring in the data, and that the sensors are able to, to some degree, recognize the different flows I decided to sample larger sets of data to analyze with MATLAB.

4.5.2 Power spectral density

The setup for the wing suspended in the water tunnel with controlled conditions should result in a relatively predictable flow behavior. A D-cylinder suspended in the flow will shed vortices at calculatable intervals. As shown in chapter 2.9 the shedding frequency for the low flow at 5v is approximately 0.58Hz while the shedding frequency for the high, 12v, flow is 1.52Hz. It should therefore follow that the signals of single sensors in the foil should follow some cyclic behavior close to the shedding-frequency of the cylinder and flow speed. To verify whether I was able to see cyclic behaviors in the sensors I sampled data in 10minute periods for all of the flow scenarios, and preformed *power spectral densit (*PSD) analysis of the data in MATLAB. In Figure 4.41 the data for a single sensor over a 100s period can be seen. There seem to be cyclic fluctuations in the signals, but the visual frequencies seem far lower than those expected for the 12v Kármán street they were sampled from.

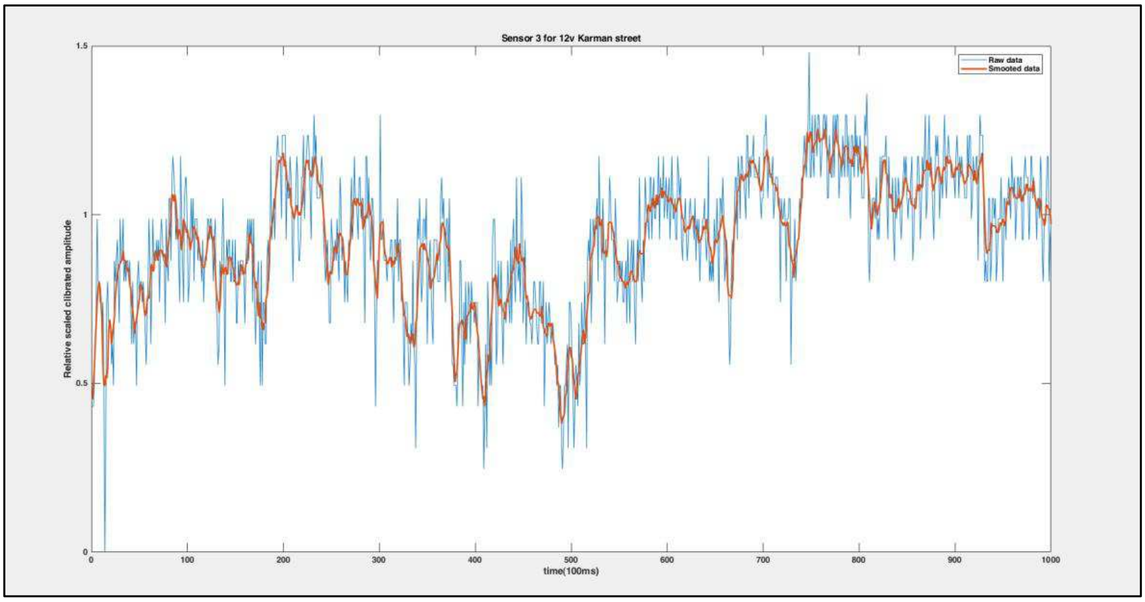


Figure 4.41 - 100 seconds of data shown for sensor 3.

The easiest way to perform a PSD analysis is through a *Fast Fourier transformatio (*FFT). An FFT analyses can be performed in MATLAB by calling the function *fft().* A PSD displays the amplitude along the Y-axis as power divided by frequency. To go from the FFT-result to a PSD periodogram, the results needs to be scaled in accordance with the power estimate. With noisy data especially, the resulting PSD graph will often show increasing values towards zero, to remove some of these tendencies one needs to account for drift and remove the mean value from the data-set. Due to the large amount of noise in my data, I still had a significant shift towards zero for longer datasets, and to better visualize the interesting frequencies I removed all frequencies lower than 0.2Hz when plotting the data. The resulting plots from the FFT analysis is very rough, due to the high amount of datapoints, but for sensors towards the front of the foil some interesting increases in amplitude and concentrations can be seen around 0.5Hz for the 5v case and 1.5Hz for the 12v case. To better visualize the PSD I use another PSD analysis tool in MATLAB called *pwelch()* which utilizes the *Welch Method*, and plot the graph against the FFT graph. The resulting graphs for single sensors can be seen in Figure 4.42 and Figure 4.43. The code for calculations and plotting of the data can be found in **appendix C**.

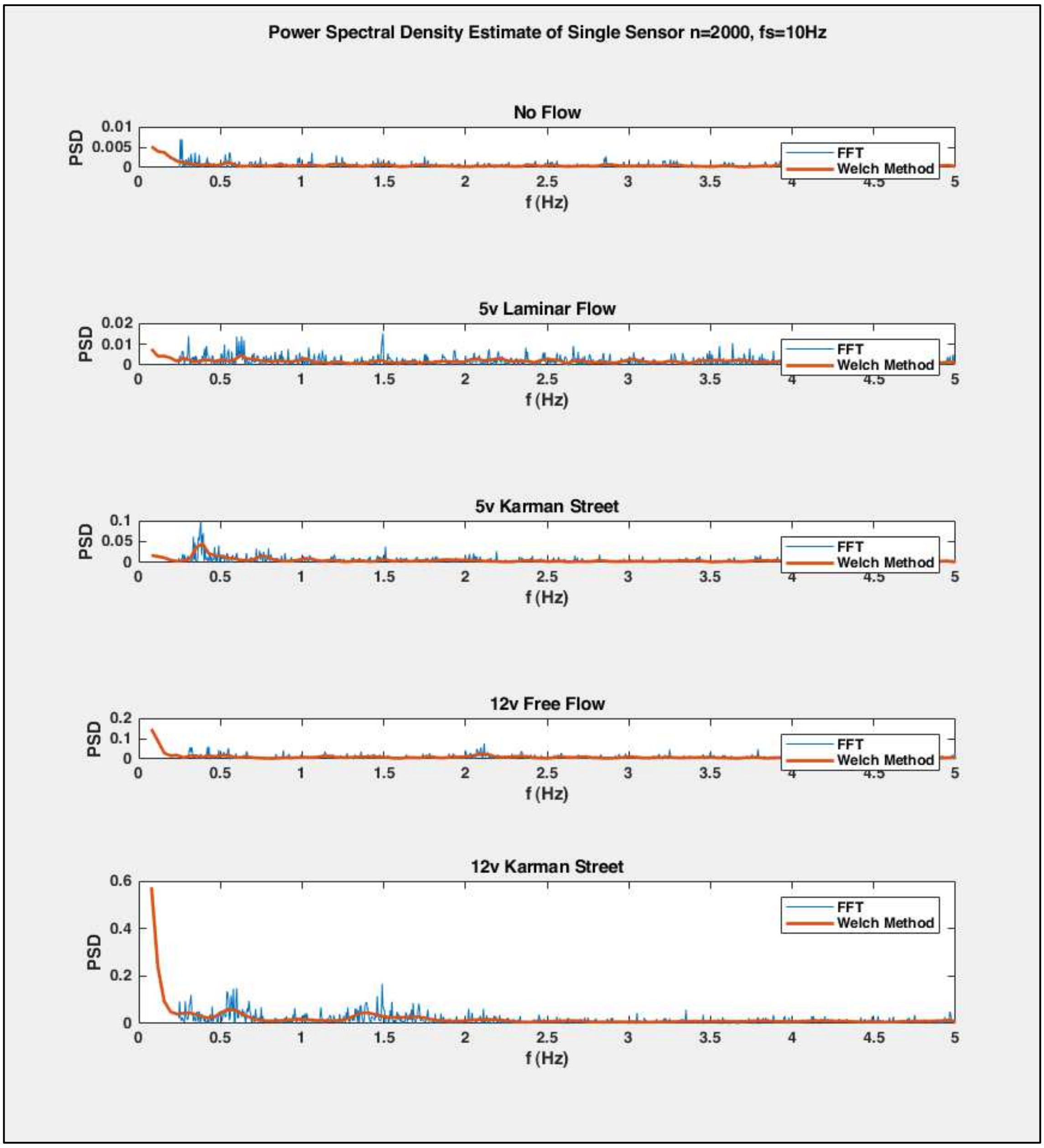


Figure 4.42 - PSD plots of all flow scenarios for sensor number 3. (top left side of foil third sensor from the front) Note that each plot is scaled differently to fit the plotted data.

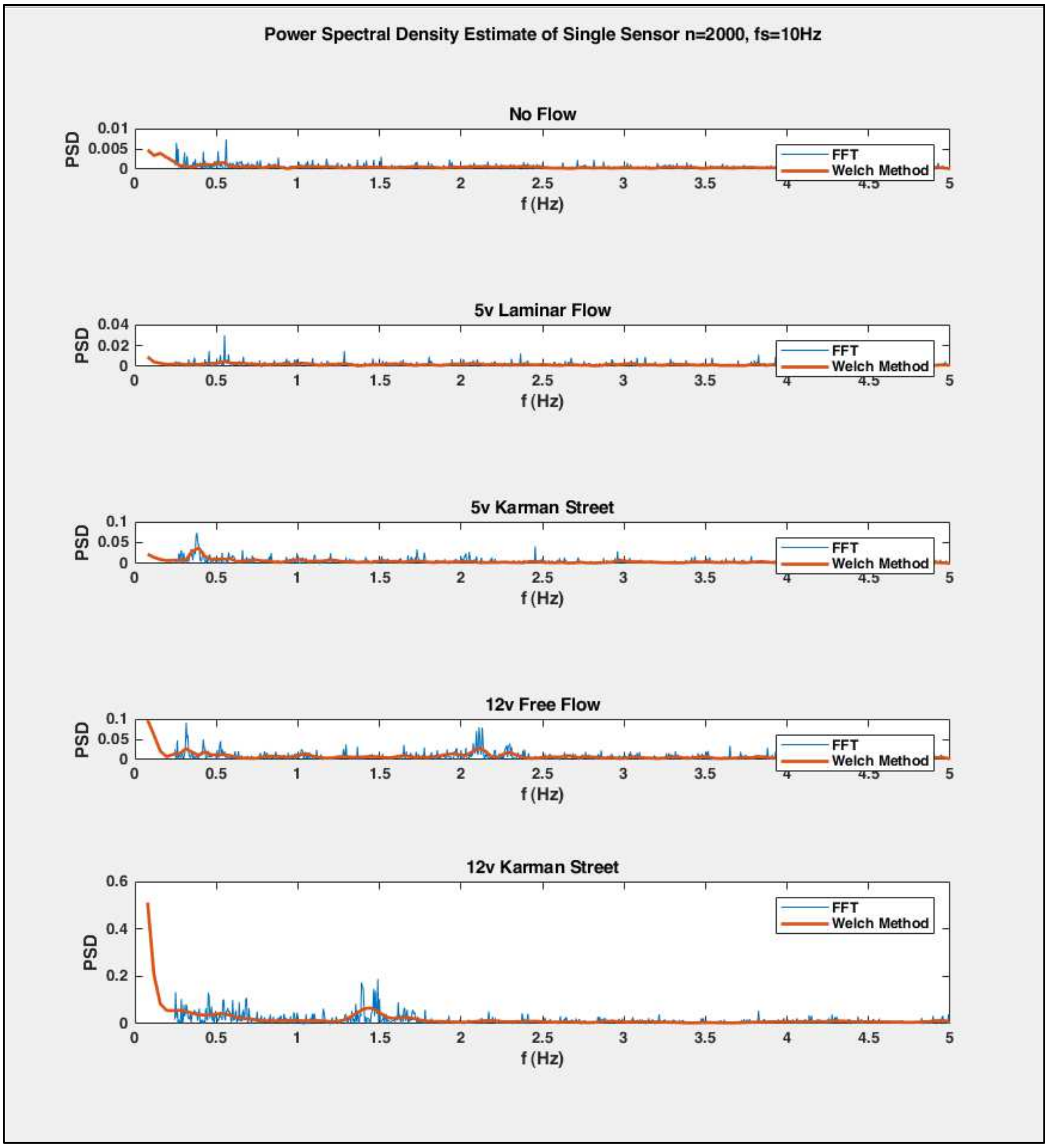


Figure 4.43 - PSD plots of all flow scenarios for sensor number 11. (second row top left side of foil third sensor from the front). Note that each plot is scaled differently to fit the plotted data.

Plotting the PSD methods against each other, the tendencies become clearer. Displaying the vast amount of data from 64 sensors this way would be somewhat difficult, so instead I checked sensors manually, while plotting the PSD for the same sensor for all of the flow scenarios.

The sensors along the first three columns, towards the front, of the foil showed much clearer cyclic behaviors. For the fastest, 12v, Kármán street a clear cyclic behavior can be seen with a frequency of about 1.5hz for almost all sensors at the front of the foil. This cyclic behavior can be seen for sensors 3 and 11 in the bottom plots in Figure 4.42 and Figure 4.43.

The 5v Kármán Street PSD does show cyclic behaviors for some of the sensors, but the frequency of 0.58Hz which it is supposed to correspond to is often inside the zero shifted “noise region” making it hard to classify whether it is an actual correct response or not. In Figure 4.42 and Figure 4.43 we see that there is a density peak for the 5v Kármán street flow scenario, but the frequency is slightly shifted as to what we would expect and shows a lower frequency close to 0.4Hz. The flow scenarios seem to be at the threshold of what the sensors are able to differentiate and sense, as the 12v Kármán street scenario is able to show correct tendencies where the lower flows, are not able to trigger sensor readings that act as we would expect.

The laminar and no flow scenarios tend to sometimes find peaks that indicate cyclic behaviors, but they are less frequent, consequent and often have lower density amplitudes. The frequencies at the "laminar" flows are often around 0.5Hz for the 5v flow and 1.2Hz and 2.1Hz for the 12v flow indicating that some sort of self-induced cyclic vortices might be made by the foil or some other obstacle in the water tunnel or that a predictable cyclic noise is affecting the model. I will also note that the sampling frequency is based on the Arduinos delay in read speed. According to the data sheet, and calculations the frequency should have been around 13Hz, yet when running data samplings over known time periods I found the mean sampling rate to be close to 10Hz. There might have been uneven speeds in the sampling rate of the Arduino, which would have caused shifts in the PSD analysis, I will also note that the large peaks at 0.4Hz for the 5v Kármán street would be shifted close to the true 0.58Hz we were expecting if calculated with a 13Hz sampling frequency.

Another interesting behavior found by accident when running an FFT analysis on the data row-wise instead of column-wise was that the pattern of the low and high-pressure regions on the skin, seemed to occur at quite constant intervals. When plotting the FFT plots of multiple still images of the foil skin against one another the resulting image seemed to repeat with varying amplitudes as seen in Figure 4.44. This might of course be a calibration issue, where the way the calibration is carried out more greatly impact how the individual sensors behave, but it could also indicate that patterns other than just time-dependent frequencies can be seen across the skin. If a pattern that is not dependent on time can be recognized, it could increase the speed at which flow scenarios are recognized, as fewer samples need to be taken before an estimate is made. If the skin had been perfectly calibrated, with sensors that all behaved equally, it would make sense that a pattern with repeating frequencies occurred across the skin for controlled flow scenarios. It might be the case that some of this behavior can still be seen with my less than perfect sensor skin. To investigate this, I made a quick neural network, machine learning algorithm to analyze the results and patterns.

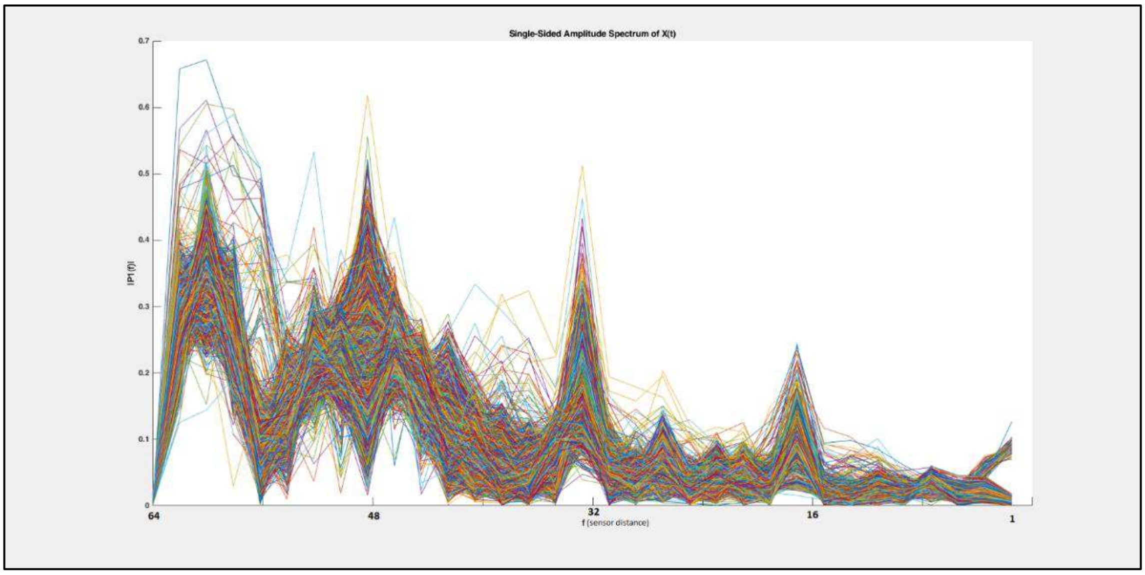


Figure 4.44 - FFT generated amplitude spectrums for 1000 skin scenarios for 12v Karman Street

4.6 Machine Learning

Looking at the heat-map visualization along with the video feed of the flow seem to suggest that there is a difference between the sensor readings of the different flow scenarios. With prior knowledge of how this dependency should look like in the data set, some of the expected trends can be seen in the power spectral density analysis, however with relatively low quality of signal to noise and scenarios with proximity in appearances, it is hard to determine features to extract for classification of flow scenarios. Purpose made algorithms to classify the data is usually the best approach, but when the classifier gets more advanced there are also other approaches to find patterns in the data. Machine learning, AI and big data are words that get thrown around a lot these days and are often used in settings where they might not fit. The strength, and reason for me to use this approach is that supervised machine learning algorithms usually are example based, meaning that by showing examples of scenarios that I know fit with certain criteria, the training algorithm will look for a way to separate it from other data.

The purpose here is not to find a classifier or algorithm that fully works, I know that a lot of refinement of both the sensors and the setup would be needed for predictable readings to occur over time I simply want to illustrate that there are differences between the observed flows in the sensors, and that there are patterns that link these differences.

4.6.1 Data sampling

To get some data to work with I used the first 1000 samples of data from the previous 10minute sampling period from the PSD analysis and sampled three additional 5minute periods and saved them to the MATLAB workspace, the first 1000 samples of the two first were added to the training data-set, making a total amount of 15000x64 datapoints. The aim is to sample some variations of the flows, so that a good classifier can be found that correctly distinguishes the differences between the flows. A dataset of 3 different variations of samples is probably far from sufficient, but as I am only trying to test a possibility and not make a final classification algorithm, I did not spend too much time sampling data. Between each sampling I calibrated the sensors with a 5v Karman street and the calibration script. The final 5minute period was sampled a different day and kept as a separate data set to later verify if a good classifier had been found in the training data.

I made an additional matrix to label the correct response for each flow for the neural network training. I decided to go with a single output variable for each flow, I also tested with two to define both type of flow and speed of flow but found this to produce a lot higher errors in the code, mainly due to its binary nature (either its wrong or right). I denoted the flows as; 1-No flow, 2 – 5v Laminar flow, 3 – 5v Kármán street, 4 – 12v Transitional flow, 5 – 12v Kármán street.

4.6.2 Neural Network

In my project thesis I constructed a neural network training and execution code, that ran on Arduino, based on *"Arduino Neural Network"* (Heymsfeld, 2018), capable of estimating conditions based on sensor data at a given moment. A more comprehensive explanation of how this was constructed can be found in **appendix-D.** For the flow scenarios that the sensors are deployed in however the change over time is probably also an important factor, I have seen shown that some frequencies can be recognized over time, which means that we need to treat the time series of the data. A powerful tool to write and run neural network training and execution is MATLAB. Not only can MATLAB handle advanced machine learning algorithms well, it also has built in tools for known algorithms such as neural networks, meaning little re-work needs to be done for the math behind the algorithms.

nnstart() starts the neural network tool, here I chose to work with time series data. I then crated a simple nonlinear input-output algorithm as I would have no verification data between each time interval.

4.6.3 Setup, testing and fitting

Although with modern tools, machine learning can be seen as some sort of black box where data is fed in and results come out, setting up a functional machine learning algorithm for your problem can be somewhat tedious, and involving a lot of guesswork and trial/error. There are mainly three aspects that can be tweaked in the model that I was using; the time delay, number of hidden nodes and the calculation method to find hidden nodes, and there is no straight forward way to correctly choose these design parameters as it will be highly individual for each dataset. While increasing the hidden nodes, might make algorithms that can perfectly separate and predict the training data, a more advanced algorithm like this can be quickly over trained and unable to find patterns in other data. Usually a good size for the hidden layer is found between the input size and output size. I found that the algorithms that best fit my training and testing data had small hidden layers, and the end algorithm that I used to plot Figure 4.45 and Figure 4.46 had a hidden layer of 5 nodes. The delay-parameter determines how many data steps the network can respond to. With lower amplitudes, I would have guessed that a long delay would be needed in my data, but I found that short delays usually made just as well fitted algorithms. The delay in Figure 4.45 and Figure 4.46 was set to 3. Another reason for shorter delays was the limitation of my computers processing powers, as delays longer than 15 would simply not calculate while running other processes. Perhaps increasing the delay further would have yielded even better algorithms, able to take the frequencies into account.

The hidden layer was calculated with the Levenberg-Marquardt training algorithm, as it yielded the best results. The training-algorithm stops, when the generalization stops improving; or the mean-square error starts to increase in the validation samples.

4.6.4 Results

I spent some time changing the parameters of my algorithm to find some decent resulting classifier. For almost all parameters and algorithms the resulting algorithm was able to properly classify data from later times in the training data. The results from the final trained neural network can be seen in Figure 4.45 and Figure 4.46. The yellow lines indicate the distance between the predicted scenario and the actual scenario.

The data from later times in the training data is data that is sure to be calibrated similarly and conducted in an identical setup, only potential noise sources would have changed between the data samples. Recognizing this data with the precision shown should indicate that there are clear patterns in the data. An interesting bias in the algorithm is where the flow changes from one to another and a high error is shown, this is due to the time delay which takes the previous measurements into account, and the fact that the dataset was not scrambled sufficiently so the neural network was not trained for changes in the data.

Recognizing data from a completely different setup, calibration and day was another story all together, while adding just small amounts of data from this set into the training would remove most of the errors, not doing so resulted in algorithms that would miss on a lot of the flows. Although, not the case for the algorithm shown in Figure 4.46, the 5v Kármán street would usually be the most recognizable scenario. Which would make sense at it is the most controlled flow. In Figure 4.46 we see that it is only the 12v free flow scenario, that show tendencies to being recognized.

With good results for the same data sets there are apparently differences between the flows, and algorithms can be found to differentiate them. The sensors are however very sensitive to change, both in drift of the sensor data and environmental change might be the reason for this. When re-calibrating the sensors, it obviously changes their response and appearance significantly, and comparing data from different calibration setups yielded high miss rates.

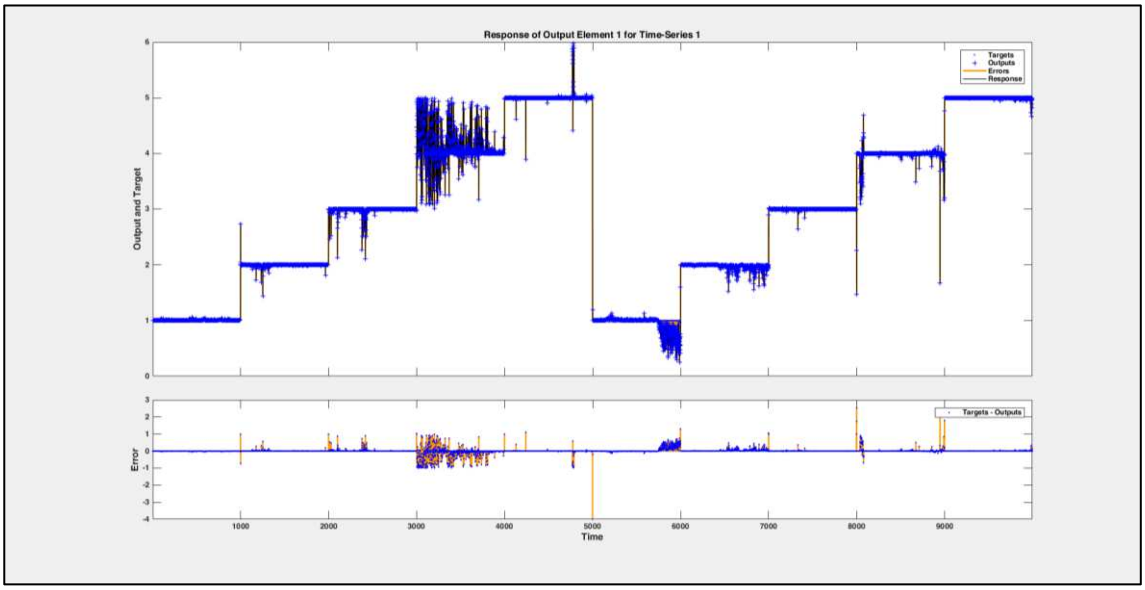


Figure 4.45 - Trained algorithm (n=5,d=3) predicting remaining data from data sets

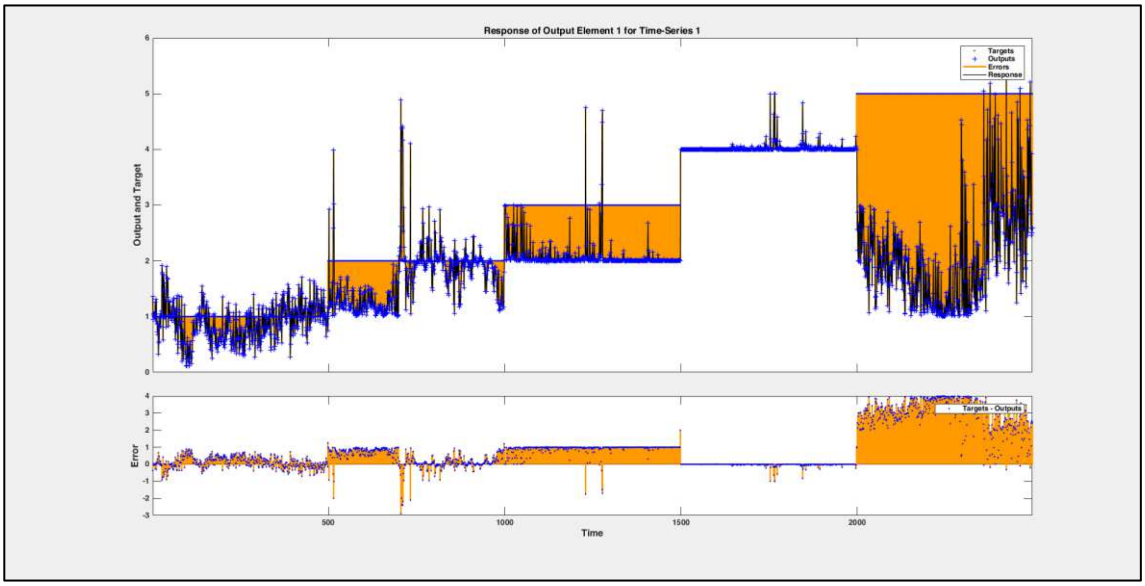


Figure 4.46 - Trained algorith (n=5,d=3) predicting new and re-calibrated data set

4.7 Testing sensors with movable and adaptive hydrofoil

The purpose of the project was to build sensors able to be installed in movable and adaptive hydrofoils. The best way to demonstrate this capability would be to actually make the foil move and adapt. When making the foil skins I designed the inner structure so that a skeleton of ribs and stringers could be installed, with the intention of installing a movable skeleton to test the capabilities of the sensors.

The most organic and natural movements I was able to generate in my artificial fish during the project thesis, found in **appendix (Chapter 4.4 and 3.4)**, consisted of inner structures with muscle wires attached to them. Muscle wires or *Nitinol* consists of a nickel-titanium alloy which, when heated retracts about 5% in length. The alloy also has a relatively high electrical resistance, which means that running current directly through the wire will generate enough heat to retract the wire. This means that it is very easy to install and use. Sticking with the technique that worked for my project thesis, I made an inner skeleton for the foils that fit with the slots made during the casting process. I oversized the foil slightly length wise by 10% so that the skin would be stretched, avoiding compression of the silicone at the inner radius during movement.

I made the skeleton out of 0.7mm Lexan poly carbonate, to keep it flexible, yet incompressible so that the chord length would remain unchanged. I cut slots for wires to go back and forth, and an arm for the foil to attach to a suspension rig. Slots were also cut for cross-bracings. The cross-bracings were made out of 1.6mm *gravoply* and installed to help retain the shape of the foil during flexing and keep the silicone skin away from the heated muscle wires.

I made electrode pads with screw connectors for the muscle wires to attach, using m3 nuts, screws and copper tape. The outermost pads were connected together and grounded, while the inner pads got separate leads to control the current through the sensors.

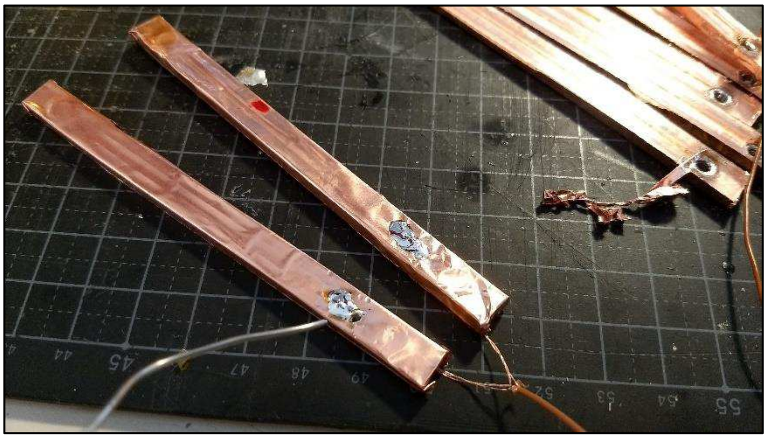


Figure 4.47 - Ground pads

Due to the high current needed to heat the muscle wires a separate motor controller is needed. I made a motor controller like in the project thesis with a potentiometer to balance the current going to the front and back muscles. For a more in dept description of how to make a movable skeleton with muscle wires please read the project thesis in **appendix D (chapter 4.4.4)**.

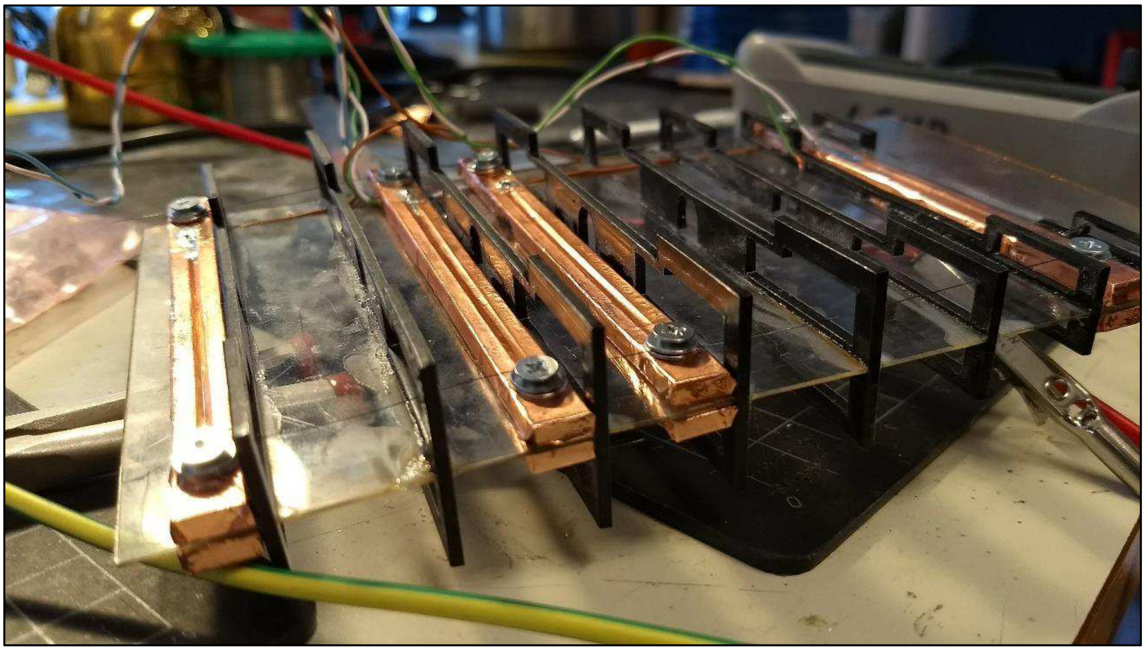


Figure 4.48 - Final actuation skeleton with muscle wires installed

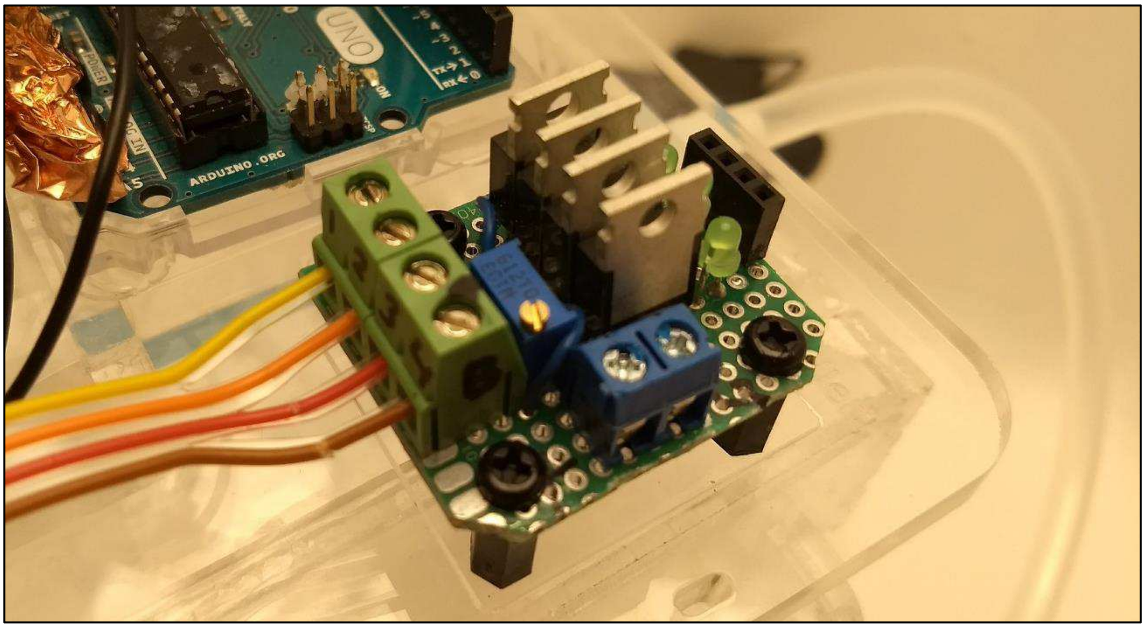


Figure 4.49 - Motor controller

4.7.1 Results

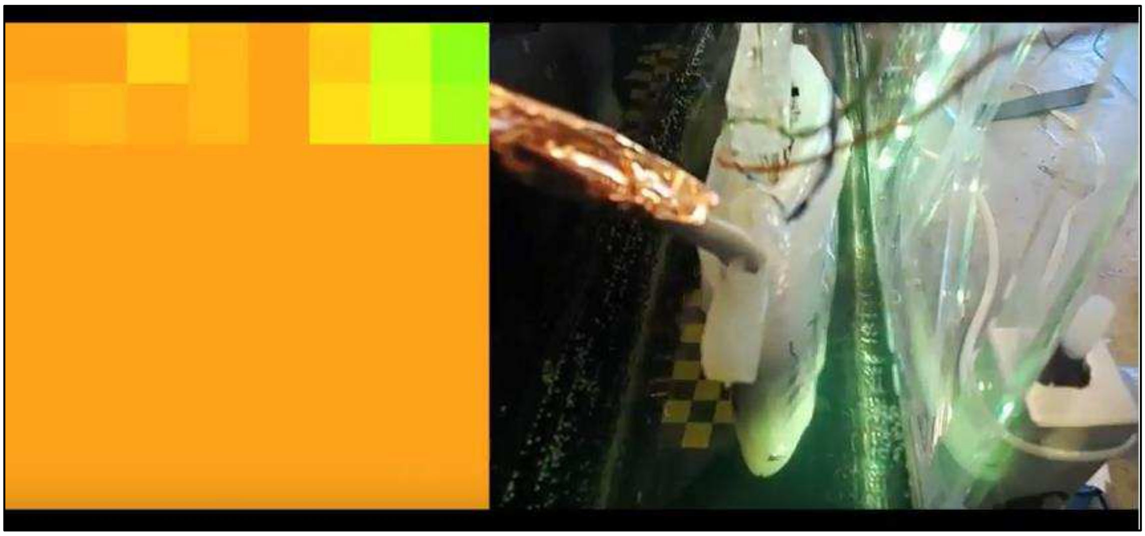


Figure 4.50 - Still image of swimming test

I was limited to use the moving skeleton with the sensor skin that was not itself sensitive enough for slow water-flows, and no movements of the foil were made based on data reading. Instead I manually moved the foil in the water tunnel by applying current to see how the sensors would react.

The sensors offered no restrictions against movement, and the foil stretched and moved organically, as if no sensors were installed. The surface of the foil was smooth with no bulging. Moving the foil induced pressure in the skin and the sensor readings from the movement exceeded the calibrated values and blacked out the readings from any additional pressure changes. When flexing the foil, two effects might have caused this; self-induced stretch and self-induced vortices/water pressure.

Compression and stretching of the skin when moving is probably the main reason for the high-pressure readings. The designed skin took little account for this. Any future skin and hydrofoil should take this into account either by removing stretching of the skin around the sensors all together, or by using softer and more flexible materials around the sensors so that the deformation can be kept to a minimum in the sensor material.

When the foil body moves it pushes against the water around it. This push, depending on the movement speed of the foil, can exceed the naturally occurring water pressure. This pressure is however not an error or design flaw, but rather an indication of the interaction between the foil and the water and part of how also aquatic animals experience water pressure.

Figure 4.51 - QR-Video link of sensor effects of swimming movements in hydrofoil https://www.youtube.com/watch?v=eclEOcS0Kh0

If algorithms that are able to account for the impact of movement can be implemented, one might be able to get additional data from the foil during adaptive movement. But it might also be that the sensitivity of the sensor gets changed or that a critical pressure is reached where additional pressure changes will no longer give readable voltage changes.

5 Summarization of results and learnings

I have tried to present the project as a sequential prototyping process, some activities of course happen simultaneously, but I have tried to highlight the way one decision and test has led to conclusions in the subsequently produced prototype or decision that had to be made. In the end I was able to produce two foils, which through self-produced sensors were able to sensitively sense pressure changes along their surfaces. One of the foils was sensitive enough to visualize changes in flows of only 0.04 and 0.1m/s and correctly show pressure fluctuations as a result of alternating vortices at 0.1m/s flow. While the other hydrofoil was not sensitive enough to show any noticeable readings in the mentioned flow velocities, it was able to showcase the flexibility of the CFRSS sensor and how it could be easily installed into a moving hydrofoil, with little footprint governing the internal electronics and structure, it also highlighted weaknesses with moving pressure sensors in the form of self-induced sensor-readings.

5.1 Innovative sensor development

By divergently exploring the solution space through an active use of prototypes I was able to come up with several models that showed promise; had I not discovered the piezo resistive properties of carbon fibers, then I think further work with the piezo resistive hair cells made from PCBs would have also yielded an interesting way to measure flows as well. But by freely exploring the solution space I was able to come up with a new way to measure flows, which to my knowledge has not been used before. In research that have undertaken similar problems, the solutions usually fall within either some piezo electric hair-cell lateral lines or commercially available pressure sensors.

5.2 Biomimicry

The sensor design is meant to function in the same way as the lateral line of fish. The model I mimicked and studied was the way fish are able to decide on adaptive actions in flows; through sensing. The end product might seem far more technically driven and not a pure example of biomimicry, and although there might be some truth to this, it is also important to remember that biomimicry is not defined as imitation of nature, but rather using a model and applying it to human problems. Using flows in adaptive hydrofoils is here the human problem, and artificial lateral lines the solution.

5.3 End sensor functionality

The capabilities of the CFRSS sensors far exceeded my expectations in their ability to measure pressure changes. Visually the sensors are able to react to pressure changes in the water very sensitively. I did not have an equivalent sensor to benchmark my sensors against, and I am not really able to think of a sensor type which it would be fair to compare the CFRSS sensors against, as sensors for the purpose of accurate flow measurement quickly become expensive and impractical when you expect accurate readings. In *"A fish perspective: detecting flow features while moving using an artificial lateral line in steady and unsteady flow"* (Chambers et al., 2014) they use 33 of the MS54 sensors, that I mentioned as an alternative in chapter 3.3, only they use the MS5401-AM version, to show how they are able to predict alternating vortex flows at 0.092, 0,186 and 0.261m/s through pressure changes. Their slowest flow velocity is almost the same as the fastest velocity in my experiments. Even with these relatively expensive sensors the accuracy is only 2Pa, which is close in proximity to the actual pressure changes that are happening in the slow flows I have been working with. Although they are also able to seemingly more accurately predict frequencies, this might as well be due to a better test environment and more realistic flow velocities.

To the best of my knowledge, the sensors preformed as well or better than any of the realistic alternatives I could have implemented into the hydrofoil, yet they could have benefited from more refinement and improved processing. Although the piezoresistive properties of carbon fiber is quite linear, I chose to not do calibrations of the sensors with known pressures/weights. The sensors had high drift and required frequent calibration, and the high number of sensors would have made such an approach impractical. My experience with the sensors was also that the sensitivity of the sensors was mostly in pressure **changes**, and not in statically applied pressures, adding difficulty to the calibration process. Instead of using few, well made and calibrated sensors I tried to show that results could also be achieved by many, less so, sensors.

5.4 Sensor failure

Throughout chapter 4.4 I attempted to refine the sensor through a more controlled approach to the manufacturing process of both the sensor and the skin in which it was integrated. Although the sensors seemed to preform much more consistently, their sensitivity was not nearly as high as expected. Although I was able to work with my previous sensor skin, and export data from it, it would be helpful for future research and production of the sensors to know which factors might have been their demise. The focus for this project was to prototype and explore the possibilities of the sensor, and I did not perform a methodological approach to determine the influence of different aspects in the production of the sensors had on the final sensor properties. Yet from the multiple sensors I made some factors seemed to play a role in the varying sensor results.

In the work of Sau et al. (1998) they suggest that the fiber content plays a very little role for response in relative resistance change and pressures applied, although we have seen that piezo resistive properties of carbon fiber vary greatly from the different fibers used (Blazewicz et al., 1997), it would be probable that some other aspect in the production would influence the sensitivity of the sensor.

Although using a laser cutter to shape the sensors gave clean even and uniform sensors, it also charred the edges of the sensors. The charring of the edges followed by using a paint thinner to clean the sensors might have created unwanted conductivity in the sensors. If more electrical contacts through charred carbons were created, the piezo resistive effect through the sensor could have become less apparent, for future production one should refrain from using cutting methods that alter the composition of the composite.

The new sensors were significantly smaller than any of the previous sensors. With less size there are shorter potential paths through the carbon fibers from one electrode to the other. Although the concrete mechanics of the piezo resistivity is still not fully understood, one can imagine that with a shorter distance over which the piezo resistive effect can take place, the less of an imprint the effect is able to make.

A clear contributor to reduced sensitivity was the increased wire length with high resistance wire. With the smaller sensors, there was a lower resistance over the sensor pads. With longer wires the resistance change over the sensor pads became a smaller contributor to the overall resistance in the voltage divider, and with a single pull down resistor in the multiplexer as described in chapter 4.2.3, there was no way to compensate for the increased resistance. Making sensors with a higher characteristic resistance over the pad would minimize the negative effect of the wires.

5.5 Manufacturing and cost

With the production method I have shown in this process, the CFRSS sensors can be cast in molds at normal conditions and can thus take almost any shape. Although casting sensors with a vacuum chamber increased the surface finish and consistency of the castings, no specialized equipment was really needed. The sensors can be made by anyone with minimal electrical and material knowledge. Even with the multiple sensors I made, I never exceeded 20x20cm in material usage of my carbon fiber sheet. When properties such as accurate weaving of the fibers is unimportant, the price of carbon fiber sheets is reduced dramatically. A 1x1m sheet of *Easy Composites Black Stuff Carbon Fibre 2/2 Twill 3k* that I used currently has the modest price of

12£, yet the fibers have the same properties as the higher end dry fiber weaves. Compared to carbon nanocomposites, the price of the fibers are near neglectable, in fact the fibers I used were acquired as part of the left over trash from another carbon fiber part production. Not only are they affordable, but due to its frequent use, carbon fibers are also very acquirable. To add to the list of positive properties of the manufacturing, working with carbon fiber poses a far lower health risk than its nanocomposite alternatives, and although protective breathing apparatuses, glasses and gloves should be used this is mainly due to irritating effects of the fibers.

5.6 Movement, critical pressure and retention

Moving the hydrofoil while reading the sensors showed on of the problems with the sensitive sensors, they become over stimulated from the high strain caused by the movement and go blind. Not only will the strain caused by the stretching of the body cause over stimulation, but the self-induced flows will blind the sensors from making any readings about the external flow scenario. The later of course would be a problem for any organism that swims in addition to sense the flows, and through processing the sensor reading along with the actuation signals it should be possible to distinguish the self-induced flows from those caused by the environment. This brings the next problem which I observed, that could be a restriction which needs to be considered when designing the sensors, which is the *critical pressure* of the sensors. The critical pressure was suggested by Sau et al. (1998) as a pressure after which little change is happening in carbon fiber-silicone mixtures with further increase in pressure. In their samples the critical pressure is quite low, at only 120g/cm^3. If movement of the foil or the body which the sensors are attached to induces critical pressure on a sensor, no further information can be gathered from that sensor, regardless of how well one processes the data.

Another effect on the sensors which becomes noticeable when you work with them is that although they mostly just reach to pressure changes, they will also retain some of the resistance change due to the pressures applied. If a high enough pressure is applied to the skin, the “zero-value” of the relaxed and unstimulated skin will be shifted towards the high pressure reading. To overcome the individual changes and retentions of the sensors I frequently calibrated the sensors, which seemed to overcome the problem temporarily but would not be a practical solution in a product.

5.7 Error modes

Throughout the testing of the self-made CFRSS sensors I’ve had a lot of trouble with different errors and noise on my signals. Making a sensor from nothing meant that finding out; what works and doesn't work, which precautions needs to be considered, and what's less important, were all things I had to find for myself. With little documentation to guide the process, it became a time consuming and often barren search for the proper procedures. With the low-pressure scenarios that I wanted to measure, any slight sources of noise and errors had great influence on the reported data. And a lot of the shown data is clearly influenced by a high noise to data ratio.

5.7.1 Noise – Shielding

When plotting the absolute voltage value of the sensors I experience surprisingly little problems with noise, from recognizable sources, on the signal wires. The fact that the wires were subject to a lot of noise first became apparent when I at one point tried to plot the absolute change of the sensor value between the readings and moving my hand in the air around the signal wires would cause fluctuations in the readings.

When working with sensitive sensors it is common to use coaxial or shielded cables to mitigate the EM and RF interference noise that the cable can be subjected to. To mitigate these effects, I shielded almost all wires that carried analog information, as explained in chapter 2.8, to mitigate the influence of cable noise. Yet removing all noise takes more than just one precaution and with the sensitive sensors, noise probably still have had a big influence on the measured signals.

5.7.2 Crosstalk

As discussed earlier, the way the sensors were wired enables a high amount of crosstalk between the sensors just through the sensor layout design. In a trade-off for simplicity of the circuit and sensor design, and making the skin more robust in that sense, I accepted this crosstalk knowing that the correct sensor would still be the main contributor to the measured voltage change and that the error from this crosstalk would be similar in size over all the sensors. With the uneven hand cut sensors this crosstalk is quite noticeable, as some sensors have a much higher sensitivity than others. With the high variation in sensitivity, stimulating the more sensitive sensors can trigger signal change at other sensor readings which might have sensors pads with very low sensitivity.

In the skin made with laser cut sensors and a consistent carbon fiber concentration, this crosstalk was much less noticeable. The areas on the skin stimulated could be much more locally defined and recognized. There are also other contributors to crosstalk in the sensor circuit, such as the multiplexer and analog to digital converter but precautions were taken in the code to mitigate these. I also added extra delays in the code to make sure that capacitance built up in the long leads would be pulled down between each sensor reading.

5.7.3 Rolling Shutter

Between each sensor sampling there is a delay. That means that each line of data does not correspond to the same exact moment in time. The distortion caused by these types of delays is often called a rolling shutter effect in photography. For the case of my sensors the delay between each sampling is rather high, both to deal with address changes in the multiplexer, analog signal purging and any residual capacitance. The time difference from the first sensor reading to the last is about 100milliseconds. Although there are many ways to remove the rolling shutter effect in post processing (Meingast, Geyer, & Sastry, 2005), with the low flow speed over the foil, and the large surface area of the sensors, 0.1seconds should not be long enough to distort the flow image completely, however it might be enough to off-set line-wise appearances of wave tops, which was expected.

5.7.4 Vibrations and environmental noise

The water tunnel and all experiments were set up in a workshop with all the vibrations and noise one could expect from such a facility. Earlier I described some of the precautions I took to minimize the effect of vibrational noise, but this far from eliminated the effect of the noise. Many of the vibrational noise-sources caused far more response in the sensors than the water flow itself. As an example, Figure 5.1 shows the turning on and off of the overhead workshop fan. To overcome this, I mainly ran the water tunnel during weekends, when the noise was at its minimum.

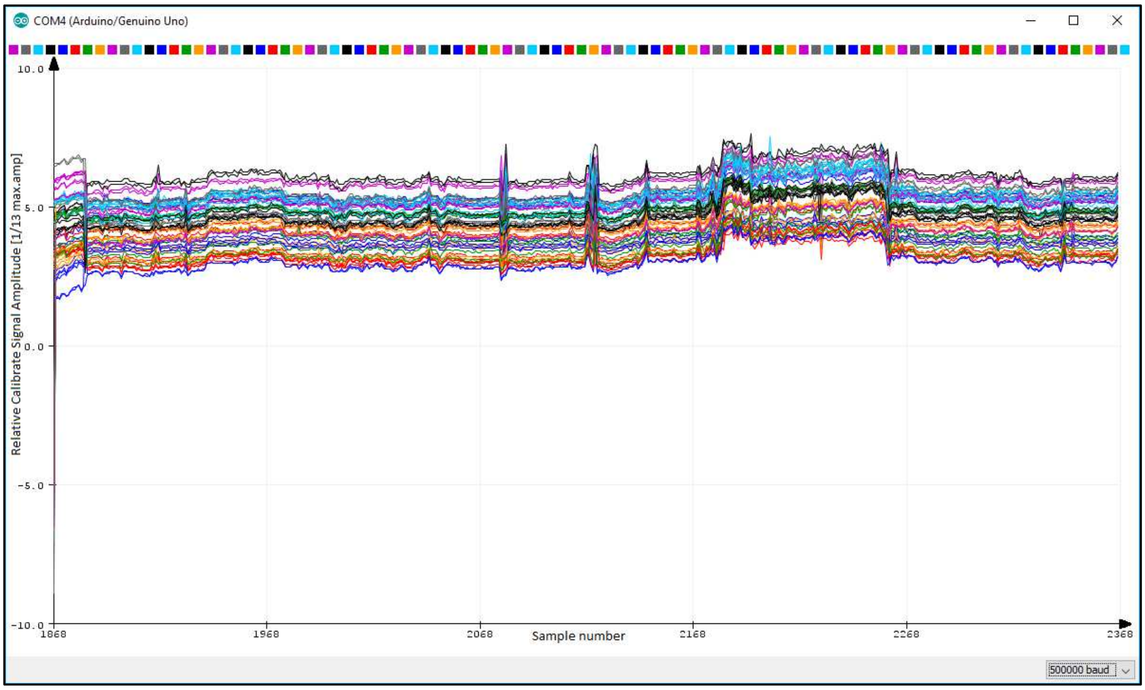


Figure 5.1 - A plot from the Arduino IDEs serial plotter. A significant increase in voltage can be seen between sample 2180 and 2250 due to overhead workshop fans being turned on and off.

5.7.5 Wall effects

The relatively wide size of the hydrofoil and the slim water tunnel means that the interaction between the flow and the wall will play a large role in the currents generated along the foil. Wall effects can play a large role in wind-tunnel simulations and in my case, it is undoubtably a big potential error, as not only are the walls close to the investigated flows, but the cross-sectional area is greatly reduced due to the size of the foil, and thus the flow-speed altered. With a maximum width of 35mm the blockage ratio is at least 35%, even without taking bulging of the foil into account. As a result, the PSD analysis plots seem to show much clearer and more correct frequencies along the first two rows of sensors at the leading edge of the foil.

5.7.6 Self-induced Turbulence and Water-tunnel imperfections

Ideally the flows generated in my water tunnel would perfectly fit the mathematical models and be completely predictable. This of course is not the real case. My equipment was made to be good enough to show tendencies, but not enough to give reliable data. Some turbulence and inconsistencies in the flow is to be expected. Furthermore, most of my tests were done with the first prototype foil, which due to bad casting and cure-inhibition had turned lumpy and uneven. Some self-induced turbulence along the foil should also be expected, making the flow pressures sensed less likely to be in accordance with the proposed models the further along the skin they were measured. Although fish and aquatic animals often use similar affects to their advantage by making turbulent boundary layers to lessen drag (Fish, 2006), for the case of my measurements it makes the readings a lot less useable, as an effect I mainly considered the first 1-3 columns of the sensor foil, where turbulence and wall effects had not yet distorted the vortex street completely. Due to the loose fastening of the foils, the eigen values of the foils might also have been a factor that caused errors in the frequency plots, if the flows caused the foils to slightly move and swing.

5.7.7 Unpredictable shedding frequencies

One of the reasons I abandoned trying to Kármán gate through manual control was the fact that even though the shedding frequency in the tunnel as an overall mean behaves mathematically sound, suddenly a vortex will be shed with a small time offset, offsetting the vortex street completely. The low speed in the water tunnel, combined with the flows not being perfect could play a role in the observed imperfections of the shedding frequencies. And the unpredictable nature of the shedding could have caused problems for the PSD analysis of the sensor data as well as the neural network.

6 Discussion and perspective

I started this project with a goal of making a better adaptive hydrofoil through biomimicry and prototyping by installing an artificial lateral line, but during the process the problem definition shifted to focus on improving and reinventing sensors for imbedding into hydrofoils.

6.1 Flow Detection

Through this project I have shown that it is possible to create sensitive pressure sensors, that are able to detect low pressure changes due to changes in water flow. The sensors are cheap and easy to produce. They are flexible and stretchable, although high flex and stretch of the sensors inhibits their ability to sense pressure changes. With good flow detection and observation comes the ability for improved flow control. Not only can proper motion and adaptation of foils in water increase the energy efficiency of the propulsion, but a better understanding of how the flows and body interact can increase the maneuverability of bodies traveling through flows. Agile fish typically have a turning radius of 10-30% of their body-length, with no speed change. Maneuvering ships will typically need to decrease their speeds while still being able to only achieve turning radiuses in dimensions 10 times larger than that of agile fish (Triantafyllou & Triantafyllou, 1995).

To keep with the way I constructed the hydrofoils in my project thesis I used stretchable materials and silicone to make the sensors, and although wings and hydrofoils are usually not made from silicone, the sensor style could still have applications in real foil scenarios for adaptation and optimization; the piezo resistive properties of carbon fiber can also be found with fibers in other matrices such as epoxies (Wang & Chung, 1995).

Wing and hydrofoil shapes have regions in which their lift and efficiency is at its maximum. Changing wings and hydrofoil shapes to fit specific working speeds can greatly increase the efficiency of the foils. Propeller hydroelastic tailoring is the act of making adaptive propeller blades to work over a bigger range of speed and flow conditions, by using the changing loads to deform the blades. By using composites in laminates, propeller structures can be changed to increase lift and energy efficiency adaptively (Mulcahy, Prusty, & Gardiner, 2010). Designing materials to self-adapt offers a potential increase in energy efficiency. If sensing properties and actuation could be implemented in composite blades, more specific changes in the foil shape can made to further increase the efficiency gain of adaptive propellers. By using carbon fiber-based sensors, the footprint of the sensing could be small and streamline structures could still be made that can give high data density feedback, and could be a potential implementation case for the sensor presented in this thesis.

6.2 Innovative aspects

Arguably my sensors is not groundbreaking in its field. There is existing research into the piezoresistive properties of carbon fiber, and even how to apply it as sensors (Wang & Chung, 1995). The increasing interest in sensors from silicone and carbon nanocomposites that have been immensely researched in recent years is probably the closest alternative to the sensors presented in this master's thesis, and their incredible capabilities is probably the reason for the relatively low interest in the piezo resistive capabilities of carbon fibers; piezo resistive properties of carbon nanocomposites are far greater than any documented carbon fibers

The efficiency of strain sensors is often compared in the form of their gauge factor., which is the relative change in electrical resistance divided by the mechanical elongation. Typical metal foil strain gauges have gauge factors of 2-5. Carbon nanotube sensors have been shown to be able to reach gauge factors of 2900 (Obitayo & Liu, 2012), although typically they are much lower in the ranges of 60-200. The single strain of HTA 5241 carbon fiber used as a sensor by Mäder et al. (2011) shows a gauge factor of roughly 2 while the most sensitive fiber in Blazewicz et al.'s (1997) study, P100s, showed a gauge factor of 8.5. The type of carbon fiber seem to greatly influence the properties of piezo resistivity and Wang & Chungs (1995) epoxy based sensor shows gauge factors up to 31 and composites of similar composition report gauge factors as high as 48.7 (Wang & Chung, 1996). The results by Sau et al. (1998), seem to suggest even higher gauge factors for their silicone and carbon fiber mixture, but no strain data is available. A better understanding of which fibers to use and how to use them might give improvement in the expected gauge factor of silicone carbon fiber sensors but reaching the same sensitivity as graphene or carbon nanotube-based sensors seems unrealistic and the sensor presented in this project is probably better suited to bridge the gap between these high end research materials and commercially available strain and pressure sensors.

With the extreme gauge factor of carbon nanotube sensors, it is hard to argue that my carbon fiber silicone sensor necessarily is an improvement from existing alternatives, yet the usability of the sensor and safe nature of the materials could make it an interesting tool in the development of products where external pressure sensing is needed.

6.3 Usability

One of the clear benefits of the carbon fiber sensor is its ease of manufacturing. It has no clear shape restrictions and can be installed pretty freely, being able to make your own with no specialized equipment within hours, sensors can be purpose built at a low cost.

6.4 Implementation into other fields

One of the use cases of carbon based nanocomposites as strain sensors is the implementation into structural materials such as cement (Yu & Kwon, 2009). This way structural loads and failures can be investigated without the need for external sensors. In their work Lee, You, Zi, and Yoo (2017) show that almost identical cement properties can be achieved for a 1vol.% multi-walled carbon nanotubes mixture, if half of the multi-walled carbon nanotubes are switched out with 0.1vol.% carbon fibers, reducing the cost of such a cement by almost half, yet achieve similar gauge factors as cement with only multiwalled carbon nanotubes.

It might be possible that similar composition changes, can be made in other use cases where carbon nanocomposite sensors are used, but using carbon fiber as sensors alone could also benefit a multitude of use cases where it would ease surface pressure sensing.

6.4.1 Empowering makers and fueling development

Although the silicone carbon nanocomposite material research has been intense, the use of these sensors in projects outside of research communities seems to be a lot more modest. The arguments for these types of flexible sensors often include robotics, and the ability to have sensors around flexible robotic joints, yet very little use of similar sensors seem to find their way into maker- and *do-it-yoursel (*DIY) communities. Creating a solution that works well, or "better than another", doesn't seem to be the true problem, but rather the applicability of the solution.

As an example, the *fused deposition modelin (*FDM) printing technology that we see so widely used in consumer 3d-printers today has been around since the 80's yet the growth and implementation into marked of such printers has only happened within the last 9 years, the reason of course being the expiration of the patent in 2009 (Riley, 2013). Power was given to the open marked of creators both through freedom to produce but also by giving the marked an easily makeable solution/recipe. The FDM plastic extruder printers do not produce the highest resolution plastic parts nor the strongest, yet they have become the most popular due to their availability and price. By making a pressure sensor that can be used in the same use-cases as the impressive carbon nanocomposite sensors, but from available and cheaper materials, the threshold for using and implementing the sensors into projects can be reduced, and possibly; development and refinement through high usage and testing, achieved.

6.4.2 Medical

The G-putty mixture of silicone and graphene (Boland et al., 2016) when tested, was sensitive enough to accurately measure both the pulse and blood pressure of a student. It is not hard to imagine that these types of highly sensitive and flexible piezoelectric materials can greatly benefit the medical industry. Monitoring health and vital signs, with non-intrusive equipment that can be worn as easily as clothes could open up for a lot of interesting research and the more documented use of carbon fibers and medical grade silicones could make it easier to get accepted as safe enough for medical research, where the somewhat more questionable toxicity of nanocomposites (Ou et al., 2016) could make this acceptance more difficult.

6.4.3 Smart-wear

A similar use case could be in smart clothing and flexible garments. With flexible sensors and circuits open up new ways for us to interact with technology. We can make electronics that not only follows and flexes with the user's body and tissue, but also electronics that could be cast to specifically fit the user. Whether using the sensors to detect motion, vital signs, or your body as a remote for other electronics, the possibilities are endless.

6.4.4 Robotics

Papers about flexible sensors often site the use in robotics as one of the main drivers into their innovation. Being able to have sensors that can flex around and with joints, while still giving readings is a big advantage in the development into humanoid as well as industrial robots, and can aid in developing trust and interaction between robots and humans as well as the added sensory inputs needed for fidelity tasks. In their masters project this spring Anne Proll Lien and Ole Mathias Samuelse (2018) were working with pneumatic McKibben-muscles to develop an exoskeleton, they asked me for a suggestion for a fitting sensor in their muscles that would be able to give feedback both about the compressive state of the muscle as well as the forces exerted by the muscle. The muscle consists of a silicone tube that is inflated inside a woven outer shell. This retracts the shell. Due to the flexible nature of the muscle, most sensors would have to be mounted externally from the muscle, but with one of the flexible silicone carbon-fiber sensors I made, they were able to mount the sensor inside the muscle and get feedback from the forces acting upon the muscle from the inside.

The cheap and flexible nature of the sensor enabled this whole test and implementation of the sensor to happen in a short period and is a great example of how versatile the sensors are. No extra ordering, manufacturing or expenses enables more rapid prototyping and implementation.

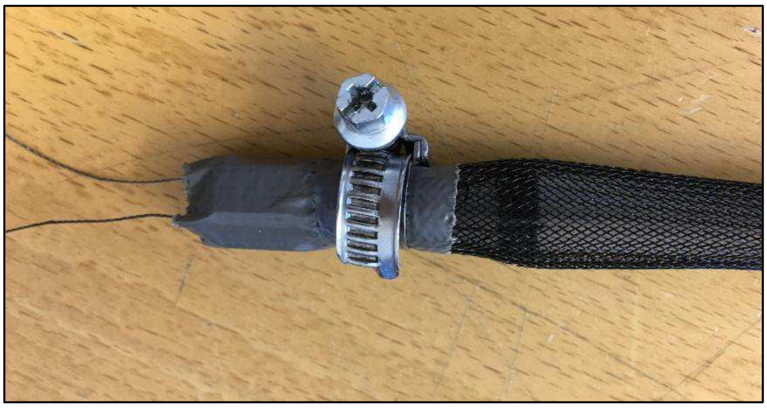


Figure 6.1 - McKibben muscle with CFRSS sensor installed

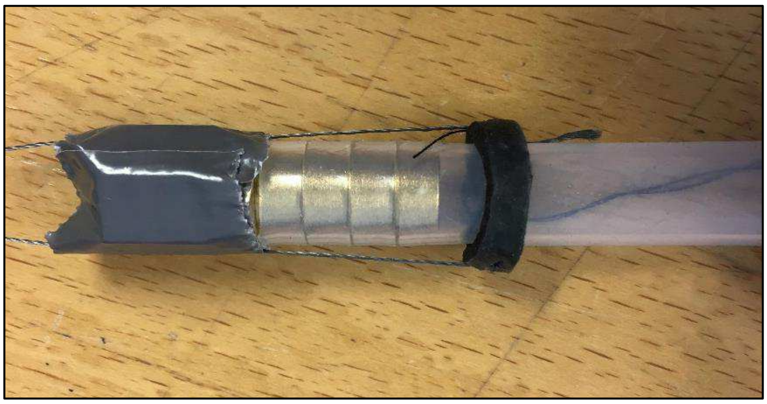


Figure 6.2 - Sensor inside McKibben muscle

6.4.5 Aeronautical

In their paper "*Srain sensing using single carbon fibres*" Mäder et al. (2011) investigate the idea of using single strains of carbon fiber to implement into structures consisting of the same fibers, so that a better the sensor and the material it is sensing show the same material properties. With an increase in composites used in aircraft construction, a natural use for composite based sensors such as these would be the aeronautical industry. Pressure at the sides of the aircraft as well as the stagnation pressure at the nose of the aircraft is used to measure the air speed of most modern aircrafts, with additional sensing of wind conditions and forces acting on the fuselage, more counter maneuvers can be made through either adapting the wings, changing altitudes, or speed to work with the air flows and not against them, and thus save energy, provide safer flights, and reduce wear on the fuselage and engines.

6.4.6 Extreme shape optimization and analysis

With increasing computational power and software, the ability to specifically engineer parts for their specific load cases has been greatly increased through tool such as topology optimization and generative design. At *"What is Generative Design"* ("What is Generative Design," 2018) Autodesk explains how using sensor data can greatly increase the load cases fed into their algorithms for generative design and topology optimization. Strain gauges can be applied along a temporary body, and the resulting deformations during real use fed into the algorithm to optimize the shape. In extreme constructions like these, the use of carbon fiber is not uncommon. Enabling direct measurements from the construction itself, through measuring resistance changes in the construction would completely remove the need or strain gauges and enable a higher density of sensor points and easier integration. This could be an alternative use case, where the same principals as those working in my sensor can be applied.

6.5 Weaknesses

The sensors made throughout this project have often been fast, and crude in their execution and their abilities and accuracy thereafter. Noise and uncertainty govern their output. I have no conclusive recipe for how to successfully tailor sensors for different sensitivity ranges and use cases, as the field is still quite unexplored. The sensors I have made have worked adequately to show promise for the technology, but not enough so that I was confident in measuring calibrated data in actual units, and only relative changes and pressures were examined. A further and more methodological research of the sensor and its composition might improve some of these traits.

6.6 Final thoughts

With their ease of use, flexibility and low cost I am surprised that making sensors from carbon fiber har not been done more frequently. Although out of the box solutions can be more comfortable and provide greater assurance in the validity of the data, specialized sensors can provide much higher flexibility in production and end products.

Although there are still ways to go in the development of hydrofoils and wings able to adapt to flows as efficiently as swimmers and flyers found in nature, an important piece in the puzzle of this development should be the sensing of local flows. In this project I have shown one way, although not necessarily the most practical in all applications, which could aid in further development of sensors for surface flow detection in adaptive hydrofoils and wings, as well as a multitude of other fields.

7 Bibliography

American Institute of Aeronautics & Astronautics (AIAA). (2017). The Wright Brothers' Wind Tunnel. Retrieved December 20, 2017, from http://legacy.wrightflyer.org/WindTunnel/testing1.html

analogRead(). (2017, December 4). Retrieved May 27, 2018, from https://www.arduino.cc/reference/en/language/functions/analog-io/analogread/

Asadnia, M., Kottapalli, A. G. P., Miao, J., Warkiani, M. E., & Triantafyllou, M. S. (2015). Artificial fish skin of self-powered micro-electromechanical systems hair cells for sensing hydrodynamic flow phenomena. *Journal of The Royal Society Interface*, *12*(111), 20150322. https://doi.org/10.1098/rsif.2015.0322

Baeckmann, W. von, Schwenk, W., & Prinz, W. (1997). *Handbook of Cathodic Corrosion Protection*. Elsevier.

Barron, J. L., Fleet, D. J., & Beauchemin, S. S. (1994). Performance of optical flow techniques. *International Journal of Computer Vision*, *12*(1), 43–77. https://doi.org/10.1007/BF01420984

Barthlott, W., & Neinhuis, C. (1997). Purity of the sacred lotus, or escape from contamination in biological surfaces. *Planta*, *202*(1), 1–8. https://doi.org/10.1007/s004250050096

Bell, A. G. (1881, July 19). *US244426A*. United States. Retrieved from https://patents.google.com/patent/US244426/en

Benyus, J. M. (2008). *Biomimicry: innovation inspired by nature* (repr). New York, N.Y: Harper Perennial.

Blazewicz, S., Patalita, B., & Touzain, P. (1997). Study of piezoresistance effect in carbon fibers. *Carbon*, *35*(10), 1613–1618. https://doi.org/10.1016/S0008-6223(97)00120-6 blorgggg. (2015, December 31). Silc Circuits: High Performance Conductive Silicone. Retrieved May 24, 2018, from http://www.instructables.com/id/Silc-Circuits-High-Performance-Conductive-Silicone/

Boland, C. S., Khan, U., Ryan, G., Barwich, S., Charifou, R., Harvey, A., … Coleman, J. N. (2016). Sensitive electromechanical sensors using viscoelastic graphene-polymer nanocomposites. *Science*, *354*(6317), 1257–1260. https://doi.org/10.1126/science.aag2879

Buehler, J., & Patel, N. V. (2015, March 11). A Real Drag. *Slate*. Retrieved from http://www.slate.com/articles/health_and_science/science/2015/03/mercedes_benz_bionic_car_boxfish_stability_and_agility_paradox_finally_solved.html

Chambers, L. D., Akanyeti, O., Venturelli, R., Ježov, J., Brown, J., Kruusmaa, M., … Megill, W. M. (2014). A fish perspective: detecting flow features while moving using an artificial lateral line in steady and unsteady flow. *Journal of The Royal Society Interface*, *11*(99), 20140467. https://doi.org/10.1098/rsif.2014.0467

Chen, J. (2007). Artificial Hair Cells for Sensing Flows. Retrieved from http://ntrs.nasa.gov/search.jsp?R=20090040745

Cremer, L., Heckl, M., & Petersson, B. A. T. (2005). *Structure-borne sound: structural vibrations and sound radiation at audio frequencies* (3rd ed). Berlin ; New York: Springer.

Dharap, P., Li, Z., Nagarajaiah, S., & Barrera, E. V. (2004). Nanotube film based on single-wall carbon nanotubes for strain sensing. *Nanotechnology*, *15*(3), 379. https://doi.org/10.1088/0957-4484/15/3/026

Doolittle, W. F. (2000). Uprooting the tree of life. *Scientific American*, *282*(2), 90–95.

Dow, S. P., Glassco, A., Kass, J., Schwarz, M., Schwartz, D. L., & Klemmer, S. R. (2012). Parallel Prototyping Leads to Better Design Results, More Divergence, and Increased

Self-efficacy. In *Design Thinking Research* (pp. 127–153). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-21643-5_8

Streitlien, K., Triantafyllou, G. S., & Triantafyllou, M. S. (1996). Efficient foil propulsion through vortex control. *AIAA*, *34*(11), 2315–2319.

Farina, S., & Summers, A. (2015). *Biomechanics: Boxed up and ready to go* (Vol. 517). https://doi.org/10.1038/517274a

Fish, F. E. (2006). The myth and reality of Gray's paradox: implication of dolphin drag reduction for technology. *Bioinspiration & Biomimetics*, *1*(2), R17-25. https://doi.org/10.1088/1748-3182/1/2/R01

Fish, F. E., Weber, P. W., Murray, M. M., & Howle, L. E. (2011). The Tubercles on Humpback Whales' Flippers: Application of Bio-Inspired Technology. *Integrative and Comparative Biology*, *51*(1), 203–213. https://doi.org/10.1093/icb/icr016

Phenix, M. (2005, August 16). Fish-Inspired Car. Retrieved December 17, 2017, from https://www.popsci.com/aerodynamic/article/2005-08/fish-inspired-car

Flock, Å., & Wersäll, J. (1962). A Study of the Orientation of the Sensory Hairs of the Receptor Cells in the Lateral Line Organ of Fish, with Special Reference to the Function of the Receptors. *The Journal of Cell Biology*, *15*(1), 19–27. https://doi.org/10.1083/jcb.15.1.19

Gawron, R. (2011, April). Homemade pressure sensor. Retrieved May 27, 2018, from https://robertgawron.blogspot.com/2011/04/home-made-pressure-sensor.html

Gonzales, P. D. (2011). *The evolution of ideas in biomimicry* (Thesis). Massachusetts Institute of Technology. Retrieved from http://dspace.mit.edu/handle/1721.1/69775

Gopalkrishnan, R., Triantafyllou, M. S., Triantafyllou, G. S., & Barrett, D. (1994). Active vorticity control in a shear flow using a flapping foil. *Journal of Fluid Mechanics*, *274*, 1–21. https://doi.org/10.1017/S0022112094002016

Heckl, M. (1982). Structure Borne Sound. In *Noise Generation and Control in Mechanical Engineering* (pp. 209–287). Springer, Vienna. https://doi.org/10.1007/978-3-7091-2894-7_3

Heymsfeld, R. (2018, May 20). Arduino Neural Network. Retrieved June 8, 2018, from http://robotics.hobbizine.com/arduinoann.html

Humpback whales inspire next generation wind turbine technology. (2009, August 18). Retrieved December 17, 2017, from https://materia.nl/article/humpback-whales-inspire-next-generation-wind-turbine-technology/

Hwang, J., Jeong, Y., Park, J. M., Lee, K. H., Hong, J. W., & Choi, J. (2015). Biomimetics: forecasting the future of science, engineering, and medicine. *International Journal of Nanomedicine*, *10*, 5701–5713. https://doi.org/10.2147/IJN.S83642

Shipping and World Trade. (2018, June 7). Retrieved June 7, 2018, from http://www.ics-shipping.org/shipping-facts/shipping-and-world-trade

Kang, I., Schulz, M. J., Kim, J. H., Shanov, V., & Shi, D. (2006). A carbon nanotube strain sensor for structural health monitoring. *Smart Materials and Structures*, *15*(3), 737. https://doi.org/10.1088/0964-1726/15/3/009

Kennedy, B. M., Sobek, D. K., & Kennedy, M. N. (2014). Reducing Rework by Applying Set-Based Practices Early in the Systems Engineering Process. *Systems Engineering*, *17*(3), 278–296. https://doi.org/10.1002/sys.21269

Kim, S., Spenko, M., Trujillo, S., Heyneman, B., Santos, D., & Cutkosky, M. R. (2008). Smooth Vertical Surface Climbing With Directional Adhesion. *IEEE Transactions on Robotics*, *24*(1), 65–74. https://doi.org/10.1109/TRO.2007.909786

Lee, S.-J., You, I., Zi, G., & Yoo, D.-Y. (2017). Experimental Investigation of the Piezoresistive Properties of Cement Composites with Hybrid Carbon Fibers and Nanotubes. *Sensors*, *17*(11), 2516. https://doi.org/10.3390/s17112516

Li, X., Zhang, R., Yu, W., Wang, K., Wei, J., Wu, D., … Zhu, H. (2012). Stretchable and highly sensitive graphene-on-polymer strain sensors. *Scientific Reports*, *2*, 870. https://doi.org/10.1038/srep00870

Liao, J. C., Beal, D. N., Lauder, G. V., & Triantafyllou, M. S. (2003a). Fish Exploiting Vortices Decrease Muscle Activity. *Science*, *302*(5650), 1566–1569. https://doi.org/10.1126/science.1088295

Liao, J. C., Beal, D. N., Lauder, G. V., & Triantafyllou, M. S. (2003b). The Kármán gait: novel body kinematics of rainbow trout swimming in a vortex street. *Journal of Experimental Biology*, *206*(6), 1059–1073. https://doi.org/10.1242/jeb.00209

LiaoLab. (2012). *Dead trout in flowing water*. Retrieved from https://www.youtube.com/watch?v=_ZBWnhzYvts

Loh, K. J., Lynch, J. P., Shim, B. S., & Kotov, N. A. (2008). Tailoring Piezoresistive Sensitivity of Multilayer Carbon Nanotube Composite Strain Sensors. *Journal of Intelligent Material Systems and Structures*, *19*(7), 747–764. https://doi.org/10.1177/1045389X07079872

Mäder, T., Nestler, D., & Wielage, B. (2011). Strain Sensing using Single Carbon Fibres (p. 6).

Marras, S., & Porfiri, M. (2012). Fish and robots swimming together: attraction towards the robot demands biomimetic locomotion. *Journal of The Royal Society Interface*, rsif20120084. https://doi.org/10.1098/rsif.2012.0084

Meingast, M., Geyer, C., & Sastry, S. (2005). Geometric Models of Rolling-Shutter Cameras. *ArXiv:Cs/0503076*. Retrieved from http://arxiv.org/abs/cs/0503076

MitchR. (2018, March 1). How to - Simplify this code? (heat map). Retrieved June 9, 2018, from https://forum.processing.org/two/discussion/26588/how-to-simplify-this-code-heat-map

Mulcahy, N. L., Prusty, B. G., & Gardiner, C. P. (2010). Flexible Composite Hydrofoils and Propeller Blades. In *International Maritime Conference 2010: Maritime Industry - Challenges, Opportunities and Imperatives, 27-29 January 2010, Sydney, Australia* (pp. 438–448). Sydney, Australia. Retrieved from http://search.informit.com.au/documentSummary;dn=510861154600090;res=IELENG

Obitayo, W., & Liu, T. (2012). A Review: Carbon Nanotube-Based Piezoresistive Strain Sensors [Research article]. https://doi.org/10.1155/2012/652438

Ou, L., Song, B., Liang, H., Liu, J., Feng, X., Deng, B., … Shao, L. (2016). Toxicity of graphene-family nanoparticles: a general review of the origins and mechanisms. *Particle and Fibre Toxicology*, *13*. https://doi.org/10.1186/s12989-016-0168-y

quark. (2016, December). Heat Map Color Resolution. Retrieved June 9, 2018, from https://forum.processing.org/two/discussion/20004/heat-map-color-resolution

Riley, J. (2013, March 14). Open Mindsets Link 3D Printing & Internet of Things. Retrieved May 24, 2018, from https://www.computerworlduk.com/it-business/open-mindsets-link-3d-printing--internet-of-things-3571147/

Rodriguez, A., Andersen, E., Bradley, J., & Taylor, C. (2007). Wind Estimation Using an Optical Flow Sensor on a Miniature Air Vehicle. In *AIAA Guidance, Navigation and Control Conference and Exhibit*. Hilton Head, South Carolina. https://doi.org/10.2514/6.2007-6614

Salumäe, T., & Kruusmaa, M. (2013). Flow-relative control of an underwater robot. *Proc. R. Soc. A*, *469*(2153), 20120671. https://doi.org/10.1098/rspa.2012.0671

Samuelsen, O. M. (2018, May 25). Facebook Conversation with Ole Mathias Samuelsen.

Sau, K. P., Khastgir, D., & Chaki, T. K. (1998). Electrical conductivity of carbon black and carbon fibre filled silicone rubber composites. *Die Angewandte Makromolekulare*

Chemie, *258*(1), 11–17. https://doi.org/10.1002/(SICI)1522-9505(19980801)258:1<11::AID-APMC11>3.0.CO;2-0

Scheiman, & Brooks. (1981). Comparison of Experimental and Theoretical Turbulence Reduction from Screens, Honeycomb, and Honeycomb-Screen Combinations | Journal of Aircraft. Retrieved December 14, 2017, from https://arc.aiaa.org/doi/abs/10.2514/3.57538?journalCode=ja

Sfakiotakis, M., Lane, D. M., & Davies, B. C. (1999). Review of Fish Swimming Modes. Retrieved from http://edge.rit.edu/edge/P14029/public/Benchmarking/benchmarking_felix/Review%20of%20Fish%20Swimming%20Modes.pdf

Snowdon, J. C. (1965). Rubberlike materials, their internal damping and role in vibration isolation. *Journal of Sound and Vibration*, *2*(2), 175–193. https://doi.org/10.1016/0022-460X(65)90095-7

Steinert, M., & Leifer, L. J. (2012). Finding-Ones-Way-Re-Discovering-a-Hunter-Gatherer-Model-based-on-Wayfaring.pdf. *International Journal of Engineering Education*, *28*(2), 251–252.

Thomke, S., & Fujimoto, T. (2000). The effect of “front-loading” problem-solving on product development performance. *Journal of Product Innovation Management*, *17*(2), 128–142. https://doi.org/10.1016/S0737-6782(99)00031-4

Triantafyllou, G. S., & Triantafyllou, M. S. (1995). An Efficient Swimming Machine. *Scientific American*, *272*(3), p64.

Ulrich, K. T., & Eppinger, S. D. (2012). *Product Design and Development* (5th ed.). New York, NY: McGraw-Hill.

Vogel, S. (1983). *Life in moving fluids: the physical biology of flow*. Princeton University Press.

Vogel, S. (2000). *Cats' Paws and Catapults: Mechanical Worlds of Nature and People*. W. W. Norton & Company.

Wang, X., & Chung, D. D. L. (1995). Short-carbon-fiber-reinforced epoxy as a piezoresistive strain sensor. *Smart Materials and Structures*, *4*(4), 363. https://doi.org/10.1088/0964-1726/4/4/017

Wang, X., & Chung, D. D. L. (1996). Continuous carbon fibre epoxy-matrix composite as a sensor of its own strain. *Smart Materials and Structures*, *5*(6), 796. https://doi.org/10.1088/0964-1726/5/6/009

What is Generative Design. (2018, June 9). Retrieved June 9, 2018, from https://www.autodesk.com/solutions/generative-design

Yan, C., Wang, J., Kang, W., Cui, M., Wang, X., Foo, C. Y., … Lee, P. S. (2014). Highly Stretchable Piezoresistive Graphene–Nanocellulose Nanopaper for Strain Sensors. *Advanced Materials*, *26*(13), 2022–2027. https://doi.org/10.1002/adma.201304742

Yu, X., & Kwon, E. (2009). A carbon nanotube/cement composite with piezoresistive properties. *Smart Materials and Structures*, *18*(5), 055010. https://doi.org/10.1088/0964-1726/18/5/055010

Zeng, H., & Zhao, Y. (2011). Sensing Movement: Microsensors for Body Motion Measurement. *Sensors*, *11*(1), 638–660. https://doi.org/10.3390/s110100638

APPENDIX A

Arduino Codes

A – 1: Arduino code for reading sensor skin data and plotting it to MATLAB arrays

```
/*MatlabFormatting
 * 08.06.2018
 * Håvard Nitter Vestad
 * This script was written as part of my masters thesis
 * the spring of 2018 at MTP, NTNU.
 *
 * This script is intended to format the data from a 8x8
 * array of analog signals, sampled through two CD74HC4067
 * Multiplexer, so that the printed information in the
 * serial monitor can be copied and saved as arrays in
 * MATLAB.
*/

//--------------- Calibration values -------------------
//Before running the scrip, run the calibration scrip
//Let the calibration run until you are confident that
//the wanted range of sensitivity has been sampled
//Paste calibration values here:
//--------------------------------------------------------
float Ceil[9][9]={
{381.00,554.00,493.00,510.00,456.00,488.00,586.00,250.00,},
{380.00,553.00,494.00,509.00,455.00,487.00,586.00,249.00,},
{335.00,439.00,343.00,513.00,498.00,419.00,443.00,214.00,},
{397.00,555.00,431.00,597.00,578.00,487.00,620.00,252.00,},
{361.00,491.00,376.00,548.00,488.00,407.00,541.00,234.00,},
{246.00,276.00,321.00,290.00,275.00,251.00,277.00,284.00,},
{543.00,407.00,417.00,534.00,404.00,379.00,441.00,238.00,},
{433.00,452.00,369.00,451.00,404.00,439.00,517.00,228.00,},
};
float Floor[9][9]= {
{368.00,537.00,475.00,488.00,444.00,476.00,572.00,211.00,},
{369.00,541.00,476.00,488.00,443.00,475.00,573.00,210.00,},
{324.00,398.00,324.00,501.00,488.00,408.00,426.00,181.00,},
{389.00,541.00,405.00,591.00,574.00,476.00,611.00,207.00,},
{355.00,477.00,361.00,543.00,482.00,398.00,535.00,199.00,},
{228.00,251.00,295.00,265.00,251.00,225.00,247.00,275.00,},
{540.00,394.00,401.00,527.00,399.00,373.00,435.00,216.00,},
{429.00,434.00,357.00,442.00,398.00,432.00,510.00,201.00,},
};

//--------------------------------------------------------


int bin[9][5] = {    //Make repository of binary numbers from 0-15
  {0, 0, 0, 0},
  {0, 0, 0, 1},
  {0, 0, 1, 0},
  {0, 0, 1, 1},
  {0, 1, 0, 0},
  {0, 1, 0, 1},
  {0, 1, 1, 0},
  {0, 1, 1, 1}
};
```

```
int    j_low = 0; //for debugging purposes
int    i_low = 0;

float sensVal = 0;
float sensValTemp = 0;

void setup() {

  Serial.begin(500000);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);

  pinMode(8, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(10, OUTPUT);
  pinMode(11, OUTPUT);

while (!Serial) { }
}

float mapfloat(long x, long in_min, long in_max, long out_min, long
out_max)
{
  return (float)(x - in_min) * (out_max - out_min) / (float)(in_max
- in_min) + out_min;
}

void loop() {

while (!Serial.available()) {}


  for (int j = 0; j < 8; j++) {
    j_low = j ;  //error finding purposes

    digitalWrite(3, bin[j_low][2]);
    digitalWrite(4, bin[j_low][3]);
    digitalWrite(5, bin[j_low][1]);
    digitalWrite(6, bin[j_low][0]);

    //delay(2);

    for (int i = 0; i < 8; i++) {
      i_low = i ; //error finding purposes


      digitalWrite(8, bin[i_low][3]);
      digitalWrite(9, bin[i_low][2]);
      digitalWrite(11, bin[i_low][1]);
      digitalWrite(10, bin[i_low][0]);

     // delay(2);
```

```
      sensVal = analogRead(A1);
      delay(1);
      sensVal = analogRead(A1);   //Read twice to purge reciding
analog value.

      sensVal   =      mapfloat(sensVal   ,   Floor[j_low][i_low],
1.3*Ceil[j_low][i_low], 0, 10);  //map value for processing

      Serial.print(sensVal, 3);
      Serial.print(",");

    }
  }

  // Serial.print("n");
  Serial.print(";");
  Serial.print("\n");
 // delay(1);

}
```

A – 2: Arduino code for calibrating sensor skin data

```
/*CalibSkinScript
 * 08.06.2018
 * Håvard Nitter Vestad
 *
 * This script was written as part of my masters thesis
 * the spring of 2018 at MTP, NTNU.
 *
 * This script is intended to find the minimum and maximum
 * values of a 8x8 array of analog signals, sampled through
 * two CD74HC4067 Multiplexers. The script is run for a period
 * of time, in the stream which you want to sample. The maximum and
 * minimum value can then be directly copied from the serial monitor
 * and pasted into other scripts intended to display and use the
 * sensor readings.
*/
int bin[9][5] = {
  {0, 0, 0, 0},
  {0, 0, 0, 1},
  {0, 0, 1, 0},
  {0, 0, 1, 1},
  {0, 1, 0, 0},
  {0, 1, 0, 1},
  {0, 1, 1, 0},
  {0, 1, 1, 1}
};
int bin_j[5] = {0, 0, 0, 0};
int blank[5] = {0, 0, 0, 0};
float sensVal = 0;
float prevSensVal[9][9][3];
float CeilSensVal[9][9];
float FloorSensVal[9][9];
float sensValTemp = 0;
```

```
int    j_low = 0;
int    i_low = 0;



int val = 0;

void setup() {

  Serial.begin(500000);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);

  pinMode(8, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(10, OUTPUT);
  pinMode(11, OUTPUT);




  memset(FloorSensVal, 0, sizeof(FloorSensVal));   //make array of
zeros
  for (int j = 0; j < 8; j++) {
    for (int i = 0; i < 8; i++) {
      FloorSensVal[j][i] = 1000;
    }
  }
  //while (!Serial) {
  // }

}

float mapfloat(long x, long in_min, long in_max, long out_min, long
out_max)
{
  return (float)(x - in_min) * (out_max - out_min) / (float)(in_max
- in_min) + out_min;
}

void loop() {
  // while (!Serial.available()) {}

  //Serial.println(10, BIN);

for (int j = 0; j < 8; j++) {
    j_low = j ;  //error finding purposes

    digitalWrite(3, bin[j_low][2]);
    digitalWrite(4, bin[j_low][3]);
    digitalWrite(5, bin[j_low][1]);
    digitalWrite(6, bin[j_low][0]);

    delay(3);

    for (int i = 0; i < 8; i++) {
      i_low = i ; //error finding purposes


      digitalWrite(8, bin[i_low][3]);
      digitalWrite(9, bin[i_low][2]);
      digitalWrite(11, bin[i_low][1]);
      digitalWrite(10, bin[i_low][0]);

      delay(3);

      sensVal = analogRead(A1);
      sensVal = analogRead(A1);   //Read twice to purge reciding
analog value.

      //sensVal = sensValTemp - prevSensVal[j_low][i_low][1];
      if (sensVal > CeilSensVal[j_low][i_low]) {      //is value a
new high?
        CeilSensVal[j_low][i_low] = sensVal;
      }
      if (sensVal < FloorSensVal[j_low][i_low]) {  //is value a new
low?
        FloorSensVal[j_low][i_low] = sensVal;
      }
      //delayMicroseconds(4);
    }
  }

  Serial.print("float Ceil[9][9]={ ");
  Serial.print("\n");

  for (int j = 0; j < 8; j++) {
    Serial.print("{");
    for (int i = 0; i < 8; i++) {
      Serial.print( CeilSensVal[j][i], 2);
      Serial.print(",");
    }
    Serial.print("},");
    Serial.print("\n");
  }

  Serial.print("};");
  Serial.print("\n");

  Serial.print("float Floor[9][9]= { ");
  Serial.print("\n");

  for (int j = 0; j < 8; j++) {
    Serial.print("{");
    for (int i = 0; i < 8; i++) {
      Serial.print( FloorSensVal[j][i], 2);
      Serial.print(",");
    }
    Serial.print("},");
    Serial.print("\n");

  }

  Serial.print("};");
  Serial.print("\n");
  // delay(100);
}
```

A – 3: Arduino code for reading and plotting PCB Hair Cell data

```
/*TestHairCell
 * 08.06.2018
 * Håvard Nitter Vestad
 *
 * This script was written as part of my masters thesis
 * the spring of 2018 at MTP, NTNU.
 *
 * This script is intended to show the analog feedback of
 * a PCB based hair cell module with piezo resistive properties.
 * The four position the four hairs is described by its analog
 * voltage sampled by a 16-channel multiplexer. Where sensor 0-3 is
 * hair 1, 4-7 is har 2, 8-11 is hair 3 and 12-15 is hair 4.
 * The values ar then plotted as values in an 8x8 array, in row 4
 * and 5, and columns 1-8, where the remaining values are set to 0.
*/


int bin[17][5] = {
  {0, 0, 0, 0},
  {0, 0, 0, 1},
  {0, 0, 1, 0},
  {0, 0, 1, 1},
  {0, 1, 0, 0},
  {0, 1, 0, 1},
  {0, 1, 1, 0},
  {0, 1, 1, 1},
  {1, 0, 0, 0},
  {1, 0, 0, 1},
  {1, 0, 1, 0},
  {1, 0, 1, 1},
  {1, 1, 0, 0},
  {1, 1, 0, 1},
  {1, 1, 1, 0},
  {1, 1, 1, 1}
};
int bin_j[5] = {0, 0, 0, 0};
int blank[5] = {0, 0, 0, 0};
float sensVal = 0;
float sensValTemp = 0;

int    j_low = 0;
int    i_low = 0;
int val = 0;

void setup() {

  Serial.begin(500000);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);

  pinMode(8, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(10, OUTPUT);
  pinMode(11, OUTPUT);



  memset(FloorSensVal, 0, sizeof(FloorSensVal));   //make array of
zeros
  for (int j = 0; j < 8; j++) {
    for (int i = 0; i < 8; i++) {
      FloorSensVal[j][i] = 1000;
    }
  }
  while (!Serial) {}

}
```

```
float mapfloat(long x, long in_min, long in_max, long out_min, long
out_max)
{
  return (float)(x - in_min) * (out_max - out_min) / (float)(in_max
- in_min) + out_min;
}

void loop() {

  while (!Serial.available()) {}
  // while (!Serial.available()) {}

  for (int j = 0; j < 24; j++) {
    sensVal = 0;
    Serial.print(sensVal, 3);
    Serial.print(",");
  }

  for (int j = 0; j < 16; j++) {
    j_low = j ;  //error finding purposes

    digitalWrite(11, bin[j_low][3]);
    digitalWrite(10, bin[j_low][2]);
    digitalWrite(9, bin[j_low][1]);
    digitalWrite(8, bin[j_low][0]);

    delay(1);

    sensVal = analogRead(A1);
    sensVal = analogRead(A1);  //Read twice to purge receding analog
value.
    sensVal =  mapfloat(sensVal , 50, 400 , 0, 10); //map value for
processing

    Serial.print(sensVal, 3);
    Serial.print(",");

  }
  for (int j = 0; j < 24; j++) {
    sensVal = 0;
    Serial.print(sensVal, 3);
    Serial.print(",");
  }
  Serial.print("\n");

}
```

A – 4: Arduino code for reading sensor skin data and writing it to the serial port

```
/*SkinData-SerialCommunicator
 * 08.06.2018
 * Håvard Nitter Vestad
 *
 * This script was written as part of my masters thesis
 * the spring 2018 at MTP, NTNU.
 *
 * This script is intended to format the data from a 8x8
 * array of analog signals, sampled through two CD74HC4067
 * Multiplexer, so that the printed information in the
 * serial monitor can be read by the processing app and
 * displayed as a heat-map.
*/




//--------------- Callibration values -------------------
//Before running the scrip, run the callibration scrip
//Let the callibration run untill you are confident that
//the wanted range of sensitivity has been sampled
//Paste callibration values here:
//-------------------------------------------------------

float Ceil[9][9]={
{460.00,459.00,458.00,458.00,460.00,459.00,459.00,459.00,},
{455.00,455.00,454.00,454.00,456.00,456.00,455.00,455.00,},
{452.00,451.00,451.00,450.00,452.00,452.00,452.00,452.00,},
{449.00,448.00,448.00,448.00,449.00,449.00,449.00,449.00,},
{444.00,444.00,444.00,443.00,445.00,445.00,444.00,444.00,},
{441.00,441.00,440.00,440.00,442.00,442.00,441.00,441.00,},
{438.00,437.00,436.00,436.00,438.00,438.00,438.00,438.00,},
{434.00,434.00,434.00,433.00,435.00,435.00,435.00,435.00,},
};
float Floor[9][9]= {
```

```
{225.00,225.00,225.00,225.00,227.00,227.00,227.00,227.00,},
{224.00,224.00,224.00,224.00,226.00,226.00,226.00,226.00,},
{223.00,223.00,223.00,223.00,225.00,225.00,225.00,225.00,},
{223.00,223.00,223.00,223.00,225.00,225.00,225.00,225.00,},
{222.00,222.00,221.00,222.00,224.00,224.00,224.00,224.00,},
{221.00,221.00,221.00,222.00,224.00,224.00,224.00,224.00,},
{221.00,221.00,220.00,221.00,223.00,223.00,223.00,223.00,},
{221.00,221.00,221.00,221.00,223.00,223.00,223.00,224.00,},
};

//-------------------------------------------------------

int bin[9][5] = {
  {0, 0, 0, 0},
  {0, 0, 0, 1},
  {0, 0, 1, 0},
  {0, 0, 1, 1},
  {0, 1, 0, 0},
  {0, 1, 0, 1},
  {0, 1, 1, 0},
  {0, 1, 1, 1}
};



int bin_j[5] = {0, 0, 0, 0};
int blank[5] = {0, 0, 0, 0};
float sensVal = 0;
float sensValTemp = 0;

int    j_low = 0;
int    i_low = 0;



int val = 0;

void setup() {
  Serial.begin(500000);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);

  pinMode(8, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(10, OUTPUT);
  pinMode(11, OUTPUT);

while (!Serial) { } //Remove to run serial plotter
}
float mapfloat(long x, long in_min, long in_max, long out_min, long
out_max)
{
  return (float)(x - in_min) * (out_max - out_min) / (float)(in_max
- in_min) + out_min;
}

void loop() {

while (!Serial.available()) {} //remove to run serial plotter


  for (int j = 0; j < 8; j++) {
    j_low = j ;  //error finding purposes

    digitalWrite(3, bin[j_low][2]);
    digitalWrite(4, bin[j_low][3]);
    digitalWrite(5, bin[j_low][1]);
    digitalWrite(6, bin[j_low][0]);

    delay(3);

    for (int i = 0; i < 8; i++) {
      i_low = i ; //error finding purposes
      digitalWrite(8, bin[i_low][3]);
      digitalWrite(9, bin[i_low][2]);
      digitalWrite(11, bin[i_low][1]);
      digitalWrite(10, bin[i_low][0]);

      delay(3);

      sensVal = analogRead(A1);
      delay(1);
      sensVal = analogRead(A1);   //Read twice to purge reciding
analog value.

      sensVal = mapfloat(sensVal , Floor[j_low][i_low],
1.3*Ceil[j_low][i_low], 0, 10);  //map value for processing

      Serial.print(sensVal, 3);
      Serial.print(",");
    }
  }
  // Serial.print("n");
  Serial.print("\n");
}
```

APPENDIX B

Processing Codes

B – 1: Processing code for drawing a heat map from serial port data

```
//---------------------------------------------
// HeatMap
// Hå vard Vestad
// 09.06.2018
//
// This script was made as part of my master thesis project
// To visualize the sensor output of an 8x8 pressure sensor
// skin.
// The script is a combination of example code segments by
// forum.processing.org users MitchR(2018)
// and quark(2016) which have been modified to fit with the
// data as i print it from  arduino serial.
//
// sources:
//                          quark,                          2016,
https://forum.processing.org/two/discussion/20004/heat-map-color-
resolution
//                         MitchR,                          2018,
https://forum.processing.org/two/discussion/26588/how-to-simplify-
this-code-heat-map

//---------------------------------------------
//Main body as described by MitchR(2018)
//---------------------------------------------
import processing.serial.*;

Serial myPort;
String myString;

int j;
int m = 0;

int r = 8;  // number of rows in input array
int c = 8;  // number of columns in input array
int t = 200;  // parameter (array resize factor)
int rows = (r-1)*t;  // height of the heat map
int cols = (c-1)*t;  // width of the heat map

float[][] array = new float[r][c];  // input array
float[][] interp_array = new float[rows][cols]; // interpolated array
String[] list = new String[r*c];

void settings()
{
  size(cols, rows);
}

void setup()
{

  printArray(Serial.list());

  myPort = new Serial(this, Serial.list()[0], 500000);

  myPort.write(65);
  myPort.write(65);
  myPort.write(65);


  noStroke();
}

void draw()
{
  myPort.write(65);

  while (myPort.available() > 0 ) {

    //Expand array size to the number of bytes you expect
    byte[] inBuffer = new byte[1024];
    myPort.readBytesUntil('\n', inBuffer);
    myString = new String(inBuffer);
    list = split(myString, ',');
    for (int i = 0; i < (list.length)/8; i++) {
      for (j = 0; j < (list.length)/8; j++) {
        array[j][i] = float(list[m]);
        m++;
      }
    }
    m = 0;
  }
  bilinearInterpolation();  //these are IN the while loop
  applyColor();
}



void bilinearInterpolation() {  // Bi-linear Interpolation algorithm

  for (int i=0; i<r; i++) {
    for (int j=0; j<c; j++) {
      int x = j*t - 1;
      int y = i*t - 1;
      if (x<0)
        x=0;
      if (y<0)
        y=0;
      interp_array[y][x] = array[i][j];
    }
  }

  for (int y=0; y<rows; y++) {
    int dy1 = floor(y/(t*1.0));
    int dy2 = ceil(y/(t*1.0));
    int y1 = dy1*t - 1;
    int y2 = dy2*t - 1;
    if (y1<0)
      y1 = 0;
    if (y2<0)
      y2 = 0;
    for (int x=0; x<cols; x++) {
      int dx1 = floor(x/(t*1.0));

      //interp_array[y][x]=array[dy1][dx1];
      int dx2 = ceil(x/(t*1.0));
      int x1 = dx1*t - 1;
      int x2 = dx2*t - 1;
      if (x1<0)
        x1 = 0;
      if (x2<0)
        x2 = 0;
//-------------------------------
//Choose interpolation or not
//-------------------------------
      //float q11 = array[dy1][dx1];
      //float q12 = array[dy2][dx1];
      //float q21 = array[dy1][dx2];
      //float q22 = array[dy2][dx2];
//-------------------------------
      float q11 = array[dy1][dx1];
      float q12 = array[dy1][dx1];
      float q21 = array[dy1][dx1];
      float q22 = array[dy1][dx1];
//--------------------------------

      int count = 0;
      if (q11>0)
        count++;
      if (q12>0)
        count++;
      if (q21>0)
        count++;
      if (q22>0)
```

```
        count++;

      if (count>2) {
        if (!(y1==y2 && x1==x2)) {

          float t1 = (x-x1);
          float t2 = (x2-x);
          float t3 = (y-y1);
          float t4 = (y2-y);
          float t5 = (x2-x1);
          float t6 = (y2-y1);

          if (y1==y2) {
            interp_array[y][x] = q11*t2/t5 + q21*t1/t5;
          } else if (x1==x2) {
            interp_array[y][x] = q11*t4/t6 + q12*t3/t6;
          } else {
            float diff = t5*t6;
            interp_array[y][x] = (q11*t2*t4 + q21*t1*t4 + q12*t2*t3
+ q22*t1*t3)/diff;
          }
        } else {
          interp_array[y][x] = q11;
        }
      } else {
        interp_array[y][x] = 0;
      }
    }
  }
}



//----------------------------------------------
// applyColor function as described by processing.org user MitchR
// December 2016
//        https://forum.processing.org/two/discussion/26588/how-to-simplify-this-code-heat-map
//----------------------------------------------


//void applyColor() {  // Generate the heat map

//  color c1 = color(0, 0, 255);  // Blue color
//  color c2 = color(0, 255, 0);  // Green color
//  color c3 = color(255, 255, 0);  // Red color
//  color c4 = color(255, 0, 0);  // Yellow color

//  for (int i = 0; i < rows; i++) {
//    for (int j = 0; j < cols; j++) {
//      float value = interp_array[i][j];
//      color c;
//      float fraction;

//      if (value>=0 && value<2) {
//        fraction = (value)/2.0;
//        c = lerpColor(c1, c2, fraction);
//      } else if (value>=2 && value<3) {
//        fraction = (value-2)/1.0;
//        c = lerpColor(c2, c3, fraction);
//      } else if (value>=3 && value<7) {
//        fraction = (value-3)/2.0;
//        c = lerpColor(c3, c4, fraction);
//      } else
//        c = c4;
//      stroke(c);
//      point(j, i);
//    }
//  }
//}

//----------------------------------------------
// applyColor function as described by processing.org user quark
// December 2016
//      https://forum.processing.org/two/discussion/20004/heat-map-
color-resolution
//---------------------------------------------
void applyColor() {  // Generate the heat map
  pushStyle(); // Save current drawing style
  // Set drawing mode to HSB instead of RGB
  colorMode(HSB, 1, 1, 1);
  loadPixels();
  int p = 0;
  for (int r = 0; r < height; r++) {
    for (int c = 0; c < width; c++) {
      // Get the heat map value
      float value = interp_array[c][r];
      // Constrain value to acceptable range.
      value = constrain(value, 0, 3);
      // Map the value to the hue
      // 0.2 blue
      // 1.0 red
      value = map(value, 0, 1, 0.2, 1.0);
      pixels[p++] = color(value, 0.9, 1);
    }
  }
  updatePixels();
  popStyle(); // Restore original drawing style
}
```

APPENDIX C

MATLAB Codes

C – 1: MATLAB code for plotting PSD analysis of data arrays

```
%%
% Written By Håvard Vestad
% As part of Masters Thesis at MTP, NTNU
% 19.05.18
%
% This script preforms a PSD analysis of data from piezoresistive
hydrofoil
% To determine correlation between observed data and flow conditions
% Using Welch PSD and FFT

%%
load dataMatrices;

delay=1*100;
Fs = 1000/delay;            % Sampling frequency     10hz
T = 1/Fs;             % Sampling period

data_array1=[data_1000hz_no(1:3000,:);data_1000hz_no(1:3000,:)];
data_array2=data_1000hz_5vLam(1:6000,:);
data_array3=data_1000hz_5vKar(1:6000,:);
data_array4=data_1000hz_12vLam(1:6000,:);
data_array5=data_1000hz_12vKar(1:6000,:);

data_array1=data_nn_no_5min(1:2000,:);
data_array2=data_nn_5vLam_5min(1:2000,:);
data_array3=data_nn_5vKar_5min(1:2000,:);
data_array4=data_nn_12vLam_5min(1:2000,:);
data_array5=data_nn_12vKar_5min(1:2000,:);

% Cut data-sets to equal length(no flow is doubbled to equal the length
for
% plotting purposes

start=3;
startf=50;



%%



for i= 42
   %2:8:18

    X1 = data_array1(:,i)';
     X2 = data_array2(:,i)';
      X3 = data_array3(:,i)';
       X4 = data_array4(:,i)';
        X5 = data_array5(:,i)';

    X1 = detrend(X1);
    X2 = detrend(X2);
    X3 = detrend(X3);
    X4 = detrend(X4);
    X5 = detrend(X5);

    L=length(X2);
    f =0:Fs/L:Fs/2;
    t = (0:L-1)*T;



    [Y1,f1] = pwelch(X1,200,0,[],Fs,'psd');
    [Y2,f2] = pwelch(X2,200,0,[],Fs,'psd');
    [Y3,f3] = pwelch(X3,200,0,[],Fs,'psd');
    [Y4,f4] = pwelch(X4,200,0,[],Fs,'psd');
    [Y5,f5] = pwelch(X5,100,0,[],Fs,'psd');



    hold on
    %no flow
    %figure(1);



    nf= subplot(3,2,1);

%     Yf1 = fft(X1);
%     Yf1 = Yf1(1:L/2+1);
%     P11 =(1/Fs*L)*abs(Yf1).^2;
%     P11(2:end-1)=2*P11(2:end-1);



Yf1 = fft(X1);
Yf1 = Yf1(1:L/2+1);
P11 = (1/(Fs*L)) * abs(Yf1).^2;
P11(2:end-1) = 2*P11(2:end-1);


p1=plot(nf,f(startf:end),P11(startf:end/1),f1(start:(end)),Y1(start:end))
;
    p1(1).LineWidth=0.5;
    p1(2).LineWidth=2;
    legend('FFT','Welch Method')


    xlabel(nf,'f (Hz)')
    ylabel(nf,'PSD')
    title(nf,'No Flow')

    %5v laminar flow
    fl5=subplot(3,2,2);

    Yf2 = fft(X2);
    Yf2 = Yf2(1:L/2+1);
    P12 = (1/(Fs*L)) * abs(Yf2).^2;
    P12(2:end-1) = 2*P12(2:end-1);


p2=plot(fl5,f(startf:end/1),P12(startf:end/1),f2(start:(end)),Y2(start:en
d));
    p2(1).LineWidth=0.5;
    p2(2).LineWidth=2;

    legend('FFT','Welch Method')
    xlabel(fl5,'f (Hz)')
    ylabel(fl5,'PSD')
    title(fl5,'5v Laminar Flow')

    %5v Karman street
    fk5=subplot(3,2,3);

    Yf3 = fft(X3);
    Yf3 = Yf3(1:L/2+1);
    P13 = (1/(Fs*L)) * abs(Yf3).^2;
    P13(2:end-1) = 2*P13(2:end-1);


p3=plot(fk5,f(startf:end/1),P13(startf:end/1),f3(start:(end)),Y3(start:en
d));
    p3(1).LineWidth=0.5;
    p3(2).LineWidth=2;

    legend('FFT','Welch Method')
    xlabel(fk5,'f (Hz)')
    ylabel(fk5,'PSD')
    title(fk5,'5v Karman Street')

    %12v laminar flow

    fl12=subplot(3,2,4);

    Yf4 = fft(X4);
    Yf4 = Yf4(1:L/2+1);
    P14 = (1/(Fs*L)) * abs(Yf4).^2;
    P14(2:end-1) = 2*P14(2:end-1);


p4=plot(fl12,f(startf:end/1),P14(startf:end/1),f4(start:(end)),Y4(start:e
nd));
    p4(1).LineWidth=0.5;
    p4(2).LineWidth=2;

    legend('FFT','Welch Method')
    xlabel(fl12,'f (Hz)')
    ylabel(fl12,'PSD')
    title(fl12,'12v Laminar Flow')

    %12v Karman street

    fk12=subplot(3,2,5);

     Yf5 = fft(X5);
    Yf5 = Yf5(1:L/2+1);
    P15 = (1/(Fs*L)) * abs(Yf5).^2;
    P15(2:end-1) = 2*P15(2:end-1);


p5=plot(fk12,f(startf:end/1),P15(startf:end/1),f5(start:end),Y5(start:end
));
```

```
    p5(1).LineWidth=0.5;
    p5(2).LineWidth=2;

    legend('FFT','Welch Method')
    xlabel(fk12,'f (Hz)')
    ylabel(fk12,'PSD')
    title(fk12,'12v Karman Street')
    suplabel('Power Spectral Density Estimate of Single Sensor n=6000,
fs=10Hz','t');



end
%%
```

APPENDIX D

Pre-Master’s Project Thesis

Appendix D-1– Project Thesis

Håvard Vestad, NTNU MTP

Biomimicry: Learning from fish for improved propulsion systems in water

Project Report
Trondheim, Autumn 2017
Supervisor: Martin Steinert

NTNU
Kunnskap for en bedre verden

Appendix D-1– Project Thesis

Abstract

Biomimicry, where biology meets engineering, is an increasingly used tool for finding new solutions in product development through inspiration from evolutionary solutions. In this project report biomimicry will be used to gather inspiration from fish and the mechanics fish use for efficient propulsion in water.

To study fish mechanics in moving waters, a water tunnel was built to create a laminar flow with a hydrogen bubble generator to visualize the flows. The water tunnel was then used to divergently test potential solutions found in literature such as drag resistance of fish-skin, muscle like actuation, swimming modes and flow controlling/sensing mechanics.

The learnings from the divergent testing were used to determine that *Kármán gaiting*, the effect where fish are able to hold a stationary position in vortex streets behind objects in flows, could hold untapped potential for learning and application in engineering. This was further investigated and tested by generating a predictable vortex street in the water tunnel and deploying prototypes that set out to test the effect's dependencies on: Stiffness, eigenfrequencies, freedom of movement and actuation. Advanced prototypes were less able to show positive effects in the Kármán street than simple models. Prototypes with high freedom to move and low stiffness of their bodies gave the results that best fit with Kármán gaiting model, in one case showing reduction in energy consumption for stationary position holding in the Kármán street of 17% as compared to in free laminar flow. Being able to properly mimic Kármán gaiting and learning from the behaviour of fish it could be possible to use these findings to create propulsion systems that adapt and actively use flows to reduce their energy consumption much more efficiently than current systems.

Appendix D-1– Project Thesis

Acknowledgements

This project thesis is the result of a challenge given by ProtoMore. Thanks to the research community at TrollLabs NTNU, Achim Gerstenberg and Martin Steinert for providing guidance and an environment in which ideas grow. Thank you to Chittiappa Muthanna and the others at SINTEF Ocean, for insights and discussions on water tunnels as well as providing me glass spheres for indication medium. Thank you to Ana Silva at *Norsk institutt for naturforskning* for good information and talks about fish mechanics.

Contents

LIST OF FIGURES

1 Introduction

Challenges and problems are a natural part of our everyday lives. And overcoming problems through finding solutions is how we move forward. Humans are natural problem solvers, we can think rationally and through communication and literature we are able base our actions on the experiences of generations of humans before us. The latter although an amazing tool can also be a source of great solution bias; If we know of a way that has worked for others previously why should we put the effort in to find a new way? In the search for innovation and freeing oneself from solution biases, many methods have been suggested to the field of *product development*, one of these is *biomimicry*.

Biomimicry is the art of taking solutions from nature and bringing the functionality of the solution into a product. Solutions found in nature are the results of millions of years of trial and error through evolution (or if you want to go back to the first common ancestor: 3.5-3.8billion years(Doolittle, 2000)). Through time, exposure to different environments has created species specialized for their specific environment and situation through natural selection of the fittest, crossing and randomization. The resulting solutions might not be the global optimal, but it is a way for the organism to function in its environment, and with no designer the solutions will be free of the associated creator biases.

Mimicking nature is not something new, arguably it is the foundation on which the modern human developed: We watered our crops to simulate rain, we used animal skins to keep warm, but at some point we stopped asking nature for answers and started basing our designs on our own technical portfolio of proven concepts and ideas or generated ideas from our understanding of the natural world.

1.1 Biomimicry

A frequently used definition of biomimicry is the act of taking a biological model and applying it to a human problem. In his book *Biomimicry* from 1997 Benryus (1997) refers to it as a "*New technology"*, suggesting the field of biomimicry, only 20 years old, might still be in a juvenile state. However, we also recognize that mimicking nature has taken place even before it got a name, and that the "new technology" might just be a classification or trial of methodological approach to something that has already been going on for a long time:

Appendix D-1– Project Thesis

Many of the early designs for aircrafts such as *Da Vinci's flying machine,* had close resemblance to birds, with flapping wings and anatomically correct structures. Although the success of these flying machines was questionable, we can see that even in the early days of engineering, mimicry played an important role when exploring concepts that were not yet understood. From more recent history, the invention of *Velcro*® is often quoted to be the first true biomimetic product. Gonzales (2011) gives a good recap of the history of George de Mestrals discovery of the *Velcro*® concept. From noticing how the bur seeds stuck to his dog's fur in the 40s, investigating and understanding the "hooks" on the seeds under a microscope and finally patenting the idea in 1951.

1.1.1 Increased ability to observe nature

Gonzales (2011) credits a lot of the recent interest and development of biomimicry to our increased ability to observe nature. We have better microscopes, and the phenomena behind specific solutions can be better studied and researched. Some examples of solutions that have come from this better observational technology are the *Lotus effect* (Barthlott & Neinhuis, 1997) and the geckos sticking abilities((S. Kim et al., 2008). The lotus effect refers to the hydrophobic and self-cleaning properties of the lotus leaves. It was discovered trough electron microscopes that the leaves had a rough surface on the Nano level, this makes it so that dirt has very little surface contact with the leaves and is easily rinsed of. Among other effects it also causes air to be trapped beneath liquids which greatly decreases the wettability of the leaves. These findings have been used to make water resistant clothing and self-cleaning paint.

Likewise, the gecko's ability to climb and stick to things has been researched with *Scanning electron microscopes*. It was found that the gecko did not use capillary adhesion such as insects and frogs but *Van der Waals* forces, generated by high amounts of microscopic hairs. These forces increase with the surface area, which means that the gecko can stick and unstick itself depending on how the foot is loaded. This type of adhesion is not dependent on air or a medium, and will stick to any surface. This has made it an interesting case for under water and even space applications (Jiang et al., 2017).

1.1.2 Recreation and testing

Although the increased ability to observe effects is a great tool in biomimicry, some discoveries also seem to be based on trial and error rather than generating an actual understanding beforehand. An example of this is *Mercedes-Benzes* concept car "*Bionic*" from 2005 (Buehler & Patel, 2015). The car drew inspiration from the Boxfish. The boxfish while bulky and square in appearance still has decent hydrodynamic properties. Additionally, *Mercedes-Benz* theorized that the shape made the fish self-stabilizing, which would be great in car applications as well. The shape of the boxfish was then transferred to a car design, with testing in wind tunnels. The resulting car was a car with generous amount of room on the inside, while still maintaining a drag coefficient of only 0.19("Fish-Inspired Car," n.d.), which is extremely low. Later it has later been found through studies of the aerodynamic effects on the boxfish that Mercedes-Benz had their theories wrong, and that the boxfish does not have inherit stabilizing properties but rather amplifies destabilizing effects to increase manoeuvrability (Farina & Summers, 2015). This goes to show, that even with a lack of understand of effects, mimicking can still be beneficial; Mercedes-Benz did create an extremely aerodynamic car with spacious interior. The car did not however have the theorized self-stabilizing properties, meaning that mimicry can only be beneficial to a certain point without understanding. For further specialization and use of the effects different fields, some understanding is often needed.

Another example where recreation(experimental research) came before a deep understanding is the turbulence and aero-/hydrodynamic properties of the leading edge of humpback whale flippers(F. E. Fish, Weber, Murray, & Howle, 2011). The flippers have a rough turbulence inducing leading edge, which experimental research has shown can increase lift and performance of wings and foils by delaying separation. The effect has been used to create wind-turbine blades, with a lift increase of 8% ("Humpback whales inspire next generation wind turbine technology • Materia," n.d.) while some tests show a 20% higher efficiency over time("Testing and Third Party Testing," 2014).

1.1.3 Find, Understand, Recreate and Specialize

The common concepts between these examples are that they to some degree all include finding a solution, an understanding of the solution, a recreation of the solution and then using this solution in a specialized setting. This might be what biomimicry process really boils down to. Biomimicry is not just about recreating something you see in nature, there also needs to be an understanding of how it works and a refinement of the solution to a specialized case for it to be a successful process. It might not always be a linear process, sometimes you are able to recreate something before you really understand what's going on, like with the humpback whale turbine blades and boxfish car, but to some degree successful biomimicry projects usually involve these stages.

We can see that not only does this describe a long process, it is also a process which goes through many scientific fields, from research to development. For me to undertake it as a whole during this six-month project would be optimistic. From a mechanical engineering perspective, the specialization and recreation stages might be the most interesting and yielding. Using accessible research and information, it is possible to use the findings of others and to focus on the later stages of the biomimicry process.

1.2 Ideation and finding a problem

This project was defined to fit a request made by *ProtoMore*. ProtoMore wished for a project that used biomimicry in the early stages of the product development process, and illustrated the benefits and challenges of this method in developing new products.

Although ProtoMore communicated to us that they wanted a project based on biomimicry they gave us no restrictions as to in which direction to run with this. We were free to form the project as we wished (which itself is a challenge). This lead to the first stages of the project being about finding our challenge, and we did some initial ideation sessions to find what we felt was a representative challenge for the involved stakeholders (ProtoMore, TrollLabs, Me).

1.2.1 Biomimicry that fits ProtoMore and the Norwegian coastal market

ProtoMore is a workshop/lab for prototyping at *Kunskapsparken* in Molde, Norway. The owners of ProtoMore are local industries, a total of 36, which are mostly maritime. The goal of ProtoMore is to inspire existing industry partners as well as start-ups to prototype and try out ideas, and they are doing this through workshop-sessions and facilitating prototyping in their lab. As such my project might not need to result in a product, but the process itself and the learnings of the process might be valuable assets for ProtoMore as well.

Another important note is that because of ProtoMore’s location at the west coast of Norway, ProtoMore’s entities are mostly from the marine industry and the project should reflect this.

1.3 The challenge

“Gray’s Paradox” is one of the popular problems that for a long time has puzzled researchers in fluid dynamics and marine biology (Bale, Hao, Bhalla, Patel, & Patankar, 2014). The problem comes from the research of James Gray (1936) and his estimate that the total muscle forces exerted by a dolphin were lower than the total drag forces working against it during swimming, which of course should be impossible. Although it is now considered to be resolved (F. e. Fish & Lauder, 2005), the paradox is a good illustration of how even with a

high density of research the totality of some of the complex solutions in nature might be hard to recreate and understand.

The same can be said about fish and other aquatic animals, the understanding of how they work and propel them self is a popular research field, and although most of their mechanics are well documented and understood(Sfakiotakis, Lane, & Davies, 1999), the recreation, use and mimicking of the found research seems to still be in a juvenile stage. Not only are some of the swimming modes like the *Thunniform swimming mode*(Sfakiotakis et al., 1999) extremely energy efficient over long distances, fish also have the ability to sense the flow situation in the waters around them as well as objects and use the situation to their advantage by swimming in the bow and wakes of others or in the right flow situation(F. e. Fish & Lauder, 2005).

If I can take the research from one or some of the mechanics that make fish efficient propulsion machines, and test ways to mimic the solution through prototyping and prototyping methods, a deeper learning into the functionality of said mechanics might be achieved, focusing on the understand and recreation stages of the biomimicry process.

I therefor wish to use fish and aquatic animals for mimicry inspiration to uncover mechanics that aid their efficient propulsion in water, and through product development tools understand and recreate the effects. The product will be the knowledge produced through this process. Research from biology will be used for inspiration and information throughout the process to aid in decision making.

2 Method

Biomimicry is a combination of biology and engineering. Through this project I will explore how to use this connection from an engineering perspective, to further build knowledge and learnings by investigating existing information and understandings through product developments tools. One of the most powerful learning tools in product development is prototyping.

2.1 Prototyping

There is no correct way to single correct way to use or create prototypes. According to Ulrich and Eppinger (2012) prototypes are tools for: communication, integration, milestones and **learning**. I will mainly focus on the latter as the setting in which I deploy my prototypes will be to answer questions. As the project is not defined with a specific direction or desired outcome, but simply with a ballpark of a solution space, an innovation model that might fit the project is a modification of the wayfaring/hunter-gatherer model as explained by Steinert and Leifer (2012). Where probing with multiple prototypes is done to find and learn the direction that is best to move forward. The high degree of unknowns in my solution space would suggest that for the initial phases of testing, a wide spectrum of solutions would need to be tested to find a direction.

2.2 Front loading and set based designs

Front loading is a theory typically deployed in lean product development which focuses on investing more resources and gathering more information and knowledge early in the product development process to take right decisions later (Thomke & Fujimoto, 2000). A way to do this is through *set based design*. A set based design model focuses on splitting tasks into subsystems of problems and generating multiple alternative solutions for the identified subsystems (Kennedy, Sobek, & Kennedy, 2014). Alternatives are discontinued when testing prove them unviable the set will converge to a single solution. This is a process that requires high amounts of resources, but is great for investigating a solution space in which you have little prior knowledge, such as in my case, as it is a knowledge generating process. I will not blindly follow the philosophy of set-based design in my process but like front loading, doing high amounts of simple divergent/directional prototypes might be the best approach early in my process to be able to make decisions later and converge towards a viable solution. An example of a successful use of set-based thinking is the wright brothers and their prototyping of flying machines. Through making a good testing environment for prototyping several simple solutions fast and with low investments they were able to succeed where others failed. A quote by Wilbur Wright summarizes this:

> In any case, as famous as we became for our "Flyer" and its system of control, it all would never have happened if we had not developed our own wind tunnel and derived our own correct aerodynamic data.
> (American Institute of Aeronautics & Astronautics (AIAA), 2017)

2.3 Creating a test environment that allows for rapid probing and testing of prototypes

To drastically generate knowledge of my solution space, I decided early on to focus on creating a testing environment that allowed for fast deployment and observation of several simple prototypes in water. The main aim at this point was to observe and compare potential candidates for mimicry to converge on. I therefore decided to make a small scale, see-through water tunnel. In which movement of water could be simulated, and thus by extension through the laws of relativity, movement of objects through water.

Although not common practice, the construction of a personal water tunnel is not necessarily very complicate. In his book *"Life in moving fluids"* Vogel (1983) goes through several possible designs for water and wind tunnels, emphasizing on how to move the water and conditioning it. For the case of my prototype tank, the most rational approach seems to be using commercially available pumps, and rather condition the flow post pumping.

2.3.1 Constructing the tank

I began constructing a tank using tools and materials easily available, thus allowing for iterations further down the line should it be needed. A rectangular design was chosen to keep the production rapid and possible to create with a laser cutter. Should the design prove unsuitable a new iteration may be made. The tunnel was designed with considerable length as compared to height and width, this way the tunnel had both a boundary layer zone up stream, while also allowing the flow to develop downstream. The entrance length for the boundary layer to fully develop depends on the turbulence of the flow. For ease of calculation the turbulent entrance length is used. The length of the entrance layer is described by Cimbala and Çengel (2006) with Equation 2.1. Equation 2.2 approximates Equation 2.1 for turbulent flow, where *D* is the characteristic dimension. For my square tunnel the characteristic dimension is described by Equation 2.4 as four times the cross-sectional area divided by the wetted perimeter of the area.

$$L_{h,laminar} = 0.05 R_e D$$

Equation 2.1 - Boundary layer development length

$$L_{h,turbulent} \approx 10D$$

Equation 2.2 - Approximated development length

With an open top tank with a square cross-section, the length should be approximately 13 times longer than the height/width. A laser cutter was used to cut sheets of 6mm Plexiglas to dimensions, with interlocking dovetails around the edges, for improved rigidity and ease of assembly. The cutting area of the laser cuter was the bottle-neck for the dimensions of the tank, thus the tank ended up being 1200mm long, 240mm high (water filled to 120) and 120mm wide, as the total length of the cutter was 1220mm. The relationship between height and length became 10 rather than 13, as a thinner design would not have allowed room for sufficiently large pumps.

Two structural beams were cut for the middle of the tank, to reduce bulging of the tank which would influence the water-flow.

The tank was fist quickly assembled and fit-tested with tape. Once it was confirmed hat all pieces fit together correctly, and no further filing or sanding was needed the tank was fixed using generous amounts of *Artifix®2R 0190* acrylic-glue along all joints. The glue, with slight cavity-filling properties, near sealed the tank. When the glue had fully dried, I applied small amounts of glass silicone along all the inner edges of the tank, I took especially care to keep the radii left behind consistent and small to reduce the effects on the flow.

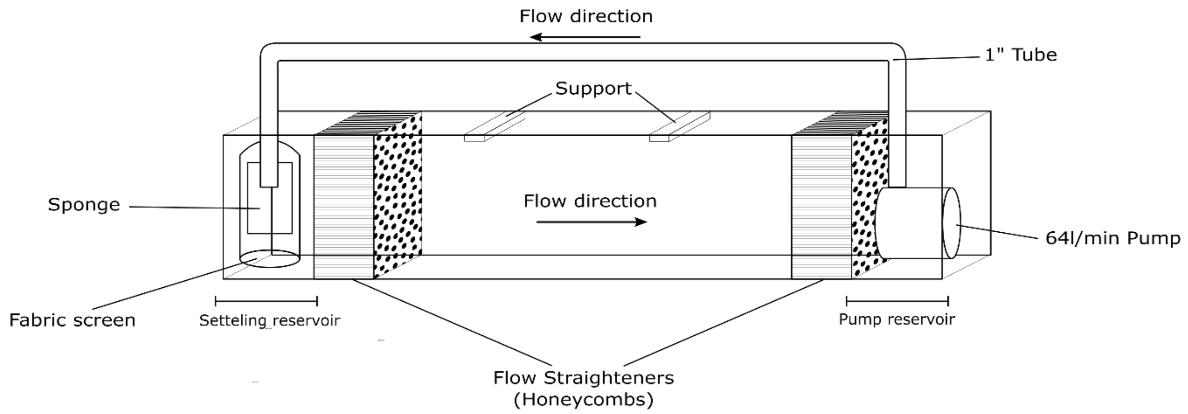


Figure 2.1 - Sketch of flow tunnel

2.3.2 Flow conditioning

As found by Scheiman and Brooks (1981)the best reduction of turbulence is through a combination of both honeycomb(HC) structure and screens. They found that screens had the greatest effect in reducing the axial turbulence, while HC reduced the lateral turbulence. The combination yielded a potential reduction of turbulence to 10% that of the original turbulence, although some literature seems to suggest even higher reduction rates.

The first iteration of the tunnel included flow straighteners both after the settling reservoir, and downstream before the pump reservoir as well. The latter did not necessarily have much function in flow conditioning, but helped prevent foreign debris from entering the pump reservoir, and offered the possibility of symmetry should the tank be moved to a different location, and the pump direction could be switched should it be preferable. As suggested by Vogel (1994) I created my flow straighteners out of drinking straws. These were cut in pieces of three equal lengths with plate scissors, where the piece with the “bendable” part was discarded. The straws were fixed in place by gluing a plexiglass plate on top of them under pressure. The physical pressure of the plate proved enough to hold them firmly in place for the flows exerted in the tank, and no further gluing was needed.

After difficulties suspending screens evenly downstream, as suggested by Scheiman and Brooks (1981), in the small tank without too big a rim, which would cause turbulence, I instead opted for only using screens around the pump hose entry in the settling reservoir. Initially small pieces of fabric were tied around the hose, as this did not sufficiently reduce the turbulence; sponges were used instead. These worked great for lower pump-outputs (around 32l/min) but were not sufficient for my largest pump at 64l/min. Additionally a sock was threaded around the tube and sponge in the settling reservoir after which the visual flow after the drinking straw flow straightener looked uniform and laminar even for the highest pumping output.

2.3.3 Pumping water

The tunnel was first run with a relatively small pump, pumping 7.5l/min. The flow at this point was nice and laminar, but later proved to be much to slow and too laminar for many of the applications I wanted to test in the tunnel. Over time two additional pumps were added, both outputting approximately 12l/min, at which point at which point the more rational solution was to upgrade to a single 64l/min pump and rather control the flow with a variable power supply should the output be too high. The theoretical *Reynolds number* in the tunnel can be calculated with the general formulas for flows in pipes, Equation 2.3, where ρ is the fluid density(kg/m^3), ν is the kinematic viscosity(m^2/s), μ is the dynamic viscosity(pa*s) and v is the velocity of the fluid(m/s), Q is the volumetric velocity (m^3/s) and D$_{\mathrm{H}}$ is the characteristic length which is given for a square tube with Equation 2.4, where A is the cross-sectional area and P is the wetted perimeter of said area

$$\mathrm{Re} = \frac{\rho v D_{\mathrm{H}}}{\mu} = \frac{v D_{\mathrm{H}}}{\nu} = \frac{Q D_{\mathrm{H}}}{\nu A}$$

Equation 2.3 - Formulas for Reynolds number

(3)

$$D_{\mathrm{H}} = \frac{4A}{P}$$

Equation 2.4 - Characteristic length of square tube

(4)

Calculations for the biggest pump show that the tunnel theoretically is well within the turbulent region, with a Reynolds number of almost 12000. With flow conditioning however, the flow is not noticeably turbulent even at maximum output, showing the effectiveness of such devices. With the possibility of running the pump at lower output, the tunnel can simulate a wide range of flows.

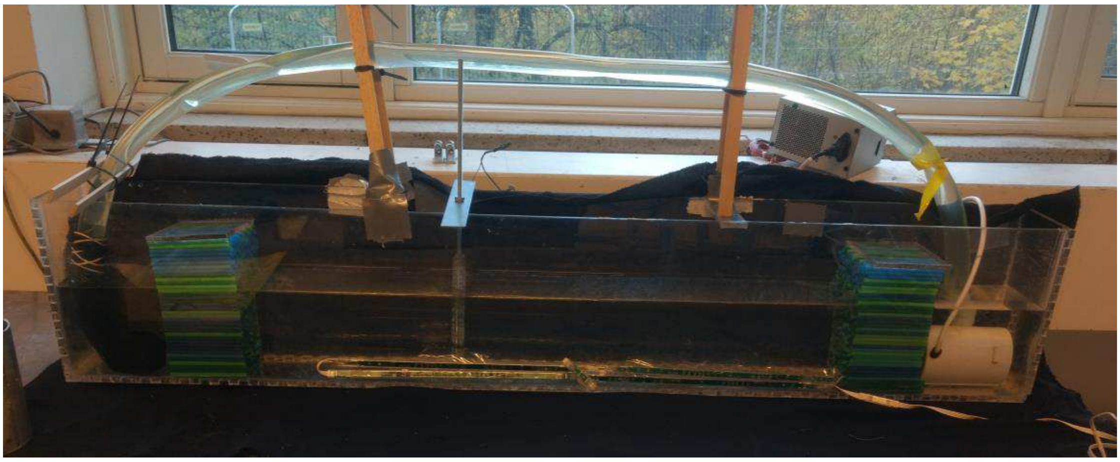


Figure 2.2 - Water tunnel in its current form

The pump required 5A of 12 volts to run at maximum capacity. All desktop power supplies in the lab were fused at 3A, meaning that to run the pump I would need two power supplies in parallel at all times. This became rather impractical, and additionally I would need power supplies for any extra equipment as well. I used an old computer power supply to create a regulated power supply that would be able to handle 6A by connecting four LM317 voltage regulators in parallel. I added a switch so that the ground could be switched from the power supply's GND wire to the power supply's -12v wire, making it possible to get regulated voltage all the way up to 24v. A total schematic of the connection diagram can be seen in Figure 2.3. The LM317 voltage regulators regulate voltage down, depending on the input voltage on the signal(common) wire, by effectively converting the excess energy into heat. This is not very efficient, so for running the pumps long periods of time, I used the non-regulated outputs that were fixed to the computer power supply's output-cables (Black=Ground, Yellow=12v, Red=5v, Orange=3.3v) to avoid overheating issues.

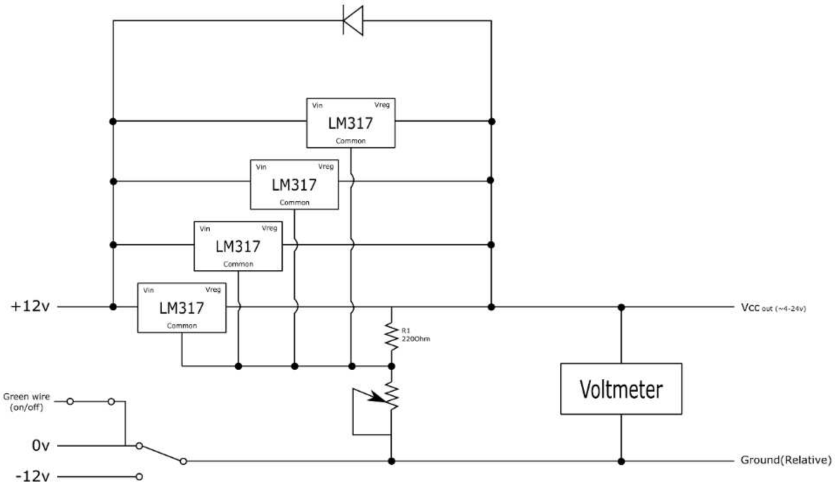


Figure 2.3 - Connection diagram regulated power supply

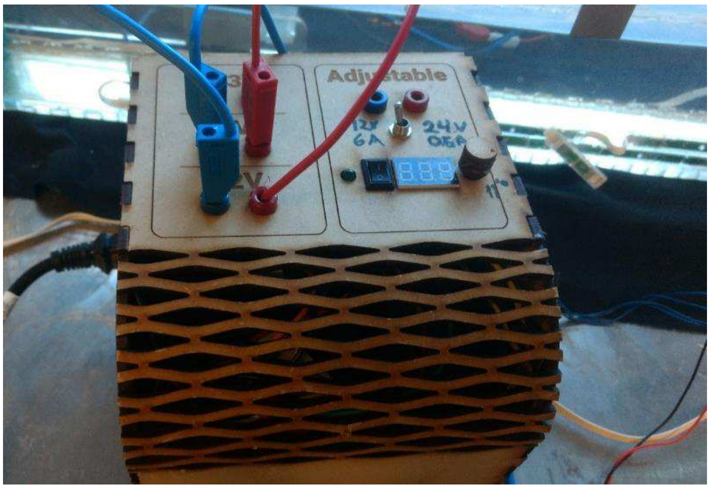


Figure 2.4 - Power supply finished case

2.4 Deciding on an indication medium through prototypes

To obtain a visual of the occurring flows in the tank and around the objects suspended in the flow, some sort of indication medium was necessary. To quickly find out what worked and what didn’t, prototypes were made for the three most prominent solutions; Injecting dyes, Laser sheets with particles and hydrogen bubble generation.

2.4.1 Dyes

Food dye was injected in the stream with a narrow syringe, secured to one of the crossmembers of the tank. Undiluted food dye would prove to have a higher density than water and got a velocity gradient towards the ground. This was improved upon by diluting the dye with water, after which it gave a nice visual stream, however without good lighting it was difficult to see and even more difficult to photograph without proper equipment.

An alternative to regular food dyes is the use of dyes with a fluorescent nature and UV-light to illuminate them. I tested tonic water and the contents of glowsticks, both of which were fluorescent on their own, but when mixed into the stream the illuminating effect was not significant enough to create a good visual of the flow.

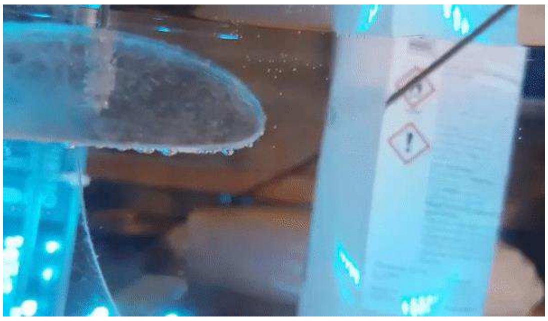


Figure 2.5 - Barely visible dye injected with syringe

2.4.2 Particles + laser sheets

A widely used method for visualizing flows in water tunnel is the use of near neutral buoyancy particles that are evenly distributed in the medium. A laser of considerable strength is then split into a sheet, which illuminates a two-dimensional plane of the flow. Together with cameras an image processing on a computer with *Particle Image Velocimetry*(PIV) an accurate vector field of the flow can be obtained. However, this requires both an extremely strong light source, and good camera equipment to function well (High resolution). I was able to obtain glass spheres, both with silver coating and without from *SINTEF Ocean* and their water tunnel facility. For a quick laser sheet, a laser level was used. While the resulting flow visualization was not bad; in fact, it some cases it gave far better visuals of the flow than the other tested methods, it only worked in extreme darkness, and getting good pictures of the experiment, for illustration purposes as well as documentation was near impossible, as Figure 2.6 might illustrate.

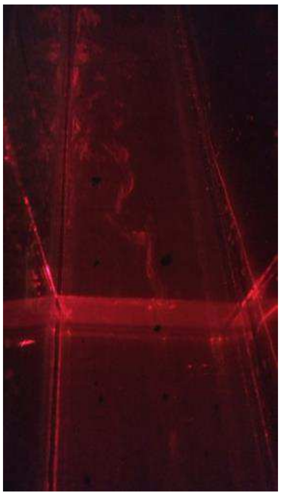


Figure 2.6 - Stream of glass particles in laser sheet

2.4.3 Hydrogen bubble generation

The final method I tested for visualizing was the use of electrolysis of the water. Electrolysis on water generates hydrogen gas from hydrogen ions in the water on the cathode, while generating oxygen and hydrogen ions from water at the anode. The half-cell reactions are:

$$2H+(aq) + 2e- \rightarrow H2(g) \quad E=0v$$

Equation 2.5- Half-cell reaction: Hydrogen reduction

$$2H_2O(l) \rightarrow O_2(g) + 4\,H^+(aq) + 4e^- \quad E=1.23v$$

Equation 2.6 - Half-cell reaction: Water oxidation

Making the total cell potential 1.23v. Meaning that submerging a cathode and an anode in the water with a voltage between them above 1.23v will generate hydrogen bubbles at the cathode, if the water is sufficiently conductive. I used *sodium chloride* as an electrolyte due to accessibility. The trick it seemed was to use a thin enough wire, so that the bubbles generated were not too influenced by their buoyancy, the first electrolysis test and the buoyancy of the bubbles can be seen in appendix **A-2**. Smits (2012) gives a good introduction to this technique, and states that wire dimensions between 10-50 µm create acceptable bubbles with low buoyancy. Acquiring thin enough wire was not straight forward without having to specifically order it. I therefor tried different types of wires that I could easily acquire; 10µm coil wire (insulation removed with heat), separated aluminium core wires from 28AWG cables and separated copper core wires from 26AWG cables, the two latter were approximately measured to be 100µm.

Out of all the wires that were tested, the aluminium wire gave by far the best visuals; generating a smooth and even plane of bubbles. The relatively short life span of the bubbles, resulted in the buoyancy playing only a small significance in the visual of the flow. The method of using hydrogen bubbles offered both continuous visualization of the flow, and did not, over time, contaminate the water in the tank as dye injection would. The only downfall being the need for salt in the solution which made it less ideal to put other electronics into the water. It also made it difficult to suspend objects of neutral buoyancy in the water, as the gravity of the water caged a lot. The anode and cathode were fused at low amps (0.5) and operated on 12 volts, posing a relatively small risk.

The only issue I found with using the hydrogen bubble generation in my flow tunnel, was the need for ions in the water. I used a relatively high concentration of NaCl/table salt in my water as it reduced the need for near contact between the cathode and anode, which in turn would cause rapid corrosion of metals left in the tank for longer periods. Even small metal fragments, such as small chips from steel pipes, could turn the water in the tank orange overnight. This lead to the need for replacing the water of the tank quite often. Thankfully the volume was only about 15 litres. A video of the early bubble sheet in Figure 2.7 can be found in appendix **A-3**.

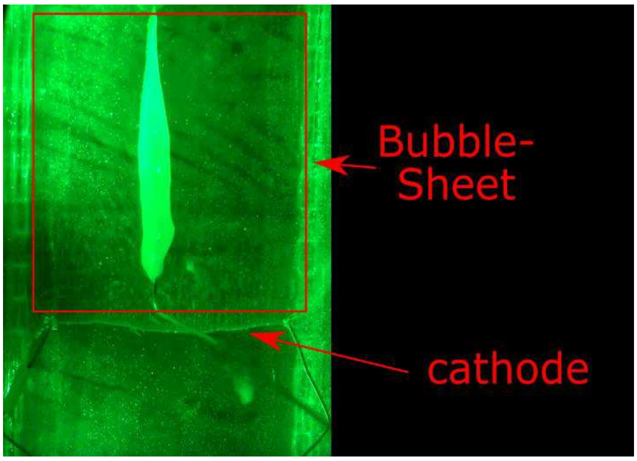


Figure 2.7 - Hydrogen bubbles generating a visual sheet

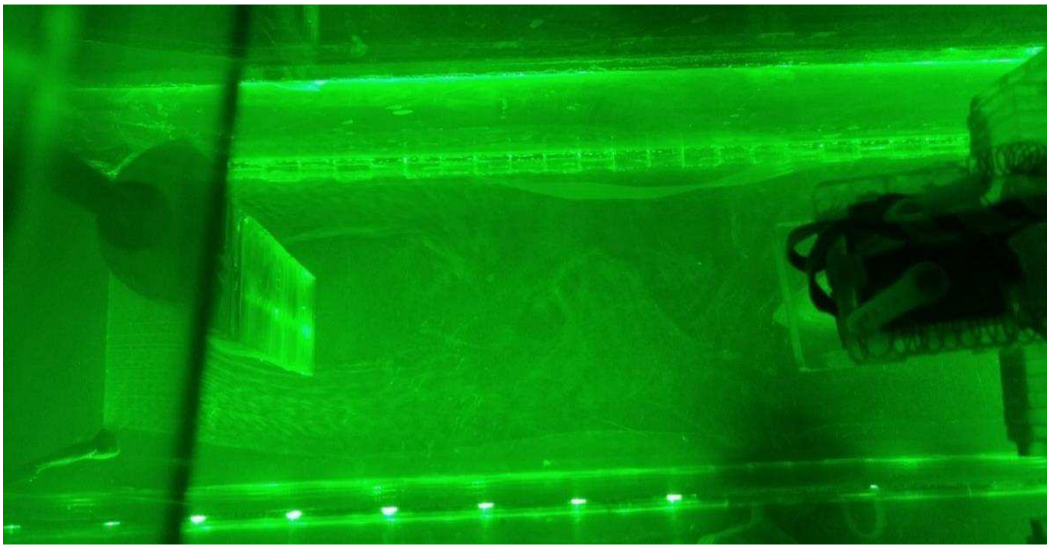


Figure 2.8 –Hydrogen bubble sheet showing a vortex street

2.5 Force Measuring

In Addition to visually be able to see what was going on in the water I also wanted to have some sort of force feedback from what was going on with the test specimen suspended in the water. As I had no idea at this point exactly how these test samples would end up being, I made a very general design for a rig that would hold the samples from atop and measure the forces in a backwards direction with a load cell. The fastening points consisted of holes of various diameters, so that different wires and rods could be used. 2mm copper coated steel welding rod turned out to be very suitable for this.

A 1kg *bending beam load cell* was used along with a *HX711 amplifier* to decode E+,E-,S+ and S- signals from the strain gauges into digitally readable signals.

The output, when uncalibrated, was a signal that followed the applied weights linearly with an offset. For my first experiments I used this data directly without converting to real values, as I questioned the accuracy of the measurements and only wanted qualitative measurements.

2.5.1 Exponential smoothing and data handling

At a later stage some improvements were made to the force measurement rig. Mainly a proper calibration of the device, as well as simple smoothing of the data. The HX711 chip already does a simple noise filtering with an onboard *programmable gain amplifier*(PGA) chip, however, for the small measurement values in my water tank this was not enough to give good visual read-outs. A lot of filters can be used to get the desired effect on the data such as PID/PD filters, Kalman/Gaussian filters etc. I opted to test the, perhaps, simplest noise reduction method fist: *Exponential smoothing*. In exponential smoothing you simply weigh your new signal together with your old results with some pre-decided constants to get the new result. This can be expressed mathematically as:

$$x_0 = x_1$$
$$x_1 = x_0 * \alpha + signal * (1 - \alpha)$$

Equation 2.7 - Exponential smoothing

Where x_1 is the results and x_0 is a memory allocating variable for the previous result. Alpha can be chosen between one and zero, where the higher it is the smoother your result. A smooth result will be at cost of sensitivity, but you will not get the same latency issues as you would if you were to do it through measuring the average of a said amount of measurements to get your result. I chose alpha to be 0.8 which gave pretty good read out, while still being able to pick up small impacts on the load cell. The easiest way to decide on an alpha is simply to test.

For coding and calibrating the load cell I used the *Sparkfun HX711 library* and example codes *(HX711-Load-Cell-Amplifier: A Breakout Board for the HX711 Load Cell Weight Measuring Amplifier* [2014] 2017), with an added line for my exponential smoothing. The calibration process was done through adding some known weights on the load cell and then adjusting the calibration variable in the serial until it read out the correct weight. I used some random items of low weights(<50g) and measured them on a scale. I'll note that the error of the scale I used to measure my known weights was by extension transferred to my load cell in this process.

3 Divergent testing and probing of potentially mimicable solutions and their results

When a good testing environment had been made, this made it easy to rapidly test multiple models and how suited they were for mimicking. That is to say; which I thought would yield the highest potential gain as compared to design input and solutions already existing. The first few weeks after the creation of the flow chamber I tested a lot of different models, some were good, and some were scraped even before they made it to the water tunnel. The process of creating these prototypes were often quick and error prone, and mainly focused on the learning outcomes. Much like in set-based prototyping, the experiments were designed to luminate whether there was any merit to the single tested mechanism and not the total system in which they might be included. This way a large solution space that I had low prior knowledge of could be investigated and explored as much as possible. The information gathered and generated in these early prototypes was saved and used later both to decide the main direction of the project, but also to make informed assumptions. Note that although most of the divergent testing was done at an early stage in the development process, some of the test ware done at later times (such as the surface drag tests in 3.2).

3.1 Piezoelectric Micro-swimming

Fish and aquatic animals swim in many different patterns and modes. Eels and eel-like organisms swim in what's called the *Anguilliform* swimming mode(Sfakiotakis et al., 1999). The mechanics of which might be of the more comprehensible than regular swimming; the movement of the body is visually not much different than a screw in two dimensions or a sinus wave. Anguilliform swimming mode relies less on the flexibility of the body, and more on actuation input. The body is dependent on at least five dimensions of freedom for the swimming mode to occur. The high amount of body control, and low dependency on non-rigid properties has made it attractive for snake and eel like robots. I wanted to test, if this swimming mode could be reproduced on a micro level in a rigid body by inducing vibrations in the body.

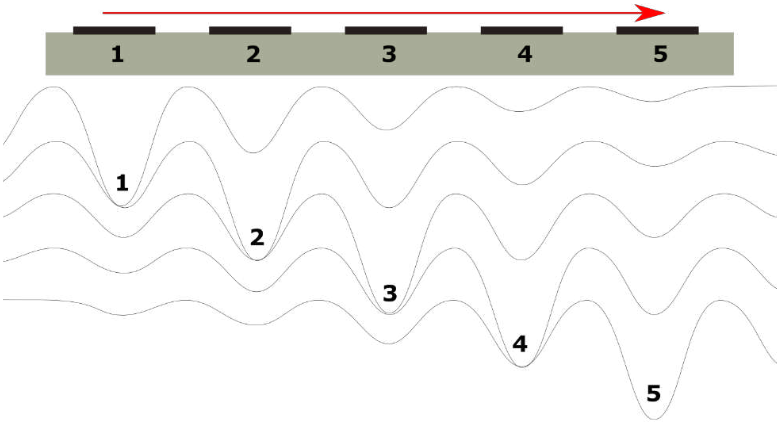


Figure 3.1 - The generated wave top (1, 2,.., 5) moves backwards.

I used five piezo electric elements to test the hypothesis. I fist mounted these inside a soda bottle cut in half, but I was not able to get a good surface contact between the elements and the bottle due to its curvature. I then opted to laser cut a small board of 6mm Plexiglas. The piezo elements were mounted using cyanoacrylate superglue.

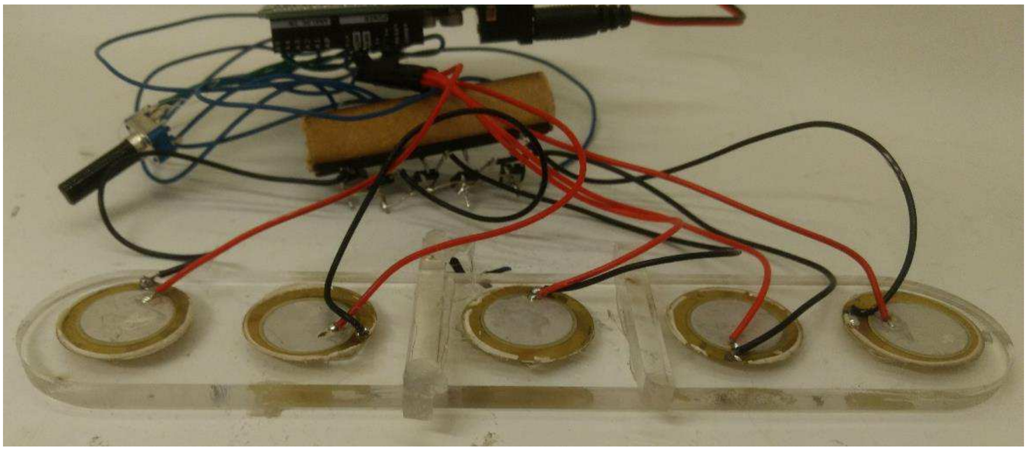


Figure 3.2 - Assembled array of piezo elements and the controlling circuit.

Piezo elements flex when a voltage differential is applied over them. By applying a voltage to the five elements in sequence with a frequency that matches the eigen frequency of the body, a wave from should occur, with an accumulated wave top traveling backwards. The piezo elements were controlled using five NPN transistors and an Arduino with a potentiometer. The Arduino and potentiometer controlled the frequency in which the piezo elements were given power. A micros(); function was used between each step, and the value of the function was determined by the potentiometer ranging from 0-1023.

3.1.1 Testing and results

To give the test rig positive buoyancy, I used duct tape along the outer edges. The Arduino and power source were suspended from above to reduce the mass of the craft. I then used the potentiometer to slowly go through the frequency range of the program (5mHz-5kHz), running the piezo elements at 10v with an external power supply. I tried to pay attention to any changes around the raft such as movement or waves forming, but visually nothing happened. I then increased the voltage to 18v, I tested a separate element and found that they would not go above 19v, and redid the steps above. This too had no visual effects on the raft or the surround water.

Both experiments however gave of an annoying sound even when it should have been outside of the human hearing range. It is possible that the Arduino was not able to run the program at the correct speeds. It is also possible that I never hit the eigenfrequency or an n-multiplication of the eigenvalues (n=1, 2, 3, …). The raft was also relatively thick as compared to the size of the piezo elements which might have made it diffuse most of the energy into the raft rather than transferring it into the water, piezo electric elements that move water are frequently used for pumps and other water applications, so I was surprised by the dullness of the experiment. The results of this experiments with all the potential errors considered might be questionable; I have yet to determine if it is an incredibly good or incredibly bad idea. The test did however show that the initial idea was not as easy to achieve as fist though and to make it work would probably be a resource draining activity.

3.2 Turbulent surface for reduced drag

“Gray’s paradox” as discussed earlier lead to a lot of research into the hydrodynamic properties of the dolphins and drag reduction in general. Although Fish (2006) argues that the reduced drag of dolphins mostly stems from their sleek shape that delays separation, one of the effects that has been thought to give a drag reducing advantage is the creation of turbulent layers along the animal body. This is especially the case for sharks which have a rough skin with teethlike structure. This effect got popular when the full body swimsuits were banned after the 2008 Beijing Olympics, and *technology doping* became a public discussion. The *FastSkin II* swim suits made by *Speedo*® supposedly use the turbulent shark effect to reduce drag, but as shown by Lauder and Oeffner (2012)the effect simply won’t work when applied on humans as we are too stiff, and the suits gave little or no effects in drag reduction related to the surface layer even when applied to a soft body. They did however find a 12.3% increase in swimming speeds of their foils when shark skin plates were applied as compared to not.

Although of high interest, few have actually produced something that uses the effect (Wen, Weaver, & Lauder, 2014). In an attempt to further learn from Lauder and Oeffners(2012) experiment, and to see if the same effects could be found with plate-like fish skins as well, I redid a simpler version of the experiment with rigid and non-rigid foils suspended in my water tunnel. The aim was to measure if there was a significant difference in drag between foils with fish-skin and plain foils.

I used a laser cutter to make foils in stacks of 6x6mm acrylics. A total of two of these stacks were made to form the rigid foils. The cut-out material was used as a silicone form to cast the non-rigid foils. So that the foils would be of near identical shape. I cast the non-rigid foils out of *Ecoflex® 00-30 liquid rubber* which cast nicely and doesn’t stick to the mould, the resulting two foils were extremely soft with a slightly stiffer than fish-like consistency. The foils were fitted with 2mm welding rods to fasten them to the force-measurement rig.

One rigid and one non-rigid foil were then applied salmon skin. The skin was cut from pieces of frozen fish where the skin structure was well preserved. Spray glue was used to secure the skin to the rigid body, while I had to resort to cyanoacrylate superglue for the rubber foil because of its non-sticking properties. I paid close attention to getting a smooth surface finish, but some overlaps and patches had to be made to fully cover the foils. These were made so that the front skin overlapped the back skin, as to reduce potential drag.

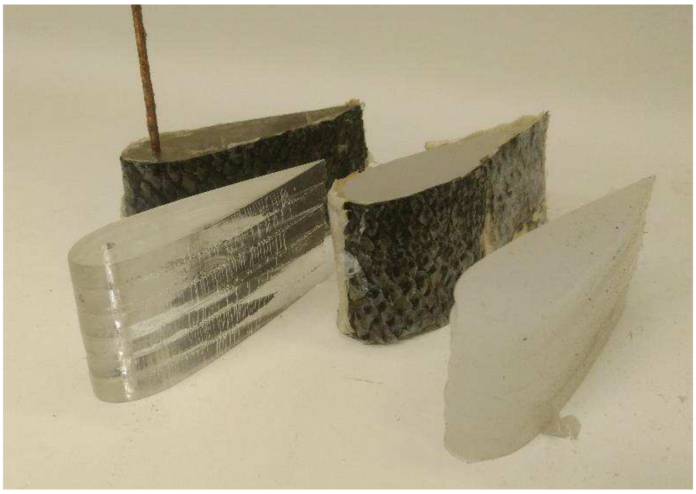


Figure 3.3 - The four different surface effect samples.

3.2.1 Testing and result

Each test piece was placed in the tank suspended by the force measuring test rig approximately one cm under the surface. The pump was stopped and once the water in the tank had settled the force gauge was zeroed in the serial plotter. The load cell was configured to do 10 samples per second. The pump was started, which can be seen in the plots as the oscillation in the force measurements. Once the forces were stabilized the drag forces working against the foil can be seen. The surface area of each test piece should be close to identical, and a calculation of the actual drag coefficients was not needed to compare the pieces.

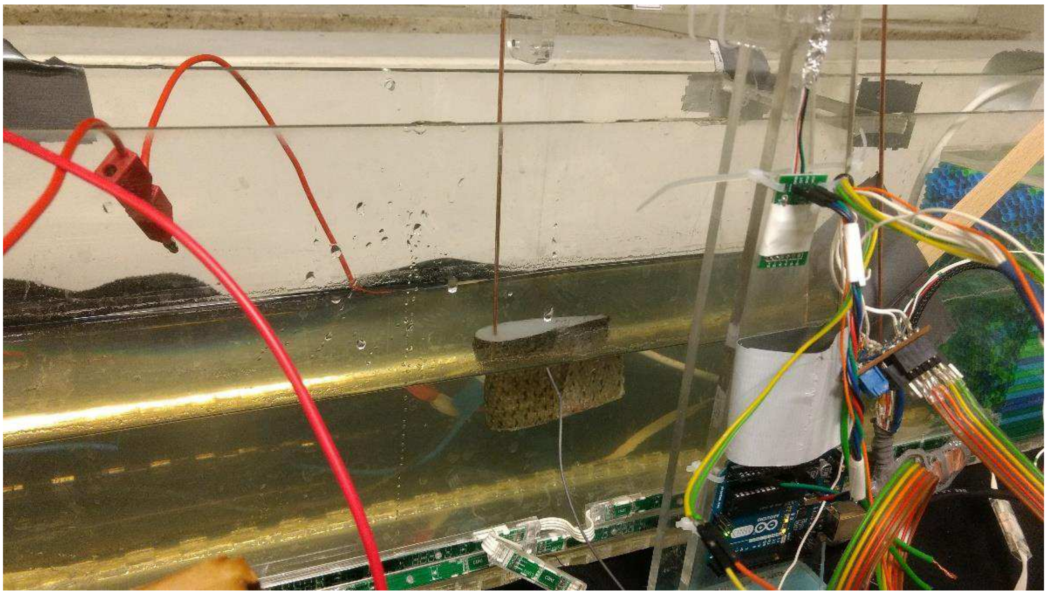


Figure 3.4 - Soft sample covered in fish skin rigged for force testing

Surprisingly in the first test there were some were some slight differences between the test pieces. The solid non-covered test piece preformed the worst. While the three other pieces preformed slightly better. The results however were not consistent. And redoing the experiment, setting everything up again, a second time gave the complete opposite results, where the solid non-covered foil performed the best.

The setup seems to be more sensitive to factors This was mainly due to the inability to get good readings. The force measuring rig did not seem sensitive enough to give good quantifiable data. The turbulence induced when turning on and of the pump could be clearly picked up in the readings, but the drag forces of the test pieces once the flow was steady were low and near zero.

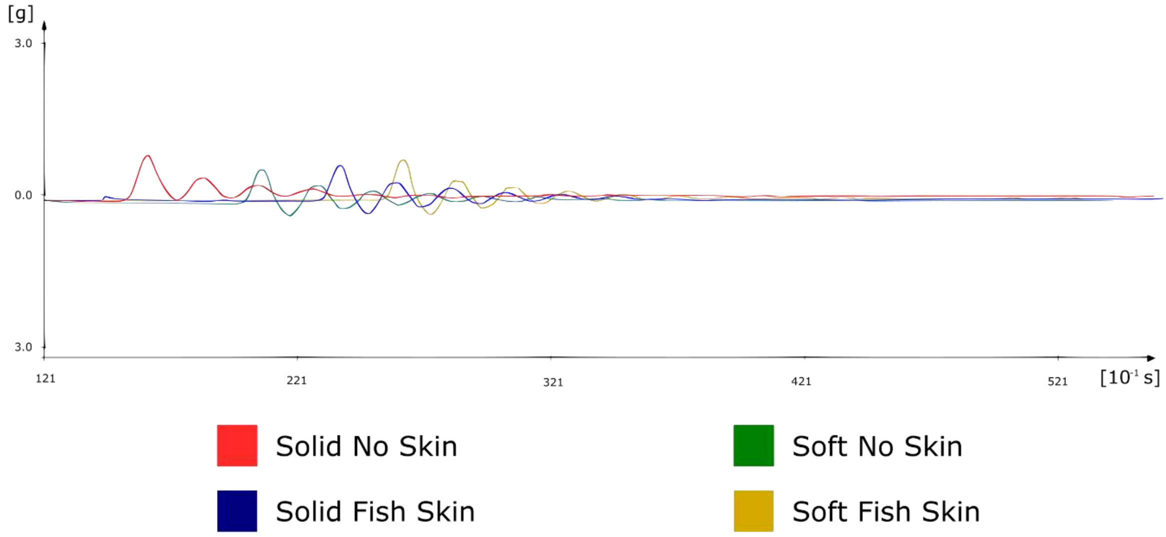


Figure 3.5 – First test: Drag forces acting on foils in flow. Measured in grams aka. [9,81/1000 N]

3.3 Sensory flow input from body surface

Fish are often thought of in their entirety as a sensory organ. Simplified, the way fish feel and detect water flow is through hair cells (Flock & Wersäll, 1962). All along the fish body we find hair cells that are imbedded in cupula. The highest concentration of hair cells is along the lateral line, and the head of the fish. The hair cells, like the ones found in human ears, detect motion. This motion can then be used to deduce pressure, flow, vibrations and movement. These relatively simple receptors can in great numbers create a good picture of the hydrodynamic situation around the fish, which is often more complicated than one would have thought such as in “Grays paradox”. The sensors we use for flow sensing today are often bulkier and require more space and installation time, thus making a grid of thousands of sensors impractical. A way to measure the flow around the body is by using pressure sensors. This is commonly used in the aeronautical industry, through pitot-pipes and atmospheric pressure sensors to find the relative speed of the aircraft. In the paper Flow-relative control of an underwater robot (Salumäe & Kruusmaa, 2013) this method is used to classify the different flow regions in the wake of an object, as well as flow speed and fish orientation, and it is shown that even with relatively few pressure sensors a good estimate of these conditions can be made.

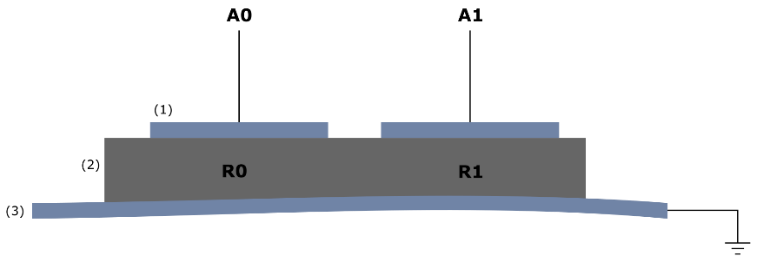


Figure 3.6 - Foam of relative resistance with leads on both sides. Reduction in distance in the foam (2) will reduce the resistance in R0 and R1, different forces in different points will skew and push the outer skin (3) differently and cause different resistances.

I wanted to investigate the possibility of a sensor that more truly resembled the way that hair cells function. The idea was to create a sandwich with a conductive layer on the outside of the skin, and small conductive data points on the inside. Between the two I would need a layer that changed resistance depending on compression. I have preciously used anti-static foam to make force-sensors with some luck and wanted to try this as the hair cell layer to see if skews and compressions could be picked up by measuring the resistance directly. The goal was to create sensors simple and cheap enough that a large grid of data points could be deployed, and machine learning used to get a good understanding of flow scenarios, so if additional amplifiers were needed to read the results, it would not meet these criteria.

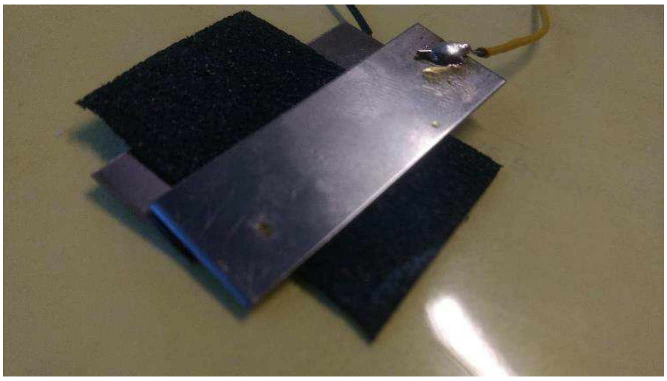


Figure 3.7 - Plates on either side of anti-static foam

3.3.1 Testing and results

To test the feasibility of this concept I cut some antistatic foam to approximately 5mm in thickness. I soldered some wires to two steel plates. I then attached a multimeter to each steel plate and measured the resistance between them when the foam was their only connection. With no forces applied I got a read out of values in the 25KOhm range. The lowest resistance I was able to achieve was in the 20kOhm range with full compression of the foam. Although this sounds measurable, the force needed to compress the foam was extremely high as compared to what will be experienced in the water. It should also be noted that for small forces the readout made no behavioural pattern, and environmental variables and noise, such as hands touching the plates and positioning of the plates, made a much larger impact on the readout than the compression itself.

3.4 Actuation of a swimming body

Swimming as compared to other propulsion methods in water is far more energy efficient. Swimming in fish is typically divided into two different types of swimming: Body and/or caudal fin (BCF) locomotion and median and/or paired fin (MPF) locomotion (Sfakiotakis et al., 1999). BCF is the types of full body swimming that is typically associated with fish-locomotion. BCF type of swimming typically moves by creating a waveform which is pushed backwards, generating a trailing vortex street. Of the four under categories of BCF the most advanced form is the Anguilliform, which as has already been discussed relies on high degrees of freedom, while the simplest is the thunniform swimming mode which only needs one degree of freedom to propel itself, but needs a soft caudal fin to generate the propelling wave form. Thunniform swimming is regarded as the most energy efficient locomotion mode that has evolved in water(Sfakiotakis et al., 1999). Being able to reproduce swimming locomotion through mechanical and mechatronic tools could help reduce energy consumption in water propulsion.

3.4.1 Magnetic

An idea that came from my co-supervisor Achim was to use electromagnets along a piece of membrane with magnets in it. By changing the direction of the magnetic field in the electromagnets, the membrane could be moved back and forth and create an *Anguilliform* swimming pattern. To test this, we taped some permanent magnets to a piece of paper and moved other permanent magnets along the paper. Getting smooth movement was difficult, and the paper moved in sudden bursts and was prone to twisting.

3.4.2 Servos

Many of the existing robotic fish used in research today such as the robotic fish used in Marras and Prfiris paper *Fish and robots swimming together: attraction towards the robot demands biomimetic locomotion* (2012), rely on servos for movement. In their paper Marras and Porfiri (2012) show that even with a single servo flipping a fin back and forth they are able to reproduce a thunniform swimming pattern convincing enough to “trick” live fish to swim in formation with the robotic fish. Although using servos for locomotion is not especially exiting, I wanted to investigate its potential to be used in my water tank, where size restrictions are one of the main issues.

To see the impact an internal servo would have on the total size of a fish-like body, I attached a servo in the middle of a piece of Styrofoam. Using a sharp knife, I cut the surrounding foam around the servo to a fish shape as close as possible to the servo, while still giving room for a reasonable chamfer between the outer edge and servo. The resulting fish was significantly larger than anticipated. And much too large to be actuated in my thin water tunnel, without wall-effects playing a large role in the results.

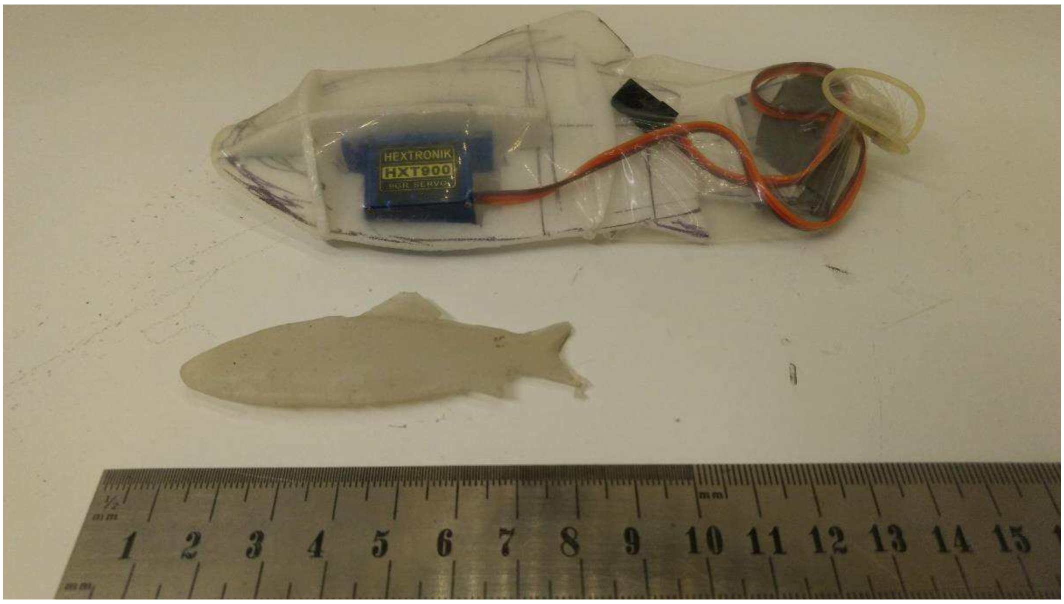


Figure 3.8 - An internal servo makes the robotic fish too large for the water tank.

3.4.3 Shape memory alloys

Shape memory alloys comes in many types and under many different names. The type I acquired for testing is often called "muscle wire" as when they are heated they shorten in length about 5%. The official name of this titanium-nickel alloy is *Nitionol.* One of the practical properties of the muscle wire is that the resistance in the wire is high, meaning that running current directly through the wire will heat it and shrink the wire without the need for a complicated control circuit. The transition temperature of the alloy is decided by small changes in its composition(Mohd Jani, Leary, Subic, & Gibson, 2014), generally they are in close range to body temperature. I choose an alloy with a transition temperature at 70ºC, for a higher temperature difference between the wire and the surrounding air thus an increased cooling speed which in turn would give the wire quicker and more controllable reaction speeds.

I made a fish body to fasten the muscle wire to out of thin 1.5mm sheets of poly carbonate(Lexan) as most poly carbonates can flex well without breaking. To form the poly carbonate I laser cut it, making additional living hinges for softer movement, and room for an Arduino Nano to try for sizing of the fish, I also made curved crossmembers along the body of the fish to give it a 3D shape, and to have something to fasten the muscle wires to. The muscle wire is only able to pull, so to use the wire to actuate back and forth I used a rubber band to pull the body in one direction while the muscle pulled in the other. The resulting body was quite small, and the actuation was remarkably organic. The strength of the muscle wire was also remarkable, and it had no problem pulling against the rubber band. The power consumption to retract the wire at reasonable speed was about 0.3A at 5V. I tested the retraction speed of the wire up to 1.5A, at which point the wire burned and sheared itself. It is possible that dirt on the wire concentrated heat at one point causing the wire to shear. But in general, it seems like applying modest amounts of power is the safest way to operate the SMA wires. In appendix **A-19** a video of me testing some SMA wires attached to a polycarbonate body can be seen.

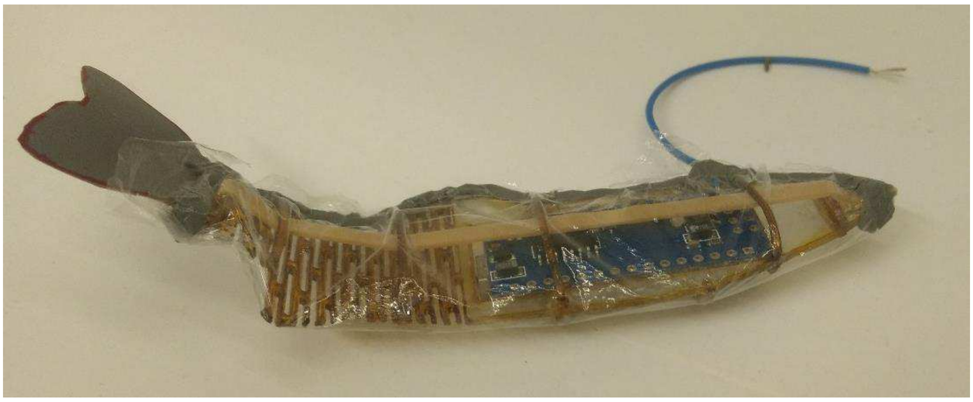


Figure 3.9 - SMA and rubber band actuated fish body

3.4.4 Actuation results and discussion

While magnetic swimming actuation in itself would be interesting, getting it to work properly would require a lot of work and as a standalone project it did not feel that it fit my objectives well. The servo required a lot of internal space, and for my small tank servo actuation would have to be from the outside. The muscle wires preformed much better than anticipated. Not only are they strong and fast, but they also move in a very organic way as they do not jump to positions, but rather shrink and stretch in accordance with the increasing and decreasing temperatures. They also take up very little space, and the only restriction would be fastening them and running lead wires to the ends. The biggest drawback to using muscle wires for actuation would be their low energy efficiency, as the rely on turning electric energy into heat. In a tank setting it would not be an issue, but for a free-swimming robot, potentially battery powered, it would be a problem.

3.5 Karman Gait

When fish swim upstream in rivers they sometimes stay stationary behind rocks or other objects in the stream, where an alternating vortex street is generated due to the right flow conditions. This effect is called *Kármán gaiting*. The effect does not seem to be fully understood, but the vortices generated by the object in the stream seem to actuate the fish so that with low muscular input a positive force forward can still be maintained(Liao, Beal, Lauder, & Triantafyllou, 2003a). Characteristically the total flow in the vortex street behind objects is still going downstream thus the fish it is not simply resting in a low flow or pressure zone, but is actively using the currents to stay stationary in the stream. The vortices in the street are similar in appearance to those that are shredded behind swimming fish, but have an opposite rotational orientation. Most interesting; the Kármán gaiting effect can also happen passively; it has been recreated with recently deceased *rainbow trout* by James Liao (LiaoLab, 2012).

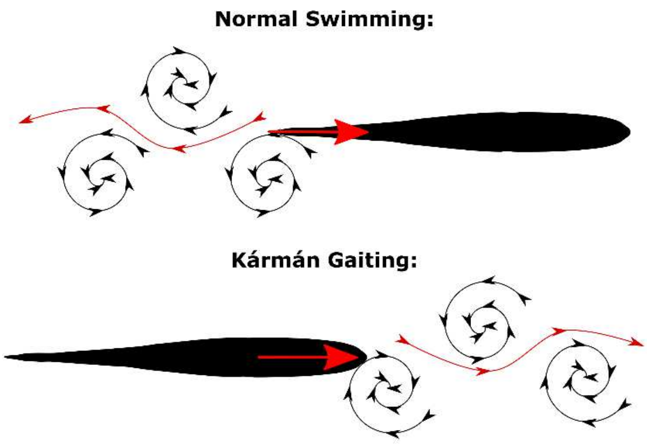


Figure 3.10 - Kármán street as compared to vortex shedding from swimming. Note the direction of the spiralling movement in the internal street.

To test if I could possibly recreate the Kármán gaiting passively I created soft rubber-silicone fish in vacuum-formed moulds of 3D printed rainbow trout 70mm in length. I used *Ecoflex® 00-30 liquid rubber* to cast the fish.

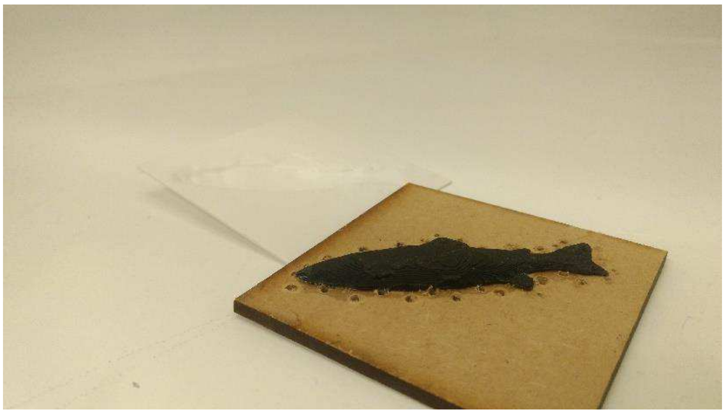


Figure 3.11 - 3D-printed and vacuum formed mould for silicone casting

3.5.1 Testing and results

The rubber fish were secured in the water tunnel with steel wire. Research has shown that the best response for passive Kármán gaiting happens when the fish body is 20% longer than the distance between the vortices in the vortex street (Toming, Chambers, & Kruusmaa, 2014), although for the initial test I put little effort into understanding the system as a whole, and rather used several cylinders of different diameter that I placed in the stream until I could observe movement of the rubber fish.

Two conditions caused swim-like movement of the rubber-fish. The first case was when I placed the largest pipes in the flow. This would cause high turbulence behind the pipe, as the cross-section of the tunnel was greatly reduced around the pipe. This turbulence caused semi-random movement of the body.

Figure 3.12 - Rhythmic movement of fish body in vortex street

The second case was when the fish was placed at some distance from the two smaller cylinders. This caused a more rhythmic movement of the fish body.

The last result seems to fit the model for Kármán Gaiting as described by Liao (Liao, Beal, Lauder, & Triantafyllou, 2003b). The Kármán Vortex street appears around three diameters in length behind the object in the stream, and the vortex shedding happens in a rhythmic fashion back and forth with a frequency that can be calculated from the objects diameter and the flow speed. My results were however not extremely surprising. The movement observed was very subtle, and there was no apparent generation of thrust. A piece of thin membrane would have moved back and forth in the stream as well, but it would be interesting to further explore if it is possible to generate a forward thrust from fishlike bodies in the stream.

4 Convergent testing and refining

The goal of any product development process and biomimicry alike is to find an idea with potential. The idea that most stood out to me during my divergent testing phase, was the Kármán Gaiting, and potential to use flows and vortices for positive gain in the propelled body. My initial tests of the principle had no real output in terms of data or understanding of what was happening to the body. I had simply observed that placing a soft body at the correct distance away from and object would cause it to move in an alternating way which might reassemble swimming. Seeking literature to fully understand Kármán gaiting shows that there lacks a single comprehensive theory of the true mechanics behind it, rather many diverging hypotheses exist that try to explain the effect. The Kármán gait might be a complex system dependent on many different mechanics, as literature might suggest, but to further learn and understand how it functions and what is needed as an input to make it work, prototyping and testing feels like a natural starting point. The goal at this point in my process was to refine and develop the idea of Kármán gaiting. Due to the loose nature of the concept and my still limited knowledge in the field, saying that this prototyping is convergent might be contradictive as new divergent hypothesis might arise and need to be tested, but the goal is still to narrow down through testing and proving/disproving theories.

4.1 Recreating Karman Vortex street in the flow tunnel

For a Kármán vortex street to occur, the local Reynolds number around the object needs to be between approximately 50 and 1000. This is dependent on the shape and size of the object as well as the flow speed around it. My initial experiments simply relied on exchanging the pipes in the stream for one of different diameter, until a vortex street was generated. For the further experiments I wanted a more reliable vortex street. Traditionally in experiments with Kármán streets, *D-shaped cylinders* (Half cylinders) are used as they create a more distinct vortex street. For my future experiments I therefore used a D-shape cylinder of 30mm in diameter. The D-shaped cylinder has a known approximated *Strouhal number*, $S_{t,}$ for vortex shedding of 0.2 (Gopalkrishnan, Triantafyllou, Triantafyllou, & Barrett, 1994). By approximating the flow-speed in the tunnel *U* to be 0.1m/s (cross-section is close to 10x10cm, and pump pumps about 60l/min or 1l/s). The shedding frequency is found to be in the proximity of 0.74Hz and the length between the vortices 14cm.

$$U_c=U(W/W-D)$$

Equation 4.1 - Effective flow speed

$$F_c= (S_t*U_c)/ D$$

Equation 4.2 - Vortex shedding frequency

Ideally as found by Beal (2006), the fish bodies put into the stream should be close to, or slightly longer than 14cm in length for maximum forward thrust in the vortex street. However in the experiments conducted by Liao (2003a) where a 50mm D-shaped cylinder was used, the calculated length between vortices should be around 21cm, yet fish of 17.8cm were able to Kármán Gait in the vortex street. I will therefor use slightly smaller fish models 10-12cm in length to better fit the narrow tank. A video of the vortex shedding around the D-cylinder can be seen in appendix **A-5**.

4.2 Passive Karman gaiting

Since James Liao showed that deceased rainbow trout were able to swim upstream for limited time periods in the right flow conditions (LiaoLab, 2012), this means that by using a soft body of right shape, without actuation, it should in theory be possible to make them Kármán gait, given that they are the right "type" of soft. With a more consistent vortex street, and a better understanding of the effects of Kármán gaiting, I further tested the concept of Karman gaiting passively.

4.2.1 Karman Gaiting with freely suspended soft inanimate object

For the first experiments I redid the experiment in 3.5, again using a metal pipe. The only factor I changed this time was the degree of freedoms the fish had to move. This time I wanted the fish to be unrestricted and free to move, to see how this affected them in the vortex street.

Appendix D-1– Project Thesis

For the first experiment I used a steel rod with a loop on the end, a thin fish-line was tied to the loop and using a needle it was threaded through the rubber fish at an angle so that the fish sat straight in the water. I started the water-pump and let the flow develop before I moved the fish into position. When the fish entered the Kármán street, it started to move back and forth much more distinct than earlier, following the vortex street. The load cell was not correctly calibrated at the time, thus collected data has no reference values, but the sinusoidal shape of the load in the period the fish was in the Karman street can be seen in Figure 4.1

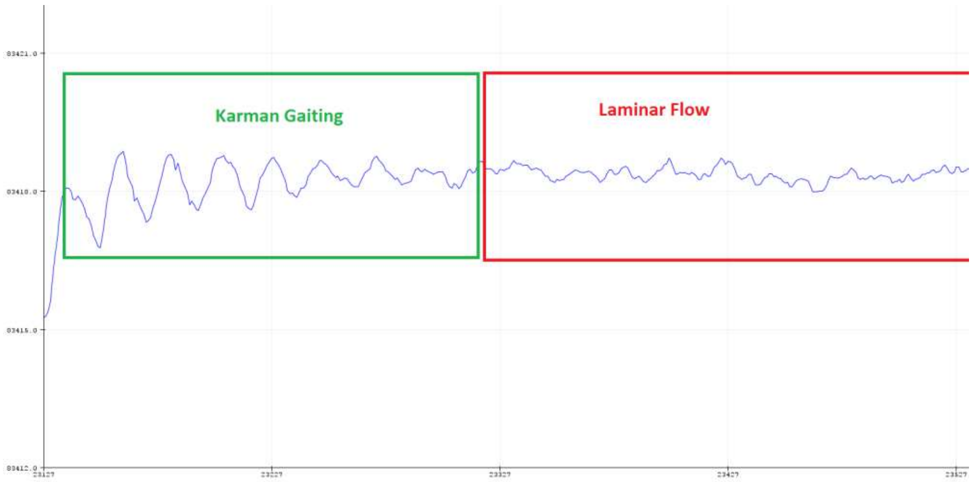


Figure 4.1 - Shape of load curve of freely suspended fish body in Kármán street and laminar flow

The results from the previous experiment did not consistently show a reduction in drag for the suspended fish body, although **Error! Reference source not found.** looks as if the mean load might be of higher value during the laminar flow than the Karman gaiting period, this difference came from drift in the sensor readings, and when the experiment was done in the opposite direction I got the opposite result, but the video of the test show a much bigger movement of the fish, than when it was held in place with steel wires.

Figure 4.2 - Rubber fish suspended by line in Kármán street.

4.2.2 Motorized raft for energy measurement of passive Kármán gaiting

At this point I felt that my choice of output and method of suspension might be to blame for the unclear results. A potential solution to this was to instead of measuring forces, measure energy, and use the water to suspend the fish in the water. So, in the second experiment I used a raft made from drinking straws that were glued tight on the ends. This created a streamline floatation device. A rubber fish was fastened underneath, with a piece of Lead-solder in the belly to keep it down. A small motor was attached to the raft with a propeller on a long shaft, well behind and above the fish so that it would not influence the flow around the fish. For power wires to the motor I used thin 10um insulated coil-wire so that it would not cause significant forces on the raft.

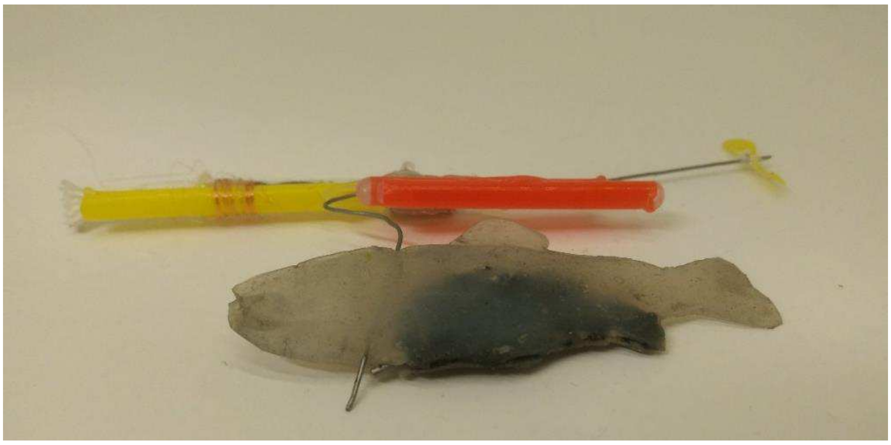


Figure 4.3 - Fish underneath motorized raft

I started the water tunnel without the D-shaped cylinder. The raft was placed in the water, with the motor connected to a power supply. I then regulated the voltage of the power supply, as it was able to make finer adjustments for voltage than amperage, until the raft stood still in the water. I then regulated the voltage down 0.1v and confirmed that this caused the fish to travel backwards.

I then placed the half-cylinder in the flow and let the vortex street establish itself. I moved the fish into the Kármán street and let it loose.

The results were quite surprising. Not only did the fish stay stationary in the lateral direction, but for the short period of time it stayed in the Kármán street it had seemingly self-correcting properties where the vortices changed the fishes' direction back and forth to keep it in the stream. I was not able to balance the fish in the vortex street for long periods of time, but it might be that the correct tweaking of power and dimensions could make it stay stationary in the flow with reduced energy usage over time. The resolution of this experiment was not very high, but the data show an estimated 17% energy reduction by Kármán gaiting as compared to staying stationary in free flow. Videos of this experiment can be found in appendix **A-9**, **A-10**, **A-11** and **A-12**.

	Amp	Voltage	Power
Laminar flow	0.02	0.60	0.012W[1]
Kármán Street	0.02	0.50	0.010W

Table 4.1 - Energy results to achieve equilibrium when suspending a passive silicone fish in different flows

Figure 4.4 - Rafted rubber-silicone fish in Kármán street

[1] The accuracy and significant digits of the power supply is not known, and this result might not hold valid. I assume two significant digits accuracy to differentiate the results.

4.2.3 Karman Gaiting with mechanical semi-rigid objects

Having experienced that Kármán gaiting-like behaviour was achievable with passive bodies, I wanted to attempt to classify the critical attributes for the Kármán gaiting. I therefore made rigid fish models, with mechanical one-dimensional joints along the bodies. One model was sanded and filed to become fully fish-shaped. While another was kept as a foil. The two bodies were then suspended in the Kármán Street with fixed rods to the force measuring rig, but showed no movement or positive thrust, even if the joints were extremely loose and willing to move. I theorized that with no spring-back in their bodies they had no ability to oscillate and generate larger movement, the bodies might need a correct eigen value for proper movement to happen. A muscle activated fish would be able to change its oscillating response in accordance with its situation by tightening and relaxing muscles. This lead to the desire to test a more active rig, to test the critical mechanical attributes of oscillation for Kármán gaiting.

4.3 Active Karman Gaiting

The passive Karman gaiting with the mechanically jointed fish yielded no results. In his paper *The Kármán Gait: novel body kinematics of rainbow trout swimming in a vortex street* James Liao(2003b) hypothesizes in the end, that one of the effects that enables Kármán gaiting is the trout actively changing its *camber* and *angle of attack*(AoA), thus establishing a differential pressure gradient and generating a positive force upstream.

Not considering the AoA for the time being, the camber of the foil can be changed slightly by moving the leading edge of the fish/wing back and forth.

In previous experiments I have tried to actuate swimming modes with different actuators. I found that the most promising was using SMA to actuate the body, yet the simplest solution is still to control servos. The biggest obstacle with the servos is their large volumetric impact and mechanical movement.

4.3.1 Mechanical actuation with springs

As servos yield the simplest way to actuate the body, I tried them first. To overcome the drawback of the servos I worked around this. I built a rig where the servos were attached outside of the body and the water. Additionally, the servos were not rigidly attached to the body, but attached through springs so that softer actuation could be achieved. A theory that I had tried to test with the passive mechanical models by adding rubber to the models was the oscillating effects of the eigenfrequencies of the body. A muscle activated body can change its stiffness, and thus the eigenfrequency. It is possible that getting the right stiffness is key to passive Kármán gaiting. With springs on either side of the body, one would think that stretching/relaxing the springs would change the stiffness of the body. However, as the forces in springs are normally linear, the resulting eigenvalue would be constant. Rubber bands are not completely linear and could be used instead for non-linear stiffness regulation, but as the servos have an arm that changes its position from the central axis of the body it should be possible to non-linearly change the eigenfrequency of the body (servos at 90 degrees normal to the central axis would cause the highest stiffness) also with linear springs.

To test this, I made soft springs from 0.1mm piano-wire. Servos were attached above a mechanical foil with two degrees of freedom and a soft tail. The entire rig was then freely suspended in plane parallel to the water surface, so that it was free to move forward and backwards and left and right, but locked in all rotational axis as well as up and down. This way I hoped to be able to observe behaviours like that of my previous experiment with the floating rig.

To control the servos, I made a rig of four linear potentiometers, each controlling the position of one servo.

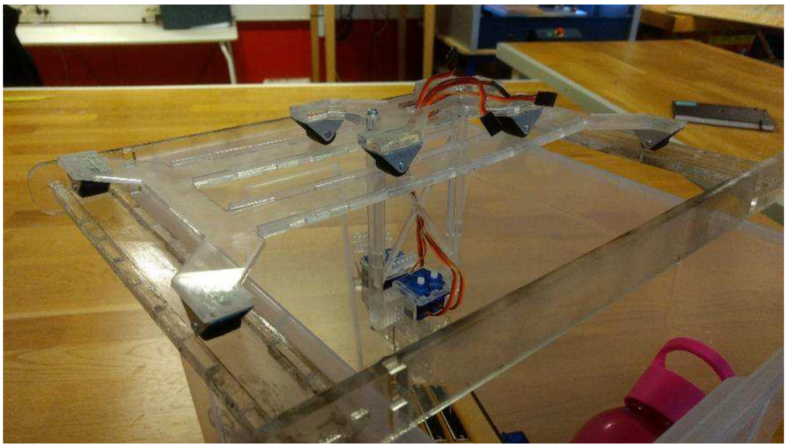


Figure 4.5 - Rig to increase dimensions of freedom in suspension of foil

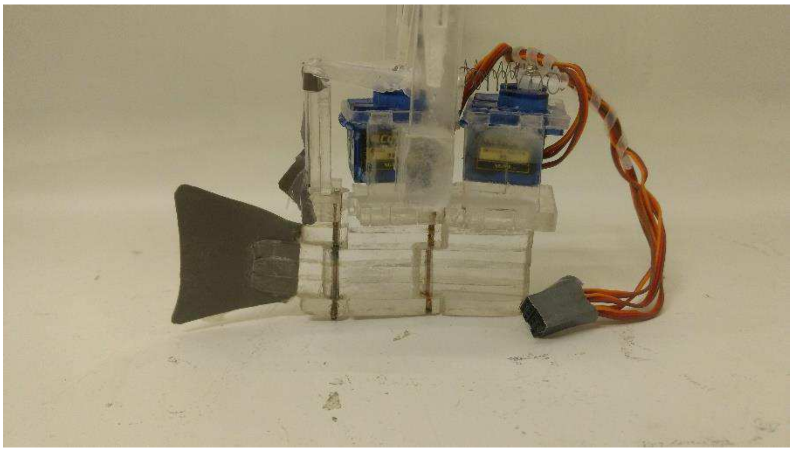


Figure 4.6 - Mechanical foil with actuation on top

I first tested the oscillation by suspending the rig in the Kármán street and slowly and repeatedly changing the servo-arm position on both sides of the fish body in parallel. No position resulted in movement in the mechanical joints of the body. The tail somewhat moved to the oscillation of the vortex street, but this showed no correlation to the stiffness of the rest of the body.

Secondly, I tested manually changing the position of the front servos in opposite directions, so that the leading edge of the body would flip back and forth. The resulting camber and angle of attack of the body should then change back and forth, potentially being able to generate lift and move the foil along its free axis. Doing this smoothly was difficult, as can be seen in the video in appendix **A-13,** and after trying for some time I decided to code the servo position with sinus waves instead, and rather change the frequency and the phase shift with the servo-controller. Since the shedding frequency was estimated to be around of 0.74Hz, the frequency range of the potentiometer was set between 0 and 1.023Hz.

Retesting with the new code I was able to get good oscillations in the tail of the body when the leading-edge frequency matched the vortex shedding frequency. However, the movement of the leading edge was very mechanical and the movement itself induced a lot of movement in the water which might have moved the tail in itself, and disturbed the vortex street. This can be seen in the video link in Figure 4.8. No lift strong enough to move the rig was generated.

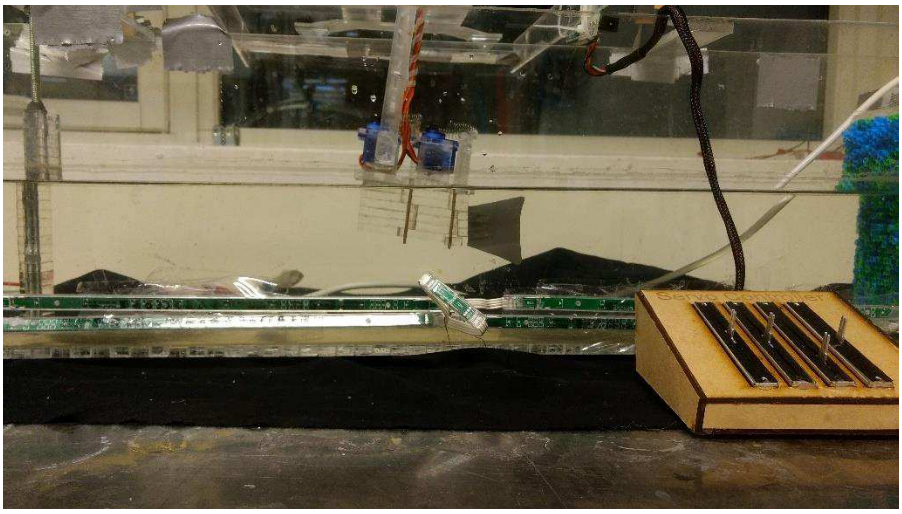


Figure 4.7 - Rig and fish in tank along with the servo-controller. The fish was adjusted to sit level.

Figure 4.8 - The mechanical movement inflicts waves and turbulence to the flow

Figure 4.9 - Oscillation in body and tail caused by leading edge movement

Figure 4.10 - Leading edge synced to the vortex street

4.4 SMA actuated Karman gaiting robot fish

The mechanically jointed fish-body that I moved with servos potentially created its own oscillating effects by pushing water back and forth. To avoid this effect, I wanted to create a soft moving body, by using the SMA-wires explored in section 3.4.3.

4.4.1 Spine and body

Like in section 3.4.3. I create the spine of the body out of *LEXAN®* but used thinner sheets of 1mm for increased flexibility. Rounded crossmembers were used to create the rough 3D shape of the fish. I added living hinges to the front, and removed the space for an Arduino on the inside. The caudal fin was made with thin rubber sheet to better mimic the fish -stiffness. On the top of the body I added a straight handle to run wires in and out of the fish body, as well as glue a 2mm welding rod to for fastening to the force measurement rig.

4.4.2 SMA-wire attachment

During my first experiments with the SMA-wires I found that one of the biggest challenges using them was to get a good electrical connection between the wire and the power source. The SMA wires are practically unsolderable as they move and flex when the heat of the soldering iron is applied on them. Even if a soldered joint were to be applied, the movement of the wire during the retraction and relaxation stages would loosen the bond between the solder joint and wire quickly. My previous solution was to wrap the lead wire tightly around the SMA wire before applying small amounts of solder to ensure good contact at all times. This was rather tedious and the wire itself had to be mechanically fastened with a different method for it to function. In the article *Bending continuous structures with SMAs: a novel robotic fish design* (Rossi, Colorado, Coral, & Barrientos, 2011) the authors had success using screws as both leads and mechanical fastenings for the SMA-wires. To incorporate screw connections in my design, I made counterbored holes for nuts to sit in at 8 points (2:middle,4:center,2:front). The holes were made for *M2* nuts, and in 1.5mm thick plastic so that the bolts would sit flush. The bolts fit snuggly in the holes, and only needed small amounts of superglue to stay in place. To get a good electrical connection with the SMA wires I made copper pads on top of the nuts with copper-tape. The lead wire was soldered to the copper pads, the front and back pads where connected together as common *Vcc*, to reduce the number of wires in the body.

Using screws to hold the SMA wires worked quite well. The only drawback was that the wires would sometimes get loose if they were subjected to too much tension as the strength of clamping of the nuts were not always sufficiently high as I did not dare to put too much torque into the small nuts in fear of shearing the threads. It should also be noted that the pulling force of the SMA wires were far greater than any of the glued connections, meaning that if the muscle wires on both sides of the body were flexed simultaneously the body would break before the wires, which it did on a couple of occasions. This acted as a failsafe, as reapplying glue is far cheaper than acquiring SMA-wires.

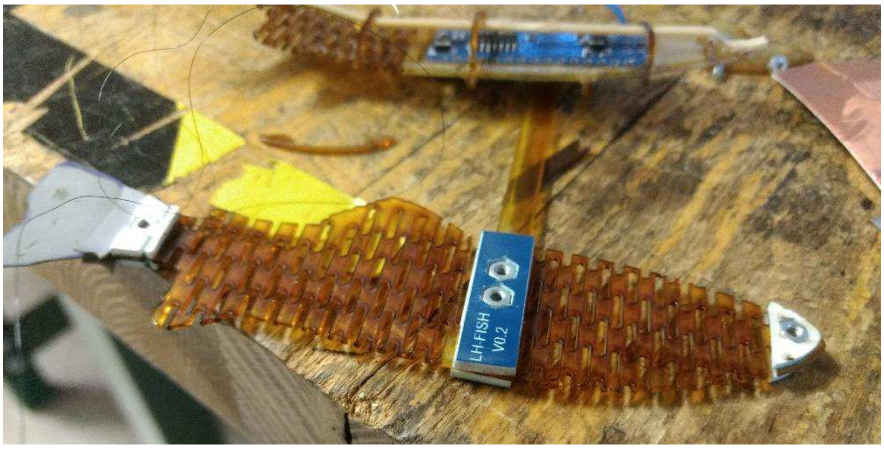


Figure 4.11 - Fish inner body glued together

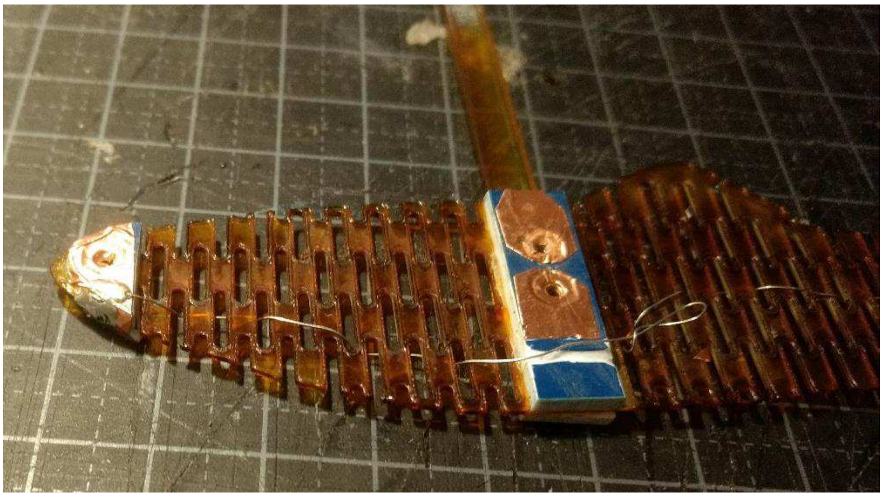


Figure 4.12 - Copper pads for electric connection with SMA wire

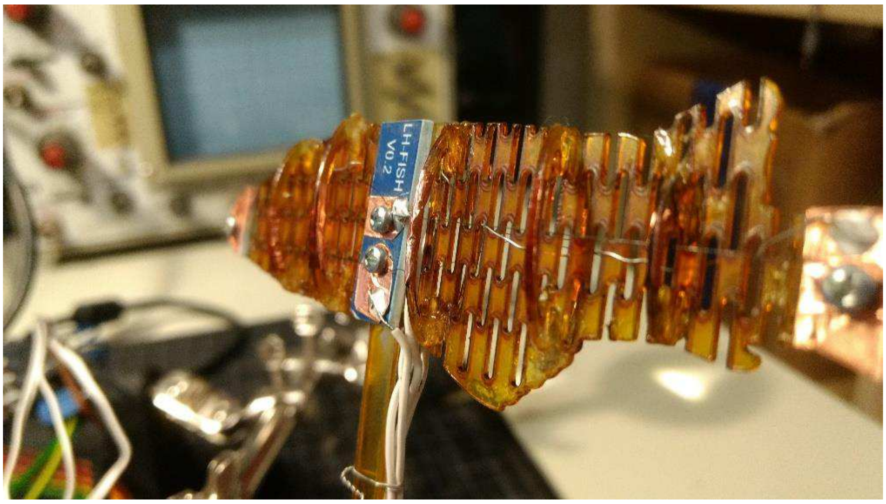


Figure 4.13 - Fish body with wires installed

4.4.3 Waterproofing

Contrary to popular belief, subjecting the electronics to clean water poses relatively small risk for simple low voltage electrical devices. My initial though was therefor to simply subject my entire fish into water without any waterproof skin. This would rid of any buoyancy issues as well as a lot of work, and the higher heat transfer rate of water as compared to air would give the muscle wires a much faster reaction time as the relaxation of the muscle wires is only dependent on the rate at which they cool down. My first test was therefore to subject my robotic fish to water. When I tested the fish in water I found that the heat transfer rate of the water was way too high, and the required current to move the wires slightly was over 3A (at 5v) which was the maximum output of the standard desktop power supplies in our lab. Not only was this impractical but the high current is also a potential risk, small air bubbles on a wire could result in burning of the wire, but more importantly high currents and living beings don’t play well together either. I therefore needed to find a way to waterproof the fish body easily, while keeping it flexible.

With limited time, I teste the resources I had available to waterproof the body. The four materials I ended up testing were; condoms/latex, Rubber-silicone sheets, cling foil, vacuum tape and 3D printed flex material.

Simply tying condoms around the body was great for waterproofing, but it ruined the shape of the body, and the caudal(back) fin was crushed. I tried to apply the latex of the condom in sheets that were cut to size, but quickly found out that no glues would properly stick to the latex. This also became the issue with the rubber-silicone sheets. To make sheets of rubber silicone I mixed up *Ecoflex® 00-30 liquid rubber* and spread it thin on a piece of acrylic glass. Once it set I was left with a very thin sheet of rubber-silicone. I tried to apply this on the body fist with spray glue which would not stick, cyanoacrylate super glue which stuck, but made a brittle and not waterproof seal and vacuum tape. The vacuum tape could not stick to the rubber for very long, but it stuck well to the body of the fish. This lead to the idea of covering the whole fish in a thin layer of vacuum tape. I rolled strips of vacuum tape thin and covered the front of the body. It looked promising waterproof-wise, so I applied power to the SMA-wires to test for movement. The joint moved fine, but around the 8th cycle the skin started to look more brittle and started to tear in a fatigue like manner. A video of this crumbling of the skin can be seen in appendix **A-17**.

Another idea was to 3D print the outer shell of the body in a thin flex material. I used *Inoflex 40* flexible filament and printed half of a body with single wall thickness, 0.4mm nozzle and no fill. The resulting body was flexible but not soft in an organic way, so I did not test this further.

My last resort was to use cling foil in place of vacuum-bag (the vacuum bag plastic was much too thick for this purpose) with the vacuum tape. The cling foil stuck to the vacuum tape well, to take it apart one would need to break the cling foil. By adding vacuum tape all along the edges and the body-crossmembers the cling foil followed the contours of the body and was waterproof, I then added a layer of rubber silicone with a brush to secure that the body was waterproof, even during movement as can be seen in appendix **A-21**.

The cling foil offered little resistance when the body was actuated. However, the SMA-wires ended up burning holes in the foil making it leak. To deal with the hot SMA-wires burning the foil I added heat shields between the wires and the body consisting of copper-tape. The tape distributed the heat over a larger area and holes were no longer burned in the foil. In the beginning the heat shield affected the movement slightly, as can be seen in appendix **A-18**. But this sticking effect disappeared over time.

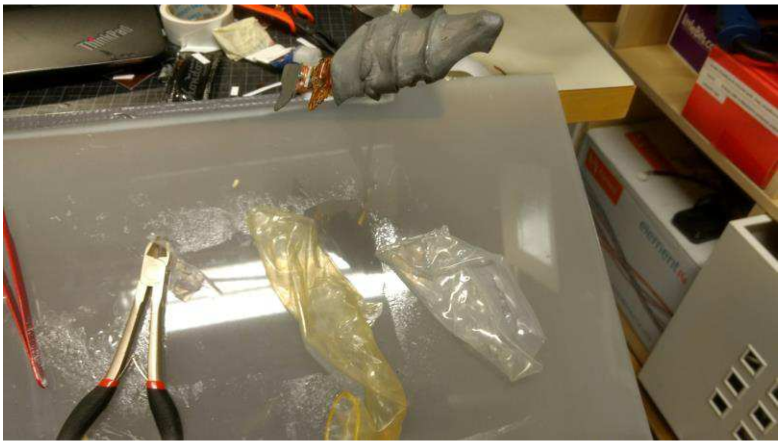


Figure 4.14 - Different tested waterproofing skins

4.4.4 Controlling movement

Activating the SMA wires in air required about 400mA at 5v. Although this is not extremely high, it is higher than the possible pin outputs of the Arduino even if one were to connect all the pins in parallel. This led to the need for a simple motor-controller circuit. I used four MOSFET switches to control the four muscle wires. The muscle wires were the main source of resistance in the circuit, and as such this was where most of the electric energy would be spent, turning to heat. As the front muscle wires were shorter than the aft muscle wires, I had to add a *continuous turning potentiometer* between the power source and the back wires to be able to balance the power that went to the back and front muscle wires. I ran the motor controller with digital signals and no analog or PWM adjustment to keep it simple.

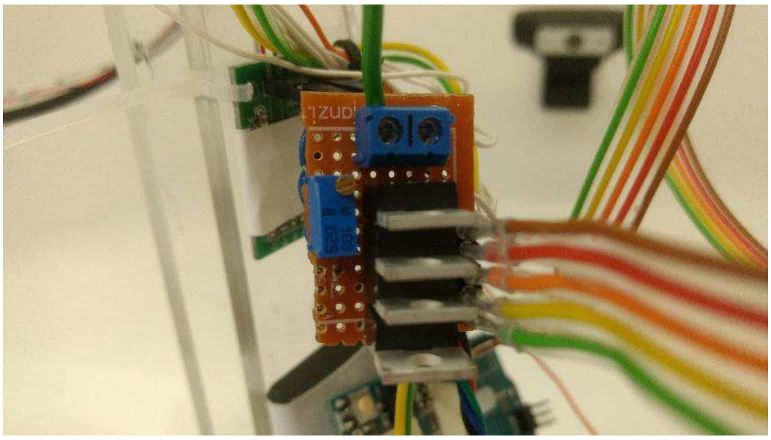


Figure 4.15 - Motor controller for SMA wires

4.4.5 Testing Karman gaiting

The frequency of the fish was controlled in the same manner as for the servo-controlled fish in section 4.3.1. The fish was placed in the Karman street and the frequency was adjusted until the leading edge followed the vortex street.

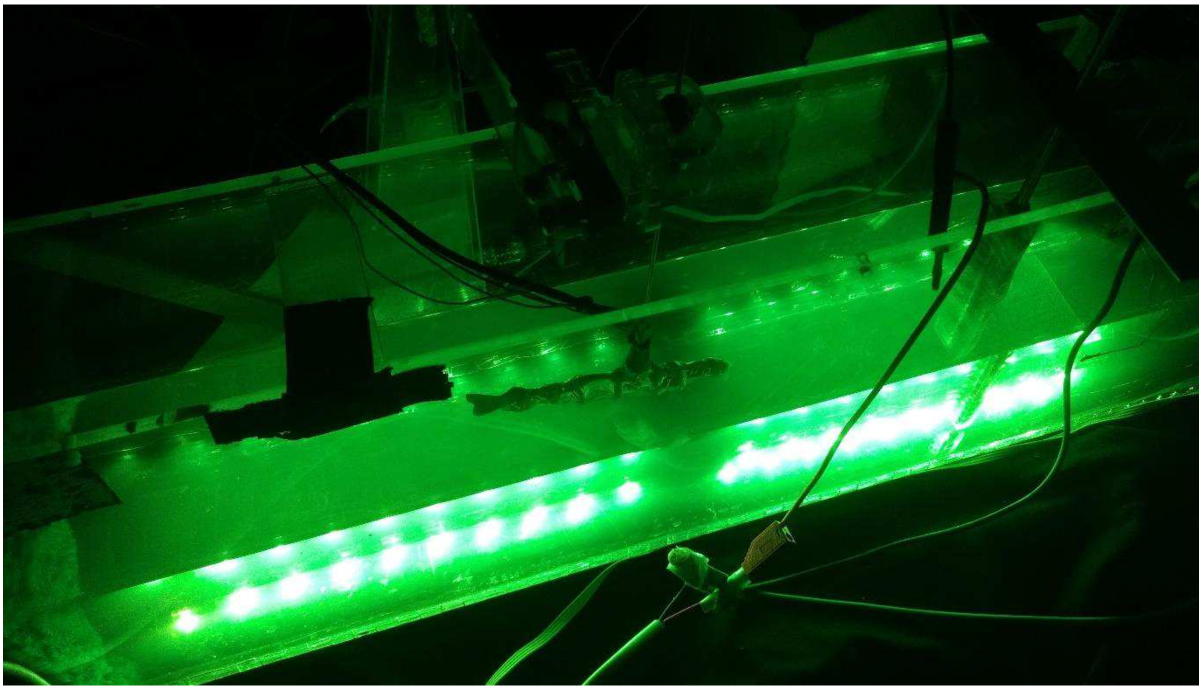


Figure 4.16 - SMA robot fish in water tunnel

The same was then repeated but controlling the tail beat rather than the leading edge.

4.4.6 Results and refinement

When matching the leading edge with the vortex shedding frequency I observed the exerted forces:

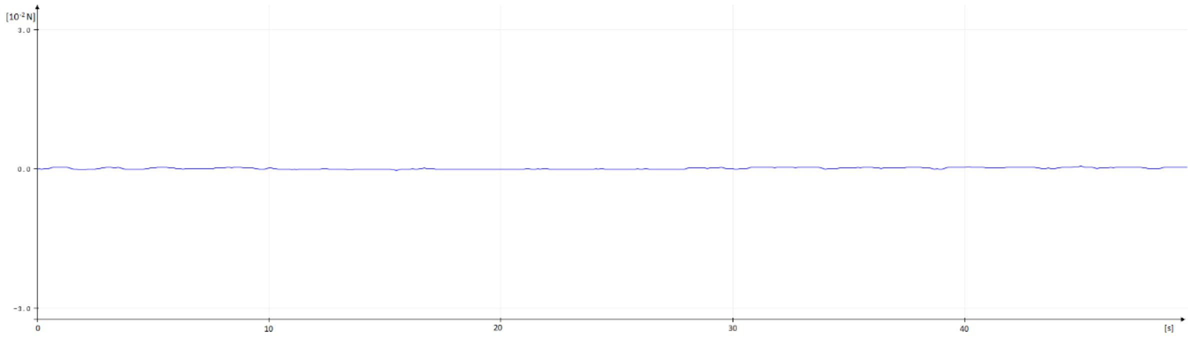


Figure 4.17 - Forces on SMA fish with regulated leading edge. Leading edge movement frequency is matched with vortex shedding.

A realization came from the initial results. Watching the fish changing its camber back and forth I realized that in doing so the effective angle of attack is also changed. I had previously assumed that this small change would be in the right direction, but as Figure 4.18**Error! Reference source not found.**(a) show the AoA changes in the opposite direction of the postulated model that Liao(Liao et al., 2003b) presented. Changing the tail porting however changes the camber and angle of attack in the correct direction according to each other. I changed over my code so that the tail could be controlled and redid the previous experiment. In Figure 4.19**Error! Reference source not found.** the forces exerted on the fish during this experiment can be seen. During the first 20 seconds the tail frequency matches the vortex shedding frequency of the vortex street. Some more dips and variations in the exerted forces with a given frequency can be seen, but the results are not dramatic. As a comparison, from t=20 and until t=45 I gradually increased the tail beat frequency for the fish to go into a full thunniform swim mode. The mean force of the swim mode is hard to determine from the graph to be lower than for the Kármán gaiting and no swim, however the fish can be visually seen generating thrust and backwards vortices in the tank. Again, suggesting that the measuring method of using a load-cell might not be well suited for getting good readings of water-propulsion, the forces are very low.

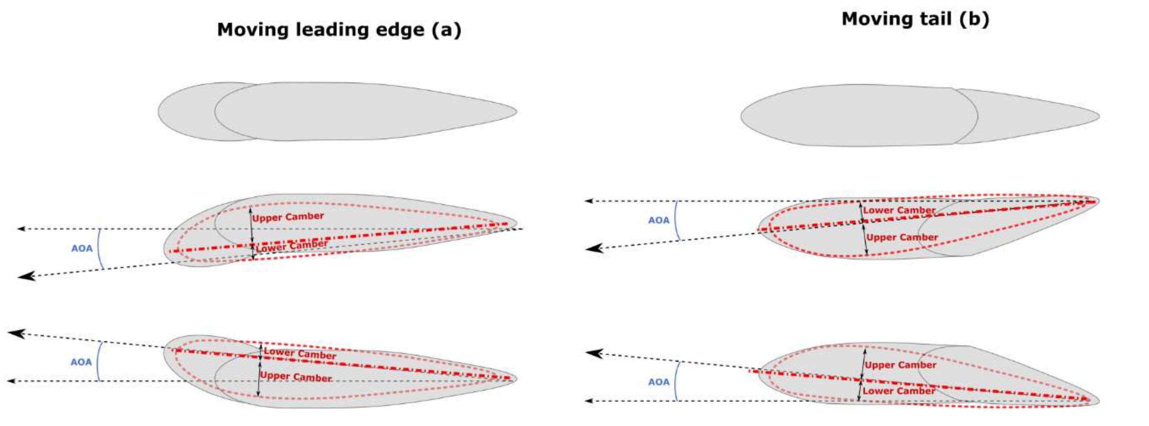


Figure 4.18 - Movement of leading edge and tail. Moving the leading edge (a) changes the camber in the opposite flip of the desired shape. Moving the tail (b) results in AoA and camber changing in the right direction.

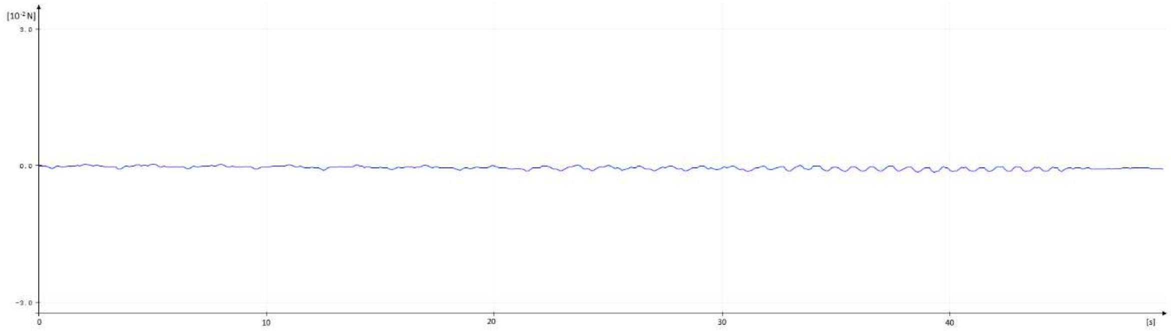


Figure 4.19 - Forces on SMA fish with regulated tail beat frequency. Tailbeat frequency matches vortex shedding for 0<t<20, tailbeat frequency is gradually increased for 20<t<45, movement is stopped at t=45

Figure 4.20 - Tail synced to vertex street

Figure 4.21 - Full swim mode

4.4.7 Artificial Neural Network

Having tested my actuated bodies in the Karman street one of the most noticeable learnings was that even with full control outside of the water, being able to correctly follow the vortex street was extremely difficult. Although the frequency of vortex shedding is somewhat predictable, small disturbances would change the predicted behaviour and the actuated body with constant frequency would be out of phase with the vortex street. Without a feel of the water, making the right adjustments for correct behaviour is near impossible. At the end of my testing period I therefore tried some simple algorithms for machine learning through *artificial neural networks(*ANN). As discussed earlier, it is possible to make good flow predictions using pressure sensors. For the sake of this experiment I used photoresistors as sensors, as they were the smallest and simplest sensors I had at hand, and what I wanted to learn was if the Arduino could handle ANN codes well.

I used two 2kOhm photoresistors along with two 2kOhm resistors to create regulated voltage dividers where the changing middle current could be read as an analog signal by the Arduino. The photoresistors were mounted on the front of the fish as eyes. I wanted to make the fish swim towards the more ideal situation. In this case the fish would like light, and seek out light sources. This meant that if the right photoresistor picked up light, the swimming pattern should move the fish to the right and vice versa. If there is no light the fish should swim fast to find light, and if there is a lot of light the fish can swim calmly until the situation changes. The inspiration for this came from Sean Hodgings neural network robot that he made for Make: (2017).

Hodgins project is based on a supervised learning code for Arduino made by *hobbizine.com* ("Arduino Neural Network," n.d.). This means that the scenarios used for training is predefined. The size of the hidden layer of nodes used to translate the inputs to outputs can also be defined. The Arduino then runs through the training data and creates batches of 1000 of randomized solutions to the hidden layer, the best solution is then chosen and re-randomized using the learning rate variable. The quality of the solutions is asses using the *mean squared error*(MSE) of the outputs compared to the training data. When the MSE is below our wanted error, the hidden nodes are saved, and the training stops.

```
if ( Error < Success ) break ;
```

To translate the swimming code to dimensions that could be used in the layers of an ANN-code I needed to translate the swimming into scenarios of values between zero and one. I gave the front muscles and back muscles three dimensions each: time_left, time_right and time_neutral. Each dimension representing the total time the muscles would be activated/relaxed for the given scenarios. Swimming towards the left would make the front muscles shift between left muscle activation and neutral muscle activation, but the time of right muscle activation would be zero. I did not consider actual mathematical models for swimming modes for these scenarios as the code got complicated enough on its own, and was mainly for experimental purposes as explained above. This gave the following training data:

```
float Input[PatternCount][InputNodes] = {
  { 0, 1},  // LIGHT ON LEFT
  { 1, 1},  // LIGHT ON both
  { 1, 0},  // LIGHT ON RIGHT
  { 0, 0 },  //NO LIGHT

};
const float Target[PatternCount][OutputNodes] = {
  { 0.5, 0, 0.5, 0.5, 0, 0.5 }, //Swim towards left
  { 1, 1, 0.5, 1, 1, 0.5}, //swim slow (low frequency
  { 0, 0.5 , 0.5, 0, 0.5, 0.5}, //swim to the right
  { 0.5, 0.5, 0.2, 0.5, 0.5, 0.2 },   //swim fast
};
```

The main body of the code looks like this:

```
void loop() {
  if ( Error > Success ) {
    train_nn();
  }
  swim(analogRead(A0),analogRead(A1));  //swim contains the InputToOutput function that uses the training
data to convert data to the 6 outputs
}
```

If the error is larger than the success variable, the data has not yet been trained, and the program will run the training algorithm. If, however the program has been trained this is used to convert the analog inputs to the swimming actuations. The full code can be found in Appendix **C-3**. With the relatively simple training data, that could have been coded with a linear approximation/mapping function the training did not take long at all. In fact, the training never went past the second cycle of 1000 solutions before reporting success. However, when I changed up the scenarios to add some more advanced/contradicting cases, the Arduino went on to calculate solutions for over 1.5 hours before I cancelled the operation. It might have been the case that a solution could not be found that satisfied the MSE I had defined, but the calculating power of the Arduino is also very limited. For supervised learning it might be beneficial to pre-calculate the learning data on a more powerful processor. The behaviour itself can be seen in the following video.

Figure 4.22 - Fish swimming through artificial neural network

5 Summarization of results and learnings from the experiments

I have this far tried to give a sequential view of the processes and tests that went into this project, by presenting prototypes, their results and the knowledge/information they generated. The information and learnings gathered at one timestep has aided in decision-making and has formed the prototypes in the next timestep. The results have typically not been statistically quantifiable, but of a qualitative and comparative nature to help in decision making and learning and move the project forward.

5.1 Initial experimental results and process output

The experimental results have to a high degree been an insight into whether or not a solution would be rewarding to keep working on and what made the solutions work/not work. For the initial convergent testing, this meant an assessment based on my own opinion of the found solutions. Trying to assess the solutions objectively, although emotional responses and deeper meanings can be important assessment criteria as well, I judged them through three factors: functionality, feasibility and degree of innovation. The results can be seen in Table 5.1.

Table 5.1 - Results assessment

Section	Solution	Functionality [1-5]	Feasibility [1-5]	Innovative [1-5]	Use Further
3.1	Piezoelectric Micro-swimming	0	2	4	NO
3.2	Turbulent surface for reduced drag	2	3	3	NO

3.3	Sensory flow input from body surface	1	3	2	
3.4.1	Magnetic (actuation)	2	2	2	NO
3.4.2	Servos (actuation)	3	5	0	YES
3.4.3	Shape memory alloys (actuation)	4	4	3	YES
273.5	Karman Gait	3	4	4	YES

The decision was then to further concentrate on the highest scoring solution, Kármán gaiting, while still using the learnings of other viable solutions found at this stage (SMA and Servos).

5.1.1 How to Kármán gait

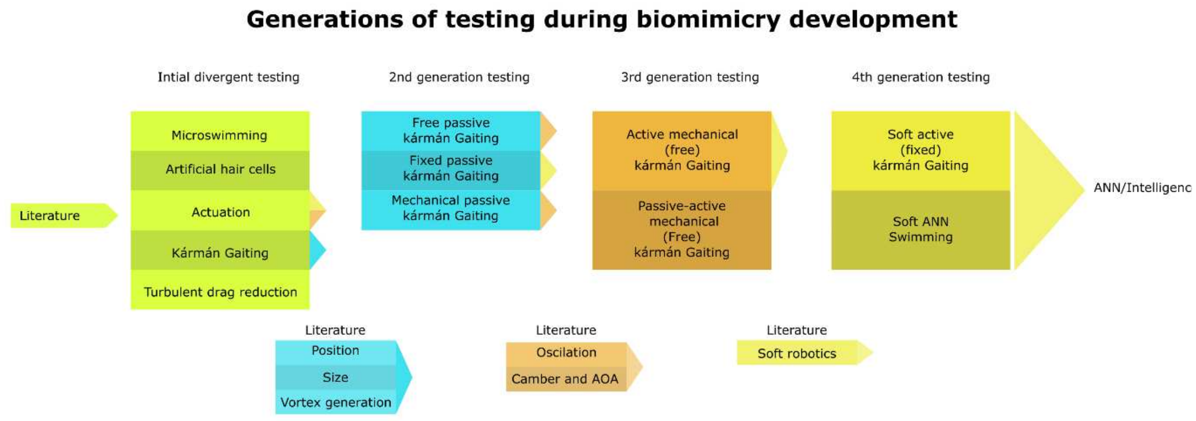
The converging end study of my project aimed at finding out how Kármán gaiting could be recreated through engineering. Although none of my prototypes were able to conclusively show proper Kármán gaiting behaviour, there were some tendencies as to what factors increased behaviours associated with Kármán gaiting.

The forces that act in the flow are relatively of small magnitudes, the Karman gaiting body therefore needs to be soft enough to be able to be acted upon by the water. Soft rubber-silicone bodies and fins made of thin rubber worked better than mechanically jointed bodies.

The best results came from the models that were suspended with a high ability to move freely. For Kármán gaiting to occur the fish body needs to be able to follow the vortex street back and forth, and not just adjust for it at a fixed position. This also includes the ability to rotate.

For prolonged Kármán gaiting there needs to be a reference of the situation in the water around the fish body and ability to adjust position, stiffness and actuation thereafter. In my experiments I used hydrogen bubble generation to visualize the vortex street and shedding frequencies to place and adjust my prototypes to. This worked for short periods in time, but for longer periods of time the prototypes would either “fall off” the vortex street or their phase would get unsynchronized. Without active adjustments and a feel of the water, Kármán gaiting is a game of luck.

5.2 The product development process with biomimicry



The original argument was that biomimicry could be a tool to learn and generate more original ideas. The idea of Kármán gaiting certainly is something that might not have come out of a traditional approach to product development. One of the other interesting outcomes of working with biomimicry is the way it has affected the development process.

My initial low knowledge of the solution space caused a high divergence in the beginning of the project, and much like in set-based design my solution converges towards the designs that worked. The way I used my previous solutions did however differ from typical set-based design. Instead of re-investigating and re-testing solutions on their own in the next generation, they were crossed with each other to generate new solutions. My knowledge of biology in this process came from literature and talks with biologists, this meant that new information surfaced throughout the project as it was uncovered, often triggered by some findings in experiments that gave me the right question. This acquired knowledge was then also crossed with my designs to generate new solutions. The resulting process is something that looks more like an evolutionary process, where the good solutions are crossed with each other with some randomness, and the fittest solutions survived.

6 Discussion

I started this project searching for the problem to solve. Through the use of divergent prototyping I was relatively quickly able to determine which models that were best suited for me to mimic with my available tools and knowledge. Tools exist that aim to make it easier to use nature to find answers such as *AskNature* ("AskNature - Innovation Inspired by Nature," n.d.), these are often large portfolios that showcase many solutions nature has to specific problems. Logically this is only useful when there is a specific problem, and you know what this problem is, it also requires one person to previously have seen a connection from one of nature’s solutions to a human problem. By rather openly exploring the solutions space and freely finding literature to base the prototyping process on I think it is possible to derive more “open” and surprising solutions from nature that might fit better than if one were to use the straight forward approach.

6.1 The results

Through a series of prototypes, I have had varying degree of success in reproducing the Kármán gait. Visually it is possible to see that bodies are affected in the Kármán street when the conditions and dimensions are just right, but to measure and isolate the effect has proven difficult. My prototypes are still simple and juvenile and do not fully grasp the underlying mechanics, and as such I still feel that the success rate has been relatively high. As a comparison, Liao(2003b) reported that his live trout were able to Kármán gait only 80% of the total time, and even lower numbers were reported by Przybilla et al. (2010) who found the rate to be only 7.9%.

Some of the factors I learned to be of highest importance during my testing were the consistency of the bodies and how freely I was able to suspend them in the water. These points might not have been of such significance had I been able to maintain a Kármán street at higher flow speeds (higher energy density), but for my tank they seemed to govern the results. As such it would have been interesting to see the effect that floating the SMA actuated robotic fish in the water would have had on the results. The simple floating rubber-silicone fish was able to Kármán gait, and show that it held a stationary position laterally better than the raft that did not Kármán gait, but it quickly fell of the vortex street. A more involved system of active control, adjustments and sensory input might be the only way for prolonged and controlled Kármán gaiting to occur.

6.2 Future research and applications

Further investigation into the underlying subsystems and mechanics of the gait and their actual significance should form the foundation of any further work into Kármán gaiting, but it would also be interesting to investigate the effect from a system perspective(Bottom-down) through more advanced but flexible and adjustable prototypes. The high amount of inputs and outputs might suggest that applying *reinforcement machine learning* or a similar machine learning algorithm could be a potential way to reduce the workload of finding an optimal solution by letting the machine do the investigation.

The exact mechanics of how fish Kármán gait might not be directly applicable in all marine product settings; a ferry would not be practical if it had the shape or consistency of a fish, but the learnings generated by putting resources into investigating the effect could lead to better understandings of flow patterns, using turbulences/flows more actively and adaptation to flow situations, that could be used to further improve turbines, propellers, wings, cars, etc.

Recent studies have been made that show that adding an upstream deflector to a system of straight-bladed vertical-axis wind turbines (D. Kim & Gharib, 2013) can greatly increase their efficiency. Although it is not because of the same mechanics as in Kármán gaiting, we can imagine that a possible outcome of Kármán gaiting knowledge could be the use of the effect for similar purposes in active or passive adjustments of blades for increased efficiencies.

6.3 Working with biomimicry

Working with biomimicry in product development has not felt much different than a normal development process. Easily available literature has been the most significant help throughout this process. The learning rate has been extremely high, at all stages in the project. Especially during the convergent testing. Even though the literature has been a great help in the process, I also feel like the project has been heavily formed by this being my main source of information in a field in which I have little prior experience. Most of the literature written from a biology point of view concludes in theories and hypothesis , and few physical tests. Theory and practice are very different, and finding out what actually works in practice has been resource demanding and time consuming. Some of the work I put into this investigating was also fundamentally flawed due to my own misunderstandings, such as the AoA change direction in section 4.4.6. Wasteful mistakes like this this could have been avoided had I spent more time investigating theory, but in my case producing prototypes and testing them has been faster and more rewarding teaching wise than a high theoretical load. This has however been a tendency for the whole duration of the project; I find that my solution does not work and when re-visit the literature to find articles that have confirmed the same or similar. Sadly, things are much more visible, once you have done it yourself. But this might also be one of the learnings from this project; failing fast increase the visibility of why you failed when you seek literature for answers. Cross-disciplinary teamwork and cooperation with someone with background in biology might have been a more streamlined way to work to reduce the amount of rework caused by making bad assumptions based on lack of correct understanding.

6.4 Conclusion

I'm not sure if working in the biological solution space has helped me produce and innovate more efficiently, but Kármán gaiting is certainly not something I would have thought to investigate otherwise. I do however feel that applying prototyping tools and set-based thinking greatly increased the learning rate in the biomimicry process. And that the biomimicry process might have been aided more by good product development than vice versa. I think that biomimicry through literature and a good attitude for rapid testing and prototyping is a good match for finding some solutions, but collaboration and cross-disciplinary teamwork might be necessary approach to get the truly astonishing results that are often associated with the word.

Through this project I have greatly investigated the finding and understanding aspect of biomimicry, by testing solutions for recreation. A true recreation was never achieved, but a much better understanding and valuable knowledge of what is needed for proper recreation came out of it. Further investigation might lead to better recreation of the found Kármán gaiting effect and ultimately a scenario where specialising the effect for gain in different fields might be achieved.

Perhaps the most fitting remark to leave this report at is Thomas Edison's words on success and learning: *"I can never find the things that work best until I know the things that don't work."* ("Thomas Edison on Failure," 2015).

7 Bibliography

American Institute of Aeronautics & Astronautics (AIAA). (2017). The Wright Brothers' Wind Tunnel. Retrieved December 20, 2017, from http://legacy.wrightflyer.org/WindTunnel/testing1.html

Arduino Neural Network. (n.d.). Retrieved December 15, 2017, from http://robotics.hobbizine.com/arduinoann.html

AskNature - Innovation Inspired by Nature. (n.d.). Retrieved December 21, 2017, from https://asknature.org/

Bale, R., Hao, M., Bhalla, A. P. S., Patel, N., & Patankar, N. A. (2014). Gray's paradox: A fluid mechanical perspective. *Scientific Reports*, *4*, 5904. https://doi.org/10.1038/srep05904

Barthlott, W., & Neinhuis, C. (1997). Purity of the sacred lotus, or escape from contamination in biological surfaces. *Planta*, *202*(1), 1–8. https://doi.org/10.1007/s004250050096

Beal, D. N., Hover, F. S., Triantafyllou, M. S., Liao, J. C., & Lauder, G. V. (2006). Passive propulsion in vortex wakes. *Journal of Fluid Mechanics*, *549*, 385–402. https://doi.org/10.1017/S0022112005007925

Buehler, J., & Patel, N. V. (2015, March 11). A Real Drag. *Slate*. Retrieved from http://www.slate.com/articles/health_and_science/science/2015/03/mercedes_benz_bionic_car_boxfish_stability_and_agility_paradox_finally_solved.html

Cimbala, J. M., & Yunus, A. Ç. (2006). *Fluid mechanics: fundamentals and applications*. Boston: McGraw-Hill Higher Education.

Doolittle, W. F. (2000). Uprooting the tree of life. *Scientific American*, *282*(2), 90–95.

Farina, S., & Summers, A. (2015). *Biomechanics: Boxed up and ready to go* (Vol. 517). https://doi.org/10.1038/517274a

Fish, F. E. (2006). The myth and reality of Gray's paradox: implication of dolphin drag reduction for technology. *Bioinspiration & Biomimetics*, *1*(2), R17-25. https://doi.org/10.1088/1748-3182/1/2/R01

Fish, F. e., & Lauder, G. v. (2005). Passive and active flow control by swimming fishes and mammals. *Annual Review of Fluid Mechanics*, *38*(1), 193–224. https://doi.org/10.1146/annurev.fluid.38.050304.092201

Fish, F. E., Weber, P. W., Murray, M. M., & Howle, L. E. (2011). The Tubercles on Humpback Whales' Flippers: Application of Bio-Inspired Technology. *Integrative and Comparative Biology*, *51*(1), 203–213. https://doi.org/10.1093/icb/icr016

Fish-Inspired Car. (n.d.). Retrieved December 17, 2017, from https://www.popsci.com/aerodynamic/article/2005-08/fish-inspired-car

Flock, Å., & Wersäll, J. (1962). A Study of the Orientation of the Sensory Hairs of the Receptor Cells in the Lateral Line Organ of Fish, with Special Reference to the Function of the Receptors. *The Journal of Cell Biology*, *15*(1), 19–27. https://doi.org/10.1083/jcb.15.1.19

Gonzales, P. D. (2011). *The evolution of ideas in biomimicry* (Thesis). Massachusetts Institute of Technology. Retrieved from http://dspace.mit.edu/handle/1721.1/69775

Gopalkrishnan, R., Triantafyllou, M. S., Triantafyllou, G. S., & Barrett, D. (1994). Active vorticity control in a shear flow using a flapping foil. *Journal of Fluid Mechanics*, *274*, 1–21. https://doi.org/10.1017/S0022112094002016

Gray, J. (1936). Studies in Animal Locomotion: VI. The Propulsive Powers of the Dolphin. *Journal of Experimental Biology*, *13*(2), 192–199.

Humpback whales inspire next generation wind turbine technology • Materia. (n.d.). Retrieved December 17, 2017, from https://materia.nl/article/humpback-whales-inspire-next-generation-wind-turbine-technology/

Jiang, H., Hawkes, E. W., Fuller, C., Estrada, M. A., Suresh, S. A., Abcouwer, N., … Cutkosky, M. R. (2017). A robotic device using gecko-inspired adhesives can grasp and manipulate large objects in microgravity. *Science Robotics*, *2*(7), eaan4545. https://doi.org/10.1126/scirobotics.aan4545

Kennedy, B. M., Sobek, D. K., & Kennedy, M. N. (2014). Reducing Rework by Applying Set-Based Practices Early in the Systems Engineering Process. *Systems Engineering*, *17*(3), 278–296. https://doi.org/10.1002/sys.21269

Kim, D., & Gharib, M. (2013). Efficiency improvement of straight-bladed vertical-axis wind turbines with an upstream deflector. *Journal of Wind Engineering and Industrial Aerodynamics*, *115*(Supplement C), 48–52. https://doi.org/10.1016/j.jweia.2013.01.009

Kim, S., Spenko, M., Trujillo, S., Heyneman, B., Santos, D., & Cutkosky, M. R. (2008). Smooth Vertical Surface Climbing With Directional Adhesion. *IEEE Transactions on Robotics*, *24*(1), 65–74. https://doi.org/10.1109/TRO.2007.909786

Liao, J. C., Beal, D. N., Lauder, G. V., & Triantafyllou, M. S. (2003a). Fish Exploiting Vortices Decrease Muscle Activity. *Science*, *302*(5650), 1566–1569. https://doi.org/10.1126/science.1088295

Liao, J. C., Beal, D. N., Lauder, G. V., & Triantafyllou, M. S. (2003b). The Kármán gait: novel body kinematics of rainbow trout swimming in a vortex street. *Journal of Experimental Biology*, *206*(6), 1059–1073. https://doi.org/10.1242/jeb.00209

LiaoLab. (2012). *Dead trout in flowing water*. Retrieved from https://www.youtube.com/watch?v=_ZBWnhzYvts

Make: (2017). *Arduino Neural Network Robot Part 3: Running Neural Networks on an Arduino*. Retrieved from https://www.youtube.com/watch?v=97R3TcUh5eI&t=978s

Marras, S., & Porfiri, M. (2012). Fish and robots swimming together: attraction towards the robot demands biomimetic locomotion. *Journal of The Royal Society Interface*, rsif20120084. https://doi.org/10.1098/rsif.2012.0084

Mohd Jani, J., Leary, M., Subic, A., & Gibson, M. A. (2014). A review of shape memory alloy research, applications and opportunities. *Materials & Design (1980-2015)*, *56*(Supplement C), 1078–1113. https://doi.org/10.1016/j.matdes.2013.11.084

Oeffner, J., & Lauder, G. V. (2012). The hydrodynamic function of shark skin and two biomimetic applications. *Journal of Experimental Biology*, *215*(5), 785–795. https://doi.org/10.1242/jeb.063040

Przybilla, A., Kunze, S., Rudert, A., Bleckmann, H., & Brücker, C. (2010). Entraining in trout: a behavioural and hydrodynamic analysis. *Journal of Experimental Biology*, *213*(17), 2976–2986. https://doi.org/10.1242/jeb.041632

Rossi, C., Colorado, J., Coral, W., & Barrientos, A. (2011, Nowember). Bending continuous structures with SMAs: a novel robotic fish design - IOPscience. Retrieved December 14, 2017, from http://iopscience.iop.org/article/10.1088/1748-3182/6/4/045005/meta

Salumäe, T., & Kruusmaa, M. (2013). Flow-relative control of an underwater robot. *Proc. R. Soc. A*, *469*(2153), 20120671. https://doi.org/10.1098/rspa.2012.0671

Scheiman, & Brooks. (1981). Comparison of Experimental and Theoretical Turbulence Reduction from Screens, Honeycomb, and Honeycomb-Screen Combinations | Journal of Aircraft. Retrieved December 14, 2017, from https://arc.aiaa.org/doi/abs/10.2514/3.57538?journalCode=ja

Sfakiotakis, M., Lane, D. M., & Davies, B. C. (1999). Review of Fish Swimming Modes. Retrieved from http://edge.rit.edu/edge/P14029/public/Benchmarking/benchmarking_felix/Review%20of%20Fish%20Swimming%20Modes.pdf

Steinert, M., & Leifer, L. J. (2012). Finding-Ones-Way-Re-Discovering-a-Hunter-Gatherer-Model-based-on-Wayfaring.pdf. *International Journal of Engineering Education*, *28*(2), 251–252.

Testing and Third Party Testing. (2014, December 14). Retrieved December 17, 2017, from https://whalepowercorp.wordpress.com/testing-and-third-party-testing/

Thomas Edison on Failure. (2015, January 13). Retrieved December 22, 2017, from http://www.edisonmuckers.org/thomas-edison-on-failure/

Thomke, S., & Fujimoto, T. (2000). The effect of "front-loading" problem-solving on product development performance. *Journal of Product Innovation Management*, *17*(2), 128–142. https://doi.org/10.1016/S0737-6782(99)00031-4

Toming, G., Chambers, L. D., & Kruusmaa, M. (2014). Experimental study of hydrodynamic forces acting on artificial fish in a von Kármán vortex street. *Underwater Technology*, *32*(2), 81–91. https://doi.org/10.3723/ut.32.081

Ulrich, K. T., & Eppinger, S. D. (2012). *Product Design and Development* (5th ed.). New York, NY: McGraw-Hill.

Vogel, S. (1983). *Life in moving fluids: the physical biology of flow*. Princeton University Press.

Wen, L., Weaver, J. C., & Lauder, G. V. (2014). Biomimetic shark skin: design, fabrication and hydrodynamic function. *Journal of Experimental Biology*, *217*(10), 1656–1666. https://doi.org/10.1242/jeb.097097

APPENDIX A

Video Archive

Appendix D-1 – Project Thesis

A-1: QR code to full video and photo archive

A-2: Fist hydrogen Bubbles:

https://photos.app.goo.gl/AlpT8tMKumuKbeaC3

A-3: Hydroogen bubble sheet:

https://photos.app.goo.gl/2G8bsXpW8mzspntA3

Appendix D-1 – Project Thesis

A-4: Difficulty of visualizing laser sheets

https://photos.app.goo.gl/bKz2IEtF1eb8nwQw1

A-5: Vortex shedding around D-sylinder

https://photos.app.goo.gl/A6HnNeWjJn0sYVZM2

A-6: Fixed Karman gaiting

https://photos.app.goo.gl/aE5tcuBX5dCZ5QMl2

A-7: Roped Silicone fish late in the Karman street
https://photos.app.goo.gl/qQeFGl2jTdTdVisy2

A-8: Roped silicone fish close to low pressure zone:
https://photos.app.goo.gl/60nvRJGQiBYN0FdH3

A-9: Raft fish without Kárman street drags behind
https://photos.app.goo.gl/8GoVB9zQ1TLUmhTw1

Appendix D-1 – Project Thesis

A-10: Raft fish Kármán gaits
https://photos.app.goo.gl/lBjiWVDRBq07Urq83

A-11: Raft in low pressere zone:
https://photos.app.goo.gl/3DAMmREepmW86uo32

A-12: Raft without fish failing to Kárman gait moves back
https://photos.app.goo.gl/0NHrDA8WffZwQfMw2

A-13: Trying to manually control mechanical body smoothly
https://photos.app.goo.gl/jZk3fSaVwkJCPP5w2

A-14: Mechanical Leading edge synced:
https://photos.app.goo.gl/r5o0GougAwQM4RkH3

A-15: Mechanical leading edge causing high movement of water:
https://photos.app.goo.gl/H6z2P0NwUesplOPb2

A-16: Mehcanical leading edge causing osciliations in body and tail:
https://photos.app.goo.gl/DzDzFIDqI5zWKxsk2

A-17: Crumbly vacuum tape body
https://photos.app.goo.gl/JzwAnraODivML6Tm1

A-18: Heat shield affecting movement:
https://photos.app.goo.gl/yRb4hvg4vlE2PHAi2

Appendix D-1 – Project Thesis

A-19: SMA functionality test

https://photos.app.goo.gl/x1iFZXGkstLJvR593

A-20: First SMA swim test:

https://photos.app.goo.gl/bzw7NoLItTl5hO483

A-21: SMA movement and waterproof test

https://photos.app.goo.gl/DzeBorD2MDkBdADt2

Appendix D-1 – Project Thesis

A-22: Leading edge SMA synced to shedding:
https://photos.app.goo.gl/AFOeLkpHgsvp8SPs1

A-23: Tail affected by leading edge synced to shedding SMA fish:
https://photos.app.goo.gl/M7jH3VUQhGWuoASV2

A-24: Tail synched to vortex street shedding SMA fish:
https://photos.app.goo.gl/mE2vj1OIsKdQ31qw1

A-25: Full Swim mode of SMA fish:
https://photos.app.goo.gl/G5oougxfa2cqHKMI2

A-26: Artificial neural network swimming
https://photos.app.goo.gl/kkCaFuyElMfXkV6l1

APPENDIX B

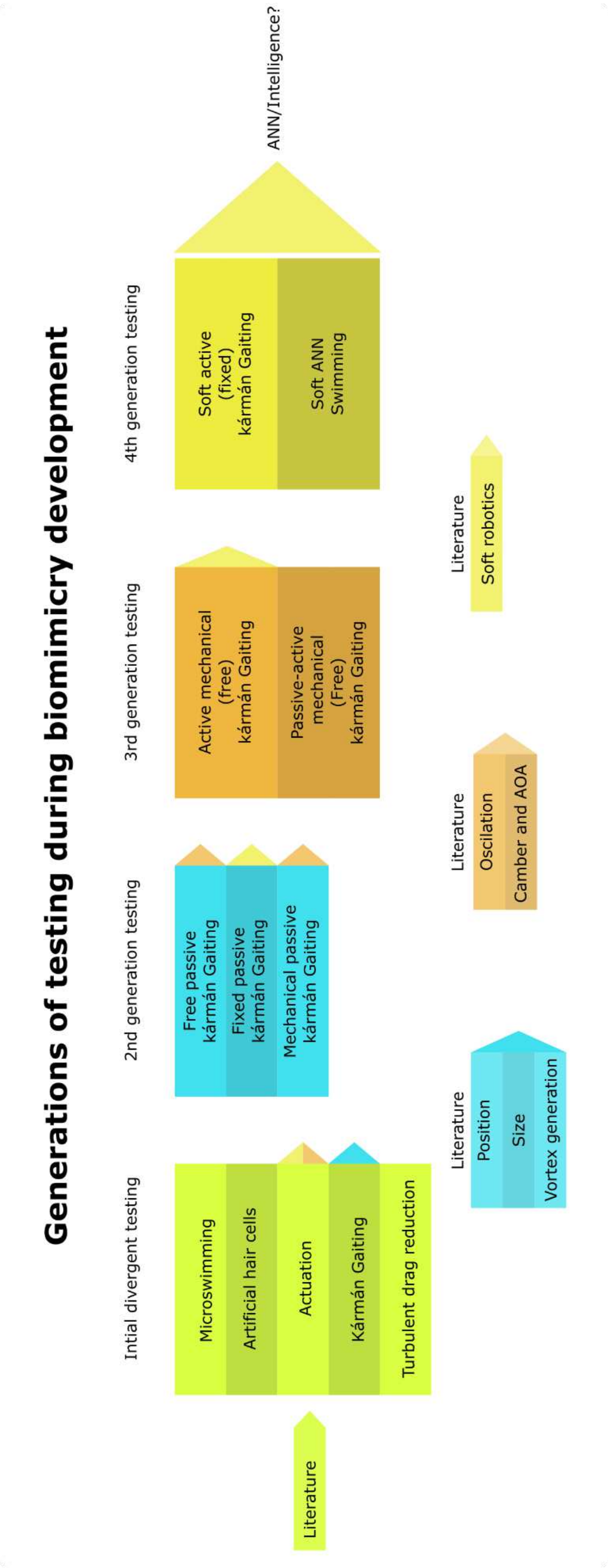
Photo and figures archive

Appendix D-1 – Project Thesis

B – 1: Full photo and video archive QR code

Appendix D-1 – Project Thesis

B – 2: Large version project progression



Appendix D-1 – Project Thesis

APPENDIX C

Arduino codes

Appendix D-1 – Project Thesis

C – 1: Code for muscle wire control of fish with variable frequency

```
/*Code For controlling muscle wire trout with potentiometer
* Håvard Vestad
*/

int MusclePins[] = {3, 4, 5, 6};
/*Code For controlling muscle wire trout with potentiometer
* Håvard Vestad
*/

int MusclePins[] = {3, 4, 5, 6};
void setup() {
  Serial.begin(9600);
  for (int i = 0; i < 5; i++) {
    pinMode(MusclePins[i], OUTPUT);
  }
}
void loop() {
  for (int i = 0; i < 2; i++) {
    digitalWrite(MusclePins[i], HIGH);
   digitalWrite(MusclePins[i + 2], HIGH);
    delay(analogRead(A2)*2);
    Serial.println(analogRead(A2));
    for (int i = 0; i < 5; i++) {
      digitalWrite(MusclePins[i], LOW);
    }
    delay(analogRead(A3));
  }
}
```

APPENDIX B – 1: Full photo and figure and video archive QR code

```
/*Code For controlling Servos
* Håvard Vestad

*Front servos are controlled manually
* Back servos follow a sinusoidial position
```

```
*Frequnecy is determined by potentiometer 3, and phase shift by pot 2
*/

#include <Servo.h>

Servo front1;
Servo front2;
Servo back1;
Servo back2;

int pin2 = A0;
int pin1 = A1;
int pin3 = A3;
int pin4 = A2;

int val1 = 0;
int val2 = 0;
int val3 = 0;
int val4 = 0;

float x = 0;

const float pi = 3.14;

void setup() {
  // put your setup code here, to run once:

  front1.attach(3);
  front2.attach(5);
  back1.attach(6);
  back2.attach(9);
  Serial.begin(9600);
}
```

```
void loop() {


  if (x >= (2 * pi)) {
    x = 0;
  }
  x = x + 2 * pi / (analogRead(pin3) / 2);


  val1 = map(analogRead(pin1), 0, 1023, 170, 10);
  front1.write(val1);
  Serial.println(val1);


   val4 = map(analogRead(pin4), 0, 1023, 170, 10);
  front2.write(val1);
  Serial.println(val1);



//Generate sinusoidial movement, with frequenzy given by datapoints from pin3
// Phase shift is deffined by pin2



  val3 = 95 + 85 * sin(x + (map(analogRead(pin2),0,1023,0,2*pi)));



  back1.write(map(val3,10,170,60,160));
  back2.write((map(val3,170,10,60,160));


  // put your main code here, to run repeatedly:
  delay(10);



}
```

C – 3: Artificial neural network code for machine learning swim patterns based on light

```
/************************************************
* This program is comprised of various open source libraries and
examples.
* Original elements: void swim(); is made by Håvard Vestad
* Original element: void InputToOutput(); is made by Håvard Vestad based
on Sean Hodgins code:
* https://github.com/idlehandsproject/makennbot
*
* This is e rewrite of original code that was deleted due to
* Arduinos autosave overwriting original files. As it stands it
* is untested and may be error-prone.
* 15.12.17
*
* Information on the Neural Network and the Neural network training code
* can be found here: http://robotics.hobbizine.com/arduinoann.html
*
*/

#include <Arduino.h> //Depends on breakout board
#include <math.h>



long eventback = millis();
long eventfront = millis();

/*******************************************************************
   Network Configuration - customized per network
 *******************************************************************/

const int PatternCount = 16;
const int InputNodes = 2;
const int HiddenNodes = 7;
const int OutputNodes = 6;
const float LearningRate = 0.3;
const float Momentum = 0.9;
const float InitialWeightMax = 0.5;
const float Success = 0.0015;
```

```
float Input[PatternCount][InputNodes] = {
  { 0, 1},  // LIGHT ON LEFT
  { 1, 1},  // LIGHT ON both
  { 1, 0},  // LIGHT ON RIGHT
  { 0, 0 },  //NO LIGHT

};

// Note : Frequenzy of back swim patern will be corrected to match front
if cases are ill-defined
const float Target[PatternCount][OutputNodes] = {
  { 0.5, 0, 0.5, 0.5, 0, 0.5 },      //Swim towards left
  { 1, 1, 0.5, 1, 1, 0.5},           //swim slow (low frequency
  { 0, 0.5 , 0.5, 0, 0.5, 0.5},      //swim to the right
  { 0.5, 0.5, 0.2, 0.5, 0.5, 0.2 },   //swim fast

};

/******************************************************************
   End Network Configuration
 ******************************************************************/

int i, j, p, q, r;
int ReportEvery1000;
int RandomizedIndex[PatternCount];
long  TrainingCycle;
float Rando;
float Error = 2;
float Accum;

float Hidden[HiddenNodes];
float Output[OutputNodes];

float HiddenWeights[InputNodes + 1][HiddenNodes];
float OutputWeights[HiddenNodes + 1][OutputNodes];
float HiddenDelta[HiddenNodes];
float OutputDelta[OutputNodes];
float ChangeHiddenWeights[InputNodes + 1][HiddenNodes];
float ChangeOutputWeights[HiddenNodes + 1][OutputNodes];
float t[OutputNodes];
int ErrorGraph[64];
```

```
void setup() {

  pinMode(A1, INPUT);
  pinMode(A2, INPUT);



  randomSeed(analogRead(A1));       //Collect a random ADC sample for
Randomization.
  ReportEvery1000 = 1;
  for ( p = 0 ; p < PatternCount ; p++ ) {
    RandomizedIndex[p] = p ;
  }

  Serial.begin(9600);
  delay(100);


}

void loop() {

  if ( Error > Success ) {
    train_nn();
  }
  swim(analogRead(A0),analogRead(A1));  //swim contains the InputToOutput
function that uses the training data to convert data to the 6 outputs

}



void swim(float right_eye,float left_eye) {
  /*
     time_back_lef = t(0)
     time_back_right = t(1)
     time_back_neutral = t(2)
     time_front_left =t(3)
     time_front_right = t(4)
     time_front_neutral = t(5)
  */
```

```
InputToOutput(right_eye, left_eye);

t[1]=Output [0];
t[1]=Output [1];
t[1]=Output [2];
t[1]=Output [3];
t[1]=Output [4];
t[1]=Output [5];

/*
   Front and back frequenzies should match so that there wont be an
   offset in the swimming pattern (not garantied with machine learning
   I make a correctional variable x so that the total times of front
   and back will match
*/
float x = t[0] + t[1] + t[2] / (t[3] + t[4] + t[5]);

/*
    the times need to be converted from variables between 0 and 1 to
    seconds. I chose to range the times from 0 to 2 seconds (2000 ms)
*/
t[0] = t[0] * 2000 ;
t[1] = t[1] * 2000 ;
t[2] = t[2] * 2000 ;
t[3] = t[3] * 2000 * x ;
t[4] = t[4] * 2000 * x ;
t[5] = t[5] * 2000 * x ;
long eventFront = 0;
 long eventBack = 0;
int prev_caseFront = 1;
int CaseFront=0;
int CaseBack=0;
int prev_caseBack = 1;


switch (CaseFront) {
  case 0: //Right
    if ( millis() > eventFront + t[2]) {
      digitalWrite(3, HIGH);
      digitalWrite(4, LOW);
```

```
        CaseFront = 2;
        prev_caseFront = 0;
        eventFront = millis();
      }
    case 1: //Left
      if ( millis() > eventFront + t[2]) {
        digitalWrite(3, LOW);
        digitalWrite(4, HIGH);
        CaseFront = 2;
        prev_caseFront = 1;
        eventFront = millis();
      }
    case 2:
      if (prev_caseFront == 1) {
        if ( millis() > eventFront + t[1]) {
          digitalWrite(3, LOW);
          digitalWrite(4, LOW);
          CaseFront = 0;
          prev_caseFront = 2;
          eventFront = millis();
        }
      }
      if (prev_caseFront == 0) {
        if ( millis() > eventFront + t[0]) {
          digitalWrite(3, LOW);
          digitalWrite(4, LOW);
          CaseFront = 1;
          prev_caseFront = 2;
          eventFront = millis();
        }
      }
  }

  switch (CaseBack) {
    case 0: //Right
      if ( millis() > eventBack + t[5]) {
        digitalWrite(5, HIGH);
        digitalWrite(6, LOW);
        CaseBack = 2;
        prev_caseBack = 0;
        eventBack = millis();
      }
    case 1: //Left
      if ( millis() > eventBack + t[5]) {
        digitalWrite(5, LOW);
        digitalWrite(6, HIGH);
        CaseBack = 2;
        prev_caseBack = 1;
        eventBack = millis();
      }
    case 2:
      if (prev_caseBack == 1) {
        if ( millis() > eventBack + t[4]) {
          digitalWrite(5, LOW);
          digitalWrite(6, LOW);
          CaseBack = 0;
          prev_caseBack = 2;
          eventBack = millis();
        }
      }
      if (prev_caseBack == 0) {
        if ( millis() > eventBack + t[3]) {
          digitalWrite(5, LOW);
          digitalWrite(6, LOW);
          CaseBack = 1;
          prev_caseBack = 2;
          eventBack = millis();
        }
      }
  }
}


//TRAINS THE NEURAL NETWORK
void train_nn() {

/******************************************************************
* Initialize HiddenWeights and ChangeHiddenWeights
******************************************************************/

  for( i = 0 ; i < HiddenNodes ; i++ ) {
    for( j = 0 ; j <= InputNodes ; j++ ) {
      ChangeHiddenWeights[j][i] = 0.0 ;
      Rando = float(random(100))/100;
      HiddenWeights[j][i] = 2.0 * ( Rando - 0.5 ) * InitialWeightMax ;
    }
  }
/******************************************************************
* Initialize OutputWeights and ChangeOutputWeights
******************************************************************/

  for( i = 0 ; i < OutputNodes ; i ++ ) {
    for( j = 0 ; j <= HiddenNodes ; j++ ) {
      ChangeOutputWeights[j][i] = 0.0 ;
      Rando = float(random(100))/100;
      OutputWeights[j][i] = 2.0 * ( Rando - 0.5 ) * InitialWeightMax ;
    }
  }
  Serial.println("Initial/Untrained Outputs: ");
  toTerminal();
/******************************************************************
* Begin training
******************************************************************/

  for( TrainingCycle = 1 ; TrainingCycle < 2147483647 ; TrainingCycle++)
{

/******************************************************************
* Randomize order of training patterns
******************************************************************/

    for( p = 0 ; p < PatternCount ; p++) {
      q = random(PatternCount);
      r = RandomizedIndex[p] ;
      RandomizedIndex[p] = RandomizedIndex[q] ;
      RandomizedIndex[q] = r ;
    }
    Error = 0.0 ;
/******************************************************************
* Cycle through each training pattern in the randomized order
******************************************************************/
    for( q = 0 ; q < PatternCount ; q++ ) {
      p = RandomizedIndex[q];
```

```
/*****************************************************************
* Compute hidden layer activations
*****************************************************************/

      for( i = 0 ; i < HiddenNodes ; i++ ) {
        Accum = HiddenWeights[InputNodes][i] ;
        for( j = 0 ; j < InputNodes ; j++ ) {
          Accum += Input[p][j] * HiddenWeights[j][i] ;
        }
        Hidden[i] = 1.0/(1.0 + exp(-Accum)) ;
      }

/*****************************************************************
* Compute output layer activations and calculate errors
*****************************************************************/

      for( i = 0 ; i < OutputNodes ; i++ ) {
        Accum = OutputWeights[HiddenNodes][i] ;
        for( j = 0 ; j < HiddenNodes ; j++ ) {
          Accum += Hidden[j] * OutputWeights[j][i] ;
        }
        Output[i] = 1.0/(1.0 + exp(-Accum)) ;
        OutputDelta[i] = (Target[p][i] - Output[i]) * Output[i] * (1.0 -
Output[i]) ;
        Error += 0.5 * (Target[p][i] - Output[i]) * (Target[p][i] -
Output[i]) ;
      }

/*****************************************************************
* Backpropagate errors to hidden layer
*****************************************************************/

      for( i = 0 ; i < HiddenNodes ; i++ ) {
        Accum = 0.0 ;
        for( j = 0 ; j < OutputNodes ; j++ ) {
          Accum += OutputWeights[i][j] * OutputDelta[j] ;
        }
        HiddenDelta[i] = Accum * Hidden[i] * (1.0 - Hidden[i]) ;
      }
```

```
/******************************************************************
* Update Inner-->Hidden Weights
******************************************************************/



      for( i = 0 ; i < HiddenNodes ; i++ ) {
        ChangeHiddenWeights[InputNodes][i] = LearningRate *
HiddenDelta[i] + Momentum * ChangeHiddenWeights[InputNodes][i] ;
        HiddenWeights[InputNodes][i] +=
ChangeHiddenWeights[InputNodes][i] ;
        for( j = 0 ; j < InputNodes ; j++ ) {
          ChangeHiddenWeights[j][i] = LearningRate * Input[p][j] *
HiddenDelta[i] + Momentum * ChangeHiddenWeights[j][i];
          HiddenWeights[j][i] += ChangeHiddenWeights[j][i] ;
        }
      }

/******************************************************************
* Update Hidden-->Output Weights
******************************************************************/

      for( i = 0 ; i < OutputNodes ; i ++ ) {
        ChangeOutputWeights[HiddenNodes][i] = LearningRate *
OutputDelta[i] + Momentum * ChangeOutputWeights[HiddenNodes][i] ;
        OutputWeights[HiddenNodes][i] +=
ChangeOutputWeights[HiddenNodes][i] ;
        for( j = 0 ; j < HiddenNodes ; j++ ) {
          ChangeOutputWeights[j][i] = LearningRate * Hidden[j] *
OutputDelta[i] + Momentum * ChangeOutputWeights[j][i] ;
          OutputWeights[j][i] += ChangeOutputWeights[j][i] ;
        }
      }
    }

/******************************************************************
* Every 1000 cycles send data to terminal for display
******************************************************************/
    ReportEvery1000 = ReportEvery1000 - 1;
    if (ReportEvery1000 == 0)
    {
      Serial.println();
      Serial.println();
      Serial.print ("TrainingCycle: ");
      Serial.print (TrainingCycle);
      Serial.print ("  Error = ");
      Serial.println (Error, 5);

      toTerminal();

      if (TrainingCycle==1)
      {
        ReportEvery1000 = 999;
      }
      else
      {
        ReportEvery1000 = 1000;
      }
    }


/******************************************************************
* If error rate is less than pre-determined threshold then end
******************************************************************/

    if( Error < Success ) break ;
  }
  Serial.println ();
  Serial.println();
  Serial.print ("TrainingCycle: ");
  Serial.print (TrainingCycle);
  Serial.print ("  Error = ");
  Serial.println (Error, 5);

  toTerminal();

  Serial.println ();
  Serial.println ();
  Serial.println ("Training Set Solved! ");
  Serial.println ("--------");
  Serial.println ();
  Serial.println ();
  ReportEvery1000 = 1;
}

void toTerminal()
{

  for( p = 0 ; p < PatternCount ; p++ ) {
    Serial.println();
    Serial.print ("  Training Pattern: ");
    Serial.println (p);
    Serial.print ("  Input ");
    for( i = 0 ; i < InputNodes ; i++ ) {
      Serial.print (Input[p][i], DEC);
      Serial.print (" ");
    }
    Serial.print ("  Target ");
    for( i = 0 ; i < OutputNodes ; i++ ) {
      Serial.print (Target[p][i], DEC);
      Serial.print (" ");
    }
/*******************************************************************
* Compute hidden layer activations
*******************************************************************/

    for( i = 0 ; i < HiddenNodes ; i++ ) {
      Accum = HiddenWeights[InputNodes][i] ;
      for( j = 0 ; j < InputNodes ; j++ ) {
        Accum += Input[p][j] * HiddenWeights[j][i] ;
      }
      Hidden[i] = 1.0/(1.0 + exp(-Accum)) ;
    }

/*******************************************************************
* Compute output layer activations and calculate errors
*******************************************************************/

    for( i = 0 ; i < OutputNodes ; i++ ) {
      Accum = OutputWeights[HiddenNodes][i] ;
      for( j = 0 ; j < HiddenNodes ; j++ ) {
        Accum += Hidden[j] * OutputWeights[j][i] ;
      }
      Output[i] = 1.0/(1.0 + exp(-Accum)) ;
    }
    Serial.print ("  Output ");
    for( i = 0 ; i < OutputNodes ; i++ ) {
      Serial.print (Output[i], 5);
      Serial.print (" ");
    }
  }

}



void InputToOutput(float In1, float In2)
{



  float TestInput[] = {0, 0};
  TestInput[0] = In1;
  TestInput[1] = In2;

  /******************************************************************
    Compute hidden layer activations
  ******************************************************************/

  for ( i = 0 ; i < HiddenNodes ; i++ ) {
    Accum = HiddenWeights[InputNodes][i] ;
    for ( j = 0 ; j < InputNodes ; j++ ) {
      Accum += TestInput[j] * HiddenWeights[j][i] ;
    }
    Hidden[i] = 1.0 / (1.0 + exp(-Accum)) ;
  }

  /******************************************************************
    Compute output layer activations and calculate errors
  ******************************************************************/

  for ( i = 0 ; i < OutputNodes ; i++ ) {
    Accum = OutputWeights[HiddenNodes][i] ;
    for ( j = 0 ; j < HiddenNodes ; j++ ) {
      Accum += Hidden[j] * OutputWeights[j][i] ;
    }
    Output[i] = 1.0 / (1.0 + exp(-Accum)) ;
  }

}
```

Appendix D-1