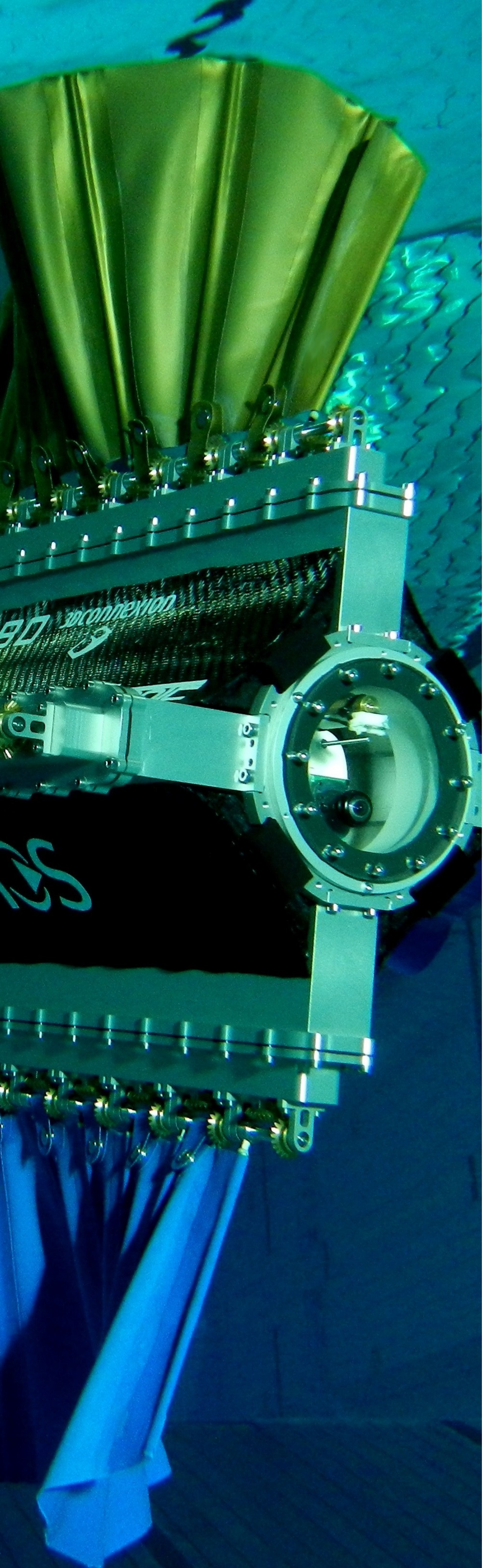


June 16, 2014

Focus Project
Sepios

Riding the Wave of Progress

www.sepios.org

ETH zürich

Autonomous Systems Lab

AUTONOUMOUS SYSTEMS LABORATORIES
ETH ZÜRICH

FOCUS PROJECT

Sepios: Riding the Wave of Progress

Authors:
Martin Peter MÖLLER
Alessandro SCHÄPPI
Pascal BUHOLZER
Markus WEGMANN
Antoine SEEWER
Fabio DUBOIS
Julian Felix FLURY
Vincent FREIERMUTH
Marjolijn HESLINGA

Supervisors:
Prof. Dr. Roland SIEGWART
Dr. Roland HAAS
Stefan BÄRTSCHI
Gregory HITZ
Andreas SCHAFFNER

www.sepios.org

June 16, 2014

ETH zürich

Autonomous Systems Lab

Declaration of Originality

We hereby declare that the written work we have submitted entitled

Focus Project Sepios, Riding the Wave of Progress

is original work which we alone have authored and which is written in our own words.[1]

Authors

Martin	Möller
Alessandro	Schäppi
Pascal	Buholzer
Antoine	Seewer
Julian	Flury
Vincent	Freiermuth
Fabio	Dubois
Markus	Wegmann

Supervising lecturers

Prof. Dr. Roland	Siegwart
Dr. Roland	Haas
Stefan	Bertschi
Gregory	Hitz
Andreas	Schaffner

With the signature we declare that we have been informed regarding normal academic citation rules and that we have read and understood the information on 'Citation etiquette'[2]. The citation conventions usual to the discipline in question here have been respected.

The above written work may be tested electronically for plagiarism.

Place and date

Signature

---

[1] Co-authored work: The signatures of all authors are required. Each signature attests to the originality of the entire piece of written work in its final form.

[2] `https://www.ethz.ch/content/dam/ethz/main/education/rechtliches-abschluesse/leistungskontrollen/plagiarism-citationetiquette.pdf`

Abstract

Sepios is a bionics project at ETH Zürich interested in mimicking the undulating side fins of squid and cuttlefish. These animals feature a very high agility when moving under water. A technical implementation of their propulsion mechanism promises to be practically silent compared to conventional propellers. It might also have a lower tendency to entangle in sea grass.

A submersible robot with cuttlefish inspired fins was built. The goal was to design the robot in a way that would enable it to move omnidirectionally. For this purpose four undulating fins were arranged symmetrically around one central cylinder. The number and arrangement of the fins can be varied modularly. They are actuated by servomotors which are placed within waterproof enclosures. This allows for a high degree of flexibility as each fin segment can be steered individually.

The robot acts as a proof of concept for the high flexibility of the described propulsion mechanism. Its primary short-term purpose is to gain a deeper understanding of the fin behaviour. Long term applications might include submarine technical inspections or the filming of marine habitats.

Preface

Many individuals and companies were so kind as to support focus project Sepios. We received a lot technical advice and sponsored materials and finances. Here we wish to extend our thanks to them for their generous support.

First of all, we would like to thank the Autonomous Systems Lab (ASL) under the direction of Prof. Dr. Roland Siegwart for giving us the great opportunity of participating in this project. We thank our coaches Gregory Hitz, Stefan Bertschi and Andreas Schaffner for brilliant advice and countless invaluable hours of counselling.

Special thanks go to Herbert Bieri and André Bitzer from Kubo Tech AG. They helped us with expertise in the field of sealing. They were very forthcoming and always ready to answer questions. The same applies to Walter Bachmann from the Zentralwerkstatt Physik at ETH. Without his constant feedback we would not have been able to finish our technical drawings in time. We also wish to express our gratitude to Fabian Günther from the Naro projects. Even during our hardest days, shortly before Christmas, he was ever present to aid us with his clear vision for technical details. Special thanks also go to Silvain Michel from EMPA. In the beginning of our project he provided us with detailed knowledge on electroactive polymers.

We also want to thank Beat Schib from National Instruments. He visited us several times in our office providing brilliant advice concerning LabVIEW programming. Furthermore we would like to thank Andreas Stuker and Marcel Wachter from the Zentralwerkstatt Physik for their inputs on manufacturing and cost optimisation. We also appreciated the support of Martin Schütz who helped us solve many CAD problems.

And finally we would like to thank Kubo, Swaytronic, National Instruments, Zentralwerkstatt Physik at ETH, Hitec, 3DConnexion, Qualicut and Kami as well as ABB, Hasler, Suter Kunststoffe, Alumex AG, Distrilec, Puag, JKI, Kundert, Jakob Antriebstechnik, Thümer Teile, Maagtechnic, Igus, UTZ, 2Xideas, Tauchbasis Inauen and ASVZ for their enormous material and financial sponsoring.

Contents

Acronyms

I^2C	Inter-Integrated Circuit. 1
ABB	Asea Brown Boveri. 1
ADC	Analog-to-digital converter. 1
ASL	Autonmous Systems Lab. 1
ASVZ	Akademischer Sportverband Zürich. 1
AUV	Autonomous Underwater Vehicle. 1
CAD	Computer Aided Design. 1
DAC	Digital-to-analog converter. 1
DC	Direct Current. 1
EAP	Electroactive Polymer. 1
EMPA	Eidgenössische Materialprüfungs- und Forschungsanstalt. 1
ESD	Electrostatic discharge. 1
ETH	Eidgenösische Technische Hochschule. 1
FPGA	Field-programmable gate array. 1
IMU	Intertial Measurement Unit. 1
LEC	Laboratory for Energy Conversion. 1
LED	Light-emitting diode. 1
MOSFET	Metal-oxide-semiconductor field-effect transistor. 1
PCB	Printed Circuit Board. 1
PoC	Proof of Concept. 1

PWM Pulse-width modulation. 1

SMD Surface-mounted device. 1

SPI Serial Port Interface. 1

SVN Subversion. 1

USB Universal Serial Bus. 1

VIA Vertical Interconnect Access. 1

Glossary

φ angle of rotation around the X-axis. 1

$\vec{E}$ Electric Field. 1

$\vec{F}$ force. 1

ν Poisson number. 1

p_{el} electrostatic pressure. 1

ε_0 vacuum permittivity. 1

ε_r relative permittivity. 1

ADC Device measuring an analog voltage an converting it into a digital value. 1

DAC Device taking a digital value and converting it into an analog voltage. 1

PCB Mechanically supports and electrically connects electronic components using conductive tracks, pads and other features etched from copper sheets laminated onto a non-conductive substrate. 1

sepiOS Sepios Operating System, the software of the robot. 1

SMD An electronic part being directly soldered onto a PCB without through-hole pins. 1

Chapter 1

Introduction

This chapter addresses the origins and goals of the project. It also provides a short summary of the robot and the conventions needed to understand this report. Furthermore our team and its organization are introduced.

Contents

1.1 The Sepios Robot

This section will introduce the basic facts in order to provide the reader with a better grasp of our robot. It will also talk about a few conventions which are very useful for reading this report.

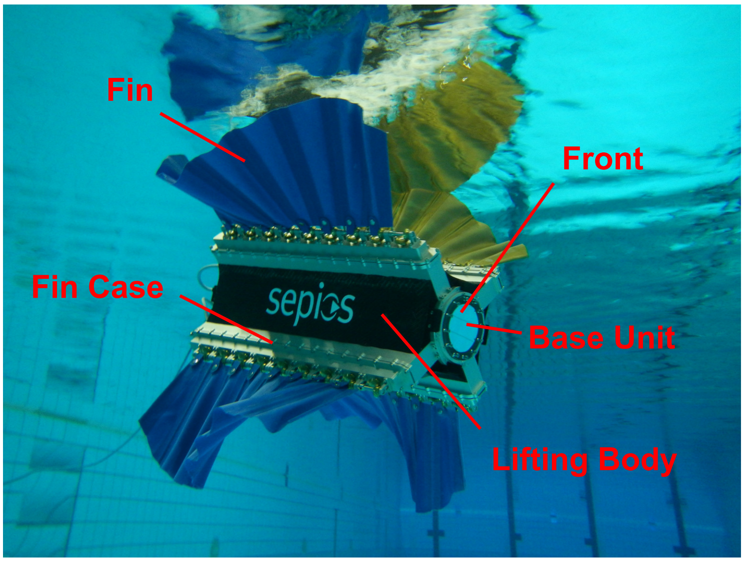


Figure 1.1: System Overview

The Sepios system consists of four so-called fin cases (Fig. 1.2a), four lifting bodies and a central base unit (Fig. 1.2b). The base unit has eight connection slots to attach fin cases and lifting bodies. It contains almost all of the electronics. Each fin case contains nine servo motors which actuate the attached fins through bevel gears.

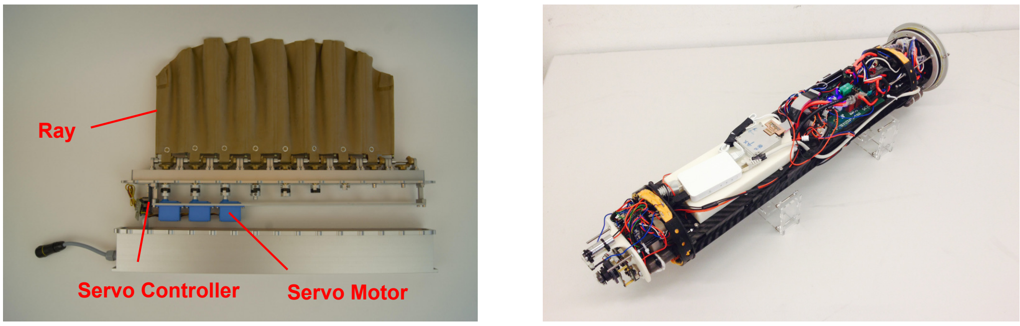


Figure 1.2: Fin case (left) and base unit (right)

1.1.1 Conventions

The front is defined as the flat face of the robot where no plugs are attached. Sepios is designed to be a modular system. The robot can easily be tuned to work with any configuration attainable with our fins. Unless stated otherwise the system is always assumed to be in the default configuration (Fig. 1.1) with all four fins mounted in a cross and the lifting bodies attached in the space between fins.

The default coordinate system (Fig. 1.3) used in this paper is a body fixed system attached to Sepios in its geometric center point, assumed to be concurrent with the center of mass S. This is the geometric center regarding only the basic components of the robot like the aluminum cases and main cylinder. The robot is assumed to be completely symmetrical and small parts causing differences in symmetry such as plugs are neglected. The X-axis points forward as the central axis of the base unit cylinder. The Z-axis points through the center of the downward fin and Y completes the right-handed system. The rotation angles around the X, Y and Z axes are referred to as ϕ, θ and ψ.

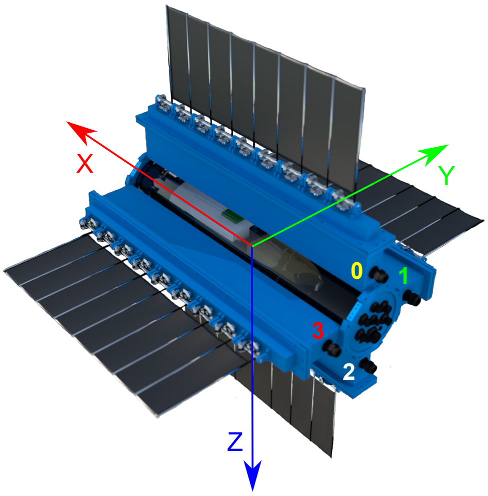


Figure 1.3: Default coordinates

Each of the nine sticks between which the fin is mounted is called a *ray*. A unique property which sets apart our robot from other cuttlefish designs is the ability to move each ray individually from $-135°$ to $135°$. Most of the waves used to generate propulsion are symmetric and oscillate around a central axis with a maximal amplitude of $30°$. This central axis can be shifted by an angle ζ referred to as *zero position* (Fig. 1.4).

The fins are numbered from zero to three, with zero being the topmost fin and the numbers continuing in positive mathematical rotation around the X-axis (Fig.1.5).

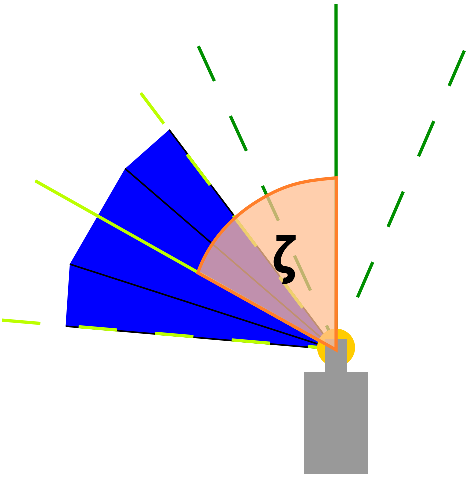


Figure 1.4: Zero position shifted

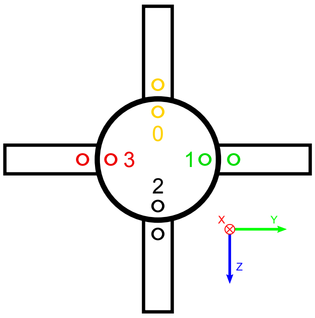


Figure 1.5: Examples of fin numbering as seen from the back

1.2 Focus Project

A Focus Project is part of the education of a bachelor student at ETH Zurich.

> “Students develop and build a product from A-Z! They work in teams, where they learn how to structure problems and identify solutions to them, analyse and simulate systems, as well as how to properly document and present a project. They build the product themselves, with access to a machine shop, and state of the art engineering tools such as Matlab, CAD, CAE, PDM systems.”
>
> ETH Zurich [2014]

We were supervised by Prof. Dr. Roland Yves Siegwart and two internal coaches named Gregory Hitz and Stefan Bertschi who all work at the Autonomous System Lab (ASL) of ETH Zürich. We also receive support from Andreas Schaffner who is a Master’s degree student and participant in the previous Focus Project Skye.

Sepios was brought to life by the idea of a colleague named Martin Möller. He was fascinated by the concept of building a robot utilising the cuttlefish’s unique propulsion mechanism. He proposed his idea along with a first hand drawing (Fig. 1.6) to interested professors. They immediately accepted the idea and added it to the varied selection of Focus Projects which people could join. It did not take long to assemble our group of eight highly motivated students.

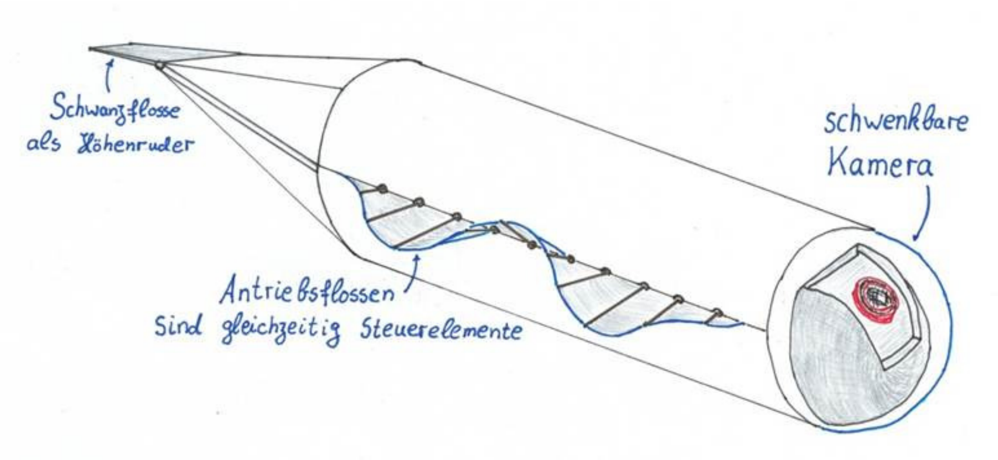


Figure 1.6: Martin's very first draft of Sepios

Project Sepios was unveiled to the public for the first time during our Roll-out presentation on the 27.05.14. There, we presented it in a swimming pool situated in front of the main ETH building. During the year the project was reviewed and graded in five individual presentations, some together with other Focus Projects. We also had to create an intermediate report, as well as this final report. For more information on our public events visit `http://sepios.org` and our Facebook page.

1.3 Motivation and Context

The name Sepios is derived from the Latin word "sepia" which is the scientific term for cuttlefish. These squid-related sea dwellers have a relatively long and streamlined body with ten tentacles encircling their mouth. They have two side-fins extending along their bodies starting above each eye and ending at the tail. By performing sinusoidal movements with their fins, water is pushed into a distinct direction. This technique, called "undulatory fin propulsion", allows the cuttlefish to move swiftly in every direction.

Figure 1.7: Cuttlefish fins as inspiration (Nova [2007])

The goal of Focus Project Sepios was to replicate this propulsion mechanism and to design and outfit a nautical robot with it. The main advantages over conventional propellers are the high flexibility and manoeuvrability. Swimming backwards is equally efficient as swimming forwards due to a symmetric body. Additionally it is possible to brake and instantly reverse the cruising direction and to drift sidewards. Also the risk of entangling in sea grass is lower than for a conventional propeller. In the future this could enable AUVs to inspect narrow environments like ship wrecks or flooded caves. Furthermore squid and cuttlefish are very elegant and almost soundless predators. This is interesting for marine wildlife filming as such a robot could approach an underwater animal without disturbing it. This way it could take spectacular pictures of marine fauna. Finally the elegance of the propulsion mechanism is can be very awe inspiring to people and is thus of potential interest for the entertainment industry.

1.3.1 The Naro Projects

The ETH already had many years of experience with underwater robotics prior to our project. One of the first bio inspired nautical robots was the “The Naro Nautical Robot” designed as part of another Focus Project in 2009. Their purpose was to mechanically recreate the propulsion mechanisms used by conventional fish (Fig. 1.8).

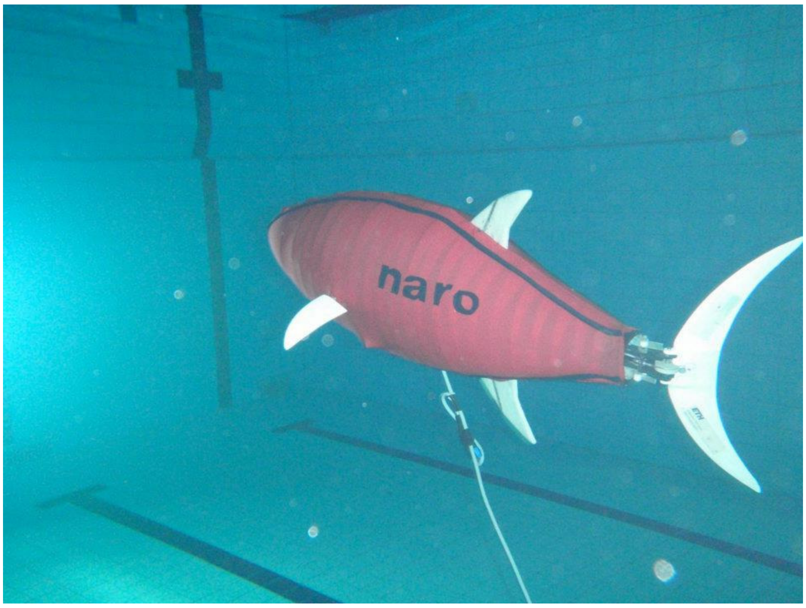


Figure 1.8: Naro Tuna (ETH [2014])

The follow-up project was Naro Tartaruga (Fig. 1.9) a project to replicate the propulsion mechanisms of a sea turtle. The robot weights 80 kg and can use its flapping fins to dive down to 150 m.

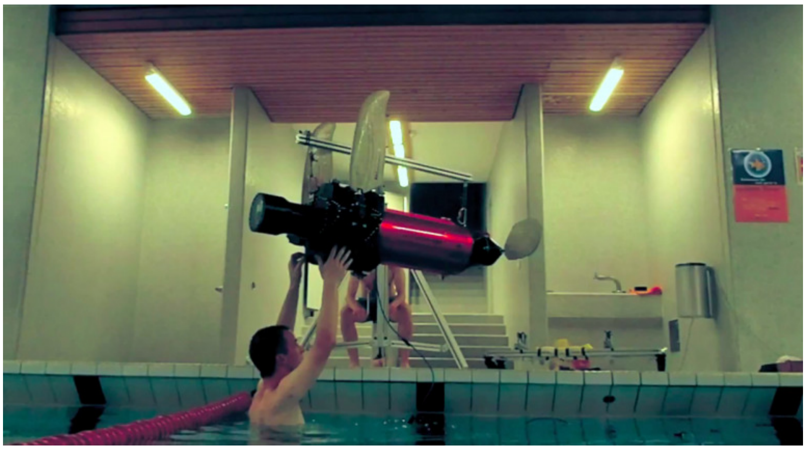


Figure 1.9: Naro Tartaruga (ETH [2014])

In the wake of these successful projects the ASL decided to build a new customizable educational entertainment robot, the “Nanins”. The robot consists of a central polycarbonate cylinder and three to four modular black motor boxes with fins (Fig. 1.10). At an educational event students were allowed to compete in a design contest, which goal it was to design an optimal fin for the robot. The students manufactured their fins themselves by laser cutter and then evaluated them in a race.

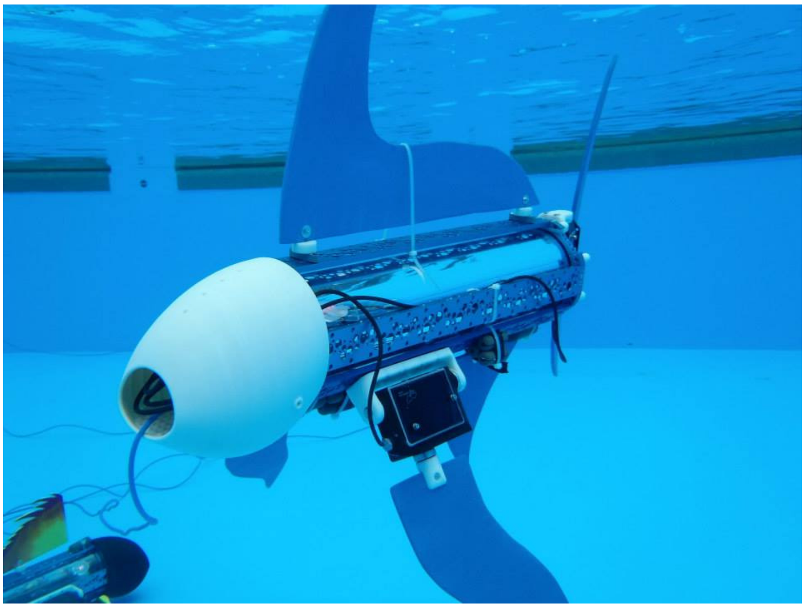


Figure 1.10: Naro Naníns (ETH [2014])

In contrary to the previous projects, Sepios is not part of the Naro chain of robots but a stand-alone robot. However, we were able to profit a lot from the experience other people had gained from the previous projects. This includes the formidable modular base unit design of the Naníns robot, which was adapted for Sepios.

1.4 Goal and Vision

After extensive debate and carefully evaluated feedback we chose the following to be our project goal:

> "Proof of omnidirectional locomotion capabilities of a nautical robot with sepia inspired fins."

Proof in this context means a proof of concept in the form of a robot actually swimming with undulating fins. The "omnidirectional locomotion capabilities" describe our demand that the robot should be able to move along all translational and rotational axes. This is further specified in the List of Requirements, section 2.4. It is important to mention that we wanted to focus on the fins themselves. A fully replicated sepia is not part of our goal. This is because we wanted to position Sepios primarily as a bionics research platform for the study of undulating fins.

Even though we designated our robot to be a research platform, we still wanted future iterations to have potential as an actual commercial product. We thus brainstormed a long list of potential applicators and used their common needs as a basis to come up with our goal's omnidirectionality clause. Some of the applications are listed in our vision, which in contrary to our goal merely presents a glimpse into the potential future of Sepios:

> “Project Sepios is aiming to design an efficient and environment friendly alternative to propellers for underwater locomotion. The use of undulating fins inspired by cuttlefish for thrust generation enables our robot to move omnidirectionally. High manoeuvrability and low disturbance allow for new applications in research, marine life filming, offshore engineering and more areas. Our team is looking to set a new trend for underwater vehicles and inspire future students to embrace engineering.”

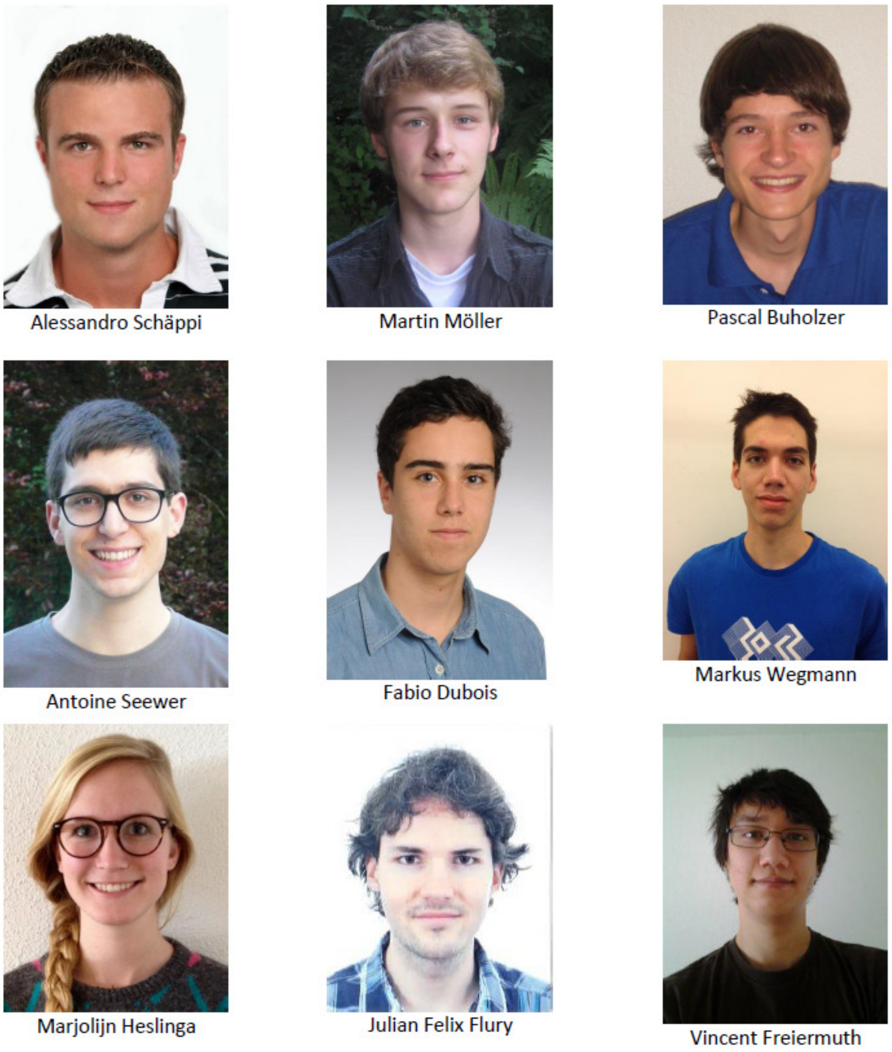
1.5 The Team

Our team consists of six Bacehlor’s students of the Mechanical (MAVT) and two of the Electrical Engineering (ITET) department of ETH Zürich, as well as one exchange student from the University of Delft in the Netherlands.

Every one of us brought their own strengths and abilities into the project. Some already knew everything concerning electronics while others already had plenty of experience with CAD. We tried to balance the team member’s jobs equally between things they were great at and new things they wanted to learn.

Figure 1.11: Team trip to Laax

To recuperate from the exhausting work we frequently organised team events. This started even before the semester (Fig. 1.11) and will surely continue into the future, the common past with the project binding us all together.



Alessandro Schäppi

Martin Möller

Pascal Buholzer

Antoine Seewer

Fabio Dubois

Markus Wegmann

Marjolijn Heslinga

Julian Felix Flury

Vincent Freiermuth

Figure 1.12: The team

1.6 Costs and Sponsoring

In the beginning it was very hard for us to estimate the total project cost. We started with our only known costs, the Naníns components, and gradually added cost blocks such as actuators, sensors and manufacturing. Hidden expenses quickly unveiled themselves as the project progressed.

Our budget was never clear-cut but a matter of negotiation. Rather than being limited to a certain amount, individual purchases had to be presented to the coaches for evaluation. They then green-lit the purchases if they deemed them reasonable.

Nevertheless, a budget had to exist to provide an overview of the spent money. We started with an estimated CHF cost of 10000 in the first review and in-

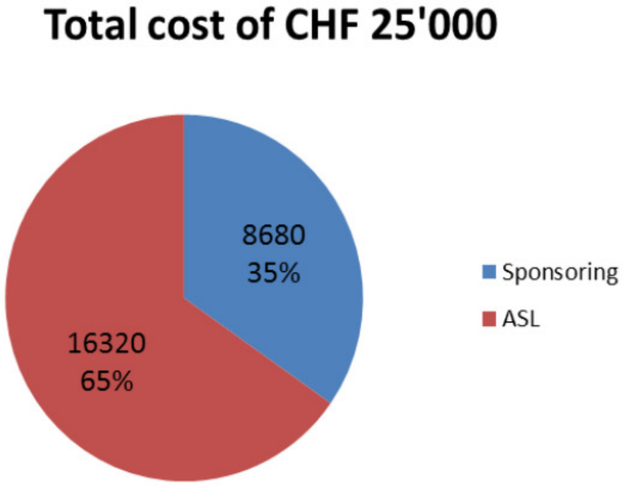


Figure 1.13: Total amount spent in CHF

creased it up to CHF 17500 for the second one. For a long time, this amount remained relatively stable, until it was increased to CHF 25000 towards the end of the project (Fig. 1.13).

The most expensive chunk consists of the 109 servos we bought. We ordered 45 servos in the beginning and bought another 10 after the first ones started breaking down because of overload. Being an university project we could arrange for a 30% discount. Once it became clear that the problems would continue and that the servos needed replacing, Hitec generously helped us choose the best alternatives and arranged for an even bigger discount of 50%. To summarise: we ordered 109 servos with a value of CHF 8442 but only paid CHF 4345.

The second most expensive block encompasses the manufacturing of our aluminium parts at the “Zentralwerkstatt Physik”. Even though it is subsidised and we only paid CHF 20 per work hour and used machine it still cost us CHF 4300. In an industrial workshop this would have cost an insurmountable amount of money.

CHF 25000 seems like a lot of money. However, it was possible to find sponsors to compensate for a large part of the expenses. Initially we contacted around 60 possible sponsors by impersonal e-mail. Only a small portion replied and almost no one was interested. We adjusted our strategies and directly contacted companies who could sponsor parts needed for the project. As recompense we offered to them to have their logo attached to our robot and website. This worked much better, especially if we visited them personally to explain our project in detail.

During the two semesters we raised over CHF 8 000 for our project. Most of

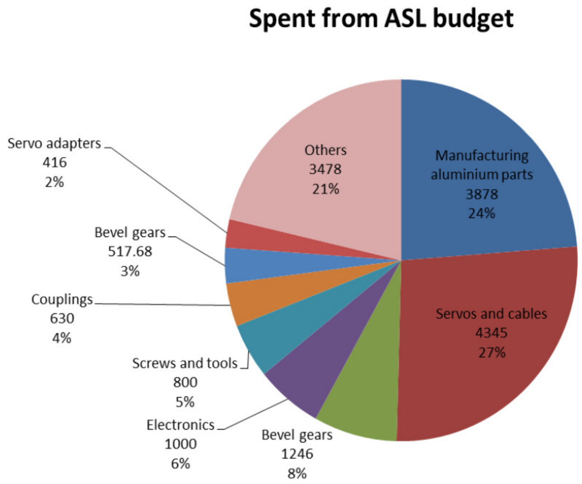


Figure 1.14: Spent amount in CHF of the ASL budget

it provided as parts or manufacturing costs. We were also offered countless hours of counselling which are not included in this budget. For these reasons calculating the exact amount of sponsored costs was not possible. A detailed list of our expenses can be found in Appendix B.

As the project's goal was to create a research platform and not a user product (as described in Figure 1.4), the developed robot is unique and many specialised parts had to be designed, which significantly increased cost. In theory building a second prototype for commercial purposes would be possible, but require severe redesign of certain parts to achieve a reasonable cost.

On the following page we list all our sponsors and summarize what they have done for us. Their logos were sent to us by mail with their agreement to use them. We would like to thank all our sponsors for their great support and fruitful feedback. Being able to profit from their expertise was truly monumental for this project.

Primary Sponsors

National Instruments Switzerland sponsors us one of their state of the art single-board computer and provides coaching.

KUBO is providing us with high quality custom made sealings for watertightness and coaching.

Swaytronic supports us with high-end LiPo batteries and charging systems.

Qualicut supports us with their knowledge in water cutting, produced some parts and anodised them as well.

3D Connexion provides us high-tech 3D navigation controllers to steer our robot in an omnidirectional way

Departement Physik
Zentrale Werkstatt

The Zentralwerkstatt of D-PHYS, ETH Zürich, supports us with their great expertise on manufacturing and nautical engineering.

Kami sponsored us with 18 printed team polo shirts.

Hitec sponsored us high end servos and we receive us a huge discount for the rest of them.

Further Sponsors

Alumex anodised all the 68 aluminium parts. This will help us having a long living and good looking robot.

Sponsored us with all our stainless steel screws and tools

Sponsored us a lot of our electronic parts

Offered us required tubes and connectors between our units

Suter Kunststoffe sponsors carbon fibre plates for our interior.

PUAG sponsored us with a high end and lightweight aluminium box to transport our robot.

Igus ponsored us all bearings for our fin case

Jakob offered us all couplings for our fin case

ABB funded us parts of our aluminium construction

Kundert sponsored us with different acrylic glass tubes and plate for our interior

Akademischer Sportverband Zürich

ASVZ supported us with diving equipment.

JKI sponsored us with licenses for their Tortoise SVN LabVIEW tool.

Tauchbasis Inauen supported us with diving equipment.

UTZ helps us store everything we need in comfortable boxes.

Teile offered us all shims greatly reduced

2Xideas sponsors all our expenses concerning our test phase in the sea of France.

1.7 Organisation

To provide an efficient working environment, it is crucial to have a well structured organization. This includes team, workload and time management.

1.7.1 Team-Management

As we started working on the project it immediately became evident that we needed to divide the team into subgroups.
A subgroup consists of two to four team members with special interests in a particular topic. Every subgroup has up to two leaders keeping the project supervisor up-to-date. This was usually done during one of the two weekly meetings.
During the autumn semester seven subgroups were established:

1. Fin Mechanics Inside: Development of fin actuation and fin case
2. Fin Mechanics Outside: Evaluation of fin surface and structure
3. Electronics and Interior: Development of power and communication distribution as well as interior assembly.
4. Control Systems & Modeling: Creation of a steering concept. The system-model is part of a Bacehlor's thesis.
5. Static Diving: Adapting the existing Nanins swim bladder to fit Sepios.
6. Outer Shell: Development of an outer shell to compensate for the lack of buoyancy.
7. Public Relations & Finances: Maintenance of an up to date internet and social media presence as well as acquirement and management of sponsors.

Martin was appointed as project supervisor. His job was to keep track of the subgroups' states, keep an overview over the entire project and distribute tasks as required. In collaboration with the individual team members he elaborated rough time schedules and monitored their execution. He was also the first contact for all official requests concerning the project.

As we mainly focused on the integration of all sub parts into one working system during spring semester, the team could not be divided as strictly anymore. However, we organized the task within the following main divisions:

1. Fin Cases and Fins: Assembly, testing and improvements.
2. Base Unit: Assembly, testing and redesign using carbon fiber material. Design and manufacturing of a buoyancy regulating shell.
3. Electronics: Design and testing of the connection board. Manufacture and test all connections and wiring.

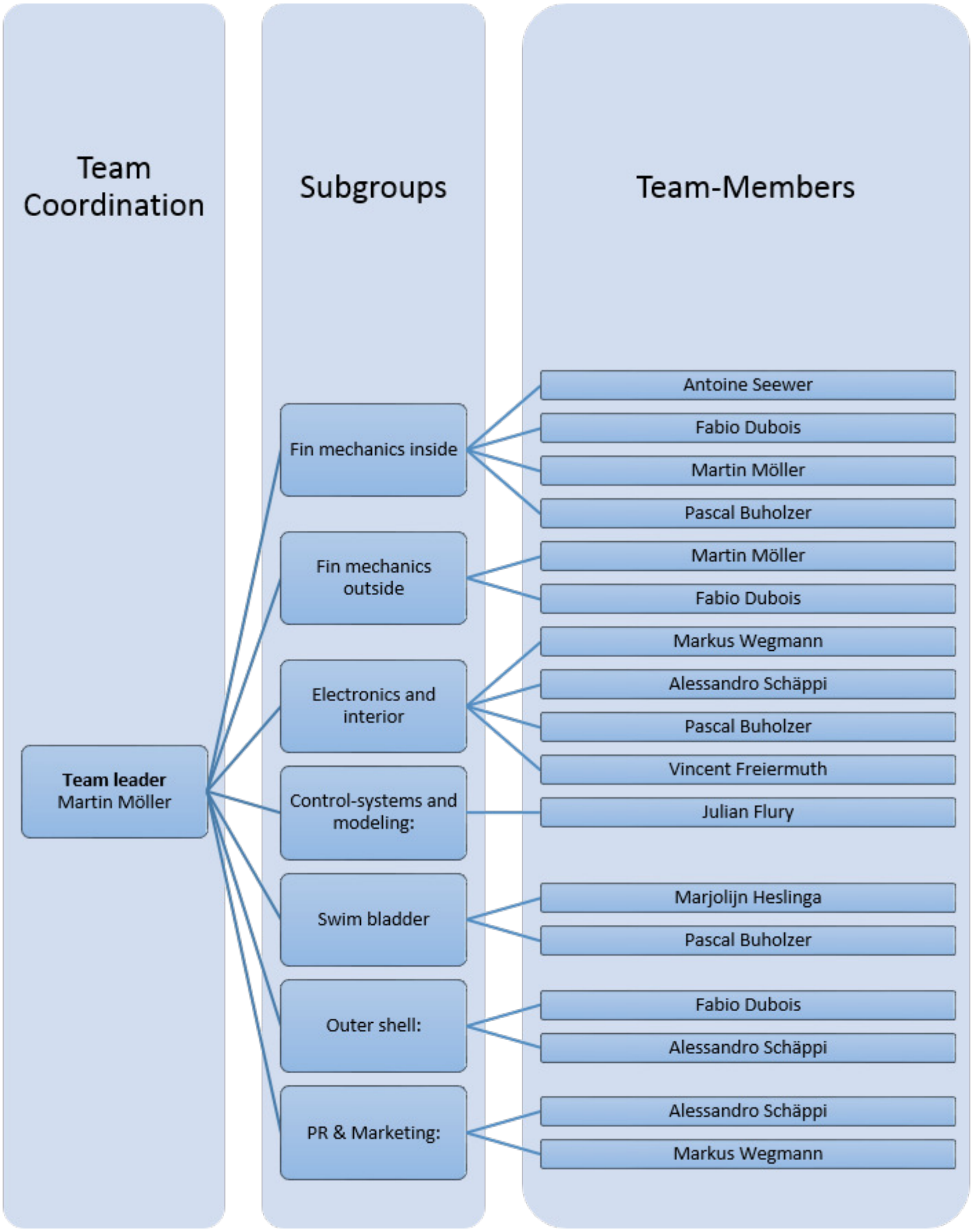


Figure 1.15: Organisation of team Sepios during the autumn semester

4. Main Software: Development and testing of the essential software required for minimal operations, including sensors management and a simple control allocation.
5. Add-on software by Bachelor Theses: Additional software parts include an attitude estimator and controller, a laser-based collision-detection and prevention, a more sophisticated control allocation as well as a framework of velocity sensors.
6. Coordination and Roll-Out preparations
7. Public Relations & Finances: Maintenance of an up-to-date internet and social media presence as well as acquirement and management of sponsors.

1.7.2 Time Management

To manage our time schedule we needed a tool that enabled us to easily keep track of the current project state. For several reasons described in Appendix G.3.2 we decided to use SmartSheet[1]. We used it to generate a list of tasks where every project member could insert their own and manage what had to be done.

A rough timetable was created early on. It was then fine tuned in collaboration with all the subgroups.
The five most important dates were the following:

- 04.11.2013: Milestone 1: Goal and vision fixed, first prototypes, search and evaluation of possible solutions created one single, fixed concept idea.
- 21.12.2013: Milestone 2: CAD model finished and drawings sent to workshops. All required hardware ordered and assembly organized.
- 21.03.2014: First successful dive with two fins in a pool. First data-recordings for evaluations.
- 25.04.2014: Milestone 3: First dive with all four fins conducted. Constant improvements on details.
- 27.05.2014: Milestone 4: Roll-Out presentation: Official presentation of the Sepios Robot to the public in Zurich.

A detailed project plan can be found in Appendix 8.2.

---

[1] SmartSheet is a cloud based project management solution

1.7.3 Infrastructure

Software

Mathematica 9.0 was used to perform dimensioning calculations. The computer aided design of the entire robotic system and numerous further components was done in *NX8.5* in combination with the server-based versioning system *TeamCenter10.1*. For the PCB design, including the customised servo shields and our connection board, the open-source software *KiCAD* was used. Furthermore *LabVIEW 13* from National Instruments, a high-level graphical programming software, combined with the SVN versioning add-on *TortoiseSVN* was used to program the robot. Some computations within *Matlab2014a* were included.

Website

Our website has existed since the beginning of our project. Thanks to Markus Wegmann a highly informative and visually appealing internet presence was designed. Check out *www.sepios.org*.

Next to being a public source of information concerning our project, the homepage was also geared towards attracting potential sponsors. It also contains an archive of all public mentions of Sepios.

Facilities

Our institute provided us with a large range of facilities. Our office was located in the CLA building, E17.2 (Fig. 1.16).

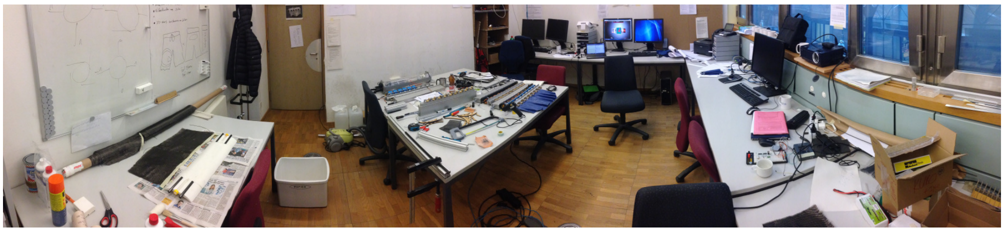


Figure 1.16: Team room with working stations and assembly table

For our extensive underwater testing we had the permission of the city of Zurich to use the Bungertwies swimming pool located at Hofstrasse 56.

In order to carry out the water tightness tests we had the opportunity to use the pressure chamber of the Composites Lab (LEC). In addition, the LEC allowed us to use their water towing tank to perform basic fin forces experiments. Last but not least the Mechanical Engineering department's own workshop helped us out countless times.

Chapter 2

Design Process

In this chapter we illustrate our path from the basic idea of building a nautical robot to our final concept. We started by studying the biological paragon and looked at already existing underwater robots. In parallel, experiments with undulating fin prototypes already gave us some first insights into this propulsion mechanism. We specified the requirements to our robot and used proven methods to arrive at our final solution.

Contents

2.1 Cuttlefish as Natural Archetypes

Our project is inspired by a very manoeuvrable little marine animal: the cuttlefish (Figure 2.1). Indeed, its two undulating lateral fins allow this sea dweller to change direction almost instantaneously. The combination of this and its ability to turn on the spot, allows the cuttlefish to navigate very narrow spaces.

Figure 2.1: The cuttlefish (Arbour [2013])

Amazed by this little-known creature, we did some research to improve our understanding of it and the way it generates thrust underwater. The following sub chapters summarize our biological studies and should help the reader to situate cuttlefish in the animal kingdom.

2.1.1 The Cuttlefish[1]

Cuttlefish are marine animals of the order Sepiida. They belong to the class Cephalopoda (meaning “head-feet” in Greek), which also includes squid, octopuses and nautiluses. 'Cuttle' is a reference to their unique internal shell, the cuttlebone. Despite their name, cuttlefish are not fish but molluscs.

They possess two lateral fins which run alongside their whole mantel length, eight arms, two tentacles with which they secure their prey and a funnel (also called siphon or simply water jet). Figure 2.2 illustrates, among others, those attributes in a body plan.

Cuttlefish range in size from 15 to 50 cm in mantle length (without tentacles) and weight up to a dozen kilograms. They inhabit tropical and temperate ocean waters and are mostly shallow-water animals, although they are known to go to depths of about 600 m.

They are sometimes referred to as the “chameleons of the sea” because of their remarkable capacity to alter their skin shape, color and pattern at will. They

---

[1] Adapted from Wikipedia [2014b]

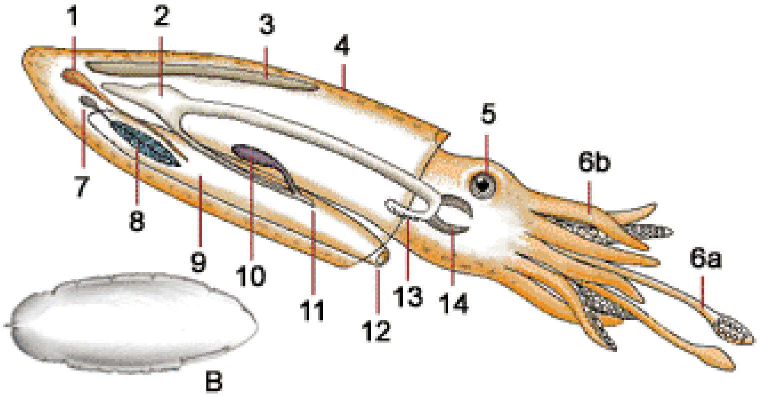


Figure 2.2: Body plan of a cuttlefish (Avon Lake Animal Clinic [2014]):
1: gonad; 2: stomach; 3 and B: internal shell (cuttlebone); 4: mantle; 5: eye; 6a: tentacles; 6b: arms; 7: heart; 8: kidney; 9: pallial cavity; 10: ink gland; 11: anus; 12: funnel (siphon); 13: radula; 14: beak.

use this incredible skill to communicate with other cuttlefish, to camouflage themselves, to scare off or momentarily distract potential predators, or to confuse their prey. At this point it is worth mentioning that Cuttlefish, although color-blind, are able through some mechanism which is not yet understood to match the color, contrast and texture of their surroundings, even in complete darkness.

A last interesting side note is the fact that the porous cuttlefish bone provides it with buoyancy, which it regulates by changing the gas-to-liquid ratio in the chambered cuttlebone. Just like the swim bladder in the Sepios' base unit (discuss in Chapter 3.2.2)!

The Latin word for this fascinating animal, *sepia*, serves as a base for our project's name.

2.1.2 Squid vs. Cuttlefish

Cuttlefish are often confused with one of their closely related cousins: the squid. Even though they share a lot of characteristics, those two cephalopods do not belong to the same order (Sepiida for the cuttlefish, Teuthida for the squid).

The main biological distinction can be found in their internal shell. Whereas squid only have an unclassified internal support, cuttlefish possess a hard bony plate, the cuttlebone. Usually, as can be seen in Figures 2.3 and 2.4, squid's bodies are much more slender than the ones of cuttlefish, their eyes are set more to the side of the head and their fins do not follow their entire mantle. However there exist so many variations between species of cephalopods that those last differences are not always reliable.

Figure 2.3: Cuttlefish
(Tier im Fokus [2014])

Figure 2.4: Squid
(Ryan Photographic [2014])

2.1.3 Locomotion Mechanisms

The French book Traité de Zoologie on cephalopods by Grassé [1997] gave us more information on those fascinating sea creatures and the way they move underwater. This scientific text provides us with knowledge about the muscles of the cuttlefish and its three different locomotion modes, namely the stationary phase, the slow swimming mode and the fast stroke.

In the stationary phase, the cuttlefish just makes small rectification movements with its fins, using them mostly to stabilise. Quite surprising is the fact that some species are too heavy to stay in place and have to produce a constant upward movement. In the slow motion mode, which is the most used locomotion principle, the cuttlefish uses both its fins and water jet for the production of thrust. In most species of cuttlefish, the muscle in charge of the jet is really well developed, but there are species which rely almost solely on their big fins. When executing very fast movements known as fast strokes, the cuttlefish wraps its side fins around its body to be as hydrodynamic as possible and uses its water jet to produce strong and quick leaps. Some species are able to reach really impressive speeds in the range of 40 to 50 km/h. This is used to attack prey and to escape from predators.

2.2 Existing Finned Underwater Robots

Of course we had a look at existing underwater robots, particularly the finned kind, at the very beginning of the project. As expected, we quickly verified that research about undulating fin propulsion was very sparse. The only advanced cuttlefish-like robot we found is the one designed at the Osaka University in Japan over the course of ten years (Fig. 2.5). Their so-called “squid robot” is about 1.4 m in length and possesses two undulating side fins and two tail fins. Its lateral fins, which have a width of 75 mm, are actuated by 17 servomotors each.

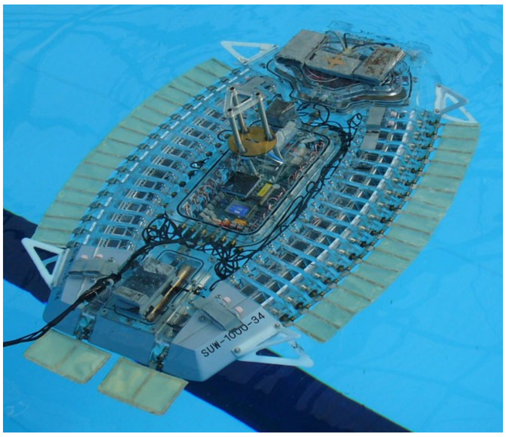


Figure 2.5: The squid robot of the Osaka University, Robot Watch [2006]

We also contacted Mr. Mahbubar Rahman, who took part in this project at the Department of Naval Architecture and Ocean Engineering in Osaka. He guaranteed us access to some really interesting papers, for example Toda et al. [2009] and Rahman et al. [2011]. The second publication is a computational study on such fins. This enabled them to establish simple relationships between the principal parameters of the fin. According to the study, thrust increases with the amplitude squared and cubic to the aspect ratio (width/length). It also rises with frequency but eventually drops again after a peak is reached. This can be explained by the increasing frictional resistance due to the higher frequency, which decreases the overall efficiency (Blasius' solution). We were able to verify some of these trends with our LEGO-Prototype (Chapter 2.3.2).

Besides the squid robot of Osaka, some other universities have built some cuttlefish-like fin prototypes. One of them is the Northwestern University in the USA, which is quite advanced in its research about undulating fins. Unlike us, they took as inspiration a small fish living in the Amazon river: the South American Black Ghost Knifefish. This fish possesses a single undulating fin along its belly. It presents astonishing manoeuvrability and is capable of moving rapidly in any direction. Figure 2.6 shows this fascinating animal and its technical counterpart.

North Western's Prof. MacIver recommended to us the articles of Epstein et al. [2006], Shirgaonkar et al. [2008], Curet et al. [2011], Sefati et al. [2012] and Ruiz-Torres et al. [2013]. They contained models and experiment results concerning the variation of frequency, amplitude and wavelength of the travelling wave and their impact on fin thrust production. They also proved that such a mechanical fin was able to produce steady forward thrust, even with a relatively small number of rays (8 in that experiment).

In addition to those publications we were also granted access to a very interesting Bachelor's thesis done at the Autonomous System Lab (ASL), Peter et al.

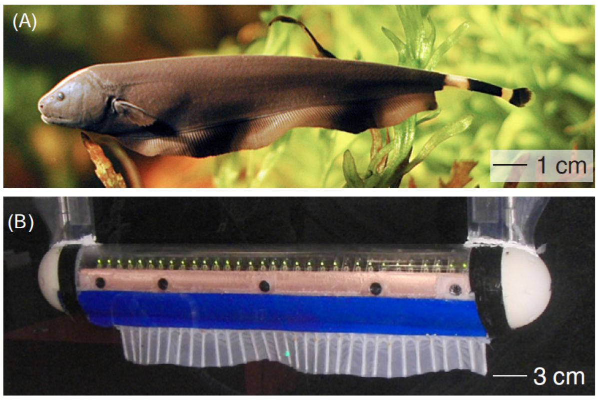


Figure 2.6: South American black ghost knifefish (A) and its robotic counterpart from the Northwestern University (B), Curet et al. [2011]

[2010]. For their thesis, B. Peter and R. Ratnaweera designed a mechanical cuttlefish fin relying on the principle of camshafts to perform the undulating motion. They were able to perform a series of experiments in the water towing tank of the Laboratory of Energy Conversion (LEC).

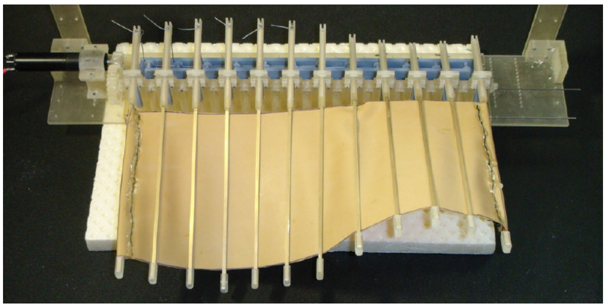


Figure 2.7: The ”CuttleFin” of Peter et al. [2010]

2.3 Early prototypes

In order to verify and further investigate the concepts of generating forward propulsion with undulating motions, we started creating prototypes very early in the project.

2.3.1 Well-Prototype

The very first functional prototype was primarily built to estimate the maximal stress on such a fin. This information was needed to perform experiments on the feasibility of EAPs as actuators (Chap. 2.8.1). The prototype was very rudimentary. Six wooden rays were attached to a central pole with holes on one side to mount a spring scale. The rays were then covered with a dismantled plastic bag.

Figure 2.8: the Well-Prototype

We fixed the construct on a well, using two heavy stones. Then we proceeded to manually imitate the wave patterns of sine and standing waves. The stress on the middle lever was repeatedly measured by pulling it with a spring scale instead of directly. Next to plastic bag foil, Lycra and PET-foil were also evaluated.

The constant distance between the outer left ray and the outer right ray was called the "length" of the fin. The distance from the pole to the tips of the rays was named "width" of the fin (see Figure 2.9). To find out whether the size of the canvas affected the strain we also varied the width. Each value represents the average of five measurements taken, which were usually off by about $0.3cm$.

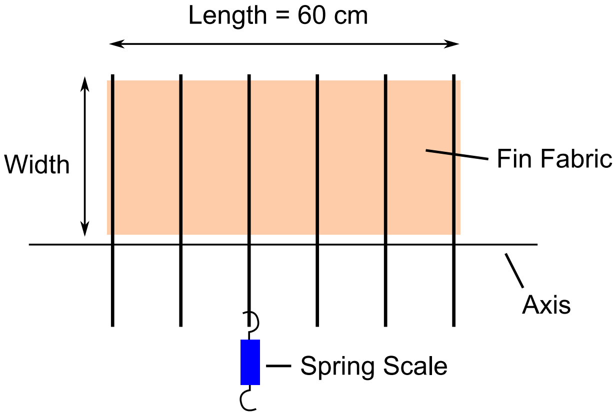


Figure 2.9: Experiment set-up

Unfortunately, because of the simple nature of the experiment, the results (shown in Table 2.1) are not very reliable. What is certain, is that the stress is significantly higher for the PET foil fabric.

More valuable than the actual results were the intuitive insights gained during the experiment. We observed some very interesting vortex structures propagating along the foil while performing a sine wave. Table 2.1 shows that actuating a PET foil takes much more effort but also seemed to result in much stronger hydrodynamic forces. This would make sense as a PET foil is much stiffer than one made of Lycra or with a plastic bag and thus likely to translate more energy directly into the water. This comes with the price of less flexibility which makes standing waves much harder to perform.

Material	**Plastic Bag (long)**	**Plastic Bag (short)**	**Lycra**	**PET Foil**
Width [m]	0.3	0.2	0.3	0.2
Length [m]	0.6	0.6	0.6	0.6
Weight on Feather Scales [kg]	1.1	0.5	0.833	2.2
Resulting Torque [Nm]	1.294	0.5886	0.981	2.589

Table 2.1: Measurements with the Well-Prototype

2.3.2 LEGO-Prototype

Another prototype built in the first weeks of the semester was the LEGO-prototype. The main goal of this was to verify whether such a propulsion mechanism would even work, how fast and agile it would be and to get some intuition about this innovative way of generating thrust in water. Being much closer to what we expected our final concept to be, it was also used to further our understanding of different fin materials. It consisted of a LEGO built body with nine step motor driven rays, controlled by three NXT bricks. These were then mounted on a raft made of wood and polystyrene.

Figure 2.10: Face down view of the LEGO-Prototype

We wrote the software for the prototype in LabVIEW and were amazed by the simplicity of creating a working program. The final version of the program was already able to mimic any possible periodic function with an output between -1 and 1. It also enabled us to adjust parameters such as amplitude, frequency, phase shift and zero position in real-time. This allowed us to observe various interesting phenomena. The software was used as a base for our primary prototypes' software.

Initial tests in the pool of the University of Zürich proved surprising speed and agility that we did not expect from a mere LEGO construct. Precise measurements were not possible, mainly due to the currents present in the pool, but a relative estimation of propulsion force for different fin materials seemed possible. Six different fins were constructed and compared. Thrust was measured by attaching a spring scale to the back of the robot and holding it in place while having it generate forward thrust through sinusoidal motion. The results are listed in Table 2.2:

Material	**Lycra (Loose)**	**Lycra (Tight)**	**Lycra (Short)**	**Latex**	**Spandex**	**Coat Fabric**
Width [cm]	16	16	9	16	16	16
Spacing [cm]	8	6	8	8	8	8
Thrust [g]	95	70	45	95	90	65

Table 2.2: Comparison between different fin materials

It is immediately evident, that Lycra, latex and spandex are able to convey the most motor force into actual forward thrust and are thus the best suited as fin material. Spandex has the disadvantage of not being water repellent and thus soaking over time. This leads to a gradual decrease in forward thrust generated. This left latex and Lycra as likely candidates for further investigation (see section 3.1.2).

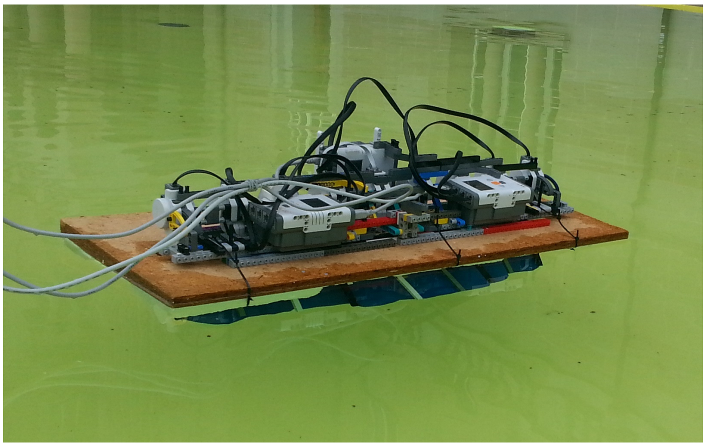


Figure 2.11: The LEGO-prototype in action

Other interesting observations corroborate what we had already read in some papers, most notably in Rahman et al. [2011] from the Osaka University in Japan. This publication states that thrust increases squared to the amplitude and cubic to the aspect ratio (width/length). As stated in the paper, there seems to exist a peak frequency, above which the efficiency of the system decreases again.

We also verified that sidewards drift by performing a standing wave is indeed possible and that leaving the last rays angled to perform curved motion works but severely limits thrust. The most force seems to act on the ray leading the wave, as those broke off the most. One last fascinating observation we made: the fabric, if allowed to move freely between rays, moves very clearly in one direction when a sine wave is applied!

Overall, this prototype was a very important step towards our final solution, verifying many concepts and providing us with a starting point for the selection of fin materials.

2.4 List of Requirements

Parallel to the experiments on our first prototypes we set up a list of requirements. These were collected during several brainstorming sessions and chosen in a way that would guarantee the achievement of our goal and the long term functionality of our robot.

In Table 2.3 we present a selection of important requirements arranged in the three categories “Locomotion”, “Design” and “Feedback”. The complete list of all 52 requirements can be found in Appendix 8.2. As not all the requirements were crucial for the functioning of our robot we separated them into the two categories “obligations” and “wishes”. Furthermore we assigned priorities to them from one (low) to four (high). This helped us decide in case of two conflicting requirements.

Upon completion of the robot, we conducted a series of experiments in order to verify the fulfilment of our requirements. These are explained in detail in Chapter 7. Let us discuss the causes for some of the more interesting requirements:

- We chose the maximum weight of our robot so that at most two people would be needed to carry it. For this we set a value of 20 kg which we exceeded slightly with a weight of 22.7 kg. Our initial goal of two people being able to carry the robot was nonetheless achieved. The more limiting factor is the size of the fins. Ensuring they do not collide with walls can be tedious.
- At 100 $Mbit/s$ our data transmission rate was sufficiently high. In our

list of requirements we had set a minimum of 20 $Mbit/s$.

- Due to time restrictions no experiments on stream resistance, noise generation or entangling in sea grass have been performed.
- We dived along the bottom of Lake Zürich without raising any sand.

Category	Requirement	Value	Description
Locomotion	Omnidirectional	-	Direction of rotation and translation in all three spatial axes of the drone
	Acceleration time	4s	Time in which the system from the rest position reaches the usual traveling speed
	Deceleration time	2s	Time in which the system decelerates from the top speed to zero
	Cruising speed	0,5 m/s	Maximum speed of the drone in standing water
	drift velocity	0,1 m/s	Maximum speed perpendicular to the main direction
	Vertical diving speed	0,1 m/s	vertical diving when horizontally aligned (no translation in other directions)
	Angular rate roll axis	20 grad/s	Angular velocity , the system rotates about its rotation axis
	Angular rate pitch-axis	15 grad/s	Angular velocity , the system rotates about its rotation axis
	Angular rate yaw-axis	10 grad/s	Angular velocity , the system is rotated about its vertical axis
Design	Max weight	20 kg	Up to two people necessary to carry the robot
	Intrinsically stable in the stream position	-	Center of mass slightly below the buoyancy point for stabilization in the normal position
	Depth use	5 m	The robot is designed for 10 m to guarantee a security factor of 2. This corresponds to 2 bar of absolute pressure (1 bar relative to atmosphere)
	Operating time at full load	25 min	At maximum propuslive force in the main direction of movement
Feedback	Depth measurement	< 10 cm	Pressure Sensor - > current depth . Deviation of less than 10cm
	Attitude measurement	< 20°/h	Speed at which the estimation error grows

Table 2.3: List of requirements

2.5 Brainstorming

Generating brilliant ideas can be hard at times. Even when tasked with finding the solution to a specific problem, creativity cannot be called upon at will. In order to help this process, creativity methods are often applied. We decided to use them to support our solution finding process.

Figure 2.12: Brainstorming post-it wall

With the goal and vision set in stone (see Chapter 1.4) we immediately set out gather all our ideas. We started by refreshing ourselves with a funny and stimulating group game. Then, equipped with lots of pens and post-its, every team member tried to come up with as many ideas and thoughts as possible. The post-its were assembled on the wall representing the whole space of possible solutions. From now on any idea or posted comment was actively discussed and challenged. Often associative ideas and inputs arose leading to the further development of an idea. By not focusing on the quality or on the feasibility of the ideas this part of the concept phase was intended to be as divergent and innovative as possible. Following the credo of thinking outside of the box we came up with many different ideas, some brilliant, some wacky, some simple and some impossible.

2.6 Functional overview

A top-down hierarchy of all the functions of the robot was set up. This helped us get a firmer grasp on what exactly the system would have to do. For further insight, see Appendix F.

2.7 Morphological box

Our intensive brainstorming brought up heaps of ideas for various functions. One of the hardest processes during the first quarter of the project turned out to be selecting the right ideas and uniting them into a whole concept.

A commonly used approach to process a lot of ideas is the morphological box, as was introduced in the "Innovation-Process" lecture during the first semester. A table is created, with every row symbolizing different sub functions of the system. The ideas are then sorted into the different rows according to their function. When all ideas are gathered, lines can be drawn to combine different ideas into whole solutions. Each line drawn represents a unique system concept. The ideas are not yet rated. Shown in Figure 2.13 is an excerpt of our morphological box. The full morphological box is attached as Appendix F.
To speed up this process and generate more concepts, we split the team into three groups of three people. Each of these groups elaborated up to three complete concepts of their choosing and presented them to the team. In a follow-up meeting the team selected the four most promising conceps:

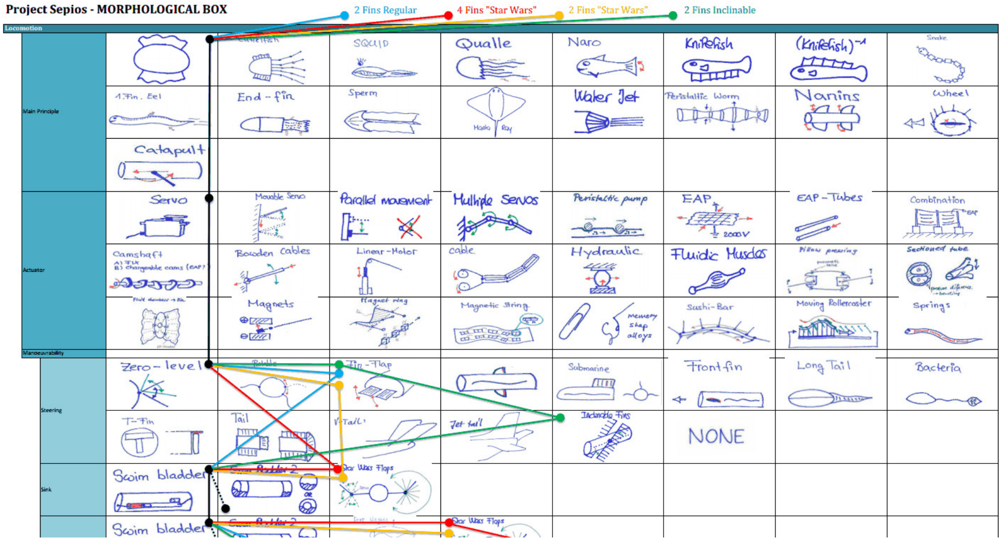


Figure 2.13: Extract of the morphological box

1. 2 Fins Regular (Figure 2.14)
2. 2 Fins "Star Wars" (Figure 2.15)

3. 4 Fins ”Star Wars” (Figure 2.16)
4. 2 Fins Inclinable (Figure 2.17)

These four concepts were then investigated more closely (see Section 2.8). Simultaneously different criteria were raised to compare the concepts in a comprehensive way (see Section 2.9).

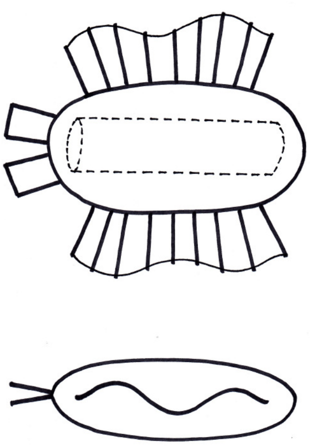


Figure 2.14: 2 Fins Regular

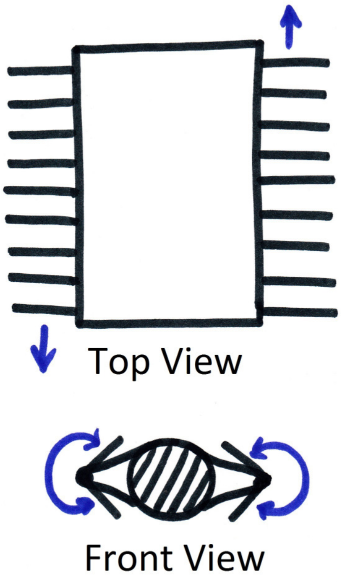


Figure 2.15: 2 Fins Star Wars

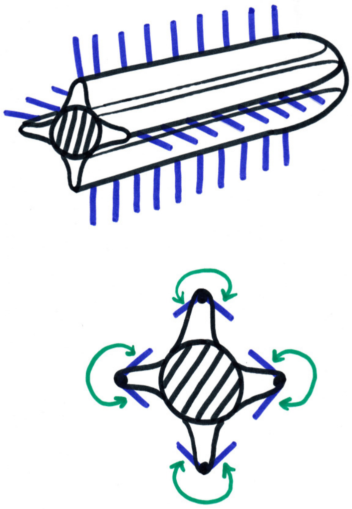


Figure 2.16: 4 Fins Star Wars

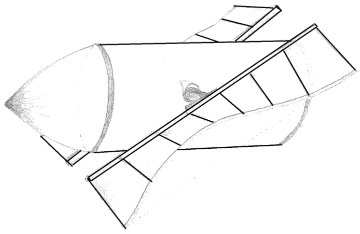


Figure 2.17: 2 Fins Inclinable

2.8 Evaluated concepts

The following sections detail the procedure which led up to our final concept. We will discuss the choice of actuators and fin configuration. The range of manoeuvres which each of these concepts allows played an important roll and is presented separately. In Chapter 2.10 our final concept is explained in detail.

For large parts of our morphological box our preferred concepts did not differ. The reason for this is that some variants had already been discussed in detail before we started combining the concepts. The possible solutions are, however, included in the morphological box for completeness' sake. This includes the main principle and the actuators.

The choice of actuators was made before we evaluated the morphological box. This is the reason why all our concepts are based on servo motors. In Chapter 2.8.1 our argumentation can be found in detail.

2.8.1 Choice of actuators

The choice of actuation concept was very pivotal for the project and was by far the most well-discussed topic during our concept process. We considered many different kinds of actuators, including bionic actuators such as Electroactive Polymers and Fluidic Muscles. In the end, we decided to pursue the classic electromechanical approach with servo motors. This because of their reliability and good performance. The following section details the reasoning behind this choice.

Electroactive Polymers

The fins of cuttlefish contain a very complex compound of muscles and organic fibers. This structure enables them to actuate small segments of their fins separately which provides them with a remarkable flexibility. As a first approach we were looking for actuators suited for imitating the natural fins. As those do not contain any fish bones we were looking for an actuation mechanism that does not require the integration of rigid bodies into the fin. Very soon the so called electroactive polymers (EAP) caught our attention. These are elastomers which experience mechanical stress in distinct directions when voltage is applied to them. They therefore also carry the name “artificial muscle”. Even though the research field is comparably young we started to investigate whether the use of EAPs might be feasible for our project.

EAP actuators are often produced in the form of thin membranes. In a physical sense they can be understood as plate capacitors (see Figure 2.18). The front- and backside of such a membrane constitute the electrodes of the capacitor. When voltage is applied to the electrodes the molecular structure inside the material is modified causing mechanical stress and the contraction of the actuator.

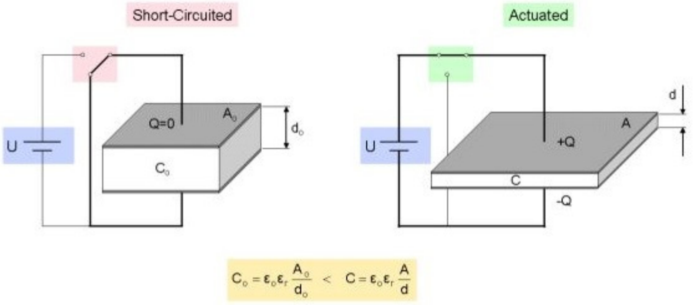


Figure 2.18: EAP as a capacitor [EMPA Switzerland, 2013]

This stress is also called electrostatic pressure and is found to be $p_{el} = \varepsilon_0 \varepsilon_r \|\vec{E}\|^2$ where $\vec{E}$ is the local electric field. The main stress vector inside the material is parallel to the electric field so that the membrane thickness decreases. As the Poisson-number of polymers can be well approximated with $\nu \approx 0,5$ (incompressible material) the electrode's surfaces tend to become larger in order to conserve the membrane's volume. Overall the membrane flattens out and relaxes.

For the use in our fins we mainly considered two EAP-configurations: an agonist-antagonist-setting and stacked-actuator-muscles. Both configurations cannot completely abstain from using some elastic skeleton-structure to hold the actuators in their positions. We also considered different elastomers. The most frequently used active polymers are conventional dielectric EAPs, e.g. VHB 4910. They are purchasable and allow the application of significant forces. Their disadvantage is the very high electric voltage that is needed for activation. Typical values needed for our purpose would be around 5 kV. While searching for isolator materials we were suggested the silicone compound “Elastosil RT 745 S” produced by the company Wacker. This material might have been suited for such high voltages while being elastic enough so that it would not limit the fins' flexibility. However, the risks that come with such high voltages are significant. The so-called ionic EAPs require much lower voltages. Unfortunately they do not provide enough force for our application. The third type of elastomer was a silicone film covered with a corrugated silver layer. The corrugated layer allows the material to actively stretch only into one distinct direction. This is convenient as no energy is lost with the undesired stretching of the material perpendicular to the main direction. However, as shown by Jordi et al. [2011] the performance of corrugated silver is still inferior to the dielectric polymer VHB 4910.

For our evaluation we received great support by the Swiss Federal Laboratories for Materials Science and Technology (EMPA) in Dübendorf. Senior scientist Silvain Michel from the group of Dr. Gabor Kovacs provided us with literature and gave us a lot od advice concerning EAPs. Their had already had some experience with the use of EAPs in a bionics project. Over the course of three years they had built an EAP-driven blimp.

The Agonist-Antagonist-Configuration In this setting one elastic plate is installed perpendicular to the robots body in order to form the fish bones of the fin. One pre-stretched EAP membrane is placed on either side of the plate. Additionally small ridges between the EAP and the elastic plate hold the membranes in their positions. There is no electrical contact between the two membranes. Figure 2.19 shows a freehand sketch of this composition.

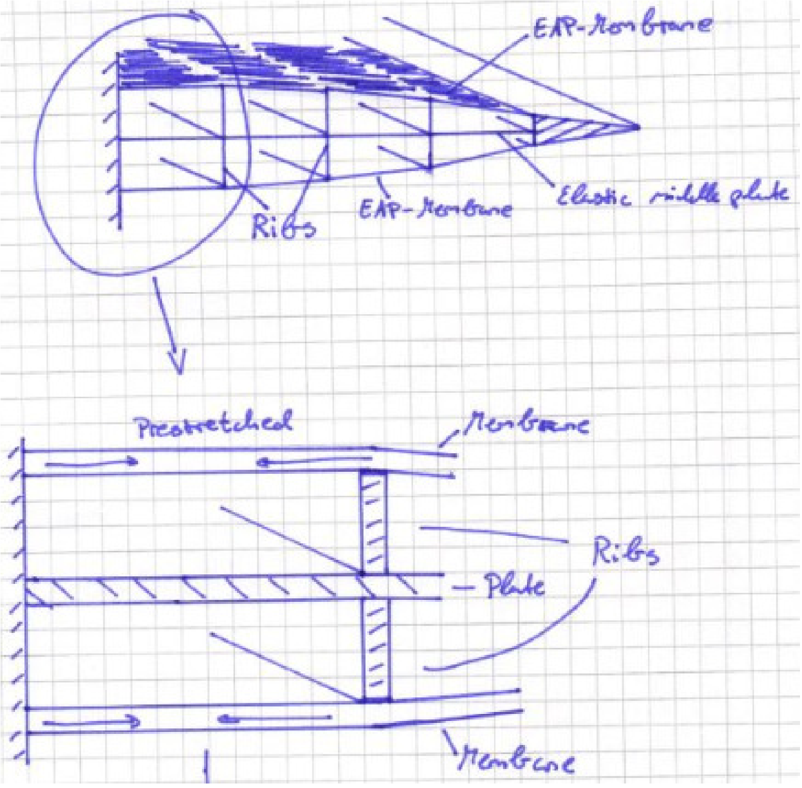


Figure 2.19: Draft of the agonist-antagonist configuration

The electrodes can be instantiated using graphite-powder which is placed on the membrane's surface and connected to the voltage supply. This has the advantage of only needing two membranes per fin. At the same time, the number of the single fin segments can be made very high by using many small stripes of graphite powder. The only limit is the number of connectors as each segment needs to be addressed and supplied with voltage individually.

When voltage is applied to one segment of the agonist membrane its thickness decreases while its overall surface area increases. As both membranes are pre-stretched the agonist membrane relaxes. Simultaneously the counterpart membrane on the backside of the elastic plate contracts due to the pre-stretch. As the agonist membrane relaxes while the antagonist membrane contracts the overall fin segment experiences a moment of torque. Thus the segment turns into the direction of the antagonist membrane. This principle is shown in figure 2.20.

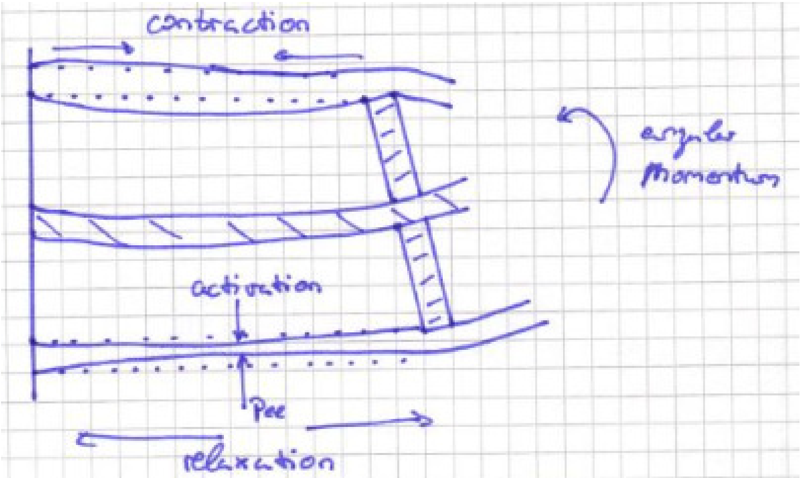


Figure 2.20: Torque generation with the agonist-antagonist configuration

This effect can be augmented by gluing several identical membranes on top of each other. The physical analogy to this is the serial connection of multiple capacitors. They all experience the same voltage per membrane and therefore relax consistently. Furthermore, as they are equally pre-stretched the resultant torques add up. As shown in Jordi et al. [2010] the torque can be scaled proportional to the number of membranes in series. By switching the voltage between the two sides it is now possible to generate an oscillating torque. For that purpose half of the voltage on one membrane can be reused for charging its counterpart. Basically the overall charge in one fin segment oscillates constantly between the opposing membranes. Coordinating the torques of all the single segments would allow the fin to perform a specified movement.

We evaluated the feasibility of the agonist-antagonist-configuration by calculating the number of membranes connected in series that would be needed to apply the necessary torque for a typical swimming mode. Our calculation (see Appendix G.3.1) showed that at least 50 EAP-layers would be necessary. During construction we would have had to consider the short lifetime of a single EAP-membrane and the fact that an electric breakthrough in only one membrane would cause the entire fin to fail. Also the production of such a large amount of layers could take over a week. Because of this the agonist-antagonist-configuration was deemed not to be a well suited actuator for our purposes.

The Stacked-Actuator-Configuration The second setting that we considered is based on a different mechanical concept. In the stacked-actuator-configuration a series of EAP-layers is again glued together in series. In contrary to the flat and thin agonist-antagonist-actuators we now have a longish cylindrical configuration. To form this shape a large number of small circular EAP-layers is cut out and glued on top of each other. Now between the two tips of the cylinder voltage is applied. Therefore its diameter increases and the actuator contracts along the main axis of the cylinder.

Stacked actuators often have a geometrical appearance similar to natural muscles. They are however also not produced industrially. Fabricating only one working actuator would take a long time and would cause a lot of waste.

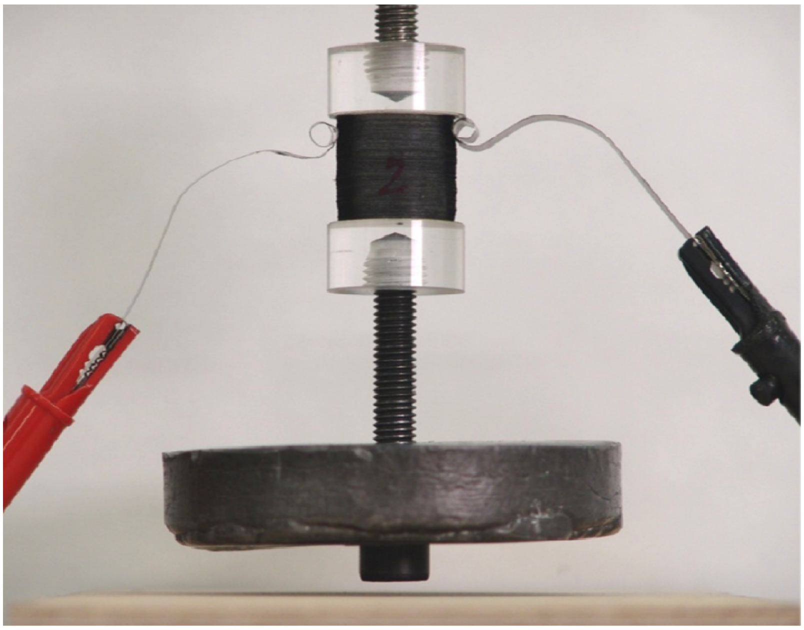


Figure 2.21: A stacked EAP actuator [Kovacs et al., 2008]

One advantage of this configuration is that the actuators do not need to be prestretched. For a fin of 10 cm width however we found the required number of EAP layers to be around 1000 (see Appendix G.3.2). In experiments at EMPA the layer thickness was chosen slightly larger. Still there were 400 layers needed which caused the production time for one actuator to be more than two months. This showed that also stacked-actuator-muscles are not feasible for the use in our project. Considering these shortcomings, we decided against using EAPs in Sepios.

Fluidic Muscles

Another kind of exotic actuators, the fluidic muscles, were also considered as possible candidates for the robots propulsion mechanism.

A fluidic muscle consists of a flexible hose made from rubber and embedded aramid fibres. When inflated with pressurized air or clean water, the fluidic muscle expands in diameter and at the same time contracts longitudinally.

Figure 2.22: Fluidic muscles of the company Festo in inactive and contracted mode (Ivo Boblan [2014])

The advantage of such a hybrid-pneumatic actuator is its extremely high power to weight ratio (up to 400 : 1). This is the primary reason why they are ever more frequently used in projects aiming to produce artificial muscle frameworks or complete bio-mimicking robots. Furthermore, the control of such systems, consisting mainly of valves, ducts and reservoirs is straightforward and reliable.

On the other hand, leakage problems can lead to catastrophic corrosion when combined with sensitive mechanical or electronic components. The pressure regulating components such as valves and filtering parts are heavy, large and costly. This, combined with the diving time limitations imposed by requiring pressurized air made it a very unattractive option.

Electromechanical actuators

Already having dismissed all the fancy bionic actuators, we focused on conventional electromechanical actuators. Typically these actuators only provide one degree of freedom. Therefore all fin designs in our morphological box that appeared in this context were based on the same principle. The majority of the fin surface would be provided by an elastic foil. This foil would be attached to a finite number of rigid rays. These rays would be rotatable around one common axis where the fin would be attached to the robot's main body. By rotating these rays single fin segments could be actuated.

One Actuator per Fin In evaluating all feasible solutions, we started with the most straightforward and thus seemingly most affordable one: a single actuator per fin. This would require a mechanism able to convey the torque of a single electric motor to multiple rays.

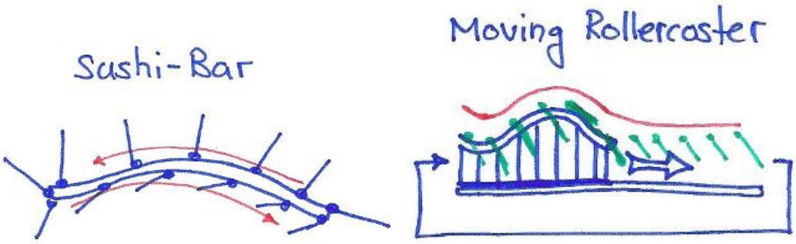


Figure 2.23: Two exotic concepts for one actuator per fin

Initial actuation concepts were based on the idea of guiding the rays on a predefined rail. A board could be produced with a wave-shaped guiding hole. By mounting the rays through said hole, bearing them appropriately and moving either the board or the rays together they would perform the desired undulating movement. Sketches of two possible implementations of this are shown in Figure 2.23. Unfortunately water sealing a moving mechanism like this seemed like a nightmare and dissuaded us from further pursuing these concepts.

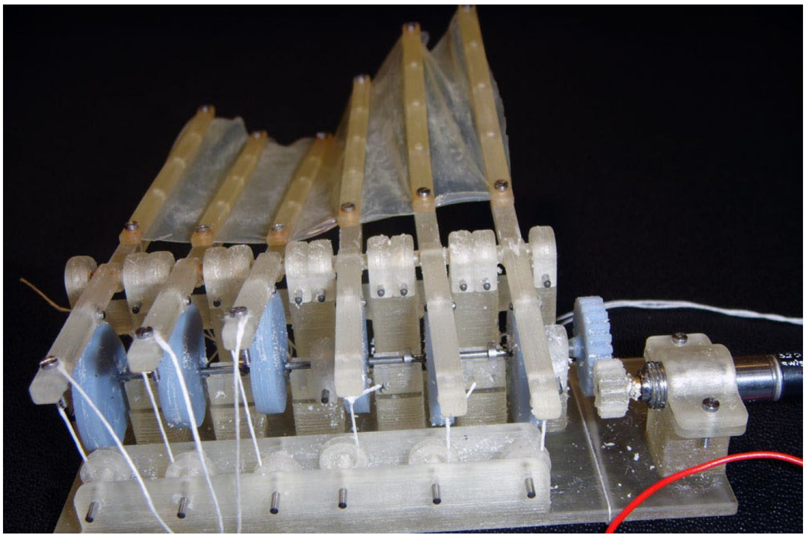


Figure 2.24: Cam shaft solution used by [Peter et al., 2010]

The second and perhaps more obvious approach was to use an ordinary cam shaft. This had already been done in Peter et al. [2010] (see Figure 2.24). This solution has some advantages: Heat losses are minimized as they only occur in one actuator and xpenses on electronics are considerably lower. However, production of a sufficiently robust cam shaft can be very expensive. The main problem of this actuation principle is that it simply does not fulfill our goal (see Chapter 1.4). Our goal is to achieve omnidirectionality and to use the robot as a research platform. Both of which rely heavily on flexibility which a camshaft solution simply cannot provide. Overly complex solutions involving camshaft revolvers or multiple additional steering surfaces might be conceivable but would undermine the very simple elegance distinguishing this solutions from the others. We regarded this restriction as a crucial handicap and thus abandoned single actuator ideas.

Servomotors Another very obvious approach was to actuate each ray individually. These solutions seemed to offer themselves because of the great flexibility they provided. This would not be restricted to a finite set of fin movements but could follow an infinitely large set of mathematical functions. This would be useful for optimizing the shapes of waves used for propulsion. Also non-periodic functions could be interesting for certain maneuvers as, for instance, large flaps produce a lot of thrust instantaneously. Additionally, the cost for a single actuator would be comparably low, allowing the immediate replacement of broken down hardware.

Multiple actuator candidates existed, including magnets and linear motors, but in the end were all deemed inferior to servomotors. Servos seemed particularly great because they have built-in position control. This greatly simplifies programming. Furthermore servomotors are rather compact, which would allow us to keep the robot at a reasonable scale.

The high flexibility and comparable simplicity to construct lead to our decision of using an individual actuated servo concept as main actuation.

Increased Flexibility One disadvantage of rays is their stiffness. Cuttlefish are able to bend their fins similar to the wings of a bird. In order to imitate this movement we were looking for mechanisms which would allow us to actively bend the fins along the rays.

This could be conducted using more than one servo per ray. The limited space however did not allow for this solution. A more promising approach was to bend flexible rays using bowden cables. The rays would then consist out of many small segments. This geometry would allow for a very flexible fin. Sealing would be reasonably easy, as the cable could be attached entirely in the water. Figure 2.25 illustrates these two concepts.

Finally, we also considered alternative ways of fixing the foil onto the rays. An interesting approach was to design the rays with a long slit through which he foil would be pulled. It would then be able to slip through the ray freely. This concept would prevent the fin from overstretching. The foil would deform so that the pressure is uniformly distributed and no segments would be unused and hang loose. This concept was used by Peter et al. [2010].

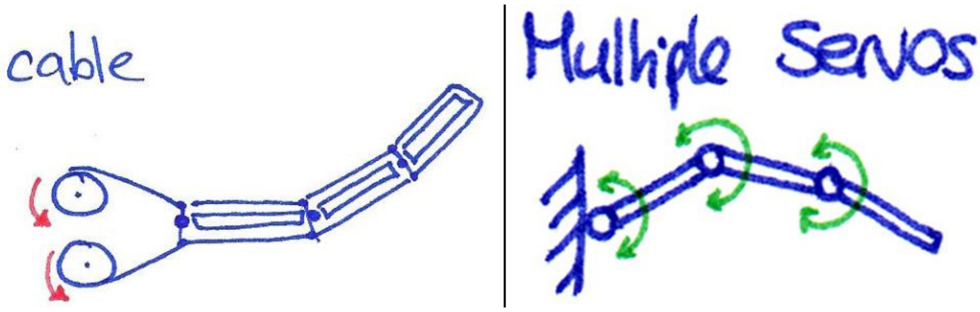


Figure 2.25: Concepts for actively bending the rays

We tested this concept on our LEGO-prototype and received very discouraging results. After giving it some thought, it seemed logical. It is exactly the pressure on the foil that generates thrust. If the foil is able to escape this pressure this results in less energy being converted into motion.

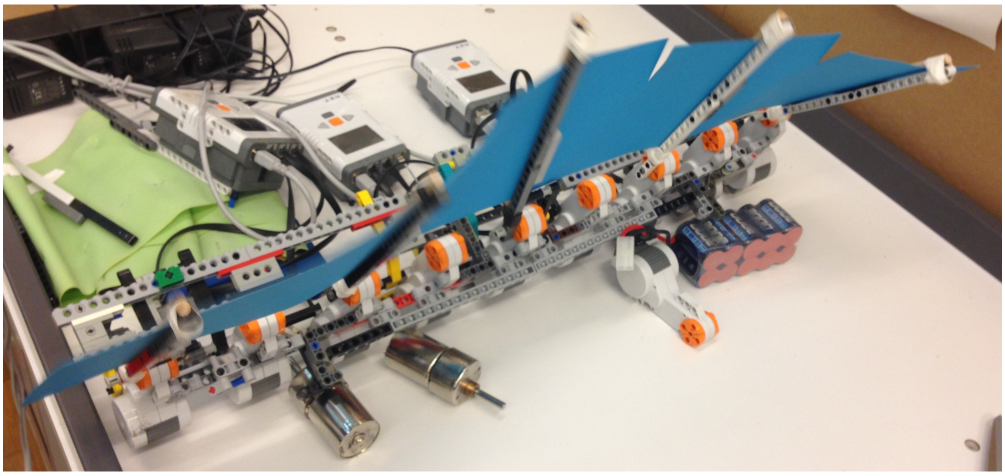


Figure 2.26: Stiff foil which can slip through the rays

In the end we decided that additional efforts to increase the fin flexibility were not necessary for our purpose and that we should rather focus on a simple system and implement that as perfectly as possible.

2.8.2 Steering mechanism

Most existing underwater robots use some kind of extra steering mechanism next to their main propulsion. Normally, this is achieved by adding secondary steering surfaces. This chapter describes the discussion of whether extra steering surfaces were needed.

No extra steering fin

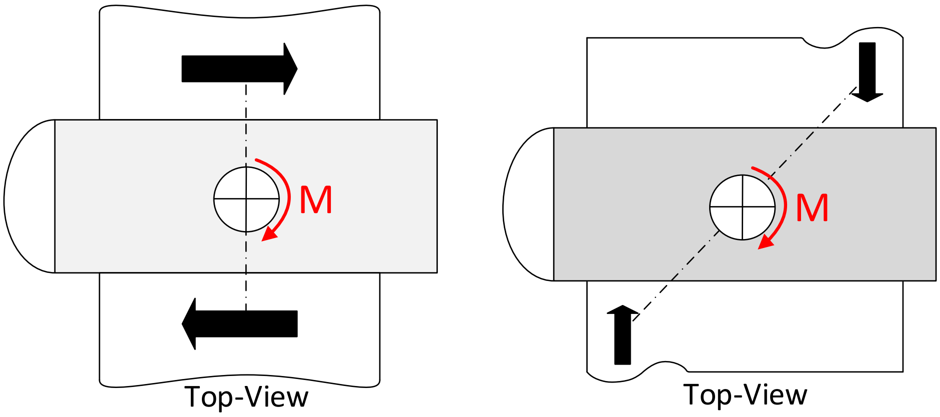
The first concept uses the undulating fins only to control the attitude. This could be achieved by the following control strategies, also illustrated in Fig. 2.27:

- **Two counter-working fins mounted in parallel:** Setting inverse wave speeds on each fin will induce a torque around the axis perpendicular to the two parallel fins. Each fin-pair can create a torque independent of any other movements along their respective axis. The rotation point will always be in the center of the distance the parallel fins are set apart. The wider this distance, the bigger the torque. This concept allows for instantaneous on-spot rotations.

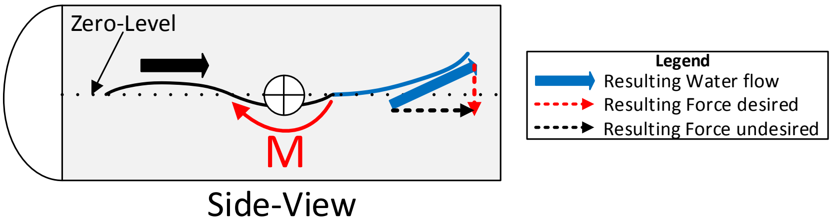
- **Standing waves in only one region of the fin:** For example by generating a standing wave only in the front area of a fin, a torque could be induced. This concept also allows for instant and on-spot rotations. However, only a small subarea of the fin can be used to generate the forces. This results in a very weak torque.

- **Varying zero position along one fin:** By a varying the wave's zero position along one fin, in theory, pitch can be controlled. However, tests with our LEGO-Prototype (see Chapter 2.3.2), showed, that this strategy also results in much weaker forward forces, as the wave is "broken". This also cannot be done without moving in X direction.
  For example: To generate a pitching torque, the zero position (see section 1.1.1 for more explanation about zero position) of the last few rays could be set to a level above zero (indicated in blue in figure 2.27c). The flow induced force will then generate a force lowering the back part of the robot and thus generating a torque on the Y-axis.

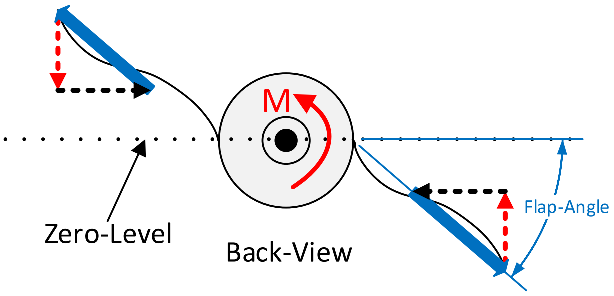
- **Flaps: Turning the whole fin into a new zero position:** By changing not only zero positions of single rays but that of the entire fin into a new position and then performing a standing wave there, a constant torque around an axis perpendicular to the fin can be generated. This is very useful for rolling.



(a) Counter working fins (b) Standing waves in one region of the fin



(c) Varying zero point along one fin



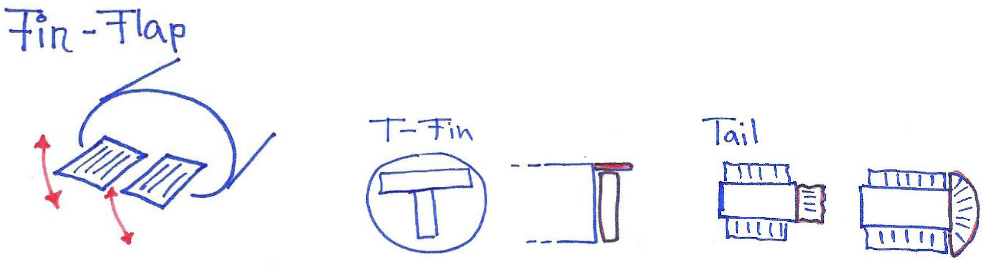
(d) Flap Principle

Figure 2.27: Steering concepts with no extra steering fin

Extra steering fin

Three concepts of additional steering fins were promising:

- **Fin-Flap:** Two separate, stiff fins mounted either in front of, or at the edge of the robot to stabilize pitch and roll-movements.
- **T-Fin:** Two separate fins mounted at the edge of the robot inspired by the elevator and rudder of aeroplanes. This would stabilize pitch and yaw
- **Steering fin allround:** An additional undulating fin going around the front- and rear-edges of the robot. By setting several rays of this fins to new zero positions, different rotations would be possible.



(a) Fin-Flap (b) T-Fin (c) Steering fin allround

Figure 2.28: Steering concepts with extra steering fin

The big advantage of these control surfaces is the independence from main thrust generating fins. However, all of these concepts involve passively flowed against control surfaces. That means that they would only work while the robot is in motion. As static rotations were a necessary requirement and zero position shift seemed to be a promising concept, we decided to discontinue concepts with separate steering fins. Furthermore, the effort of building another, entirely new, piece of mechanical hardware would be prevented.

2.8.3 Fin arrangements

One of the core decisions which had to be taken was what number of fins and what their alignment would have to be to fulfil all of the requirements (see Chapter 2.4).

This section introduces the four most promising concepts found trough the Morphological Box (see Chapter 2.7) and their expected characteristics.

For the expected force directions mentioned in this chapter we assumed the superposition principle for the forces created by our fins would work in general terms, neglecting linearity. As shown in section 7.1 this assumption could be proven during evaluation.

Two Fins Regular

As introduced in Chapter 2.2, most existing cuttlefish robots use two undulating side fins exactly as their natural archetypes do. These robots deflect their fins within a range of approximately ±30° [2] and are therefore unable to shift their zero position. This restricts the ability create forces perpendicular to the fins. These fins limited to ±30° are referred to as *"regular"* within this report.

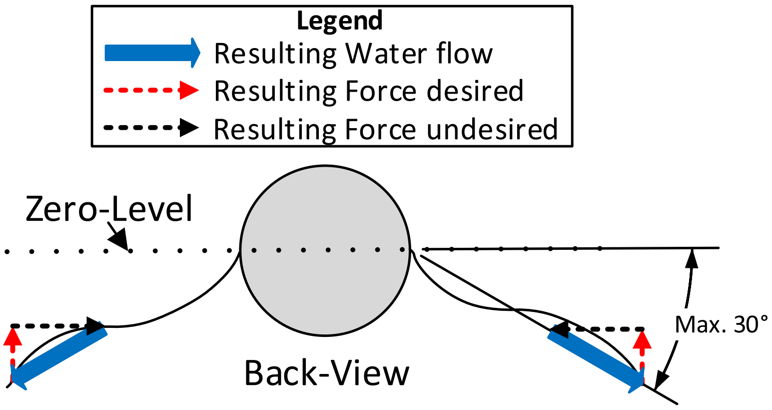


Figure 2.29: Two Fins Regular

Two Fins "Star Wars"

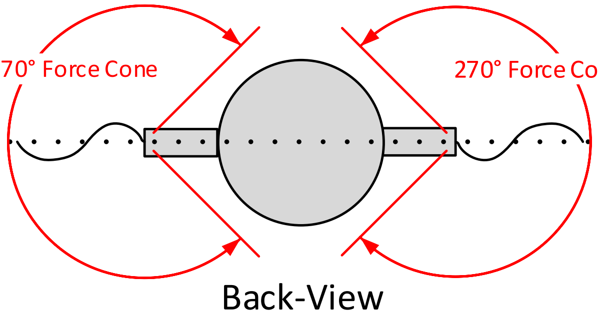
In order to create a truly omnidirectional robot, we developed the idea of fins with a maximal deflection angle of ±135°. Such a fin would be able to generate forces within a ±270° cone. We refer to these ±270° deflectable fins as the *"Star Wars"* configuration [3].

Mounting two such fins on either side of the robot would allow for much higher thrust in vertical directions and possible force generation in any desired direction (see Chapter 2.8.2). However, we expected the large changes of the zero position to be difficult to control. Furthermore, to avoid a collision of the fins with the base unit of the robot, the turning point of the rays has to be distanced by a few centimetres, requiring an additional construction (the gray box in Figure 2.30c).

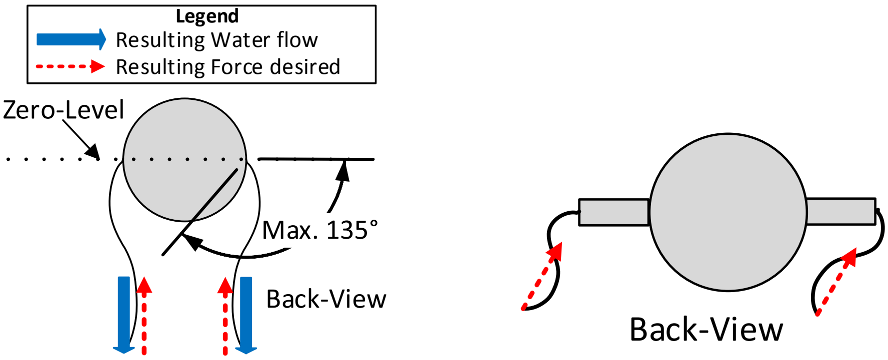
---

[2] This angle was estimated with a video-analysis on existing robots - see section 2.2

[3] The incredible transformation processes of Rebel spaceships at the back of our minds



(a) 270° force cone



(b) Increased Lift with ±135° deflection

(c) Possibility of a sloped ascent

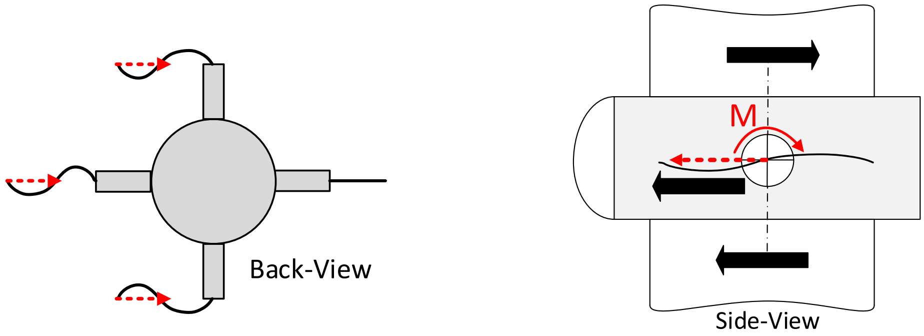
Figure 2.30: Two Fins “Star Wars”

Four Fins “Star Wars”

As mentioned in Chapter 2.8.2 we found that pitch control by varying the zero position along the fin’s length was not satisfyingly possible. Hence we developed a concept with a more direct control of the pitch axis. Mounting four fins aligned in a cross would enable us to control pitch and yaw simultaneously and independently (illustrated in Figure 2.31b). Furthermore we assumed the overall available propulsion force as well as the force transition area to the water would be almost doubled. This would allow for faster and more precise manoeuvres.

To allow for sideways movements with the strongest thrust possible (see figure 2.31a), we decided to add our ±270° deflectable “Star Wars” concept as described in the paragraphs above.

On the other hand, manufacturing and maintaining four instead of only two fins would increase the overall cost of our propulsion mechanism significantly. To minimize this increase and allow for maximal modularity we decided that all four fins would have to be identical.



(a) Side drift with three standing waves

(b) Combined, decoupled pitch and forward movement

Figure 2.31: Four Fins ”Star Wars”

Two Fins Inclinable

Another approach to generate pitching motion was to rotate the two side fins around a central axis. This concept would at least allow for simple pitch correction of the robot's body when stationary. Furthermore, only two fins would be required reducing costs and weight.

However, without varying the zero position, the pitch angle of the fins themselves could hardly be controlled. Furthermore, the rotation point would require a lot of engineering compared to the concepts with fins directly attached to the base unit. Having the entire propulsion mechanism be attached at a single, rotatable axis would also greatly increase the frailty of the whole system.

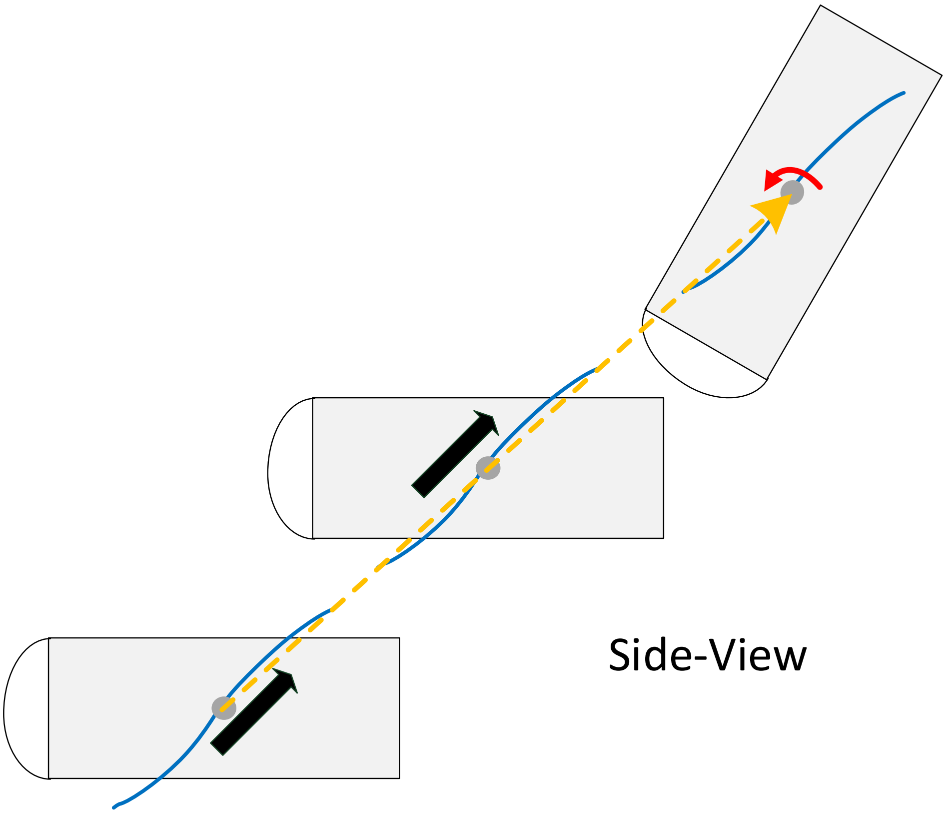
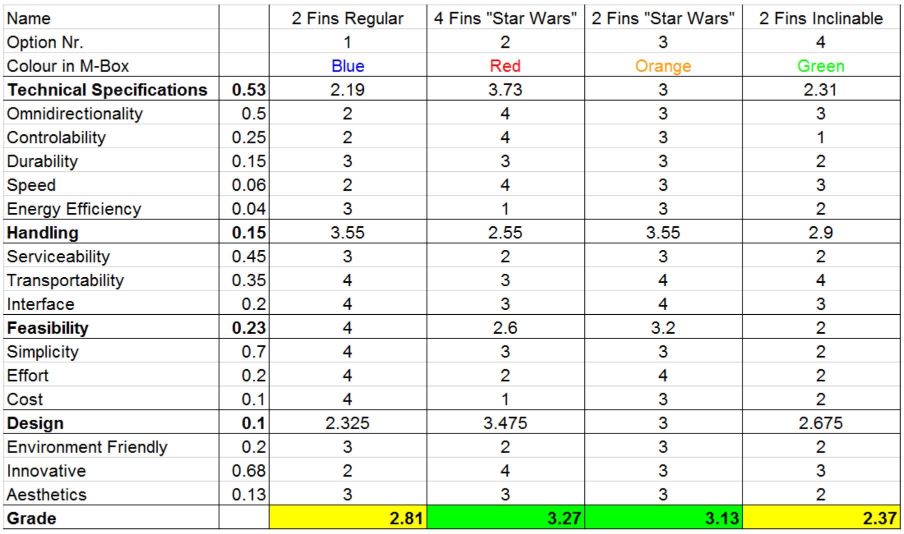


Figure 2.32: Possible maneuver with two inclinable fins

2.9 Criteria analysis

To be able to better compare our four solutions, we we developed a number of criteria. We divided them into four categories:

1. **Technical Specifications:** As we are primarily a research project this includes the most important criteria. The most important criterion of all, omnidirectionality, is a direct representation of the project goal (see Chapter 1.4). Note that the least important criterion is energy efficiency. This stems from our teams desire to create something brand new instead of optimising something existing towards perfection.
2. **Feasibility:** The limited resources had to be taken into account.
3. **Handling:** all the omnidirectionality would be in vain, if weren't possible to steer the robot accordingly.
4. **Design:** While we wanted to innovate the world of submarine robotics, outer appearance was not deemed important. Regardless, the resulting prototype is rather pretty.



Name		2 Fins Regular	4 Fins "Star Wars"	2 Fins "Star Wars"	2 Fins Inclinable
Option Nr.		1	2	3	4
Colour in M-Box		Blue	Red	Orange	Green
Technical Specifications	**0.53**	2.19	3.73	3	2.31
Omnidirectionality	0.5	2	4	3	3
Controlability	0.25	2	4	3	1
Durability	0.15	3	3	3	2
Speed	0.06	2	4	3	3
Energy Efficiency	0.04	3	1	3	2
Handling	**0.15**	3.55	2.55	3.55	2.9
Serviceability	0.45	3	2	3	2
Transportability	0.35	4	3	4	4
Interface	0.2	4	3	4	3
Feasibility	**0.23**	4	2.6	3.2	2
Simplicity	0.7	4	3	3	2
Effort	0.2	4	2	4	2
Cost	0.1	4	1	3	2
Design	**0.1**	2.325	3.475	3	2.675
Environment Friendly	0.2	3	2	3	2
Innovative	0.68	2	4	3	3
Aesthetics	0.13	3	3	3	2
Grade		**2.81**	**3.27**	**3.13**	**2.37**

Figure 2.33: Our criteria analysis table

With all the criteria in place, we graded our favourite solutions accordingly. As can be seen in Figure 2.33, the ”Four Fins Star-Wars” solution, described in-detail in Chapter 2.8.3 prevailed. This is direct result of the added omnidirectionality of using four fins. We also thought that the controllability might be improved for this solution, as more basic movements can be decoupled.

2.10 Final Concept

We chose “Four Fins “Star Wars”” (see Chap. 2.8.3) as our final concept.

A base unit derived from the Nanins (see Chap. 1.3.1) would form the centrepiece of the robot. 65 *cm* of polycarbonate tube would hold together metal rings, onto which the fins would be attached. The interior was to be made out of laser cut polycarbonate.

Aluminium cases would enclose the nine servo motors per fin. The fin surface was to be an elastic foil of Latex attached to rigid carbon rays. These rays would be rotatable around one common axis. Bevel gears were to be used to convey power from the high-end servos to the rays. To provide additional buoyancy, polystyrene lifting bodies were supposed to be fastened between the fins.

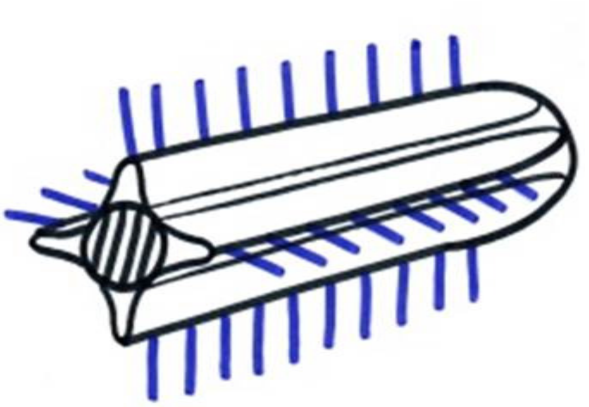


Figure 2.34: Final Concept

Inside the base unit there would be a myRIO single-board computer for steering computation, a built-in camera for marine life filming, and a huge battery to provide power. An IMU, as well as pressure- and leakage sensors would provide the user with constant feedback while diving. A swim bladder would serve as an additional actuator to control the buoyancy.

A highly flexible cable would connect the robot to our terminal station running a comprehensive LabVIEW interface.

2.11 Risk Analysis

When publicly announcing the concept for the first time, many people warned us of the complexity of the system. We realised that we had to take potential problems seriously and prepare for all eventualities if we wanted to succeed. Therefore, we decided to do an elaborate risk analysis.

Table 2.4 provides an overview of our risk analysis, listing the risks, their consequences, possible countermeasures and their gravity (calculated as the probability of risk multiplied by its severity). This classification helped us identify the risks with the highest gravity, those most likely to lead to a complete project failure. It is evident, that risks such as "Manufacturing defects or faults", "Delivery delays" or "Servo blocking" were likely to be big issues for our project. The first two being mostly external, there was little we could do to prevent them. Good communication and the strict adherence to planned deadlines, which accounted for external failures with buffers, was key.

Category	**Risk**	**Explanation**	**Consequences**	**Measures**	**Prob.**	**Impact**	**P·I**
General	Specifications misinterpretation, missing details or hardware	Hardware does not work as thought, or is missing (e.g. "not enough ADCs on MyRIO"). Unexpected dimensions of mechanical parts.	Parts need to be substituted and reordered. Place for new and bigger devices must be allocated. Doom loop possible. Time problem.	Keep enough free place for additional components. Keep it modular. Keep enough rest budget. Don't buy things, lease them until they work properly in the system.	3	4	12
	Cost		No more money available for vital components.	Sponsoring, cheap construction in CAD.	4	3	12
	Schedule	Failing important time goals, despite the planned time buffers.	Other things will get delayed.	Reduce parts to a minimum. Safe solutions first, no perfectionism.	2	4	8
	No testing facilities available		Tests delayed or not possible.	Arrange testing facilities as soon as possible.	1	4	4
External	Manufacturing defects or faults	Or errors in CAD-files or technical drawings.	Pieces not compatible, possible leakage, etc. Very expensive and time consuming.	Time buffer. Discussion and negotiation about the potential problems. Good communication.	3	5	15
	Delivery delays		Less time for control, Roll-Out, Reviews. Goal delay.	Time buffer. Discussion and negotiation about the potential problems. Look for additional manufacturing possibilities. Work with prototypes for programming until delivery.	4	4	16
Electronics	Too long charging time	Too big battery or too slow charger due to costs or size. Very high currents while charging.	Charging takes a lot of time during testing. System can not be in action and crew has to wait.	Buy second battery-set (keep costs in mind).	3	3	9
	Overcharging of battery inside robot	By misconfiguration of charger or battery failure.	Fire in the robot. Complete destruction of main-electronics.	Preprogrammed settings on charger, instructions.	2	5	10
	Short circuit or reverse polarity	By human error while handling battery.	Battery and electronics fail, might cause fire.	Connectors mechanically not connectible when wrong, circuit-breaker, inverse polarity protection on board.	3	4	12
	Higher currents required than expected	Dimensioning of electronics is based on simple prototype-measurements.	To low running-time, might require bigger battery (difficult to find), longer charging time.	Plugs and cables slightly overdimensioned (up to +50%).	2	2	4

Computer System Failure (Hardware and Software)	Won't fix bugs	System has a bug that won't get fixed by third party. Risk of dependency.	The software won't run as expected and has to be adapted around the problem. Very time consuming.	Keep a second system in mind.	3	4	12
	Hardware failure	System gets destroyed by overheating, inappropriate handling (voltage), etc.	System needs to be repaired or replaced. Very expensive and time consuming.	Protect parts from ESD and other voltages. Do not unnecessary load. Keep an eye on process statistics.	3	4	12
	MyRIO gets destroyed		Too expensive to buy a new one. Software may need to be rewritten. Very time consuming.	Keep a MyRIO replacement in sight.	2	5	10
	Update bugs	System or software gets broken by update.	System needs to be reinstalled.	Keep image of software. Continuous integration as a solution. Do not install unnecessary updates.	3	3	9
	Loss of Signal		In lake or ocean the robot might be lost in the deep. In seaworld a diver might have to bring it out of the aquarium.	Fail-safe action (surface). Safety rope.	4	2	8
	Too low performance	Systems performance is not designed for certain tasks, or is at its limit.	Features have to be cut: e.g. only half the update cycle rate.	Keep software as resource efficient as possible.	3	2	6
	No software for the task	Software is not available for some task as it was expected.	Software needs to be written. Time consuming.		2	3	6
Communication	Broken wire		Wire needs to be replaced.	Keep replacement wires.	2	2	4
Leakage	Fin Case	Water rinsing inside the Fin Case and destroying servos and servo controller.	Useless servos, very expensive to replace. Time consuming.	Separated boxes or compartments, sensors, waterproof servos (IP 67). Discuss with experts.	3	4	12
	Connectors	Water rinsing into the connectors may damage the cables or enter in the Base Unit.	Single components not controllable, redesign of the connectors might be expensive.		2	4	8
	Swim bladder	Water rinsing into the Base Unit via the swim bladder.	Redesign of the swim bladder required, time consuming.	Grease, experiments already done.	2	5	10
	Base Unit	Water flowing through the sealings into the cylinder and harming the electronics.	Loss of connection, destruction of the electronics, very expensive.	Sensors, extra control.	2	5	10
Overheating	Servos in action	While actuating for long time on high frequencies.	Servo failure or destruction.	Temperature sensor, Aluminum contact, reduce frequency or continuous operation time.	3	3	9
	Servos at rest	Overheating while holding zero position during long programming because of weight of fins.	Servo failure or destruction. Short time for programming with running system.	Fins are turnable-off during programming and armed when testing movements.	2	4	8
	Electronics	To high loads.	Shut-down by internal protection, redesign required.	Read trough datasheets carefully, overdimensioning, fan.	2	3	6
	Battery		Electronic components might be harmed.	Temperature sensor, power measurement.	1	5	5
Fin	Servo blocking		Fin movement hindered. Maneuvering more difficult. May damage the foil.	Make servos replaceable. Just make sure the sealing holds. Plan intended breaking points in the mechanics. Watch the current consumption.	4	4	16
	Material break	Scratches in the foil or cracks in the sticks.	Maneuvering gets slower and more difficult. Might required redesign.	Exchangeable fins.	3	3	9
Swim bladder	Not working			Active diving with fins.	2	3	6
	Sinking	Due to pump failure (full swim bladder cannot be emptied). Overweight by misconstruction.	In the lake or ocean the robot might be lost in the deep. In seaworld a diver might have to bring it out of the aquarium.	Dive up actively with the fins. Rescue Manually (e.g. via rope). Safe testing environment with not more than 10 meter depth. Calculate weight conservatively.	2	2	4
Transportation	Damage due to transport		Damages have to be fixed. Some pieces might have to be replaced.	Transport-box. Transport with care.	2	4	8

Table 2.4: Risk Analysis

Chapter 3

Mechanical Design

The following chapter will cover the design and fabrication of our hardware. This includes detailed conceptual decisions as well as their practical implementation. We will present the design of the fins, the base unit and the outer shell. The sealing of the entire robot is also addressed.

Contents

3.1 Fins

Four fins are arranged symmetrically around one base unit and can be interchanged modularly. Each fin contains nine servomotors. These are aligned inside a watertight fin case. There, each servo actuates one rotatable ray via two small shafts and a bevel gear. The rays are placed outside in the water. A latex foil glued to the rays constitutes the actual fin. By actuating the servos in the right order this foil performs the desired undulating movements. In the following chapter, we will provide details on the design of the fins and the decisions taken during the process.

3.1.1 Actuation Concept

Having chosen servomotors as actuators, the next step was outlining the power transmission from the servos to the fin. For this purpose we considered multiple options. The main criteria were fabrication effort, space consumption and sealing. We now present you the most promising concepts:

Note: To illustrate these concepts we will use the original drawings created during our design process. The fin case interior, containing the servo motors, is always situated in the lower part of the image. The water is always on top. The fin case wall is either represented by dashed faces or by angled lines. Large rectangles represent servomotors.

Direct Transmission

The first concept proposes to position the servos such that their shafts align to be parallel to the central axis of the base unit. No gears needed, the shafts point directly out into the water. There, the ray is fixed to the shaft in a 90° angle (Fig. 3.1).

Unfortunately the shape of the fin case becomes quite complicated. This require very specialized methods of manufacturing and thus requires enormous costs. Furthermore, this configuration would result in very long fin cases with a limited wave resolution because of a large minimal servo distance.

Dry Bevel Gear

Here the servos are positioned to have their axes face outwards, towards the fin. The rays are attached to a second shaft perpendicular to the servos' own and once again aligned parallel to the central axis of the base unit. It passes from the inside to the outside of the case. An x-ring seal positioned around this shaft keeps the fin case dry. Torque between the shafts is transmitted using bevel gears mounted on the inside of the case (Fig. 3.2).

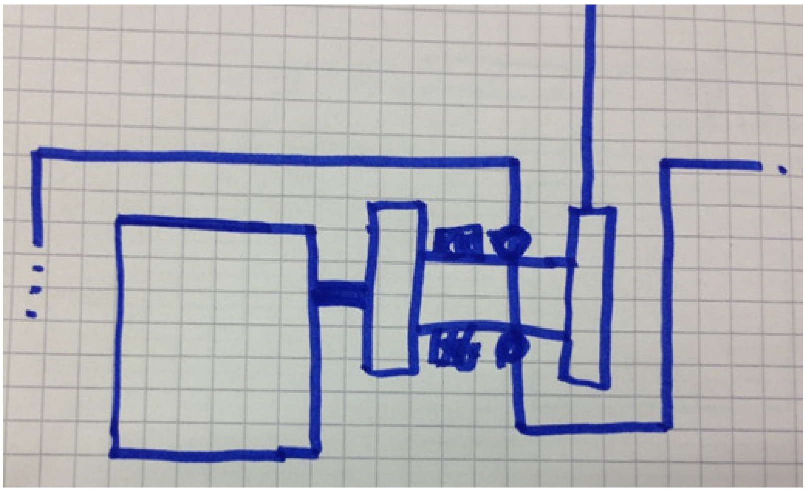


Figure 3.1: Direct transmission

This configuration has the advantage that the rays can be rotated by very large angles, enabling zero position shift (see Chapter 2.8.2). The vertical placement of the servos allows a comparably high wave resolution. Again, as extrusions for every single servo are needed, it is not possible to mill the entire configuration from a single solid. This increases the amount of parts to be manufactured and the amount of seals needed. More seals result in a higher risk of leakage. Additionally, designing a proper bearing for the outer shaft would be hard. Finally the assembling would be tedious, as many parts are not easily accessible.

Wet Bevel Gear

This configuration is very similar to the one described in Section 3.1.1. Again two shafts, arranged in the same geometry, are connected with bevel gears. The main difference is the placement of the bevel gears and second shafts entirely in the water. An x-ring seal seals a part of the servo-shaft instead of the second one (Fig. 3.3).

In contrary to the dry bevel gear solution, the case can be milled as a single part. Furthermore, the entire outside structure bearing the shafts can be done very simply. These conditions are ideal for simple and thus cheap manufacturing and also reduce the number of seals. Handling should be very straightforward because of the simple arrangement of the components. The main disadvantage of this solution is that the bevel gears have to be selected out of a corrosion resistant material.

Tooth Belt

As an alternative to bevel gears we considered the use of tooth belts. Another two shaft concept. Here servo-shafts and outer shafts are both parallel to the

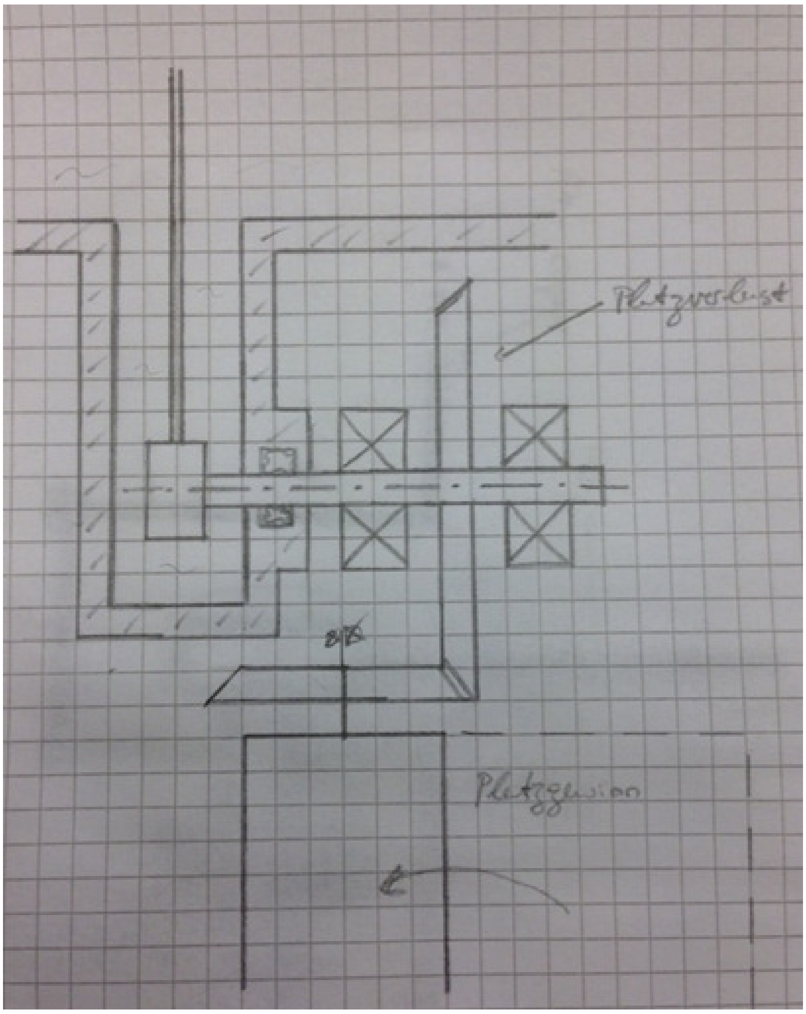


Figure 3.2: Draft of the Dry Bevel Gear solution

X-axis. The torque between the shafts is transmitted using a toothed rubber belt (Fig. 3.4).

Again, the rays can be deflected by very large angles, enabling zero position shift (see Chapter 2.8.2). Nesting the servos in pairs might reduce the space needed and allow a high density of actuators. However, similarly to most other concepts, this would require costly extrusions in the fin case. Assembling would also be very cumbersome, as rubber belts require prestretching. This would be very hard to achieve in such a tight space.

Watertight Servomotors

One very straightforward approach was to look for watertight servos. Unfortunately, many companies falsely declare their servos to be watertight even though they are only splash proof. That means that they are resistant only to short

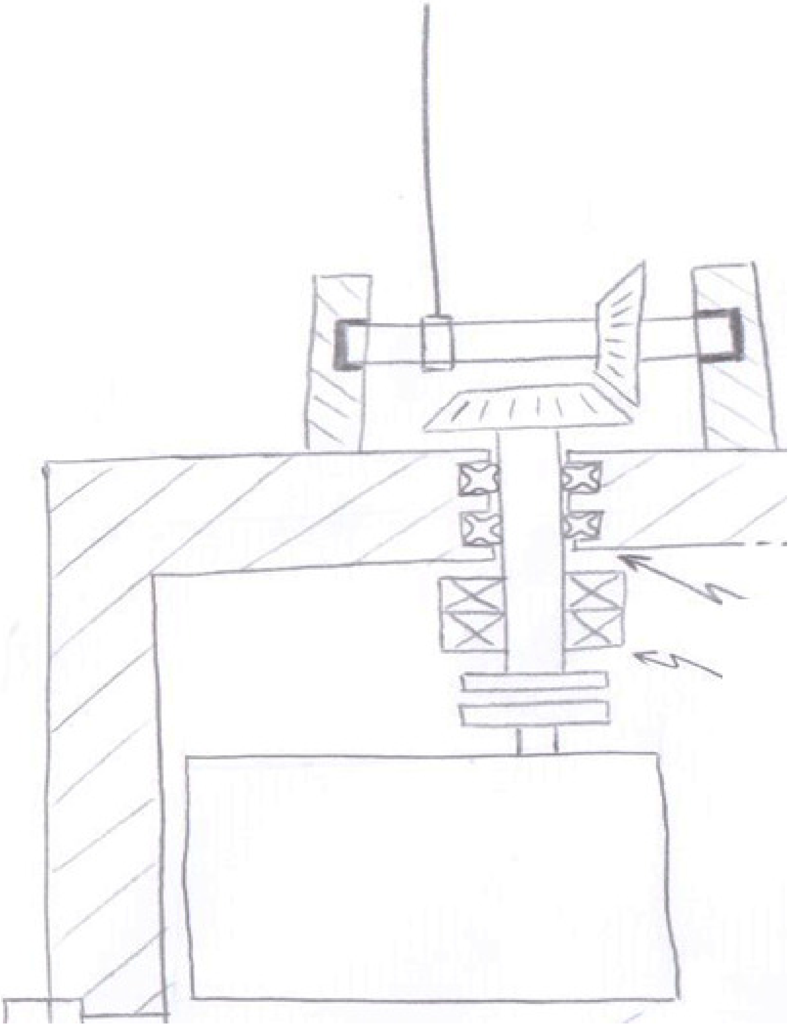


Figure 3.3: Draft of the Wet Bevel Gear solution

exposures of water, not a long, deep dive. We located genuinely watertight servos, the DA 22-SUB distributed by the company VOLZ SERVOS. Unfortunately they completely exceeded our reasonable price range.

One Fin Case per Servomotor

Instead of buying waterproof servos we then considered to place each servo in its own watertight enclosure. This would be a very safe concept, as the failure of one such case would not destroy any other servo. Its mechanics are also very straightforward as the servo would directly transmit the torque to the ray. Unfortunately, only a low number of servos could be installed per unit length because they need to be separated by the walls of the fin cases. Additionally each case would need a dedicated cable with its own watertight connection.

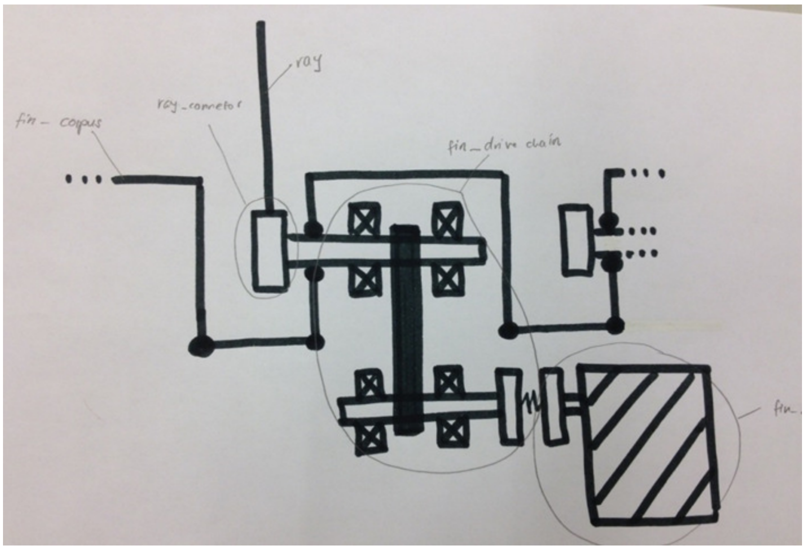


Figure 3.4: Tooth Belt Concept

Final choice for one actuation mechanism

At first we reduced the list of concepts to two favoured options. Our first choice was the bevel gear outside of the fin case. We believed that this was the most straightforward concept, the easiest to produce and assemble. The tooth belt solution was kept as a backup plan, mostly because we had doubts about the bevel gears resistance to corrosion.

In order to choose properly, we decided that more data was needed. For this purpose we built two prototypes which were placed over a water tank of about 1 m^3 volume (Fig. 3.5).

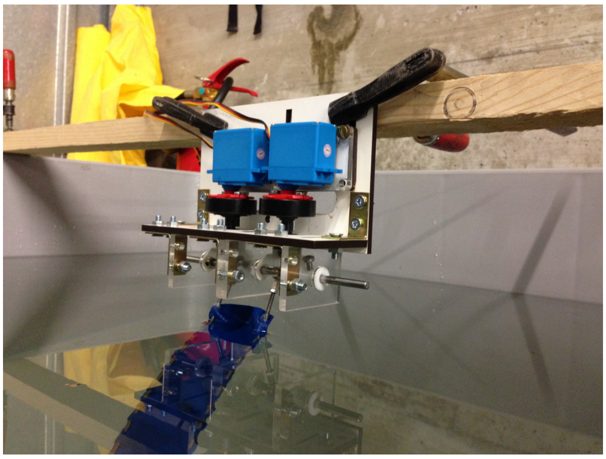


Figure 3.5: Experiment: Bevel Gear vs. Tooth Belt

As expected, the tooth belt prototype was very hard to assemble. Surprisingly, the experiment showed that the prototype using bevel gears required a higher current than the one using a tooth belt (see Appendix 8.2). Acquiring non-corrosive gears turned out to be easy. This meant that the more elegant bevel gear prototype was a clear favourite. See Figure 3.6 for a comparison of the two concepts.

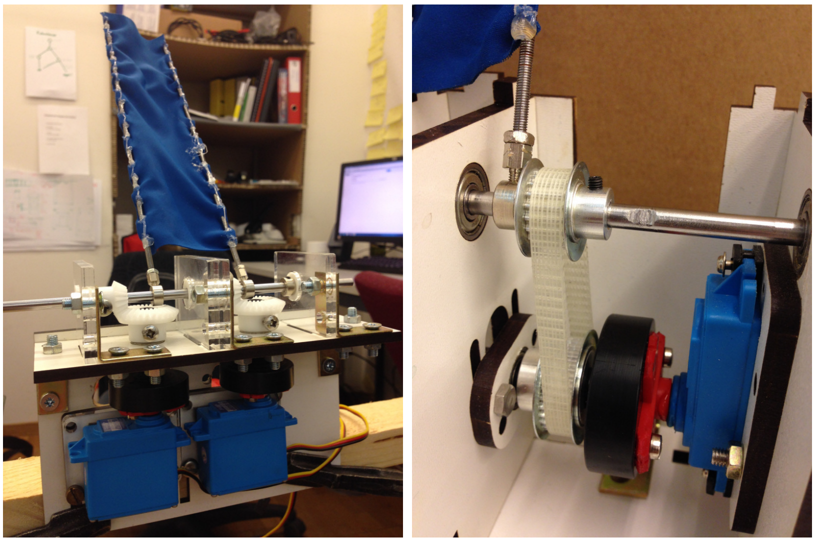


Figure 3.6: Experiment: Bevel Gear vs. Tooth Belt

3.1.2 Components

The subsequent section depicts the solution-finding and implementation process of the mechanical parts of the robot.

Servomotors

The driving force behind our fin propulsion system is the servomotor HS-5646WP produced by the company Hitec. Below we present the reasoning which led up to this choice.

Position feedback is integrated in every common servomotor. This and space considerations made the choice of not using conventional motors evident. The main drawback of a servo is its inability to turn further than a limit. For our purpose this was not cumbersome at all, as the fins are required to perform at most 270° deflections.

Figure 3.7: Servomotor HS-5646WP from Hitec
1

A list of specifications towards the servomotors were raised in order to guarantee an optimal accordance with the electronic power supply and the mechanical working principle. Table 3.1 depicts these specifications.

	Specification	**HS-5646WP**
Length [mm]	max. 40	40
Weight [g]	max. 100	61
Torque [Ncm]	min.100	129
Rotation angle [°]	>180, ideally 270	180, programmable
Time for 60° [s]	< 0.15	unspecified
Voltage [V]	7.4	7.4

Table 3.1: Servo specifications

Many servomotors matching these criteria were found and had to be evaluated in order to choose the optimal device. For an insight into the selection process, please consult the scheme in Appendix 8.2. Together with the HP-DH20-UCD and the Turnigy 1269HV, the HS-5646WP had the highest performance to price ratio. Unfortunately the Turnigy servo was not in stock and the HP product only had a carbon-polymer gear. Servo longevity often hinges on the durability of the gear and metal gears are a lot sturdier.

Before making a final decision, samples were ordered and various suitability tests performed:

- **Watertightness** was tested by running a submersed servo at full load for over 4 hours. Several parameters were varied during the test: Frequencies up to $3Hz$, large amplitudes of max. 30° and different phase shifts.
- **Operating Temperature** was observed with a heat sensitive camera. Temperatures of up to 80° C were measured (Fig. 3.8).
- **Power Consumption** was calculated using the current which was logged during the whole test cycle. This knowledge was used to dimension our battery.
- **Overload** was simulated by applying heavy weight and torque forces onto the servomotor (Figures 3.9 and 3.10) and running them for multiple hours. The servos resisted this marvellously well, only breaking upon being strained like this over the course of an entire night. These tests indicated a high mechanical reliability.

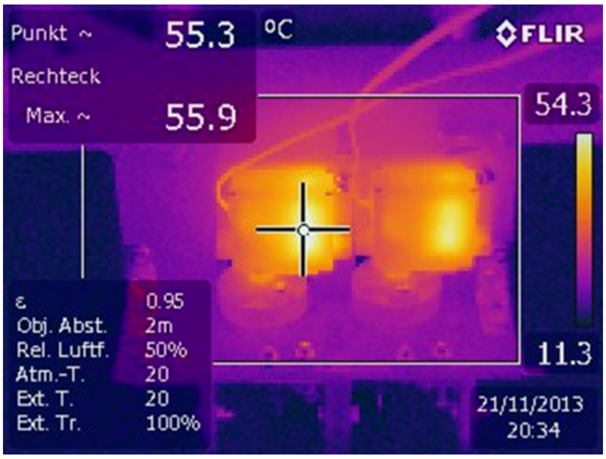


Figure 3.8: Heat generation

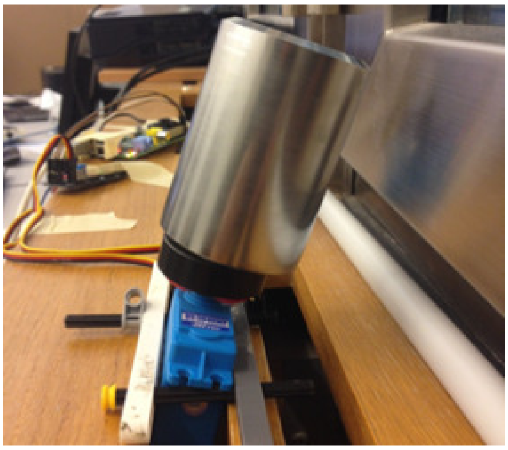


Figure 3.9: Non-concentric weight

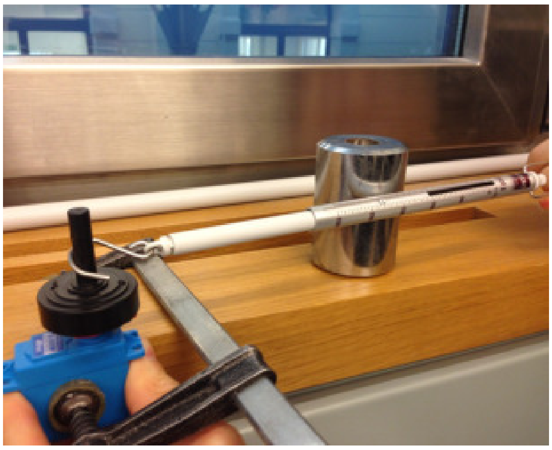


Figure 3.10: Pulling with $2kg$ on shaft

For detailed experiment results and their conclusions, please consult the data found in the experiment protocol in Appendix 8.2.

Problems with HS-5646WP Servomotors: During our initial tests with the finished robot these servos performed exactly as expected. However, the older they got, the more of them got destroyed. As this seriously inhibited our ability to conduct tests, each and every one of the failures was investigated closely. Unfortunately, it seems that our choice of servo motor was not as well-suited as initially assumed. This may have been the single most costly mistake made during the project. A list of all the different failure cases and their consequences was created:

- **Data transmission failure:** The I^2C (see Chapter 4) connection was not perfectly stable and triggered faulty commands from time to time. Especially critical was the malfunctioning of the data transmission regarding incorrect initialisation of the PWM duty cycle frequency. Rather than $60Hz$ a different frequency was initialised. This resulted in a humming noise, similar to the one heard when servos are overloaded, even when the servos were doing nothing. Simultaneously, three servos were destroyed. 105 Hz was later empirically determined to be the eigenfrequency of the oscillating LC-circuit, leading to fatal destruction of the servo interior.
- **Mechanical overload:** Another servo was destroyed in an incident involving a completely blocked ray. The servo kept applying force to its shaft and eventually broke down. As a consequence we attached droplets of glue to the rays just above their fixation to the shaft. Like this they could no more slide in the direction of the cover and cause a blockage of the shaft. On the hardware side we also installed circuit brakers for each fin which would limit the the current to 16 A. Of further note on this subject is that, even before this incident, the servos built-in overload protection was in use (see section 7.6.4).
- **Duty cycles:** Eventually, more and more servomotors started to fail of unknown causes. After multiple servos were destroyed, we talked to the product manager of Hitec, explaining to him the symptoms observed. He immediately diagnosed ageing effects due to number of duty cycles. Apparently the average number of work cycles for an HS-5646WP is 200000. We quickly calculated a reasonable minimum of absolved cycles. The assumption that we had used the robot for at least 50 hours was made. The average wave frequency would be around 1 Hz this resulted in a minimal number of 1 $Hz \cdot 180000\ s$ cycles performed. This was frighteningly close to the end of the expected servo lifetime. Our mistake was to not take the number of cycles into account when choosing the servo. To be fair, numbers are not generally provided as the maximum depends very much on the load. While the servo was able to exercise the necessary torque

> without problems within a short time frame, it was not dimensioned to do so while alternating constantly.

In close collaboration with Hitec, a suitable replacement has been chosen: the Hitec HS-5646WP. As fin cases had been entirely dry until that point, choosing a non splashproof servo to decrease costs seemed obvious. The new servos are dimensioned to supply thrice the torque of the old ones, thus increasing the maximal load cycles manifold. We are now in the process of phasing out the old HS-5646WP and replacing them with our brand new servos.

Transmission

The following section will provide an overview of how the torque of the servomotors is transmitted to the rays.

The biggest challenges during the design of the powertrain were caused by the complex interactions of the components in terms of tolerances and montage considerations. In particular, dimensioning a sufficiently robust transmission with so little space was very hard. The weight limits we needed to abide by in order to keep the robot afloat further restricted us.

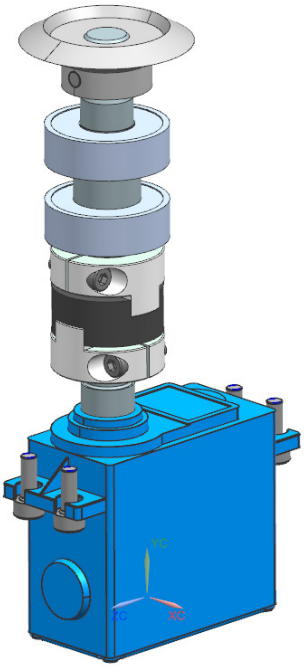


Figure 3.11: Rendering of our powertrain

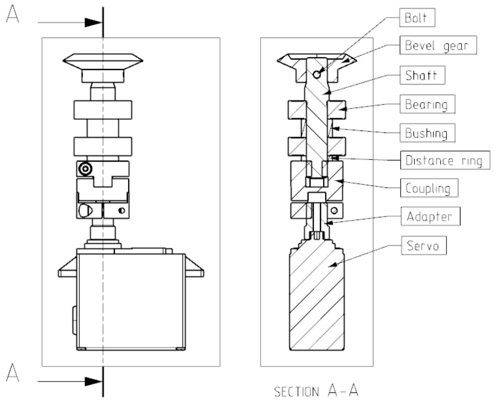


Figure 3.12: Drawing of our powertrain

Table 3.2 serves as a legend to Figure 3.12 which depicts the components of our powertrain. Figure 3.11 depicts the powertrain's CAD model.

Component	Part name and brand	Special characteristics
A: Bolt	3.3882.03016, Hasler AG	Were dimensioned with a security factor of >1.5
B: Bevel gear	35055300, Maedler AG	Gear ratio: 1:1.5
C: Shaft	Custom design	Chrome steel 1.4301
D: Ball Bearing	608-ZZ-MAE, Maedler AG	Top: fixed; Bottom: loose
E: Bushing	Custom design	Improve stability against bending
F: Distance ring	83.3570.08012, Hasler AG	Keep distance between bearing and coupling
G: Coupling	MOHC20, Jakob Antriebstechnik AG	Elastomer coupling for angular and planar deviations
H: Adapter	375HS, Servocity	Linking part between servo and coupling
I: Servo	HS-5646WP, Hitec	Robust and IP 67

Table 3.2: Components of our powertrain

Fincase

As explained in Chapter 3.1.1, we chose a solution based on submersed bevel gears as our final concept. The starting point of our design was the geometry of the whole servo container, the so-called "fin case". Indeed, the construction would have to be compact, watertight, feasible for production, reasonably priced and easy to dismantle for maintenance. It took several weeks of iteration to reconcile all these constraints. The result was the following concept (Fig. 3.13):

A simple aluminium box into which a skeletal servo holder structure is lowered. The advantage of such a design is that the entire powertrain can be assembled, adjusted and tested in a first step. Only once everything works the structure is lowered into the aluminium box and sealed off.

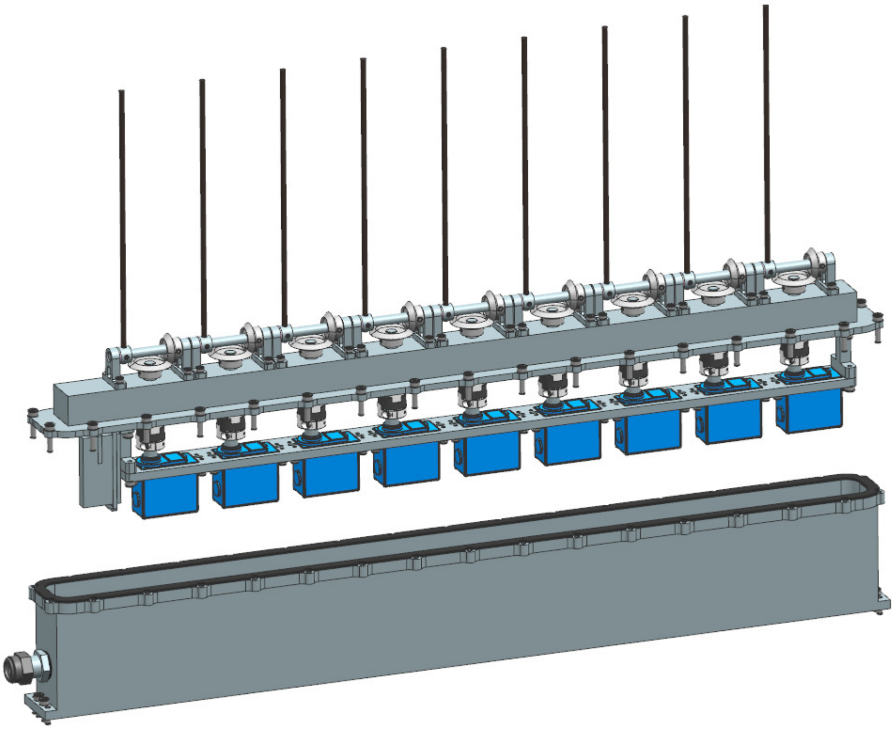


Figure 3.13: Final fin actuation concept

In parallel to the box design, we also looked for the best bearing solution for the shaft attached to the servo. We decided to use a fix and and loose ball bearing. It is mounted as close as possible to bevel gears in order to absorb the induced bending torque. Furthermore it should allow for a small diameter of the shaft section which is in contact with the x-ring seal in order to minimize friction. Figure 3.14 shows how this was implemented in our CAD model.

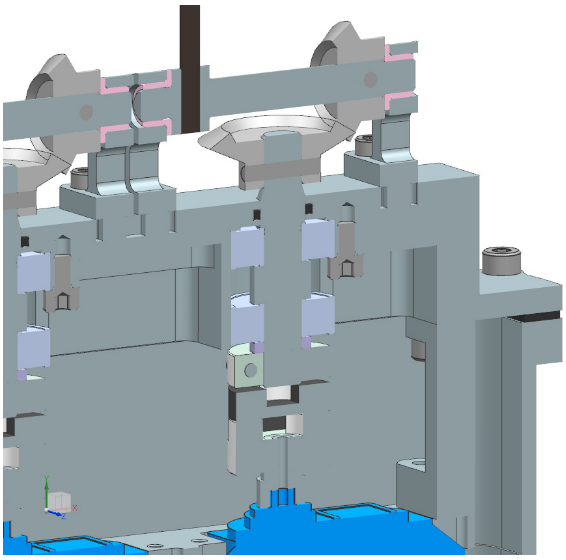


Figure 3.14: Detailed view of the powertrain, including and bearings (light blue)

For the outside part of the fin mechanics, we decided on having the gears fixed onto the shafts with bolts. For each outer shaft, two holders with sliding bearings are screwed onto the fin case. This allows us to remove any pair of bevel gears modularly, without dismantling the entire fin case. An exploded view of the entire fin case CAD can be found on the DVD.

Fin Design

Already with our early LEGO-Prototype experiments we had narrowed down the selection of fin materials to latex and Lycra (see Chap. 2.3.2). Two further experiments were conducted to gather enough data for an informed decision. During the first one we measured the torque needed to reach a given angle between two neighbouring rays in a fin. During the second one we compared the permeabilities of the materials.

Elasticity The elasticity of Lycra and latex were compared. We also wanted to estimate the torque needed to deflect two adjacent rays to a given angle for both materials. The experiment set-up is shown in Figure 3.15.

Each sample fabric was fastened between two rays of 25 *cm* length. One ray was in a fixed position while the other one was free to move. The required torque was measured with a spring scale attached to the tip of the movable ray. Each sample was tested twice. The results are visualised in Figure 3.16. Red represents the two experiments made with Lycra while blue represents latex. For geometric reasons, the range of the measured angles is slightly smaller for latex than for Lycra.

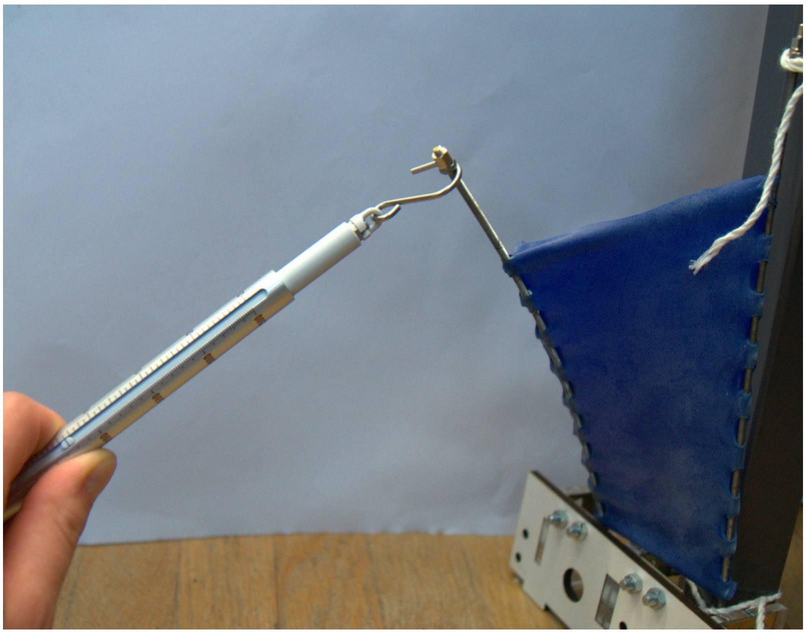


Figure 3.15: Set-up of the stretch test

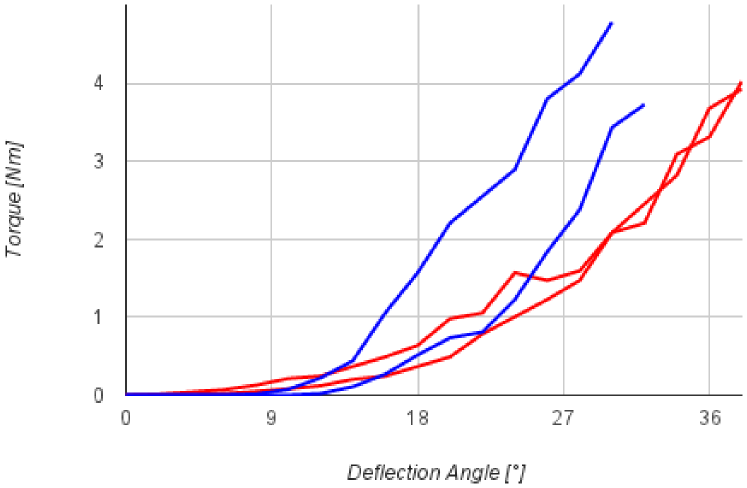


Figure 3.16: Elasticity experiment with Lycra (red) and latex (blue)

Unsurprisingly latex turned out to be much stiffer than Lycra. It could be shown that the torque for stretching latex to more than 15° is close to the maximum torque of our servomotors. This would favour Lycra, as less torque avoids overload. The trade-off was that less torque would be converted into forward thrust. Weighing both points equally important, we used the permeability experiment as tiebreaker.

Permeability This experiment investigated the permeability of the fabrics. The higher it is the more pressure is lost on the fin surfaces as water diffuses through the material. This results in a loss of thrust. The set-up of the experiment is shown in Figure 3.17.

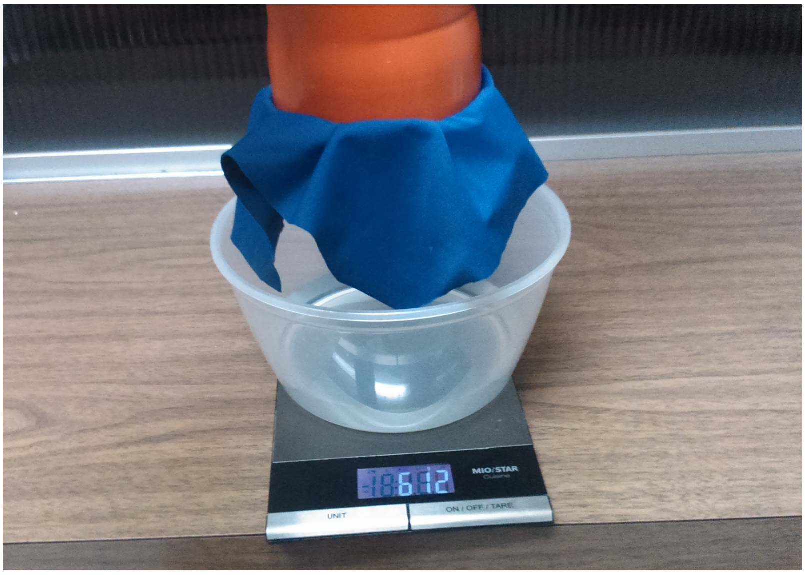


Figure 3.17: Set-up of the permeability test

The tested material is stretched across the opening of a short pipe of about half a metre length. The circumference of the pipe is 0.37 m. This results in a diameter of $\frac{0.37\ m}{\pi} = 0.12\ m$ and a cross-section area of $A = (0.12\ m)^2 \cdot \frac{\pi}{4} = 0.011\ m^2$. The opening of the pipe can be closed with a cover. The basic idea of this experiment is to close the pipe on one side after spanning the material over the opening and to fill in water from the other side. Afterwards the pipe is held over a bucket which is placed on an electric scales. Then the cover is removed so that only the fin material prevents the water from emptying into the bucket. The more permeable the material, the faster the bucket is filled.

Let ρ be the density of water and V_0 the volume of the water trapped inside the pipe before the cover is removed. Further let $m(t)$ be the mass of the water inside the bucket and $V(t)$ the volume of water in the pipe at time t after removing the cover. Conservation of mass enforces $m(t) + \rho \cdot V(t) = \rho \cdot V_0$. The volume $V(t)$ depends on the height $h(t)$ of the water column in the bucket by $V(t) = A \cdot h(t) \Longleftrightarrow h(t) = \frac{V(t)}{A}$. We call a point on the free surface of the water ”1” and a point just above the foil ”2”. Applying Bernoulli’s law between 1 and 2 we find:

$$\begin{aligned}\int_1^2 \dot{u} ds + \frac{u_2^2 - u_1^2}{2} + \frac{p_2 - p_1}{\rho} - gh &= \\ \dot{u} \cdot h + \frac{u_2^2 - u_1^2}{2} + \frac{(p_1 + \Delta p) - p_1}{\rho} - gh &= \\ \ddot{h} \cdot h + \frac{\dot{h}^2 - \dot{h}^2}{2} + \frac{\Delta p}{\rho} - gh &= \\ \frac{\Delta p}{\rho} - (g - \ddot{h})h &= 0\end{aligned}$$

$$\Longrightarrow \Delta p = \rho(g - \ddot{h})h$$

Therefore if the water is accelerated slowly enough, the pressure difference can be approximated using the primary hydrostatic equation. Such a material would be regarded as impermeable. If, however, the water is accelerated too quickly only a small pressure difference appears across the foil.

The purpose of the scales was to measure the mass $m(t)$ in the bucket and to find $h(t) = \frac{V_0}{A} - \frac{m(t)}{\rho \cdot A}$. The rate with which $h(t)$ decreases would allow us to compare the permeability of the tested materials. In the experiment we found that latex was practically impermeable. It was clearly visible that the water only emerged through the small gaps between the latex and the wall.

In stark contrast, all of the water immediately passed through the Lycra sample. The pipe was nearly empty after only two seconds. These results showed us that latex is much better suited to convey servo torque into thrust. Subsequently, we chose latex to be our fin foil material.

The fins (Fig. 3.18) consist of a cone shaped fabric glued to nine small carbon sticks. These sticks are designed to be the weakest link in case a fin collides with an object. This prevents the servos or bevel gears from getting destroyed during such calamities. Because of this and also for more flexibility, the rays are fastened to the shafts with only a single screw each, allowing their quick attachment or detachment. Extra space was left between the rays so that they can be operated at a reasonable phase shift without stretching the material. In the standard configuration, three fins are blue and one fin is golden. This characteristic allows the operator to recognise the robot's attitude, which would otherwise be very challenging as the robot is almost perfectly symmetric.

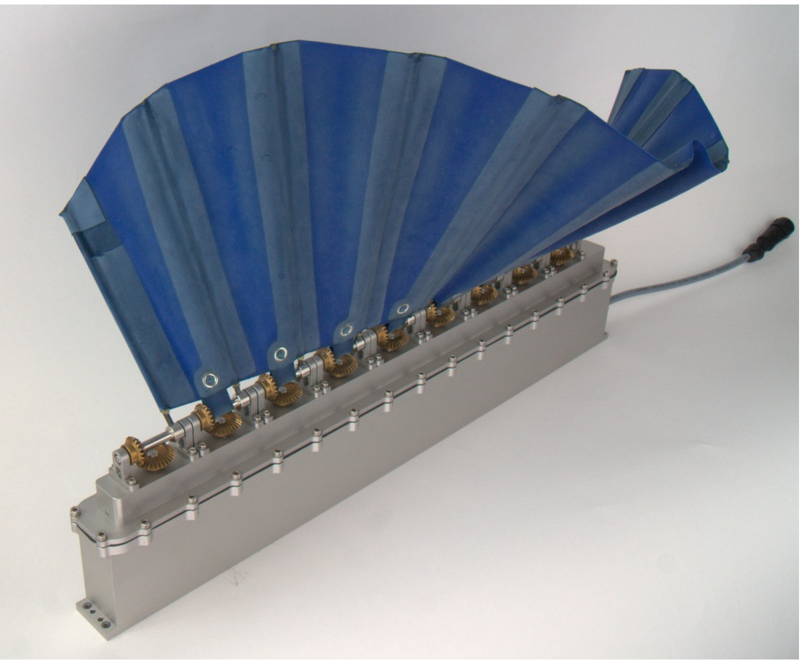


Figure 3.18: Final design of the fins

3.2 Base Unit

The central structure of Sepios is called the base unit. It is a cylinder made out of polycarbonate with two aluminium covers at each end. A carbon fibre construct keeps all the interior electronics in their place and is easily retrieved for maintenance. To tare the robot there is a swim bladder in the base unit. This chapter deals with the design of these structures.

3.2.1 Central Structure

As described in Chapter 2.10, we decided to derive from the existing and well-tested base unit concept of the Nanı́ns. The cylindrical shape ensures to keep potential invading water from reaching the electronics. Having an already sealed concept to build upon saved us a lot of valuable time and effort and certainly contributed to the project's success.

As our robot only had to withstand relatives pressures of 1 *bar* a polycarbonate tube provided adequate rigidity at low costs. Transparency allows for easy checking of the interior. The length of the base unit was constrained by the length of the fin cases. Finally, the following specifications were given:

- Length of polycarbonate cylinder: 623 mm
- Outer diameter of polycarbonate cylinder: 120 mm
- Wall thickness of polycarbonate cylinder: 3 mm

The cylinder is enclosed by two aluminium covers sealed with O-ring seals. These covers are screwed onto two octagonal aluminium rings imposed onto the polycarbonate cylinder. The octagonal rings allow for a modular attachment of the fin cases.

Figure 3.19 illustrates the principle: The octagonal ring is displayed in blue while the polycarbonate cylinder is shown in a light gray-blue. The back cover is dark gray.

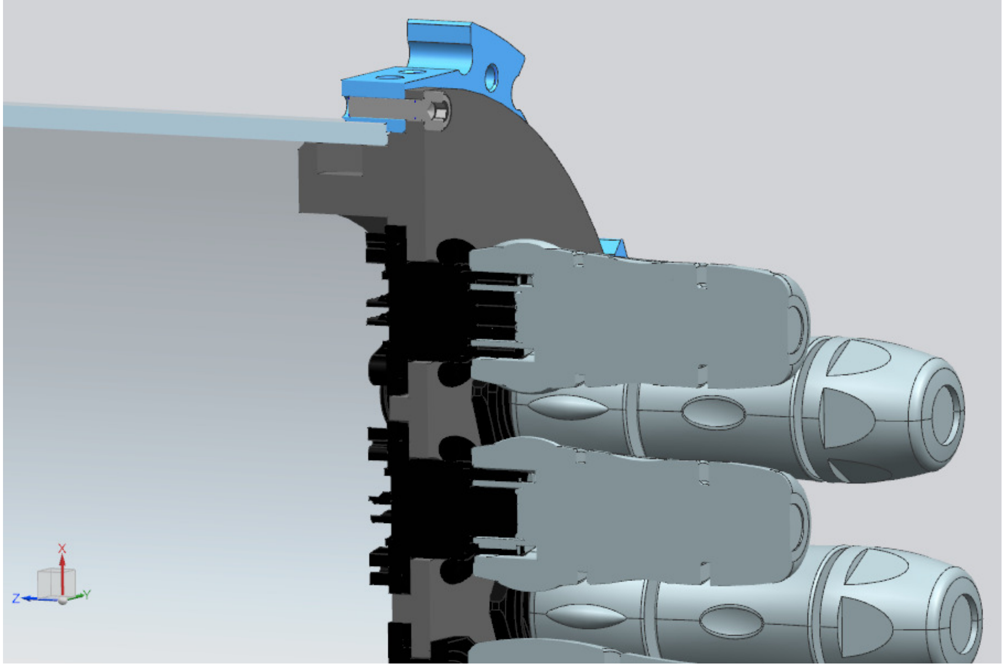
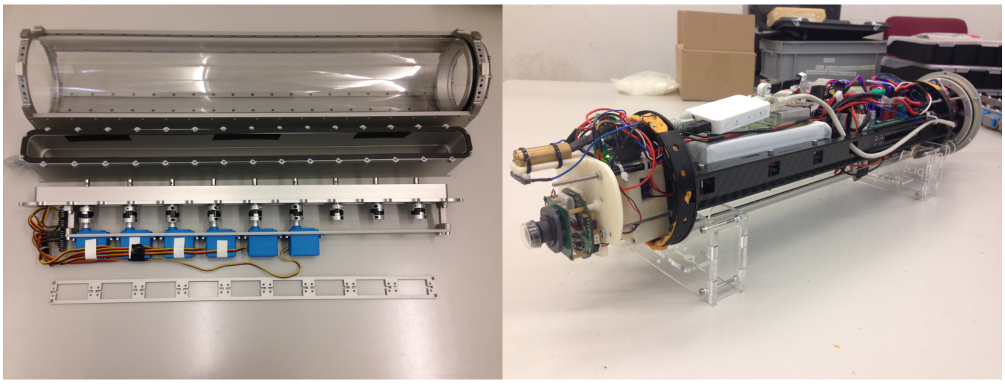


Figure 3.19: Section view of base unit cover

The interior structure (Fig. 3.20b) is attached to the back cover by a metal thread for easy adjustment of the centre of gravity. Furthermore this connection to the back cover allows for an easy extraction of the interior structure by pulling on the cover. All the cables leading outside pass through this cover, meaning that no interior cables ever have to be detached to extract the structure.Next to all the plugs for cables leading outside, the back cover features holes for the swim bladder pump and for the pressure sensor.

On the front cover, a big acrylic glass acts as a porthole for the built in camera to see through. During construction a lot of time was spent to ensure an even weight distribution to align the centres of gravity and buoyancy.



(a) Polycarbonate cylinder next to fin case (b) Carbon fibre interior of base unit

Figure 3.20: Base unit

3.2.2 Swim Bladder

This section covers the swim bladder, a mechanism that enables our robot to sink and lift without using its fins. This mechanism consists of a hydraulic cylinder and a gear pump which can fill or empty the cylinder with water from outside the robot. It is located inside the base unit.

Concepts

We started by analysing the existing swim bladder in the Nanı́ns. This swim bladder had a volume of approximately 3 *dl*, two magnetic end-position sensors and a pump that worked in both directions. There were, however, some limitations to this swim bladder:

- The magnetic sensor only reported when the cylinder was completely full or completely empty. It did not show the actual position between these states.
- The original volume of 3 *dl* did not provide a lot of sinking/lifting ability: With a bigger volume of the cylinder this would be improved. [2]
- The swim bladder of the Nanı́ns did not work symmetrically. As the cylinder is filled from one side a small torque, parallel to the lateral axes, was induced. This caused a slight pitching motion, which had to be corrected constantly.

---

[2] expecting our robot to be more than twice as heavy as the original robot for which the swim bladder was laid out.

To improve on these points, the following concepts were considered:

- Better distance sensor for continuous position feedback:
    1. Laser/ultrasonic sensor
    2. Winch position sensor
    3. Pressure sensor
    4. Flow sensor
    5. Liquid level sensor
- Symmetry problem:
    1. Two separate swim bladders, inside or outside of the base unit
    2. A swim bladder on a rail that corrects the momentum with an additional motor

Evaluation of concepts

1. A short internet research revealed several problems considering a *laser distance measurement*: Most low cost laser distance sensors are designed to work from a minimal distance of several centimetres as they are often designed as collision warning sensors. As we had very small space for the sensor at the end of the swim bladder, we needed a sensor with a minimum measurement distance of a few millimetres. Such sophisticated laser sensors were too expensive as well as too big for our purpose. The same problem applied to ultrasonic-sensors. Ultrasonic measurements would also have been subject to more measurement noise due to reflected ultrasonic waves in the enclosed pipe.

2. A *winch position sensor* counts the rotations of a winch with an attached string. The string is connected to a moving part whose position is supposed to be measured. Knowing the diameter of the winch and the starting location, the position of the part can easily be calculated.
   By choosing diameter and rotational sensing type, a very high resolution at very low noise levels can be achieved. However, attaching a cable to the moving part of the swim bladder would introduce additional strain onto the pump. Furthermore, the string's path would have to be carefully chosen to minimize drag and failures.

3. A *pressure sensor* measuring the rising air pressure when the enclosed bladder is filled seemed to be an easy solution as these cheap sensors are widely available and reliable. However, we made a rough estimate of the expected air pressure assuming a one-litre bladder would be filled half-way:

$$p \cdot V^n = const$$
$$n_{air} = 1.4,\ p_1 = 1bar,\ V_1 = 1l,\ V_2 = 0.5l$$
$$\Rightarrow\ \underline{\underline{p_2 = 2.6\,bar}}$$

As our favoured pump maximally generated 2.3 bar this was no option. We needed to bleed the air replaced by the incoming water into the big volume of our base unit to keep pressure increase small. This would have led to the need of a high resolution and therefore expensive sensor or accepting a high noise level. Both were undesirable.

4. Low-cost *flow sensors* are common in garden watering and drink machine industry. They are relatively small and precise enough for our purposes (accuracy of several millilitres). However, all standard flow sensors can only measure flow in one direction. As we fill and empty the swim bladder via the same pump and therefore pipe an additional valve plus a second sensor would have been necessary to estimate the bladder content. The biggest problem was the fact that this method provides no absolute measurement of the bladder state. This would lead to a drifting state estimate over time.

5. *Capacitive liquid level sensors* (as shown in Figure 3.21) can be mounted on the outside of a container and detect the liquid level. They are very thin, flexible and reasonably priced. The downside was there were only predefined lengths available from stock. Furthermore, as we control our swim bladder filling by a moving aluminum part we expected noise problems due to the conducting aluminum.

Figure 3.21: Capacitive fluid level sensor

Decisions

We decided to test one of the cheapest and easiest status feedback options: The winch position sensor. An internet research only turned up too big sensors, designed for industrial-purposes. This is why we decided to fabricate it ourselves.

We constructed a winch from a retractable key ring rope and connected the end of the rope to the diaphragm of the cylinder. A potentiometer attached to the winch measures its angular position. A section view through the CAD-model is shown in Figure 3.22. When measuring the output of the potentiometer we were able to estimate the status of the swim bladder. Since that concept worked out very nicely we decided to use this status feedback.

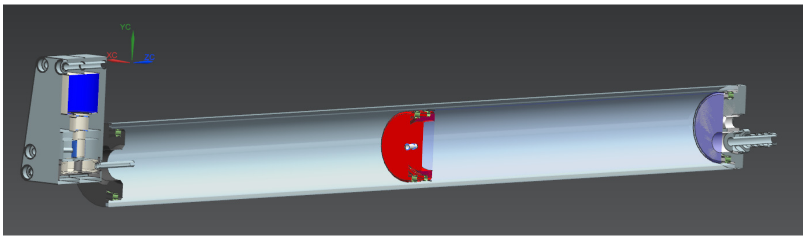


Figure 3.22: Section view of the swim bladder

For the rest of the hardware, we decided to copy the swim bladder of the Nanı́ns in order to save time. Only 3 *dl* of volume meant that the swim bladder dynamic would be almost completely negligible except for trimming the robot and static diving without fins. Sepios is trimmed to be at an equilibrium when the swim bladder is halfway filled.

The pump used for filling and emptying the bladder has the following specifications:

- Volume flow: 1.7 l/min
- Starting current: 2.3 A
- Maximum pressure: 2.3 bar

The pump is controlled by a motor driver in combination with a micro-controller. For the motor driver we had initially designed our own PCB. Unfortunately it was not powerful enough and had to be replaced with a professional solution [3], controlled directly by our single-board computer. This further enabled us to implement the motor controller as well as the status feedback directly into our main software.

3.2.3 Cover and Plugs

To respect the modularity and maintainability of our system, detachable cables were required for connecting the fin cases to the base unit. This meant that waterproof plugs were required. As explained in Chapter 3.2.1 these connectors are integrated into the back cover of the central structure of the base unit.

---

[3] Polulu Dual MC33926 Motor Driver: `http://www.pololu.com/file/download/MC33926.pdf?file_id=0J233`

As the Nanìns project had made good experiences with the UTS-Line-connectors from Souriau[4], we decided to use plugs from the same product line. These connectors are rated according to the IP69K-certification to be tight for two weeks of dynamic use at depths of up to 10 m. However, our fins require higher currents than Nanìns fins(see Chapter 4.2.1), so larger connectors were required. Many spacious connectors on a small-diameter cover lead to extremely cramped placement of the individual connectors. To be able to tighten the connectors using conventional tools, the inside of the cover had to be fabricated in a way it could serve as a counter support. This resulted in a very unconventional and thus expensive part (Fig. 3.23).

Furthermore, in contrary to the small connectors, these large ones are not tight if unmated, meaning a protection cap is required at all times if nothing else is connected.

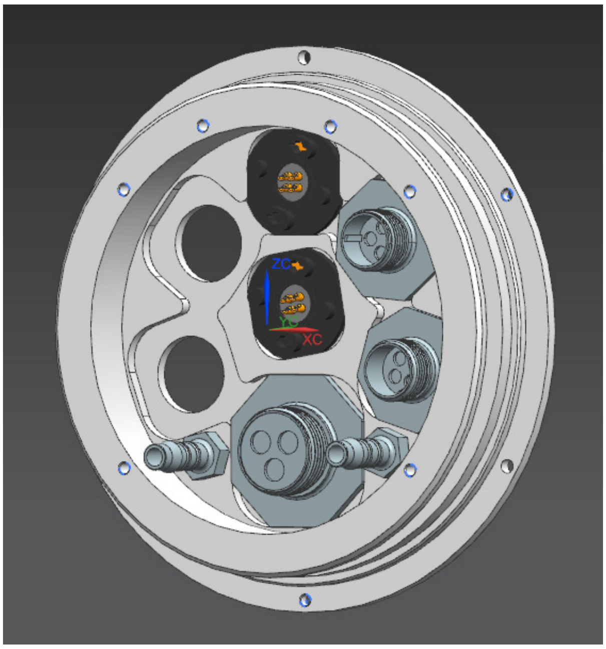


Figure 3.23: CAD model of highly customized back cover with plugs

Even tough we were aware of the disadvantages, the high currents required the use of the larger plugs. We also found no smaller connectors by other manufacturers with sufficient specifications. Therefore we decided to use these bigger connectors.

However, during the tests of our robot we realised, that these plugs watertightness is very susceptible to a bending torque on the cable (see also Chapter 7.3). The specific design of the back cover prevented us from exchanging these connectors with other types. We managed to temporarily halt water intrusion by

---

[4] Souriau is a French connector manufacturer

using hot glue to “seal” the cables. This is very unfortunate, as time and time again, a little water managed to intrude into our fin cases. Our ability to test was severely hampered by these circumstances. Replacement of the plugs is currently under way.

3.3 Sealing

All seals were generously provided by the company Kubo Tech AG. They also helped us design the many unique seals required for our robot. Most of them are made out of FPM Viton as it is very chlorine resistant. Some seals inside the base unit were directly ported from the Naníns project (Chap. 1.3.1).

3.3.1 Sealing of the Fin Cases

Each fin case contains eleven seals. One large gasket is used in a flange connection between the main case and the cover. This flange was designed by us and custom-made by Kubo Tech AG. Gaps were left for the screws which connect the cover to the main case. These gaps also help us to precisely position the gasket on the mating surface.

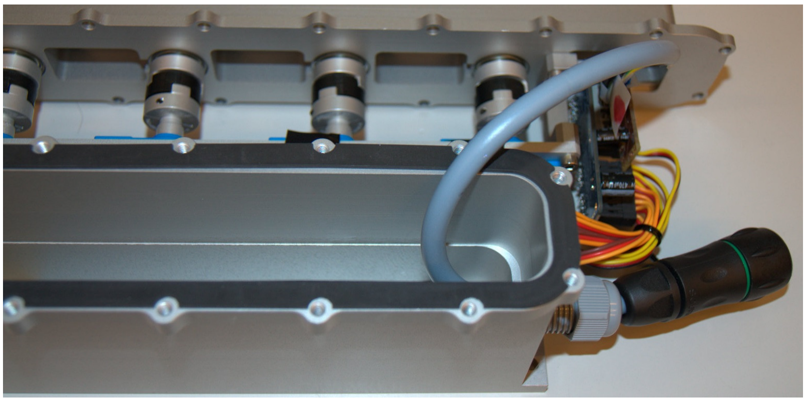


Figure 3.24: Flange gasket sealing the fins

The gaps between the fin case covers and the shafts attached to the servomotors is sealed using one x-ring per shaft. First we considered using two x-rings instead of one. This was deemed unnecessary by sealing specialists. Assembling the x-ring seals proved to be tricky. The x-rings had to be placed inside a round notch. Afterwards the shafts were pushed through the hole. During design we had to consider that the seal could be easily damaged if it were pulled over sharp edges. This fact significantly influenced the shape of the shafts as well as that of the overall assembly.

The fin supply cables enter through a single hole into the fin cases. A watertight M-12-plug is used to seal the gap between the fin cases and the cables. The mating surface of the plugs is only realized with two small plastic ridges. However, it

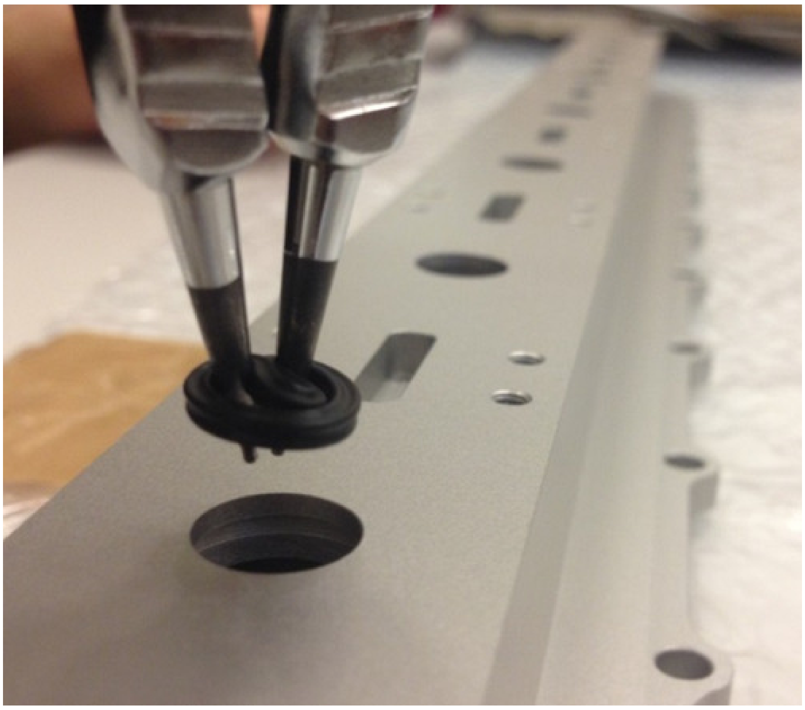


Figure 3.25: Assembling of an x-ring into the corresponding notch

turned out that it was not necessary to reinforce them with an additional o-ring.

3.3.2 Sealing of the Base Unit

Most of the seals in the base unit could be adapted from the drawings of the Naníns robot. One new flange gasket was designed to seal the camera's porthole.

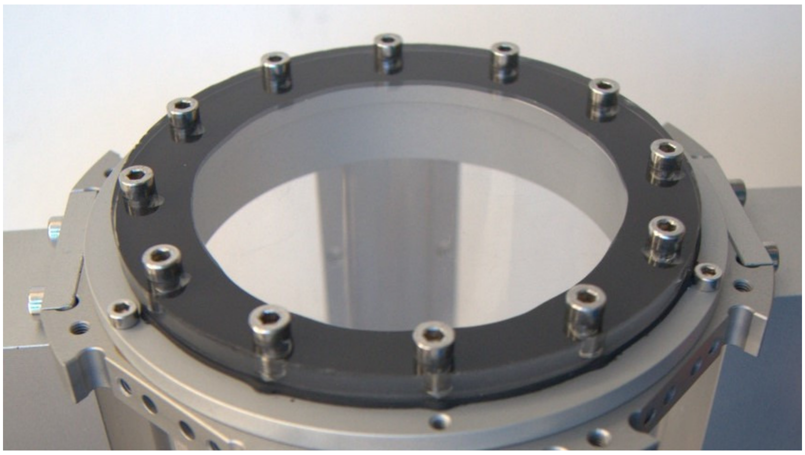


Figure 3.26: Gasket sealing the porthole in front of the camera

3.4 Outer shell

During the design of any submersible, buoyancy is an important concern. This chapter elaborates our treatment of the subject.

One of the first equations we had a look at for our project was how to gain buoyancy in order to prevent our robot from sinking. As we already knew that we were going to use the Nanı́ns base unit we were able to calculate its generated uplift. We were also aware that our fin cases were going to be made out of aluminium and would likely generate some downwards force. Therefore we came up with the idea of creating an outer shell made out of lifting bodies to compensate for the missing uplift.

To calculate the net uplift we had to measure the total displaced volume of the robot minus its weight. This equation is shown below, where $F_B[N]$ is the buoyancy, V the volume in $[m^3]$. ρ_w is the density of water and ρ_{parts} the density of robot parts.

$$F_b = F_V - F_g = g \cdot (\rho_w \cdot V_{displaced} - \rho_{parts} \cdot V_{parts}) \tag{3.1}$$

In total our robot weighs 22.7 kg and generates an uplift of 18 kg. So we required an 4.7 kg of additional buoyancy. The exact volume that the shell needed to be was calculated with a CAD program. To cross check, we also measured the net uplift of the whole robot without shell with a spring scale. The measurement confirmed our expected values.

We wanted to keep the base unit as short as possible to prevent the inertia around the Y- and Z axes from growing too far apart from the one around the X axis. Increasing its diameter would have created tremendous design effort. Therefore the only reasonable option left was to add lifting bodies externally.

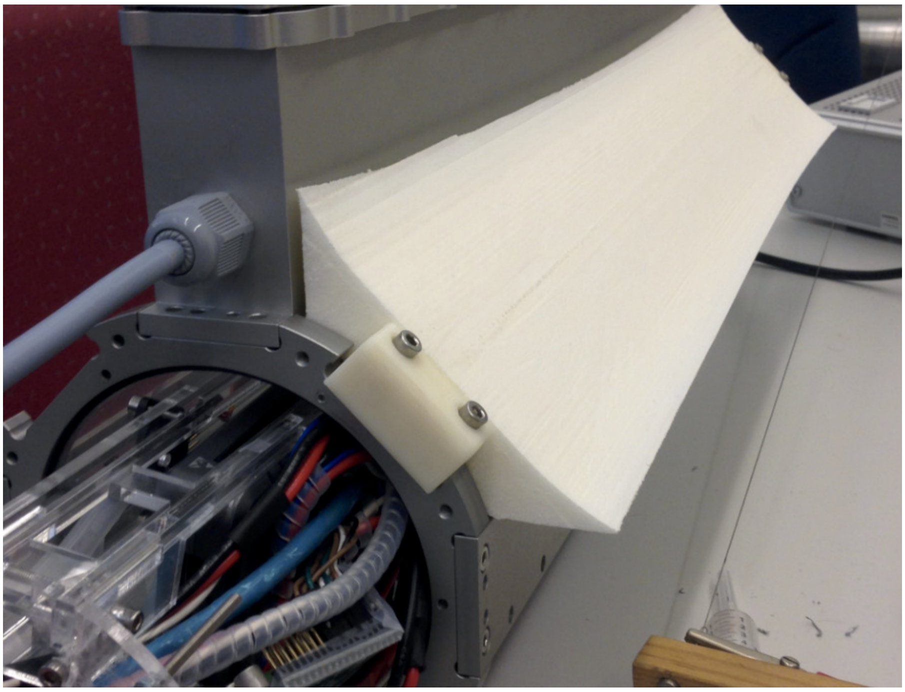


Figure 3.27: Polystyrene shell and 3D printer connector

The buoyancy pads were finally made out of special kind of polystyrene (Fig. 3.27). This was very light and cheap, while also being able to resist a little force and chlorine. The polystyrene was cut into shape with a hot wire. To provide a visually appealing and sturdy surface, the bodies were laminated with two layers of carbon fibre (Fig. 3.28). Water resistant stickers with logos of our sponsors were attached to the laminate. The connection between the polystyrene and the base unit was realized by using little 3D printed connectors. A metal stripe is embedded below the laminate. This can be used to attach little magnets to fine-tune the robots weight distribution.

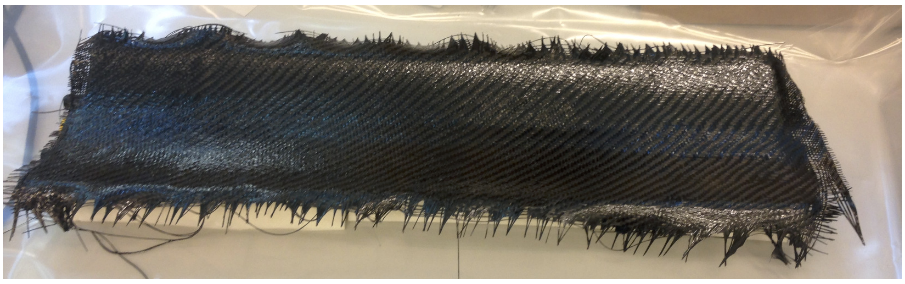


Figure 3.28: Lamination in progress

Chapter 4

Electronics

In this Chapter we will provide you with an overview of the electronics and then present the single electronic components which are installed in our robot.

Contents

4.1 Overview

The electronics inside Sepios have two main tasks: Provide the required energy and connect the software to all actuators, sensors and communication devices.

To simplify the set-up we located all of the “intelligent” electronics and power storage in the base unit. This includes our central microprocessor, a National Instruments myRIO, as well as all general sensors and the battery. With waterproof connectors, external periphery and communication-cables can be connected to the base unit. The most important periphery are the fin cases. The only electronics they contain are servo controllers. These compile the digital values sent by the myRIO to analogue voltage information for the servomotors. Figure 4.1 shows an overview of all electronic components, including their location.

Locating all of the critical electronic parts inside the base unit, provides additional security in case of water leaks. Invading water is always pulled towards the walls of the base unit cylinder by gravity, while all the electronics are located in the centre. Like this, excluding major hull breaches, no electronics get damaged when water enters the robot. In the unlikely case of a complete flooding of the fin cases, the only destroyed electronic part would be the servo controller.

Communication between servo controller and the myRIO is based on the serial I^2C-Bus. I^2C is an on-board bus and protocol widely used in robotics and industrial applications. Various compatible chips are available, including sensors and actuators. I^2C only requires four cables, including the power lines. This helped to keep the amount of cables low.

Another measure to keep cabling down was the design of a connection board. It handles all power distribution, monitors of the most crucial states and provides a pin out for easy accessibility of all required myRIO ports. It is also addressed using I^2C.

Communication to the surface is established through a flexible four conductor TCP/IP-connection cable. The maximal bandwidth achieved is 100 Mbit. The biggest advantage of using conventional Ethernet cables, is the ability to install any conceivable network device on Sepios, similar to the network camera already installed.

The battery is dimensioned to guarantee a continuous operation time of at least 30 minutes under full load as described in Chapter 4.2.1.

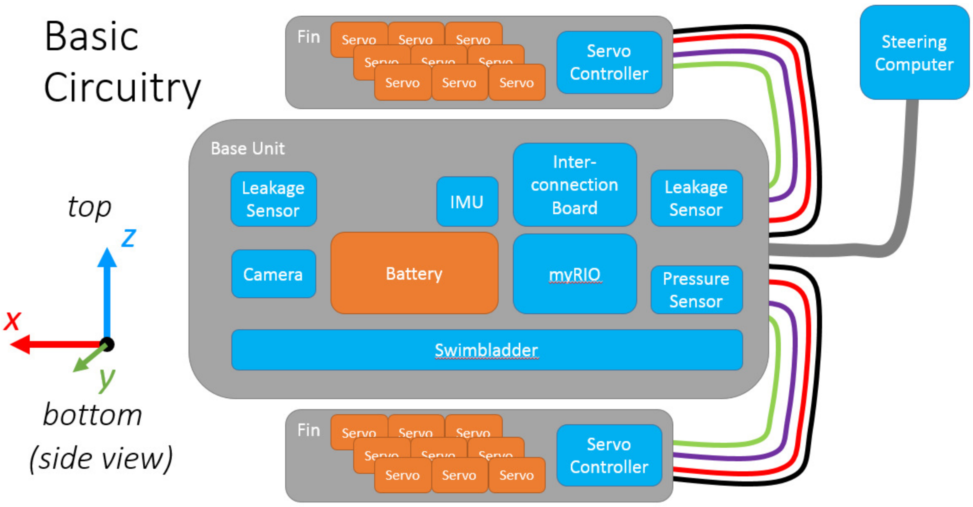


Figure 4.1: Overview of Sepios' electronic parts

4.1.1 Sensors

The following sensors are integrated into Sepios to provide the required observability:

- **Water pressure:** To detect the diving depth we used the MPX4250AP-Sensor.[1] It is normally used to detect manifold pressure for engine control and generates an analogue output. Using a Kalman Filter supported by accelerometer data, we achieved a resolution of about 5 cm. See Chapter 7.4 for more information regarding the Kalman Filter.

- **Leakage in fin cases:** Because our initial model of servo motor was splashproof, we deemed leakage sensors in fin cases unnecessary. In the base unit, a home-made leakage sensor was installed. See Chapter 7.6.2 for more information on our leakage detection principle.

- **Battery voltage and current:** Both are measured directly on the connection board. Current is measured by a high precision shunt resistor directly connected to an analogue pin of the myRIO. Battery voltage is measured by the secondary micro controller of the connection board. This controller is also able to forcefully shut down the entire system in case of critical events (such as exceedingly low battery power). This controller can then by accessed by the myRIO using an I^2C interface. This information enables us to estimate the remaining battery capacity and measure the actual power consumption.

- **Inertial Measurement Unit:** Our system is equipped with a high end IMU providing the Kalman Filter with its necessary input.

---

[1] `Datasheet:http://www.freescale.com/files/sensors/doc/data_sheet/MPX4250A.pdf`

- **Flow sensor:** A biologically inspired low cost sensor is being designed and developed in order to improve the state estimation. This sensor's feedback enables the system to navigate more precisely which increases the autonomy. See Chapter 4.2.8 for more information on our flow sensor.
- **Reed-switch:** A magnetic field sensor serves as an on/off switch which requires no physical contact. It is connected to the connection board and controls its main MOSFET.[2] This enables us to turn the robot on or off without having to open the frame. See also Chapter 4.2.2.
- **Camera:** A Full-HD network camera with live stream serves as the "eye" of SEPIOS. Being an surveillance camera, it delivers good pictures even in dark situations. Lenses are interchangeable thanks to an universal mount. The camera is also used as an input for our collision detection algorithm (see Chapter subsection 7.6.1).

4.2 Components

As with any robotics project, electronics are a very important part of Sepios. Many components had to be custom-made as industrial solutions did not match all our requirements or were too expensive. As an example, the interconnection of all the components is done by our central "Connection Board", a single PCB. In this section we present our most important electronic components and their interactions with the system.

4.2.1 Battery

After the decision to use four fins of nine servos each (see Chapter 2.10) we were finally able to estimate the required amount of power the battery had to provide. Our list of requirements specified 25 minutes of continuous operation at full load (see Chapter 2.4).

Our biggest energy consumer are the 36 servomotors. This is the reason why we optimised our energy system to suit the servos. Servos deliver different torque at different voltages. The higher the voltage, the higher the torque. To keep the current and with it the required cable diameters and battery capacity low, we decided to use "high voltage" servomotors. See also Chapter 3.1.1 for an elaboration of our servos. These servos, common in big RC-models, directly connect to 2-cell Lithium-Polymer batteries. This avoids the need of a voltage converter and associated losses. Therefore, using a 2-cell LiPo battery with a nominal voltage of 7.4V seemed obvious.

---

[2] A MOSFET is a high-current electronic "switch"

Lithium-Polymer-batteries are common in many devices as they offer the highest energy density per kilogram of all commercial accumulators. Wikipedia [2014a] If handled correctly LiPo-batteries are safe to use and can deliver very high power.

Figure 4.2: Sepios' LiPo-Batteries (only one at a time in the robot)

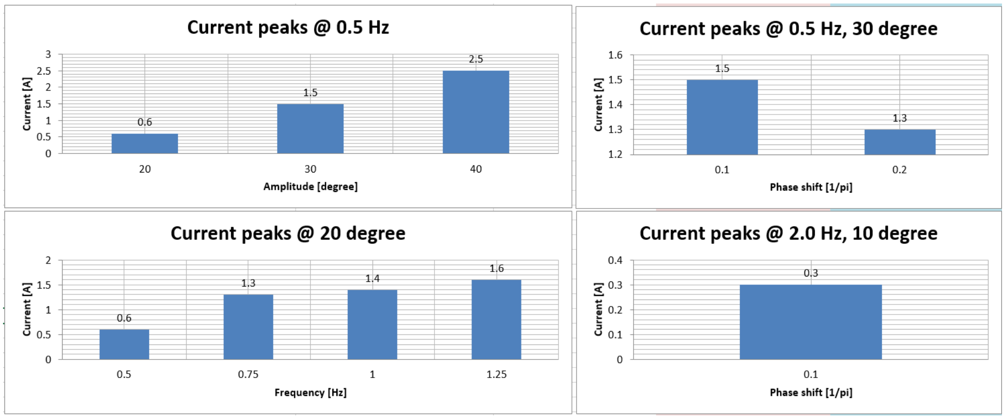


Figure 4.3: Servo current peaks at different fin settings

From our concept prototype with bevel gears (see Chapter 3.1.1) we estimated the required continuous and peak current values. The peak value is highly unlikely to be reached, as this would require all servos travelling at full speed at the exact same time. As we will usually perform sinusoidal waves, speed average over all servos will be more or less steady and not reach any maxima.

Figure 4.3 shows the required servo current depending on the fin movement frequency. This data enabled us to derive the required capacity C_{req} of our battery:

- Nominal battery voltage of a 2s-LiPo battery: $U_{batt} = 7.4V$
- Estimated continuous current per servo: $I_{cs} = 0.6A$
- Amount of servos: $N_{Servos} = 36$
- Safety factor (considering the power required by the micro controllers and various attached devices): $S_c = 2$
- Minimal operation time: $t_{min} = 25min = 0.41666h$

$$C_{req} = I_{cs} \cdot N_{Servos} \cdot S_c \cdot t_{min} = 18Ah \tag{4.1}$$

The following specifications were calculated:

- Continuous total current: $I_{ctot} = N_{Servos} \cdot I_{cs} = 42.2A$
- Continuous required power: $P_c = U_{batt} \cdot I_{ctot} \approx 320W$

As a capacity of 18 $A \cdot h$ is not market standard, we upped the specifications to 20 $A \cdot h$. However, procuring such batteries is not very easy.

Swaytronic offered to sponsor us three customized batteries. They assembled and sent the batteries to us, including the required 20 A-charger. This charger enables us to recharge the battery to about 90 % within one hour. The battery can be charged without opening the robot, through a large 40 A connector. One battery weighs around 850 g.

To ease handling of the battery, we designed a connection board which connects the battery, its balancing ports for charging inside the robot and all power consumers with a minimum of cables. See Chapter 4.2.2 for a detailed description of the connection board.

This set-up allows us to carry out tests without having to worry about the battery level. As the robot never constantly operates at full load, it can dive for up to 2 h with one fully charged battery. The battery state is constantly being monitored by a voltage sensor to prevent deep discharge, which would severely damage the battery. See Chapter 4.1.1 for more information on sensors.

4.2.2 Connection Board

To have an neatly wired system, a “connection board” (Figure 4.4) was designed. Mounted directly on top of the myRIO, it performs all the low-level power-management tasks. These include:

- Ensuring there is a standby mode controllable by a Reed switch (operatable by a magnet from outside the robot).
- Measuring voltage, as well as current levels of the battery.
- Providing status-feedback with an LED.
- Protecting the system from over-current with fuses.
- Routing power to the servos and the myRIO.

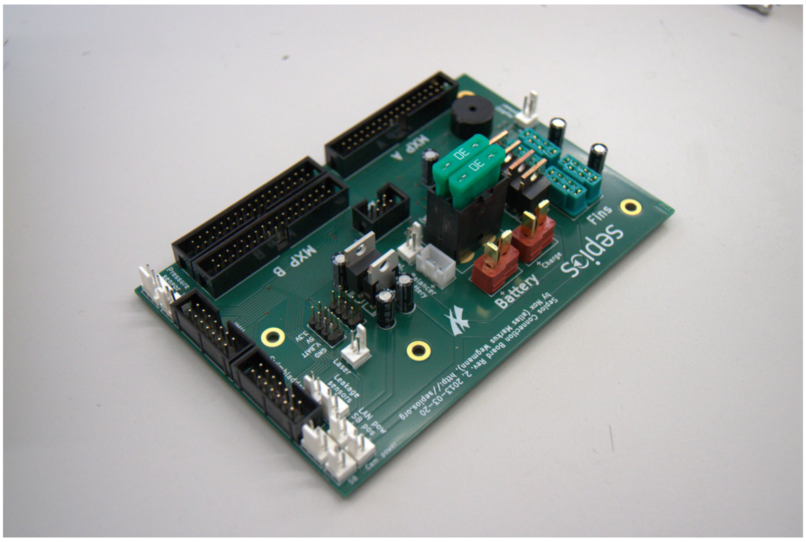


Figure 4.4: The connection board

Schematics are separated into a low-power part, which is responsible for power management and a power-on part which provides power to the different interconnected components. The board is mounted on top of the myRIO with a hook and some loop fasteners. The myRIO is directly connected to the board with a ribbon cable. Through-hole parts were used to reduce count of VIAs (vertical interconnections) on the the two sided board, SMDs were used on the bottom layer. Higher parts, such as the blade fuses and voltage regulators were aligned in the middle along the base unit cylinder, where the space was sufficient. All connectors are labelled with silk screen labels.

The NMOS solid-state relay, based on four IRF1018EPBF transistors, switches the main power on the secondary side (servos, myRIO, periphery) and is controlled by the ATtiny85 microcontroller. The solid state relay is high-side, meaning the secondary side is drawn to battery ground potential in off state. This prevents short circuits between the primary and secondary side ground over other conductive parts. The high side switching voltage is accomplished with a 3.3 V to 12 V insulated DC-to-DC regulator. The primary side is always on, if a battery is installed. It draws around 5 mA in standby mode, and runs on 3.3 V digital voltage. As soon as the reed switch is toggled, the solid state is toggled as well, indicating the status over a buzzer signal (long beep meaning off, three short beeps meaning turning on). This allows us to turn the high power secondary side completely off, even when submersed, in case severe malfunctions.

Peripheral power is then branched off directly from main power at 7.4 V to the myRIO and the servos, or converted by voltage regulators to 5 V or 3.3 V (providing 1 A and 3 A respectively) to components like the laser (3.3 V), LED ring indicator (3.3 V), network switch (5 V) or IMU (3.3 V).

Total battery current is measured on the high-side as a voltage over two “Kelvin contact” shunt resistors and amplified by a INA169. The output of the INA169 is then directly routed to a myRIO ADC input. Kelvin contacts and shunt amplifiers ensure around 1 A precision if no disturbance by the Servos occurs. The conversion of the measured voltage at the ADC pin to a current is the following:

Considering the shunt resistor of $R = 0.5\,\mathrm{m\Omega}$ a current $I = 1\,\mathrm{A}$ induces a voltage of $V = 0.5\,\mathrm{mV}$ at the INA169 terminals. This voltage gets amplified by the internal operational amplifier to a current of

$$\begin{aligned} I_{\text{amp}} &= \frac{V}{1\text{k}\Omega} && (4.2)\\ &= \frac{I \cdot R}{1\,\text{k}\Omega} && (4.3)\\ &= 0.41\,\mu\text{A} && (4.4) \end{aligned}$$

This current flows over a $R_{\text{amp}} = 82\,\text{k}\Omega$ resistor to ground, inducing a voltage of

$$\begin{aligned} V_{\text{ADC}} &= I_{\text{amp}} \cdot R_{\text{amp}} && (4.5)\\ &= \frac{I \cdot R}{1\text{k}\Omega} \cdot R_{\text{amp}} && (4.6)\\ &= 41\,\text{mV} && (4.7) \end{aligned}$$

Concluding that with a voltage measured at the ADC, dividing it by 41 mV gives us the corresponding current I over the battery (also including the charging device). R_{amp} was chosen so that the ADC voltage is limited to 3.3 V at 80 A.

The microcontroller is constantly checking the battery voltage over a resistor network and built-in ADC. It switches the secondary system to low voltage (6.6 V) to prevent battery from exploding, while providing an acoustic warning. The microcontroller provides a software based I^2C slave-interface for the myRIO. The myRIO can read the voltage at any time, trigger alarms on the buzzer, and give the command to switch the solid state relay off in case of malfunction.

The board was manufactured at Multi-CB, a German PCB company. For a better understanding please consider the production files with schemes and PCB layouts or the firmware source code on the accompanied DVD or on the public Sepios GitHub [Wegmann, 2014b].

4.2.3 myRIO

The primary processor of the robot serves as its brains. It computes everything necessary, such as individual servo positions and coordinates all the subsystems. We chose a National Instruments myRIO for this job. Below follows and explanation of our reasoning.

The requirements towards our central microprocessor platform were as follows:

- **Size:** it had to be small enough to fit the base unit (single-board if possible).
- **Robustness:** built in ESD and over-voltage protection were required.
- **Cost:** it had to be affordable (price below CHF 1 000).
- **Computational power:** it had to boast sufficient computational power
- **Connectivity:** many external interfaces would be needed to interact with the rest of the system.
- **User-friendliness:** the board had to be easy to program and well-documented.
- **Power consumption:** Consumed energy was not deemed an important criterion as either way it would be negligible compared to the servos' consumption.

After detailed research, two candidates emerged: The *National Instrument myRIO-1900* and the *Raspberry Pi Model B, 512 MB* as seen in Figure 4.5.

Figure 4.5: From left: The myRIO and Raspberry Pi in comparision

The Raspberry Pi is a 700 MHz ARMv7 based single-board computer. Main feature is its low price of USD 40, excluding the required SD card. It provides all the necessary hardware and software interfaces. Maintained by a open source community, it runs several Linux distributions, including many open software projects. Thus a detailed documentation is available. It had already been used successfully in the Naníns, although performance was at the limits. This was possibly due to poor optimization of the Robot Operating System (ROS) framework on the Raspberry Pi. As a Raspberry Pi was kindly provided by one of the team members, we were able to successfully test it in concert with our servo controllers.

The myRIO has many advantages over the Raspberry Pi, including built-in ADCs, DACs, more GPIOs (general purpose input-output), several powering output voltages (5V, 3.3V), graphical programmability in LabVIEW, dual-core ARM and reconfigurable FPGA System-on-Chip (SoC), built-in WiFi, as well as protection from electrostatic discharge (ESD) and over-voltage. LabVIEW had already been used for the LEGO-Prototype (see Chapter 2.3.2. Communication between desktop- and myRIO-based Virtual Instruments (VIs, name for any LabVIEW program) was very easy to set-up. A LabVIEW license was provided by ETH Zürich. In contrary to the Pi, conventional open-source software would not run on the myRIO. The lack of an Ethernet port was another disadvantage, as a USB-Ethernet dongle would have to be employed. Also, the myRIO takes thrice the space of the credit card-sized Raspberry Pi. It is also placed in a much higher price range (CHF 700 for academic applications), which meant that we could afford only one of them.

Figure 4.6: myRIO beside the connection board. Note the box headers on both components where they get interconnected.

As the sophisticated and easy-to-use graphical programming of the myRIO provided a huge benefit to our coding efficiency and flexibility, we decided for this solution, keeping the Raspberry Pi as a backup plan. The decision was supported by the fact that National Instruments Switzerland would sponsor us a myRIO and provide us with valuable support. Furthermore, one of our coaches was able to provide us with spares to enable parallel coding.

4.2.4 Servo Controller

The servo controller “Adafruit 16-Channel 12-bit PWM/Servo Driver” is a breakout-board (as seen in Figure 4.7) for the NXP PCA9685 chip, a LED driver with programmable PWM (pulse-width modulation) signal output channel. It can be used to set positions of 16 standard servos. The servo controller is connected via I^2C to the myRIO and is supplied with 7.4 volt battery voltage for the servo wires and 3.3 volts for the logic inside the chip. Adafruit provides Python source code for connecting and writing to the servo-controller from the Raspberry Pi. We translated this code to the myRIO.

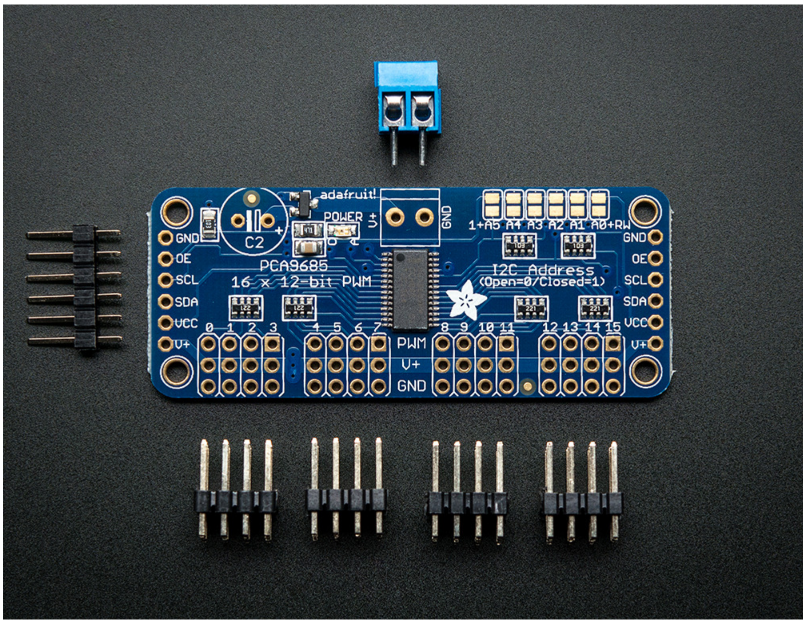


Figure 4.7: The unsoldered top side of our servo controller.

One servo controller is placed in each fin case. Directly on top of it a “supply shield” PCB as shown in Figure 4.8 is mouned. It provides the 3.3 volts for the logic by converting down the 7.4 volts. Like this, only four wires have to be connected to the base unit. The supply shields were manufactured professionally at Eurocircuits, a european PCB company. The board also contains ready-to-use soldering pads for an ATtiny85 micro-controller and connectors for a leakage sensor as possible additional security mechanism.

The servo controller has to be initialized with the right PWM frequency of 60 Hz upon establishing connection. For each servo, two I^2C commands have

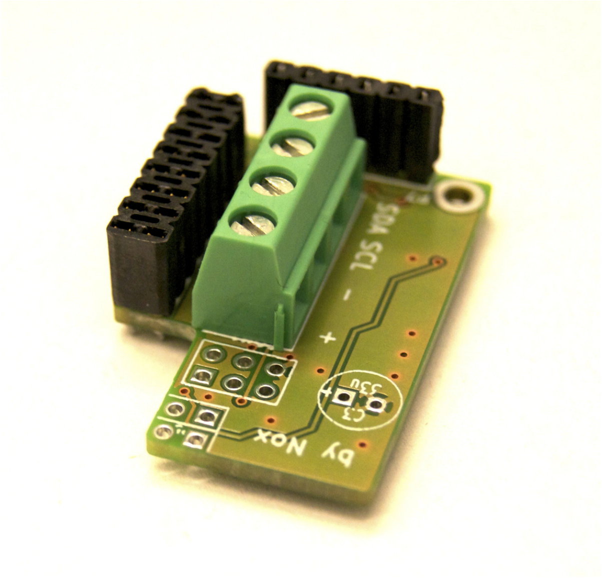


Figure 4.8: A supply shield

to be issued to set the 12-bit duty cycle for that channel, one for the higher byte, one for the lower. Each command consists of an I^2C device address and access mode (1 byte)), the duty cycle register address (1 byte), and the value (1 byte)). For each byte there is an additional acknowledge bit, resulting in $n_b = 27$ bits in total. To have a movement rate of $f_m = 60\,\text{Hz}$ on the 36 servos, the bit-rate f_b is calculated as follows:

$$f_b = 36 \cdot f_m \cdot n_b \tag{4.8}$$
$$= 58.320\,\text{kbit/s} \tag{4.9}$$

Therefore f_b is around eight times lower than our I^2C bus speed of 400.000 kbit/s leaving enough bandwidth for other communication on the bus.

4.2.5 Control and Supply Terminal – Base Station

The base station consists of a notebook running LabVIEW for Windows and a 3D (six axis) navigation controller from 3Dconnexion called a SpaceMouse. There is also a power supply and charging system for charging the LiPo battery directly inside the robot without having to open it. Simultaneous recharging and operating is possible, but limited because of the very short charging cable. The cable which feeds into the robot is connected to a wireless hotspot, thus enabling a very mobile base station.

4.2.6 Camera and Laser

As already shown in Figure 3.20b, the front part of the base unit is occupied by a camera and a line laser. The Full-HD 1080P IP network board camera serves several purposes. First of all, it greatly simplifies the control of the robot by the user, providing him with live-stream images. This is of great help to determine the orientation of the robot or to perceive potential obstacles, even if the surface of the water is disturbed. Second, it can be used to create video footage of dives. This might be useful for experiments on filming oceanic fauna.

The network camera is also used, coupled to the green line laser, as input to detect and avoid collision with the pool's walls. This enables a basic autonomous swimming of Sepios. The laser, which operates at a wavelength of 532 nm and possesses a fan angle of 90°, was chosen for its small size and its low output power of 5 mW (eye-safe). This allows us not to take particular safety dispositions, but on the other hand makes the laser line detection more challenging. To learn more about our collision detection, visit Chapter 7.6.1.

4.2.7 Inertial Measurement Unit

An inertial measurement unit (IMU) is an electronic device that measures and reports a crafts angular velocities, orientation, and accelerations, using a combination of accelerometers and gyroscopes, sometimes also magnetometers.Wikipedia [2014c]

The "ADIS16488"[3], a high-end IMU which measures ten degrees of freedom, is installed on Sepios. Besides being internally temperature compensated it features a pressure sensor (not used within this project), a tri-axis gyroscope, a tri-axis accelerometer as well as a tri-axis magnetometer. Data exchange with the myRIO is accomplished on an SPI-bus. A sampling rate of 100 Hz turned out to be a good compromise regarding estimation accuracy, data transmission via TCP/IP and processor capacity.

The IMU data is used to run an extended Kalman Filter and estimate Sepios' current attitude. More information can be found in Chapter 7.4.

4.2.8 Flow Sensor

In order to control the velocity of the robot a reliable measurement is required. The design and evaluation of a biologically inspired sensor framework for this purpose is explained in a Bachelor't thesis by Dubois [2014]. Next to enabling velocity control, the complete state estimation of the Kalman filter and thus the attitude control would also be improved.

---

[3] Datasheet: `http://www.analog.com/static/imported-files/data_sheets/ADIS16488.pdf`

Chapter 5

Modelling and Control

As is to be expected for a system trying to be omnidirectional using 37 actuators, coordinating them to always do exactly what you want is no easy task. This chapter discusses how and to what extent the robot currently is controlled, as well as what might be implemented in the future and what is not realistically possible.

Contents

5.1 Modelling Conventions

As already briefly touched upon in the introduction, to better unify discussions concerning the robot, we decided on some conventions regarding the systems of coordinates. We based these systems on commonly used drone coordinate conventions, which means that the Z-axis points downwards. The most important one is the body fixed system (Figure 5.1), attached to the robot in its geometric center S (also assumed to be the centre of mass). X points forward as the central axis of the base unit, Z downwards through fin number 2 and Y complements a right handed system by pointing through fin number 1.

For the applications requiring it, a space fixed coordinate system is defined at the surface of the fluid, using the same orientation as the body fixed coordinate system (X and Y in the fluids surface plane, Z pointing downward).

Finally, each fin was given its own set of coordinates (Figure 5.2). These are transformed versions of the body fixed coordinate system, shifted such that the X-axis is concentric to the axes of the bevel gears. The systems are also rotated by $-90 \cdot N$ degrees around the X-axis, where N is the fins designated number. Its origin is referred to as point A. The simplified mean force point of attack of the generated fin force is called P.

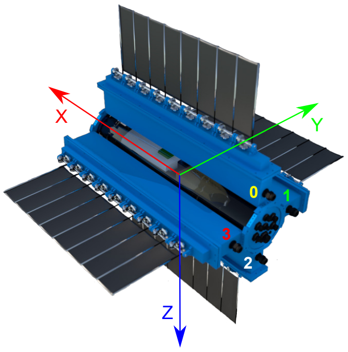


Figure 5.1: Fixed Body Coordinates

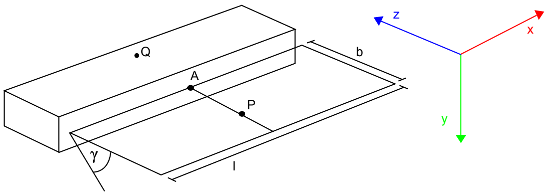


Figure 5.2: Fin Coordinates

5.2 Physical Model

The base for any sophisticated control is a model of how the system interacts with its environment depending on the inputs. Two different models were created, a global one concerning the distribution of a central force among the fins and a local one predicting which forces are generated by the fins.

In our case, many simplifying assumptions had to be made in order to create a feasible model. Naturally, every simplification taken decreases the accuracy of such a model. We assumed that:

- The robot is a rigid body and all the individual fin forces can be added up according to classic rigid body dynamic laws.
- For each fin, the force always acts in the same fixed point $(0, 0, -2/3 * \cdot h)$ in fin coordinates.
- The primary model only looks at the steady state periodic mean force of each fin. Transitions between different states are modelled separately.
- Only sine waves and standing waves are considered, as those suffice to provide omnidirectionality.

For the individual force generation of the fins, an experiment was performed and the results were adapted to a simple model. Fin forces are influenced by four different parameters inside the program. The three needed to fully describe any sine (or standing) wave, in our case: Frequency, amplitude and phase shift and also the zero position shift angle (see Chapter 1.1). Zero position shift simply rotates the force around the shafts' axes on which it is mounted. The experiment confirmed that a peak in frequency existed, around $1.2 - 1.3\ Hz$. It also verified that almost the entire spectrum of forces could be produced for any frequency close to those. We thus decided to use a fixed frequency of around $1Hz$ (slightly lower than optimal to go easy on our servos). The experiment very clearly shows that the generated thrust proportionally increases to the given amplitude of the wave, more or less independent of the phase shift. The angle of the force towards Z is influenced primarily by the phase shift, the lower it was, the smaller the angle. A logical conclusion, as a phase shift of 0 results in a purely flapping motion. This is not as accurately true as the linearity of force magnitude towards amplitude, but was simplified this way for the first iteration of our model. In the near future, tests involving the application of these models will be performed. The entire model and experiment, alongside with the control allocation of the robot are the subjects of the Bachelor's thesis of Flury and Möller [2014].

5.3 Control

For any robot where movement is important, it would be ideal to be able to perfectly control position and velocity at all times. Unfortunately the only states which we can reliably measure are acceleration and angular velocity as provided by the IMU. Fluid based velocity sensors are in planning and may later be used to control velocity. Detailed information can be found in a Bachelor's thesis by Dubois [2014]. Any control of states using integrated data is unfortunately impossible, as even after short moments the values begin to become unreliable because of drifts and biases. The only other useful state measurement is the global Z position, depth, which is provided by a pressure sensor. This leaves four states to be controlled:

- The attitude of the robot (its angular position around each of the three axes of space)
- The depth of the robot

A rudimentary attitude control has already been implemented. To learn more about it and the necessary state estimation filter to extract useful data from the IMU, consult Alessandro Schäppi [2014].

5.4 Control Allocation

One very important problem of Sepios is its overactuation. That means that it has many more actuators than degrees of freedom to act out. In control systems theory, finding the ideal distribution of actuator inputs to create a certain force for too many actuators is called a control allocation problem.

5.4.1 Pragmatic Approach

Our system was ready for testing as early as March, but the control allocation was not. A more pragmatic approach was required.

Considering waves are used as primary method of thrust generation, the idea of superposing basic wave motions seemed apparent. For every of the six degrees of freedom, we created a basic movement. Figure 5.3 illustrates this, with → representing standing waves ⊗, ⊙ "running" waves and 0 no motion at all. With a limited resolution of nine rays per fin, it was obvious that we would not be able to reproduce any arbitrary superposition of sines. We had to minimize the complexity of the waves. To achieve this, frequency and phase shift were kept globally constant for all the fins and only amplitude was adjusted individually.

Given that standing waves are nothing else than the superposition of two "running" sines moving in opposite directions, superposing those basic movements should result in something tangible. Before implementing the solution, a quick programmatic check was performed. Indeed, all the superposition seemed feasible, except for the combination of Y or Z movements with ϕ. This was temporarily addressed by prioritizing the movements according to their intensity.

However, this intuitive way of controlling our robot has some shortcomings. One critical issue is the fact that the basic concept relies on the following idea: in order to keep the undesirable Z force component of each "running" wave under control, a compensation on the other side is attempted. This lead to the symmetry in the basic movements shown above. In order to superpose them, we assumed that the Z components of a standing wave would cancel out the Z component of a "running" wave. This turned out to be not entirely accurate, as standing waves produce slightly less thrust.

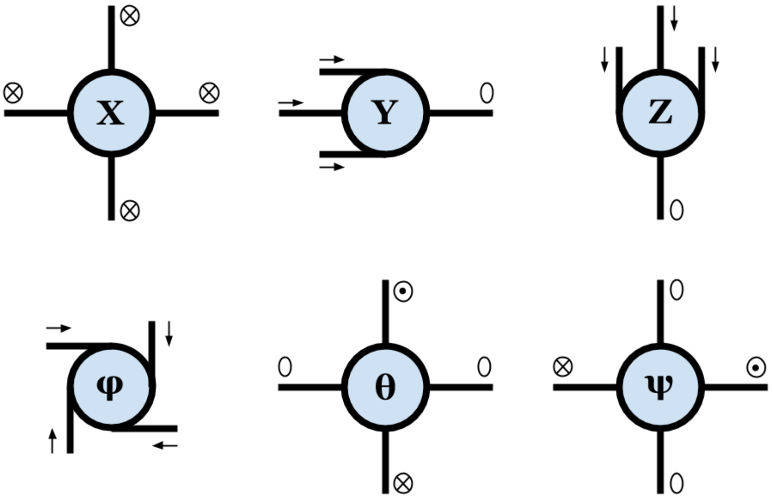


Figure 5.3: Basic movements of our pragmatic control allocation

Another issue was caused by the flaps occurring during the transition between some of the basic movements. They generated an abrupt but strong thrust into one direction, instantaneously displacing the robot. To master this issue, smoother flaps were implemented. They are set to be proportional to the input until an adjustable saturation is reached, where they come to a halt. This greatly improved the ability to control the robot, both for the operator and the PI attitude controller.

The biggest issue was the loss of efficiency due to the superposition. Our superposition works by adding the sines together and then dividing them by the number of sines added. This is done in order to prevent potential overload of the servomotors by exceeding the maximal amplitude. This means that, the more degrees of freedom are exercised simultaneously (and thus sines added), the weaker the movements get. Superposing up to two motions worked fine, more than that reduced the robot's movements to an insignificant speed. While, in theory, this issue could have been resolved, it was never attempted as a more sophisticated control allocation strategy was already in development (see Chapter 5.4.2).

Submarine tests exceeded our expectations. Our approach worked very well and allowed us to perform all the tests needed to verify our list of requirements. We were, for example, able to perform nice loopings (X and θ), to roll while navigating forward (X and φ) or to turn around an object while filming it (Y and ψ).

5.4.2 Mathematical Approach

The classic approach for solving such a problem is to model it as an optimization problem. In the end the problem boils down to nothing more than a linear equation with certain constraints and too many equations to properly determine all the unknowns. This is addressed by optimizing the unknowns for a specific criterion, in this case the classic minimization of theoretical total energy (Figure 5.4).

$$E = X_0^2 + Y_0^2 + Z_0^2 + X_1^2 + ... + Y_3^2 + Z_3^2$$

Figure 5.4: Optimization criterion for four finned configuration

Sepios' control allocator takes the inputs as modified by the attitude controller, a force and a torque in X, Y and Z, and applies the optimization to compute the ideal force for each individual fin to generate. In a second step that information is converted into frequency, amplitude and phase shift which are then converted into PWM's for the servomotors.

To learn more about the problem and how exactly we solved it, consult Flury and Möller [2014].

Chapter 6

Software

Software was very central during the second half of the project. The myRIO and computer interface had to be programmed reliably. This chapter will explain that process, starting with our first approaches and ending in a fully fledged control software called sepiOS.

The final software is fully coded in LabVIEW, a graphical programming language. This allowed the very quick creation of a functional program and easy implementation on the myRIO. The source code can be found on the accompanying DVD or the public Sepios GitHub repository [Wegmann, 2014b].

Contents

6.1 First Approaches

As explained in5 Chapter 2.3.2, we built a one-finned raft prototype out of LEGO. Having already programmed LEGO NXT bricks kits with LabVIEW during the “Innovation Project” in the first year of our bachelor studies, using it again seemed apparent. The ease with which this program was created was crucial for the choice to use the myRIO single-board computer and with it LabVIEW as programming language (see Chapters 6.2 and 4.2.3) for the primary prototype.

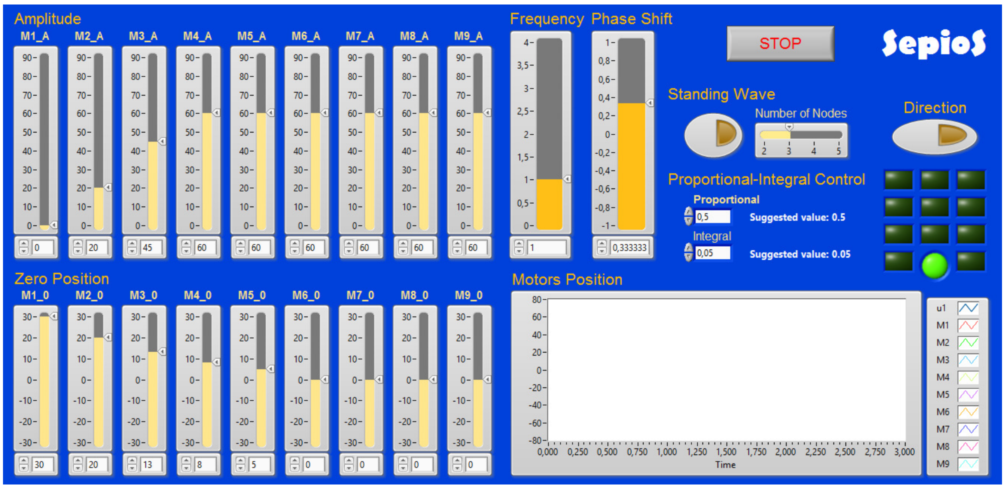


Figure 6.1: Interface of the LEGO-Prototype

The final version of the LEGO-Prototype software calculates the desired position of each step motor and regulates it with the help of a proportional-integral controller. The interface (Fig. 6.1) allows the user to adjust many wave parameters, such as frequency, amplitude, phase shift and zero position shift in real-time. An additional button switches between “running” and standing waves.

Figure 6.2: Cuttlefish performing a “progressive wave”

Although initially based on sine waves, the LEGO-Prototype software can be fed with any 2π-periodic function with an output bounded between -1 and 1. This feature allowed us to try different waves, for example progressive ones derived from the observation of real cuttlefish (see Figure 6.2).

6.2 LabVIEW and myRIO

LabVIEW is a software development environment and a graphical programming language. The user creates a flowchart-like code, connecting different functions with wires. This code is then compiled and executed. A very intuitive and clean method of programming, LabVIEW has saved us countless weeks of debugging.

The myRIO is an embedded hardware device created specifically for students to help them design complex engineering systems. Isuses the LabVIEW Reconfigurable Input/Output (RIO) architecture, which is based on four components: a processor, a field programmable gate array (FPGA), electronic inputs and outputs and a graphical design software. Together, these components allow the rapid creation of custom hardware circuitry with great flexibility.National Instruments [2014]

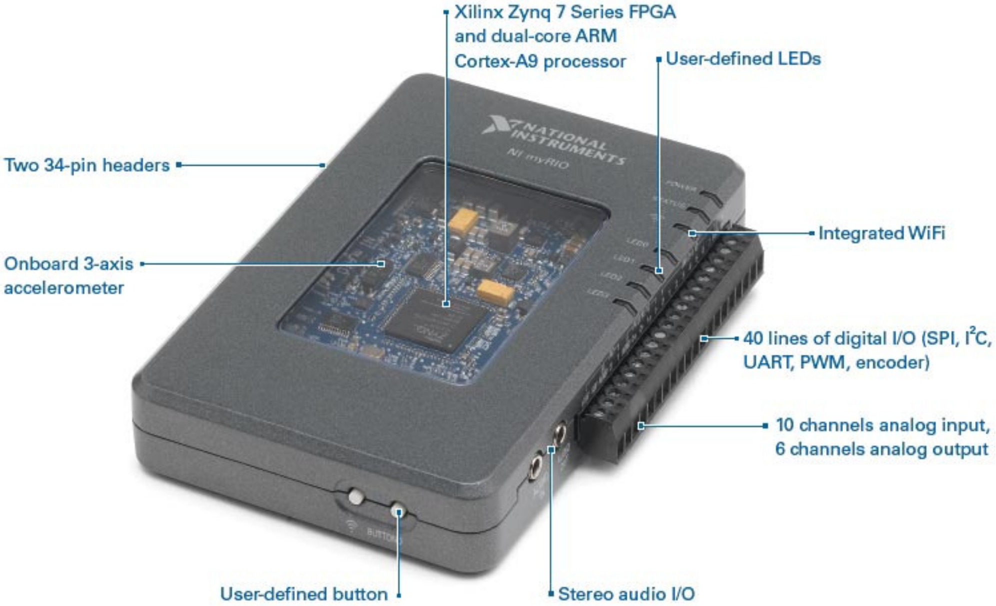


Figure 6.3: NI myRIO

When creating a program in a team, revision control is required so that no code is accidentally deleted or destroyed. We decided to use SVN for that purpose. Every change in the code is committed to the SVN server. Every commit is saved onto a new revision on the server allowing the code to be rolled back to any old revision.

6.3 Software Structure

The following chapter provides some insight into sepiOS (short for Sepios Operating System), our robot's final software. The general software structured is discussed, followed by an in-depth look at some of sepiOS' most important features: The interface, the communication between different parts of the program and our error handling strategy.

6.3.1 General Structure

Generally speaking, sepiOS can be separated into three distinct parts (Figure 6.4). The heart of the software is the real-time application running on the myRIO. It acts as a variable server, hosting all global variables used to communicated between the three sections. Simultaneously it manages all sensors and addresses the servomotors over a self-made I^2C-protocol. The application of attitude control to the user input and subsequent transformation into PWN values also occurs in this sub-program.

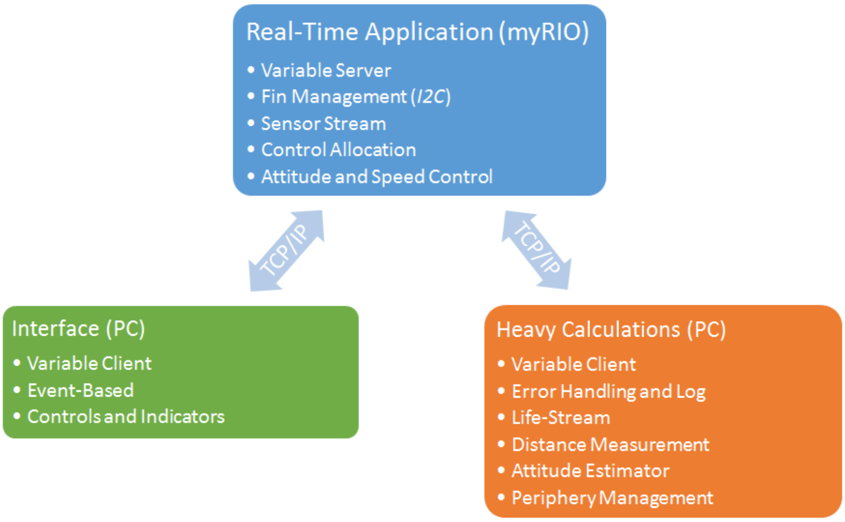
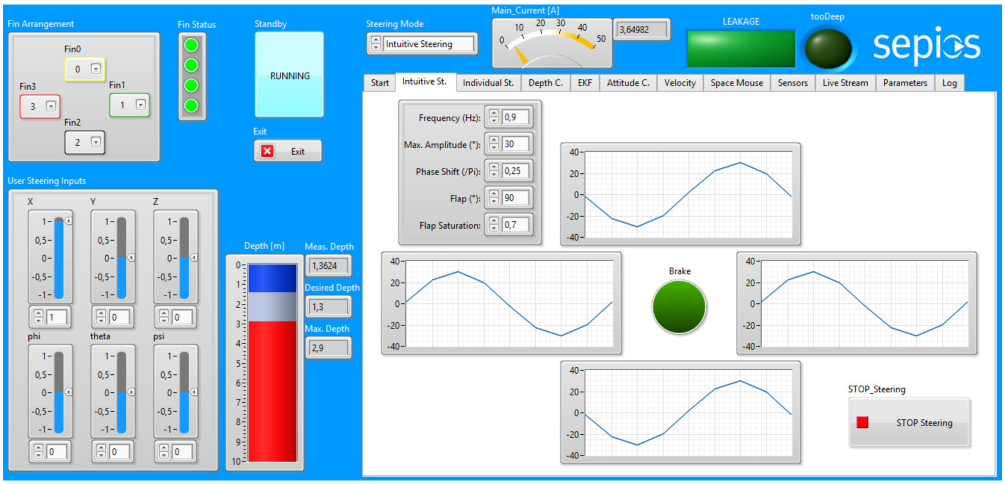


Figure 6.4: General software structure

The two other parts of sepiOS run on the interface computer and are both subscribed as variable clients to the real-time application.

One of these two sections is the interface. It regroups all the controls and indicators (Figure 6.5). The interface is designed to be intuitive for any novice user, allowing the robot to be started with the click of a single button. A more advanced user, will find a massive amount of settings, allowing him to customise everything. The interface is event-based, which means that network variables are only refreshed if they experience a change. This keeps the communication channels clear of many superfluous packets.



Frequency (Hz): 0,9
Max. Amplitude (°): 30
Phase Shift (/Pi): 0,25
Flap (°): 90
Flap Saturation: 0,7

Figure 6.5: The user interface of sepiOS

The second half of the interface computer's program performs all the processing power intensive calculations, handles errors and logs interesting data. This includes live-stream management, the vision based distance measurement (Chapter 7.6.1) and the attitude estimator (Chapter 7.4).

6.3.2 User Interaction

The primary source of user input is a 3Dconnexion SpaceMouse, a six axis input device. This was a logical choice for an omnidirectional robot.

Figure 6.6: Wireless SpaceMouse provided by 3Dconnexion

All other inputs are entered over the graphical user interface. This includes the "Steering Mode" where you can choose between three available options, the "User Steering Inputs" which allows direct force inputs, a standby button and an exit button. In the different tabs the user can change various parameters, including some for steering, depth control, attitude control and the space mouse. Additionally the user can enable or disable any periphery, such as sensors or the laser.

Besides the inputs, there are also several displays on the interface, the most important of them are always visible. These are: the depth, the main current, a leakage warning and an excessive depth warning. The tabs contain an attitude display, a tachometer, waveform charts of the PWN values, the filtered SpaceMouse input and the live-stream of the camera.

6.3.3 Steering Modes

Three different steering modes are implemented in sepiOS: "Intuitive Steering", which is the result of the pragmatic control allocation approach (Chapter 5.4.1), "Control Allocation" (Chapter 5.4.2) and "Individual Steering". This last mode serves as an undulating wave research platform. All parameters (frequency for each fin, amplitude, phase shift and zero position shift for each ray) can be modified in real-time or preprogrammed. This research mode also allows the replacement of conventional sine waves with any 2π-periodic function. For example, we implemented a "CuttleSine", derived from the movements of real cuttlefish. They often perform progressive waves with their fins (see Figure 6.2), which we approximated as follows:

$$
\begin{array}{l}
if\ (floor(mod(x/pi, 2)) = 0) \\
\qquad y = (sin(x) + 0.2 \cdot sin(2 \cdot x))/1.06868 \\
else \\
\qquad y = (sin(x) - 0.2 \cdot sin(2 \cdot x))/1.06868
\end{array}
$$

This shape (Figure 6.7) could reveal itself to be more efficient than a simple sine due to the steeper initial rise of the wave. Unfortunately we did not have the time to verify this.

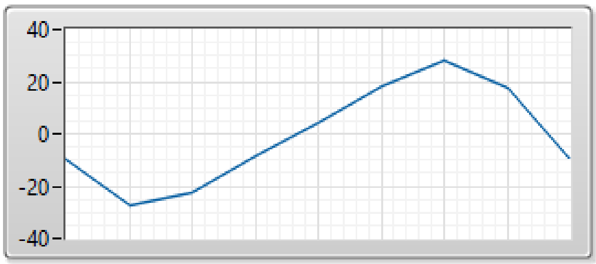


Figure 6.7: "CuttleSine" as represented on the interface

To conclude this chapter, Figure 6.8 shows the steering input flow through the entire sepiOS. A key feature of the flow is the so-called switchboard, which receives and dispatches the steering information in accord with the current status of the robot and the chosen modes.

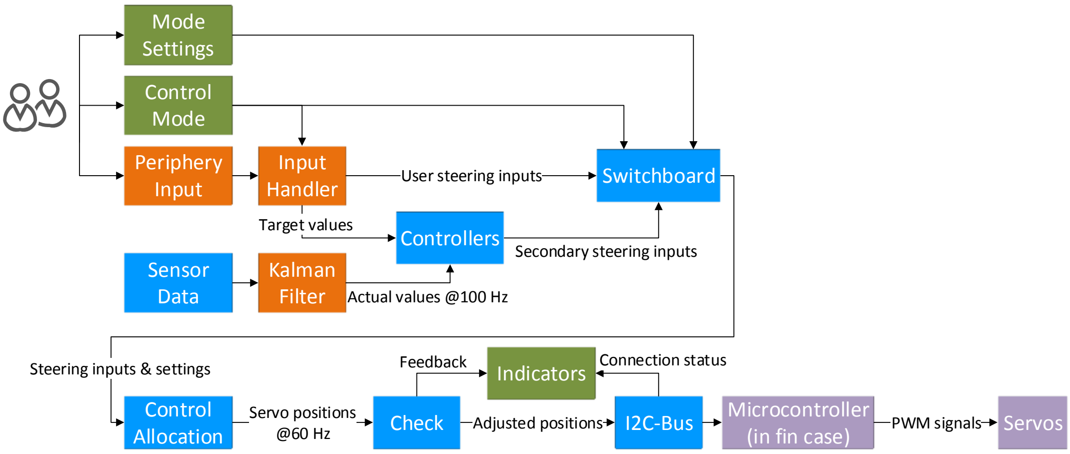


Figure 6.8: Steering input flow

6.3.4 Communication

The software had to use different approaches of inter-process communication, depending on how real-time we needed the participating tasks to be. All of the approaches used were already directly implemented into LabVIEW itself.

Data between processes running on the same device is exchanged using shared variables. They match the principle of shared memory in other programming languages. An example is the status of the main loop and the pattern on the ring LED indicator.

Data between the interface computer and the myRIO is exchanged via net-published shared variables. If written to on one device, they can be read on the other, and vice-versa. Examples include the steering inputs being sent to the myRIO, error transmission and the on/off switch of the swim bladder. If the data communication has to be real-time compatible, the real-time first-in-first-out (RT-FIFO) variant of shared variables has to be used. This allows for deterministic run-times as the variables are preallocated at a fixed size. RT-FIFOs automatically come with a buffer.

Low-level communication over Ethernet was handled by the Linux distribution running on the myRIO using a USB dongle. I^2C values are relayed through a built-in interface on the myRIO's FPGA.

Initially, due to our poor understanding of the host process of shared variables, the program produced non-deterministic errors. These could be fixed by slowing down loops which write into shared variables, to allow the values to propagate through the system.

6.3.5 Error Handling

Our software contains a sophisticated error handling strategy (Figure 6.9). Errors in LabVIEW consist, by default, out of a code, a string containing the location and a boolean flag. There is also a library containing and error message for every code. It is possible to implement custom errors unique to every system. As our program is fairly large and complex and consists of multiple loops running simultaneously, an advanced strategy was required. Wherever an error could occur we placed our own local error dispatcher. This VI interprets, prioritizes and classifies the incoming error and then writes it into a global RT-FIFO variable. It also takes any local precaution needed based on the error, such as stopping specific VIs. The local dispatcher also decides whether the error signal is cleared or passed on to subsequent VIs.

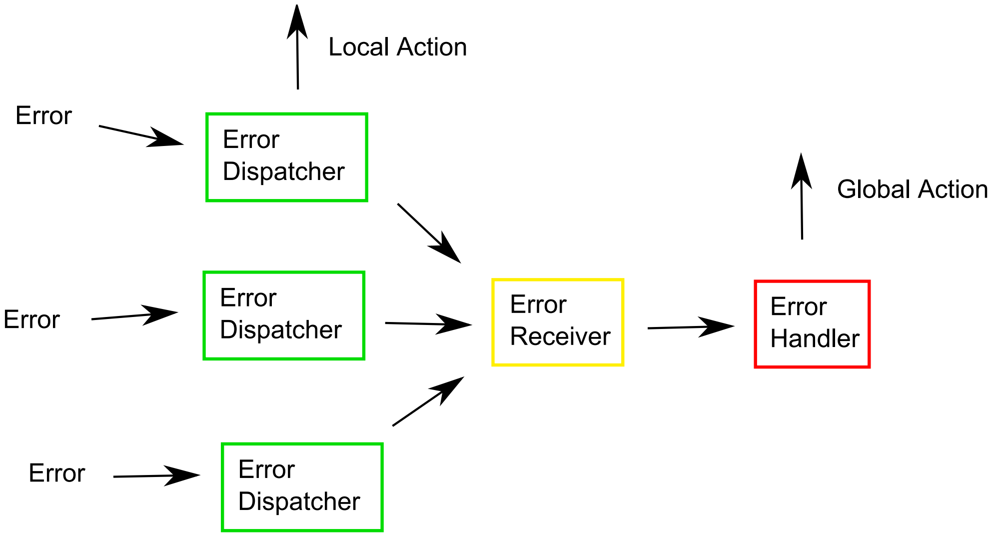


Figure 6.9: Error Handling

On the main interface loop, the error variable is constantly read by the error receiver. Its only purpose is clearing the variable and writing the values into a local array according to priority. This happens at a very high frequency in order to prevent buffer overflow of the error variable. The error handler VI then extracts the errors from the array and takes global actions. These include shutting down and restarting the system, displaying error messages and writing into an error log. The default action is logging the error.

Chapter 7

Evaluation

Upon completion of the robot, there was one more important task to tackle: the verification of our requirements (see Appendix 8.2). This chapter encompasses the most important experiments performed in order to accomplish this.

Contents

7.1 Omnidirectionality

The verification of our central goal, omnidirectionality, took precedence over all the other requirements. However, we soon realised that quantifying and thus measuring this attribute was not straightforward at all.

In the widest sense omnidirectionality can be understood as the ability to follow any trajectory in space. This definition does not specify the amount of time required for reaching the end of the path. In practice it is not possible to prove this ability as there are infinitely many possible trajectories. Therefore it would take an infinitely long time for a real object to follow all of them. So in order to decide whether our robot had reached the desired manoeuvrability we defined omnidirectionality in our list of requirements as follows:

> “robot can accelerate along all three main axes of its body fixed system of coordinates and it can rotate around them”

As this definition only describes a finite set of paths we were able to experimentally verify them. For this purpose we conducted an experiment in a swimming pool where we successfully performed all the described actions with our robot. Video footage showing these can be found on the attached DVD. Therefore we can state that we fulfilled our goal following the specification in our list of requirements.

However, as mentioned above, our list of requirements only contains a simplified definition of omnidirectionality which we could prove in practice. But we were also curious if the robot could succeed at performing more complicated manoeuvres as well. So we continued the experiment by superposing the basic manoeuvres from our list of requirements. Here we could follow a large number of trajectories and the intuitive approach for our control allocation exceeded our expectations. There were, however, a few combinations which we could not perform after two or three tries. Due to the limited time on that evening we skipped the verification of these. This does not mean that we are generally not able to perform the manoeuvre. We rather wanted to test a large number of them and therefore did not lose time if one did not work immediately.

Nevertheless we can state that there were two major problems which caused difficulties for certain manoeuvres. One was the stiff data cable which sometimes limited the agility of our robot. The other problem was connected to our intuitive approach for a control allocation. In this approach superposed movements turned out to be rather inefficient as explained in Chapter 5.4.1. We expect these inefficiencies to be eliminated with the control allocation developed by Flury and Möller [2014].

This thesis also brought up another, more theoretical, approach of proving omnidirectionality. The optimization used to distribute forces and torques among the fins has been solved for every possible combination of force and torque. For every direction (not for every magnitude) a solution has been found. This is the closest we can get to a mathematical proof of omnidirectionality. However, it has to be stated, that this approach uses a very simple model of the actual physical properties of the robot. This may lead to some discrepancy between the proof and the robots actual ability.

7.2 Acceleration and Velocity

Velocity measurement under water proved to be rather challenging. Because we were unable to turn off the pool's water jets, we had to take the fairly strong currents generated by them into account. Measurements over large distances were made impossible, as the robot would drift too much from its course.

The experimental set-up was very simple, but required many team members working in concert. One person had to operate the robot, one person had to record the measurements and one person had to perform the actual measurements using a stopwatch. Additionally a diver was required to steady the robot in place before each lap. The robot then moved in a straight line for twice 2 m and the elapsed time was noted.

We measured forward propulsion as well as side drift and rotation with different amplitudes, phase shifts and frequencies of the fins. The maximum velocity measured for forward propulsion is 0.56 m/s whereas for side drift it is 0.23 m/s. The maximum angular velocity measured for rolling is 32 °/s, for pitching it is 37 °/s and for yawing it is 40 °/s. All of these values are in the range necessary to comply with our requirements.

Sepios' acceleration to full speed is almost instantaneous and impossible to measure with our limited equipment.

7.3 Watertightness

As our main area of operation would be a conventional swimming pool of at most 5 m depth, we specified this as our maximum diving depth, with a hefty security factor of two. To evaluate the watertightness of our seals we used an autoclave pressure chamber supplied by the ETH. We inserted the assembled pieces of our robot and raised the pressure of the air to 1 *bar* relative to the surrounding. This corresponds a depth of 10 m in water. While in the autoclave, the servos were actuated to simulate dynamic load on the x-ring seals. Air has a higher tendency to diffuse through a seal than water due to the far lower surface tension. This meant, that if the system held at these conditions, it would be more than enough to satisfy our requirement.

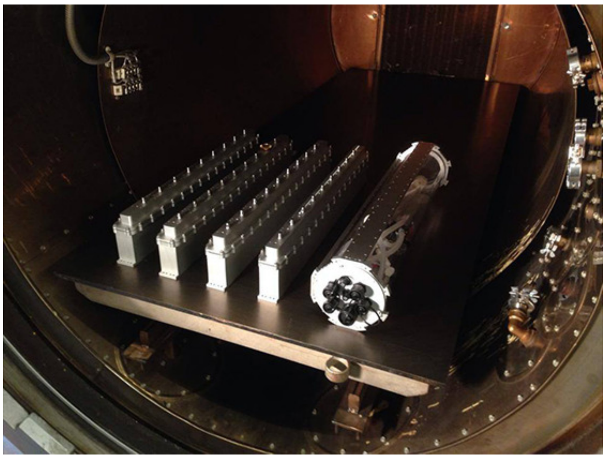


Figure 7.1: Dynamic test in the autoclave

Space inside the fin cases is sparse, therefore the pressure difference between the inside and outside could not be measured with an electronic sensor. Instead a makeshift pressure sensor consisting of a paper snippet placed inside a 10 *ml* syringe was used. A droplet of solder on the tip of the syringe guaranteed a constant pressure on the inside. Once the outside pressure increases, the plunger pushes the snippet forward, where it remains even when pressure recedes, thus indicating that a change in pressure has occurred. One such primitive sensor was placed in every fin case for the test.

Figure 7.2: Primitive pressure sensor using a syringe

All seals designed by us and produced by Kubo Tech AG passed the test. Initial problems involving the hose connecting the swim bladder to the outside were quickly solved by tightening the connection with binders.

The plugs which connect the base unit electronics to the fins servo controllers turned out to be the gravest source of problems. Because of us using a stiff and short cable, the side load exercised onto the cable seal was not uniform. At the point where the stress was lowest water was able to enter the cables. We temporarily fixed this by filling the cables with glue. Nevertheless, the water inside the cables caused problems. First it began to corrupt the data connection to the fins causing erratic behaviour of the servos. Eventually it also entered the fin cases and destroyed several servos. Finally the only option was to replace the plugs and increase the length of the cable. Currently we are waiting to test the new plugs. These will hopefully fulfil their task to our complete satisfaction.

Figure 7.3: Plugs for connecting the fins to our base unit

7.4 Alignment Precision

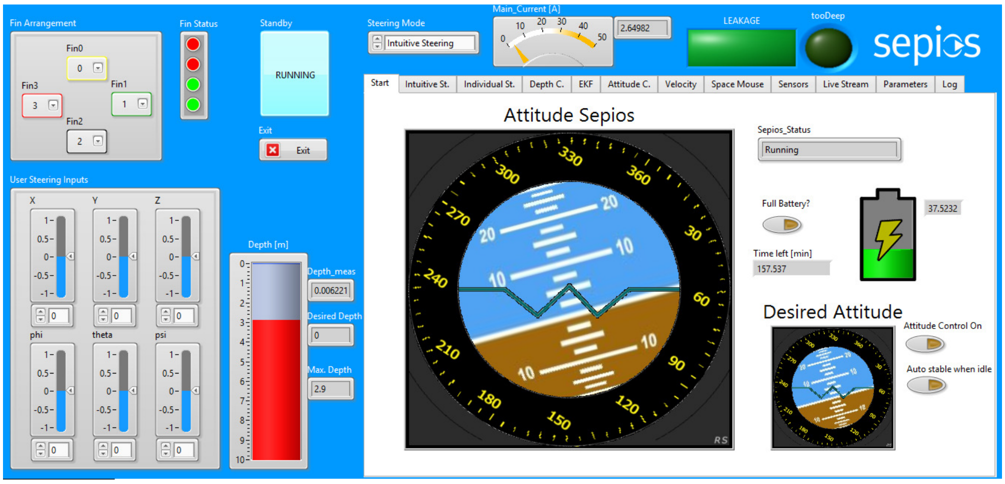
Having implemented the attitude estimator closely described in Pascal's and Alessandro's Bachelor's thesisAlessandro Schäppi [2014] we are able to directly read the actual roll, pitch and yaw values from the attitude display (Fig. 7.4). We are also able to plot these values after each dive to check on how the robots acceleration, rotation speed and the attitude behaved while submerged. All the different filter parameters can be adjusted directly on the interface to compensate for external circumstances such as magnetic disturbances or strong currents.

To verify whether our estimator worked properly we performed a test using a Vicon system. The Vicon is a state of the art position sensor, which detects an object by measuring laser reflections on special tags. Comparing this data to our estimated data, we were able calculate the following mean estimation errors:

- RMS-Error Roll: 1.1°
- RMS-Error Pitch: 1.4°
- RMS-Error Yaw: 1.8°

Unfortunately, because of magnetic disturbances from within the Base Unit, the magnetometer is unable to point to the absolute magnetic north. However it's relative yaw values can still be processed by the Kalman filter to improve the state estimate.

The implemented robots attitude controller is able to keep Sepios stable within a range of approximately 10 degrees of deviation towards the desired attitude. As the current sensor setup does not provide any position information when submerged (for example as GPS) only the attitude but not the position can be controlled autonomously.



Fin0: 0
Fin1: 1
Fin2: 2
Fin3: 3
Steering Mode: Intuitive Steering
Main_Current [A]: 2.64982
Sepios_Status: Running
37.5232
Time left [min]: 157.537
Depth_meas: 0.006221
Desired Depth: 0
Max. Depth: 2.9

Figure 7.4: Attitude display on the interface

7.5 Depth Measurement

As our pressure sensor is mounted on the back edge of the robot the depth measurement refers to this sensor position. However the sensor position is dependent on the actual attitude of the robot. As we are interested in the depth of the robots center of mass, the values from the pressure sensor have to be recalculated. To perform this calculation we use the attitude of the robot as provided by the Kalman filter. For example as shown in figure 7.5 on-spot positive pitching lowers the robots tail thus resulting in a (wrong) deeper depth measurement. By the knowledge of the pitch angle this measurement error can be compensated completely which results in a better depth estimation.
Combined with the accelerometer sensor and the mentioned Kalman filter we achieved a depth measurement accuracy of approximately 5 cm.

A more in-depth look into these calculations can be found in Pascal's and Alessandro's Bachelor's thesis. Alessandro Schäppi [2014]

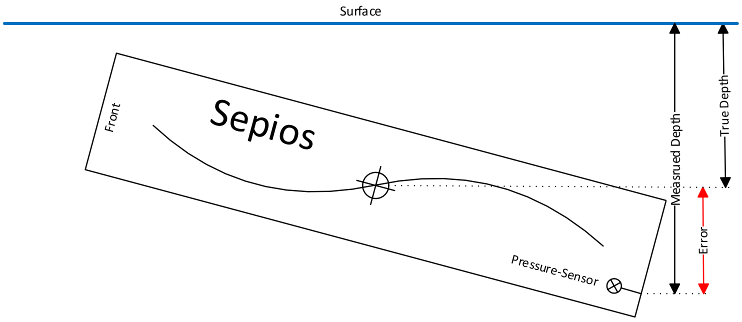


Figure 7.5: Attitude dependent depth measurement error due to offset sensor position

7.6 Security Mechanisms

As underlined in the risk analysis (section 2.11), a system as complex as the Sepios robot needs multiple security mechanisms to guarantee safe operation. Many of these measures were defined during the early design process (morphological box, section 2.7 and appendix F, and risk analysis, section 2.11), others were added along the way. The following subchapters describe the most important countermeasures.

7.6.1 Collision Detection and Avoidance

A nice safety feature of the Sepios robot is the pool wall collision detection and avoidance, which is part of the Bachelor's thesis "Vision Based Wall Detection for the Sepios Underwater Robot" (Seewer [2014]). Employing the green line laser and the on-board camera (section 4.2.6), Sepios is able to determine its distance and angle towards the wall it is facing. These data enable a basic autonomous swimming of Sepios, which was one of our wish requirements (see appendix 8.2, List of Requirements). For more information on this feature, please refer to the above mentioned Bachelor's thesis.

7.6.2 Leakage Sensor

To minimize the risk of parts being destroyed by water damage we have implemented three different types of leakage prevention on our robot.

As our initial model of servo motor was splash proof, we decided that no leakage detection was necessary in the Fin Cases. The estimated worst case, would have been a servo controller failing. This quickly alarms us through the failure of the fins electronics. Servo controllers are comparably cheap and easy to replace.

If little water enters the Base Unit it will not harm any of the electronics because of the cylindrical design. No matter the current attitude, gravity will always pull the water towards the wall, away from the electronics, which are mounted in the center of the cylinder.

Last but not least, we created our own leakage sensors. They consist of two separated wires woven into synthetic felt. Our board computer measures the voltage difference between the two wires. When water enters the Base Unit it is sucked into the felt, increasing its conductivity and thus decreasing the differential voltage. The rings are positioned at each end of the mainframe. The rings are aligned in two 3D printed rings at each of the two caps of the cylinder. The holders ensure that the rings are always tightly pressed onto the cylinder's wall in order to minimize the time of detection of penetrating water. These sensors have already saved us a lot of money in electronic parts.

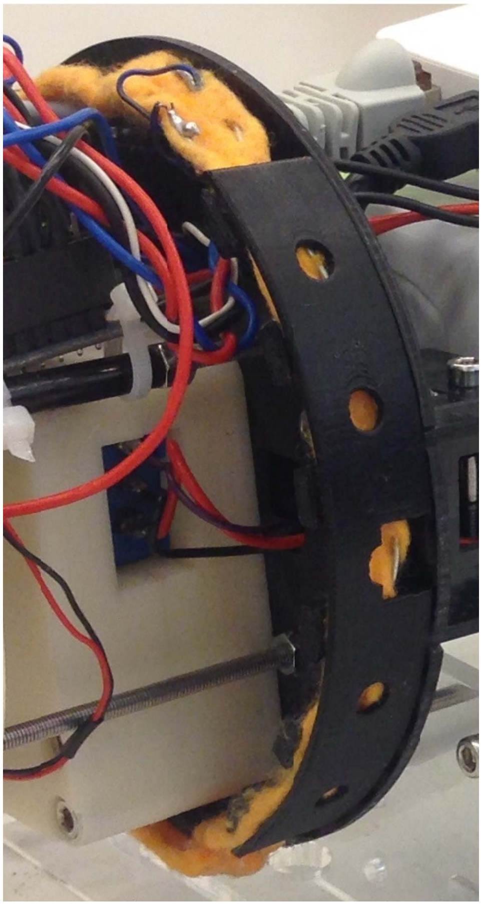


Figure 7.6: Leakage sensor (orange felt) built in the 3D printed holder

7.6.3 Low Voltage Protection

The low voltage protection prevents the battery from dropping below 6.6 V. The micro controller on the Connection Board continuously measures the battery voltage and turns off the secondary systems in case of low voltage. For further elaboration please refer to subsection 4.2.2. The system was tested with a laboratory power supply and works flawlessly.

7.6.4 Servo Overload Protection

Two distinct measures were taken to prevent an overload of the servomotors. The first one is the use of the servos own built-in overload protection. We programmed them to reduce their torque to 10% of their maximum in case of overload.

The second safety feature is a software check in sepiOS, where the PWM values are verified to be within a reasonable range before being sent to the motors. The software blocks any PWN values which are out of range, ensures that the "stretch angle" does not exceed the limit (see Figure 7.7) and coerces excessive movement speeds into reasonable ranges.

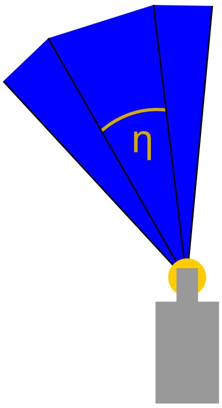


Figure 7.7: Stretch angle between rays

Chapter 8

Conclusion

Contents

8.1 Results

A nautical robot using four side fins inspired by cuttlefish was designed and built. The number and configuration of fins can be varied modularly. By performing undulating fin movements the robot is able to generate thrust underwater. Nine servomotors per fin provide the robot with a high manoeuvrability which enables it to move omnidirectionally. The fins themselves consist of elastic foils and rigid sticks driven by the servomotors. The servos are placed in waterproof cases which are connected to the base unit through supply cables.

All components are controlled by one single-board computer. This is where all user commands are processed before execution. All data and power distribution is managed in one, central, self-designed connection board. A swim bladder inside the base unit allows vertical diving without using the fins and holding the robot at a stable height. The robot can be steered manually using a SpaceMouse as is known from CAD. A full-HD camera simplifies navigation for the user by providing live-stream images. An outer polystyrene shell provides the robot with the necessary buoyancy.

So far, there have been successful dives using four- and two-finned configurations. Using only its fins the robot is able to cruise forwards and backwards, to drift sidewards and upwards and to roll, to pitch and to yaw by any angle. This fulfils one widespread definition of omnidirectionality. There are however some manoeuvres left which could not yet be performed. One major problem are the weight and stiffness of the data cable. Even a fairly thin cable already has a strong mechanical influence on the robots attitude. The manoeuvrability is still being improved with a control allocation approach based on measurements of the fin forces.

The mechanical components designed by team Sepios have proven to be adequately robust and the self-designed seals were watertight. Major problems concerning watertightness were encountered with industrial plugs. These plugs are used to connect the supply cables of the fins to the base unit. They were responsible for the majority of the leaks that occurred. A makeshift solution could be found to operate the robot without having water enter. For the long term a solution using plugs that are actually watertight is required.

Another important problem concerned the servomotors. They appeared to be too weak for the described application and gradually kept dying. Yet it is not clear whether the main reason for this is overheating or mechanical overload. The servos have been replaced with another type capable of applying three times the torque of the old ones. These servos are expected to last for a long time.

8.2 Outlook

The status of project Sepios as a Focus Project ends officially in summer 2014. The robot remains property of ETH Zürich. Several team members have already declared interest in continuing their work on the robot within the range of semester or Master's theses. During the brainstorming sessions in autumn possible gadgets and follow-up projects have been listed which could be realised within this context.

We are planning to present the robot on several exhibitions. We would also want to test it in more challenging terrain. If the new plugs are deemed waterproof, a test dive in the Mediterranean Sea will be attempted.

Sepios is still very far from any commercial applications. Currently no further prototypes are planned. Research on this prototype will hopefully continue for a long time and provide valuable insights into undulating fin propulsion.

Bibliography

Pascal Buholzer Alessandro Schäppi. Attitude Estimation and Control of the Sepios Robot. Technical report, 2014.

David Arbour. The Amazing Cuttlefish, 2013. URL `http://www.stories-for-children.ca/cuttlefish.php`.

Avon Lake Animal Clinic. Body plan of a cuttlefish, 2014. URL `http://www.avonlakeanimalclinic.com/exotics_search.php?category=&id=521`.

O M Curet, N A Patankar, G V Lauder, and M A MacIver. Mechanical properties of a bio-inspired robotic knifefish with an undulatory propulsor. *Bioinspiration & Biomimetics*, 6(2), 2011. ISSN 1748-3182. doi: Artn026004Doi10.1088/1748-3182/6/2/026004. URL `<GotoISI>://WOS:000291154500004`.

Fabio Dubois. *Development of a biologically-inspired strain gauge sensor for AUV velocity estimation*. Bachelor thesis, ETH Zürich, 2014.

EMPA Switzerland. EAP Technology @ Empa, 2013. URL `http://www.empa.ch/plugin/template/empa/*/72289`.

M Epstein, J E Colgate, and M A MacIver. Generating thrust with a biologically-inspired robotic ribbon fin. *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vols 1-12*, pages 2412–2417, 2006. doi: Doi10.1109/Iros.2006.281681. URL `<GotoISI>://WOS:000245452402088`.

Paolo Ermanni. Dimensionieren 1. 3(3):187, 2011.

ETH. No Title, 2014. URL `http://www.naro.ethz.ch/p2/index.html`.

ETH Zurich. ETH ASL Focus Project, 2014. URL `http://www.asl.ethz.ch/education/bachelor/focus`.

Julian Felix Flury and Martin Möller. Fin Force Identification and Control Allocation on the Sepios Robot. *Bachelor thesis ETH Zurich*, 2014.

Pierre-Paul Grassé. *Traité de Zoologie, Tome V, Fascicule 4: Céphalopodes*. P. Grassé, masson edition, 1997.

Ivo Boblan. Artificial Muscle, 2014. URL `http://www.boblan.de/artificial-muscle.html`.

C. Jordi, S. Michel, G. Kovacs, and P. Ermanni. Scaling of planar dielectric elastomer actuators in an agonist-antagonist configuration. *Sensors and Actuators A: Physical*, 161(1-2):2 , 6, June 2010. ISSN 09244247. doi: 10.1016/j.sna.2010.05.015. URL `http://linkinghub.elsevier.com/retrieve/pii/S0924424710002281`.

C Jordi, A Schmidt, G Kovacs, S Michel, and P Ermanni. Performance evaluation of cutting-edge dielectric elastomers for large-scale actuator applications. *Smart Materials and Structures*, 20(7): 1–10, July 2011. ISSN 0964-1726. doi: 10.1088/0964-1726/20/7/075003. URL `http://stacks.iop.org/0964-1726/20/i=7/a=075003?key=crossref.99504cb642cc6ff673a065f732fb4343`.

Gabor Kovacs and Lukas Düring. Contractive tension force stack actuator based on soft dielectric EAP. In *Proc. SPIE*, volume 7287, pages 72870A–72870A–15, 2009. doi: 10.1117/12.815195. URL `http://dx.doi.org/10.1117/12.815195`.

Gabor Kovacs, Empa Prof, and Paolo Ermanni. EAP stack actuator. *ETH Zürich*, page 15, 2008.

National Instruments. From Student to Engineer : Preparing Future Innovators With the NI LabVIEW RIO Architecture. pages 1–3, 2014.

Nova. Kings of camouflage, 2007. URL `http://www.pbs.org/wgbh/nova/camo/anat-nf.html`.

B. Peter, R. Ratnaweera, W. Fischer, C. Pradalier, and R.Y. Siegwart. Design and evaluation of a fin-based underwater propulsion system. *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, 2010.

Md. Mahbubar Rahman, Yasuyuki Toda, and Hiroshi Miki. Computational Study on a Squid-Like Underwater Robot with Two Undulating Side Fins. *Journal of Bionic Engineering*, 8(1):25–32, March 2011. ISSN 16726529. doi: 10.1016/S1672-6529(11)60003-6. URL `http://linkinghub.elsevier.com/retrieve/pii/S1672652911600036`.

Robot Watch. Squid Robot, 2006. URL `http://robot.watch.impress.co.jp/cda/static/image/2006/10/23/aqua0807.jpg`.

R Ruiz-Torres, O M Curet, G V Lauder, and M A MacIver. Kinematics of the ribbon fin in hovering and swimming of the electric ghost knifefish. *Journal of Experimental Biology*, 216(5):823–834, 2013. ISSN 0022-0949. doi: Doi10.1242/Jeb.076471. URL `<GotoISI>://WOS:000314883000017`.

Ryan Photographic. Teuthoidea - Squid, 2014. URL `http://www.ryanphotographic.com/squid.htm`.

Antoine Seewer. *Vision Based Wall Detection for the Sepios Underwater Robot.* Bachelor thesis, ETH Zürich, 2014.

S Sefati, I Neveln, M A MacIver, E S Fortune, and N J Cowan. Counter-propagating waves enhance maneuverability and stability: a bio-inspired strategy for robotic ribbon-fin propulsion. *2012 4th Ieee Ras & Embs International Conference on Biomedical Robotics and Biomechatronics (Biorob)*, pages 1620–1625, 2012. ISSN 2155-1782. URL `<GotoISI>://WOS:000313014600272`.

A A Shirgaonkar, O M Curet, N A Patankar, and M A MacIver. The hydrodynamics of ribbon-fin propulsion during impulsive motion. *Journal of Experimental Biology*, 211(21):3490–3503, 2008. ISSN 0022-0949. doi: Doi10.1242/Jeb.019224. URL `<GotoISI>://WOS:000260165100023`.

Tier im Fokus. Der Tintenfisch, 2014. URL `http://www.tier-im-fokus.ch/info-material/portraets/tintenfisch/`.

Yasuyuki Toda, Masashi Danno, K Sasajima, and H Miki. Model experiments on the squid-like under-water vehicle with two undulating side fins. *The 4th International Symposium on Aero ...*, pages 2–7, 2009. URL `http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Model+Experiments+on+the+Squid-Like+Under-Water+Vehicle+with+two+Undulating+Side+Fins#0`.

Markus Wegmann. Sepios: Nautical Robot, 2013. URL `http://sepios.org`.

Markus Wegmann. Sepios Facebook, 2014a. URL `https://www.facebook.com/sepiosETH`.

Markus Wegmann. Sepios GitHub, 2014b. URL `https://github.com/sepios`.

Markus Wegmann. Sepios YouTube, 2014c. URL `https://www.youtube.com/channel/UC0JrltNbSgAVg5zuUs7Oyqw`.

Wikipedia. Energy density, 2014a. URL `http://en.wikipedia.org/w/index.php?title=Energy_density&oldid=596036586`.

Wikipedia. Cuttlefish, 2014b. URL `http://en.wikipedia.org/wiki/Cuttlefish`.

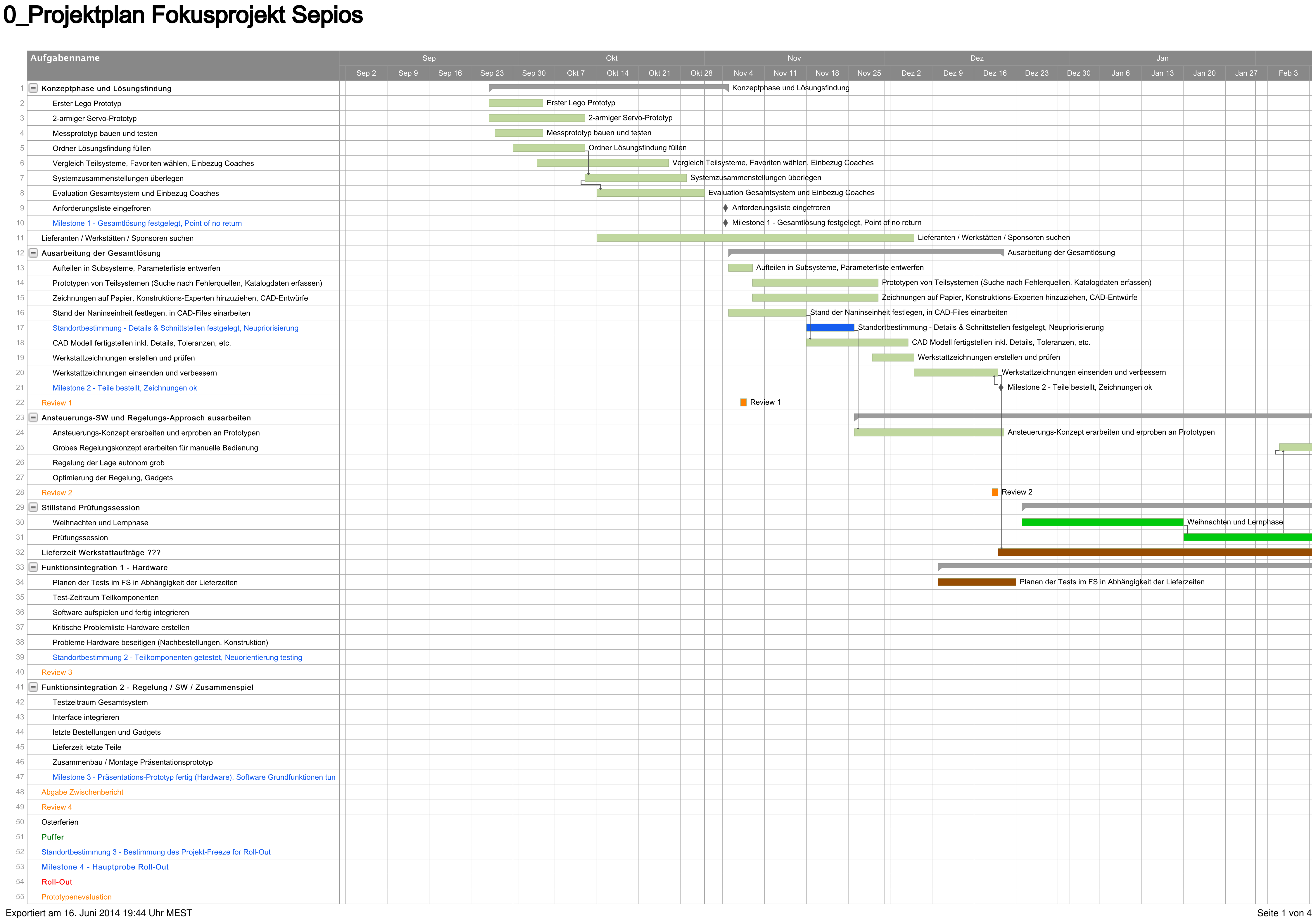
Wikipedia. IMU, 2014c. URL `http://en.wikipedia.org/wiki/Inertial_measurement_unit`.

List of Figures

List of Tables

Appendix A

Project and Subgroups Plan

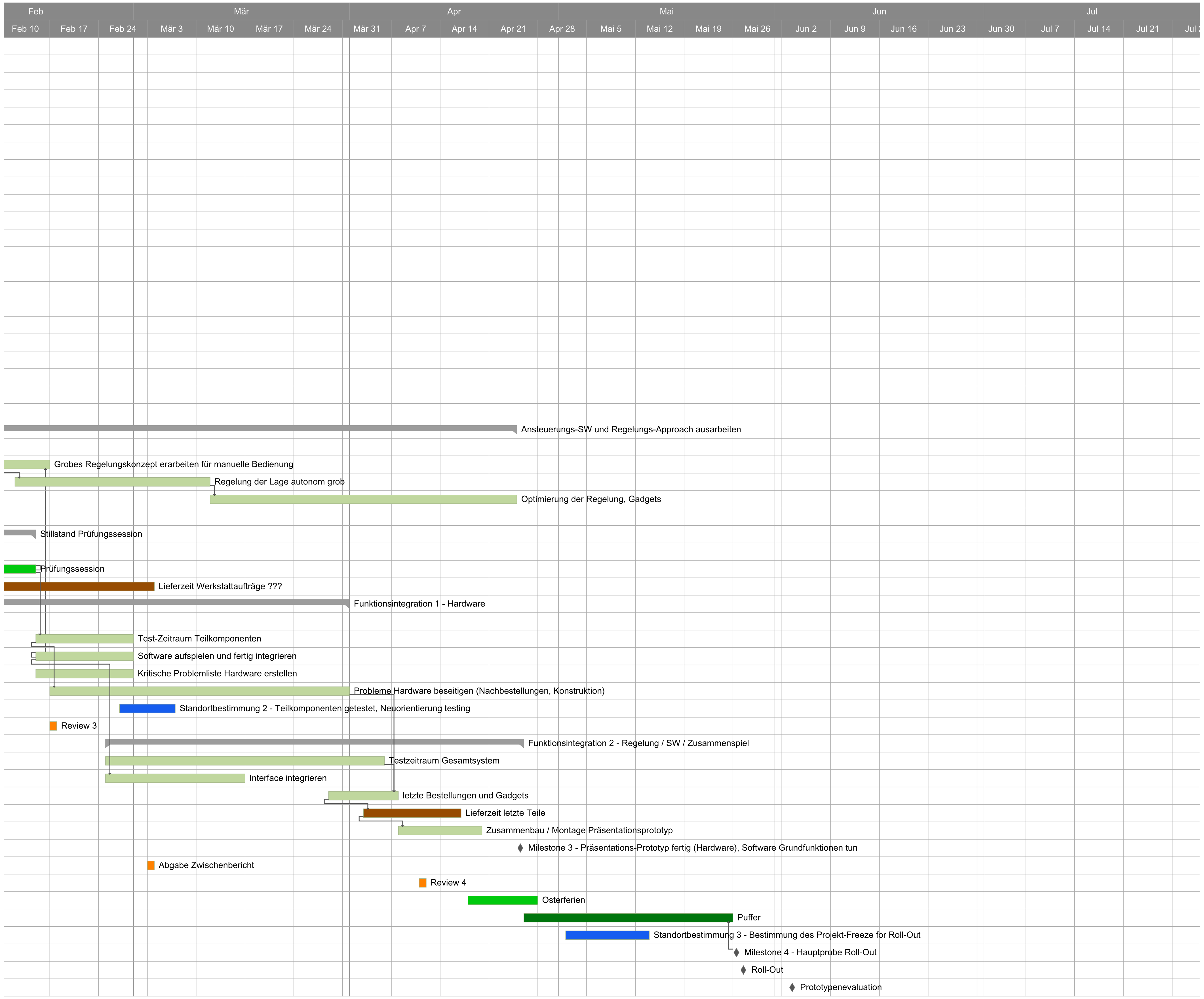


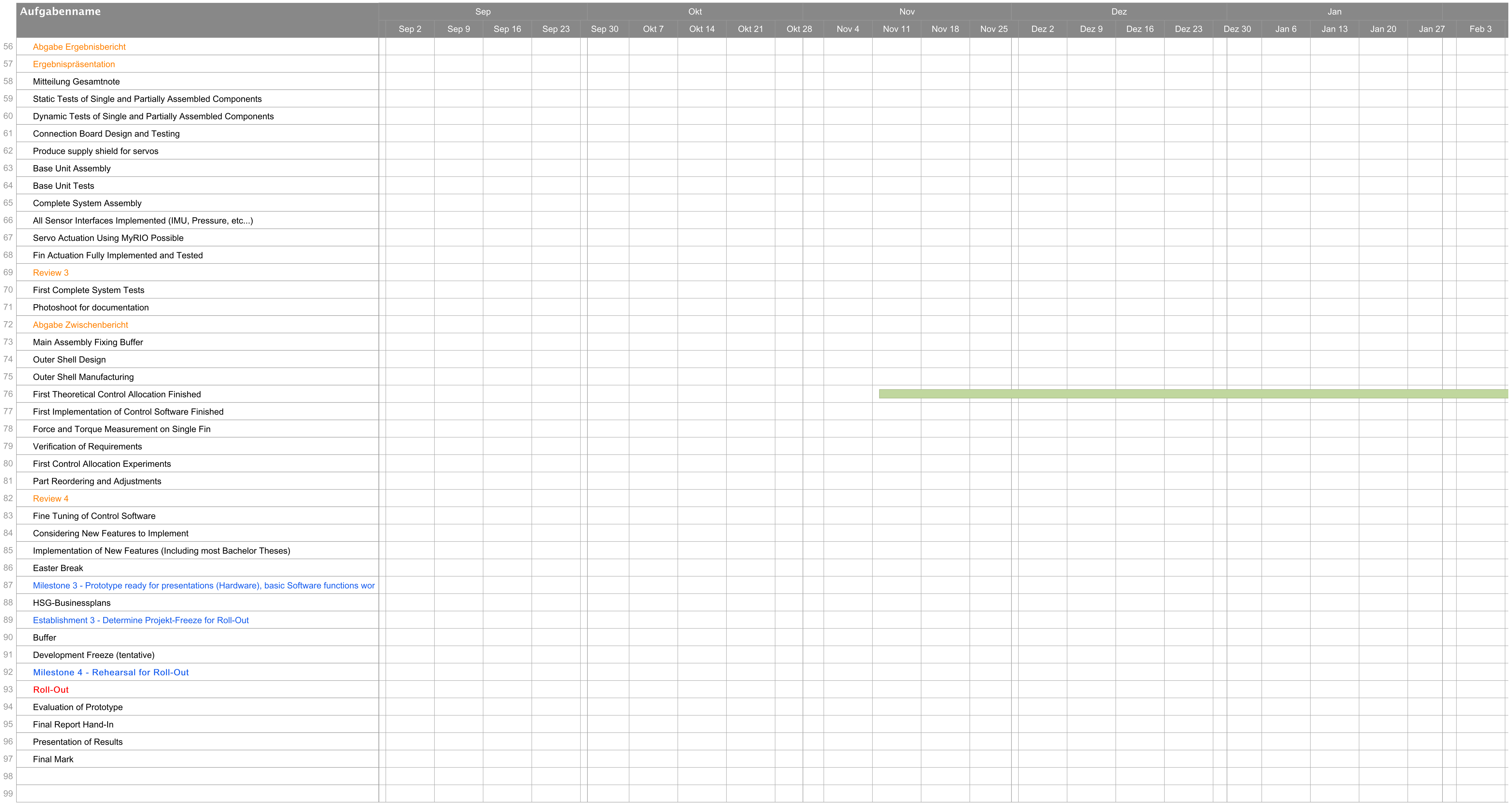
0_Projektplan Fokusprojekt Sepios

	Aufgabenname
1	**Konzeptphase und Lösungsfindung**
2	Erster Lego Prototyp
3	2-armiger Servo-Prototyp
4	Messprototyp bauen und testen
5	Ordner Lösungsfindung füllen
6	Vergleich Teilsysteme, Favoriten wählen, Einbezug Coaches
7	Systemzusammenstellungen überlegen
8	Evaluation Gesamtsystem und Einbezug Coaches
9	Anforderungsliste eingefroren
10	Milestone 1 - Gesamtlösung festgelegt, Point of no return
11	Lieferanten / Werkstätten / Sponsoren suchen
12	**Ausarbeitung der Gesamtlösung**
13	Aufteilen in Subsysteme, Parameterliste entwerfen
14	Prototypen von Teilsystemen (Suche nach Fehlerquellen, Katalogdaten erfassen)
15	Zeichnungen auf Papier, Konstruktions-Experten hinzuziehen, CAD-Entwürfe
16	Stand der Naninseinheit festlegen, in CAD-Files einarbeiten
17	Standortbestimmung - Details & Schnittstellen festgelegt, Neupriorisierung
18	CAD Modell fertigstellen inkl. Details, Toleranzen, etc.
19	Werkstattzeichnungen erstellen und prüfen
20	Werkstattzeichnungen einsenden und verbessern
21	Milestone 2 - Teile bestellt, Zeichnungen ok
22	Review 1
23	**Ansteuerungs-SW und Regelungs-Approach ausarbeiten**
24	Ansteuerungs-Konzept erarbeiten und erproben an Prototypen
25	Grobes Regelungskonzept erarbeiten für manuelle Bedienung
26	Regelung der Lage autonom grob
27	Optimierung der Regelung, Gadgets
28	Review 2
29	**Stillstand Prüfungssession**
30	Weihnachten und Lernphase
31	Prüfungssession
32	**Lieferzeit Werkstattaufträge ???**
33	**Funktionsintegration 1 - Hardware**
34	Planen der Tests im FS in Abhängigkeit der Lieferzeiten
35	Test-Zeitraum Teilkomponenten
36	Software aufspielen und fertig integrieren
37	Kritische Problemliste Hardware erstellen
38	Probleme Hardware beseitigen (Nachbestellungen, Konstruktion)
39	Standortbestimmung 2 - Teilkomponenten getestet, Neuorientierung testing
40	Review 3
41	**Funktionsintegration 2 - Regelung / SW / Zusammenspiel**
42	Testzeitraum Gesamtsystem
43	Interface integrieren
44	letzte Bestellungen und Gadgets
45	Lieferzeit letzte Teile
46	Zusammenbau / Montage Präsentationsprototyp
47	Milestone 3 - Präsentations-Prototyp fertig (Hardware), Software Grundfunktionen tun
48	Abgabe Zwischenbericht
49	Review 4
50	Osterferien
51	**Puffer**
52	Standortbestimmung 3 - Bestimmung des Projekt-Freeze for Roll-Out
53	**Milestone 4 - Hauptprobe Roll-Out**
54	**Roll-Out**
55	Prototypenevaluation

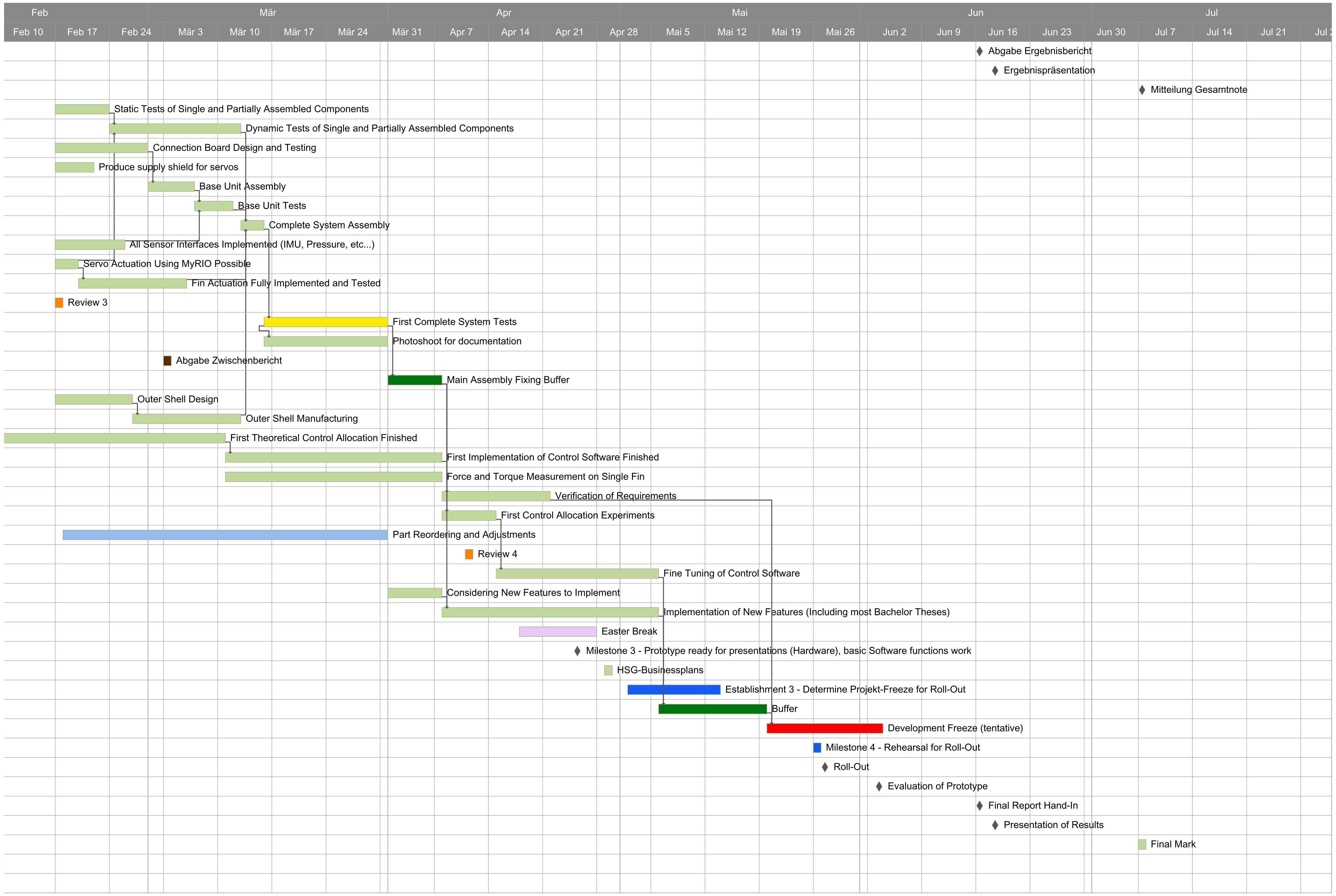
Exportiert am 16. Juni 2014 19:44 Uhr MEST

Seite 1 von 4


Feb
Mär
Apr
Mai
Jun
Jul
Feb 10
Feb 17
Feb 24
Mär 3
Mär 10
Mär 17
Mär 24
Mär 31
Apr 7
Apr 14
Apr 21
Apr 28
Mai 5
Mai 12
Mai 19
Mai 26
Jun 2
Jun 9
Jun 16
Jun 23
Jun 30
Jul 7
Jul 14
Jul 21
Jul 2
Ansteuerungs-SW und Regelungs-Approach ausarbeiten
Grobes Regelungskonzept erarbeiten für manuelle Bedienung
Regelung der Lage autonom grob
Optimierung der Regelung, Gadgets
Stillstand Prüfungssession
Prüfungssession
Lieferzeit Werkstattaufträge ???
Funktionsintegration 1 - Hardware
Test-Zeitraum Teilkomponenten
Software aufspielen und fertig integrieren
Kritische Problemliste Hardware erstellen
Probleme Hardware beseitigen (Nachbestellungen, Konstruktion)
Standortbestimmung 2 - Teilkomponenten getestet, Neuorientierung testing
Review 3
Funktionsintegration 2 - Regelung / SW / Zusammenspiel
Testzeitraum Gesamtsystem
Interface integrieren
letzte Bestellungen und Gadgets
Lieferzeit letzte Teile
Zusammenbau / Montage Präsentationsprototyp
Milestone 3 - Präsentations-Prototyp fertig (Hardware), Software Grundfunktionen tun
Abgabe Zwischenbericht
Review 4
Osterferien
Puffer
Standortbestimmung 3 - Bestimmung des Projekt-Freeze for Roll-Out
Milestone 4 - Hauptprobe Roll-Out
Roll-Out
Prototypenevaluation

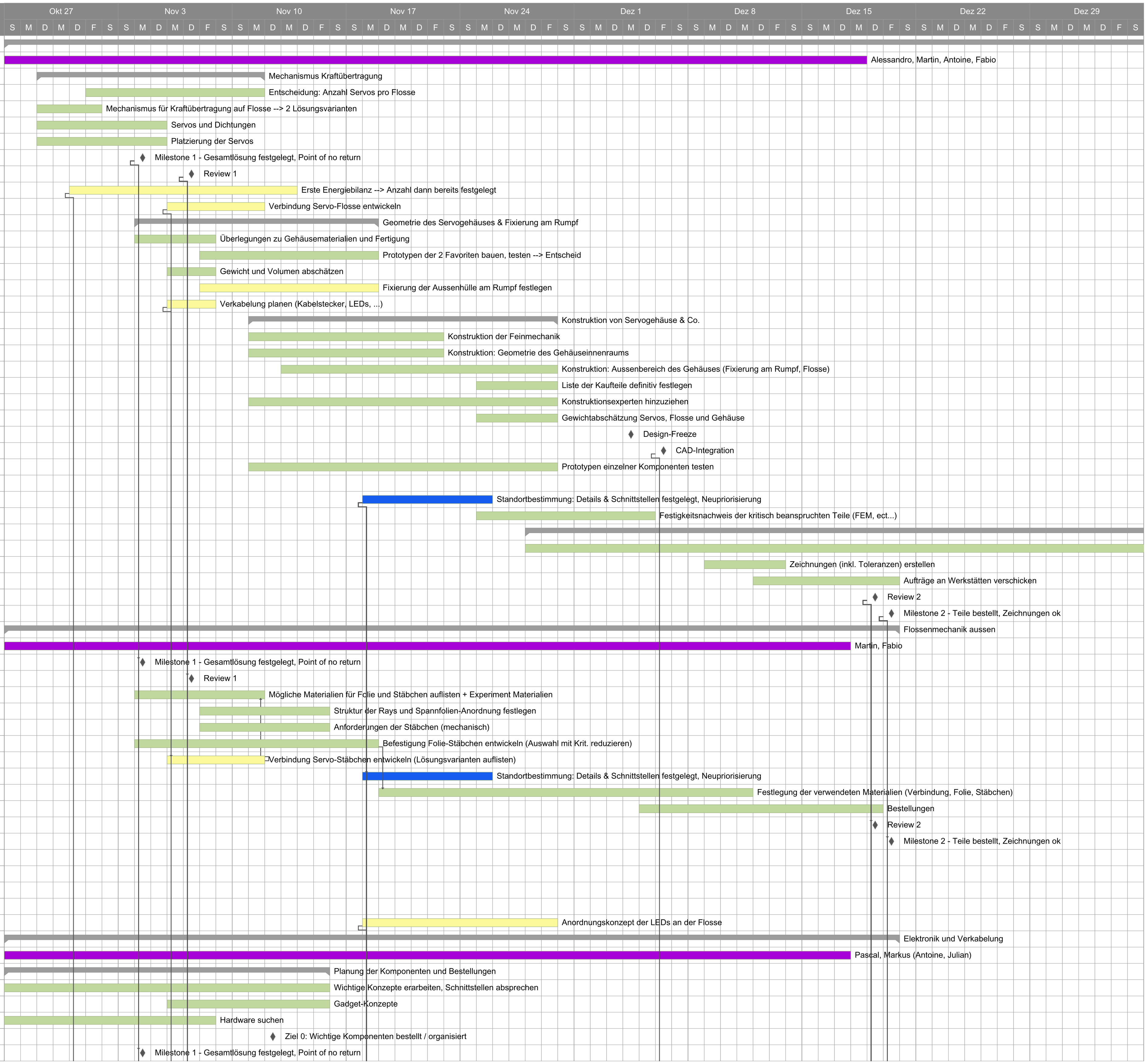


	Aufgabenname	Sep					Okt				Nov				Dez					Jan				
		Sep 2	Sep 9	Sep 16	Sep 23	Sep 30	Okt 7	Okt 14	Okt 21	Okt 28	Nov 4	Nov 11	Nov 18	Nov 25	Dez 2	Dez 9	Dez 16	Dez 23	Dez 30	Jan 6	Jan 13	Jan 20	Jan 27	Feb 3
56	Abgabe Ergebnisbericht																							
57	Ergebnispräsentation																							
58	Mitteilung Gesamtnote																							
59	Static Tests of Single and Partially Assembled Components																							
60	Dynamic Tests of Single and Partially Assembled Components																							
61	Connection Board Design and Testing																							
62	Produce supply shield for servos																							
63	Base Unit Assembly																							
64	Base Unit Tests																							
65	Complete System Assembly																							
66	All Sensor Interfaces Implemented (IMU, Pressure, etc...)																							
67	Servo Actuation Using MyRIO Possible																							
68	Fin Actuation Fully Implemented and Tested																							
69	Review 3																							
70	First Complete System Tests																							
71	Photoshoot for documentation																							
72	Abgabe Zwischenbericht																							
73	Main Assembly Fixing Buffer																							
74	Outer Shell Design																							
75	Outer Shell Manufacturing																							
76	First Theoretical Control Allocation Finished											🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
77	First Implementation of Control Software Finished																							
78	Force and Torque Measurement on Single Fin																							
79	Verification of Requirements																							
80	First Control Allocation Experiments																							
81	Part Reordering and Adjustments																							
82	Review 4																							
83	Fine Tuning of Control Software																							
84	Considering New Features to Implement																							
85	Implementation of New Features (Including most Bachelor Theses)																							
86	Easter Break																							
87	Milestone 3 - Prototype ready for presentations (Hardware), basic Software functions wor																							
88	HSG-Businessplans																							
89	Establishment 3 - Determine Projekt-Freeze for Roll-Out																							
90	Buffer																							
91	Development Freeze (tentative)																							
92	**Milestone 4 - Rehearsal for Roll-Out**																							
93	**Roll-Out**																							
94	Evaluation of Prototype																							
95	Final Report Hand-In																							
96	Presentation of Results																							
97	Final Mark																							
98																								
99																								

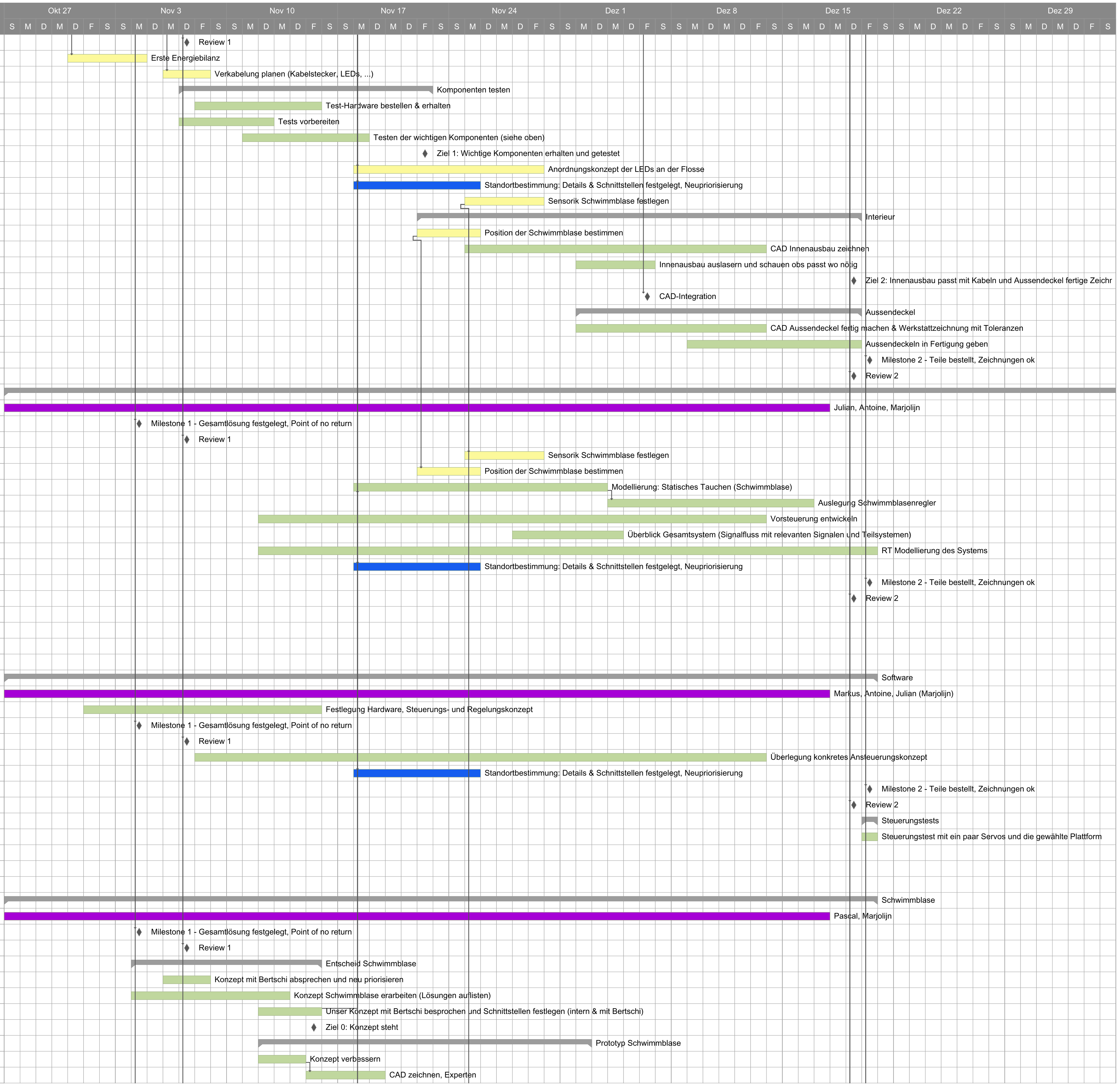

Feb
Mär
Apr
Mai
Jun
Jul
Feb 10
Feb 17
Feb 24
Mär 3
Mär 10
Mär 17
Mär 24
Mär 31
Apr 7
Apr 14
Apr 21
Apr 28
Mai 5
Mai 12
Mai 19
Mai 26
Jun 2
Jun 9
Jun 16
Jun 23
Jun 30
Jul 7
Jul 14
Jul 21
Abgabe Ergebnisbericht
Ergebnispräsentation
Mitteilung Gesamtnote
Static Tests of Single and Partially Assembled Components
Dynamic Tests of Single and Partially Assembled Components
Connection Board Design and Testing
Produce supply shield for servos
Base Unit Assembly
Base Unit Tests
Complete System Assembly
All Sensor Interfaces Implemented (IMU, Pressure, etc...)
Servo Actuation Using MyRIO Possible
Fin Actuation Fully Implemented and Tested
Review 3
First Complete System Tests
Photoshoot for documentation
Abgabe Zwischenbericht
Main Assembly Fixing Buffer
Outer Shell Design
Outer Shell Manufacturing
First Theoretical Control Allocation Finished
First Implementation of Control Software Finished
Force and Torque Measurement on Single Fin
Verification of Requirements
First Control Allocation Experiments
Part Reordering and Adjustments
Review 4
Fine Tuning of Control Software
Considering New Features to Implement
Implementation of New Features (Including most Bachelor Theses)
Easter Break
Milestone 3 - Prototype ready for presentations (Hardware), basic Software functions work
HSG-Businessplans
Establishment 3 - Determine Projekt-Freeze for Roll-Out
Buffer
Development Freeze (tentative)
Milestone 4 - Rehearsal for Roll-Out
Roll-Out
Evaluation of Prototype
Final Report Hand-In
Presentation of Results
Final Mark

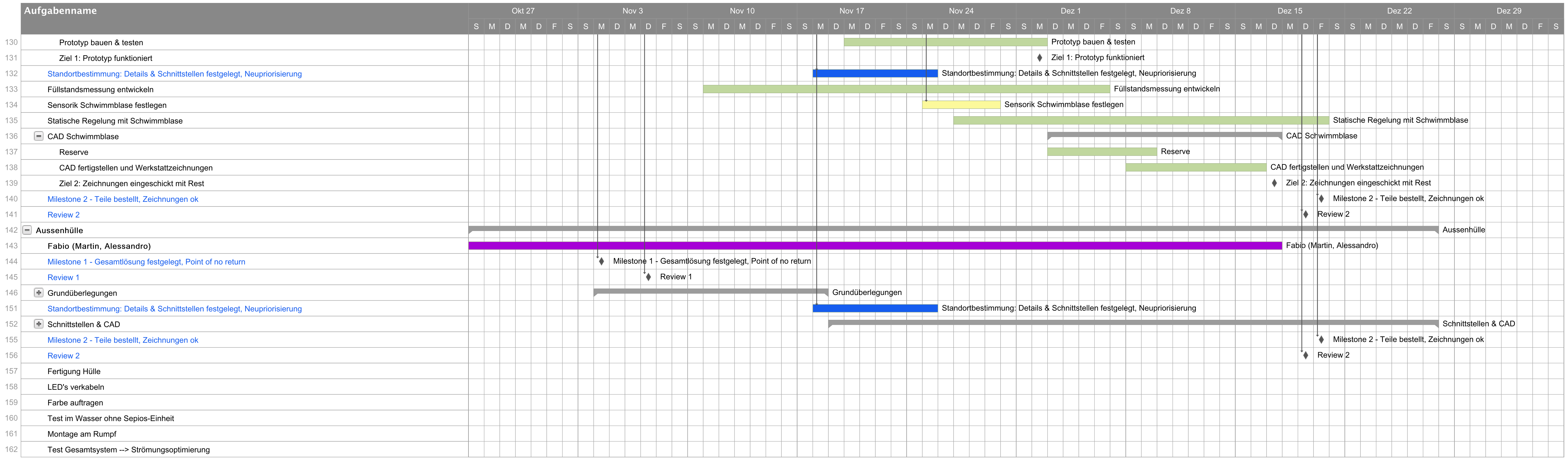
4_Zeitplan Subgroups

	Aufgabenname
1	**Flossenmechanik innen**
2	**Alessandro, Martin, Antoine, Fabio**
3	Mechanismus Kraftübertragung
4	Entscheidung: Anzahl Servos pro Flosse
5	Mechanismus für Kraftübertragung auf Flosse --> 2 Lösungsvarianten
6	Servos und Dichtungen
7	Platzierung der Servos
8	Milestone 1 - Gesamtlösung festgelegt, Point of no return
9	Review 1
10	Erste Energiebilanz --> Anzahl dann bereits festgelegt
11	Verbindung Servo-Flosse entwickeln
12	Geometrie des Servogehäuses & Fixierung am Rumpf
13	Überlegungen zu Gehäusematerialien und Fertigung
14	Prototypen der 2 Favoriten bauen, testen --> Entscheid
15	Gewicht und Volumen abschätzen
16	Fixierung der Aussenhülle am Rumpf festlegen
17	Verkabelung planen (Kabelstecker, LEDs, ...)
18	Konstruktion von Servogehäuse & Co.
19	Konstruktion der Feinmechanik
20	Konstruktion: Geometrie des Gehäuseinnenraums
21	Konstruktion: Aussenbereich des Gehäuses (Fixierung am Rumpf, Flosse)
22	Liste der Kaufteile definitiv festlegen
23	Konstruktionsexperten hinzuziehen
24	Gewichtabschätzung Servos, Flosse und Gehäuse
25	Design-Freeze
26	CAD-Integration
27	Prototypen einzelner Komponenten testen
28	3D-Print einer Kammer des Servogehäuses
29	Standortbestimmung: Details & Schnittstellen festgelegt, Neupriorisierung
30	Festigkeitsnachweis der kritisch beanspruchten Teile (FEM, ect...)
31	Zeichnungen erstellen und Aufträge verschicken
32	Sponsoren anschreiben und beraten lassen
33	Zeichnungen (inkl. Toleranzen) erstellen
34	Aufträge an Werkstätten verschicken
35	Review 2
36	Milestone 2 - Teile bestellt, Zeichnungen ok
37	**Flossenmechanik aussen**
38	**Martin, Fabio**
39	Milestone 1 - Gesamtlösung festgelegt, Point of no return
40	Review 1
41	Mögliche Materialien für Folie und Stäbchen auflisten + Experiment Materialien
42	Struktur der Rays und Spannfolien-Anordnung festlegen
43	Anforderungen der Stäbchen (mechanisch)
44	Befestigung Folie-Stäbchen entwickeln (Auswahl mit Krit. reduzieren)
45	Verbindung Servo-Stäbchen entwickeln (Lösungsvarianten auflisten)
46	Standortbestimmung: Details & Schnittstellen festgelegt, Neupriorisierung
47	Festlegung der verwendeten Materialien (Verbindung, Folie, Stäbchen)
48	Bestellungen
49	Review 2
50	Milestone 2 - Teile bestellt, Zeichnungen ok
51	Zusammenbau und Tests (z.B. mit Lego-Prototyp)
52	Montage an Sepios-Einheit
53	Tests und Vergleich
54	Evtl. Nachbestellungen
55	Anordnungskonzept der LEDs an der Flosse
56	**Elektronik und Verkabelung**
57	**Pascal, Markus (Antoine, Julian)**
58	Planung der Komponenten und Bestellungen
59	Wichtige Konzepte erarbeiten, Schnittstellen absprechen
60	Gadget-Konzepte
61	Hardware suchen
62	Ziel 0: Wichtige Komponenten bestellt / organisiert
63	Milestone 1 - Gesamtlösung festgelegt, Point of no return



	Aufgabenname
64	Review 1
65	Erste Energiebilanz
66	Verkabelung planen (Kabelstecker, LEDs, ...)
67	⊟ Komponenten testen
68	Test-Hardware bestellen & erhalten
69	Tests vorbereiten
70	Testen der wichtigen Komponenten (siehe oben)
71	Ziel 1: Wichtige Komponenten erhalten und getestet
72	Anordnungskonzept der LEDs an der Flosse
73	Standortbestimmung: Details & Schnittstellen festgelegt, Neupriorisierung
74	Sensorik Schwimmblase festlegen
75	⊟ Interieur
76	Position der Schwimmblase bestimmen
77	CAD Innenausbau zeichnen
78	Innenausbau auslasern und schauen obs passt wo nötig
79	Ziel 2: Innenausbau passt mit Kabeln und Aussendeckel fertige Zeichnung
80	CAD-Integration
81	⊟ Aussendeckel
82	CAD Aussendeckel fertig machen & Werkstattzeichnung mit Toleranzen
83	Aussendeckeln in Fertigung geben
84	Milestone 2 - Teile bestellt, Zeichnungen ok
85	Review 2
86	⊟ **Modellierung und Regelung**
87	**Julian, Antoine, Marjolijn**
88	Milestone 1 - Gesamtlösung festgelegt, Point of no return
89	Review 1
90	Sensorik Schwimmblase festlegen
91	Position der Schwimmblase bestimmen
92	Modellierung: Statisches Tauchen (Schwimmblase)
93	Auslegung Schwimmblasenregler
94	Vorsteuerung entwickeln
95	Überblick Gesamtsystem (Signalfluss mit relevanten Signalen und Teilsystemen)
96	RT Modellierung des Systems
97	Standortbestimmung: Details & Schnittstellen festgelegt, Neupriorisierung
98	Milestone 2 - Teile bestellt, Zeichnungen ok
99	Review 2
100	⊟ Auslegung und Implementierung Regler
101	Auslegung Regler
102	Implementierung Regler
103	Weitere Optimierung Regler
104	⊟ **Software**
105	**Markus, Antoine, Julian (Marjolijn)**
106	Festlegung Hardware, Steuerungs- und Regelungskonzept
107	Milestone 1 - Gesamtlösung festgelegt, Point of no return
108	Review 1
109	Überlegung konkretes Ansteuerungskonzept
110	Standortbestimmung: Details & Schnittstellen festgelegt, Neupriorisierung
111	Milestone 2 - Teile bestellt, Zeichnungen ok
112	Review 2
113	⊟ Steuerungstests
114	Steuerungstest mit ein paar Servos und die gewählte Plattform
115	Steuerung einer Flosse
116	Steuerung 2 Flossen
117	Steuerung 4 Flossen
118	⊟ **Schwimmblase**
119	**Pascal, Marjolijn**
120	Milestone 1 - Gesamtlösung festgelegt, Point of no return
121	Review 1
122	⊟ Entscheid Schwimmblase
123	Konzept mit Bertschi absprechen und neu priorisieren
124	Konzept Schwimmblase erarbeiten (Lösungen auflisten)
125	Unser Konzept mit Bertschi besprochen und Schnittstellen festlegen (intern & mit Bertschi)
126	Ziel 0: Konzept steht
127	⊟ Prototyp Schwimmblase
128	Konzept verbessern
129	CAD zeichnen, Experten





	Aufgabenname
130	Prototyp bauen & testen
131	Ziel 1: Prototyp funktioniert
132	Standortbestimmung: Details & Schnittstellen festgelegt, Neupriorisierung
133	Füllstandsmessung entwickeln
134	Sensorik Schwimmblase festlegen
135	Statische Regelung mit Schwimmblase
136	CAD Schwimmblase
137	Reserve
138	CAD fertigstellen und Werkstattzeichnungen
139	Ziel 2: Zeichnungen eingeschickt mit Rest
140	Milestone 2 - Teile bestellt, Zeichnungen ok
141	Review 2
142	**Aussenhülle**
143	**Fabio (Martin, Alessandro)**
144	Milestone 1 - Gesamtlösung festgelegt, Point of no return
145	Review 1
146	Grundüberlegungen
151	Standortbestimmung: Details & Schnittstellen festgelegt, Neupriorisierung
152	Schnittstellen & CAD
155	Milestone 2 - Teile bestellt, Zeichnungen ok
156	Review 2
157	Fertigung Hülle
158	LED's verkabeln
159	Farbe auftragen
160	Test im Wasser ohne Sepios-Einheit
161	Montage am Rumpf
162	Test Gesamtsystem --> Strömungsoptimierung

Appendix B

Cost

Sepios	HS 13 FS 14	Material cost
date	text	16'321.76
22.10.13	Ueberspace, Bastli, name.com	80.61
29.10.13	Interdiscount (USB Kabel, etc.)	160.4
29.10.13	Smartsheet; Back-Up License	151.63
30.10.13	SRLG Sektion Höngg	160
04.12.13	Ricardo, Conrad, Coop (Prototypenmaterial, Büromat.)	292.21
04.12.13	Playzone Adafruitchannel	31.9
04.12.13	ZVV Empa, SBB	29.8
04.12.13	Coop, Migros, SAB, Apropos (Büromaterial)	90.85
16.12.13	Mädler; Kegelräder	36.6
06.01.14	Zollgebühr Your Tong	36.8
24.12.13	Zollgebühr Thümer-Teile D-Hainichen	20.2
20.12.13	Play Zone; Adafruit 16-Channel Servo	92.6
20.12.13	Thümer Teile; Passcheiben € 52.0	63.96
09.01.14	Zoll	64.7
10.12.13	Kamera 2000 : Weitwinkel € 136.44	170.55
09.01.14	Zeko; Dual Motor Driver	42.6
11.02.13	Jakob Antriebstechnik; Distanzscheiben	84.9
24.01.14	MwSt Jakob € 53.0	20.2
21.01.14	Suter; Carbon Rundstäbe, Laminat, Harz, Härter	231.7
16.01.14	Mädler; Kegelrad	1246.45
21.01.14	Jakob; Kreuzschieberkupplung € 512.62	630.5226
28.01.14	Bossard Schrauben	91.48
28.01.14	ServoCity, Servokoppler	415.9
28.01.14	Multiplex 44 Servo mit 12 Verlängerungskabel	1473.25
21.02.14	Nozag ;Kegelrad Paar	140.03
25.02.14	Dv. Kleinmaterial für Sepios	204.07
06.02.14	Kundert; PC Glasklar	124.85
28.02.14	SBB HSG St. Gallen	34
26.02.14	Mädler; Kegelrad	19.7
26.02.14	Distrelec; Laborstecker	168.1
18.03.14	Supermagnete; Magnete	32.2
04.03.14	SAB-Shop Cash 2 Binden	8
05.03.14	Coop; Büroexpress; Apotheke (Spritzen)	25.9
05.03.14	Multiplex 1 Ersatz Servo	35.88
25.03.14	Farnell; Jumperkabel, Buchse; Kabel	60.85
25.03.14	Farnell; CMOS 15 V	8.1
25.03.14	Farnel; Wandler, Widerstand 100k	76.35
25.03.14	Suter; Arbeitspack Epoxy	145.15
31.03.14	Multi-CB; Leiterplatte	102
02.04.14	Abdeckklappe	183.66
04.04.14	Kleinmaterial und Autospesen Alumex Muri	233.2
25.03.14	Zoll Alexander Rub addiction	26.5
31.03.14	Mouser; Ohmite Current Sense	39.24
04.03.14	Werkstatt D-Phys: Werkstattnutzung und Alumaterial	4378.1
15.04.14	ABB	-500
18.03.14	Rub addiction,Latex € 13.44	16.58
15.03.14	Hasler und Rubaddiction	127.05
26.03.14	Vistaprint; Coop	228.9
26.03.14	Diverses	38.95
15.04.14	10 Servos (blau)	436.3302
30.04.14	Digitec und Jumbo, diverses	65.59
03.04.14	UPS Multi-PcB	28
24.04.14	Bauhaus Bastlerglas	10.3
09.05.14	Playzone Farnell; diverses	68.4

20.05.14	Conrad; Coop, diverses	342
20.05.14	Suter; Carbon Rundstab	30.75
27.05.14	Multiplex; 20 + 10 Servos und 12 Verlängerungskabel	1331.53
02.06.14	Playzone Channel, Printklemme	114.9
02.06.14	Franz Carl Weber und ARS Longa; Dekoration	30.65
02.06.14	Merci (für Sponsor) Freiermuth	8.6
02.06.14	ZVV Stäfa	24.8
03.06.14	Schutzklappe Distrelec	133.35
13.06.14	Ausstehend: 20 neue Servos von Multiplex	1100
13.01.14	Reserve (Noch unbezahlte Rechnungen) (Bastli, 3D Druck, …)	1000

Appendix C

List of Requirements

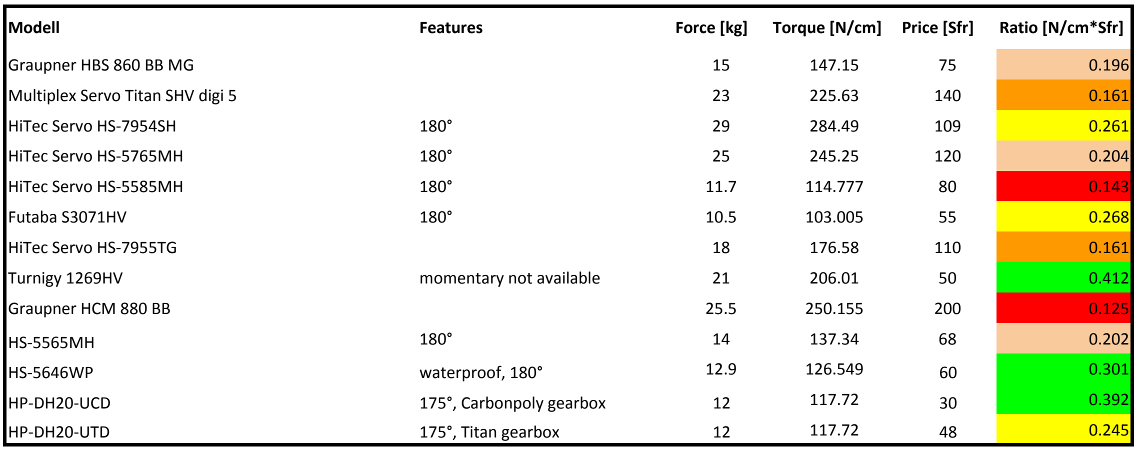
Nr.	Requirement	Category	Value	Type*	Explanation	Comments	Priority	Actual Value	Status	Comment	Legend
1	Intrinsically stable in the ventral position	generally	-	FF	Buoyancy point slightly above the center of massfor stabilization in the normal position	Above is clearly defined (e.g. coloured fin)	4		approved		FF: Obligation
2	Front is clearly visible	generally	-	FF	Front is clearly defined (e.g. coloured fin)		4		approved		mF: Minimum Value
4	omnidirectionality	locomotion	-	FF	robot can accelerate along all three main axes of its body fixed system of coordinates and it can rotate around them		4		approved		MF: Maximum Value
3	Optimize flow resistance	design	cW < 0.45	FF	CW-value is smaller than for a sphere with the same cross-sectional area	Sphere (Re < 1.7 E^5) Penguin: 0.03	4		not tested		W: Wish
5	robust control software	interaction	-	FF	Can handle connection loss		4		see comment	Not yet implemented, currently not approved but easy implementation	
7	Status feedback to ROV interface	interaction	-	FF	Battery level, all sensor data, alarms		4		approved		
8	depth measurement	sensors	< 10 cm	FF	current depth is measured with a maximum error of 10cm	Can be realized with filter and existing sensor	4	5cm	approved		
10	suitable for use in indoor swimming pool	fabrication	-	FF	easily washable, resistant to corrosion in the pool		4		approved		
12	Emergency mechanism	ability	-	FF	rope can be attached to the robot		4		approved		
9	attitude measurement	sensors	< 20°/h	FF	software estimates attitude. the error of the estimated angular coordinates grows with less than 20°/h	Necessary for the regulation of the attitude detectable by eye	4	0°/h	approved		
11	modularity	design	-	FF	All the fins share the same geometry and can be interchanged arbitrarily Electronics and software are designed in modular principle -> simple extension		4		approved		
6	Status directly recognizable on ROV	interaction	-	FF		LEDs	4		approved		
13	operation depth	generally	5m	mF	Normal maximal operation depth		4		approved		
14	Water resistant to depth	generally	10 m	mF	This guarantees a security factor of 2 when the maximum operation depth is 5m. The maximum absolute pressure is therefore 2 bar (1bar relative to the atmospheric pressure)		4		approved		
15	Operating time at full load	generally	25min	mF	At maximum thrust in the main direction of travel	Also consider temperature of the actuators	4	60 min	approved		
16	acceleration time	locomotion	4s	MF	Time in which the system reaches the usual traveling speed from the rest position	Along the main axis of locomotion	4		approved		
17	Deceleration time	locomotion	2s	MF	Time in which the system decelerates from its top speed to zero		4		approved		
18	cruising speed	locomotion	0,5 m/s	mF	Speed of the drone in standing water	Along the main axis of locomotion	4	0,566 m/s	approved		
19	drift velocity	locomotion	0,1 m/s	mF	Speed perpendicular to the main direction of travel	Along the lateral axis	4	0,18 m/s	approved		
20	Vertical diving speed	locomotion	0,1 m/s	mF	vertical diving when horizontally aligned (no translation in other directions)		4		not tested		
21	Angular rate roll axis	locomotion	20 grad/s	mF	Angular velocity , the system rotates about its rotation axis		4	31,5 °/s	approved		
22	Angular rate pitch-axis	locomotion	15 grad/s	mF	Angular velocity , the system rotates about its rotation axis		4	40,4 °/s	approved		
23	Angular rate yaw-axis	locomotion	10 grad/s	mF	Angular velocity , the system is rotated about its vertical axis		4	40 °/s	approved		
24	Max weight	design	20 kg	MF		Up to two persons necessary for carrying the robot	3	22,7 kg	not approved		
25	Robot can be recharged without being opened	design	-	W	Charging socket for charging with external balancer / charger		4		approved		
26	Easy On / Off Switch	design	-	W	Reed switch for main power line - > On / Off by magnet proximity		4		approved		
27	Update and / or flashable when submerged	interaction	-	W	If surfaced		4		approved		

Nr.	Requirement	Category	Value	Type*	Explanation	Comments	Priority	Actual Value	Status	Comment	Legend
28	Maximum attitude deviation without user input	navigation	10 grad	W	Maximum deviation from the nominal attitude in standing water		4		not tested		
29	maximum compass deviation	sensors	< 10 grad	W	3D compass that works even when system is tilted		4		see comment	only with gyro + kalman filter support	
30	noise	generally	> 10 kHz < 60 dB	W	Measurement is done ashore at a distance of 1 meter	Normal speaking intensity -> 60 dB Measurements on land with QAM	3		not tested		
31	Handling / transportability	design	-	W	In a suitcase / carrying box and stowed in pairs portable	A large rolling suitcase constitutes everything necessary for operation on a show	3		approved		
32	Wireless connection between the remote control and drone	interaction	7.5m, 1 KBit/s	W	duplex communication	Minimum distance 7.5 m	3		not approved		
33	Autonomy / semi- autonomously in certain scenarios	navigation	-	W	e.g. predefined manoeuvres (collision avoidance, circle, square , triangle, star, loops, rolls , etc.)	Important for demonstration purposes , triggered by user or software	3		not approved		
34	Leak alarm	sensors	-	W	Water or moisture sensor inside : warning and automatic surfacing if operator does not respond.		3		approved		
35	mechanical robustness	generally	-	W	Robot can stand frontal crash into a wall with half maximum speed five times. Verify using FEM or replacement test	Possible additional categories for robustness at land (drop test , etc.) and robustness for example, Side impact , frontal impact , excessive depth , wave height of surface	2		not tested		
36	charging time	generally	<1h	W			2	59 min	approved		
37	Camera with electric tripod	generally	-	W	e.g. from modelbuilding shop		2		not approved		
38	Child-safe operation	design	-	W	A 14- year-old child can control the ROV		2		approved		
39	attractive design	design	-	W	natural appearance		2		approved		
40	Emergency mechanism (fail-safe)	ability	-	W	Surface robot after an extended inactivity, software- or hardware problems	Can also be activated externally	2		not approved		
41	Does not entangle	ability	-	W	Drone does not wravel up in algae or sea grass	Test in the lake with sea plants	2		not tested		
42	Crash warning	interaction	4s	W	Warning to user interface if an object at the current speed is closer than 4s (only in main direction of travel from the front)	ultrasound	2		not tested		
43	Control App (smartphone or web)	interaction	-	W		demos	2		not approved		
44	location	navigation	-	W	Drone determines its own position in space (coordinate feedback)	In swimming pools easier than in the lake	2		not approved		
45	Interior temperature measurement	sensors	< 1° Deviation	W	security feature		2		see comment	measuring IMU, ~30° hotter	
46	biocompatibility	design	2 Meter Distance	W	System integrated in zoological environment , does not scare fish at distances >2m		1		not tested		
47	drone swarm network	ability	-	W			1		not approved		
48	External light source	ability	-	W	controlled via interface For shooting at greater depths, in hideouts and caves	Intensity is a question of price; the brighter the better	1		not approved		
49	object Recognition	ability	95%	W	The drone detects for example fish and is able to follow them		1		not approved		
50	fast / broadband data transmission (underwater)	interaction	> 20 Mbit/s	W	Via Ethernet Cable	Live Video	1	100 Mbit/s	approved		
51	fixation system	navigation	-	W	Drone can attach itself to a predefined/arbitrary location	Must decide whether only pre- docking or any suitable surface	1		not approved		
52	mapping	navigation	-	W	System developes spatial map (2D/3D) of its surroundings	For instance map of the sea bed or model of a ship wreck. ASL multi -floor mapping	1		not approved		
53	Water temperature measurement	sensors	< 1° Deviation	W	Water temperature measurement		1		not approved		

Appendix D

Servo Selection Table

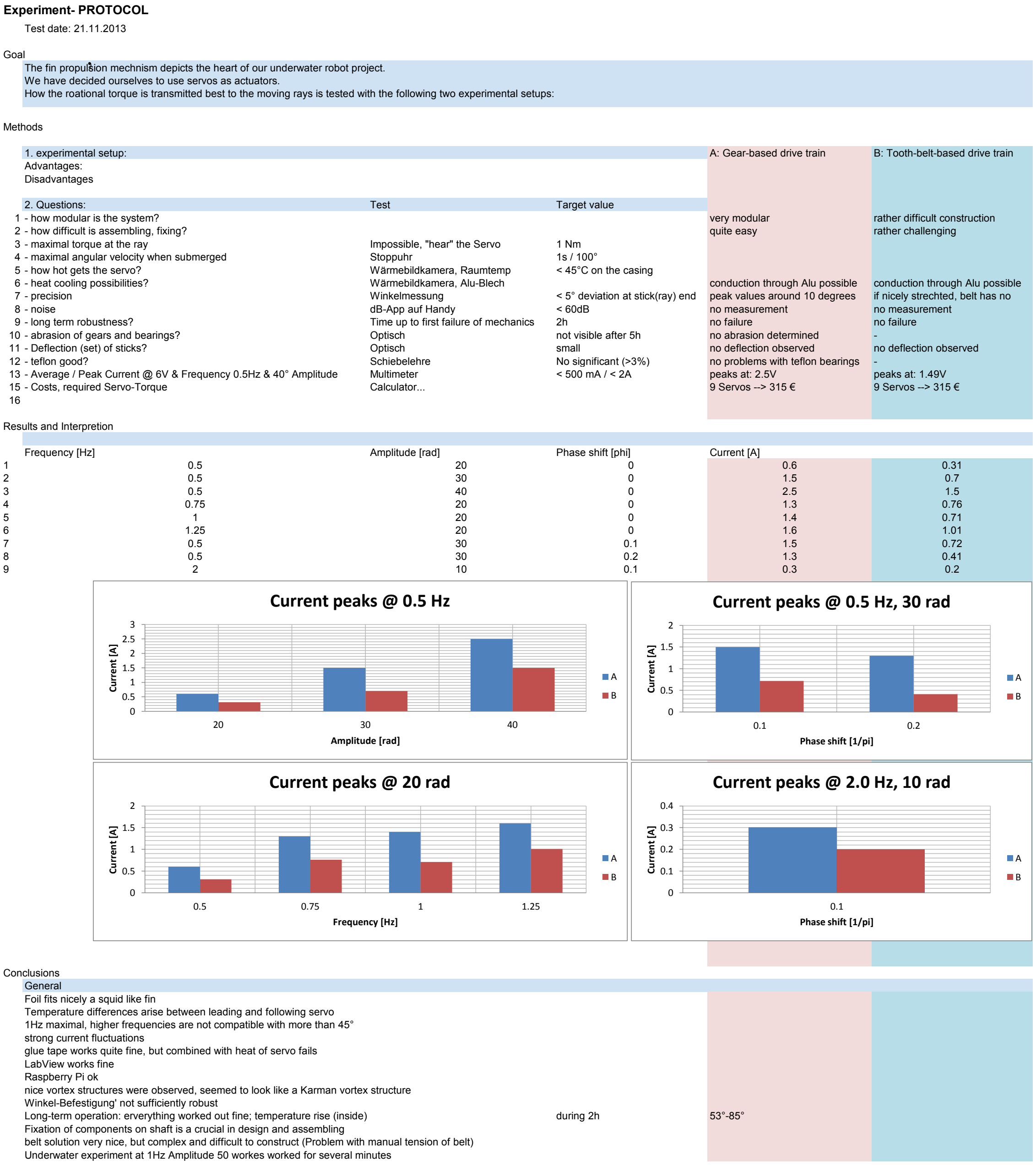
,



Modell	Features	Force [kg]	Torque [N/cm]	Price [Sfr]	Ratio [N/cm*Sfr]
Graupner HBS 860 BB MG		15	147.15	75	0.196
Multiplex Servo Titan SHV digi 5		23	225.63	140	0.161
HiTec Servo HS-7954SH	180°	29	284.49	109	0.261
HiTec Servo HS-5765MH	180°	25	245.25	120	0.204
HiTec Servo HS-5585MH	180°	11.7	114.777	80	0.143
Futaba S3071HV	180°	10.5	103.005	55	0.268
HiTec Servo HS-7955TG		18	176.58	110	0.161
Turnigy 1269HV	momentary not available	21	206.01	50	0.412
Graupner HCM 880 BB		25.5	250.155	200	0.125
HS-5565MH	180°	14	137.34	68	0.202
HS-5646WP	waterproof, 180°	12.9	126.549	60	0.301
HP-DH20-UCD	175°, Carbonpoly gearbox	12	117.72	30	0.392
HP-DH20-UTD	175°, Titan gearbox	12	117.72	48	0.245

Appendix E

Protocol of Servo Testing



Experiment- PROTOCOL

Test date: 21.11.2013

Goal

The fin propulsion mechnism depicts the heart of our underwater robot project.
We have decided ourselves to use servos as actuators.
How the roational torque is transmitted best to the moving rays is tested with the following two experimental setups:

Methods

			A: Gear-based drive train	B: Tooth-belt-based drive train
1. experimental setup:				
Advantages:				
Disadvantages				
2. Questions:	Test	Target value		
1 - how modular is the system?			very modular	rather difficult construction
2 - how difficult is assembling, fixing?			quite easy	rather challenging
3 - maximal torque at the ray	Impossible, "hear" the Servo	1 Nm		
4 - maximal angular velocity when submerged	Stoppuhr	1s / 100°		
5 - how hot gets the servo?	Wärmebildkamera, Raumtemp	< 45°C on the casing		
6 - heat cooling possibilities?	Wärmebildkamera, Alu-Blech		conduction through Alu possible	conduction through Alu possible
7 - precision	Winkelmessung	< 5° deviation at stick(ray) end	peak values around 10 degrees	if nicely strechted, belt has no
8 - noise	dB-App auf Handy	< 60dB	no measurement	no measurement
9 - long term robustness?	Time up to first failure of mechanics	2h	no failure	no failure
10 - abrasion of gears and bearings?	Optisch	not visible after 5h	no abrasion determined	-
11 - Deflection (set) of sticks?	Optisch	small	no deflection observed	no deflection observed
12 - teflon good?	Schiebelehre	No significant (>3%)	no problems with teflon bearings	-
13 - Average / Peak Current @ 6V & Frequency 0.5Hz & 40° Amplitude	Multimeter	< 500 mA / < 2A	peaks at: 2.5V	peaks at: 1.49V
15 - Costs, required Servo-Torque	Calculator...		9 Servos --> 315 €	9 Servos --> 315 €
16				

Results and Interpretion

	Frequency [Hz]	Amplitude [rad]	Phase shift [phi]	Current [A]	
1	0.5	20	0	0.6	0.31
2	0.5	30	0	1.5	0.7
3	0.5	40	0	2.5	1.5
4	0.75	20	0	1.3	0.76
5	1	20	0	1.4	0.71
6	1.25	20	0	1.6	1.01
7	0.5	30	0.1	1.5	0.72
8	0.5	30	0.2	1.3	0.41
9	2	10	0.1	0.3	0.2

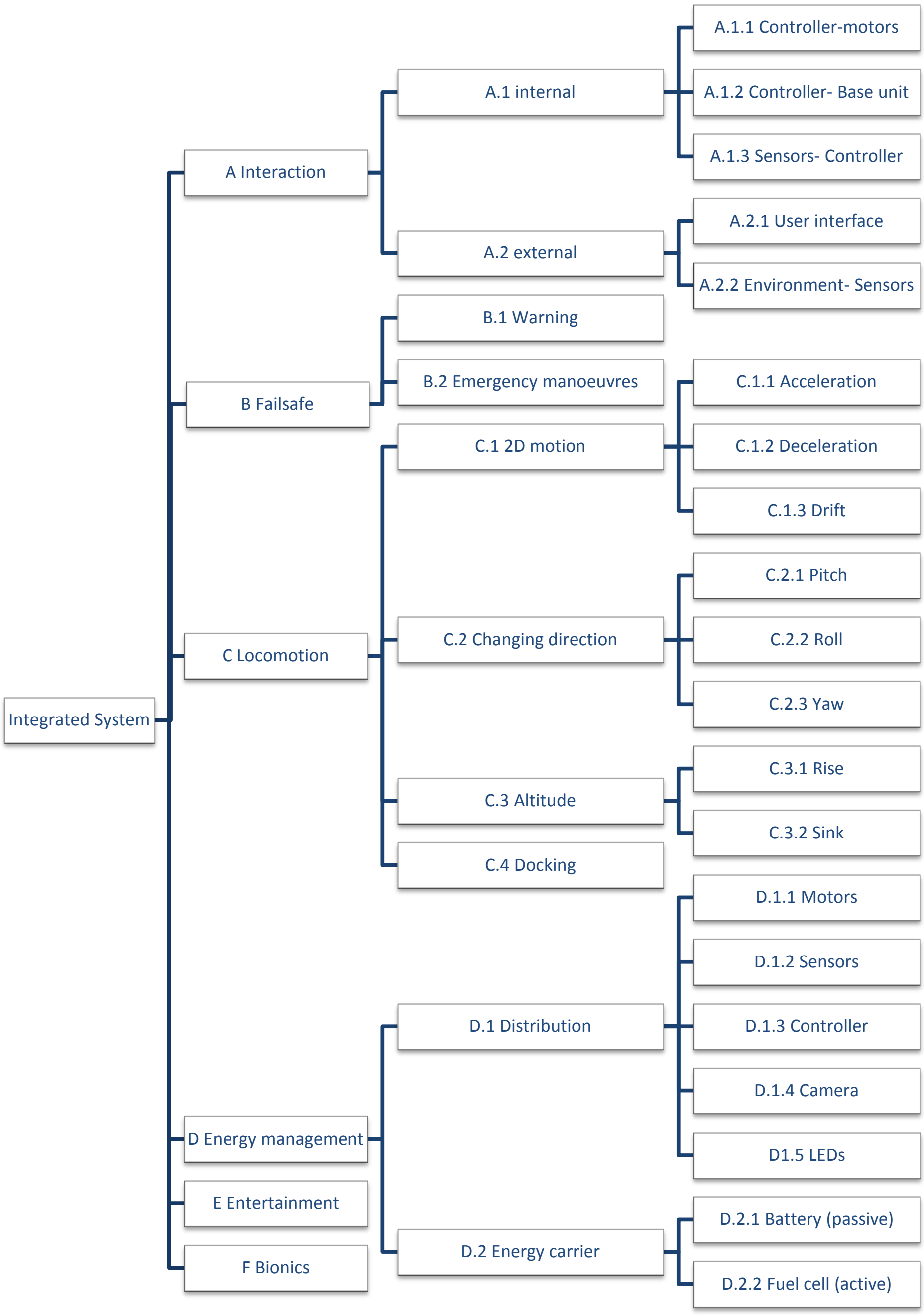
Conclusions

General

Foil fits nicely a squid like fin
Temperature differences arise between leading and following servo
1Hz maximal, higher frequencies are not compatible with more than 45°
strong current fluctuations
glue tape works quite fine, but combined with heat of servo fails
LabView works fine
Raspberry Pi ok
nice vortex structures were observed, seemed to look like a Karman vortex structure
Winkel-Befestigung' not sufficiently robust
Long-term operation: everything worked out fine; temperature rise (inside) during 2h 53°-85°
Fixation of components on shaft is a crucial in design and assembling
belt solution very nice, but complex and difficult to construct (Problem with manual tension of belt)
Underwater experiment at 1Hz Amplitude 50 workes worked for several minutes

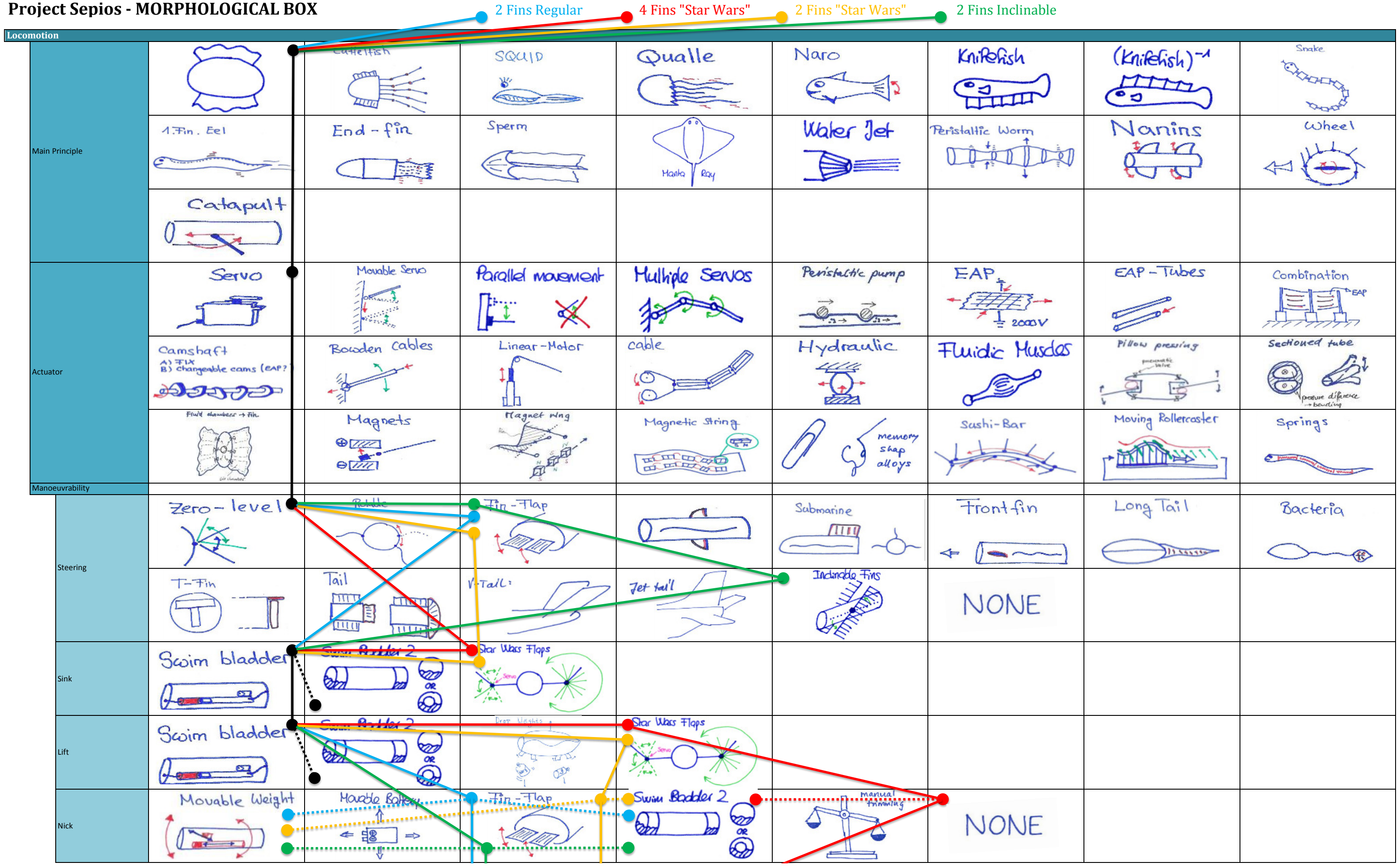
Appendix F

Functional System Overview and Morphological Box

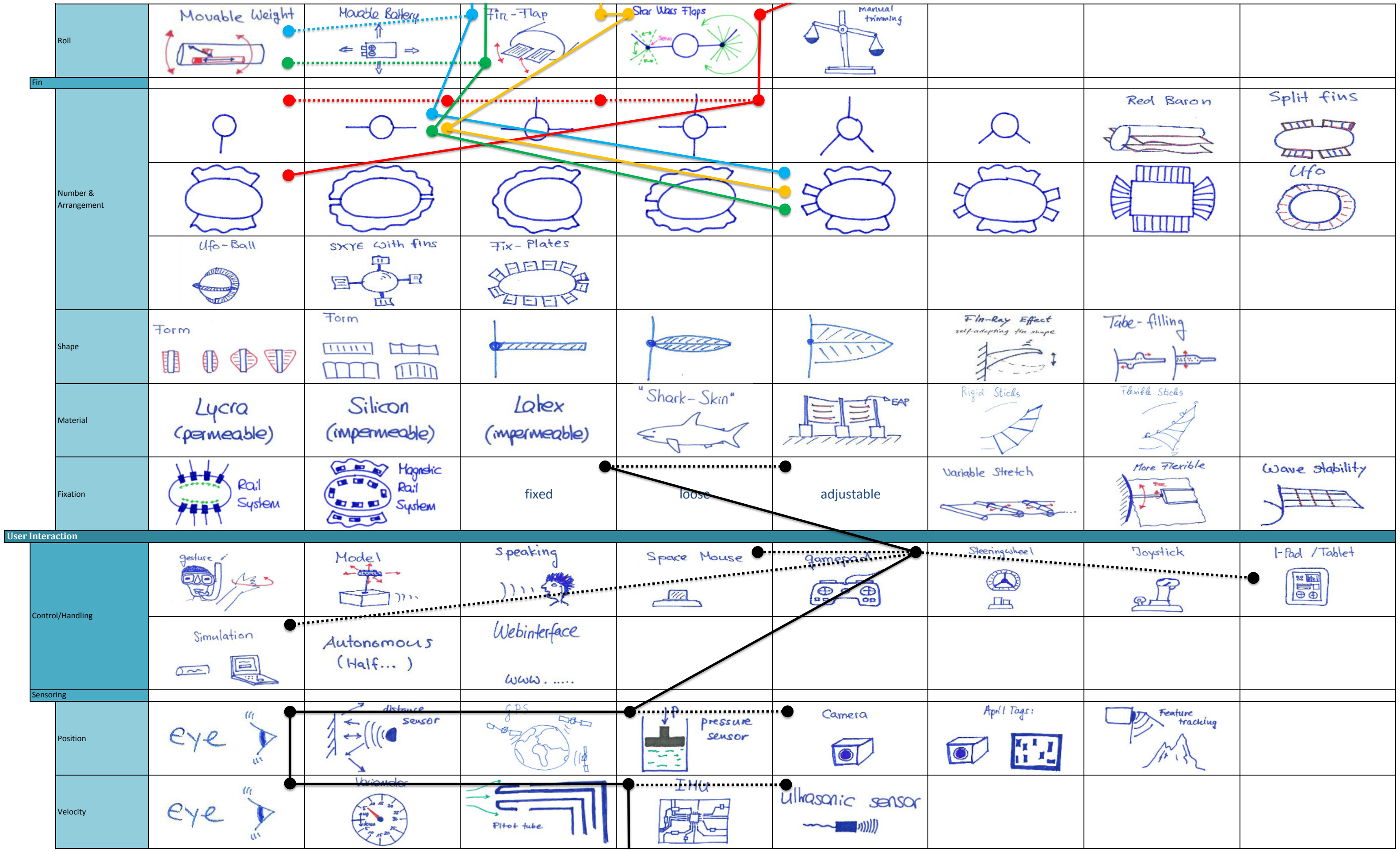


Project Sepios - MORPHOLOGICAL BOX

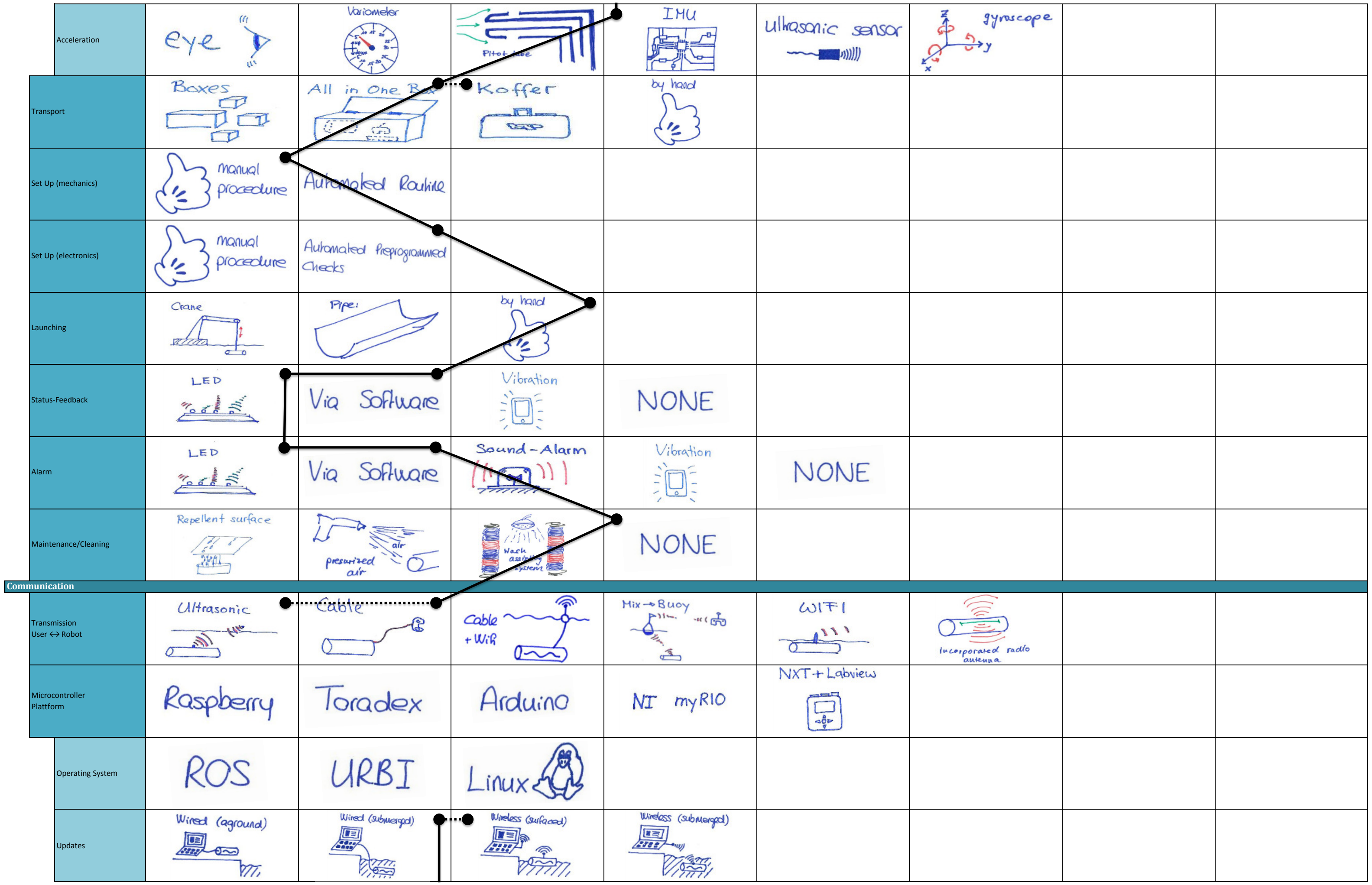
2 Fins Regular · 4 Fins "Star Wars" · 2 Fins "Star Wars" · 2 Fins Inclinable



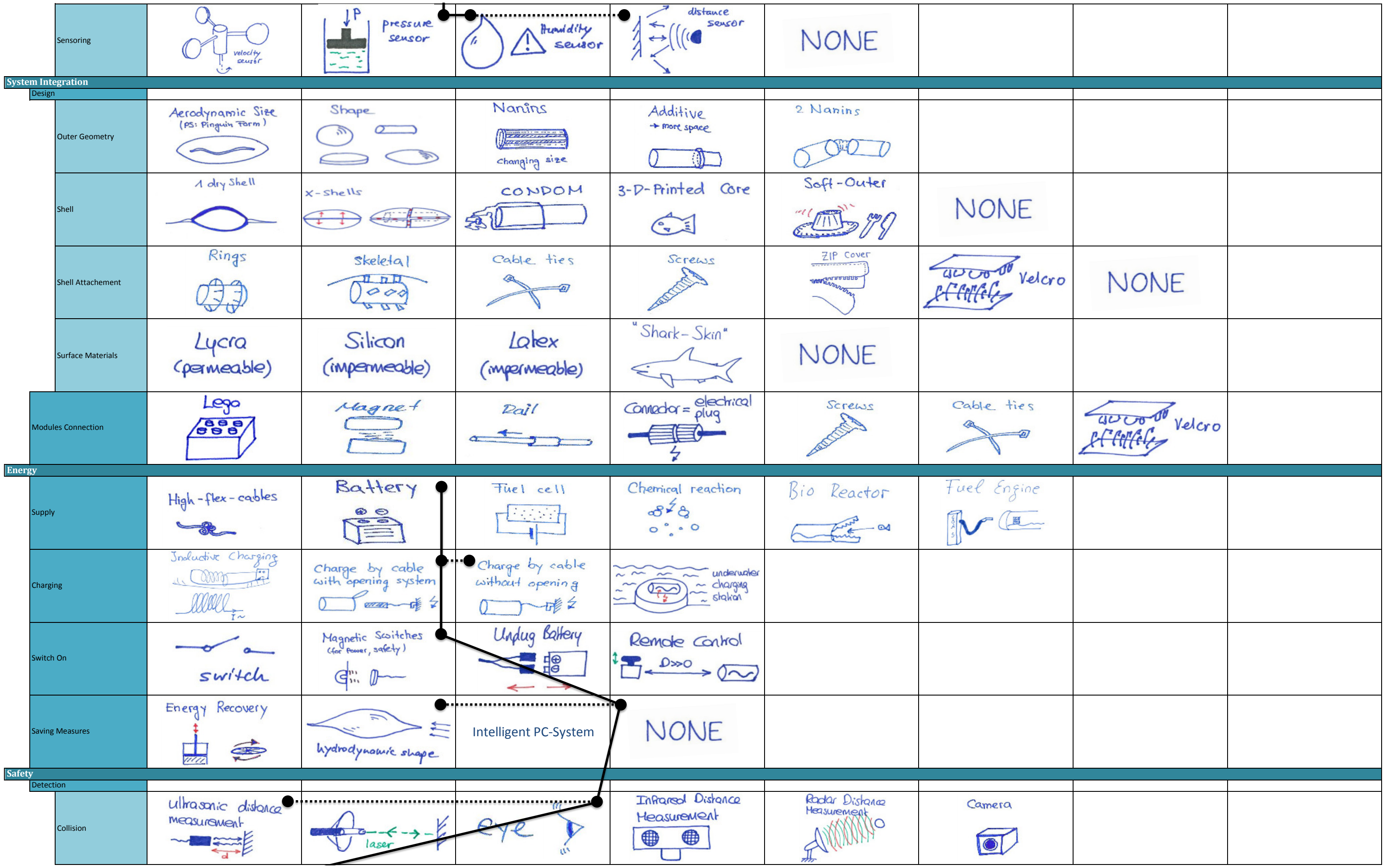
Locomotion									
	Main Principle		Cuttlefish	SQUID	Qualle	Naro	Knifefish	(Knifefish)$^{-1}$	Snake
		1 Fin. Eel	End-fin	Sperm	Manta Ray	Water Jet	Peristaltic Worm	Nanins	Wheel
		Catapult							
	Actuator	Servo	Movable Servo	Parallel movement	Multiple Servos	Peristaltic pump	EAP 2000V	EAP-Tubes	Combination EAP
		Camshaft A) FIX B) changeable cams (EAP?)	Bowden Cables	Linear-Motor	cable	Hydraulic	Fluidic Muscles	Pillow pressing pneumatic valve	Sectioned tube pressure difference → bending
		Fluid chambers → Fin air chamber	Magnets	Magnet ring Fin N S N S N S	Magnetic String	memory shap alloys	Sushi-Bar	Moving Rollercoster	Springs
Manoeuvrability									
	Steering	Zero-level	Paddle	Fin-Flap		Submarine	Front fin	Long Tail	Bacteria
		T-Fin	Tail	V-Tail:	Jet tail	Inclinable Fins	NONE		
	Sink	Swim bladder	Swim Bladder 2 OR	Star Wars Flaps Servo					
	Lift	Swim bladder	Swim Bladder 2 OR	Drop Weights	Star Wars Flaps Servo				
	Nick	Movable Weight	Movable Battery	Fin-Flap	Swim Bladder 2 OR	manual trimming	NONE		



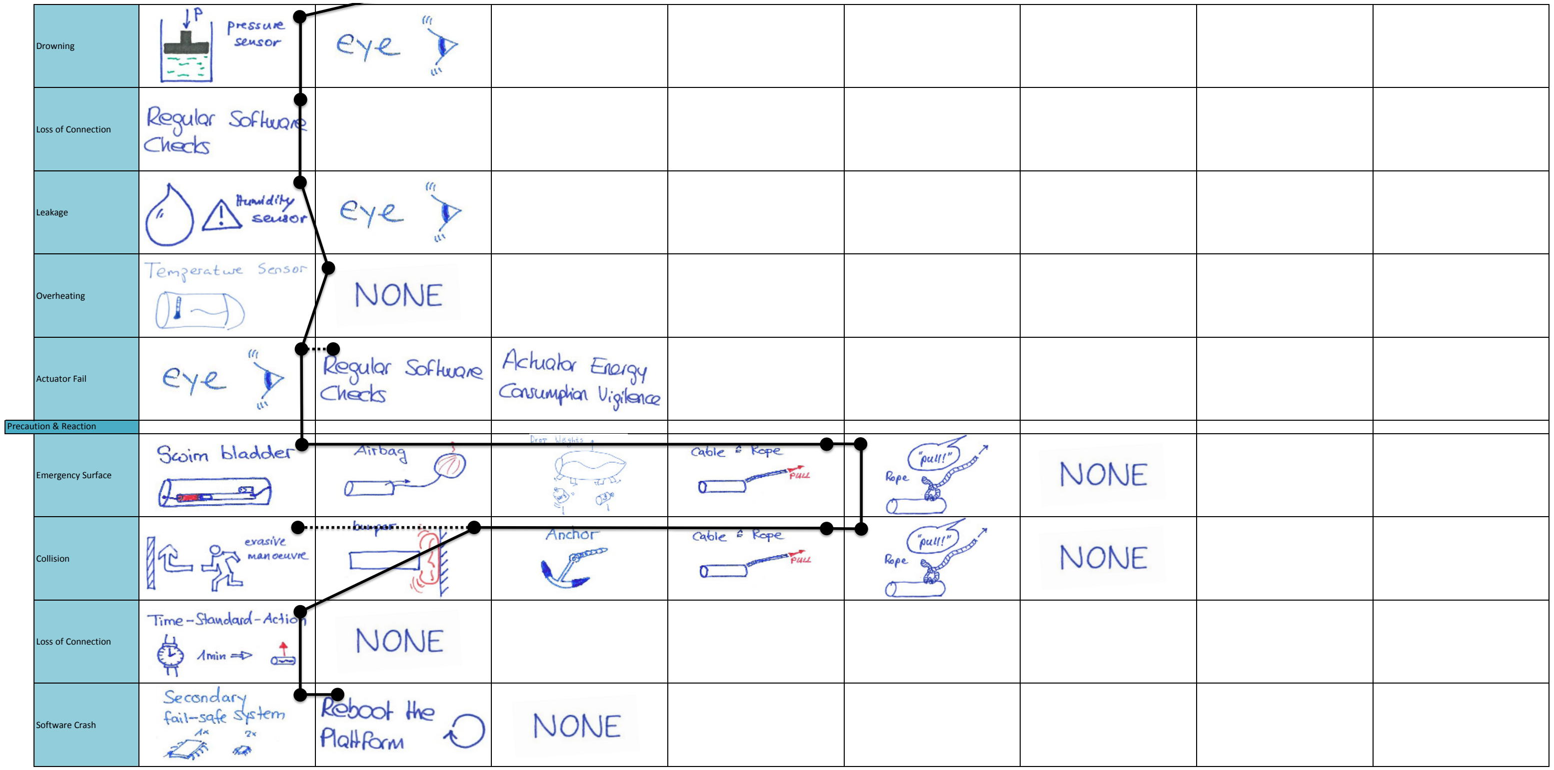
Roll	Movable Weight	Movable Battery	Fin-Flap	Star Wars Flaps	manual trimming			
Fin								
Number & Arrangement							Red Baron	Split fins
								Ufo
	Ufo-Ball	SKYE with fins	Fix-Plates					
Shape	Form	Form				Fin-Ray Effect self-adapting fin shape	Tube-filling	
Material	Lycra (permeable)	Silicon (impermeable)	Latex (impermeable)	"Shark-Skin"	EAP	Rigid Sticks	Flexible Sticks	
Fixation	Rail System	Magnetic Rail System	fixed	loose	adjustable	Variable Stretch	More Flexible	Wave stability
User Interaction								
Control/Handling	gesture	Model	Speaking	Space Mouse	gamepad	Steeringwheel	Joystick	I-Pad / Tablet
	Simulation	Autonomous (Half...)	Webinterface www.					
Sensoring								
Position	eye	distance sensor	GPS	pressure sensor	Camera	April Tags:	Feature tracking	
Velocity	eye	Variometer	Pitot tube	IMU	Ultrasonic sensor			



Acceleration	eye	Variometer	Pitot tube	IMU	ultrasonic sensor	gyroscope		
Transport	Boxes	All in One Box	Koffer	by hand				
Set Up (mechanics)	manual procedure	Automated Routine						
Set Up (electronics)	manual procedure	Automated Preprogrammed Checks						
Launching	Crane	Pipe:	by hand					
Status-Feedback	LED	Via Software	Vibration	NONE				
Alarm	LED	Via Software	Sound-Alarm	Vibration	NONE			
Maintenance/Cleaning	Repellent surface	presurized air	Wash assisting system	NONE				
Communication								
Transmission User ↔ Robot	Ultrasonic	Cable	Cable + Wifi	Mix → Buoy	WIFI	Incorporated radio antenna		
Microcontroller Plattform	Raspberry	Toradex	Arduino	NI myRIO	NXT + Labview			
Operating System	ROS	URBI	Linux					
Updates	Wired (aground)	Wired (submerged)	Wireless (surfaced)	Wireless (submerged)				



Sensoring	velocity sensor	P pressure sensor	Humidity sensor	distance sensor	NONE			
System Integration								
Design								
Outer Geometry	Aerodynamic Size (PS: Pinguin Form)	Shape	Nanins changing size	Additive → more space	2 Nanins			
Shell	1 dry Shell	x-Shells	CONDOM	3-D-Printed Core	Soft-Outer	NONE		
Shell Attachement	Rings	Skeletal	Cable ties	Screws	ZIP Cover	Velcro	NONE	
Surface Materials	Lycra (permeable)	Silicon (impermeable)	Latex (impermeable)	"Shark-Skin"	NONE			
Modules Connection	Lego	Magnet	Rail	Connector = electrical plug	Screws	Cable ties	Velcro	
Energy								
Supply	High-flex-cables	Battery	Fuel cell	Chemical reaction	Bio Reactor	Fuel Engine		
Charging	Inductive Charging I~	Charge by cable with opening system	Charge by cable without opening	underwater charging station				
Switch On	switch	Magnetic Switches (for Power, safety)	Unplug Battery	Remote Control D>>0				
Saving Measures	Energy Recovery	hydrodynamic shape	Intelligent PC-System	NONE				
Safety								
Detection								
Collision	Ultrasonic distance measurement d	laser	eye	Infrared Distance Measurement	Radar Distance Measurement	Camera		



Drowning	P pressure sensor	eye						
Loss of Connection	Regular Software Checks							
Leakage	Humidity sensor	eye						
Overheating	Temperature Sensor	NONE						
Actuator Fail	eye	Regular Software Checks	Actuator Energy Consumption Vigilence					
Precaution & Reaction								
Emergency Surface	Swim bladder	Airbag	Drop Weights	Cable & Rope PULL	Rope "pull!"	NONE		
Collision	evasive manoeuvre	bumper	Anchor	Cable & Rope PULL	Rope "pull!"	NONE		
Loss of Connection	Time-Standard-Action 1min ⇒	NONE						
Software Crash	Secondary fail-safe system 1x 2x	Reboot the Plattform	NONE					

Appendix G

Calculations

G.1 Dimensioning

The dimensioning of the critical moving parts has been done according to Ermanni [2011]. The detailed calculations can be found in the attached DVD.

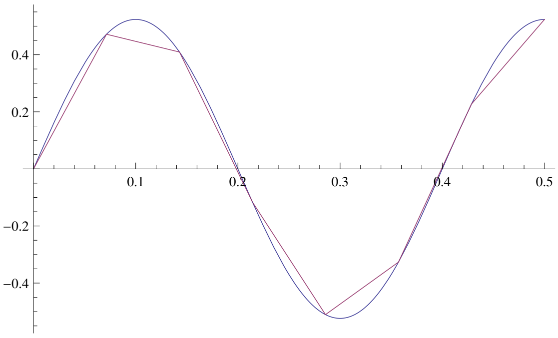
G.2 Mathematica-file used to evaluate the influence of the number of rays on the propulsion force

The Influence of the Number of Rays on the Fin Propulsion Force

```
In[243]:= finlength = .5; (* length of the fin along the squids body*)

In[244]:= nrays = 8; (* number of rays along one sidefin*)

In[245]:= drays = finlength / (nrays - 1); (* distance between two neigbouring rays*)

In[246]:= amplitude = 30 °; (*maximum deflection angle*)

In[247]:= wavelength = .4; (*distance between two rays with equal deflection angle*)

In[248]:= frequency = .6; (*oscilation frequency of one ray in the fin*)

In[249]:= wavespeed = wavelength * frequency;
          (*speed of the wave travelling along the sidefin*)

In[250]:= k = 2 * Pi / wavelength; (*wave number*)

In[251]:= omega = 2 * Pi * wavespeed / wavelength; (*circular frequency*)

In[252]:= period = 2 * Pi / omega;

In[253]:= xu[x_] = drays * Floor[x / drays]; (*position of the ray left of x*)

In[254]:= xo[x_] = drays * Ceiling[x / drays]; (*position of the ray reft of x*)

In[255]:= Angle[x_, t_] = amplitude * Sin[k * x - t * omega]; (*theoretical sine-wave*)

In[256]:= ddtAngle[x_, t_] = D[Angle[x, t], t];

In[257]:= Deflection[x_, t_] = Angle[xu[x], t] +
              (Angle[xo[x], t] - Angle[xu[x], t]) * (x - xu[x]) / (xo[x] - xu[x]);
          (*deflection angle of a point on the fin at position x and
           time t. the foil between the rays is assumed
           infinitely stretchable and the find being in vacuum.*)

In[258]:= ddxDeflection[x_, t_] = (Angle[xo[x], t] - Angle[xu[x], t]) / (xo[x] - xu[x]);

In[259]:= ddtDeflection[x_, t_] = ddtAngle[xu[x], t] +
              (ddtAngle[xo[x], t] - ddtAngle[xu[x], t]) * (x - xu[x]) / (xo[x] - xu[x]);
```

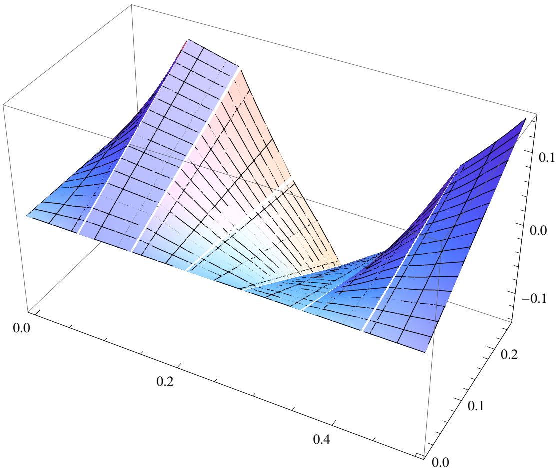
```
In[260]:= Plot[{Angle[x, 0], Deflection[x, 0]}, {x, 0, finlength},
           PlotRange → {-1.1 * amplitude, 1.1 * amplitude}]
```

Out[260]=



```
In[261]:= Displacement[x_, y_, t_] = y * Tan[Deflection[x, t]];
          (*Function describing the threedimensional
            shape of the fin via the distance from the xy-plane*)

In[262]:= Plot3D[Displacement[x, y, 0], {x, 0, finlength},
           {y, 0, .5 * finlength}, BoxRatios → Automatic]
```

Out[262]=



```
In[263]:= normal[x_, t_] = {-Abs[ddxDeflection[x, t]], Sign[ddxDeflection[x, t]]} /
             Sqrt[1 + ddxDeflection[x, t] ^ 2]; (*normal vector to the foil*)

In[264]:= Pressure[x_, t_] = ddtDeflection[x, t] ^ 2 * normal[x, t];
          (*pressure on a fin segment at position x and time t*)

In[265]:= Force[t_] =
           Integrate[Pressure[x, t] * Sqrt[1 + ddxDeflection[x, t] ^ 2], {x, 0, drays}] * nrays;
          (*force on a single fin segment times the total number of segments*)

In[266]:= averageforce = Integrate[Force[t] / period, {t, 0, period}]
           (*total force averaged over time*)

Out[266]= {-5.63203, -9.29965 × 10^-17}
```

G.3 Electroactive Polymers

G.3.1 The Agonist-Antagonist Configuration

The number of EAP-membranes which an agonist-antagonist-configuration requires to perform a typical swimming mode is calculated in the following. We consider the situation where the highest torque is acting on the fin. This is where it passes the zero-deflection-line normal to the body of the cuttlefish. The assumptions made in the following are rather optimistic.

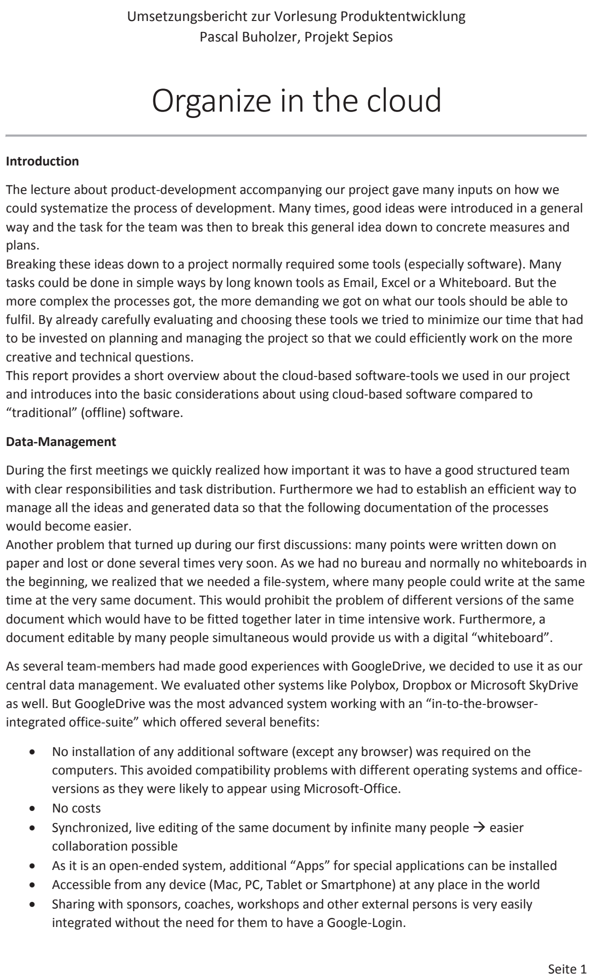
According to table 2.1 we set the minimum torque to be 0.5 Nm for a fin segment of 10 cm length. This corresponds to 5 Nm/m angular momentum per unit length. For high voltages and low activation frequencies Jordi found the blocking moment to be smaller than 0.1 Nm/m (pp. 7, figure 13) for one layer of VHB 4910 elastomer which we assume to be 60 μm thick. According to Jordi et al. [2010] we may sum up the forces applied by single membranes to find the overall force of the membranes arranged parallelly. Thus in order to reach the required angular momentum of 5 Nm/m we would need $\frac{5\ Nm/m}{0.1\ Nm/m} = 50$ layers of EAP-membranes.

G.3.2 The Stacked Actuator Configuration

The stacked-actuator-muscle is considered as an alternative to the agonist-antagonist-configuration. According to the test actuator fabricated by EMPA and mentioned by Kovacs and Düring [2009] we assume the electrostatic pressure to be $p_{el} = 0.06\ MPa$. The required angular momentum per unit length is again 5 $Nm = 15mm \cdot Area \cdot p_{el} = 15mm \cdot d \cdot 1m \cdot 0.06MPa$ wich results in an actuator thickness of $d = \frac{5N}{15\ mm \cdot 0.06MPa} = 5.6\ mm$. For a fin of 10 cm width the number of EAP-layers required is about $\frac{100\ mm}{0.1\ mm} = 1000$.

Appendix H

Cloud-software-tools



Umsetzungsbericht zur Vorlesung Produktentwicklung
Pascal Buholzer, Projekt Sepios

Organize in the cloud

Introduction

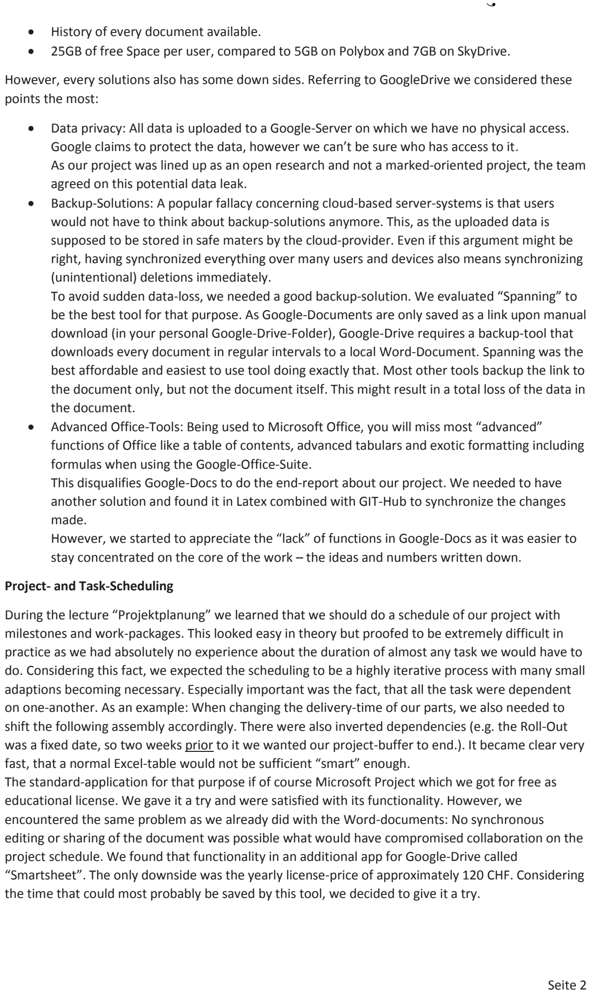
The lecture about product-development accompanying our project gave many inputs on how we could systematize the process of development. Many times, good ideas were introduced in a general way and the task for the team was then to break this general idea down to concrete measures and plans.
Breaking these ideas down to a project normally required some tools (especially software). Many tasks could be done in simple ways by long known tools as Email, Excel or a Whiteboard. But the more complex the processes got, the more demanding we got on what our tools should be able to fulfil. By already carefully evaluating and choosing these tools we tried to minimize our time that had to be invested on planning and managing the project so that we could efficiently work on the more creative and technical questions.
This report provides a short overview about the cloud-based software-tools we used in our project and introduces into the basic considerations about using cloud-based software compared to “traditional” (offline) software.

Data-Management

During the first meetings we quickly realized how important it was to have a good structured team with clear responsibilities and task distribution. Furthermore we had to establish an efficient way to manage all the ideas and generated data so that the following documentation of the processes would become easier.
Another problem that turned up during our first discussions: many points were written down on paper and lost or done several times very soon. As we had no bureau and normally no whiteboards in the beginning, we realized that we needed a file-system, where many people could write at the same time at the very same document. This would prohibit the problem of different versions of the same document which would have to be fitted together later in time intensive work. Furthermore, a document editable by many people simultaneous would provide us with a digital “whiteboard”.

As several team-members had made good experiences with GoogleDrive, we decided to use it as our central data management. We evaluated other systems like Polybox, Dropbox or Microsoft SkyDrive as well. But GoogleDrive was the most advanced system working with an “in-to-the-browser-integrated office-suite” which offered several benefits:

- No installation of any additional software (except any browser) was required on the computers. This avoided compatibility problems with different operating systems and office-versions as they were likely to appear using Microsoft-Office.
- No costs
- Synchronized, live editing of the same document by infinite many people → easier collaboration possible
- As it is an open-ended system, additional “Apps” for special applications can be installed
- Accessible from any device (Mac, PC, Tablet or Smartphone) at any place in the world
- Sharing with sponsors, coaches, workshops and other external persons is very easily integrated without the need for them to have a Google-Login.



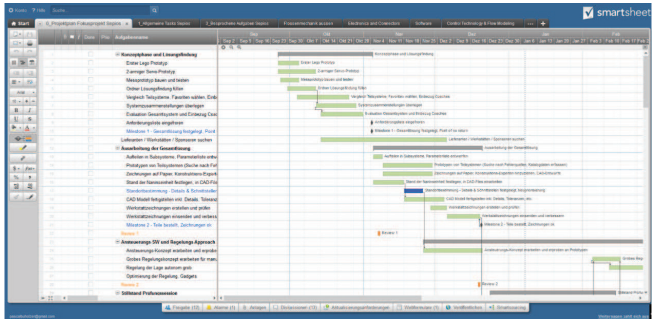
- History of every document available.
- 25GB of free Space per user, compared to 5GB on Polybox and 7GB on SkyDrive.

However, every solutions also has some down sides. Referring to GoogleDrive we considered these points the most:

- Data privacy: All data is uploaded to a Google-Server on which we have no physical access. Google claims to protect the data, however we can’t be sure who has access to it.
  As our project was lined up as an open research and not a marked-oriented project, the team agreed on this potential data leak.
- Backup-Solutions: A popular fallacy concerning cloud-based server-systems is that users would not have to think about backup-solutions anymore. This, as the uploaded data is supposed to be stored in safe maters by the cloud-provider. Even if this argument might be right, having synchronized everything over many users and devices also means synchronizing (unintentional) deletions immediately.
  To avoid sudden data-loss, we needed a good backup-solution. We evaluated “Spanning” to be the best tool for that purpose. As Google-Documents are only saved as a link upon manual download (in your personal Google-Drive-Folder), Google-Drive requires a backup-tool that downloads every document in regular intervals to a local Word-Document. Spanning was the best affordable and easiest to use tool doing exactly that. Most other tools backup the link to the document only, but not the document itself. This might result in a total loss of the data in the document.
- Advanced Office-Tools: Being used to Microsoft Office, you will miss most “advanced” functions of Office like a table of contents, advanced tabulars and exotic formatting including formulas when using the Google-Office-Suite.
  This disqualifies Google-Docs to do the end-report about our project. We needed to have another solution and found it in Latex combined with GIT-Hub to synchronize the changes made.
  However, we started to appreciate the “lack” of functions in Google-Docs as it was easier to stay concentrated on the core of the work – the ideas and numbers written down.

Project- and Task-Scheduling

During the lecture “Projektplanung” we learned that we should do a schedule of our project with milestones and work-packages. This looked easy in theory but proofed to be extremely difficult in practice as we had absolutely no experience about the duration of almost any task we would have to do. Considering this fact, we expected the scheduling to be a highly iterative process with many small adaptions becoming necessary. Especially important was the fact, that all the task were dependent on one-another. As an example: When changing the delivery-time of our parts, we also needed to shift the following assembly accordingly. There were also inverted dependencies (e.g. the Roll-Out was a fixed date, so two weeks prior to it we wanted our project-buffer to end.). It became clear very fast, that a normal Excel-table would not be sufficient “smart” enough.
The standard-application for that purpose if of course Microsoft Project which we got for free as educational license. We gave it a try and were satisfied with its functionality. However, we encountered the same problem as we already did with the Word-documents: No synchronous editing or sharing of the document was possible what would have compromised collaboration on the project schedule. We found that functionality in an additional app for Google-Drive called “Smartsheet”. The only downside was the yearly license-price of approximately 120 CHF. Considering the time that could most probably be saved by this tool, we decided to give it a try.

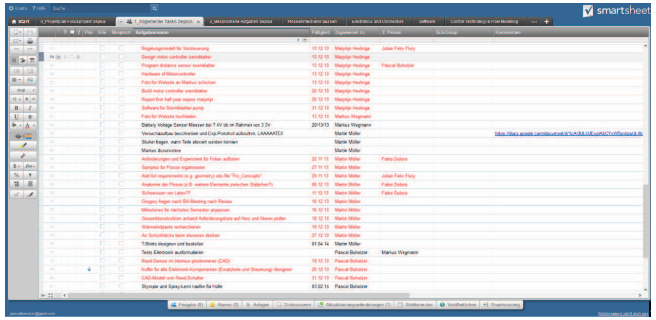


Project-Schedule in SmartSheet

In addition SmartSheet allowed us to create lists with tasks to be done by the team-members. The lists can be filtered by different criteria as due-date, assigned persons or priority. It is directly interconnected with the project-schedule (so the tasks change in dependence of the schedule or vice-versa) and highlights delayed tasks automatically so everyone can see easily what needs to be done next.

The team-leader is responsible to distribute the tasks on to the different subgroups of the team or single persons while always keeping an overview over the ongoing work.

We considered the organization of bigger and important tasks with "work-packages" to be a good idea and adopted it to our project. The work-package created from a template in Google Drive can easily be attached to the task-entry in the list so the persons assigned to the task can immediately start with it. After completion of the task, the results could be noted directly in the work-package. This system eliminated the need of a complicated and fault-prone system with work-packages written down on paper and sorted into folders. Especially it superseded the need to go to the office to get your new tasks as well as retyping the results for the report later.



Task-List in SmartSheet

Conclusion

We were very happy about the theoretical inputs gathered in the lecture "Product-Development" and tried to transfer many ideas into our project in a way to increase our efficiency and minimize management-related risks. Especially in the beginning, we invested a lot of time in evaluating tools to support us in the management of the project and found a solution that is very flexible, interactive, innovative and functional while still being cheap and safe.

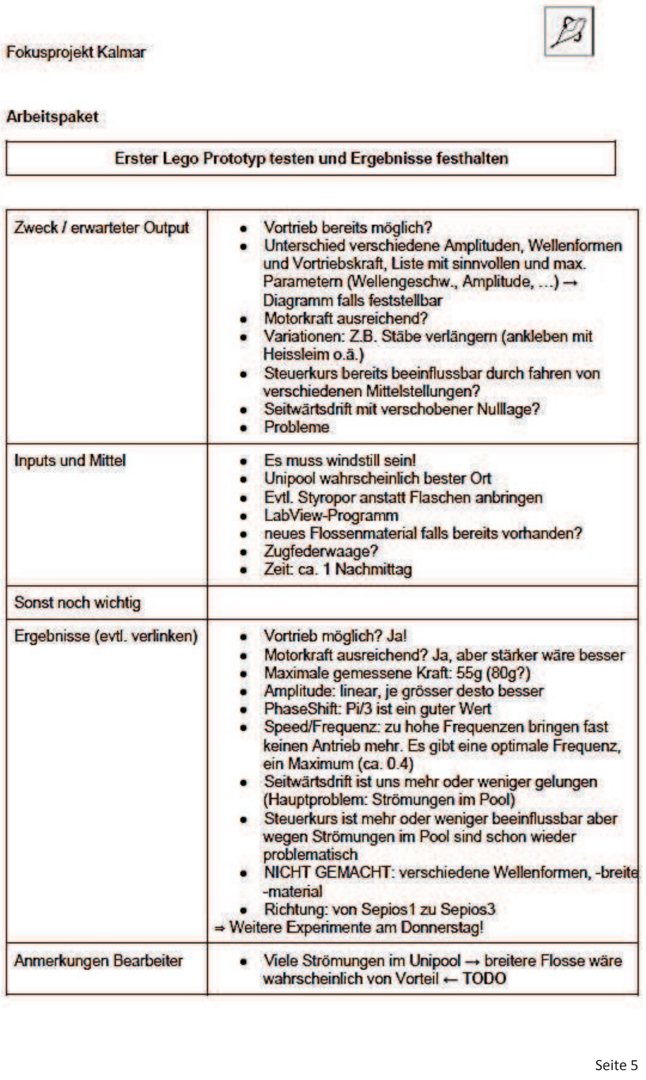
It can be said that the cloud-based data-management fulfilled most of our expectations and offered superior work efficiency compared to the "traditional" (offline) way - especially for collaboration.

However, even though Google-Drive combined with SmartSheet and GIT-Hub for the report proofed to be a very efficient combination, one must keep in mind, that software changes extremely fast and there might be many new innovative providers in the future. This applies especially to all data privacy related topics.

In any case working in the cloud should definitely be considered during the setup of a new project as a time- and cost-saving alternative to "known" offline-work.

Appendix:

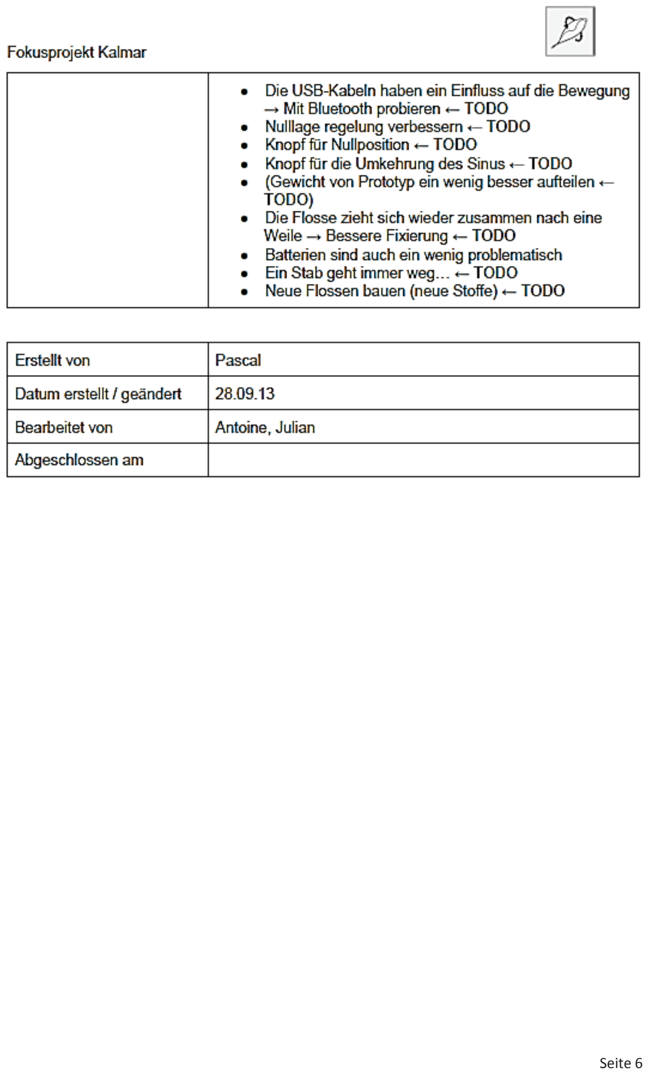
- 1x Sample Work-Package



Arbeitspaket

Erster Lego Prototyp testen und Ergebnisse festhalten

Zweck / erwarteter Output	• Vortrieb bereits möglich? • Unterschied verschiedene Amplituden, Wellenformen und Vortriebskraft, Liste mit sinnvollen und max. Parametern (Wellengeschw., Amplitude, …) → Diagramm falls feststellbar • Motorkraft ausreichend? • Variationen: Z.B. Stäbe verlängern (ankleben mit Heissleim o.ä.) • Steuerkurs bereits beeinflussbar durch fahren von verschiedenen Mittelstellungen? • Seitwärtsdrift mit verschobener Nulllage? • Probleme
Inputs und Mittel	• Es muss windstill sein! • Unipool wahrscheinlich bester Ort • Evtl. Styropor anstatt Flaschen anbringen • LabView-Programm • neues Flossenmaterial falls bereits vorhanden? • Zugfederwaage? • Zeit: ca. 1 Nachmittag
Sonst noch wichtig	
Ergebnisse (evtl. verlinken)	• Vortrieb möglich? Ja! • Motorkraft ausreichend? Ja, aber stärker wäre besser • Maximale gemessene Kraft: 55g (80g?) • Amplitude: linear, je grösser desto besser • PhaseShift: Pi/3 ist ein guter Wert • Speed/Frequenz: zu hohe Frequenzen bringen fast keinen Antrieb mehr. Es gibt eine optimale Frequenz, ein Maximum (ca. 0.4) • Seitwärtsdrift ist uns mehr oder weniger gelungen (Hauptproblem: Strömungen im Pool) • Steuerkurs ist mehr oder weniger beeinflussbar aber wegen Strömungen im Pool sind schon wieder problematisch • NICHT GEMACHT: verschiedene Wellenformen, -breite -material • Richtung: von Sepios1 zu Sepios3 ⇒ Weitere Experimente am Donnerstag!
Anmerkungen Bearbeiter	• Viele Strömungen im Unipool → breitere Flosse wäre wahrscheinlich von Vorteil ← TODO



	• Die USB-Kabeln haben ein Einfluss auf die Bewegung → Mit Bluetooth probieren ← TODO • Nulllage regelung verbessern ← TODO • Knopf für Nullposition ← TODO • Knopf für die Umkehrung des Sinus ← TODO • (Gewicht von Prototyp ein wenig besser aufteilen ← TODO) • Die Flosse zieht sich wieder zusammen nach eine Weile → Bessere Fixierung ← TODO • Batterien sind auch ein wenig problematisch • Ein Stab geht immer weg… ← TODO • Neue Flossen bauen (neue Stoffe) ← TODO

Erstellt von	Pascal
Datum erstellt / geändert	28.09.13
Bearbeitet von	Antoine, Julian
Abgeschlossen am	