École Polytechnique
Fédérale de Lausanne

Control - Dahu

Semester Project

Author
SAVIOZ Baptiste

Professor - Lab
SALZMANN Christophe
Laboratoire d'automatique 3

Supervisor
AMACHER Robin,
NEBULONI Stefano

Swiss Solar Boat

Spring 2023

Abstract

The semester project *Control - Dahu* summarises the changes applied to control in the spring 2023 semester, as well as giving some ideas for the control of the next boat: the Ref. The changes made to the control algorithm are :

- The addition of antiwindup (section 2.3)
- The removal of the problematic coefficients $coeff1$ and $coeff2$ (section 2.6.4)
- The way to convert an angle into a motor increment (section 2.5)

The complete current control algorithm is shown in figure 1. The gains of the controllers used are shown in table 1. A good summary of how the control algorithm works can be found in section 2.1.2. In addition, the section 3.1 is dedicated to the testing of position control, whose approach is perhaps interesting in order to simplify control. Section 3.3.1 tackles the problem of roll control and proposes points for improvement.

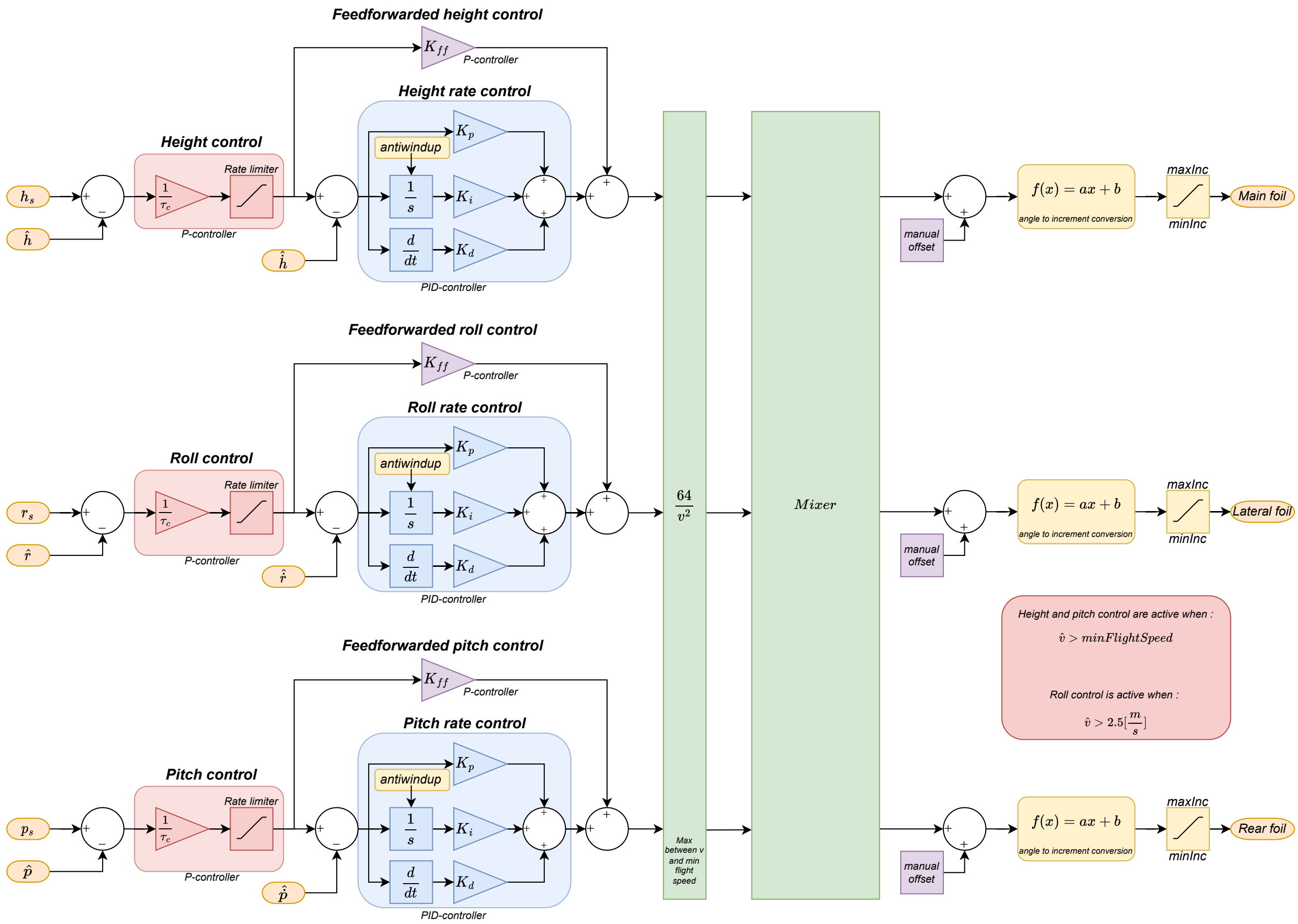


Figure 1: Cascaded control scheme - Implemented at time of writing

Parameter	Height Control	Roll Control	Pitch Control
T_c	2	0.75	3
K_{ff}	42	1.7	1
K_P	0.25	0.2	0.1
K_I	7	0.45	0.1
K_D	0	0	0

Table 1: Control gains used at time of writing

Contents

1 Introduction **5**
1.1 Global objectives . . . 6
1.2 Specific objectives . . . 6
1.3 Timeline . . . 7

2 Control Dahu **9**
2.1 Summary of the control at SSB before the start of the project . . . 9
2.1.1 Hardware & Software . . . 9
2.1.2 Control algorithm . . . 9
2.1.3 Method . . . 12
2.1.4 Performances . . . 12
2.1.5 Conclusion . . . 13
2.2 Updating the FCU and documentation . . . 14
2.2.1 Motivation . . . 14
2.2.2 Implementation . . . 14
2.2.3 Results analysis . . . 14
2.2.4 Conclusion . . . 14
2.3 Antiwindup . . . 15
2.3.1 Motivation . . . 15
2.3.2 Overview of the existing solutions . . . 16
2.3.3 Choice and Design . . . 16
2.3.4 Implementation and testing . . . 17
2.3.5 Results analysis . . . 18
2.3.6 Conclusion . . . 19
2.4 Empirical Boat Model . . . 19
2.4.1 Motivation . . . 19
2.4.2 Design . . . 19
2.4.3 Results analysis . . . 21
2.4.4 Conclusion . . . 22
2.5 Foil calibration . . . 23
2.5.1 Motivation . . . 23
2.5.2 Implementation . . . 24
2.5.3 Result analysis . . . 25
2.5.4 Conclusion . . . 25
2.6 Lake tests . . . 25
2.6.1 Motivation, context and timeline . . . 25
2.6.2 Test on week 3 - Objectives . . . 25
2.6.3 Test on week 3 - Implementation . . . 26
2.6.4 Test on week 3 - Result analysis . . . 26
2.6.5 Test on week 3 - Conclusion . . . 27
2.6.6 Test on week 9 - Objectives . . . 27
2.6.7 Test on week 9 - Implementation . . . 28
2.6.8 Test on week 9 - Result analysis . . . 28
2.6.9 Test on week 9 - Conclusion . . . 34

3 Control REF **35**
3.1 Position control . . . 35
3.1.1 Motivation . . . 35
3.1.2 Implementation . . . 35
3.1.3 Lake tests - week 11 . . . 36
3.1.4 Result analysis . . . 36
3.1.5 Conclusion . . . 37
3.2 Height measurement - Radar . . . 37
3.2.1 Context & motivation . . . 37
3.2.2 Implementation . . . 37
3.2.3 Result analysis . . . 39
3.2.4 Conclusion . . . 39

3.3 Recommendation and future directions for the REF 40
3.3.1 Yaw measurement for roll compensation 40
3.3.2 Hardware & Software 40
3.3.3 Safety reflexes and soft crash 41
3.3.4 Estimator & Optimal control 41

4 Conclusion **43**

A Regression Analysis **44**
A.1 Matlab script for data preprocessing 44
A.2 Python script for regression analysis 49

B Foil calibration procedure **54**

C Radar Baumer - factory reset code **59**

Abbreviation

AW	Antiwindup
FCU	Flight Control Unit
GS	Ground Station
IDE	Integrated Development environment
LTI	Linear Time Invariant
LQR	Linear Quadratic Regulator
MIMO	Multiple Inputs Multiple Outputs
MPC	Model Predictive Control
OCD	On Chip Debugger
OS	Operating System
REF	Renewable Energy Foiler
REX	Range Extender
RMSE	Root mean square error
SSB	Swiss Solar Boat
SISO	Single Input Single Output

1 Introduction

Swiss Solar Boat is an interdisciplinary association, part of the EPFL MAKE project whose objective is to build and optimize foiling boats powered by renewable energy. After two participations in the Monaco Energy Boat Challenge in 2021 and 2022 with the Dahu, a famous Polynesian foiling prao, the association is looking towards new objectives and is starting to design a new boat: the REF. The optimisation of the current boat, the Dahu, is coming to an end and the desire to design for competition is fading. The team wants to get closer to real-world issues, with greater autonomy, more passengers and greater cruising speed. To achieve this, we needed to increase the density of the energy on board, and we opted for hydrogen.
This is how the REF (Renewable Energy Foiler) project was born, a new hybrid boat: solar - hydrogen, with the aim of launching it by 2025. Integrating hydrogen into a boat is a major challenge for a team of students, so we decided to gain some experience by hybridising our current boat, the Dahu, with a small hydrogen module to extend its autonomy, the REX - Range Extender project. The aim is that in the summer of 2023, we will be able to cross the lake - Lausanne - Evian - Lausanne - more than twice our current autonomy, thanks to hydrogen and the REX project. The success of this project will be measured not only by the achievement of this objective, but above all by the amount of experience we will have gained. There will be plenty of mistakes to make, and we'll be glad to make them if they help us to learn and improve for the REF.

Figure 2: The most famous flying Polynesian prao : The Dahu

My name is Baptiste Savioz and I'm a MA4 robotics student with a bachelor's degree in micro-engineering. In addition, I joined the association in spring 2022, took part in monaco 2022 and since fall 2022 I've been in charge of the hardware, software and control electronics team. It was in that spirit that my semester project, laconically entitled *Control Dahu*, was born. Indeed, in the winter of 2022-2023, we had experienced a number of control setbacks and capsized the boat several times. We were nowhere near the level of reliability of Monaco 2022 and it was not feasible in the current state to foil with the extra 50kg of the REX. I was already in charge of control at the time, so I had a good understanding of the system and practices at SSB, as well as the likely areas for improvement. However, I didn't have enough time to devote to the project and it was imperative to make the control system more reliable if we were to have any hope of successfully crossing the lake in the summer of 2023.

1.1 Global objectives

With this in mind, it is possible to define the overall objectives that have guided this project - *control Dahu.*

	Objective	Priority
1.	Maintain the control of the Dahu this semester and prepare for the lake tests	Highest
2.	Ensuring reliable control with an additional 50kg load	Highest
3.	Document the control system and its practices at SSB.	Medium
4.	Train one person to maintain control next semester	Medium
5.	Provide the basis of control for the next boat: the REF	Low

Table 2: Project's global objectives

The imminent goal of crossing the lake with the REX project is clearly the main objective. This involves, on the one hand, maintaining the current control of the Dahu, i.e. ensuring that the boat can take to the water by getting it ready before each lake test session, with sufficient expertise in the FCU - Flight Control Unit - which is the unit that handles the control algorithm, as well as the boat's state machine, to ensure that the boat can be operated and, why not, fly. On the other hand, this not only involves maintaining the control, but also improving it, because as it stands, it is not reliable enough to fly for extended periods of time, nor to fly with an additional load of 50 kg. Then, objectives three and four arise from a recurring problem at SSB and in all the associations where the students change regularly: the transmission of information. Indeed, there are countless projects where students have to start projects from scratch because previous work was not designed to be carried over and maintained by subsequent students. This was the case with my project, for example, where out of the thousands of lines of code contained in the FCU and two semester projects, I only found 5 lines of comments... Furthermore, the last objective is to take advantage of the experience generated by this project to learn from it and pass on the benefits to the next boat. Finally, I'd like to add a comment about the objectives of this project, as I'm both electronic team leader and in charge of control (through my project). Certain team or association objectives take precedence over my personal project objectives and when there is a problem with electronics, it's my responsibility to manage the team or even often to resolve the problem personally, which limits the amount of time I have to allocate to the project. Control is the top of the pyramid, so keep this sentence in mind when reading:

Le contrôle, c'est quand tout le reste fonctionne.

1.2 Specific objectives

The global objectives were clearly established at the start of the project. However, it was more difficult to establish a list of specific objectives, as many of them emerged or were abandoned during the semester, as progress was made, failures were encountered and lessons learned, as well as the risks they involved for the association. It is therefore impossible for me to write this paragraph rigorously without rewriting the history or cherry-picking the information. I will therefore limit myself to presenting the parts of my work that I have completed in order to keep this report concise and interesting. Nevertheless, if anyone is interested in the chronological follow-up of my work, I invite them to consult the design reviews, which are available on the association's drive.

Objectives	**Motivation**
FCU update and documentation	The lack of comments and documentation on how the FCU works makes it difficult to maintain. Additionally, since the start of the project, no work has been done to clean up the code.
Take into account actuator saturation	The physical saturation of the actuators is not taken into account by the control algorithm. This poses problems and there is considerable margin for improvement.
Develop a boat model	In order to evaluate the boat's performance or to use more advanced control methods, a model of the boat is required.
Actuation calibration	No rigorous actuation calibration procedure has been implemented at SSB, and results vary widely from one test session to the next.
Tune the regulators and adapt the control algorithm	Improve the robustness of the boat's flight and adapt the control system to the various changes made (physical, electronic, software).
Test position control	A complex and opaque cascade control structure is used. A simpler structure has never been tested.
Contactless altitude measurement	Testing new ways of acquiring altitude measurement

Table 3: Project's specific objectives

1.3 Timeline

In order to carry out the project correctly, the following timeline was drawn up. It take place during the sping semester 2023.

week 1	Understanding the FCU code and reading reports
week 2	Prepare the boat for lake test, comment on the FCU code and production of documentation
week 3	**Lake test**
week 4	Analysis of boat performance & Decoding Baumer radar messages & Anti-windup Literature review
week 5	Restructuring the FCU code and preparing the framework for future implementation
week 6	Code debugging & Designing antiwindup scheme
week 7	Validation of antiwindup scheme and implementation
week 8	Implementation of Baumer radar
week 8+	Prepare the boat for lake test & finalise implementations & actuators calibration
week 9	**Lake test**
week 10	Analysis of boat performance & development of boat model & Position control scheme implementation
week 11	**Lake test**
week 12	Analysis of boat performance
week 13	Production of documentation
week 14	Report writing

Table 4: Timeline of the semester project

Finally, I would like to add a few words about a project in an association like SSB. It is not limited to this simple project and we are quickly overtaken by the responsibilities. It is an exciting job that involves us in a human way and not only for our technical skills. It is not possible to summarise everything done and learned in this report, nor is it the purpose of the latter. The long nights spent debugging the boat, preparing for lake test. The countless kilometres of cable we saw pass by, to which we sometimes modified a connector, changed a sleeve or even completely replaced. The datasheet or norms that we read and reread until we hurt our eyes to make sure that everything is in order. The countless coffees we drank at the local, rarely paying for them. The books we filled out with schematics and drawing to coordinate ourself between the different systems. All the helping hands given to colleagues out of passion for what we do. All this will go unnoticed in this report.

2 Control Dahu

2.1 Summary of the control at SSB before the start of the project

Swiss Solar Boat operates a foiling boat called the Dahu. It has three foils: the main, which is close to the centre of mass and carries most of the boat's weight, the lateral, which is off-centre and has a major influence on the roll, and the rear, which mainly influences the pitch. In addition, the boat is propelled by two motors at the rear and guided by the rudder, which can be turned. The boat therefore has five degrees of freedom: the angles of the three foils, the command of the propulsion motors and the angle of the rudder. Two degrees of freedom are controlled by the pilot (steering and propulsion) while the remaining three, the angles of the foils, are controlled by the FCU and its control algorithm. So, the primordial unit that manages the boat is the FCU - Flight Control Unit. This is also where the boat's sate machine is implemented. I won't go into any more detail, as several excellent reports have already done so [1][2]

2.1.1 Hardware & Software

The FCU is located in the main electrical box, under the headrest, behind the pilot. The hardware is a NUCLEO-F446RE card with a custom shield to integrate two CAN lines for communication with the rest of the boat. No modifications have been made to it this semester as it has been extensively tested and has proved its robustness. The card runs at 180 [MHz], which is more than sufficient for our application. On the software side, an embedded operating system, FreeRTOS, has been implemented, enabling real-time multi-threading. In addition, an On-Chip-debugger (OpenOCD) is used, enabling the FCU to be reprogrammed remotely when the boat is switched on, and the Ground Station connected. In short, our on-board computer (RaspberryPi 3) is physically connected to the FCU's debugger (the chip that physically enables reprogramming). By running a program on our on-board computer, we can connect to the debugger remotely and reprogram the FCU remotely during tests on the lake, for example. The programming environment is :

IDE	STM32CubeIDE V1.7.0
OS	FreeRTOS V10.0.1
Firmware	STM32Cube FW_F4 V1.24.1

Table 5: FCU programming environment

No major changes have been made. There was talk of updating it, but this was aborted because it would have involved changes to Ground Station V1 (GS), while a V2 was under development. The cost/benefit ratio was too low, while the risk involved was too high.

2.1.2 Control algorithm

No up-to-date and complete schematics of the control algorithm existed at the start of the project, and the control team was simply unaware of certain apesct/parts of the algorithm. So the first thing to do was to reconstruct the algorithm from the FCU code (Figure 3). Unfortunately, I can't guarantee the complete accuracy of this diagram, but it's the most complete and correct one available.

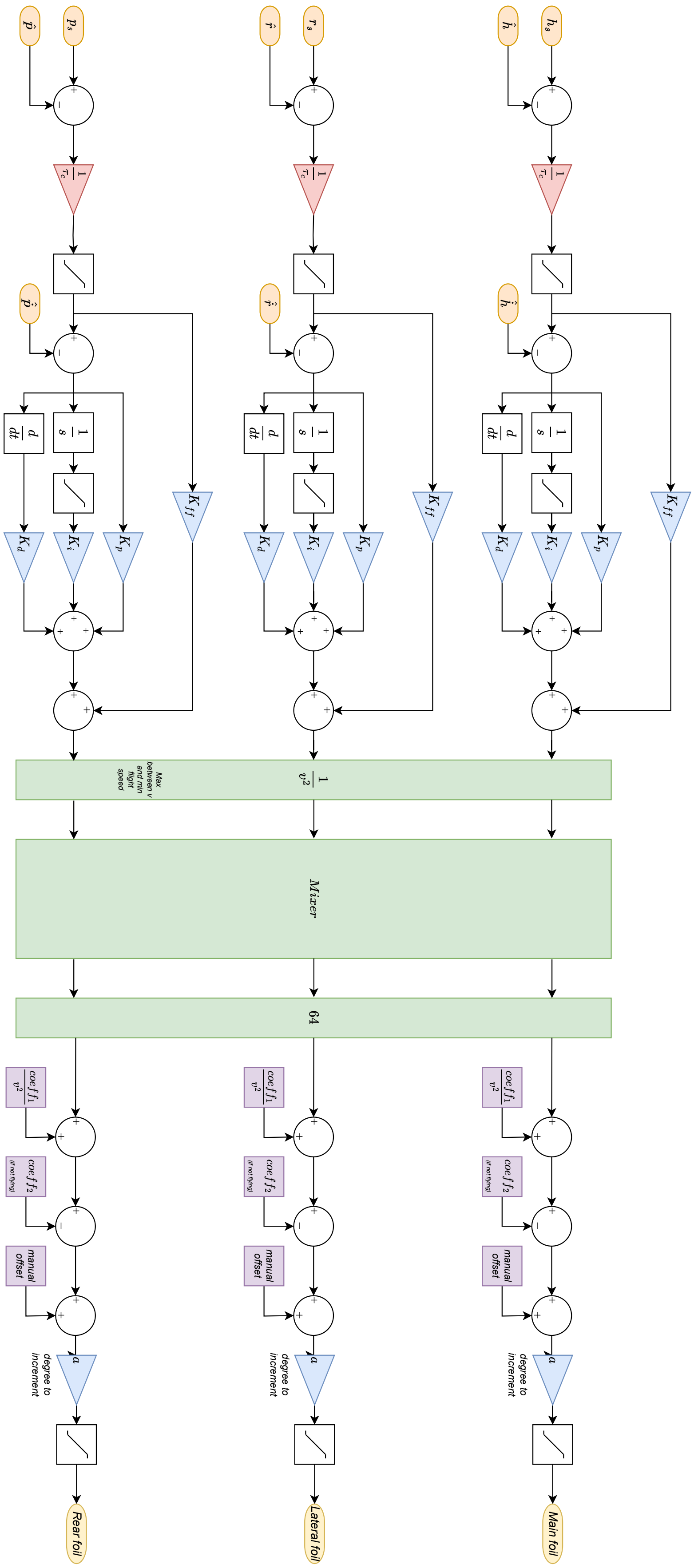


Figure 3: Full Control scheme at the beginning of the project (Out of date!)

The control algorithm is a cascade control algorithm, with first a position controller whose output is a speed reference which the second controller tries to follow. The second controller is therefore a rate controller, its output is an angle in degrees[1], which is then scaled by a factor $\frac{64}{v^2}$. The idea is that the lift of a foil is proportional to its speed squared.

$$L_{foil} \propto v^2 \tag{2.1.1}$$

Scaling the controller output by $\frac{64}{v^2}$ therefore gives the *nominal* controller (i.e. scale by a factor of 1) at a speed of $8[\frac{m}{s}]$, which is the nominal cruising speed, but above all gives a force control, rather than an angle control. The idea of this force control is to be invariant to the change in boat speed and therefore that the controller's aggressivity on the boat is the same at $6[\frac{m}{s}]$ as at $10[\frac{m}{s}]$.

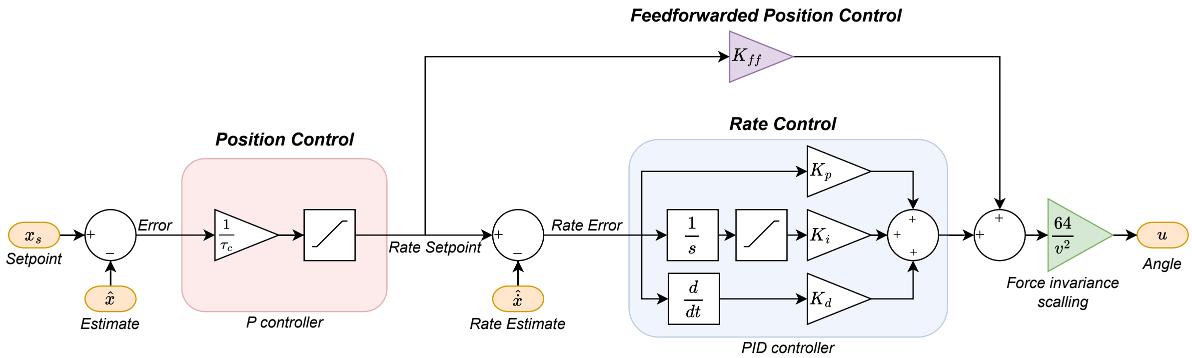


Figure 4: Simplified cascade control scheme (Out of date!)

I'd like to focus for a moment on the two cascaded controllers. The controller in position is a P-controller with gain $K_p = \frac{1}{\tau_c}$. This detail has never been properly explained in previous test reports and it took me quite a while to convince myself. The output of the position controller is then limited in order to limit the rate setpoint and therefore indirectly the boat's rate. The rate controller is a PID controller. As physic saturation of the actuators is not taken into account, the integrator is limited, thus limiting the wind-up phenomenon. Strictly speaking, this is not an anti-wind-up system, because it does not prevent the phenomenon, it only limits its effects. Nevertheless, it does have the consequence of also limiting the integrator in normal conditions. In practice, because the bounds are so tight, the integrator is saturated all the time and the controller is not a PID but simply a P, as shown in figure 5.

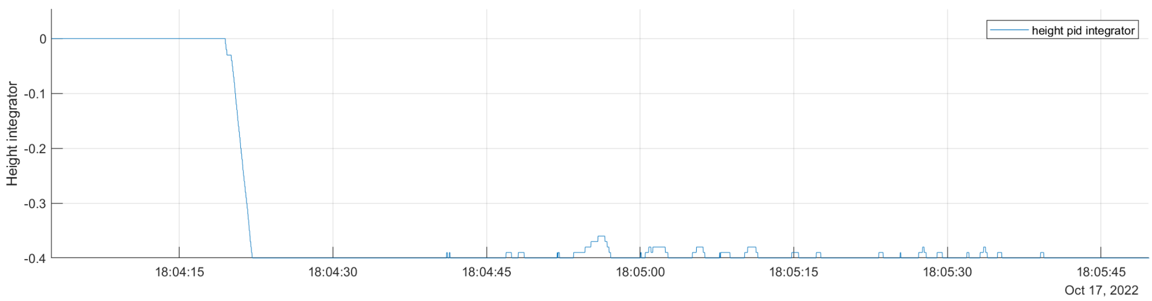


Figure 5: Height rate controller - Integrator saturation - October 17 2022, clean flight

Moreover, the most important detail of the control algorithm, which was never explained in a report, is the feedforward coefficient K_{ff}. It is systematically absent from previous diagrams [1, p. 45][2, p. 30]. I don't think the previous control teams fully understood how it works. This feedforward controller allows the output of the position controller to be applied directly to the foils, thus bypassing the rate control. It can be interpreted as a position controller in parallel with our cascade controller, as illustrated in figure 6. Furthermore, although the units are not identical and therefore directly comparable, the K_{ff} coefficient is typically two orders of magnitude greater than the K_p coefficient, which underlines its crucial importance in controlling the boat.

[1]It cannot be directly interpreted as a physical angle, but the unit is still degrees.

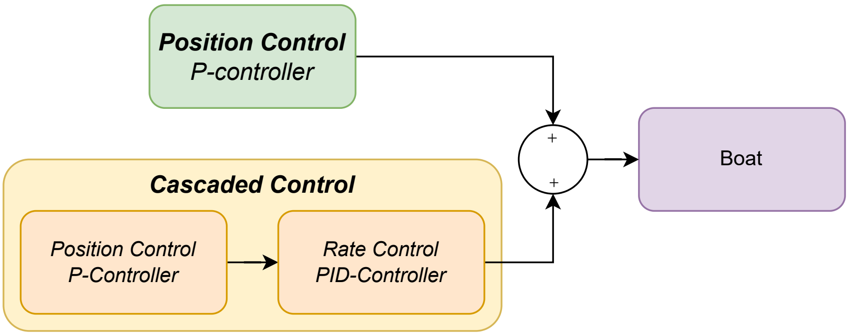


Figure 6: Feedforward controller as a position controller in parallel of the cascade controller

Our system is a multivariable system and although relatively well decoupled, there is still interaction between our different outputs and degrees of freedom. This is taken into account when the controller output, which is an angle, is then passed through our *mixer* to redistribute control across the three foils. This report won't go into it any further. There was some talk of adapting it to the change in centre of mass associated with the integration of the REX, but given the performances, it wasn't judged necessary.

Besides, there are three more parameters added to this command. The steady state $\bar{u}$ (coeff1), which is calculated from an interpolation and scaled to be invariant to speed. The coefficient of *no lift* (coeff2) which is only subtracted from the command when the boat is below its take-off speed. And finally, an offset that can be manually corrected via the groundstation when the boat is on the lake.

Finally, the angle command is converted into motor increment by a linear function ($f(x) = ax$) and bounded by the physical limits of actuation.

As a reminder, this control algorithm has now evolved, and I invite readers to consult the diagram shown in Figure 17 if they are interested in the version implemented at the time of writing.

2.1.3 Method

No up-to-date model of the boat has been able to make its way up to this report, so the methods of adjusting the controllers are all empirical and, to my knowledge, always have been. The methods that have made their way up to this report organically are based on instinct and experience of the boat. Generally speaking, the boat goes out on the water with the settings from the previous tests and as it goes along, the values of the controllers are refined according to the boat's performance and the instinct of the person in charge of control on the vessel that follows the boat. As log analysis is only possible after the test, it is not possible to use live data from the boat for tuning. You have to rely on what you see and on the few values displayed by the groundstation, always with a delay of course. Generally, the only parameters modified by the control manager are : The gains of the rate controllers K_p and K_i, the gain of the feedforward controller K_{ff} and the manual offset of the foil angles.

2.1.4 Performances

The boat has been flying reliably and robustly for two years now. There are obviously a few glitches and a significant improvement margin, with roll control and lateral actuation for example, but on the whole it's a tried and tested system that has proved its worth. As a benchmark for comparing the results obtained this semester, we will use the runs made on 19 April 2022 (figure 7), shortly before the Monaco 2022 competition. These runs were made in ideal conditions, with a very calm lake, on the day the sponsor photo was taken and are considered to be among the best ever made at SSB.

Nevertheless, if we look closely at these results, we can see that the actuators are working intensively and that the motor controllers are often in saturation. Table 6 summarises these benchmark values. A more detailed analysis will be carried out where necessary.

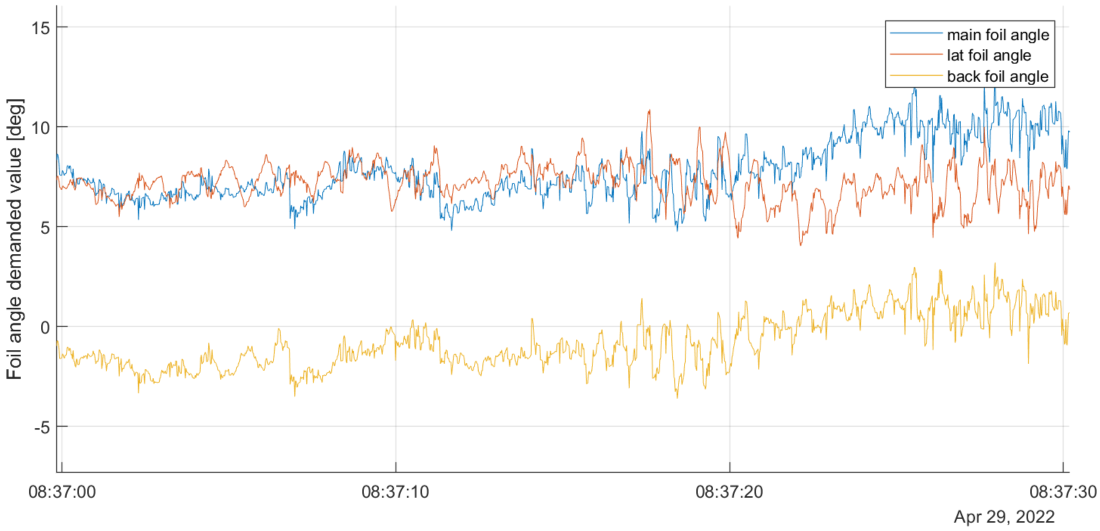


Figure 7: foil angles - April 29 2023 - Stable flight - Benchmark

		Foil Angle [deg]	Motor position [incr]	Motor consumption [W][2]
Main Foil	Mean	7.79	-843	173
	Std. dev.	1.50	82.3	128
Lateral Foil	Mean	7.20	-113	171
	Std. dev.	0.936	7.31	124
Rear Foil	Mean	-0.78	-18.7	32.5
	Std. dev.	1.33	14.3	27.3

Table 6: Benchmark value of interest - April 29 2022 - Stable flight

2.1.5 Conclusion

At the start of this project, we had a complex but well-tried and proven algorithm. On the other hand, there were a lot of blind spots and a lot of ignorance about what was actually going on. It was only after several weeks of work that this level of understanding was reached. That was the starting point for the work, questioning what had been learned and digging into the details of what was going on in order to find the elements that needed to be retained, those that needed to be removed and finally those that needed to be improved. I think this quote sums up the philosophy with which the work was approached:

> *Il semble que la perfection soit atteinte non quand il n'y a plus rien à ajouter, mais quand il n'y a plus rien à retrancher.*
>
> Antoine de Saint-Exupéry, Terre des Hommes

[2]The motor consumption is computed by using the mean value of the battery tension ($44.4[V]$) and not the actual one. Moreover, it uses the current inside the motor, and not the current really consumed by the motor controller. A certain amount of the energy inside the motor may be regained by the controller, thus the value presented here are over-evaluated. However, it is consistent and systematic making them still relevant and meaningful for comparison. True power consumption require the use of additional current sensor, which have been implemented once in December 2022.

2.2 Updating the FCU and documentation

2.2.1 Motivation

The FCU is a monumental code written over several years. In total, there are 9 different authors, 18 Git branches, 153 commits and over 6000 lines of code [3] spread over more than 3 years. There are hardly any lines of comments in the code, and the documentation work that had previously been done had been lost in the maze of SSB's storage spaces. Furthermore, no clean-up work has ever been done on the code. There are many parts or functions that are now obsolete, without having been cleaned up or removed. Some functions have evolved so much since they were first implemented that we've had to tweak the code to the limit. In a nutshell, the FCU needed a major rework to document, unify, optimise and rewrite it.

2.2.2 Implementation

First of all, it was necessary to comment the code in order to *re-discover* how it worked. All the most important files have been commented, although unfortunately not all of them. Next, a major revision of the state machine implementation was carried out. Firstly, to unify and centralise security checks. And secondly, to take ownership of the code and provide the framework for future implementations. It should be noted that no changes have been made to the state machine itself (just to its implementation), and its description in a previous report is still correct ([1, pp. 55–58]). Additionally, although a lot of work has gone into updating the code, the architecture has not changed either. The choice was made to keep the documentation as close to the code as possible, to avoid it getting lost again (I didn't have access to the documentation work done by the previous control team, Yann Boudigou and Maxime Zufferey, until week 7, which is far too late in the project's lifecycle). This is why it now takes the form of a *readme* located directly with the code in the Git repository. When the project is consulted online [4] the *readme* is automatically formatted and displayed on the home page. This means that the documentation can be easily consulted and edited (even directly online), and it will no longer be possible to lose it.

2.2.3 Results analysis

The work done on commenting the code can be directly quantified (table 7), with an increase of more than 600% in the number of lines of comments. However, the documentation work has not been of the same quality, and much remains to be written. The foundations have been laid and it will now be easy to do and maintain, but it hasn't been entirely fulfilled because it's partly redundant with the comment work, and a lack of time, motivation and need was felt.

	line of code	comment line[5]	Comments in percentage
Monaco 2022 - 7.7.22	1774	85	4.6%
Spring 2023 - 3.6.23	1706	574	25.2%

Table 7: Descriptive statistics of the *controller* folder of the FCU code.

The code rewriting has been successful, with no bugs reported so far, and has enabled the rapid and efficient implementation of the features tested this semester. Without this upstream work, it would have been much more difficult.

2.2.4 Conclusion

A healthy code basis has been established this semester. Of course, the code remains colossal and without a substantial personal investment of time, it's not possible to understand it properly. The restructuring work is not perfect and the commentary and documentation work is not entirely accomplished. Nevertheless, it will now be easier to maintain and appropriate.

[3] 6000+ lines of pure code written by SSB, without comments, without blank lines, without lines written by third-party authors (Lib, etc.).

[4] All code written by SSB is stored in an EPFL Gitlab group - https://gitlab.epfl.ch/swiss-solar-boat1 - to which all members have access.

[5] Automatically generated comment have been filtered, Counting is done with the *VS Code Counter* plugin.

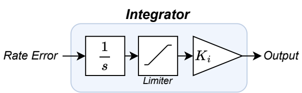
2.3 Antiwindup

First of all, let's start by reminding ourselves what *wind up* is.

> *All actuators have limitations: a motor has limited speed, a valve cannot be more than fully opened of fully closed, etc. For a control system with a wide range of operating conditions, it may happen that the control variables reaches the actuator limits. When this happens the feedback loop is broken and the system runs as an open loop because the actuator will remain at its limit independently of the process output. If a controller with integrating action is used, the error will continue to be integrated. This means that the integral term may become very large or, colloquially, it "winds up". It is then required that the error has opposite sign for a long period before things return to normal. The consequence is that any controller with integral action may give large transients when the actuator saturates[3, pp. 80–81].*
>
> Åström K. J. & Hägglund T. , Advanced PID Control

2.3.1 Motivation

As we saw earlier in the section 2.1.2, the Dahu control algorithm includes three integrators, none of which is protected against a *wind up* effect. But is this really a problem, and does it happen in practice with the Dahu?



As we can see in Figure 8, during a take-off the main foil actuator is saturated for nearly 5 seconds, during which the integrators wrongly accumulate error and thus conclude that *wind up* does happen in practice with the Dahu. This does not seem to pose any substantial problem because the flight and take-off were stable. This can be explained by the fact that the integrator output is bounded by very restrictive limits. These limits effectively mitigate the *wind up* problem, but they have significant drawbacks. As seen in Figure 5, the integrator enters saturation immediately after take-off and does not leave this state for the duration of the flight. The PID controller is therefore no more than a P-controller. Implementing a proper antiwindup solution would make it safe to remove the restrictive limit on the integral action thus it would solve this problem, and the improvement margin seems to be considerable.

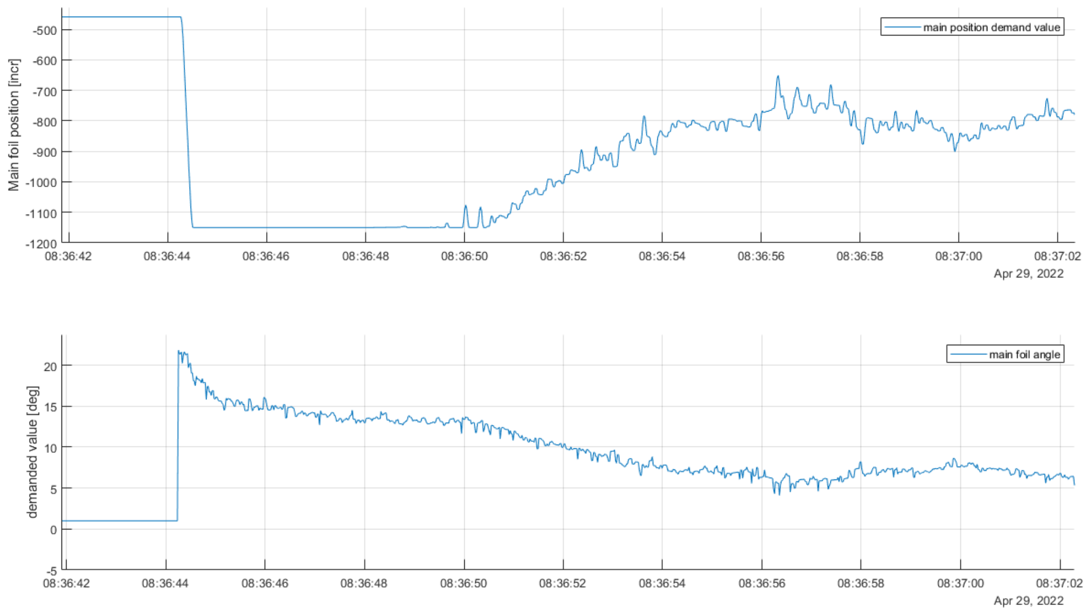


Figure 8: Main foil position vs FCU demanded value - stable take-off - April 29 2023 - Benchmark

2.3.2 Overview of the existing solutions

The first thing to do was to do a literature review of the various existing techniques in order to determine which would be the most appropriate for our case. The first problem encountered was the limited resources available in the literature. Indeed, as the *wind up* problem is a very practical problem, it does not generate much pressure for academic publications and the industries that develop methods for these problems keep them carefully as industrial secrets. "The ideas were often kept as trade secret and not much was spoken about."[3, p. 82].

Nevertheless, five techniques have been identified:

1. Clamping or Conditional Integration [4][3, pp. 82–83, 88–89] [5]
2. Back Calculation [4] [3, pp. 83–87] [5]
3. Proportional Band [4][3, pp. 87–88]
4. Direct Linear Antiwindup (DLAW) [6, pp. 290–300] [7]
5. Model Recovery Antiwindup (MRAW) [6, pp. 300–304] [7]

I will only go very briefly into the details of these solutions, and I invite the reader to consult the references for more information. Techniques 1 to 3 can be operated more or less without a model, while techniques 4 and 5 do require a model. The first method is the simplest and also the most brutal. It simply stops integration when the actuator reaches saturation. Secondly, the back-calculation method is an elegant way of desaturating the integrator with a first-order transfer function and conserving the linearity of the system. However, it requires an additional parameter to be tuned, thus increasing the complexity of the system. Finally, the proportional band method is a more elaborate method of the clamping method.

2.3.3 Choice and Design

All the antiwindup methods mentioned above are designed for a Single Input Single Output (SISO) system and unfortunately I haven't found anything in the literature that comes close to our case: Multiple Inputs Multiple Outputs (MIMO). It was therefore necessary to adapt a technique to our case before being able to implement it. In fact, this adaptation is far from trivial, as it is not possible to directly identify the integrator responsible for the saturation of an actuator, as the three integrators control the three actuators. The DLAW and MRAW methods require a model, and we didn't have a reliable model of the boat, so it was possible to discard these two techniques directly. Then, the Back Calculation method increases the complexity of the system and its tuning, so it was judged more prudent to discard it as well at this stage. The two methods developed are therefore an extension of the clamping and proportional band method.

Restrictive Clamping

This is the simplest form of antiwindup you can think of for a multivariable system. When one of the three actuators reaches saturation, integration is stopped for all three controllers. The *wind up* phenomenon is therefore effectively prevented, but it is also the most restrictive policy imaginable. In fact, the controllers are blocked no matter which direction they integrate, and although one of the three actuators is saturated, it would still be possible to increase the control action using the other two actuators. In other words, three degrees of freedom are blocked when one would be sufficient.

Saturation Redistribution

The principle of this technique is to subtract the excess control action that causes saturation from the integrator, in order to maintain the system in its linear regime, out of saturation. The problem in a MIMO system is to know who is responsible for the part of the control action that causes saturation... The idea was therefore to pass the saturation part through the mixer's inverted matrix.

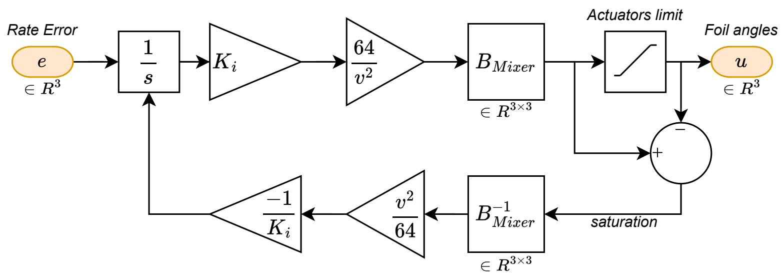


Figure 9: Saturation redistribution antiwindup scheme

This technique has the advantage of not being conservative and allowing maximum use of the degrees of freedom of the system. On the other hand, it is relatively complex to understand, and its functioning is opaque. I am firmly convinced that this technique works and that it does not pose any stability problems, having examined it in a few simple cases. To be perfectly rigorous, we would have to deal with the 6 possible cases of saturation when only one actuator saturates, which would correspond to the 6 faces of a cube, there would still be the 12 edges (when two actuators saturate) and the 8 corners (when three actuators saturate), which I haven't done. Unfortunately, I haven't found any mention of this technique in the literature and I have no evidence that it works beyond my intuition.

To summarise, in the past, integrators were bounded in a very restrictive way and in practice the controller was only a P-controller. Two techniques were developed, the first more restrictive but also more intuitive and easier to understand. A second technique, this time not restrictive at all but more complex and opaque, with no support in the literature for its effectiveness. For these reasons, it was decided to test the first solution: restrictive clamping because it is simple to implement, potentially more reliable and because the incremental gain between solutions one and two is small compared to to the risk taken.

2.3.4 Implementation and testing

The antiwindup was implemented in the middle of the semester, before the tests in week 9. The saturation check is done in the *flightStack* class, after each of the control actions has been added. Note that the minimum and maximum angle ranges are hard-coded at this point and that if a physical change were to be made to the actuation, it would be necessary to make a change at this place too. Conditional integration is implemented in the *PID* class and is a little more permissive than in its original formulation above, as we only block integration in the direction that increases the control action[6].

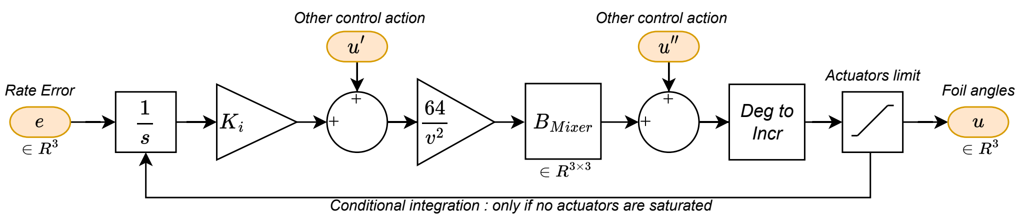


Figure 10: Restrictive Clamping antiwindup scheme - implemented

The implementation was tested on land during the preparation of the tests using the test mode[7] of the boat. It works perfectly, with one exception: the constant offset of one degree between the theoretical and measured limits. One possible explanation for this is that the C++ implementation of the *abs* function or the $<$ comparator with an object of type *Eigen* doesn't quite work as intended. However, once you are aware of this mismatch, it poses no practical problem. All you have to do is set the limit in the code to $[-3°; 12°]$ if you want the effective limit to be $[-4°; 13°]$. Although this solution lacks elegance, due to

[6]This is not perfectly rigorous, and we could imagine a case where this would pose a problem (due to the interaction between several controllers with control actions of opposite sign). However, it is more likely that this relaxation will not cause any problems and will reduce the restrictivity of the controller

[7]The complete boat control algorithm is active and a virtual speed, proportional to the throttle command, is applied

time constraints and lack of necessity, no further effort was devoted to resolving this issue. The real-life tests with the boat foiling on the lake were carried out in week 9 during the tests on the lake.

2.3.5 Results analysis

Tests on the lake in week 9 gave excellent results. However, as several parameters were modified simultaneously, it is not possible to quantify the impact of the antiwindup specifically. It should be noted, however, that in practice the control action of the integrators was multiplied, and the typical values are shown in table 8.

Controllers	Height rate	Roll rate	Pitch rate	
Integrator bounded value (Benchmark)	0.4	1	30	[deg][8]
Integrator typical value (April 19 2023)	-1.1	12	2.5	[deg]
K_I (Benchmark)	0.1	0.06	0.02	
K_I (April 19 2023)	7	0.45	0.1	
Augmentation	7000%	750%	500%	

Table 8: Integrators typical value - Benchmark and after antiwindup implementation (April 19 2023)

Looking at these values, it would be tempting to say that the operating ranges have been increased, sometimes by a factor of 10. But in reality, it's even better than that, it's the whole integral action that has been recovered and which no longer saturates, as illustrated in figure 11 compared with the previous results illustrated in figure 5. The controller now behaves effectively as a PI-controller and not just as a P-controller. Furthermore, previously the person in charge of the control would set the gain K_I, thinking he was setting the integral action, but the latter, when saturated, did not behave as such, but rather as a constant offset. This had the effect of disturbing the tuning process, and I'm convinced did more harm than good to the control. This effect is measurable. In order not to disturb the control too much with a constant offset, the integral gain K_I was typically small, the value being found empirically by the people in charge of control judging that the boat behaved better that way. During the tests in week 9, it was possible to increase the integral gain K_I by up to two orders of magnitude, allowing the integral action to finally play its role. The overall flight performance of the Week 9 tests will be analysed in a later section.

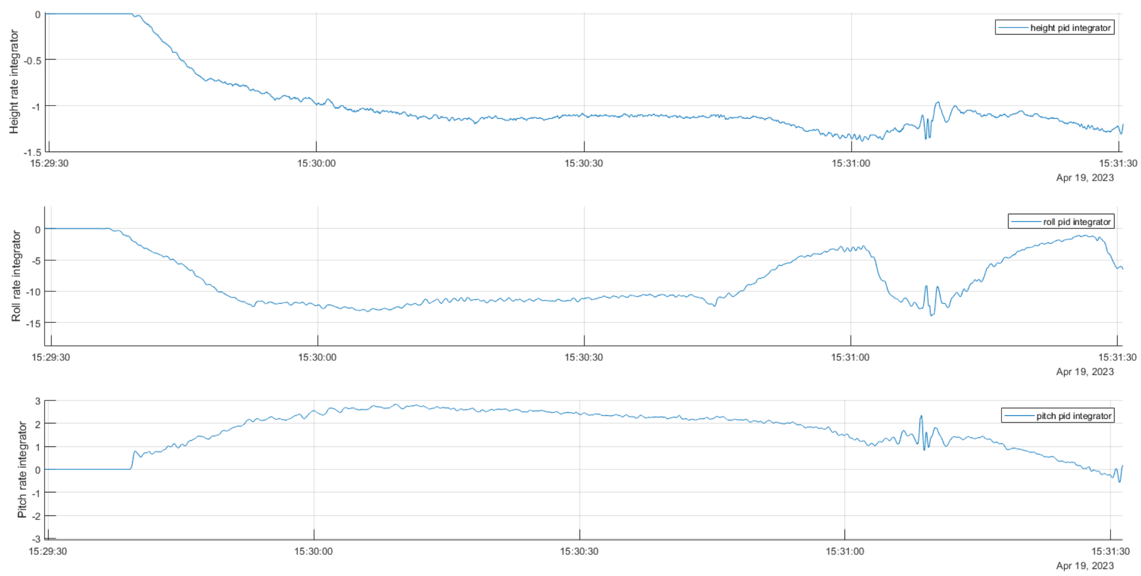


Figure 11: Integrator value - Stable fight April 19 2023 - with additional 80$[kg]$ load

[8]The integrator unit is degree, nonetheless it cannot be directly interpreted since it is latter scaled and spread among the three foils.

2.3.6 Conclusion

The conclusion of this section is simple: implementing an antiwindup system has greatly benefited the boat's control. The implementation of an antiwindup safety measure has made it possible to get rid of the restrictive limit on the integral action without risking the safety of the controller and therefore indirectly of the boat and its pilot. It also allowed the integral action to be restored, as well as reducing disturbances and opacity in the boat's tuning, which had excellent results on the performances.

2.4 Empirical Boat Model

2.4.1 Motivation

Until now, all boat tuning techniques at SSB have been empirical. These methods have advantages of convenience, but if we wish to turn to more modern, more rigorous methods of tuning we need a model. Furthermore, without a model, it is not possible to quantify the robustness of the controller, or to establish an estimator. The motivation for having a model of the boat is therefore clear: to be able to do quantitative work rather than qualitative work. However, this is not a trivial task. A model of the boat had previously been developed when the boat had not yet been launched and maintained for some time by the previous control teams. However, this was not enough and at the start of the project this work was buried deep in the maze of SSB's storage spaces and no longer usable in its current state. Reviving this work is a major task, which is not the subject of this project, but of another semester project being carried out simultaneously at SSB by Noé Tambourin and Maxime Teuber, who are in charge of the *Design of a multi-physic simulator to enhance modeling and control of the new boat.* Making a theoretical model that was as complete and reliable as possible was therefore out of the question. However, another approach was possible. To develop a model based on the empirical data now available for the boat, which had never been used in this way. The aim was to quickly have a simple model on which to verify our various theories as well as checking the margins of stability and the values of the controllers.

2.4.2 Design

Model Definition

The simplest model we can think of to use the tools of control theory is a Linear Time Invariant (LTI) model, in its state-space representation:

$$\dot{x} = Ax + Bu \tag{2.4.1}$$

As a reminder, the FCU only controls 3 degrees of freedom of the boat: height, roll and pitch. A reduced model with 6 state variables is therefore sufficient to describe the complete system. We will use the usual control theory notation. In addition, we will assume the state space measurement and therefore the output $y = x$.

State	x	$[h, \dot{h}, r, \dot{r}, p, \dot{p}]$	height $[m]$, height rate$[\frac{m}{s}]$, roll $[deg]$, roll rate$[\frac{deg}{s}]$, pitch$[deg]$, pitch rate $[\frac{deg}{s}]$
Input	u	$[\alpha_M, \alpha_L, \alpha_R]$	Main foil angle $[deg]$, Lateral foil angle $[deg]$, Rear foil angle $[deg]$

In order to solve for the three degrees of freedom of our system $[h, r, p]$, we will use three equations of the dynamics. These are second-order differential equations (equations 2.4.2 to 2.4.4). In order to be able to write them as LTI systems, we will transform them into pairs of ordinary first-order differential equations[9], introducing in the process 3 additional intermediate variables $[\dot{h}, \dot{r}, \dot{p}]$.

$$\sum \vec{F} = m\vec{a} \quad \Rightarrow \quad m\ddot{h} = M + L + R - mg \tag{2.4.2}$$

$$\sum \vec{M} = J\vec{\dot{\omega}} \quad \Rightarrow \quad J_R\ddot{r} = L_M M + L_L L + L_R R \tag{2.4.3}$$

$$\sum \vec{M} = J\vec{\dot{\omega}} \quad \Rightarrow \quad J_P\ddot{p} = L'_M M + L'_L L + L'_R R \tag{2.4.4}$$

The above equations make a number of assumptions, considering the most simplistic case. They ignore the influence of the boat's attitude on the moments and forces as well as the various interactions that may exist. Starting with a simple model is always a good idea, but we'll see later if the equations have managed to grasp the complexity of reality. In addition, 12 new parameters have been introduced:

[9] Classic double integrator problem

M, L and R, which are respectively the lift forces of the main foil, lateral foil and rear foil. Then m, J_R, J_P which are respectively the mass, roll inertia & pitch inertia of the boat. Finally, the last 6 parameters L_{xx} are the distances that allow the forces of the foils to be reported as momentum to the boat's centre of mass.

Before putting everything together to form the simplified model of the boat, a final assumption is made:

$$M = C_m v^2 \alpha_m, \quad L = C_l v^2 \alpha_l, \quad R = C_r v^2 \alpha_r \tag{2.4.5}$$

In other words, the lift force of the foil is a function of the foil's angle of attack [10], the speed squared and a parameter C that embeds all the complexity of the foil. This hypothesis is acceptable over a certain range of speeds and angles of attack, as well as in quasi-static conditions where the variation in angle is negligible compared with the flow velocity.

The simplified model of the boat can now be developed:

$$\underbrace{\begin{bmatrix} \dot{h} \\ \ddot{h} \\ \dot{r} \\ \ddot{r} \\ \dot{p} \\ \ddot{p} \end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} h \\ \dot{h} \\ r \\ \dot{r} \\ p \\ \dot{p} \end{bmatrix}}_{x} + \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ \frac{C_m}{m} v^2 & \frac{C_l}{m} v^2 & \frac{C_r}{m} v^2 \\ 0 & 0 & 0 \\ \frac{C_m L_m}{J_R} v^2 & \frac{C_l L_l}{J_R} v^2 & \frac{C_r L_r}{J_R} v^2 \\ 0 & 0 & 0 \\ \frac{C_m L'_m}{J_P} v^2 & \frac{C_l L'_l}{J_P} v^2 & \frac{C_r L'_r}{J_P} v^2 \end{bmatrix}}_{B} \underbrace{\begin{bmatrix} \alpha_m \\ \alpha_l \\ \alpha_r \end{bmatrix}}_{u} \tag{2.4.6}$$

Parameter Identification

A reduced model with 6 states, 3 inputs and 12 unknowns has been introduced. It is now time to identify the unknowns. To do this, we will use the data from the boat during the tests. In addition, we are actually only interested in the 9 unknown terms of the matrix B, reducing the number of unknowns from 12 to 9.

In order to identify these parameters, we will take advantage of the linear formulation of our system and perform a linear regression. I won't go into the details and subtleties of such a regression [8]. First of all, let's reformulate our problem so that it can be used in a linear regression. Let's start with the height equation:

$$\ddot{h} = -g \frac{C_m}{m} v^2 \alpha_m + \frac{C_l}{m} v^2 \alpha_l + \frac{C_r}{m} v^2 \alpha_r \tag{2.4.7}$$

It is now possible to define the dataset (X,y) of interest:

$$X = [v^2, \alpha_m, \alpha_l, \alpha_r] \Rightarrow [x_1, x_2, x_3, x_4], \quad y = \ddot{h}$$

This leads us to the following formulation (for the i-th datapoint) of our linear regression:

$$y_i = \beta_0 + \beta_1 x_{i,1} x_{i,2} + \beta_1 x_{i,1} x_{i,4} + \beta_1 x_{i,1} x_{i,4} + \epsilon_i \tag{2.4.8}$$

ϵ_i being the error between the regression and the true value of y_i. Linear regression attempts to minimise ϵ by minimising an error calculating function, typically the sum of squared errors. If the model is accurate then we have identified the following parameters:

$$\begin{aligned} \beta_0 &= -g \quad &&\text{, used as a verification parameter} \\ \beta_1 &= \frac{C_m}{m} \\ \beta_2 &= \frac{C_l}{m} \\ \beta_3 &= \frac{C_r}{m} \end{aligned}$$

We have now formulated our first regression. However, the first step in any data analysis is to prepare the data correctly, and a model cannot do better than the data it is based on: *Garbage in - Garbage out.*

[10] calculated with respect to the plane perpendicular to the hydrodynamic appendix

The data we have at our disposal are the Dahu logs, with all the noise they contain, missing data, etc. The first problem was to obtain reliable angle values. As the actuators were not systematically calibrated, these values fluctuated greatly from one test to the next and could not give a reliable model. We had to wait until the tests in week 9 to obtain usable data. More details are given in the section 2.5. The second problem was to match the data through time. The model is based on the assumption that all the measurements are taken at the same time. In practice, each sensor sends its measurements independently and there is no synchronisation. The on-board computer, which logs all the data, then associates a time stamp with the data it receives. Note that there may be a delay in this timestamp due to the priority and arbitration system for sending CAN messages, and this cannot be determined, as well as an arbitrary delay in the measurement depending on the sensors. In order to obtain sufficient data, we matched the data to the same datapoint when the difference between the timestamps was less than $10[\frac{m}{s}]$. This is a compromise between the accuracy of the data and the number of data points available for regression. We then matched the data only during stable phases of flight, when altitude and speed were reasonable, in order to remain within the *a priori* validity limits of our model. The data preprocessing is done in a matlab code (Appendix A.1). Once the data was preprocessed, it was finally possible to run the linear regression. To do this I wrote a python script (Appendix A.2) and used 4 libraries: Pandas to manipulate the data[9], Seaborn for visualisation[10][11], Numpy for mathematical operations[12] and Statsmodel for the linear regression tools[13]. I won't go into detail about the roll and pitch regressions. The approaches are strictly equivalent. The data used are those from April 18 2022, first run of the afternoon.

2.4.3 Results analysis

In general, the results are not conclusive for the regression on vertical acceleration. The verification parameter is well identified ($\beta_0 \approx -g$), but the three parameters of interest are not. We can see that the effect (coeff) is small, of the order of 10^{-4}, and that it is not statistically significant (P> |t| is large). This can be explained by the fact that the effect we are trying to measure is too small compared to the sensor noise, or by the fact that the assumptions under which we designed the model may not be valid. Moreover, the vertical acceleration of the boat is almost zero, as well as the variation in the angle of the foils[11], which makes it difficult to detect any effect. I don't think we can do any better with the data we have right now. We would have to generate data where the effect is more visible, for example with a step variation in the height setpoint at several speeds during a test on the lake. However, this involves a high risk for the pilot and the boat and I'm not sure that the risk/benefit ratio is particularly interesting.

Dep. Variable:	$\ddot{h}$	**R-squared:**	0.007
Model:	OLS	**Adj. R-squared:**	0.002
Method:	Least Squares	**F-statistic:**	1.351
Date:	Mon, 05 Jun 2023	**Prob (F-statistic):**	0.257
Time:	11:14:23	**Log-Likelihood:**	-339.00
No. Observations:	612	**AIC:**	686.0
Df Residuals:	608	**BIC:**	703.7
Df Model:	3		
Covariance Type:	nonrobust		

parameter	**coef**	**std err**	**t**	**P> |t|**	**[0.025**	**0.975]**
$\beta_0 : Intercept$	-9.8116	0.086	-114.009	0.000	-9.981	-9.643
$\beta_1 : v^2\alpha_m$	0.0010	0.001	1.559	0.120	-0.000	0.002
$\beta_2 : v^2\alpha_l$	-0.0002	0.000	-0.688	0.492	-0.001	0.000
$\beta_3 : v^2\alpha_r$	-0.0009	0.001	-1.551	0.121	-0.002	0.000

Table 9: Results of linear regression on height acceleration

Next, for regression on roll acceleration, the parameters of interest linked to the angle of attack of the main and lateral (β_1, β_2) give good results. The effects are not ridiculous, but above all these values are statistically significant (small p-value, confidence interval of the same sign, large t-statistic). Furthermore, the effects of the main foil and the lateral foil are of opposite sign, as expected. In contrast, the rear foil coefficient is not significant. This is partly expected, given the small influence of the rear foil on the roll. Nevertheless, these three effects are two orders of magnitude smaller than the $\beta_0[\frac{deg}{s^2}]$ intercept. Here, we

[11] variance of foil angle was below 0.1°on the dataset used

would have expected this coefficient to be close to zero, as it represents the sensor bias. This huge value is partly explained by the fact that the roll acceleration is obtained by double derivation of the gyroscope signal and is therefore potentially very noisy. The first derivation is done by the inertial central unit itself, but the second was done by myself, using the forward Euler formula to derive a discrete signal. The sources of error are therefore numerous and consequent. To conclude, the relatively good results obtained for these coefficients need to be balanced against the magnitude of the sensor bias. I would say that although potentially fraudulent, these results can be used and that there is a good margin for improvement if we were to calibrate the sensor or filter the data better.

Dep. Variable:	$\ddot{r}$	**R-squared:**	0.047
Model:	OLS	**Adj. R-squared:**	0.042
Method:	Least Squares	**F-statistic:**	9.870
Date:	Mon, 05 Jun 2023	**Prob (F-statistic):**	2.31e-06
Time:	12:05:04	**Log-Likelihood:**	-3176.8
No. Observations:	611	**AIC:**	6362.
Df Residuals:	607	**BIC:**	6379.
Df Model:	3		
Covariance Type:	nonrobust		

parameter	**coef**	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
$\beta_0 : Intercept$	-17.8672	8.959	-1.994	0.047	-35.462	-0.272
$\beta_1 : v^2\alpha_m$	0.2903	0.068	4.241	0.000	0.156	0.425
$\beta_2 : v^2\alpha_l$	-0.1276	0.031	-4.123	0.000	-0.188	-0.067
$\beta_3 : v^2\alpha_r$	-0.0531	0.059	-0.905	0.366	-0.168	0.062

Table 10: Results of linear regression on roll acceleration

The results of the last linear regression on pitch acceleration are similar to those of the regression on roll acceleration. This time it's the main foil and rear foil coefficients that are relevant, and their effect is not negligible. The effect of the lateral foil is very small, but this is partly expected as it has little influence on the pitch. The intercept is again large, indicating a strong bias on the sensor. The conclusions to be drawn are similar.

Dep. Variable:	$\ddot{p}$	**R-squared:**	0.037
Model:	OLS	**Adj. R-squared:**	0.032
Method:	Least Squares	**F-statistic:**	7.743
Date:	Mon, 05 Jun 2023	**Prob (F-statistic):**	4.41e-05
Time:	12:24:09	**Log-Likelihood:**	-2419.2
No. Observations:	611	**AIC:**	4846.
Df Residuals:	607	**BIC:**	4864.
Df Model:	3		
Covariance Type:	nonrobust		

parameter	**coef**	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
$\beta_0 : Intercept$	-9.3963	2.593	-3.624	0.000	-14.489	-4.304
$\beta_1 : v^2\alpha_m$	0.0954	0.020	4.818	0.000	0.057	0.134
$\beta_2 : v^2\alpha_l$	-0.0068	0.009	-0.765	0.445	-0.024	0.011
$\beta_3 : v^2\alpha_r$	-0.0515	0.017	-3.035	0.003	-0.085	-0.018

Table 11: Results of linear regression on pitch acceleration

2.4.4 Conclusion

To conclude, I'd say that the approach was interesting and that it was worth trying to make use of this data in a way that we'd never done at SSB. It enabled us to illustrate the strong bias of the inertial measurement unit[12] on pitch and roll acceleration. With a good calibration, as well as data filtering, I think this approach could give excellent results for identifying roll and pitch parameters, although I

[12]This is the IMU Ellipse2-N inertial navigation sensor

consider that it can be used cautiously in its current state. On the other hand, the values obtained for height parameters are not at all usable, which makes our model as a whole difficult to use. The obstacles to overcome in order to obtain something acceptable in terms of height appear to be considerable and no further work has been undertaken in this direction, as this is also, I would remind you, the subject of another project which is running this semester. As a result, we have had to abandon the hope of using quantitative methods, at least for this semester, as well as setting up an estimator to measure the state of the boat, which also requires a model.

2.5 Foil calibration

2.5.1 Motivation

The Dahu has three foils that are mechanically mounted on the verticals. They are driven by mechanical transmission which transmits the rotation of the actuation motors to the foils. The actuation motors are position-controlled *Maxon* motors. In short, the motor controller is given a reference in increments and then, using its own control system, it moves the motor to the desired position. The transmission system and the actuation in general are frequently disassembled and reassembled and there are small variations from one time to the next, which changes the angle of the foil in relation to its equivalent in increments. We can see in figure 12 that during three stable flights spaced out in time (2021, 2022, 2023) we obtain very different average angle values. Obviously, there have been some physical modifications to the boat which may explain a small part of these differences. However, it is much more likely that the majority of these differences are explained by a difference in the actuation set-up and therefore in the conversion between foil angle and motor increment rather than in the physical angle of the foil. The fact that the boat could have flown with a negative main foil angle should convince you.

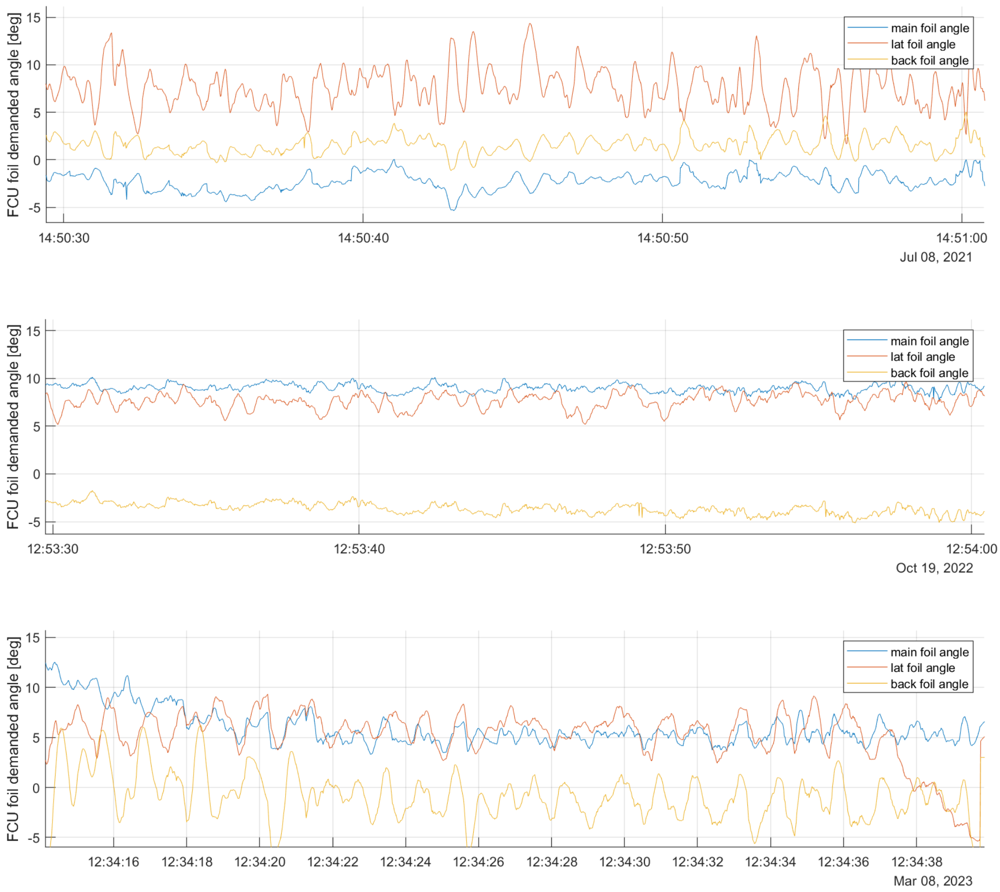


Figure 12: FCU demanded foil angle in stable flight - July 2021, October 2022, march 2023

Indeed, re-calibrating the conversion between foil angle and engine increments is not (or is no longer) part of SSB's standard procedures and we need to incorporate it into our practices. A disparity between

	Main foil	Lateral foil	Rear foil
Monaco 2021	-3°	8°	2°
October 2022	9°	7°	-4°
March 2023	5°	5°	-2°
April 2023 (with calibration)	3°	4°	2°

Table 12: Mean value of foil angles during stable flight

the angle commanded by the FCU and the value of the physical angle poses problems for control because certain values such as *coeff1* or *coeff2* shown in figure 3 are based on theoretical values of the foils. They represent $\bar{u}$ and the no-lift angle respectively. But now, because of a disparity between the physical angle and the requested angle, they no longer represent anything at all and it's becoming harder to know what's actually happening and which element is responsible for which behaviour on the boat. On the one hand, they disturb the behaviour of the boat and, on the other, they make the work of the person in charge of control more difficult. In addition, if you can't rely on the angles you read, you can't draw conclusions from them either, for example when building a model or optimising a foil. Correctly calibrating each foil before setting off on the lake therefore seems to be an important point.

2.5.2 Implementation

A calibration procedure has been drawn up and can be found in the appendix B. First of all, we had to rediscover how the angle value is converted into a motor increment. In the FCU code, in the *boat* class during initialisation (*boat.cpp* file) the conversion values are defined, originally *MinInc, MaxInc, ratio-inc-deg)*. The conversion algorithm is illustrated in figure 13. This algorithm is practically inexplicable and far too complicated. It is indeed a linear function ($f(x) = ax + b$) but we are not able to correctly identify the zero, nor to change the y-intercept. The concept behind it, if I've guessed correctly, was to set the zero increment to correspond to the physical zero. It's possible to do this through the software used to program the motor controller, but the procedure is fastidious, opaque and prone to errors. The slope a is updated in the FCU, whereas the y-intercept b is defined in the *Maxon* controller...

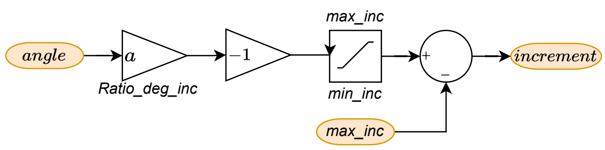


Figure 13: Angle to increment conversion Scheme at the beginning (Out of date!)

Instead, it was decided to do something much simpler and more robust, i.e. simply make a linear function ($f(x) = ax+b$). All the parameters are defined in the FCU code to standardise and facilitate the procedure. Bounding the output is done only once and at the very end, to avoid errors that compromise the hardware integrity of the actuation. The algorithm is illustrated in Figure 14 and implemented in the *maxon-position-motor* class.

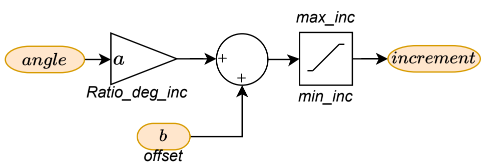


Figure 14: New angle to increment conversion Scheme

A second difficulty was to define the angle of the foil properly. It's a complex profile and defining a plane on which to measure the angle is not at all trivial. For this, I'd like to thank Mattia Valitutti, who designed and 3D-printed some foil fitting blocks. They follow the profile of the foils and give a reliable and consistent reference from one time to the next. In addition, we had to buy an electronic level to be able to read the angles reliably. The reference is made with respect to the horizontal, so you have to make sure that the boat is flat (or compensate the measurement with the angle values read by the IMU).

2.5.3 Result analysis

The results are unequivocal. During the tests in week 9, the first time the calibration was carried out properly, the foil angle values during the flight phases were much closer to the theoretical values initially predicted[14]. This can be seen in figure 15.

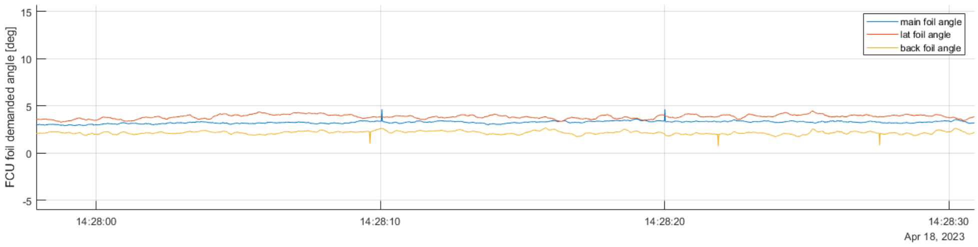


Figure 15: foil angle - Stable fight April 18 2023

Furthermore, we are now certain that the angles demanded by the FCU correspond to the physical angles of the foils. This makes the data usable for other applications, such as model development from section 2.4. Additionally, the control system has been particularly efficient since then. As several changes have been made simultaneously, it's not possible to assess the impact of a good calibration accurately. However, it is reasonable to think that it has only made the work of the control team easier.

2.5.4 Conclusion

In order to calibrate the foils properly, we had to do a bit of clean-up in the FCU code and practices at SSB. On the one hand, we didn't have the equipment to do it properly. On the other hand, nobody knew how to do it properly and where to change which parameter. For example, there was no record and no one could justify why *maxInc* was being subtracted or why there was an offset hard-coded in the *Maxon* controller. Questioning practices in order to document and unify them has certainly been a good thing and will lead to greater reliability in the future. Finally, although the way in which the foils are calibrated has been challenged, this has not been the case for the linear function which links an angle to an increment. This is justified for the rear foil, whose conversion is linear due to its mechanical mounting with a ball screw. However, it's only barely justified for the lateral actuation and its connecting rod. In my opinion, this is an area where there's still considerable margin for improvement.

2.6 Lake tests

2.6.1 Motivation, context and timeline

Swiss Solar Boat is an association whose vocation is to sail boats, so tests on the lake are naturally a considerable part of our work. Organising such tests is a major logistical operation, with dozens of students involved over several weeks. In the spring semester of 2023, three test sessions took place, in week 3, week 9 and week 11. Note that the week 11 tests will be dealt with in a later section as they relate to the REF. This report concentrates on the control aspects of the tests, although it should be remembered that a great deal of electronic and mechanical work is required to get the boat up and running. The times when it's fully functional and when it's possible to test things in condition are very limited, usually just a few hours while off the water. The main aim of these tests is to prepare the boat for the Lausanne-Evian crossing with 50kg of extra weight.

2.6.2 Test on week 3 - Objectives

The objectives of the tests in week three are relatively limited, the aim being simply to get the boat working, taking off and flying reliably, which it hasn't done since October 2022 in a reliable way. The main problem at the time that prevented us from achieving this goal was our ignorance of the system... So the objectives were to get to know the control system properly and to invest enough time in fine-tuning the controllers.

In addition, the lateral control was constantly in saturation at the motor control level. The lateral actuation motor controller, in an attempt to follow the increment reference imposed by the FCU, almost systematically delivered the maximum current, in order to deliver the maximum torque on the foil.

Under normal conditions, this didn't pose any problems. However, this maximum current ($10[A]$) is only permissible for a limited period of time ($\sim 30[s]$), after which the motor by safety [13] limits itself to the rated current ($5[A]$). In this case, the motor torque is not sufficient to follow the reference imposed by the FCU, the control loop is broken (open loop, the actuator is saturated) and the reference is not followed. This has a very harmful effect on the boat, as the axis of the roll is the dangerous axis, the side floater can either crash into the water, or in the worst case scenario go over the main hull and capsize the boat... At the time, several theories were put forward to explain the problems with the lateral actuation, one being that the actuator was simply undersized.

2.6.3 Test on week 3 - Implementation

As these tests came very early in the semester, no implementation had yet been made with regard to the control, the time having been invested in reading the reports, understanding the code and preparing the boat for the launch.

With regard to lateral actuation, a spring has been implemented by the *hydro* team to preload the motor so that at nominal speed, it does not have to give the torque required to carry the boat. The aim was to reduce the effort made by the controller and therefore avoid saturation.

2.6.4 Test on week 3 - Result analysis

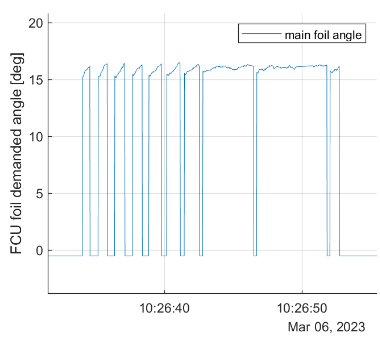
Actuators saturation - Coeff1 and Coeff2

On the first day of these tests, we found that on take-off, the angle demand on the actuators was too high, immediately saturating the actuators. We were able to identify the source of this problem, which was the $\bar{u}$ steady-state command added to the control action at take-off and during the flight phase.

$$\bar{u} = \frac{Coeff1}{v^2} \tag{2.6.1}$$

Because of the scalling of $\bar{u}$ by the speed squared, the control action is very large during take-off where the speed is much lower ($5.2[\frac{m}{s}]$) than the nominal cruise speed ($8-9[\frac{m}{s}]$). Added to this is the fact that on take-off, there is a step change in the setpoint and therefore a large proportional error (in position and rate) and therefore also a large cascade control action from the controller. These two effects combine to saturate the actuators on take-off.

Saturating[14] the actuators is never a good idea, it breaks the control loop and can cause a number of problems. In this case, when everything is going well, it causes a very aggressive take-off, as well as an overshoot. The overshoot is partly compensated for by a high proportional gain[15]. However, once the boat is in the air and the overshoot begins, the error changes sign and the action of this gain effectively counteracts the excessive action of $\bar{u}$, until the boat reaches its nominal speed. This is not quite an appropriate behaviour and there is a considerable margin for improvement.



By design, the controllers are only activated when a minimum take-off speed is reached ($5.2[\frac{m}{s}]$). The foils are relatively flat before, to limit drag and facilitate acceleration. Then, once the controllers are active, the boat is supposed to take off. A hysteresis effect has not been included. During these tests, a funny effect was observed. The boat rocked around the take-off speed. At the moment of take-off, the foils saturated upwards, creating lift but also drag, which slowed the boat down considerably. It slowed down below take-off speed and the foils returned to their flatter position, accelerating the boat and so on, as illustrated in the figure alongside.

To solve this problem of back and forth movement, as well as actuator saturation, we set the coefficients $Coeff1$ and $Coeff2$ ($\bar{u}$ and *no drag*) to zero. From then on, the actuators no longer entered saturation, and the boat was able to take off calmly and stably. As a reminder, at this point the actuators had not been properly calibrated, with all the problems we raised in the 2.5 section.

[13] Protection parameter I^2t

[14] Please note that the term saturated is used extensively in this section, however it does not always refer to the same saturation. Here it refers to the foils being at mechanical limit.

[15] The gain K_{ff} which feeds the proportional action of the position controller to the system.

Lateral actutation - motor saturation

Generally speaking, these tests were not very conclusive. The conditions were difficult and the lake rough. The boat rose well and flew very steadily through the waves. However, the lateral actuation motor controller systematically went into saturation after about thirty seconds, breaking the control loop and causing the lateral to crash. The most spectacular example can be seen in Figure 5, where the setpoint could no longer be tracked by the motor, causing the boat to capsize.

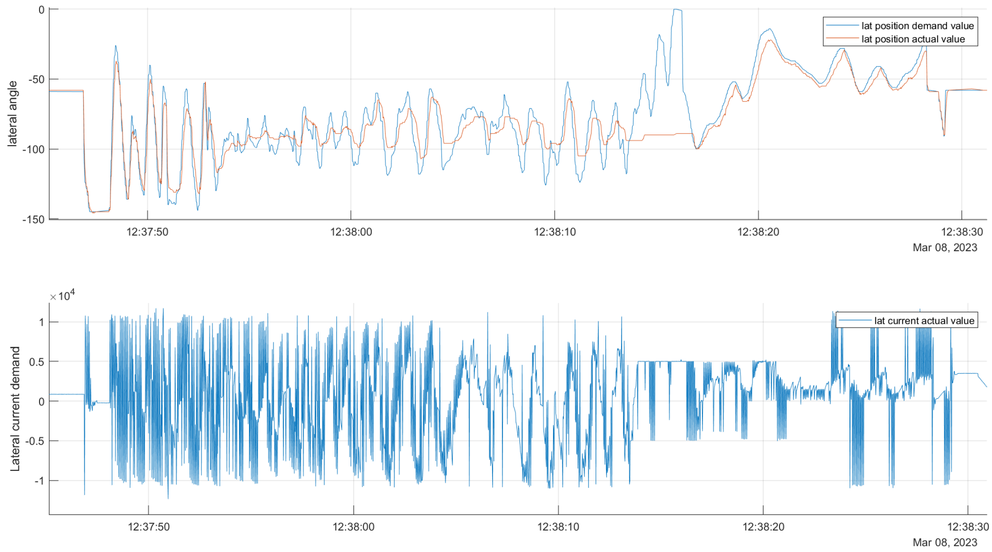


Figure 16: Saturation of the lateral motor controller - March 2023 - The Dahu capsized

The lessons are clear. On the one hand, the spring doesn't help, because this situation happens in all cases, even faster with the spring. On the other hand, the motor controller is undersized to follow the imposed reference. Nevertheless, on visual inspection of the graph, it seems that the control action is too aggressive and that we're off-tunig, close to the instability limit. This is an intuition that cannot be supported by values. Better tuning would allow less aggressive control and therefore less use of the actuator, avoiding engine saturation.

2.6.5 Test on week 3 - Conclusion

In conclusion, these tests enabled us to gain experience with the operation of the control algorithm and to identify a number of blocking points and areas for improvement. Removing the parameters ($Coeff1$ and $Coeff2$) has already simplified the control algorithm. Removing these elements allowed the boat to take off. Although theoretically justified, their practical application was not entirely within the framework in which it had been theoretically defined nor the implementation was well executed. They posed more problems than they solved.

2.6.6 Test on week 9 - Objectives

The week 9 tests had consequent objectives for control, with a number of new implementations. First of all, there's the major work of restructuring the state machine and the security checks, which need to be tested, verified and proven. In addition, antiwindup has been added. This feature also needs to be tested. Implementing antiwindup will drastically change the way the boat is controlled, since by removing the strict bounds on the integrator, it will allow the integral action of the controllers to be restored and therefore control in PI rather than P. This will require a great deal of tuning work. In addition, this is the first time that an absolutely rigorous foil calibration procedure has been implemented, even modifying the physical mounting of the actuators where necessary in order to restore them to their originally intended range of actuation. Being able to rely on the angles read as well as knowing whether the planned range is appropriate are also two objectives. In addition, a new motor controller for lateral actuation will be tested. The aim is to find out whether this will be sufficient to resolve the problems with the lateral actuation, as well as finding out whether it does not cause additional problems (overheating, for example) with the motor, which has not been resized. Finally, the main aim of these tests is to achieve sufficient reliability for the Lausanne-Evian route. The control system has to be sufficiently reliable to be able to load the boat with ballast, at least an additional $50[kg]$, and ensure that no incidents occur, never.

2.6.7 Test on week 9 - Implementation

I won't go into the details of the implementations as these have already been covered in other sections. For the antiwindup implementation, I refer the reader to section 2.3.4, for the foil calibration to section 2.5.2 and for the restructuring of the state machine and FCU code to section 2.2.2..

To solve the lateral actuation problems, we decided to change the motor controller, to one that could deliver more current and allow the motor to deliver more torque. We therefore replaced the *Maxon EPOS4 Compact 50/5 CAN*[15] controller, which delivers a nominal current of $5[A]$, with the *Maxon EPOS4 Compact 50/8 CAN*[16] controller, which delivers a nominal current of $8[A]$, meaning that we increased the maximum actuation force by 60%. The *hydro* team took care of the mechanical fitting and I took care of programming the controller. However, this solution involves a risk, as the motor is rated for a current of $7.25[A]$. After discussions with the technical team at *Maxon*, this shouldn't pose any problems, but it is worth mentioning that there is a non-zero risk of breaking the motor through overheating.

The control algorithm has not undergone any major changes, apart from the addition of the antiwindup, the suppression of the integrator bounds and the suppression of $coeff1$ and $coeff2$. The complete algorithm is shown in Figure 17.

2.6.8 Test on week 9 - Result analysis

The results specific to the different implementations will not be discussed here, instead I refer the reader to the analyses made in the specific sections. Section 2.2.3 for FCU code restructuring, section 2.3.5 for antiwindup, section 2.5.3 for foil calibration.

This section focuses more on analysing the boat's performance against the April 2022 benchmark. Generally speaking, these tests went very well and all the objectives were achieved. The boat was on the water for two days and 5 runs were made on the lake. A lot of tuning work was carried out, in particular to adjust the gain of the integrators. After the first day, the boat was able to fly in a perfectly stable way, with a very smooth but also slower take-off. There were no lateral crashes and we had achieved the desired level of performance. The second day of testing was dedicated to testing and adjusting the controllers with the extra weight and changing the centre of mass. At the end of the day we were flying perfectly stable and had a controlled take-off with $72.5[kg]$ worth of sandbags spread out on the deck of the boat and a pilot about ten kilos heavier than usual [16]. We carried out manoeuvres and short turns in the waves of the support ship in order to disturb the Dahu and it remained very stable. We made long runs (over a minute) to obtain data on consumption at different heights ($[30, 35, 40, 45, 50][cm]$). Finally, we made two tribord turns, i.e. in the unfavourable axis of the boat, steered to the maximum, passing twice through the waves of the support ship, with no signs of instability. We will consider this last test as the definitive validation that the boat is sufficiently reliable and robust for the Lausanne-Evian crossing. Objectives : - 1. maintain the control of the Dahu this semester and prepare for the lake tests - 2 Ensuring reliable control with an additional 50kg load are therefore met.

The table 13 compares the evolution of controller values. The most obvious changes are the general decrease in the aggressiveness of the proportional gains (K_p, K_{ff}), and the increase in the integral gain (K_I), following the implementation of the antiwindup. Note also that the rear foil has been mechanically reset to its correct actuation range. The physical angles are certainly not very different, but here we can clearly see the impact of the calibration, 2-3 degrees being a much more realistic value for the boat's flight.

Figure 18 shows the angles of the foils during the flight phases in different scenarios. First of all, in the first case (figure 18a), we can see that the actuation hardly works at all. This can be explained by the fact that the lake was very calm, but also, I hope, by the improvement in the control algorithm and its tuning. Visually, the inspection of the worst case (figure 18d) is still very acceptable.

[16] For comparison with the benchmark, we estimate an extra $80[kg]$

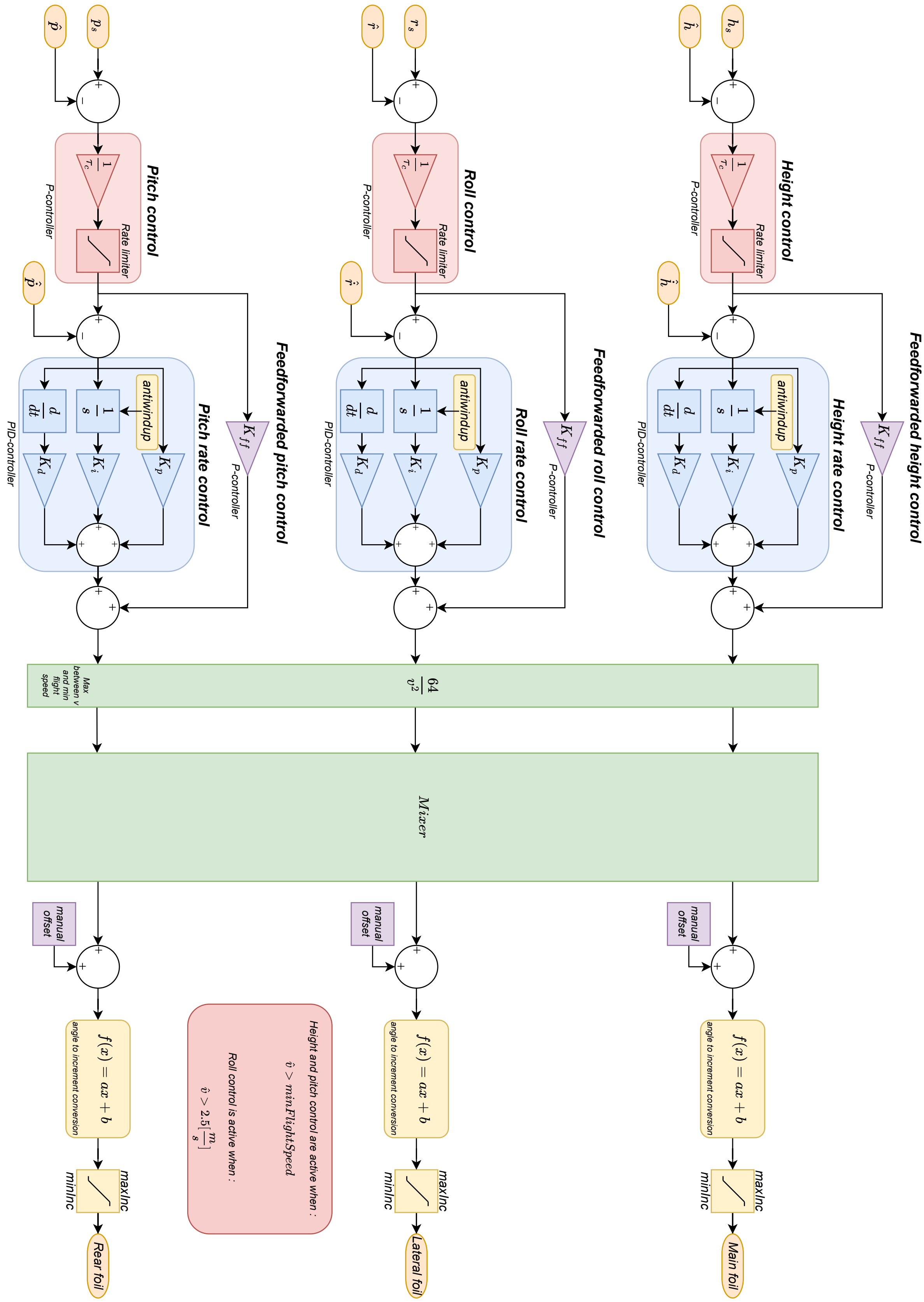
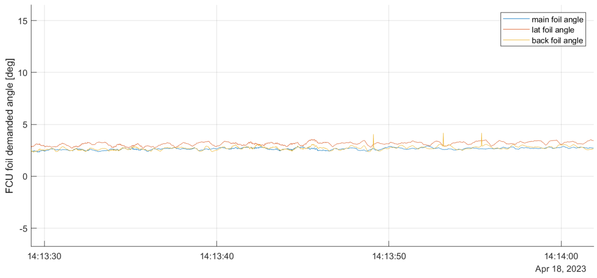


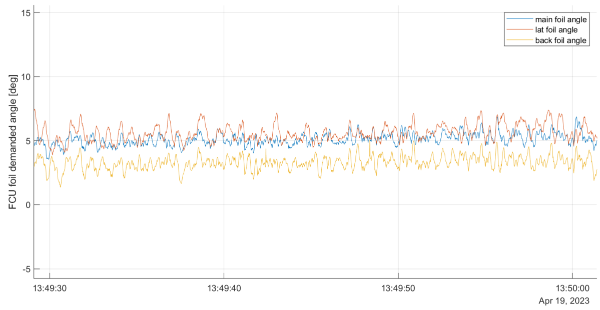
Figure 17: Full Control scheme - Implemented

		Monaco 2022	29 April 2023	Variation
Foil offset	Main foil	2°	2°	
	Lat. foil	0.5°	0°	-0.5°
	Rear foil	-4.5°	2.5°	+7°
Height controller	T_c	2	2	
	K_{ff}	40	42	
	K_P	0.8	0.25	-68.8%
	K_I	0.04	7	+17400.0%
	K_D	0	0	
Roll controller	T_c	0.75	0.75	
	K_{ff}	3.2	1.7	-46.9%
	K_P	0.35	0.2	-42.9%
	K_I	0.08	0.45	+462.5%
	K_D	0	0	
Pith controller	T_c	4	3	-25%
	K_{ff}	1	1	
	K_P	0.02	0.1	+400%
	K_I	0.02	0.1	+400%
	K_D	0	0	

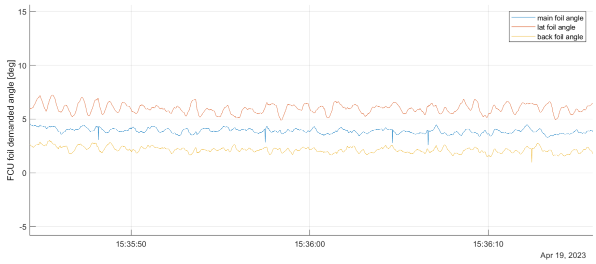
Table 13: Controller value comparison - Monaco 2022 and April 19 2023



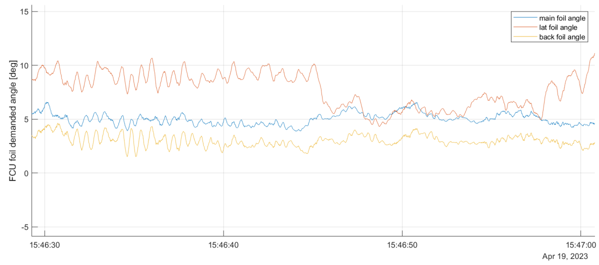
(a) Stable flight - no load



(b) Stable flight - 50 [kg] additional mass



(c) Stable flight - 80 [kg] additional mass



(d) Stable flight with steep turn and perturbation - 80 [kg] additional mass

Figure 18: FCU foil angle in stable flight - in various case - April 2023

The table 14 summarises the interest values and provides a comparison with the April 2022 benchmark. These two test sessions are relatively comparable. The lake was very calm in both cases. Note, however, that during the April 2022 tests, the Dahu used the lateral foil *V3* whereas the *V1* was used this time. What's more, the lateral actuation motor controller has also changed. These differences can perhaps explain a small part of the variations, but certainly not everything. We can almost systematically see a reduction of almost an order of magnitude in the standard deviation of the foil angle. In my opinion, this is the most relevant metric for measuring the use *intensity* of the actuator. The general reduction in the aggressiveness of the controllers explains this difference in my opinion. The integral action made it possible to reduce the proportional action and therefore to stop constantly *over-reacting*, particularly to sensor noise for example.

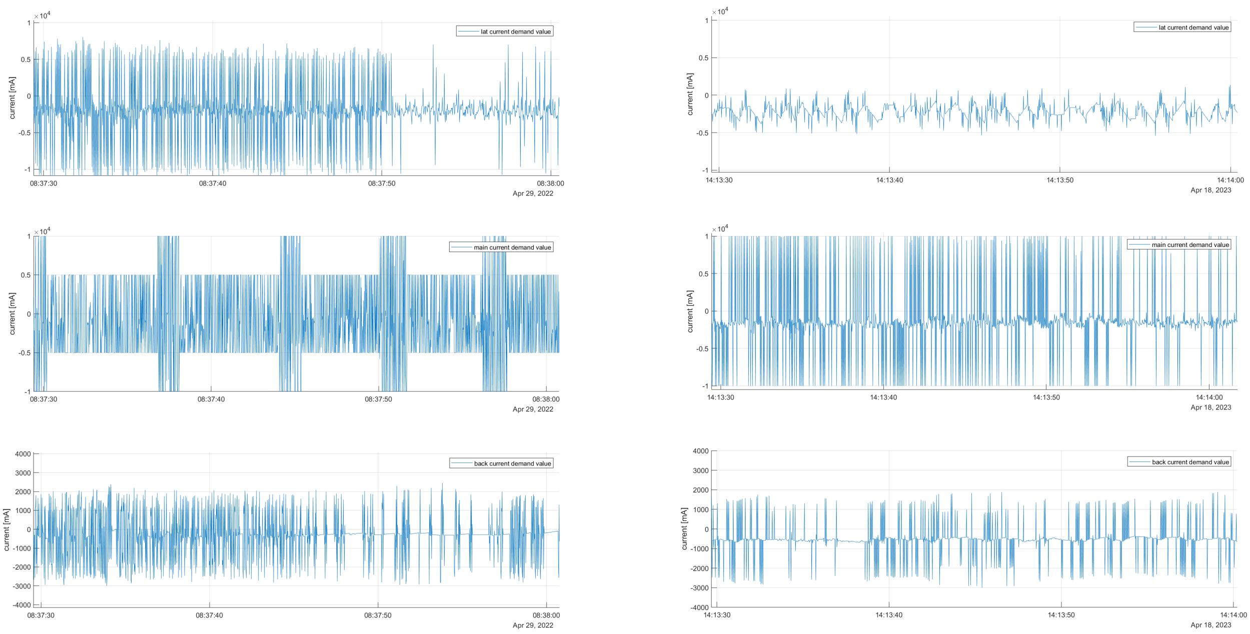
We've also seen a general drop in the consumption of actuation, but these results should be taken with reservations as they're based on a lot of assumptions.

		Foil Angle [deg]		Motor position [incr]		Motor consumption [W]	
		no load	+80[kg]	no load	+80[kg]	no load	+80[kg]
Main Foil	Mean	2.64	4.96	846	660	144	187
	Std. dev.	0.102	0.248	7.76	18.6	143	138
Comparison with Benchmark[17]		-93.2%	-83.7%	-90.6[18]%	-77.4%	-16.8%	+8.09%
Lateral Foil	Mean	3.13	5.15	-48.8	-65.0	98.9	112
	Std. dev.	0.174	0.379	1.68	3.00	38.4	49.9
Comparison with Benchmark		-81.4%	-59.5%	-77.0%	-59.0%	-42.1%	-34.5%
Rear Foil	Mean	2.71	3.22	-49.9	-57.6	31.3	37.9
	Std. dev.	0.168	0.24	1.67	3.39	21.8	27.9
Comparison with Benchmark		-87.4%	-82.0%	-88.3%	-76.3%	-3.69%	+16.6%

Table 14: Value of interest - April 18-19 2023 - Stable flight

Lateral actuation

The problems with the lateral actuation seem to have been resolved once and for all. It is clear that the actuator is working less and that its current demand has been significantly reduced. In addition, figure 19b shows that it never reaches saturation, or even approaches a peak value. As several modifications were made simultaneously, it's difficult to draw any precise conclusions, but it seems that increasing the available actuation force was not necessary, and that simply providing *better* control was sufficient.



(a) Benchmark

(b) 18 April 2023 - no load

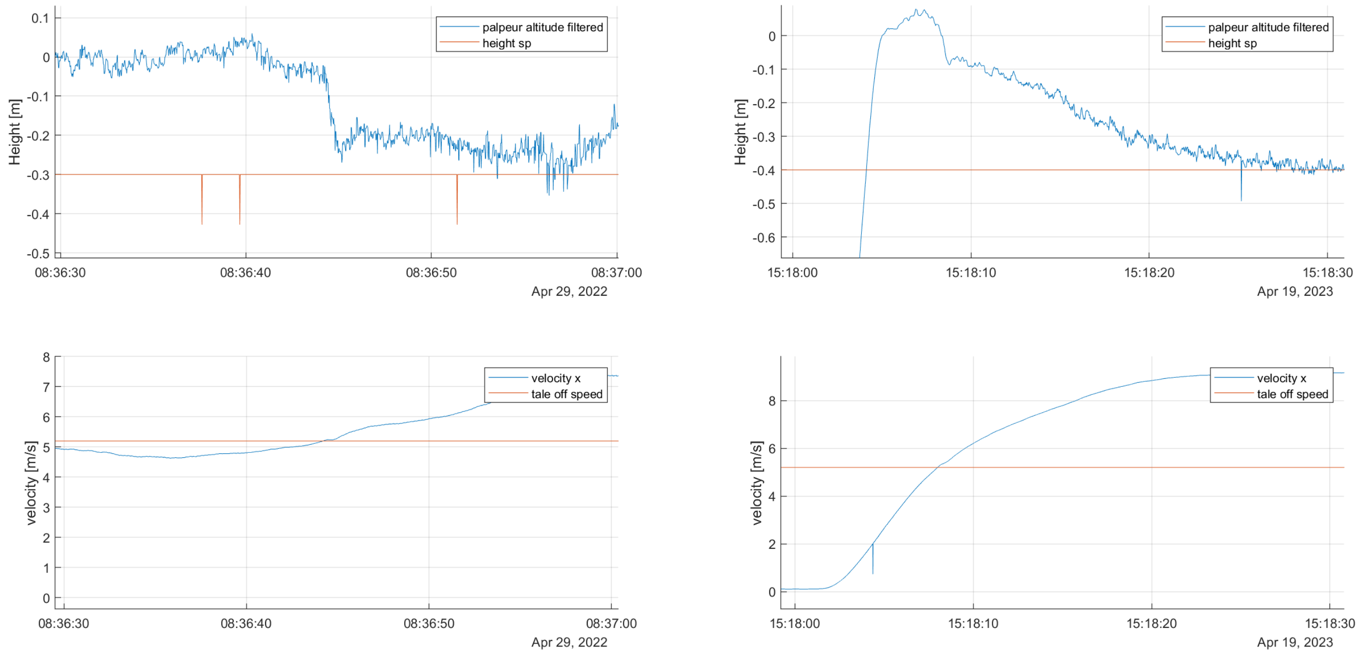
Figure 19: Actuators current consumption in stable flight and calm lake

[17] For the foil angle and motor position, it is the standard deviations that are compared, for the motor consumption, the average values are compared

[18] In an ideal world, the variation in percentage should be the same between the foil angle and the motor increment comparison. The differences are explained with the actuators calibration that changed the ratio $DegToInc$ and also because the foil angle is a reference, while the motor position is the actual position. There is a controller in between that tries to match both.

Take-off behaviour

The most obvious difference in the boat's behaviour is when it takes off. We saw that the $coeff1$ parameter caused saturation of the actuators and other problems, but it also allowed the boat to take off very quickly. We can see in figure 20a that the boat used to take off in the very second it reaches the take-off speed, threshold which starts the height controller. However, when we removed this coefficient the take-off became a lot smoother (figure 20b). The integrator gently increase the angle of attack of the foils, allowing a slow but controlled take-off, despite the additional load of $80[kg]$. It would have been possible to have a more aggressive take-off, but this has often caused problems, especially with the extra load, and let's remember that we're no longer preparing the boat for a race, but for a journey. This strategy of a gentle take-off was therefore preferred.



(a) Benchmark

(b) 18 April 2023 - 80 [kg] load

Figure 20: Take-off comparison in calm lake[19]

Setpoints tracking

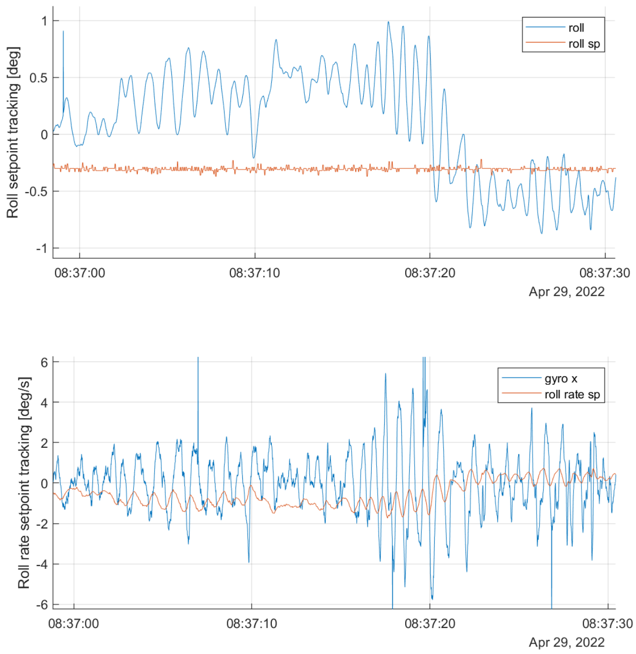
To finish this analysis, we have seen that the use of actuation has been improved, but what about the tracking of the references, which is still the main aim of the control. A visual inspection of figure 21 seems to confirm that tracking has indeed been improved. In order to make a quantitative assessment, we chose to use the root mean square error as a metric to compare the quality of the tracking. This evaluation is shown in table 15. It can be seen that the tracking has been improved by an order of magnitude.

	Height RMSE	Height rate RMSE	Roll RMSE	Roll rate RMSE	Pitch RMSE	Pitch rate RMSE
Benchmark	0.135	0.251	0.662	1.85	0.964	0.680
April 2023 - no load	0.0150	0.0471	0.143	0.665	0.0826	0.196
Variation	-88.9%	-81.2%	-78.4%	-64.1%	-91.4%	-71.2%

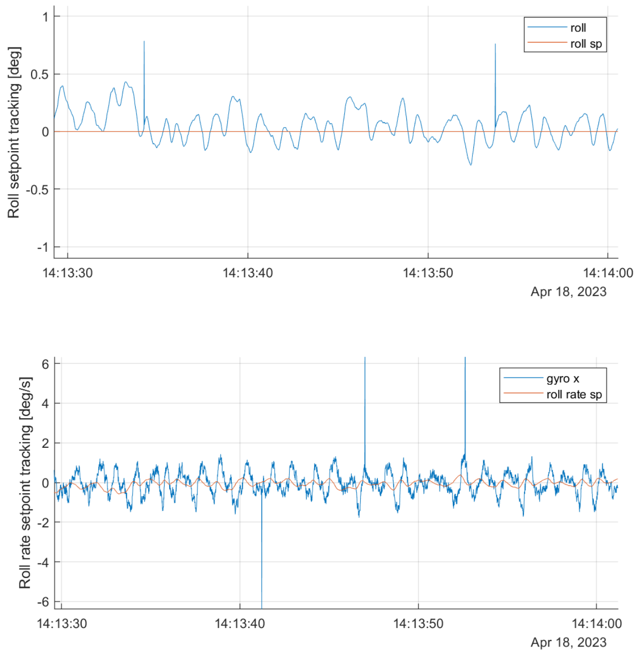
Table 15: Root mean square error[20] of the tracked variables

[19] Please note that z-axis point downwards, thus it's inverted compared to our intuition. A height of $-0.4[m]$ means that the boat flies above the water. Moreover, the palpeur is drown when the boat is not moving, thus giving false height measurement.

[20] Linear interpolation have been used to reconstruct signal of same length and make RMSE computation possible. Cubic spline interpolation gives similar results



(a) Benchmark



(b) 18 April 2023 - no load

Figure 21: Roll and roll rate tracking in stable flight and calm lake

Validity of the analysis and problem of cherry-picking the data

The results obtained should be nuanced and challenged, especially when they seem too good to be true. First of all, each of the comparisons made compares 30 seconds of stable flight only. These are systematically the same 30 seconds, those displayed in the graphs. The other navigation phases are ignored. Secondly, these are straight-line flight phases with a calm lake (unless otherwise indicated). The data has been chosen to be as comparable as possible, but obviously a perfect match is not possible. The Benchmark data has been cherry-picked (as it serves as a benchmark), which should limit the problems of cherry-picking results. I've tried to use data that is as representative as possible, but unfortunately I can't guarantee that the comparison is perfectly fair. Power consumption data has been filtered to remove artefacts, all other data has been compared as is.

2.6.9 Test on week 9 - Conclusion

Many of the elements discussed here have already had their own conclusions. In short, these tests went as well as possible and fulfilled all their objectives. At this stage the boat is ready for the Lausanne-Evian crossing, which is a real relief. However, two questions remain: what is the luck factor in these results and will we be able to reproduce them? Furthermore, having achieved good results increases the risk associated with new implementations. Every time we make a difference, we risk losing quality, and that's something we might want to avoid...

Figure 22: Sharp turn of the Dahu with 80[kg] of additional load - April 2023

3 Control REF

3.1 Position control

3.1.1 Motivation

Until now, Swiss Solar Boat has always used a cascade control architecture to fly the Dahu, and as far as I know, this goes back as far as hydrocontest, the association that gave birth to SSB. This architecture comes from the world of drones [1] and in particular from the opensource project *px4* [17]. The problem of controlling a drone is relatively similar to that of our boat. It is a multivariable problem, where the actuators' control action is a force and the objective is to control the attitude of an vessel. However, it is a complex and opaque architecture. It is difficult to identify which parameter influences which behaviour. In addition, the large number of parameters makes tuning difficult. The motivation to test a simpler position control architecture came from a very simple observation: the parameter with the greatest control action to date is the gain $\frac{K_{ff}}{T_c}$. As explained previously (figure 6), it acts as a position controller in parallel with the cascade controller. So, if our position controller is already the main controller, why not directly test a position control architecture. This would have the advantage of being much simpler and less opaque. This would reduce the number of variables and make tuning easier.

3.1.2 Implementation

The algorithm is relatively simple and the number of modifications is minimal. The base remains the same, except that the cascade controller has been replaced by a position controller. So there is still the *mixer* matrix, the scaling by $\frac{64}{v^2}$ to control in force and be invariant to the change of speed. The complete algorithm is shown in figure 23.

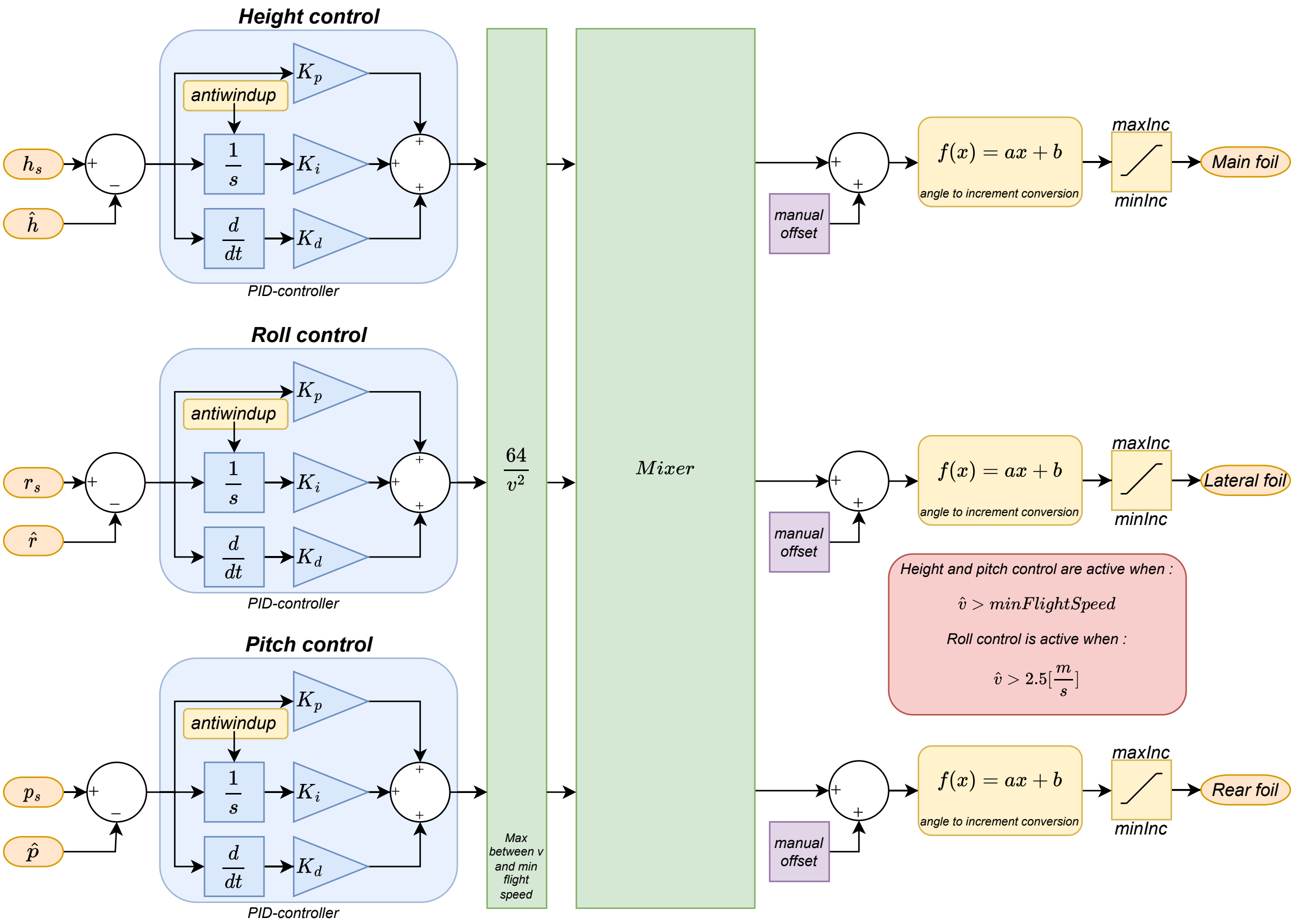


Figure 23: Position control scheme - tested in week 11

This controller was implemented during weeks 10 and 11. The modifications were made directly in the FCU's *pid* class. These particularly busy weeks and the difficulties encountered in reassembling the boat's electronics meant that we didn't have enough time with the boat fully operational to properly test the code modifications on land.

3.1.3 Lake tests - week 11

The tests in week 11 were carried out over a single day, and the boat was not stored in the port of Vidy. It was an intense logistical operation, but the boat was able to take to the water in good time. These tests had several objectives, in addition to testing the position control. Firstly, to ensure that we could replicate the results obtained in week 9. Secondly, to train a team on the control so that it could be maintained in the following semester. A lot of time and effort has been invested in passing on as much knowledge as possible about how the control works. Obviously on how the algorithm works, but also on the boat's state machine, how to reprogram the boat and all the tips and tricks needed to make it work.

Figure 24: Overshoot with the feeler rod fully outside water

The first test on the water was to see if the boat took off the first time and flew stably with the Week 9 controller. This proved to be the case, so we were able to move on to testing the controller in position. All the tuning work had to be redone, but we could still use certain values as a starting point, such as $\frac{K_{ff}}{T_c}$ being the proportional gain of our previous position controller. There were few failures and frightful moments before we arrived at a decent setting. Notably, an overshoot at take-off that saw the entire feeler lifted out of the water, causing the boat to crash with significant slamming on the side float. Apart from this incident, the tests went well and produced some interesting results.

3.1.4 Result analysis

Figure 25: Formation of the next control team

Unfortunately, the test logs could not be extracted from the boat that evening due to lack of time, and it has not been switched back on since (a condition that is almost essential for extraction). This section will therefore have to be restricted to a qualitative analysis based on the many photos and videos taken during the tests. Firstly, the roll was relatively stable and was able to compensate for the steady state error when a roll up or roll down reference was given. Secondly, after a considerable period of tuning, the boat was able to take off in a stable and controlled manner. However, the boat systematically pitched up or down after take-off, a very curious behaviour. In other words, the boat takes off slowly and horizontally. And the more time passes, the more it tilts in either direction. The reason for this behaviour is that the pitch control was not active, whereas the roll and height control were correctly active. This is probably due to an error in the implementation of the FCU code, perhaps linked to the way in which the *pitch* instance of the *pid* class is initialised. A surprising problem given the very few changes made. Nevertheless, it wasn't a problem that could be fixed on the lake. After a few unsuccessful attempts, we went back to the week 9 controller and continued training the team. We also noticed that when the boat pitches up, the alitude measurement is becoming increasingly incorrect. The boat thinks it's lower than it actually is. Part of the height overshoot is therefore due to an erroneous measurement rather than a poor controller setting. The height is measured at the front of the boat using a feeler. The sensor is in contact with the water and an angle encoder gives the required angle with the XY plane of the boat. After inspection, the transfer of this measurement to the centre of mass as well as taking into account the pitch angle is not correctly done. Correcting this problem will be a job for the future.

3.1.5 Conclusion

Although there have been some glitches, I think that the position control is promising and could allow the boat to be operated in a reliable and robust way. I don't think it would improve performance, but it would at least make the control team's job easier. Nevertheless, there's still some work to be done before we reach the same level as with our regular algorithm. It is part of the continuity of this semester project to want to simplify things in order to control them better. The actuation system will change with the new boat, and I don't have enough information or experience to recommend position control for the next boat, but I think I can say that it's an option to consider. Finally, a lot of time and effort has been invested in training a control team for the upcoming semester. I have every confidence in them and I think it's safe to say that the objective is to: *4. Train one person to maintain control next semester* has been achieved.

3.2 Height measurement - Radar

3.2.1 Context & motivation

Measuring altitude is a challenge for Swiss Solar Boat, but also for flying boats in general. Many different sensors are being tested, but the research has not yet converged and for the moment there is no standard. Nor are there any (or very few) sensors dedicated to this application. At SSB, altitude measurement is currently provided by a feeler at the front of the boat. This is a simple mechanical arm that maintains contact with the water. An angular encoder uses a simple trigonometric relationship to determine the altitude. This sensor is very simple, robust and reliable. On the other hand, it has a tendency to bounce off the water at high speed, causing considerable noise. Additionally, it's also mechanically fragile and we occasionally break its carbon rod. For all these reasons, SSB is experimenting with new altitude measurement methods. Several semester projects have already been dedicated to this, and the latest sensor to be tested is a radar. The company *Baumer* kindly offered us one of their models and we have a good chance of thinking that it will be suitable for our application because the company, although oriented towards agricultural machinery sensors, also equips other hydrofoil boats. Nevertheless, their sensor uses a particular CAN protocol, CAN J1939, and two semester projects struggle with it without ever managing to implement it on the boat. I'd like to thank Valentine Houlier in particular, who had almost every possible problem with this sensor and made it possible to implement it this semester.

Figure 26: Baumer radar - R600V.DAE0

3.2.2 Implementation

The first problem with implementing this radar is being able to correctly receive the messages it sends. We had already managed to receive data frames thanks to the *peak-CAN*, which is a CAN data-logger that can be controlled via a PC, but we had never succeeded with one of our microcontrollers that can be embedded on the boat. The source of the problem has not been properly identified and in my opinion it is multifactorial, but we had given up with the MKRzero microcontrollers. The first clue to a solution came thanks to John Taylor, whom I'd like to thank warmly, who wrote a package *Arduino* [21] for the V2.2 microncontroller that we developed at SSB[18] last spring. This package solves the problem of using both CAN lines simultaneously, as well as using a different CAN library to the one we were using until now. This allowed us to unblock the situation and finally receive the CAN messages from the radar correctly.

A second problem then arose, that of correctly decoding the frame it was sending us. It turned out that the datasheet that *Baumer* had sent us last autumn was not the right one. Indeed, we had asked for a first model and as it wasn't available, the company sent us another relatively equivalent one and that's how the confusion arose. We were decoding a frame for the R600V.DAH5 radar when we had the R600V.DAE0. The radars are identical and the datasheets almost similar, except for the main frame which is completely different. The frame to be decoded is shown in figure 27. In addition, a factory reset was performed on the radar in an attempt to understand what was happening. The necessary code provided by *Baumer* can be found in appendix C.

0x 8C AB 27 0C 34 0C 00 00 [1000 1100 1010 1011 0010 0111 0000 1100 0011 0100 0000 1100 0000 0000 0000 0000]₂					
Sensor status (bit 1-2)	Canopy confidence (bit 3-9)	Ground confidence (bit 10-16)	Canopy distance (bit 17-32)	Ground distance (bit 33-48)	Reserved (bit 49-64)
$[00]_2$	$[1100011]_2$	$[1010101]_2$	0x 0C 27	0x 0C 34	0x 00 00
0	99%	85%	3'111mm	3'124mm	N/A

Note: Little endian format, bits 1 underlined; colours indicate correspondences

Figure 27: Exemplary decoding the *ground & canopy* message (PGN 0xC000)[19, p. 23]

Everything was now ready for the first implementation on the boat. The best place for this was in the feeler box. At that time, there was still an *Arduino* MKRzero that made the feeler measurement, but it was relatively feasible to replace it with the MCU V2.2 in order to take advantage of its two CAN lines. The first dedicated to communication with the rest of the boat, and the second for the sensor interface. The schematic diagram is shown in Figure 28. This involved porting the MKRzero sensor code to our new MCU V2.2 as well as writing code to interface with the radar. The code was designed to be as modular as possible, with one library written for each sensor. The code has been fully commented and is available in the gitlab repository *altitude-measurement-baumer*. It was then rigorously tested and debugged in preparation for the Week 9 tests.

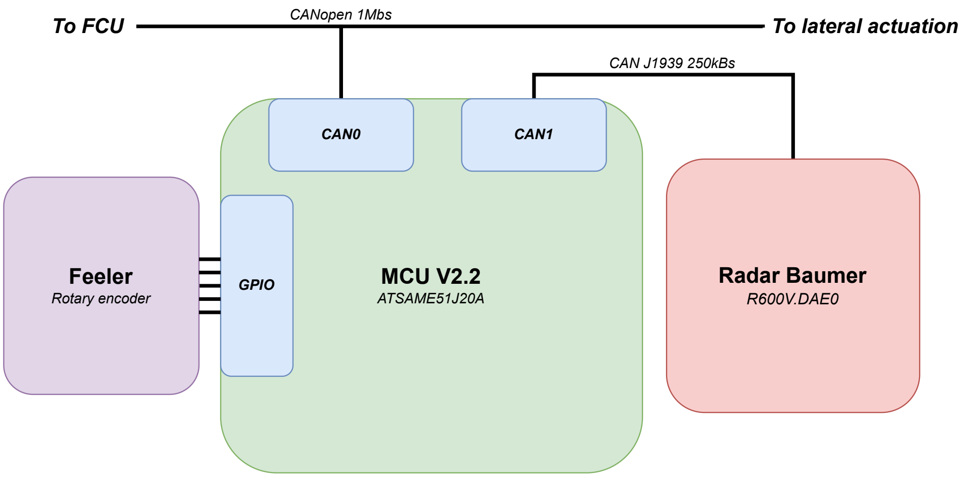


Figure 28: High level schematic of the feeler box with the Radar Baumer implementation

In addition, I'd like to thank Simon Dorthe, who took care of the mechanical fitting of the radar to

[21] It's a hardware abstraction layer that lets you program a board and a microcontroller with the *Arduino* environment.

the feeler box, as well as the production of a new iteration of it, and without whom the implementation of this radar would not have been possible in time.

3.2.3 Result analysis

The radar was mounted on the boat for testing in week 9 and we were able to collect some data, a sample of which is shown in figure 29. First of all, we can see that the radar is indeed capable of measuring altitude, which is good news. However, there are a few problems. First of all, the measurement scale is completely wrong... The most likely problem is a conversion error (in the matlab code or in the MCU V2.2 code) or a frame decoding error. The second problem is that the measurement is not valid when the boat is not flying, even though the radar is out of the water ($\approx 25[cm]$). It doesn't seem to be sensitive that close. Although the datasheet indicates a sensitivity range from 200 to 6000 $[mm]$. In order to solve this problem, moving the sensor is one option, but it would also be necessary to ensure that the radar is correctly configured, as the sensitivity range is adjustable. In addition, the measurement is much noisier than that of the feeler. This can be explained by poor mechanical mounting of the radar and poor setting of the radar sensitivity (also a configurable parameter). Further tests and in-pool characterisation would be necessary to answer this question. There are also two other factors that may explain these results. On the one hand, we can't rule out the possibility that this radar simply isn't suited to our application and that we can't do any better. On the other hand, here it's the canopy measurement that is displayed and not the ground measurement, which would certainly be better suited to our application... The ground measurement is completely erroneous, so there are no usable results. The most likely explanation is a poor decoding of the frame or a conversion error somewhere. It could also be that the radar configuration is not right for our application. Once again, only more extensive testing and proper characterisation could answer this question.

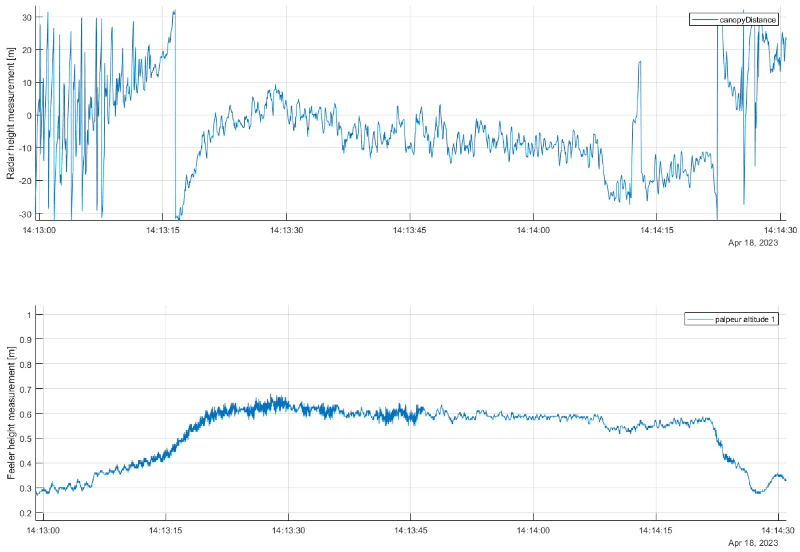


Figure 29: Radar and feeler height measurement - stable flight - 18 April 2023 [22]

3.2.4 Conclusion

To conclude this project, which has been running for three semesters, I would say that this sensor is a serious possibility and that we should continue to explore it. There's almost nothing missing to make it usable. It has proved its mechanical and electrical robustness, the only thing lacking is its configuration and characterisation. Unfortunately, this hasn't been a priority this semester and I haven't had any more time to devote to it. Nevertheless, the hardware and software foundations have been solidly laid and the characterisation work should be much easier in the future. I hope to be able to devote some more time to it before the Lausanne-Evian route.

3.3 Recommendation and future directions for the REF

3.3.1 Yaw measurement for roll compensation

Roll control is the most crucial element in the boat's performance, at least from a racing perspective. In fact, it determines how aggressive you can be with the curves and how short you can take them. With our latest control algorithm, I think we've reached the limits of what's possible. In fact, the control of the roll is very stable, both in a straight line and when turning. Nonetheless, the boat is leaning a little at the start of the turn, rolling up on tribord turns and rolling down on babord turns. This error is gently compensated for by the integrator, which brings the boat back to a flat position during the turn. Obviously, once the boat straightens out, the opposite behaviour is observed, until the integrator returns to normal. Increasing the proportional gain can limit this behaviour, but it leads to more instability in normal conditions, and is therefore undesirable. To improve this behaviour, adding a derivative gain would be an option, but that would require filtering the data in my opinion. It's for these reasons that I think we've reached what's possible with the roll and our current tools. In my opinion, the best option is to tackle the source of the problem: the boat's change of direction. We know the source of the disturbance in the roll, so it would be possible to compensate for its effect. However, this requires an additional measure. We already know the yaw of the boat thanks to the IMU, which also integrates GPS data. However, this measurement suffers from a bit of a delay and when the yaw changes, the boat has already reacted and the roll has therefore already been disturbed. The solution would therefore be to measure directly what is disturbing the boat, i.e. the rudder angle. This requires an additional sensor. Last semester, an angle encoder was installed to measure the angle of the pilot's wheel, which is equivalent to the rudder angle. This solution was implemented once in December 2022, however the implementation was not fully complete and could not be reiterated. I had to rewrite some code to make it work and it was ready for testing in week 11. In addition, I had modified the FCU to take it into account in the control algorithm (as in figure 30). However, because of the problems encountered with position control, this could not be tested. In my opinion, the control algorithm could be modified in two ways in order to take this measure into account and improve roll control:

- Feedforward the disturbance caused by the variation in rudder angle to keep the boat flat.
- Modify the roll setpoint to angle the boat during a turn.

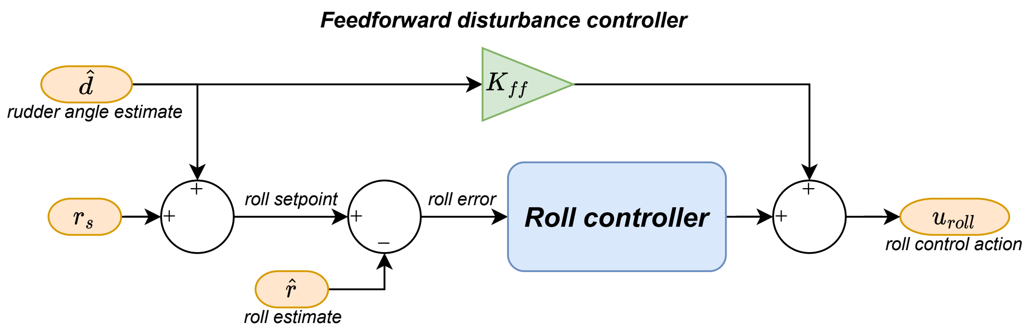


Figure 30: Control scheme for roll compensation of rudder perturbation

It is in this direction that I think there is the greatest potential for improving the performance of the Dahu, and I believe that this thought is also perfectly valid for the REF. There's almost nothing missing for these solutions to be implemented on the Dahu and without the risk involved by the REX implementation, they would already have been tested.

3.3.2 Hardware & Software

On the hardware side, it was decided, in conjunction with Mathias Arnold and Kai Ween Loo who are in charge of design projects for the REF's low-power electronics this semester, to integrate the REF's FCU directly into the on-board computer. This has a number of advantages. Firstly, many of the messages between the various units will be internal to the on-board computer, limiting external communication, which would mean there would only be one CAN bus on the boat and would greatly simplify data management and the boat's electronic robustness. Secondly, the on-board computer is a much more powerful unit than what we currently have, which would make it possible to use more demanding control or data analysis algorithms, such as MPC. The on-board computer would use a standard OS, probably a version of Ubuntu. These advantages also come with challenges. Three functionalities, currently separated in three distinct physical systems, would be grouped together. The OS would make it possible to run programs in parallel, but this would involve a lot of work in managing shared data. Additionally, bugs will be harder to identify and fix because it will be harder to identify their source.

In terms of software, I think we need to build a new FCU from scratch, because the one we have at this stage, is too complex, with 6 or even 7 levels of abstraction. These levels of abstraction don't always make sense and the quest for modularity has gone too far in my opinion. Porting the current code to a new platform will require too much investment for a far from optimal result, in my opinion. However, the boat's state machine [1, pp. 55–58] makes a lot of sense to me and the numerous security conditions it implies are good practice for a boat built by students. I think it should be kept.

3.3.3 Safety reflexes and soft crash

At time of writing, when a behaviour triggers a safety error, we don't have any strategies and simply shut down the system. For example, an excessive roll angle cuts the propulsion and puts the three foils in neutral position, without any further consideration of the state of the boat. Or when a battery cell reaches an undervoltage threshold, the power supply to the whole boat is cut immediately, without any consideration for the state of the boat, or the appropriate cut-off sequence. I'd very much like to see the notion of *soft-crash or soft-error* implemented, where when a first threshold is reached, a safety procedure is triggered to stop the process cleanly. The idea of this first threshold is to take the time to do things properly. There would also be a second threshold, which this time would trigger a *hard-error or hard-crashed* that would shut everything down without consideration.

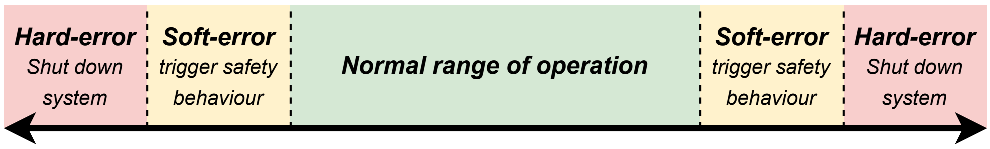


Figure 31: Soft-error scheme

A first implementation of these softs-errors will take place very soon with the Hydrogen safety systems of the REX project. The *hard* limits have been implemented physically in such a way to shut down the system, no matter what, while the *soft* limits are managed in software with a shutdown sequence to be respected. I think this is a more intelligent way of dealing with errors. Of course, some of them require an immediate reaction to protect people's safety, but for most of them, intelligent error management would improve people's safety as well as preserving hardware integrity. For the REF, I think it's important to manage these errors intelligently in order to trigger a behaviour that puts the boat down calmly and gradually cuts power. Indeed, as speeds increase and energy management becomes more important, I think that our brute-force strategy will no longer be sufficient. It's necessary to anticipate these behaviours from the design stage, in order to coordinate all the systems efficiently during the design of their software.

Similarly, I was interested in planning a safety reflex framework to force a certain foil configuration in order to recover the boat's attitude when it took on problematic values. For example, forcing the side foil to downforce when the roll reached dangerous values in order to avoid a possible capsized situation. For risk management reasons, this was not tested this semester. It could create problems elsewhere, for example with cases of slamming for which the Dahu was not initially designed. I still think it's an interesting solution that's worth investigating for the REF.

3.3.4 Estimator & Optimal control

The next area where control can make great progress is in estimating the state of the boat. Today, simple low-pass filters are implemented. However, it is difficult to do better in the absence of a reliable model of the boat. A project launched this semester should solve this problem in the future, and open up new possibilities. At the time of writing, there is still too much unknown about the actuation system to give a pertinent opinion on any control algorithm. However, if there is one lesson to be learned from this semester's project, it is the importance of the integrator. Optimal control strategies such as LQR or MPC are *model-based* strategies, so they suffer from the error between the model and reality. An integrator compensates for these modelling errors and is often absent from the theoretical formulations discussed in lectures. I would therefore like to emphasise this last point because I am convinced that it greatly improves the boat's performance.

4 Conclusion

This semester's project has been a bit of a patchwork. I had the opportunity to explore many different directions, sometimes by choice, sometimes because of the association's needs. I hope I was able to make the narrative as interesting as possible and also to lay solid foundations for the future and raise relevant points for the next boat. The risks associated with the project were high, which also limited the freedom I had to make changes to the algorithm, bearing in mind that the FCU is really the central unit of the boat and the slightest bug could compromise the whole route. I think I've achieved each of the project's objectives, at least in part. Control has been maintained and the boat has always been operational on time. It is as ready as it can be for the Lausanne-Evian route, although we won't know if that's enough until D-Day. A healthy foundation for the future has been laid, the successors have been trained and the control diagrams are as detailed as possible and, for once, complete and up to date. However, there's still a lot to do and it promises to be interesting. I would have liked to have done more, especially with regard to more ambitious control strategies, to have been able to test an optimal control, but I haven't managed to do that despite the 5 to 6 days a week I've devoted to this project. Instead, I did a lot of questioning of practices, coefficients, ways of doing this or that calculation, explanations given for this or that phenomenon. Although fundamentally I haven't changed much about the way things are done, I do think that it's much more accurate today. I'd like to conclude by saying what a pleasure it was to do this project: technically very interesting, varied, full of responsibilities but also in a great atmosphere, well supervised and well surrounded by the members of the association. It will remain a great memory.

Acknowledgment

Finally, there are so many people I'd like to thank for this project, first and foremost the entire Swiss Solar Boat community, who have made this adventure first and foremost a human one. I'd also like to thank Mathias Arnold in particular, who worked with me to lead the electronics team and with whom I often found myself finishing work on Sunday evenings so that everything was ready on Monday morning. Stefano Nebuloni for his discreet but relevant and much appreciated supervision. Simon Dorthe for his constant commitment and for the passion he brings to his work. Jules Bervillé and David Croce for their boundless commitment to the electronics team. Noé Tambourin, with whom I've been able to share ideas on control and who has put a lot of himself into the association and the project. Mark Dei-Kofi and Max Molina for their investment in the lateral actuation problems. Mattia Valitutti for his invaluable help during reassembly, day and night. Robin Amacher for always finding a solution and arranging things for us. Valentine Houlier and John Taylor for all their effort and perseverance with the radar problem. Juliette Pelissier who welcomed me and allowed me to join the association. Yoann Lapijover, Maxime Zuffrey and Yann Boudigou, despite all the bad things I said about their diagram, for their excellent report and for all the colossal and fundamental work done beforehand. André Hodder for his invaluable help with electronic security. Christope Salzmann for his help with the antiwindup when I couldn't find anyone to compare my ideas with. Benoît Prudhomme-Lacroix for, among other things, his excellent organisation of the test logistics. Judith Mayencourt, my mother, who financed an extra semester of study so that I could play with a flying boat. Théophile Dufour and Aliénor Hamoir for taking the risk of piloting the boat when I was experimenting with position control.

Appendix A Regression Analysis

A.1 Matlab script for data preprocessing

```
%% Exctract data for reg analysis

%% Create array data X = (t, vx, az, h, aoa_m, aoa_l, aoa_r) with same sample time

time_tol = 0.01; % 10 [ms]
min_height = 0.5;
max_height = 1;
min_speed = 6; % [m/s]
max_speed = 12; % [m/s]

% data to match
vx = data.ahrs_sensor.velocity_x;
az = data.ahrs_sensor.d_vel_z;
h = data.palpeur.palpeur_altitude_1;
aoa_m = data.main_position_motor.main_position_actual_value;
aoa_l = data.lat_position_motor.lat_position_actual_value;
aoa_r = data.back_position_motor.back_position_actual_value;

% index from data to match
index_matched_data = 1;
index_az = 1;
index_h = 1;
index_aoa_m = 1;
index_aoa_l = 1;
index_aoa_r = 1;

% Bool if match data has been found
found_matching_az = false;
found_matching_h = false;
found_mathing_aoa_m = false;
found_mathing_aoa_l = false;
found_mathing_aoa_r = false;

% preallocating for speed
matched_data = zeros(length(vx),7);

for index_vx = 1:length(vx)
    % Match az to vx
    while(true)
        if abs( az(index_az,2) - vx(index_vx, 2) ) < time_tol/2
            % same time found
            found_matching_az = true;
            break;
        elseif az(index_az,2) > vx(index_vx, 2)
            % time_az is greater than time_vz
            found_matching_az = false;
            break;
        elseif index_az >= length(az)
            % we've finnished the table
            found_matching_az = false;
            break;
        else
            % increment the time_az
            index_az = index_az + 1;
        end
```

```
    end

    % Match h to vx
    while(true)
        if abs( h(index_h,2) - vx(index_vx, 2) ) < time_tol/2
            % same time found
            found_matching_h = true;
            break;
        elseif h(index_h,2) > vx(index_vx, 2)
            % time_az is greater than time_vz
            found_matching_h = false;
            break;
        elseif index_h >= length(h)
            % we've finnished the table
            found_matching_h = false;
            break;
        else
            % increment the time_az
            index_h = index_h + 1;
        end
    end

    % Match aoa_m to vx
    while(true)
        if abs( aoa_m(index_aoa_m,2) - vx(index_vx, 2) ) < time_tol/2
            % same time found
            found_matching_aoa_m = true;
            break;
        elseif index_aoa_m >= length(aoa_m)
            % we've finnished the table
            found_matching_aoa_m = false;
            break;
        elseif aoa_m(index_aoa_m,2) > vx(index_vx, 2)
            % time_az is greater than time_vz
            found_matching_aoa_m = false;
            break;
        else
            % increment the time_az
            index_aoa_m = index_aoa_m + 1;
        end
    end

        % Match aoa_l to vx
    while(true)
        if abs( aoa_l(index_aoa_l,2) - vx(index_vx, 2) ) < time_tol/2
            % same time found
            found_matching_aoa_l = true;
            break;
        elseif aoa_l(index_aoa_l,2) > vx(index_vx, 2)
            % time_az is greater than time_vz
            found_matching_aoa_l = false;
            break;
        elseif index_aoa_l >= length(aoa_l)
            % we've finnished the table
            found_matching_aoa_l = false;
```

```
            break;
        else
            % increment the time_az
            index_aoa_l = index_aoa_l + 1;
        end
    end

        % Match aoa_r to vx
    while(true)
        if abs( aoa_r(index_aoa_r,2) - vx(index_vx, 2) ) < time_tol/2
            % same time found
            found_matching_aoa_r = true;
            break;
        elseif aoa_r(index_aoa_r,2) > vx(index_vx, 2)
            % time_az is greater than time_vz
            found_matching_aoa_r = false;
            break;
        elseif index_aoa_r >= length(aoa_r)
            % we've finnished the table
            found_matching_aoa_r = false;
            break;
        else
            % increment the time_az
            index_aoa_r = index_aoa_r + 1;
        end
    end

    % save data point
    if found_matching_az && found_matching_h && found_matching_aoa_m &&↙
found_matching_aoa_l && found_matching_aoa_r
        matched_data(index_matched_data,:) = [vx(index_vx, 2), vx(index_vx, 1), az↙
(index_az,1), h(index_h,1), aoa_m(index_aoa_m, 1), aoa_l(index_aoa_l, 1), aoa_r↙
(index_aoa_r, 1)];
        index_matched_data = index_matched_data + 1;
    end

    found_matching_az = false;
    found_matching_h = false;
    found_matching_aoa_m = false;
    found_matching_aoa_l = false;
    found_matching_aoa_r = false;
end

matched_data = matched_data(matched_data(:,1) > 0,:); % get rid of zeros
matched_data = matched_data(matched_data(:, 2) > min_speed,:); % get rid of datapoints↙
below min speed
matched_data = matched_data(matched_data(:, 2) < max_speed,:); % get rid of datapoints↙
above max speed
matched_data = matched_data(matched_data(:, 4) > min_height,:); % get rid of↙
datapoints below min height
matched_data = matched_data(matched_data(:, 4) < max_height,:); % get rid of↙
datapoints above max height

disp(['Found ',num2str(length(matched_data)),' datapoints with a time tolerance of ',↙
num2str(1000*time_tol),' [ms].']);
```

```
%% plots for insight
figure(1)
scatter(matched_data(:,2),matched_data(:,5))
xlabel('speed v_x [m/s]')
ylabel('Main foil angle of attack [Incr]');

figure(2)
scatter(matched_data(:,3),matched_data(:,5))
xlabel(' vertical acceleration a_z [m/s]')
ylabel('Main foil angle of attack [incr]');


%% save data to CSV
writematrix(matched_data,'data/extract_CSV_reg_analysis/2023-04-19_14_12_13-reg-↙
analysis.csv')
```

A.2 Python script for regression analysis

In [ ]:
```
import pandas as pd
import statsmodels.formula.api as smf
import seaborn as sns
import numpy as np
```

In [ ]:
```
#load data
data = pd.read_csv('regAnalysis.csv',
                   names = ['time', 'vx', 'az', 'h', 'aoa_m','aoa_l','aoa_r'])
data
```

Out[ ]:

	time	vx	az	h	aoa_m	aoa_l	aoa_r
0	1.681822e+09	7.11	-9.64	0.541	817	-50	-21
1	1.681822e+09	7.09	-9.81	0.782	819	-46	-30
2	1.681822e+09	6.66	-9.03	0.843	855	-40	-24
3	1.681822e+09	7.51	-9.45	0.795	857	-40	-27
4	1.681823e+09	7.38	-10.17	0.673	843	-49	-33
...	...	...	...	...	...	...	...
607	1.681825e+09	7.56	-9.75	0.565	736	-57	-51
608	1.681825e+09	7.56	-10.04	0.565	735	-58	-51
609	1.681825e+09	7.55	-10.04	0.588	741	-57	-51
610	1.681825e+09	7.56	-9.68	0.588	744	-58	-51
611	1.681825e+09	7.56	-9.82	0.588	746	-58	-51

612 rows × 7 columns

In [ ]:
```
# conversion increment to angle (To modify if they change !)
a = -81.5
b = 1054
data.aoa_m = data.aoa_m.apply(lambda x : (x - b)/a)

a = -8.05
b = -24
data.aoa_l = data.aoa_l.apply(lambda x : (x - b)/a)

a = -14.9
b = -10
data.aoa_r = data.aoa_r.apply(lambda x : (x - b)/a)
```

In [ ]:
```
# add the vx square columns
data['vx2'] = data['vx']*data['vx']
data
```

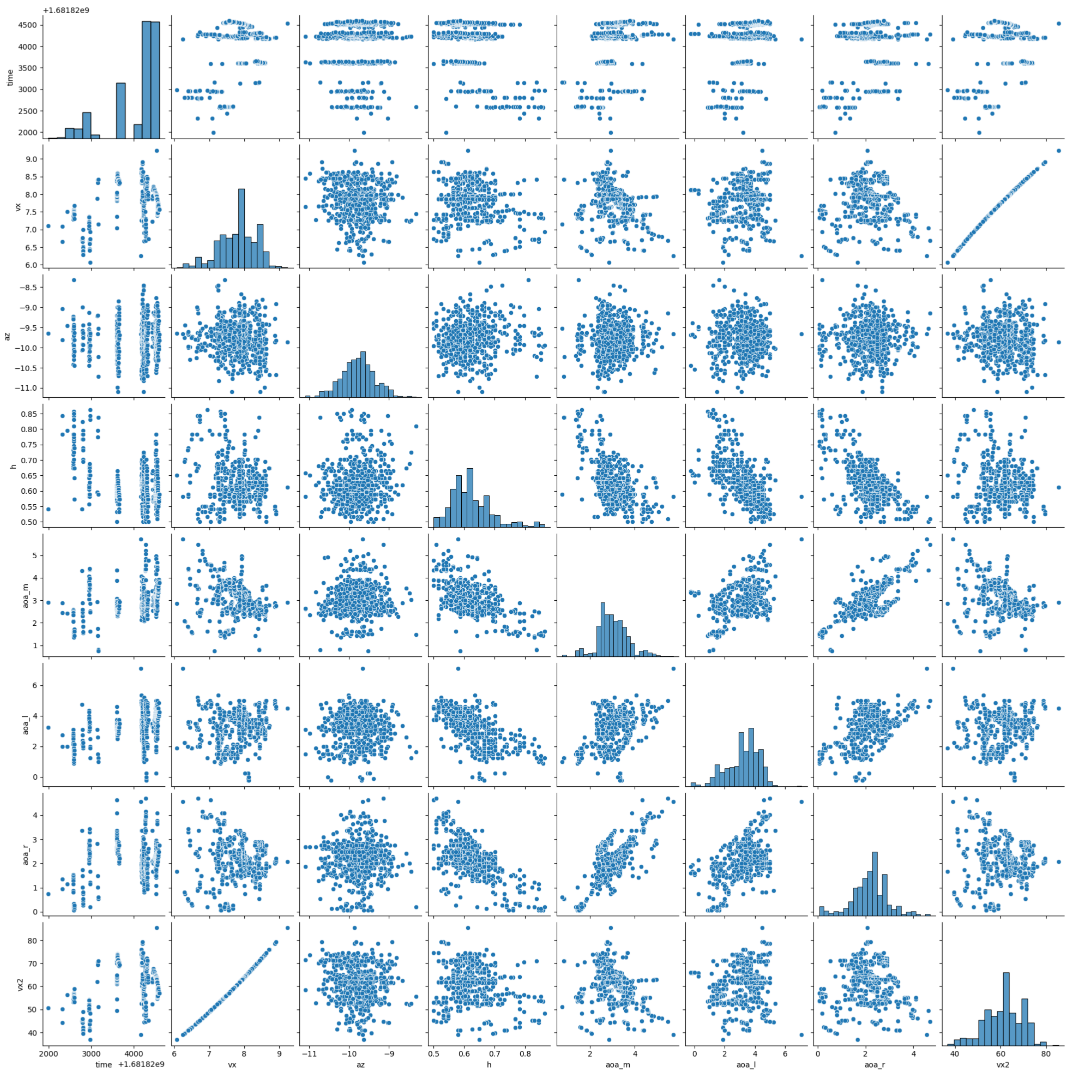
Out[ ]:

	time	vx	az	h	aoa_m	aoa_l	aoa_r	vx2
0	1.681822e+09	7.11	-9.64	0.541	2.907975	3.229814	0.738255	50.5521
1	1.681822e+09	7.09	-9.81	0.782	2.883436	2.732919	1.342282	50.2681
2	1.681822e+09	6.66	-9.03	0.843	2.441718	1.987578	0.939597	44.3556
3	1.681822e+09	7.51	-9.45	0.795	2.417178	1.987578	1.140940	56.4001
4	1.681823e+09	7.38	-10.17	0.673	2.588957	3.105590	1.543624	54.4644
...	...	...	...	...	...	...	...	...
607	1.681825e+09	7.56	-9.75	0.565	3.901840	4.099379	2.751678	57.1536
608	1.681825e+09	7.56	-10.04	0.565	3.914110	4.223602	2.751678	57.1536
609	1.681825e+09	7.55	-10.04	0.588	3.840491	4.099379	2.751678	57.0025
610	1.681825e+09	7.56	-9.68	0.588	3.803681	4.223602	2.751678	57.1536
611	1.681825e+09	7.56	-9.82	0.588	3.779141	4.223602	2.751678	57.1536

612 rows × 8 columns

In [ ]:

```
# Plot and inspect data
g = sns.pairplot(data)
```



In [ ]:
```python
# Declares the model : Linear regression
Lin_reg_model = smf.ols(formula='az ~ vx2:aoa_m + vx2:aoa_l + vx2:aoa_r',
                        data=data)

# Fits the model (find the optimal coefficients,
# adding a random seed ensures consistency)
np.random.seed(2)
result = Lin_reg_model.fit()

# Print thes summary output provided by the library.
print(result.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                     az   R-squared:                       0.007
Model:                            OLS   Adj. R-squared:                  0.002
Method:                 Least Squares   F-statistic:                     1.351
Date:                Mon, 05 Jun 2023   Prob (F-statistic):              0.257
Time:                        10:53:40   Log-Likelihood:                -339.00
No. Observations:                 612   AIC:                             686.0
Df Residuals:                     608   BIC:                             703.7
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -9.8116      0.086   -114.009      0.000      -9.981      -9.643
vx2:aoa_m      0.0010      0.001      1.559      0.120      -0.000       0.002
vx2:aoa_l     -0.0002      0.000     -0.688      0.492      -0.001       0.000
vx2:aoa_r     -0.0009      0.001     -1.551      0.121      -0.002       0.000
==============================================================================
Omnibus:                        2.249   Durbin-Watson:                   1.645
Prob(Omnibus):                  0.325   Jarque-Bera (JB):                2.279
Skew:                           0.035   Prob(JB):                        0.320
Kurtosis:                       3.291   Cond. No.                     1.57e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly s
pecified.
[2] The condition number is large, 1.57e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

Appendix B Foil calibration procedure

Foil Calibration Procedure

Préparation

1. S'assurer d'avoir toute l'actuation montée. Une calibration ne se fait qu'à la toute fin, lorsque plus aucun changement n'est fait au montage mécanique des foils !
2. S'assurer d'avoir un bateau fonctionnel, avec le high power allumé et une GS connectée.
3. S'assurer que les moteurs *Maxon* soient électroniquement bien connectés et fonctionnels.
4. S'assurer d'avoir les trois cales pour mesurer l'angle de chacun des foils.
5. S'assurer d'avoir le niveau électronique, avec de la batterie.
6. S'assurer d'avoir un bateau plat. La mesure se fait avec les angles donnés par l'IMU. Le pitch peut se corriger grâce à la manivelle à l'avant du bateau qui contrôle la hauteur de la roue. Le roll se corrige en modifiant la hauteur de la fixation du soutien du flotteur sur la remorque. Il faut être au minimum quatre pour faire cette opération.
7. **Attention ! Avant de commencer à reporter les valeurs, il faut avoir fait un homing des foils !** Lorsque le moteur démarre, il a un zéro, qu'il réinitialise lorsqu'un homing est fait (le zéro est au niveau du switch).

Variables d'intérêts

Nous sommes intéressés par les incréments minimum et maximum que peuvent prendre les foils. D'un côté, ils viennent taper le switch, de l'autre côté, ils rentrent en butée mécanique. De plus, nous sommes également intéressés par le rapport degré/incrément (attention au signe !) ainsi que par l'ordonnée à l'origine, c'est-à-dire le nombre d'incrément lorsque l'angle du foil est nul.

MaxInc	MinInc	a	b
Nombre maximal d'incrément admissible	Nombre minimal d'incrément admissible	rapport degré/incrément	incrément lorsque l'angle du foil est nul

Attention ! Merci de prendre une marge de sécurité pour MaxInc et MinInc, typiquement 10%.

Main Foil

1. Allumer le bateau.
2. Se connecter à la GS.

3. Allumer le high power.
4. Mettre la FCU en calibration.
5. **Faire un homing du main foil** (! être attentif, monitorer les valeurs du contrôleur à travers la GS et désarmer le moteur s'il touche une butée (valeur de courant élevé)!).
6. Monter la calle sur le foil et mettre le niveau sur la calle.
7. Faire tourner le moteur à la main jusqu'à ce qu'il ne touche plus le switch et que la marge de sécurité soit confortable. Relever la première limite (min ou max inc). Il faut relever la valeur en incrément du moteur, ainsi que la valeur en angle du foil (attention au signe, déportance = signe négatif). **La valeur en incrément peut être lue à travers l'onglet Maxon de la GS.**
8. Faire tourner le moteur à la main jusqu'à ce que la valeur d'angle lue par le niveau soit nulle (=0). Relever la valeur en incrément (il s'agit de l'ordonnée à l'origine).
9. Faire tourner le moteur à la main jusqu'à la limite suivante, qui est une limite mécanique (butée). Prendre une marge de sécurité confortable et mesurer la valeur en incrément et la valeur en angle (min ou max inc).

Lateral Foil

1. Allumer le bateau
2. Se connecter à la GS
3. Allumer le high power
4. Mettre la FCU en calibration
5. **Faire un homing du lateral foil** (être attentif, surveiller les valeurs du contrôleur à travers la GS et désarmer le moteur s'il touche une butée (valeur de courant élevé) !)
6. Monter la calle sur le foil et mettre le niveau sur la calle
7. Faire bouger le foil à la main (ou le moteur si le capot est enlevé), jusqu'à ce qu'il ne touche plus le switch et que la marge de sécurité soit confortable. Il est important que le capot soit retiré ou d'être très attentif au "click" du switch. Relever la première limite (min ou max inc). Il faut relever la valeur en incrément du moteur, ainsi que la valeur en angle du foil (attention au signe, déportance = signe négatif). **La valeur en incrément peut être lue à travers l'onglet Maxon de la GS.**
8. Faire bouger le foil à la main (ou le moteur si le capot est enlevé) jusqu'à ce que la valeur d'angle lue par le niveau soit nulle (=0). Relever la valeur en incrément (il s'agit de l'ordonnée à l'origine).
9. Faire bouger le foil à la main (ou le moteur si le capot est enlevé) jusqu'à la limite suivante, qui est une limite mécanique (butée). Prendre une marge de sécurité confortable et mesurer la valeur en incrément et la valeur en angle (min ou max inc).

Rear Foil

1. Allumer le bateau.
2. Se connecter à la GS.
3. Allumer le high power.
4. Mettre la FCU en calibration.
5. **Faire un homing du rear foil** (! être attentif, surveiller les valeurs du contrôleur à travers la GS et désarmer le moteur s’il touche une butée (valeur de courant élevé)!).
6. Monter la calle sur le foil et mettre le niveau sur la calle.
7. Afin de faire bouger le foil, il faut le saisir délicatement (on peut s’aider de la calle) et effectuer un moment sur le foil. Il s’agit d’une vis à bille, elle est donc réversible mais cela demande un peu de force. Il faut tout de même être précautionneux, il s’agit de pièces fragiles.
8. Faire bouger le foil à la main jusqu’à ce qu’il ne touche plus le switch et que la marge de sécurité soit confortable. Il est important que le couvercle du CUS soit enlevé afin de voir le switch. Relever la première limite (min ou max inc). Il faut relever la valeur en incrément du moteur, ainsi que la valeur en angle du foil (attention au signe, déportance = signe négatif). **La valeur en incrément peut être lue à travers l’onglet Maxon de la GS.**
9. Faire bouger le foil à la main jusqu’à ce que la valeur d’angle lue par le niveau soit nulle (=0). Relever la valeur en incrément (il s’agit de l’ordonnée à l’origine).
10. Faire bouger le foil à la main jusqu’à la limite suivante, qui est une limite mécanique (butée). Prendre une marge de sécurité confortable et mesurer la valeur en incrément et la valeur en angle (min ou max inc).

Mettre à jour la FCU avec les valeurs de calibration

Une fois les 9 paires de valeurs obtenues (Min, Max et b fois 3 en angle et en incrément), il est possible de calculer les trois coefficients a en faisant le rapport

$$(MaxInc - MinInc)/(MaxDeg - MinDeg)$$

Faites attention au signe, il compte !

À ce stade, vous avez obtenu toutes les valeurs. Il est maintenant possible de les mettre dans la FCU, dans la classe *boat*, au moment de l'initialisation, tout en haut du fichier *boat.cpp*.

Il faut ensuite reprogrammer la FCU avec les nouvelles valeurs. Pour cela, il existe d'autres procédures.

Se connecter et contrôler un moteur

Avec le logiciel de chez *Maxon*, *Epos Studio*, il est possible de se connecter à un moteur, de lire les valeurs en incrément, de faire une calibration et même d'imposer une valeur en incrément. Cela n'est généralement pas nécessaire, mais cela peut faciliter la procédure de calibration, par exemple avec le rear foil qui est compliqué à manipuler.

Pour cela, il faut d'une part posséder (et ouvrir) le logiciel *Epos Studio* et ouvrir un nouveau projet de type *Epos4*. Une fois que c'est fait, il est possible de se connecter au contrôleur *Maxon* avec un câble microUSB. Notez que le contrôleur doit être sous tension, c'est-à-dire que High Power doit être allumé.

Il s'agit d'une procédure risquée, dans le sens où les contrôleurs Maxon sont fragiles et que le moindre faux contact/court-circuit leur est fatal. Veillez à être très précautionneux et à protéger toutes les surfaces conductrices avec du scotch approprié.

Une fois connecté, le logiciel est très facile à prendre en main. Il suffit d'ouvrir le panneau de contrôle, de se mettre en mode HMM (c'est-à-dire en mode Homing. Il existe également le mode PPM : position control mode et VVM : velocity control mode) et d'effectuer les commandes que l'on souhaite. Les commandes sont identiques à celles de la GS et il est en plus possible d'imposer une valeur en incrément au moteur.

Appendix C Radar Baumer - factory reset code

```python
import can
import time
# Releases:
# RevA:
#  - Initial relase
# Rev B:
#  - Fix datalength bug with newer radar firmware
# Rev C:
#  - Include sensor information to the skript


# Note: Only one sensor on the canbus supported, is sensor_address is None

#
# Set the new value
#
value = 1 #Parameter value

#
# Set the SPN number
#
# spn_number = 517000 # 0x0007E388 Detect Range end, mm,  default 6000
# spn_number = 517001 # 0x0007E389 Detect Range Start, mm, default 200
# spn_number = 517030 # 0x0007E3A6 Sensitivity, %, default 100
# spn_number = 517036 # 0x0007E3AC Detection Mode 0=ground and Crop
                      # Mode 1=single target, resets with pwr cycle
# spn_number = 516010 # Green LED Duty Cycle %, resets with pwr cycle
spn_number = 517018 # 0x0007E39A CAN Bus speed; 0 = 250k, 1 = 500k

#
# Show can log
#
doCanLog = True

#
# Print Sensor information
#
doSensorInfo = True

#
# Sensor Address
#
sensor_address = None

#
# Select the desired canbus interface
#
# bus = can.interface.Bus(bustype='socketcan', channel='vcan0', bitrate=250000)
bus = can.interface.Bus(bustype='pcan', channel='PCAN_USBBUS1', bitrate=250000)
# bus = can.interface.Bus(bustype='ixxat', channel=0, bitrate=250000)
# bus = can.interface.Bus(bustype='vector', app_name='CANalyzer', channel=0, bitrate=250000)

# ----------------------------------------------------------------------------------------------------
# Do not change after here
# ----------------------------------------------------------------------------------------------------

# Simple log function
```

```python
def _recv_msg(msg):
    if doCanLog : print(msg)

# Create a buffered CAN reader
can_rcv_buffer = can.BufferedReader()
# Register the can Notifier
notifier = can.Notifier(bus, [_recv_msg, can_rcv_buffer])

# Prepare the SPN value
spn_bdata = spn_number.to_bytes( 4, 'little')

# Send a message on the bus
def send_msg(msg):
    try:
        if doCanLog: print(msg)
        bus.send(msg)
        return True
    except can.CanError:
        return False

# Recv a message on the bus
def recv_msg(expected_pgn, timeout=1.25):
    tstart = time.time()
    while ( time.time() - tstart ) < timeout:
        # Receive messages
        msg = can_rcv_buffer.get_message()
        if msg:
            pgn = ( msg.arbitration_id & 0xFF0000 ) >> 8
            if pgn == expected_pgn:
                return msg
    return None

def get_sensor_address():
    if sensor_address:
        return sensor_address
    else:
        msg = can.Message(
            arbitration_id=0x18EAFFFE,
            data=[0x00, 0xEE, 0x00],
            is_extended_id=True)

        if not send_msg(msg):
            print("Couldn't send the request for address claim")

        msg = recv_msg(0xEE00)
        if msg == None:
            print("Didn't get the response of the request")
            return
        # Extract source address of the sensor and store it as sensor address
        return msg.arbitration_id & 0xFF

def print_sensor_info():
    msg = can.Message(
        arbitration_id=0x18EAFFF9,
        data=[0xC5, 0xFD, 0x00],
        is_extended_id=True)

    if not send_msg(msg):
        print("Couldn't send PGN 0xFEDA message")
```

```
    msg = recv_msg(0xEC00, timeout=0.5)
    data = msg.data
    bytes_len = data[1]
    pkt_len = data[3]

    res = bytearray(bytes_len)
    for n in range(pkt_len):
        msg = recv_msg(0xEB00, timeout=0.1)
        if msg is None:
            break
        data = msg.data
        res[n*7:(n+1)*7] = data[1:min(8, bytes_len-n*7)]

    print(f'ECU Identification Info: {res.decode()}')

    msg = can.Message(
        arbitration_id=0x18EAFFF9,
        data=[0xDA, 0xFE, 0x00],
        is_extended_id=True)

    if not send_msg(msg):
        print("Couldn't send PGN 0xFEDA message")

    msg = recv_msg(0xEC00, timeout=0.5)
    data = msg.data
    bytes_len = data[1]
    pkt_len = data[3]

    res = bytearray(bytes_len)
    for n in range(pkt_len):
        msg = recv_msg(0xEB00, timeout=0.1)
        if msg is None:
            break
        data = msg.data
        res[n*7:(n+1)*7] = data[1:min(8, bytes_len-n*7)]

    print(f'ECU Software Identification: {res[1::].decode()}')

def set_parameter(dist):
    dst = get_sensor_address()
    print('Sensor address: {}'.format(dst))

    msg = can.Message(
        arbitration_id=0x18EEFFF9,
        data=[0xE8, 0x03, 0x00, 0x00, 0x00, 0x81, 0x00, 0x00],
        is_extended_id=True)

    if not send_msg(msg):
        print("Couldn't send address claim")

    msg = can.Message(
        arbitration_id=0x18D900F9 | (dst << 8),
        data=[0x01, 0x15] + list(spn_bdata) + [0x01, 0x00],
        is_extended_id=True)

    if not send_msg(msg):
        print("Couldn't send the first DM14 message")

    msg = recv_msg(0xD800)
    if msg == None:
```

```
            print("Didn't get the response of the first DM14 message")
            return
        # Copy the data from the message
        data = msg.data
        bdata = data[-2::]
        val = int.from_bytes(bdata, byteorder='little')
        if val != 0xFFFF:
            # Login Required
            data = [0x01, 0x15]  + list(spn_bdata)
            data.append(val & 0x00FF)
            data.append((val & 0xFF00) >> 8)
            msg = can.Message(
                arbitration_id=0x18D900F9 | (dst << 8),
                data=data,
                is_extended_id=True)

            if not send_msg(msg):
                print("Couldn't send the second DM14 message")

            msg = recv_msg(0xD800)
            if msg == None:
                print("Didn't get the response of the second DM14 message")
                return
            # Copy the data from the message
            data = msg.data
            bdata = data[-2::]
            val = int.from_bytes(bdata, byteorder='little')
            if val != 0xFFFF:
                print("Seed/Key Login not successful")
                return
        print("Login successful")

        # Convert value to byte list
        val = int(dist)
        bdata = val.to_bytes(max(1,(val.bit_length() + 7) // 8), 'little')
        #if len(bdata) > data[0]:
        #    print(f'dist: {dist}, val: {val}') # delete
        #    print(f'data: {data}, bdata: {bdata}') # delete
        #    print(f'length of bdata: {len(bdata)}, Data: {data[0]}') # delete
        #    print('Data size too big, sensor supports {} but {} where
    requested'.format(data[0], len(bdata)))
        #    return
        data = [len(bdata), 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF]
        for n in range(len(bdata)):
            data[n+1] = int(bdata[n])

        msg = can.Message(
            arbitration_id=0x18D700F9 | (dst << 8),
            data=data,
            is_extended_id=True)

        if not send_msg(msg):
            print("Couldn't send the DM16 message")

        msg = recv_msg(0xD800)
        if msg == None:
            print("Didn't get the response of the DM16 message")
            return
        # Copy the data from the message
        data = msg.data
```

```python

    if data[1] != 0x09:
        print("Transfer was not successfuly")
        return

    msg = can.Message(
        arbitration_id=0x18D800F9 | (dst << 8),
        data=[0x04, 0x19]  + list(spn_bdata) + [0xFF, 0xFF],
        is_extended_id=True)

    send_msg(msg)
    print('Successfuly update the parameter')

if __name__ == "__main__":
    if doSensorInfo: print_sensor_info()
    set_parameter(value)

```

References

[1] Ahmad Abou Daher, Joseph Nasr, and Yoann Lapijover. *Design and implementation of an altitude estimator and a controlalgorithm for a foiling Proa*. Lausanne, 2021.

[2] Yann Boudigou and Maxime Zufferey. *Digital Twins and control strategy*. Lausanne, 2022.

[3] Karl Johan Åström and Tore Hägglund. *Advanced PID Control*. English. ISA - The Instrumentation, Systems and Automation Society, 2006. ISBN: 978-1-55617-942-6.

[4] Prof. Alberto Bemporad. *Lecture notes Automatic Control 2 - Anti-windup techniques*. University of Trento, 2011.

[5] Dew Toochinda. "PID Anti-Windup Schemes". In: *Scilab.Ninja* ().

[6] Sophie Tarbouriech et al. *Stability and Stabilization of Linear Systems With Saturating Actuators*. Jan. 2011. ISBN: 978-0-85729-940-6. DOI: 10.1007/978-0-85729-941-3_2.

[7] Sergio Galeani et al. "A Tutorial on Modern Anti-windup Design". In: *European Journal of Control* 15.3 (2009), pp. 418–440. ISSN: 0947-3580. DOI: https://doi.org/10.3166/ejc.15.418-440. URL: https://www.sciencedirect.com/science/article/pii/S0947358009709987.

[8] Robert West. *Lecture notes Applied Data Analysis (CS401) - Regression analysis*. EPFL, 2022.

[9] Wes McKinney. "Data Structures for Statistical Computing in Python". In: *Proceedings of the 9th Python in Science Conference*. Ed. by Stéfan van der Walt and Jarrod Millman. 2010, pp. 51–56.

[10] Michael L. Waskom. "seaborn: statistical data visualization". In: *Journal of Open Source Software* 6.60 (2021), p. 3021. DOI: 10.21105/joss.03021. URL: https://doi.org/10.21105/joss.03021.

[11] J. D. Hunter. "Matplotlib: A 2D graphics environment". In: *Computing in Science & Engineering* 9.3 (2007), pp. 90–95. DOI: 10.1109/MCSE.2007.55.

[12] Charles R. Harris et al. "Array programming with NumPy". In: *Nature* 585.7825 (Sept. 2020), pp. 357–362. DOI: 10.1038/s41586-020-2649-2. URL: https://doi.org/10.1038/s41586-020-2649-2.

[13] Skipper Seabold and Josef Perktold. "statsmodels: Econometric and statistical modeling with python". In: *9th Python in Science Conference*. 2010.

[14] Adrien Peltier. *Control, foil and foil actuation design for the first boat of Swiss Solar Boat*. Lausanne, 2019.

[15] *EPOS4 Module/Compact 50/5 Positioning Controller, Hardware reference*. Maxon motor AG. Sachseln, Switzerland, April 2022.

[16] *EPOS4 Module/Compact 50/8 Positioning Controller, Hardware reference*. Maxon motor AG. Sachseln, Switzerland, April 2022.

[17] px4. *PX4 Autopilot User Guide*. 2023. URL: https://docs.px4.io/main/en/.

[18] Baptiste Savioz, Blanche Brognart, and Mathias Arnold. *PCBs design and production*. Lausanne, 2022.

[19] *R600V.DAE0 Operating Manual (EN)*. Baumer Electric AG. Frauenfeld, Switzerland, October 2021.