Second Year Internship Report
Development Of Fitness Robots And Their Human-Robot Interface

FISE 2025 - SOIA

October 2, 2024

Internship at the Tokyo University of Science
Takuya Hashimoto Sensei Laboratory
1st of May 2024 - 25th of August 2024
Clément PATRIZIO
clement.patrizio@ensta-bretagne.org

Abstract

With the collaboration of Nippon Sport Science University, Professor Takuya Hashimoto and his students have designed and developed fitness robots that could replace traditional fitness machines and help professional athletes track their improvements. To control these fitness robots, we needed to create a user interface. Therefore, we developed a desktop application called CoachBot. Nine sub-versions were produced before the first fully functioning version, version 1.0. This document is the report of a four-month internship that concluded the development of the first fully operational application version.

The application needs to answer the following requirements.

— The user should be able to choose any load value to lift.
— The user should be able to set a "load profile" to make the robot change dynamically the load as the movement is done.
— The application must be able to calculate the One Repetition Maximum (One RM) and the Endurance of the athlete.
— It must also be able to save the user's workout session data in a database.

This document is divided into two parts. The first part is a state-of-the-art analysis of what exists in the market and what was developed before the internship in the laboratory. The second part relates to the contributions and the value added to the project during the internship.

Keywords

Fitness robot, One Repetition Maximum, Endurance calculation, Athlete Fatigue, C++ application, Human Machine Interface

Acknowledgements

I am incredibly grateful to Professor Takuya Hashimoto, who trusted me and welcomed me into his laboratory at the Tokyo University of Science. His support gave me the opportunity to gain valuable experience in a country that was entirely new to me. Under his advice, I enhanced my technical skills in both development and robotics. Moreover, I discovered the unique and fascinating Japanese work culture, which enriched my personal and professional growth. This experience has been exceptional, and I owe much of it to his mentorship.

I also want to thank my friends from ENSTA Bretagne Noé Parker, Antoine Morvan, Romain Bornier, Enora Danet, Antoine Madrelle and Tom Bergamelli, with whom I spent a lot of time pacing the wonderful landscapes of Japan.

I also have a particular thought for Naoki, Sousuke and Nagashima. The great people I met in Japan and who let me feel the Japanese culture from the inside. They taught me that even though Japanese people might be short, they can jump as rockets into the sky when attacking in volleyball or basketball.

Figure 1 – Tokyo University of Science. The laboratories are in the building on the left.

Contents

Glossary **6**

Introduction **7**

1 State of the Art (Market and Pre-Internship Analysis) **10**
1.1 Analysis of the client's needs and assessment of the current market trends 10
1.2 Mechanical and electronic structure of the robots 10
1.2.1 Mechanical structure 10
1.2.2 Electronic structure of the robots 11
1.3 One Repetition Maximum 12
1.4 Endurance 13

2 Contributions and Value Added to the Project **16**
2.1 Hardware modifications 16
2.2 General presentation of the software project 16
2.3 General presentation of the classes 17
2.4 Data acquirement with a Kalman filter 18
2.5 Linear and polynomial regressions 20
2.5.1 Linear regression 20
2.5.2 Polynomial regression 21
2.5.3 Comparing polynomial and linear regression 21
2.6 Peak detection 21
2.7 Software development concepts 24
2.7.1 The Analysers as observers of the `TrainerPredictor` 24
2.7.2 The plotting mechanism 25
2.8 Various features implemented in the application 26
2.8.1 The Load Profile creation 26
2.8.2 The database creation 27

Conclusion **29**

Appendix **32**

List of Figures

1 Tokyo University of Science. The laboratories are in the building on the left. 3
2 Screenshot of the application home page during a training session. CoachBot V1.1 . . . 8
3 From left to right: Antoine Morvan, a French engineering student who worked in the laboratory as an intern, Professor Hashimoto, Clément Patrizio 9
4 First robot made by students . . . 11
5 Second robot . . . 11
6 PEX-361116 from ©Interface . . . 11
7 LPC-632104 from ©Interface . . . 11
8 Explanation of the One RM protocol - author: vbtcoach (graph modified) 13
9 Example of a real data set that was used to define the athlete's Endurance law (biceps curl, set at 10kg, 15kg and 21kg) . . . 14
10 Picture of the LPC, the PEX, the CONTEC counter and the CONTEC AIO. . . . 16
11 Class diagram of the Project. . . . 18
12 Screenshot of the application after acquiring a data set using the moving average. The angular velocity (green) is choppy. . . . 19
13 Representation of the filter used. . . . 20
14 Screenshot of the application using the filter. The distance and velocity are less choppy and closer to the real data. . . . 20
15 Screenshot of the application when the `TrainerPredictor` object understands that the last repetition of the set is not high enough. The computer can also notify when the repetition is not smooth enough. . . . 23
16 Screenshot of the application using the failing peak detection method on a theoretical data set. . . . 23
17 Screenshot of the application using the failing peak detection method on a real data set. The polynomial regressions don't fit, and some peaks are not considered repetitions. . . . 24
18 Explanation of the observable/observers concept and its application in the Project. . . . 25
19 Example of five plots' data saved in the lists-type variables of `TrainerApp`. 26
20 Screenshot of the Load Profile option. Fully flexible, the user can drag and drop the points to modify it. . . . 27
21 Calendar-looking page to access previous training sessions . . . 27
22 Summary of the past training session . . . 27
23 Example of a past One RM experiment accessed via the calendar . . . 28
24 The database structure . . . 29
25 Estimated and real Gantt Diagram . . . 33
26 Expanded class diagram . . . 34
27 Datasheet of the application counting the user manual and and the technical manual (August 2024, 33 pages) . . . 35
28 Assessment report page 1/2 . . . 36
29 Assessment report page 2/2 . . . 37

Glossary

- **One RM** The One Repetition Maximum is the maximum load that an athlete can lift in one repetition.
- **One RM velocity** The One RM Velocity is the estimated velocity at which the user could lift the One RM load.
- **Endurance** of the athlete refers to a calculation of fatigue using velocity loss during a set.
- **LPC board** is a counter board used by the previous generation of robots. It requires a compatible slot on the PC motherboard.
- **PEX board** is an analog input-output board used by the previous generation of robots. It requires a compatible slot on the PC motherboard.
- **CONTEC AIO board** is an analog input-output board used by the new generation of robots. It can be directly plugged in on a USB port.
- **CONTEC CNT board** is a counterboard used by the new generation of robots. It can be directly plugged in on a USB port.
- **CONTEC boards** refers to both the CONTEC AIO board and the CONTEC CNT board.
- **Set** refers to a series of repetitions of the same exercise.

Introduction

Professor Takuya Hashimoto is a Japanese professor who leads a laboratory at the Tokyo University of Science. His laboratory's mission is to conduct studies on new topics such as the interaction between humans and robots, new technologies for medical and welfare use, the development of underwater robots that look like fish or manta rays, and the development of AI and robots in the sports field.

Around twenty to thirty students from Bachelor's to Master's level worked in the departments listed above. Some of them had close subjects and could work in teams. Every week, each team of students presented their improvement in their research to Professor Takuya Hashimoto, who was also available to answer any questions during the rest of the week. Twice a month, the students needed to present individually their work to the entire laboratory.

My subject was not close enough to another student's, so I worked alone. Despite the lack of teamwork, the internship was exciting, as I needed to fit in with the Japanese way of working. It was my first experience in Asia, and it took time, but I managed to understand how Japanese people work. I tried to integrate with the rest of the students, but the language barrier was high, and I could only make friends with a few. It was a challenging experience.

The subject I was assigned to is fascinating. The goal is to develop a robot that replaces conventional fitness machines. Instead of lifting a bar with heavy plates, the user could lift a bar linked to motors by a wire and through a pulley. The motors imitated the weight of the plates by applying an equivalent torque to the bar. The different robots were already developed when I integrated the lab. My job was to improve them by changing the interface boards that allow the computer to communicate with the robots and to create a desktop application to make a convenient user experience.

Powerlifters or bodybuilders could use such robots. Indeed, they are limited in their sport by the lack of hefty plates in fitness centres. Here, the robot's only limit is the power of the motors. Professional athletes could also use robots to provide data on the velocity of the repetitions. Then, they can trace their improvements through the training sessions. In ageing societies, robots could benefit greatly as they adapt to the athlete's raw strength and fatigue. Patients who need muscular healing could also use these robots, as they allow the user to apply smooth torques instead of one coarse force. This could prevent injuries, especially to the tendons. The uses are multiple. This is why this project is part of a program led by the collaboration between the Tokyo University of Science and the Nippon Sport Science University (based in Tokyo).

Moreover, the robots can dynamically apply a specific load. The goal is multiple.

— The user can choose any load kilogram by kilogram under the robot's maximum power.
— The load applied to the bar is effective only when the user is ready for the exercise.

This can protect the athlete's back when they move around the bar to get into position.

- Sensors are used to get data on the velocity of the bar during the exercise. This can provide professional athlete with data on their performances.
- In case of failure during a movement, the user can switch off the motors with an emergency switch. (This means the user can do a bench press without a spotter.)
- As we can change the load dynamically, we can apply a low load at the beginning of the movement and then apply the correct load when the muscles are in the best position to lift. This can prevent tendon wounds.
- Users can save the data of their workout session and see their progress.

These goals are especially interesting when the user wants to lift high loads (that can lead to dangerous situations) or when the user is a patient who needs to heal a muscular wound (the exercise needs to be as smooth as possible).

The exciting part of the project was developing an application from scratch using my C++ and software development skills. I also needed to create the application's intelligent features, such as analyzing the athlete's repetitions and sets.

Figure 2 – Screenshot of the application home page during a training session. CoachBot V1.1

Figure 3 – From left to right: Antoine Morvan, a French engineering student who worked in the laboratory as an intern, Professor Hashimoto, Clément Patrizio

1 State of the Art (Market and Pre-Internship Analysis)

1.1 Analysis of the client's needs and assessment of the current market trends

The Nippon Sport Science University and the Tokyo University of Science collaborate to develop robots to help professional athletes train in their fitness centres. Therefore, we can see the athletes and coaches from the Nippon Sport Science University as the project's clients. They need a robot with a human interface machine that could acquire data on their sets of repetitions (of any fitness exercise), save this data, and analyse it.

In general, fitness robots are not much developed and democratised. They are usually expensive, hardly scalable, and poorly targeting the needs of professional athletes.
For example, the Tonal Society [1] developed a mural robot that allows users to pull ropes from the wall. Motors inside the robot apply a specific torque to the rope. Compared to our project, Tonal's robot is the closest option on the market. However, the application linked to the robot does not provide professional athletes with all the data they require to trace their improvements.

The Nippon Sport Science University expressed the need to estimate the athlete's One-Repetition Maximum and Endurance. Students from Professor Takuya Hashimoto's laboratory have conducted research studies to understand these notions in the sports field and tried to elaborate protocols to estimate them. Please refer to the section "One Repetition Maximum" and "Endurance" to know more about these notions. We will discuss these notions, but first, let's discuss robots' mechanical and electronic structure.

1.2 Mechanical and electronic structure of the robots

1.2.1 Mechanical structure

The robots were made of two motors (left and right), two pulleys, two ropes, and one bar. For the athlete's convenience, the bar is a fitness bar, and there is also a bench for bench pressing or inclined bench pressing exercises.

There are two different robots.

The first one is called a "first-generation robot" and was developed by former students from Professor Hashimoto's laboratory. It is convenient for bench pressing and squatting. In this robot, the motors are not powerful enough to apply a torque equivalent to 100 kg on the bar, so the former students used clutches to reduce the angle velocity of the motors and increase the final torque.

The second robot, a "new-generation robot," was developed by engineers who work with Professor Hashimoto on the fitness robots program. This new robot also has a metal frame to allow athletes to practice different exercises. With this frame, the athlete could work on the cable lat pull-down, cable chest press, cable triceps push-down, cable bicep curve, etc. This robot is, therefore, much more flexible.

Figure 4 – First robot made by students

Figure 5 – Second robot

1.2.2 Electronic structure of the robots

Before my internship, both robots used a PEX-361116 [2] and an LPC-632104 board [3]. The PEX board is an Analog Input Output (or AIO) board used to send commands to the motors and potential clutches. It can also acquire sensor values such as potential potentiometers. The LPC board is a counterboard (LPC stands for Low Pin Counter) used to acquire the position of encoders present at the crankshaft of the motors. These boards are inconvenient because they need the computer to have the correct PCIe slots on the motherboard. This issue will lead to a modification of the robots during my internship. Please refer to the section "Hardware modifications"

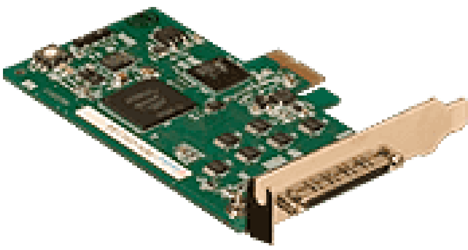


Figure 6 – PEX-361116 from ©Interface

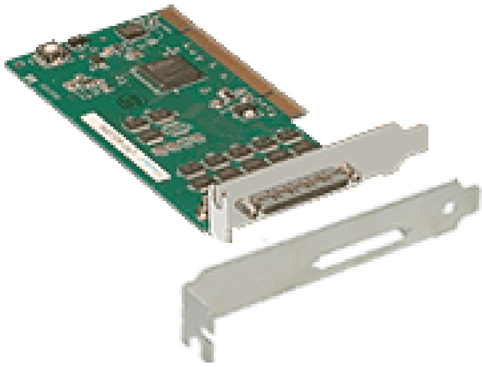


Figure 7 – LPC-632104 from ©Interface

1.3 One Repetition Maximum

The One-Repetition Maximum is the maximum load an athlete can lift in only one repetition. Professional athletes use this value to track their improvements in powerful movements. However, lifting such a load is hurtful for the athlete's joints and tendons, so we need to estimate it. We found two different methods to estimate it.

One RM Experiment with Epley, Lander and Mayhew.

The first way to calculate the One RM is to make as many repetitions as possible with a particular load. Then, a quick formula returns the One RM value. The scientists Epley, Mayhew and Brzycki (or Lander) developed different formulas to estimate One RM. [4] [5] [6]

Epley:

$$1\text{RM} = w\left(1 + \frac{r}{30}\right)$$

Lander:

$$1\text{RM} = \frac{100 \cdot w}{101.3 - 2.67123 \cdot r}$$

Mayhew:

$$1\text{RM} = \frac{100 \cdot w}{48.8 - 53.8 \cdot e^{-0.075 \cdot r}}$$

With w being the load value lifted during the set and r being the number of repetitions done during the set. Information to notice: Lander's formula is really close to Brzycki's.

These methods are quick to follow because a single set of repetitions can estimate the One RM value. However, these methods rely on fixed formulas and do not consider the athlete's level of lifting. The formulas are the same whether the user is a beginner or a trained athlete. Moreover, the formulas are not really accurate if the athlete uses a load that is too light compared to their actual One RM. The heavier the athlete lifts during the experiment, the more precise the formulas are. Lastly, these methods can estimate the One RM value but not the One RM velocity (the estimated Velocity of the repetition at the One RM load). This is why another method is implemented in the application.

The One RM calculation by the three steps method is a more convenient way to estimate one's One RM value and One RM velocity. Their estimation is based on Velocity during repetitions. [7] We will ask the athlete to lift a specific load as quickly as possible and do as many repetitions as possible. They can do such a set with three different load values. We can predict that the Velocity during each set will decrease as the athlete gets tired. Something found through Reynolds's ([8]) and Picerno's ([9]) research and verified in the former student Miyu Kokubu's ([7]) research is that One RM could be estimated as the cross point between the linear regression of the first values of each set and the average value of the last values of each set. This cross point will give the One Repetition Maximum value and the One RM Velocity.

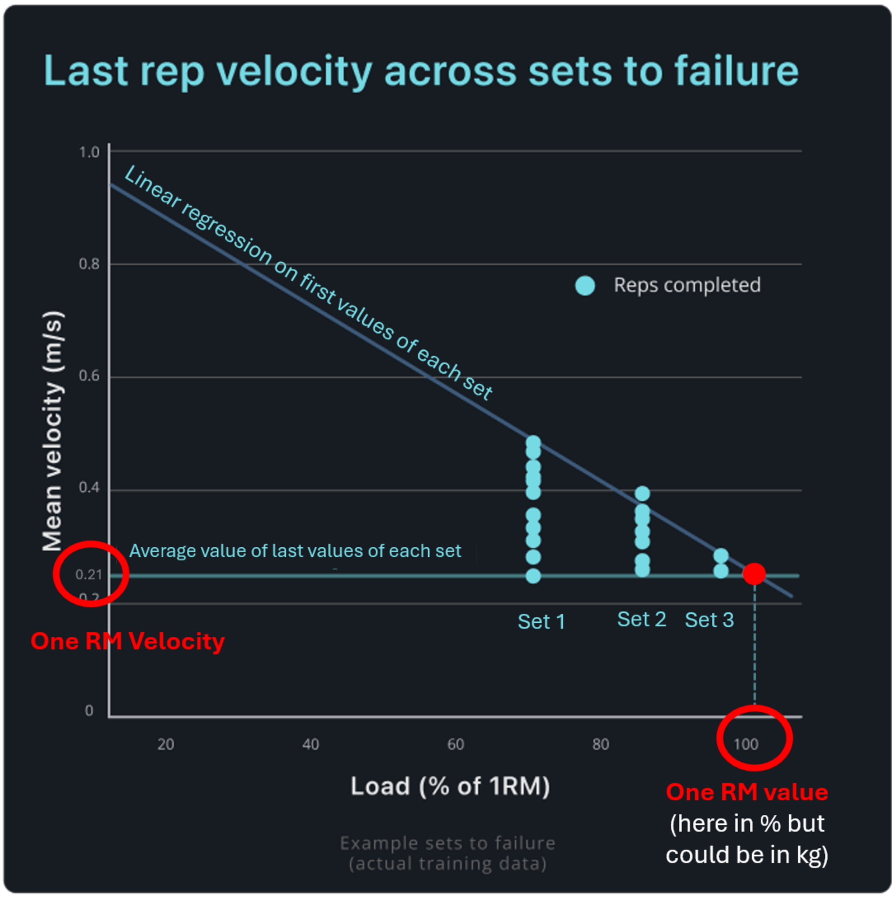


Figure 8 – Explanation of the One RM protocol - author: vbtcoach (graph modified)

1.4 Endurance

The other feature that professional athletes need to know about their performances is Endurance. Endurance is the athlete's ability to continue lifting the bar even if muscular fatigue rises. We will define it as an estimated maximal number of repetitions at a specific load.

First, during the user's One RM experiment, the computer stores the velocity loss from repetition to repetition. At the end of the experiment, the computer plots the data sets in a chart titled "Velocity loss / Number of repetitions in percentage of the set". It then computes the polynomial regression of this scatter plot.

In a future set, when the athlete proceeds with the exercise, the computer will acquire the velocity loss of the current repetition. It will then use the image of the polynomial regression expression to compute the remaining set percentage corresponding to the current velocity loss. Using the formula developed by former student Natsuo Tojo [10], the computer can estimate the number of repetitions remaining in the current set.

$$N_{reps\_remaining} = \frac{N}{N_{per} \cdot \frac{1}{100}}$$

Former student Natsuo Tojo's formula for computing the number of repetitions remaining knowing the athlete's muscular fatigue

$N_{reps\_remaining}$ is the estimated number of repetitions remaining in the set.
N is the number of repetitions already done in the current set.
N_{per} is the progress in the set (in percentage) corresponding to the measured velocity loss at the current repetition.

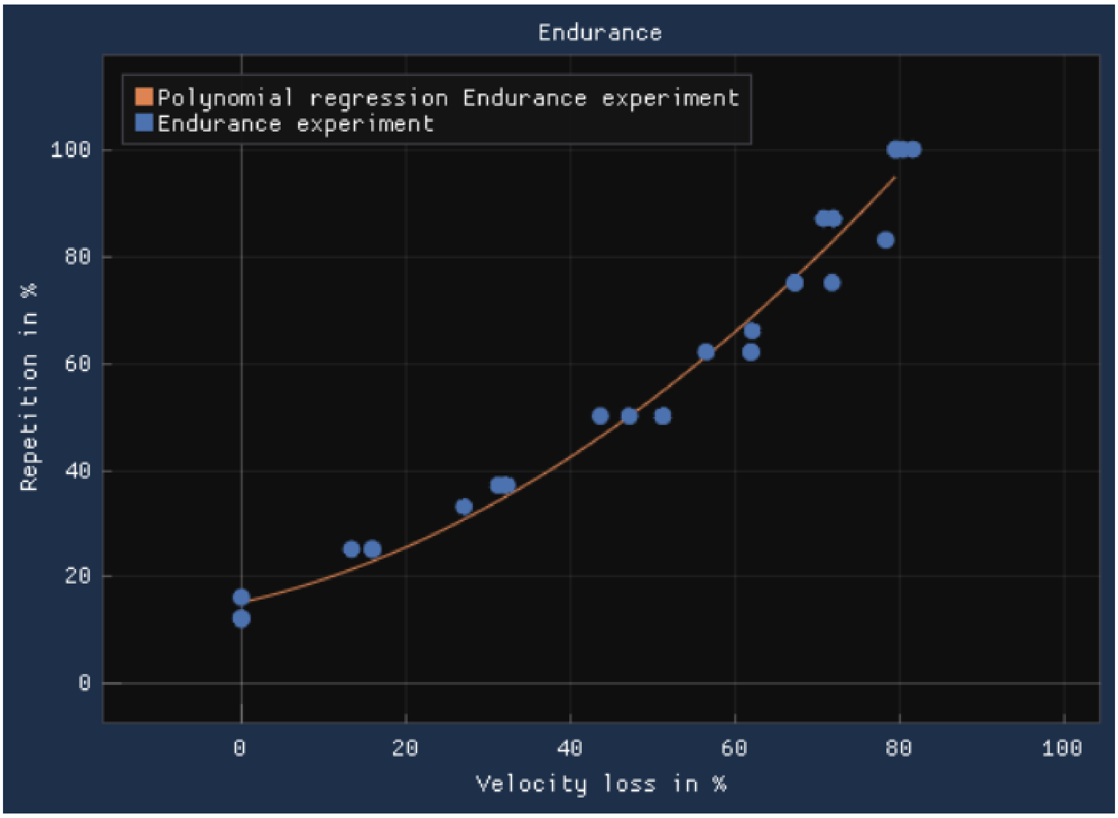


Figure 9 – Example of a real data set that was used to define the athlete's Endurance law (biceps curl, set at 10kg, 15kg and 21kg)

To conclude, the robots are working, and the Nippon Sport Science University athletes could use them. However, to control them, the user can only set a specific load and does not receive feedback on their performances. Moreover, the robots should be delivered with their particular computer as they need the correct PCIe slots on the motherboard for the interface boards. The challenge for the internship is to develop an application that could render the data for the repetitions (distance of rope pulled out, Velocity of the bar, acceleration of the bar, and torque applied). The application should be as scalable as possible. It should also implement the One RM protocol and the Endurance calculation. It should be able to save data from one session to another to let the athlete track their improvements. Finally, it should allow the athlete to use a profile to dynamically change the load applied to the bar as the athlete lifts it.

2 Contributions and Value Added to the Project

2.1 Hardware modifications

As explained before, both robots use LPC and PEX boards to communicate with the computer. This requires the computer to have the correct PCIe slots on the motherboard. This is highly limiting because the only computers with these slots on the motherboard are obsolete computers with the Windows Vista version installed. Firstly, Windows Vista is outdated, and the application has not been designed to work on this version of Windows. Secondly, we would like the robot to be able to be plugged into any computer.

The first modification was to replace the LPC and the PEX boards with USB-compatible interface boards. This way, the robot could be plugged into any computer with a USB port. Professor Hashimoto chose the AIO-163202FX-USB (called CONTEC AIO board in the rest of this document)[11] to replace the PEX board and the CNT-3204IN-USB (called CONTEC counter in the rest of this document) [12] to replace the LPC board.

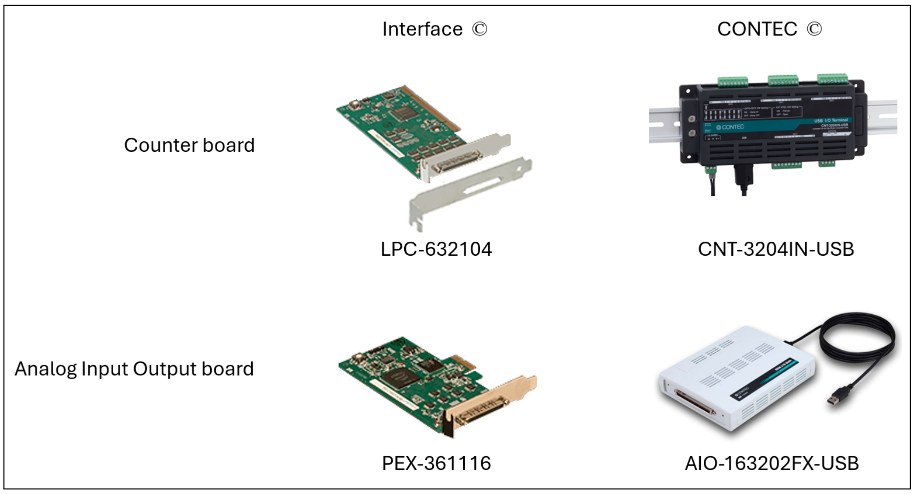


Figure 10 – Picture of the LPC, the PEX, the CONTEC counter and the CONTEC AIO.

My work at the beginning of the internship was to start developing the application and change the wire connections in the robot to use these new boards. We could do such work on the first robot (the one made by students) but not on the second one, as the connections were welded. For the second robot, with the help of Professor Hashimoto, we managed to plug in a converter that translate the PEX and LPC connections into USB ones. In order to have only one version of the application working on both robots, we needed to make the application able to recognise which board was plugged into the computer.

2.2 General presentation of the software project

The code has a tentacular structure. The class Trainer is the keystone of the code. The `main.cpp` file creates an object, TheTrainer, which would be the Project's main object. The `main.cpp` also manages the requirements linked to the ImGui library. ImGui is the module for the graphic interface developed by Ocornut and available on the internet in open source [13]. The graphs plots are managed with ImPlot, a library compatible with ImGui windows, developed by Epezent and available in open source on the internet [14].

2.3 General presentation of the classes

There are seven main classes, one dependent class (which could not exist without the main classes), and three tool classes (used by other classes).

- **`Trainer`** is the keystone of the application. We will use the six other main classes to create objects within `TheTrainer` object.
- **`BotManager`** refers to the robot's manager. It acquires data from the robot, computes the new command and sends the commands to the robot.
- **`TrainerApp`** refers to the graphic interface. Any other object that needs to render a piece of information will go through this `TheTrainer.app`
- **`DataManager`** refers to the manager of the data acquired from the robot. It saves the data in the database and plots the raw data in `TheTrainer.app`.
- **`TrainerPredictor`** is the "brain" object that will analyse the data during the sets and predict the estimated One RM and the Endurance of the athlete.
- **`TrainerSimulator`** is a way to create theoretical data instead of real data acquired from the robot. This `TheTrainer.simulator` replaces the `TheTrainer.botManager` in simulation mode. This is useful when the engineer wants to work on the application without any robot plugged in.
- **`TrainerDataRenderer`**. This last main class of the Project renders the saved past data. This object helps show a calendar with the user's past workout sessions.
- **`TrainerAnalyser`**. This class depends on the `TrainerPredictor`. We use it to create small objects that can analyse the sets in many ways. Every Analyser has a different mission, so please refer to the subsection "Software design concepts" in this report.
- **`Trapeze`**. This tool class is used to define the load profile shape.
- **`Line`**. This other tool class is used to render lines in the plots.
- **`Point`**. This last tool class is used to render points in the plots.

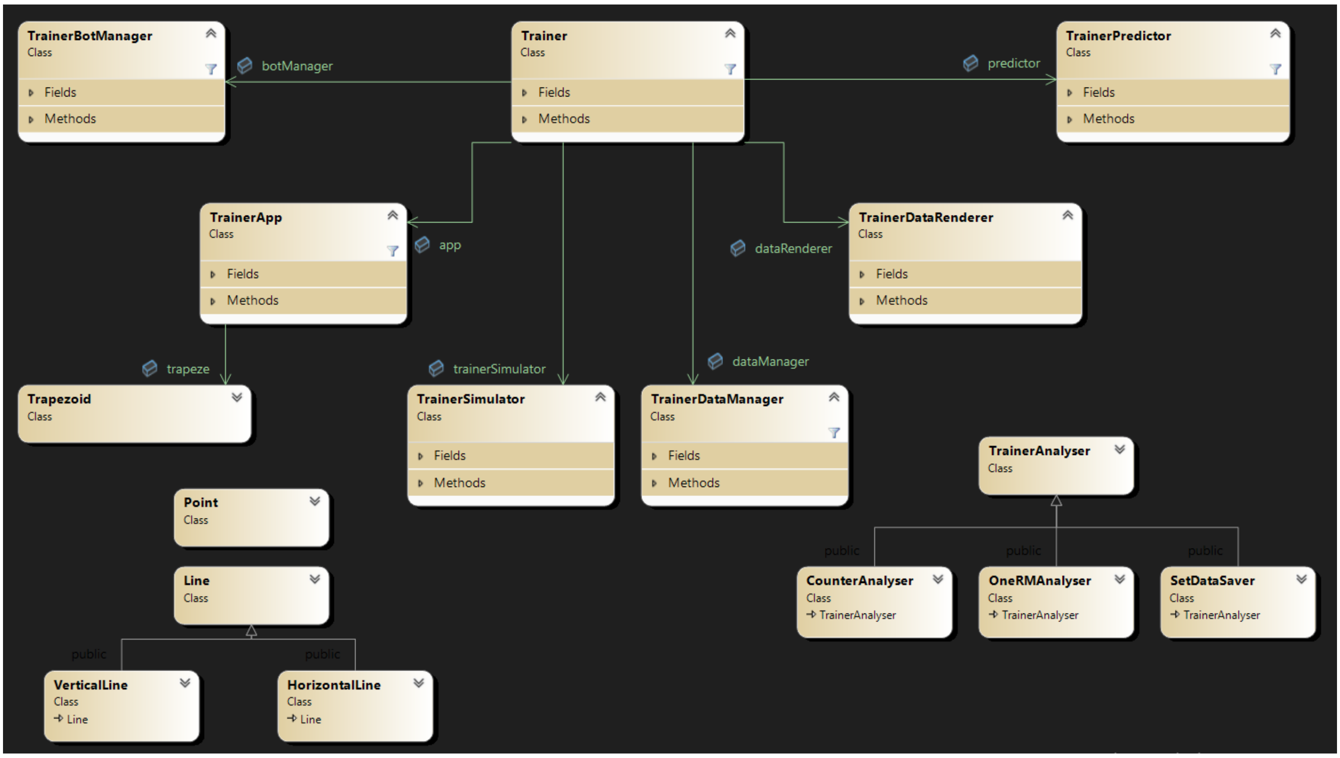


Figure 11 – Class diagram of the Project.

2.4 Data acquirement with a Kalman filter

As expected, the digital encoders used to get the motors' angular position are precise, but they still present some noise. In future calculations of the athlete's energy consumption, we will raise the speed to power two. This calculation will also raise the noise to power two. This is why it is necessary to reduce the noise dynamically.

The former students working on this Project used a simple moving average calculation to smooth the data acquired. The moving average consists of calculating the average of the last five values saved and the new raw value freshly acquired, then considering this estimated value as the acquired value.

Moving Average counting the last five acquired values

$$\widetilde{x}_k = x_k + \sum_{i=1}^{5} \widetilde{x}_{k-i}$$

k is the current value to acquire. $\widetilde{x}_j$ is the j smoothed value. x_k is the current raw acquired value

This method reduces the risk of high noise at the current acquired value. However, this method adds some delay in the acquisition and does not dynamically remove the noise.

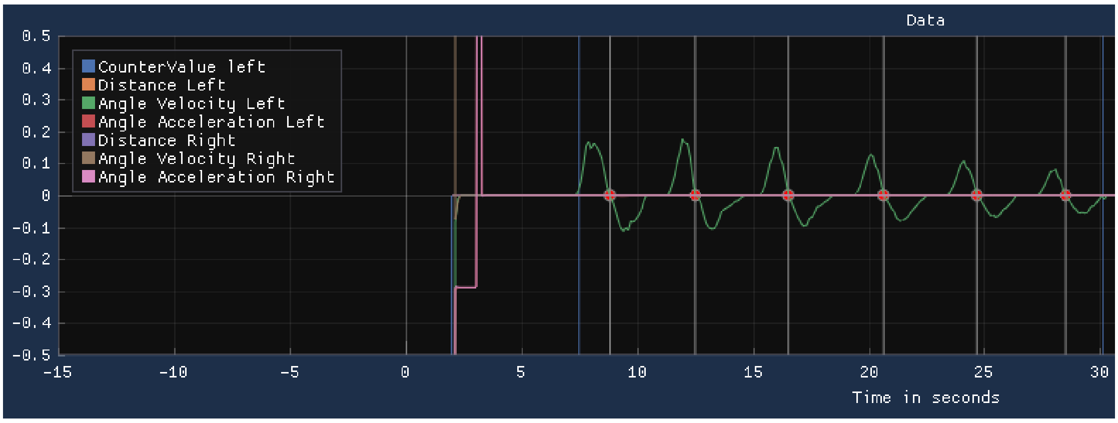


Figure 12 – Screenshot of the application after acquiring a data set using the moving average. The angular velocity (green) is choppy.

We decided to use a Kalman filter [15] to reduce noise dynamically and avoid any delay in the acquisition. However, using such a filter still caused a little delay. This is why we decided to use a filter that mixes the Kalman filter and an estimation filter.

The estimation filter will predict the next value. We will trace a line between the last two values. Then, we trace a vertical line at the x-value corresponding to the time of the future acquired point. The intersection of the two lines gives the estimated value of the future point. We call it the `predicted_value`. Then, we calculate the Kalman-filtered value of the freshly acquired point. And we calculate a weighted average between the `predicted_value` and the `Kalman_value`. This will be the smoothed value that we will consider as the result of the filter.

This method reduces the noise and the delay. It also smooths a potential micro rest of the user during the repetition. But it slightly exaggerates the extremal values of height. As a matter of fact, the predicted value will call for a continuing rise of the height when the user pulls up the bar (or a decrease of the height when the user pulls down the bar). So, at the top of the repetition, the prediction filter will wrongly make the height appear higher than it really is. However, this error is on the order of 0.5 degrees, so we consider it negligible.

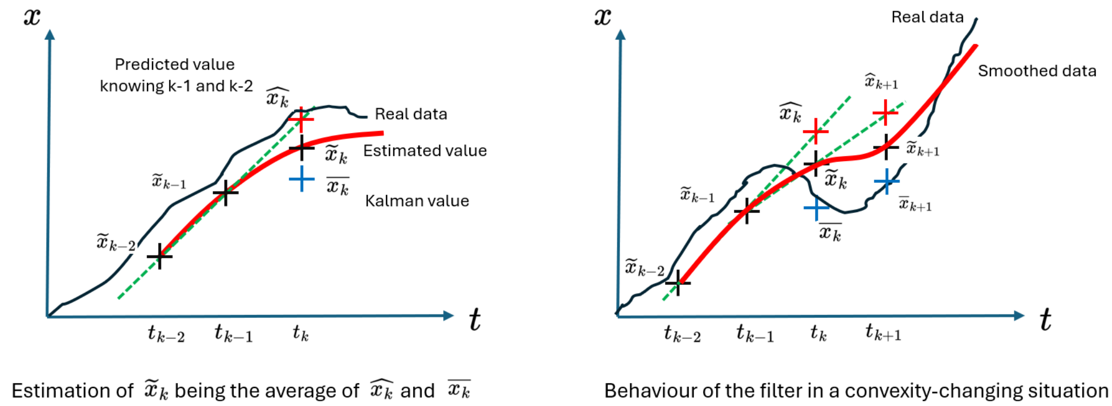


Figure 13 – Representation of the filter used.

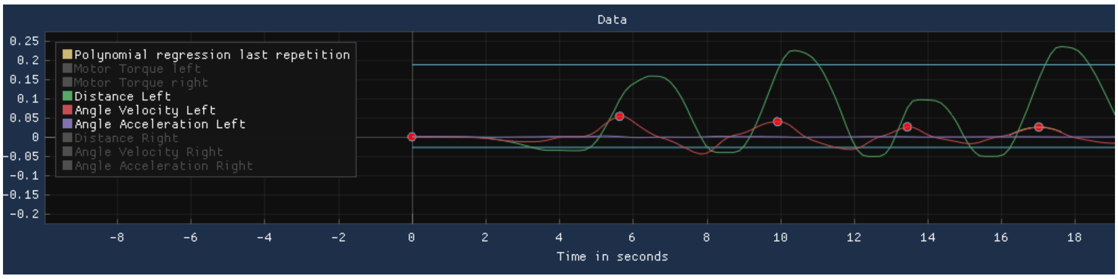


Figure 14 – Screenshot of the application using the filter. The distance and velocity are less choppy and closer to the real data.

2.5 Linear and polynomial regressions

The One Repetition Maximum experiment and the Endurance calculation require the development of tools to calculate regressions on data sets. To confirm the former students' studies about the polynomial asset of the Endurance law, we also need to compare the polynomial and the linear regressions.

2.5.1 Linear regression

Here is the protocol to compute the linear regression of a data set. Let $X = (x_i)_i$ be the abscissa of the set and $Y = (y_i)_i$ the y-value of the set.

— Compute $x_{mean} = \frac{1}{n}\sum_{i=1}^{n} x_i$ and $y_{mean} = \frac{1}{n}\sum_{i=1}^{n} y_i$

— Compute $z = \sum_{i=1}^{n} x_i y_i$ and $w = \sum_{i=1}^{n} x_i^2$

— Calculate $a = (z - n.x_{mean}y_{mean})/(w - n.x_{mean}^2)$;

— And Calculate $b = y_{mean} - ax_{mean}$

a and b give the slope and y-intercept of the linear regression for the data set. [16]

2.5.2 Polynomial regression

To calculate the polynomial regression, we used the `jacobiSvd` function from the `eigen-3.4.0` module. [17]

```
Eigen::VectorXd coeffs = X1.jacobiSvd(Eigen::ComputeThinU |
Eigen::ComputeThinV).solve(Y1);
```

2.5.3 Comparing polynomial and linear regression

We need the linear regression to compute the One RM experiment. To compute the Endurance, the former student's studies conjuncture a polynomial shape of the law, but we need to confirm it. This is why we decided to compute both the polynomial and the linear regression from the Endurance data set, and we will compare them. We used the F-test described in Claire Della Vedova's lessons to do so. [18]

F is a variable calculated so:

$$F = \frac{(RSS_{\text{linear}} - RSS_{\text{polynomial}})/(p_{\text{polynomial}} - p_{\text{linear}})}{RSS_{\text{polynomial}}/(n - p_{\text{polynomial}})}$$

— RSS_{linear}: Residual sum of squares for the linear model.
— $RSS_{\text{polynomial}}$: Residual sum of squares for the polynomial model.
— $p_{\text{polynomial}}$: Number of parameters (coefficients) in the polynomial model. (For our case, it is equal to 3)
— p_{linear}: Number of parameters in the linear model. (Here, it is equal to 2)
— n: The size of the data (the number of observations).

Then, we determine the p_{value} using F-distribution tables and compare it with an offset value of 0.05.
— If $p_{value} < 0.05$, the polynomial regression is a model that fits better to the data set than the linear model.
— On the other hand, if $p_{value} \geq 0.05$, the linear regression fits better.

In approximately 100 experiments, the polynomial regression was always the best model that fitted the Endurance. So, we can empirically confirm the former student's conjuncture.

2.6 Peak detection

Detecting the peaks is really important to analyse the sets automatically. But there are some challenges. First, without knowing the minimum height of the repetitions nor the maximum height, we should still be able to detect the beginning and the end of the repetitions. Secondly, concerning the first robot (made by students), the motors roll back the potential rolled-out rope before the set. This means that the minimum height of the repetition is not at zero and can be negative. This is another difficulty in the math problem. Finally, we need to detect the peaks in real-time to use the Endurance experiment results and tell the user the remaining number of repetitions as they proceed with the exercise. This is also a critical difficulty.

We then developed two calculations to detect the peaks. The first way is the most secure and could be done at the end of the set:

— Find the maximum height of the set. (We will name it `maxHeightOfSet`)
— Detect all the entries and exits of the height graph in the tube of radius 20% made around the `maxHeightOfSet` value.
— Isolate what we call the "counter reps". These are the canyons in between the repetitions.
— Store in a list the minimum heights during these counter reps. This represents the moment when the bar is at the lowest position between each repetition.
— Detect the maximum value of this list. (We name this value `maxMinHeightsOfCounterReps`)
— Detect all the entries and exits of the height graph in the tube of radius 10% made around the `maxMinHeightsOfCounterReps` value. These entries and exits are the repetitions' boundaries.

This method is safe but requires to have access to all the data of the set as we compute the protocol based on the maximal height of the entire set.

Now, we need a real-time method to compute the Endurance of the athlete at the end of each repetition. For this method we do not need to be precise on the repetitions' boundaries, we only need to count the repetitions. Even if we have no origin for the height graph, we can still analyse the velocity. When the velocity is negative and becomes positive, it is the beginning of a repetition. When the velocity is positive and becomes negative, it is the maximum height point. Knowing these two boundary points, we can detect the maximum velocity of the repetition and add one iteration to the variable that counts the number of repetitions.

This method is slightly less effective because when the user moves around the bar, it can cause a peak detection even if it is not a repetition. This is why we also store the maximum height of those peaks. If the maximum height of the current peak is inferior to 50% of the maximum height of the set, we consider that the algorithm made a mistake and misdetected a peak as a repetition. So, we delete the given peak from the repetition counter.

With the first method, we can also consider the acceleration and detect unsmoothed repetitions. If the acceleration becomes zero many times during the repetition, this means that the user makes pauses during the repetition. We consider this a mark of fatigue, so we inform the user of this weakness. By detecting the maximum height of each repetition, we can also detect when the athlete is too tired to bring the bar all the way up. We also consider this a hint of fatigue, so we inform the user of this weakness. With these tools, we can now give the athlete intelligent feedback on their set.

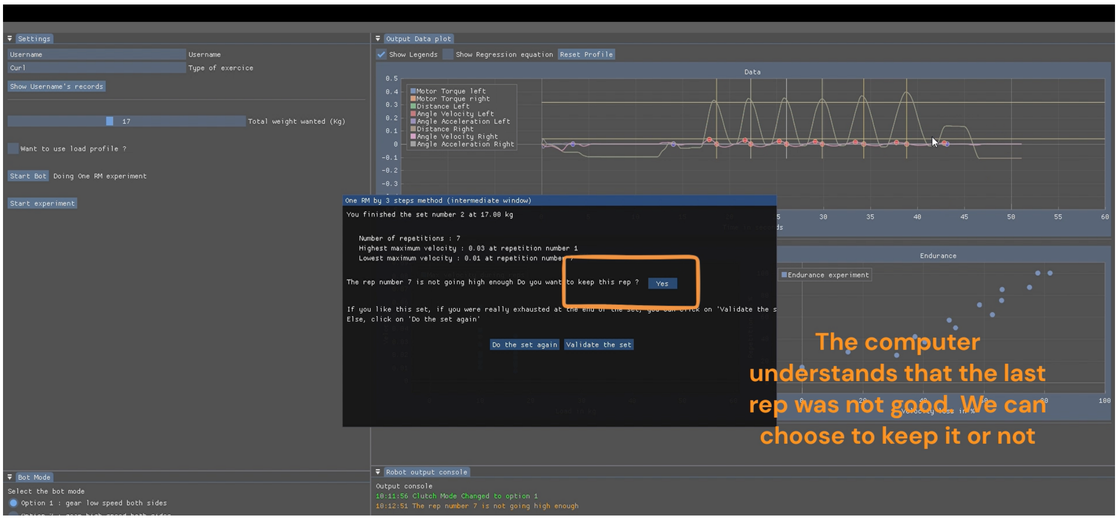


Figure 15 – Screenshot of the application when the `TrainerPredictor` object understands that the last repetition of the set is not high enough. The computer can also notify when the repetition is not smooth enough.

It is interesting to note that I struggled for three weeks before developing this solution. Another method I tried to implement was the local maxima research in the angular velocity data set. I also tried to estimate the actual maximum velocity by computing a polynomial regression on repetitions that I managed to delimit (thinking a repetition would have a polynomial form).

Theoretically, the peak detection and the maximum velocity research were effective (please refer to the following figure).

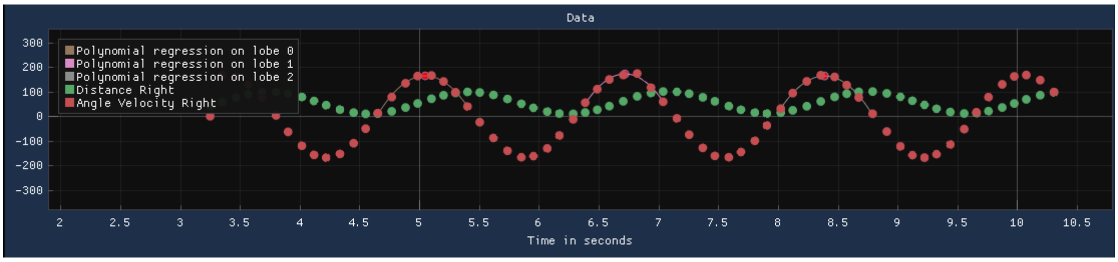


Figure 16 – Screenshot of the application using the failing peak detection method on a theoretical data set.

However, this method failed with the actual data as the athlete did not pull up the bar with a perfectly smooth and polynomial or sinusoidal movement (please refer to the following figure).

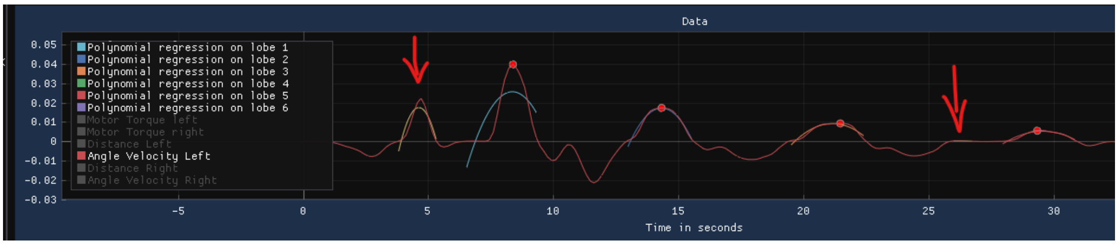


Figure 17 – Screenshot of the application using the failing peak detection method on a real data set. The polynomial regressions don't fit, and some peaks are not considered repetitions.

Since this failure, I implemented the `TrianerSimulator` class to generate real saved data to test the application's features instead of working on mathematical formulas to create the testing data.

2.7 Software development concepts

2.7.1 The Analysers as observers of the `TrainerPredictor`

In the ninth week of the internship, I realised the software was too linear and could not be scalable enough to add some missions (One RM by Epley, lander and Mahyew, One RM with three steps method, Endurance calculation, Repetitions counter, etc...). Then, I changed the code structure to gain scalability using the Observer/Observable concept [19]. I implemented little objects named `Analysers` that work as observers of the `TrainerPreditor` object. Each type of Analyser has its mission while observing the `TrainerPredictor`'s calculations.

— The basic analyser is a simple `DataSaver` analyser. It simply saves a set's data and stores its eigenvalues (maximum velocities, minimum velocities, beginning and ending points of repetitions, etc.)

— One clever analyser is the `OneRMAnalyser`. When it observes an end of set detected by the `predictor` object, it computes the One RM eigenvalues (One RM value, One RM velocity value, One RM linear regression, etc. Please refer to the "One Repetition Maximum" section for further explanations of these values.) and asks the `app` to plot the results.

— Another analyser used is the `CounterAnalyser`. It counts the number of repetitions in a set, which is really useful for the One RM experiment using Lepley's method.

Every type of `Analyser` inherits from the `TrainerAnalyser` class. This polymorphism concept makes the `Analyser` class highly scalable. As a matter of fact, we can easily add a new type of Analyser without changing the philosophy of the application.

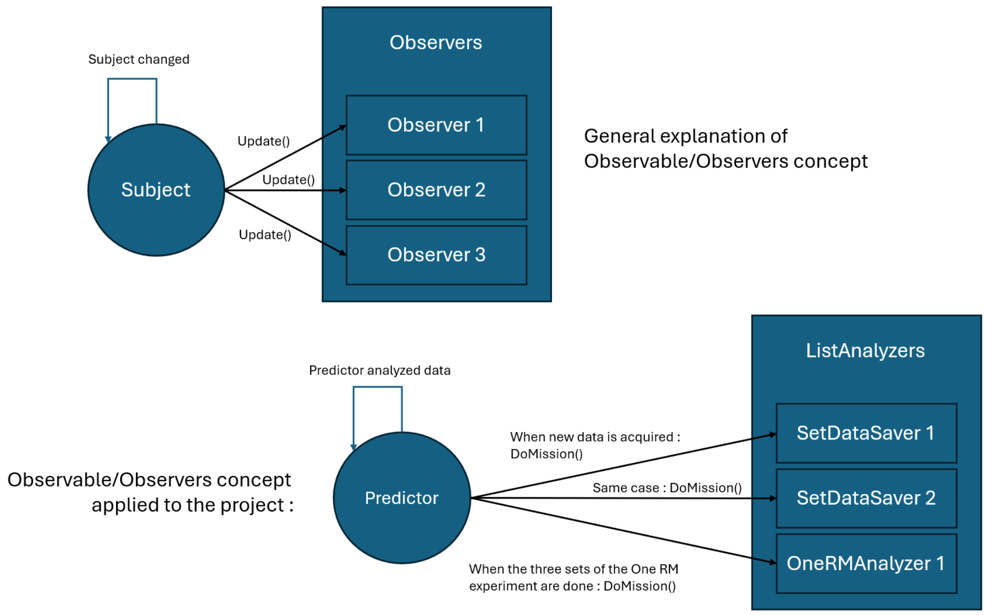


Figure 18 – Explanation of the observable/observers concept and its application in the Project.

2.7.2 The plotting mechanism

As explained previously, the `TrainerApp` object plots the data, but plotting is one of the primary missions of the application. So, we need the solution to be highly scalable and flexible. Knowing these requirements, we used the concept of Observer/Observable but slightly modified. The `TrainerApp` has a list-of-lists-type variable to save the information on every scatter plot (`listXToPlot, listYToPlot, listLabelsToPlot, listStylesToPlot,` etc. These lists are detailed in the CoachBot App Datasheet. Still, this report is not a technical manual). Each of these lists is accessible by the other objects of the Project. Any object can plot a data set in an empty slot of these lists. The `TrainerApp` object just plots what is saved in these lists. This way, we separated the Observable (the `TrainerApp`) and the Observers (any other object of the Project).

	Plot 1 (Raw data)	Plot 2 (Raw data)	Plot 3 (empty)	Plot 4 (One RM)	Plot 5 (OneRM regression)	Plot 6 (Endurance)
`listLabelToPlot`	Motor torque	Angular position	« »	Maximum velocities	« »	Endurance
`listXToPlot`	{0.0, 0.5, 1.0, 1.5} (time in seconds)	{0.0, 0.5, 1.0, 1.5} (time in seconds)	{0.0, 0.0, 0.0}	{10, 10, 10, 15, 15} (One RM exp for 3 reps at 10 kg and 2 at 15)	{10, 15}	{10, 15, 40, 90} (velocity loss in %)
`listYToPlot`	{15, 15, 15, 15} (Motor Torque in kg)	{5.0, 6.7, 7.9, 9.0} (Angular position)	{0.0, 0.0, 0.0}	{0.7, 0.6, 0.5, 0.5, 0.4} (Max Velocity for each rep)	{0.66, 0.48}	{12, 40, 60, 100} (number of reps in %)
`listIndexesToPlot`	0.0	0.0	0	1.0	1.0	1.1
`listStylesToPlot`	lines	lines	« »	scatter	lines	scatter
`listAnnotationsToPlot`	« »	« »	« »	« »	y = −0.036x + 1.02	« »
`listColorsToPlot`	ImVec4(0.0f, 0.0f, 0.0f, 0.0f) (default will let ImPlot choose)	ImVec4(0.0f, 0.0f, 0.0f, 0.0f) (Same)	ImVec4(0.0f, 0.0f, 0.0f, 0.0f)	ImVec4(1.0f, 0.0f, 0.0f, 1.0f)	ImVec4(1.0f, 1.0f, 0.0f, 1.0f)	ImVec4(0.0f, 1.0f, 0.0f, 1.0f)
`listOfPlotsToKeep`	False	False	True	True	True	True

Figure 19 – Example of five plots' data saved in the lists-type variables of `TrainerApp`.

2.8 Various features implemented in the application

2.8.1 The Load Profile creation

One of the most surprising but interesting features the client wanted is a flexible Load Profile. When the athlete moves the bar around to get into position for the exercise, the robot shall apply a low torque to the bar (only to keep the ropes well rolled in). Then, at the beginning of the movement (20% of the movement, for example), the robot shall apply a torque raising linearly from zero to the final wanted value. During the next part of the movement (60% of the movement, for example), the robot shall apply a torque equal to the final wanted value. For the last part of the movement (the remaining 20%), the motors apply a linearly decreasing torque. This protocol prevents muscular wounds and preserves the tendons, especially in traction-type exercises.

To sum up, this option is a dynamic torque command depending on the distance of the rope pulled out by the user.

We made this feature fully flexible; the user can drag and drop the points of the trapeze anywhere they want to create their own load profile. They can also add or remove points on the trapezoidal shape.

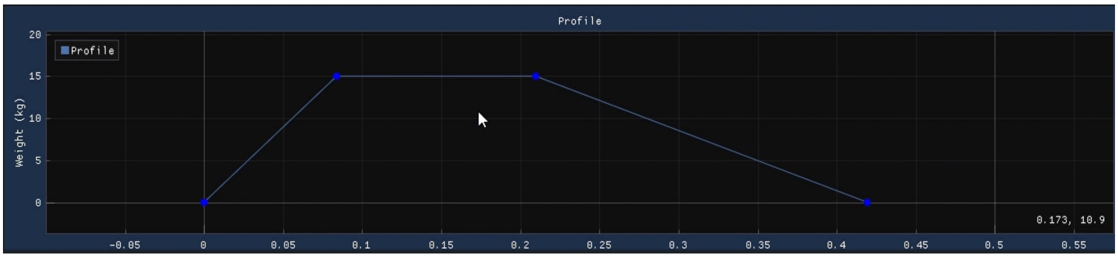


Figure 20 – Screenshot of the Load Profile option. Fully flexible, the user can drag and drop the points to modify it.

2.8.2 The database creation

Another feature I implemented was the communication with a local SQL database [20]. In the future, the database should be installed on a server and accessible online, but I did not have time to do so. The database is still locally installed on the computer. Users can access past training sessions on a calendar-looking page when they enter their username.

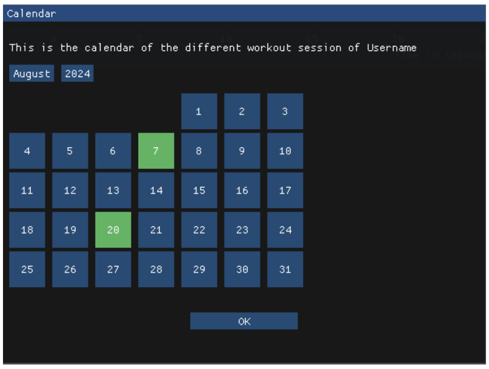


Figure 21 – Calendar-looking page to access previous training sessions

Figure 22 – Summary of the past training session

They can also access data on previous One RM experiments they could have done.

Figure 23 – Example of a past One RM experiment accessed via the calendar

The database structure is still in beta test, as I needed to change it many times before creating a stable version, which my successor on this Project might need to modify. Therefore, I tried to make a complete structure, avoiding redundancy.

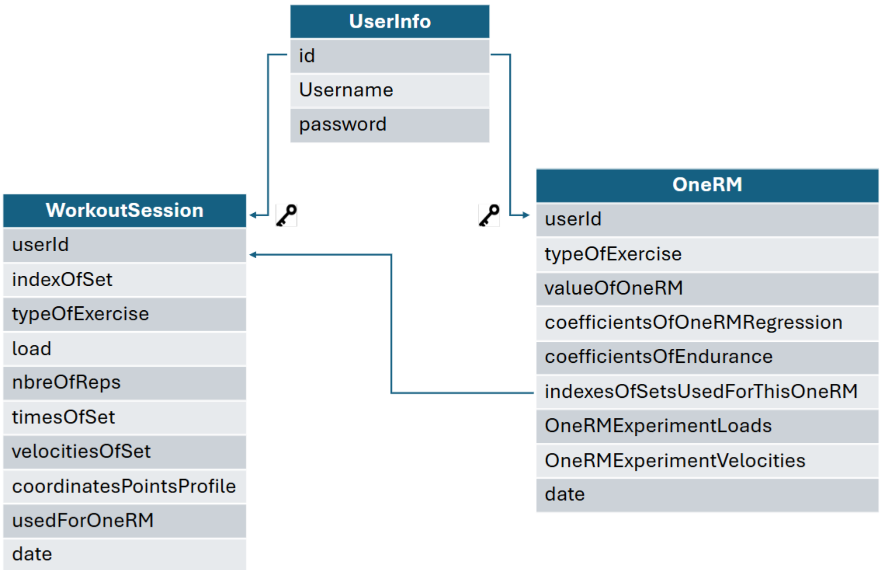


Figure 24 – The database structure

Conclusion

The development of this application was a real challenge; we succeeded in producing a fully functioning version. Through version 1.1, we also implemented the calendar option, which exceeded the planned expectations. We tried to make the CoachBot application as scalable as possible at a technical scale. This internship was a great challenge for me on a cultural and linguistic dimension as well as on a technical scale. I really enjoyed leaving my safe zone and improving my development skills.

I had the chance to present the application at the Nippon Sport Science University to professional sports coaches and multiple representatives of different companies in the tech industry. I therefore highly thank Professor Takuya Hashimoto for this fantastic opportunity. I was proud to work on such an exciting and promising project.

References

[1] Tonal. Tonal website - existing robot in the market. https://www.tonal.com/. 10

[2] Interface. Interface pex-361116 board documentation. https://www.interface.co.jp/pub/prdc/catalog/PEX-361116.htm. 11

[3] Interface. Interface lpc-632104 board documentation. https://www.interface.co.jp/pub/prdc/catalog/LPC-632104.htm. 11

[4] Boyd Epley. Poundage chart. In *Boyd Epley Workout*. Body Enterprises, 1985. 12

[5] Jerry Mayhew. Accuracy of prediction equations for determining one repetition maximum bench press in women before and after resistance training. In *Journal of Strength and Conditioning Research*, page 1570–1577. Body Enterprises, 2008. 12

[6] Matt Brzycki. In *A Practical Approach To Strength Training*. Body Enterprises, 1985. 12

[7] Miyu Kokubu 国分 美柚. Estimation of 1rm and muscular endurance based on velocity. In 挙上速度に基づく *1RM* および筋持久力の推定 *, Estimation of 1RM and Muscular Endurance based on Velocity*. Tokyo University of Science, Hashimoto lab, 2023. 12

[8] Jeff M Reynolds. Prediction of one repetition maximum strength from multiple repetition maximum testing and anthropometry. https://pubmed.ncbi.nlm.nih.gov/16937972/, 2006. 12

[9] Pietro Picerno. 1rm prediction: a novel methodology based on the force–velocity and load–velocity relationships. https://link.springer.com/article/10.1007/s00421-016-3457-0, 2016. 12

[10] Natsuo TOJO 東城夏生. Calculation of endurance based on velocity loss. In 多種目筋トレ装置の開発と負荷-速度関係に基づく筋特性評価*, Development of Multi-purpose Resistance Training Device and Muscle Characteristics Evaluation Based on Load-Velocity Relationship*, page 69. Tokyo University of Science, Hashimoto lab, 2021. 13

[11] CONTEC. Contec aio board documentation. https://www.contec.com/jp/products-services/daq-control/pc-helper/usb-module/aio-163202fx-usb/feature/, Ver.1.06. 16

[12] CONTEC. Contec counter documentation. https://www.contec.com/products-services/daq-control/pc-helper/usb-module/cnt-3204in-usb/feature/, Ver.1.02. 16

[13] Ocornut. Imgui open source code. https://github.com/ocornut/imgui, Ver.1.91. 17

[14] Epezent. Implot open source code. https://github.com/epezent/implot, Commit f156599. 17

[15] Fabrice COMBLET. Kalman filter lessons. In *Kalman filter*. Ecole nationale des études et techniques avancées Bretagne, Brest, France, 2024. 19

[16] MIT. Linear regression mit lessons. https://math.mit.edu/~dav/05.dir/class26-slides.pdf, 18.05 Spring 2018. 21

[17] Doxygen. Documentation for the jacobi module in c++. https://tmp.mosra.cz/eigen-docs/classEigen_1_1JacobiSVD.html. 21

[18] Claire Della Vedova. In *Della Data*. Della Data, 2020. 21

[19] Joel CHAMPEAU. Software design. In *Conception logicielle*, page 15. Ecole nationale des études et techniques avancées Bretagne, Brest, France, OOD-CI2-Part1-V5-29-01-24. 24

[20] Loïc LAGADEC. Sql and database lessons. In *Bases de données*, page 15. Ecole nationale des études et techniques avancées Bretagne, Brest, France, 2022. 27

Appendix

List of Versions

— **Version 0.1:** The initial release with the four-window structure and a working plot renderer.
— **Version 0.2:** Creating linear and polynomial regressions is possible. The computer can define the best regression to keep. The load profile option is developed.
— **Version 0.3:** The application can communicate with the robots through new interface boards from CONTEC company. A modified Kalman filter has been developed.
— **Version 0.4:** By manually setting boundaries around the repetitions, the user can make the computer understand the repetition amplitude and maximum velocity and proceed to a One Repetition Maximum Experiment. Epley, Mahyew, and Lander's methods have been implemented.
— **Version 0.5:** The computer can detect peaks and isolate each repetition to proceed automatically to the One Repetition Maximum Experiment.
— **Version 0.6:** The application can calculate the Endurance of the athlete using the data from a One RM Experiment.
— **Version 0.7:** The application includes all the dependencies and external libraries, making it fully transportable.
— **Version 0.8:** The application can adapt to the type of robot plugged in. The interface boards from CONTEC company and Interface company can be used. The detection of the robot type is automatic.
— **Version 1.0:** First fully operational application version. Developed on July 29th and presented at the Japanese Sport University the next day.
— **Version 1.1:** Development of the calendar option. The load profile option is more convenient.
— **Version 1.1.1:** A new option is available; the user can upload a past One RM experiment in the current session data.

Gantt Diagram

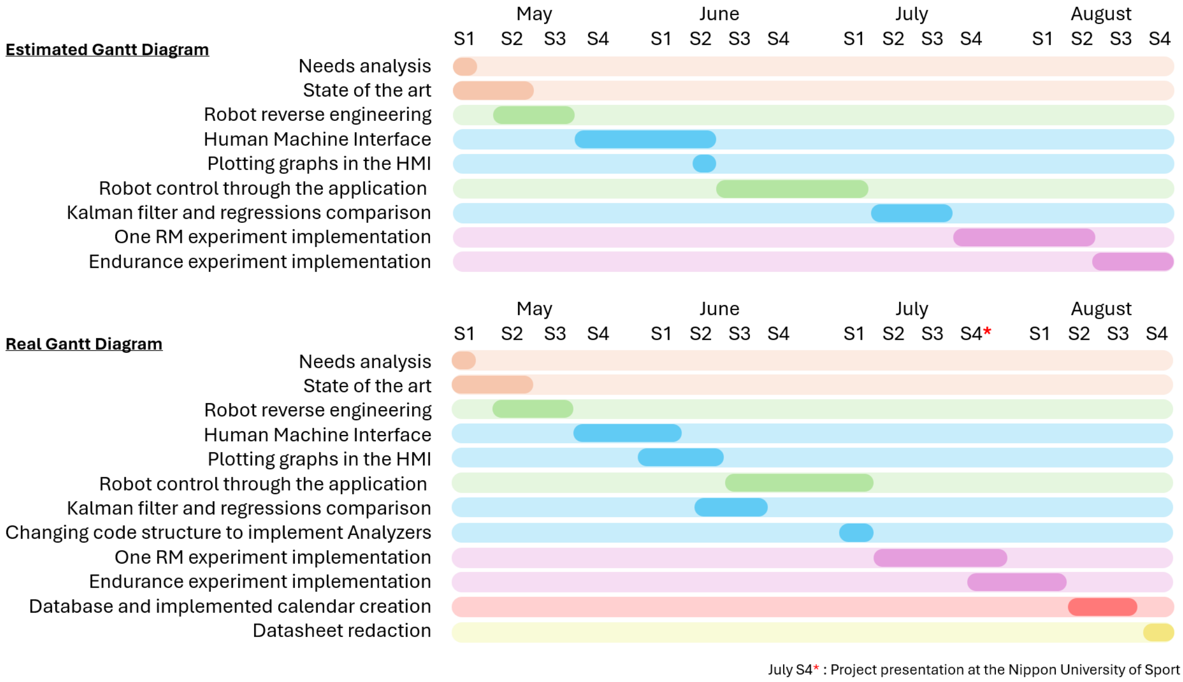


Figure 25 – Estimated and real Gantt Diagram

Class Diagram

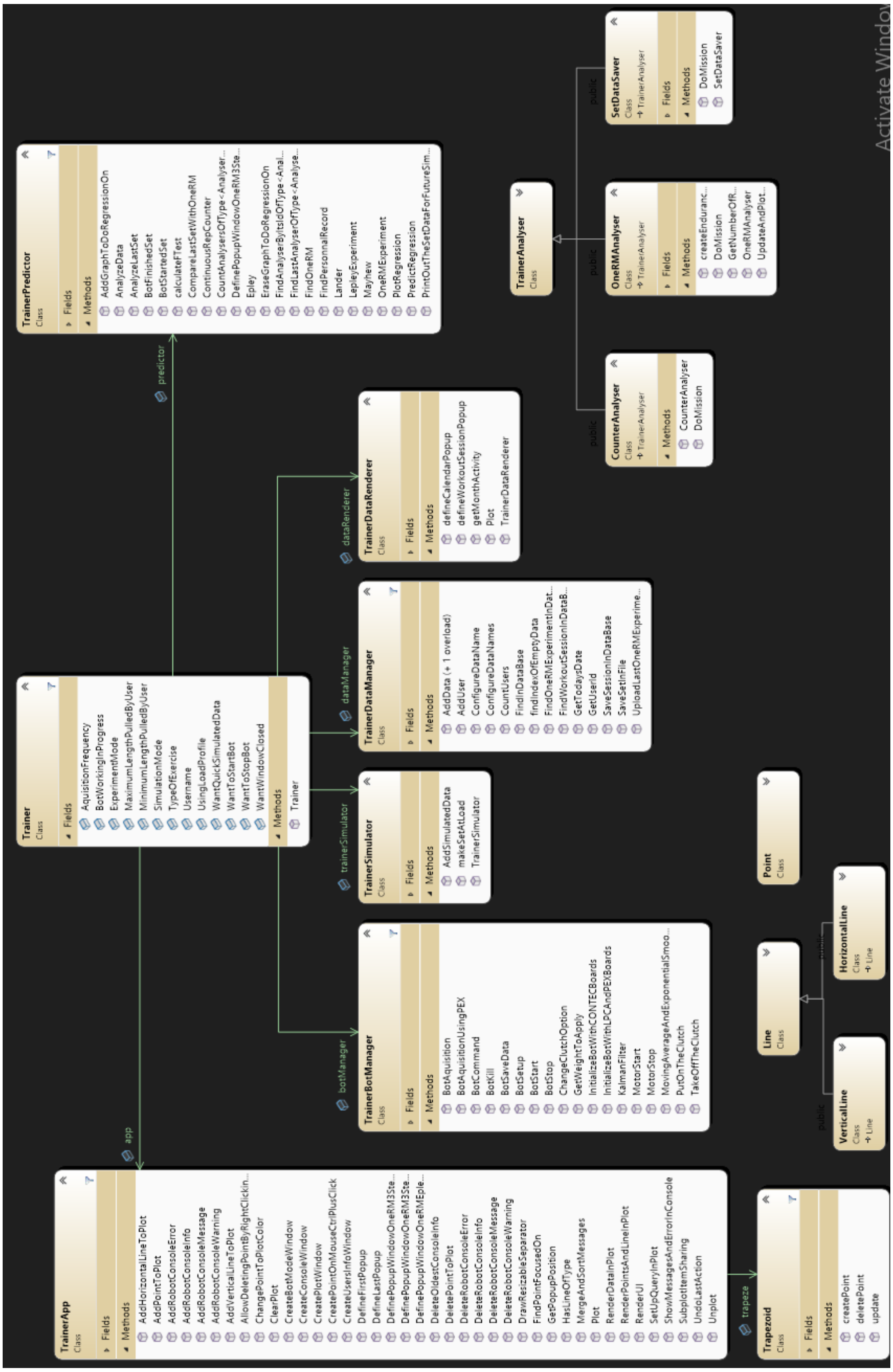
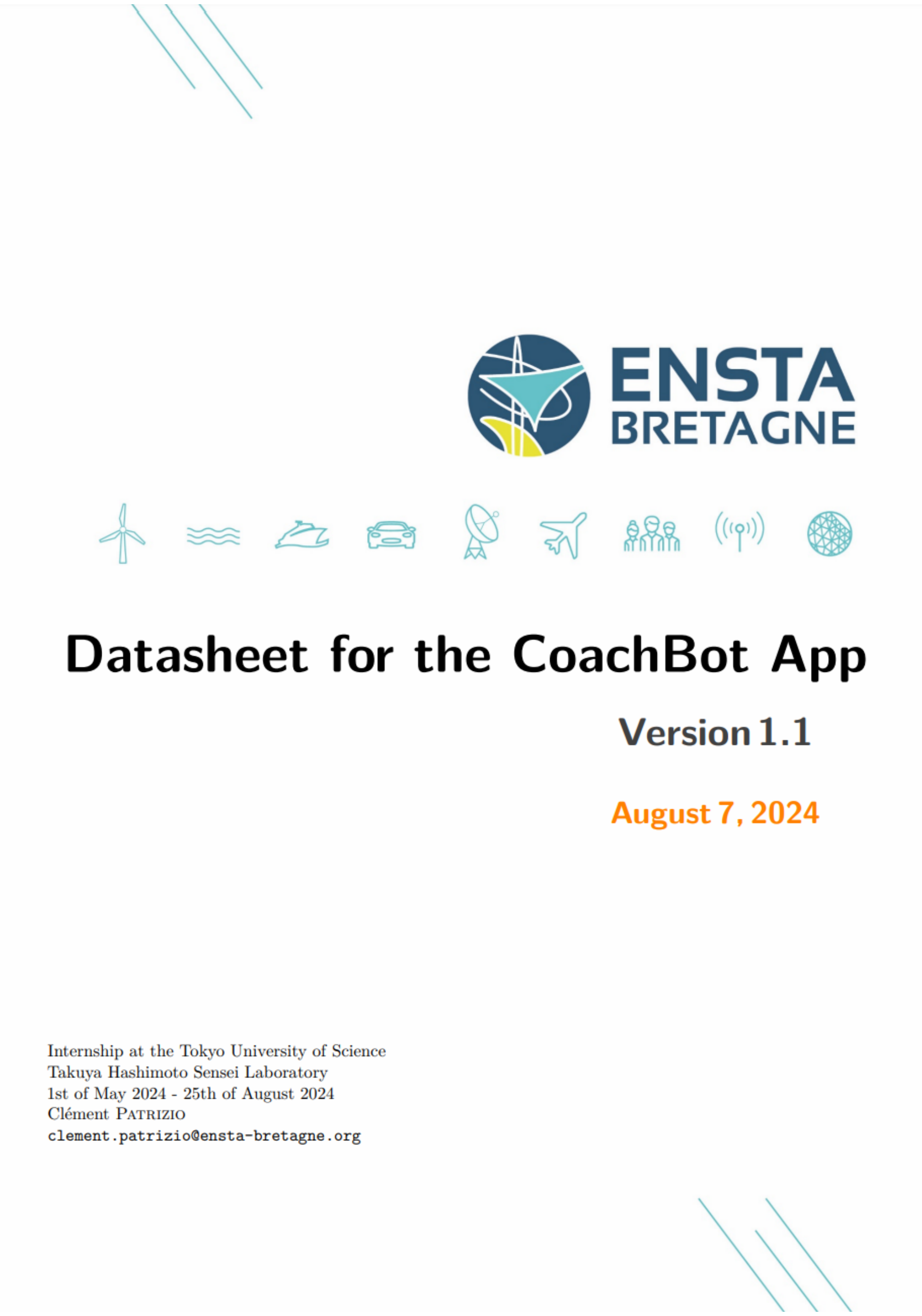


Figure 26 – Expanded class diagram

Application Datasheet



ENSTA BRETAGNE

Datasheet for the CoachBot App

Version 1.1

August 7, 2024

Internship at the Tokyo University of Science
Takuya Hashimoto Sensei Laboratory
1st of May 2024 - 25th of August 2024
Clément Patrizio
clement.patrizio@ensta-bretagne.org

Figure 27 – Datasheet of the application counting the user manual and and the technical manual (August 2024, 33 pages)

Assessment report

RAPPORT D'EVALUATION
ASSESSMENT REPORT

Merci de retourner ce rapport par courrier ou par voie électronique en fin du stage à :
At the end of the internship, please return this report via mail or email to:

ENSTA Bretagne – *Bureau des stages - 2 rue François Verny - 29806 BREST cedex 9 – FRANCE*
☎ 00.33 (0) 2.98.34.87.70 / stages@ensta-bretagne.fr

I - ORGANISME / *HOST ORGANISATION*

NOM / *Name* Tokyo University of Science

Adresse / *Address* 1-chōme-3 Kagurazaka, Shinjuku City, Tokyo 162-8601, Japan

Tél / *Phone (including country and area code)* +81 3-3260-4271

Nom du superviseur / *Name of internship supervisor*
Professor Takuya HASHIMOTO

Fonction / *Function*

Adresse e-mail / *E-mail address* tak@rs.tus.ac.jp

Nom du stagiaire accueilli / *Name of intern* PATRIZIO Clément

II - EVALUATION / *ASSESSMENT*

Veuillez attribuer une note, en encerclant la lettre appropriée, pour chacune des caractéristiques suivantes. Cette note devra se situer entre **A (très bien)** et **F (très faible)**
*Please attribute a mark from **A (excellent)** to **F (very weak).***

MISSION / *TASK*

- La mission de départ a-t-elle été remplie ? A B C D E F
  Was the initial contract carried out to your satisfaction?
- Manquait-il au stagiaire des connaissances ? ☐ oui/*yes* ☑ non/*no*
  Was the intern lacking skills?

  Si oui, lesquelles ? / *If so, which skills?*

ESPRIT D'EQUIPE / *TEAM SPIRIT*

- Le stagiaire s'est-il bien intégré dans l'organisme d'accueil (disponible, sérieux, s'est adapté au travail en groupe) / *Did the intern easily integrate the host organisation? (flexible, conscientious, adapted to team work)* A B C D E F

Souhaitez-vous nous faire part d'observations ou suggestions ? / *If you wish to comment or make a suggestion, please do so here*

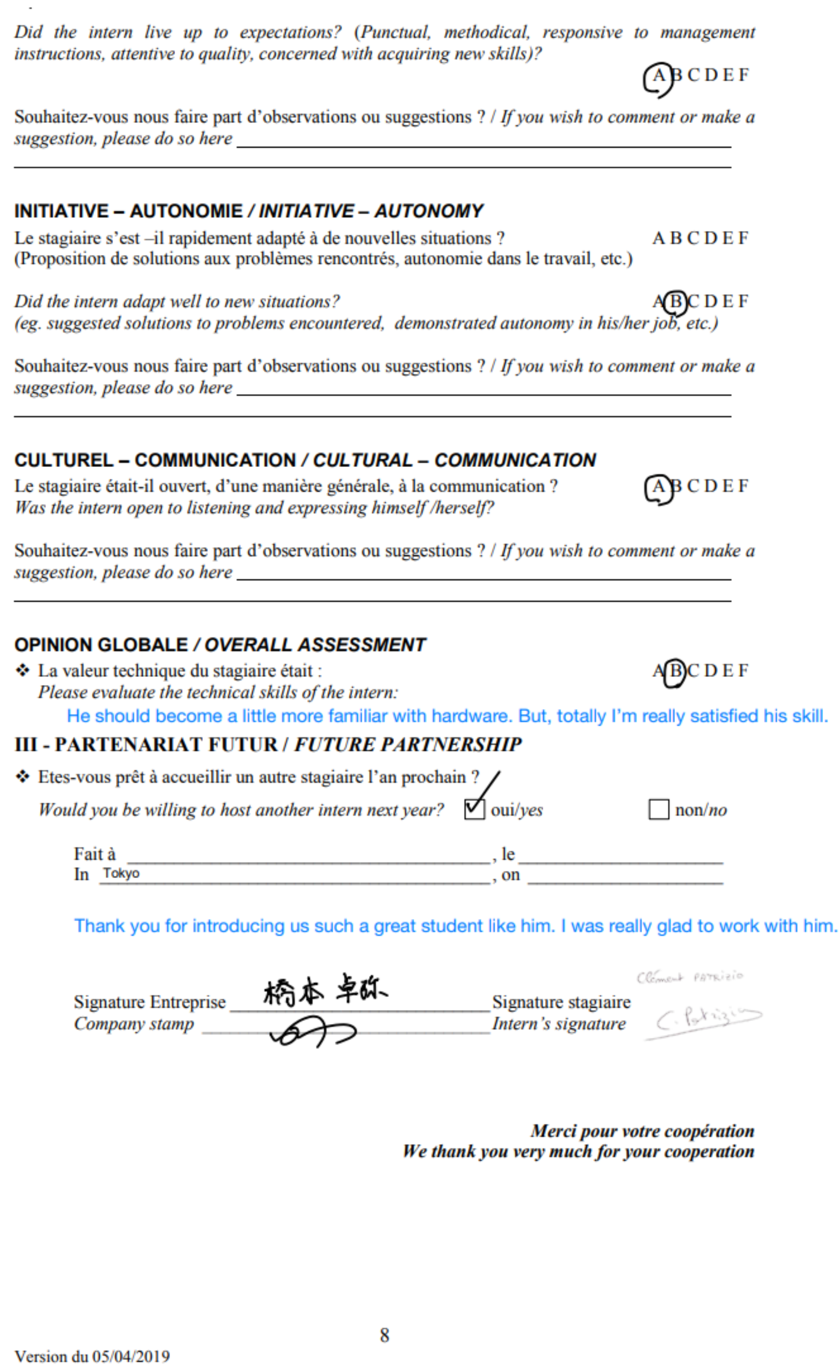
COMPORTEMENT AU TRAVAIL / *BEHAVIOUR TOWARDS WORK*

Le comportement du stagiaire était-il conforme à vos attentes (Ponctuel, ordonné, respectueux, soucieux de participer et d'acquérir de nouvelles connaissances) ?

7

Version du 05/04/2019

Figure 28 – Assessment report page 1/2



Did the intern live up to expectations? (Punctual, methodical, responsive to management instructions, attentive to quality, concerned with acquiring new skills)? A B C D E F

Souhaitez-vous nous faire part d'observations ou suggestions ? / *If you wish to comment or make a suggestion, please do so here* ______

INITIATIVE – AUTONOMIE / *INITIATIVE – AUTONOMY*

Le stagiaire s'est –il rapidement adapté à de nouvelles situations ? A B C D E F
(Proposition de solutions aux problèmes rencontrés, autonomie dans le travail, etc.)

Did the intern adapt well to new situations? A B C D E F
(eg. suggested solutions to problems encountered, demonstrated autonomy in his/her job, etc.)

Souhaitez-vous nous faire part d'observations ou suggestions ? / *If you wish to comment or make a suggestion, please do so here* ______

CULTUREL – COMMUNICATION / *CULTURAL – COMMUNICATION*

Le stagiaire était-il ouvert, d'une manière générale, à la communication ? A B C D E F
Was the intern open to listening and expressing himself /herself?

Souhaitez-vous nous faire part d'observations ou suggestions ? / *If you wish to comment or make a suggestion, please do so here* ______

OPINION GLOBALE / *OVERALL ASSESSMENT*

❖ La valeur technique du stagiaire était : A B C D E F
Please evaluate the technical skills of the intern:

He should become a little more familiar with hardware. But, totally I'm really satisfied his skill.

III - PARTENARIAT FUTUR / *FUTURE PARTNERSHIP*

❖ Etes-vous prêt à accueillir un autre stagiaire l'an prochain ?
Would you be willing to host another intern next year? ☑ oui/*yes* ☐ non/*no*

Fait à ______, le ______
In Tokyo, on ______

Thank you for introducing us such a great student like him. I was really glad to work with him.

Signature Entreprise 橋本 卓弥 Signature stagiaire
Company stamp *Intern's signature* C. Patrizio

Merci pour votre coopération
We thank you very much for your cooperation

8

Version du 05/04/2019

Figure 29 – Assessment report page 2/2