Innovative Project

Heart rate controlled electrical assistance for the Maillon Capitole lightweight vehicle

Brian Biendou
Marie Brunetto
Timothé Bigot
Achille Caute

Tutor : Thierry Monteil

Institut National des Sciences Appliquées de Toulouse

5A ISS - 2024/2025

January 28, 2025

Contents

Introduction

Nowadays, technologies tends to be integrated in more and more in our everyday life, mainly through the arrival of connected objects. A connected object is an object able to achieve tasks and to communicate, often wirelessly, with other devices. The purpose of the object can vary from a sensor, an actuator to complete tools such as phones. As future engineer specialized in the Internet of Things, it is essential that we understand the benefit and limitation of these objects through a project.

Chapter 1

Overview of our project

1.1 Innovative Project

In the Innovative and Smart System option, we have the opportunity to work on a project throughout the year to apply the knowledge that we acquired during our years of study. The innovative project is done by groups of four to five students, lead by a tutor. The subject is given by teachers, research laboratories or external companies.

1.2 Maillon Mobility

Maillon Mobility is a company founded in January 2023 by two Toulousain Paulin Fabre and Charles Pugnet. This company aims to create a lightweight vehicle, hybrid between an electric bike and a license-free car. The Maillon Capitole, their first patented vehicle, is an electric tricycle conceived for peri-urban travel. This vehicle is able to transport up to 2 adults and 2 children or 1 adult and 4 children, or 250kg of payload.
The vehicle is license-free and therefore can travel at a maximum speed of 25km/h. For now, the vehicle is still a prototype, but a commercialized version should be available at the end of 2025.

1.3 The requirements

The requirement that was given to us was to implement one or more functionalities to improve their Maillon Mobility vehicle. With that open subject, we had a wide range of possibilities when it comes to implementation related to our formation.

1.4 Technical Overview

1.4.1 Our ideas of implementations

Sensors for obstacle detection

Our first idea of implementation aimed to improve the integrity of the car and security of the driver and passengers by detecting the obstacles and warning the driver. The goal would be to use sensors to detect potential obstacles. However, we deemed that, even though the system would be interesting to implement, the connected aspect of the detection would not add enough to be pertinent.

Adaptive light

A second idea was to create adaptive light depending on the ambient luminosity. For instance, this would avoid blinding driver coming from the opposite side of the road.

Vitals monitoring for better control

A final idea was to gather vital information such as Heart Rate and VO2 to adapt the power provided to the pedals. This would improve the experience of the user and prevent risks in case of efforts that would be too important.

1.4.2 Vitals monitoring for better control

We decided to work on the vital monitoring for many reasons:

- This idea is the most useful from the three we had, combining comfort with security,
- The size of the project could allow us to have enough work to go through the semester and ideally end up with a functional prototype.
- The project combined electronic and a more programming oriented part, fitting our formations,
- Maillon Mobility expressed there interest on that specific subject. Our project could be a good insight into a first research and production on that subject.

The final idea of the implementation would be to gather the vital information from a smart watch, communicated and display on the user's phone. Finally, the information, along with the the wanted level of effort, are sent to the tricycle to be applied.

1.5 Organization

The project took place as a group of four person through an entire semester. We needed to be organize to avoid redundancy of the work and to ensure that our project is realistic and the deadline could be meet.

1.5.1 Work repartition

One important step of the project was the division of the work. We ideally needed to divide the whole project into four equally important part, that would not be dependent with one another. This aspect is important to avoid the need of waiting for another student to finish a part to be able to start another. The division with which we ended up working was:

- One would work on the watch and the communication with the phone, as well as the displaying of these information on the phone
- One would develop the backend of the phone application and set up the communication between the phone and the tricycle
- One would work on the communication and application of the information within the vehicle
- One would research and work on the control law of our system.

1.5.2 AGILE method

As advised by our teachers, we followed the AGILE method for our project to have a better control over the project. It consisted mainly on dividing our project into needs called user stories, that will be allocated to sprints, periods on time, often one or two weeks long. Our user stories are

- As a user, I want to be able to connect my phone to the tricycle so I can use the application.
- As a user, I want to be able to see my data so I can be informed on my activity.
- As a user, I want to be able to choose the level of effort so I can adapt my pedaling to my needs.
- As a user, I want to be able to enter my information so the system can adapt better to my age and physiology.

We then did a planning to decide when we should have the user stories realized.

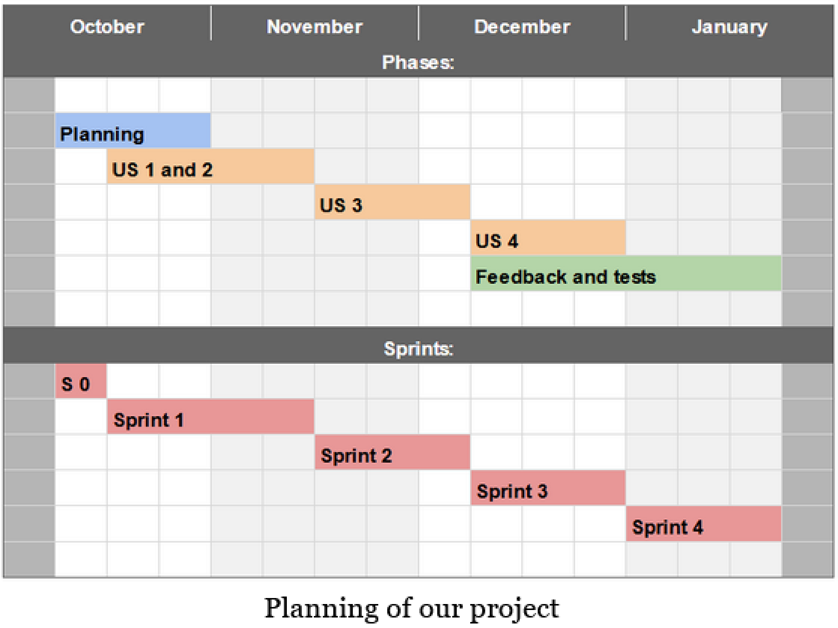

Figure 1.1: Planning of our project

1.5.3 Tool

To organized ourselves, multiple tools have been used:

- **Notion** is a productivity and note taking website. We used it to write our brainstorming and keep track of the progress of our project.
- **Github** i a developer platform that we used to store our project. Through is versioning capabilities, it allowed us to work on the same project and even files at the same time.
- **Overleaf** is a collaborative online LaTeX editor that we used to write our reports.

Chapter 2

Communication Watch - Phone

2.1 Problem and objectives

The communication between the smartwatch and the phone is a crucial step in our project, as it allows us to obtain the heart rate data needed for the adaptive control of the tricycle. The main objective is to acquire this physiological data in real time and transmit it to our mobile application so that it can be used according to the parameters defined by the user.

2.2 Initial Approach: Using a Garmin Watch

2.2.1 Developing with the Garmin SDK

Our first approach was oriented towards the use of a Garmin Vivoactive 3 watch, selected for its SDK (Software Development Kit) accessible to developers. We developed an application using the Flask framework in Python, capable of interacting with the Garmin API to retrieve cardiac data via the user account.

2.2.2 Limitations encountered

This approach, however, revealed several significant limitations:

Synchronization problem

The main constraint lies in the data synchronization mechanism. Indeed, the measurements remain stored locally on the watch until synchronization with the Garmin Connect application. This synchronization process is not automatic, the data accessible via the API remains static between two synchronizations, which does not correspond to our real-time monitoring needs.

API Access Restrictions

A second limitation concerns API usage quotas. While the exact specifications are not clearly documented, our testing revealed a limit of about 10 requests per 90-minute period, which is insufficient for our use case.

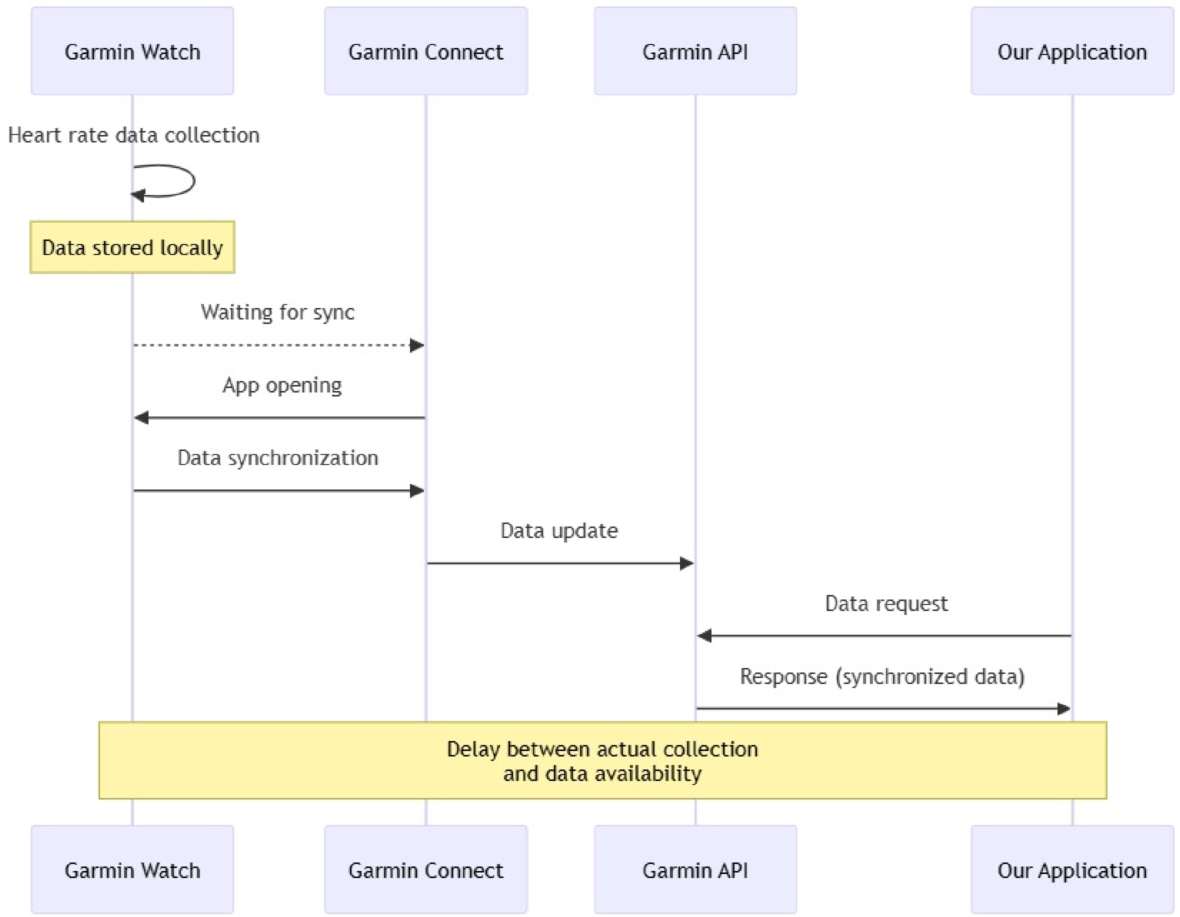

Figure 2.1: This diagram highlights timing and constraints.

2.3 Alternative solutions explored

2.3.1 API Garmin Health

One solution considered was to use the Garmin Health API, which offers more advanced features. However, since this API is reserved for enterprises and is subject to pricing based on request volume, this option was not retained for our university project.

2.3.2 ANT protocol

We also explored the possibility of using the ANT protocol, available on Garmin watches. While this solution allows direct data transmission, it requires specific receiver equipment not native to smartphones, adding unwanted hardware complexity to our system.

2.4 Final solution adopted

2.4.1 Dedicated heart rate sensor

Faced with these constraints, we turned to a more direct solution: the use of a heart rate sensor (pulse sensor) coupled with a microcontroller. This approach has the advantage of allowing real-time data acquisition and transmission via Bluetooth to our application.

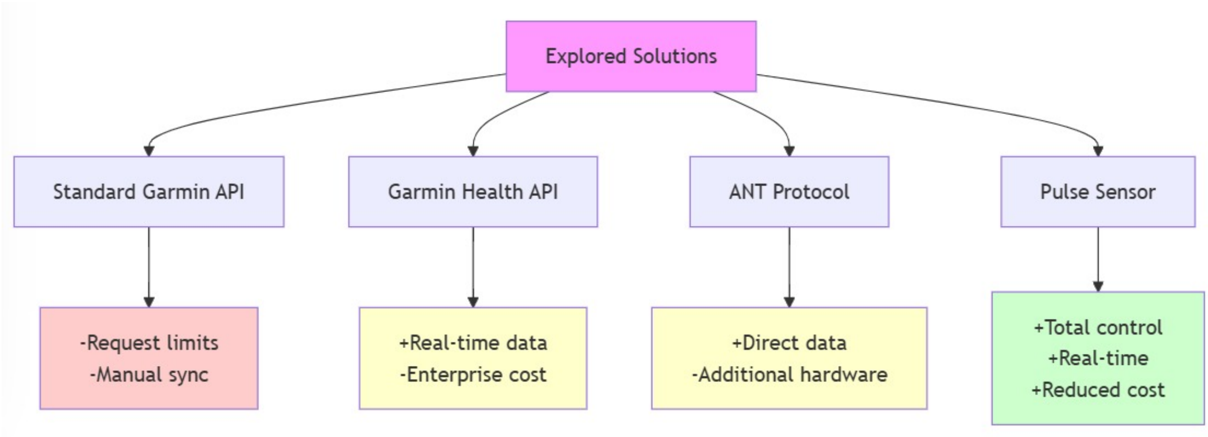

Figure 2.2: This representation allows to visualize the advantages and disadvantages.

2.4.2 Temporary implementation

Due to development delays and difficulties in routing data from the microcontroller to the application, we have temporarily implemented a simulated heart rate generator directly into the application. This interim solution allows us to continue development and testing of the other components of the system while waiting for the real data acquisition to be completed.

Chapter 3

Phone application

3.1 Front End

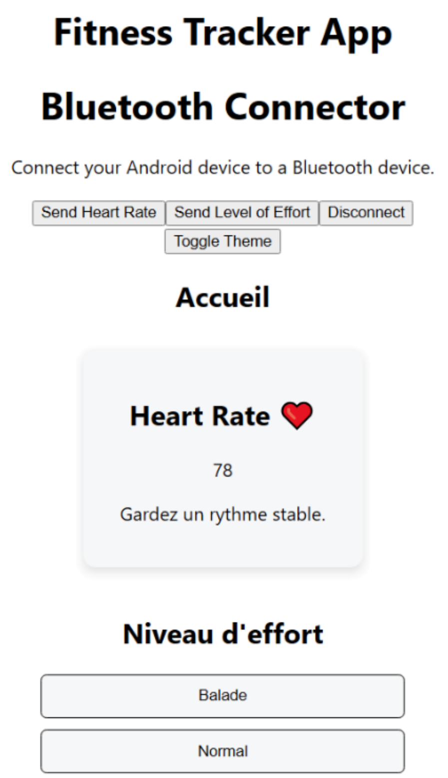

Figure 3.1: Former visual of our application

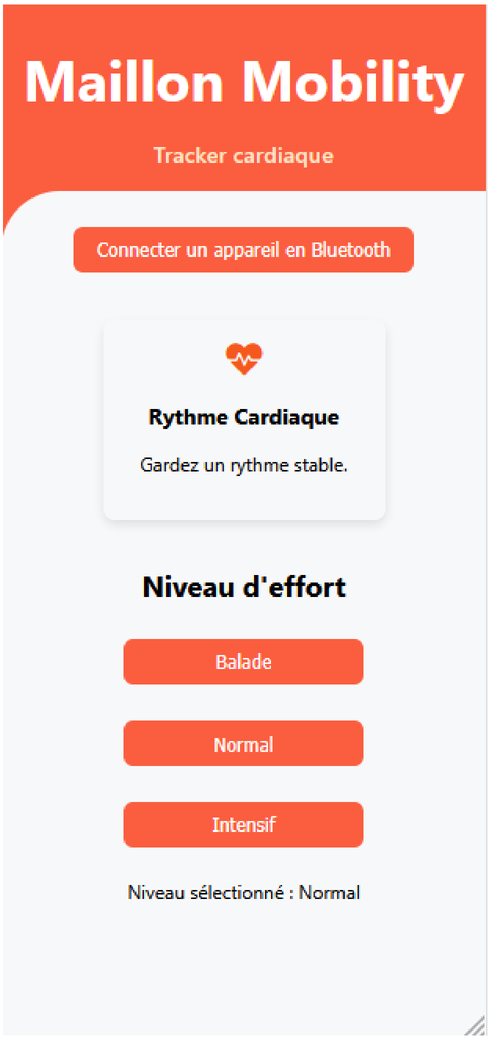

Figure 3.2: Current visual for our application

The Front End part of our application was developed using React Native, a JavaScript framework for building cross-platform mobile applications. The user interface evolved significantly during development, as illustrated in the figures3.1 et 3.2.

3.1.1 Initial interface

The first version of the interface (Figure 3.1) presented a minimalist approach with :

- A Bluetooth connector for pairing devices
- A simple heart rate display
- Basic control buttons for data management
- A two-level effort level selector (Ride and Normal)

3.1.2 Final interface

Final interface (Figure 3.2) has been significantly improved to offer a better user experience :

- A visual identity consistent with the Maillon Mobility graphic charter
- A refined design with cards for better visual organization
- A more intuitive display of cardiac data with explicit iconography
- A three-level effort level selection system (Ride, Normal, Intensive)
- A more professional interface with harmonious colors and better contrast

The new interface better meets user needs while maintaining consistency with Maillon Mobility's visual identity. Emphasis was placed on ease of use and clarity of the information displayed, essential for an application intended to be used during physical activity.

3.2 Back End

The back-end of an application is the section managing the data and communicating through APIs to other device. Even though the user does not interact directly with the back end, it is the part making the system functional and achieving its tasks. In our project, we needed to develop our back-end to get the information from the watch and the ESP, as well as the web server on which it will be hosted.

3.2.1 Development Language

The language of the back-end of an application is independent to the choice of the framework for the front-end. In our case, as we decided to use React for the front-end, we chose to develop our back end using JavaScript. Because React use JavaScript for its component, it was easier for us as it did not require to pass through an additional interface for the development. JavaScript is also a language that is well documented on internet, with enough library for functionalities such as BLE communication or random number generators.

3.3 Deployment

3.3.1 Type of deployment

Once the front-end and back-end were made, we needed to make it so the user can easily access and use it. As the aim is to use it on a phone, we had two solutions: a web page or an application. At first, we wanted to try to develop an application as the use of BLE (explained in the Chapter 5) would require a Bluetooth connection. However, after some difficulties with with Android Studio, a software to create APK to be run on an Android Phone, we decided to think about the possibility of a website. In the table 3.1 are the points we discussed about the advantages and inconvenient of a web page compared to an application. We concluded that a website would be easier, at least for development. Even more because, by the fact that we used react, we could conveniently convert it as an application.

Website	Application
+ Accessible from any device with a browser, meaning platform-independent. + Does not require installation. + Easier to update and maintain. − Limited offline functionality.	+ Can work offline in many cases. + Better integration with device features (e.g., push notifications, hardware access). − Platform-dependent (may need separate apps for iOS, Android, etc.). − Requires installation and regular updates. − Consumes device storage space.

Table 3.1: Comparison of Advantages and Inconveniences of Website and Application for deployment

3.3.2 Hosting the page

As we decided to make a website, we needed to host our page on a remote server. For this, we needed online storage space as well as a domain name to publicly access our project. Multiple online platforms could provide us these services, such as Google Cloud, OVH or Wix. However, we decided to take our chance with a service offered by Github called Github Pages. It offers up to one gigabyte of storage, and is perfect for development. As our project is already stored on Github, it was a convenient opportunity. Also, as the aim of the project was to give the final draft to Maillon Mobility, hosting it on our repository for the code to be accessed was a good option.

Chapter 4

Communication Phone - Tricycle

The link between the phone and the motor is important, as it is the one interfacing the user with the tricycle. It is important to properly define technologies and methods to develop this communication.

4.1 Technology for the communication

The first point we need to explore is the technology that will be used to pass the data between the phone and the tricycle. As you will see in chapter 6, we choose an ESP32 to interface the motor with the outside.

4.1.1 Existing Technologies

To choose which technology we want to use for the communication, we first need to explore the available technologies on both the phone and the ESP32.

For the phone, the available communication protocols are:

- **NFC**, or Near Field Communication. It is a very energy-efficient way of communicating, but with a very limited range of a few centimeters.
- **BLE and classic Bluetooth** are protocols developed for short ranged connected object communications. The estimated range of 10 meters for normal functioning.[**?**] BLE, Bluetooth ow Energy, the energy efficient version of Bluetooth is also available on mobile phone. This version gave more use of Bluetooth for IoT purpose.
- **Wi-fi** is a long-range communication protocol, allowing the exchange of a very large amount of data. Due to its important energy consumption, it is rarely used for IoT purpose.
- **GSM, GPRS, LTE** and other protocols relying on the mobile networks. It has a really long range, and allows to transport an important amount of data. Due to its large energy consumption, it is rarely used for IoT purpose.

For the ESP32, the communication protocols that are natively available are: [7]

- **BLE and classic Bluetooth**, previously described.
- **ESP-NOW**: developed by Espressif, a Chinese company working in affordable communication, ESP-NOW is an interesting protocol communicating 250-byte payload on a range up to 220 meters long, interesting for IoT applications.
- **Wi-fi**, previously described.
- **MQTT** is a publish/subscribe based protocol that is widely used in IoT. But this would require to use a broker, an additional element to our system.

Other protocols such as GSM, GPRS, LTE or LoRa are available using additional chips, SIM card or antenna. However, we will not consider them here, as we want to keep the system simple, and the ESP already share enough protocols with the phone to make our choice.

4.1.2 Reasons of our choice

Numerous protocols can be used on both mobile phone and ESP32. As we do not want to add any module to our ESP if necessary, we decided to focus on the pre-existing ones. The common protocols are Wi-Fi, Bluetooth and BLE. For our application, we require to transmit low amount of data between a tricycle and a phone in direct communication, a few meters apart maximum. For this reason, even though Wi-Fi has an alternative, called Wi-Fi direct, allowing communication without Access Point[1], it remains an energy-intensive protocol. Meanwhile, Bluetooth and BLE have the correct range for communication and light payload. BLE is energy efficient as well, making it the perfect choice for our application.[2]

4.2 Security

See corresponding Annexe B about Security

4.3 Data to exchange

4.3.1 Uplink data

At first, it was planned to send some data uplink, the power and cadence, to display the evolution during the activity or for statistics. However, we decided to not implement it as we encountered some delays and implementing the upward link would have required some additional work. We could also question the need of using energy to frequently send data that would probably not be really understood by the user, and might not be used by the company. We then decided to focus on the downlink data

4.3.2 Downlink data

As we decided to send the data downlink, it was decided that the calculation of the required power would be lead by the

Heart rate

A first value we want to send down to the ESP is the heart rate, as it will be the main variable for the torque control. It is represented as an integer between 0 and 255, even though it is unlikely that any of this two value would be reached. This value wouldbe sent at a regular page, every 5 to 10 seconds

Level of Effort

The level of effort is represented by an integer between 0, 1 and 2, respectively meaning stroll mode, normal mode and sport mode. This value will define how high the heart rate is allowed to go. This value would be sent when the link is first established as well as every time it changes.

Additional data

Finally, additional data thought to be implemented such as age or weight, to adapt better the torque control. For now, it is not implemented on the application, but it could be communicated when the communication is first established, but also if the user change the value (for instance, another driver takes the wheel). Even though this variable could be interesting to be implement to have a better control over the response, it requires to have a model acting with those variables as well. As we might not have it, it would be needed to be tested with numerous people from different age and weight, but this goes beyond the scope of our current project.

Chapter 5

Torque control

5.1 Testbench overview

As we did not have the opportunity to work directly on a full tricycle, we instead had a test bench provided by Maillon Mobility. It is made up of multiple elements.

First, we have the potentiometer that allows us to control the level of effort on the pedaling part, following what the user wants. This potentiometer, for the project we are working on, has to be set through the software, so a work of characterization and adaptation has been done in this way. Then we have the motor, to provide the feedback to the user when driving the bike. Both forward and backward are possible (for reverse park) but we are only using forward direction. The GenePi Generator is indeed part of the feedback assistant provided to the user, with the possibility to set various levels of torque to apply, and it is very important for us, as we can have the rotation per minute, cadency, power, and other interesting data for the regulation part. Regarding this regulation part, we are only focusing on 20w to 150w power range of values, as the model we are working on is based on this, which is reasonable. The other important element is the Flipsky controller, that centralizes all the data of the testbench and makes all the actions. Finally, we have the battery, here to power the system. We will explain later how we decided to power the ESP32. All these elements are shown in the next figure, except for ESP32, which we added to the testbench to have the control law and the communication with the smartphone to be set.

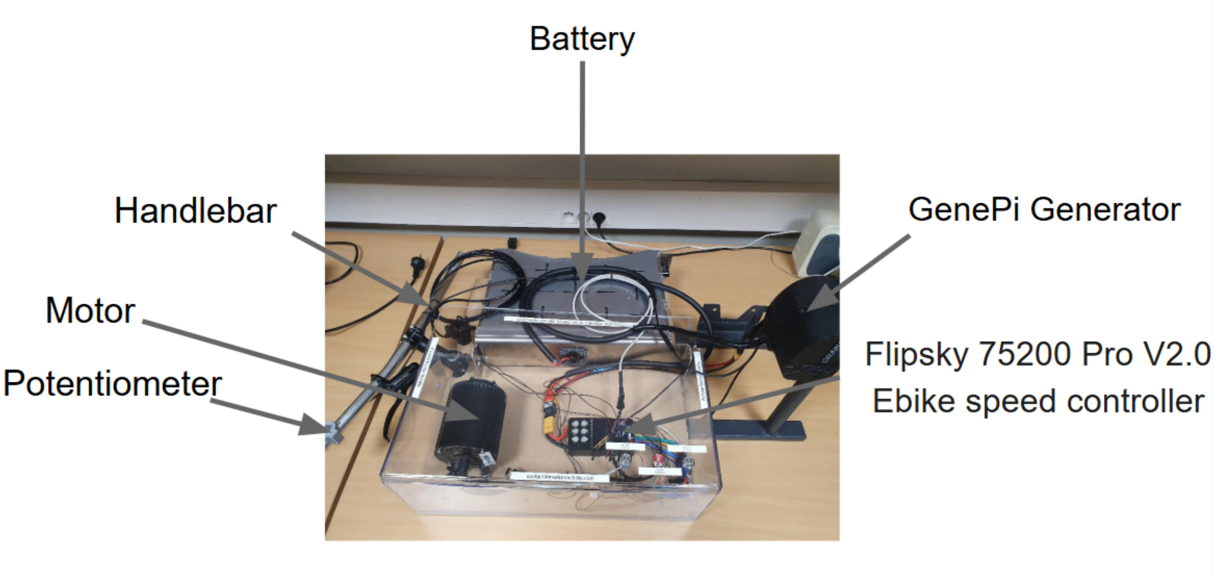

Figure 5.1: Key elements composing the testbench

In this part, we are not very focusing on the command law part, but more on the connectivity of the testbench with the smartphone, and the ability to catch all the information from the testbench, to then apply the control law. Actually, the goal is to connect to the flisky controller, with VESC programming, and then to the ESP32 to have connectivity with the smartphone.

To answer this, we had to characterize the testbench, within the data it can provide to us. The fact is that we spent a lot of time undertanding how the elements were working together, and it has been a lot of reverse engineering, as we did not have that much documentation : and how this testbench has been designed and how it works. We only had a short datasheet on the GenePi, with anything else more. However, we contacted Maillon Mobility and we had more information throughout the project.

5.2 Collecting data from the testbench

The main problematic is the following one : how to get the data from Flipsky speed controller and the GenePi Generator, in order to understand the system and then set the level of effort with control law regulation.

As mentioned before, the Flipsky controller manages all the testbench interactions, and that is why we focus first on this element.We explored different possibilities to achieve this connection with this controller. In fact, the most reliable way to connect to it, is via the VESC Tool software, that has been used to program it.

5.2.1 Mobile version of VESC : BLE protocol

We first tried it on my mobile, with the dedicated version, but we were not able to read and extract the data we received. In fact, it was not clear what we were receiving, and there were some elements in the frame that indicated errors. A screen shot is provided below.

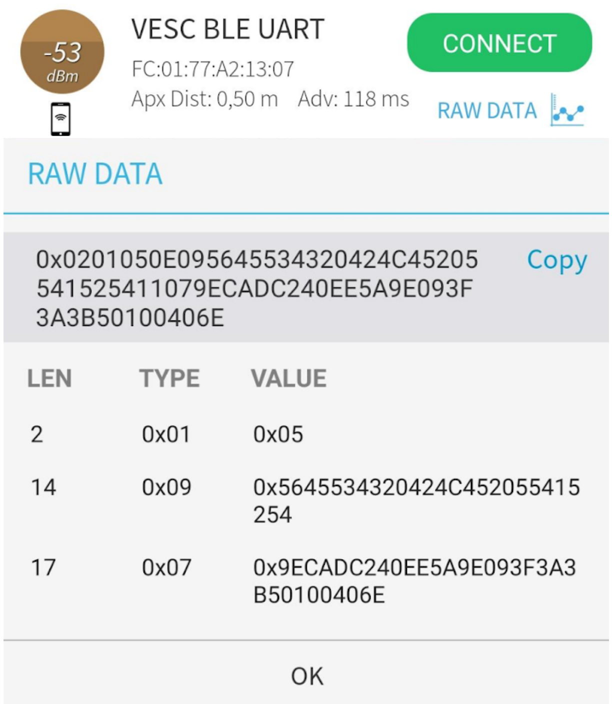

Figure 5.2: Data received on the VESC smartphone version

The flags (0x01) indicates that the BLE device is in Limited Discoverable Mode and does not support BR/EDR (Bluetooth Classic), which is a good sign. The complete local name (0x09) indicates the BLE device's name VESC BLE UART, and the Service Data (0x17) could be a custom service data related to the specific BLE service being implemented. In fact, the device we were trying to connect is the Flipsky, but we cannot process these data, and identify the one we were looking for. That's why we moved to the VESC tool software for computers.

5.2.2 Computer Version of VESC : CAN and UART protocols

So we moved to the computer version. On it, we were able to connect via an usb cable to the testbench. Yet, VESC Tool is a professional but complex tool to use when no any experience with it. We first made research on the internet to understand the multiple sections in it, that allows us to see and manage various kinds of data. The one useful for us is the power that the user provides to the pedaling part. It is possible to deduce it from the rpm and the torque.

We first tried to see if we could get information via the CAN bus. Indeed, even if the datasheet detailed the frame format and the data we were able to get, the GenePi itself was not set up for CAN frame format. This has been fixed later, during January. The only element set in CAN communication was the battery, but in fact, it was not the element relevant for what we were looking for, considering the control law to implement. So, at the time we tried to communicate with it, but we did not get any information.

Finally, we got through the UART communication, in which the motor was set. There, in the LISP programming interface of VESC, we were able to acquire the following data: the brake value (0.95-3.41V) applied, the information on if the Forward motor is used (0 or 3.41V) and finally the RPM value applied to GenePi (0-120). The last one is important for us, and will be useful for the next part. However, we were not able to catch the torque level. The three elements we were able to catch were physically linked to a pin on the flispky controller, within two ADC for the brake and forward motor detection, and a direct float value for the rpm value. The figure below show this data in the VESC tool.

	Binding	Value
1	adc2	3.412167
2	adc1	3.408834
3	rpm	69.306259

Figure 5.3: Data from VESC SOftware

On this part, we faced multiple issues. First with the VESC tool version, that was not the same version as the same used for the development of the testbench. In fact, once this problem was identified, Maillon Mobility provided us the right version, as well as the configuration files in order to use it properly. At a time, when trying to connect to the GenePi, and due to the wrong version used (using 6.05 instead of 6.02), the feedback from the motor that contributes to the work of the GenePi has been suspended. Hopefully, Paulin Fabre from Maillon came to INSA to fix this problem. A second time, he came to INSA, to make the GenePi able to work with CAN bus and so we were able to have a better view on how to connect the ESP32 to the controller and with the GenPi, and we would like to thank him for his time and contributions during this process. Even if there were some problems, we were able to get the RPM value from the GenePi, which would be useful for the next part.

5.2.3 Connecting the ESP32 to the testbench

Once we had more information on the working of the testbench itself, we had a better view on how it would be possible to include the ESP32. The first step was to define an architecture to work one with the analyze of data exchanges, and formant, as well as the power of the ESP32.

The architecture we chose is shown in the picture below. The data and the control law will be processed at the ESP32 level. First because it is the intermediate between the smartphone/watch and the testbench, but also because it is the simpler way to do it. In fact, we could have processed the control law in the controller directly, but it would have been more complex, because of the specific language it uses, and still, we would have needed the ESP32 to have the heart level. With this architecture, we can compute the data directly on the ESP32. For the final implementation, we can even consider getting the RPM value from the GenePi, which will simplify the model. With CAN communication, we can have bidirectional communication and ensure reactivity and reliability. There are three native UART port in the ESP32 but, the GenePi works only with CAN. In order to make the ESP32 communicate in CAN, we added a CAN grove module that is connected via the ESP32 with a serial port. The module then sends and receives CAN frames within the CANH and CANL ports from both Flipsky controller and GenePi Generator. Besides these communication aspects, we will power the ESP32 directly from the Controller, with both 5V and GND wire added by Maillon Mobility.

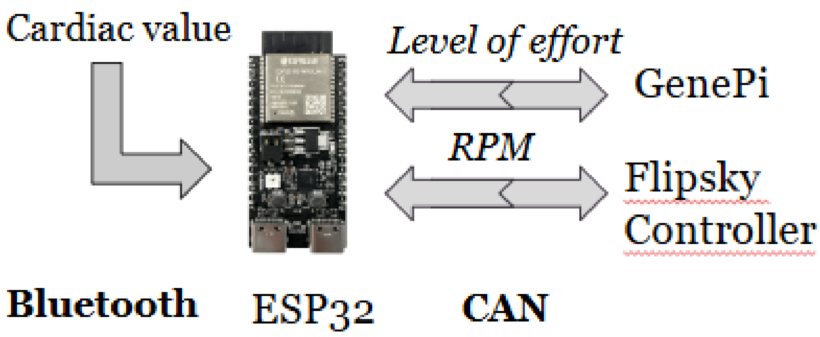

Figure 5.4: System architecture

One of the challenges of this architecture, and of our project indeed, is to connect all our work properly, and especially here, to respond to strong time constraints. The data incoming from the smartphone via BLE will be a challenge to take into account and make it work.

Considering the potentiometer, that was initially here to change the level of effort manually, we have two possibilities to replace it and integrate it in the control law. First, by using a digital voltage reference, that regulates the level of effort to apply with a continuous set of values. To do so, a filter has to be made to remove the perturbation and smooth the signal. We will try this option. The other one, is to set the level of effort directly via the CAN bus. To do, we can have among five level of effort, but it brings hysteresis and non-linearities. We will test the two solutions, because we are able to do it, and select the best one.

The control law will be explained in the next section. At this time, the objective is to communicate with the GenePi. To do, we programmed the ESP32 to first send a different level of effort. In order to do this, we send various values on the 0x300 address to set this level of torque to apply.

At this time, we are still working on the part and on the integration of the control law, and it is not finished and functional yet. After the previous communication working, and the control law implement in C++, we will probably face real-time issues and have to correct it also.

5.3 Control law

Heart rate dynamic model

Modeling the heart rate dynamic is essential for designing a control law for our application. It is a tricky system to model as the dynamic depends on age, training, and other factors.

Hence the best approach would be to have a rough idea of how the model behave, and estimate its parameters on the mock-up by measuring all the relevant data (heart rate, power, cadence, etc.). Unfortunately, the mock-up is not made in a way that allows that (it actually lacks pedals, and a saddle). We choose not to take VO2 into account as according to [**?**], it is too correlated with

Fortunately for us, S. Mohamad et al [6] have done a fabulous work by estimating a model of the heart rate dynamic accordingly tho the power output of an individual, we can use to train an IA model, or design and verify our control law onto.

The heart dynamic is quite difficult to model as it has a non-linear behavior. The approximation done in [6] proposes a Takagi-Sugeno model for input between 20 and 150 watts.

Takagi-Sugeno models are a special type off nonlinear model that actually are fuzzy model, this means that is a succession of linear models weighted by a fuzzy set (see eq: 5.1), hence the nonlinearity. But it has really interesting properties in terms of stability analysis, and control with parallel distributed compensation law (PDC for short). It also is a universal approximation for every non-linear model as Kevin Guelton has mentioned it in his HAL paper.[3]. The structure of the model is the following :

$$\begin{cases} x(t+1) = \sum_{i=1}^{r} h_i(z(t))\left(A_i x(t) + B_i u(t)\right) \\ y(t) = \sum_{i=1}^{r} h_i(z(t))\left(C_i x(t) + D_i u(t)\right) \end{cases} \tag{5.1}$$

with the following conditions :

$$\begin{aligned} h_i(z(t)) &\geq 0 \text{ and} \\ \sum_{i=1}^{r} h_i(z(t)) &= 1 \end{aligned}$$

The model identified in S.Mohamad et al. [6] has the following expression :

$$\begin{cases} x(t+1) = Ax(t) + \sum_{i=1}^{r} h_i(z(t)) B_i u(t) \\ y(t) = Cx(t) \end{cases} \tag{5.2}$$

We see the $r = 2$, so their is only to linear membership models, and the matrix A, and C are the same for both. Hence in the following part, when there will be A_1 or A_2, they will effectively be the same matrices in our case, same for C and D (which is all zeros in our case as well).

We were able to simulated it in python for simulation purposes as you can see in figure 5.5, which is actually a step response of the model. We also see that it is stable as both membership systems are stable individually [3].

5.4 Control law design

Notations : for simplicity the following notation will be adopted : $Y_z = \sum_{i=1}^{r} h_i(z(t))Y_i$, $Y_{z+} = \sum_{i=1}^{r} h_i(z(t+1))Y_i$, with Y_i represent any matrix/vector. As well (*) in a symmetric matrix denote the transpose element in the symmetric position.

5.4.1 State feedback stabilization

This part aimed at ensuring input to state stability. This means that the internal states are bounded and that a small variation on the input imply a small variation of the states.

As described in Guerra's et al. paper [4], we can design a state feedback (see eq.5.3), that guarantee global asymptotic stability (GAS), if the linear matrix inequalities (LMI) shown in equation 5.6 are

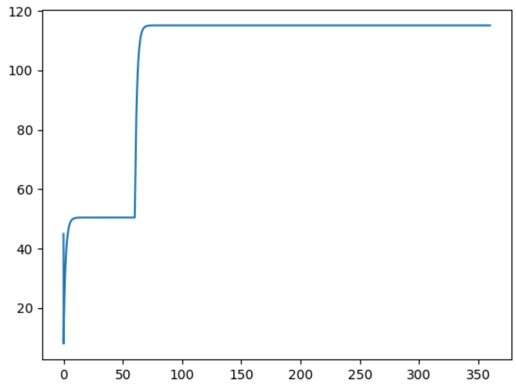

Figure 5.5: Step response of the HR model (100W to 150W), in open loop

satisfied by an existing solution. Note that we only check if the LMI are satisfied here, and do not try to find an optimal solution by minimizing a criteria.

In equation 5.3, the P_z^{-1} matrix actually is a Lyapunov matrix of the system. And the lyapunov function used in this case is a non-quadratic one, described in equation 5.4.

$$\begin{aligned} u(k) &= -F_z P_z^{-1} \\ \text{Hence} \quad x(t+1) &= (A - B_z F_z P_z^{-1})x(t) \end{aligned} \tag{5.3}$$

$$\begin{aligned} V(x(t)) &= x^T(t) \cdot P_z^{-1} \cdot x(t) \\ \Delta V(x(t)) &= V(x(t+1)) - V(x(t)) \\ &= x^T(t+1) \cdot P_z^{-1} \cdot x(t+1) - x^T(t) \cdot P_z^{-1} \cdot x(t) \\ &= x^T(t)((A - B_z F_z P_z^{-1})^T \cdot P_{z+}^{-1} \cdot (A - B_z F_z P_z^{-1}) - P_z^{-1})x(t) \end{aligned} \tag{5.4}$$

The GAS property is guaranteed by the Lyapunov conditions written in equation 5.5.

$$\begin{aligned} V(x(t) = 0) &= 0 \\ V(x(t)) &> 0 \\ \Delta V(x(t)) &< 0 \end{aligned} \tag{5.5}$$

Hence we get from our Lyapunov function (see eq.5.4) :

$$\begin{aligned} &\Leftrightarrow \; x^T(t)((A - B_z F_z P_z^{-1})^T \cdot P_{z+}^{-1} \cdot (A - B_z F_z P_z^{-1}) - P_z^{-1})x(t) < 0 \\ &\Rightarrow \; (A - B_z F_z P_z^{-1})^T \cdot P_{z+}^{-1} \cdot (A - B_z F_z P_z^{-1}) - P_z^{-1} < 0 \;\; \text{(i.e. NSD)} \\ &\Leftrightarrow \; \begin{bmatrix} P_z & (*) \\ AP_z - B_z F_z & P_{z+} \end{bmatrix} > 0 \text{ (i.e PSD) This comes from Schur's complement} \\ &\Leftrightarrow \; \begin{bmatrix} P_i & (*) \\ AP_i - B_i F_i & P_j \end{bmatrix} > 0 \text{ for each } i, j \in \{1, .., r\} \end{aligned} \tag{5.6}$$

See a more detailed proof in Gruerra et al. paper [4]. Also this condition implies that P_i is PSD.

Running an LMI solver with those conditions allow us to find both the P_z matrices, and the F_z vectors. Note that for our project we use two state feedback vector for each sub-systems as it yields better conditioned Lyapunov matrices, and as our microcontroller is powerful enough to afford the extra computations.

5.4.2 State space augmentation for integral effect

In order to acheived a null steady state error, i.e. the difference between the output of the system and the command. We add an integrator on the output.

To do this we extend the state space representation as shown in equation 5.7, by adding a state x_I.

$$\begin{cases} \begin{bmatrix} x(t+1) \\ x_I(t+1) \end{bmatrix} & = \begin{bmatrix} A & 0 \\ -C & I \end{bmatrix} \cdot \begin{bmatrix} x(t) \\ x_I(t) \end{bmatrix} + \begin{bmatrix} B_z \\ 0 \end{bmatrix} \cdot u(t) + \begin{bmatrix} 0 \\ I \end{bmatrix} y_c \\ y(t) & = \begin{bmatrix} C & 0 \end{bmatrix} \cdot \begin{bmatrix} x(t) \\ x_I(t) \end{bmatrix} \end{cases} \tag{5.7}$$

Hence we get :

$$\begin{aligned} x_I(t+1) &= x_I(t) - Cx(t) + y_c \\ &= x_I(t) - y(t) + y_c \end{aligned}$$

In order to get proper Lyapunov matrices and state feedback vectors, we had to run the LMI solver on this new state space representation. Note that the 4^{th} element of F_z are now the integrator gain. We also set the pre-command gain to $1/4$ on the 4^{th} element of W after finding out that they were a static gain of 4 on the controlled system output.

From now on, we will use the following notation when speaking about the augmented state space with the integral effect :

$$A_{aug} = \begin{bmatrix} A & 0 \\ -C & I \end{bmatrix} \quad \in M_{4\times 4}(\mathbb{R}) \qquad W = \begin{bmatrix} 0 \\ I \end{bmatrix} \quad \in M_{4\times 1}(\mathbb{R})$$

$$B_{z\ aug} = \begin{bmatrix} B_z \\ 0 \end{bmatrix} \quad \in M_{4\times 1}(\mathbb{R}) \qquad C_{aug} = \begin{bmatrix} C & 0 \end{bmatrix} \quad \in M_{1\times 4}(\mathbb{R})$$

$$x_{aug}(t) = \begin{bmatrix} x(t) \\ x_I(t) \end{bmatrix} \quad \in M_{4\times 1}(\mathbb{R})$$

5.4.3 State observer

As you might have noticed, the control law we have yet is not implementable as we do not know $x_{aug}(t)$ for every value of t. Then we need a state observer to estimate such a vector, and compute our $u(t)$, the whole scheme is described in figure 5.6. Note that L is the observer gain. We will get into the detail of the design bellow.

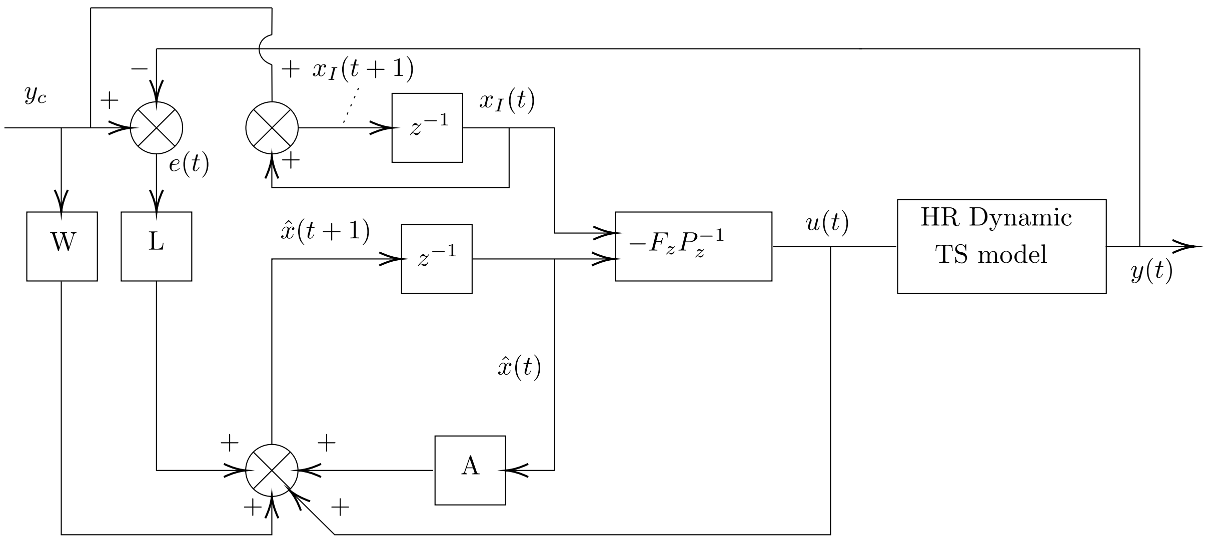

Figure 5.6: Block schematic of the control scheme with the state observer

Separation principle and observability

The separation principle said that for linear system, we can design the state feedback and the observer separately. Yoneyama et al.[8] show that it is still applicable to TS models. This way, we were able to design an observer feedback vector for each subsystem while maintaining stability.

We get another problem, A_{aug} with C_{aug} is not observable, more accurately, $x_I(t)$ is not observable. Thus for the pole placement, we do not change its pole. For the other ones, we choose them to be both stable and faster, i.e. we decrease their module to make them closer to zero while still in the unit circle, which is trivial as all our poles were already stable.

State observer expression

To simulate the observer dynamic, and check out stability, response time, and other relevant parameters, we augmented the state space once again as shown bellow :

We get the following state feedback :

$$\begin{cases} x_{aug}(t+1) & = A_{aug} \cdot x_{aug}(t) + B_{z\ aug} \cdot u(t) \\ y(t) & = C_{aug} \cdot x(t) \end{cases} \tag{5.8}$$

With our usual state feedback in equation 5.8, such as $u(t) = -F_{z\ aug} \cdot P_{z\ aug}^{-1} \cdot \widehat{x}_{aug}(t)$, with $\widehat{x}_{aug}(t)$, the estimation of $x_{aug}(t)$, we get equation 5.9 :

$$\begin{cases} x_{aug}(t+1) & = A_{aug} \cdot x_{aug}(t) - B_{z\ aug} \cdot F_{z\ aug} \cdot P_{z\ aug}^{-1} \cdot \widehat{x}_{aug}(t) + Wy_c \\ y(t) & = C_{aug} \cdot x(t) \end{cases} \tag{5.9}$$

When it comes to estimating $x_{aug}(t)$, we will use the Luenberger observer, also know as the identity observer, as shown in equation 5.10, it actually simulate the dynamic of the system with the model, and adjust itself with the error between the estimated output and the measured one.

$$\begin{aligned} \widehat{x}_{aug}(t+1) &= A_{aug} \cdot \widehat{x}_{aug}(t) - B_{z\ aug} \cdot F_{z\ aug} \cdot P_{z\ aug}^{-1} \cdot \widehat{x}_{aug}(t) + L(y(t) - \widehat{y}(t)) + Wy_c \\ &= (A_{aug} - L \cdot C_{aug} - B_{z\ aug} \cdot F_{z\ aug} \cdot P_{z\ aug}^{-1}) \cdot \widehat{x}_{aug}(t) + Ly(t) + Wy_c \\ \text{as} \quad \widehat{y}(t) &= L \cdot C_{aug} \cdot \widehat{x}_{aug}(t) \end{aligned} \tag{5.10}$$

Hence, if we extend the system to take into account the observer dynamic we get the expression 5.11.

$$\begin{cases} \begin{bmatrix} x_{aug}(t+1) \\ \widehat{x}_{aug}(t+1) \end{bmatrix} = \begin{bmatrix} A & -B_{z\ aug} \cdot F_{z\ aug} \cdot P_{z\ aug}^{-1} \\ L \cdot C_{aug} & A_{aug} - L \cdot C_{aug} - B_{z\ aug} \cdot F_{z\ aug} \cdot P_{z\ aug}^{-1} \end{bmatrix} \cdot \begin{bmatrix} x_{aug}(t) \\ \widehat{x}_{aug}(t) \end{bmatrix} + \begin{bmatrix} W \\ W \end{bmatrix} \cdot y_c \\ y(t) = C_{aug} \cdot x(t) \end{cases} \tag{5.11}$$

Design of the state observer gain L

In order to guarantee that the system will still be stable with the state observer, and in respect to the separation principle, we can place the poles of $A - L \cdot C_{aug}$ to the poles we want, in the same fashion as previously mentioned. We can do this as this matrix the dynamic of the error $e(t) = x_{aug}(t) - \widehat{x}_{aug}(t)$, thus if it is stable, $\lim_{t\to+\infty} e(t) = 0$, and the estimation of the states follow th real state values faster then they change, i.e. faster then the system dynamic.

Results

From those design steps, we were able to design a functioning and implementable state feedback. It would have been great to enhance the controller property by introducing an objective function in the LMI conditions, in order to decrease response time for example.

As you can see in figure 5.7, we simulate a step response between 50 to 150 bpm, and the output (x_0) follow the step satisfactorily, with a response time ($t_{90\%}$) of 20 seconds or so.

5.4.4 Embedded torque control loop

The state feedback to control the heart rate work very well, however the input $u(t)$ is a power, and we can only control the torque on Maillon mobility tricycle. Hence we design another control law that take the command of the other model ($u(t)$), and apply a PI control law to it to command the power produce by the cyclist depending on its cycling speed (in rpm).

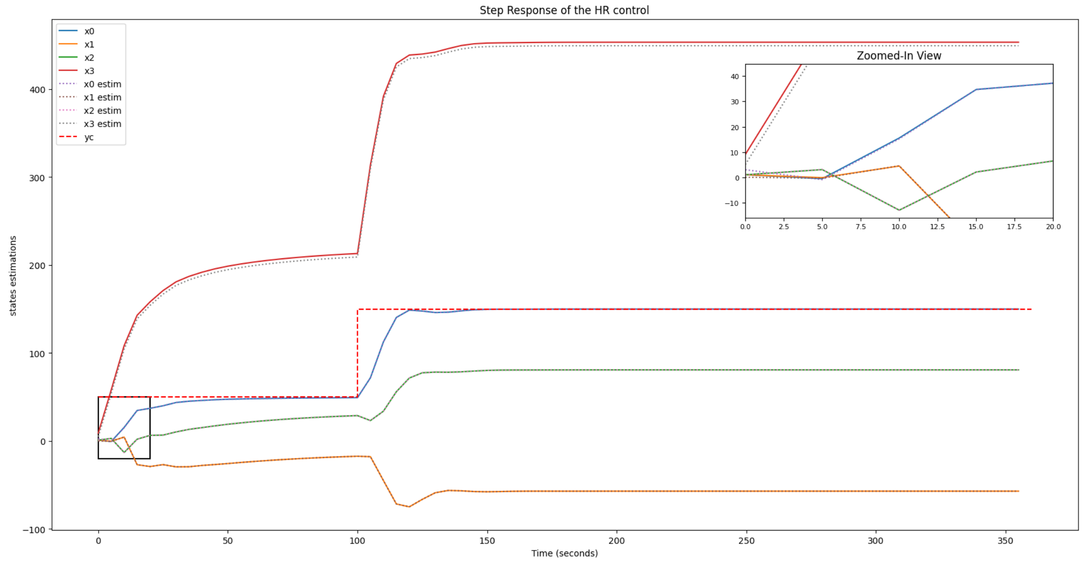

Figure 5.7: Step response (from 50 to 150 bpm) of the system controlled, augmented with the state observer

$$P = T \cdot \omega \quad and \quad \omega = \frac{2\pi \cdot rpm}{60} \tag{5.12}$$

We define $e(t) = P_c - P_m$, with P_c the wanted power input, and P_m the measured power. Thanks to the relation in eq.5.12, and with K_p, K_i and T_s respectively the proportional gain, the integrator gain, and the sample time ($T_s = 0.1s$), we can write the following expressions :

$$\begin{aligned} e(t) &= P_c - T(t) \cdot \omega(t) \\ I(t) &= I(t-1) + K_i \cdot e(t) \cdot T_s \\ T(t+1) &= K_p \cdot e(t) + I(t) \end{aligned} \tag{5.13}$$

With well chosen gains and with a varying cycling speed, we can get an effective control of the power via a variation of the torque as shown in figure 5.8

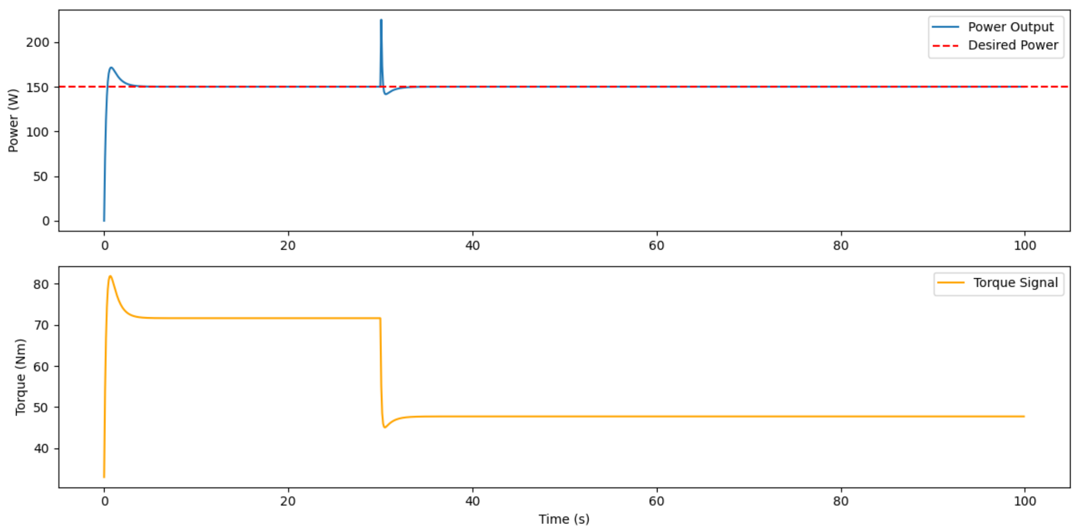

Figure 5.8: Torque regulation loop response to a variation in the cycling speed (from 10 to 30 rpm

As you can see in figure 5.8, the response is a bit rought, one good objective wound be to limit the derivative of the Torque control, hence reduce its slope, such as the user would not have a sharp change in the Torque he needs to apply.

5.4.5 Reinforcement learning and other enhancement

The control law we used works well in simulation, however it comes from a strong hypothesis, that the model identified by Mohammad et at. [6] fits for a wide variety of people. In fact people heart rate dynamic depends on a lots of different parameters, mainly age, health, and training. What we could do for improving performances is identify the system online with a identification sequence (sort of calibration), and then define a user class, such as the people that connect to the bike can share their profile and have the best regulation possible. Also, this implies to do the design of the feedback vectors on the micro-controller or the phone.

In this regard reinforcement learning and more specifically deep differential gradient policy with continuous action space network (see this paper from Lillicrap et al. [5]) could be used to estimate the model, and converge to a control law by learning iteratively over the action and output domain. This comes with a bunch of drawbacks however, as it is very computationally expensive, and we would likely have switched to a single board computer for implementing the control law. It also needs to explore the action and output domain, which mean that the torque would have to vary over its whole range, and same for the cycling speed, in order to get a good image of the output (i.e. the heart rate, and other states). This might not even converge, and leads us to the final drawback, which it that the system and its control law is a black box, and we have no guarantee upon what is going on for a given state of the system.

Conclusion

Through this project, we were able to work together to achieve the creation of a system aiming to adapt the cadence of an electric tricycle depending on the heart rate of the driver. This application helped us to apply all the knowledge you accumulated through our years at INSA Toulouse, in programming, automatic, electronic and project management.

We also have some further improvements we want to make in order to make it a proper application fit to be deploy on such tricycles. For example, the control law still needs some improvement for tuning the response for example, such as the cyclist do not feel the change of torque directly, but more gradually over time. Also, the identification of the model as explained in the section 5.4.5, would be a great addition.

This project appears to be more of an accessory than a practical solution. At its core, it builds upon an already expensive electric tricycle. The chain-less transmission, which relies on a dynamo and an electric motor, seems less efficient than a traditional chain drive paired with electric assistance. While the tricycle's unique architecture with two front wheels might allow for sharper turns at higher speeds, yet, its benefits are marginal in urban environments where frequent stops at intersections are unavoidable. Additionally, this design increases costs, poses safety risks, and can be challenging to handle for new users.

While integrating heart rate monitoring and phone connectivity offers educational value, these features ultimately come across as gimmicks. They provide limited real-world utility and position the product as more of a toy than a genuinely practical or innovative solution.

Bibliography

[1] Wi-fi direct | wi-fi alliance.

[2] Mohammad Afaneh. Bluetooth low energy (ble): A complete guide.

[3] Kevin Guelton. Modèles Flous de Type Takagi-Sugeno : des Origines à la Problématique Actuelle de leur Commande à Base de Signaux Echantillonnés. In Cépaduès-Editions, editor, *LFA 2020*, Sète, France, 2020.

[4] Thierry Marie Guerra and Laurent Vermeiren. Lmi-based relaxed nonquadratic stabilization conditions for nonlinear systems in the takagi–sugeno's form. *Automatica*, 40(5):823–829, 2004.

[5] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning, 2019.

[6] Sami MOHAMMAD, Thierry Marie GUERRA, Jean Marie GROBOIS, and Bernard HECQUET. Heart rate control during cycling exercise using takagi-sugeno models. *IFAC Proceedings Volumes*, 44(1):12783–12788, 2011. 18th IFAC World Congress.

[7] Sara Santos. Esp32 wireless communication protocols.

[8] Jun Yoneyama, Masahiro Nishikawa, Hitoshi Katayama, and Akira Ichikawa. Output stabilization of takagi–sugeno fuzzy systems. *Fuzzy Sets and Systems*, 111(2):253–266, 2000.

.1 Annex A - Energy section

This annex includes our report about the energy consumption in our project.

Introduction

Innovation is an important part of our curriculum as future engineers specialized in the world of Internet of Things. In our present and future, we are and will be led to conceive systems. Those might be of any size, of any purpose. But a constant constraint that will be met in any IoT system is the energetic one. Connected objects often are modest in energy source, and it is important to ask ourselves how we want to power and save energy to make it as efficient as possible. In this report, you will find our reflection regarding the energy side of our last year Innovative Project.

Presentation of our System

Our Innovative project concerns an electric tricycle created for suburban transportation. Our goal is to implement a device that would communicate the cardiac rhythm and adapt the effort the driver needs to put to have a comfortable and safe ride. Our system is composed by three main components:

- A smart watch that will measure the vital constants of the driver
- The phone that will display and manage the communication
- An ESP32 that will transmit the change of cadence to the motor

Powering the System

The ESP32 that we are using for the command law, is directly connected to the flisky controller of the tricycle and the battery, and so does not need external power input. In that way, there is no specific optimization to do on this part. The second part of the system is the watch and the smartphone in which it is linked to. On this side, small batteries are already designed and embedded within the watch and smartphone devices. Hence, we are more thinking in terms of consuming the minimum power needed, which is the element we are focusing on in the next parts of this report.

Ambient energy recovery from the tricycle

One of the huge challenges on this is ensuring that the energy spent to catch this wasted energy is negligible compared to the energy obtained. This is not necessarily the case, especially because there is only a small amount of energy that we would be able to catch. The other question is how to use it in our system. In fact, we can be sure that the place available on the tricycle and the battery on it allows us to not have power issues. In fact, there are not so many possibilities in our project. The only one we can see is a mechanical energy restore. Maybe an optimization is done on the GenePi Generator, but we do not really have this information within the documentation.

Saving Energy

Communication between devices

A first point of the consumption that might be energy intensive is the wireless communication between the components of our system. As we need real time response from the watch to the motor to ensure the proper use of the effort adjustment, we need to send data regularly at a rapid pace.
To keep the efficiency while reducing the energy used for the communication, multiple solutions can be deployed.

Reduction of the size of the exchanged data

A first step can be the reduction of the payload, to make sure that only the necessary information is transferred. As we are using floats or integers for most of our values, the size of the payload remains light. This applies if the payload is treated as such and not as string, in which case it would be quite heavy.
As we are currently using the Bluetooth Low Energy Protocol, we are limited when it comes to the size of the header. We are currently using the Generic Attribute Profile, or GATT, that is working using Services and Attributes to communicate the data (after connection)

Reduction of the number of exchange

Another way of reducing the energy put into the exchange is to reduce the frequency of the exchange itself.

A first way to reduce the energy could be to ensure that we only send data that will be used. Some of the information that we want to exchange will be used to calculate. A problem that we had was to choose where some of the math would be done. For instance, do we want to calculate the new cadence on the phone by getting the cadence and power from the esp32 or do we want to calculate on the esp32 by getting the cardiac rhythm and the chosen level of effort from the phone. This choice might change the energy for the communication, but it is hard to estimate without proper testing.

A second way to approach this solution is to send the data only when a big enough difference is calculated between the previous communicated measure and the new one. However, the lack of messages from a device could be from an unchanged value or a problem in the communication: this last one could not be known with the previous method. A way to address this issue could be to send periodic messages to let the other device know that the communication is still on. However, this might require clock synchronisation, and it is out of the subject of this report about energy.

Reduction of the number of exchange

Besides the elements cited before, related to the format of their data exchanged, we can also consider code optimizations, with for example, the use of interruption in the code, rather than delays that may cause periodic energy losses, and the possibility of failures. Using interruptions allows better management of the system, even if the implementation is likely more complex and time consuming when developing the code.

Conclusion

Our project is highly dependent on existing systems: the tricycle, the watch and the phone. For that reason, the input power of our system is not a limitation. However, It remains interesting to ask ourselves how we could reduce the consumption of the energy, as we have seen, in the communication and redaction of the code.

.2 Annex B - Security section

This annex includes our report about the security issues and solutions in our project.

Introduction

In simple or complex systems, any electronic device and digital system could be exposed to security issues, by malevolent attackers or simply by some untreated systemic issues. During the development of our Innovative Project, we have to question ourselves about potential weaknesses of our implementation. Through this report, we will explore our system, divided into sections, and question ourselves about security topics that could be associated, and which approach we should take to solve them.

Our report will be divided into two main sections: the bluetooth communication between the device and the confidentiality of the data that our system has or might implement. More specifically on BLE (Bluetooth Low Energy) that is suitable for our application with low power consumption objectives.

On the following figure, we can see an overview of the possible threats that may occur on the BLE protocol. We will not be able to cover the entire threats but we can have a global point of view of the risks that our system could face.

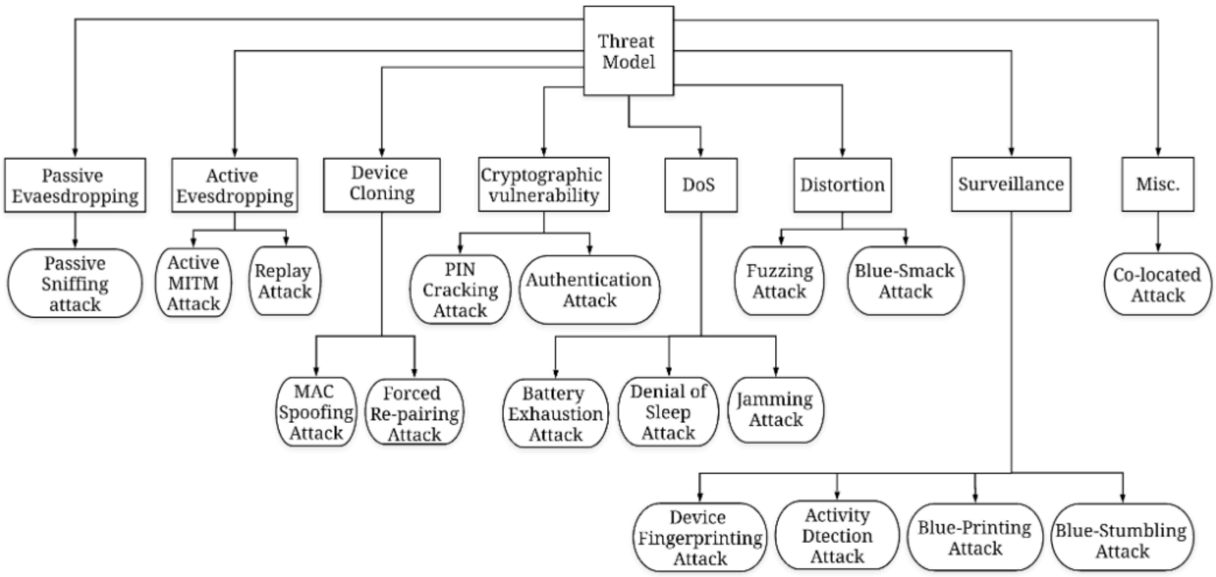

Figure 1 : BLE threat model on the basis of the attack domain [1]

Bluetooth connection between devices

Our project relies on a phone to support an application interfacing with the user and manage the communication between the devices. Although it allows us to create a complex application and facilitate the implementation of further communication, such as a potential database, it restricts the protocols that we can use for communication to Wi-fi, Bluetooth and Cellular (2G, 3G, 4G or 5G)

Due to the short range of our application, the energetic needs and the facility of development, our choice went to Bluetooth, more specifically Bluetooth Low Energy (BLE). BLE implements multiple security procedures, with levels of authentication and encryption.

Connection between the phone and the tricycle

The first connection that we need to securize is the one that the phone will have with the tricycle, more precisely with an ESP32 that will be embedded on the vehicle, that communicates directly with the motor. This development board integrates a Bluetooth module.

The main issue that we might encounter for this communication is the identification of the user: we want to make sure that the BLE connection is made with the associated vehicle, and not by a nearby one. We could imagine that a potential attacker could try to connect the vehicle we drive, and control some aspect of the driving regardless of what we wish for.

Currently, the ESP allows only one Bluetooth connection. We could ensure that the connection is the appropriate one by associating an identification number to the displayed bluetooth name. In case this one is not found, it means that it is currently associated with another device, and the user should be able to reset the connection.

Finally, if the user does not want to use the cardiac rhythm functionality, they should be able to turn it off. The system should be able to work without any information from the phone, without having issues or errors.

Key based security

A way to implement the security regarding the proper identification is the use of asymmetrical encryption, using an identification number and a pair of keys to ensure the identity of the phone/device pair. The public key would be available on a web server, while the private key would be stored directly on the tricycle. The key exchange would involve 3 actors: a web server, a tricycle and a phone:

1. The phone asks the public key assigned to this specific tricycle to the web server
2. The web server gives the key
3. The Phone encrypts a challenge (random number) using the public key
4. This encrypted challenge is sent to the tricycle, which deciphers it through its own private key.
5. The challenge is returned to the phone. If it matches the original one, the connection can be established, as both devices are recognized.

However, this implementation still have some issues:

- The generation of the private and public key needs to be defined before use.
- The tricycle (ESP32) needs to be able to store the private key. Currently, there is no persistence of the data when the system is turned off, making it tricky to have a private key. Otherwise, it might need to be generated at activation and be able to communicate with the website to give it. An internet connection would then be required.
- Each tricycle needs to have a proper identification to be able to get the matching public key on the website.

Connection between the phone and the watch

To get the information of the heart rate, we decided to use a smart watch. The smart watch offers an API that allows to get the data gathered by the watch. We are currently using a Garmin Watch that requires a connection by entering the username and password.

We are still not sure if this account would be the one from the user to which the watch belongs, or the company's account to which a paid API key might be applied. In both cases, we need to ensure that the credentials will be kept confidential and in case they need to be stored, will be encrypted.

Another solution we had to get the heart rate was the direct use of a sensor that would get the rhythm. In that case, a system would embed the sensor, and the communication would be managed by us. In that case, we would also need to ensure the availability, confidentiality and integrity of the transferred data by using a reliable protocol. In that case, BLE would remains the best option.

Machine learning and Blockchain to improve security

We can expect to improve bluetooth communication with the recent development and applications of machine learning. Currently, there is an open research opportunity to protect the BLE mesh network from zero-day vulnerability, DoS attacks, spoofing attacks using intrusion detection systems and intrusion prevention systems, and watchdogs.Introducing new aspects of machine learning algorithms is a promising area of research for enhancing the security and privacy of IoT devices.In the wrong hands, the power of AI can be exploited. AI is frequently used by attackers to uncover and exploit flaws far faster than developers can repair them.[1]

BLE enabled smart wearable and IoT devices have been seamlessly integrated into our everyday life. So secure data management and robust access control of IoT devices are becoming very important. The idea of using a server to connect with an IoT device rather than connecting with an individual device directly may enhance device management and users' privacy significantly. This server will store users preferences, thus preventing IoT devices from accessing personal information. But if the network/server is breached, then every device connected to that network will be compromised. So there is a research opportunity to use decentralized block-chain technology to secure IoT devices connected in a BLE mesh network. [1]

Data confidentiality

Information relative to the calculation

A functionality we did not implement is the ability for the user to enter their age and weight to adapt with more precision the model. If done, as this information is personal, we should ensure confidentiality in case we decide to keep it stored so the user does not have to enter them at each connection. In our case, as we developed a website, this information could be kept as a cookie. Using Json Web Tokens (JWT), we could properly store and even encrypt the data so it might not be gathered by other agents.

Conclusion

We saw that a system as simple as the effort on a tricycle adapted from the heart rate has many security threats and risks. However, due to the lack of time, we were not able to implement any proper security prevention, outside of the use of the BLE protocol, embedding some solution to keep the integrity and availability if our data.

Sources

[1] 1. Barua, M. A. Al Alamin, M. S. Hossain and E. Hossain, "Security and Privacy Threats for Bluetooth Low Energy in IoT and Wearable Devices: A Comprehensive Survey," in *IEEE Open Journal of the Communications Society*, vol. 3, pp. 251-281, 2022, doi: 10.1109/OJCOMS.2022.3149732.