RACING DRIVING DATA VISUALIZATION

PAU MATAS ALBIOL

Thesis supervisor: PERE PAU VÁZQUEZ ALCOCER (Department of Computer Science)

Thesis co-supervisor: IMANOL MUÑOZ PANDIELLA (Department of Computer Science)

Degree: Bachelor's Degree in Data Science and Engineering

Thesis report

Facultat d'Informàtica de Barcelona (FIB)
Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona (ETSETB)
Facultat de Matemàtiques i Estadística (FME)

Universitat Politècnica de Catalunya (UPC) - BarcelonaTech

Aknowledgements

M'agradaria agrair a tothom qui ha fet possible aquest Treball de Final de Grau.

En primer lloc, vull agrair al grau de dades (GCED) per proporcionar-nos la formació necessària, tant a mi com als meus companys, perquè siguem capaços d'adaptar els nostres coneixements a qualsevol àmbit, com en el meu cas, els cotxes de competició.

En segon lloc, vull agrair a l'equip de BCN eMotorsport. Sense aquest grup d'inconscients, que dóna sang, suor i llàgrimes perquè el projecte segueixi endavant, tot això no tindria sentit. Vull agrair especialment les seccions de control i percepció; també al Rodas, al Biosca, al Puigde, al Víctor, al Martínez, al Kike, al Pareras, a la Cris, al Pereztol i al Bergas; per haver fet més amè el meu pas per l'equip.

Tampoc vull oblidar-me dels professors, tant en Pere-Pau com l'Imanol, per l'ajuda que m'han suposat i la paciència que han tingut durant la redacció d'aquest TFG. Agrair també, a en Javi i en Josep Ramón per la seva ajuda incondicional a l'equip.

Finalment, l'agraïment més important va per als familiars, els amics, els companys i per la Marina. Tots ells han estat les persones que han estat al meu costat, donant-me suport en els millors i pitjors moments, tant durant aquests quatre anys com durant aquest treball. Han estat estimant-me, animant-me i fent-me els dies una mica més feliços de forma desinteressada. A tots vosaltres, gràcies per tot el passat, present i futur.

English

Abstract

Formula Student teams not only need to build the best car to win competitions but also need to select and train their top drivers. In this thesis, we propose a visual analysis application for the data collected by the car during driving, aiming to evaluate and compare the drivers. This tool has resulted in an improvement in terms of user-friendliness, communication, and interpretability of the data. Thus, decisions regarding the selection of team drivers can now be done data-driven.

Keywords

Formula Student, Data visualization, Data science, Car telemetry, User interfaces, Data-driven decision processes

Català

Resum

En els equips de Fórmula Student, no només és necessari construir el millor cotxe per a guanyar competicions, sinó també trobar i entrenar els millors pilots. En aquesta tesi, es proposa una aplicació d'anàlisi visual per a les dades recollides durant la conducció del cotxe, amb l'objectiu d'avaluar i comparar els pilots. Aquesta eina ha suposat una millora significativa en termes de facilitat d'ús, comunicació i interpretació de les dades. Així, les decisions sobre la selecció dels pilots per a l'equip ara es poden prendre de manera guiada per les dades.

Paraules Clau

Formula Student, Visualització de dades, Ciència de dades, Telemetria del cotxe, Interfícies d'usuari, Processos de presa de decisions basats en dades

Español

Resumen

Los equipos de Fórmula Student no solo necesitan construir el mejor coche para ganar competiciones, sino que también necesitan encontrar y entrenar a sus mejores pilotos. En esta tesis se propone una aplicación de análisis visual para los datos recopilados por el coche durante la conducción con el fin de evaluar y comparar a los pilotos. Esta herramienta ha supuesto una mejora en términos de facilidad de uso, comunicación e interpretación de los datos. De esta forma, ahora las decisiones sobre la selección de pilotos del equipo pueden tomarse guiadas por los datos.

Palabras Clave

Formula Student, Visualización de datos, Ciencia de datos, Telemetría del coche, Interfaces de usuario, Procesos de toma de decisiones basados en datos

Contents

1 Introduction

This bachelor's thesis is carried out at the *Universitat Politècnica de Catalunya* (UPC) under the supervision of Pere-Pau Vázquez and Imanol Muñoz.

The purpose of this project is the creation of a visual analytics tool that facilitates the visual comparison and evaluation of different drivers of a Formula Student car.

In this section two main aspects of this bachelor's thesis will be discussed:

1. Where this project originates from and how the idea behind it is formed.
2. Which other similar tools are used in Formula Student and which functionalities provide to the users.

With the project introduced, its main goals will be detailed to give a general vision of what is expected for this document to be. Then, the technology used in this project will be explained and afterwards that the development of the thesis and the results will be presented. Finally, we will show the user study that we have committed and the final conclusions to which we have arrived, after finalizing the whole process.

1.1 Project Background

Six months before the deadline to submit the idea for this bachelor's thesis I joined BCN eMotorsport, the Formula Student (FS) team of the UPC. The team is managed by the Barcelona School of Telecommunications Engineering (ETSETB) and the Barcelona School of Industrial Engineering (ETSEIB). BCN eMotorsport annually recruit around 50 students from fields as telecommunications, industrial engineering, mathematics, physics, electronics, computer science or data science to fill their departments. In the team there are eight departments: Aerodynamics, Chassis & Composites, Vehicle Controls, Vehicle Dynamics, Perception, Electronics, Powertrain and Management; that work together to build a formula-type car with a budget of ~ 500K€ based on financial sponsorships and the value of material contributions. This season's team photography is shown in Figure 2

As hundreds of other worldwide student teams, BCN eMotorsport competes in FS. The FS competition is an annual motorsport challenge held worldwide where teams design and build a formula racing car to compete against each other. The teams compete in different nations

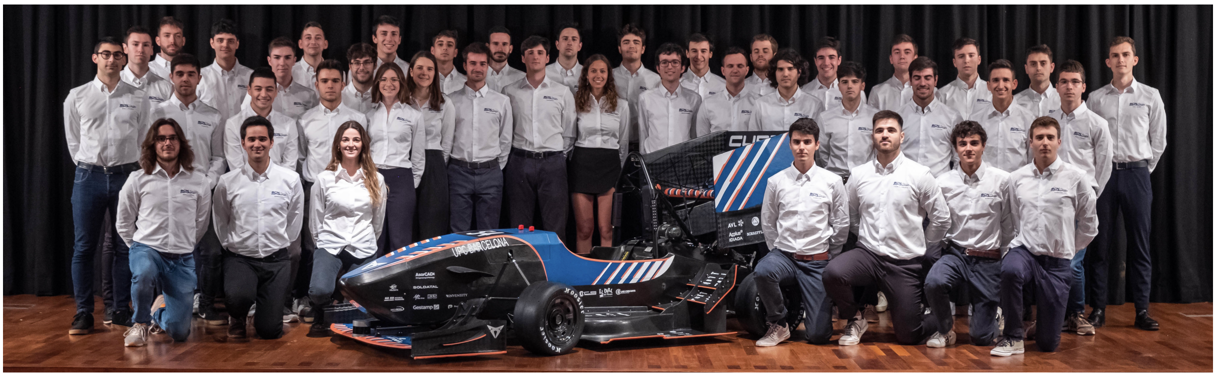


Figure 2: BCN eMotorsport 2023 team with Cat15x


Figure 3: *Technische Universität München*'s car racing at FSG 2019

during the summer and not all the national competitions have the same prestige. Formula Student Germany (FSG) -of which is the image of Figure 3- is for excellence the most important event of the year followed by Formula Student Austria and Formula Student East (held in Hungary). The more prestige a competition has, the more difficult the admission tests are, the more teams want to join it and are trying the exams, and since the number of slots is limited the more difficult is to compete in them. This year we will compete at FSG, Formula Student Spain (FSS) and Formula Student Italy.

Formula Student competitions also have three divisions: Electric Vehicles (EV), Combustion Vehicles (CV) and Driver-less Vehicles (DV). BCN eMotorsport's 2022-2023 car (Cat15x), shown in Figure 2, is an electric vehicle which is also able to race autonomously and therefore can compete in both EV and DV divisions. To better understand what a DV competition consists of, one of the team's best performances ever at FSG with Cat13e or the last year's car Cat14x at FSS are available in the respective links.

1.1.1 Definition of the problem

During the six months before starting this project and already being at the team, I was working on the perception department. Perception is in charge to detect the track limits for the DV part of the car, using the different sensors available on the car (cameras and LiDAR). At the department, I conducted research on the usage of neural networks on LiDAR data to detect and classify objects, working with two ETSETB professors. I was commited to continue this research in form of a bachelor's thesis when weeks before the deadline I found that properly evaluating the drivers' performace was very challenging with the tools we had. The perception pipeline was in a very advanced point and I thought that solving the problem of accurately evaluating performance could have a bigger impact on the team than keep working on the perception pipeline.

This big problem starts when, for EV events, as a team, you need a drivers line-up and since it is uncommon to have professional drivers studying at your university, some of the engineering students have to be drivers. Therefore, if you want to win competitions, it is necessary not only to build the best possible car but also to choose the best driver among the team members and train them to perform. The decision of selecting drivers and the improvements in their performance is achieved through hours of testing. However, if those hours are not supported by analysis, there could be some missing progress for the driver, and it could be impossible for the team to select the best drivers.

The situation of the team was that the analysis tool that was being used did not provide the necessary information. This caused the drivers draft to be based on instincts rather than data-driven, and the drivers evolution on performance to be practically non-existent. As far as my driving skills are pretty bad, I wanted to contribute my bit to make the team more competitive providing for a better analysis tool.

1.1.2 Previous environment

Up until now, the team has been using the GEMS' Data Analysis Software (hereafter referred to as GEMS) extensively. Initially, GEMS was employed for analyzing data related to car components such as tire pressure, cooling system flow rate, motor and battery power and energy, as well as aerodynamic data, among others.

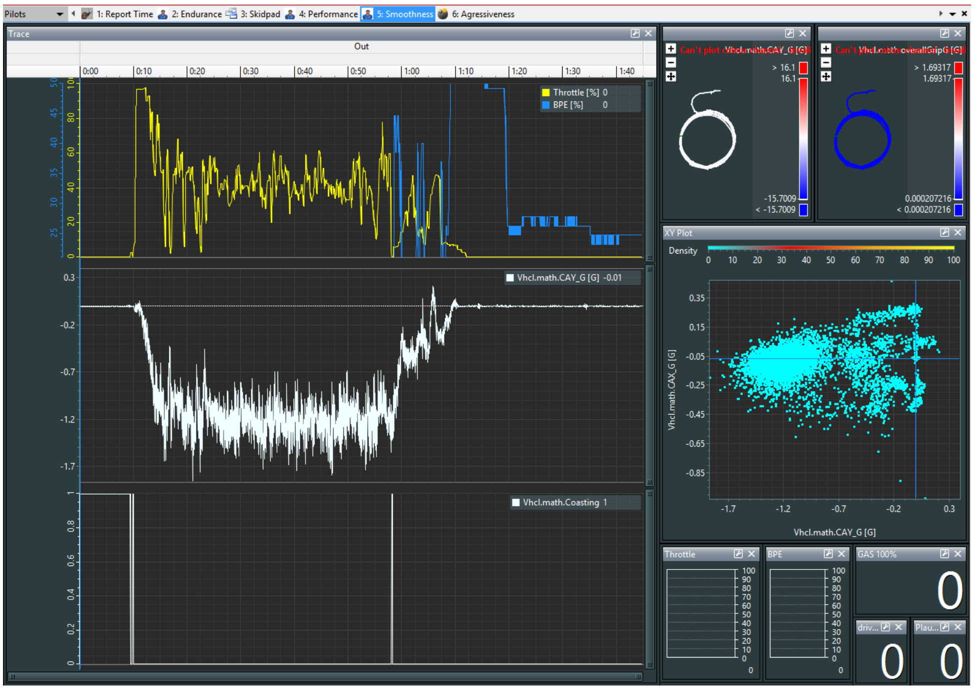
In recent seasons, inspired by other teams, the decision was made to analyze driver performance as well. Leveraging our familiarity with GEMS, it was chosen as the tool for this task as well. However, over the years, we have encountered certain limitations with GEMS.

Firstly, it requires paid licenses to unlock its full functionality. One of the tasks limited by these licenses is the ability to upload and use our own data, which is a limiting factor for us. The team has a restricted budget, and one license is not enough to properly evaluate and compare drivers.

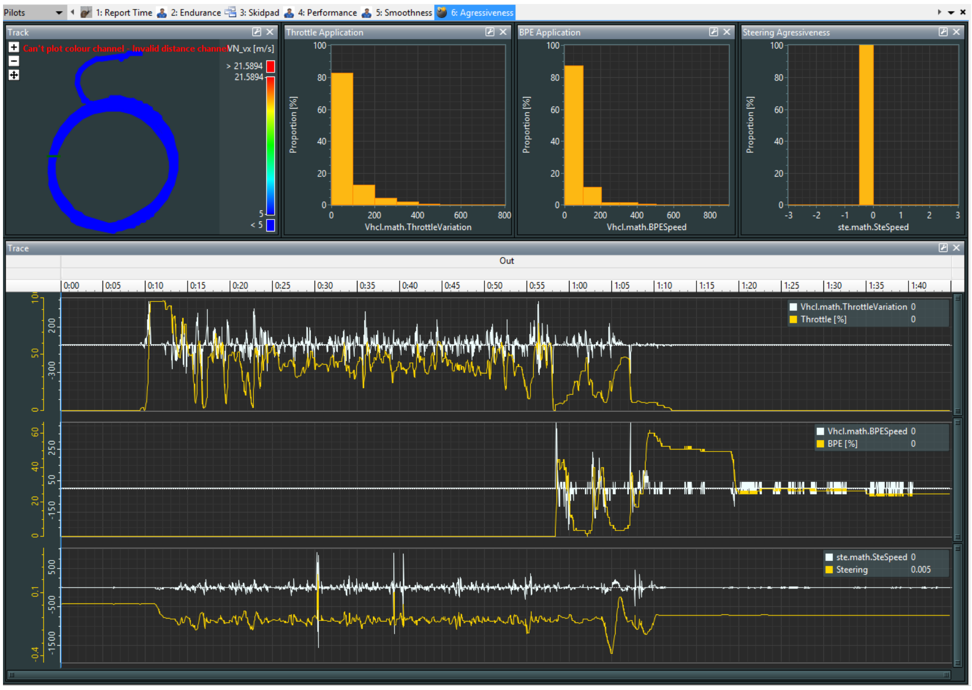
Additionally, while GEMS enables excellent dynamic visualizations, where multiple synchronized time-based data can evolve simultaneously, it can be somewhat limiting when it comes to comparing driver skills. These dynamic visualizations applied to drivers data are shown in Figure 4. This kind of visualizations only allow comparison between adjacent frames, and this comparison is difficult to understand by the user as they can not see both values at the same time. These visualizations are played as a video, as can be seen in the left part of the toolbar in Figure 4, and thus on-board videos can be added, which is very captivating.

Figure 4: Dynamic visualizations with the track layout and on-board videos used last season to analyze drivers data

Furthermore, GEMS provides a set of predefined graphs that cannot be customized as extensively as desired to meet our specific needs. For example, to visualize the drivers' harshness data, some team members have ideas for more communicative charts, but GEMS only allows the line charts, scatter plots, and bar charts visible in Figure 5.



(a) Last season's GEMS smoothness panel



(b) Last season's GEMS harshness panel

Figure 5: Last season's visualizations to analyze harshness limited by lack of other charts.

Considering these factors, the team began exploring alternative options for analyzing drivers' data.

2 Project Goals

The initial goal of this thesis is to create a visual analytics tool capable of analyzing the drivers' performance. This goal needs to allows us to solve the two problems exposed in Section 1.1.1:

2.1 Necessities of the team

The team needs this tool to evaluate the performance of their drivers. First, to select four drivers among all the driver candidates who will comprise their driver lineup. Second, to determine which of the four drivers will finally compete in the official events.

Not all the key attributes of a driver can be evaluated analytically, such as their ability to provide clear feedback to the engineers. However, there is a wide range of metrics from which we can take more advantage, such as time, racing line, or braking point, among others.

Other aspects that would help the team make data-driven decisions are the consistency of the drivers and their ability to push the car to the limit.

2.2 Necessities of the drivers

It is important to make this tool useful for the drivers as well. They are not used to having any criteria to improve other than intuition. Considering that we already collect a large amount of data related to their performance, it would be game-changing to develop a tool capable of helping them to understand better what is happening in their training runs.

Apart from mechanical aspects, it would be essential for the drivers to know their strengths and weaknesses. This means that they do not only need a overview of the entire lap based on the time needed to complete it, but more fine-grained detail is needed. This level of detail is achieved when analyzing specific parts of the circuit (sectors, microsectors, turns,...). On the other hand, like the team, drivers might also want to know their racing line and braking points. Finally, it would be critical to know more about their driving style, the optimal driving style of the layout, and their ability to change driving styles.

2.3 Requirements

Both problems discussed before led us to the following questions, important for our team, our drivers or both. Then we will define some project requirements from each of the questions that will help us to lead our developments.

- Which is the fastest driver throughout a lap?

  This question is as important as it is common. As the basics of racing are, being the fastest on the track is the way of winning. So time will be our main indicator to evaluate the improvement or deterioration in the performance. If the team needs to evaluate the performance of the drivers (need 2.1), evaluating time throughout a lap is the most generic way.

  This question can also be approached from a different angle, because in terms of comparison discretizing how fast a driver is in a lap may not be the best approach. This is because the final time difference may not be representative of the real time difference that there has been throughout all the lap. For that reason being able to evaluate which is the fastest driver *throughout* the different instants of the lap is important for us, too.

Requirement 1 We must know the different laptimes as an evaluation criteria, but also be able to compare them.

Requirement 2 We must be able to compare the differences of time throughout a lap.

- Which is the fastest driver for every sector (or microsector) of the lap?

  Despite not being very different from the previous question, the sectors question is not only evaluative, it allows us to start analyzing. We can extract which parts of the layout favor each driver -which helps the team for need 2.1-, help the drivers to understand where they might focus -which benefits the drivers needs exposed in Section 2.2- or know where drivers lose their advantage over other ones -which is in both interests-.

  Requirement 3 We must know the times for each sector or microsector, be able to compare them and know which are the best ones.

- Which racing line allows for faster lap times on the circuit?

  This is a challenging question, as the selection of the best racing line has always been a contentious matter. We do not intend to offer a universal answer; instead, we aim to provide a visualization that can guide drivers (need 2.2) and teams (need 2.1) in evaluating and choosing the optimal racing line for each turn of the circuit.

  Requirement 4 We must be able to compare different racing lines over the same circuit to analyze them by their own and to make comparisons among them.

  Requirement 5 We must be able to zoom the racing line visualization to specific sections of the circuit such as sectors, microsectors or turns.

- Which driver has more cornering ability?

  Cornering is a key attribute of a driver, turns are the parts of the circuit where usually more time is lost. Is for that reason that braking when it is needed, keeping in the racing line and not being slower than needed can make a difference.

  Braking, in addition to be key for a proper cornering, is one of the most important controls that a driver has over the car. Braking too much can result in losing valuable time, while not braking enough can cause the driver to go off-track. Braking too early can lead to a loss of top speed in the section preceding the turn, while braking too late can cause the driver to miss the apex and be slower in the following section. Thus, good braking performance is crucial in motorsports.

  Therefore, answering where drivers start braking in relation to the apex of each turn can answer also which drivers have more cornering ability. This information can help the team (need 2.1) identifying which drivers are not pushing the limits enough, which are pushing too hard, and which can serve as a reference for others to improve (need 2.2).

Requirement 6 We must be able to know where does each driver brake for each turn and compare it with the braking points of other laps in that turn.

- Which driving style (harsh/smooth) allows for faster lap times on the circuit?

  Beyond brakes, the driver has two additional controls over the car: the throttle and the steering. The intensity with which they are used can largely vary the response of the car. Applying the throttle at 10% of its travel is not the same as going full throttle. Similarly, going from 0% to 100% of throttle in one second is not the same as doing it in five seconds. The same applies to the steering. While it may seem obvious, there is no comparison between turning five degrees and turning 180 degrees.

  It should be noted that the brake pedal also can not be applied like a switch; it is a pedal too and it must be applied gently to avoid losing grip. However, as mentioned before, there is more room for improvement in the moment of braking than in the way it is applied. On the contrary, applying the throttle gently or harshly and turning the steering smoothly or not can have a significant impact on lap times.

  Therefore, in this question, we aim to determine how smooth or harsh a driver is with both controls and to determine if a particular driving style is more effective for the track we are practicing on. This information can clearly help drivers improving their performance (need 2.2).

  From another point of view it can help the team to detect each driver tendencies, knowing if a driver is smooth or harsh on the controls helps choosing which events are assigned to each driver in the competitions.

  Requirement 7 We must be able compare throttle harshness and smoothing harshness, and evaluate how they affect to the laptimes.

  Requirement 8 We must be able detect each driver tendencies in terms of driving style.

- Which driver exhibits more consistency?

  While it is challenging to assess consistency using quantifiable metrics, it is a crucial factor for winning races. A driver's ability to consistently deliver strong laps is often more valuable for the team (need 2.1) than sheer speed over a single lap. For example some DV competitions have two different events, one of only one lap and another of endurance which needs a larger amount of laps. For this second kind of events it is very important for the team to know which driver is the most consistent.

  To address this question, we aim to develop a metric that considers various factors, such as lap times, differences in braking points at turns, and measures variations in other metrics across multiple laps.

  Requirement 9 We must be able evaluate the consistency of a driver.

- Which driver is the best at pushing the car to its limits?

  We can measure how much the car's physics are being pushed to the limit with the accelerations it receives. If we had a proper car model, we would be able to know the theoretical limits of the car in terms of accelerations. However, due to the limited resources of a Formula Student team, such a model is typically not achieved by the end of a season, and even when it is, it may not be very accurate.

Therefore, the approach to answering this question is to compare the acceleration profile that a driver achieves during a lap with the composite limits reached by all the drivers over all the laps, thus creating a sense of empirical limits. This measure will not tell us how close the drivers are to the actual limits, but it will indicate which reachable accelerations the drivers are not reaching. In a less technical sense, it will inform the team which drivers are more capable of understanding and pushing the limits of the car to be quicker, and which ones do it less than their mates. If the drivers have access to this information, they will know where they have gap of improvement (need 2.2).

Requirement 10 We must be able evaluate how a driver is pushing the physic limits of the car and compare it to other drivers and laps.

Apart from the requirements derived form the questions, we have two more general extra requirements. They are not directly related to solve any technical questions, but they are needed to accomplish the exposed needs in sections 2.1 and 2.2.

Requirement 11 We must be able to select which circuit we want to work with, which lap we want to focus in or which laps we want to compare.

Requirement 12 Facilitate comparison and relations between information in different charts.

With all this requirements the main goal of the project now becomes: to create a visual analytics tool able to solve both mentioned problems and meet the requirements.

3 Data

In this section, we will explain the data available for analysis and its sources. It is crucial to ensure that we have the necessary information to visually address the defined questions and requirements. Without data, we will not be able to provide visual responses, whether it is data captured by the car or estimated from real data.

3.1 Data Collection

Our Formula Student car has several sensors distributed in key locations of the car represented in Figure 6. Some of them are very important for the team but can not help analyzing drivers such as suspension pressures, battery voltages or temperature of the processing unit among others.

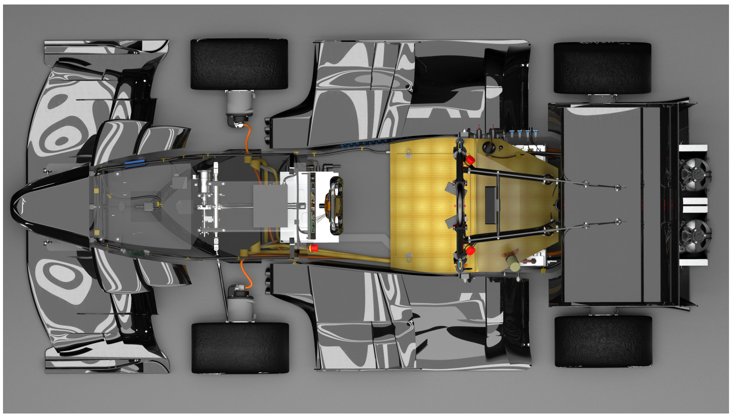


Figure 6: Detail of the sensors available in Cat15x

The data relevant for us, involving drivers, comes from two different methods: directly from sensors or from algorithms. The first ones' data is automatically saved, however the second ones' data is firstly calculated or estimated and then saved.

3.1.1 Data of sensors

VectorNav

The first relevant sensor for driver analysis is the VectorNav. It is represented as the white box visible under the steering in Figure 6, and which truly ends located under the driver knees. It has a inertial measurement unit (IMU) which provides us for velocity, orientation angle and gravitational forces data. Among this data, for us it is important the velocity for posterior calculus that will give us data to solve various requirements. The longitudinal and transversal accelerations are also very important data for us, they are needed to later solve the **Requirement 10** with a visualization.

Apart from the IMU sensor, the VectorNav also includes a GPS whose data will be important for the position.

Steering

The steering wheel apart from being a controller over the car it is also a sensor because registers its angle at every time instant. This data gives us direct information of how the driver works with the car controls and is needed to meet the **Requirement 7**. The steering wheel can be identified clearly in the center of Figure 6. The rotor which registers the angles is the grey box located behind the steering wheel from the perspective of a driver, and at its left if we look in the figure.

Throttle pedal

The throttle pedal is both a controller and a sensor, too. In this case it registers the percentage of the pedal's stroke pressed by the driver. Again, it is very important for the **Requirement 7**. The throttle pedal is located under the steering rotor and between the front wheels together with the braking pedal.

Braking pedal

The braking pedal, despite being also a controller, a sensor and a pedal, it registers the pressure applied to the braking system instead of the pedal's pressed stroke percentage. The braking data allows us to answer the questions related to the braking points and the cornering ability detailed in **Requirement 6**.

3.2 Data of algorithms

Perception pipeline

The LiDAR sensor is one of the most important ones for our car if not the most because our perception that allows it to race driverless. Despite the LiDAR being a sensor we store the data resulting from the perception pipeline. The sensor is not visible in the render of Figure 6 as it is located in the middle of the front wing, under the nose.

The original data coming out the LiDAR sensor consists in point clouds of the surrounding environment of the car. The perception's pipeline obtains the track limits of the circuit from the point clouds. These circuit boundaries are important to analyze the racing lines of the **Requirement 4**.

One of the algorithms of the perception pipeline is LIMO-Velo[6]. This algorithm accumulates different frames of point clouds in a global map. This map is used to have a better estimation of the position (discussed below).

Position estimation

The data of the VectorNav's GPS is not precise enough to have a proper positioning of the car which results in bad racing lines. To solve that, a few years ago a former team member developed a estimation algorithm combining the VectorNav's GPS localizations and the mentioned LIMO-Velo's mapping data. It provides us for data of the x and y positions of the car at every instant, and for instance it provides us for good racing lines. These racing lines are needed to compare them as **Requirement 4** demands.

LapCount

The last algorithm is LapCount. It is a program run in the car Processing Unit (PU) when the car is moving. This program takes advantage of the Robotic Operating System (ROS) process held in the PU which communicates all the other processes and signals existing in the car. Moreover, ROS has a internal clock shared by all the devices of the car connected to it. This allows LapCount to receive -for each time instant of this internal clock- all the data published by the chosen sensors and store it as relational data. Besides storing data, LapCount also adds some parameters to this table:

- **Time**: It stores the internal clock time instants. It will help us to calculate time differences and laptimes, the data important for several requirements such as 1, 2, 3 and 7.
- **Laps**: It keeps track of the car's position at the first time instants to detect when a lap is completed, this way it is capable to know at which lap corresponds every time instant. Without it we would not be able to achieve the **Requirement 11**.
- **Sectors and microsectors**: With a method similar to the Laps detection method, it divides the track in sectors and microsectors and assign them to the timestamps.
- **Velocity**: With the accelerations data and the time traking it also does a velocity estimation.

3.3 Data simulation

Despite being able to record all this data, we have not found any historical registration of car data with all the information we are interested in together. However, we have found individual data registrations for every sensor, estimation or subsets of them and therefore we have been able to build visualizations from real data. It can seem enough but to display the entire dashboard and meet the **Requirement 12** (which requires coherence of the entire application) we needed complete data registrations for all sensors and estimators.

The first option was to simply go to an EV testing session and record it. However, FS season starts in September and while the season's car is being designed and the materials have not being delivered yet the previous season car is maintained, which usually means until December. Then the new car is expected to be built by April and to be completely running in May, but that are only expectations and the reality is that the car uses to be ready to drive in June. Taking into account that this thesis duration is from November to June and that we faced this problem once the previous season's car was already not maintained, we decided to simulate data to have a complete visual analysis tool. This meant that we needed to be able to have coherent data in terms of accelerations, track limits, positions, steering, throttle, braking and time which is not trivial.

Luckily, the process of simulating data has been easier than expected thanks to the DV team. On the one hand, we benefited of Pol Puigdemont's TILK-E[4]; it is a track generator originally used to evaluate the performance of our track limits algorithms. On the other hand, we exploited GRO[2] from Albert Rodas; which is a racing line planner that estimates also the accelerations suffered by the car used to decide the best racing line for the car while running without driver. After slightly adapting the GRO code to work with circuits generated by TILK-E we were able to generate different circuits. Moreover, adapting -with the guidance of Rodas- some of the parameters of GRO's iterative optimization process we were able to generate different laps on those circuits. Finally, we defined different simulated drivers with means and deviations for each of the modified GRO parameters. For different laps we chose the parameters with a normal distribution with the mentioned mean and deviation. This way we were able to have simulated circuits and laps coherent among drivers.

4 Technologies

In this section, the different used technologies will be explained. It has to be said that I already had some experience working with Python and Altair. Apart from that I also had some knowledge of C++, but other technologies were unknown for me. Apart from that, in terms of number of different technologies, we have focused in not using a wide range of them if not needed and keeping it as simple as useful.

The technologies used in this Bachelor's Thesis have been divided in two main sections:

- **Data**: All the software that has been used to collect or generate the data for the visual analysis tool.
- **Web Application**: All the software that has been used to build the web app and to build all the visualizations and interactive graphics appearing on it.

Figure 7: Used technologies' logos

4.1 Data Technologies

As mentioned in Section 3, data can be collected or generated and the technologies for both processes are different.

4.1.1 Data Collection Technologies

All the code running on the car is programmed in C++ which is a compiled programming language and allows a fast time performance. To manage all the communication processes between nodes, sensors and actuators of the car ROS is used. ROS is the reference open source robotics software. By the time I started working on the FS team I did not know anything about it. I had to learn it and all that ROS knowledge was important for this project. Concretely, the knowledge was important to adapt the existing code and efficiently retrieve and store all the needed data. I also needed to master C++ because despite having some knowledge of it, I had no prior experience working on a C++ project, particularly of such significant scale.

4.1.2 Data Generation Technologies

The data generation do not need to run in the car so a faster time performance is not a requirement, for this reason Puigdemont[4] and Rodas[2] decided to implement their algorithms with Python. Some noteworthy libraries used by them that I needed to work with to merge both algorithms are scikit-learn[1], scipy[2] and pymoo[3].

- **Scikit-learn**: In order to discretize the outputs of the racing line planning algorithm and find nearest points efficiently scikit-learn's KDTrees has been used.
- **Scipy**: Both racing lines and track limits objects are built on scipy's Splines and in order

---

[1] You can find the scikit-learn documentation here.

[2] You can find the scipy documentation here.

[3] You can find the pymoo documentation here.

to plot them in charts we needed to work with this kind of objects.

- **Pymoo**: Pymoo is the library chosen by Rodas to perform the iterative optimization process, to not find always the optimal solution and have different driver profiles it was necessary to work with this library.

4.2 Web Application Technologies

To build the application, we wanted to keep it simple, as already mentioned. For this reason, and due to my previous knowledge of the Altair library in Python, we decided to use Python to build the entire application.

4.2.1 Altair

Altair is a powerful open-source data visualization library in Python that allows for the creation of elegant and efficient static and dynamic graphs. It has been used for this bachelor's thesis, drawing upon the knowledge I acquired during my course on Information Visualization[4]. By leveraging Altair's capabilities, we have developed interactive and expressive visualizations to explain the drivers' data, and after include them to the application.

It has to be said that a new version of Altair was released[5] while the thesis was underway, but the decision was made to continue using the previous version for compatibility reasons with Streamlit.

4.2.2 Streamlit

Streamlit[6] is a Python framework designed for creating web applications with interactive data visualizations. It seamlessly integrates with Altair, allowing for low-effort development of web-based applications incorporating Altair visualizations. For this project, we realised that Streamlit was a perfect fit for showcasing our Altair visualizations in a user-friendly manner, enabling easy exploration and interaction. Although I had no prior experience with Streamlit, I took the initiative to learn it and successfully leveraged its capabilities to build a compelling web application for my thesis.

---

[4] The description of the course can be found here in catalan or, briefly, in english.

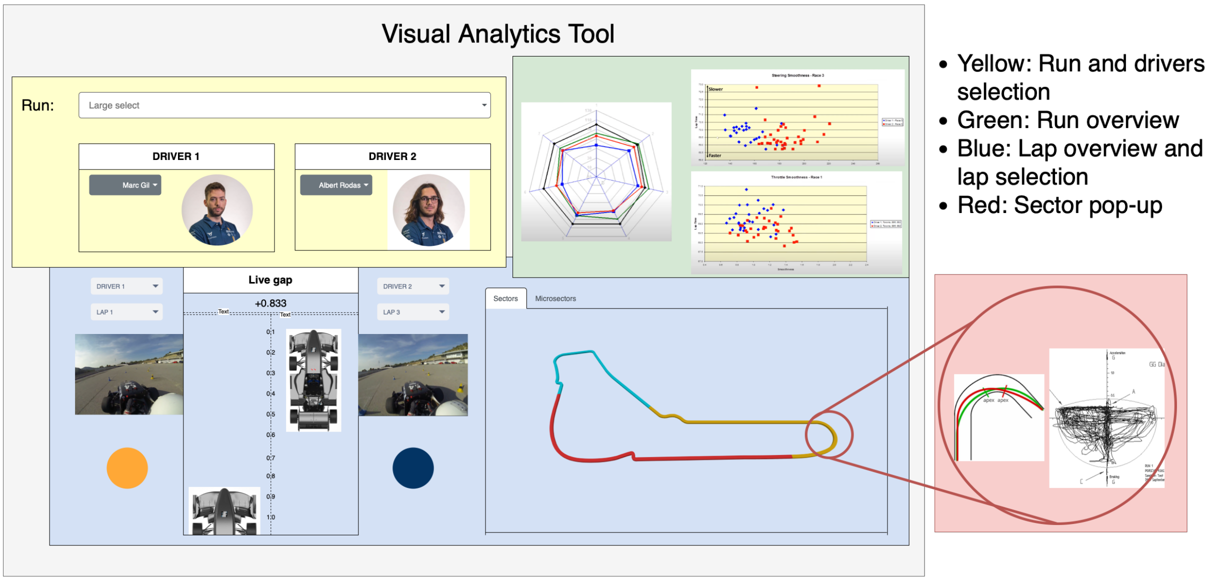
[5] Altair documentation for the most recent version 5.0.1 and for the one used in this project 4.2.2.

[6] Streamlit documentation is able here.

5 Project Development

In this section, the creation of the different charts conforming the application will be explained. First, we will show the first sketch to explain sequentially for each chart which problems solves, which options have been considered, if any setback or challenge appeared, etc. After detailing all the charts form the sketch we will show the final application.

The first approach (shown in Figure 8) consists of dividing the application in three zones. First, the *Selectors Section* (colored in yellow in the sketch) where the user can make selections to access the information that he is interested in. Then, *Run Overview* (green-colored in the sketch) where, given a circuit with all the different laps driven on it (a run), we can get insights of each lap and driver and compare it with the rest. At last, the *Lap Overview*, over a blue background, where lap-versus-lap comparisons will be performed. This last zone will also include a subsection for sector (or microsector) comparison which is colored in red in the figure and primarly was thought as a pop up view.



- Yellow: Run and drivers selection
- Green: Run overview
- Blue: Lap overview and lap selection
- Red: Sector pop-up

Figure 8: Sketch of the application's first approach

5.1 Selectors

For this section we need the user to be able to make selections and change them at any time. Also, we want to transmit to the user that the making the selection precedes working with the visualization. Guided for this section's second need, in the first approach represented in the sketch (Figure 8) we located the selector section in the upper part of the dashboard. This made sense for a fix visual analytics tool but as we advanced developing different charts they needed for more space and for scrolling the page. The scroll resulted in a occlusion of out selectors section when we were working with certain parts of the dashboard and that supposed a breach of our first need. To make the selectors section fix and able any time the best streamlit (Section 4.2.2) object able to match that need was the sidebar. This way we will have always a side bar section opened with the selectors ready to be changed.

Apart from being crucial in order to meet **Requirement 11**, selectors are also important to ease comparison and relations between the data displayed in different charts as the **Requirement 12** specifies. They are important because they allow the user to filter the data, make selections and mute the application aspect. Therefore, we have to be able to respond coherently to these view changes.

5.1.1 Run selector

For us a Run are all the laps performed in a circuit, so the easiest way to select a run is by the name of the circuit where the laps have been performed. Taking into account that there will be an arbitrary number of runs in the datasets, the best selection method will be a dropdown. This way, to select a lap you will simply expand a dropdown with all the runs' circuit name and select the one you are interested in. It is our first step to achieve the **Requirement 11**.

5.1.2 Lap selector

5.1.2.1 First approach

In the first approach we required first to select two drivers and then a lap of each driver (driver selection in the bottom of the yellow zone of Figure 8 and lap selection above the two on-board images). When commenting that with the team's drivers they complained about it and asked us to make sure that two laps of the same driver could be compared. This meant that if we continued with the "first driver, then lap" approach we needed to select two times the same drivers to compare two laps of their, which seemed anti-intuitive. In consequence, we changed it to a simpler laps selection with each lap tagged with its driver.

5.1.2.2 New solution

The new approach meant that after a run is selected, its time to select or not, one or multiple laps. These three casuistics guide us to build a selector able to be empty or to make multiple selections. The first idea to come to mind (using streamlit[4.2.2]) is a `st.multiselect`[7] object, which consists in a dropdown with all the options and every time you select one it is added to a list of selections. In that list of selections you can also undo individual selections. However there is a possibility that certain charts are not able to be built with more than two laps selected. Later, in the following sections, if this problem is recurrent in several charts, we can change to a different approach. This new approach will consist in two dropdowns, one for *Lap A* and another one for *Lap B*. Both must have the *Not selected* option to be able to have the empty selection casusitic. Moreover, *Lap B* dropdown should be disabled if no selection is done in *Lap A* guiding the user to first select only one lap and evaluate it and later selecting the second to compare with.

Both approaches meet the **Requirement 11**; therefore, the decision of which to use will be detailed after explaining the different charts. This decision is be based on the chart's versatility to display different numbers of laps.

Finally, we have noticed that picking the lap blindly could not be the best idea, so we thought to try responding one of the Section 2.3's questions in the selectors sidebar.

---

[7] Streamlit's multiselect object reference and examples can be found here.

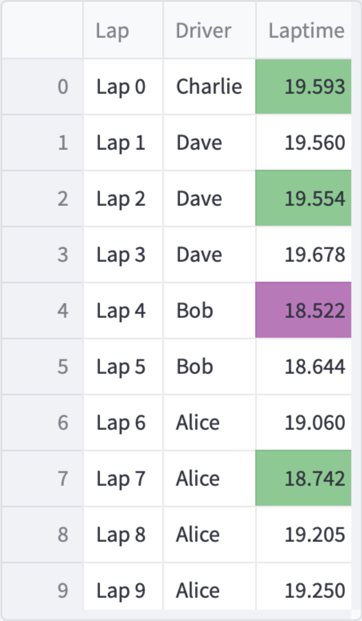
5.1.3 Fastest driver

"Which is the fastest driver throughout a lap?"

To make a more intelligent selection of the lap we want to work with, it can be very interesting to be able to partially respond to the above question before. This way, if given a run we have a intuition of their laps' times we will be able to select more consciously.

If we want to strictly respond to the question above it is necessary to clearly convey which driver has completed a lap in the minor time possible. This, has a simple and effective solution that consists in displaying a table with all the drivers and their best laptime. Although, the solution answers the question, it avoids information and does not help the selection as it is about laps, not drivers. The fact is that every driver uses to have completed more than a lap in a circuit and only showing the best time is not as faithful as it is to show all the laps.

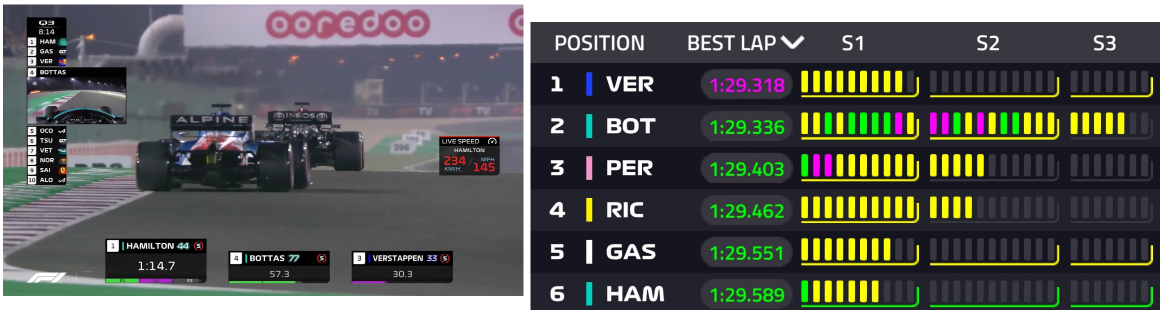
This option of a table with all the laps on a circuit with their driver and laptime can easily escalate in dimensions and density of information. Moreover it can lead to confuse the user and have difficulties to get the key information. For that reason and to enhance the interpretation of the raw time numbers by the user, we included a color code to the cells of the table, resulting in a table like the one in Figure 9. The resulting table will help selecting the lap interesting for what we want to know while meeting the **Requirement 1**.



	Lap	Driver	Laptime
0	Lap 0	Charlie	19.593
1	Lap 1	Dave	19.560
2	Lap 2	Dave	19.554
3	Lap 3	Dave	19.678
4	Lap 4	Bob	18.522
5	Lap 5	Bob	18.644
6	Lap 6	Alice	19.060
7	Lap 7	Alice	18.742
8	Lap 8	Alice	19.205
9	Lap 9	Alice	19.250

Figure 9: Laptimes table with color encoding.

The selected color code is widely used in motorsport (as shown in Figure 10) and involves coloring the best time overall in purple, while using green for the best time of each driver and no color or yellow for the rest.



POSITION		BEST LAP	S1	S2	S3
1	VER	1:29.318			
2	BOT	1:29.336			
3	PER	1:29.403			
4	RIC	1:29.462			
5	GAS	1:29.551			
6	HAM	1:29.589			

(a) The color code used in Formula 1 TV broadcast

(b) The color code used in Formula 1 app

Figure 10: Examples of the use of the explained color code in Formula 1.

5.2 Run overview

As mentioned, the idea for this section of the dashboard is to work with the selection of a run. For that reason we have to answer the questions which can be responded without any further detail.

The questions that can be responded with only the run selection are the ones responding to the cornering ability and the driving style. In fact the comparison of racing lines and the evaluation of the ability to push the car to its limits can also be performed but the occlusion of racing lines and the density of physics data, respectively, makes both visual answers unfeasible.

The Run Overview is the one with the green background in Figure 8 and it is designed to have two views:

- The *laps* view displays information for all the laps of a run. Despite being interesting for the team, it may help more the drivers (need 2.2) improving their performance.

- The *drivers* view displays the average data of all the laps corresponding to each of the unique drivers of the run. It is intended to be more helpful for the team (need 2.1) as it compares drivers.

This dual modality can be implemented with tabs allowing the user to swap from one to another in any moment.

5.2.1 Cornering Ability Evaluation

"Which driver has more cornering ability?"

5.2.1.1 First Approach

The first approach to know more about the cornering ability of a driver through the braking points is represented in the chart at the left of the green zone. It takes into account that braking points can be evaluated as the distance to the turn apex at the moment of start pressing the braking pedal. Those evaluations can be easily compared as it is a distance measure. To visualize this evaluation and comparison we use the approach proposed by Claude Rouelle in the FS Russia *"Driver and chassis debrief"* webinar [5]. It can be seen in the the Figure 11 and consists in a radar chart that illustrates the braking points of each lap of each driver on a circuit.

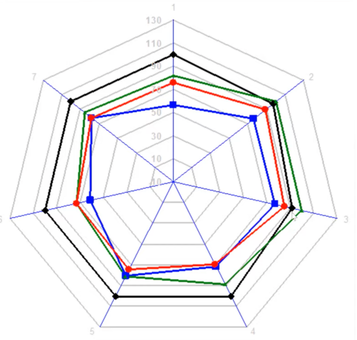


Figure 11: Rouelle's braking radar chart

A radar chart is none other than a graph in the form of a polygon with radiating axes from a central point. Each dimension is represented by an axis, and the extent of the axis indicates the value or magnitude of the variable. The lines connecting the different axes form a polygonal shape that allows for the comparison of data across the different dimensions. It is a interesting chart because it allows visualizing multiple dimensions at the same time, it allows to make a relative comparisons between different individuals across different categories and it emphasises differences between values. The individuals with higher values in the polygon indicate values that are more significant or have a greater impact. Despite these advantages, radar charts are very sensible to scales and are more suitable for datasets with a relatively small number of dimensions and datapoints, the bigger number of them, the more cluttered and less useful for analysis it becomes.

For the Rouelle's example in Figure 11 we can observe that each axis of the chart represents a turn of the track, and each polygon profile line represents a lap. The axis measures the distance in meters from where the driver starts braking to the apex of the turn. If we are capable to adapt this visualization to our data we will be able to meet the **Requirement 6**, which needs to know were does a driver brake for each turn and compare it with other laps.

5.2.1.2 First approach problems

After starting developing the charts this first approach presented certain problems:

- Firstly, evaluating the cornering ability with the braking points only it is not the most complete way to do it. In order to be more accurate we can redefine the cornering ability: considering the braking point as the start of the cornering and the moment after the turn when the maximum throttle is reached as the end. The ability to reduce the separation of these two points while keeping the car performing at its most in between, is a more accurate definition of a good cornering ability.
- Despite knowing that a good turn its not only defined by the braking point and that braking as late as possible does not have to be the best option, if we only look on the braking points we will consider them "likely to be better" as nearer to the apex. This collides with the natural interpretation of a radar chart, where -in terms of evaluation- the further from the origin a value is, the better that value is expected to be, as explained before. This means that we will need to be able to evaluate an inverse measure that was higher for braking points nearer to the turn apex.
- Furthermore, defining where is the apex is not trivial. Theoretically, we could look for local minimums in the curvature but that is a data not at our disposal. Another option was to simply look for the braking points of the different laps and try to make clusters in order to detect where the turns were. This option do not worked well as there were laps with no braking in some turns or more than one braking point per turn. The fact is that we were facing problems to define the apexes or the braking zones and without them it was impossible to create the chart proposed by Claude Rouelle.

5.2.1.3 New solution

Given these problems faced, we presented a solution solving them and not leaving **Requirement 6** behind. First of all, we started working with microsectors. They are defined easier than an apex and are along all the circuit, so the solution involves manually defining turns by selecting the microsectors that cover it, and result with selections like the one in Figure 12. As microsectors are bounded by doors, with them we can have a notion of start and end of the turn besides the interval itself. This helps us to solve our first problem because now we will have information of the start, duration and end of the corner and with that data we will plot three charts shown in Figure 13.

The multichart located at the bottom of Figure 12, corresponds to the braking points plot and shows the distance from the door of the first microsector assigned to the turn to the braking point. This new measure will plot later breaks as higher values giving the expected intuition to the chart. However, the fact that doors are set without taking into account where the apex is can cause really different values for different turns. Also if microsectors are not chosen properly, the braking point can be before the microsector door so they have to be manually set consciously. This chart was designed to be a radar chart at first instance but the heterogeneity in the locations of the microsectors with respect to the turns apexes caused the turns to not share the axis domains. This happened to make impossible to distinguish different data points for the same turn so we decided to flatten the chart to a column-separated 1D scatter plot. This visualization allows us to have a different domains for each turn which eases the comprehension of the data shown.

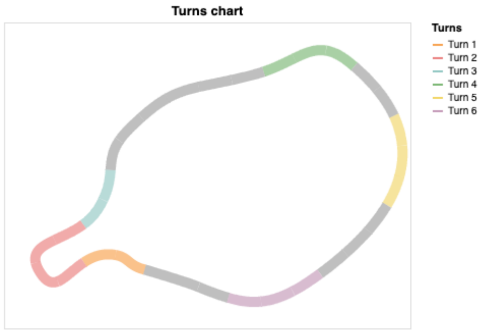


Figure 12: Example of a turns microsectors selection

Secondly, on the top-left of Figure 13, the chart corresponding to the duration of the corner will display the average velocity of the car during the microsectors defining the turn. This chart is a radar chart, as the following one, because their data did not show the multiple domains problem.

Finally, the evaluation of the end of the turn will ideally be the point when maximum throttle is reached, however maximum throttle can not always be reached, for example if there is another turn after. Our solution is a radar chart shown in the top-right position of the mentioned Figure and, taking advantage of the microsector doors, it plots the velocity at the end of the last microsector assigned to the turn.

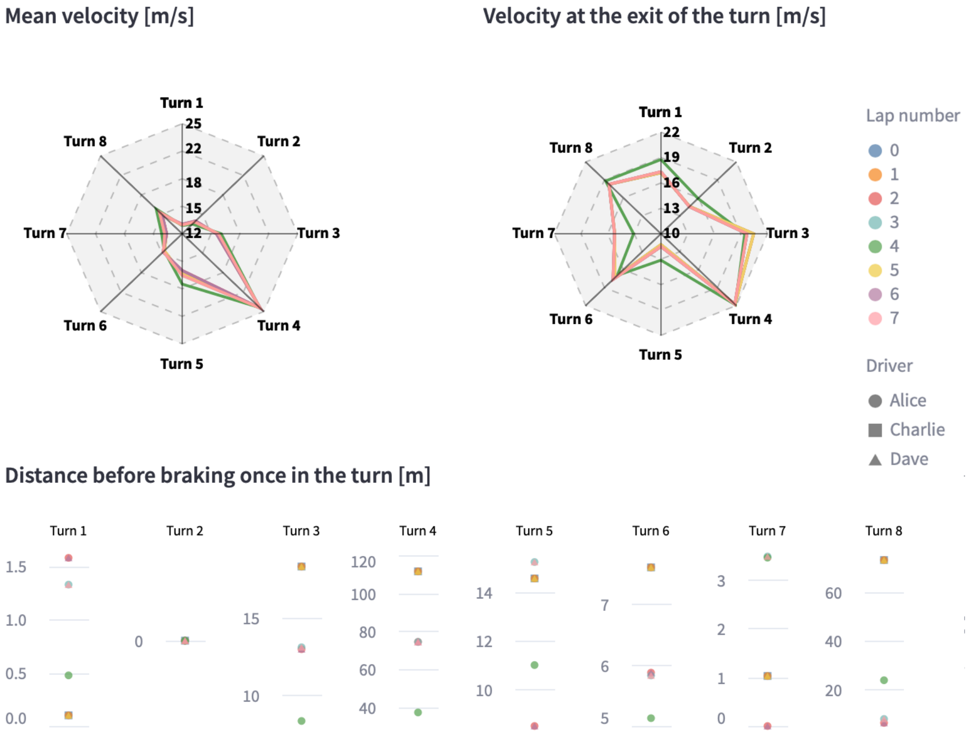


Figure 13: Braking points and cornering final solution charts.

To complement this new approach and to make it easier for the user to be plenty conscious of how are metrics obtained, it will be interesting to add a plot of the circuit with the microsectors that define turns colored (Figure 12). This chart will give us a sense of the width of the turns area in case that we face problems with the braking points before the first microsector door.

Finally, to meet the **Requirement 12**, about the coherence of the entire application it is important for all the charts to compare the information equally aggregated, so it needs the chart to mutate when laps are selected or only driver metrics are displayed. For the laps we will simply display the selected laps and for the drivers we will average the different values of each driver.

5.2.2 Best Driving Style

"Which driving style (harsh/smooth) allows for faster lap times on the circuit?"

To answer the next question, we will use a proposal of Claude Rouelle in the FS Russia webinar [5], too. We will evaluate the most efficient driving style with a scatter plot where we plot points corresponding to the lap time versus the harshness of the control (throttle or steering). Scatter plots use lengths (or 2D positions) in common scale to lay out datapoints, and this is perfect to compare because we are good at perceiving those visual variables.

Each scatter plot point shape will define the driver of that lap and the color will identify the lap itself. The harshness measure proposed by Rouelle consists in the following procedure:

1. We will work with the throttle and steering signals (or control signals), which are the pedal pressure and the rotation angle over time, respectively.
2. First, we take the original control signals and smooth them. We can smooth the signals with any low-pass filter, in our case we will use a Hamming window of size 21 since Rouelle does not make any suggestion and this one is pretty standard and works properly. Results for the two signals of one of our simulated laps are displayed in the upper-left chart of the Figure 14.
3. Next up, we subtract the smoothed signal to the original one and obtain the absolute value of the result. The result can be seen in the top-right chart of the Figure 14.
4. Also in the top-right chart, we can see the colored area under the signal resulting from the absolute subtraction. This area will be our harshness measure and we will obtain it using discrete integration techniques applied to the signals.
5. Finally and as mentioned, the total area values will be plotted versus the laptimes. Following the logic of the procedure, lower values of the harshness measure indicate smoother-driven laps, while higher values indicate more harshly driven ones.

For the *laps* view, the scatter plot of the Laptime versus the Throttle harshness and the one of the Laptime versus the Steering harshness display the harshness and time values for each lap. These charts are shown in the bottom of Figure 14. The charts also make us accomplish the **Requirement 7**, about comparing controls' harshnesses and evaluate their effects on laptimes.

For the *drivers* view, the scatter plots' values need to change to the driver average ones. Analyzing distributions with the previous view and with this view's driver averages we meet the **Requirement 8** because we allow the user to understand each driver driving style.

Moreover, if we implement the same options than for the cornering plots and we filter for laps when selected, we make the chart coherent with the entire application and in consequence meet

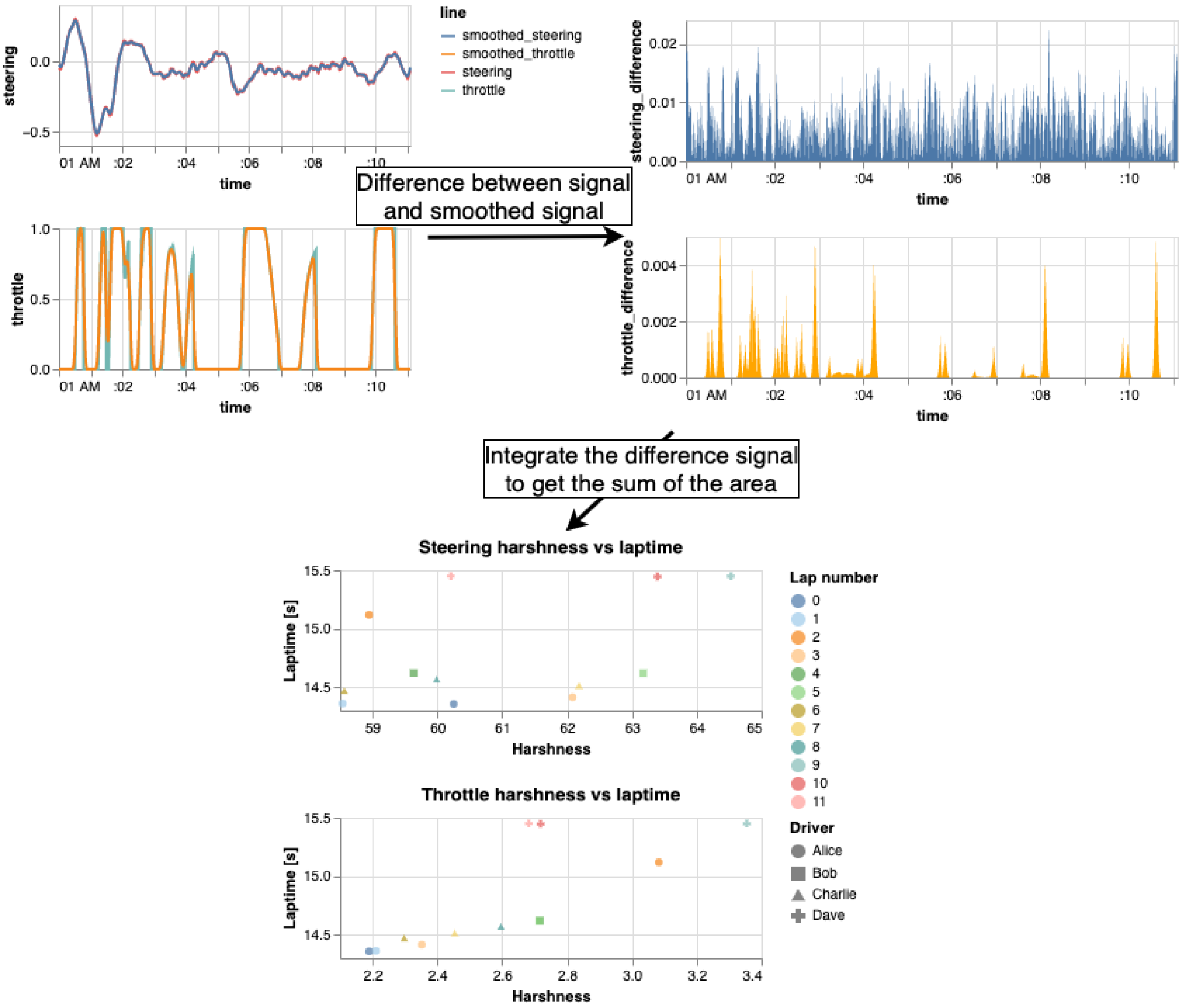


Figure 14: Process for obtaining the harshness values

the **Requirement 12**.

5.3 Lap overview

This last section provides for insights when a selection is applied to laps. It is important to take into account that there are two types of selections for laps: selecting one lap (for evaluation) or selecting multiple laps (for comparison). This will lead us to two different views depending on those selections, each view will be guided by the goals of each task (comparison or evaluation).

To be able to provide response to **Requirement 3** and **Requirement 5** which require track sections to be acomplished, we will need to be able to work with sectors or microsectors. As the pop-up option thought for the sketch (red part of the Figure 8) is not supported by Streamlit, we changed it to a switcher-like option as done in the Run Overview (Section 5.2) for the laps/driver modes. In this section it will change from sectors view to microsectors' and vice versa and it will increase us the number of possible views for this section to four.

Finally, in order to keep meeting the **Requirement 12** we will need a neutral display for when no lap is selected. This will be the track colored in grey -a neutral color- with the sectors or microsectors delimited (based on the view selected).

5.3.1 Fastest driver

"Which is the fastest driver throughout a lap?"

Despite being already mentioned, this question is not completely answered as not all its requirements are met. The partial solution of Section 5.1.3 meets the **Requirement 1** about laptimes but not the **Requirement 2** which intend to work with laptimes as continuous data throughout the lap.

5.3.1.1 First approach

To meet **Requirement 2**, our first approach for this two lap's speed comparison was inspired by the Spanish DAZN F1 broadcast[1]. The analysts of the program compared the poleman and the second fastest driver of the session with a visualization conveying what we wanted. Despite the program not being published, the visualization we are interested in can be seen in this YouTube video and also in Figure 15.

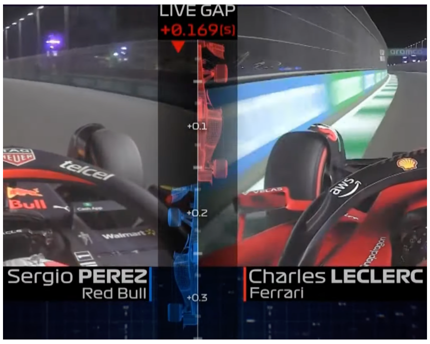


Figure 15: Spanish DAZN F1 broadcast's time difference comparison throughout a lap chart

Our version of the broadcast's visualization, can be seen in the first sketch's *lap overview* (blue section) left part, in Figure 8. It can be described as an animated chart with an axis representing the time difference between the two cars and distance covered by the car in the lead as the temporal dimension of the video. Both cars are represented as two top profiles of the single-seater, which makes the visualization as catchy as easy to understand what is conveying. The representation of the car ahead keeps on the zero axis while the other one moves along the axis corresponding to the time difference between cars at the covered distance of the frame.

5.3.1.2 New solution

The truth is, this dynamic visualization was included because of how catchy it was. However, after evaluating the chart we realised that the temporal dimension prevented a wider understanding of the lap that was exactly what we were looking for. Moreover, it only allowed us to focus on one frame and compare among their neighbour frames. Taking advantage of working only with two dimensions, we are able to plot the time difference in one axis and the covered distance of the circuit in the other axis. This plot is shown in Figure 16 where we can see for this example that the lap 5 was faster almost until the end of sector 0, during the sector 1 lap 1 gained advantage and in sector 2 this advantage was kept.

This visualization can only be displayed when exactly two laps are selected and needs to be hidden otherwise to keep the entire dashboard coherent (as required by **Requirement 12**). This mean that it would be able for the two views with two laps selected over the four possible; sectors and microsectors views do not affect in this case. The needs of this chart make us lean towards limiting the lap selection to a maximum of two laps instead allowing a unlimited selection of laps as proposed in Section 5.1.2.

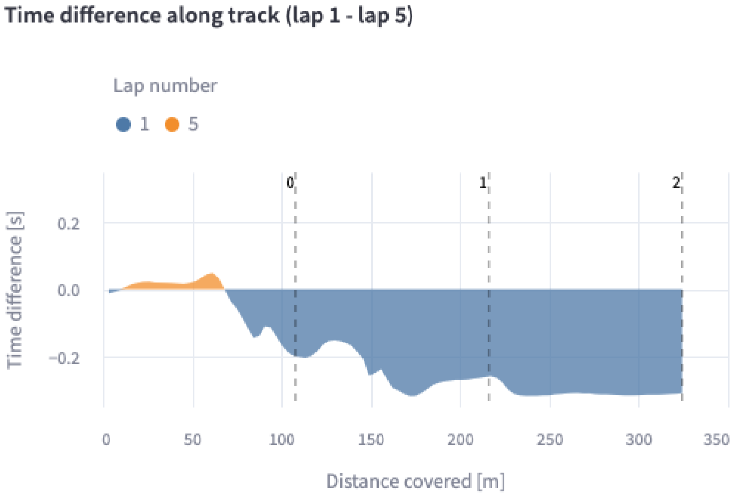


Figure 16: Time difference between drivers throughout a lap chart's new solution

5.3.2 Fastest driver per sector

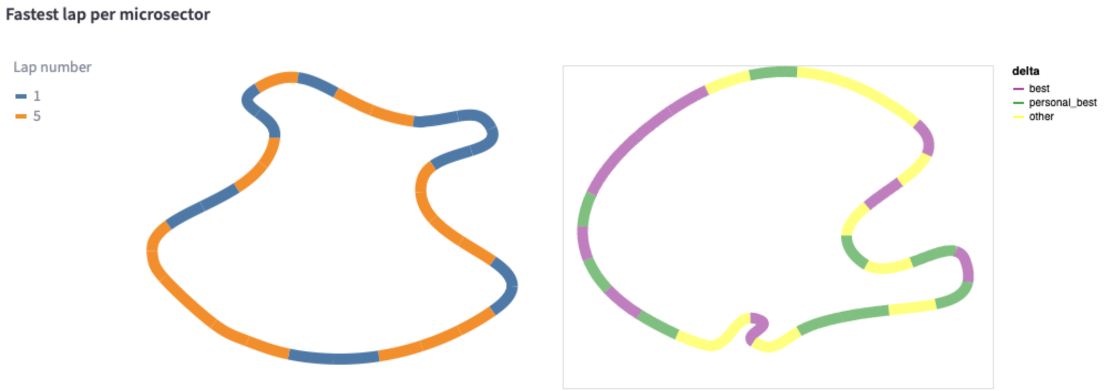
"Which is the fastest driver for every sector (or microsector) of the lap?"

The second question to be answered in the Lap overview, concerns to the fastest driver of every sector or microsector of the lap. This information is important for the drivers and the team when they analyze driver performances on different circuits' shapes of the layout. With information that responds to this question, it is possible to understand if certain parts of the track benefit more a driver or another and how it is reflected on their times.

Depending on how specific we want to be, we can work with sectors or microsectors. Usually, sectors are meant to be thirds of the track and microsectors tenths of a sector. This cause the sectors to have relevant information as they almost every time comprehend turns; unlike some microsectors that can comprehend just a straight and thus almost no relevant information. On the other side, sectors can be too general and comprehend various different layouts. For example a sector can have a slow corner and some linked curves, these different layout shapes are usually analyzed separately by the team. On the other hand, microsectors allow to be more explicit and select only the layouts the user is interested in.

As can be extracted from the previous paragraph, the answer to this question is more about comparison than evaluation. It is for this reason that the proposed chart needs at least a lap to be selected to perform comparisons with it. The proposed chart for the four available views consist in:

- For single-lap-selection views, like the case of Figure 17b, it will plot the track divided by sectors or microsectors -depending on the view-, each one colored following the color code explained for the table in Section 5.1.3. If the time recorded for the sector is the best overall ,it will be purple; if it is the best of that driver, green; and, otherwise, a neutral color as yellow.
- For multiple-lap-selection views, the sectors or microsectors -depending on the view- will be colored after the lap with the best time annotation for that part of the circuit. This case can be seen in the chart of Figure 17a.



(a) Colored microsectors chart for a multiple-lap-selection view

(b) Colored microsectors chart for a single-lap-selection view

Figure 17: Colored microsectors chart options

This leaves us four possible chart views, two for each of the two track division possibilities (sectors and microsectors) times the two selection possibilities. If in each of the views we display the exact time, for example when hovering the mouse over each section, we meet the **Requirement 3**.

5.3.3 Fastest racing line

"Which racing line allows for faster lap times on the circuit?"

The next question asks about the racing line that allows faster lap times on the circuit. We can not evaluate a racing line per se but we can visualize the racing lines knowing which belong to the fastest laps.

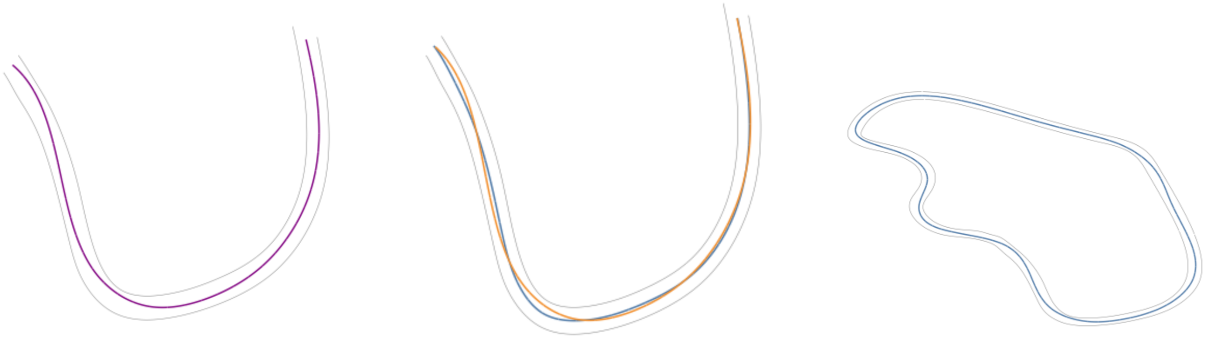
5.3.3.1 General idea

The whole behaviour of this chart consists in plotting the track limits as grey lines and one or many racing lines with a wider line than the track limits. The racing lines can be colored entirely, by different sections of the chart of with a gradient depending on the propose of the view. The simplest definition of this chart can be seen in Figure 18c.

This representation of the racing line with the track limits allow the user to observe how close to the track limits a racing line goes, if it tries to take a constant curvature while surpassing the turns more than doing a sharper apex, among others.

This base chart has an intuitive variation for views with different number of laps selection: which consists in plotting only a racing line for the one lap selection view, and simply plot multiple racing lines in different colors for the multiple lap selection view -this second view makes meet the **Requirement 4**-. This second view is represented in the first approach's sketch, at the left of the *sector overview* (red zone) of the sketch in Figure 8. This view allows us to compare different racing lines added to the evaluation that allows plotting just one racing line, explained before.

The sectors and microsectors views allow the users to filter by parts of the circuit, in the sectors view the filters are over sectors and in the microsectors one over microsectors. This way we allow the user to focus its analysis in only a certain part of the circuit with more zoom. Simple and multiple lap selection views filtered by sector can be seen in Figures 18a and 18b, respectively.



(a) The racing line chart's single-lap-selection view with a selected sector which corresponds to the fastest lap in that sector.

(b) The racing line chart's multiple-lap-selection view with a sector selected.

(c) The racing line chart view for a single-lap-selection and no sector nor microsector selected.

Figure 18: Different examples of views of the racing lines chart

5.3.3.2 Redesigns

Besides having already a visualization meeting the requirements that we defined to answer the racing line question (**Requirement 4**), the idea of enhancing the different views with more information specific for each one came up:

Single-lap-selection views. For the case single-lap-selection views, we came up with the idea to fusion the chart with the fastest sectors (or microsectors) for one-lap-selection views of section shown in Figure 17b. This way we would be able to respond two questions with a chart. The merge means plotting the different sections (sectors or microsectors) with the color code explained for Figure 17b. An example of the resulting visualization can be seen in Figure 19.
This does not apply to the multiple-lap-selection views as the color codes for distinguish the different racing lines and to emphasize which lap has been the fastest in that sector or microsector do not match.

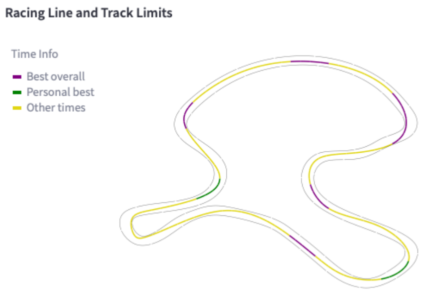


Figure 19: Racing line chart colored with sector color code.

Multiple-lap-selection views. For the multiple-lap-selection views, and guided by the objective of enhancing the chart, the idea that came up was to plot the chart with a color gradient for the velocity of each car at each position. This way the user is able to understand at which velocity is each driver doing his racing line, to better understand their behaviours. For example, if a driver does a more open racing line than another driver but he keeps a higher velocity we can think that it is because he has a different driving style. Otherwise, if it is more opened that the one of the rival but it needs to slow down equally, we can understand that the first driver has made a mistake.

5.3.3.3 Design decisions

After implementing both evolutions we evaluated them to decide whether we used them in the final application or not.

Firstly we realised that the more racing lines are plotted, the more difficult it is to compare them. Moreover, the larger racing lines are plotted, the more difficult it is to detect differences and the less detail can be observed. With this in mind we decided that one racing line for the entire circuit is not too much and can be properly evaluated (as in Figure 18c). For this reason we decided to apply the one-lap-selection views evolution.

On the other hand, two racing lines on the entire circuit seemed too much to properly compare them. It did not suppose to discard the multiple-lap-selection views evolution, on the contrary, it supposed to merge the racing lines charts and the colored sectors charts for all the views. The solution resulted in: for the one-lap-selection views, the racing line chart plotted as explained on the first evolution. However, for multiple-lap-selection views, the chart display the colored sectors chart (as in Figure 17a) when no sector -or microsector- is selected and the sector-filtered racing lines chart (Figure 18b) when a section of the circuit is selected.

Even filtering by sectors, plotting more than two racing lines hinders exponentially the ability to compare them. This conclusion reinforces the idea to limit the lap selection to two.

Apart from that, we have not already analyzed deeply the multiple-lap-selection views evolution. We realised that comparing two color gradients of different colors is not a good practice. Moreover, for the comparison goals described in Section 2.1, it remains more important to be able to identify each lap than knowing the velocity. We tried several visual variables to encode different laps racing lines and facilitate the perception of differences or similarities on the velocity gradient. After these tries, we reached the following conclusions:

- Different patterns with discontinuities may potentially result in a loss of information since the details we are interested in are highly precise in terms of position and velocity.
- Lines of the same color or scheme (as the gradient) may lead to confusions as racing lines intersect constantly.
- Different line widths may lead to occlusions as the track limits are quite narrow and all the lines have to be inside of them.

Apart from these discarded options, increasing the size of the chart does not scale well, and would not leave enough room for the other charts, so a compromise had to be taken. We finally decided to reject the velocity gradient evolution for the racing line charts in multiple-lap-selection views.

5.3.4 Sector and microsector selections

The previous section has raised the need to be able to select sectors or microsectors to get more precise insights of them. The first idea for it was to be able to click in any part of the circuit to open a pop-up window with that sector insights. We faced that with the dashboard technology that we were using (explained in Section 4.2.2) it was not possible. However, the program offered us another viable option, radio buttons with each of the sectors as options plus a *"All* *sectors"* option (and the same for microsectors). The *"All (micro)sectors"* option is intended to keep the original representation when no sector has been selected.

For sectors, it was a perfect solution as it resulted in 4 options, but for microsectors it was overwhelming as it resulted in 31 radio buttons. To facilitate interaction, we tried to overcome one of the problems of working with microsectors (previously mentioned in Section 5.3.2): the possibility of some microsectors to comprehend just a straight and thus almost no relevant information. The solution we came up with was to select microsectors with a slider, this way we can either select only one microsector, a range of microsectors of our interest or all microsectors replacing the *"All microsectors"* button. This selection method has two problems: it is not as intuitive as the other two proposals and might need some extra explanation to the user, and it allows to select almost all the track microsectors and make the racing line chart difficult to read as in Figure 18c. The first problem is solved giving the user a brief explanation if need, and the second one, we think is compensated by the versatility offered to the user and the fact that displaying a less readable plot is under the users responsability.

The resulting selection with radio buttons for sectors and a range slider for microsectors makes us meet the **Requirement 5** that asked for the ability to "zoom" in the racing line visualization.

5.3.5 Drivers ability of pushing the car to its limits

"Which driver is the best at pushing the car to its limits?"

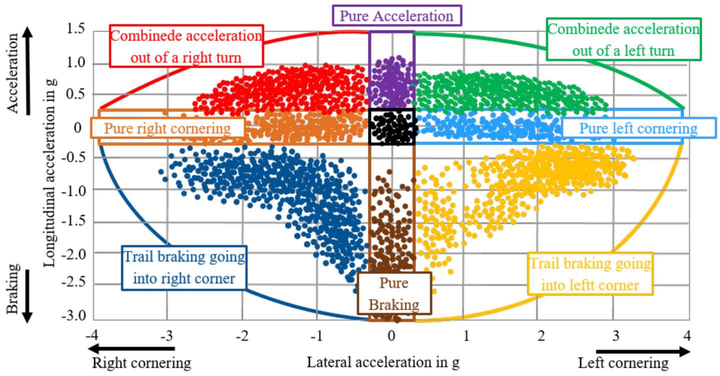


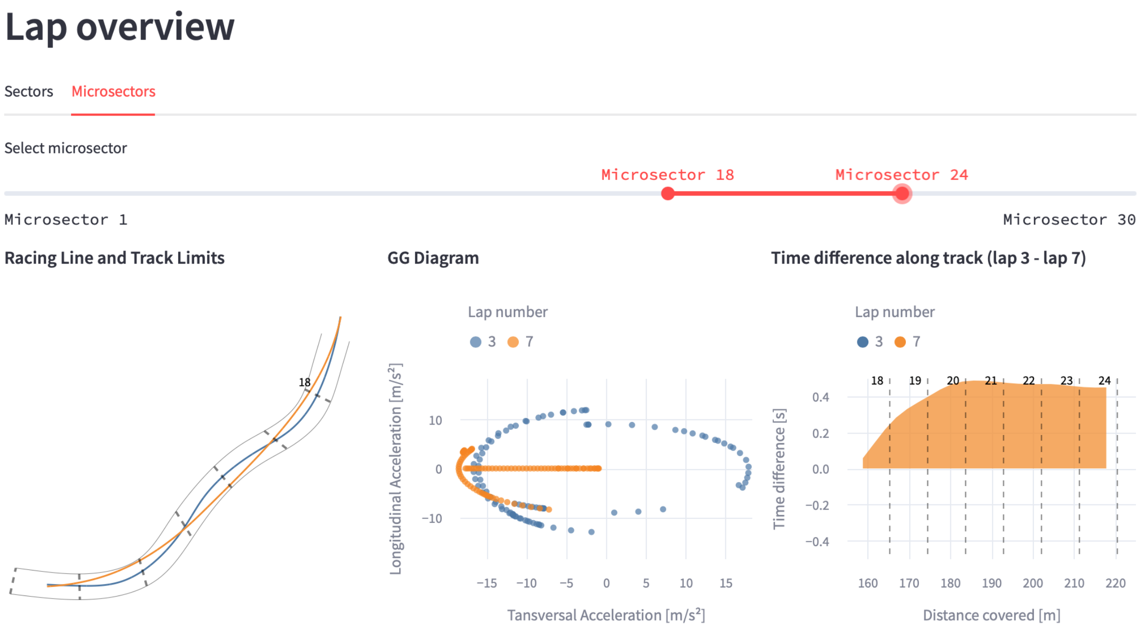
Figure 20: Diagram extracted from Betz *et al.* article[3] explaining the different zones of a GG diagram.

As mentioned in Section 2.3 the question of which driver pushes the limits of the car more is quite abstract and the response is linked with the accelerations profile. It was also mentioned, in Section 5.2 in this case, that the accelerations profile is a very dense plot and that needs to be careful with the amount of data displayed. Therefore, we will visualize this response with the racing line when sectors are selected. Moreover, when a lap is selected it will only print that lap's data and when multiple laps are selected it will display all laps' data distinguished by color. The fact that the chart easily becomes very dense and difficult to interpret supports the idea of limiting the number of laps that can be selected.

We have talked about different options of the chart but not have explained itself. The visual response to the question will be a plot of different drivers' accelerations data known in the motorsport sector as GG diagram. It consists in plotting all the longitudinal accelerations versus the transversal ones registered by the car sensors during the sector as a scatter plot.

The diagram zones can be related with certain combinations of controls as it is explained in detail in the article proposed by Betz *et al.*[3] and also plotted in their chart in Figure 20. For example acceleration points near the positive X axis will correspond to pure left cornering or points near the negative Y axis will be related to pure braking. These zones are easier to identify if we plot the GG diagram near the racing line chart like in Figure 21. In this Figure we can understand that the orange driver is capable to go through three corners with one curve in the racing line which means keeping the accelerations to the same side of the car. On the other hand, the blue driver needs to do three curves in the racing line and this way it needs to shift the accelerations from one side of the GG diagram to the other.

Moreover, the further a point is from the origin of coordinates, the more acceleration is the car receiving and so the more their limits are pushed. This implication makes the GG diagram a good answer for the **Requirement 10**.


Lap overview

Select microsector: Microsector 18 – Microsector 24

Figure 21: Lap overview panel for simulated circuit *TILK-E:3412*, laps 3 and 7 and filtered from Microsector 18 to 24.

The *Lap overview* (Figure 21) as a comparison panel gives a big amount of information to both the team and the drivers. It tells the team that the orange driver is capable to choose a better racing line which allows to gain $0.4s$ (a significant difference in motorsports) to the other driver in just three turns. Apart from that, it tells the team that the orange driver is capable to push more the car to its limits (as seen in the GG Diagram). Finally, it also tells the team that if the orange driver is able to go through three corners with only one steering wheel turn, he might have a better cornering ability. On the other hand, this *Lap overview* section tells the blue driver that the car limits can be pushed more and that the three corners can be surpassed with a more efficient racing line. In a more subtle way, the orange driver can see that in the last microsectors of this range he loses a bit of time advantage (as seen in the time difference chart). It can mean that the blue driver goes out of the last turn with more velocity and in an hypothetical long straight after the last turn the blue driver might gain advantage. With the gradient proposal of Section 5.3.3.2 we would have been able to observe in this charts if this turn exit velocity hypothesis is true. However, we are still able to observe it in the *Run overview*'s velocity at the exit of the turn radar chart (shown in Figure 13).

5.4 Unanswered questions

Despite having been able to answer the vast majority of questions, not all the questions have been answered yet. This has been due to both technical problems and not finding the ideal way to answer that question.

More consistent driver

"Which driver exhibits more consistency?"

The last question to answer is about the consistency of the drivers. Our first approach was to provide for a metric measuring the variances of different values used in the application (such as laptime, braking points, harshness, distance to the track limits, etc.). However, after discussing this metric with some of the drivers of the team, we concluded that the key variation to track was the consistency to push the limits of the car. It has been already exposed that our way to answer to this ability was with the GG diagram, so we tried to find the empirical limits of the car. We tried to estimate the convex hull of all the GG diagrams to then plot it and calculate for each lap the mean distance of all the points to the convex hull. The estimation of this convex hull adapted to the selections of the dashboard is not trivial and this idea was left behind.

Despite that, you can get a sense of consistency for some variables displayed:

- For laptimes, all drivers' different time laps are displayed in the table of the sidebar and are also plotted with a different shape for each driver in the Y axis of the harshness charts.
- For the driving style consistency you can observe the dispersion of the points with the same shape -corresponding to a same driver- in the harshness charts.
- You can do the same approach for braking points in the distance before braking charts.

However this is not enough for us and we consider the **Requirement 9** not achieved and something to work on in the future work.

5.5 Final application design

As mentioned, after all the iterations of each individual chart the final visualization will result in the vertical concatenation of the Run overview and the Lap overview. The application's final version can bee seen in Figures 22, 23 and 24. The parts of the application can be distinguished by their title and the selectors menu can be found in the side bar.

The final decision is to set a limit of maximum selected laps at two. Even when laps are selected, every lap will have a color assigned. This way, every lap will be plotted with the same color in every chart. This coloring measure will ensure that we meet the **Requirement 12**.

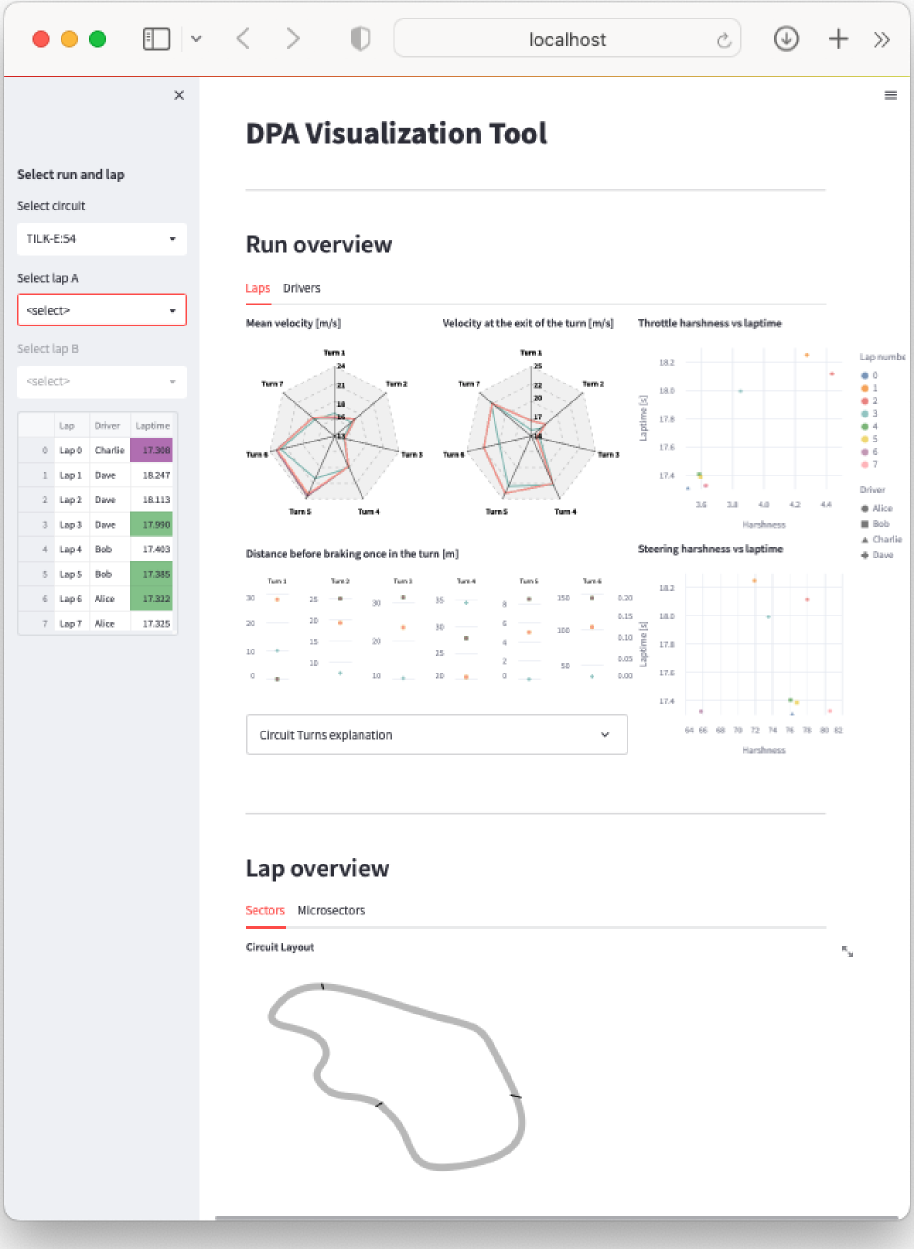
All the code of this application[8] is open and accessible at this GitHub repository. We want the software to be licensed, however, we are not sure of which license to use yet. It will be defined soon and applied to the repository, but it will not be done before the presentation of this project.

---

[8] Data simulation algorithms are not considered part of the application. More specifically, TILK-E[4] is public but GRO[2] is not.

The different available views are the following ones:

- When no lap is selected:
  - **Laps view**: Display a lap-detailed version of the *Run overview* panel, as shown in Figure 22.
  - **Drivers view**: Display a driver-detailed version of the *Run overview* panel.
  - **Lap overview**: the view can be switched among sectors and microsectors but only the circuit layout with the selected sections marked is displayed.



Select run and lap

Select circuit

TILK-E:54

Select lap A

<select>

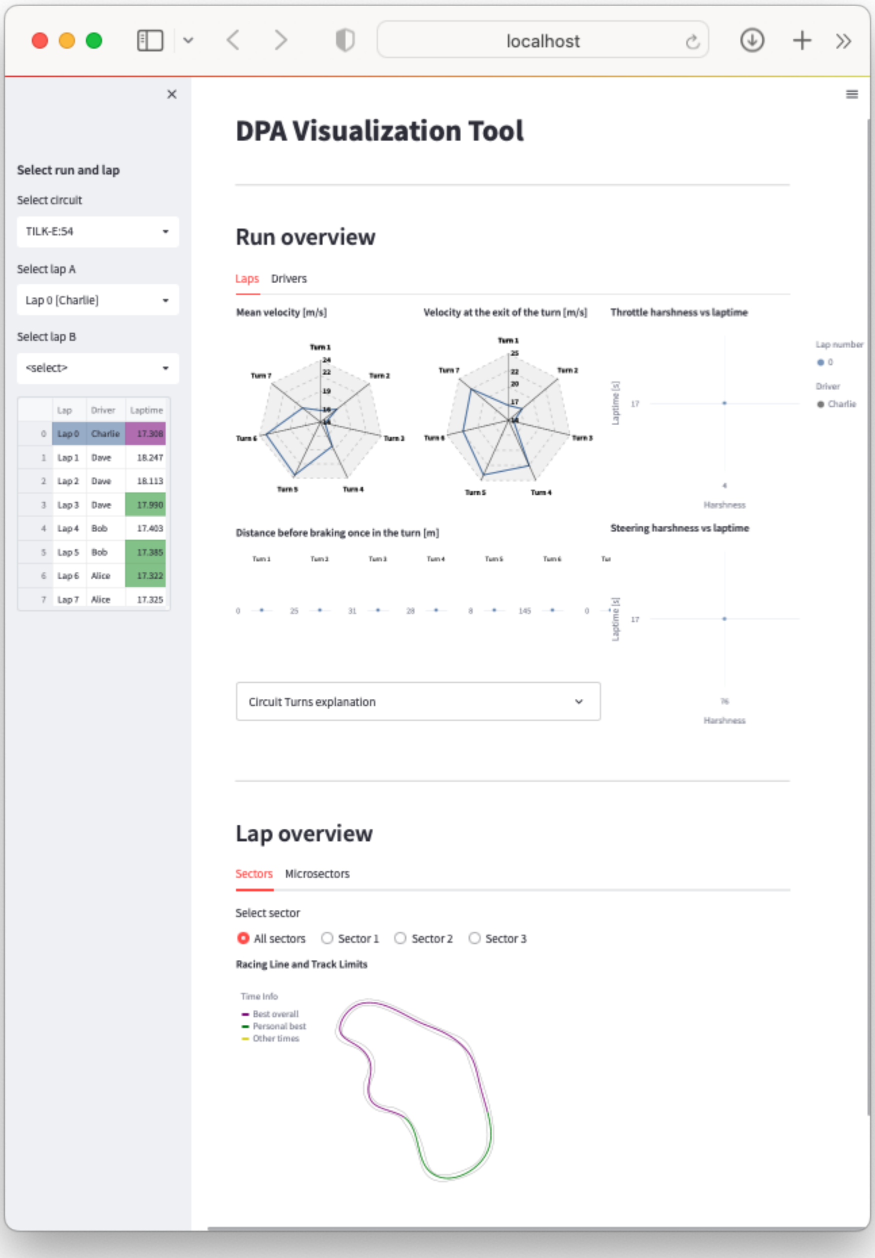
Select lap B

<select>

	Lap	Driver	Laptime
0	Lap 0	Charlie	17.308
1	Lap 1	Dave	18.247
2	Lap 2	Dave	18.113
3	Lap 3	Dave	17.990
4	Lap 4	Bob	17.403
5	Lap 5	Bob	17.385
6	Lap 6	Alice	17.322
7	Lap 7	Alice	17.325

Figure 22: Application's complete design for *Run overview*'s *Laps* view without selected laps.

- When a lap is selected
    - **Run overview**: *Laps* view is filtered by the lap selected and *Drivers* view is disabled.
    - **Lap overview**: the view can be switched among sectors and microsectors. Both show the lap's racing line colored by the time code of that section of the track. Sectors and microsectors can be filtered individually and in a range mode respectively; when filtered, they zoom the racing line and show insights about the ability to push the limits of the car. This view can be seen in Figure 23.



Select run and lap

Select circuit: TILK-E:54

Select lap A: Lap 0 (Charlie)

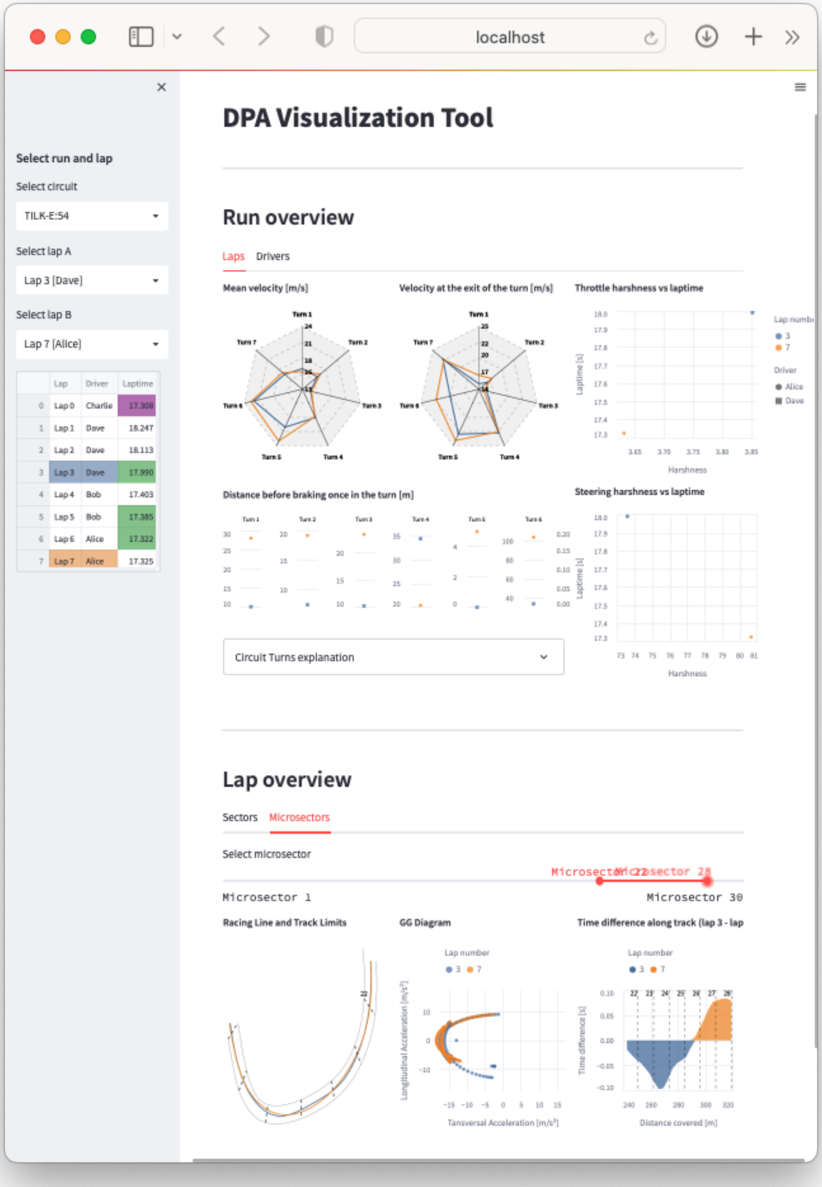
Select lap B: <select>

	Lap	Driver	Laptime
0	Lap 0	Charlie	17.308
1	Lap 1	Dave	18.247
2	Lap 2	Dave	18.113
3	Lap 3	Dave	17.990
4	Lap 4	Bob	17.403
5	Lap 5	Bob	17.385
6	Lap 6	Alice	17.322
7	Lap 7	Alice	17.325

Select sector: ● All sectors ○ Sector 1 ○ Sector 2 ○ Sector 3

Figure 23: Application's complete design for *Lap overview*'s *Sectors* view without a lap selected and without any sector selection.

- When two laps are selected:
  - **Run overview**: *Laps* view is filtered by the laps selected and *Drivers* view is disabled.
  - **Lap overview**: the view can be switched among sectors and microsectors. Both show the circuit layout colored by the fastest lap for that section of the track next to the complete time difference evolution chart for the circuit. Sectors and microsectors can be filtered individually and in a range mode respectively; when filtered, they show the section zoomed racing lines of both laps, insights about the ability to push the limits of the car for each lap and, finally, the time difference evolution chart for that section. It is shown in Figure 24.



DPA Visualization Tool

Select run and lap

Select circuit: TILK-E:54

Select lap A: Lap 3 [Dave]

Select lap B: Lap 7 [Alice]

	Lap	Driver	Laptime
0	Lap 0	Charlie	17.308
1	Lap 1	Dave	18.247
2	Lap 2	Dave	18.113
3	Lap 3	Dave	17.990
4	Lap 4	Bob	17.403
5	Lap 5	Bob	17.385
6	Lap 6	Alice	17.322
7	Lap 7	Alice	17.325

Run overview

Laps Drivers

Circuit Turns explanation

Lap overview

Sectors Microsectors

Select microsector: Microsector 22 – Microsector 28

Microsector 1 Microsector 30

Figure 24: Application's complete design for *Lap overview*'s *Microsectors* view with two selected laps and a range of microsectors selected.

6 User study

To analyze the usability of the visual analytics tool we have developed, we conducted a user study. The objective of the study is to analyze how easy it is for participants to respond to the tasks and questions defined in Section 2.3. Additionally, we aim to evaluate the overall usability and user-friendliness of the tool in a general sense.

6.1 Subject recruitment

The tool we have developed is specifically designed for a narrow sector and a specific task. For this reason it would be a mistake to recruit a wide range of subjects for evaluation. To ensure a proper evaluation, we need to target domain experts expected to use our application. In our case, as explained in Section 2, we have two types of users that will fit the application domain: the team members in charge of choosing the drivers and the drivers themselves.

With the given conditions, we have a total of six participants from BCN eMotorsport who are able to participate in our user study. These participants include: the two individuals responsible for selecting the drivers, two experienced drivers whose judgment is trusted in the driver selection process, and the other two drivers who have been selected for this season. However, it is important to note that all these participants work with me in the team and their responses can be biased. It is important to take this into account when analyzing the results.

6.2 Experimental Procedure

In this section, we explain the experimental protocol that the experimenter has to follow upon participants' arrival and the resources that will be used.

When a participant begins the experiment, they will be provided with a computer that has the visual analysis tool already launched and all the necessary data loaded. Additionally, the participant will receive a questionnaire, which they will be asked to open on a mobile phone or an alternative computer to allow them to respond while working with the tool.

The complete questionnaire is designed to take approximately ten minutes to complete and is divided into six parts:

1. **Consent form**: Firstly, the consent to take part of this case study process is required.

2. **Tutorial**: with the consent given, the entire visual analysis tool is explained. All their parts and uses are presented to the user with the following tutorial. This first part of interaction is thought to have a duration of 2 minutes.

   > *To start working with it, you can firstly interact with the selectors panel in the sidebar. There you can select the circuit you want to work with, for example circuit 54.*
   > *If you move to the "Run overview" panel of the main dashboard you can analyse the cornering ability with the three charts at the left and the driving style at the right. Each of these analysis can be performed over each one of the laps or over the drivers' average with the tabs above the plots.*
   > *If you want to know more about a lap, you can go back to the sidebar and select it in the "Lap A" dropdown (for example lap 0). Now on the "Lap overview" panel you can see more information about the lap, first of all you can select to work with sectors or with*

microsectors with a tab. Focusing on sectors (but the same applies to microsectors) you can first see the entire circuit and the lap racing line colored after the time of each sector compared with the rest of the times of the run. Then, if you select any of the sectors (or a range of microsectors) you can analyse the ability of the driver to push the car to their limits.
Lastly if you want to compare the lap with another one, back in the sidebar menu, you can select the second one in the "Lap B" dropdown -either of the same driver or from another- (for example lap 7 from Alice). Now, also in the "Lap overview" panel you can see the fastest driver by sector (or microsector) and the time difference evolution throughout the lap. If you select a sector (or a range of microsectors) you can compare both racing lines, both GG diagrams and also the time difference for that part of the circuit.

3. **Suggested exercises**: In order to make the user more used with the tool and their uses, different exercises related with all the requirements (Section 2.3) are recommended. They are detailed below and the expected time to solve all the suggested exercises -by a domain expert- is 6 minutes.

   (a) *Analyse each driver's driving style tendencies for circuit 27.*
   (b) *In the same circuit, which lap breaks later for turn 4 and which one exits with the higher speed the 6th turn?*
   (c) *For laps 2 and 3 in circuit 4 between microsectors 24 and 30 analyse the racing line differences, are the differences also reflected in the way both laps push the car to its limits?*
   (d) *Detect if any of these two laps has been upfront throughout all the lap.*
   (e) *Where would you focus on to improve Charlie's fastest lap in the circuit 3412?*
   (f) *Which driver would you select to run in circuit 6811?*

4. **Tool usage questionnaire**: This part consists of a Likert-scale[9] survey asking about the capacity to answer the raised questions of Section 2.3 with the statements below. The response to this questionnaire should not take more tan a minute.

   - *I think cornering ability is evaluable and comparable with the available charts*
   - *I believe it was easy to observe how much drivers push the limits of the car*
   - *It has been easy for me to understand in which sector is faster than another*
   - *I find easy to understand if a sector time is the best*
   - *I find the sectors and microsectors selections useful to analyse racing lines*
   - *It has been easy for me to detect differences in driving styles*
   - *It has been easy for me to evaluate which driving style was better for each circuit*
   - *I believe the entire dashboard is coherent*
   - *I think that selectors are user-friendly and intuitive*
   - *I find easy to understand how the difference between drivers evolve during a lap*
   - *I think racing lines can be properly evaluated and compared*
   - *I think I would know where to focus if I wanted to improve a lap time*
   - *I believe I would be able to decide which driver is better with the data in the dashboard*

---

[9] The Likert scale is a commonly used response scale for measuring attitudes or opinions in surveys or studies. It involves statements or questions with a response scale ranging from *strongly disagree* to *strongly agree*. More information on the Likert scale is available here.

5. **General usage questionnaire**: Another Likert-scale survey is presented to the user, in this case the questions asked are adaptions of the System usability scale (SUS) questions. With the responses we will be able to have a global opinion about the general usability. As the previous questionnaire it should be answered in a minute.

6. **Satisfaction and feedback**: Finally users are asked about the difficulty of the exersises proposed, the usability and satisfaction compared to the GEMS tool[10] and any feedback about the application.

6.3 Results

In this section, the results of the user study will be explained and analyzed. Each questionnaire will be commented and finally we will give a global overview.

6.3.1 Tool usage questionnaire

The thirteen statements about the tool usage -detailed in the 4th step of the procedure in the previous section- are related directly with the requirements (Section 2.3). All the statements of this questionnaire are positive, which means that if the subjects agree with them, the tool is accomplishing what it is expected to solve. The study subjects' answers are detailed in the following Figure:

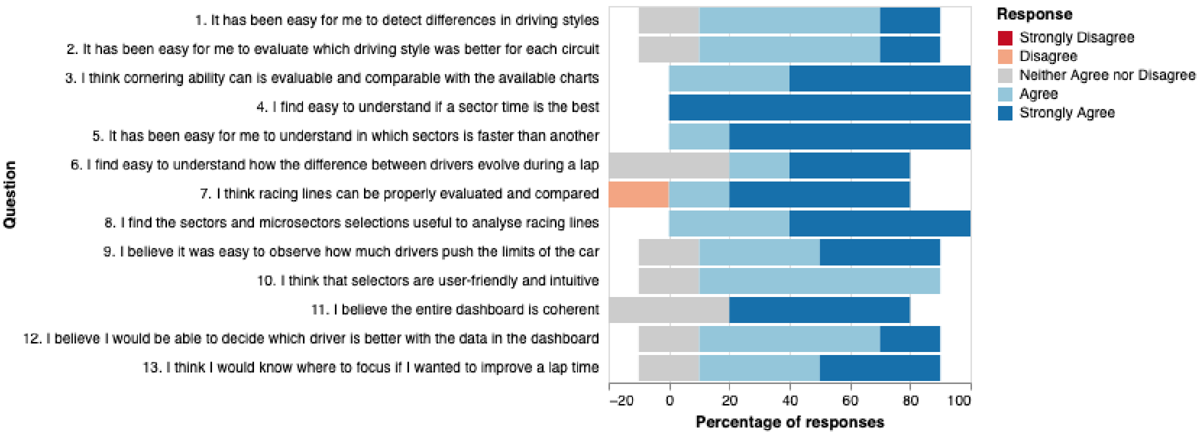


Figure 25: Percentage of responses for each answer and question in the tool usage questionnaire.

We can observe that the majority of the answers are agreeing with the statement or neutral, which is positive for us and gives us a first impression that the tool is properly solving the requirements.

When analyzing concrete statements, we found out that the only statement with a disagreeing answer was the number 7 in Figure 25 which states: *"I think racing lines can be properly evaluated and compared"*. This makes us think that maybe **Requirement 4** is not completely achieved and we need to improve their visual answer. On the other side, we realised that the 4th statement about the capacity to understand if a sector time is the best has received a 100% of "Strongly Agree" answers. This can be biased because of, as mentioned, the color code used is extended

---

[10] The current driver performance analysis tool used by BCN eMotorsport explained in Section 1.1.2

in motorsports sector. However, it encourages us to think that **Requirement 3** is met and representations as the colored table (Section 5.1.3) or the one in Figure 19 communicate properly the information.

Other answers to remark are the ones for statements 6 and 11 of Figure 25 where the percentage of neutral answers is bigger than for the rest of statements. These two statements ask about **Requirement 2** and **Requirement 12**. It seems so that the evolution of the time difference chart that we have made (represented in Figure 16) could have been better and that the entire application coherence can be improved. On the contrary, statements 3 and 5 do not have neutral answers and the responses are divided among "Agree" and "Strongly Agree". Statement 5 is related with the sectors coloring and the **Requirement 3** as the 4th one (which received only "Strongly Agree" responses). It reinforces the idea of having done a good job with it while reminds us that extended visualizations are more communicative to the users than new ones. Is for that reason that we are so surprised that the 3rd statement received only positive answers. This statement asks about the cornering ability charts -responding to **Requirement 6**- which we have extensively redesigned (Section 5.2.1).

The last two statements were thought to ask for the necessities described in sections 2.1 and 2.2. Their answers are meanly positive so we can understand that the application is fulfilling both needs.

The rest of the statements have a bigger percentage of "Agree" answers and smaller ones of neutral and "Strongly Agree" ones which makes us validate the rest of requirements.

6.3.2 General usage questionnaire

The statements for this questionnaire are ten standard statements used for SUS analysis for different kinds of user interfaces. This general usability analysis will help us to locate the user-friendliness of the application. Not all the statements of this questionnaire are positive, as it happened in the previous one. In Figure 26 below the answers are detailed and distinguished by if they are positive or negative.

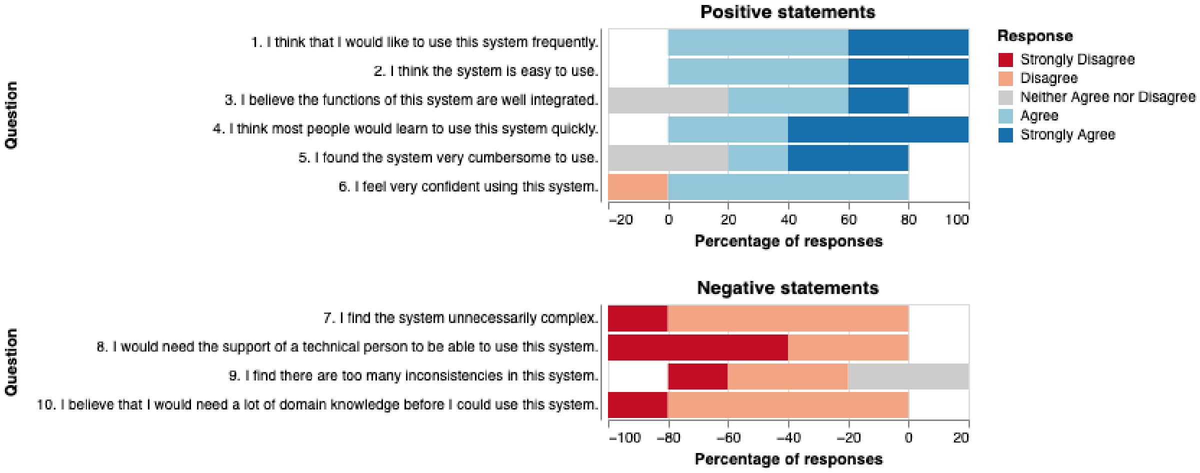


Figure 26: Percentage of responses for each answer and question in the general usage questionnaire.

At first sight, it is remarkable that the positive statements tend to have agreeing responses and the negative ones, disagreeing. This pattern can tell us that our application is user-friendly.

More specifically, we can highlight three statements for their answers:

- First, we want to remark the 6th statement because it is the only statement to receive a adversary answer (disagreeing for a positive statement or agreeing for a negative one). It states *"I feel very confident using this system"* and the "Disagree" answer can show us that an user (the 20% of our domain experts) does not completely trust our application.
- Secondly, the fourth statement is the one with more "Strongly Agree" answers. It is the statement saying that they think people would learn to use the system quickly and we can understand that they are all domain experts and may think about the ease to learn how to use the system by another domain expert. For this reason, we can be satisfied to have created an application for domain experts that they think it is easy to learn to use, so it is intuitive and easily communicates.
- Finally and on the contrary, the 8th is the statement with more "Strongly Disagree" responses. It goes on the same line as the statement whit more "Strongly Agree" because it states: *"I would need the support of a technical person to be able to use this system"*. This result reinforces the believe that we have built an intuitive and communicative application.

6.3.3 Satisfaction questionnaire

The statements for this questionnaire are different from the other ones, there are no positives nor negatives per se. We wanted to evaluate three different aspects: the difficulty of the exercises, if the application improved GEMS (Section 1.1.2) and if they would like to use the application for the team as the standard visual analysis tool.

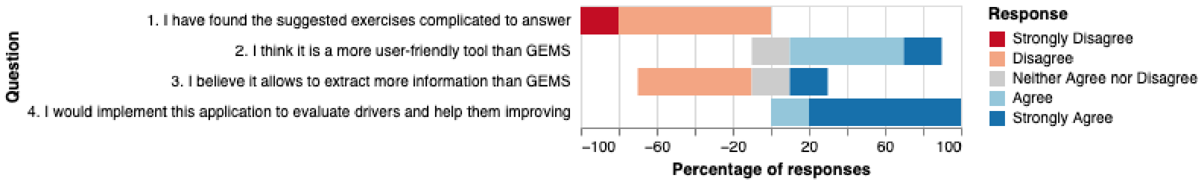


Figure 27: Percentage of responses for each answer and question in the satisfaction questionnaire.

As we can see in Figure 27's first question, the exercises were not found difficult by the testers. They were usual tasks for them so it has been reflected in how easy the problems have been found.

When comparing with GEMS' tool, in the second and third questions of the results' chart we can see that there is no clear decision. While it seems that the tested users think of this application as a more user-friendly framework, some users think that this application can not contribute with more information than GEMS.

Finally, almost every one strongly agree in implementing this tool in the team, the exception is only a subject who just agrees. This are good news as it means that the satisfaction of the testers with the tool is high.

6.3.4 Global overview

In more general terms we can say that the results of the User Study are very satisfactory. The application was liked by the domain experts who apart form responding good to the satisfaction questions, almost all of them asked about the possibility of reading real data to use this tool when the car is ready. It has to be said, again, that the subjects of the user study were my friends and teammates. For this reason, the good results of the first questionnaire (Section 6.3.1) have to be taken cautiously. On the other hand, the statements with lower agreement clearly need to be redesigned in future work. The results may also be biased for the reason of having a small test population. It may be interesting to repeat this experiment with more people related to FS competitions. This will probably modify the results of the SUS questions and it would give us a better understanding of where we stand in terms of user-friendliness.

Finally, we have also received feedback from some of these users whom took part of the user study. Usually these kind of feedback is discarded as it is not representative when comes from wide subjects' ranges, but in our case, with domain experts, we though that at least it was interesting to mention. Both comments received were related to the time differences chart (Figure 16). The first one stated that it would be interesting to be able to track velocity differences versus time differences. The second one suggested a marker in the racing line chart synced with the mouse hovering in the time differences chart to be able to know where in the track are drivers gaining/losing advantage. These suggested charts will be analyzed in future works, where they will be designed and it will be decided whether they are added to the application or not.

7 Conclusions

7.1 Achieved objectives

The necessities described in Section 2 have been covered by achieving most of the requirements in Section 2.3.

Requirement 1 *We must know the different laptimes as an evaluation criteria, but also be able to compare them.*
Laptimes are fixedly displayed and colored to be able to succesfuly compare them.

Requirement 2 *We must be able to compare the differences of time throughout a lap.*
The evolution of time differences between two laps is available. However, this have not been the most easy-to-understand visualization for the testers.

Requirement 3 *We must know the times for each sector or microsector, be able to compare them and know which are the best ones.*
Sector (or microsector) time comparison charts are displayed with the most variety depending on views and have been one of the most liked charts by the testers.

Requirement 4 *We must be able to compare different racing lines over the same circuit to analyze them by their own and to make comparisons among them.*
Racing lines are visualized precisely over the circuit, despite that, it has been the worst punctuated chart by the user study testers.

Requirement 5 *We must be able to zoom the racing line visualization to specific sections of the circuit such as sectors, microsectors or turns.*
Racing line charts are available for single sectors and for ranges of microsectors.

Requirement 6 *We must be able to know where does each driver brake for each turn and compare it with the braking points of other laps in that turn.*
The visualizations to communicate the drivers' cornering ability are the ones that have evolved the most. The domain experts that tried the application agreed in the fact that they allow to evaluate and compare the cornering skills among drivers.

Requirement 7 *We must be able compare throttle harshness and smoothing harshness, and evaluate how they affect to the laptimes.*
Harshness charts are successfully built and shown in the application.

Requirement 8 *We must be able detect each driver tendencies in terms of driving style.*
Driving styles are interpretable and differentiable with the charts built.

Requirement 9 *We must be able evaluate the consistency of a driver.*
The evaluation of the consistency is not achieved to the desired extent.

Requirement 10 *We must be able evaluate how a driver is pushing the physic limits of the car and compare it to other drivers and laps.*

With the GG diagrams we have successfully shown the drivers ability to push the limits of the car in each lap. They also allow the users to compare the push of the physic limits in two different laps.

Requirement 11 *We must be able to select which circuit we want to work with, which lap we want to focus in or which laps we want to compare.*
Selectors for circuits (runs), laps and lap sections (sectors and microsectors) have been implemented and included to the application. They contributed to increase the user-friendliness which has been recognised as a big step forward by the testers.

Requirement 12 *Facilitate comparison and relations between information in different charts.*
Despite having synchronized all charts for different views and in terms of colors, coherence has not been one of the adjectives in which testers agreed more.

7.2 Discussion

In order to provide for a data-driven system to help evaluating and comparing drivers for a Formula Student team, we have developed a two-parts application that gives a visual analysis system to solve these tasks.

Run overview gives a global view with information about the laptimes, cornering ability and driving style for all the drivers and laps. The domain experts that tested the application, have agreed in the ease to evaluate and compare both aspects with the available visualizations.

Lap overview gives the user the ability to evaluate and compare lap sections' times, racing lines, capacity of pushing the cars' physic limits and time differences evolution of the different laps. These can be single-selected to evaluate them or selected in pairs to compare the laps in pairs. This part of the application has visualizations claimed to be very communicative by the domain experts, as the ones showing the lap sections' times, but also visualization that can be improved to communicate better as the racing line one.

As said, both have evaluating aspects which help drivers to understand how they are performing and how can they improve, and also both parts provide for comparison charts to allow the team to understand which drivers are performing better.

7.3 Future work

We are very satisfied with what we have built, despite that there is always gap to improve. For example, a 20% of the user study subjects have claimed to not feel confident using our system and our application needs to transmit confidence. It is important to detect why this happens and solve it for future versions. Some aspects that it has been detected that can be improved are the following ones:

- Firstly, driver consistency goals (**Requirement 9**) have not been achieved. As explained in Section 5.4, difficulties have been found and a chart answering the consistency question has not been developed. In next iterations of the application new research can be done to find or design a proper visualization for the consistency.

- Racing lines charts are the ones less agreed to communicate sufficiently by the user study individuals. It suggest us to redesign them focusing in the ease to evaluate and compare racing lines. The zooming capacity for sectors and microsectors has been agreed to be useful so it could be the starting point for next developments.

- The evolution of the time difference throughout a lap is tied for the most neutral answers which could indicate us that it could be improved. If we also take into account the feedback provided by the domain experts, all the suggestions were for this chart so it could be interesting to redesign it.

- Besides having been designed and built for real car data, the application has not been tested with complete real datasets. For this reason, when the car starts rolling and when drivers are able to show their ability, the application needs to be tested and adjusted because the main goal of this project is to be able to help with real data in real situations.

- The user study would be more representative with a larger range of domain experts and it would be also more truthful with people linked with other Formula Student teams not knowing and working with me. This would give us other insights of how our application works and this way we could keep improving it.

7.4 Relevant data science background

This project suppose the end of my Bachelor Degree in Data Science and Engineering at UPC. The knowledge acquired during this time has been really important for the execution of this final project. Among what I have learnt, I would like to highlight the following aspects:

- **Information Visualization:** The knowledge received from this subject has been the most used in this project. Not only the technical knowledge of Altair, but also notions of how to be communicative and how to be able to make the users understand the data with their less effort and our less ink-waste.
- **Signal Processing:** The different subjects of this branch of my degree have provided me for a way to work and understand that helped me when dealing with car sensors and their data.
- **Algorithmics and Programming:** the three levels of this subject have trained me to code clean and tidy. This training with the knowledge of Object Oriented Programming have helped me to build a better code, which was easy to adapt or refactor while the application was evolving. One of the key moments when I realised of the importance of these habits was when we moved from real data to simulated data. The way of how the visualization classes were agnostic of the building of the data classes made possible to generate new data classes which were perfectly read by the visualization ones.
- **Engineering Projects and Internships:** Both, the Engineering Projects subject and the internship I did during my degree have given me a critical view to detect necessities of people -other than data scientists- working with data. This detection capacity and the aim to effectively address user needs without getting lost in technological extravagances have been crucial to achieve useful results for the project.

References

[1] *GP Arabia Saudí: Previo Carrera*. DAZN F1, Mar. 2022. TV program.

[2] Albert Rodas Barceló. *La traçada ideal per a un Formula Student Autònom*. Master's thesis, FME - Universitat Politècinica de Catalunya (UPC), Barcelona, Spain, May 2023.

[3] Johannes Betz, Alexander Heilmeier, Alexander Wischnewski, Tim Stahl, and Markus Lienkamp. Autonomous driving—a crash explained in detail. *Applied Sciences*, 9(23), 2019.

[4] Pol Puigdemont Plana. *TILK-E*. `https://github.com/puigde/TILKE`, 2023.

[5] Claude Rouelle and Andrei Plakhotnichenko. *Driver and chassis debrief - Claude Rouelle (OptimumG) | FS Russia webinar*. `https://www.youtube.com/watch?v=bR8mE9UJX2c`, Aug. 2021.

[6] Andreu Huguet Segarra. *LIMO-Velo*. `https://github.com/Huguet57/LIMO-Velo`, 2022.