Phase 1: RFCAR

Product concept, foreseen specifications, planning, tests, initial designed

Master Degree in Industrial and Computer Electronics Engineering
Laboratórios e Práticas Integradas 2

Integrator Project

Group 7

Nuno Rodrigues A85207
Hugo Carvalho A85156

Hugo Ferreira A80665

João Faria A85632

João de Carvalho A83564

José Mendes A85951

José Pires A50178

April 17, 2020

Contents

1 Product concept

The envisioned product consists of a remote controlled car used to assist exploration and maintenance domains, hereby, denominated as Radio Frequency Camera Assisted Rover (RFCAR). To satisfy such requirements, the vehicle must contain a remotely operated camera that provides a live video feed to the user. Additionally, the vehicle must include an odometric system that assists the driving and avoids unintentional collisions when remote control is compromised, e.g., when connection is lost. The vehicle provides means for exploration and conditions assessment in critical or unaccessible areas to human operators, such as fluid pipelines and other hazardous locations.

2 Foreseen product specifications

In this section the foreseen product specifications of the system to be developed are provided. Such specifications were obtained through the intersection of customer, functional requirements and project restrictions.

2.1 Quality Function Deployment

The customer requirements are usually abstract and can collide with the functional requirements, compromising the fulfilment of the project. Thus, it raises the need of a methodology which converts abstract requirements into a series of concrete engineering specifications.

An efficient quality assessment methodology is the use of a Quality House. In this method, the desired requirements are laid out as rows and the engineering specifications/restrictions as columns. In the intersections lies a symbol representing the strength (weak, moderate or strong — Figure 2) of the relationship requirement-specification. This symbol is one of the many tools that permit the quantification of relations existing between the customer requirements and engineering specifications. For instance, the 'engine power' specification and the 'fast' requirement have a very strong correlation (9) since the power of the engine is directly responsible for the speed of the car.

Along with the requirements, the importance given to each is also specified, ranging from 1 (lowest importance) to 5 (highest importance) these, along with the number at each intersection, will be used to calculate the importance of each specification and thus assign priorities for the Design Team.

Lastly, the triangle shape (the 'roof' within the house metaphor) serves as another way of measuring relationships, this time between each specification: such is achieved by placing a symbol (ranging from very negative to very positive, see Figure 1) in the diagonal intersection of two specifications. I.e., the battery life will have a very negative correlation with the battery temperature, due to the fact that the increase of the temperature will cause a decrease in life time. As such a 'very negative' correlation was placed in the diagonal intersection betwixt 'Battery Life' and 'Battery Temperature'.

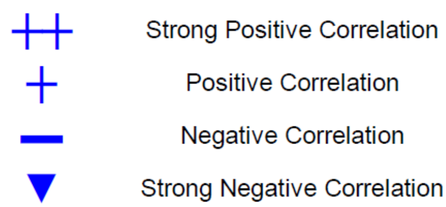

Figure 1: Quality House - Specification Correlation Strength Symbols

Figure 3 shows the 'Quality House' for the RF CAR containing:

- **Customer Requirements**: Vehicle Integrity; Obstacle Avoidance; Reliable Feedback; Fast Response; Fast; Budget Friendly; Low Consumption; Small.
- **Functional Requirements or Restrictions**: Autonomy; Battery Temperature; Minimum Distance to Obstacle; Maximum Velocity; Motor Expectancy; Cost of Production; Motor Power; Ramp-Up Speed Time; Frame Rate; Camera Range; Resolution; Communication Range; Communication Speed; Dimensions; Mass.
- **Intersection Values** (referencing the strength of the requirement-specification correlation) — see Figure 2.
- **Analytical Data**, depicting, in a quantifiable manner, the aims of the project and the relevance of each entity:
  - Target or Limit Value: The metrics the design team will be based on, white spaces are left for either further discussion and refinement.
  - Difficulty: Allows a subjective input to be added so that 'importance' can be changed to balance unforeseen circumstances.

- Importance and Relative Weight: The main conclusion for which the QFD was used, it assigns the priorities for the design team in an objective manner.

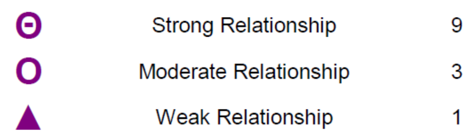

Figure 2: Quality House - Relationship Strength Symbols

With the QFD, the prioritized ranks and specification targets were obtained and diffused within the Design Team with a straightforward guideline. For instance, the low cost requirement should be prioritized over all other specifications, followed by the maximum speed, Ramp-Up Speed Time and so on. On the other hand, the engine expectancy is of little to no consequence (note that the importance added up to a mere 3%), followed by the camera-related specifications. This could be regarded as a point of discussion, which should be prioritized? The functionality of the car or the the feedback provided by the camera?

With the last point in mind, the QFD has the advantage of promoting further discussion, simply by changing the importance of a requirement the priority ranking will change, ergo the priorities can be altered, easily and efficiently, if deemed appropriate.

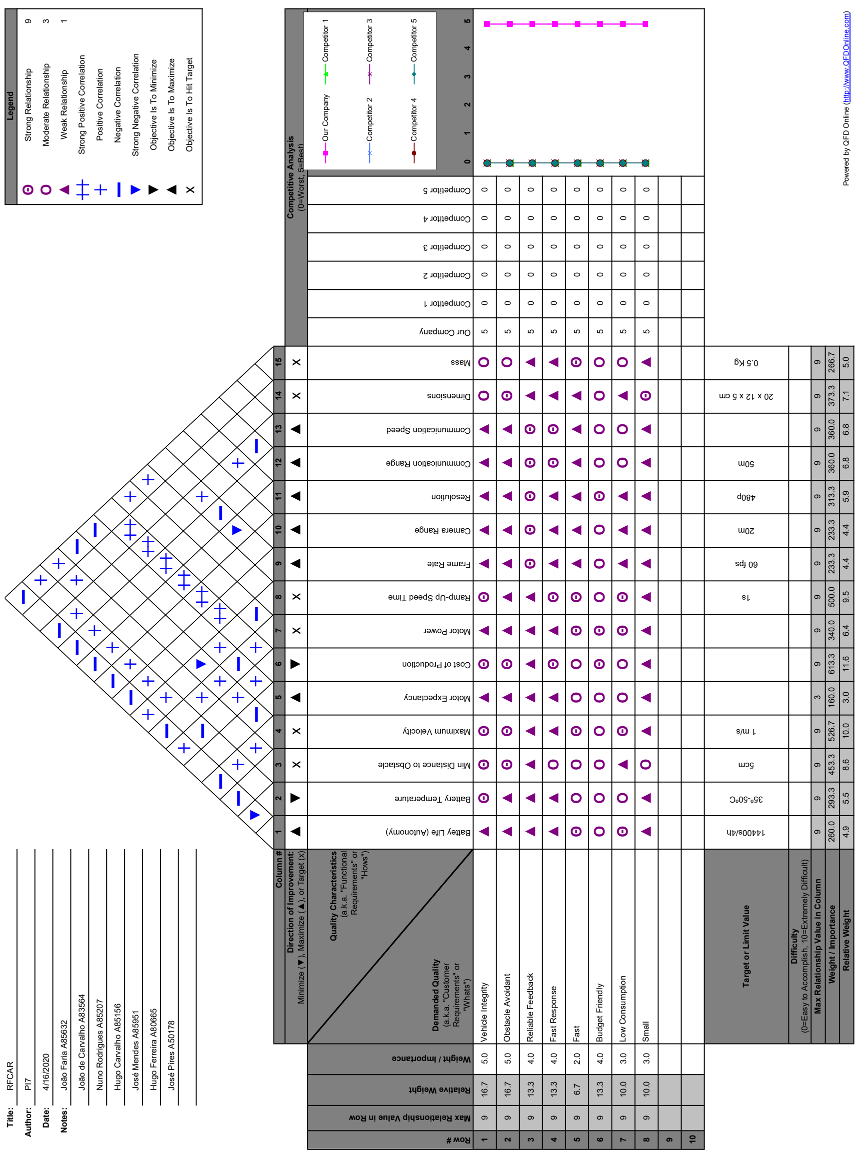

Figure 3: Project Study — RF Car Quality House

2.2 Vehicle Autonomy

The vehicle is operated in wireless mode, thus, a portable power source must be included. The autonomy refers to the time interval between battery fully charged and safely discharged and should be observed for the following scenarios:

- No load and vehicle operating at maximum speed;
- No load and vehicle operating at mean speed;
- Maximum load and vehicle operating at maximum speed;
- Maximum load and vehicle operating at mean speed.

2.3 Speed

The vehicle must be operated within a safe range of speed, while also not increasing excessively the power consumption. Thus, these speed boundaries should be tested in the absence of an external load and in the presence of the maximum load.

2.4 Safety

Vehicle self integrity protection is a requirement in product design, especially considering the vehicle is to be remotely operated. The safety of operation can be analysed in two ways, and considers the preservation of people and goods. For the former, it is important to assure safe interaction as well as user operation — the vehicle may encounter several people along its path, but it must not inflict any damage. For the latter, the vehicle under operating conditions must not inflict any damage to goods. Thus, in the presence of conflicting user commands violating the safety of people and goods, the local system should override them, taking corrective measures to prevent it. The same holds true if the communication between user and system is lost.

2.5 Image acquisition

The vehicle is equipped with a camera to assist in its navigation, thus, requiring it to be fed to the user's platform appropriately.

2.5.1 Frame rate

Frame rate refers to the frequency at which independent still images appear on the screen. A better image quality is the result of a higher frame rate but the processing overhead increases as well, so a compromise must be achieved between the quality of the image and the increased processing overhead required. The minimum frame rate defined must be such that allows a clear view of the navigation.

2.5.2 Range

How far can the camera capture images without loosing resolution and record them. The range must be such that allows the user to see the obstacles when the car is heading to them and provide enough time to change the direction.

2.5.3 Resolution

The amount of detail that the camera can capture. It is measured in pixels. The quality of the acquired image is proportional to the number of pixels but a greater resolution requires a greater data transfer and processing overhead, thus, a compromise must be achieved. The minimum resolution must be such that provides the least amount of information required for the user.

2.6 Communication

2.6.1 Reliability

A communication is reliable if it guarantees measures to deliver the data conveyed in the communication link. As reliability imposes these measures, it also increases memory footprint, which must be considered depending on the case. For the devised product, an user command must be acknowledged to be processed, otherwise, the user must be informed; on the other hand, loosing frames from the video feed is not so critical user can still observe conveniently the eld of vision if the frame rate is within acceptable boundaries.

2.6.2 Redundancy

The communication protocols are not flawless and the car relies on them to be controlled. If the communication is lost, the car cannot be controlled. A possible solution for this issue is using more communication protocols (e.g Wi-Fi and Bluetooth), so when one protocol fails, the car can still be controlled by the other.

2.6.3 Range

The communication protocols have a limited range of operation, and, as such, regarding the environment on which the car is used the range can be changed. The range refers to the maximum distance allowed between user and system for communication purposes.

2.7 Responsiveness

The movement of the car will be determined by the tilt movement of the smartphone. Sensibility refers to the responsiveness of the car on the minimum smartphone tilt movement. The sensibility must be in an range of values in which small unintentional movements will be enough to change the state of the car and it does not take big smartphone tilts for the car to move.

2.8 Closed loop error

The speed, direction and safe distance to avoid collisions must be continuously monitored to ensure proper vehicle operation. The closed loop error must then be checked mainly in three situations as a response to an user command:

- speed: the user issued an command with a given mean speed, which should be compared with the steady-state mean speed of the vehicle.
- direction: the user issued an command with a given direction, which should be compared to the vehicle direction.
- safe distance to avoid collisions: the user issued an command with a given direction and speed which can cause it to crash. The local control must influence, to prevent collision, and the final distance to the obstacles must be assessed and compared to the defined one.

2.9 Summary

Table 1 lists the foreseen product specifications.

Table 1: Specifications

	Values	Explanation
Autonomy	4 h	Time interval between battery fully charged and safely discharged
Speed Range	0.1 to 1 m/s	Speed at which the car can operate
Frame Rate	60 fps	Frequency at which independent still images appear on the screen
Camera Range	20 m	How far can the camera capture images without loosing resolution
Camera resolution	480p	Amount of detail that the camera can capture
Communication Range	50 m	Maximum distance between the car and the smarphone without losing connection
speed Error	5 %	Maximum difference between desired and real speed
Direction Error	5%	Maximum difference between desired and real direction
Distance Error	5 %	Maximum difference between desired and real distance to the obstacle
Dimensions	20x12x5 cm	Dimensions of the car
Weight	0.5 kg	Weight of the car

3 Initial design

Following an analysis of the products family tree (remote controlled cars), the state of the art and the QFD matrix in fig. 3, an initial design of the product itself can be produced (fig. 4). The selected approach was top-down, in the sense that the requirements and specifications were addressed and that resulted in a general diagram of the product concept. Some macro-level decisions were made in this stage to narrow the problems solutions pool, as follows:

- The car itself should be battery-powered, as it is a free-moving object that is intended to work in environments where trailing cables could interfere with its regular movement.
- The device used to control the car should ideally be one already owned by the user, with an integrated screen (e.g. smartphone), as it would make it more affordable and have a more straightforward interface.
- The protocol for communication between the controlling device and

the Rover should be chosen from within the pool of those readily available to smartphones (e.g. Wi-Fi, GPRS) to keep the price of the overall product down and make it as practical as possible.

- The control and communication unit for the car should be divided into two modules: one which can interface directly with the camera module and manage data transmission and reception at the applicational level of the TCP/IP protocol stack, with enough throughput for the specified video resolution and framerate. And another one which can measure and process sensor inputs and control the actuators in real-time.

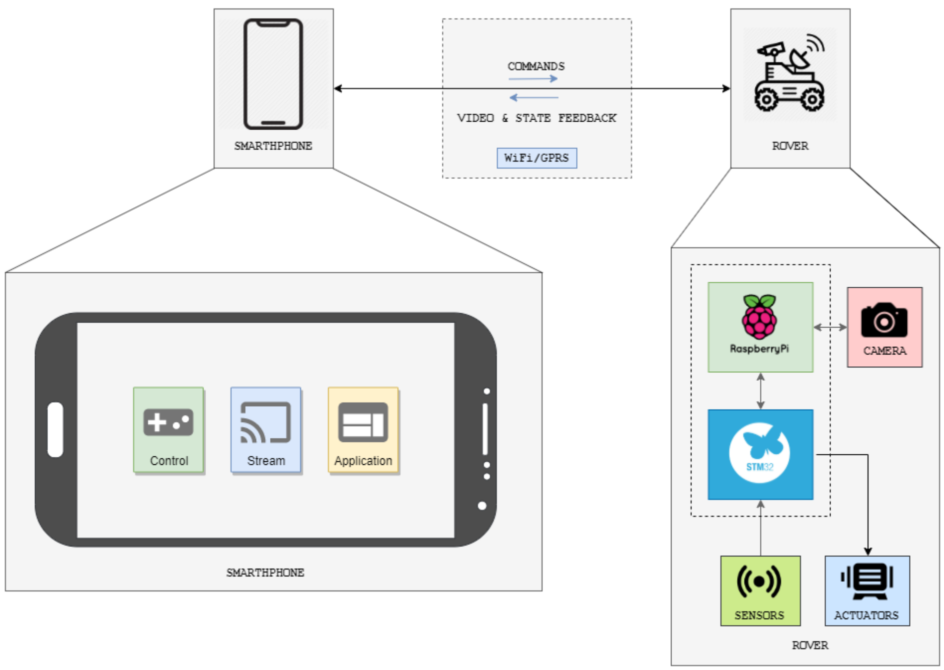

Figure 4: Initial design: Block diagram view

Thus, summarising, the initial design yields the system illustrated in fig. 4, comprised of:

- **Raspberry Pi**: Interfaces with the camera directly, transmitting the information it receives to the smartphone. Receives user commands and sends sensorial information back to it;

- **STM32**: Sends sensorial information to the Raspberry Pi module and receives commands from it. Controls the actuators according to the given instructions and sensor readings;

- **Actuators**: DC Motors that control the cart́s movement and headlights for nocturnal or low light conditions;

- **Sensors**: Odometric sensors that supin this senseport the detection of obstacles and luminosity sensors;

- **Camera**: Device connected to the Raspberry Pi that allows the live stream of the cart́s surrounding environment;

- **Smartphone**: Grant visual feedback from the camerat́s live feed also allowing the user to control the movement of the vehicle intuitively;

Due to the extraordinary conditions imposed by the recently enacted confinement measures, the need rose to create a non-physical connection between both modules of the Rover, which is depicted in fig. 5.

For that purpose, a virtual environment will be created to simulate the Rover's various subsystems. This solution allows for integrated testing without the need to deploy it to the hardware.

The first of said subsystems is the Physical Environment Virtual Subsystem, which simulates the Rover, the physical environment and receives actuator inputs from the second one, the Firmware Virtual Subsystem, for which it generates the sensor readings.

The Interface and Processing Virtual Subsystem interfaces with the computer's Web Camera, which is meant to simulate the onboard camera of the Rover. It also exchanges that information, as well as the Rover's status and the user's commands with the Smartphone module through a Wi-Fi communication channel, which separates it from the non-simulated environment.

Moreover, one redundant connection is established using the GSM network, to ensure proper feedback in case of failure in the Wi-Fi communications. Additionally, one more connection is established between the FVS and the Smartphone to ensure minimal communication between the Rover and the Smartphone in case of IPVS's integral failure.

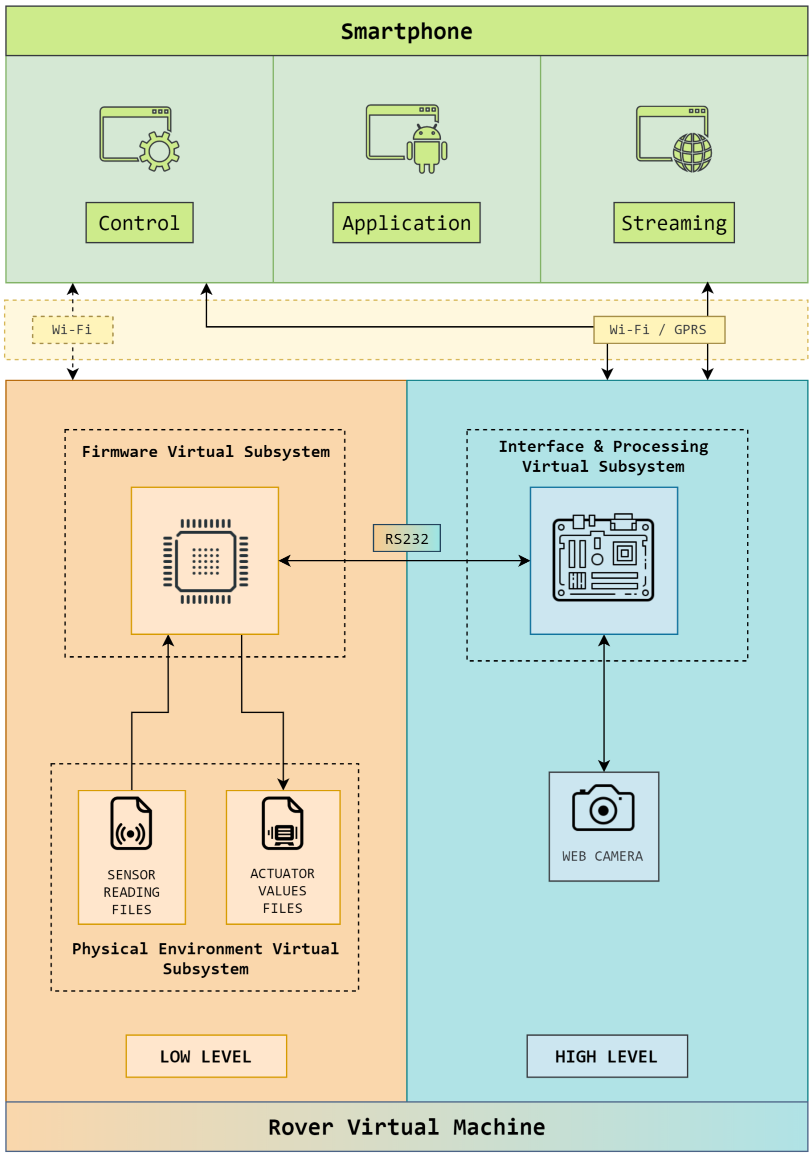

Figure 5: Initial design: Virtual environment block diagram view

4 Planning

In fig. 6 is illustrated the Gantt diagram for the project, containing the tasks' descriptions. It should be noted that the project tasks of Analysis, Design, Implementation and Tests are performed in two distinct iterations as corresponding to the Waterfall project methodology.

Due to unpredictable circumstances, limiting the mobility of team staff and goods, the implementation stage will not be done at full extent, but rather at a simulation stage. Thus, to overcome these constraints, the project focus is shifted to the simulation stage, where an extensive framework as to built to model the system operation, test it, and providing valuable feedback for the dependents modules. As an example, the modules previously connected just by an RS232 link, must now include upstream a web module (TCP/IP) — the data is now effectively sent through the internet, and must be unpacked and delivery serially as expected if only the RS232 link was used.

The tasks are described as follows:

- Project Kick-off: in the project kick-off, the group is formed and the tutor is chosen. A brainstorming about conceivable devices takes place, whose viability is then assessed, resulting in the product concept definition (Milestone 0).

- State of the Art: in this stage, the working principle of the device is studied based on similar products and the system components and its characteristics are identified.

- Analysis: In the first stage — Analysis 1 — contains the analysis results of the state of the art. It should yield the specifications document, containing the requisites and restrictions to the project/product, on a quantifiable basis as required to initiate the design; for example, the car maximum velocity must be, at maximum, `2 m/s`. The second stage — Analysis 2 — contains the analysis of the first iteration of the development cycle.

- Design: it is done in two segments: modules design — where the modules are designed; integration design — where the interconnections between modules is designed. It can be subdivided into *conceptual design* and *solution design*.

  - In the conceptual design, several problem solutions are identified, quantifying its relevance for the project through a measuring scale, inserted into an evaluation matrix, for example, Quality Function Deployment (QFD).

  - In the solution design, the selected solution is developed. It must include the solution modelling, e.g.:
    * Control system design: analytically and using simulation;
    * Transducer design: circuit design and simulation;
    * Power system design: power supply, motors actuation and respective circuitry design and simulation;
    * Communications design: communication protocols evaluation and selection;
    * Software design: for all required modules, and considering its interconnections, at distinct levels:
      · front end level: user interface software, providing a easy and convenient way for the user to control and manage the system.
      · framework level: software required to emulate/simulate and test the required system behaviour, providing seamless interfaces for the dependents modules
      · back end level: software running *behind the scenes*, handling user commands received, system monitoring and control.

- Implementation: product implementation which is done by modules and integrated. Once again, it should be noted that the implementation is mostly done in simulation and coding stages, due to the aforementioned constraints. In the first stage, the implementation is done in a prototyping environment — the assisting framework developed, yielding version alpha; in the second stage it must include the coding on the final target modules, yielding prototype beta.

- Tests: unit tests — by modules — and integrated tests are performed. Tests are generally considered as those performed over any physical component or prototype. Here, it is used as a broader term, to reflect the tests conducted into the system and the several prototypes.

- Verification/Validation: in normal circumstances, after the alpha prototype is built the specifications listed in the analysis must be verified and the prototype validated by an external agent (an external user to the group). Due to abnormal circumstances, the verification must now be performed, not on the physical prototype, but over the chain of modules developed, checking their performance against the specifications listed, i.e., subsystem verification. System verification may be performed to validate overall function, but not for quantifiable measurement, due to the latencies involved. Regarding validation, once again, there is limited access to the physical modules, specially for an external agent, thus, it should be limited to user interface validation.

- Delivery: — project closure encompassing:
  1. Final prototype
  2. Support documentation: how to replicate, instruction manual.
  3. Final report
  4. Public presentation

5 Tests

Tests are generally regarded as those performed over any physical component or prototype. Here, however, a broader sense is used, to reflect the tests conducted into the system and the several prototypes, under the abnormal present circumstances. Moreover, as indicated in the design, the current development strategy encompasses the virtualization of all hardware components, enclosed in a single virtual environment.

Thus, it does not make sense to perform hardware related tests such as velocity measurements, autonomy, safety, etc. As such, the focus is shifted towards software and control verification, encompassing the following tests: functionality, image acquisition, communication, and control algorithms correctness.

The tests are divided into verification and validation tests.

5.1 Verification tests

The verification tests are tests performed internally by the design team to check the compliance of the foreseen specifications. These tests are done after the prototype alpha is concluded.

5.1.1 Functionality

The remotely operated vehicle is composed of several modules distributed along several different platforms, some of which distanced from each other.

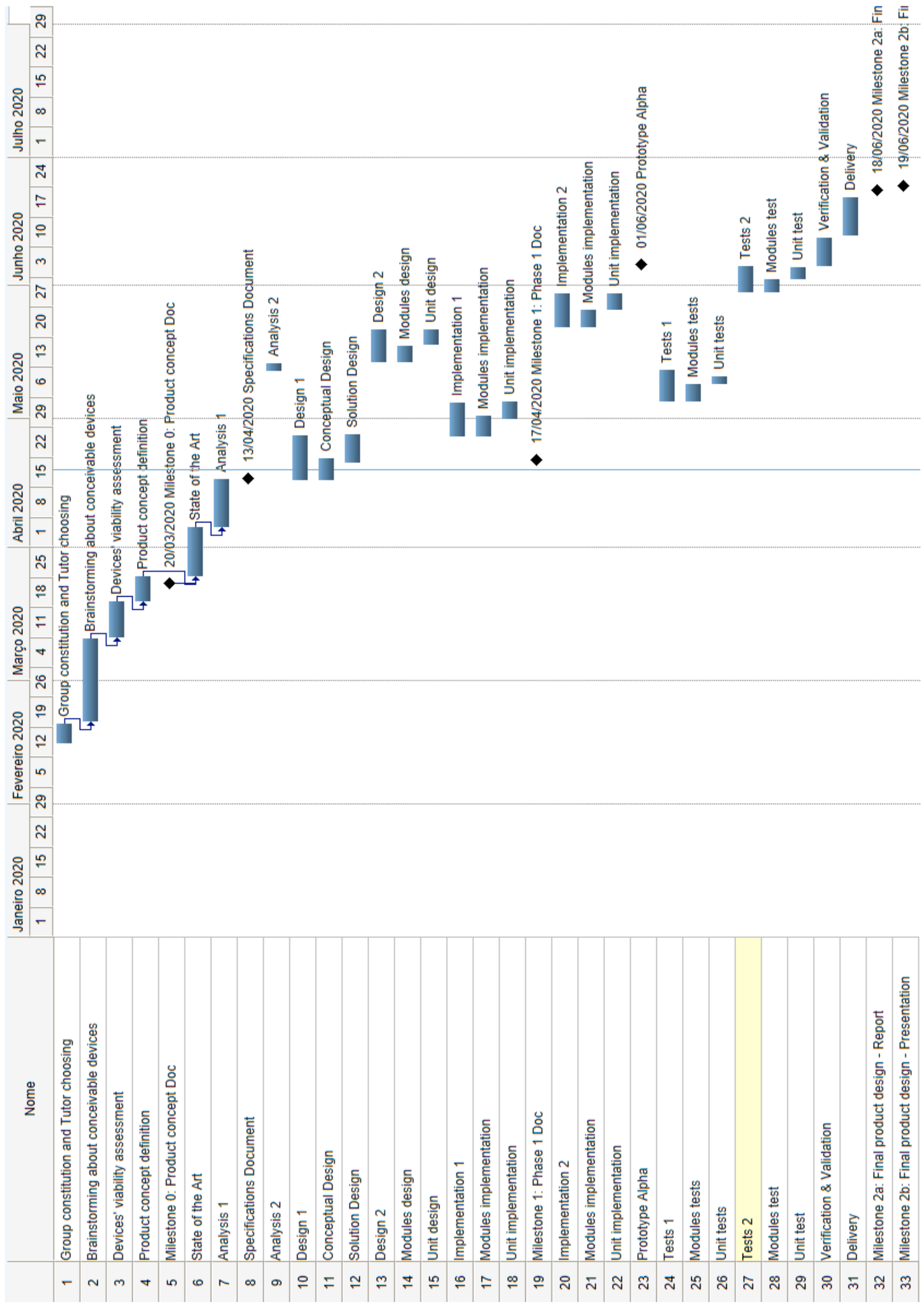

Figure 6: Project planning — Gantt diagram

Thus, the proposed sets of functionalities should be tested in the integrated system, by tracking and analysing the user commands issued along the way until it finally reaches the vehicle (in the virtual environment), assessing if it is correctly processed. For example, if the user issues the vehicle to move to a given place (via smartphone interaction), the message sent to the vehicle must be signalled in each endpoint hit, and the vehicle should move to that place, symbolically detected by the modification of its virtual coordinates.

5.1.2 Image acquisition

The vehicle is equipped with a camera to assist the user in its navigation, thus, requiring the following variables to be tested: frame rate, range, and resolution. In the current scenario, the virtual environment should provide access to a built-in or external camera, with the former being fairly common nowadays in every personal computer, thus, enabling easy testing.

Frame rate To test frame rate, the user screen must be updated with the number of frames received from the camera per second and checked against the defined boundaries.

Range To test camera's range, an object must be captured at increasing distances, until the image resolution is lost.

Resolution The minimum resolution should be tested as providing the least amount of information required for the user, while minimizing data transfer and processing overhead.

5.1.3 Communication

The communication tests are performed in compliance to the specifications provided in Section 2.6, namely reliability and redundancy.

Reliability To test communication reliability, the most critical communication link is chosen, namely the wireless communication between user's platform (smartphone) and vehicle's platform (virtual environment). Then, the communication link and protocol selected are tested by monitoring the ratio of dropped packets versus total packets, using an appropriate tool on both directions for transmission and reception.

Redundancy To test communication redundancy, one communication channel should be turned off, verifying if the communication between nodes is still possible through another communication channel. For example, in the communication between host (user's platform) and remote (virtual environment) systems, the priority communication is performed between host and high-level subsystem. However, if this is turned off, the host must also be able to communicate with the remote system via low-level subsystem.

5.1.4 Correctness of the control algorithms

As previously mentioned, the speed and position must be continuously monitored to ensure proper vehicle operation. Under the current circumstances, it is difficult to properly and realistically stimulate the control loops, as both sensor and actuator data are not available. The solution could reside in the physical modelling of both actuator and sensor (the plant), as well of controller's logic modelling, and perform formal verification using model-checking and finite automata. Later on, those models could be tested using software-in-loop (and hardware-in-loop, if hardware was available).

However, physical modelling is not trivial, and it may still produce unrealistic data. Thus, the solution found resides in external stimulation of the control loops through input files containing the relevant data. Then, the behaviour of the system can be analysed and verified for some corner case situations, assessing the control algorithms correctness.

5.2 Validation tests

The validation tests should be performed by the client using the products manual, so it is expected that a user without prior experience with the product should be able to use it correctly and safely. On the present abnormal circumstances, with limited access to the physical modules, specially for an external agent, the validation is severely limited. Thus, it should be limited to user interface validation.

For this purpose, an external agent will be provided with the software application and the respective installation and usage manuals, and the feedback will be collected and processed to further improve the product.