Phase 1: RFCAR

Product concept, foreseen specifications, planning, tests and initial design

Nuno Rodrigues A85207, Hugo Carvalho A85156, Hugo Ferreira A80665, João Faria A85632, João de Carvalho A83564, José Mendes A85951, José Miguel Alves Pires A50178*

April 13, 2020

Contents

*a50178@alunos.uminho.pt

1 Product concept: Radio Frequency Camera Assisted Rover (RFCAR)

The envisioned product consists of a remote controlled car used to assist exploration and maintenance domains. For this purpose, the vehicle should contain a remotely operated camera feeding back video to the user. Additionally, the vehicle must contain odometric sensors to assist in driving and prevent crashes when user is not in control, e.g., when connection is lost. The vehicle can be used for exploration of unaccessible areas to human operators like fluid pipelines and other hazardous sites.

2 Foreseen product specifications

The foreseen product specifications are listed as topics below.

2.1 Autonomy

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

- No load;
- Vehicle operating at maximum speed;
- Vehicle operating at minimum speed.

2.2 Velocity

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

2.3 Safety

For a remote controlled car, safety concerns not only the car itself and all of the equipment, but also the humans that interact with the car:

- Car: If the user issues a command that would cause damage to the system, the system should take corrective measures to prevent it. The same holds true if the communication between user and system is lost. The system uses odometric navigation.
- Human: Due to the odometric sensors safely fixed in the car, crashes will not occur, making it much harder for the car to hit a person or for any part of the car to jump and cause harm to the user or anyone around.

2.4 Image acquisition

2.4.1 Frame rate

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

2.4.2 Range

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

2.4.3 Resolution

The amount of detail that the camera can capture. It is measured in pixels. The quality of the aquired image is proportional to the number os 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.

2.5 Load

The remotely controlled vehicle can be used in applications involving load carrying (besides its own), e.g., packet delivery. For this purpose it is important to determine the maximum load the vehicle can carry safely at the minimum velocity dened. As the load increases, also increases the power consumption, diminishing the autonomy.

2.6 Overall System latency/responsivess

The overall system latency is the sum of all systems' latencies, which must be under a maximum tolerated value for the user.

2.7 Communication

2.7.1 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 adds overhead to the communication protocol, which must be considered depending on the case. For example, 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.

2.7.2 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.

2.8 Sensibility

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.

2.9 Closed loop error

The velocity, direction and distance to obstacles must be continuosly 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:

- velocity: the user issued an command with a given mean velocity, which should be compared with the steady-state mean velocity of the vehicle.
- direction: the user issued an command with a given direction, which should be compared to the vehicle direction.

- distance to obstacles: the user issued an command with a given direction and velocity which can cause it to crash. The local control must take over control, preventing this to happen, and the final distance to the obstacles must be assessed and compared to the defined one.

2.10 Summary

Table 1 lists the foreseen product specifications.

Table 1: Specifications

	Values	Explanation
Max Velocity	0.2 m/s	Maximum velocity of the conveyor belt in steady state
Dimensions	60x30x30 cm	Dimensions of the conveyor belt in cm [l w h]
Time Min	3 s	Time taken to transport a load the full extent of the conveyor belt at maximum velocity
Max Load	1 Kg	Load the belt can hold without causing any harm to the product
Max slope	15°	Maximum slope in which the conveyor belt can operate at nominal conditions
Slope levels	[0,5,10,15]°	Different levels of slope manually handled
Settling time	$0.2 \cdot T$	This means that it takes up to 20% of the full travel time to reach steady velocity
Overshoot	110% Vss	Maximum velocity the conveyor belt reaches before settling time
Margin of error	95%-105% of Vss	Admissible error in steady velocity
Power supply	12V batteries, 6W	The main power supply will be 12V batteries

3 Planning

In fig. 1 is illustrated the Gantt diagram for the project and in fig. 2 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:
      · frontend 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
      · backend 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.

- <u>Delivery</u>: — project closure encompassing:

  1. Final prototype
  2. Support documentation: how to replicate, instruction manual.
  3. Final report
  4. Public presentation

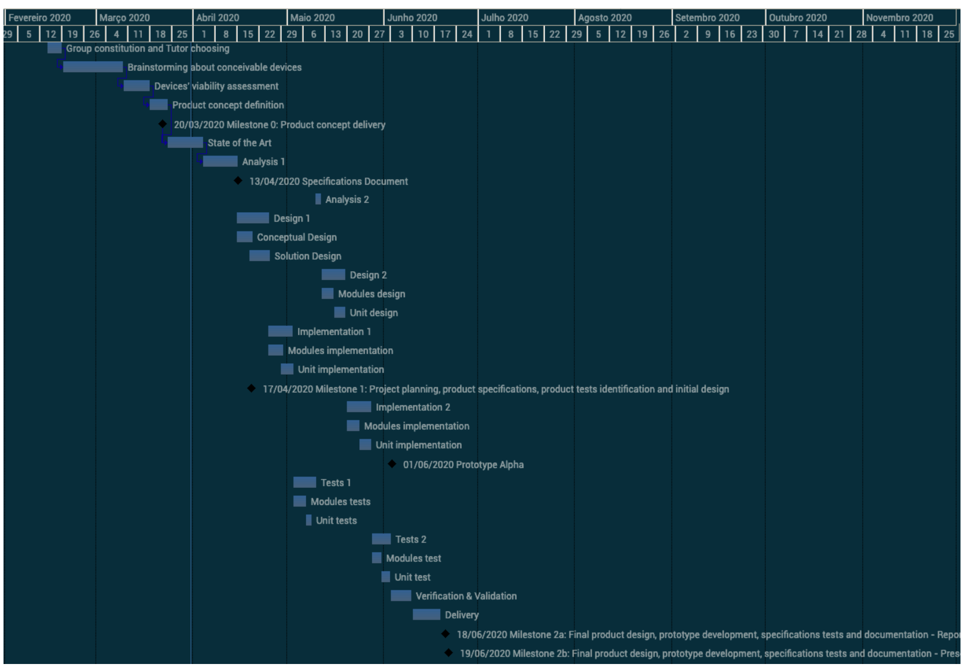

Figure 1: Project planning: Gantt diagram 1

	Nome	Duração	Início	Fim
1	Group constitution and Tutor choosing	3dias	14/02/2020	18/02/2020
2	Brainstorming about conceivable devices	57.13dias?	19/02/2020	09/03/2020
3	Devices' viability assessment	6.84dias?	09/03/2020	17/03/2020
4	Product concept definition	3.46dias?	17/03/2020	23/03/2020
5	Milestone 0: Product concept delivery	1dia	20/03/2020	20/03/2020
6	State of the Art	9.55dias?	23/03/2020	03/04/2020
7	Analysis 1	6.63dias?	03/04/2020	14/04/2020
8	Specifications Document	0dia	13/04/2020	13/04/2020
9	Analysis 2	4.94dias	09/05/2020	10/05/2020
10	Design 1	31.08dias?	14/04/2020	24/04/2020
11	Conceptual Design	14.63dias?	14/04/2020	19/04/2020
12	Solution Design	19.5dias?	18/04/2020	24/04/2020
13	Design 2	22.67dias?	11/05/2020	18/05/2020
14	Modules design	11.46dias?	11/05/2020	15/05/2020
15	Unit design	10.25dias?	15/05/2020	18/05/2020
16	Implementation 1	23.55dias?	24/04/2020	02/05/2020
17	Modules implementation	14.13dias?	24/04/2020	29/04/2020
18	Unit implementation	11.63dias?	28/04/2020	02/05/2020
19	Milestone 1: Project planning, product specifications, product tests identification and initial design	43.13dias	03/04/2020	17/04/2020
20	Implementation 2	23dias?	19/05/2020	27/05/2020
21	Modules implementation	12.17dias?	19/05/2020	23/05/2020
22	Unit implementation	11.25dias?	23/05/2020	27/05/2020
23	Prototype Alpha	0dia	01/06/2020	01/06/2020
24	Tests 1	21.88dias?	02/05/2020	09/05/2020
25	Modules tests	12.05dias?	02/05/2020	06/05/2020
26	Unit tests	4.92dias?	06/05/2020	07/05/2020
27	Tests 2	17.67dias?	27/05/2020	02/06/2020
28	Modules test	9.21dias?	27/05/2020	30/05/2020
29	Unit test	8.38dias?	30/05/2020	02/06/2020
30	Verification & Validation	19.75dias?	02/06/2020	08/06/2020
31	Delivery	26.05dias?	09/06/2020	18/06/2020
32	Milestone 2a: Final product design, prototype development, specifications tests and documentation - Report	0.05dia	18/06/2020	18/06/2020
33	Milestone 2b: Final product design, prototype development, specifications tests and documentation - Presentation	0.33dia	19/06/2020	19/06/2020

Figure 2: Project planning: tasks

4 Tests

Tests are generally considered as those performed over any physical component or prototype. Here, however, it is used in a broader sense, to reflect the tests conducted into the system and the several prototypes, under the abnormal present circumstances. The tests are divided into verification and validation tests.

4.1 Verifications tests

The verifications 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.

4.1.1 Functionality

The remotely operated vehicle is composed of several modules distributed along several different platforms, some of which distanced from each other. In the present abnormal circumstances, this is even more true. 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, assessing if it is correctly processed. For example, if the user issues the vehicle to move to a given place, the message sent to the vehicle must be signaled in each endpoint hit, and the vehicle should move to that place.

4.1.2 Maximum Load

The remotely controlled vehicle can be used in applications involving load carrying (besides its own, obviously), e.g., packet delivery. For this purpose it is important to determine the maximum load the vehicle can carry safely at the minimum velocity defined. As the load increases, also increases the power consumption, diminishing the autonomy. Thus, two alternative definitions, and consequently, tests arise for the maximum load determination:

1. maximum load (at minimum velocity): maximum load the vehicle can carry safely at the minimum velocity defined.

2. maximum load (at 50% over the mean power consumption): maximum load which causes a 50% increase in the mean power consumption, i.e., while operating at mean velocity.

To test the former, load should be increased slowly, measuring the vehicle mean velocity, until the minimum velocity defined is achieved. To test the latter, load should be increased slowly, measuring the power consumption, until a 50% increase over the mean power power consumption is detected, while operating at the mean velocity.

4.1.3 Autonomy

The vehicle is operated off-the-grid, thus, a portable power source must be included. The autonomy — time interval between battery fully charged and safely discharged — should be observed for the following scenarios:

1. No load and vehicle operating at maximum speed
2. No load and vehicle operating at mean speed
3. Maximum load and vehicle operating at maximum speed
4. Maximum load and vehicle operating at mean speed

The autonomy is related to product's power consumption and the capacity of the battery chosen. Under the present abnormal circumstances is not reasonable to expect the product's power consumption to match the real one, thus, for all purposes, this will be considered as the one drawn by the car module itself, namely, the installed motors and sensors.

Then, the autonomy can be measured as the time interval between battery fully charged and safely discharged (the car stops), by fixating the car to a supporting structure with free moving wheels, and imposing the aforementioned conditions.

4.1.4 Velocity

The vehicle must be operated within a safe range of velocity, while also not increasing excessively the power consumption. Thus, these velocity boundaries should be tested in the absence of an external load and in the presence of the maximum load. It is important to define these boundaries as follows:

- minimum velocity: minimum velocity defined for the vehicle, which must be attained even in the condition of maximum load. This value must be selected to assure safe motor operation.
- maximum velocity (no load): maximum velocity for the vehicle in the absence of an external load. This is the absolute maximum velocity for the vehicle.

- maximum velocity (maximum load): maximum velocity for the vehicle in the presence of the maximum load. This value must be selected to prevent excessive power consumption and motor overheating.

The aforementioned velocities should be tested in the designed conditions, within a sufficiently long distance to assure velocity reach and stabilization, and compared to the ones provided in the foreseen specifications.

4.1.5 Safety

Safety is paramount in product design, especially considering the vehicle is to be remotely operated. Safety can be analysed in two ways, considering the preservation of people and goods. For the former, it is important to assure safe user operation as well as safe human interaction — 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.

To test human safety, it is important to identify the interactions between the user and the product, and which are the most prevalent and dangerous. Even so, the exhaustive test is outside the scope of the present work; a small set of features will be tested accordingly to the devised user manual, containing the safety measures. For example, battery installation and conditions should be tested, eventually leading to the posterior incorporation of safety measures in the product.

To test goods safety, it is reasonable to assume the operating conditions of the vehicle. Under these it is important to consider the most critical ones that concern the moments when the vehicle is left to be controlled locally, instead of user controlled operation. The critical conditions for local operations are divided into two sets:

- processing of user commands and vehicle operation: user commands can conflict with safety measures and, thus, should be overriden locally.
- communication loss: the vehicle is left to odometric navigation, preserving the safety of people and goods.

To test these two scenarios, they should be replicated, observing the system response and tolerance.

4.1.6 Image acquisition

The vehicle is equipped with a camera to assist the user in its navigation, thus, requiring it to be feed to the user's platform within proper conditions.

The following variables are to be tested: frame rate, range, and resolution.

Frame rate The frame rate is the rate at which the user platform screen is updated with new image information. It should be maintained within acceptable boundaries to serve the purpose of assisting the user in the vehicle's navigation. To test it, the number of frame displayed per second in the user screen must be updated and checked against the defined boundaries.

Range The range is the maximum distance the camera can clearly effectively capture an image without losing resolution. To test this, an object must be captured at increasing distances, until the image resolution is lost.

Resolution The image resolution quantifies how close lines can be to each other and still be visibly resolved, giving an information on the its detail. The minimum resolution should be tested as providing the least amount of information required for the user, while minimizing data transfer and processing overhead.

4.1.7 Communication 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 adds overhead to the communication protocol, which must be considered depending on the case. For example, 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 field of vision if the frame rate is within acceptable boundaries.

Thus, given the critical nature of user commands issued, the focus will be on this communication link. To test the reliability dummy packets should be sent from the user platform to the vehicle and be acknowledged and parsed correctly.

4.1.8 Closed loop error (Control loops)

The velocity, direction and distance to obstacles must be continuosly 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:

- velocity: the user issued an command with a given mean velocity, which should be compared with the steady-state mean velocity of the vehicle. This can be tested by comparing the user defined velocity and the vehicle's;

- direction: the user issued an command with a given direction, which should be compared to the vehicle direction. This can be tested by measuring the angle between final and initial points and comparing it with the user defined direction.

- distance to obstacles: the user issued an command with a given direction and velocity which can cause it to crash. The local control must take over control, preventing this to happen, and the final distance to the obstacles must be assessed and compared to the defined one.

4.2 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.

5 Initial design

Following an analysis of the products family tree (remote controlled cars) and the state of the art, an initial design of the product itself can be produced (fig. 3). 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.

Thus, summarising, the initial design yields the system illustrated in fig. 3, 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 cart's movement and headlights for nocturnal or low light conditions;

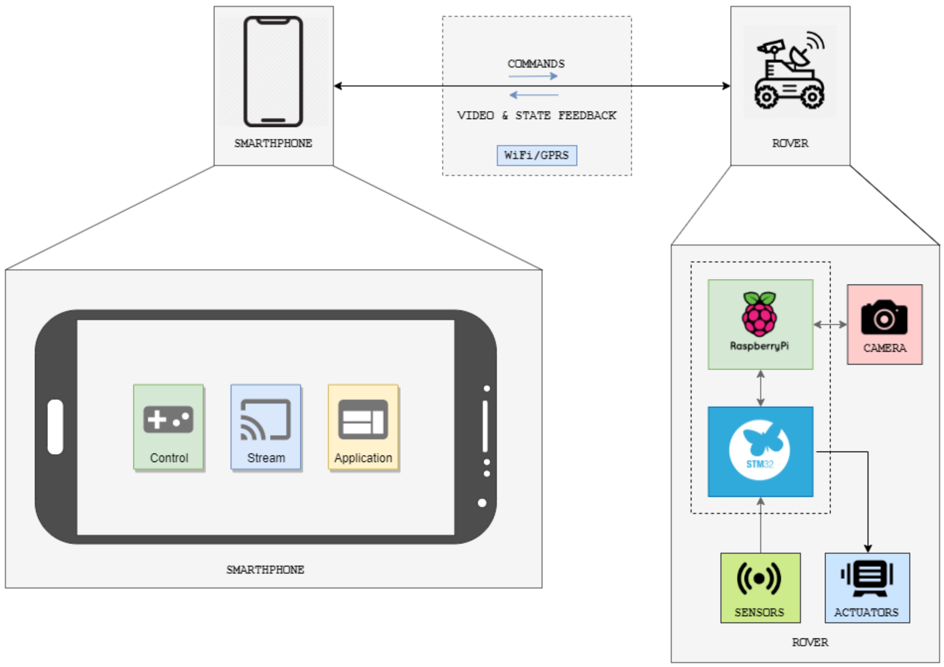

Figure 3: Initial design: Block diagram view

- **Sensors**: Odometric sensors that support 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;