Brainwave-Controlled Robot Car

4G06 Capstone Project – Software Requirement Specifications

Group 11

Butt, Arfa (buttaa3)
Dong, Kairui (dongk7)
Jin, Ziyi (jinz27)
Li, Jiahao (li577)
Liu, Jianpeng (liu433)
Luo, Jie (luoj3)
Premachandran, Aravi (premaa)

Revision 0 – November 2, 2018

Contents

1. Introduction

1. Purpose

This documentation is served as a System Requirement Specification document for our project Brainwave-Controlled Robot Car. This is an agreement between stakeholders and customers on what our project should behave. It also specifies the conditions of our project that are likely to be changed during the development. It provides a rigorous assessment of requirements before the more specific system design stages [1]. The requirement specification document lays out functional and nonfunctional requirements for the project to-be-built.

The current section is an introduction, which introduces this document and a description to our project. It also defines terminology, acronyms, abbreviations that will be used in the following sections. The reference section is then followed. The next section, Overall Description, provides a brief description on the overall project. The third section is the requirements section, which lists the details on functional requirements, non-functional requirements, constraints, event-handling, and lists of requirements that are likely to be changed or not changed.

2. Scope

This project will utilize pre-collected data to train a machine learning model and the machine learning model will recognize the user's brain activity signal and translate the signal into one of 5 commands to manipulate a small robot car. The brain activity signal will be collected by Muse and the robot car will be controlled by an Arduino microcontroller. Bluetooth will be communication protocol to transfer the data and commands between Muse and Arduino. There are total five different commands for the robot car, move forward, move backward, turn left, turn right and stop. The project will be completed by April 2018 and the total budget for this project is $750 CAN.

3. Definitions, Acronyms, and Abbreviations

Term	**Definition**
EEG	Electroencephalograph
Muse	EEG data collection device
Arduino	Programmable Microcontroller
Bluetooth	Communication protocol
DESC	Description
RAT	Rationale
DEP	Dependency

4. References

[1] "Software requirements specification," *Wikipedia*, 26-Oct-2018. [Online]. Available: https://en.wikipedia.org/wiki/Software_requirements_specification.

[2] “Bluetooth Technology | Bluetooth Technology Website,” *Bluetooth*. [Online]. Available: https://www.bluetooth.com/bluetooth-technology.
[3] *Arduino - Education*. [Online]. Available: https://www.arduino.cc/en/Main/Education.

2. Overall Description

This section will give an overview of the whole system. The system will be explained in its context to show how the system interacts with other systems and introduce the basic functionality of it.

1. Product perspective (With Context Diagram and Constants)

As being mentioned earlier, the goal of the product is to allow uses to control a robot car by their own brain activities. the project can be divided and categorized into three fields, data collection and processing field, feature extraction and model training field and Arduino programming field (See Figure 1). There three fields are in sequential order and they depend on each other closely. For the data collection and processing will be first part to be worked on and it will also affect the accuracy of the machine learning model significantly. In order to raise the quality of the training data and make them more representative, the training data will be collected from all the team members internally.

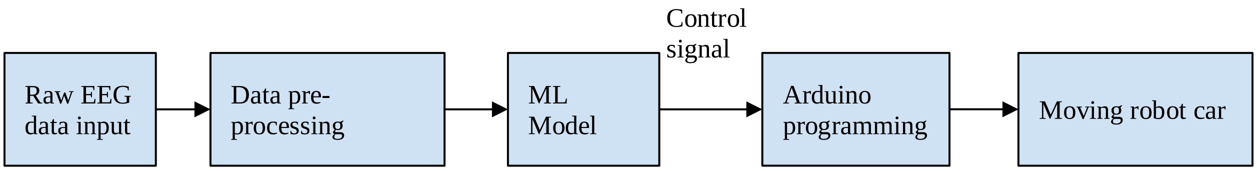

Figure 1

2. Product Functions (With Functional Decomposition Diagram)

The product will consist of three parts which are Muse, the robot car, and the software. Both Muse and the robot car will be connected to the software with Bluetooth in a reasonable distance. Muse will be able to collect the signal from the user’s brain activity and send them to the software when the user wear muse on his head. The signal will be finally sent to the robot car and used to control its motions. The user will be able to control the robot car to go forward, go backward, turn left, turn right and stop by simply thinking these commands in his mind. The whole system will work in a real time situation which means the robot car will do the motion immediately after the user give the command. For security purposes, the system will be able to shut down at any time to avoid the unexpected situation. The function decomposition diagram is shown in Figure 2.

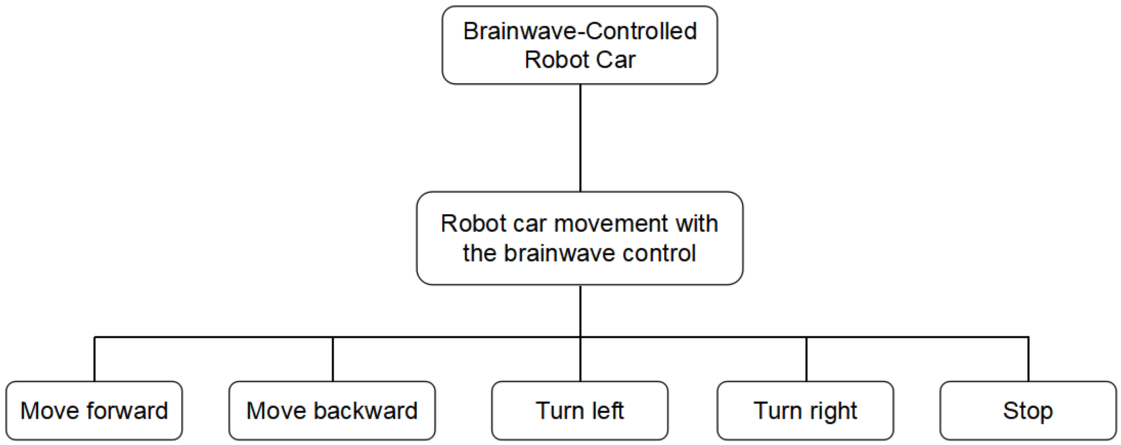

Figure 2

3. Required Behavior Description

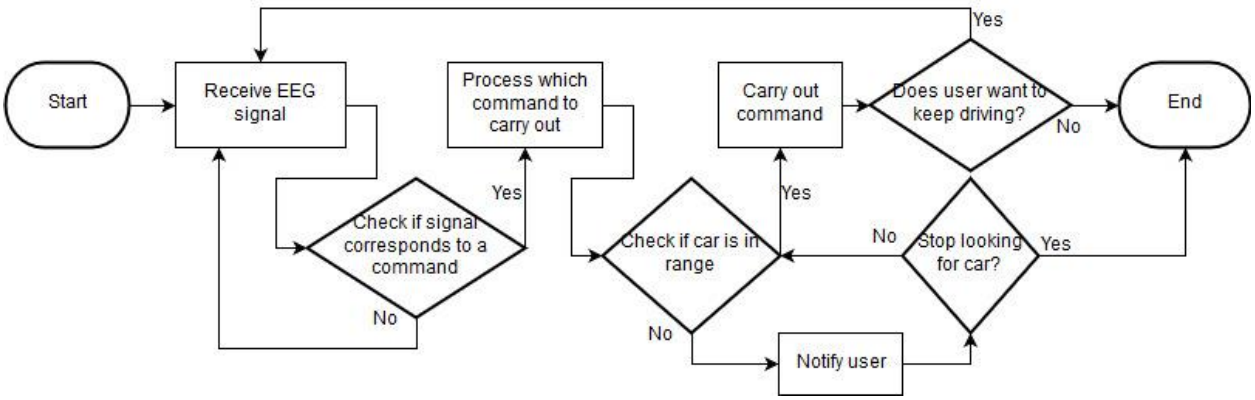

Figure 3

State	Description
Receive EEG signal	In this state, system will receive data from Muse headband.
Process which command to carry out	In this state, system will compare received data with pre-processed data to compute which command to carry out.
Carry out command	In this state, system will send user’s command to the robot car in order to be carried out.
Notify user	In this state, system will notify user about the absence of robot car from the bluetooth range.

3. Specific Requirements

This section contains all the functional and quality requirements of the system. It gives a detailed description of the system and all its features.

1. Functional Requirements

This section includes the requirements that specify all the fundamental actions of the software system.

3.1.1 Functional requirement 1.1

ID: FR1

TITLE: Reading EEG signal

DESC: The system, especially the EEG signal collecting device (Muse), should be able to read the EEG signal from the user's brain in real time.

RAT: This is the main input source.

DEP: N/A

3.1.2 Functional requirement 1.2

ID: FR2

TITLE: EEG signal to control signal mapping

DESC: The system should be able to map the EEG signal to robot car's control signal with at least 70 percent accuracy for each control instruction category. There are five control instruction categories: move forward, move backward, turn left, turn right and stop.

RAT: It is necessary to convert the raw EEG signal to control signals.

DEP: N/A

3.1.3 Functional requirement 1.3

ID: FR3

TITLE: User interface

DESC: User should be able to control the robot car using the brainwave (EEG signal) in real time. The latency of the system should be relatively low. The detail of the latency requirements is defined in FR4.

RAT: This is the way how user interacts with the robot car.

DEP: FR1, FR2, FR4

3.1.4 Functional requirement 1.4

ID: FR4

TITLE: System latency

DESC: The latency of the end-to-end system must be less than 2 seconds and is preferred to be less than 1 second. The end-to-end latency is defined as the time interval between the system getting the EEG signal from the user and the car starting to respond to the corresponding control signal.

RAT: This is necessary for a good user experience.
DEP: FR1, FR2

3.1.5 Functional requirement 1.5

ID: FR5

TITLE: Safety

DESC: The system should provide a way for the user to shut down the entire system, especially the moving robot car at any time. As the brainwave interface can be unreliable under certain conditions, it is necessary for the system to have another reliable way for the user to interact with.

RAT: Safety is critical.

DEP: N/A

3.1.6 Functional requirement 1.6

ID: FR6

TITLE: Muse and backend server communication

DESC: Muse should be able to communicate with the backend server over either wired or wireless communication channel, so that the server can send the instruction to Muse and Muse will be able to send the collected EEG signal back to the server.

RAT: Muse and the server should be able to communicate.

DEP: N/A

3.1.7 Functional requirement 1.7

ID: FR7

TITLE: Robot car and backend server communication

DESC: Robot Car and the backend server should be able to communicate with each other over a wireless communication channel.

RAT: Robot car and the server should be able to communicate.

DEP: N/A

3.1.8 Functional requirement 1.8

ID: FR8

TITLE: Robot car control

DESC: Robot Car should be able to respond to the corresponding control signal sent by arduino. The speed of the car should be relatively low (less than 10km/hour).

RAT: Robot have to be able to go forward, backward, left, right or stop based on the control instructions.

DEP: N/A

2. Non-functional requirements

The requirements in this section provide a detailed specification of the user interaction with the software and measurements placed on the non-functional requirements of the system.

3.2.1 Availability

ID: QR1

TITLE: Capabilities of handling unexpected errors and exceptions

DESC: This system should be able to handle a fair amount of unexpected errors and exceptions for the sake of both safety and reliability. Coding should be done in a way where maintenance can be done in a fair amount of time to increase the time of the system being operable. Hardware of this system should be tuned fine to prevent hardware glitches as much as possible. Deployment of the system should be designed to be user-friendly and in a fair amount of time.

RAT: Availability is one of the terms in measuring the reliability of a system. When unexpected errors and exceptions occur, the system should still operable partially. In such cases, determining where the problem is via maintenance in due time is crucial to restarting the system as soon as possible. Hardware problems should be prevented by frequent maintenance including greasing, checking loose screws, etc.

DEP: QR3, QR6

3.2.2 Security

ID: QR2

TITLE: Protection against Man in the Middle attacks

DESC: Due to the fact that the Muse will communicate with the Robot Car via bluetooth the software must protect against Man in the Middle attacks.

RAT: If someone was able to intercept the signal and modify it before it reaches the Robot car they could change the car's path thus taking over control.

DEP: N/A

3.2.3 Maintainability

ID: QR3

TITLE: Allows for the addition of new features

DESC: The coding should be done in a certain way such that adding new features would be simple. Detailed and proper comments as well as easy-to-read and consistent format should be applied throughout the process of coding. Only necessary information should be included in comments. Implementation of modules would be preferred.

RAT: Detailed comments of code and a clean format is crucial in the process of maintenance for even the memory of the developer of this system can fade over time. By implementing modules, change of code during maintenance and addition of new features do not require corresponding changes in the entire system.

DEP: N/A

3.2.4 Compatibility

ID: QR4

TITLE: Capable of use in various situations

DESC: The coding should be done in a way such that it could be used in completing other tasks. Writing a user-friendly API and implementing the API in the product would be preferred since APIs can be easily implemented in other projects.

RAT: For instance, if a person is interested moving a bionic arm left or right they should be able to do that with some simple adaptations to the software.

DEP: QR3

3.2.5 Integrity

ID: QR5

TITLE: Data Integrity

DESC: The software should be written in such a way that the data pertaining to brain signals is well protected and non-modifiable by those without access. Signals are preferred to be encrypted before being sent out by Muse. Based on the computing power of the device we are planning to use, public key cryptosystems like RSA is recommended to be used.

RAT: If a person was interested corrupting the data and causing the system to work erratically it could have catastrophic consequences.

DEP: QR2

3.2.6 Portability

ID: QR6

TITLE: Capability of being mobile

DESC: The entire system should not be such volume or mass that an average adult is not capable of carrying it around with or without the help of simple tools.

RAT: An average adult should not be stumbled by the volume of mass of the device when trying to move the device for whatever purpose.

DEP: N/A

3. Design Constraints

This section includes the constraints coming from hardware and potential constraints from the system itself.

3.3.1 Hardware constraints

ID: DC1

TITLE: Accuracy of the Muse device

DESC: There is no guarantee that the Muse device which is planned to be used to collect users' brain activity can be always accurate.

RAT: N/A

DEP: N/A

ID: DC2

TITLE: Bluetooth device constraints

DESC: Beyond a certain range of distance, Bluetooth signals might not be well received by the robot vehicle device and can thus cause glitches.

RAT: N/A

DEP: N/A

3.3.2 System constraints

ID: DC3

TITLE: Time delay

DESC: Based on the choice of communication protocols of either wired or wireless connection, time delays always exist but varies.

RAT: Encoding and decoding of any communication protocol would cause a short time delay.

DEP: N/A

ID: DC4

TITLE: Noises of brain signals

DESC: Noises of brain signals always exist and can be very different among different individuals. This can cause potential problems if some of the signals have overlaps and thus are not separable by the machine learning model.

RAT: N/A

DEP: N/A

ID: DC5

TITLE: Machine learning limitations

DESC: No ML system is guaranteed to be 100% correct.

RAT: Essentially, by feeding brain signals into a neural network model, the model produces clustering of the 5 target signals and try to predict probability of the aforementioned 5 classes. Since it is only predicting a probability, there is always a chance of the prediction going wrong.

DEP: N/A

4. Undesired Event Handling

Undesired event	Handling method

Communication lost (system failure, out of range)	The arduino should stop the robot car and reset itself to the initial state, when the communication with the server has lost.
High Noise Signal causing the car doing unexpected movements	The arduino should detect the irregular data partners causing by the noise signal and issue the warnings regardingly

5. List of Requirements That are Likely to Be Changed

Requirement ID	Reason
FR1	Muse may be replaced by other EEG signal collection device if it can provide better performance.
FR4	The system end-to-end latency requirements may change (current value is less than 2 seconds), if the current latency threshold cannot fulfill the functionality needs.