DEVELOPMENT OF A COMPUTATIONAL TOOL FOR CALCULATING ELECTRICAL POWER FLOW IN THE DC LOW VOLTAGE DISTRIBUTION SYSTEM OF MODERN VEHICLES

By

JUAN ANDRÉS PÉREZ RÚA

Submitted to the Department of Electrical Engineering,
Electronics, Computers and Systems in partial fulfillment of the
requirements for the degree of
Erasmus Mundus Master Course in Sustainable Transportation
and Electrical Power Systems

At the

UNIVERSITY OF OVIEDO
September 2016

Author…..……………………………………………………

Certified by.……………………………………………………

PhD. Pablo Arboleya Arboleya
Associate Professor
Thesis Advisor

Development of a computational tool for calculating electrical power flow in the DC low voltage distribution system of modern vehicles

By

Juan Andrés Pérez Rúa
E-mail: japerezrua@gmail.com

Submitted to the Department of Electrical Engineering, Electronics, Computers and Systems on
September 2016, in partial fulfillment of the requirements for the degree of
Erasmus Mundus Master Course in Sustainable Transportation and Electrical Power Systems

Abstract

In this research work is embodied the result of gathering information and the establishment of a general procedure by which the manufacturing data of the on-board network distribution system of modern vehicles are adapted to a proposed format, in order to perform electrical power flow simulation on the aforementioned system. First at all, the state of the art information related to the philosophy and manufacturing procedures of the wiring systems currently applied by the Volkswagen Group are deployed. Secondly, the data organization system is described from scratch in accordance to the official practices of the Group. Later, the proposed data input format for an external power flow solver-visualization tool is explained in details along with coded routines in MATLAB to get them automatically parting from structures formed in pre-processing stages. Finally, it is tackled a descriptive analysis regarding the mathematical approaches available in the literature to solve linear systems defined by DC resistive networks such as the on-board distribution systems of vehicles; as result of this study, a method based in the graph theory of cyclic-direct pair-wise graphs is proposed as main algorithm to solve the electrical power flow of these systems: the Meshed Network Back and Forward Sweep method, which has been applied to static distribution networks and even to train systems, is newly explored from a point of view of applicability to the nature of the on-board distribution systems of modern vehicles; the advantages respect other methods based in mathematical techniques of the linear algebra field are listed and a test is carried out by means of an exemplification of the network. The main contributions of this work is the settling of a general procedure and methodology with aims of simulating the full electrical system of vehicles based in a modular wiring system design approach, which will help thoroughly to the improvement of the current techniques used by the designers when sizing wires and fuses in the department of electrical and electronic development in SEAT.

Key words: Data organization, automotive modularity, wiring harness optimization, DC power flow, resistive networks, algorithms.

Thesis Advisor: PhD. Pablo Arboleya Arboleya.
Associate Professor

External Co-Advisor: Eng. Manuel Diaz Millan
Chief at the Electric and Electronic Department in SEAT S.A

Institution where the work was developed: Technical Center of SEAT S.A

ACKOWLEDGEMENTS

I would first like to thank my thesis advisor PhD Pablo Arboleya Arboleya of the department of electrical engineering at University of Oviedo. His support and advices were a constant throughout all the process of this work and also in the internships.

I also would like to thank my company tutor Manuel Diaz Millan, for his guide in the electrical and electronic department of the technical center at SEAT and to my colleague Armand Rius Rueda for his constant insight and collaboration. Additionally, I want to express my gratitude to the professors Pablo Garcia Fernandez as a coordinator of the Master STEPS and to Cristina González Morán who also helped in this internship semester.

Lastly but no less important, very warm thanks to my family for their support along the past two years in the master and to my friend Gülce Onbaşılı her constant support and all the experiences shared in these 4 semesters of studies.

TABLE OF CONTENTS

Chapter Page

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

The on-board power supply network of road vehicles has been earning a slow but always increasing importance in the last decades. From the beginning of the first electrification of classic vehicles, the low voltage distribution network at 12 Volts is a key factor in the design of vehicles, being the costs of wires, both for power and communication uses, the complexity of industrialization and logistic and the complexity of assembly the main driving points. Additionally, in the last years the level of complexity of the required equipment in the on-board systems is growing up considerably, due to the advancements of new technologies related with silicon and electronics for providing more satisfaction to the final costumers, which has caused the more intensive use of Electronic Control Units (ECUs) (embedded system for controlling one or more appliances in the vehicle), bringing as a consequence, finally, greater amount of wires, more compound harnesses. Likewise, a similar effect is provoked by the current trend for designing and manufacturing electric and hybrid vehicles, which of course will require even more electrical and electronic subsystems to be operable.

The aforementioned might be in some sense in contraposition with the internationals efforts for cutting down the emissions of greenhouse gases, - in fact, the European Union introduced in 2009 mandatory passenger car GHG emissions standards, specifying up to 95 g/km of maximum emissions for new car fleet in 2020 – since the increase of weight due to more wires and harnesses provoke more consume of energy in the manufacturing processes and even also in the driving fuel consumption. Hence, recent researches have been carried out in order to reduce on board harness network by means of the reduction of the size of the wires through thermodynamic analysis techniques, which might bring not only the compliance with the emerging standards of GHS reduction, but also economic profits for the vehicles manufacturers [1], [2],[3] .

The actual sizing and dimensioning techniques used by the main vehicles manufacturers follow an empirical specific procedure with a step by step guide given by existing regulations, from that well established practice two aspects are put under lens: the thermal behavior of the cables and their voltage drop throughout the on-board distribution systems. The voltage drop might be the most important checking point while sizing the wires due to the length of several cables, and also it has been determined that for the designers of the department of electrical and electronic development of the technical center in SEAT[1] to count with a tool for calculating the voltage drop might reduce the time invested in this process, also, as an important step for optimizing the wiring systems through more complex and automatic procedures, is required recurrent simulations for estimating the compliance of minimum voltage at terminals of equipment, making the power flow calculation a determinant aspect in the chain of the design of a car model.

The power flow analysis of static networks such as transmission and distribution lines is a well-known topic for power system engineers; however, the application of such techniques for electrical on-board distribution systems needs to be adapted considering the complex structure of a vehicle and the available information of factory, which is not organized from the start to be used for a power flow simulator. Technical aspects like the greater current densities [4], the presence of electronic loads and the generally weakly meshed characteristics of networks on vehicles must be addressed in order to develop the computational tool for calculating the voltage drops in the system.

---

[1] SEAT is a car manufacturer company located at Martorell, Barcelona. SEAT today is the only major Spanish car manufacturer with the ability and the infrastructure to develop its own cars in-house.

For all the above mentioned reasons, this thesis focuses in the tasks related to the description of the adopted procedure for organizing the factory data coming from datasheets of equipment and from the network design produced in the cabling department for a vehicle, the confectioning of algorithms for generating automatically the required inputs for an external power solver which will provide, between other benefits, the dynamic visualization of the power flow and voltage drops in a particular vehicle. Additionally, a solution considering a backward and forward sweep algorithm for weakly meshed networks and constant current loads is proposed for solving the system from a pure console point of view.

This report is structured in this way: In the Chapter IV a general explanation of the concept of modularity, families and the main parts of the harnesses design process is done. In the Chapter V is detailed the adopted organization of the factory data to standardize it. In the Chapter VI the inputs provided to the external power flow solver are explained, and in the Chapter VII the proposed power flow solution of the system assuming constant current consumption and applying an algorithm used commonly for radials statics distribution networks is described. Finally in last Chapters the conclusions are presented along with other information regarding the internships.

CHAPTER II

OBJECTIVES

The main goal of this master thesis, is the settling of a general procedure and methodology with aims of simulating the full electrical system of the on-board distribution network of vehicles based in a modular wiring system design approach, and the selection and implementation of an algorithmic method for carrying out the solution of power flow in such networks and then providing the designers of the electric and electronic department of SEAT with a computational tool for voltage drop validation. In order to achieve this main objective the next specific targets are established:

- To study of the state of the art of the philosophy for designing wiring harness in modern vehicles, i.e, understanding the different approaches used by the car manufacturers throughout the world and the one mainly applied in the Volkswagen Group.
- To study the current methodology adopted by the Volkswagen Group (specifically in SEAT) for sizing fuses and the wires composing the harnesses.
- To study and learn the software tools used by the designers of the electric and electronic department of SEAT for dimensioning the on-board network of vehicles.
- To establish general formats by means of Excel files to organize and adapt towards simulation purposes, the data produced by the software tools used by the designers.
- To understand and design pre-processing stages in order to take the Excel files formats and transform them into appropriate structures to perform simulations.

- To establish general input formats for an external power flow solver/dynamic visualization tool according to the common practices adopted by commercial software tools and IEEE recommendations.
- To program automatic routines in order to get the aforementioned formats (tables and numeric matrices) for the external solver.
- To study the state of the art of methods and algorithms used to solve resistive-DC networks.
- To select a method and adapt it in order to solve the on-board distribution network of vehicles.
- To code algorithmic routines to solve the on-board network distribution of a vehicle from a pure console point of view and considering constant currents.

CHAPTER III

DESCRIPTION OF THE INTERNSHIPS

This work has been carried out during an internships period as a part of the fourth semester of the EMMC STEPS program version 2014-2016, in the technical center of SEAT department of electric and electronic development at Martorel – Catalunya - Spain. Therefore, all the activities described here have been done in the timeframe of the internships which was between March of 2016 and July of 2016.

Additionally to the opportunity to make research in the company regarding the subject of the master thesis, I have had the chance to participate in some experimental routine tests on vehicles, such as tests on the motor of the windshield cleaner of the new model SUV ATECA, which was having some particular issues about excess of speed and torque, making it to hit the upper part of the vehicle. On the other hand, I have been providing support to the designers in handling the software tools used in the design of the on-board network for specific new models projects. Finally, I have been participating actively in current research projects about the optimization of the wiring harnesses regarding their validation under thermodynamic behaviors in normal steady state and under contingency (faults); in this matter, algorithmic routines have been done to obtain models of liner regression for calculating the minimum area of a bundle in order to keep the working temperature around permissible values, and test bench of wires bundles have been implemented to validate such off-line models. It is important to remark that the voltage drop validation process is other important step in the whole process of wires optimization which SEAT is working nowadays exhaustively.

CHAPTER IV

OVERALL VIEW OF THE HARNESS DESIGN: STAGES, MODULARITY AND FAMILIES

This Chapter has been mainly based in the information contained in the reference [1] due to this is the official reference of SEAT for this related information. The aim is to describe the important concepts of modularity and families in the car manufacturing industry, which is adopted in SEAT and even though is a great advantage for the final costumer, it represents a huge complexity for designing all the embedded systems of a vehicle, of course in that group is included the wire harnesses. Likewise, the practiced regulations for sizing the wires and overcurrent/short circuit protection fuses are presented, making evident the voltage drop checking as an important step in such primary engineering stages.

The development of the on-board electrical supply system is, without place to doubts, one of the most complicated processes inside of the design chain of a vehicle. This is because of the evolution of electronic and electrical components and their huge associated variety, making to extend the wires inside of harnesses up to 1 km and with 20 kg. In the Figure 1 is displayed an example of a wire harness, it can be seen this is basically an assembly of cables which transmit signals and/or electrical power from supply to the components of the system. The extremes of the harness have connectors for allowing the joining between the points one desire to establish an electrical coupling, and it is covered by tapes and other fittings for protecting it against the extreme conditions that might appear in the surroundings. One can see that a harness has a tree configuration for reaching the physical locations intended to connect through the inside part of the body's car, in such a way that is not possible for the user to see the harnesses for both preserving and aesthetical reasons.

Figure 1: Wire harness [5]

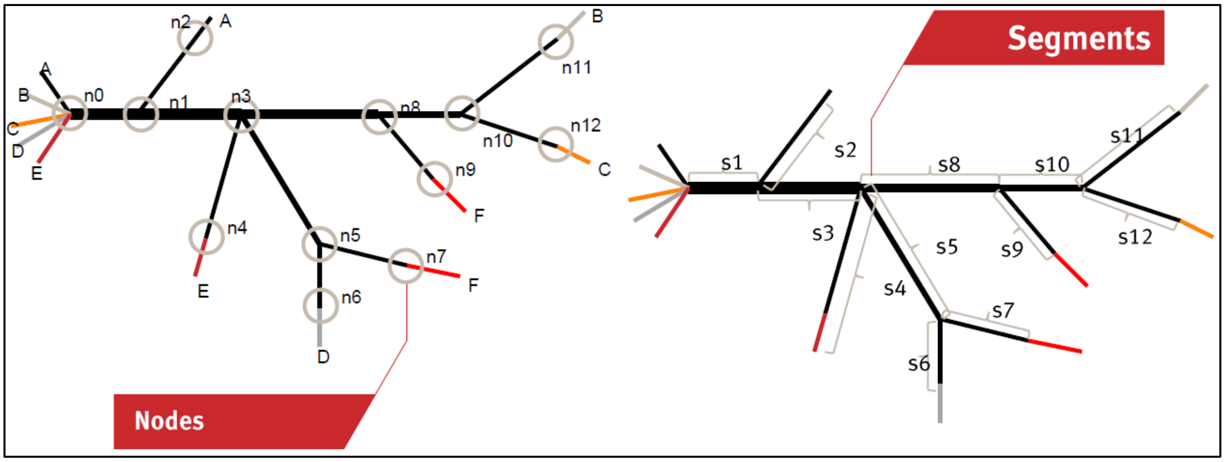


Figure 2: Geometrical descriptions of a wire harness

It is also worth it to comment that a wire harness is a manufacturing unit, i.e., 2D layouts after designing are given to the provider for its systematic fabrication (assembly drawings). With that objective, a physical detailed description is also required to perform, in the Figure 2 is shown the geometrical description a harness. As it was said previously, a wire harness has a tree configuration, consequently it is formed by geometrical nodes and segments, where throughout them, cables are conducted to their source and destination. Therefore, one can see that a wire *A* (Figure 2) parts from node *n0* and reaches its destination in the node *n2*, passing by the segments *s1* and *s2* to do so. Inside of the geometrical nodes, there may be also electrical nodes done by soldering of wires.

For the design of the wires harness, the point from which is started, is the technical report of the vehicle, where it is contained all the information produced during the *pre-engineering* stage, such as the main specific equipment to install in the new car model and in the same way the functionalities to be offered. Hence, the detailed engineering is called for the definition of the architecture of components system by system. After this point, the elaboration of the electrical schematics is carried out, indicating in diagrams the connections between the components of the system, which has a related complexity due to the great variety of options offered in the European car manufacturer, being needed to establish a relation between the selectable equipment of a model and its associated components. Later the elaboration of electrical schematics is performed; the wiring schematics are developed, aimed to display physical features of the wire systems, such as connector, fuses, and color standards and so on. Simultaneously, a 3D routing definition using specialized CAD software is done according to the electrical schematics and the physical location of the components inside of the car, defining the shortest path through the harness a cable must track in order to optimize its length. Finally, the aforementioned assembly drawings are produced taking into the different combinations of equipment, after some adaptations these layouts are handed out to the manufactures for the systematic fabrication of the harnesses. Next on, it is explained in more details the previous concepts.

4.1. Functional systems, electrical schematics and families

As it was commented previously, once the pre-engineering stage of the design of a new model car model is executed, the functional systems are defined along with the architecture of each one.

A functional system might be defined as a set of functions to be performed by a group of components. For instance, the lighting system of a vehicle is a functional system, being composed by different components such the brake light, indicators, fog lights, headlights, rear lamps, and so on. The sound system is also other functional system formed by the radio, front loudspeakers, and back loudspeakers, touch screen, etc. In some cases, some component may belong to more than one functional system, for example, the wheel rotational speed sensor can be part of the systems *ABS* (Anti-Lock Braking System) and navigation; due to this two systems have different functionalities they are separated but both require a common component. The concept of family is basically the same of functional system, so, the lightening system is a family and the sound system as well. The mapping of connections between components belonging to a family are presented in the wiring schematics, along with other important information like type of wires, materials, connectors, etc.

4.2. Types of harnesses and different approaches for their definition

As it was mentioned before, the conception of harnesses is important for having in a compact way a group of elements like wires, connectors, protections and other fittings. This is really useful for the massive productions of cars, since in this way the assembling of harnesses become one more part of the overall vehicle assembling chain. Perhaps it can be understood easier, using the analogous concept of puzzle, where each harness is a token.

According to this, it can be understood that the definition of different types of harnesses are given, between other reasons, mainly due to the assembly lines of a car and the routing of the wires defined in the 3D layouts. A common example is the vehicle’s door: This part is assembled into the body exclusively when it is completely finished and with all its elements mounted to it, including, obviously, the door’s harness where the wires are transported for connecting its electrical components. Therefore, it would not do any sense if the wires that supply, for instance, the electrical motor for moving up and down the door’s window are in the harness of the main engine, or in other way, it would be senseless to install the door’ wires after the door is mounted into the whole car. For these reasons, a car is formed for several harnesses, with a *main* harness where all the *auxiliary* harnesses are connected to, through couplings or cut-off points. Thus, the harnesses assembling match with the whole construction of the vehicle. Other factors that force the separation of harnesses to several smaller, is the constraint in size of some wires, making difficult to handle them in the same groups due to reduced space.

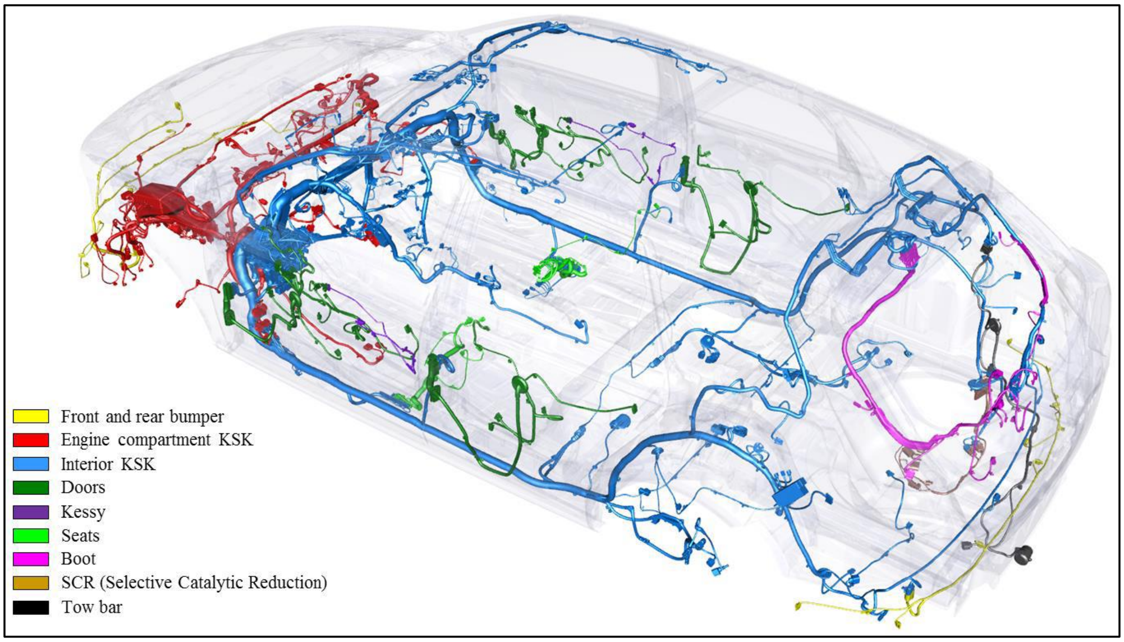


Figure 3: Full view of the wiring system

The Figure 3 presents the overall view of the wiring system determined by the different harnesses contained in a car. In this figure it can be appreciated the *main* harnesses named as *KSK kundenspezifischer Kabelstrang* (German for customer-specific wiring harness). In this specific model of car, there are two *KSK* harnesses where the other ones (*auxiliaries*) are connected to, the interior *KSK* (in blue color) and the engine compartment *KSK* (red color); these ones are the biggest and collectors of the wiring system of a vehicle, counting with the most of the functions, in contrast with the others which are worth it for less functions due to their small size. Both types have to be constructed taking into account that they have to fulfill all the requirements given by the variety of selectable equipment (modularity).

Currently, the cars manufactured throughout the world offer two types of versions regarding the wiring system: the few variants model and the many variants one (See Figure 4). The difference is due to the variety of car models offered to the costumers. In the Figure 5 can be seen that the few variants are offered mainly in Asia and North America, but the manufacturers in Europe give the option of selecting between several modules the desired final configuration.

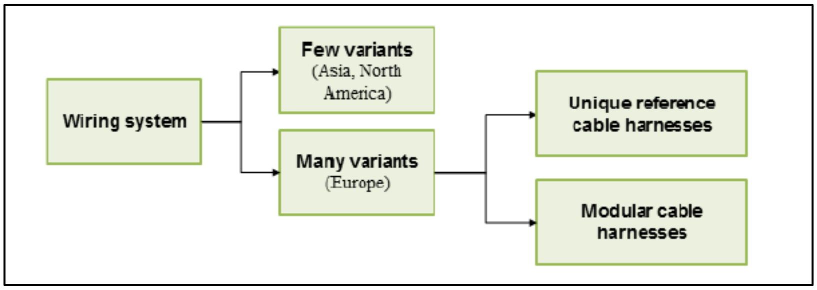


Figure 4: Wiring systems approaches

Figure 5: Locations of the use of the different wiring systems approaches

The manufacturers, who offer few variants, design the wiring systems for each available configuration of the car. Of course this is only possible when they offer to the customer not many options for buying a vehicle. For instance, the variations *basic*, *standard* and *premium* are the typically different models proposed under this approach. The advantages of this scheme are noticeable mainly from an industrialization and logistical point of view, since with less variety the complexity is dramatically decreased and therefore easier to do such activities, joint to the fact of being easier the assembling in the manufacturing chain. Also the cost for the time saved in these activities is optimized. The disadvantages are evident from marketing and technical perspectives; in the first one the fact of offering to clients less options impact on the amount of people interested in the brand, and it is oriented to more specific interest of customers. The latter can be explained, taking into account that this approach still needs to divide the harnesses into several groups (as explained before), so, maybe some of them have to be shared between the different car configurations, causing that a particular model has some wires or elements which are not required for its particular needs; increasing in this way the costs in materials.

The manufacturers that consider the many variants approach for designing the wiring system basically are prone to pick up one of these two methodologies (See Figure 4): *unique reference* and *modular* wiring systems. In the first, functions may be combined by subdividing the wiring systems into many harnesses and in the latter, manufacturers prefer to assemble customer-specific wiring harnesses (*KSK* in German). Next, it is explained more deeply about the two approaches.

The *unique reference* wiring system approach consists in splitting the whole wiring system into many smaller harnesses, in such a way that each harness encloses a more or less small number of variations in function of the vehicle's modularity. With this scheme, it is possible to associate a specific reference number to all the set of wiring harnesses, thus facilitating the spare parts logistics when they need to be replaced or in the manufacturing process line. One can relate this concept imaging each harness as a Lego piece, with each piece with a unique code through which can be identified, so if you loss one brick you can go easily to the shop and buy it by specifying its code. But as it can be supposed the main disadvantage of this model is the high complexity in the assembling, due to for a specific car model hundreds of harnesses may be required and must be carefully handled by the operators in the factory, and also the plugging connector between one to other. On other hand, it would bring more vulnerability to the quality control, because the increase of connections and wires segments may cause mistakes in the process of construction.

The *modular* wiring system approach basically encompasses the techniques related with grouping all of the variants of a car model into a set of unique bid sized wiring harnesses (this approach counts with *main* and *auxiliaries* harnesses, as explained before). This approached compared with the previous one, has less amount of harnesses but with bigger size .This the wiring system technique implemented by SEAT and the whole Volkswagen Group[2]. As it was commented before, a family is showed to the customers as packs of functions (a functional system), and each family counts with a group of modules where the client can select only one (in many less cases a family allows selecting several modules). In this way, a selectable component is strongly correlated to its so-called *module*, and therefore the whole wiring system is defined in function of the combinations of modules selected by the final user of the product, i.e., it ends up that there are as many configuration of harnesses as amount of possible selection of modules. One can associate this concept as dynamics harnesses which changes with the inputs of modules, in contrast with the *unique reference* approach where each small harness is *static* in function of such inputs, what it changes are the connections between them. The main advantage is the saving of costs of materials for manufacturing the harness, because, each harness is personalized for each customer and hence, only the exact amount of wires and other fittings are used in the harness in that case, and consequently, a great variety of options are supported to be offered to the customer for a car model, increasing in this way the image of the brand and potential buyers. Nevertheless, as main

---

[2] The Volkswagen Group is the first vehicle manufacturer of Europe and the second in the world. It is formed by international brands such as Volkswagen, Audi, SEAT, Lamborghini, Porsche, Ducati, Bentley, among others.

drawback, it can be established the huge complexity for development, industrialization and logistic, and the spare parts management.

Criteria	Few variants	Unique reference wiring	Modular wiring
Number of combinations	$\sim 10^2$	$\sim 10^5$	$\sim 10^{10}$
Complexity of industrialization and logistics	Low	High	Very High
Costs of industrialization and logistics	Low	High	Very High
Unit cost of parts	High	Medium	Low
Complexity of development	Low	High	Very High
Complexity of assembly	Low	High	Very High
Pre-allocated elements	~5%	~3%	0%
Complexity of the spare parts management	Low	Low	High

Table 1: Comparison of the different wiring systems approaches

The Table 1 presents a comparison between the different wiring systems approaches. It can be appreciated that the enormous number of combinations, the low unit costs of parts and the low amount of pre-allocated elements, are the main advantages of the modular wiring approach, but its very high complexity in the development, industrialization, logistic and spare parts managements are the main drawbacks. Even though, the reduction of unit costs and mainly the great number of variants overcome the negative points and therefore this is the approach used in SEAT.

In the *modular* wiring system appears a very important concept: *basic* modules and *variable* modules. The *basic* modules are those which have to be always present in a car model regardless the rest of used equipment; they are a constant in the equation of the manufacturing process. All the same types of harnesses share their basic module. For instance, the harnesses of the type door (See Figure 3) must share the basic module of the loudspeaker and the door locking, but, the four doors may differ between them by the variable modules, like in the case the client picks-up the variable module of the mirror’s electrical motor module only for the pilot and co-pilot doors, but not for the back ones. This provokes that the harnesses of the pilot and co-pilot door change compared to the rest. The variable modules are associated to two types of functional systems, or what it is the same, to two families: *obligatory* and *optional*. In the obligatory families one module has to be selected mandatorily (in some very few cases two or more modules are allowed to be chosen). In the optional families is allowed either to select one module or no one, depending on the needs of a particular client.

Going back to the harnesses concept, a specific type or harness, say again the pilot-door, is formed by the basic modules, plus the module associated with the obligatory family and the module of the optional family, in case it is present.

To emphasize in the comprehension of the concept of *basic* and *variable* modules and *obligatory* and *optional* families, the next example applied to other topic is presented. Let's assume one is looking for a good house to live in, so one is interested in buying a property. A house construction company is offering us a property on plans in the following formal proposal:

Basic modules:

- ✓ Living room
- ✓ One double room
- ✓ One single room

Variable modules:

- Obligatory functional families:
    - Kitchen
        - ✓ Only with the stove included
        - ✓ Stove plus cook table
        - ✓ Fully furbished, including all the required appliances
    - Backyard
        - ✓ Full natural grass floor
        - ✓ Full paved floor
        - ✓ Full paved floor plus pool
- Optional functional families:
    - Guests room
        - ✓ Single room
        - ✓ Double room
    - Garden
        - ✓ Full furbished garden

In the previous example one can see that the company offers a house at least with living room, one double room and a one single room. Additionally, the customer has to select his preferred type of kitchen and backyard, those will be in the house, but it is up to the buyer to decide what options of them likes the most. Finally, the house may have or not guests room and garden, and in case the buyer wants any of those Sections, is up to him choosing the desired facilities.

An important aspect to highlight is that a specific module always points out to a particular family, i.e, a module is associated unequivocally to a functional family, thus when the module *stove plus cook table* of the example is mentioned, is directly know that it belongs to the family *Kitchen* and never to other one. Therefore, a module never belongs to more than one family.

In this example family *Kitchen* has three modules which share functions between them, i.e, the module 2 (*Stove plus cook table*) includes the stove available in the module 1 (*Only with stove included*), and the module 3 includes both 1 and 2. But the family *Guests room* offers two modules with totally independent functions. From the first case is realizable the fact that for a family is not possible (generally) to select more than one module, because in that case, it might appear redundancy of components.

The first case, where shared functions between modules of the same family are presented, is also applicable in the cars industry. With the aims of showing this point, next is summarized this situation for a real family of a car model: The *Airbag* family. In the Table 2 is displayed for this example the 6 modules belonging to the *Airbag* family with their functions associated; it is evident that the module 1 only encompasses the basic airbag, and the others 5 include this basic function, and also, for instance, the switch for disconnection of the airbag is present in the modules 3, 4 and 6. The Table 3 represents this same information but in a matrix form, answering to the question: *Is the function 'A' included in the Module 'n'?* with '1' and '0' for positive and negative answers, respectively. The definition showed in the Table 2 is not visualized in any of the electrical or wiring schematics, but as presented in the Table 3 for defining the presence of components and wires in the different modules of a specific harness. The Figure 6 presents the example in a graphic way, where it can be seen how the harness change dynamically in function of the selected module.

Module	Function
Module 1	Basic airbag
Module 2	Basic airbag+Side
Module 3	Basic airbag+Passanger disconnection
Module 4	Basic airbag+Side+Passenger disconnection
Module 5	Basic airbag+Side+Head
Module 6	Basic airbag+Side+Head+Passenger disconnnection

Table 2: Example of modules of the airbag family

Function	Mod. 1	Mod. 2	Mod. 3	Mod. 4	Mod. 5	Mod. 6
Basic airbag	1	1	1	1	1	1
Side airbag	0	1	0	1	1	1
Airbag disconnection	0	0	1	1	0	1
Head airbag	0	0	1	1	0	1
Complexity of the spare parts management	0	0	0	0	1	1

Table 3: Presence matrix of airbag functions

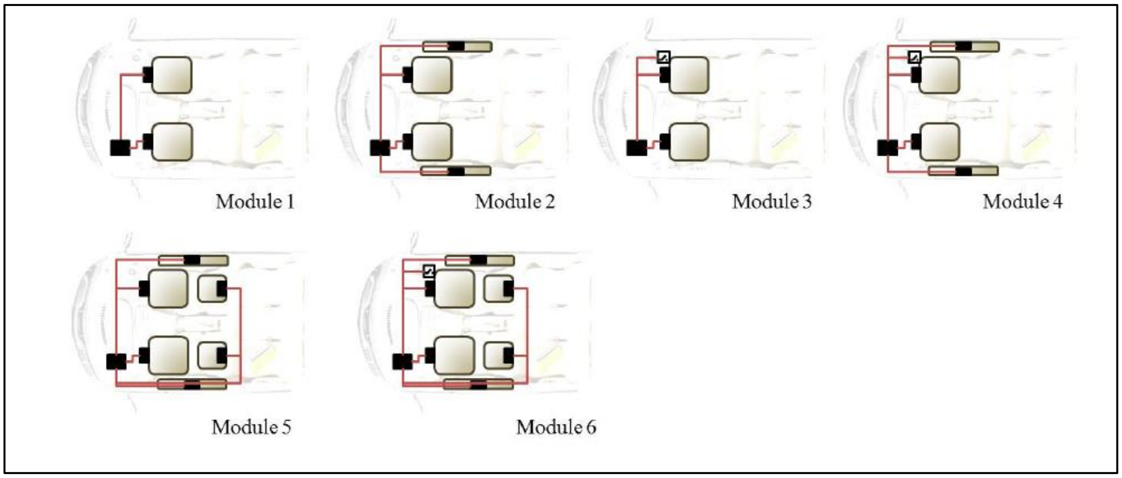


Figure 6: Example of modules of the airbag family

The distinction between obligatory and optional functional families is useful to take the concept of the demands of variety of the vehicle to the manufacturing of wiring harnesses. The wiring harnesses are assembled on special platforms, starting from wires and components belonging to the basic module. Meanwhile the variable modules are assembled in separated platforms, and they are later coupled to the basic wiring harness, until the customer-specific wiring harness is complete.

4.3. Wiring schematics

As it was explained at the beginning of this Chapter, after the pre-development stage of a car, the functional systems are defined (hardware network, component architecture), therefore the components to be used also and as well the electrical connections between them, being produced so the electrical schematics. Moreover, the electrical schematics lack of valuable information for manufacturing, such as size of wires, color and mechanical data of the wiring system. Therefore, the wiring schematics are carried out in order to provide this information. It is commonly known by the cabling engineers to assign to the wiring schematics a scope between electrical schematic and assembly drawings. The connectors between wires are described and the equipotential splices are also established, which are the soldering points. Manufacturing information of the cable such as type, material of the insulation, materials of connectors, etc. are provided.

An important point is that in the confectioning of these schematics, the definition of the fuses protection scheme arrangement, references and fuses electrical parameters are done. The fuses are located in fuse boxes, and their aim is to protect against overcurrent and short circuit the wires and the components connected to it. A special feature in these schematics is that they describe all the possible wiring systems resulting of the different variants according to the modules selected. This is achieved by assigning to each component in a car, a presence matrix, as showed in the

Table 3. Therefore, for a given combination of modules, a unique interpretation of the wiring schematic is obtained.

4.4. Wire routing definition

In the wire routing definition a more accurate -by means of 3D models-representation of the physical assembling into the inner part of the car's body is performed. This product requires as main inputs the wiring schematics and the *package* of the vehicle, which are documents containing the location of the components. This routing is done through specialized 3D CAD software tool, and one of the objectives is to define optimal trajectories of the wires taking into account the harnesses and devices locations. Other aim is to obtain the digital mock-up of the car.

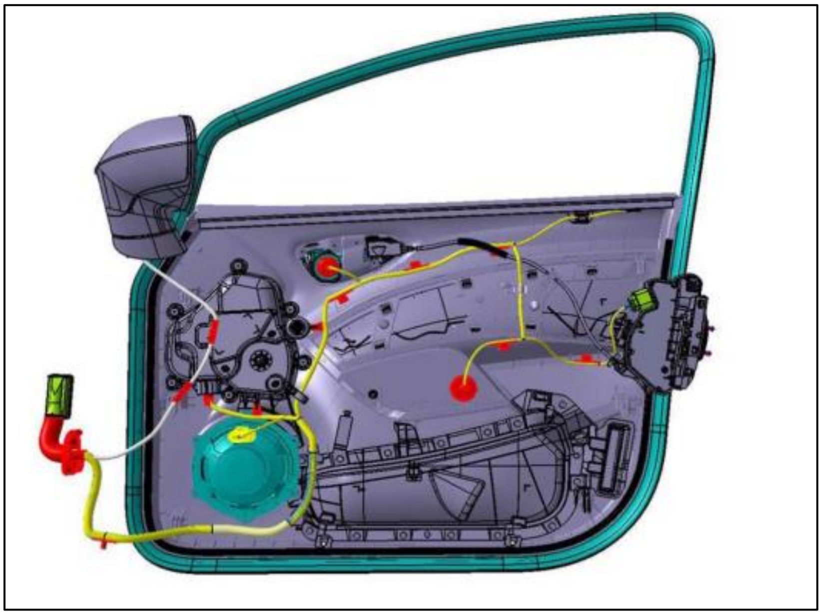


Figure 7: Example of the door's digital mock-up

An example of the door's digital mock-up is presented in the Figure 7, it is possible to see the yellow harness through where the wires are conducted. The software once this trajectory is defined is able to define the paths of each wire in an optimal way. From this can be analyzed the fact that the routing definition does not work with single wires, but with bundle branches of the harnesses which conducts a number of wires. At this stage are defined the thickness and maximum curvature of every harness branch segment; the position, orientation and type of fasteners and the amount of these elements, taking into account the three dimensional structure of the vehicle, and if it is necessary, some modifications of the bodywork might be carried out, for instance, to define holes and clamping points for hanging the harnesses. The protections for the bundle branches of the harnesses are also defined by means of the inspection of the surroundings of the wires; if there are metallic sharps or if the environment is corrosive, the designers establish the proper materials for covering the bundles, also in case there are no risks associated a simple wrapping tape is projected to cover them. As in the case of the wiring schematics, all the possible routings to be considered in function of the variants are defined, this is clear since with the number of possible combinations of modules change the location of components, sizes of wires, among others.

4.5. Assembly drawings

The assembly drawings are bi-dimensional plans, which represent all the information included in the wiring schematics and in the 3D routing definition. These drawings consider all the possible combinations of modules, and they are ready to be interpreted by the manufacturers for their fabrication- the designers hand out them in format paper-.This is carried out by means of specialized software tools, where the designer can add rules to ensure the obtained schematic is congruent, such as: electric coupling between the extremes of connectors, the loop of a circuit is complete, wires continuity at geometrical nodes (see Figure 2), and so on.

The assembly drawings provide the following important information:

- Physical description of each harness: lengths of the bundle branch segments, positions of all of the components and splices, type of wrappings for each branch, and so on.

- Wire list of each harness: for each wire of a given harness its part number is showed and the modules it belongs.
- Component list of each harness: for each component of a given harness its part number is showed and the modules it belongs.

The assembly drawings are formed by several pages, each one represent a type or harnesses which form the whole on board distribution network of the vehicle. These products also give important additional information like standards, modifications, detailed views, layouts, cut-views, etc.

In the Figure 8 is presented the abstract of the chain in the design of the harness in the modular approach. The use of the modular reference enormously eases tasks of modification, since they affect only to one drawing.

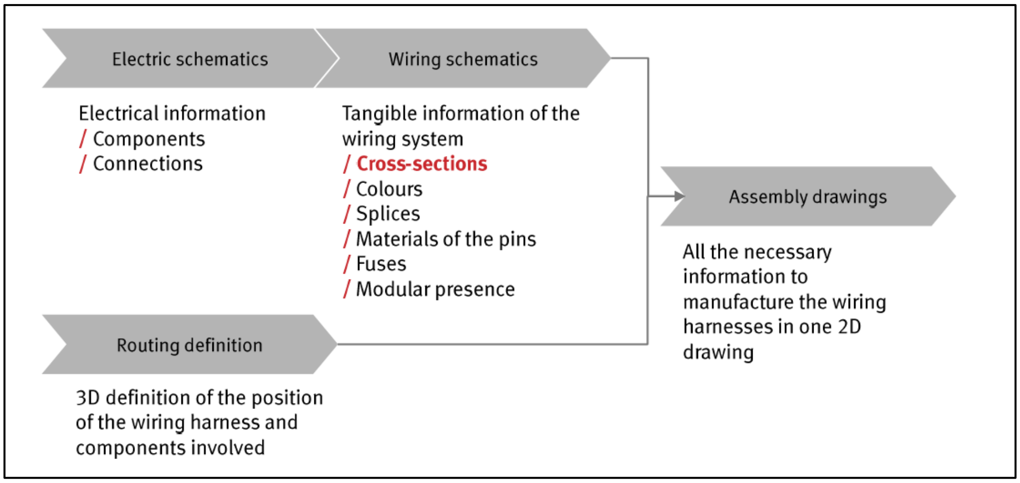


Figure 8: Summarize of the wiring system design scheme

4.6. The process of sizing fuses and wires

Currently, the designers belonging to the department of electrical and electronic development of the Volkswagen Group abide the recommendations contained in the international standard ISO 8820-1 [6], regarding the sizing of fuses for protecting DC low voltage distribution systems for road vehicles, and the internal regulation of the group, VW75212 [7]. Basically, in those standards, the guidelines are oriented to ensure the reliable operation under normal conditions (fault free) and protection of the components and wires under contingencies conditions, such as short circuits. For ensuring that a wire is reliably protected against the aforementioned conditions, the most critical factor to study is its thermal behavior, since it is covered by an insulator which cannot resist limitless high temperatures due to the Joule effect while it is carrying current and also due to external thermal loads, for instance, the exhaust pipe.

Every single wire in the cars is defined with a specific cross-section, which must be sized according to the load power, the thermal characteristics throughout the trajectory of it inside of the harness, i.e, the heat provided by the other wires that surround it and by devices under operation, its matching with the protecting fuse, its behavior under short circuit conditions, and the voltage drop in the terminal of the device which is being supplied with electricity. A wire is, basically, formed in the core by copper strands and is wrapped for mechanical protection and isolation with polymers. The wiring system in a vehicle must withstand as much as the expected life of the car under the different types of degradations observed, which are documented in [6] and [7]. One of such degradation effects (and the most critical) is the one produced by the increase of temperature that the polymers are undergone under a specific range of time. For these reasons, in [6] and [7] the recommend maximum temperatures for wires and their exposition time are detailed, in such a way that the maximum degradation rates are not reached, and thus, with uncertainty but acceptable margin of probability, to guarantee the life cycle of the wiring system. Since the endurance of the wire insulator is directly related with the cross-section of the cooper core, the engineers must select a wire with this parameter in such a way that the temperatures operation times are abided and the critical voltage drop as well. The previous commented temperatures ranges are the next:

- T_{LT} Long-term: Wires withstand this temperature (105 ℃) or less for 3000 hours

- T_{ST} Short-term: Wires withstand between T_{LT} and T_{LT} (130 °C) for 240 hours
- T_{OL} Thermal overload: Wires withstand between T_{ST} and T_{ST} (155°C) for 6 hours.

Code	$U_{B_{min}}$	$U_{B_{max}}$	**Description**
a	6 V	16 V	For functions to be preserved during the process of starting-up
b	8 V	16 V	For functions not to be preserved during the process of starting-up. Use this point only when is not possible to use a, c and d
c	9 V	16 V	For function to be preserved when the engine is OFF
d	9.8 V	16 V	For function to be preserved when the engine is ON

Table 4: Voltage regulation in the Volkswagen Group [8]

Regarding the checking of the voltage drop throughout the extension of the wiring system, it is clear to know that the main constraint of voltage must be given by the device which is being supplied; in the datasheets of a component, is likely to be able to find the recommended operation conditions of it, among which the minimum voltage in terminals is specified. However, in the Volkswagen Group, the database of components is really huge, making really difficult to analyze device by device to determine such critical values. Therefore, in the Group is tried to abide the regulation VW80000 [8] to determine the minimum voltage in terminals of a device, as it can be appreciated in the Table 4; there, the range of voltage operation is presented in function of 4 different scenarios: devices to be operated in the starting-up process, not to be operated in such condition, to be operated when the engine is OFF and when it is ON. Even though the low voltage distribution system in a vehicle is so-called to be 12 V, under normal charged conditions the operation voltage is 14 V, and in some cases to ensure a voltage of 9.8 V minimum may be critical. Likewise, a way less conservative approach is also followed, in [9] is established a maximum voltage drop of 1.5 V for general loads (when the result of $50mV/A$ is higher than 1.5 V, otherwise the same value $50mV/A$) and for illumination with loads lower than 15 W is indicated 0.7 V and for higher 1.1 V. Currently, the checking of voltage drop is carried out too empirically by the designers, in many cases doing assumptions too far to be real, which might distortion considerably the required robust system that is intended to have, for this reason the tool for calculating rapidly and precisely the power flow is important. In the Figure 9 is displayed the process for sizing fuses and wires in the Group Volkswagen, where it can be seen the presence of the block in charge of checking the voltage drop enclosed in a red rectangle.

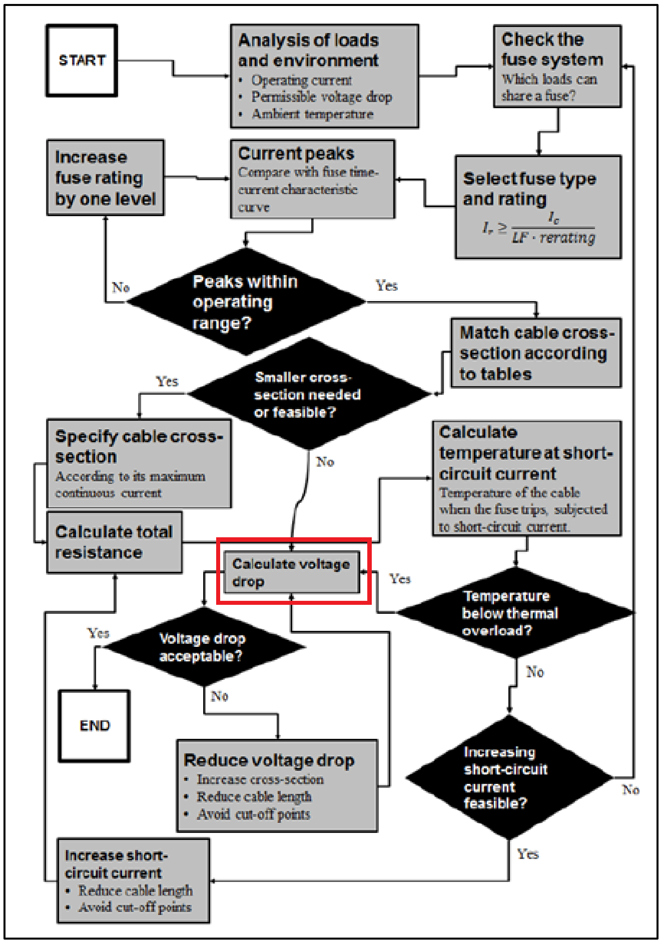


Figure 9: Wire and fuse dimensioning process diagram [7]

In the Figure 9 it is visible that, naturally the first step in the chain is to analyze the loads to be supplied and the environment, in the first, the operating currents, inrush currents, maximum currents are determined, and in the latter, the surrounding temperature of the wire is established.

After, it is carried out the fuse system distribution, trying to decrease the number of fuses to be places in the box, due to space constraints and also even costs; here is determined if two or more feeders can share one fuse or individual protections have to be projected. It is worth it to mention that in case a fuse is shared between several feeders, the gotten drawback is the oversizing of the cross-section of wires, in order to protect them successfully to reach the temperatures for the time given previously (T_{LT} in faultless conditions and T_{OL} under faults). The next step is to calculate the rated current of the fuse, which is done by dividing the maximum operating current of load or loads, over a Load Factor (typically 0.7) and the rerating factor that takes into account the change of temperature in the surroundings of the fuse out of the nominal condition tests (it can be taken as 1 in case is not necessary the correction). Afterwards, the inrush currents or current peaks of the loads are compared with the time-current characteristic curve of the fuse, and it is defined if nuisance tripping is not obtained; in case such peaks are not located under the curve, the rated current of the fuse is increased until that point is under the characteristic curve of the fuse. Once the rated current of the fuse is calculated the matching of the cable cross-section of the wire is performed. The Figure 10 presents an example for matching a fuse and wire time-current characteristic curves.

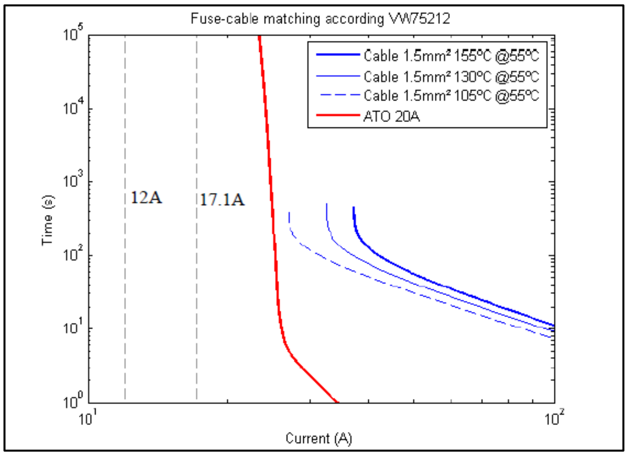


Figure 10: Example of the matching wire-fuse according to [7]

In the Figure 10 the lines of 12 A is the maximum operating current of load, the line of 17.1 is obtained by dividing 12 A over 0.7 (load factor), hence, the fuse ATO 20 A is selected. From this point, pre-established curves available in [7] are used by the designers in order to choose the cross-section of the wire, in such tables is guaranteed that the recommended value is far enough to ensure a reliable operation of the cable under nominal and critical conditions, as it can be seen in this figure, all the characteristic withstanding curves of the wire (blue ones) are above the fuse curve (red one). The dashed blue line represents the time- current withstanding capability of the wire for T_{LT} (nominal operation) considering an ambient temperature of 55 ℃, and the thick blue lines for T_{OL} (contingency operation). Basically, such wires curves explain how long it takes for the cable to reach the specific temperature for a given value of current. The engineers must check that under the most critical short circuit current the thick blue line is above the red line; under this condition the complete protection is ensured, otherwise the wire is exposed to have failures in its insulation because after a time given by its characteristic curve, it will reach the T_{OL} temperature and it will continue under that condition an undefined time (the fuse will not burn) and it may damage after the exceeding of the established 6 hours. The most critical short circuit is that one which provides the lower value of current, since as it can be appreciated in the Figure 10, the lower the current, the smaller the range between the fuse and wire curves.

In the step *'Smaller cross-section needed or feasible'* is decided whether is possible to decrease or not the cross-section of the wire, taking into account that in the previous step that value might be oversized due to a too conservative approach, in case this is not desired to do, the next step is to check the voltage drop and finally if it is satisfactory the process is finished, otherwise strategies to reduce the drop are carried out, such as increase the cross-section, reduce the cable length or avoid cut-off points until the approval criteria is reached. Regularly, in the design process this is the path followed, but current researches work in the optimization of the cross - section according to thermal analysis through computational tools [1], [2] and [3]. The basic scheme to reduce the cross-section is visible in the diagram flow, after decreasing this value and the calculation of the total resistance of the cable between the fuse box terminal and the component's terminal, is evaluated again if the matching of curves is satisfactory (for the lowest value of short circuit current), with positive results, the process is pointed again to the voltage drop checking; with negative results, is analyzed if the short circuit current may be augmented by means of the reduction of the wire length, with the aim of going out of the most critical condition as explained before, otherwise the fuse system must be rearranged, i.e, to assign to the wire its own protecting fuse in order to decrease the rated value of the fuse and consequently diminish the size of the wire; this is a undesired solution, because the fact of increase the fuses is a worse condition than decreasing the cross-section, in economical and logistical points of view, hence if finally this is the only solution the optimization process is cancelled and the big cross-sections are used. When the cross-section of a wire is changed, the total resistance of cables connected to the last one are affected, therefore when optimizing is necessary to carry out iterations and updating systematically.

Additionally, in the Chapter IV the importance for optimizing the cross-sections of the wires was discussed, for that the importance of exploring techniques that allow implementing them in the design of the wiring systems of vehicles. The systematic voltage drop checking plays an important role in this situation, since rapid, precise and robust simulations of power flow calculations would make part of the optimization systems.

CHAPTER V

PROCESS OF ORGANIZATION OF FACTORY DATA

5.1. Bills of Materials (BOMs) and Wire lists

As it was discussed in the previous Chapter, the last link of the chain in the design process of the wiring systems is when the assembly drawings are obtained. In those drawings, information such as the physical description of each harness (lengths of the branches, location of both mechanical and electrical components), type of wrapping materials, etc., in general, all the required information for manufacturing the harnesses is available. To obtain these drawings, a specialized software tool is used (*LDorado*), which basically is a viewer-manager of the XML files obtained after other software tool joins into a single design the outputs of the wiring schematics and the 3D routing.: This software tool is the *Elena*, and the softwares for building the wiring schematics and 3D routing designs are *EB Cable* and *Catia*, respectively.

Likewise, for each harness inside of a vehicle, excel files with valuable information of the components and wires forming it are available for their use by the software *LDorado*; in such files are deployed in the same format of the presence matrix of the Table 3, if an element *X* is belonging to the module *N*, and in this way, a specific interpretation of the drawings is possible to carry out when is defined which modules of a particular car model are selected by the final customer, and finally, the elements to manufacture the big size harnesses are picked-up and assembled. This methodology is very useful for managing the logistics and construction processes taking into account the complexity associated to the modularity of this wiring system approach.

When the *basic* modules, along with the *variable* modules of a car model are activated, unique harnesses designs are generated automatically; therefore having the particularity of being able to offer unique solutions and thus saving material costs and the pre-allocation of elements is avoided as well.

The *VOBES* is the Volkswagen On-Board Network Development System [10], and gives the tools to develop the system wiring plan, for constructing the wiring harnesses in the cable wiring plan and also to laying the wiring harnesses in the virtual vehicle, and thus documenting the construction steps in wiring harness drawings. The *VOBES* allows the standardization of the processes adopted in the whole chain of the wiring design, and it defines all the steps and softwares previously explained, in other words, is a functional package of softwares, guidelines and processes used for the Volkswagen Group for the on-board network development.

The BOMs, standing for Bills of Materials are excel files generated automatically by the design software of the assembly drawings (*LDorado*), they display not only the list of electrical and electronic equipment of a vehicle model (defined in the basic engineering stage along with the functional systems of a vehicle), but also mechanical elements such as clamps, connectors, protectors, bases, bolts, washers, and in general all the required materials to build a harness. The on-board electrical distribution system is formed by several types of harnesses, as discussed and showed in the Figure 3; for each one of those types a BOM excel file is available after its configuration regarding which information is desired to be contained and displayed on it. The Table 5 shows an example sheet of a BOM excel file; it can be appreciated that with the number *1* is indicated the family that is present in this harness, is also mentioned the name of such family, in this example *family01*. As it was discussed before, a harness might be able to content devices for carrying out more than one function, i.e., to encompass more than one family. For those cases, the number *1* will appear in the same amount than the number of families present in the harness; their description will be available in other rows under the breaking down of the components belonging to the previous family. In the Table 5 is realizable that the *family02* is listed in the row 19 after the descriptions of the components belonging to the *family01*, in the coming rows after the 19 the components and modules associated to the *family02* would be presented, and in case a third family is required, it will show up in a subsequent row after the breaking down of components of the *family02*.

1	family01					
2	V1	module01				
2	V2	module02				
2	V3	module03				
0	Bezeichnung	VOBES-ID	Teilenummer	V1	V2	V3
3		vws_fuse01.1	pn010	x	-	-
3		vws_fuse01.1	pn011	-	x	-
3		vws_fuse01.1	pn012	-	-	x
3		vws_01.1	pn021	-	x	-
3		vws_01.1	pn022	-	-	x
3		vws_02.1	pn031	x	-	-
3		vws_02.1	pn032	-	x	-
3		vws_02.1	pn033	-	-	x
3		vws_03.1	pn041	-	-	x
3		vws_04.1	pn051	-	-	x
3		vws_05.1	pn061	-	x	-
3		vws_05.1	pn062	-	-	x
3		vws_09.1	pn073	-	-	x
2	family02					

Table 5: Example sheet of a BOM file

After the family, with the number *2* are indicated the list of modules associated to the family described upper. In this example, for the *family01* are associated the modules *V1*, *V2* and *V3*, along with their descriptions *module01*, *module02* and *module03*. In the row with number *0* are deployed the set of columns for describing the components, such as, the most important: *Bezeichnung* or description in English, *VOBES-ID* which is the identifier of the component for its location inside of the vehicle according to the package (as discussed in the 3D drawings). This is supposed to be more or less like a physical address of the component, and hence, is a unique parameter in the car; *Teilenummer* or part number of the component and the next columns are the modules associated previously. On other hand, with the number *3* are listed the components belonging to this family.

Basically, the BOM is a presence matrix, where it is specified with an *X* or a binary *1* if a component with location *VOBES-ID YYY* and *Teilenummer WWW* belongs to the module *Z*. In this example, it can be seen that for the module *V1* the component with location *vws_fuse01.1* has a device with part number *pn010*, for the module *V2 pn011* and for *V3 pn012*. It means that for different modules the same component might have several references (part numbers or *Teilenummer*) being differentiated by, for instance, the electrical features between them, i.e., the devices *pn010*, *pn011* and *pn012* can have different maximum (or nominal, peak, etc) current capabilities. This is properly projected in the design stage. From this, it is important to remark that a component with a given part number can belong to more than one module or even to none of them, and also a component can be present in more than one family and/or wiring harness.

In the example of the Table 5 are presented components such as fuse boxes, ECU and consumers -the last two are conceived by the engineers of SEAT as *black boxes* with respective datasheets for determining their consumptions of current, it means, for the sizing of wires and fuses, dynamic and detailed analysis of the behavior of the electronic inside of these components are not taken into account-. This will be discussed in next Chapters. Additionally, it is expected to have in the Table 5 not only electrical and electronic components, but also mechanical elements for manufacturing the harness, which also are going to be dependent on the presence matrix and the modularity.

The wire lists are excel files produced automatically by the software *LDorado* which, as mentioned before, is a viewer-manager of XML files generated by the merger software *Elena*. The wire lists display all the single cables which are inside of a specific type of harness, as in the case of the BOM’s files, there are as many excel files as the number of types of harnesses projected in a particular car model. The Table 6 presents an example sheet of a wire list file.

1	family 01										
2	V1	module01									
2	V2	module02									
2	V3	module03									
0	Ltg-Nr.	Verbindung	von	Pin	nach	Pin	Leitung	Länge	V1	V2	V3
3	1		XF.vws_fuse01.1	1A	spl01	1	wire01	0,28	x	-	-
3	2		spl01	1	vws_01.1	1	wire01	0,51	-	x	x
3	3		spl01	1	vws_02.1	1	wire01	0,70	x	x	x
3	5		spl01	1	vws_03.1	1	wire01	0,94	-	-	x
3	6		spl01	1	XF.vws_fuse01.1	4	wire01	0,71	-	-	x
3	7		vws_01.1	2	vws_02.1	4	wire_prim	0,60	-	x	x
3	8		vws_02.1	2	vws_03.1	3	wire_prim	0,64	-	-	x
3	9		vws_03.1	2	XA.vws_mp_02.1	1	wire01	0,71	-	-	x
3	10		XF.vws_fuse01.1	4A	vws_04.1	1	wire02	0,55	-	-	x
3	11		vws_04.1	2	XB.vws_mp_02.1	1	wire02	0,86	-	-	x
3	12		vws_01.1	3	vws_05.1	1	wire02	0,25879841	-	x	x
3	13		vws_01.1	4	SPM01	1	wire01	0,67	-	x	x
3	14		vws_02.1	3	SPM01	1	wire01	0,25	x	x	x
3	15		vws_05.1	2	SPM01	1	wire02	0,94	-	x	x
3	16		XF.vws_fuse01.1	1A	spl01	1	wire00	0,44	-	x	x
3	17		SPM01	1	vws_mp_01.1	1	wire01	0,78	x	x	x
3	18		XF.vws_fuse01.1	3	XC.vws_fuse01.1	1	wire01	0,92	-	x	x
3	19		XF.vws_fuse01.1	5A	spl_apr	1	wire01	0,95	-	-	x
3	20		spl_apr	1	XF.vws_fuse01.1	50	wire01	0,65	-	-	x
3	21		spl_apr	1	vws_09.1	1	wire_prim	0,61	-	-	x
3	22		vws_09.1	2	XF.vws_fuse01.1	51	wire_prim	0,97	-	-	x
3	23		vws_09.1	3	SPM01	1	wire_prim	0,55	-	-	x
3	26		XF.vws_fuse01.1	53	spl01	1	wire02	0,95	-	-	x
1	family 02										

Table 6: Example sheet of a Wire list file

In the Table 6 it can be appreciated that the structure of the wires presentation is pretty similar than the aforementioned for the BOM files. With the number *1* are displayed the families, with *2* the modules and with *3* the wires which form the harness. Again, the number *0* indicates the kind of information to be presented for the wires, and also, a presence matrix approach is followed for determining if the wire *Z* belongs to module *N*. In the row *0* the information related to wires is:

- *Leitung-Nr.*: German for number of *wire reference*. This is the numeration of the wires according to the project. Each family has a particular list of wires and numeration.
- *Verbindung:* German for *connection description*.
- *Von:* German for *from*. Indicates from where a wire starts its connection. In this column can appear terminals of components and also soldered splices. In the case of components here is indicated the location and the slot.
- *Pin:* In this column is indicated the pin of the component associated to the previous column. In case of splices, is shown 1 by default. One has to take into account that the terminals for connections of an equipment are addressed by means of slots and pins, being the first one the set of the second, thus, a component can have two slots, for instance, *A* and *B*, and the slot *A* can have pins from 1 to 10, and the *B* from 1 to 5.
- *Nach*: German for *to*. Indicates to where a wire finished its connection. In this column can appear terminals of components and also soldered splices. In the case of components here is indicated the location and the slot.
- *Pin:* In this column is indicated the pin of the component associated to the previous column. In case of splices, is shown 1 by default.
- *Leitung:* German for *wire reference*. In this column is indicated the factory reference number of a wire. Through this parameter is possible to find information of factory such as materials used, type of wire, cross-section, specific resistance, etc.

- *Länge:* German for *length*. In this column is shown the linear length of the wire. Additionally, in other columns might appear other information related to the wire, but these ones are the most relevant.
- *Modules:* In the subsequent columns are listed the modules associated to the family with the respective indicator to establish if a wire belongs or not to one, more or even none of the modules.

It is worth it to comment that a specific wire can only be connected to one point in both sides and can only belong to one harness. On other hand, a wire can be present, as mentioned before, to several modules, but never to more than one functional family.

5.2. *QT* – Technical information file for simulations

The QT is an excel file which summarizes the information of the BOM and wire list files. QT stands for *Quadern Tècnic* in Catalan, and it means Technical information file for simulations. The QT file condenses the information required for being able to perform simulations in the on-board network wiring system in SEAT, such as the under-development software for optimizing the sizing of wires and the power flow calculation tool, which is being explained in this work. The QT is expected to be obtained systematically for any model of car, in such a way that through visual basic macros all the sheets of this file are elaborated automatically; with this aim, engineers of SEAT are working currently in this matter, however, some of the parts of the QT files are impossible to be automatized, taking into account the way how the factory information is provided, which causes that in many situations the human interpretation is indispensable to process it. In order to automatize the QT, not only the BOM and wire lists files of a car model are required, but also the XML files provided for the merger software *Elena*. In the scope of this work is not considered the aforementioned construction of the QT, since this is a project previously conceived in SEAT for other objectives, such as the commented optimizer tool;

moreover, is important to explain the content of it, bearing in mind that is an essential input for the power flow simulator.

Currently, the QT file model counts with 13 worksheets, these ones are going to be explained in the next:

- List of drawings: In this worksheet are deployed the set of drawings and the corresponding BOMs and wire list files to each one associated to a wring system of a car model. Basically, it has three columns: *drawing*, *BOM file* and *wires list file*, and in each row is written the name of the files using the same identifier available by the software 2D *LDorado*. In general for each harness presents in a car vehicle there is a drawing plan and are associated a BOM file and a wire list file.
- Supplementary families: In here is listed in a column the name of the functional families which have the particularity of allowing the selection of more than one module. In previous Chapters, it was explained that in very few cases some families conceive this idea.
- Battery: This worksheet is designated to contain the information related to the battery of the system, the provider of energy in the distribution on-board network. It is formed by 5 columns. In this sheet the main objective is to indicate the location of the positive and negative battery terminals. It is projected to storage more than one battery in case of electrical vehicles which may use them. So, the first column is for listing the name of the drawings where the battery appears, the second column shows the VOBES-location of the equipment, and the next two columns are for indicating the slot and pin or the terminals, finally the last column describes the voltage potential of the specific terminal.
- Locations master: This is a very useful worksheet since in here is indicated the list of all the locations linked to the package of a car model. In its column location is written such list of VOBES tags. The column *drawing* specifies the name of the drawing where the

location can be found. The column *description* shows an alphabetic chain for which can be understood the function or nature of the element represented by the given location. The column *type* basically encompasses six different variations: Fuse box, battery, component, coupling, soldered junction (splice) and ground bolt. An element tagged as a component might be in broad terms either an ECU or a consumer (including interrupters, sensors and among others), for both elements the electrical features are available in the database of SEAT. The remaining two columns are expected for containing information related to the fuse boxes, such as the name and the surrounding temperature; this may be important for correcting the time-current characteristics of the fuses when the sizing of a wire is carried out and the matching of curves is performed.

- Wires segments: In this worksheet physical information of the harness is available. Here three columns display the following content: In the column *segment_id* is contained the string identifier of a harness segment; in the Figure 2, for instance, it is realizable segments such as *s1*, *s2* or *s3*. In the column *wire_number* is showed the number of a wire according to the numeration of wires present in a harness, i.e., this is not a general wire list. Finally, in the column *harness* is expressed the drawing where a particular *segment_id* can be found. The main purpose in here is to mark the trajectory followed by a specific wire inside of a harness through the different segments which compose it.
- Segments_junctions: In this worksheet is described how the different segments of a harness are assembled between them. So, in the column *segment_id* are listed all the segments, in the columns *junction_id_1* and *junction_id_2* are deployed the string identifiers of the extremes of the segments, and the column *harness* simply shows again the corresponding drawing. As it has been insisted before, one has to take into account that for each harness there is a drawing. By last, in the column *Tamb* is indicated the surrounding temperate of the segment, which is dependent of the particular location of a

segment inside of the vehicle because of the presence or not of thermal loads such heat dissipaters, and so on.

- Juctions: In this worksheet are listed the aforementioned geometrical junctions, by indicating the harness where they are present, and in two other columns are specified the geometrical coordinates *X* and *Y* in order to make a graphical representation of the harnesses in case is needed.
- Modules: In this worksheet are showed 4 columns: The column *module_name* shows the names of the modules available in a car model. In the field *description* is written for each module the corresponding description associated; on other hand the *pr_formula* expresses a short internal formula used in the Volkswagen Group in order to indicate to a particular module the components and functionalities which compose it. Also, in the column *family* is linked to which functional group is belonging each one of the modules; however as discussed in the previous Chapter a module is associated unequivocally to a family, but this information allows establishing with precision the relation. Finally, the column *mix* is a measure of frequency about how much is selected a particular module by the clients, taking into account the historical of sell in the company. This may be useful for applications such as the sizing of some components by bearing in mind if it is highly demanded or not.
- Device pinout: This worksheet is very determinant for the power flow simulation tool, because in here is established how is the direction of power flow inside of a component by means of establishing of the electrical relationships between the output and input pins. In order to automatize the elaboration of this field, the responsible engineers in SEAT play with the characteristics of the pin types according to pre-established names, for instance, for a ECU, the input power pin is named as *Senke Last* and the output power pin as *Senke Ausgang*, meanwhile the ground connection as *Senke Komponent*. In the same manner the communication terminals are tagged as *Senke CAN*. This is evident in the

next worksheet. Consequently, in here there are 6 columns, in the first is specified the description of the component, in the next one its part numbers is represented, and the next four columns inform about the input pin, slot and output pin and slot, respectively. This sheet may be complicated to automatize since the human interpretation is required.

- Pin master: The pin master is a table where is contained all the information from the *System42* (storage software of datasheets of all the components used in the Volkswagen Group) regarding consume of currents of the components pin by pin. This worksheet is deployed by indicating the description of the component, the part numbers, the slots and pins of each component for a given part number, the pin type description where is described the different types explained in the previous point, the pin characteristic (material of fabrication of the terminal), the minimum voltage, the cycle and until 6 different types of currents and two different times (for transient behavior) for each terminal of each part number associated to a component by location. The meaning of the electrical parameters such currents and times will be explained in the next Chapter.
- Couplings: In this worksheet are indicated the elements classified as couplings, and is determined the electrical connections between them. Thus, in one column is specified the VOBES-location of the coupling of one extreme and other column the location of the coupling of the other extreme.
- Fuse box pinout: In this worksheet is done basically the same than in the device pinout but for the elements inside of a fusebox. Thus, here is showed the VOBES-location of the fusebox, the input slots and pins, the output slots and pins, the connection type which are so far fuse, plate, relay coil and relay switch and their connection string ID.
- Fuse box modularity: In this worksheet are deployed all the fuse box connections mentioned before, with their string ID, and additionally is indicated the modularity of them, i.e. to know if for a given module a given connector element belongs to it or not with a specific part number.

CHAPTER VI

DEFINITION OF THE INPUTS FOR THE POWER FLOW VISUALIZATION/CALCULATION TOOL

6.1. Pre-processing of the factory data files

In the Chapter V has been explained all the process for gathering the factory information data of a car model in SEAT. Basically, there are three set of excel files: The QT, which is roughly an abstract, and the bills of materials and the wire lists, which are available in a number equal to the amount of harnesses forming a vehicle. However, with the aim of performing the whole processing of such information, it is required to use any specialized software able to deal with information organized as structures and matrices; for that objective the tool MATLAB R2013b© has been applied. The systematization into MATLAB of the aforementioned excel files is part of ongoing projects in SEAT, especially the already commented optimizer tool for the wiring system.

All the information is firstly pre-processed in such a way that a structure named *dataContainer* is obtained; in order to automatize the generation of the inputs for the power flow visualization/calculation tool a processing of that structure is carried out by means of algorithmic routines implemented also in MATLAB, therefore is important to describe what its content is, and thus in next, to describe the standardization of the inputs and the processes to get them.

The structure *dataContainer* counts with 21 fields; information of each of them is deployed in next:

- qt: This field is other structure. In here it is loaded all the information contained in the QT file, hence each field of this sub-structure is one of the worksheets of the original excel document. In order to do this, pre-established functions of MATLAB are used, such as *readtable* which is useful to create tables by reading column-oriented data from a file.
- Modules: In this field is showed the same information already explained in the worksheet Modules of the QT file.
- WireModules: In this field an important processing of the wire list files is performed since in here is available a table with information such as a consecutive numeration ID of all the wires of the wiring system, grouping all of them no matter if they belong to different modules, harnesses and/or families. In the Table 6 was displayed an example of a wire list file, it was shown that in the column *Leitung-Nr.* a harness-discriminated numeration of the wires is available, but in order to identify individually each wire, an unique ID is associated in this field. Additionally, a string wire id is linked to each wire in order to name them as a part of a harness. In this field is also indicated each wire to which module and family it belongs; from here is worth it to remember that a wire with an unique ID might be part only of one family, but may belong to more than one module of that family and it is restricted to be part of only one harness as well (the concept of a harness conceived as a manufacturing unit allows understanding this particularity).
- NodeList: This is also a very important field, because in here is elaborated a general assigning of IDs to all the elements considered to be nodes in the wiring system. This is the equivalent to the previous wire modules table, but of course regarding electrical junction points of the whole system. As nodes of the system are considered basically the next types: components' terminals, couplings' terminals, battery's terminals, fuse boxes' terminals, soldered splices in the positive potential layer and in the negative potential layer, ground bolts, and in general all the points were the wires can be connected to, i.e., a wire with a specific ID can be connected only between two nodes, which are at the

same time are described in this field. The table deployed in this field has 7 columns, in the first one is indicated the unique ID of a node, in the column two is described the type of a particular node, that as it was already discussed might be up to 5 variants, the column 3 contents an assigned string node ID where is merged the location of the node's element and the information of the terminal (slot and pin); in this manner the next three columns show the location, slot and pin of the node, respectively. The nodes not belonging to an element type component, battery, coupling or fuse box have by defect an empty slot and a pin equal to one. Finally, in the last column is indicated the description associated to the element where the node belongs, this description is coincident with the one shown in the worksheet of the QT *location master*.

- WireList: In this field a table regarding more information about the wires forming the wiring system is deployed. For each wire with a specific ID, is indicated its harness-discriminated numeration, the harness where it belongs, the nodes IDs where it is connected (without having into consideration any order, i.e, the node 1 might be for one wire the higher potential point or for other the lower), this is a direct translation from the wire lists, in the Table 6 is realizable that for each wire is indicated the two connection points (*von* and *nach*), so in this Wire List table such points are expressed instead with their corresponding node ID already assigned in the previous field. The part number of each wire is also listed in this table, one must bear in mind that one or more wires (each one with different ID) might have the same part number, because simply they can use the same reference of cable. The next columns describes information about physical features of the wires, such as the gauge (cross-section in mm^2) and the length (in meters). It is important to remark that the last three fields (*WireModules*, *NodeList* and *WireList*) are master tables with information about the wires and nodes of the wiring system, i.e., if one needs to know certain information about a single wire or node, one goes to the interest table and with the ID is possible to find it, each row of those tables represent a different element with its particular data.

- Def_InOut: In this field is contained the same information already described in the work sheet of the QT *device_pinout*. In this definition is expected to have the relations between the input and output pins, without having connections between the inputs themselves or between the outputs themselves.
- Couplings: In this field is indicated the same information of the worksheet of the QT 'Couplings'. However, additionally is elaborated a numeration of the electrical linking couplings (ID of electrical coupling connections); each row represents an electrical junction formed by two couplings with locations also indicated.
- Fuses: In this field is indicated in general, the same information of the worksheet of the QT *Fuse box pinout*. But as in the case of the field Couplings, in this one also is done a consecutive and unique numeration of the connections inside of a fuse box. Therefore, the connections type fuse, plate, relay coil and relay switch are all grouped together for numerating purposes.
- FusesModularity: This field contains basically the same information of the worksheet of the QT *Fuse box modularity*, but as main difference instead of displaying the string ID of each connection, the numeric ID assigned in the previous field *Fuses* is showed. Also, the part number to each connection with the modules and family where it belongs is presented.
- Devices: In this field are listed with a unique ID all the elements categorized as components in the work sheet of the QT *Locations master*. The aim of this table is to identify unequivocally all the components of the on-board system of a vehicle, therefore in here is deployed the mentioned ID and the corresponding VOBES-location of the component.

- ➢ DevicesModularity: In this field is presented a table with all the elements categorized as components, fuse boxes, battery and couplings, by indicating their modularity and part numbers, i.e., for an element with a specific location all the modules and families associated are displayed and the part numbers for each case as well.
- ➢ Pins: The information contained in this field is the same than the already described in the worksheet *Pin master*. As it was discussed in the previous Chapter, for each terminal is deployed up to 6 different types of currents and up to 2 different times. The meaning of each variable is explained in the following:
  - *i1:* Nominal current of consumption. This parameter has been measured for 5 minutes at 14 V (See Figure 11 as an example).
  - *i2:* Maximum current in the worst case (See Figure 11 as an example).
  - *i3:* Maximum peak transient current at start-up or break (See Figure 11 example)
  - *i4:* Minimum current of consumption. Standby current.
  - *i5:* Tailing current
  - *i6:* Bias current
  - *t1:* peak transient time for the current *i3*
  - *t2:* tailing time for the current *i5*

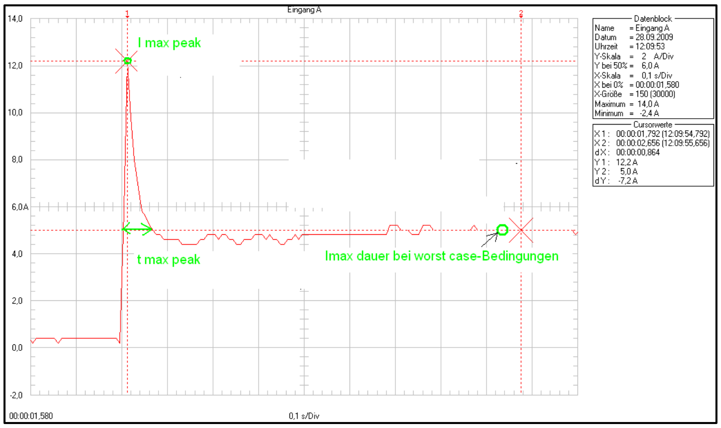


Figure 11: Operation consumption of a fuel pump [12]

- WiresSegments: In this field is presented a table with the same information than the homonymous worksheet of the QT, but additionally is added a new column indicating the general ID of a wire.
- SegmentsJunctions: In this field is presented a table with the same information than the homonymous worksheet of the QT.
- Junctions: In this field is presented a table with the same information than the homonymous worksheet of the QT.
- Battery: In this field is presented a table with the same information than the homonymous worksheet of the QT.
- WireMaster: In this field is deployed a table with information regarding manufacturing data of the wires discriminated by part number. For each wire part number is indicated the conductor material, the isolation material, the cross-section are of the conductor part and for the isolation part, the specific weight (grams/length) and the specific electrical resistance (ohms/length) at nominal temperature.
- VW75212: In this field the information of the regulation [7] is summarized. In here there are available two tables about the matching fuse-wires and the current-temperature curves of the wires.
- FuseRefs: In here are described the references of the fuses available in the database and their rated current.
- FusesCurvesData: In this field is available the points current-time of the characteristic of each fuse reference.
- All series sets: This is the most determinant field of the *dataContainer* because in here are established all the possible paths from the positive terminal of the battery to the negative terminal of the same. In order to explain this structure the example of a simple electrical arrangement in a vehicle of the Figure 12 is displayed.

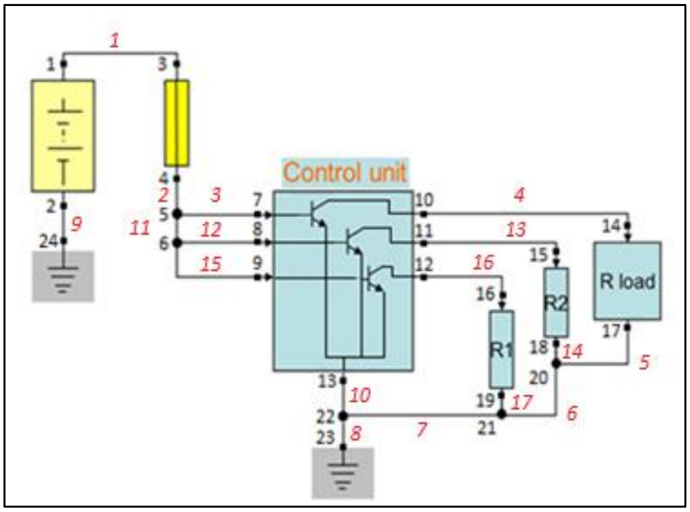


Figure 12: Example of an electrical arrangement in a vehicle [13]

Let’s consider that the electrical schematic shown in the Figure 12 has associated one BOM and one wire list excel file, and of course its QT file; it means that all the wires belong to one harness. Additionally let’s consider that the components and wires listed in the example have only one module associated, consequently representing the same functional family. After the three excel files (BOM, wire list and QT) are introduced to MATLAB, automatic pre-processing routines are run and all the fields of the *dataContainer* previously described are obtained. For this example samples of the most important fields are generated, they can be seen in the next:

Node ID	Type	Node id_0	Location	Slot	Pin	Description
1	Positive battery terminal	A.21A	A.2	A	1	Battery
2	Negative battery terminal	A.21B	A.2	B	1	Battery
3	Fuse box node	vws.box1.11A	vws.box1.1	A	1	Fusebox
4	Fuse box node	vws.box1.12A	vws.box1.1	A	2	Fusebox
5	Splice	Sp011	Sp01		1	Splice
6	Splice	Sp021	Sp02		1	Splice
7	Device node	vws.box2.11	vws.box2.1		1	Component1
8	Device node	vws.box2.12	vws.box2.1		2	Component1
9	Device node	vws.box2.13	vws.box2.1		3	Component1
10	Device node	vws.box2.14	vws.box2.1		4	Component1
11	Device node	vws.box2.15	vws.box2.1		5	Component1
12	Device node	vws.box2.16	vws.box2.1		6	Component1
13	Device node	vws.box2.17	vws.box2.1		7	Component1
14	Device node	vws.box3.11	vws.box3.1		1	Component2
15	Device node	vws.box4.11	vws.box4.1		1	Component3
16	Device node	vws.box5.11	vws.box5.1		1	Component4
17	Device node	vws.box3.12	vws.box3.1		2	Component2
18	Device node	vws.box4.12	vws.box4.1		2	Component3
19	Device node	vws.box5.12	vws.box5.1		2	Component4
20	Splice	Sp031	Sp03		1	Splice
21	Splice	Sp041	Sp04		1	Splice

Node ID	Type	Node id_0	Location	Slot	Pin	Description
22	Splice	Sp051	Sp05		1	Splice
23	Ground node	vws.g11	vws.g1		1	Ground bolt
24	Ground node	vws.g21	vws.g2		1	Ground bolt

Table 7: Example table of the field NodeList

In the Table 7 is presented the field *NodeList* after the pre-processing of the electrical schematic of the Figure 12. As it was explained all the nodes of the system are listed and a unique ID is linked to each of them; in the Figure 12 it is possible to see that all the device's terminals, battery's terminal, fuse box' terminals, ground bolts and soldered splices are counted as nodes of the system (the nodes have been numerated from 1 to 24 in black color). The Table 7 shows information for all the nodes, such as their exact location and an assigned string ID.

Wire ID	Wire number	Harness	Node_id_1	Node_id_2	Part number	Gauge	Length
1	1	H1	1	3	Wire01	0.45	1.5
2	2	H1	4	5	Wire02	0.35	1.2
3	3	H1	5	7	Wire03	0.35	1.1
4	4	H1	10	14	Wire03	0.25	0.8
5	5	H1	17	20	Wire03	0.25	0.7
6	6	H1	20	21	Wire03	0.15	0.6
7	7	H1	21	22	Wire02	0.15	0.3
8	8	H1	22	23	Wire01	0.15	0.2
9	9	H1	24	2	Wire01	0.45	1.5
10	10	H1	13	22	Wire01	0.25	0.6
11	11	H1	5	6	Wire01	0.25	0.6
12	12	H1	6	8	Wire01	0.25	0.9
13	13	H1	11	15	Wire02	0.15	1.3
14	14	H1	18	20	Wire02	0.15	0.8
15	15	H1	6	9	Wire02	0.10	0.7
16	16	H1	12	16	Wire03	0.15	0.2
17	17	H1	19	21	Wire02	0.10	0.7

Table 8: Example table of the field WireList

In the Table 8 is presented the field *WireList* which is obtained after assigning to all the wires of the whole wiring system an unique numeric ID (column *Wire ID*, in the column *Wire number* is deployed the internal numeration of the wires inside of the harnesses, but in this example there is only one harness, hence the numeration of both columns is coincident). The Figure 12 shows in red color numbers the consecutive numeration of the wires of the example, and with the Table 8 by means of the unique ID, one can find related information such as the harness where it is contained, the two nodes where the wire is connected using the same numeration of the Table 7, the part number and physical magnitudes like the cross-section and the length. By means of the part numbers and with help of the *WireMaster* field, one can find the specific resistance and the specific weight in order to make power flow calculations.

Description	Part number	Pin_in	Slot_in	Pin_out	Slot_out
Component1	vws.box2.1-A	1		4	
Component1	vws.box2.1-A	2		5	
Component1	vws.box2.1-A	3		6	
Component1	vws.box2.1-A	1		7	
Component1	vws.box2.1-A	2		7	
Component1	vws.box2.1-A	3		7	
Component2	vws.box3.1-A	1		2	
Component3	vws.box4.1-A	1		2	
Component4	vws.box5.1-A	1		2	

Table 9: Example table of the field Def_InOut

The Table 9 presents the field with the pin in and pin out definition for the entire elements type device. In this example there are 4 components, all of them are shown in this table.

Connection ID	Location	Slot_1	Pin_1	Connection_id0	Connection type	Slot_2	Pin_2
1	vws.box1.1	A	1	F1	fuse	A	2

Table 10: Example table of the field Fuses

The field *Fuses* represented in the Table 10 lists the internal connections in the fuse boxes. In this example there is only one connection type fuse, therefore it is indicated in this table.

Device ID	Part number
1	vws.box2.1
2	vws.box3.1
3	vws.box4.1
4	vws.box5.1

Table 11: Example table of the field Devices

The Table 11 shows the list of elements type component of the example. All after the elaboration of the whole *dataContainer* structure, the next step is the formation of matrices with the all the possible logical paths between the terminals of the battery. Basically, two matrixes contain such information: the type matrix (Table 12) and the ID matrix (Table 13). In both matrixes each row is a possible logical path, and the followed methodology to represent them is the next: In the odd columns are located the nodes and in the even columns the connections between them. Therefore, both matrices will have always an odd integer number as column size.

In order to represent this methodology numerically, a nomenclature of the elements types that might be finding in the on-board network of a vehicle has been carried out. The nomenclature is:

- **1**: Element categorized as a fuse box node.
- **2**: Element categorized as wire connector
- **3**: Element categorized as soldered splice node
- **4**: Element categorized as a coupling node.
- **5**: Element categorized as a device node.
- **6**: Element categorized as a ground bolt node.
- **7**: Element categorized as a component (device) internal connector
- **8**: Element categorized as a coupling internal connector
- **9**: Element categorized as a fuse connector
- **10**: Element categorized as a negative terminal node
- **11**: Element categorized as a positive terminal node
- **12**: Element categorized as a plate connector
- **13**: Element categorized as a relay coil connector
- **14**: Element categorized as a relay switch connector
- **15**: Element categorized as vehicle’s body

To assign unique IDs to the elements belonging to each element type, some of them have been grouped together according to their features and also taking into account where they can be found. The adopted rules for assigning IDs are:

- The elements with types 1, 3, 4, 5, 6, 10 and 11 are organized together as nodes, and their ID is unique between them as it was showed in the Table 7.
- The elements with type 2 are forming a particular big group. The wires are assigned with a unique ID between as it was presented in the Table 8.

- The elements with type 7 are grouped also in a particular big group. The list is displayed in the Table 11. They have a unique ID between them.
- The elements with type 8 are grouped also in a particular big group. The list is displayed in the field *Couplings*. They have a unique ID between them.
- The elements with types 9, 12, 13 and 14 are organized together as internal connections of fuse boxes, and their ID is unique as it was showed in the Table 7. They have a unique ID between them.
- There is only one element type 15, since it is the vehicle's body for connecting the ground bolts.

Node type	Connector type	Node type	Connector type	Node type	Connector type	Node type	Connector type	Node type	Connector type	Node type	Connector type	Node type	Connector type	Node type	Connector type	Node type	Connector type	Node type	Connector type	Node type	Connector type	Node type	Connector type	Node type	Connector type	Node type	Connector type	Node type
11	2	1	9	1	2	3	2	5	7	5	2	5	7	5	2	3	2	3	2	3	2	6	15	6	2	10	-	-
11	2	1	9	1	2	3	2	5	7	5	2	3	2	6	15	6	2	10	-	-	-	-	-	-	-	-	-	-
11	2	1	9	1	2	3	2	3	2	5	7	5	2	5	7	5	2	3	2	3	2	3	2	6	15	6	2	10
11	2	1	9	1	2	3	2	3	2	5	7	5	2	3	2	6	15	6	2	10	-	-	-	-	-	-	-	-
11	2	1	9	1	2	3	2	3	2	5	7	5	2	5	7	5	2	3	2	3	2	6	15	6	2	10	-	-
11	2	1	9	1	2	3	2	3	2	5	7	5	2	3	2	6	15	6	2	10	-	-	-	-	-	-	-	-

Table 12: Example of an all paths type matrix

Node ID	Connector ID	Node ID	Connector ID	Node ID	Connector ID	Node ID	Connector ID	Node ID	Connector ID	Node ID	Connector ID	Node ID	Connector ID	Node ID	Connector ID	Node ID	Connector ID	Node ID	Connector ID	Node ID	Connector ID	Node ID	Connector ID	Node ID	Connector ID	Node ID	Connector ID	Node ID
1	1	3	1	4	2	5	3	7	1	10	4	14	2	17	5	20	6	21	7	22	8	23	1	24	9	2	-	-
1	1	3	1	4	2	5	3	7	1	13	10	22	8	23	1	24	9	2	-	-	-	-	-	-	-	-	-	-
1	1	3	1	4	2	5	11	6	12	8	1	11	13	15	3	18	14	20	6	21	7	22	8	23	1	24	9	2
1	1	3	1	4	2	5	11	6	12	8	1	13	10	22	8	23	1	24	9	2	-	-	-	-	-	-	-	-
1	1	3	1	4	2	5	11	6	15	9	1	12	16	16	4	19	17	21	7	22	8	23	1	24	9	2	-	-
1	1	3	1	4	2	5	11	6	15	9	1	13	10	22	8	23	1	24	9	2	-	-	-	-	-	-	-	-

Table 13: Example of an all paths ID matrix

Finally, the Table 12 as discussed before displays per row all the possible logical paths between the battery's terminals by indicating the type of each element. Thus, in the first row one can find as first element the type 11 (battery's positive terminal), the second element is type 2 (wire), the third fuse box node (type 1) and so on, the last element is type 10 (negative's negative terminal). All the paths (rows) start with the same type and finish with the same, as it is logic. Additionally, those paths with length smaller than the biggest one will have as elements *NaN* (Not a Number) values after their last occupied position. A very important characteristic of the paths detector code is that it ignores the ways through which the communication terminals of the components are connected, because the wires used for those objectives handle very less power and their size cannot be optimized more: basically they are neglected due to their low affectation in the electrical power circuits.

The Table 13 presents the same information of the Table 12 but in each position instead of showing the type of the element its ID is placed. Thus, in the first row the first element with type 11 (position (1, 1) of the type matrix) has an ID of 1 (position (1, 1) of the ID matrix), which can be found in the Table 7 with deeper information; also the ID of the wire located in the position (1,2) of the type matrix can be determined in the position (1,2) of the ID matrix: this is 1, and its complete information can be found in the Table 8. The same logic is applied for the other elements, bearing in mind the rules for ID's assignation already explained in order to point out correctly to find the features of a specific element with a determined type and ID.

Both matrixes of the Table 12 and Table 13 have 6 possible paths which is correct if one analyzes the Figure 12. In this arrangement there is an Electronic Control Unit (ECU) supplying 3 loads; the ECU has its ground connection point for addressing the current from the internal BJTs emitters and its 3 collectors are feeding the loads (obviously this is a simplified scheme) from the power coming through their 3 bases. So, each input pin of the ECU is electrically coupled to its direct output and to the ground connection point: this makes the 6 possible paths. The paths detector code is able also to discriminate the compatibility of paths according to the concept of modularity, i.e., a path cannot appear if it is formed by wires belonging to different modules of functional families restricted to only one.

6.2. Processing of the data - Standardization of the inputs for the power flow visualization/calculation tool

After the pre-processing stage of the factory data, it is required to perform a final processing before sending the inputs to the external power flow visualization/calculation tool, since as it was discussed in previous Chapters, the factory data of the cars' manufacturing world is not properly organized from a point of view of electrical solvers which require some standardization of the inputs to be able to run rapidly, dynamically, robustly and systematically simulations for getting the results of electrical magnitudes throughout the on-board network distribution system of a vehicle, such voltages at the nodes, differential voltages at the equipment's terminals, branches current, power transfer, efficiency, etc.

With the aim of doing so, in here is described the input data format to be use, based in the so-used standard of the IEEE [14] for this kind of applications, which is as well similar to those implemented by commercial software in general, although with the necessary modifications to adapt the conditions and specific features of the cars' wiring system manufacturing world into this scheme. The next structures are put forward:

Connection matrix:

This numeric matrix is projected to deal with the connection between nodes in the system; therefore a connection will never have an imposed consumption of current. The nominal currents of connection elements will not be treated as injections to the system, but as limit restrictions values of the current that could be transported through them, i.e., after the power flow solver calculates the real current through all the branches, these values might be compared to the nominal ones of these elements and thus to produce alarms when the real current are higher than the mentioned limit values. This applies for all the elements categorized as connections, which are: wires (code 2), ECUs (code 7), couplings (code 8), fuses (code 9), plates (code 12), relays coils (code 13), relay switches (code 14) and the car's body (code 15). Each row of this connection matrix represents a possible link between nodes in the system and the information of the columns is explained the following. It is worth it to remark that this matrix is intended to storage only numeric data, master tables with string type information will be also provided in order to give information such as part numbers and others. The columns are:

- General ID: Unique numeric ID of the connections. Each row is a different connection and therefore an individual ID is assigned here.
- Source: Node with highest voltage potential in the connection. In this column are deployed the nodes using the unique ID assigned in the *NodeList* field. With the objective of identifying the data of a given node a master table is provided as well. A given connection can only be connected to one source node.
- Destination: Node with lowest voltage potential in the connection. In this column are deployed the nodes using the unique ID assigned in the *NodeList* field. With the objective of identifying the data of a given node a master table is provided as well. A given connection can only be connected to one destination node.
- Type: In this column is specified the type of the connection using the numeric nomenclature already explained before: wires (code 2), ECUs (code 7) and so on.
- ID: In this column is specified the ID of the connection according to the rules of grouping elements type already explained. Master tables with information in function of the type and individual IDs per group are provided.
- Module: Module where the connection is associated. It is necessary to take into consideration that the same connection might appear in several modules; as it was explained in the Chapter IV, Table 2, for different modules of the same functional family some functions could be shared, hence some connections and loads can be present in several modules. In this column are deployed the modules but with a numeric identifier,

therefore a master list of modules and families is also given in order to associate the number with the string names. If a special connection does not have associated any modules because is always present, in this column is presented a zero (such as the vehicle's body).

- Harness: In this column is presented a numeric identifier of the harness where a connection belongs. Since harnesses only carry wires, this column is only logical for connection type wire, otherwise a zero is showed. In order to associate the numeric identifier of a harness with its string name, a master table of harnesses is also given.
- Specific resistance: Resistance in Ohm/meter of the connection. In case the connection is not type wire in this space a zero is deployed, otherwise the parameter is displayed according to the manufacturing information and the surrounding maximum temperature of the wire. The specific resistance available in tables are measured under nominal test conditions ($\boldsymbol{R}(\boldsymbol{T_0})$) such as a stable temperature of 20 ℃ to 25 ℃ ($\boldsymbol{T_{Nom}}$) [15]; however it has been proved the linear temperate-dependence of the electrical resistance, therefore by means of the temperature coefficient of the material (α, for cooper researches have determined a value of $3.9 \times 10^{-3}\ \frac{\Omega}{K}$), the determined maximum temperature $\boldsymbol{T_{real}}$ (the list of temperatures of each segment of a harness are expressed in the field *SegmentsJunctions*) and the Equation 1 , the correction of the resistance is performed.

$$\boldsymbol{R}(\boldsymbol{T}) = \boldsymbol{R}(\boldsymbol{T_0})(\boldsymbol{1} + \boldsymbol{\alpha}(\boldsymbol{T_{real}} - \boldsymbol{T_{Nom}}))$$

Equation 1

In a post-processing stage to be carried out by the external solver, typical values of resistance can be assigned to each type of connection different than wires, and also when a resistance is lower than a minimum value the nodes can be joined in order to avoid convergence issues when calculating the power flow.

- Maximum specific resistance: In this column the maximum resistance of the wire under the most critical thermal operation condition is presented. As it was discussed in the Section 4.6 a wire is expected to operate at maximum 105℃ under nominal conditions, hence and since it could be very difficult to make a more accurate model for determining the real surrounding temperature taking into account the currents on the wires, this value could be important if it is desired to consider the wire working in the worst case scenario.
- Length: In this column the length of the wires is represented. In case of connections different than wires in here is showed zero.
- Maximum current: Maximum current resistible by the connection. This value is used for generating warnings when the current through its connection is higher to this value. The value of maximum current in general is available systematically for components type ECU; in the Figure 13 is deployed an example of how the relations Pin-Pout of ECUs are treated: all the connections between pins are considered individual connections, therefore each one has a specific General ID (indicated in red colors number), and also the maximum current is determined by the current available in datasheets of the output pin, in this way if a output pin is connected to more than one input pin, the maximum current per connection is gotten by a linear split of such current. For connections of other types (wires, etc) this parameter is not so determinant because as a general rule the wires are oversized with respect to the current they are expected to carry, this is due to they are sized from the a thermal point of view under short circuits as explained in the Section 4.6. According to the previous for those types in this column is showed zero.
- Minimum current: Nominal current resistible by the connection. This parameter is not in general available systematically; therefore in a first version of the power flow calculator tool is not taken into consideration. Hence a value of zero is always presented for all the connections. This will not affect the results because this parameter is merely informative.
- Nominal current: Apply the explained in the previous point.

- State: In this column is presented always a value of 1, meaning that the connection is activated. In the external power flow calculation/visualization tool this value can be change dynamically to zero when is desired to open a connection, simulating thus the function of interrupters.

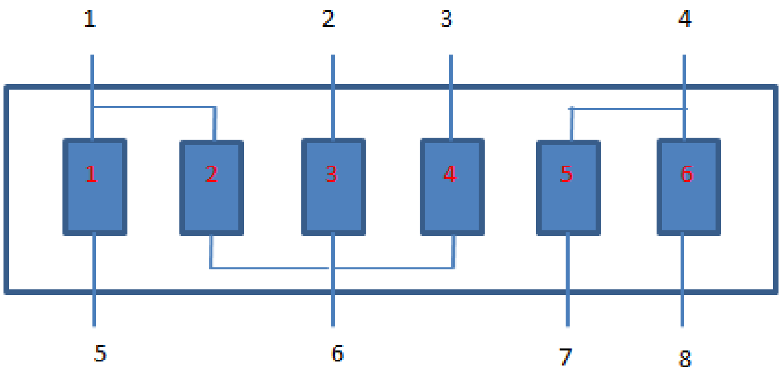


Figure 13: Exemplification of the connections of an ECU

Load matrix:

This numeric matrix is projected to deal with the bi-nodal loads of the system, i.e., elements with an associated electrical consumption by injection of nodal current. In general in this matrix are represented elements type component with behavior of consumers: coils and components (treated as a *black box*, i.e., an element with consume given by its pin without modeling the behavior inside of it). Coils are considered injectors only if they do not have a controller which in turn would be classified as a consumer type component. If there are loads connected between two nodes of the same layer, they will be treated as constant impedance loads, otherwise as constant current sources; this is going to be explained ahead. Each row of this load matrix represents a possible bi-nodal load in the system and the information of the columns is explained the following:

- General ID: Unique numeric ID of the bi nodal loads. Each row is a different load and therefore an individual ID is assigned here.

- Positive node: Node with highest voltage potential in the load. In this column are deployed the nodes using the unique ID assigned in the *NodeList* field.
- Negative node: Node with lowest voltage potential in the load. In this column are deployed the nodes using the unique ID assigned in the *NodeList* field.
- Type: In this column is specified the type of the load using the numeric nomenclature already explained before: component (code 7) and coil (13).
- ID: In this column is specified the ID of the bi-nodal load according to the rules of grouping elements type already explained. Master tables with information in function of the type and individual IDs per group are provided.
- SubID: The SubID is meant to be a numeric identifier of the different part numbers associated to a component with a specific location. Thus, if a component has 3 different part numbers, the SubID goes from 1 until 3. With a master table is possible to link the type, ID and subID of a load with its string identifiers, such as its part number.
- Module: Module where the bi-nodal load is associated. It is necessary to take into consideration that the same bi-nodal load might appear in several modules; as it was explained in the Chapter IV, Table 2, for different modules of the same functional family some functions could be shared, hence some connections and loads can be present in several modules. The part number (or SubID) can vary or not from one module to other for a given device. In this column are deployed the modules but with a numeric identifier, therefore a master list of modules and families is also given in order to associate the number with the string names.
- Nominal current: Current *i1* of the bi-nodal load (See Section 6.1 for the meaning of this current).
- Maximum current: Current *i2* of the bi-nodal load (See Section 6.1 for the meaning of this current).

- Maximum peak transient current: Current *i3* of the bi-nodal load (See Section 6.1 for the meaning of this current).
- Minimum current: Current *i4* of the bi-nodal load (See Section 6.1 for the meaning of this current).
- Nominal Voltage: In this column is indicated the nominal voltage of the bi-nodal loads. Since the distribution system of a vehicle is a direct current electrical network, basically there can be two types of loads: Those connected between two nodes with nominal voltage of 14 V (given by the higher potential terminal of the provider of the system: the battery) and those connected between a node with nominal voltage of 14 V and a node with nominal voltage of zero (given by the lower potential terminal of the battery). In the Figure 14 is schematized an example of the previous situation: There are two elements type component with behavior of consumers connected in series; the *Consumer 1* has three pins which can be modeled as two bi-nodal loads, *A* and *B*, and the *Consumer2* has two pins which are modeled as a single bi-nodal load *C*.

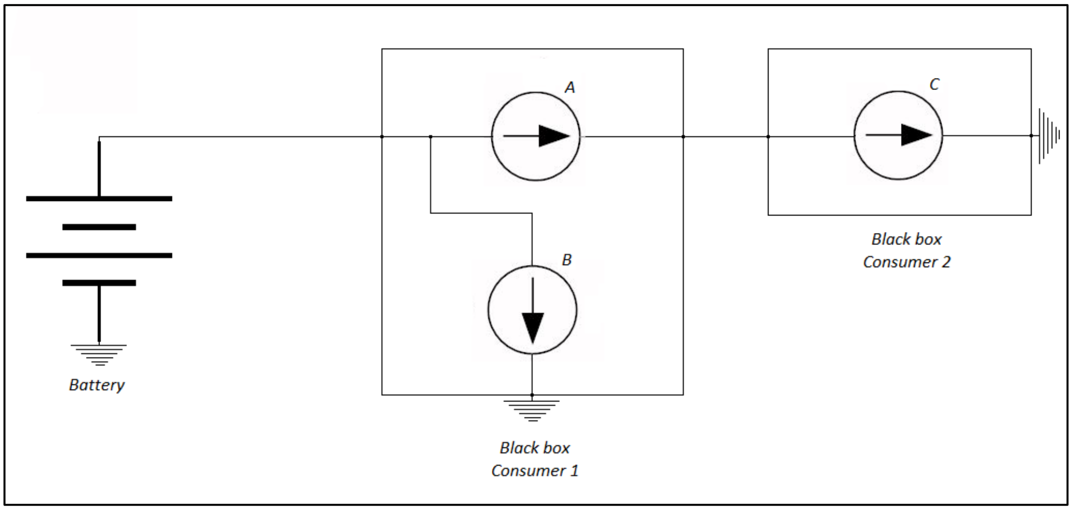


Figure 14: Exemplification of bi-nodal loads

It is simple to conclude that the bi-nodal loads *B* and *C* must have associated the nominal voltage of the battery, and the load *A* a voltage way less than the battery, because as it was discussed before, the nominal voltage is dependent of the nominal characteristics of the nodes where the loads are connected: *A* is connected between a input pin and a output pin of the *Consumer 1*, in contrast with *B* and *C* which are connected to a ground node. To establish precisely with complete accuracy the nominal voltage of a bi-nodal load such as *A* might be complicated due to the scarcity in most of the cases of precise information from the components used in the Group, however it could be possible to consider in this kind of situations nominal voltages between 0.5 V and 0.7 V according to typical voltages drops of semiconductors, because at the end, components like the *1* of this example are internally electronics arrangements of discrete electronic elements such as diodes, transistors, etc.

- Level of load: In this column is presented always a value of 1, meaning that the nominal current of the bi-nodal is selected. In the external power flow calculation/visualization tool this value can be change dynamically between 0, 1 and 2 in order to switch the level of the load between the minimum, nominal and maximum, respectively.
- Type of load: The static models of loads are widely used to represent precisely the behavior of static loads such as resistances, illuminations and so on, and also are important to make an approximation of the behavior of dynamic loads such as motors or generators. Since in the industry of cars all the loads can be categorized as statics, this approach is far enough to model them. In this approach of static loads exists the so-known polynomial approximation technique of loads [16]. In this technique the type of a load is limited to three variants: constant current, constant power and constant impedance, and the real power of a load is given by the Equation 2, where P_0 is the nominal power, V_0 is the nominal voltage, V is the real voltage, P_R the real power and *N* a coefficient that may change between 0, 1 and 2 to consider constant power, constant current and constant impedance load, respectively. In fact, in this column is presented one

of the above numbers to indicate the nature of a particular load. According to the example of the Figure 14 the bi-nodal loads *A* and *C* cannot be considered both constant current, because otherwise a solution of the power flow will not be possible to perform due to basically both are in series; therefore in this kind of situations the load *A* is treated as constant impedance load and the *B* and *C* as constant current.

$$\boldsymbol{P_R = P_0 \times \left(\frac{V}{V_0}\right)^N}$$

Equation 2

- State: In this column is presented always a value of 1, meaning that the load is activated. In the external power flow calculation/visualization tool this value can be change dynamically to zero when is desired to disable a load, simulating thus a contingency in the system.

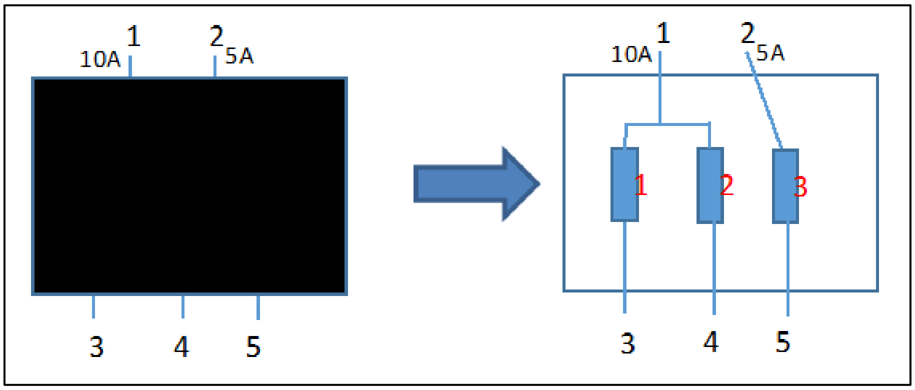


Figure 15: Modeling of components *black box* with consumer behavior

For loads elements type component and behavior of consumers, it was already explained the nature of them to be treated as black boxes, due to the scarcity of information and even, not need of a more complete and complex modeling from the point of view of the designers in charge of the sizing of fuses and cables in vehicles. For that reason, in the Figure 15 is deployed the simple modeling strategy adopted to distribute the consumptions of a component with a given number of pins and pin-pout description; as it was explained at the beginning of the matrix load Section, the loads in here are conceived as bi-nodal loads, hence the currents of a pin are split proportionally to the number of connected pins to it in order to assign the current of consumption to each single load. This is realizable in the right image of the Figure 15, the pin 1 is connected to the pins 3 and 4, therefore two single bi-nodal loads appear each one with a current consumption equal to the half of the input pin current, on other hand the pin 2 is only electrically coupled to the pin 5, so there is only one load. The proportional split is adopted since to establish other distribution of currents is required to abide the Kirchhoff current law inside of the device and this is in most of the cases not possible to ensure.

Car Setup:

This structure will be defined by the external simulator tool. In this structure will be established all the functional families and modules associated to them, each row would represent a module available in the variety of a specific car model. Finally there will be a column to select which modules to select and thus, the respective connections and loads will be activated.

Car Scenario:

This structure will be defined by the external simulator tool. In this structure will be defined the operative scenario for a simulation, such as the activation or not of wires and loads for a given setup.

Master Tables:

- Master nodes table: In this table is deployed the *NodeList*, i.e, all the nodes of the system with their related information about location, slot, pin and string identifier.
- Master wires table: In this table is shown all the list of wires of the system and their part number.

- Master modules table: In this table is presented the list of modules and the families where each one them belong.
- Master harness table: In this table is deployed the list of harnesses and their drawing name.
- Master fuses table: In this table is shown all the internal connection of the fuse box and their related information such as input slot, input pin, string IDs, output slot and output pin.
- Master coupling table: In this table is presented the list of couplings with ID and VOBES locations of both extremes.
- Master coupling modularity table: In this table is available information about the VOBES locations of couplings, their modularity and part numbers.
- Master ECU table: In this table is shown all the components with behavior of ECU with their unique ID.
- Master ECU modularity table: In this table is presented the VOBES locations of the ECUs, their modularity and part numbers.
- Master connections table output: Basically in this table is shown the same columns of the matrix connections but to each side of each column of that matrix a string column is deployed to represent the string identifier of the associated numeration.
- Master loads table output: Basically in this table is shown the same columns of the matrix loads but to each side of each column of that matrix a string column is deployed to represent the string identifier of the associated numeration.

6.3. Pseudo codes - Automatization of the process for getting the inputs for the power flow visualization/calculation tool

In the Section 6.1 the pre-processing stage of the data coming from the Excel files (BOM lists, wire lists and QT) was explained, the information available in MATLAB was described in detail.

In the Section 6.2 the proposed organization of the input data for the external power flow visualization/calculation tool was presented as well; in this Section the aim is to show how to get those inputs automatically according to the available information. The adopted procedure to carry out such automatization is by means of programmed routines through MATLAB.

In both structures connection matrix and load matrix, one of the key factors is the need to be able to classify the elements type component as consumer or an Electronic Control Unit (ECU) according to its behavior. Additionally, it was explained that those components have to be treated like *black boxes* due to the lack of information and mainly, the lack of need to deepen into more complete analysis under the point of view of wires and fuse sizing which is the biggest aim of this power flow calculation tool, at least in this first version. However, it is important still to differentiate when a *black box* may be considered as a distribution unit (ECU) or as an element which demands electrical power for its operation (load, sensors, interrupters, etc.). The Figure 16 shows an example of schematization for understanding a particular situation where it is present a control unit providing power to loads.

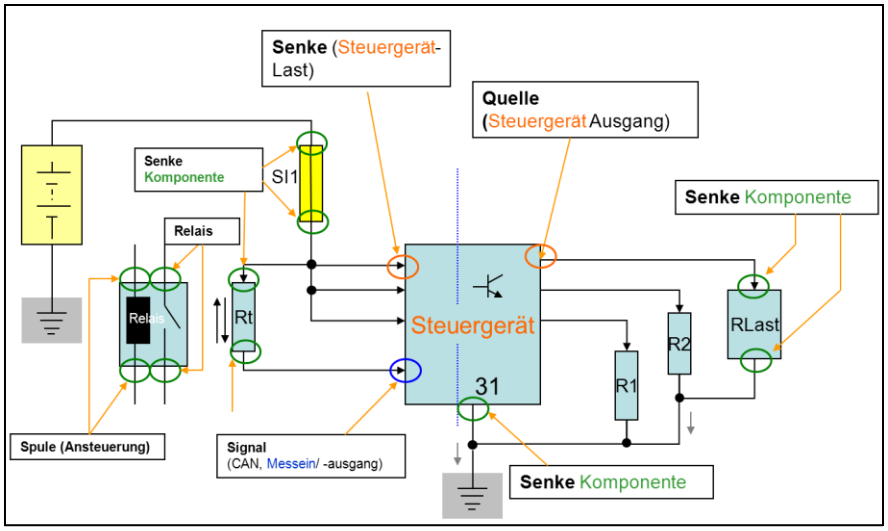


Figure 16: Pin types of components [13]

When previously the word ‘behavior’ was used, it is more referred to a point of view of an overall analysis of the pin types of components (commented in the Section 5.2, worksheets of the QT *Device pinout* and *pin master*) rather than a formal study using modeling techniques and experimentations, due to, as discussed before, it is not feasible. The Figure 16 shows the example of components’ arrangement tagging the pins with their corresponding reserved words: all the components of the Volkswagen Group (which are grouped in the database *Sys42*) have two or more of the next pin types:

- *Senke (SG Last): SG* stands for *Steuergerät* in german, in english Electronic Control Unit (ECU). *Last* on other hand, translates load. Pins labeled under this name are basically power input pins in components. See Figure 16.
- *Quelle (SG Ausgang): Ausgang* is the german for output. This is the pin type name used for power distribution pins in components (ECUs). See Figure 16.
- *Senke (Komponent): Senke* is the german for Sink, and the pins tagged under this name are basically those which do not exhibit characteristic of *Last* nor *Ausgang* pins. Pins *Senke (Komponent)* might represent a ground connection of a component. See Figure 16.
- *Signal (CAN, Mess.):* Description associated to a pin designated for handling communication signals between components. As it was discussed before, the power flow calculator neglects this part of the network because of its low levels of voltages and power. See Figure 16.
- *Schaltkontak (Relais/Schalter):* Description associated to a pin belonging to a contactor switch of an electromechanical relay. See Figure 16.
- *Spulle (Ansteuerung):* Description associated to a pin belonging to a coil of an electromechanical relay. See Figure 16.
- *Nicht verbunden:* German for not connected.

From this list of pin types one should be able to make a classification of a component, determining if it fits as a consumer of as an ECU. An important aspect to remark is in this tool for calculating power flow, when a component is classified as ECU it is considered simply like a point of division of currents (See the Figure 13, kind of substation in HV electrical power systems) and real behaviors inside of it are overridden (an ECU may increase or decrease the level of voltage to supply a load, make the conditioning of current, or even to provide PWM signals; in other terms, it adapts the electrical magnitudes according to the need of the consumer and the conditions imposed by the user). In order to make more accurate analysis of such components, PSPICES tools to simulate them might be explore, taking into account specific features of a particular device. Nevertheless, from the point view of a static analysis, considering always DC signals is a good approximation for dimensioning wires and fuses, always bearing in mind that nominal operation conditions are assumed.

From the explained all above the Algorithm 1 has been coded in MATLAB in order to get the function *Pin Type*, useful to translate the reserved pin type names used for the components into their electrical meaning. For instance, in the Algorithm 1 if one enters the input *String=Senke (SG Last)* and *Type=Component_power_in* the output is *Cert=1* meaning that the pin with such name has to be considered as a power input pin, in the case that *Type=Component_power_out* the result would be in contrast, *Cert=0*. This is extensible for the other pin types:

Signal (CAN, Mess.) → *Signal*
Senke (SG Last) → *Component_power_in*
Quelle (SG Ausgang) → *Component_power_out*
Senke (Komponent) → *Sink*
Schaltkontak (Relais/Schalter) → *switch_power*
Spulle (Ansteuerung) → *switch_coil*
Nicht verbunden → *not_connected*

In general the function of the Algorithm 1 answers to the question: Is the pin with type name *XXX* an electrical type *YYY*?

Algorithm 1: Pin Type

```
Input:  Type, String
Output: Cert
1. If String is cell go to 2. If String  is char go to 5.
2. for i=1:1:length(String) do
3.   Cert(i)=PinType (Type,String{i})
4. end for
5. Switch (Type)
6.   Case 'Signal'
7.        Comp{1}='Signal'
8.   Case 'Component_power_in'
9.        Comp{1}='SG'
10.       Comp{2}='Last'
11.  Case 'Component_power_out'
12.       Comp{1}='SG'
13.       Comp{2}='Ausgang'
14.  Case 'Sink'
15.       Comp{1}='Komponent'
16.  Case 'switch_power'
17.       Comp{1}='Schaltkontakt'
18.  Case 'switch_coil'
19.       Comp{1}='Ansteuerung'
20.  Case 'not_connected'
21.       Comp{1}='Verbunden'
22. end Switch
23. for i=1:1:length(Comp) do
24.   Cert=Cert&&~isempty(strfind(String,Comp{i}))
25. If ~ Cert go to 26
26. end for. End
```

Algorithm 2: Device Type

```
Input:  dataContainer, Location
Output: device_type
1. PN=locateInTable(dataContainer.DevicesModularity,'partnumber','and','location', Location)
2. for i=1:1:length(PN) do
3.   pins=locateInTable(dataContainer.DevicesModularity,'pin_type','and','partnumber',PN{i})
4.   Signal=sum(PinType('Signal',pins))
5.   P_in=sum(PinType('Component_power_in',pins))
6.   P_out=sum(PinType('Component_power_out',pins))
7.   Sink= sum(PinType('Sink',pins))
8.   Switch_p= sum(PinType('switch_power',pins))
9.   Switch_coil= sum(PinType('switch_coil',pins))
10.  NC= sum(PinType('not_connected',pins))
11.  If  P_in>0 go to 12. Otherwise to 17
12.  If P_out>0 go to 13. Otherwise to 14
13.  device_type{i}='Source'. Go to 23
```

```
14. If Sink>0 go to 15. Otherwise to 16
15. device_type{i}='Consumer'. Go to 23
16. device_type{i}='Unclear' . Go to 23
17. If P_out==0 go to 18. Otherwise to 22
18. If Sink>0 go to 19. Otherwise to 20
19. device_type{i}='Consumer'. Go to 23
20. If Signal>0 go to 21
21. device_type{i}='low_power'. Go to 23
22. device_type{i}='Unclear' . Go to 23
23. end for
24. If length(device_type)>1 go to 25. Otherwise 33
25. device_type=unique(device_type). Go to 26
26. Unclear=sum(SScompare(device_type,'unclear'))
27. alternatives=device_type(~SScompare(device_type,'unclear')))
28. If Unclear==length(device_type) go to 29. Otherwise to 30
29. device_types='Unclear' End
30. If 1==length(alternatives) go to 31. Otherwise to 32
31. device_type= alternatives End
32. Error('Oxymoron') End
33. device_type End
```

In the Algorithm 2 is presented a function programmed in order to classify a component (device) as an ECU or as a consumer. This code carries out a complete analysis from the point of view of a device location including all its associated part numbers. A component is considered an ECU (also called 'Source') if and only if it has at least one *Component_power_in* terminal and one *Component_power_out* terminal. If a component has none pin *Component_power_in* but at least one terminal *Component_power_out*, it is not possible to conclude what type it should be, hence it is considered as 'Unclear'. On other hand, a component is allocated as Consumer if it has only pins *Sink* or with *Component_power_in* terminals but without *Component_power_out* pins. If a component has associated *Component_power_in* pins but none *Component_power_out* terminals nor *Sink*, is not possible to conclude anything about it, therefore the device would be named as 'Unclear'. In the Algorithm 2 an overall study of all the part numbers associated to a device location is done, therefore if different conclusions are obtained for different part numbers an error warning is executed (line 32), and also if for one or even for several part numbers 'Unclear' results are gotten but for the rest a congruent result is produced, the final classification is done according to the logical result (line 27 and line 30). Components only with *Signal* pins are neglected. Finally, if under all points of view an ‘Unclear’ result is obtained the device with that location is considered ‘Unclear’ as well; theoretically this condition should not be reached, nevertheless errors in the data coming from the database may appear.

Algorithm 3: Determining Master Tables

Input: dataContainer
Output: Master_modules_table, Master_harness_table, Master_nodes_table, Master_wires_table, Master_fuses_table, Master_couplings_table, Master_couplings_modularity_table, Master_ECU_table, Master_ECU_modularity_table, Master_potential_loads_table

```
modules=unique(dataContainer.Modules{:,2})
for i=1:1:length(modules) do
  numeration(i)=i
  family(i)=locataInTable(dataContainer.Modules,’family’,’and’,’module’,modules(i))
end for
Master_modules_table=FormTable(numeration,modules,family)
harness=unique(dataContainer.WiresSegments{:,3})
for i=1:1:length(harness) do
  numeration(i)=i;
end for
Master_harness_table= FormTable(numeration,harness)
Master_nodes_table=dataContainer.NodeList
Master_nodes_table(:,7)=[]
Master_wires_table=dataContainer.WireList
Master_wires_table(:,9:7)=[]
Master_wires_table(:,5:2)=[]
Master_couplings_table=dataContainer.Couplings
for i=1:1:length_rows(Master_couplings_table) do
  location_1=Master_couplings_table{i,2}
  location_2=Master_couplings_table{i,3}
  modules_location_1=unique(locateInTable(dataContainerDevicesModularity,’module’,’and’,’location’,location_1))
  modules_location_2=unique(locateInTable(dataContainerDevicesModularity,’module’,’and’,’location’,location_2))
  for j=1:1:length(modules_location_1) do
    PN_loc1=unique(locateInTable(dataContainerDevicesModularity,’partnumber’,’and’,’location’, location_1,’module’, modules_location_1{j}))
    PN_loc2=unique(locateInTable(dataContainerDevicesModularity,’partnumber’,’and’,’location’, location_2,’module’, modules_location_2{j}))
  end for
  Master_coupling_modularity_table=FormTable(repeat(location_1), modules_location_1, PN_loc1, repeat(location_2), modules_location_2, PN_loc2)
end for
Master_fuses_table=dataContainer.Fuses
Master_ECU_table=dataContainer.Devices
for i= length_rows(Master_couplings_table):-1:1 do
  device_location=Master_ECU_table{i,2}
```

```
33.   device_type= DeviceType(dataContainer, device_location)
34.   If strcmp(device_type,’Source’) ~=1 go to 35. Otherwise to 36
35.   Master_ECU_table(i,:)=[]
36. end for
37. for i=1:1:length_rows(Master_ECU_table) do
38.   ECU_location=Master_ECU_table{i,2}
39.   Modules_ECU=unique(locateInTable(dataContainerDevicesModularity,’module’,’and’,’location’, ECU_location))
40.     for j= 1:1:length(Modules_ECU) do
41.       PN_ECU=unique(locateInTable(dataContainerDevicesModularity,’partnumber’,’and’,’location’, ECU_location,’module’, Modules_ECU {j}))
42.     end for
43.   Master_ECU_modularity_table=FormTable(repeat(ECU_location), Modules_ECU,PN_ECU)
44. end for
45. for i=1:1:length_rows(dataContainer.Devices) do
46.   device_location=dataContainer.Devices{i,2}
47.   device_ID=dataContainer.Devices{i,1}
48.   device_type= DeviceType(dataContainer, device_location)
49.   If strcmp(device_type,’Consumer’) 1 go to 50. Otherwise to 57
50.   modules_load=unique(locateInTable(dataContainerDevicesModularity,’module’,’and’,’location’, device_location))
51.     for j= 1:1:length(modules_load) do
52.       PN_load=unique(locateInTable(dataContainerDevicesModularity,’partnumber’,’and’,’location’, device_location,’module’, modules_load {j}))
53.       PN_load_num=(1:1:length(unique(PN_load)))
54.       load_module_number= locateInTable(Master_modules_table,’Numeration’,’and’.’Modules’, modules_load {j})
55.     end for
56.   Master_potential_loads_table=FormTable(repeat(7), repeat(‘component’),repeat(device_ID),repeat(device_location), PN_load, PN_load_num,modules, load_module_number)
57. end for
58. IDcoils=locateIntable(dataContainer.Fuses,’connection_id’,’and’,’connection_type’,’relay_coil’)
59. for i=1:1:length (IDcoils) do
60.   location_coil=unique(locateInTable(dataContainer.FusesModularity,’location’,’and’,’connection_id’,IDcoils(i)))
61.   modules_coil=unique(locateInTable(dataContainer.FusesModularity,’modules’, ’location’, location_coil,’connection_id’, IDcoils(i)))
62.     for j= 1:1:length(modules_coil) do
63.       PN_coil=locateInTable(dataContainer.FusesModularity,’partnumber’,’and’,’connection_id’,’ IDcoils(i)’,’modules’, modules_coil{j})
64.       PN_load_num=(1:1:length(unique(PN_coil)))
65.       load_module_number= locateInTable(Master_modules_table,’Numeration’,’and’.’Modules’, modules_coil{j})
66.     end for
67.   Master_potential_loads_table=FormTable(repeat(13), repeat( ‘coil’),repeat(IDcoils),repeat(location_coil), PN_coil, PN_load_num,modules_coil, load_module_number)
68. end for. End
```

The pseudo code displayed in the Algorithm 3 belongs to the function *Determining Master Tables* which allows getting the master tables already described in the Section 6.2. This function requires the structure *dataContainer* exclusively, and by almost straightforwardly assignation the modules table, harnesses table, nodes table, wires table, fuses table and couplings table are obtained, i.e., they are no more than replications of some fields of the aforementioned structure; in the case of the tables of the modules and harnesses, the code assigns to each of them a particular numeric ID as it was explained for the connection matrix and load matrix. The master couplings modularity table is also confectioned by means of iterative processes (lines 18 to 28) in order to associate the locations of couplings with their corresponding modules and part numbers. Likewise, to get the master ECU table and the master ECU modularity table the function of the Algorithm 2 *DeviceType* (line 34) is called with aims of filtering out those components which are not considered ECUs (or ‘Sources’), and afterwards, relating each location with its corresponding modules and part numbers. On other hand, an important table is also processed, the master potential loads table; this is not presented to the external solver as the previous ones, but it is important for the function to get the load matrix (to be explained ahead). To get this table the function *DeviceType* is again used (line 48) and the objective is to list all the locations of components which are classified as consumers and the coils as well; thus for each location the modules and part numbers associated are displayed, also their numeric IDs (7 for components and 13 for coils) and SubIDs (related to the part number).

From the Appendix 6 to the Appendix 16 an example with all the previously explained master tables are available to see.

Algorithm 4: Forming Connections

```
Input:  dataContainer, Master_modules_table, Master_harnesses_table
Output: connection_matrix, Master_connections_table
1. nodep=1
2. for row=1:1:length_rows(dataContainer.all_series_sets.types) do
3.   actual_node=1
4.      for column=1:2:( length_columns(dataContainer.all_series_sets.types) -2) do
```

```
5.   If  nodep==1 go to 6. Otherwise to 7
6.   vnp(nodep)=dataContainer.all_series_sets_ids(row,column). nodep++
7.   If  nodep>1 && ~isnan(dataContainer.all_series_sets,ids(row,column+2))  go to 8.
     Otherwise to 46
8.   pos=find(vnp== dataContainer.all_series_sets,ids(row,column+2)). check=0. check2=0
9.   If ~isempty(pos) go to 10. Otherwise to 24
10.  check=1
11.  If dataContainer.all_series_sets,types(row,column+1))==2  go to 12. Otherwise to 23
12.  cont_ver1=0. cont_ver2=0
13.    for l=1:1:length_rows(connection_matrix) do
14.        If  connection_matrix(l,1) == vnp(actual_node) &&  connection_matrix(l,2) ==
           vnp(pos) go to 15. Otherwise to 18
15.        check2=1. cont_ver1++
16.        If connection_matrix(l,3) == dataContainer.all_series_sets,types(row,column+1) &&
           connection_matrix(l,4) ~= dataContainer.all_series_sets,ids(row,column+1) go to 17.
           Otherwise to 18
17.        cont_ver2++
18.    end for
19.  If cont_ver1== cont_ver2 && cont_ver1~=0 && cont_ver2~=0 go to 20. Otherwise to 21
20.  Add information to the connection_matrix. Add information to the
     Master_connections_table
21.  If check2==0 go to 22. Otherwise to 23
22.  Add information to the connection_matrix. Add information to the
     Master_connections_table
23.  actual_node=pos
24.  If check==0 go to 25. Otherwise to 46
25.  vnp(nodep)=dataContainer.all_series_sets_ids(row,column+2)
26.  Switch (dataContainer.all_series_sets_types(row,column+1)
27.    Case 2
28.      Add information to the connection_matrix. Add information to the
         Master_connections_table
29.      If dataContainer.all_series_sets.types(row,column+3)==15 go to 30. Otherwise to 44
30.      Add information to the connection_matrix. Add information to the
         Master_connections_table
31.    Case 7
32.      If strcmp(device_type,'Source') 1 go to 33. Otherwise to 44
33.      Add information to the connection_matrix. Add information to the
         Master_connections_table
34.    Case 8
35.      Add information to the connection_matrix. Add information to the
         Master_connections_table
36.    Case 9
37.      Add information to the connection_matrix. Add information to the
         Master_connections_table
38.    Case 12
39.      Add information to the connection_matrix. Add information to the
         Master_connections_table
40.    Case 13
41.      Add information to the connection_matrix. Add information to the
         Master_connections_table
42.    Case 14
```

```
            Add information to the connection_matrix. Add information to the
            Master_connections_table
        end Switch
        actual_node=nodep. nodep++
    end for
end for.
Organize connection_matrix. Organize Master_connections_table. End
```

The pseudo code of the Algorithm 4 allows understanding the function *Forming Connections,* which has been programmed in order to get both the connection matrix and the master connections table already explained in the Section 6.2. The Table 12 and Table 13 present the matrices contained in the field *all series sets* of the *dataContainer* main structure. The idea of the Algorithm 4 is to convert both matrices into the aforementioned inputs for the external solver tool.

The followed strategy is roughly the next: Both matrices are examined extensively through cycle routines as the *for,* in such a way that a vector V_{n_p} is obtained simultaneously; that vector contains saved all the unique IDs of the nodes of the system organizing them electrically, so when the algorithm is placed in the column *N* of the row *i*, it analyses if the node of the position *(i,N+2)* is already included in V_{n_p}, in case of not, the ID of the node belonging to the position *(i, N+2)* is saved in the next position of the already commented vector, and consecutively according to the type of the connection in *(i, N+1)* a particular action is carried out in order to save new elements in the connection matrix. For instance, in the case of a type 2 (wire) the code goes to the field *WireModules* to get the modules associated to that particular wire; additionally the maximum temperature of wire is determined by means of the inspection of the harness' segments through which it is extended, so applying the Equation 1 both the real specific resistance and the maximum specific resistance are calculated. All the columns of the connection matrix described in detail in the Section 6.2 are systematically added for each row appearing for each connection. In the case of a type 7 (component), it is evaluated if it is classified as ECU, in that condition each one of its connections given by its pinout relationships are also included (neglecting the signal connections); likewise the maximum withstand current of the component between pins is displayed in the corresponding column, with the aim of limiting the current (specially to the ground) in the power flow calculation.

In the way the code is done the columns *Source* and *Destination* of the final matrices represent the highest and lowest potential pins, respectively. On the other hand, if the node of the position *(i,N+2)* is already present in the vector V_{n_p}, it is studied if the connection *(i,N+1)* is a wire, and in that condition may be necessary to add other rows to the connection matrix, since the ID of that wire could be different to the IDs of the wires already added between the nodes *(i,N)* and *(i,N+2)* or even it is feasible that none connection was considered before between them. In other words, only two or more different connections of type wire (with different ID) can connect two added nodes, and the code is able to detect such condition and discern if add them to the final matrices. According to this operation principles, in the connection matrix are saved all the connections of the system taking into account their modularity, i.e., a wire with particular ID, for instance, can be present in different modules or only in one.

Finally, both the matrix connection and its mater table are given as outputs of the function *Forming Connections*, the last one is just the same than the numeric structure but with columns to each side with string identifier to describe its corresponding value. In the Appendix 15 is presented an example of the table and in Appendix 17 of the matrix.

Algorithm 5: Forming Loads

Input: dataContainer, Master_potential_loads_table, Master_connections_table, connection_matrix

Output: load_matrix_output, Master_loads_table_output, connection_matrix_output, Master_connections_table_output

```
1. for i=1:1:length_rows(dataContainer.all_series_sets.types) do
2.   for j=2:2:length_columns (dataContainer.all_series_sets.types) do
3.     If  ~isnan(dataContainer.all_series_sets.types(i,j)) go to 4. Otherwise to 11
4.     Switch (dataContainer.all_series_sets_types(i,j))
5.     Case 7
6.       location=unique(locateInTable(dataContainer.Devices,'location','and','device_id',dataContainer.all_series_sets_types(i,j)))
7.       type_vector{i,j}= DeviceType(dataContainer, location)
```

```
8.    Case 13
9.      type_vector{i,j}='consumer'
10.   Otherwise
11.  end for
12. end for
13. for i=1:1:length_rows(type_vector) do
14.   Last=''
15.    for j= length_columns(type_vector):-1:1 do
16.       If  strcmp(type_vector{i,j},'Consumer') && isempty(Last) go to 17. Otherwise to 25
17.       Last=type_vector{i,j}
18.       If  strcmp(Last,'Unclear')  go to 19. Otherwise to 20
19.       error('Unclear device in the position (i,j). Please check it out')
20.       If  dataContainer.all_series_sets.types(i,j)==7 go to 21. Otherwise to 23
21.       Add information to the load_matrix_output
22.       consumer_ground(pos_load)=1. pos_load++
23.       If  dataContainer.all_series_sets.types(i,j)==13 go to 24. Otherwise to 29
24.       delete_coil=1. id_coild_to_delete(del)= dataContainer.all_series_sets.ids(i,j). del++
25.       If  strcmp(type_vector{i,j},'Consumer') && ~isempty(Last)&&
          dataContainer.all_series_sets.types(i,j)==7  go to 26. Otherwise to 29
26.       Add information to the load_matrix_output
27.        series_consumer(1,series)= dataContainer.all_series_sets.types(i,j)
28.        series_consumer(2,series)= dataContainer.all_series_sets.ids(i,j). series++
29.     end for
30. end for
31. for i=1:1:length_columns(load_matrix_output) do
32.    for j=1:1:length_columns(series_consumer) do
33.       If  load_matrix_output(3,i)==series_consumers(1,j)&&
          load_matrix_output(4,i)==series_consumers(2,j)  go to 34. Otherwise to 35
34.        binodal_series(i)=1. Break
35.    end for
36. end for
37. Delete repeated binodal loads in load_matrix_output, consumer ground and binodal_series
38. for i=1:1: length_columns(load_matrix_output)
39.   If  consumer_ground (i)==1 && binodal_series (i)==1 go to 40. Otherwise to 41
40.   current_split(i)=1/(length(find(load_matrix_output(2,i)==load_matrix_output(2,:))))
41.   current_split(i)=1/(length(find(load_matrix_output(1,i)==load_matrix_output(1,:)))) Neglect
      connections to ground pines
42. end for
43. load_matrix_output=transpose(load_matrix_output)
44. for i=1:1: length_rows(load_matrix_output)
45.   Add information to the load_matrix_output by searching in the
      Matrix_potential_loads_table
46.   Add information to the Master_loads_table_output by searching in the
      Matrix_potential_loads_table
47. end for
48. If  delete_coil==1 go to 49. Otherwise to 52
49. for i=1:1: length_rows(id_coild_to_delete)
50.       Delete information of the connection_matrix and Master_connections_table
51. end for
52. Organize connection_matrix_output. Organize Master_connections_table_output. End
```

The pseudo code of the Algorithm 5 allows understanding the function *Forming Loads*, which has been programmed in order to get all the final products to the external solver: the connection matrix output, the master connections table output, the load matrix output and the master loads table output. All of them already explained in the Section 6.2.

The code works in this way: From the line 1 to the 12 the matrix type cell *type_vector* is formed containing in the even columns (connections) the corresponding type of the element, consumer or source. Obviously not all the positions are occupied by a char variable, only those where either an element type 7 (Device) or a 13 (coil) is located. This big cell matrix is obtained through the exhaustive evaluation of the already discussed matrices of the field *all series sets* in the *dataContainer*. Afterwards, the function goes through the elements of type*_vector* row by row in ascending direction, but in the reverse way for the columns. This is done in order to take advantage of the benefits of the detecting paths code already explained, which finds intelligently and efficiently all the possible paths between terminals of the battery in the direction of the DC power flow; therefore when ones check the columns in reverse, the first element classified as consumer represents the last bi-nodal load of the path where it is located. For this reason between the lines 13 to 30 the before is carried out, and in parallel the load matrix output is started to be constructed. In these lines also is determined which components should be considered as series consumers located before of other consumer in the direction of the power flow (for instance, the black box Consumer 1 of the Figure 17), also which bi-nodal loads are connecting to the ground (like the loads *B* and *D* of the Figure 17)) and if any coil of the whole system is the 'Last' load of its path, in that case it is added to the load matrix and deleted from the connection structures. Additionally, if the code detects a component classified as 'Unclear' an error message is showed to alarm the user for checking the information of database of such element.

In the lines 31 to 36 is determined which of the bi-nodal loads already saved in the partial load matrix have to be considered as series single consumers (like the load *A* of the Figure 17) by means of a comparison of the loads with the addresses of the components classified as series black box consumers (gotten in the lines 13 to 30). Two or more components behaving as consumers might appear in series in the system because one or more of them in reality are ECUs and due to errors in the information of the database they seem to be loads, in that case warning messages are displayed to advice the user to check if it is necessary to correct this information, because the voltage drop would be higher since an ECU is simply treated as a point of division of current instead of an element causing voltage drop for energy consumption. The line 37 represents a script to delete the information related to repeated single bi-nodal loads, which may appear due to loops in the system.

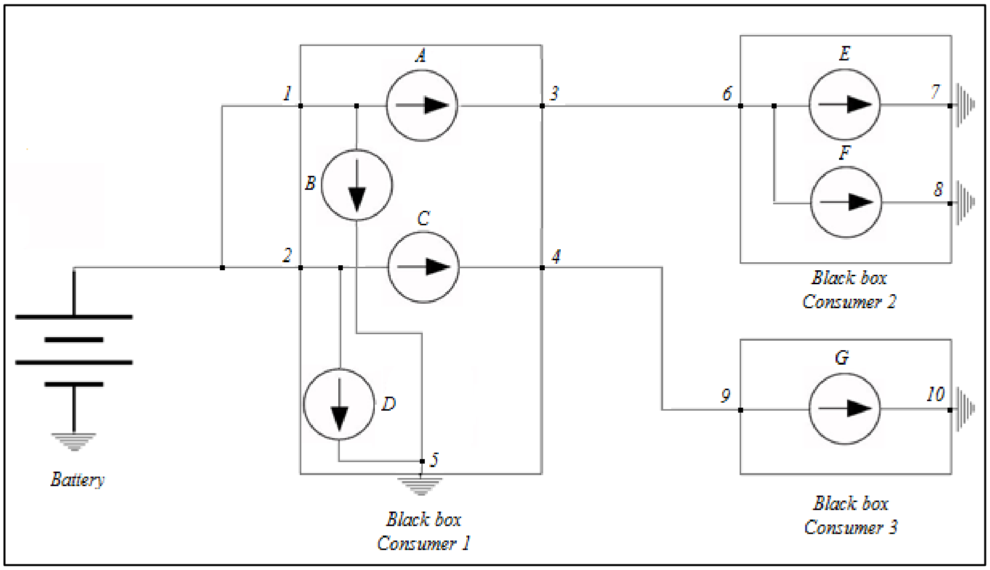


Figure 17: Exemplification of division of current in bi-nodal loads

In the lines 38 to 42 the coefficients of current division for each bi-nodal load are obtained. Previously it has been explained that for each bi-nodal load it is determined if it is belonging to a component classified as consumer located before the last one and in a series chain (Consumer 1 of the Figure 17. The Consumer 2 and 3 are the last ones). The criteria to determine the aforementioned coefficient changes according to the next:

- If a bi-nodal load belongs to a component considered as in-series and that load is connected to ground (loads *B* and *D* of the Figure 17), the current associated to the load is equal to the current of its ground pin (number 5 in the image) over the number of input pins related to it (that is two, the pins 1 and 2). So, the coefficient for the loads *B* and *D* in the example is 0.5 (1/2). With current coefficient is meant the factor by which the current of the pin is multiply to get the current of a particular bi-nodal load.

- If a bi-nodal load belongs to a component considered as in-series and that load is not connected to ground (loads *A* and *C* of the Figure 17), the current associated to the load is equal to the current of its input pin (number 1 for *A* and number 2 for *C*) over the number of output pins related to it and neglecting the connections to ground pins (that is one for both, the pins 3 and 4, respectively). So, the coefficient for the loads *A* and *C* is 1.
- If a bi-nodal load belongs to a component considered the last of a path (loads *E*, *F* and *G* of the Figure 17), the current associated to the load is equal to the current of its input pin (number 6 for *E* and *F* and number 9 for *G*) over the number of output pins related to it (that is two for *E* and *F*, the pins 7 and 8 and one for *G*, the pin 10). So, the coefficient for the loads *E* and *F* is 0.5 and for *G* is 1.

In the line 43 to 47 the load matrix output and the Master loads table output are finally confectioned. This is done by relating to each bi-nodal load the information of its associated component (location, part numbers, and modularity) by means of the Master potential loads table produced with the Algorithm 3. Regarding the type of load and the nominal voltage, the next scenarios are contemplated:

- If a bi-nodal load belongs to a component considered as in-series and that load is connected to ground (loads *B* and *D* of the Figure 17), the initial type of load sent to the external solver is constant current and the nominal voltage is considered 14 volts.
- If a bi-nodal load belongs to a component considered as in-series and that load is not connected to ground (loads *A* and *C* of the Figure 17), the initial type of load sent to the external solver is constant impedance and the nominal voltage is considered 0.5 V.

- If a bi-nodal load belongs to a component considered the last of a path (loads *E*, *F* and *G* of the Figure 17), the initial type of load sent to the external solver is constant current and the nominal voltage is considered 14 volts.

Also it is considered the current of each bi-nodal current according to its part number. In the lines 48 to 51 if any coil was found to be a consumer of path it is deleted of the connection matrix and table and the final ones are rearranged. In the Appendix 16 is presented an example of the master load tables and in the Appendix 18 is shown examples of the matrix.

CHAPTER VII

POWER FLOW SOLUTION APPROACHING

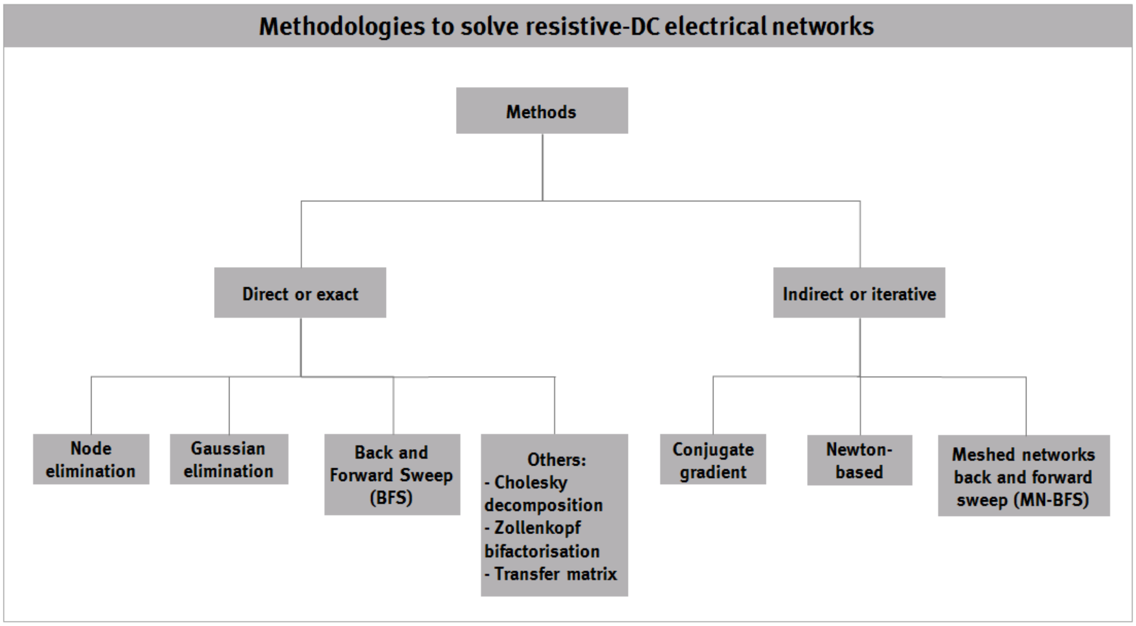


Figure 18: Different methods applied for solution of resistive-DC electrical networks

The state of the art regarding the solution of resistive-DC electrical networks is wide and basically based on mathematical models developed in the field of linear algebra. In the Figure 18 is displayed a rough classification of some methodologies available in the bibliography to solve this kind of systems, from a point of view of the number of iterations for the solutions assuming constant current loads (it will be explained that those methods are equally valid for voltage-dependent loads, however as explained for the Figure 9 an approach considering constant current is the one used for dimensioning wires and fuses in the *VOBES* process). In the Section 7.1 are explained in more detail each one of the methodologies schematized in the Figure 18.

7.1. Description of the methods

Node elimination:

This method represents pure physical substitutions by means of replacements of physical elements with equivalent physical ones. A resistive network can be considered as a set of points joint by resistances; thereafter each node has a voltage V_o and the conductance between the node *i* and *j* is $G_{ij} = \frac{1}{R_{ij}}$. A node in a resistive network can be either internal or external, being the latter the ones connected to the surroundings and with a certain voltage or current flowing out of them. Examples of such nodes are the junctions *A*, *B*, *C* and *D* in the Figure 19. Internal nodes are those surrounded by the externals without any voltage or current source connected to them, being their voltages the unknown variables of the system; nodes *E*, *F*, *G* and *H* are examples of them.

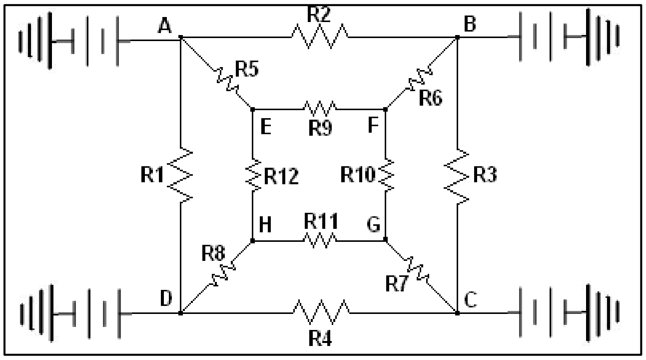


Figure 19: Example of external and internal nodes

Through the Kirchhoff's current law the balance equation in a node O, connected to N nodes through elements with electrical conductance expressed by $\boldsymbol{G_{oi}}$, is given by the Equation 3:

$$\sum_{i=1}^{N} G_{oi}(V_o - V_i) = \begin{cases} 0, & \textit{Internal node} \\ I_o, & \textit{External node} \end{cases}$$

Equation 3

In the Equation 3 if one considers an internal node, the voltage on it is given by the Equation 4

$$\boldsymbol{V_o} = \frac{\sum_{i=1}^{n} \boldsymbol{G_{oi} V_i}}{\sum_{i=1}^{n} \boldsymbol{G_{oi}}}$$

Equation 4

From the Equation 4 can be concluded that to know the voltage of a node is enough to know the voltage of its neighbors.

Now, in order to solve the system, one starts 'eliminating' nodes, in such a way, that the connections between the node to eliminate and its neighbors are substituted by additions of incremental conductance between the neighbor nodes. The Figure 20 displays an example of such elimination: the node *O* is made to disappear, and instead of it, new connections between all the neighbors are projected. In this example before the elimination of the node *O* there was no connection between the other nodes, therefore new ones are added, in case a connection did exist, the new conductance is summed up to the old one. The Equation 5 allows obtaining the incremental conductance between the nodes 'j' and 'k' when eliminating the node 'O'.

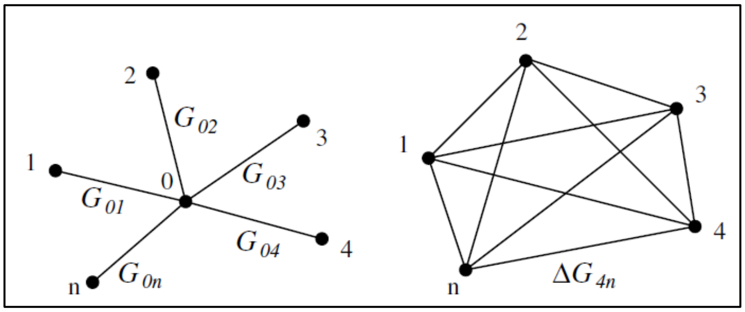


Figure 20: Removing a node from the system [17]

$$\Delta \boldsymbol{G_{jk}} = \frac{\boldsymbol{G_{oj} G_{ok}}}{\sum_{i=1}^{n} \boldsymbol{G_{oi}}}$$

Equation 5

This procedure is repeated until all the internal nodes are eliminated and the corresponding updating of the network is carried out according to the Equation 5. This procedure is called the *forward substitution* and is finished only when the external nodes are remaining in the system. If one is interested in finding the voltage at each node, the *backward substitution* process is called, i.e., by means of the Equation 4 the voltages are calculated which is equivalent to 'insert back' the nodes to the system in the exact reverse order of the elimination. From the before is concluded that in order to calculate the voltage of one node, one only needs to know with certainty the indexes (before the elimination, indexing of the nodes is necessary to implement) of the neighbors nodes at the moment of the removal. Hence it is necessary to define an efficient elimination order before the whole process is started, with the aim to count with a robust and fast algorithm [17].

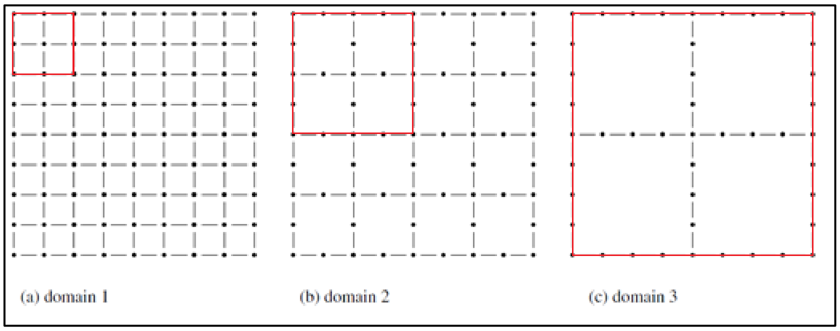


Figure 21: Hierarchical domain structure technique [17]

Many researches have been carried out in order to find an optimal elimination order. In [17] has been proposed a strategy based in domain structures where larger domains are made each from four smaller domains: In the Figure 21, the image (a) is the initial system, the square of the left upper part formed by 10 nodes is a larger domain (enclosed in red lines), formed itself by 4 smaller domains (one of them is the square of the very left corner). When the center nodes of the larger domains are removed the remaining boundaries look like the image (b) of the Figure 21. In the image (b) appears 4 new larger domains (one of them enclosed with red lines), where each of them have 5 internal nodes to remove: by applying the principle of keeping the number of connections as low as possible, it is doable to establish an ideal order to eliminate them, first at all the node between the two upper smaller domains is removed, secondly the node between the lower smaller domains, and finally the horizontal line inside of each larger domain. After that the system is transformed as shown in the image (c), the external nodes define the larger domain and the vertical and horizontal lines are eliminated successively. This strategy proposed in [17] is easily extendable for larger and larger domains, and it has demonstrated to work quite fine in tests of random resistive networks, allowing getting very close to optimal results in terms of time processing.

Gaussian elimination:

The Gaussian elimination represents the optimal matrix representation of the physical substitution method explained before as nodal elimination. This means both methods are equivalent in the sense of the use of same mathematical principia [18]. This is going to be explained in the following.

The matrix representation of the Equation 3 is given by the Equation 6:

$$\mathbb{G}\boldsymbol{V} = \boldsymbol{I}$$

Equation 6

Where V is a voltage vector containing the nodal voltages, I is a vector with the nodal current injections representing the values of current flowing out of the nodes. For external nodes one of the two positions are specified: either their voltages or current injections are given by the boundary conditions. For internal nodes the current injections are equal to zero and their voltages are unknown variables of the system. $\mathbb{G}$ is the conductivity matrix and it stores the values of conductance between the nodes i and j and to a node itself, i.e., a position $\boldsymbol{G_{ij}}$ contains the value of conductance between the aforementioned nodes, but the position $\boldsymbol{G_{ii}}$ contains the negative value of the total conductance connected to the node i. The matrix $\mathbb{G}$ represents a linear cause-effect relationship, because each row is a node and each columns represents a nodal voltage, hence the matrix expresses for a row i which nodal voltages j are involved in the Kirchhoff current law in that node i [19]. An expansion of the conductivity matrix is given by the Equation 7.

$$\mathbb{G} = \begin{pmatrix} -\boldsymbol{S_0} & \boldsymbol{G_{01}} & \cdots & \boldsymbol{G_{0k}} & \cdots & \boldsymbol{G_{0N}} \\ \boldsymbol{G_{10}} & -\boldsymbol{S_1} & \cdots & \boldsymbol{G_{1k}} & \cdots & \boldsymbol{G_{1N}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{G_{k0}} & \boldsymbol{G_{k1}} & \cdots & -\boldsymbol{S_k} & \cdots & \boldsymbol{G_{kN}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{G_{N0}} & \boldsymbol{G_{N1}} & \cdots & \boldsymbol{G_{Nk}} & \cdots & -\boldsymbol{S_N} \end{pmatrix}$$

Equation 7

In the Equation 7, $S_i = \sum_{\substack{\boldsymbol{n=0} \\ \boldsymbol{n \neq i}}}^{\boldsymbol{N}} \boldsymbol{G_{in}}$ is the total conductance connected to the node i, these terms are located in the main diagonal of $\mathbb{G}$ as negative values. N represents the total number of nodes of the system. As it can be appreciated the conductivity matrix is a symmetric structure, since the conductivity between two connected nodes is not dependent on the direction ($\boldsymbol{G_{ij}} = \boldsymbol{G_{ji}}$), other attribute is its considerable sparseness due to only a small number of nodes are linked between them (in comparison to the complete network).

To solve the system given by the Equation 6, the Gaussian elimination method conceives summing up multiples of the zeroth row (see Equation 7) to all the other rows (from 1 to N), in such a way that the elements under the column where the element S_0 is, are made zero. In order to do so, each element of the row zero is multiplied into the term $\frac{G_{i0}}{S_0}$ and then element by element the row i is added to the modified row zero (0). The Equation 8 represents mathematically the term to add to each term of the row i, where $i \neq 0$:

$$\Delta \boldsymbol{G_{ik}} = \boldsymbol{G_{0k}} \frac{\boldsymbol{G_{i0}}}{\boldsymbol{S_0}} = \frac{\boldsymbol{G_{0k}} \boldsymbol{G_{i0}}}{\sum_{\substack{\boldsymbol{n=0} \\ \boldsymbol{n \neq i}}}^{\boldsymbol{N}} \boldsymbol{G_{0n}}}$$

Equation 8

Once the elements G_{10} to G_{N0} are equal to zero the matrix $\mathbb{G}$ is turned into $\mathbb{G}'$ according to the Equation 9.

$$\mathbb{G}' = \begin{pmatrix} -\boldsymbol{S_0} & \boldsymbol{G_{01}} & \cdots & \boldsymbol{G_{0k}} & \cdots & \boldsymbol{G_{0N}} \\ \mathbf{0} & -\boldsymbol{S_1} & \cdots & \boldsymbol{G_{1k}} & \cdots & \boldsymbol{G_{1N}} \\ \mathbf{0} & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \boldsymbol{G_{k1}} & \cdots & -\boldsymbol{S_k} & \cdots & \boldsymbol{G_{kN}} \\ \mathbf{0} & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \boldsymbol{G_{N1}} & \cdots & \boldsymbol{G_{Nk}} & \cdots & -\boldsymbol{S_N} \end{pmatrix}$$

Equation 9

From now the process is repeated having as a reference row the row one (1) instead of the already processed row zero (0), it means that this row and the column zero (0) are discarded and the remaining conductivity matrix is still symmetric (columns and rows from one until N). The final objective is to get an upper triangular matrix and to thus calculate the nodal voltages through *backward substitution*.

This method is exactly the same than the already discussed node elimination [18]. If one compares the Equation 5 and Equation 9 it is simple to realize that both are totally equal. Actually the order of the equations in the matrix $\mathbb{G}$ represents the elimination order already explained in the node elimination approach. The next advantages of these both methods can be remarked:

- Through these methods not only electrical systems can be studied, but also in general terms, physical problems governed by transportation laws, such as fluid flow through porous media (network of tubes in which the fluid flows) [20] or fracture models oriented to study the cascade effect of fuses blowing in distribution networks [21].
- In resistive-DC electrical networks, if the boundary conditions are given with constant current sources, these methods allow solving exactly the equations system of Equation 6 in the way already explained; however if the current sources show a voltage-dependence (such as constant power loads explained already in the Equation 2), an iterative process

where the value of the currents in the boundary conditions can be systematically changed until a convergence point is found in the system. This can be carried out with the Gaussian elimination optimally since the right-side of Equation 6 is the only part modified (the left-side contains the conductance of the connections which are static variables), so under this condition in the *forward substitution* process just the current vector require to be updated after the calculation of the first solution of the system; therefore taking into consideration that the *backward substitution* is way faster compared the previous process, time saving is gotten by means of the application of these methods.

- In the research performed by the authors in [17] was demonstrated the Gaussian elimination is the fastest method for solving highly-meshed random networks compared to techniques such as transfer matrix and conjugate gradient method for small-medium size systems. Additionally was also determined that the Gaussian elimination is insensitive to the range of variation of the coefficients in the matrix $\mathbb{G}$, i.e., those coefficients can change more than one order magnitude and the time in solving is not compromised.

Likewise, the next disadvantages might be highlighted:

- Since for getting a fast solution is required to establish a smart elimination order, and lists of nodes need to be stored for each node, for big systems the memory consumption starts being an issue when applying these methods.
- In [17] was demonstrated as well the increase in time for solving systems in function of the size of the system.
- If from the outset the topology makes it difficult to create domains and sub-domains structures to find the optimal elimination order, these methods may not be the ideal ones to be selected.

- These methods could be considered more difficult to program compared to other simpler options (to be explained ahead).

Back and Forward Sweep (BFS):

This method is based in the graph theory of direct acyclic connected graphs. The graph theory is a field of mathematics in charge of the study of structures which express pair-wise relations between objects. In the Figure 22 is presented an example of a graph, which is basically a collection of nodes joined by edges; in this example there are $N = 9$ nodes (the ones enclosed in circles) and $m = 18$ edges (the lines). Some of the attributes of a graph are its direction and level of interconnection: regarding the first one a graph is classified as direct if there is an associated direction in the edges (this particularity depends on the nature of the phenomenon being modeled as a graph), otherwise is considered as indirect. On other hand, a graph is acyclic if it doesn't have cycles, i.e., without loops in the connections between nodes, and in the other way around is tagged as cyclic. In order to determine mathematically the number of cycles or loops of a connected graph the Euler's formula [23] is used (See Equation 10).

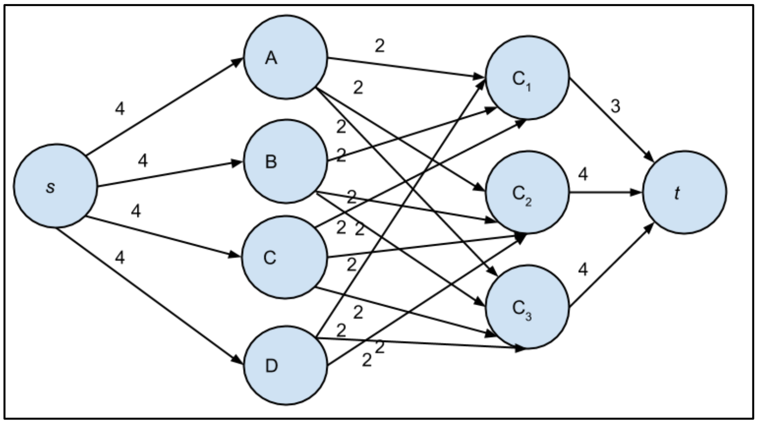


Figure 22: Example of a graph [22]

$$\boldsymbol{N - m + loops = 1}$$

Equation 10

As commented before, the BFS method is a straight-forward strategy to solve transportation equations of resistive-DC electrical networks that can be modeled as direct (the flow of current causes the system to be direct since it defines the voltage drop, i.e., $V_{ij} \neq V_{ji}$) and acyclic (without loops) under the graph theory field applied into electrical networks. In electrical power system terms this means totally radial systems, i.e., without redundant paths. In this work is commented the application of the BFS method for DC networks, but in general it can be easily extended for AC systems as is explained in [24] where a power flow is executed through this technique.

The formulations of the BFS method have been extracted from [25] following the compact matrix-based form easily deducted by means of the application of the graph theory:

$$\boldsymbol{\Gamma^T \cdot I_B^T + I_d \cdot I_N^T = 0}$$

Equation 11

$$\boldsymbol{\Gamma \cdot V^T - R_B \cdot I_B^T = 0}$$

Equation 12

In both Equation 11 and Equation 12 the matrix **Γ** appears. This is the node incidence matrix where each row represents a branch of the system and each column a node of the same. The construction of **Γ** is done following the next rules [23]:

- $\boldsymbol{\Gamma_{ij} = 1 \; when \; the \; tail \; of \; the \; edge \; i, is \; vertex \; j}$
- $\boldsymbol{\Gamma_{ij} = -1 \; when \; the \; head \; of \; the \; edge \; i, is \; vertex \; j}$
- $\boldsymbol{\Gamma_{ij} = 0 \; Otherwise}$

The Equation 11 is no other thing but the extension of the Kirchhoff Current Law (KCL) in a matrix form. $\boldsymbol{I_B}$ is a row vector containing the branch currents in the same order than the rows of **Γ**. $\boldsymbol{I_d}$ is the identity matrix. $\boldsymbol{I_N}$ is a row vector with the nodal current injections being positive when the current is flowing out from the node and negative otherwise. Since the system is completely radial (zero loops) through this equation all the branches current can be calculated isolating the term $\boldsymbol{I_B}$, for that the inverse of $\boldsymbol{\Gamma^T}$ is calculated and since is a considerably sparse matrix is no a computational high-demanding operation. However, to be able to perform $\left(\boldsymbol{\Gamma^T}\right)^{-1}$ one first need to discard the first row of $\boldsymbol{\Gamma^T}$ because according to the Equation 10 when $\boldsymbol{loops} = \boldsymbol{0}$, $\boldsymbol{N} = \boldsymbol{m} + \boldsymbol{1}$, meaning that the number of nodes is 1 unit higher than the number of edges (branches), therefore $\boldsymbol{\Gamma^T}$ is not square and hence no-invertible. The application of this equation is denominated *backward sweep* (BS).

The Equation 12 is the matrix form of the Kirchhoff Voltage Law (KVL). $\boldsymbol{V}$ is a row vector with the nodal voltages all with the same reference than the slack node (by convention the first position, i.e., $\boldsymbol{V}(\boldsymbol{1})$). $\boldsymbol{R_B}$ is a diagonal matrix with the value of branch resistances in the same order than the rows of $\boldsymbol{\Gamma}$. Once the Equation 11 has been applied the Equation 12 is used replacing the obtained value of $\boldsymbol{I_B}$ into this expression and getting $\boldsymbol{V}$. Note that in this operation is required to perform the inverse of $\boldsymbol{\Gamma}$, applying the already explained about the simplicity and procedure of this operation. The application of this equation is denominated *forward sweep* (FS).

The main advantage of this method obviously is its simplicity but also the main drawback is its limitation regarding the gamma of applications where it can be used: only to systems where the number of loops is equal to zero (totally radial distribution networks). However this method has been presented because with some modifications the MN-BFS method is created, offering more flexibility and robustness as it is going to be explained ahead.

Others:

In [17] has been demonstrated that the transfer matrix method is just a specific case of the Gaussian Elimination method, using an elimination order far to the optimal. Thereafter the Gaussian approach is faster at any case, but the transfer matrix also shows good results for very narrow and long strips.

The Cholesky decomposition method consists on expressing a symmetric matrix $\mathbb{G}$ as the product of a lower triangular matrix $\boldsymbol{L}$ into its transposed $\boldsymbol{L}^{\boldsymbol{T}}$ [26] and afterwards solving the formula $\boldsymbol{Ly} = \boldsymbol{I}$ (keeping the same symbols of the Equation 6) and finally getting the nodal voltages $\boldsymbol{V}$ with the expression $\boldsymbol{LV} = \boldsymbol{y}$. A variation of this method is the LDL decomposition.

The Zollenkopf bifactorisation is a special modification of the Gaussian Elimination [27], consists on expressing the inverse of the matrix $\mathbb{G}$ as the product of $2N$ factor matrices: $\boldsymbol{L}^{\boldsymbol{N}} \cdot \boldsymbol{L}^{\boldsymbol{N-1}} \cdots \boldsymbol{L}^{\boldsymbol{1}} \cdot \mathbb{G} \cdot \boldsymbol{R}^{\boldsymbol{1}} \cdots \boldsymbol{R}^{\boldsymbol{N}} = \boldsymbol{U}$; so the problem is turned into the calculation of the individual factor matrices $\boldsymbol{R}$ and $\boldsymbol{L}$.

Other exact methods to solve linear equations systems have been proposed in the mid of the last century (decades 60, 70 and 80) using complex matrix mathematical analysis; the aim of this work is not to analyze them and test them profoundly in a comparison framework, but to show the wide variety of techniques present in the state of the art. It is clear that the reasons for selection of one method over other are dependent directly on the particular considerations of each problem to solve, for instance, the size, boundary conditions, variations of the coefficients, required speed solution, complexity, etc. In the field of study of random networks (field resulted from the mix of probability theory and graph theory) at the percolation threshold (or upper or lower) the selection of the method is even more critical in function of the aforementioned parameters. Likewise, as it is going to be showed with the method chosen to solve the power flow of the on-board distribution network of a vehicle, the particular needs and characteristic of this system may fit better in one strategy rather than in the others.

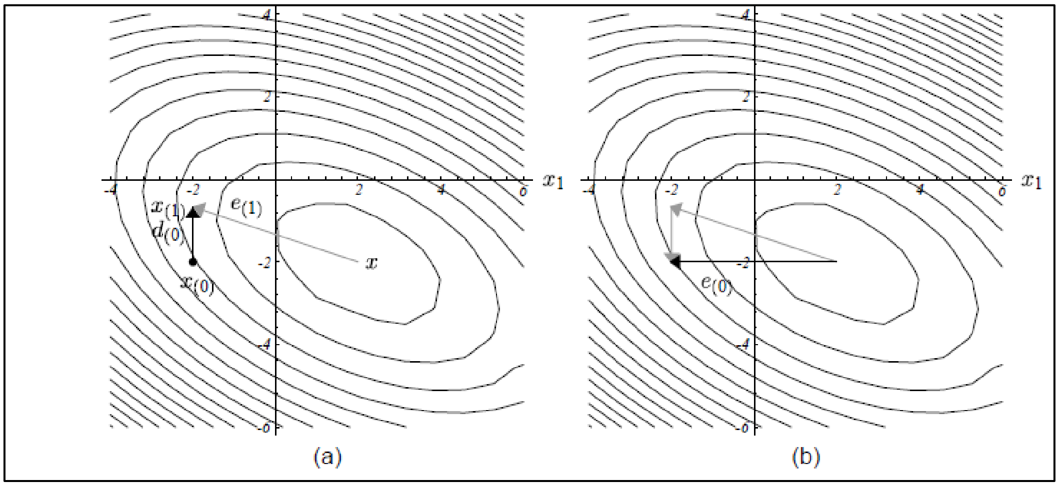
Conjugate gradient:

When the matrix $\mathbb{G}$ is too large and considerably sparse the conjugate gradient method is the one recommended to solve the system of equation given in the Equation 6 [28]. This is an iterative process applicable for square-symmetric-positive-definitive matrices, such as the case given in resistive-DC electrical networks. This is based roughly in the fact that for a symmetric positive-definite matrix $\mathbb{G}$ the gradient of its quadratic form can be expressed according to the Equation 13:

$$\boldsymbol{f}'(V) = \mathbb{G}\boldsymbol{V} - \boldsymbol{I}$$

Equation 13

The right-side part of the Equation 13, according to the Equation 6 if $\boldsymbol{V}$ is its solution, is equal to zero. So one can conclude that the solution of this system of equations is equivalent to the minimization of its quadratic form expression: $\frac{1}{2}\boldsymbol{V}^{\boldsymbol{T}}\mathbb{G}\boldsymbol{V} - \boldsymbol{I}^{\boldsymbol{T}}\boldsymbol{V} + \boldsymbol{c}$.



(a)

(b)

Figure 23: Illustration of the conjugate gradient solver [28]

The aforementioned can be appreciated in the Figure 23. Let's consider that the matrix $\mathbb{G}$ has a dimension $\boldsymbol{N = 2}$ and the vector $\boldsymbol{X}$ in the image (a) (in this situation $\boldsymbol{X}$ represents the unknown variables, that is $\boldsymbol{V}$) is the final solution (point where $\boldsymbol{f}'(V) = \boldsymbol{0}$). The first guess of the solution is given by $\boldsymbol{X}_{(\boldsymbol{0})}$; the first step of the iteration to solve the system is taken along some direction $\boldsymbol{d}_{(\boldsymbol{0})}$,, the minimum point $\boldsymbol{X}_{(\boldsymbol{1})}$ is chosen by the constraint that $\boldsymbol{e}_{(\boldsymbol{1})}$ must be $\mathbb{G} - \boldsymbol{Orthogonal}$ to $\boldsymbol{d}_{(\boldsymbol{0})}$. In the image (b) of the Figure 23 the initial error $\boldsymbol{e}_{(\boldsymbol{0})}$ can be expressed as a sum of $\mathbb{G} - \boldsymbol{Orthogonal}$ components drawn with gray arrows, each step of the conjugate direction eliminates one of those components, therefore the final solution is found in $\boldsymbol{N}$ iterations. The fact that $\boldsymbol{e}_{(1)}$ is $\mathbb{G}-\boldsymbol{Orthogonal}$ to $\boldsymbol{d}_{(0)}$ is the reason for naming the method as conjugate gradient. According to the explained the Algorithm 6 presents the pseudo code of this method.

Algorithm 6: Conjugate Gradient Method (CGM)

```
Input: $\mathbb{G}, I, V_{initial}$
Output: $V$
$e_{initial} = I - \mathbb{G}V_{initial}$
$d_{initial} = e_{initial}$
for i=1:1:length_rows($\mathbb{G}$) do
    $\alpha = ({e_{initial}}^T \cdot e_{initial})/({d_{initial}}^T \cdot \mathbb{G} \cdot d_{initial})$
    $V_{actual} = V_{initial} + \alpha d_{initial}$
    $e_{actual} = e_{initial} - \alpha \cdot \mathbb{G} \cdot d_{initial}$
    $\beta = ({e_{actual}}^T \cdot e_{actual})/({e_{initial}}^T \cdot e_{initial})$
    $d_{actual} = e_{actual} + \beta d_{initial}$
    $V_{initial} = V_{actual}$
    $e_{initial} = e_{actual}$
    $d_{initial} = d_{actual}$
end for
$V = V_{actual}$. End
```

In the Algorithm 6 if the system counts with one or more slack voltage nodes, before solving it one should discard the rows and columns in $\mathbb{G}$ corresponding to those nodes, as in the vector $\boldsymbol{V}$ those positions don't need to be calculated: they are inputs for the system. The next advantages of this method can be remarked:

- According to the results found in [17], this method is faster and more robust for big size systems (the concept of the size is relative, but in the literature a system with number of nodes in the order of thousands is considered 'big'), which have as feature a sparse matrix $\mathbb{G}$.
- The programming is easy according to the presented in the Algorithm 6.

On other hand these are the drawbacks:

- For boundary conditions changing over time, and even for loads with currents dependent on voltage, the system needs to be solved systematically from scratch in function of the displayed in the Algorithm 6; in contrast with other methods such as the Gaussian Elimination where only the left side of the equations system needs to be updated. This disadvantage causes of course an increase in the time for solving the system when comparing both methods.
- The conjugate gradient method doesn't handle so properly the solution of systems when the coefficients of $\mathbb{G}$ vary several magnitude orders when compared to the Gaussian Elimination: the method gets slower [17].

Newton-based methods:

The techniques based in Newton-Raphson numerical method are widely used for solving no-linear systems through approximations by means of series expansions such as the Taylor series. It is based in the Jacobian matrix which contains the derivatives of the objective functions respecting the unknown variables of the system. This method is used not only for calculating power flow in AC systems, but also in dynamic DC systems like train lines [29]; however this method is highly affected by the initial conditions of the system, its lack of robustness and problematic for no-smooth no-linear voltage-dependent loads [25]. For constant current loads does not make any sense the application of this method, this is only applicable for no-linear consumers.

Meshed Networks Back and Forward Sweep (MN-BFS):

This method is a variation of the Back and Forward Sweep (BFS) technique already explained before. It is proved to be a robust, efficient and fast method for solving big-size weakly meshed networks not only for constant current loads (the case for the power flow approaching in this Chapter) but also for voltage-dependent loads [25] (the external power flow solver will use also this method taking into consideration loads with constant power and constant impedance characteristics). This is the main reason to select this method for the case of the present work regarding the solution of the on-board distribution network of vehicles, as in the Section 7.2 the details of the MN-BFS method are disclosed it is going to be commented its advantages when applied for this system and also in the Section 7.3 where it is explained its adaptation.

7.2. In-detail description of the adopted method (Pseudo codes)

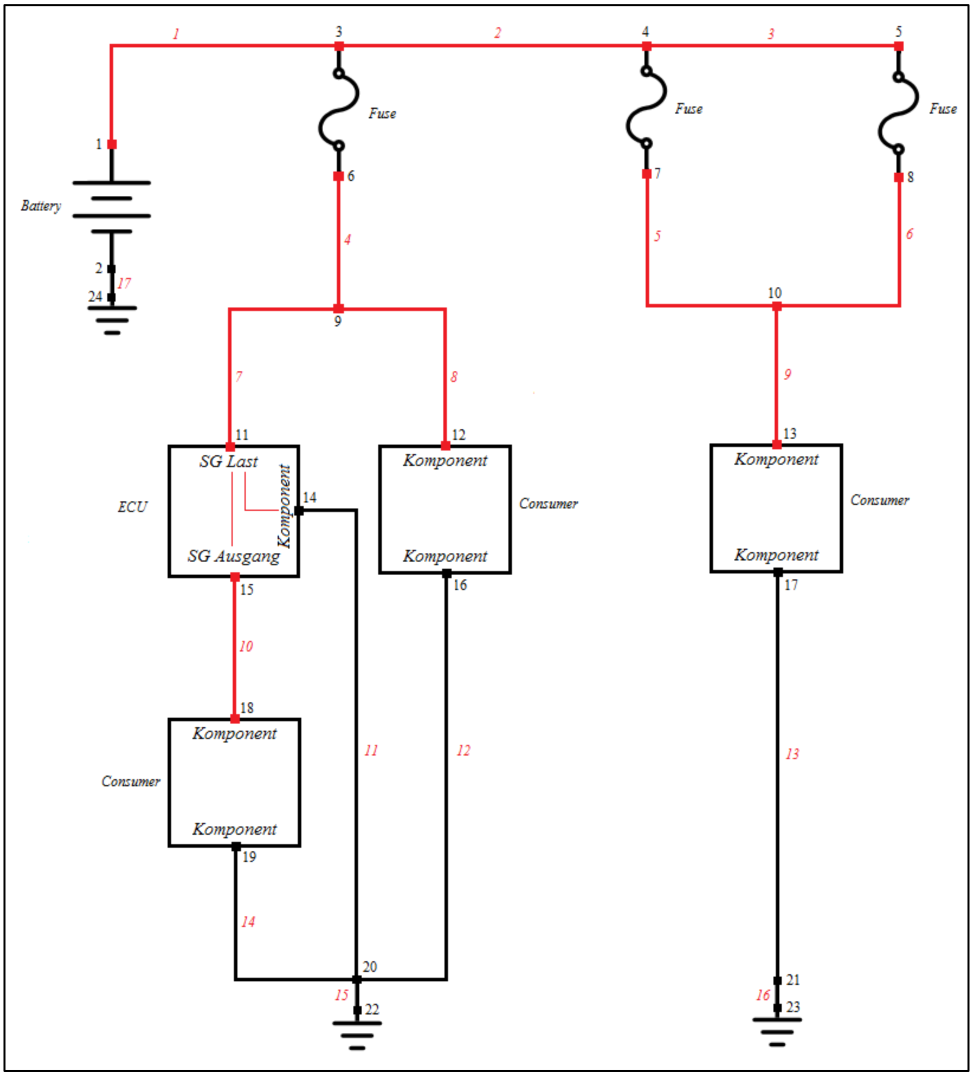


Figure 24: Electrical diagram example of an on-board network distribution system in a vehicle

An on-board distribution network in a vehicle is formed by two sub-grids: the grid where all the connections from the positive battery terminal are done, and on other hand, the grid of connections to the negative battery terminal. See the Figure 24 (the ground symbol represent ground bolts, interconnected between them through the body of the vehicle). In the Figure 24 can be appreciated an example of an electrical diagram, in red lines are drawn the wires of the first grid, whereas in black lines the wires of the second one. Since all the information of the network is stored and organized as a whole in the QT file, BOM lists and Wire lists, the indexing of nodes, wires, components, fuses, etc, is also carried out as a full set, just as displayed in the example. The interface between the 'positive grid' and the 'negative grid' are the constant current bi-nodal loads; for instance, between the nodes 18 and 19 (belonging to a black box consumer), there is a bi-nodal load, which for practical effects is considered as two independent but related nodal current injections for both grids. See Figure 25 (a) and (b).

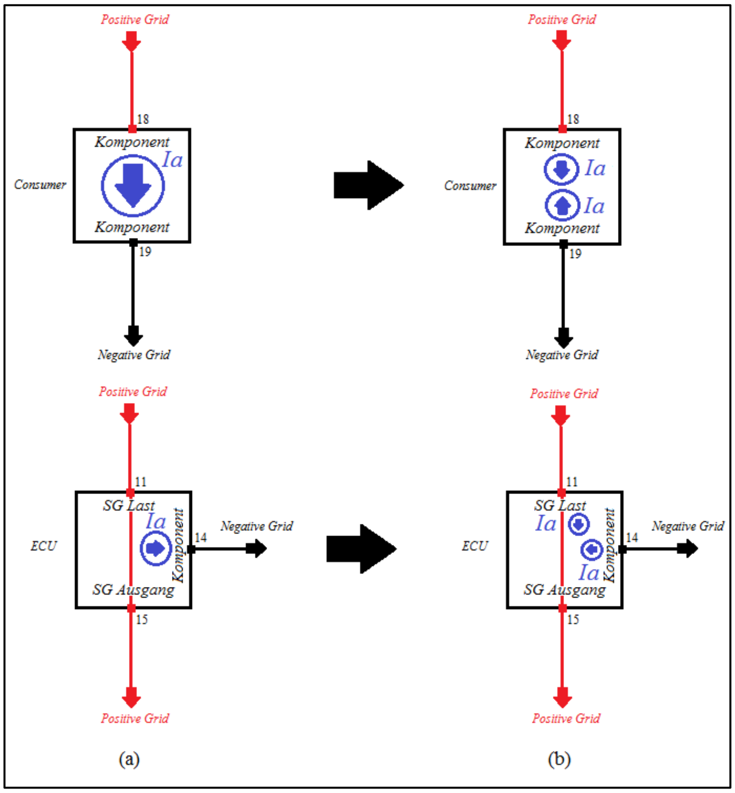


Figure 25: Transformation of components into current loads

In the Figure 25 is presented the aforementioned conversion: For black boxes behaving as consumers appear a number of loads equal to the number of relationships pin-pout associated to that component. The loads can be seen as current injections to the nodes where they are connected, in order to guarantee the same total amount of current in battery's terminals, a distribution of current between the nodes need to be done. In the case of the example, since for the consumer component there is only one relationship (18-19) the nodal injection current in the node 18 (positive grid) and in the node 19 (negative grid) are equal, but in other situation would be possible to find two relationships inside of a component, for instance (18-19) and (18-50), (let's assume the pins 18, 19, 50 are associated all to the same component), therefore the current injections assigned to the nodes 19 and 50 (both belonging to the negative grid) are exactly the half of the current associated to the pin 18. In the case of components classified as ECUs the procedure is exactly in the other way around: the current associated to a ground pin is split between its associated input pins; in the case of this example, in the lower image of the Figure 25 (a) there is only one input pin linked to the ground pin 14 (pin 11), therefore there is only one current load inside of this component, equal to the current associated to the pin 14, which is on other hand modeled as two independent but related nodal current injections for both grids (See Figure 25-b). Other approach might consider calculating the current between the nodes 11-14 by means of equivalent Thevenin resistance and voltages between those pins; however in most of the ECUs components the calculated value of those currents might be higher than the current given in their datasheets, hence it might be a good approximation to consider that the device itself limits internally the current to ground and consequently to make a distribution of that current is a reasonable solution.

The final conclusion is that for solving the whole system of an on-board distribution network, two different but related systems have to be solved: the positive and the negative grids. The final results are gotten by means of the merger of the results of both subsystems. In the Figure 26 image (a) the red wires all are belonging to the positive grid, which has a slack voltage equal to the value assumed in the battery. The black wires belong to the negative grid, which has a slack voltage of 0 Volts (Figure 26 (b)).

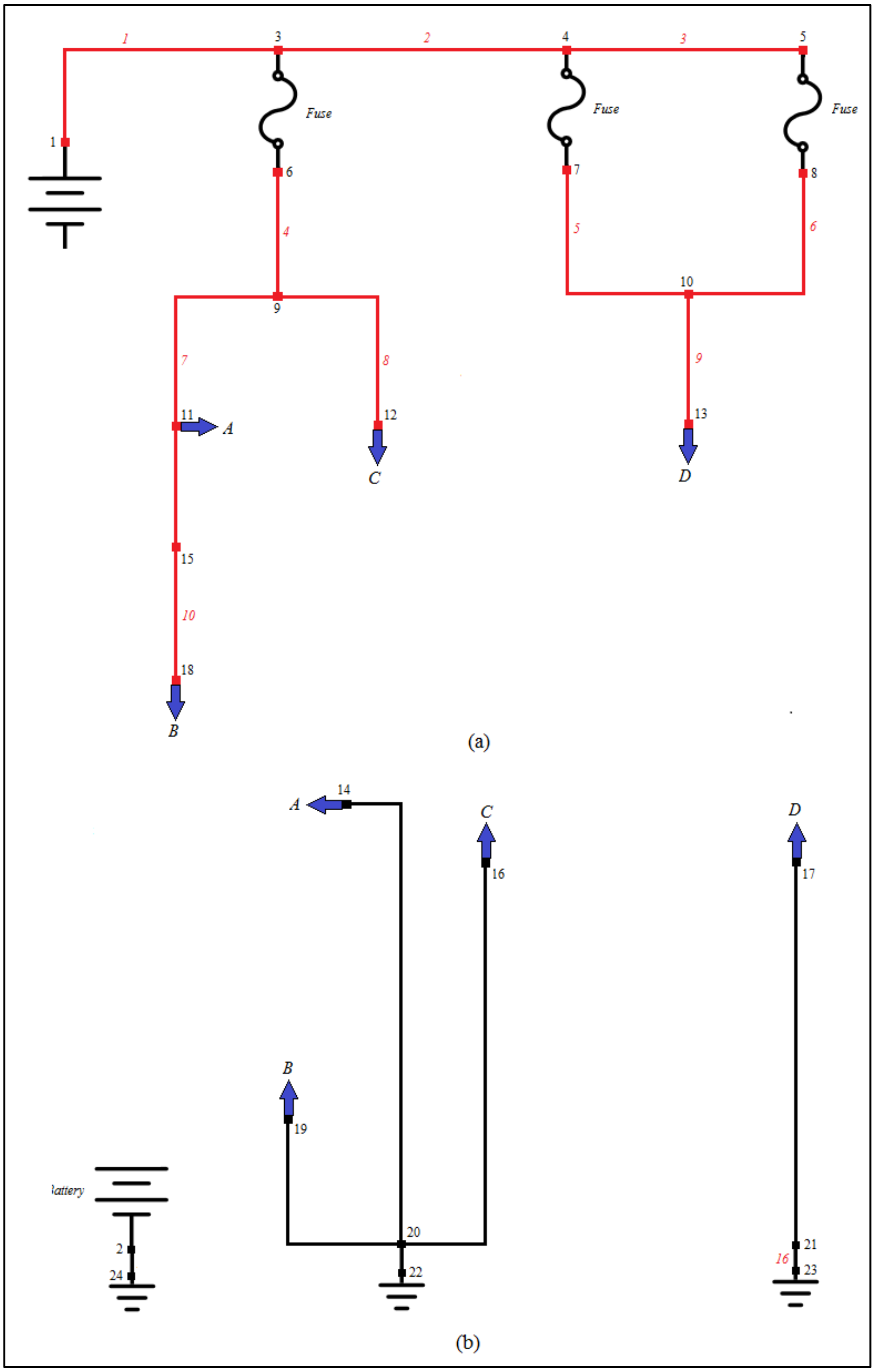


Figure 26: Division of the electrical diagram of the Figure 24

The constant current MN-BFS algorithm fits properly under this philosophy for tackling the system's solution. In next a step by step guide with pseudo codes of this method is provided based on the work of [25].

It was said before that the method BFS only works for radial systems; In fact, it is a direct process which doesn't require iterations because basically is a straight-forward matrix calculation approach. The MN-BFS is a variation of the BFS algorithm useful for weakly meshed networks. A weakly meshed network is a grid with low number of loops in comparison with the total amount of lines or connections on it. The system given in the image (a) of the Figure 26 is an example of a weakly meshed network, because it has only one loop according to the Equation 10 ($N = 14$ and $m = 14$); in this way is not possible to use the Equation 11 and Equation 12 because the matrix $\mathbf{\Gamma}$ is not square when the column of the slack node is removed from it. In order to solve that system, if one 'cuts' the connection between the nodes 8-10, then the system becomes totally radial and the back and forward sweep is implementable. This is other of the reasons to choose this method: it is almost a straight-forward calculation except the cut branches which require a fast-iterative process to determine its current; therefore for networks with high degree of sparseness this algorithm is way faster and robust than Gaussian Elimination (GE) or Conjugate Gradient Method (CGM), for instance. Loops like the one presented in the Figure 26 are common in the on-board distribution networks of vehicles, mainly caused due to soldered splices such as the nodes 9 and 10 in that image, and are done in order to share fuses or divide them between several loads. Algorithms such as GE and CGM are more used for random networks-highly meshed.

According to this, let's consider that the vector l_c contains the indexes of the branches to be cut in order to make the system fully radial (in the case of the example of the Figure 26, either the branch 5 or 6 can be cut but only one is needed to), in the same order given by the rows of $\mathbf{\Gamma}$, i.e., the indexes in l_c must be corresponding to the positions of the branches in the incidence matrix.

Likewise, the vector l_{nc} saves the branches remaining in the system. From this, if one divides the Equation 11 keeping the cut and no-cut branches separated and ignoring the slack node (by convention the slack node is supposed to be in the first position of the vector $\boldsymbol{V}$, and it has to be ignored because at this point the current injected in that node is unknown), the Equation 14 is obtained (the adopted representation on the equations is corresponding to the used in MATLAB):

$$\boldsymbol{\Gamma^{T}(2\colon N, l_{nc}) \cdot I_{B}^{T}(l_{nc}) + \Gamma^{T}(2\colon N, l_{c}) \cdot I_{B}^{T}(l_{c}) + I_{d} \cdot I_{N}^{T} = 0}$$

Equation 14

Isolating the term of no-cut branches currents vector the Equation 15 is found:

$$\boldsymbol{I_{B}^{T}(l_{nc}) = (\Gamma^{T}(2\colon N, l_{nc}))^{-1} \cdot (-\Gamma^{T}(2\colon N, l_{c}) \cdot I_{B}^{T}(l_{c}) - I_{N}^{T})}$$

Equation 15

With the Equation 15 is possible to do the backward sweep if one knows the currents through the cut branches. If the vector $\boldsymbol{I_{B}^{T}(l_{nc})}$ is known then the nodal voltages are possible to be calculated: To get the equation to do so, from the Equation 12 is separated the slack node and only the no-cut branches are considered, as follows:

$$\boldsymbol{\Gamma(l_{nc}, 2\colon N) \cdot V^{T}(2\colon N) - R_{B}(l_{nc}, l_{nc}) \cdot I_{B}^{T}(l_{nc}) + \Gamma(\colon, 1) \cdot V(1) = 0} \rightarrow$$
$$\boldsymbol{V^{T}(2\colon N) = \left(\Gamma(l_{nc}, 2\colon N)\right)^{-1} \cdot \left(-\Gamma(\colon, 1) \cdot V(1) + R_{B}(l_{nc}, l_{nc}) \cdot I_{B}^{T}(l_{nc})\right)}$$

Equation 16

The Equation 16 represents the forward sweep. However, in order to perform the Back and Forward Sweep given by the Equation 15 and the Equation 16, one requires determining correctly the currents through the cut branches. With that aim an iterative process is proposed base in the work of [25]. The procedure consists in representing all the branches forming a loop as a set of dipoles connected to the radial network (after those branches are cut the system is radial). An example can be found in the Figure 27, the nodes where the cut branches are connected can be seen as a two port model. In this example let’s assume that the branch 6 is the one to disconnect, so in the image of the left the dipole representation is presented.

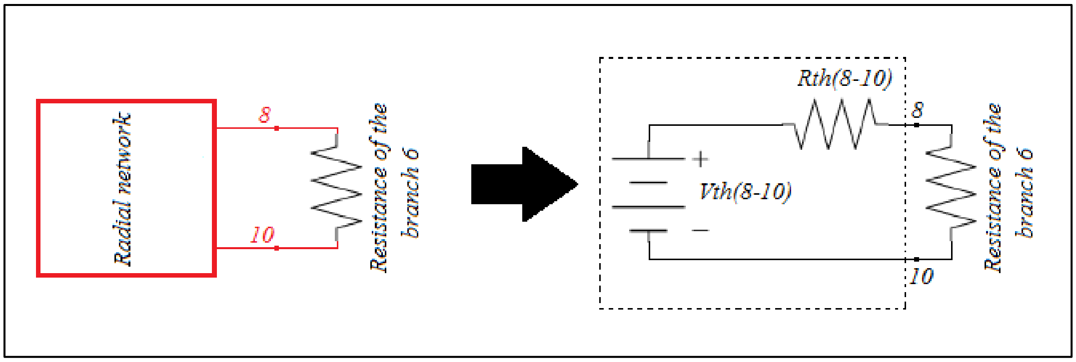


Figure 27: Dipole representation of the example of the Figure 24

In this sense, the image to the right in the Figure 27 is an equivalent transformation of the dipole representation. This transformation is carried out by means of the theory of Thevenin [30], because the entire block named as 'radial network' can be replaced by a Thevenin voltage source in series with a Thevenin resistance. The Thevenin voltage source is calculable by means of the Equation 16, but the Thevenin resistance is obtained by a sequential activation of unitary current sources connected between all the nodes where the cut branches are located. In the Figure 28 is presented a schematization of the procedure, in this example there is only one loop, but this is extensible for several loops. Since the current source has a value of 1 A, the obtained differential voltage between the nodes is equal to the Thevenin resistance of that branch.

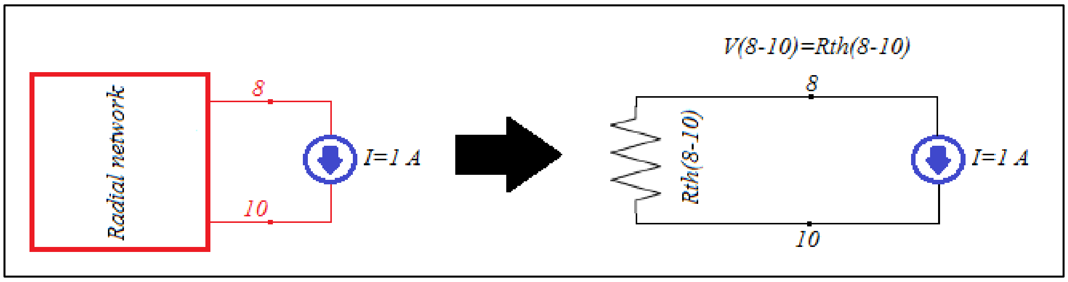


Figure 28: Strategy to determine the equivalent Thevenin resistance

According to the theory of circuits [30], to calculate the Thevenin resistances between two nodes, all the independent sources are shut down, this means the voltage sources are short circuited and the current sources are opened. For this, in the Equation 15, $\boldsymbol{I_N^T}$ is made all equal to zero and the result of $\boldsymbol{I_B^T(l_{nc})}$ is replaced in the Equation 16, where $\boldsymbol{V(1) = 0}$. The resulting expression is:

$$\boldsymbol{V^T(2:N) = -\left(\Gamma(l_{nc}, 2:N)\right)^{-1} \cdot R_B(l_{nc}, l_{nc}) \cdot \left(\Gamma^T(2:N, l_{nc})\right)^{-1} \cdot \Gamma^T(2:N, l_c) \cdot I_B^T(l_c)}$$

Equation 17

But the Equation 17 only gives the nodal voltages referenced to the same point; hence to obtain the differential voltages between nodes the Equation 18 is used:

$$R_B^{TH}(:,i) = \boldsymbol{\Gamma(l_c, 2:N)\ \cdot V^T(2:N)}$$

Equation 18

With the Equation 18 the Thevenin resistances between nodes of the cut branches are calculated. It was mentioned that there can be cases where more than one branch are cut to transform the entire system into a radial network, therefore the unitary current source $\boldsymbol{I_B^T(l_c)}$ are activated sequentially, i.e., the Thevenin resistances are calculated one by one.

Algorithm 7: Calculation Thevenin Resistance In Branches (CTRB)

```
Input: $\Gamma$, $R_B$, $l_c$ , $l_{nc}$
Output: $R_B^{TH}$
N = length_columns(Γ)
for i=1:1:length($l_c$) do
    $I_B^T = zeros$(length($l_c$): 1)
    $I_B^T(l_c(i)) = 1$
    $V^T(2:N)$ ← Equation 17
    $R_B^{TH}(:,i)$ ← Equation 18
end for
End
```

In the Algorithm 7 is presented the pseudo code for calculating the Thevenin resistance between the nodes of all the cut branches (branches forming a loop in the system). As it was explained before, fictitious unitary current sources are connected sequentially between the nodes corresponding to the particular cut branch; the Thevenin resistances for each cut branch are obtained step by step in a ‘for’ cycle, and finally in the diagonal matrix $\boldsymbol{R_B^{TH}}$ are stored the equivalent resistances (consequently this matrix is sized $l_c \times l_c$). This part of the algorithm is needed to be done at the beginning of the process only once, because the indexes $\boldsymbol{l_c}$ and $\boldsymbol{l_{nc}}$ are not supposed to change after the system is formed; therefore the inverses of the incidence matrix in the Equation 17 are required to determine only once. In the following the procedure for solving the system is explained.

When the system solver is started all the cut branches currents are considered equal to zero ($\boldsymbol{I_B^T}(\boldsymbol{l_c}) = \boldsymbol{0}$), in order to obtain in the first iteration the value of the nodal voltages through the Equation 15 and Equation 16. Once the first iteration is finished, the $\boldsymbol{I_B^T}(\boldsymbol{l_c})$ needs to be updated. This is done by means of adding incremental currents to the cut branches. See Equation 19.

$$\Delta \boldsymbol{I_B^T}(\boldsymbol{l_c}) = \left(R_B^{Total}\right)^{-1} \cdot \boldsymbol{\Gamma}(\boldsymbol{l_c}, :) \cdot \boldsymbol{V_0^T}$$

Equation 19

In the Equation 19, R_B^{Total} is a diagonal matrix where the total resistance per cut branch is stored: the Thevenin plus the branch resistance (See Figure 27) and $\boldsymbol{V_0^T}$ are the results of voltages using the Equation 16. This process is repeated until the voltages drops across the cut branches are corresponding to the results of the BFS, for this the Equation 20 is used:

$$\left| \boldsymbol{R_B}(\boldsymbol{l_c}, \boldsymbol{l_c}) \cdot \boldsymbol{I_B^T}(\boldsymbol{l_c}) - \boldsymbol{\Gamma}(\boldsymbol{l_c}, :) \cdot \boldsymbol{V^T} \right| < \boldsymbol{\varepsilon}$$

Equation 20

To understand better the whole procedure, the pseudo code of the MN-BFS method for constant current loads is showed in next:

Algorithm 8: Meshed Network Back and Forward Sweep (MN-BFS)

```
Input: $\Gamma$, $R_B$, $I_N^T$, $l_c$ , $l_{nc}$, $V_{slack}$
Output: $V^T$, $I_B^T$
1. N = length_columns($\Gamma$)
2. $I_B^T = zeros(\text{length\_rows}(\Gamma): 1)$
3. $V^T = zeros(\text{length\_columns}(\Gamma): 1$
4. $V^T(1) = V_{slack}$
5. $R_B^{TH} \leftarrow$ Algorithm 7
6. $R_B^{Total} = R_B^{TH} + R_B(l_c, l_c)$
7. for i=1:1:maximum_number_of_iterations do
8.   $\boldsymbol{I_B^T}(\boldsymbol{l_{nc}}) \leftarrow$ Equation 15        Backward Sweep
9.   $\boldsymbol{V^T}(\boldsymbol{2}:\boldsymbol{N}) \leftarrow$ Equation 16        Forward Sweep
10.  If Equation 20 go to 10. Otherwise 11
11.  break
12.  $\Delta\boldsymbol{I_B^T}(\boldsymbol{l_c}) \leftarrow$ Equation 19
13.  $\boldsymbol{I_B^T}(\boldsymbol{l_c}) = \Delta\boldsymbol{I_B^T}(\boldsymbol{l_c}) + \boldsymbol{I_B^T}(\boldsymbol{l_c})$
14. end for
15. End
```

In the Algorithm 8 is presented the pseudo code of the whole MN-BFS method for constant current loads. As it was said before, the Algorithm 7 (CTRB) is used at the beginning of the process. Inside of the 'for' cycle of the step number 7, the inverse operations of the incidence matrix and R_B^{Total} can be done before this cycle, avoiding an excessive computational consumption; this is possible since the physical information of the network does not change when the currents calculation process is carried out.

The MN-BFS method for constant current loads is an iterative process; however its number of iterations is relatively low because the values of currents through cut branches converge rapidly. For this reason this method is ideal for weakly meshed networks just like the distribution system of a vehicle. This method is easily extendable for no constant current loads (voltage dependent), because the Algorithm 8 can be part of a bigger algorithm which changes the load currents in function of the load voltages calculated in the previous iteration. Additionally, if ones wants to calculate exactly the currents through inner connections of ECU (the connections to ground), it would be possible to apply the concept of equivalent Thevenin circuits, instead of the constant current propagation considered in this approach (reasonable though).

This method has been tested for systems with total number of nodes in the order of 8000 [25], in a vehicle one can find up to 5000 nodes, which makes this method really appropriate for this application.

Other advantage is the simplicity for programming it according to the Algorithm 7 and Algorithm 8. In order to solve the on-board network distribution system in a vehicle, as it was explained before, the MN-BFS method is repeated for the two grids: positive and negative; finally to get the differential voltages at the terminals of the components both results are merged.

7.3. Solution of the power flow for on-board distribution networks in vehicles

The methodology for solving the on-board network distribution system in a vehicle has been explained; the Meshed Network Back and Forward Sweep (MN-BFS) algorithm is the method considered the most appropriated to be implemented in this application. In the Chapter VI was mentioned that the structure *All series sets* of the *dataContainer* contains all the possible paths between the positive and the negative battery terminal; therefore, the most natural first step for solving the system is to obtain the equivalent *All series sets* structure but according to the modules selected by the user. Once such structure is gotten, the immediate step is to obtain the inputs for the MN-BFS algorithm; these inputs are the incidence matrix (Γ), the diagonal matrix with branches resistances (R_B), the nodal injection currents (I_N) and the slack voltage V_{slack}. In fact, since there are two sub-systems forming the whole network those inputs are needed to be determined equally for both cases. When the inputs are correctly formed, the code of the Algorithm 8 is applied and the voltage drops are calculated.

In order to do the first step, the activation of modules, a function in MATLAB named *Filtering* has been designed; this function receives as input parameters the structure *dataContainer* and a char matrix one can name as *Fam*. The char structure *Fam* is projected to indicate per column the list of modules to select for one family as it can be seen in the Table 14: in the first row are deployed the families of a vehicle, in the second row the number of modules to active for the corresponding family (generally per family should be possible to pick up only one module, but there are special cases where several can be chosen), in the following rows the name of the modules to select per family are listed.

family01	**family02**	···
1	2	···
module01_family01	module01_family02	···
---	module02_family02	···
⋮	⋮	⋮

Table 14: Example of char data for activating modules ‘Fam’

The aim of the function *Filtering* is to get the matrices *Mctype*, *Mcid*, *com_modules* and *com_family*. The matrices *Mctype* and *Mcid* are basically the all series sets per type and ID according to the modules selected. The matrices *com_modules* and *com_family* are projected to store in the odd column positions, string identifiers with the name of the families and modules where the wire between those nodes belongs, and thus, being able to associate to a component’ terminals their currents.

Algorithm 9: Filtering

```
Input:  dataContainer, Fam
Output: Mctype, Mcid, com_modules, com_family
1.cont=0; Mctype= dataContainer.all_series_sets_types; Mcid= dataContainer.all_series_sets_ids)
2. for i=1:1:length_columns(Fam) do
3.   for j=1:1:length_rows(dataContainer.all_series_sets_ids) do
4.     for k=2:2:length_columns(dataContainer.all_series_sets_ids) do
5.       If dataContainer.all_series_sets_types(j,k)==2  go to 6. Otherwise to 16
6.       cont=cont+1
7.       for l=1:1:Fam{2,i} do
8.         A=strcmp(unique(locateInTable(dataContainer.WireModules,’wire_modules’,’and’,’wire_id’,dataContainer.all_series_sets_ids(j,k),’family’,Fam{1,i})), Fam{l+2,i})
9.         If A==1  go to 10. Otherwise to 15
```

```
10.           presence_matrix(j,k)=1
11.           com_modules{j,k-1}=Fam{l+2,i}
12.           com_modules{j,k+1}=Fam{l+2,i}
13.           com_family{j,k-1}=Fam{1,i}
14.           com_family{j,k+1}=Fam{1,i}
15.        end for
16.     end for
17.     check_array(j)=cont
18.     cont=0
19.  end for
20. end for
21. for j= length_rows(dataContainer.all_series_sets_ids):-1:1 do
22.    If sum(presence_matrix(j,:))<check_array(j)  go to 23. Otherwise to 24
23.    Mctype(j,:)=[]; Mcid(j,:)=[]; com_modules(j,:)=[]; com_family(j,:)=[];
24. end for
25. End
```

The Algorithm 9 shows the pseudo code for getting the aforementioned matrices *Mctype, Mcid*, *com_modules* and *com_family*. Through this logic a given path in a row can be formed by elements of different families or even by more than one module of the same family. It is based in eliminating paths according to the presence or not of all the wires in function of the desired modules expresses in the char *Fam*. All after the application of this process, it is ready to get the inputs for the MN-BFS based solver. These are: Γ_p, R_{B_p}, I_{N_p} , V_{N_p}, V_{slack_p} and Γ_g, R_{B_g}, I_{N_g} , V_{N_g}, V_{slack_g}. The first set of 4 inputs is the one representing the ‘positive grid’ and the second set the ‘negative grid’. In addition to this, it is required to get in a structure, information related to the ‘positive nodes’ and ‘negative nodes’ which are corresponding for being terminals of components where one is interested to find the differential voltage. The values of slack voltages are opened to set by the user; the ‘positive grid’ may have a value of 12 V, 14 V or 16 V depending which condition of the battery want to be tested, but of course the ‘negative grid’ is forced to have 0 V. Since the indexing of nodes available in the *dataContainer* is carried out taking into account all the modules together, the vectors V_{N_p} and V_{N_g} store the indexes of nodes according to the filtering done in the first stage and organized in the same order than the columns of the incidences matrices, in this way when the voltages are computed it is easy to link the results to the corresponding nodes. In order to get all the previous inputs a function in MATLAB called *Constructor* has been coded, which takes for itself the matrices *Mctype*, *Mcid*, *com_modules*, *com_family* and the structure *dataContainer* from the function *Filtering*, to process them and to get its outputs. The main key of the function *Constructor* is to go through the rows and columns of the matrices *Mctype* and *Mcid* taking into consideration that each row is a possible path between terminals of battery and each odd column is a node but each even column a connection. For this, the formation of Γ_p, R_{B_p}, I_{N_p} , V_{N_p} and Γ_g, R_{B_g},I_{N_g} ,V_{N_g} is done simultaneously when *Mctype* and *Mcid* are examined; if the codes detects a node not saved yet in V_{N_p} or V_{N_g} then it is added to it, and the resistance of the connection between two consecutive nodes as well to R_{B_p} or R_{B_g}, and finally to the incidences matrices. A very important point is to establish the interface node per row which determines the end of the 'positive grid' and the beginning of the 'negative grid'; this is the node of the last element found per row connected to ground. Additionally, the current injection vectors are formed following the next rules:

- If in a row there is only one element categorized as 'Consumer' through two nodes, say 1 and 2, then to the node 1 belonging to the 'positive grid' is assigned its current according to the modules selected (using the information contented in the matrices c*om_modules* and *com_family*), and to the node 2 belonging to the 'negative grid' is assigned the current of the node 1 divided over all its related output pins (ignoring the signals).
- If in a row there is only one element categorized as 'Consumer' through two nodes, say 1 and 3, since in the previous case all the current of the pin 1 was already assigned, then in here is added zero (0) amperes to this node; however since the wise-pair 1-3 is new, to the node 3 belonging to the 'negative grid' is assigned the current of the node 1 divided over all its related output pins (ignoring the signals). In this way is guaranteed that the currents in terminals of battery are the same.

- If in a row there is only one element categorized as ‘Consumer’ through two nodes, say 4 and 2, then to the node 4 belonging to the ‘positive grid’ is assigned its current according to the modules selected (using the information contented in the matrices c*om_modules* and *com_family*), and to the node 2 belonging to the ‘negative grid’ is added its previous current plus the current of the node 4 divided over all its related output pins (ignoring the signals).
- If in other row appears again the nodes 1-2 representing the same consumer, then to both current injection nodes are summed up zero (0) amperes because both nodes were already considered.
- If in the same row are found 2 or more elements categorized as consumers, then the minimum current of them is taken in order to assign it to the nodes were that current is going to be injected.
- If in a row there is only one element categorized as ‘Source’ through two nodes, say 5 and 6, then to the node 6 belonging to the ‘negative grid’ is assigned its current according to the modules selected (using the information contented in the matrices c*om_modules* and *com_family*), and to the node 5 belonging to the ‘positive grid’ is assigned the current of the node 6 divided over all its related input pins (ignoring the signals).
- All the previous cases explained for a ‘Consumer’ are the same for a ‘Source’, i.e., in the case of a ‘Consumer’ the current of a node not connected to ground (input) is assigned only once but the currents to the related output pins are summed up using the concept of accumulator; in the case of a ‘Source’ is in the other way around, the fixed injection current is given to the pin connected to ground and the input pins are accumulators of current.
- The code is able to correct a ‘black box’ tagged as ‘Consumer’ but being in fact a ‘Source’. This is likely to happen due to errors in the information coming from the

system. In this way the current distribution in the pins of a black box ‘Consumer’ is done following the procedure already explained for components type ‘Source’.

The Algorithm 10 shows the pseudo code for the function ‘Constructor’.

Algorithm 10: Constructor

```
Input: Mctype, Mcid, com_family, com_modules, dataContainer
Output: Vnp, Inp, Incp, Rbp, Vng, Ing, Incg, Rbg, Relationships
1. nodep=1; branchp=1; nodeg=1; branchg=1; inp=zeros(1,10000); ing=zeros(1,10000)
2. for i=1:1:length_rows(Mctype) do
3.   for j=2:2:length_columns (Mctype) do
4.     If  ~isnan(Mctype(i,j)) go to 4. Otherwise to 11
5.     Switch (Mctype (i,j))
6.       Case 7
7.         location=unique(locateInTable(dataContainer.Devices,’location’,’and’,’device_id’, Mctype (i,j)))
8.         type_vector{i,j}= DeviceType(dataContainer, location)
9.       Case 13
10.        type_vector{i,j}=’consumer’
11.      Otherwise
12.  end for
13.  number_consumer=0; number_source=0
14.  for k= length_columns(type_vector):-2:2 do
15.    If strcmp(type_vector{i,j},’Unclear’) go to 14. Otherwise to 15
16.    error(‘Unclear device in the position (i,j). Please check it out’)
17.    If (strcmp(type_vector{i,k},'consumer')) | ( strcmp(type_vector{i,k},'source'))  go to 16. Otherwise to 17
18.      stoppage_node_type=type_vector{i,k}; stoppage_node_pos=k+1
19.      If (strcmp(type_vector{i,k},'consumer'))  go to 18. Otherwise to 19
20.      number_consumer=number_consumer+1; consumers(k-1)=Mcid(i,k-1);
21.      If (strcmp(type_vector{i,k},'source'))  go to 20. Otherwise to 21
22.      number_source=number_source+1; source(k-1)=Mcid(i,k-1);
23.   end for
24.   [current_pos, current_neg] = get_currents(dataContainer,number_consumers,number_sources,consumers,stoppage_node_pos,stoppage_node_type,com_family,com_modules,i,Mcid,Mctype)
25.   actual_node=1
26.   for column=1.2:(stoppage_node_pos-4) do
27.     If nodep==1 go to 28. Otherwise to 29
28.     Vnp(nodep)=Mcid(i,nodep); nodep=nodep+1
29.     If nodep>1 go to 30. Otherwise to 54
30.     check=0; check2=0
31.     pos=find(Vnp==Mcid(i, column+2))
32.     If ~isempty(pos) go to 33. Otherwise to 48
33.     check=1
34.     for l=1.1:length_rows(Incp) do
```

```
35.    If (abs(Incp(l,actual_node))==1) && (abs(Incp(l,pos))==1) go to 36. Otherwise to 41
36.    check2=1
37.    If rrefp(1,l)==Mctype(i,column+1) && rrefp(2,l)~=Mcid(i,column+1) go to 38.
       Otherwise to 41
38.    resistance=get_resistance(Mctype(i,column+1),Mcid(i,column+1),dataContainer)
39.    rbp(l)=(rbp(l)*resistance)/(rbp(l)+resistance）
40.    actual_node=pos
41.   end for
42.   If check2==0 go to 43. Otherwise to 47
43.   Incp(branchp, actual_node)=1; Incp(branchp, pos)=-1
44.    Rbp(branchp)= get_resistance(Mctype(i,column+1),Mcid(i,column+1),dataContainer)
45.    rrefp(1,branchp)=Mctype(i,column+1); rrefp(2,branchp)=Mcid(i,column+1); branchp++
46.    actual_node=pos
47.    actual_node=pos
48.    If check==0 go to 49. Otherwise to 54
49.    Vnp(nodep)=Mcid(i,column+2)¸ Incp(branchp, actual_node)=1; Incp(branchp, nodep)=-1
50.    Rbp(branchp)= get_resistance(Mctype(i,column+1),Mcid(i,column+1),dataContainer)
51.    rrefp(1,branchp)=Mctype(i,column+1); rrefp(2,branchp)=Mcid(i,column+1); branchp++
52.    actual_node=nodep
53.    nodep++
54.    If column==(stoppage_node_pos-4) go to 55. Otherwise to 56
55.    Inp(actual_node)=Inp(actual_node)+current_pos
56.  end for
57.  actual_node_g=1
58.  for h=length_columns(Mctype):-2:(stoppage_node_pos) do
59.    If (nodeg==1) && (~isnan(Mcid(i,h)) ) go to 60. Otherwise to 63
60.    Vng(nodeg)=Mcid(i,h)
61.     nodeg=nodeg+1
62.     prim_g=Mcid(i,h)
63.     If (nodeg>1) && (~isnan(Mcid(i,h)) ) && (Mcid(i,h)~=prim_g) go to 64. Otherwise to
       88
64.     checkg=0; check2g=0
65.     posg=find(Vng== Mcid(i,h))
66.     If ~isempty(posg) go to 67. Otherwise to 82
67.     checkg=1
68.     for lg=1.1:length_rows(Incg) do
69.       If (abs(Incg(lg,actual_node_g))==1) && (abs(Incg(lg,posg))==1) go to 70. Otherwise
          to 75
70.       check2g=1
71.       If rrefg(1,lg)==Mctype(i,h+1) && rrefg(2,lg)~=Mcid(i,h+1) go to 72. Otherwise to 75
72.       resistanceg=get_resistance(Mctype(i,h+1),Mcid(i,h+1),dataContainer)
73.       rbg(lg)=(rbg(lg)*resistanceg)/(rbg(lg)+resistanceg）
74.       actual_node_g=posg
75.    end for
76.     If check2g=0 go to 77. Otherwise to 81
77.     Incg(branchg, actual_node_g)=1; Incg(branchg, posg)=-1
78.     Rbg(branchg)= get_resistance(Mctype(i,h+1),Mcid(i,h+1),dataContainer)
79.     rrefg(1,branchg)=Mctype(i,h+1); rrefg(2,branchg)=Mcid(i,h+1); branchg++
80.     actual_node_g=posg
81.     actual_node_g=posg
82.     If checkg==0 go to 83. Otherwise to 88
```

```
83.       Vng(nodeg)=Mcid(i,h)¸ Incg(branchg, actual_node_g)=1; Incg(branchg, nodeg)=-1
84.       Rbg(branchg)= get_resistance(Mctype(i,h+1),Mcid(i,h+1),dataContainer)
85.       rrefg(1,branchg)=Mctype(i,h+1); rrefg(2,branchg)=Mcid(i,h+1); branchg++
86.       actual_node_g=nodeg
87.       nodeg++
88.       If h== stoppage_node_pos go to 89. Otherwise to 90
89.       Ing(actual_node_g)=Ing(actual_node_g)+current_neg
90.     end for
91.   end for
92. relationships=current_nodes
93. inp(length(vnp)+1:length(inp))=[];
94. ing(length(vng)+1:length(ing))=[];
95. rbg=diag(rbg);
96. rbp=diag(rbp);
97. Fix if two or more black box consumers are found in a same row
98. End
```

The pseudo-code shown in the Algorithm 10 allows getting all the inputs for the solver: Γ_{p} (in the code named as Incp), R_{B_p}, I_{N_p} , V_{N_p} in the ‘positive grid’ and Γ_{g} (in the code named as Incg), R_{B_g}, I_{N_g} , V_{N_g} in the ‘negative grid’, and the information about the terminals of the components where one needs to be interested in finding its differential voltage (between the inputs pins and output-grounded pins) given by the matrix $Relationships$. This code is basically divided in the next stages: classification of the elements as consumer or source (lines 3 to 12), determination of the stoppage node –the interface between the positive and the negative grids –, number of consumer and sources in a same row (lines 13 to 23), calculation of the currents to inject in the stoppage node and neighbor (line 24, where a function named ‘get currents’ is being used which gets the currents according to the requirements already explained), forming the matrices Γ_{p}, R_{B_p}, I_{N_p} , V_{N_p} where the resistance of the branches for wires is calculated taking into account the effects of the temperature according to the Equation 1 (lines 25 to 56) and forming the matrices Γ_{g}, R_{B_g}, I_{N_g} , V_{N_g} similarly to what is done for the ‘positive grid’ (lines 57-90). All the previous processes are inside of a big ‘for’ cycle going row by row of the *Mctype* and *Mcid* matrices. Finally, between the lines 92 to 96 the matrices are arranged. In the line 97 is executed a process where the current injection vectors are fixed up in case there is no balance between the currents of terminals in the battery; this may happen if there is a ‘Source’ element wrongly classified as ‘Consumer’ due to errors in the information coming from factory or in case of two consumers black boxes in series.

At this point almost everything is served to complete the algorithms for simulating power flow in a vehicle. One last sub-process required to do before the implementation of the Algorithm 8 is the detection of the indexes of branches which define loops in both positive and negative grid. To understand how to do it, one needs to take into consideration that the Algorithm 10 forms the incidence matrix from the *all series sets* matrices, which are gotten following the direction of the current from the positive battery terminal and the negative terminal of the same, this allows obtaining something like the presented in the Table 15; in this example the *Node K* has for three different lines three values of -1 (*branch 2*, *branch L-1* and *branch L*). When a system is completely radial, a new line added to that system causes the addition of a new node, however for meshed networks, as in this example, when the *branch L-1* is added the *Node K* was already added for the *branch 2*, and also when the *branch L* is added again the incidence on the *Node K* is repeated. From this and by means of the Table 15, one can detect easily when there are loops: A node having associated more than one value of -1 is receiving lines which provoke cycles or loops in the system. Obviously only two out of the three lines related to the *Node K* have to be indexed in the vector $\mathbf{l_c}$ and the remaining one must be kept in $\mathbf{l_{nc}}$. This concept can be applied because as it was discussed before, as main input the matrices with *all the series sets* is used and the Algorithm 10 is conceived to go through those matrices in that order and logic.

	Node 1	***Node 2***	⋯	***Node K***	⋯	***Node N***
Branch 1	1	-1	0	0	0	0
Branch 2	1	0	0	-1	0	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮
Branch L-1	0	1	0	-1	0	0
Branch L	0	0	1	-1	0	0

Table 15: Example of an Incidence Matrix

Algorithm 11: Detection of Loops (DOL)

```
Input: $\Gamma$
Output: $l_c$ , $l_{nc}$
1. pos_cont _loops=1
2. for i=1:1:length_columns($\Gamma$) do
3.   loops=-1
4.   for j=1:1:length_rows($\Gamma$) do
5.     If $\Gamma_{\mathrm{p}}(j,i)$ == -1 go to 6. Otherwise to 10
6.     loops=loops+1
7.     If loops>=1 go to 8. Otherwise to 10
8.     pos_loops(pos_cont _loops)=j
9.     pos_cont _loops++
10.  end for
11. end for
12. $l_c$=unique(pos_loops)
13. pos_cont _no_loops=1
14. for i=1:1:length_rows($\Gamma$) do
15.   If ~ismember(i, $l_c$) go to 16. Otherwise to 18
16.   $l_{nc}$(pos_cont _no_loops)=i
17.   pos_cont _no_loops++
18. end for
19. End
```

Finally, in the Algorithm 12 is presented the whole process of the power flow solver for the on-board network distribution system in modern vehicles considering constant current consumptions. The Algorithm 11 is used for both the positive and negative grid, just like the Algorithm 8 to solve independently both subsystems; lastly both results are merged and the voltages at terminals of the devices are obtained.

Algorithm 12: Power Flow Solver of the On-board Network Distribution System In Modern Vehicles considering Constant Current Consumption (PFS-ONDSV-CCC)

```
Input: dataContainer, Fam
Output: $V^T$, $V^T_{diff}$, $I^T_B$
1. Mctype, Mcid, com_modules, com_family ← Algorithm 9
2. Vnp, Inp, Incp ($\mathbf{\Gamma_p}$), Rbp, Vng, Ing ($\mathbf{\Gamma_g}$), Incg, Rbg, Relationships ← Algorithm 10
3. $\mathbf{l_{cp}}$ , $\mathbf{l_{ncp}}$ ← Algorithm 11
4. $\mathbf{V^T_p}, \mathbf{I^T_{B_p}}$ ← Algorithm 8
5. $\mathbf{V^T_g}, \mathbf{I^T_{B_g}}$ ← Algorithm 8
6. $\mathbf{V^T} = \left[\mathbf{V^T_p}; \mathbf{V^T_g}\right]$
7. $\mathbf{I^T_B} = \left[\mathbf{I^T_{B_p}}; \mathbf{I^T_{B_g}}\right]$
```

```
8. $\mathbf{V}_{\mathbf{diff}}^{\mathbf{T}}$ ← Getting the differential voltages at terminals of components using the matrix Relationships where is saved all the input pins – out put (grounded) pins pair-wise relations
9. End
```

In the Appendix 1 to the Appendix 5 are displayed some examples of electrical schematics from a distribution system vehicle's section. This example is defined by two functional families and 6 modules in total. After the pre-processing stage in MATLAB (by means of the associated BOM and Wire lists files of each drawing) the *dataContainer* structure is obtained and afterwards by applying the Algorithm 12 the power flow simulations is run and voltages drops are gotten. In the Appendix 19, Appendix 20 and Appendix 21 the results for the system - when the module01 and module_fam02_03 are selected – can be found; the Appendix 21 displays the differential voltages at components terminals, where it is possible to see an assessment of that result according to the criteria already explained in the Chapter IV. With the module02 and module_fam02_03 picked up the results are available in the Appendix 22, Appendix 23 and Appendix 24. Finally, the most critical topology of the system can be formed by activating the module03 and module_fam02_03: there are more components connected and therefore at terminals of the one located at *vws.07.1* one can detect a critical value of voltage (Appendix 27) and hence the electrical designer should consider the increase of the wire's cross-section. It is important to remark that this results neglect the connection of signal cables for communication purposes, since their power is considerably low.

With this is realizable how dynamic the results are in function of the vehicle's modularity: In this example there are 6 modules from only two functional families, therefore the total number of combinations of modules ascend to 9; according to the Table 1 with the modular system approach a vehicle can have up to 10^{10} possible combinations considering its overall design. The full simulation of a vehicle is under process; specifically the preparation of the QT file is on progression.

CHAPTER VIII

CONCLUSIONS AND FUTURE DEVELOPMENTS

The wiring system design in the on-board distribution network of modern vehicles counts with a variety of techniques currently implemented differently by the vehicle manufacturers throughout the world. In the Volkswagen Group, where SEAT is belonging, the adopted technique is the modular wiring approach, which allows getting a very high number of combinations (in the order of $\sim 10^{10}$), this means more variety offered the customers, also the unit cost of parts is optimized along with the pre-allocation of elements; however this causes a higher complexity in the network development, assembling and managing the spare parts system. The important concepts of modules and functional families have been explained regarding their impact on the design and construction of wiring harnesses. Additionally, the VOBES process which defines the methodologies, strategies, frameworks and sub-processes in the vehicle design with certain software tools haven been described. As a result, it has been established the great importance in calculating accurately the voltages drops at component’s terminals in order to comply with the regulations and to ensure the correct operation of the whole system, taking into consideration current researches pointing out the optimization of the wirings and the need of dynamic-recurrent simulation to validate the new strategies for sizing from a point of view of stable power flows.

It has been defined the set of elements that can be found in the on-board network of a vehicle, between others, one can highlight as very important the Electronic Control Units (ECUs) and general loads. Both are treated as black boxes in the electrical and electronic department and a correct identification of them is basic in order to perform power flow simulations; this is done by means of the identification of standard pin tags used in the Volkswagen Group. In this work the inputs for an external power flow solver-visualization tool has been standardly proposed, basically dividing the full system as a set of elements for connections and loads: as connectors one should include ECUs and other logic elements such as wires, switches, devices, etc, and as loads all those equipment which are expected to inject currents into the system or generate voltage drops: in general black boxes with information of pin currents. The dynamic behavior of ECUs is neglected, considering them just as a point of distribution of current. To start-up the external power flow tool the elaborated code carried out in this work, takes into account that none constant current loads can be connected in series, instead of that it establishes in such cases, some loads as constant impedance thus to guarantee the convergence of the external tool. The final aim to obtain with the external power flow solver is to be able of simulating power flow with much flexibility, such as the option to set manually the type of a load (constant current, constant power and constant impedance), to switch wires, turn off loads, etc., by looking in a friendly user interface the results and thus getting results easily interpretable.

The Meshed Networks Back and Forward Sweep method (MN-BFS) newly developed to be applied in train distribution systems has been considered as the best option to perform power flows solutions in the network of a vehicle: it is an almost-straight forward (direct) mathematical approach which for constant currents, is only iterative to solve the cycles or loops that the system may have (they are low compared to the total amount of branches). This an approach easily programmable and fits properly with the characteristics of this kind of systems: since the matrix of all sets between the battery' terminals is available, is simple to get the incidence matrices and the other parameters, additionally it allows performing solutions considering constant power and constant impedance loads that may be set throughout the system. Other methods such as the Conjugate Gradient are slower when the boundary conditions are changed (no linear) and also requires more number of iterations (equal to the order of the system, i.e., number of nodes), and in the case of the Gaussian Elimination are ideal for small-medium size systems which is not the case for the on-board network which can have around 5000 nodes. The MN-BFS is not a computational-costly method since the matrix inversions required to calculate are all needed to do only once and the matrices are highly sparse, representing a low consumption of memory and time. This method is ideal for radial systems and its high speed is appropriate for future new developments where probabilistic power flows are required to do and the speed starts being a key factor in the solution. The comparisons between methods have been carried out only qualitatively, but in future researches might be interesting to elaborate quantitative comparisons.

From the development of this static tool, new improvements can be added such as dynamic simulations, the study of use of smart fuses, and Hardware-In-the-Loop (HIL) systems to model dynamically the vehicle and thus to evaluate the impact on the electrical distribution systems of variables like driving profile, climate, external conditions on the road, etc. All is aiming the optimization of the sizing of the system and the increase on monitoring of the system to improve the experience of the driver and acquire more benefits for the manufacturers.

It is recommended to validate the static simulator with real tests, and also make load characterizations with experimentations to define the type of loads , trying to go beyond the approach of black boxes commonly adopted in the electrical and electronic department when sizing the wires, fuses and in general the distribution system.

In this work has been explained the methodologies pointing out the tackling to solve this kind of electrical systems. The very high number of modules and the organization of manufacturing data are the main complex aspects for simulating statics and dynamics behaviors in vehicles, because all the information is generated from a point of view of construction but is not systematically available and organized to perform this kind of analysis; even sometimes, the manufacturing data are contradictories and some information is lacking, because the VOBES process points out to standardize the vehicle design, however it is not based in a full-model-based software approach, but as a set of tools driven by engineers which may conduce to problems in the updating of new designs, errors in the architecture definition, misinformation, lack of integration, etc.

However, with the methodology proposed in this work is optimistic to project the possibility to carry out a high gamma of simulations on the on-board network; from the point of view of power flow to determine voltages drops, the method based in the Meshed Network Back and Forward Sweep (MN-BFS) has been showed to be perfectly applicable, adaptable, and qualitatively more robust, faster and efficient than other alternatives available in the state of the art for similar systems. The simulation on the wiring system of vehicles is a novelty research, because currently all the manufacturers throughout the world make estimations of voltages drops and power based in fast and manual calculations, which may conduce to oversizing of wires at some points. In this work the bases for future more complex, complete and deeper analysis of the electrical system in a vehicle has been settled.

CHAPTER IX

QUALITY REPORT

SEAT is concerned deeply about the confidentiality, therefore in some occasions was difficult or even impossible to access to specific information and to software which have licenses designated exclusively for plant employees of the company. Additionally, the speed for resolution of internal agreements for completing the negotiation of the design of the power flow software tool with the provider was not as fast as expected, causing the impossibility of finishing fully the project at the end of the internships, according to the terms offered at the beginning of it, such as the programming of the external solver which is projected to do the power flow calculation and the dynamic visualization of the system. On the other hand, it has been found the VOBES system lacking of a robust integration of the all their softwares, so in some cases there was data missed and/or generating conflicts.

The QT file has been put forward as an important document for organizing the information and thus being able to simulate the on-board network of vehicles in SEAT. In order to improve for future works the elaboration and precision of the content of this document it is recommended to formalize the QT as a binding and relevant documentation, standardize its content for all projects SEAT and introduce it mandatorily in the specification of wiring to be delivered along with the plans made in the VOBES process for a vehicle model.

REFERENCES

[1] Rius, R., Armand. *A novel optimization methodology of modular wiring harnesses in modern vehicles: weight reduction and safe operation.* Universitat Politécnica de Catalunya, 2015.

[2] Frei, S., Diebig, M. *Simulation-based optimization of multi voltage automotive power supply systems*. Vehicle power and propulsion conference (VPPC), IEEE. Pages 1-6. 15-18 Oct. 2013

[3] Diebig, M., Frei, S., *Modeling of the automotive power supply network with VHDL-AMS*. Vehicle power and propulsion conference (VPPC), IEEE. Pages 1-3. Sept. 2010

[4] Čiegis, R., Ilgevičius, A., Ließ, H., Meilūnas, M., Suboč, O. *Numerical simulation of the heat conduction in the electrical wires. Mathematical Modelling and analysis* Vol. 12 No. 4. pages 425-439. 2007

[5] http://delphi.com/ images/news/2011/delphi-wiring-harness.jpg. Link 2014

[6] ISO 8820-1 – *International standard for road vehicles-Part I*. 2014

[7] VW75212 – *Volkswagen group internal regulation for sizing fuses and wires*

[8] VW80000 - *Volkswagen group internal regulation for road vehicles*

[9] AUDI. *Load Analyse Prufkatalog*. Electric and Electronic department of AUDI, Page 8, 2010.

[10] https://servicenet.t-systems.com/t-systems-plm-en/vobes. Link 2016

[11] MATLAB online help web http://es.mathworks.com/help/matlab/?requestedDomain=www.mathworks.com

[12] Datasheet *5Q0_614_517_Ab* extracted from *Sys42*

[13] Datasheet *4H0_951_253_C* extracted from *Sys42*

[14] IEEE. *Recommended Practice for Software Requirements Specifications*. Software Engineering Standards Committee, IEEE Computer Society. New York, 20

[15] Kasap, S. O. *Principles of Electronic Materials and Devices (Third edition.)*. Mc-Graw Hill. p. 126. 2006

[16] Robba, E. João. *Introdução a Sistemas Elétricos de Potencia*. Universidade de São Paulo, 2011.

[17] Arendt Knudsen, H, et al. *Robust algorithm for random resistor networks using hierarchical domain structure*. Journal of Computational Physics 211. 2005

[18] B. Derrida, J.G. Zabolitzky, J. Vannimenus. *A transfer matrix program to calculate the conductivity of random resistor networks*. J.Stat. Phys. 36. Pages 32-42. 1984

[19] Grainger John, Stevenson William D. *Electrical power system analyses*. Book from McGRAW-HILL. Pages 350-400- 2001

[20] E. Aker, K.J. Maloy, A. Hansen, G.G. Batrouni. *A two dimensional network simulator for two-phase flow in porous media*. Transport Porous Med. 32, 1998.

[21] G.G. Batrouni, A. Hansen. *Fracture in three-dimensional fuse networks*. Phys. Rev. Lett. 80. Pages 325-328. 2011

[22] http://cstheory.stackexchange.com/questions/19449/algorithm-for-max-network-flow-with-lower-bounds-and-its-complexity. Link 2016.

[23] Massachusetts Institute University class notes. *Graphs, networks and incidence matrices. Linear Algebra*. MIT productions. 2015.

[24] J.A. Michline Rupa, S. Ganesh. *Power flow Analysis for Radial Distribution System using Backward/Forward Sweep Method*. International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering Vol. 8, No 10, 2014.

[25] Arboleya Pablo, Mohamed Bassam, González-Morán Cristina, El-Sayed Islam. *BFS Algorithm for Voltage-Constrained Meshed DC Traction Networks With Nonsmooth Voltage-Dependent Loads and Generators*. IEEE Transactions of Power Systems. 2015

[26] P. William H, Saul A. Teukolsky, William T. Vetterling. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University England. 1992

[27] Wexler Alvin. *Large Engineering Systems*. International Symposium of EPS. 1976

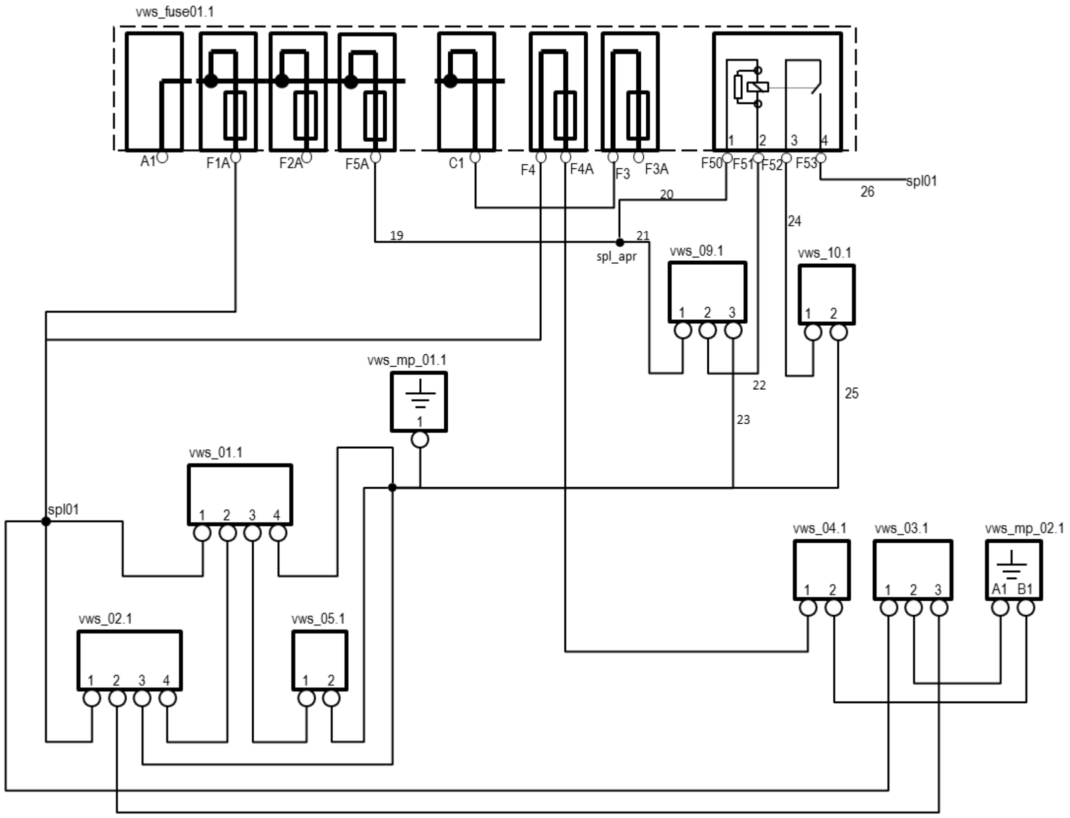
[28] Richard Jonathan. *An introduction to the Conjugate Gradient Method without the Agonizing Pain*. School of Computer Science Carnegie Mellon University. 1994

[29] D. Braunagel, L. Kraft, and J. Whysong. *Inclusion of dc converter and transmission equations directly in a newton power flow*. IEEE Trans. Power App. Syst., vol. PAS-95, no. 1, pp. 76–88, Jan. 1976.

[30] Sadiku. N.O. Matthew, Alexander K. Charles. *Fundamentals of Electric Circuits*. Fifth Edition. Pages 340-360. 2013

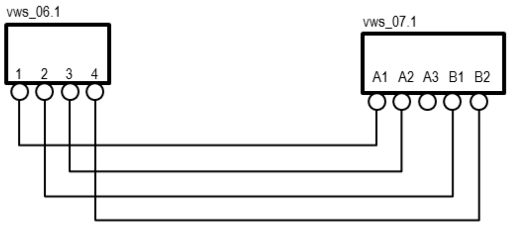
APPENDICES

Appendix 1. Electrical schematic harness 01



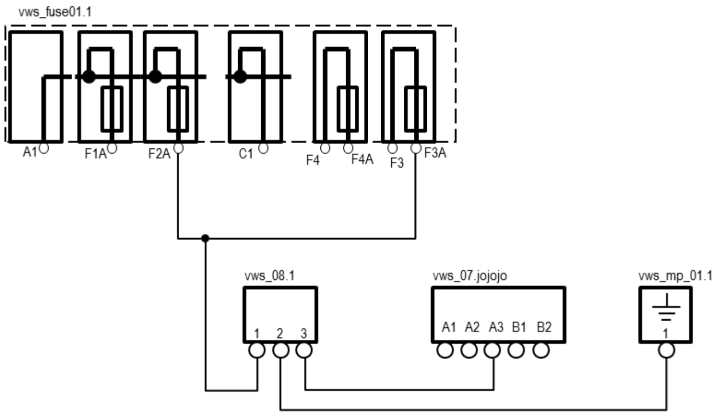
drawing01

Appendix 2. Electrical schematic harness 02



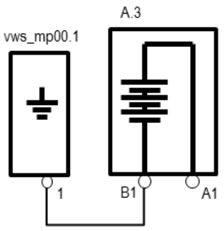
drawing02

Appendix 3. Electrical schematic harness 03



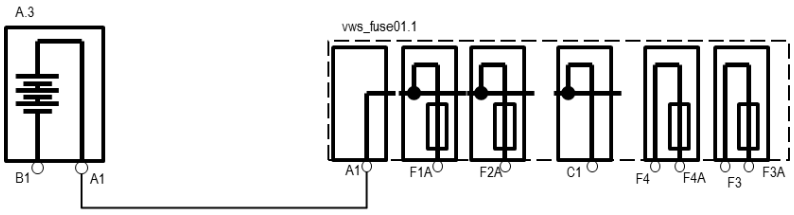
drawing03

Appendix 4. Electrical schematic harness 04



drawing_bat_negative

Appendix 5. Electrical schematic harness 05



drawing_bat_positive

In the appendices 1, 2, 3, 4 and 5 are displayed electrical schematics of wiring harnesses for an example of a part of an on-board distribution network in a vehicle. This example constitutes 5 wiring harness defined by 2 functional families (family01 and family02) and 5 modules in total: module01, module02, module03 (this three belonging to the family01), module_fam02_01, module_fam02_02 and module_fam02_03 (this last three are from the family02). For each of the harnesses there are associated corresponding BOM Excel files and Wire lists Excel file, and for the whole system a QT Excel file where is summarized and available critical information for being able to simulate. Along with the previous files, a master table of the part numbers of wires is also available as an input.

Appendix 6. Master nodes table

node_id	type	node_id0	location	slot	pin
1	fusebox_node	vws_fuse01.11A	vws_fuse01.1	A	1
2	positive_battery_terminal	A.31A	A.3	A	1
3	ground_node	vws_mp_00.11	vws_mp_00.1		1
4	negative_battery_terminal	A.31B	A.3	B	1
5	fusebox_node	vws_fuse01.11AF	vws_fuse01.1	F	1A
6	splice	spl011	spl01		1
7	device_node	vws_01.11	vws_01.1		1
8	coupling_node	vws_02.11	vws_02.1		1
9	device_node	vws_03.11	vws_03.1		1
10	fusebox_node	vws_fuse01.14F	vws_fuse01.1	F	4
11	device_node	vws_01.12	vws_01.1		2
12	coupling_node	vws_02.14	vws_02.1		4
13	coupling_node	vws_02.12	vws_02.1		2
14	device_node	vws_03.13	vws_03.1		3

node_id	type	node_id0	location	slot	pin
15	device_node	vws_03.12	vws_03.1		2
16	ground_node	vws_mp_02.11A	vws_mp_02.1	A	1
17	fusebox_node	vws_fuse01.14AF	vws_fuse01.1	F	4A
18	device_node	vws_04.11	vws_04.1		1
19	device_node	vws_04.12	vws_04.1		2
20	ground_node	vws_mp_02.11B	vws_mp_02.1	B	1
21	device_node	vws_01.13	vws_01.1		3
22	device_node	vws_05.11	vws_05.1		1
23	device_node	vws_01.14	vws_01.1		4
24	splice	SPM011	SPM01		1
25	coupling_node	vws_02.13	vws_02.1		3
26	device_node	vws_05.12	vws_05.1		2
27	ground_node	vws_mp_01.11	vws_mp_01.1		1
28	fusebox_node	vws_fuse01.13F	vws_fuse01.1	F	3
29	fusebox_node	vws_fuse01.11C	vws_fuse01.1	C	1
30	fusebox_node	vws_fuse01.15AF	vws_fuse01.1	F	5A
31	splice	spl_apr1	spl_apr		1
32	fusebox_node	vws_fuse01.150F	vws_fuse01.1	F	50
33	device_node	vws_09.11	vws_09.1		1
34	device_node	vws_09.12	vws_09.1		2
35	fusebox_node	vws_fuse01.151F	vws_fuse01.1	F	51
36	device_node	vws_09.13	vws_09.1		3
37	fusebox_node	vws_fuse01.153F	vws_fuse01.1	F	53
38	device_node	vws_10.11	vws_10.1		1
39	fusebox_node	vws_fuse01.152F	vws_fuse01.1	F	52
40	device_node	vws_10.12	vws_10.1		2
41	coupling_node	vws_06.11	vws_06.1		1
42	device_node	vws_07.11A	vws_07.1	A	1
43	coupling_node	vws_06.12	vws_06.1		2
44	device_node	vws_07.11B	vws_07.1	B	1
45	coupling_node	vws_06.13	vws_06.1		3
46	device_node	vws_07.12A	vws_07.1	A	2
47	coupling_node	vws_06.14	vws_06.1		4
48	device_node	vws_07.12B	vws_07.1	B	2
49	fusebox_node	vws_fuse01.12AF	vws_fuse01.1	F	2A
50	splice	sp31	sp3		1
51	device_node	vws_08.11	vws_08.1		1
52	device_node	vws_08.12	vws_08.1		2
53	ground_node	vws_mp_03.11	vws_mp_03.1		1
54	device_node	vws_08.13	vws_08.1		3
55	device_node	vws_07.13A	vws_07.1	A	3
56	fusebox_node	vws_fuse01.13AF	vws_fuse01.1	F	3A

Appendix 7. Master wires table

wire_id	partnumber	wire_id	partnumber
1	wireba1	**21**	wire00
2	wireba2	**22**	wire01
3	wireba3	**23**	wire01
4	wireba1	**24**	wire01
5	wireba2	**25**	wire01
6	wireba3	**26**	wire_prim
7	wire01	**27**	wire_prim
8	wire01	**28**	wire_prim
9	wire01	**29**	wire02
10	wire01	**30**	wire02
11	wire01	**31**	wire02
12	wire_prim	**32**	wire01
13	wire_prim	**33**	wire01
14	wire01	**34**	wire_prim
15	wire02	**35**	wire_prim
16	wire02	**36**	wire03
17	wire02	**37**	wire03
18	wire01	**38**	wire03
19	wire01	**39**	wire_prim
20	wire02	**40**	wire03

Appendix 8. Master modules table

Numeration	Modules	Family
6	mod_fam02_01	family02
7	mod_fam02_02	family02
8	mod_fam02_03	family02
9	module01	family01
10	module02	family01
11	module03	family01

Appendix 9. Master harness table

Harness	Numeration
drawing01	1
drawing02	2
drawing03	3
drawing_bat_negative	4
drawing_bat_positive	5

Appendix 10. Master fuses table

connection_id	location	slot_1	pin_1	part_slot_1	part_pin_1	connection_id0	connection_type	slot_2	pin_2	part_slot_2	part_pin_2
1	vws_fuse01.1	A	1			F1	fuse	F	1A		
2	vws_fuse01.1	A	1			F2	fuse	F	2A		
3	vws_fuse01.1	A	1			kl30..C1	plate	C	1		
4	vws_fuse01.1	F	4			F4	fuse	F	4A		
5	vws_fuse01.1	A	1			F5	fuse	F	5A		
6	vws_fuse01.1	F	3			F3	fuse	F	3A		
7	vws_fuse01.1	F	50		1	R1_coil	relay_coil	F	51		2
8	vws_fuse01.1	F	52		3	R1_switch	relay_switch	F	53		4

Appendix 11. Master coupling table

coupling_id	location_1	location_2
1	vws_02.1	vws_06.1

Appendix 12. Master coupling modularity table

location_1	modules_1	partnumbers_1	location_2	modules_2	partnumbers_2
vws_02.1	module01	pn031	vws_06.1	module01	pn081
vws_02.1	module02	pn032	vws_06.1	module02	pn082
vws_02.1	module03	pn033	vws_06.1	module03	pn083

Appendix 13. Master ECU table

device_id	location
1	vws_01.1

Appendix 14. Master ECU modularity table

location	module	partnumber
vws_01.1	module02	pn021
vws_01.1	module03	pn022

Appendix 15. Master connections table output

General_ID	Source	Source	Destination	Destination	Type	Type	SEAT_ID	Module	Module	Harness	Harness	Specific Res	Max. Sp. Res	Length	Max. Curr.	Min. Curr.	Nom. Curr.	State
1	2	A.31A	1	vws_fuse01.11A	2	wire	1	9	module01	5	drawing_bat_positive	0,0048	0,0056	0,8000	0	0	0	1
2	1	vws_fuse01.11A	5	vws_fuse01.11AF	9	fuse	1	9	module01	0	no_one	0,0000	0,0000	0,0000	0	0	0	1
3	1	vws_fuse01.11A	5	vws_fuse01.11AF	9	fuse	1	10	module02	0	no_one	0,0000	0,0000	0,0000	0	0	0	1
4	1	vws_fuse01.11A	5	vws_fuse01.11AF	9	fuse	1	11	module03	0	no_one	0,0000	0,0000	0,0000	0	0	0	1
5	5	vws_fuse01.11AF	6	spl011	2	wire	7	9	module01	1	drawing01	0,0127	0,0149	0,2880	0	0	0	1
6	6	spl011	8	vws_02.11	2	wire	9	9	module01	1	drawing01	0,0127	0,0149	0,7055	0	0	0	1
7	6	spl011	8	vws_02.11	2	wire	9	10	module02	1	drawing01	0,0127	0,0149	0,7055	0	0	0	1
8	6	spl011	8	vws_02.11	2	wire	9	11	module03	1	drawing01	0,0127	0,0149	0,7055	0	0	0	1
9	8	vws_02.11	41	vws_06.11	8	coupling	1	9	module01	0	no_one	0,0000	0,0000	0,0000	0	0	0	1
10	8	vws_02.11	41	vws_06.11	8	coupling	1	10	module02	0	no_one	0,0000	0,0000	0,0000	0	0	0	1
11	8	vws_02.11	41	vws_06.11	8	coupling	1	11	module03	0	no_one	0,0000	0,0000	0,0000	0	0	0	1
12	41	vws_06.11	42	vws_07.11A	2	wire	32	9	module01	2	drawing02	0,0127	0,0149	0,1566	0	0	0	1
13	41	vws_06.11	42	vws_07.11A	2	wire	32	10	module02	2	drawing02	0,0127	0,0149	0,1566	0	0	0	1
14	41	vws_06.11	42	vws_07.11A	2	wire	32	11	module03	2	drawing02	0,0127	0,0149	0,1566	0	0	0	1
15	46	vws_07.12A	45	vws_06.13	2	wire	34	9	module01	2	drawing02	0,1467	0,1719	0,3590	0	0	0	1
16	46	vws_07.12A	45	vws_06.13	2	wire	34	10	module02	2	drawing02	0,1467	0,1719	0,3590	0	0	0	1
17	46	vws_07.12A	45	vws_06.13	2	wire	34	11	module03	2	drawing02	0,1467	0,1719	0,3590	0	0	0	1
18	45	vws_06.13	25	vws_02.13	8	coupling	1	9	module01	0	no_one	0,0000	0,0000	0,0000	0	0	0	1
19	45	vws_06.13	25	vws_02.13	8	coupling	1	10	module02	0	no_one	0,0000	0,0000	0,0000	0	0	0	1
20	45	vws_06.13	25	vws_02.13	8	coupling	1	11	module03	0	no_one	0,0000	0,0000	0,0000	0	0	0	1
21	25	vws_02.13	24	SPM011	2	wire	19	9	module01	1	drawing01	0,0127	0,0149	0,2588	0	0	0	1
22	25	vws_02.13	24	SPM011	2	wire	19	10	module02	1	drawing01	0,0127	0,0149	0,2588	0	0	0	1
23	25	vws_02.13	24	SPM011	2	wire	19	11	module03	1	drawing01	0,0127	0,0149	0,2588	0	0	0	1
24	24	SPM011	27	vws_mp_01.11	2	wire	22	9	module01	1	drawing01	0,0127	0,0149	0,7226	0	0	0	1
25	24	SPM011	27	vws_mp_01.11	2	wire	22	10	module02	1	drawing01	0,0127	0,0149	0,7226	0	0	0	1
26	24	SPM011	27	vws_mp_01.11	2	wire	22	11	module03	1	drawing01	0,0127	0,0149	0,7226	0	0	0	1
27	27	vws_mp_01.11	3	vws_mp_00.11	15	body	1	0	no_one	0	no_one	0,0000	0,0000	0,0000	0	0	0	1
28	3	vws_mp_00.11	4	A.31B	2	wire	4	9	module01	4	drawing_bat_negative	0,0048	0,0056	0,8000	0	0	0	1
29	2	A.31A	1	vws_fuse01.11A	2	wire	2	10	module02	5	drawing_bat_positive	0,0032	0,0037	0,8000	0	0	0	1
30	5	vws_fuse01.11AF	6	spl011	2	wire	21	10	module02	1	drawing01	0,0076	0,0089	0,4467	0	0	0	1
31	5	vws_fuse01.11AF	6	spl011	2	wire	21	11	module03	1	drawing01	0,0076	0,0089	0,4467	0	0	0	1
32	6	spl011	7	vws_01.11	2	wire	8	10	module02	1	drawing01	0,0127	0,0149	0,5940	0	0	0	1
33	6	spl011	7	vws_01.11	2	wire	8	11	module03	1	drawing01	0,0127	0,0149	0,5940	0	0	0	1

General_ID	Source	Source	Destination	Destination	Type	Type	SEAT_ID	Module	Module	Harness	Harness	Specific Res	Max. Sp. Res	Length	Max. Curr.	Min. Curr.	Nom. Curr.	State
34	7	vws_01.11	21	vws_01.13	7	ECU	1	10	module02	0	no_one	0,0000	0,0000	0,0000	30,01	0	0	1
35	7	vws_01.11	21	vws_01.13	7	ECU	1	11	module03	0	no_one	0,0000	0,0000	0,0000	30,02	0	0	1
36	21	vws_01.13	22	vws_05.11	2	wire	17	10	module02	1	drawing01	0,0191	0,0223	0,2588	0	0	0	1
37	21	vws_01.13	22	vws_05.11	2	wire	17	11	module03	1	drawing01	0,0191	0,0223	0,2588	0	0	0	1
38	26	vws_05.12	24	SPM011	2	wire	20	10	module02	1	drawing01	0,0191	0,0223	0,6680	0	0	0	1
39	26	vws_05.12	24	SPM011	2	wire	20	11	module03	1	drawing01	0,0191	0,0223	0,6680	0	0	0	1
40	3	vws_mp_00.11	4	A.31B	2	wire	5	10	module02	4	drawing_bat_negative	0,0032	0,0037	0,8000	0	0	0	1
41	7	vws_01.11	23	vws_01.14	7	ECU	1	10	module02	0	no_one	0,0000	0,0000	0,0000	0,51	0	0	1
42	7	vws_01.11	23	vws_01.14	7	ECU	1	11	module03	0	no_one	0,0000	0,0000	0,0000	0,52	0	0	1
43	23	vws_01.14	24	SPM011	2	wire	18	10	module02	1	drawing01	0,0127	0,0149	0,7933	0	0	0	1
44	23	vws_01.14	24	SPM011	2	wire	18	11	module03	1	drawing01	0,0127	0,0149	0,7933	0	0	0	1
45	1	vws_fuse01.11A	49	vws_fuse01.12AF	9	fuse	2	10	module02	0	no_one	0,0000	0,0000	0,0000	0	0	0	1
46	1	vws_fuse01.11A	49	vws_fuse01.12AF	9	fuse	2	11	module03	0	no_one	0,0000	0,0000	0,0000	0	0	0	1
47	49	vws_fuse01.12AF	50	sp31	2	wire	36	10	module02	3	drawing03	0,0254	0,0298	0,7338	0	0	0	1
48	49	vws_fuse01.12AF	50	sp31	2	wire	36	11	module03	3	drawing03	0,0254	0,0298	0,7338	0	0	0	1
49	50	sp31	51	vws_08.11	2	wire	37	10	module02	3	drawing03	0,0254	0,0298	0,8606	0	0	0	1
50	50	sp31	51	vws_08.11	2	wire	37	11	module03	3	drawing03	0,0254	0,0298	0,8606	0	0	0	1
51	52	vws_08.12	53	vws_mp_03.11	2	wire	38	10	module02	3	drawing03	0,0254	0,0298	0,9931	0	0	0	1
52	52	vws_08.12	53	vws_mp_03.11	2	wire	38	11	module03	3	drawing03	0,0254	0,0298	0,9931	0	0	0	1
53	53	vws_mp_03.11	3	vws_mp_00.11	15	body	1	0	no_one	0	no_one	0,0000	0,0000	0,0000	0	0	0	1
54	1	vws_fuse01.11A	29	vws_fuse01.11C	12	plate	3	10	module02	0	no_one	0,0000	0,0000	0,0000	0	0	0	1
55	1	vws_fuse01.11A	29	vws_fuse01.11C	12	plate	3	11	module03	0	no_one	0,0000	0,0000	0,0000	0	0	0	1
56	29	vws_fuse01.11C	28	vws_fuse01.13F	2	wire	23	10	module02	1	drawing01	0,0127	0,0149	0,6974	0	0	0	1
57	29	vws_fuse01.11C	28	vws_fuse01.13F	2	wire	23	11	module03	1	drawing01	0,0127	0,0149	0,6974	0	0	0	1
58	28	vws_fuse01.13F	56	vws_fuse01.13AF	9	fuse	6	10	module02	0	no_one	0,0000	0,0000	0,0000	0	0	0	1
59	28	vws_fuse01.13F	56	vws_fuse01.13AF	9	fuse	6	11	module03	0	no_one	0,0000	0,0000	0,0000	0	0	0	1
60	56	vws_fuse01.13AF	50	sp31	2	wire	40	10	module02	3	drawing03	0,0254	0,0298	0,8298	0	0	0	1
61	56	vws_fuse01.13AF	50	sp31	2	wire	40	11	module03	3	drawing03	0,0254	0,0298	0,8298	0	0	0	1
62	2	A.31A	1	vws_fuse01.11A	2	wire	3	11	module03	5	drawing_bat_positive	0,0019	0,0022	0,8000	0	0	0	1
63	6	spl011	37	vws_fuse01.153F	2	wire	29	11	module03	1	drawing01	0,0191	0,0223	0,5863	0	0	0	1
64	37	vws_fuse01.153F	39	vws_fuse01.152F	14	relay_switch	8	11	module03	0	no_one	0,0000	0,0000	0,0000	0	0	0	1
65	39	vws_fuse01.152F	38	vws_10.11	2	wire	30	6	mod_fam02_01	1	drawing01	0,0191	0,0223	0,8707	0	0	0	1
66	39	vws_fuse01.152F	38	vws_10.11	2	wire	30	7	mod_fam02_02	1	drawing01	0,0191	0,0223	0,8707	0	0	0	1
67	39	vws_fuse01.152F	38	vws_10.11	2	wire	30	8	mod_fam02_03	1	drawing01	0,0191	0,0223	0,8707	0	0	0	1
68	40	vws_10.12	24	SPM011	2	wire	31	6	mod_fam02_01	1	drawing01	0,0191	0,0223	0,6820	0	0	0	1

General_ID	Source	Source	Destination	Destination	Type	Type	SEAT_ID	Module	Module	Harness	Harness	Specific Res	Max. Sp. Res	Length	Max. Curr.	Min. Curr.	Nom. Curr.	State
69	40	vws_10.12	24	SPM011	2	wire	31	7	mod_fam02_02	1	drawing01	0,0191	0,0223	0,6820	0	0	0	1
70	40	vws_10.12	24	SPM011	2	wire	31	8	mod_fam02_03	1	drawing01	0,0191	0,0223	0,6820	0	0	0	1
71	3	vws_mp_00.11	4	A.31B	2	wire	6	11	module03	4	drawing_bat_negative	0,0019	0,0022	0,8000	0	0	0	1
72	6	spl011	9	vws_03.11	2	wire	10	11	module03	1	drawing01	0,0127	0,0149	0,5155	0	0	0	1
73	15	vws_03.12	16	vws_mp_02.11A	2	wire	14	11	module03	1	drawing01	0,0127	0,0149	0,9198	0	0	0	1
74	16	vws_mp_02.11A	3	vws_mp_00.11	15	body	1	0	no_one	0	no_one	0,0000	0,0000	0,0000	0	0	0	1
75	6	spl011	10	vws_fuse01.14F	2	wire	11	11	module03	1	drawing01	0,0127	0,0149	0,9213	0	0	0	1
76	10	vws_fuse01.14F	17	vws_fuse01.14AF	9	fuse	4	11	module03	0	no_one	0,0000	0,0000	0,0000	0	0	0	1
77	17	vws_fuse01.14AF	18	vws_04.11	2	wire	15	11	module03	1	drawing01	0,0191	0,0223	0,9856	0	0	0	1
78	19	vws_04.12	20	vws_mp_02.11B	2	wire	16	11	module03	1	drawing01	0,0191	0,0223	0,5784	0	0	0	1
79	20	vws_mp_02.11B	3	vws_mp_00.11	15	body	1	0	no_one	0	no_one	0,0000	0,0000	0,0000	0	0	0	1
80	1	vws_fuse01.11A	30	vws_fuse01.15AF	9	fuse	5	11	module03	0	no_one	0,0000	0,0000	0,0000	0	0	0	1
81	30	vws_fuse01.15AF	31	spl_apr1	2	wire	24	11	module03	1	drawing01	0,0127	0,0149	0,5994	0	0	0	1
82	31	spl_apr1	32	vws_fuse01.150F	2	wire	25	11	module03	1	drawing01	0,0127	0,0149	0,7886	0	0	0	1
83	32	vws_fuse01.150F	35	vws_fuse01.151F	13	relay_coil	7	11	module03	0	no_one	0,0000	0,0000	0,0000	0	0	0	1
84	35	vws_fuse01.151F	34	vws_09.12	2	wire	27	11	module03	1	drawing01	0,1467	0,1719	0,9267	0	0	0	1
85	36	vws_09.13	24	SPM011	2	wire	28	11	module03	1	drawing01	0,1467	0,1719	0,6844	0	0	0	1
86	31	spl_apr1	33	vws_09.11	2	wire	26	11	module03	1	drawing01	0,1467	0,1719	0,7979	0	0	0	1

Appendix 16. Master loads table output

General_ID	Positive_pin	Positive_pin	Negative_pin	Negative_pin	Type	Type	SEAT_ID	SEAT_ID	SubID	SubID	Module	Module	Nom. Curr.	Max. Curr.	Peak. Curr.	StandBy Curr.	Nom. Volt.	Level	Level	Ty_Of_Load	Ty_Of_Load	State
1	42	vws_07.11A	46	vws_07.12A	7	black_box	5	vws_07.1	1	pn091	9	module01	7,01	8,01	9,01	0	14	1	NC	1	CC	1
2	42	vws_07.11A	46	vws_07.12A	7	black_box	5	vws_07.1	2	pn092	10	module02	7,02	8,02	9,02	0	14	1	NC	1	CC	1
3	42	vws_07.11A	46	vws_07.12A	7	black_box	5	vws_07.1	3	pn093	11	module03	7,03	8,03	9,03	0	14	1	NC	1	CC	1
4	22	vws_05.11	26	vws_05.12	7	black_box	4	vws_05.1	1	pn061	10	module02	15,01	16,01	16,11	0	14	1	NC	1	CC	1
5	22	vws_05.11	26	vws_05.12	7	black_box	4	vws_05.1	2	pn062	11	module03	16,02	16,02	16,12	0	14	1	NC	1	CC	1
6	51	vws_08.11	52	vws_08.12	7	black_box	6	vws_08.1	1	pn102	10	module02	8,02	8,52	8,92	0	14	1	NC	1	CC	1
7	51	vws_08.11	52	vws_08.12	7	black_box	6	vws_08.1	2	pn103	11	module03	8,03	8,53	8,93	0	14	1	NC	1	CC	1

General_ID	Positive_pin	Positive_pin	Negative_pin	Negative_pin	Type	Type	SEAT_ID	SEAT_ID	SubID	SubID	Module	Module	Nom. Curr.	Max. Curr.	Peak. Curr.	StandBy Curr.	Nom. Volt.	Level	Level	Ty_Of_Load	Ty_Of_Load	State
8	38	vws_10.11	40	vws_10.12	7	black_box	8	vws_10.1	1	pn083	6	mod_fam02_01	1,00	2	3	0	14	1	NC	1	CC	1
9	38	vws_10.11	40	vws_10.12	7	black_box	8	vws_10.1	1	pn083	7	mod_fam02_02	1,00	2	3	0	14	1	NC	1	CC	1
10	38	vws_10.11	40	vws_10.12	7	black_box	8	vws_10.1	1	pn083	8	mod_fam02_03	1,00	2	3	0	14	1	NC	1	CC	1
11	9	vws_03.11	15	vws_03.12	7	black_box	2	vws_03.1	1	pn041	11	module03	3,00	4	4,1	0	14	1	NC	1	CC	1
12	18	vws_04.11	19	vws_04.12	7	black_box	3	vws_04.1	1	pn051	11	module03	4,00	5	5,1	0	14	1	NC	1	CC	1
13	34	vws_09.12	36	vws_09.13	7	black_box	7	vws_09.1	1	pn073	11	module03	2,00	2,5	3,2	0	14	1	NC	1	CC	1
14	33	vws_09.11	36	vws_09.13	7	black_box	7	vws_09.1	1	pn073	11	module03	1,00	1,2	1,5	0	14	1	NC	1	CC	1

NC = Nominal Current
CC = Constant Current

Appendix 17. Connection matrix

1	2	1	2	1	9	5	0,0048	0,0056	0,8000	0	0	0	1
2	1	5	9	1	9	0	0,0000	0,0000	0,0000	0	0	0	1
3	1	5	9	1	10	0	0,0000	0,0000	0,0000	0	0	0	1
4	1	5	9	1	11	0	0,0000	0,0000	0,0000	0	0	0	1
5	5	6	2	7	9	1	0,0127	0,0149	0,2880	0	0	0	1
6	6	8	2	9	9	1	0,0127	0,0149	0,7055	0	0	0	1
7	6	8	2	9	10	1	0,0127	0,0149	0,7055	0	0	0	1
8	6	8	2	9	11	1	0,0127	0,0149	0,7055	0	0	0	1
9	8	41	8	1	9	0	0,0000	0,0000	0,0000	0	0	0	1
10	8	41	8	1	10	0	0,0000	0,0000	0,0000	0	0	0	1
11	8	41	8	1	11	0	0,0000	0,0000	0,0000	0	0	0	1
12	41	42	2	32	9	2	0,0127	0,0149	0,1566	0	0	0	1
13	41	42	2	32	10	2	0,0127	0,0149	0,1566	0	0	0	1

14	41	42	2	32	11	2	0,0127	0,0149	0,1566	0	0	0	1
15	46	45	2	34	9	2	0,1467	0,1719	0,3590	0	0	0	1
16	46	45	2	34	10	2	0,1467	0,1719	0,3590	0	0	0	1
17	46	45	2	34	11	2	0,1467	0,1719	0,3590	0	0	0	1
18	45	25	8	1	9	0	0,0000	0,0000	0,0000	0	0	0	1
19	45	25	8	1	10	0	0,0000	0,0000	0,0000	0	0	0	1
20	45	25	8	1	11	0	0,0000	0,0000	0,0000	0	0	0	1
21	25	24	2	19	9	1	0,0127	0,0149	0,2588	0	0	0	1
22	25	24	2	19	10	1	0,0127	0,0149	0,2588	0	0	0	1
23	25	24	2	19	11	1	0,0127	0,0149	0,2588	0	0	0	1
24	24	27	2	22	9	1	0,0127	0,0149	0,7226	0	0	0	1
25	24	27	2	22	10	1	0,0127	0,0149	0,7226	0	0	0	1
26	24	27	2	22	11	1	0,0127	0,0149	0,7226	0	0	0	1
27	27	3	15	1	0	0	0,0000	0,0000	0,0000	0	0	0	1
28	3	4	2	4	9	4	0,0048	0,0056	0,8000	0	0	0	1
29	2	1	2	2	10	5	0,0032	0,0037	0,8000	0	0	0	1
30	5	6	2	21	10	1	0,0076	0,0089	0,4467	0	0	0	1
31	5	6	2	21	11	1	0,0076	0,0089	0,4467	0	0	0	1
32	6	7	2	8	10	1	0,0127	0,0149	0,5940	0	0	0	1
33	6	7	2	8	11	1	0,0127	0,0149	0,5940	0	0	0	1
34	7	21	7	1	10	0	0,0000	0,0000	0,0000	30,01	0	0	1
35	7	21	7	1	11	0	0,0000	0,0000	0,0000	30,02	0	0	1
36	21	22	2	17	10	1	0,0191	0,0223	0,2588	0	0	0	1
37	21	22	2	17	11	1	0,0191	0,0223	0,2588	0	0	0	1
38	26	24	2	20	10	1	0,0191	0,0223	0,6680	0	0	0	1
39	26	24	2	20	11	1	0,0191	0,0223	0,6680	0	0	0	1
40	3	4	2	5	10	4	0,0032	0,0037	0,8000	0	0	0	1
41	7	23	7	1	10	0	0,0000	0,0000	0,0000	0,51	0	0	1
42	7	23	7	1	11	0	0,0000	0,0000	0,0000	0,52	0	0	1
43	23	24	2	18	10	1	0,0127	0,0149	0,7933	0	0	0	1

44	23	24	2	18	11	1	0,0127	0,0149	0,7933	0	0	0	1
45	1	49	9	2	10	0	0,0000	0,0000	0,0000	0	0	0	1
46	1	49	9	2	11	0	0,0000	0,0000	0,0000	0	0	0	1
47	49	50	2	36	10	3	0,0254	0,0298	0,7338	0	0	0	1
48	49	50	2	36	11	3	0,0254	0,0298	0,7338	0	0	0	1
49	50	51	2	37	10	3	0,0254	0,0298	0,8606	0	0	0	1
50	50	51	2	37	11	3	0,0254	0,0298	0,8606	0	0	0	1
51	52	53	2	38	10	3	0,0254	0,0298	0,9931	0	0	0	1
52	52	53	2	38	11	3	0,0254	0,0298	0,9931	0	0	0	1
53	53	3	15	1	0	0	0,0000	0,0000	0,0000	0	0	0	1
54	1	29	12	3	10	0	0,0000	0,0000	0,0000	0	0	0	1
55	1	29	12	3	11	0	0,0000	0,0000	0,0000	0	0	0	1
56	29	28	2	23	10	1	0,0127	0,0149	0,6974	0	0	0	1
57	29	28	2	23	11	1	0,0127	0,0149	0,6974	0	0	0	1
58	28	56	9	6	10	0	0,0000	0,0000	0,0000	0	0	0	1
59	28	56	9	6	11	0	0,0000	0,0000	0,0000	0	0	0	1
60	56	50	2	40	10	3	0,0254	0,0298	0,8298	0	0	0	1
61	56	50	2	40	11	3	0,0254	0,0298	0,8298	0	0	0	1
62	2	1	2	3	11	5	0,0019	0,0022	0,8000	0	0	0	1
63	6	37	2	29	11	1	0,0191	0,0223	0,5863	0	0	0	1
64	37	39	14	8	11	0	0,0000	0,0000	0,0000	0	0	0	1
65	39	38	2	30	6	1	0,0191	0,0223	0,8707	0	0	0	1
66	39	38	2	30	7	1	0,0191	0,0223	0,8707	0	0	0	1
67	39	38	2	30	8	1	0,0191	0,0223	0,8707	0	0	0	1
68	40	24	2	31	6	1	0,0191	0,0223	0,6820	0	0	0	1
69	40	24	2	31	7	1	0,0191	0,0223	0,6820	0	0	0	1
70	40	24	2	31	8	1	0,0191	0,0223	0,6820	0	0	0	1
71	3	4	2	6	11	4	0,0019	0,0022	0,8000	0	0	0	1
72	6	9	2	10	11	1	0,0127	0,0149	0,5155	0	0	0	1
73	15	16	2	14	11	1	0,0127	0,0149	0,9198	0	0	0	1

74	16	3	15	1	0	0	0,0000	0,0000	0,0000	0	0	0	1
75	6	10	2	11	11	1	0,0127	0,0149	0,9213	0	0	0	1
76	10	17	9	4	11	0	0,0000	0,0000	0,0000	0	0	0	1
77	17	18	2	15	11	1	0,0191	0,0223	0,9856	0	0	0	1
78	19	20	2	16	11	1	0,0191	0,0223	0,5784	0	0	0	1
79	20	3	15	1	0	0	0,0000	0,0000	0,0000	0	0	0	1
80	1	30	9	5	11	0	0,0000	0,0000	0,0000	0	0	0	1
81	30	31	2	24	11	1	0,0127	0,0149	0,5994	0	0	0	1
82	31	32	2	25	11	1	0,0127	0,0149	0,7886	0	0	0	1
83	32	35	13	7	11	0	0,0000	0,0000	0,0000	0	0	0	1
84	35	34	2	27	11	1	0,1467	0,1719	0,9267	0	0	0	1
85	36	24	2	28	11	1	0,1467	0,1719	0,6844	0	0	0	1
86	31	33	2	26	11	1	0,1467	0,1719	0,7979	0	0	0	1

Appendix 18. Load matrix

1	42	46	7	5	1	9	7,01	8,01	9,01	0	14	1	1	1
2	42	46	7	5	2	10	7,02	8,02	9,02	0	14	1	1	1
3	42	46	7	5	3	11	7,03	8,03	9,03	0	14	1	1	1
4	22	26	7	4	1	10	15,01	16,01	16,11	0	14	1	1	1
5	22	26	7	4	2	11	16,02	16,02	16,12	0	14	1	1	1
6	51	52	7	6	1	10	8,02	8,52	8,92	0	14	1	1	1
7	51	52	7	6	2	11	8,03	8,53	8,93	0	14	1	1	1
8	38	40	7	8	1	6	1	2	3	0	14	1	1	1
9	38	40	7	8	1	7	1	2	3	0	14	1	1	1
10	38	40	7	8	1	8	1	2	3	0	14	1	1	1
11	9	15	7	2	1	11	3	4	4,1	0	14	1	1	1
12	18	19	7	3	1	11	4	5	5,1	0	14	1	1	1
13	34	36	7	7	1	11	2	2,5	3,2	0	14	1	1	1
14	33	36	7	7	1	11	1	1,2	1,5	0	14	1	1	1

Appendix 19. Nodal voltages results with module01 and mod_fam02_03 activated

Location_node	Slot_node	Pin_node	Voltage_(V)
A.3	A	1	14,000
vws_fuse01.1	A	1	13,973
vws_fuse01.1	F	1A	13,903
spl01		1	13,877
vws_02.1		1	13,815
vws_06.1		1	13,780
vws_07.1	A	1	13,766
A.3	B	1	0,000
vws_mp_00.1		1	0,027
vws_mp_01.1		1	0,034
SPM01		1	0,116
vws_02.1		3	0,139
vws_06.1		3	0,174
vws_07.1	A	2	0,543

Appendix 20. Branches currents results with module01 and mod_fam02_03 activated

Location_extreme_1	Slot_extreme_1	Pin_extreme_1	Location_extreme_2	Slot_extreme_2	Pin_extreme_2	Branch_current_(A)	Branch_Power_transfer_(W)	Branch_Power_losses_(W)
A.3	A	1	vws_fuse01.1	A	1	7,01	98,140	0,187
vws_fuse01.1	A	1	vws_fuse01.1	F	1A	7,01	97,953	0,491
vws_fuse01.1	F	1A	spl01		1	7,01	97,461	0,180
spl01		1	vws_02.1		1	7,01	97,281	0,441
vws_02.1		1	vws_06.1		1	7,01	96,841	0,246
vws_06.1		1	vws_07.1	A	1	7,01	96,595	0,098
A.3	B	1	vws_mp_00.1		1	7,01	0,187	0,187
vws_mp_00.1		1	vws_mp_01.1		1	7,01	0,237	0,049
vws_mp_01.1		1	SPM01		1	7,01	0,811	0,574
SPM01		1	vws_02.1		3	7,01	0,973	0,162
vws_02.1		3	vws_06.1		3	7,01	1,218	0,246
vws_06.1		3	vws_07.1	A	2	7,01	3,806	2,588

Appendix 21. Differential voltages at components' terminals results with module01 and mod_fam02_03 activated

Location_ positive_node	Slot_ positive_node	Pin_ positive_node	Location_ negative_node	Slot_ negative_node	Pin_ negative_node	Differential_ voltage_(V)	Condition
vws_07.1	A	1	vws_07.1	A	2	13,222	Admissible value

Appendix 22. Nodal voltages with module02 and mod_fam02_03 activated

Location_node	Slot_node	Pin_node	Voltage_(V)
A.3	A	1	14,000
vws_fuse01.1	A	1	13,922
vws_fuse01.1	F	1A	13,697
spl01		1	13,620
vws_01.1		1	13,517
vws_01.1		3	13,502
vws_05.1		1	13,428
vws_02.1		1	13,557
vws_06.1		1	13,522
vws_07.1	A	1	13,508
vws_fuse01.1	F	2A	13,881
sp3		1	13,780
vws_08.1		1	13,594
vws_fuse01.1	C	1	13,918
vws_fuse01.1	F	3	13,873
vws_fuse01.1	F	3A	13,833
A.3	B	1	0,000
vws_mp_00.1		1	0,078
vws_mp_01.1		1	0,100
SPM01		1	0,364
vws_05.1		2	0,517
vws_01.1		4	0,369
vws_02.1		3	0,387
vws_06.1		3	0,422
vws_07.1	A	2	0,792
vws_mp_03.1		1	0,086
vws_08.1		2	0,248

Appendix 23. Branches currents results with module02 and mod_fam02_03 activated

Location_ extreme_1	Slot_ extreme_1	Pin_ extreme_1	Location_ extreme_2	Slot_ extreme_2	Pin_ extreme_2	Branch_ current_(A)	Branch_Power _transfer_(W)	Branch_Power _losses_(W)
A.3	A	1	vws_fuse01.1	A	1	30,560	427,840	2,375
vws_fuse01.1	A	1	vws_fuse01.1	F	1A	22,540	313,809	5,081
vws_fuse01.1	F	1A	spl01		1	22,540	308,728	1,731
spl01		1	vws_01.1		1	15,520	211,384	1,600
vws_01.1		1	vws_01.1		3	15,010	202,890	0,225
vws_01.1		3	vws_05.1		1	15,010	202,665	1,112
spl01		1	vws_02.1		1	7,020	95,613	0,442
vws_02.1		1	vws_06.1		1	7,020	95,171	0,246
vws_06.1		1	vws_07.1	A	1	7,020	94,925	0,098
vws_fuse01.1	A	1	vws_fuse01.1	F	2A	4,093	56,983	0,168
vws_fuse01.1	F	2A	sp3		1	4,093	56,816	0,415
sp3		1	vws_08.1		1	8,020	110,516	1,488
vws_fuse01.1	A	1	vws_fuse01.1	C	1	3,927	54,674	0,015
vws_fuse01.1	C	1	vws_fuse01.1	F	3	3,927	54,658	0,179
vws_fuse01.1	F	3	vws_fuse01.1	F	3A	3,927	54,479	0,154
vws_fuse01.1	F	3A	sp3		1	3,927	54,325	0,210
A.3	B	1	vws_mp_00.1		1	30,560	2,375	2,375
vws_mp_00.1		1	vws_mp_01.1		1	22,540	2,260	0,508
vws_mp_01.1		1	SPM01		1	22,540	8,198	5,939
SPM01		1	vws_05.1		2	15,010	8,028	2,305
SPM01		1	vws_01.1		4	0,510	5,532	0,002
SPM01		1	vws_02.1		3	7,020	5,806	0,162
vws_02.1		3	vws_06.1		3	7,020	2,962	0,246
vws_06.1		3	vws_07.1	A	2	7,020	5,557	2,595
vws_mp_00.1		1	vws_mp_03.1		1	8,020	0,602	0,064
vws_mp_03.1		1	vws_08.1		2	8,020	1,015	1,302

Appendix 24. Differential voltages at components' terminals results with module02 and mod_fam02_03 activated

Location_ positive_node	Slot_ positive_node	Pin_ positive_node	Location_ negative_node	Slot_ negative_node	Pin_ negative_node	Differential_vol tage_(V)	Condition
vws_07.1	A	1	vws_07.1	A	2	12,716	Admissible value
vws_05.1		1	vws_05.1		2	12,911	Admissible value
vws_01.1		1	vws_01.1		4	13,148	Admissible value
vws_08.1		1	vws_08.1		2	13,346	Admissible value

Appendix 25. Nodal voltages results with module03 and mod_fam02_03 activated

Location_node	Slot_node	Pin_node	Voltage_(V)
A.3	A	1	14,000
vws_fuse01.1	A	1	13,935
vws_fuse01.1	F	1A	13,619
spl01		1	13,512
vws_fuse01.1	F	53	13,493
vws_fuse01.1	F	52	13,492
vws_10.1		1	13,480
vws_01.1		1	13,402
vws_01.1		3	13,386
vws_05.1		1	13,307
vws_02.1		1	13,449
vws_06.1		1	13,414
vws_07.1	A	1	13,400
vws_03.1		1	13,476
vws_fuse01.1	F	4	13,463
vws_fuse01.1	F	4A	13,423
vws_04.1		1	13,355
vws_fuse01.1	F	2A	13,894
sp3		1	13,793
vws_08.1		1	13,607
vws_fuse01.1	C	1	13,931
vws_fuse01.1	F	3	13,885
vws_fuse01.1	F	3A	13,846
vws_fuse01.1	F	5A	13,905
spl_apr		1	13,883
vws_fuse01.1	F	50	13,864
vws_fuse01.1	F	51	13,844
vws_09.1		2	13,649
vws_09.1		1	13,746
A.3	B	1	0,000
vws_mp_00.1		1	0,065
vws_mp_01.1		1	0,093
SPM01		1	0,415
vws_10.1		2	0,424
vws_05.1		2	0,579
vws_01.1		4	0,420
vws_02.1		3	0,438
vws_06.1		3	0,473
vws_07.1	A	2	0,843
vws_mp_02.1	A	1	0,068
vws_03.1		2	0,096
vws_mp_02.1	B	1	0,069
vws_04.1		2	0,131
vws_mp_03.1		1	0,073
vws_08.1		2	0,236
vws_09.1		3	0,775

Appendix 26. Branches currents results with module03 and mod_fam02_03 activated

Location_ extreme_1	Slot_ extreme_1	Pin_ extreme_1	Location_ extreme_2	Slot_ extreme_2	Pin_ extreme_2	Branch_ current_(A)	Branch_Power _transfer_(W)	Branch_Power _losses_(W)
A.3	A	1	vws_fuse01.1	A	1	42,600	596,400	2,769
vws_fuse01.1	A	1	vws_fuse01.1	F	1A	31,570	439,928	9,967
vws_fuse01.1	F	1A	spl01		1	31,570	429,962	3,396
spl01		1	vws_fuse01.1	F	53	1,000	13,512	0,019
vws_fuse01.1	F	53	vws_fuse01.1	F	52	1,000	13,493	0,002
vws_fuse01.1	F	52	vws_10.1		1	1,000	13,492	0,012
spl01		1	vws_01.1		1	16,540	223,484	1,817
vws_01.1		1	vws_01.1		3	16,020	214,698	0,257
vws_01.1		3	vws_05.1		1	16,020	214,441	1,267
spl01		1	vws_02.1		1	7,030	94,988	0,443
vws_02.1		1	vws_06.1		1	7,030	94,544	0,247
vws_06.1		1	vws_07.1	A	1	7,030	94,297	0,098
spl01		1	vws_03.1		1	3,000	40,535	0,108
spl01		1	vws_fuse01.1	F	4	4,000	54,047	0,195
vws_fuse01.1	F	4	vws_fuse01.1	F	4A	4,000	53,852	0,160
vws_fuse01.1	F	4A	vws_04.1		1	4,000	53,692	0,271
vws_fuse01.1	A	1	vws_fuse01.1	F	2A	4,098	57,106	0,168
vws_fuse01.1	F	2A	sp3		1	4,098	56,938	0,416
sp3		1	vws_08.1		1	8,030	110,754	1,492
vws_fuse01.1	A	1	vws_fuse01.1	C	1	3,932	54,792	0,015
vws_fuse01.1	C	1	vws_fuse01.1	F	3	3,932	54,776	0,180
vws_fuse01.1	F	3	vws_fuse01.1	F	3A	3,932	54,597	0,155
vws_fuse01.1	F	3A	sp3		1	3,932	54,442	0,210
vws_fuse01.1	A	1	vws_fuse01.1	F	5A	3,000	41,805	0,090
vws_fuse01.1	F	5A	spl_apr		1	3,000	41,715	0,065
spl_apr		1	vws_fuse01.1	F	50	2,000	27,767	0,039
vws_fuse01.1	F	50	vws_fuse01.1	F	51	2,000	27,728	0,040
vws_fuse01.1	F	51	vws_09.1		2	2,000	27,688	0,390
spl_apr		1	vws_09.1		1	1,000	13,883	0,137
A.3	B	1	vws_mp_00.1		1	42,600	2,769	2,769
vws_mp_00.1		1	vws_mp_01.1		1	27,570	2,922	0,760
vws_mp_01.1		1	SPM01		1	27,570	13,097	8,885
SPM01		1	vws_10.1		2	1,000	0,424	0,010
SPM01		1	vws_05.1		2	16,020	0,579	2,625
SPM01		1	vws_01.1		4	0,520	0,420	0,003
SPM01		1	vws_02.1		3	7,030	7,244	0,163
vws_02.1		3	vws_06.1		3	7,030	7,579	0,247
vws_06.1		3	vws_07.1	A	2	7,030	13,510	2,602
vws_mp_00.1		1	vws_mp_02.1	A	1	3,000	0,478	0,009
vws_mp_02.1	A	1	vws_03.1		2	3,000	0,675	0,084
vws_mp_00.1		1	vws_mp_02.1	B	1	4,000	0,485	0,016
vws_mp_02.1	B	1	vws_04.1		2	4,000	0,393	0,249
vws_mp_00.1		1	vws_mp_03.1		1	8,030	0,292	0,064
vws_mp_03.1		1	vws_08.1		2	8,030	0,942	1,305
SPM01		1	vws_09.1		3	3,000	3,098	1,079

Appendix 27. Differential voltages at components' terminals results with module03 and mod_fam02_03 activated

Location_ positive_node	Slot_ positive_node	Pin_ positive_node	Location_ negative_node	Slot_ negative_node	Pin_ negative_node	Differential_vol tage_(V)	Condition
vws_07.1	A	1	vws_07.1	A	2	12,556	Critical value
vws_05.1		1	vws_05.1		2	12,728	Admissible value
vws_09.1		2	vws_09.1		3	12,874	Admissible value
vws_09.1		1	vws_09.1		3	12,972	Admissible value
vws_01.1		1	vws_01.1		4	12,982	Admissible value
vws_10.1		1	vws_10.1		2	13,056	Admissible value
vws_04.1		1	vws_04.1		2	13,224	Admissible value
vws_08.1		1	vws_08.1		2	13,371	Admissible value
vws_03.1		1	vws_03.1		2	13,380	Admissible value