

Power Flow Simulation in the Product Development Process of Modern Vehicular DC Distribution Systems

Paola Mantilla-Pérez, Juan-Andrés Pérez-Rúa, Manuel Alberto Díaz Millán, Xavier Domínguez, *Student Member, IEEE* and Pablo Arboleya, *Senior Member, IEEE*

Abstract—Nowadays, automakers face unprecedented requirements to comply and exceed quality standards while attending to consumer expectations. Among the challenges are the insertion of new functions for driving assistance towards highly automated vehicles, electrification and connectivity. Introduction of simulation – driven design in the broad wide disciplines involved within vehicles development yields significant savings in both costs and product release time. This paper introduces an approach to vehicle electrical distribution systems (EDS) simulation adapting the methods used conventionally in transmission and distribution systems to the special features found in the vehicle EDS. To this purpose, a procedure based on a backward/forward sweep (BFS) algorithm for solving power flows in weakly meshed dc traction networks is applied and described. An important part of the work has to do with the information pre-processing from the modular based format used in automotive industry into standard simulation matrices. Constant current load profiles are assumed for the consumers, while the electronic control units (ECU) are considered static power distribution boxes. The main outputs of the proposed methodology are nodal voltages, branch currents and differential voltages at components terminals in the vehicle EDS. The knowledge extracted from the simulation will help the designers during the dimensioning and validation process of modern vehicles EDS and will be a powerful tool to reach the zero-prototypes goal before the start of production.

Index terms—Automotive modularity, vehicle electrical distribution system, DC power flow, Backward/Forward Sweep.

NOMENCLATURE

Acronyms

ASDS	AllSeriesDataSets
BFS	Backward-Forward Sweep
BOM	Bill of Materials
EDS	Electrical Distribution Systems

P. Arboleya is with the Department of Electrical Engineering, University of Oviedo, Spain e-mail: arboleyapablo@uniovi.es,

P. Mantilla-Pérez is with the Department of Electrical Engineering, University of Oviedo, Spain and SEAT S.A e-mail: paola.mantilla@seat.es

J.A. Pérez-Rúa is with the Technical University of Denmark, Denmark e-mail: juru@dtu.dk

M. Diaz Millan is with SEAT S.A e-mail: manuel-alberto.diaz@seat.es

Xavier Domínguez is with the Department of Electrical Engineering of the University of Oviedo, Spain and the Facultad de Ingeniería en Ciencias Aplicadas, Universidad Técnica del Norte, Ibarra, Ecuador, email: uo233585@uniovi.es

This work was partially supported by SEAT under grant for Industrial PhD. project FUIO-371-17 (Development of tools for electrical and thermal simulation of the on board electrical network of vehicles), and by Rucker-Lypsa under grant for Industrial PhD. project FUIO-200-18 (Development of a visual analytics tool for analysing and representing and analysing the on-board electrical network of the vehicles).

ECU	Electronic Control Unit
FAT	Research Association for Automotive Technology
FWD	Full Wiring Drawings
KBL	Automotive branch XML file
KCL	Kirchhoff current law
KVL	Kirchhoff voltage law
MN-BFS	Meshed-Network Backward/Forward Sweep
NR	Newton-Raphson
OEM	Original Equipment Manufacturer
QT	File condensing BOM,WL and loads information
VDA	German Association of the Automotive Industry
VHDL	Very High Speed Integrated Circuit Hardware Description Language
WHDP	Wiring Harness Development Process
WL	Wirelist
WP	Wiring Plans
WS	Wiring Schematics
XML	Extensible Markup Language

Variables

I	Branch current
n_N	Number of nodes
S	Node apparent power
T	Ambient temperature
V	Node voltage
Z	Branch impedance
ϵ	Voltage drop error

Vectors

\mathbf{I}	Branch currents vector
\mathbf{P}	Nodes active power vector
\mathbf{Q}	Nodes reactive power vector
\mathbf{V}	Nodes voltage amplitude vector
$\boldsymbol{\theta}$	Nodes voltage angle vector

Matrices

\mathbf{R}_b	Branch resistances matrix
$\boldsymbol{\Gamma}$	Incidence matrix
\mathbf{J}	Jacobian Matrix

Superscripts

k	k_{th} iteration number
TH	Thevenin

Subscripts

c	Cut branches
i	Node i
j	Node j
n	Negative grid
nc	Non-cut branches
p	Positive grid

Operators

Δ	Incremental
T	Transpose
*	Conjugate

I. INTRODUCTION

IN the last years, complexity of vehicle electrical distribution systems has considerably grown due to the aim of automotive OEMs (Original Equipment Manufacturer) to provide customization and advanced comfort, entertainment, and safety functions to users. Such trend imposes an increased amount of power consumers and wiring in vehicle EDS that reaches around 10^{10} possible architectures. It is only through the deployment of electrical simulation or equivalently, the construction of digital twins of the vehicle EDS that the multiple possible scenarios could be addressed and analyzed. Moreover, implementation of autonomous driving will add stricter demands on the fail-safe operation of every system and component within vehicles including its EDS. This turns EDS simulation into an imperative step inside the product development process. By means of computational simulation, early detection of possible electrical network failures can be done, such as excessive voltage drops or wrong components sizing for the whole range of feasible architectures, and consequently, to discard those ones which do not comply with operational and safety requirements in an early project stage. In addition, critical or damaging conditions could be safely reproduced in a virtual environment to observe the system response. With a simulation phase within the EDS design, the prototype phase would serve merely to corroborate the results already obtained by simulations, where naturally, a wider amount of architectures could be evaluated. As a consequence, the required number of prototypes could potentially be reduced through the use of virtual experimentation.

Although simulation-driven design is widely extended inside OEMs in areas such as vehicle crash and vehicle dynamics, it is not the case for vehicle EDS modeling. However, a few efforts have been devoted to it. Some commercial tools are available, such as the Power Net Simulation by Bosch [1] or SABER by Synopsys [2]. These tools are used to simulate a complete energy transfer system that include three major parts: the alternator, battery and the electrical loads, all operating under selected drive cycles or scenarios. These programs center their usefulness on an early phase of design to determine the correct dimensioning of the battery and alternator, given the required loads and the different driving scenarios. In these programs the characteristics of the EDS itself with all its connecting wires and flow control components such as fuses are not taken

into account. On the other hand, commercial tools for the design of vehicles EDS like LDorado [3] or EBCable [4], which contain the detailed information of the full EDS are not suited for simulation. From these, important differences of the present work in comparison to the existing commercial tools are summarized as follows:

- The group of commercial software for simulation does not center its functionality in the vehicle EDS itself including the complex wiring and components forming the harnesses. Instead, the focus lies on the proper dimensioning of battery and alternator (energetic balance)
- The underlying simulation methods used in the commercial tools are not always known by the user
- The simulation oriented commercial tools are mainly thought for an user to insert every single functional box with its corresponding parameters, while in the present work, a great advantage is the possibility for the user to extract the information of a complete harness and connected loads from the factory data, as received, thus saving the amount of effort required from the user
- The existing commercial software tools for EDS design consider all the details of the EDS components but do not allow the option for electrical simulation

II. STATE OF THE ART IN EDS SIMULATION

Not many works have been published to introduce a complete methodology for power flow simulation in vehicle EDS. There are a few examples of related work such as the development of freely-distributed Very High Speed Integrated Circuit Hardware Description Language models (VHDL models) for different components by the Research Association for Automotive Technology (FAT) in the German Association of the Automotive Industry (VDA) [5]. Other reference is the work by Petit [6] which shows the modeling of an alternator, battery and light bulb using Matlab Simulink. In this work the transient behavior of the network is simulated and experimentally corroborated using detailed lumped parameter models. A similar work by Bilyi [7] uses a combination of analytical calculations, lumped parameter models of power components and SimPowerSystems-blocks in Matlab Simulink for the simulation of a 12 V vehicle electrical network containing an alternator, excitation current control, different types of loads and battery, showing a solid match with experimental data.

In [8] Ruf and coworkers combine a modified discrete Particle Swarm Optimization with a physical power net simulation to meet voltage stability requirements in transient behaviors. Such physical power net simulation is performed using a Dymola-based tool with the models implemented in Modelica language. The simulation includes the model of chassis ground, wiring harness, battery, alternator, loads, dc/dc converter and electric double layer capacitors. However, the importance is given to the description of the optimization algorithms to achieve a minimum network weight with voltage stabilization topology rather than to the power flow description itself. In a more recent work, Gorelik et al. [9] introduce a simulation model for energy management systems

to perform analysis of power net designs taking into account the validation of safety requirements and the design of system fault reactions for automated driving applications. Simulation and optimization lie on the energy management, where under critical network conditions the degradation of loads will occur under a priority criteria where safety prevails over comfort. Similar to the work of [7] and [8], the wire harnesses are not object of study. On the other hand, the need for weight reduction has motivated the modeling and optimization of wire harnesses as described in [10], [11]. In these works, the authors have applied heat transfer equations to represent the thermal behavior of automobile cable harnesses in steady state and transient regime, under application of constant currents through the wires. The models have been validated against experimental data showing remarkable match. The modeling is, however, limited to the simulation of single wires or bundles but not applied to solution of full automotive electrical networks.

The related existing literature reports usually lack the methodic where it is clearly stated the step by step procedure employed to carry out the simulations of the power flow through the harnesses in the vehicle EDS. In particular, because the translation of the standard vehicle EDS information into a suitable input for a solver algorithm is challenging and secondly, in the existing works there is a scarce description of the methods employed to solve the equations derived from the network. In some cases, extremely detailed models are employed for each element in a Simulink or Modelica based approach which are hardly scalable to a more realistic network containing more than a hundred of power consumers. Although the power flow analysis of static networks such as transmission and distribution lines is a well-known topic within the power systems field, and it has been successfully employed in other applications like railways [12], [13], aircrafts [14]–[17] or ships [18]–[23], the application of such techniques for the vehicle EDS has been rarely addressed and needs to be adapted according to the complex structure of the vehicle and the available factory information. Typically, the information required to launch simulations is spread into different source files.

In this work, the authors propose a general procedure for simulating the complete electrical system of vehicles based in a modular wiring system design approach, as it is used within many European automotive OEMs. In addition, the application of an efficient and robust method for calculating the vehicle EDS based on a modified version of an algorithm used commonly for radial static distribution networks is demonstrated. The initial algorithm is known as Backward/Forward Sweep (BFS), and the cyclic-graph version as Meshed-Network Backward/Forward Sweep (MN-BFS).

The paper is structured as follows: in Section III the structure of the factory data is discussed together with the concept of modularity and the process of wire harness development, specifically, the vehicle EDS organization data is selected and grouped in three files which are subjected to pre-processing algorithms; Section IV describes the structure in which data is assembled in order to deliver it to the power flow solver, where the results from the pre-processing act as inputs

for the power flow algorithms. In section V the simulation strategy is explained including the treatment of the loads, and required inputs. Different power flow solvers are explained as the Newton-Raphson and the BFS including a performance comparison between them. The proposed power flow solution of the system assuming constant current consumption and applying the MN-BFS is described. At this point, the voltages and currents throughout the network are calculated. Section VI presents case studies based on two different vehicle sample networks that employ the described methodology to extract the node voltages and branch currents of the circuits. Finally, a set of conclusions are stated.

III. FACTORY DATA

Depending on the model, the full EDS of a vehicle can be composed by a set of different wire harnesses. For instance, a harness for the interior which transports energy to most of the consumers, but also smaller harnesses such as those for the doors or bumper. The different harnesses are interconnected by means of couplings in specific parts of the vehicle. The present method is thought to be applied one by one in each of the different wire harnesses co-existing in a vehicle independently, to approximate the response of the full EDS. The most important source of factory information are the electrical schematics describing each single harness. There exists one data container file per each of the wire harnesses present in the vehicle. Such data container files are represented in the form of an Extensible Markup Language file (XML) denominated KBL, which was designed for the automotive branch. As described in [24], each KBL document has both a logical and a physical structure. Physically, the KBL document is composed of units called entities, in a tree-structure. A document begins in a *root* or document entity. For the automotive case, the KBL files contain electrical components data information and they can be represented as 2D schematics by the use of specific software. A description of the structure for the KBL files has been published by the German Association of the Automotive Industry VDA (Verband der Automobilindustrie) in [5].

One limitation within the use of the KBL files for simulation is that their structure and definitions are not conceived for electrical simulation. The data containers structure has been developed to facilitate exchange of product data between OEMs and wire-harness manufacturers mainly for production purposes. For instance, although a definition of nodes is present in the KBL, they do not correspond to what would be specified as an electrical node and instead are associated in many cases to constructional aspects of the wire harness. Another limitation lies in the need to extract additional information of the power consumers, electrical and wiring elements in external databases, given that not all the necessary information is contained in the KBL files. Examples of such information are the time-current characteristics of fuses, pin-out information of the different consumers, temperature class of each of the cables, among others. Apart from the challenges that arise from the available product data scheme in vehicle EDS, there are technical aspects like the greater current densities, the presence of electronic loads and the generally

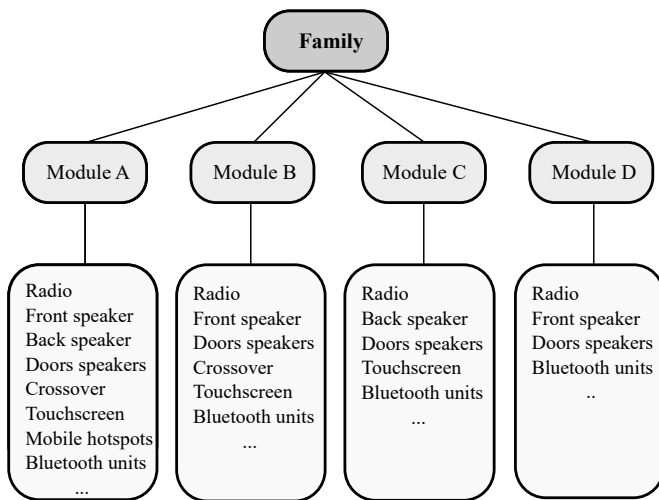


Fig. 1. Concept of modularity.

weakly meshed characteristics of this kind of distribution systems which must be attended in order to develop a computational tool for calculating the electrical variables in the system.

There exist two important variants within OEMs that determine the way in which factory data is organized: the few variants model and the many variants model. In the first case, the OEMs offer a restricted amount of selectable options for a particular car model. On the contrary, in the latter, OEMs offer a vast range of selectable options to customers. This is an advantage from the customer point of view but it requires a strong logistic effort in order to provide fully-customized vehicles. The strategy behind the flexibility of a vehicle configuration lies in the concept of modularity.

Modularity is the strategy followed in the wiring harness development to allow full customization of electrical functions of vehicles. It operates by means of families and modules. A family represents a group of elements able to perform a specific function. This function can be, for instance, the lighting. The sound system is another functional system formed by the radio, front loudspeakers, back loudspeakers, touch screen, etc. To operate the lighting functionality of the vehicle there are units as the brake light, indicators, fog lights, head lights, rear lamps, among others. The components forming the families are in turn organized in modules. In this way a functional family can be formed by a certain group of modules, all of them capable of performing the function and where each module is composed by diverse parts and might represent a different level of complexity, as shown in Figure 1. Following the example of the sound system family the possible configurations might include modules A and B. Module A could represent a higher equipped sound system compared to module B. The type of loudspeakers, which are loads in the vehicle electrical system, would also be classified according to modules. The reason is that the load characteristics have impact on the wiring harness components. For instance, a higher power loudspeaker would require a bigger wire cross-section. On the other hand, the elements in a specific family

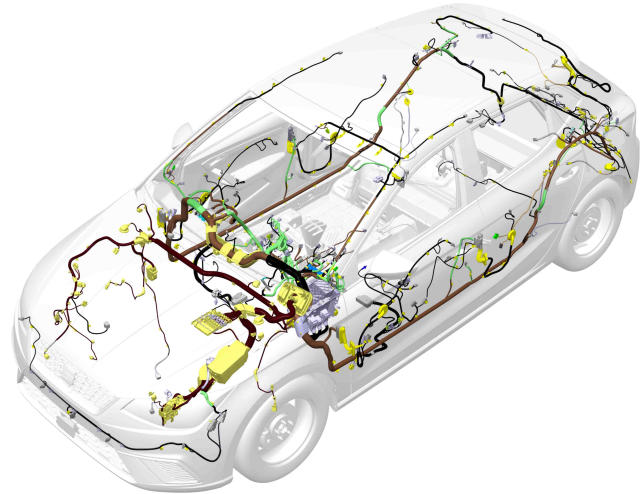


Fig. 2. Sketch of the wire harnesses inside a vehicle.

are not only electrical as fuses, wires, relays and connectors but also fixing and protective elements such as clips and tapes. Each single element in the vehicle wiring harness is associated to a module, which in turn is associated to a family. As a general rule, for each family in the vehicle only one module is present. Then, a full description of a specific and unique wiring harness configuration is based on a list where for each functional family defined there is a module selected.

The final product of the EDS wiring harness development is the full wiring harnesses description as they would be installed inside the vehicle, maintaining the modularity information all along. Figure 2 shows the scheme of all the wiring harnesses present in a car for a given user-defined configuration and vehicle model.

Databases containing the electrical information and properties of all components to be used within the developed electrical network are built before hand and used all along the process.

The Wiring Harness Development Process (WHDP) then starts with the design of system schematics and ends with the actual development of the wiring harness inside the vehicle, going through different stages.

The first step for the wiring department is the creation of the system diagrams for a particular car model. These diagrams are general representations of the logical connections among all the components inside the on-board electrical network and will be referred to as wiring schematics WS. The components are sketched as boxes having a specific pin-out. In overall, the WS are a set of drawings that include clearly the electrical power distribution from the battery to the last of the consumers, indicating the connections through fuses, relays and other electrical elements and including cross-references between WS files. By employing the system diagrams it is possible to determine all the connections that occur inside the vehicle excluding the technical details such as type of wiring, wiring cross-section, couplings, among others.

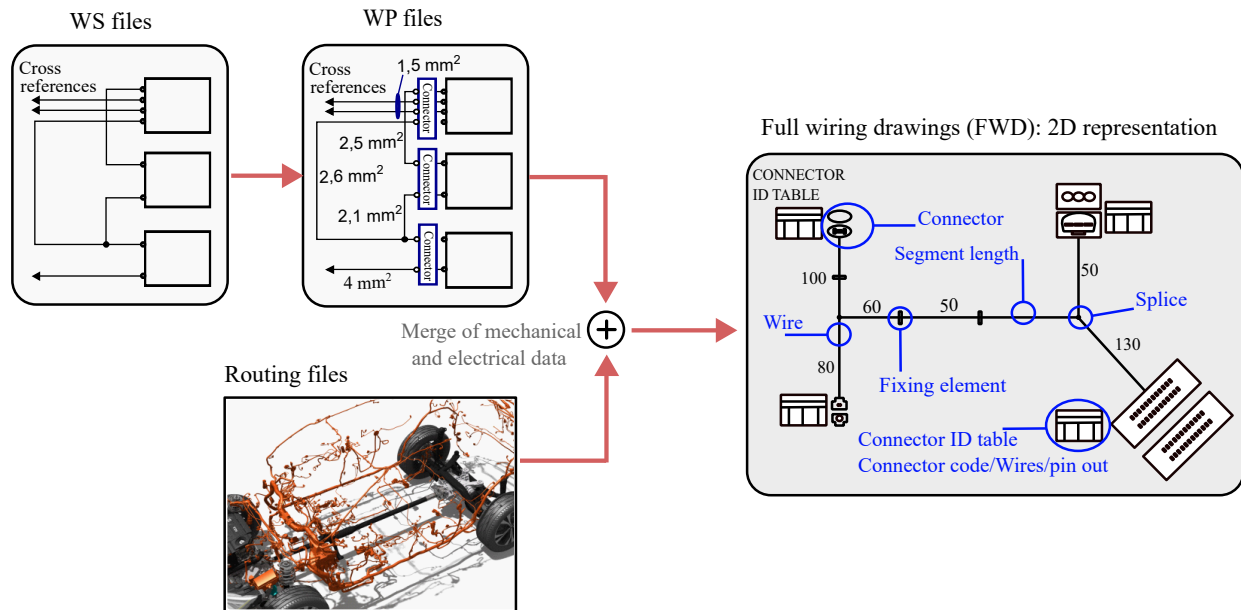


Fig. 3. Wiring Harness Development Process (WHDP).

The second step in the WHDP is the elaboration of the cable wiring plan, also described as WP files. Graphically, the wiring diagrams are similar to the WS files, except for the fact that they include the numbering of the cables together with the physical characteristics of the cables holding the connections specified in the WS. Technical information such as cables type, cables cross-section and couplings are included in the WP.

Simultaneously to the development of the WS and WP files takes place the development of the routing files for the wires. All the wire paths which connect one component to another are designed taking mechanical considerations into account and saved into 3D CAD files. Similarly to the case of the WS and WP, there exists a database containing the graphical information of every component that can be reached through the CAD software.

A third step in the WHDP is the merging of the cable wiring plans WP with the routing diagrams to generate a full graphical description of the wiring harness containing also the cable lengths information. A KBL file type that contains the mixed mechanical representation of the wires and components and their electrical data is formed. The resulting KBL files contain 3D information that can be later represented into 2D schematics. Such are saved with XML extension. These elaborated 2D wiring drawings, which are the result of the WHDP contain all the relevant manufacturing information of the wiring harness and will be referred to as full wiring drawings FWD.

As the last step of the process, all the generated information is uploaded to specific databases. A scheme with the representation of the WHDP flow is shown in Figure 3 including a section of a FWD.

Specific software tools allow the automatic generation of two files; namely, the Wirelist (WL) and the Bill of materials (BOM) for each of the KBL representing a wire harness. In the case of the Wirelist, this includes a numbered list of

all the wires present in the harness, along with information such as identification tag, origin and destination node with pin information, total length, cross-section, color, insulation type, and others; all together with the family and module to which each cable belongs. This list is critical because it contains all the connections performed with wires. On the other hand, the BOM file lists electrical components which are not wires, such as fuses, relays, but also elements from the wire harness that have only mechanical functions such as fixings, clips, wire-guides, etc, specifying module and family for each one.

Both WL and BOM documents are excel files and are used without modifications for the pre-processing of the vehicle EDS data. However, a third file denominated QT must be manually designed for the design of a multi-dimensional data structure denominated dataContainer. The QT is an excel file with condensed information from the automatically generated BOM and Wirelist, but complemented with information about each electrical component in the network taken from the WHDP database. Examples of these components are sensors, sources, electronic control units, actuators and loads. As already mentioned, there is a WL and BOM for each wire harness to be simulated. If the simulation aims to calculate the operation of three wire harnesses, there would be 3 WLs and 3 BOMs, respectively. However, a unique QT is designed that merges the information for the different wire harnesses under study. A scheme with the factory data inputs is shown in Figure 4.

The components information is extracted from the databases and include the pin diagram, the type of pin and also the current consumptions under different operational states (inrush current, minimum, normal and maximum regime currents). Each pin can be classified as input power pin, output power pin, ground connection, signal or coil connection. Loads current information are used to assign the input node currents for the MN-BFS algorithm. At the post-processing stage,

a comparison between the calculated branch currents and the nominal maximum cable currents is performed, returning alarms and warnings in case the obtained currents exceed the nominal values.

IV. THE DATACONTAINER STRUCTURE

The files shown in Figure 4 are used to create the structure denominated dataContainer in Matlab. The fields composing the dataContainer are shown in Figure 5 and provide all the required information of the vehicle EDS. In many cases, the dataContainer copies directly the information from the input excel files, while in other occasions, operations are performed to associate together data from the three files to generate the dataContainer tables.

In the group of Wire related fields there are four tables, as seen in Figure 5: WireList, WireMaster, WireModules and

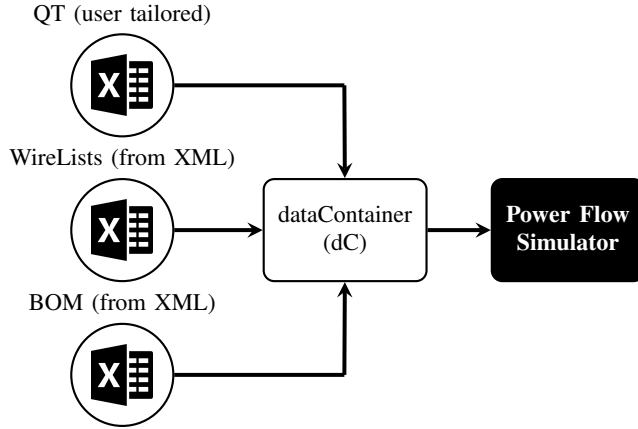


Fig. 4. Factory data inputs.

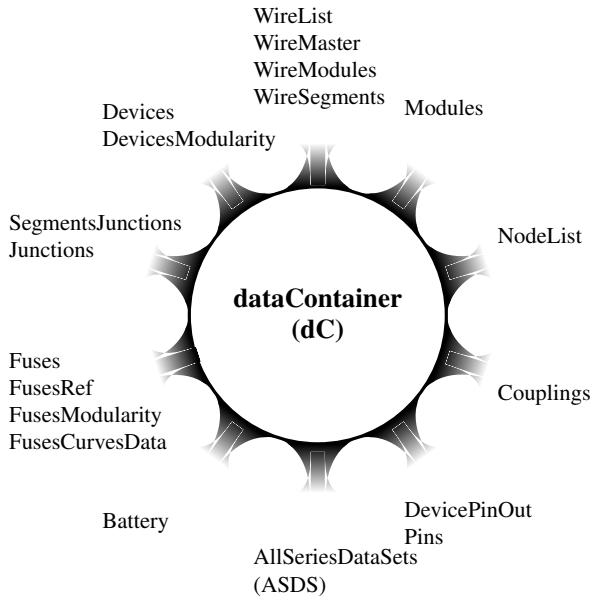


Fig. 5. Fields composing the data container.

Identifier	General	Specific
1	Node	Fuse box
2	Connector	Wire
3	Node	Soldered splice
4	Node	Coupling
5	Node	Device
6	Node	Ground bolt
7	Connector	Component internal
8	Connector	Coupling internal
9	Connector	Fuse
10	Node	Negative terminal
11	Node	Positive terminal
12	Connector	Plate
13	Connector	Relay coil
14	Connector	Relay switch
15	Node	Vehicle's body

TABLE I: Identifiers assigned to the different part types.

WireSegments. The WireList is a condensed version of the one obtained directly from the WHDP. It contains selected information such as: wire id, wire number, parent harness, source node, destination node, part number and length. Within the factory data it is common to have multiple identification tags associated to an element. Examples are the wire id, wire number and partnumber fields of the WireList. WireMaster is a table providing geometrical and materials information. In this file each row contains the wire partnumber together with the conductor material, isolation material, conductor cross section area, isolation cross section area in mm^2 , specific weight (g/m) and specific resistance (Ω/m). The WireModules table contains a list of every single wire that can be present in the harness together with the logistics wire id, the module to which it belongs and the family. In this list there can be rows where the wire id repeats because a wire with a given id can be present in different modules. Likewise, WiresSegments contains a list of all the defined physical segments inside of a harness with their respective identification tags, describing the segments trajectory of each cable. Complementary, the table SegmentJunctions contains a list of each segment but instead of giving information about the parent wires, it provides the identification tags for the geometrical nodes interconnecting the segments inside of the parent harness and the ambient temperature. This is important given that all along a wire there might be segments exposed to different external temperatures (for instance those harnesses installed in the engine compartment are exposed to higher temperatures compared to those installed in the air-conditioner compartment).

The field of Fuses information contain the list of fuses with their identification tags and the specific input an output pins, while the FusesModularity provides the modularity information per fuse. The table of FuseRefs contains the list of product numbers of the existent fuses with their type and current rating. Finally, the fuses information is completed with the table FuseCurvesData which contains the tripping curves.

Table I shows a list of identifiers for the nodes and connectors. Depending on the element to which a node belongs or the type of connector, a different number is assigned. This should not be confused with the unique id that every node or element on the network has. Precisely, for this matter

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	5	7	5	2	10

TABLE II: Example row of an AllSeriesDataSets.types table.

exists the NodeList. It contains a table of all the nodes, id tag, classification according to Table I and specific connection pin. From Table I Nodes of the type 5 are considered Device Nodes. A device or also called component is basically every unit that might consume energy, such as a load, or provide energy for loads such as an ECU. Apart from a part number and id, another identification for the devices is a concept denominated “location”. For a particular device or component there can only exist one location associated.

The table of Devices contains a list of the components with an id and their location, and DevicesModularity contains the information of all elements categorized as components, fuse boxes, battery and couplings, by indicating their modularity, part numbers and location for the case of components. The table Pins contains all the elements categorized as components with their respective pin out and the type of pin. There are 7 defined pin types: signal, component power input, component power output, sink, switch power, switch coil or not connected. In addition, information of current consumption and current peaks is given in this table. For the calculations, as it will be explained in section V, it is necessary to establish a relation of the input pins with correspondent output pins in the components. This is shown in the table DefInOut for every component. The pin out for the battery is given in the table Battery.

The tables Modules and Couplings are simply lists of modules with the families and the couplings with their id, respectively. Finally, the AllSeriesDataSets (ASDS) is a two-table structure containing all the possible current paths from the positive terminal of the battery to the negative terminal. The first table is the ASDS.types and is constructed by having a column with the Node type followed by the Connector type sequentially until it finishes with a column of Node type which is the negative terminal for the battery. In this way, the column connector type describes the type of connector used in between each two nodes. This type of connector, as shown in Table I, can be a wire, a component internal connection, an internal coupling, a fuse, a plate, a relay coil or a relay switch. The second table is the ASDS.ids which in turn contains unique ids of the elements. The id for an element is unique within its type classification. For instance, for the type wires (2), there

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	1	6	3	2

TABLE III: Example row of an AllSeriesDataSets.ids table.

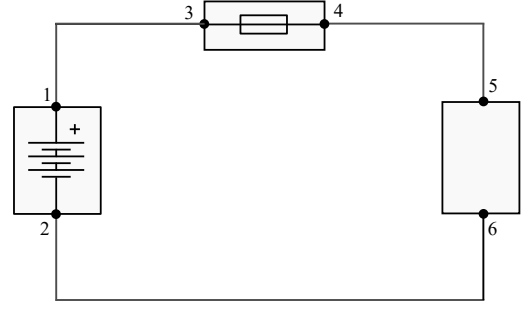


Fig. 6. Circuit represented by Table III.

is only a wire with wire id equal to 1. Similarly, for the type fuses (9) there is only a fuse with a fuse id equal to 1. An example of a row within the ASDS tables is shown in Tables II and III, corresponding to Figure 6.

Algorithm 1 reads the DeviceModularity information within the dataContainer and with this classifies a component or device as an ECU or as a consumer by analysing its pinout. A component is considered an ECU or “source” if and only if it has at least one *component power in* terminal and one *component power out* terminal. On the other hand, a component is allocated as Consumer if it has only pins of the type *Sink* or power inputs but without power output pins. Components containing only pins of the type *Signal* are neglected. Two situations lead to an inconclusive classification: a component without pins of the type *component power in* but at least one terminal *component power out*, and a component with *component power in* pins but none *component power out* or *sink* pins.

The WL and BOM as depicted in Figure 4 contain the information directly from the XML. XML files are designed to contain all the possible modules that exist for a given family within a car model, therefore, the generated dataContainer provides by default the full modules information. In practice, a specific vehicle is described by a selection of only one module per family, thus, its simulation requires filtering of the dataContainer. The way to achieve this is to allow for the user introduction of modules per family in the form of a table, corresponding to the car configuration to be studied. Once this table is introduced together with the dataContainer,

Input: dataContainer as dC

Output: device type

1. From dC.DevicesModularity extract a vector with all devices
 2. Count number of pins of the type 'Signal', 'Power in', 'Power out', 'Sink', 'Switch p', 'Switch coil' per each device
 3. Classify each component as 'Source' or 'Consumer' according with the number of pins of each type and save classification in the vector device type
-

Algorithm 1: Selecting Device Type

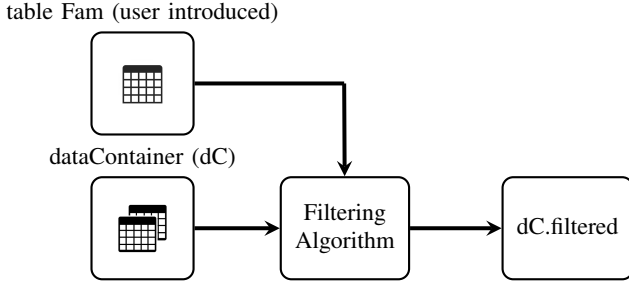


Fig. 7. Filtering process.

a pre-processing algorithm removes unnecessary data. Filtered data is forward used as input for the power flow solver (see Figure 7).

V. POWER FLOW SOLVER

Traditionally, the most used algorithm for solving the power flow problem in conventional electrical systems was the so called Newton-Raphson. For very specific networks, with very specific characteristics, the Backward/Forward Method proved to be faster. In the next subsections, a very brief description of both methods is going to be presented. A detailed analysis of them can be found in [25].

A. Conventional Power Flow Solvers

1) *Newton-Raphson (NR) Method*: The Newton-Raphson method is a derivative-based method in which a vector containing all the voltages amplitudes and angles in the nodes is initialised and updated according to:

$$\begin{bmatrix} \mathbf{V} \\ \boldsymbol{\theta} \end{bmatrix}^{k+1} = \begin{bmatrix} \mathbf{V} \\ \boldsymbol{\theta} \end{bmatrix}^k + \begin{bmatrix} \Delta \mathbf{V} \\ \Delta \boldsymbol{\theta} \end{bmatrix}^k \quad (1)$$

The voltage amplitude and angle increment vectors ($\Delta \mathbf{V}$ and $\Delta \boldsymbol{\theta}$) can be obtained using the Jacobian matrix (\mathbf{J}) as follows:

$$\begin{bmatrix} \Delta \mathbf{V} \\ \Delta \boldsymbol{\theta} \end{bmatrix}^k = \begin{bmatrix} \Delta \mathbf{P} \\ \Delta \mathbf{Q} \end{bmatrix}^k \cdot \left([\mathbf{J}]^k \right)^{-1} \quad (2)$$

The active and the reactive power incrementals ($\Delta \mathbf{P}$ and $\Delta \mathbf{Q}$) represent, respectively, the error between the specified active and reactive powers in the nodes, and the ones calculated with the nodal voltages of the ongoing iteration. More information about the calculation of the Jacobian matrix can be obtained in Chapter 3, Section 3.5 of [25]. The convergence is achieved when all voltage increments are below a given threshold.

2) *Backward-Forward Sweep (BFS)*: The BFS method assumes an initial voltage profile in all the nodes and it calculates the nodal current for each node according to the next expression:

$$I_i^k = \left(\frac{S_i^k}{V_i^k} \right)^* \quad (3)$$

where S represent the apparent power and V the nodal voltage. The subindex i represents the specific node and the superindex k the iteration number. Once the nodal voltages are obtained, the branch currents can be obtained in a backward way using

the Kirchoff Current Law (KCL) according to the following expression:

$$I_{ij}^k = I_j^k + \sum_{m \in \mathcal{J}, m \neq i} I_{jm}^k \quad (4)$$

I_{ij} represents the branch currents connecting the nodes i and j . The branch currents are used to update the node voltages starting from the slack node and sweeping the network in forward direction according to:

$$V_j^{k+1} = V_i^{k+1} - Z_{ij} I_{ij}^k \quad (5)$$

where Z_{ij} represents the impedance between the nodes i and j . Once all the voltages are updated the next step is the calculation of the currents (see eq. (3)) until the convergence is achieved. Convergence is achieved when the difference between the voltage profiles in two successive iterations is below a given threshold.

3) *Performance comparison between (NR) and (BFS)*: BFS method performs better in radial systems in terms of speed. The tests have been done with a fully radial DC system with 200 nodes, half of them loads and the other half generators. The authors generated a battery test of 10^4 random load cases solved in a conventional computer with a processor Intel(R) Core(TM) i7-2670QM CPU @ 2.20GHz and 4GB of RAM. The convergence achieved by both algorithms was 100%. But in terms of speed the BFS algorithm was nearly 8 times faster. In the Figure 8 are shown the amount of solved cases in percentage versus the time invested per case. As it can be observed the BFS algorithm solve 80% of the cases investing less than 80ms per case while the NR algorithm solve 80% of the cases in less than 570ms per case. In Table IV, more details about the convergence speed can be found. The mean time for solving a case for the NR algorithm is 0.563s, while the BFS invests as an average 0.071s. In the table, it can be found also the minimum and the maximum time invested for solving a case for both algorithms.

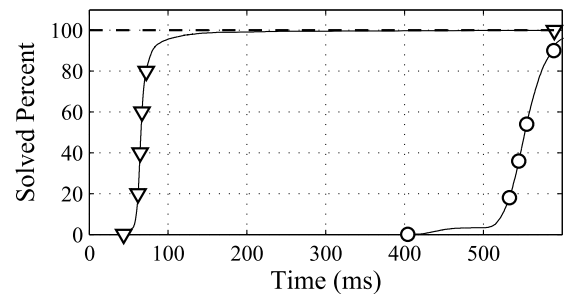


Fig. 8. Comparison of the speed of convergence of the conventional NR and BFS algorithms. \circ NR: Newton-Raphson algorithm. ∇ BFS: Backward/Forward Sweep algorithm.

Selected algorithm	Min. time per case(s)	Max. time per case(s)	Mean. time per case(s)	Achieved converg.(%)
NR	0.403	2.524	0.563	100.0
BFS	0.043	0.916	0.071	100.0

TABLE IV: Comparative behaviour of the original BFS and NR algorithms.

B. Proposed solver

The chosen algorithm to solve the vehicle EDS is the method known as Backward/Forward sweep. This algorithm uses the Kirchhoff current and voltage laws (KCL and KVL) iteratively instead of using the traditional power based formulation of the conventional Newton-based methods. This method has been selected for several reasons; 1) Obtaining the formulation for this specific case of application is much simpler; 2) The speed of convergence of these BFS methods has been proved to be higher than the Newton-based methods in radial or slightly meshed systems [26]. In the specific application of vehicle EDS the system cannot be purely radial, but slightly meshed, hence a Meshed Network BFS must be implemented. This algorithm is based on the conventional BFS, however, it includes minor modifications to deal with the existent loops that will be explained later on this section.

The vehicle EDS can be seen as a two-fold grid: the positive and negative grid. The first one refers to all the nodes and segments connecting the positive terminal of the battery to components and between them, while the latter refers to the nodes and segments that connect to the negative terminal of the battery. An illustration of a simple network with the two sub-grids is seen in Figure 9, showing in red the wires and nodes of the positive grid and in black the ones of the negative grid. The three ground symbols represent ground bolts, interconnected between them through the vehicle body.

The strategy of studying the vehicle EDS as two separated networks is based on the fact that in such networks the path to ground or equivalently, the negative terminal of the battery, represents an intricate network itself. Typically, reaching the negative terminal involves passing through current splices and sequential ground bolts before reaching the main ground bolts in the vehicle body. The selected approach benefits the accuracy of the model. In addition, the MN-BFS adopts only one voltage as the nominal value required. Both the battery positive terminal voltage and negative terminal voltage can be interpreted as the nominal voltages for two different grids. The variable that connects the two networks is the battery current, in other words, the current that flows from the battery positive terminal must be equal to the one that flows into the negative terminal. With this approach, addition of current in the positive grid and negative grid at all branches is always equal to zero. Voltage drops in the network are easily addressed by subtracting node voltages of the respective nodes, ones calculated for the positive grid and the others for the negative grid.

In general, the following is applied: for black boxes behaving as consumers a number of loads appear equal to the number of relationships pin-pout associated to that component. These relationships are clearly stated in the QT file. The loads can be seen as current injections to the nodes where they are connected. A related nodal current injection for each of the grids, the positive and the negative, is defined. In order to guarantee the same total amount of current in the battery terminals, a distribution of current between the nodes need to be done. Considering the case of consumer C2 in Figure 9, it is observed that there is only one relationship (1-2), where 1

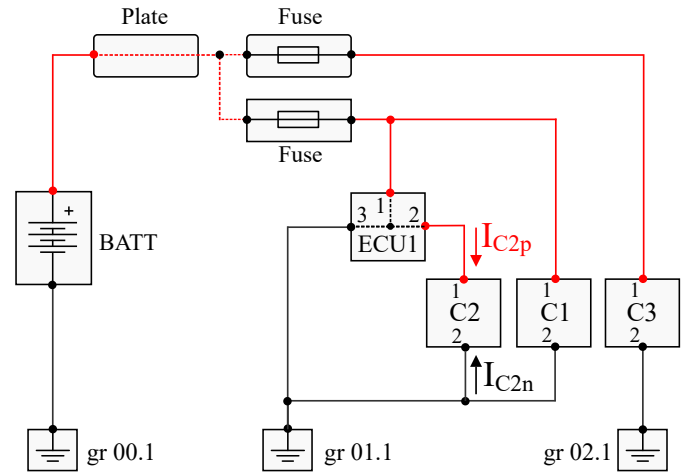


Fig. 9. Example network divided into positive grid (red) and negative grid (black).

is the input and 2 the output pin, therefore the nodal injection current in the node 1 or I_{C2p} (positive grid) and in the node 2 or I_{C2n} (negative grid) are equal. However, it would be possible to find two relationships inside of a component. That were the case for a C2 consumer with 3 pins, having for instance the pin relations (1-2) and (1-3), being pin 3 another output pin. In such an event, the current injections assigned to the nodes 2 and 3 (both belonging to the negative grid) were exactly the half of the current associated to the input pin 1. In the case of components classified as ECUs the procedure is exactly the other way around: the current associated to a ground pin is split among its associated input pins. Back to Figure 9, as seen for ECU1, pin 1 is the only input pin linked to the ground pin 3, therefore there is only one current load per each of the grids inside of this component, equal to the current associated to the pin 3.

Algorithm 2 also denominated *Voltage Constructor* executes several tasks. It goes line by line through all of the possible current paths between the positive and negative terminal of the battery, taking into consideration that each odd column is a node and each even column is a connection. The tasks performed are, in order: classification of the elements as consumer or source using Algorithm 1, determination of the interface between the positive and the negative grid (stop node), calculation of the currents to inject in the stop node and neighbors or node current injection vector I_{np} , formation of the incidence matrix Γ_p , branch resistances matrix R_{bp} , voltage node id vector V_{np} for the positive grid and their equivalent for the negative grid Γ_n , R_{bn} , I_{ng} , V_{ng} . Finally, the algorithm creates a pairing matrix which contains the information about the terminals of the components where it is required to obtain their differential voltage (between the inputs pins and output-grounded pins). For the example network in Figure 9 one possible current path goes from the positive of the battery passing through ECU1, C2 and ending in the negative of the battery through ground bolt gr01.1. The stop node for this current path would be pin 2 of C2.

A set of rules are followed to determine the nodal injection

currents I_n . Namely:

- 1) if in a row there is only one element categorized as ‘Consumer’ through two nodes, the current of the input node belonging to the ‘positive grid’ is assigned according to the modules selected, and the current of the output node belonging to the ‘negative grid’ is equal to that of the input node divided over all its related output pins (ignoring the signals)
- 2) when an output pin is associated to more than one input node, then its current is calculated by adding up the currents that result from following Rule 1 for each of the input nodes
- 3) the current of an input node is assigned only once. If the same input node appears in different rows forming new node pairs, the value of input node current will remain the first assigned while the second node gets the current of the input divided over all its related output pins (ignoring pins carrying communication signals)
- 4) for 2 or more consumers connected in series within a single row the minimum current among them is assigned to the input node of the consumer which is connected to ground
- 5) if in a row there is only one element categorized as ‘Source’ through two nodes, the current of the output node belonging to the ‘negative grid’ is assigned according to the modules selected, and the current of the input node belonging to the ‘positive grid’ is equal to that of the output node divided over all its related input pins (ignoring the signals)
- 6) when an input pin is associated to more than one output node, then its current is calculated by adding up the currents that result from following Rule 5 for each of the output nodes. Exactly the opposite to the case of the consumers

Figures 10 and 11 are graphical representations of the aforementioned rules.

The incidence matrix Γ provides the connections data, where each row represents a branch of the system and each

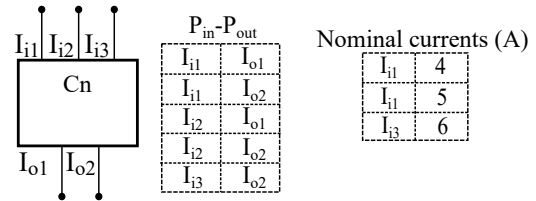
Input: dataContainer as dC with dC.ASDS Filtered

Output: V_{np} , I_{np} , Γ_p , R_{bp} , V_{ng} , Γ_n , R_{bg} , Pairing matrix

1. pathnumber=lengthrows(dC.ASDS.types)
2. **for** currentpath=1:1:pathnumber **do**
3. *Create device type vector using Algorithm 1*
4. *Find position of the input node of the last consumer connected to ground or Stop node*
5. *Assign injection currents to nodes according to the Rules 1 to 6*
6. *Create incidence matrix I_{ncp}*
7. *Calculate branch resistances R_{bp}*
8. *Repeat for the negative grid to get V_{ng} , I_{ncg} , R_{bg}*
9. *Create the node pairs matrix or Pairing matrix*
10. **end for**

Algorithm 2: Voltage Constructor

Positive grid



Negative grid

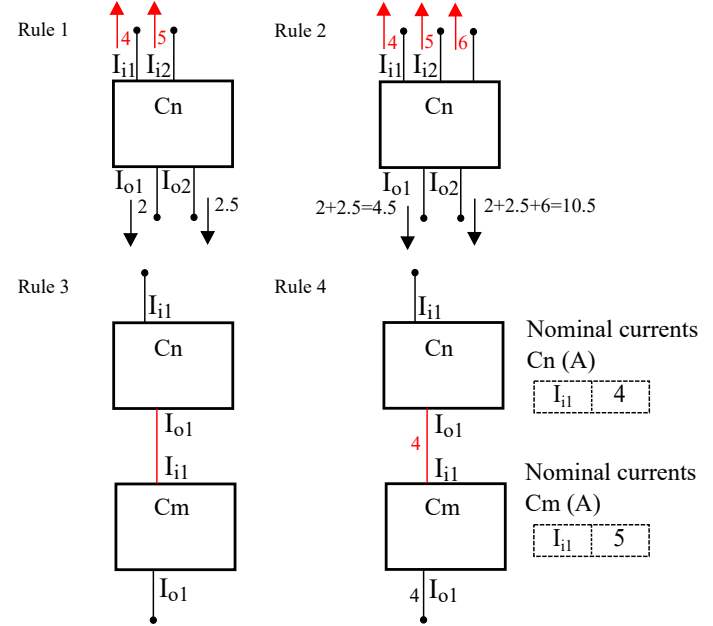
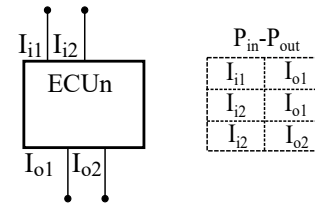


Fig. 10. Representation of nodal injections current rules for the case of consumers.

Positive grid



Negative grid

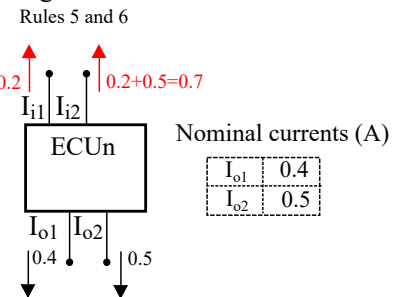


Fig. 11. Representation of nodal injections current rules for the case of ECUs.

Input: Γ

Output: l_c, l_{nc}

1. **for** nodes=1:1:lengthcolumns(Γ) **do**
 2. **for** currentpath=1:1:lengththrows(Γ) **do**
 3. *Detect loops*
 4. *Save to l_c the lines to be removed*
 5. *Save to l_{nc} the lines to be kept*
 6. **end for**
 7. **end for**
-

Algorithm 3: Detection of loops

column a node. The construction of Γ is done following the next rules:

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

The KCL and KVL can be expressed in matrix form using the incidence matrix, respectively:

$$\Gamma^T \cdot \mathbf{I}_B^T + \mathbf{I}_d \cdot \mathbf{I}_n^T = 0 \quad (6)$$

$$\Gamma^T \cdot \mathbf{V}^T - \mathbf{R}_b \cdot \mathbf{I}_B^T = 0 \quad (7)$$

where the newly introduced parameters are \mathbf{I}_B which is a vector containing all the branch currents and \mathbf{I}_d which is the identity matrix.

To calculate the branch resistances, the algorithm uses information of wire length between nodes given in the dataContainer, together with the selected conductor resistivity. The calculated resistance takes into account effects of temperature by means of the linear approximation of resistivity with temperature shown in equation 8, where T is the ambient temperature and α is the temperature coefficient of resistivity in K^{-1} .

$$R(T) = R(T_0) * [1 + \alpha(T - T_0)] \quad (8)$$

Before applying the algorithm Meshed-Network Back and Forward Sweep it is necessary to detect and eliminate the branches which define loops in both positive and negative grid. A straight forward strategy to do it is by analyzing the incidence matrix: a node having associated more than one value of -1 is receiving lines which provoke cycles or loops in the system. Branches are eliminated following Algorithm 3.

Once the system is radial, a conventional BFS algorithm can be used for solving the problem, afterward, a compensation algorithm must be applied in order to estimate the currents in the cut branches. The compensation algorithm uses the Thevenin equivalent resistance in the cut branches. A very detailed explanation of the electric principle used for making this Thevenin calculation can be found in [26]. Basically it can be stated that connecting fictitious unitary current sources between the nodes of the cut branches, the voltage obtained between these nodes are the Thevenin equivalent resistance of the circuit from the point of view of a given pair of nodes.

Input: $\Gamma, R_B, \mathbf{I}_n, l_c, l_{nc}, V_{slack}$

Output: \mathbf{V}, \mathbf{I}_B

1. *Init. all \mathbf{I}_B to zero and all \mathbf{V} to 1*
 2. *Set $V(1)$ to V_{slack}*
 3. *Calculate Thevenin resistance of cut branches (R_{th}) using fictitious unitary current sources [26]*
 4. *Calculate the total branch resistance of cut branches R_B^{Total}*
 5. **for** i=1:1:max number of iterations **do**
 6. *Calculate all currents of non-cut branches $\mathbf{I}_B(l_{nc})$*
 7. *Calculate network voltages, except the slack $V(2:N)$*
 8. *Calculate the voltage drop in the cut-branches using the previous voltages and the Ohm's law*
 9. **if** Calculated voltage drops are consistent **then**
 10. *break*
 11. **else**
 12. *Calculate the new branch currents in the cut-branches using the new voltage profile*
 13. **end if**
 14. **end for**
-

Algorithm 4: Meshed Network Back and Forward Sweep (MN-BFS) Solver

Having calculated the Thevenin resistance of the cut branches the compensation algorithm can be embedded in the radial power flow algorithm as observed in Algorithm 4. The algorithm starts with the initialization of all branch currents to zero and all voltages to one, except the slack voltage (battery) that is set to its predefined value. After that, all Thevenin resistances of the cut-branches are calculated using the unitary current sequential injection method. It must be pointed out that the Thevenin resistances are calculated just once and they don't have to be updated through the iterative process. Subsequently, we launch the iterative process, starting with the calculation of the branch currents in the non-cut branches $\mathbf{I}_B(l_{nc})$. This calculation is direct from the nodal currents \mathbf{I}_n , already assigned according to the six rules previously stated, using the matrix Γ . By means of the branch currents and the resistances of the non-cut branches we can update the value of the voltages of all nodes (except the slack one). At this stage, the voltage drop in the cut branches can be calculated using two different methods. In one side, since we have all voltages in the network, we can use them to obtain the voltage drop directly in the cut branches. The other approach is the use of the branch currents and the resistance of the cut branches to calculate the voltage drop. Both calculations must match when the algorithm convergence is achieved. If not, the second approach is used to update the branch currents in the cut branches and a new iteration is launched. It has been demonstrated that this method is quite simple and yet, quite robust and effective to solve this kind of systems.

To illustrate the power flow solver described in Algorithm 4, let us consider the sample network in Figure 12. A given resistance value is only considered for the wires where it is clearly stated. For the fuses, a resistance value is assigned. The

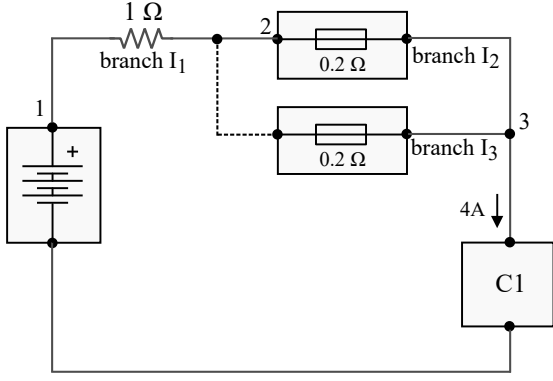


Fig. 12. Sample network to describe the power flow algorithm.

circuit is composed by a battery feeding a consumer C1 through a couple of fuses. The nominal battery voltage (slack voltage) considered is 14V and the consumer nodal injection current is 4A. There are three main branches identified: l_1 connecting the battery to the fuses, l_2 corresponding to one of the fuses and l_3 to the second parallel fuse. The incidence matrix Γ together with the nodal current injection vector I_n and the branch resistances R_B are considered inputs for the Algorithm 4, obtained in previous steps. The branch selected to be cut is branch l_3 .

$$\Gamma = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ 0 & 1 & -1 \end{bmatrix}, \quad I_n^T = \begin{bmatrix} 0 \\ 0 \\ 4 \end{bmatrix} A, \quad R_B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.2 & 0 \\ 0 & 0 & 0.2 \end{bmatrix} \Omega$$

$$l_c = [l_3] \quad (9)$$

$$l_{nc} = [l_1, l_2] \quad (10)$$

After applying steps 1 and 2, the Thevenin resistance of the cut branch l_3 is calculated following step 3, using the Equation 7 in the form:

$$\mathbf{V}^T(2:n_N) = -(\Gamma(l_{nc}, 2:n_N))^{-1} \cdot \mathbf{R}_B(l_{nc}, l_{nc}) \cdot \Gamma^T(2:n_N, l_{nc})^{-1} \cdot \Gamma^T(2:n_N, l_c) \cdot \mathbf{I}_B^T(l_{nc}) \quad (11)$$

where n_N is the number of nodes, equal to 3 for the sample network in Figure 12.

$$\mathbf{V}^T(2:3) = - \begin{bmatrix} -1 & 0 \\ 1 & -1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} 1 & 0 \\ 0 & 0.2 \end{bmatrix} \cdot \begin{bmatrix} -1 & 1 \\ 0 & -1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} 1 \\ -1 \end{bmatrix} \cdot 1 = \begin{bmatrix} 0 \\ -0.2 \end{bmatrix} V$$

The Thevenin resistance of the cut branch is calculated according to:

$$R_B^{TH}(:, i) = \Gamma(l_c, 2:n_N) \cdot \mathbf{V}^T(2:n_N) \quad (12)$$

$$R_B^{TH}(l_3) = [1 \quad -1] \cdot \begin{bmatrix} 0 \\ -0.2 \end{bmatrix} = 0.2\Omega$$

In step 4, the total branch resistance of l_3 is calculated:

$$R_B^{Total}(l_c) = R_B^{TH} + R_B(l_c, l_c) \quad (13)$$

$$R_B^{Total}(l_3) = 0.2\Omega + 0.2\Omega = 0.4\Omega$$

At this point, following step 5, the iterations start. In the first iteration, steps 6 and 7 of Algorithm 4 are as follows, considering $I_B(l_c) = 0$ from step 1:

$$\mathbf{I}_B^T(l_{nc}) = \Gamma^T(2:n_N, l_{nc})^{-1} \cdot (-\Gamma^T(2:n_N, l_c) \cdot \mathbf{I}_B^T(l_c) - \mathbf{I}_n^T(2:n_N)) \quad (14)$$

$$\mathbf{I}_B^T(l_1:l_2) = \begin{bmatrix} -1 & 1 \\ 0 & -1 \end{bmatrix}^{-1} \cdot - \begin{bmatrix} 0 \\ 4 \end{bmatrix} = \begin{bmatrix} 4 \\ 4 \end{bmatrix} A$$

$$\mathbf{V}^T(2:n_N) = (\Gamma(l_{nc}, 2:n_N))^{-1} \cdot \left((-\Gamma(:, 1) \cdot V(1) + \mathbf{R}_B(l_{nc}, l_{nc}) \cdot \mathbf{I}_B^T(l_{nc})) \right) \quad (15)$$

$$\mathbf{V}^T(2:n_N) = \begin{bmatrix} -1 & 0 \\ 1 & -1 \end{bmatrix}^{-1} \cdot \left(\begin{bmatrix} 1 \\ 0 \end{bmatrix} \cdot 14 + \begin{bmatrix} 1 & 0 \\ 0 & 0.2 \end{bmatrix} \cdot \begin{bmatrix} 4 \\ 4 \end{bmatrix} \right)$$

$$\mathbf{V}^T(2:n_N) = \begin{bmatrix} 10 \\ 9.2 \end{bmatrix} V$$

The stop criteria makes use of the KVL in the cut branches. The voltage drops are calculated using two methods as indicated in step 8 and explained before:

$$\|\Gamma(l_c, :) \cdot \mathbf{V} - \mathbf{R}(l_c, l_c) \cdot \mathbf{I}_B(l_c)\| \leq \epsilon \quad (16)$$

For the current example, the algorithm is considered convergent when calculating a voltage drop error value equal or below 0.01V. This condition is evaluated as stated in step 9.

$$\|0.2 \cdot 0 - [0 \quad 1 \quad -1] \cdot \begin{bmatrix} 14 \\ 10 \\ 9.2 \end{bmatrix}\| \leq 0.01V$$

In this first iteration the criteria in Equation V-B is false. Then, the current in the cut branch is updated, following step 12.

$$\Delta \mathbf{I}_B^T(l_c) = (R_B^{Total})^{-1} \cdot \Gamma(l_c, :) \cdot \mathbf{V}_0^T \quad (17)$$

$$\Delta \mathbf{I}_B^T(l_3) = (0.4)^{-1} \cdot [0 \quad 1 \quad -1] \cdot \begin{bmatrix} 14 \\ 10 \\ 9.2 \end{bmatrix}$$

$$\Delta \mathbf{I}_B^T(l_3) = 2A$$

$$\mathbf{I}_B^T(l_3) = 2A$$

A second iteration starts.

$$\mathbf{I}_B^T(l_1 : l_2) = \begin{bmatrix} -1 & 1 \\ 0 & -1 \end{bmatrix}^{-1} \cdot - \begin{bmatrix} 1 \\ -1 \end{bmatrix} \cdot 2 - \begin{bmatrix} 0 \\ 4 \end{bmatrix} = \begin{bmatrix} 4 \\ 2 \end{bmatrix} A$$

$$\mathbf{V}^T(2 : n_N) = \begin{bmatrix} -1 & 0 \\ 1 & -1 \end{bmatrix}^{-1} \cdot \left(- \begin{bmatrix} 1 \\ 0 \end{bmatrix} \cdot 14 + \begin{bmatrix} 1 & 0 \\ 0 & 0.2 \end{bmatrix} \cdot \begin{bmatrix} 4 \\ 2 \end{bmatrix} \right)$$

$$\mathbf{V}^T(2 : n_N) = \begin{bmatrix} 10 \\ 9.6 \end{bmatrix} V$$

$$\|0.2 \cdot 2 - [0 \ 1 \ -1] \cdot \begin{bmatrix} 14 \\ 10 \\ 9.6 \end{bmatrix}\| \leq 0.01V$$

For this iteration the criteria in Equation V-B is true. Then, it is considered that the algorithm has successfully converged. The obtained values are:

$$\mathbf{V}_n^T = \begin{bmatrix} 14 \\ 10 \\ 9.6 \end{bmatrix} V, \quad \mathbf{I}_B^T = \begin{bmatrix} 4 \\ 2 \\ 2 \end{bmatrix} A \quad (18)$$

VI. CASES OF STUDY

The previously described methodology has been applied to different scenarios of a sample vehicle network after introducing a specific modularity information. In these examples, we have considered the case of two families, where each of these two families has three electable modules each, as shown in Table V.

Family	Modules
family 01	mod01 fam01
	mod02 fam01
	mod03 fam01
family 02	mod01 fam02
	mod02 fam02
	mod03 fam02

TABLE V: List of families with their electable modules.

The selection of modularity is inserted in the algorithm in the form of a table denominated 'Fam' (see Figure 7). Table VI shows an example of modularity selection ('Fam'), where for each of the two families studied, there is a single module specified. To configure a full vehicle electrical network, there should be a module selection for more than 50 families.

family 01	family 02
mod01 fam01	mod03 fam02

TABLE VI: Example of Table 'Fam' for Sample network 1.

Sample network 1 in Figure 13 corresponds to the selection depicted in Table VI. This electrical network contains a 14V battery connected to the vehicle ground as the feeder. The node A1 represents a connection to a metal plate where two fuses (F1A and F2A) and a single connection (C1) distribute the current. The fuse identified with node F3 is not connected to the common plate and receives the current through C1.

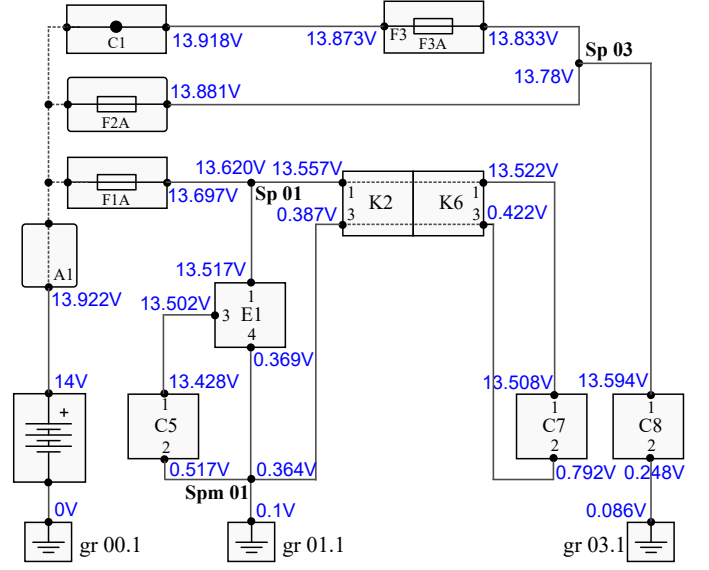


Fig. 13. Calculated node voltages for Sample network 1.

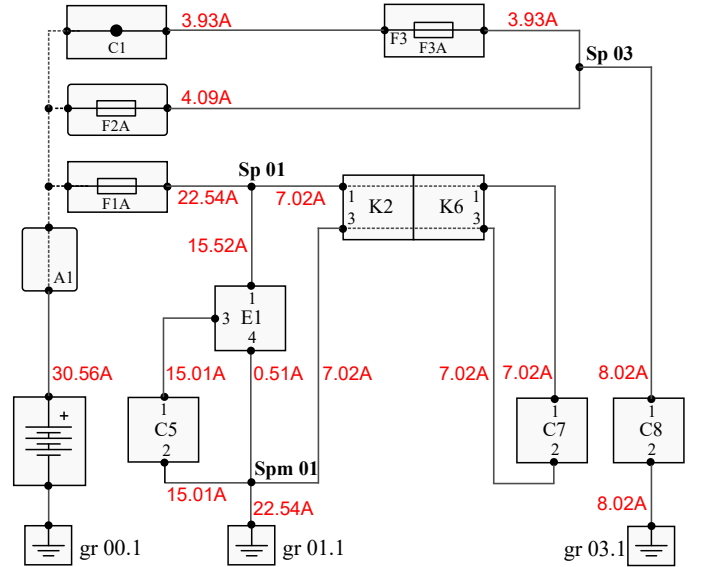


Fig. 14. Calculated branch currents for Sample network 1.

The nodes identified with the prefix *Sp* are splices nodes. Splices represent ultrasonic soldered connections of multiple wires. On the other hand, consumers have been named with the prefix *C* and couplings have been named with the prefix *K* followed by a number. Electronic control units are named ECU followed by a numeration. The reason the numbers are not exactly consecutive is to represent the fact that given the modules selected, certain components have been filtered out. The type of component (whether a consumer or source) is defined using Algorithm 1.

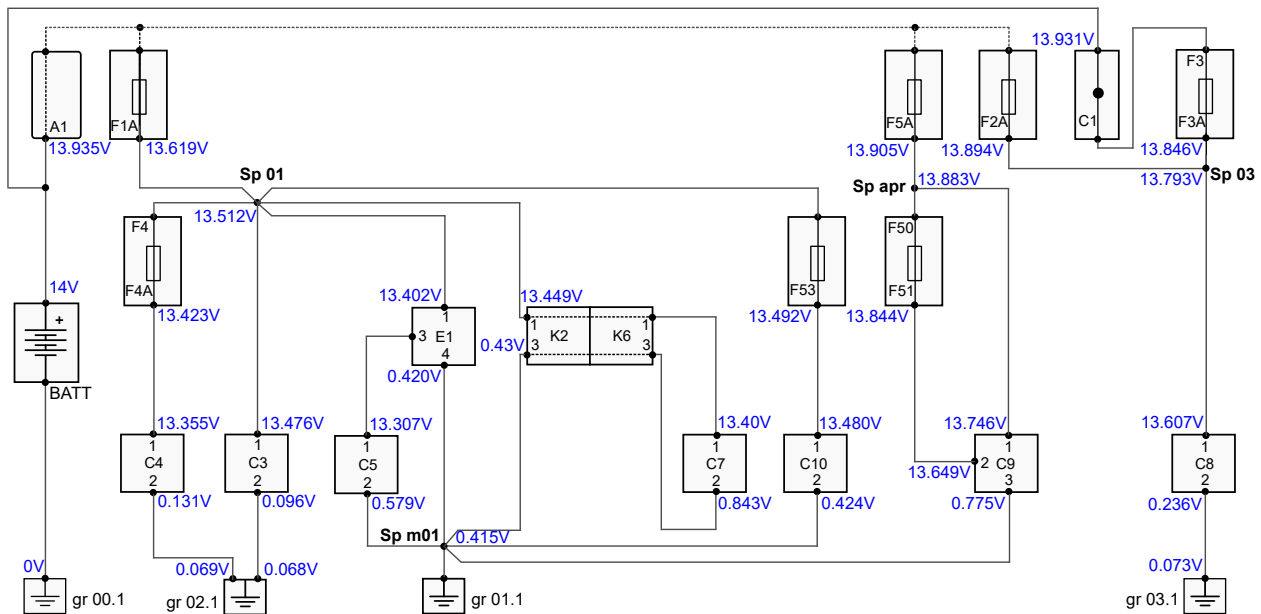


Fig. 15. Calculated node voltages for Sample network 2.

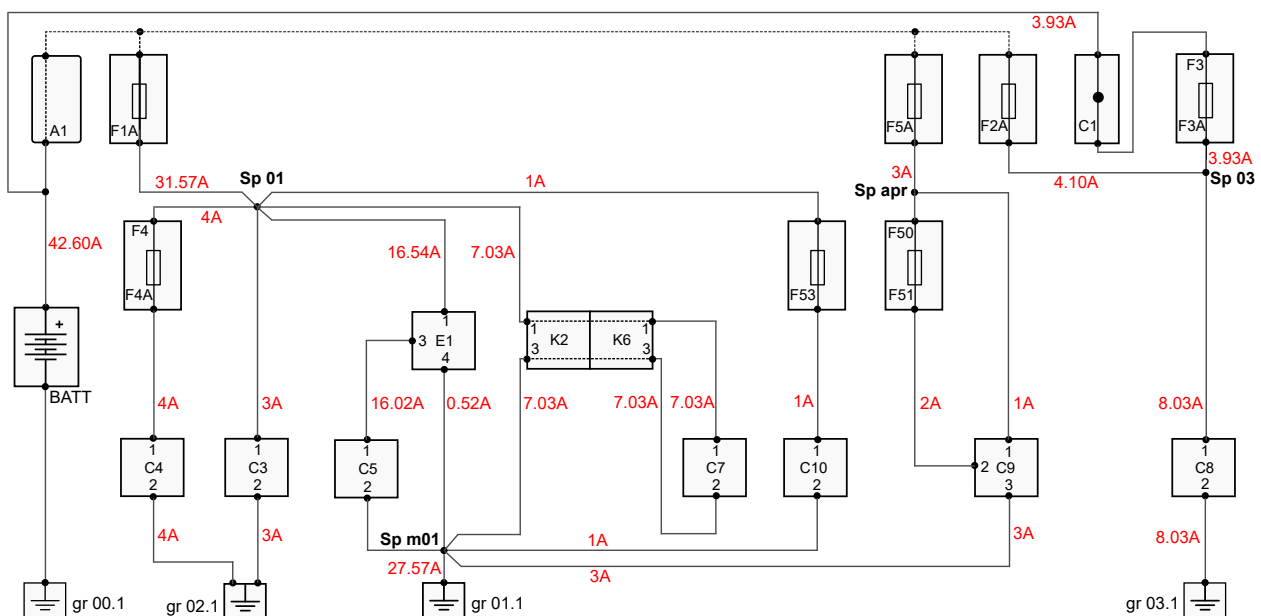


Fig. 16. Calculated branch currents for Sample network 2.

Component ECU1 is an electronic control unit, therefore classified as 'Source'. It has 4 pins where pin 1 and 4 are power inputs and pin 3 is a power output. Pin 2 is a signal pin and thus neglected. The numbered grounds represent bolt ground points located in different places of the vehicle body and connected through the vehicle body itself. The battery ground or negative terminal is identified with the code gr00.1.

Figures 13 and 14 show the calculated node voltages and branch currents of Sample network 1, respectively. Table VII gives a list of the components on both sample networks.

Voltage drops are observed in the lines following the calculated line resistances given their physical characteristics.

Code	Description
E1	ECU/Source
K2	Coupling
C3	Consumer
C4	Consumer
C5	Consumer
K6	Coupling
C7	Consumer
C8	Consumer
C9	Consumer
C10	Consumer

TABLE VII: Components list

Nominal node voltages decrease following the power flow as expected. In addition, calculated branch currents are consistent all along the circuit.

family 01	family 02
mod03 fam01	mod03 fam02

TABLE VIII: Example of Table 'Fam' for Sample network 2

Sample network 2 corresponds to a different modularity selection, as observed in Table VIII. In this case, the network is formed by a higher number of consumers, thus increasing also the amount of fuses and splices present in the network. Voltage drops are observed along the wires and fuses. The amount of voltage drop in the fuses is dependent on their type. As to sample network 1, the obtained power flow all along the network remains coherent, as observed in Figures 15 and 16.

Usually different parts of the EDS are fed by the battery or a different set of ECUs that split the whole network in different subsystems that can be simulated and studied independently.

VII. CONCLUSION

The high complexity found in the wiring system of modern vehicles translates to several aspects such as EDS development, assembling, managing of spare parts and also documenting. To reach the level of vehicle EDS simulation is therefore mandatory to understand the way in which the network information is presented and be able to extract it and transform it.

The main contributions of this work are:

- A detailed methodology to take and adapt the vehicle EDS factory data into suitable and standard matrices that serve as inputs for a power flow solver
- The description and implementation of the Meshed Network Backward and Forward sweep algorithm, which for systems such as vehicle EDS where the network is not purely radial, performs much better compared to Newton-based methods. This method includes a minor modification on top of the traditional BFS, that allows it to be used in weakly meshed networks such as the vehicle EDS.
- The introduction of a low computational-costly method, since the required matrix inversions are calculated only once and the matrices are highly sparse, representing a low consumption of computer memory and computation time. This method is ideal for radial systems and its high speed is appropriate for future new developments where probabilistic power flows are needed and the speed becomes a key factor.
- The usability of a method able to detect faults in the vehicle EDS such as excessive voltage drops or overcurrents, that is applicable to multiple EDS configurations far beyond of what is feasible with real prototypes. The applicability of the methodology has been demonstrated by selecting different configurations and verifying the consistency of the calculated electrical variables.

REFERENCES

- [1] Bosch, "Power net simulation," 2019 (accessed 05-August-2019).
- [2] SYNOPSIS, "Saber for automotive systems," <https://www.synopsys.com/verification/virtual-prototyping/saber/saber-automotive-systems.html>, 2019 (accessed 05-August-2019).
- [3] COMSA, "Bordnetz-design mit Idorado," <https://www.comsa.de/>, 2019 (accessed 05-August-2019).
- [4] Engineering Base Cable, "Aucotec product portfolio," <https://www.aucotec.com/en/solutions/engineering-base.html>, 2019 (accessed 05-August-2019).
- [5] VDA, "Harness description list (kbl)," <http://www.w3.org/TR/xml>, 2015 (accessed 05-August-2019).
- [6] M. Petit and P. Dessante, "Modeling and experimental set-up of an automotive electrical network for transient studies," *19th International Conference on Electrical Machines, ICEM 2010*, pp. 1–6, 2010.
- [7] D. Bilyi and D. Gerling, "Modeling and design simulation of vehicle power network in Simulink/MATLAB," *2016 IEEE Smart Energy Grid Engineering (SEGE)*, no. 1, pp. 196–202, 2016.
- [8] F. Ruf, M. Schill, A. Barthels, T. P. Kohler, H. U. Michel, J. Froeschl, and H. G. Herzog, "Topology and design optimization of a 14 v automotive power net using a modified discrete pso in a physical simulation," *2013 9th IEEE Vehicle Power and Propulsion Conference, IEEE VPPC 2013*, pp. 469–475, 2013.
- [9] K. Gorelic, A. Kilik, and R. Obermaisser, "Modeling and simulation of optimal and adaptive real-time energy management system for automated driving," *2017 IEEE Transportation and Electrification Conference and Expo, ITEC 2017*, pp. 356–363, 2017.
- [10] M. Diebig, S. Frei, H. Reitingger, and C. Ullrich, "Modeling of the automotive power supply network with VHDL-AMS," *2010 IEEE Vehicle Power and Propulsion Conference, VPPC 2010*, 2010.
- [11] A. Rius-Rueda, A. Garcia, and M. A. Diaz, "Custom integer optimization method for wire bundle dimensioning," *IECON Proceedings (Industrial Electronics Conference)*, pp. 4389–4394, 2016.
- [12] R. A. Jabr and I. Džafić, "Solution of DC railway traction power flow systems including limited network receptivity," *IEEE Transactions on Power Systems*, vol. 33, no. 1, pp. 962–969, Jan 2018.
- [13] B. Mohamed, P. Arbolea, and C. Gonzalez-Moran, "Modified current injection method for power flow analysis in heavy-meshed DC railway networks with nonreversible substations," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 9, pp. 7688–7696, Sept 2017.
- [14] P. Saenger, N. Devillers, K. Deschinkel, M. C. Pera, R. Couturier, and F. Gustin, "Optimization of electrical energy storage system sizing for an accurate energy management in an aircraft," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 7, pp. 5572–5583, July 2017.
- [15] H. Nademi, R. Burgos, and Z. Soghomonian, "Power quality characteristics of a multilevel current source with optimal predictive scheme from more-electric-aircraft perspective," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 1, pp. 160–170, Jan 2018.
- [16] L. Herrera, W. Zhang, and J. Wang, "Stability analysis and controller design of dc microgrids with constant power loads," *IEEE Transactions on Smart Grid*, vol. 8, no. 2, pp. 881–888, March 2017.
- [17] J. M. Crider and S. D. Sudhoff, "Reducing impact of pulsed power loads on microgrid power systems," *IEEE Transactions on Smart Grid*, vol. 1, no. 3, pp. 270–277, Dec 2010.
- [18] G. Sulligoi, A. Vicenzutti, V. Arcidiacono, and Y. Khersonsky, "Voltage stability in large marine-integrated electrical and electronic power systems," *IEEE Transactions on Industry Applications*, vol. 52, no. 4, pp. 3584–3594, July 2016.
- [19] Q. Xu, B. Yang, Q. Han, Y. Yuan, C. Chen, and X. Guan, "Optimal power management for failure mode of mvdc microgrids in all-electric ships," *IEEE Transactions on Power Systems*, vol. 34, no. 2, pp. 1054–1067, March 2019.
- [20] S. Fang, Y. Xu, Z. Li, T. Zhao, and H. Wang, "Two-step multi-objective management of hybrid energy storage system in all-electric ship microgrids," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 4, pp. 3361–3373, April 2019.
- [21] S. Fang, Y. Wang, B. Gou, and Y. Xu, "Towards future green maritime transportation: An overview of seaport microgrids and all-electric ships," *IEEE Transactions on Vehicular Technology*, pp. 1–1, 2019.
- [22] S. Fang and Y. Xu, "Multi-objective robust energy management for all-electric shipboard microgrid under uncertain wind and wave," *International Journal of Electrical Power Energy Systems*, vol. 117, p. 105600, 2020. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0142061518336238>

- [23] E. H. Trinklein, M. D. Cook, G. G. Parker, and W. W. Weaver, "Exergy optimal multi-physics aircraft microgrid control architecture," *International Journal of Electrical Power Energy Systems*, vol. 114, p. 105403, 2020. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0142061518330874>
- [24] W3C, "Extensible markup language (xml) 1.0 (fifth edition)," <http://www.w3.org/TR/REC-xml/>, 2008 (accessed 05-August-2019).
- [25] A. Gomez-Exposito, A. Conejo, and C. Canizares, *Electric Energy Systems: Analysis and Operation*, ser. Electric Power Engineering Series. CRC Press, 2017.
- [26] P. Arboleya, B. Mohamed, C. Gonzalez-Moran, and I. El-Sayed, "BFS algorithm for voltage-constrained meshed DC traction networks with nonsmooth voltage-dependent loads and generators," *IEEE Transactions on Power Systems*, vol. 31, no. 2, pp. 1526–1536, March 2016.

Pablo Arboleya (SM'13) received the M.Sc. and Ph.D. (with distinction) degrees from the University of Oviedo, Gijón, Spain, in 2001 and 2005, respectively, both in Electrical Engineering. Nowadays, he works as Associate Professor in the Department of Electrical Engineering at the University of Oviedo, he is Managing Editor of the International Journal of Electrical Power and Energy Systems and holder of the Gijón Smart Cities Chair at the University of Oviedo. Presently his main research interests are focused in the micro-grid and smart-grid modeling and operation techniques, Internet of the Energy applications, railway traction networks simulation and combined AC/DC power flow algorithms.

Paola Mantilla-Perez received the B.Sc. degree in electronics engineering at the Universidad del Norte, Colombia in 2007 and the M.Sc. degree in optics and photonics from the Karlsruhe Institute of Technology (KIT), Germany, in 2012. She obtained a Ph.D in Photonics for her research applied to solar energy, from the Universitat Politècnica de Catalunya (UPC) Spain in 2017. Currently she is working at SEAT S.A. while pursuing an industrial Ph.D in Energy and process control at the University of Oviedo, Spain, in cooperation with the spanish automotive producer. Her current areas of interest are modeling and simulation of DC automotive vehicle power networks, automotive power networks for autonomous vehicles and integration of renewable energies in transportation systems.

Juan-Andrés Pérez-Rúa received the B.Sc. degree in electrical engineering from the Technological University of Bolivar, Colombia, with Summa Cum Laude distinction in 2012, and the M.Sc. degree in Sustainable Transportation and Electrical Power Systems from the Superior Institute of Engineering from Coimbra, Portugal, the University of Nottingham, England and the University of Oviedo, Spain, in 2016. Currently he is pursuing the Ph.D. degree in the Department of Wind Energy in the Technical University of Denmark. From 2012 to 2014 he worked with HMV Engineers in Colombia as a design engineer of electrical protection systems and primary design of substations and transmission lines in national and international projects. His present-day areas of interest include renewable energies, optimization techniques applied to electrical systems of offshore wind energy, dynamic lines rating and control.

Manuel-Alberto Díaz Millán is the Head of the On-Board Electrical Network Development Department within SEAT S.A. He has more than 20 years of engineering and management experience within the electrical distribution systems of vehicles. His technical expertise include the vehicle EDS design from the concept to constructional aspects, vehicle wire harness manufacturing technologies, risks analysis, saving strategies and reliability improvement.

Xavier Dominguez received the Engineering Degree in Electronics and Control at the Escuela Politecnica Nacional-Ecuador in 2010. Thanks to an Ecuadorian scholarship (Senescyt), in 2014 he attained a M.Sc. degree in Electrical Energy Conversion and Power Systems at the University of Oviedo-Spain, where he is nowadays pursuing an industrial Ph.D in Energy and Process Control related to a research project supported by the automotive Spanish manufacturer SEAT S.A. His current research areas are modeling, simulation and visualization of on-board electrical networks of vehicles, smartgrids and renewable energy integration.