# BFS Algorithm for Voltage Constrained Meshed DC Traction Networks with Non-Smooth Voltage-Dependent Loads and Generators

Pablo Arboleya, *Senior Member, IEEE,* Bassam Mohamed, Cristina González-Morán, *Senior Member, IEEE,* and Islam El-Sayed

*Abstract*—**In this paper a new procedure based on a Backward/forward sweep (BFS) algorithm for solving power flows in weakly meshed DC traction networks is presented. The proposed technique is able to consider the trains as non-linear and non-smooth (non-differentiable) voltage dependent loads or generators. This feature permits the inclusion of the trains overcurrent protection and the squeeze control. With the use of the mentioned controls, the conventional power flow problem becomes a voltage constrained power flow problem, and the interaction between the trains and the network can be accurately modeled. However, the train control induces a highly non-smooth voltage dependent load characteristics, causing convergence problems in most of the derivative based algorithms. The proposed algorithm is faster, more robust and stable than the derivative based ones. In addition, the authors present all the formulation in a compact matrix based form by means of the graph theory application and the node incident matrix.**

*Index Terms*—**Backward/forward sweep (BFS) algorithm, distribution system modeling, DC traction networks, railway systems, voltage dependent load modeling.**

## I. INTRODUCTION

THE use of power flow techniques applied to DC traction systems simulation is widely extended for planning or operation purposes [1]–[7]. Some of these works consider the DC system and also the AC system feeding the DC one [1], [2], [5]–[8] and others consider only the DC subsystem [4], [9]. The former provide a more accurate solution because the effect of the AC subsystem is considered, but the latter are faster, and for most applications they have enough accuracy. As it is stated in [4], some of these works solve the system of equations using direct approaches based on Gaussian elimination, Cholesky decomposition or Zollenkopf bifactorisation considering that DC substations are DC voltage or current sources [**?**], [4], [9] . Others use Newton based methods [1], [2], [10]. One of the drawbacks of these last iterative methods is the lack of robustness [11]. It must be pointed out that in all the Newton based power flow methods the initialization point has a big influence over their convergence. This is another important disadvantage of these kind of methods when they are applied to the simulation of traction systems. The trains can drastically change their behavior from one instant to other, specially when they are equipped with regenerative braking systems. They can behave as a load in one instant, and a generator in the next one. In such cases, using the previous instant current and voltage profile as an initialization point is useless.

The existence of non-smooth (non-differentiable) defined loads or generators in the problem definition [12] constitutes another important obstacle for all derivative based methods. In this particular case, the non-smooth characteristics are basically derived from the trains squeeze control and the overcurrent protection systems. At each instant, a specific amount of electrical power is demanded by the train (train in traction mode), or is available to be injected in the network (train in regenerative braking mode). This is an input of the problem and is beyond the scope of this work to develop an electromechanical model of the train to calculate this power reference. If the catenary voltage is lower than a given value, the overcurrent protection reduces the demanded power when the train is in traction mode. On the other hand, if the catenary voltage is higher than a given quantity, the squeezing control will not inject all the available regenerated power into the network. It will derive part of the power to the rheostatic braking system, reducing the amount of power injected into the electrical network. With these two mechanisms, all trains collaborate to maintain the grid voltages within the limits. In [2] an optimal power flow approach was used to simulate these controls and add the voltage constrains to the problem.

From the point of view of the initialization problem, the use of Backward/forward sweep (BFS) methods could be a good option. Their convergence is not as much affected by the initial point as it is in derivative based methods. However, these methods have not been applied to this kind of systems, maybe for two reasons. The first one has to do with the topology of the DC traction systems. It is not unusual to find DC traction systems forming rings or other more complex meshed configurations, and the BFS methods were designed to be used with radial systems. Some authors already proposed solutions for using modified BFS methods in electrical meshed systems [13]–[16]. The second reason has to do with the use of complex loads or generators models in BFS methods. Most of the studied papers deal with constant impedance and constant power loads or generators, but it has been demonstrated that even PV generators and voltage dependent loads can be embedded in these sort of solvers [13], [17]–[19].

The main contribution of this work is the modification of

P. Arboleya, B. Mohamed, C. Gonzalez-Moran and I. El-Sayed are with the Department of Electrical Engineering, University of Oviedo, Spain. e-mail: arboleyapablo@uniovi.es, gonzalezmorcristina@uniovi.es, engbassam@gmail.com, islam@isa.uniovi.es

Manuscript received XXX, 2014; revised XXX, 2014.

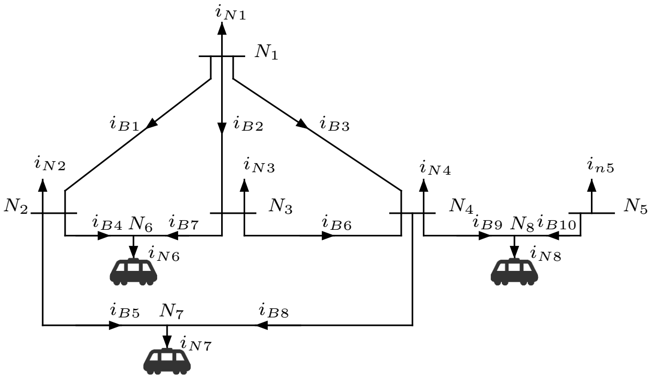


Fig. 1: *Test System n^o1* considering all meshes.

the BFS algorithm for solving power flows in weakly meshed DC traction networks with non-linear and non-smooth voltage dependent loads and generators. The authors formulate all the problem in a compact matrix form based in graph theory and node incident matrix.

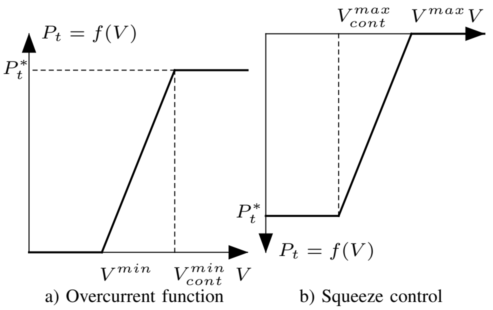
The paper is structured as follows. In section II the system that must be solved is described and the problem of overcurrent and squeeze control modeling is presented. Section III details the algorithms for solving the system with the proposed compact matrix based formulation. Section IV shows and analyses the algorithm performance and convergence when compared with a derivative based method. Finally in section V the conclusions are presented.

## II. System description

Considering the test system labelled as *Test System n^o1* represented in Fig. 1. The first node $N_1$ is defined as the slack node and it represents the AC network. Branches $B_1, B_2$ and $B_3$ represent the connection between the AC network and the DC traction network. The impedance of these branches will depend on the short-circuit impedance of the AC network, the impedance of the power transformer installed at each branch and the commutation impedance of the power converter. A detailed analysis of how these impedances can be obtained can be found in [20], [21].

The branches connecting the slack node to the DC network are called links. The DC nodes that are not trains or slack are called substations. A substation can be connected or not to the slack bus through a link. In this work, all links are considered as reversible allowing the power flow in both directions.

In Fig. 2 the selected electrical model to simulate the train behavior is represented. It represents the actual power that the train can inject or demand from the catenary as a function of the catenary voltage and the power reference $(P_t^*)$. This non-linear and non-smooth piecewise definition of the trains characteristic has been also used in [12], [22], [23]. The authors assume that the power reference is an input and can be calculated with complex train models like the one proposed in [24]. Once the power reference is known (positive if the train is in traction mode or negative if the train is in braking mode), this power reference is absorbed or injected into the grid just if the catenary voltage $(V)$ is among the limits $(V_{cont}^{min}, V_{cont}^{max})$. When the train is in breaking mode ($P_t^* < 0$), the power injection increases the catenary voltage. If this voltage rises above $V_{cont}^{max}$ the power injection must be reduced. Before exceed the maximum value permitted for the catenary voltage $V_{max}$, the injected power must be totally restricted. The difference between the power available for being regenerated and the actual power injected into the grid will be burned using the rheostatic braking system. This injection power strangulation is called squeeze control. It is graphically represented in Fig. 2b) and mathematically in expression (1). When the train is in traction mode ($P_t^* > 0$), the power absorption tends to reduce the catenary voltage. If the voltage is too low, the overcurrent protection reduces the power demand and before the voltage falls below $V_{min}$ the train power electronics equipment disconnect the train from the grid. The graphical representation of the overcurrent protection model can be observed in Fig. 2a) and can be mathematically formulated according expression (2).



a) Overcurrent function (traction mode)

b) Squeeze control (breaking mode)

Fig. 2: Squeeze and overcurrent protection functions for a given power reference $(P_t^*)$.

$$P_t = \begin{cases} P_t^* & V \leq V_{cont}^{max} \\ P_t^* \cdot \frac{V^{max}-V}{V^{max}-V_{cont}^{max}} & V_{cont}^{max} < V \leq V^{max} \\ 0 & V > V^{max} \end{cases} \tag{1}$$

$$P_t = \begin{cases} 0 & V \leq V^{min} \\ P_t^* \cdot \frac{V-V^{min}}{V_{cont}^{min}-V^{min}} & V^{min} < V \leq V_{cont}^{min} \\ P_t^* & V > V_{cont}^{min} \end{cases} \tag{2}$$

This piecewise definition of the train, makes possible to model its real behavior and the interaction between the trains and the electrical network. However,it makes the problem not too much tractable for being solved with conventional derivative-based algorithms. On the contrary, modeling the trains just like non-linear constant power loads or generators, will lead to unconstrained power flow. The calculated voltages and power flows would not represent the real situation.

## III. Algorithm description

### A. Core BFS algorithm

In this section, the core of the BFS algorithm that will constitute the most internal loop will be described. This specific algorithm is applied when the network is radial. For this reason, all branches that generate meshes are cut and the currents through those branches are going to be assumed as known currents. In further sections, it will be explained how these currents are updated, but for now, the currents through the cut branches are assumed as known. In the same way, the currents demanded or injected by the loads (trains and substations) are considered as constant known values. For the first iteration, all nodal currents are initialized using the their rated values, the ones that they would demand or inject at the rated voltage. In further sections, it will be explained how this value will be updated considering the dependence of these currents with the network voltage.

The problem is formulated using the node incidence matrix ($\mathbf{\Gamma}$). Each row of $\mathbf{\Gamma}$ represent a branch and each column a node. For a given row representing a branch connecting the nodes $p$ and $q$, the column $p$ will be filled with 1, and the column $q$ will be filled with $-1$ ($p < q$). Therefore, $\Gamma_{ij}$ element can be defined as:

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

Using the $\mathbf{\Gamma}$ matrix, the Kirchhoff Current and Voltage Laws (KCL and KVL) in all nodes and branches are formulated as follows:

$$\begin{aligned} \mathbf{\Gamma}^T \cdot \mathbf{I}_B^T + \mathbf{I}_d \cdot \mathbf{I}_N^T &= 0 \qquad (3)\\ \mathbf{\Gamma} \cdot \mathbf{V}^T - \mathbf{R}_B \cdot \mathbf{I}_B^T &= 0 \qquad (4) \end{aligned}$$

Where:

- $\mathbf{I}_B$ is a vector containing all branch currents $i_{Bi}$.
- $\mathbf{I}_N$ is a vector containing all nodal currents $i_{Ni}$.
- $\mathbf{V}$ is a vector containing all nodal voltages $V_i$.
- $\mathbf{R}_B$ is a diagonal matrix. Each term of the diagonal $R_{Bi}$ represents the resistance of branch $i$. In this first approach these terms will be constant for a given instant.
- $\mathbf{I}_d$ is the identity matrix.

The matrix equation (3) represents the KCL in all nodes. It has $n_N$ equations and $(n_N + n_B)$ unknowns, being $n_N$ the number of nodes and $n_B$ the number of branches. The unknowns for this equation are the nodal currents and the branch currents. The matrix equation (4) represents the KVL in all branches. It has $n_B$ equations and $(n_N + n_B)$ unknowns, the voltages in all nodes and the currents in al branches.

The total number of equations represented in (3) and (4) are $(n_N + n_B)$ and the total number of unknowns is $(2 \cdot n_N + n_B)$. It is obvious that it will be necessary the use of the load/generator models and the voltage definition in the slack to solve the problem.

A radial version of the *Test System n$^o$1* in Fig. 1 could be obtained cutting all the links except the first one and all the DC branches creating meshes (for instance $B_5$) (see Fig.3).

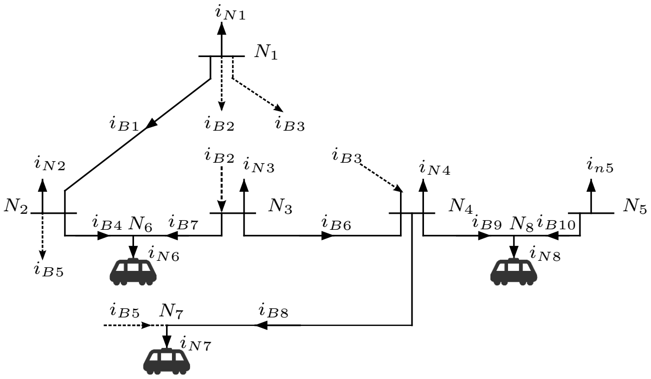


Fig. 3: Radial version of the *Test System n$^o$1*. Links $B_2$ and $B_3$ have been cut as well as DC line $B_5$.

The currents through the cut branches are considered as known currents (zero for the first iteration). The vectors $\mathbf{l}_c$ and $\mathbf{l}_{nc}$ are defined as vectors containing respectively the indices of the cut and non-cut branches. Under these assumptions, the currents through the non-cut branches can be obtained in a general form using the expression (3) as follows:

$$\begin{aligned} \mathbf{I}_B^T(\mathbf{l}_{nc}) = \Big(\mathbf{\Gamma}^T(2:n_N, \mathbf{l}_{nc})\Big)^{-1} \cdot \\ \cdot \Big(-\mathbf{\Gamma}^T(2:n_N, \mathbf{l}_c) \cdot \mathbf{I}_B^T(\mathbf{l}_c) - \mathbf{I}_N^T(2:n_N)\Big) \end{aligned} \qquad (5)$$

The colon operator (:) is defined as follows. When it separates two indices in a vector or a matrix it means *"from ... to"*. For instance, in $\mathbf{X}(i : j, k)$ it represents the column $k$ of the matrix $\mathbf{X}$ from row $i$ to row $j$. When it is used alone, as for instance in $\mathbf{X}(k, :)$, it represents *"the whole"* row or column. For instance, $\mathbf{X}(k, :)$ represents the whole kth row of the matrix.

It must be remarked that for the first iteration the currents through the cut branches $\mathbf{I}_B^T(\mathbf{l}_c)$ are initialised as zero. In conventional AC distribution systems, when the variation of the loads is slow, it can be a good strategy to initialise the currents through the cut branches with the value obtained in the previous instant. However, in DC traction networks, the change of the trains power reference can vary drastically from one instant to the next one. For instance, a train can be braking at a given instant acting as a generator and accelerating in the following instant acting as a load. This situation can make the current flow direction through a cut branch vary in two successive instants, so the best solution is to initialise the cut branches currents to zero in all instants.

The currents injected in all nodes are computed assuming a plain voltage profile with the voltage value equal to the rated one. As it can be observed in equation (5), the slack current ($i_{N1}$) is not considered. It is important to remark that the matrix $\mathbf{\Gamma}^T(2 : n_N, \mathbf{l}_{nc})$ is always a square regular matrix. Once the cut branches are selected, $\mathbf{\Gamma}^T(2 : n_N, \mathbf{l}_{nc})$ will be constant during all iterative process. Its inverse has to be calculated only once. With a single matrix equation, the backward sweep (BS) is completed. In successive iterations, the injected nodal currents will be calculated assuming the voltage profile of the previous iteration.

The forward sweep (FS) can be conducted by means of (4) removing the rows corresponding to the cut branches and assuming that the slack voltage is a known variable. The matrix expression for the FS step is expressed in a general form as:

$$\begin{aligned}\mathbf{V}^T(2:n_N) &= (\mathbf{\Gamma}(\mathbf{l}_{nc}, 2:n_N))^{-1} \cdot \\ &\cdot \left(-\mathbf{\Gamma}(\mathbf{l}_{nc}, 1)\mathbf{V}(1) + \mathbf{R}_B(\mathbf{l}_{nc}, \mathbf{l}_{nc})\mathbf{I}_B^T(\mathbf{l}_{nc})\right)\end{aligned} \tag{6}$$

In this case, it can be also stated that the matrix $\mathbf{\Gamma}(\mathbf{l}_{nc}, 2:n_N)$ is a square regular matrix. If the network is radial and the loads are constant current loads, the algorithm is a direct non-iterative algorithm, there is no need of iterate. This algorithm will be designated as a *Core BFS Algorithm* (C-BFS) and its pseudocode is represented in Algorithm 1.

```
Input: $\mathbf{\Gamma}, \mathbf{R}_B, \mathbf{l}_c, \mathbf{l}_{nc}, \mathbf{I}_N(2:n_N), \mathbf{I}_B(\mathbf{l}_c), V_{slack}$
Output: $\mathbf{I}_B(\mathbf{l}_{nc}), \mathbf{V}(2:n_N)$
  1. $\mathbf{I}_B(\mathbf{l}_{nc}) \leftarrow expr.(5)$ Backward Sweep
  2. $\mathbf{V}(2:n_N) \leftarrow expr.(6)$ Forward Sweep
```

**Algorithm 1:** C-BFS (Core Backward/Forward Sweep Algorithm).

For meshed networks, the currents through the cut branches must be updated while the stop criteria is not fulfilled. The stop criteria will use the KVL in the cut branches and it is expressed as:

$$||\mathbf{\Gamma}(\mathbf{l}_c, :) \cdot \mathbf{V} - \mathbf{R}(\mathbf{l}_c, \mathbf{l}_c) \cdot \mathbf{I}_B(\mathbf{l}_c)|| \leq \epsilon \tag{7}$$

The procedure for updating the currents through the cut branches is described in the next section.

### *B. Cut branches current updating procedure*

Once the first C-BFS iteration is run, the currents through the cut branches must be updated. For this purpose, an improved method, based on the one presented in [15] is going to be described. In the *Test System n$^o$1* represented in Fig. 1, let us consider that only one branch (Branch 2) was cut. The current update through the cut branch ($\Delta I_{B2}$) can be calculated solving the circuit depicted in Fig. 4a) by means of the Thevenin equivalent. Fig. 4a) represents the Thevenin equivalent of the system observed from nodes $N_1$ and $N_3$. The Thevenin voltage is calculated as $V_{th2} = V_1^0 - V_3^0$.The super indices ($^0$) express that the voltages are the ones obtained from the C-BFS algorithm. For calculating the Thevenin resistance $R_{th2}$, neither train currents nor nodal currents injections are considered. Once the Thevenin resistance is calculated, the current through the branch $B_2$ is updated using the next incremental current:

$$\Delta I_{B2} = \frac{V_{th2}}{R_{B2} + R_{th2}} = \frac{V_1^0 - V_3^0}{R_{B2} + R_{th2}} \tag{8}$$

The radial network represented in Fig. 3, obtained by cutting the *Test System n$^o$1* branches $B_2$, $B_3$ and $B_5$, can be represented as a multipole system like the one depicted in Fig. 5.

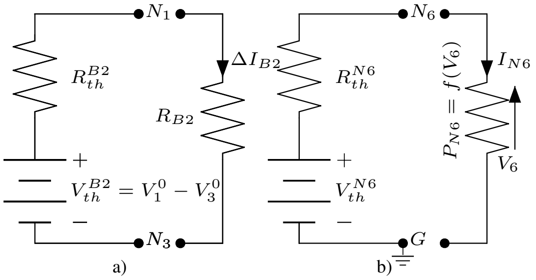


Fig. 4: a) Thevenin equivalent of the network observed from nodes $N_1$ and $N_3$. b) Thevenin equivalent of the network observed from train $N_6$.

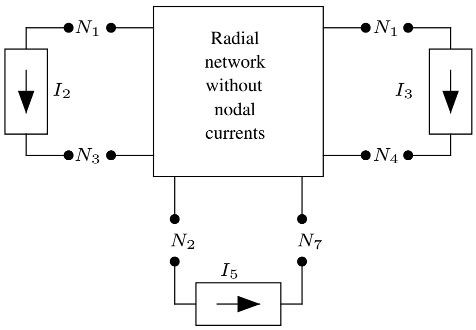


Fig. 5: Multipole representation of the *Test System n$^o$1*. Links $B_2$ and $B_3$ have been cut as well as DC line $B_5$. Unitary current sources $I_2$, $I_3$ and $I_5$ will be sequentially connected to obtain the Thevenin resistances.

The Thevenin voltage sources can be calculated using the voltages already obtained from C-BFS algorithm. For calculating the Thevenin resistances, a current source connected to each dipole corresponding to a cut branch will be considered. With all current sources activated at the same time, the relationship between the current sources and the dipole voltages can be expressed as follows:

$$\begin{pmatrix} V_1 - V_3 \\ V_1 - V_4 \\ V_2 - V_7 \end{pmatrix} = \begin{pmatrix} R_{22} & R_{23} & R_{25} \\ R_{32} & R_{33} & R_{35} \\ R_{52} & R_{53} & R_{55} \end{pmatrix} \cdot \begin{pmatrix} I_2 \\ I_3 \\ I_5 \end{pmatrix} \tag{9}$$

Applying the current sources sequentially, the Thevenin resistances (those in the main diagonal) can be easily obtained by means of the voltages in the left side of the equation (9). For instance, with the sequence $(I_2, I_3, I_5) = (1, 0, 0)$, the Thevenin resistance $R_{th2}$ will be $R_{th2} = R_{22} = V_1 - V_3$. The voltages $V_1$ and $V_3$ can be obtained using the C-BFS algorithm, the circuit is radial and it only takes one iteration for each pair of voltages. This process is repeated for all cut branches.

To express this procedure in a general way using a compact matrix formulation, expressions (5) and (6) are used in a simplified form. In expression (5), nodal currents are not considered. In (6), the slack voltage is set to zero because in this part of the algorithm, only voltage differences are needed and not absolute values. Under these assumptions, all voltages derived from the current injections can be calculated as:

$$\begin{aligned}\mathbf{V}^T_{ci}(2:n_N) = -\boldsymbol{\Gamma}(\mathbf{l}_{nc}, 2:n_N)^{-1} \cdot \mathbf{R}_B(\mathbf{l}_{nc}, \mathbf{l}_{nc}) \cdot \\ \cdot\, \boldsymbol{\Gamma}^T(2:n_N, \mathbf{l}_{nc})^{-1} \cdot \boldsymbol{\Gamma}^T(2:n_N, \mathbf{l}_c) \cdot \mathbf{I}^T_{Bci}(\mathbf{l}_c)\end{aligned} \tag{10}$$

Where the subindex $(_{ci})$ represents the current source that is activated. It varies from 1, to the number of cut branches $(n_c)$.

Using the voltages calculated in (10), each column of the resistance matrix can be computed as follows:

$$\mathbf{R}^B_{th}(:, ci) = \boldsymbol{\Gamma}(\mathbf{l}_c, 2:n_N) \cdot \mathbf{V}^T_{ci}(2:n_N) \tag{11}$$

The main diagonal of Thevenin resistance matrix ($\mathbf{R}^B_{th}$) will contain the Thevenin resistances. This matrix, only needs to be updated if the indices of cut branches change. It does not have to be updated at each iteration. The algorithm to calculate this matrix is labeled as *branches Thevenin equivalent resistance calculation* (BTHRC) and its pseudocode is expressed in Algorithm 2.

```
Input: $\boldsymbol{\Gamma}, \mathbf{R}_B, \mathbf{l}_c, \mathbf{l}_{nc}$
Output: $\mathbf{R}^B_{th}$
for $i = 1$ to $n_c$ do
    $\mathbf{I}_{Bci} = 0$
    $I_{Bci}(i) = 1$
    $\mathbf{V}_{ci}(2:n_N) \leftarrow expr.(10)$
    $\mathbf{R}^B_{th}(:, i) \leftarrow expr.(11)$
end for
```

**Algorithm 2:** BTHRC (Branches Thevenin Equivalent Resistance Calculation Algorithm).

Finally, the correction terms for updating the currents through the cut branches are calculated as:

$$\Delta\mathbf{I}^T_B(\mathbf{l}_c) = \left(\mathbf{R}_{B+th}\right)^{-1} \boldsymbol{\Gamma}(\mathbf{l}_c, :) \cdot \left(\mathbf{V}^0\right)^T \tag{12}$$

Where $\mathbf{R}_{B+th}$ is a diagonal matrix of size $(n_c, n_c)$. The terms in the diagonal are calculated by the addition of the cut branch resistances $(R_B)$ and the Thevenin equivalent resistances $(R_{th})$. Again, the super-index $(^0)$ indicates that the voltages were obtained from the C-BFS algorithm. The currents through the cut branches are updated using the correction term:

$$\mathbf{I}^T_B(\mathbf{l}_c) = \left(\mathbf{I}^0_B(\mathbf{l}_c)\right)^T + \Delta\mathbf{I}^T_B(\mathbf{l}_c) \tag{13}$$

From now on, this current correction procedure will be labeled as *Cut Branch Correction* Algorithm (CBC). Its pseudocode can be analyzed in Algorithm 3.

```
Input: $\boldsymbol{\Gamma}, \mathbf{R}_B, \mathbf{l}_c, \mathbf{l}_{nc}, \mathbf{V}^0, \mathbf{R}^B_{th}$
Output: $\mathbf{I}_B(\mathbf{l}_c)$
$\Delta\mathbf{I}_B(\mathbf{l}_c) \leftarrow expr.(12)$ Cut branch current corr. term
$\mathbf{I}_B(\mathbf{l}_c) \leftarrow expr.(13)$ Cut branch current correction
```

**Algorithm 3:** CBC (Cut Branch Correction Algorithm).

In algorithm 4 the combination of the *core BFS algorithm* (C-BFS) with the *branches Thevenin equivalent resistance calculation* (BTHRC) and *cut branch correction procedure* (CBC) is represented. The three combined algorithms will be denoted as *Meshed Networks with Constant Current Loads Backward/Forward Sweep Algorithm* (MNCCL-BFS).

```
Input: $\boldsymbol{\Gamma}, \mathbf{R}_B, \mathbf{l}_c, \mathbf{l}_{nc}, V_{slack}, \mathbf{I}_N$
Output: $\mathbf{I}_B, \mathbf{V}(2:n_N)$
$\mathbf{I}_B(\mathbf{l}_c) \leftarrow 0$
$\mathbf{V} \leftarrow V_{slack}$
$\mathbf{R}^B_{th} \leftarrow$ BTHRC() Algorithm 2
for i = 1 to maximum number of iterations do
    $\mathbf{I}_B(\mathbf{l}_{nc}) \leftarrow$ C-BFS() Algorithm 1
    if $expr.(7)$ is true then
        break
    else
        $\mathbf{I}_B(\mathbf{l}_c) \leftarrow$ CBC() Algorithm 3
    end if
end for
```

**Algorithm 4:** MNCCL-BFS (Meshed Networks with Constant Current Loads Backward/Forward Sweep Algorithm).

### *C. Nodal currents correction*

Until now, the currents demanded or injected in the nodes (trains or substations) were assumed as constant. However, as it was mentioned in previous sections, the nodal currents (trains or substations) are defined as voltage dependent current sources. At every instant, the train will try to inject or absorb a given power ($P^*_t$). Still, the real power injection or absorption ($P_t$) will be constrained by the squeeze control or the overcurrent protection. These two controls create a non-linear but also non-smooth (non-differentiable) voltage-dependent piecewise defined characteristic (see Fig. 2).

Usually, this situation is solved using an *if-then* approach. Depending on the previous iteration voltage, an interval of the piecewise defined characteristic is selected. However, sometimes the new obtained voltage is out of the previously selected interval, so a new iteration is required. In addition, when the slope of the squeeze control or the overcurrent protection characteristic is too high, the algorithm can iterate in an infinite loop between two non-adjacent segments. The correct solution cannot be reached because is in the segment situated in the middle.

To overcome this obstacle, the authors propose a method based on the Thevenin equivalent. The method calculates the Thevenin equivalent of the network from the point of view of each train (or substation). For calculating the Thevenin voltage the MNCCL-BFS will be used, the nodal currents will be set sequentially to zero and thus, the obtained voltage in the node with zero nodal current will be the thevenin voltage.

The procedure to calculate the Thevenin equivalent impedance of the network from the point of view of the nodal loads (trains/substations), is similar to the one described in the previous subsection. Unitary current sources will be connected sequentially between all nodes (except the slack node) and ground. The obtained voltages will be the columns of the equivalent resistance matrix. The algorithm used for this purpose is again the MNCCL-BFS. In this case, as it happened with the CBC algorithm, the slack voltage can be set to zero because only the induced voltages derived from the injected currents are needed. This procedure is labeled as *nodes Thevenin resistance calculation* (NTHRC) and its pseudocode is detailed in Algorithm 5.

**Input:** $\mathbf{\Gamma}, \mathbf{R}_B, \mathbf{l}_c, \mathbf{l}_{nc}$
**Output:** $\mathbf{R}_{th}^N$

```
$\mathbf{V}_N \leftarrow 0$
$\mathbf{I}_B(\mathbf{l}_c) \leftarrow 0$
for $i = 2$ to $n_N$ do
    $\mathbf{I}_N \leftarrow 0$
    $\mathbf{I}_N(i) \leftarrow -1$
    $\mathbf{V}_N \leftarrow$ MNCCL-BFS()
    $R_{th}^{Ni} \leftarrow \mathbf{V}_N(i)$
end for
```

**Algorithm 5:** NTHRC (Nodes Thevenin equivalent resistance calculation).

For calculating the Thevenin equivalent voltage of the network from the point of view of the nodal loads (trains/substations), the MNCCL-BFS will be used. The nodal currents will be set sequentially to zero and thus, the obtained voltage in the node with zero nodal current will be the Thevenin voltage. This algorithm is labeled as *nodes equivalent Thevenin voltage calculation* (NTHVC) and its pseudocode can be observed in Algorithm 6.

**Input:** $\mathbf{\Gamma}, \mathbf{R}_B, V_{slack}, \mathbf{I}_N^0, \mathbf{l}_c, \mathbf{l}_{nc}$
**Output:** $\mathbf{V}_{th}^N$

```
$\mathbf{V}_N \leftarrow V_{slack}$
$\mathbf{I}_B(\mathbf{l}_c) \leftarrow 0$
for $i = 2$ to $n_N$ do
    $\mathbf{I}_N \leftarrow \mathbf{I}_N^0$
    $\mathbf{I}_N(i) \leftarrow 0$
    $\mathbf{V}_N \leftarrow$ MNCCL-BFS()
    $V_{th}^{Ni} \leftarrow \mathbf{V}_N(i)$
end for
```

**Algorithm 6:** NTHVC (Nodes Thevenin equivalent voltage calculation).

The Thevenin voltages and resistances are calculated for all nodes containing a voltage dependent load or generator. Then, an equivalent circuit like the one depicted in Fig. 4b) is obtained for each of those nodes. Usually, if the node is demanding power and the voltage is too low, the power demand is reduced due to the overcurrent protection. If the node is injecting power into the DC network and the voltage is too high, the injected power is reduced by the squeeze control. In both cases, the load/generator characteristic is a piecewise-defined function that may have several segments. In the specific case represented in the figure 6 it has three segments.

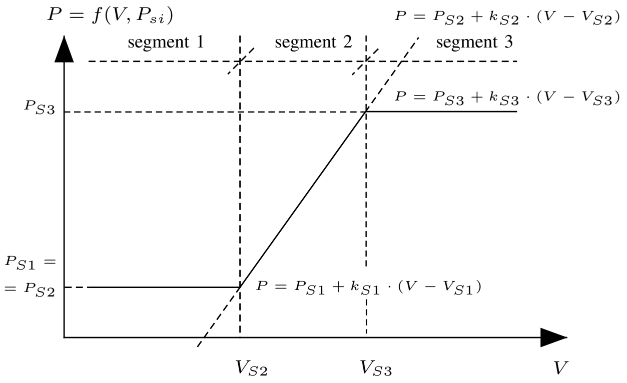


Fig. 6: Generic piecewise defined power load.

Each segment is characterized by its power and voltage. The characteristic voltage of a segment is defined as the lowest voltage of the segment, and it is labelled as ($V_{Si}$), being $i$ the index of the segment. The characteristic power of a segment is the one associated with its characteristic voltage and it is denoted by ($P_{Si}$). The characteristic voltage of the first segment must be defined as a voltage lower than the minimum catenary voltage (zero, for instance). To select the active segment, the Thevenin equivalent circuit is used (see Fig. 4b)). The algorithm checks all the segments starting from the highest index one until the condition (14) is fulfilled. Then the catenary voltage of the node under analysis ($V_i$), is determined by means of expression (15). $k_{Si}$ represents the slope of the segment. This expression is derived from the basic analysis of the circuit represented in Fig. 4b).

$$P_{Si} \leq \frac{(V_{th} - V_{Si}) \cdot V_{Si}}{R_{th}} \tag{14}$$

$$\begin{aligned} V_i = {} & (1/2)(V_{th} - k_{Si}R_{th} + \\ & + \sqrt{(k_{Si}R_{th} - V_{th})^2 + 4R_{th}(P_{Si} - V_{Si}k_{Si})}) \end{aligned} \tag{15}$$

Finally the actual injected or demanded power is obtained with the active segment characteristic equation (see Fig. 6 and expression (16)).

$$P_i = P_{Si} + k_{Si} \cdot (V_i - V_{Si}) \tag{16}$$

Once the power $P_i$ and the voltage $V_i$ are determined, the load can be treated as a constant current load. The power flow in the next iteration can be solved again using the MNCCL-BFS algorithm. The pseudo-code of the algorithm to correct the nodal currents injections can be observed in Algorithm 7.

The integration of the *nodal current correction algorithm* (NCC) with the *meshed network with constants current loads backward/forward sweep algorithm* (MNCCL-BFS), allows to solve meshed networks with non-smooth voltage-dependent piecewise defined loads or generators. The nodal currents

```
Input: $\mathbf{\Gamma}, \mathbf{R}_B, V_{slack}, \mathbf{I}_N^0, \mathbf{l}_c, \mathbf{l}_{nc}, \mathbf{P} = \mathbf{f}(\mathbf{V}), \mathbf{R}_{th}^N$
Output: $\mathbf{I}_N$
$\mathbf{V}_{th}^N \leftarrow$ NTHVC()
for $j = 2$ to $n_N$ do
    for $i = n_S$ to $1$ do
        if $expr.(14)$ is true then
            segment $i$ is active
            $V_i(j) \leftarrow expr.(15)$
            $P_i(j) \leftarrow expr.(16)$
            $I_N(j) \leftarrow P_i(j)/V_i(j)$
            break
        end if
    end for
end for
```

**Algorithm 7:** NCC (Nodal currents correction).

are initialized in the first iteration assuming a plain voltage profile. Then, the MNCCL-BFS algorithm is used to update the voltage profile and the branch currents. Finally, the nodal currents will be corrected using the NCC algorithm. The updated nodal currents will be used as a new input for the MNCCL-BFS algorithm. The process is repeated until the stop criteria or the maximum number of iterations is reached. The stop criteria is expressed as:

$$\left|\left|\mathbf{I}_N(2:n_N) - \mathbf{I}_N^0(2:n_N)\right|\right| \leq \epsilon \tag{17}$$

In Algorithm 8, the combination of these two algorithms is described. The new algorithm is labeled as (MNVDL-BFS) (*Meshed Networks with Voltage dependent Loads Backward/Forward Sweep Algorithm*).

```
Input: $\mathbf{\Gamma}, \mathbf{R}_B, \mathbf{l}_c, \mathbf{l}_{nc}, V_{slack}, \mathbf{P} = \mathbf{f}(\mathbf{V})$
Output: $\mathbf{I}_B, \mathbf{V}(2:n_N), \mathbf{I}_N(1)$
$\mathbf{R}_{th}^N \leftarrow$ NTHRC()
$\mathbf{I}_N(2:n_N) \leftarrow \mathbf{P}(V_{slack})/V_{slack}$
for i = 1 to maximum number of iterations do
    $\mathbf{I}_B, \mathbf{V} \leftarrow$ MNCCL-BFS() Algorithm 4
    $\mathbf{I}_N^0(2:n_N) \leftarrow \mathbf{I}_N(2:n_N)$
    $\mathbf{I}_N(2:n_N) \leftarrow$ NCC() Algorithm 7
    if $expr.(17)$ is true then
        break
    end if
end for
```

**Algorithm 8:** MNVDL-BFS (Meshed Networks with Voltage dependent Loads Backward/Forward Sweep Algorithm).

## IV. Algorithm performance analysis

The algorithm was first validated using a previously developed software package [1], [2], [12]. The previous software solves the same problem but using the well-known trust-region dogleg algorithm [25]. The initial conditions are the same for both methods:

- Plain voltage profile (1 p.u.) in all nodes.
- Zero for all branch currents.
- Currents at rated voltage for all nodal currents.

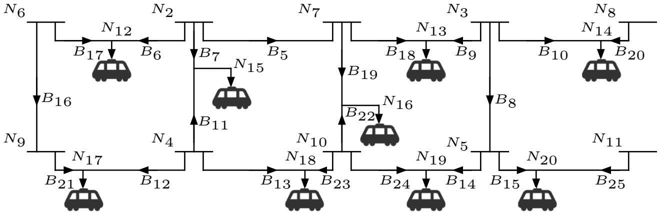


Fig. 7: *Test System n$^o$2* Meshed DC traction network.

The algorithm was implemented using MATLAB. All tests were conducted in a conventional personal computer with a processor Intel(R) Core(TM) i7-2670QM CPU @ 2.20GHz and 4GB of RAM.

To analyze the performance of the algorithm, four test batteries of $10^5$ cases/battery are presented. The rated voltage of the DC system is 750V in all cases. The catenary and the rail resistance are assumed to be 7m$\Omega$/km and 3.5m$\Omega$/km in all cases. It is also assumed, that each train has a perfect connection to ground so the rail resistance is added to the catenary resistance. The method described in [21] was used for obtaining the equivalent impedance of the substations as a function of their rated power and the power transformer short circuit voltage. The line lengths oscillates between 2 and 3 km in the three systems. All the trains have a maximum traction power demand and a maximum regenerated power capacity of 1MW. The reference power for all trains and cases in a battery test is uniformly distributed between the maximum traction power and the maximum regeneration capacity (±1MW).

This section is structured as follows. First the behavior of the algorithm will be evaluated in two small scale systems by means of three test batteries. Then the convergence and the accuracy of the proposed algorithm are evaluated by means of the test battery 4. This test battery is performed in a large network based in a real topology.

### A. Analysis of the algorithm behaviour in small networks

The test battery $n^o1$ was carried out using the *Test System n$^o$1* in Fig. 1, while the test batteries $n^o2$ and $n^o3$ were effectuated using the *Test System n$^o$2* in Fig. 7. The substations rated power and short circuit voltage in *Test System n$^o$1* and *2* were considered as 3MW and 5% respectively.

The results obtained in the batteries $n^o1$, $n^o2$ and $n^o3$ are represented in Fig. 8. In the first and the third battery tests (rows 1 and 3 of Fig. 8) , the squeeze control is activated at 775V and the maximum catenary voltage is set to 780V, the overcurrent protection is activated at 725V being the minimum catenary voltage 720V. This set up generates extremely stressed scenarios. In test battery $n^o2$ (row 2 of Fig. 8), the squeeze control is activated at 850V and the maximum catenary voltage is set to 900V. The overcurrent protection is activated at 700V and the minimum catenary voltage is set to 650V.

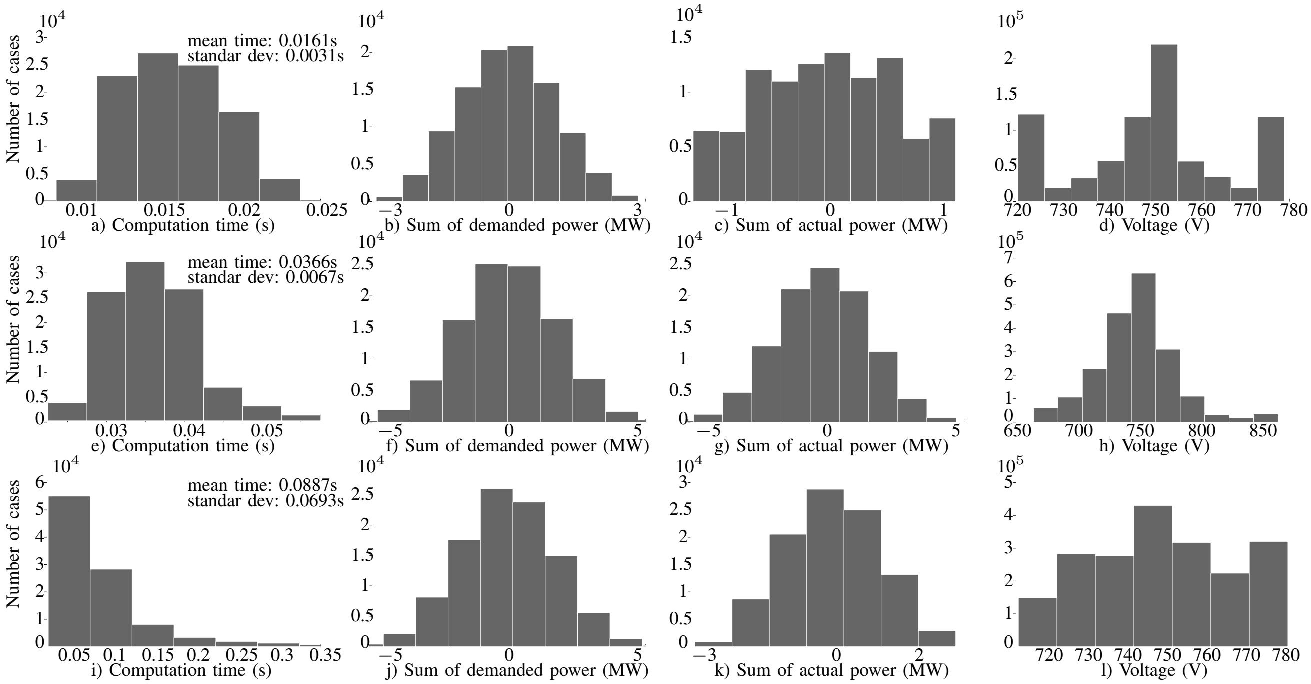


Fig. 8: Proposed algorithm performance indices. Row 1 represent the test battery 1 with *Test System n$^o$1*. Rows 2 and 3 correspond to the test batteries 2 and 3 with *Test System n$^o$2*. Column 1 (a, e and i) represents the computation time in seconds. Column 2 (b, f and j) represents the sum of all demanded (positive) or available for regenerate (negative) power in all trains, while column 3 (c,g and k) represents the sum of the real injected or absorbed power by all trains. Finally in column 4 (d,h and l), all voltages obtained in all cases and in all nodes of the system can be observed.

*1) Test battery n$^o$1:* The mean time spent for simulating each instant was 0.0161s and the standard deviation 0.0031s. Comparing these results with those provided by the previous derivative based software package, it can be observed that the mean time is reduced in more than 90%. With the previous software the mean time per case was 0.31s with a standard deviation of 0.16s. Comparing Fig. 8b) and c), it can be observed that this case is quite stressed. Fig. 8b) represents the sum of the total power that all trains try to inject or absorb into the network. In some cases, the three trains try to inject the maximum power or absorb the maximum power at the same time (±3MW). However, as it was explained in previous sections, because the squeeze control and the overcurrent protection, not always the train is able to inject or absorb the required power. This effect can be observed in Fig. 8c), where the real injected or absorbed power is represented. The maximum real injected or absorbed power by all trains in a single instant is around 1MW. The stress in the network can be also noticed analyzing 8d) where the histogram of voltage in all nodes and all cases is depicted. There are more than $10^5$ node voltages near the minimum, this is an indicator of overcurrent protection activation. There are more that $10^5$ nodes with a voltage near the maximum, indicating the activation of the squeeze control.

*2) Test battery n$^o$2:* It is applied to the scheme represented in Fig. 7. In this case the network is more complex and the mean simulation time is 0.0366s. The time reduction comparing with the derivative based method is about 95%. This is a non-stressed network due to the relaxation of the maximum and minimum voltages up to the real values. In this case, the histograms of the sum of the total demanded or absorbed power (Fig. 8f)) and the actual injected power (Fig. 8g)) are similar. There are also nearly zero nodes with the maximum or the minimum voltages (900V and 650V respectively) as it can be observed in Fig. 8h).

*3) Test battery n$^o$3:* It corresponds to the same topology used for the test $n^o2$. However, in this case, the maximum and minimum voltages were set to 780V and 720V respectively to created a extremely stressed network. The sum of the maximum demanded power oscillates between -5MW and 5MW (see Fig. 8j)), while the actual demanded power oscillates between -2.5MW and 2.5MW. More than 40% of the nodes are in the voltage band where the squeeze control or the overcurrent protection is active. In this test case the mean time to solve an instant was 0.0887s. With the derivative based method, the mean time was 0.96s, but in more than 25% of the instants the convergence was not achieved. The proposed method is not only faster but also more stable.

### *B. Analysis of the algorithm behaviour in large networks*

The $10^5$ cases of the Test battery $n^o3$ were conducted using the network depicted in Fig. 9. This topology is based on a 64 nodes real commuter network. It has 40 stations, 10 of them connected to the AC system (the slack node in this specific case of study). The number of operative trains is 24.

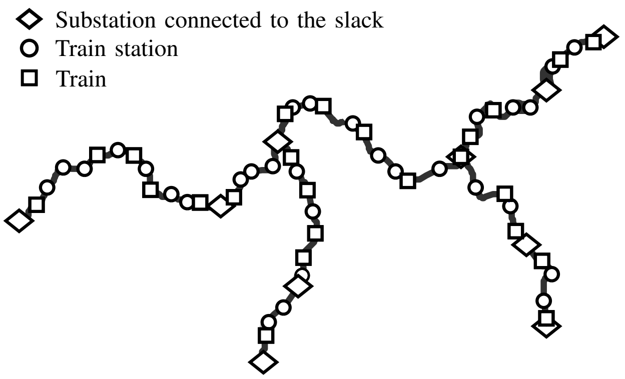


Fig. 9: *Test System n$^o$3* Real large conmuter topology.

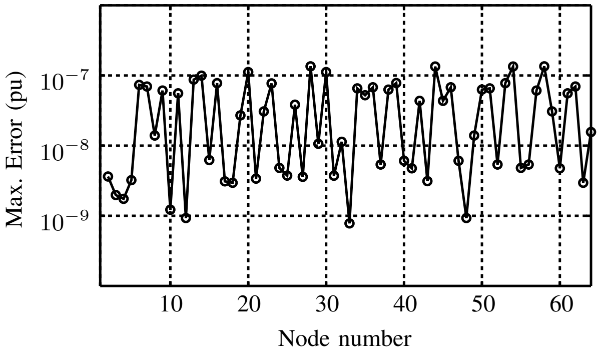


Fig. 10: The node voltage error of BFS-based method relative to derivative based software results.

In Fig. 9, the slack and the 10 lines connecting the slack to the substations are not represented. However, these 10 lines create 9 loops, transforming the radial topology in a weakly meshed one. The rated power of each substation and the power transformer short circuit voltage are assumed to be 15MW and 5% respectively. This scenario tries to simulate a real one, so the squeeze control is activated at 850V and the maximum catenary voltage is set to 900V. The overcurrent protection is activated at 550V and the minimum catenary voltage is set to 500V. In Fig. 10, the maximum voltage error in all nodes except the slack, between the proposed algorithm and the derivative based one considering the $10^5$ cases is represented. As it can be observed the maximum error is always lower than $1.1 \cdot 10^{-7}$ (p.u.). The selected threshold ($\epsilon$) for the stop criteria expressed in the expression (17) is $10^{-6}$.

The convergence of this method will be demonstrated by means of empirical methods [26]–[31]. The evolution of the error with respect to the time, the number of iterations and other parameters described in the following paragraphs have been analyzed for a large number of test batteries. In Figs. 11 and 12, the convergence in this specific test battery can be analyzed and compared with the derivative based algorithm convergence.

In Fig. 11a) and b) the cumulative percentage of solved cases as a function of the number of iterations is depicted. In terms of iterations, the derivative method is more effective because most of the cases are solved in a lower number of iterations. However, the analysis of this single parameter is not determinant.

In Fig. 11c) and d) the mean, maximum and minimum error versus the iteration number are represented for the proposed method and the derivative based one respectively. For calculating the maximum error at the iteration $i$ ($Max.Error_i$), the expression (18) is used. $A_i$ represents the set of all active cases at iteration $i$, avoiding to compute those cases that already converged in previous iterations. $n\left(A_i\right)$ represent the number of active cases.

$$Max.Error_i = \frac{\sum\limits_{j \in A_i} \left|Max.Error_j\right|}{n\left(A_i\right)} \tag{18}$$

The minimum and the mean error in Fig. 11c) and d) are computed in the same way, only the active cases at each iteration are considered for calculating the values.

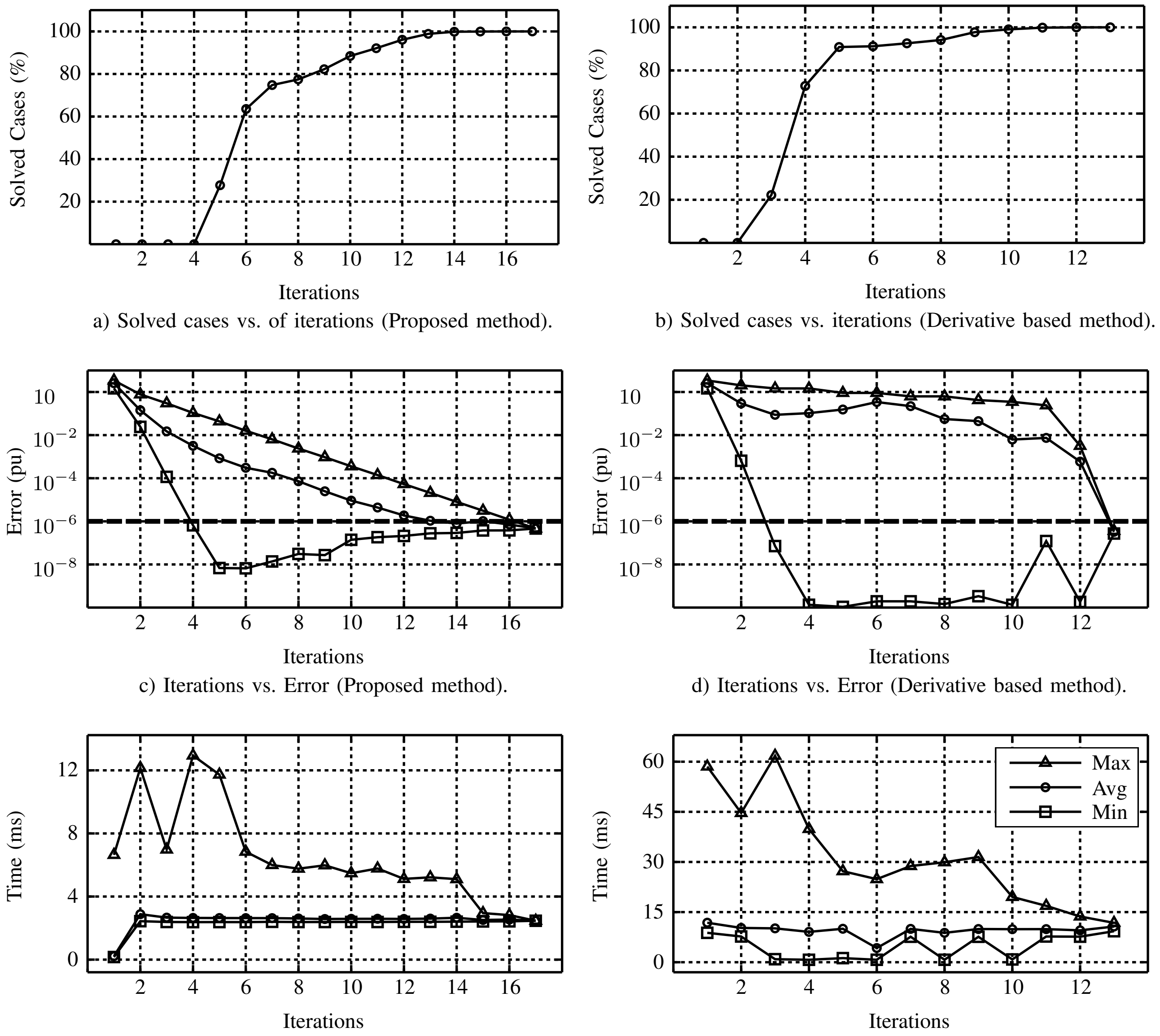
The maximum, minimum and mean time invested at each iteration are represented in Fig 11e) and f). For the proposed method the mean time per iteration is very homogeneous and close to the minimum (see Fig 11e)). The standard deviation of the time invested at each iteration is very low (0.29ms according Table I), and only a few iterations present times higher than 10ms. When comparing with the time per iteration in the derivative based method (see Fig. 11f)), it can be observed that the mean time per iteration is higher higher than 10ms in most of the iterations and there are some iterations with a maximum time higher than 30ms. Again for calculating these times per iteration, only the active cases have been considered. At this point it can be stated that the derivative based method needs more iterations. However, each iteration is much faster in the proposed BFS based method.

Fig. 12 represents the algorithm behavior with respect to the time. In Fig. 12a) the cumulative solved cases (in percentage) versus the time can be compared for both methods. Then, in Fig. 12b) the mean error of the active cases versus the time is compared for both algorithms. In this figure, the advantage of the proposed method in terms of time saving is notable.

Finally, In Table I, the behavior of both algorithms, the derivative based and the BFS based is summarized and compared. The derivative based method invest a mean number

| | BFS based method | Derivative based method |
|---|---|---|
| Minimum total time (ms) | 8.66 | 24.38 |
| Mean total time (ms) | 16.25 | 44.38 |
| Maximum total time (ms) | 58.31 | 185.90 |
| Total time (Std.Dev.) | 6.36 | 14.30 |
| Minimum number of iterations | 4 | 3 |
| Mean number of iterations | 6.99 | 4.40 |
| Maximum number of iterations | 17 | 13 |
| Number of Iterations (Std.Dev) | 2.33 | 1.58 |
| Minimum time per iteration (ms) | 1.96 | 6.14 |
| Mean time per iteration (ms) | 2.30 | 10.41 |
| Maximum time per iteration (ms) | 5.46 | 38.64 |
| Time per iteration (Std.Dev.) | 0.29 | 2.00 |

TABLE I: Summary of the comparison between the two methods.



a) Solved cases vs. of iterations (Proposed method).

b) Solved cases vs. iterations (Derivative based method).

c) Iterations vs. Error (Proposed method).

d) Iterations vs. Error (Derivative based method).

e) Time per iteration (Proposed method).

f) Time per iteration (Derivative based method).

Fig. 11: Convergence behaviour of the proposed algorithm compared with the derivative based one in terms of error per iteration and time per iteration.

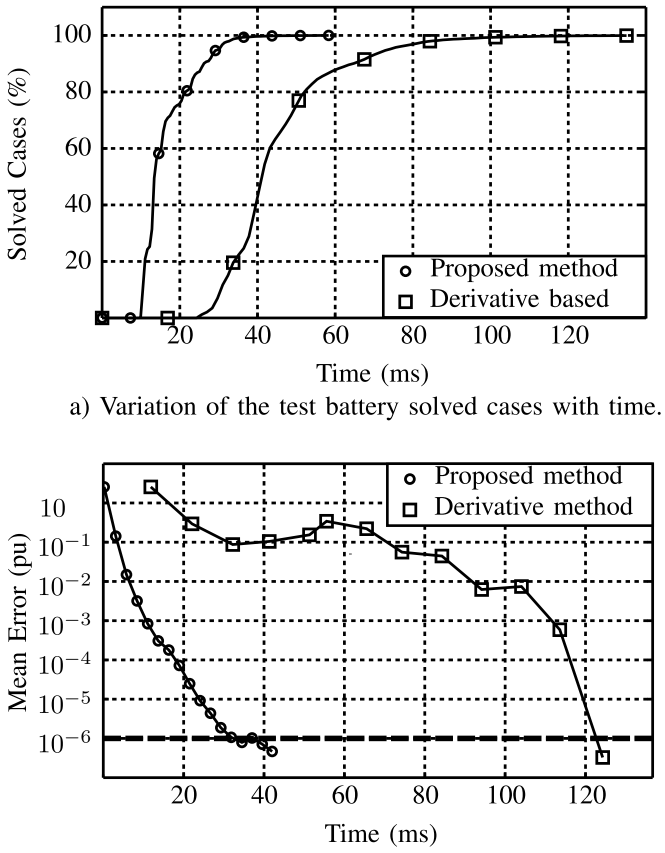
of iterations of 4.4, while the BFS based mean number of iterations is nearly 7. However, the iterations in the BFS based algorithm are much faster, 2.3 ms versus 10.41 ms in the derivative based. The total mean time for solving one instant with the new algorithm is 16.25ms. The time saving with respect to the previous software is more than 270%.

## V. Conclussions

A modified Backward/Forward based power flow algorithm for solving weakly meshed DC traction networks with trains equipped with regenerative braking system was proposed and evaluated in this paper. The main contributions of this work deals with the compact matrix formulation of the algorithm and the ability for solving meshed power systems with non-smooth voltage-dependent loads or generators. This last feature allows the authors to simulate the important effect in the network of the squeeze control and the overcurrent protection of the trains. Comparing the proposed BFS based algorithm with the traditional derivative based one, the authors can determine that the proposed algorithm is equivalent in terms of accuracy, but much more robust and faster. From the analysis of the results, it can be stated that the proposed BFS based methods for solving DC traction networks have been revealed as very effective.

## References

[1] P. Arboleya, G. Diaz, and M. Coto, "Unified ac/dc power flow for traction systems: A new concept," *Vehicular Technology, IEEE Transactions on*, vol. 61, no. 6, pp. 2421–2430, 2012.



a) Variation of the test battery solved cases with time.

b) Mean error variation relative to time.

Fig. 12: Proposed BFS based method Vs. derivative based method.

[2] M. Coto, P. Arboleya, and C. Gonzalez-Moran, "Optimization approach to unified ac/dc power flow applied to traction systems with catenary voltage constraints," *International Journal of Electrical Power & Energy Systems*, vol. 53, no. 0, pp. 434 – 441, 2013.

[3] L. Abrahamsson and L. Soder, "Fast estimation of relations between aggregated train power system data and traffic performance," *Vehicular Technology, IEEE Transactions on*, vol. 60, no. 1, pp. 16–29, 2011.

[4] C. Pires, S. Nabeta, and J. Cardoso, "Iccg method applied to solve dc traction load flow including earthing models," *Electric Power Applications, IET*, vol. 1, no. 2, pp. 193–198, 2007.

[5] Y. Tzeng, R. Wu, and N. Chen, "Electric network solutions of dc transit systems with inverting substations," *Vehicular Technology, IEEE Transactions on*, vol. 47, no. 4, pp. 1405–1412, 1998.

[6] ——, "Unified ac/dc power flow for system simulation in dc electrified transit railways," *Electric Power Applications, IEE Proceedings*, vol. 142, no. 6, pp. 345–354, 1995.

[7] C. Pires, S. Nabeta, and J. Cardoso, "Dc traction load flow including ac distribution network," *Electric Power Applications, IET*, vol. 3, no. 4, pp. 289–297, 2009.

[8] W. Liu, Q. zhan Li, and M. Chen, "Study of the simulation of dc traction power supply system based on ac/dc unified newton-raphson method," in *Sustainable Power Generation and Supply, 2009. SUPERGEN '09. International Conference on*, April 2009, pp. 1–4.

[9] Y. Cai, M. Irving, and S. Case, "Iterative techniques for the solution of complex dc-rail-traction systems including regenerative braking," *Generation, Transmission and Distribution, IEE Proceedings*, vol. 142, no. 5, pp. 445–452, 1995.

[10] D. Braunagel, L. Kraft, and J. Whysong, "Inclusion of dc converter and transmission equations directly in a newton power flow," *Power Apparatus and Systems, IEEE Transactions on*, vol. 95, no. 1, pp. 76–88, 1976.

[11] Y. Saad, "Iterative methods for sparse linear systems," *Society for Industrial Applied Mathematics*, vol. 2nd ed., Philadelphia 2003.

[12] P. Arboleya, M. Coto, C. Gonzlez-Morn, and R. Arregui, "On board accumulator model for power flow studies in dc traction networks," *Electric Power Systems Research*, vol. 116, no. 0, pp. 266 – 275, 2014.

[13] Y. Ju, W. Wu, B. Zhang, and H. Sun, "An extension of fbs three-phase power flow for handling pv nodes in active distribution networks," *Smart Grid, IEEE Transactions on*, vol. 5, no. 4, pp. 1547–1555, July 2014.

[14] E. Janecek and D. Georgiev, "Probabilistic extension of the backward/forward load flow analysis method," *Power Systems, IEEE Transactions on*, vol. 27, no. 2, pp. 695–704, May 2012.

[15] D. Shirmohammadi, H. W. Hong, A. Semlyen, and G. X. Luo, "A compensation-based power flow method for weakly meshed distribution and transmission networks," *Power Systems, IEEE Transactions on*, vol. 3, no. 2, pp. 753–762, May 1988.

[16] G. X. Luo and A. Semlyen, "Efficient load flow for large weakly meshed networks," *Power Systems, IEEE Transactions on*, vol. 5, no. 4, pp. 1309–1316, Nov 1990.

[17] D. Rajicic and A. Dimitrovski, "A new method for handling pv nodes in backward/forward power flow for radial and weakly meshed networks," in *Power Tech Proceedings, 2001 IEEE Porto*, vol. 3, 2001, pp. 6 pp. vol.3–.

[18] D. Bhujel, B. Adhikary, and A. K. Mishra, "A load flow algorithm for radial distribution system with distributed generation," in *Sustainable Energy Technologies (ICSET), 2012 IEEE Third International Conference on*, Sept 2012, pp. 375–380.

[19] A. Augugliaro, L. Dusonchet, S. Favuzza, M. Ippolito, S. Mangione, and E. Sanseverino, "Decoupled solution of radial and weakly meshed distribution networks through a backward method," in *Transmission and Distribution Conference and Exposition: Latin America, 2008 IEEE/PES*, Aug 2008, pp. 1–9.

[20] F. Milano, *Power System Modelling and Scripting*. Springer, 2010.

[21] P. Pozzobon, "Transient and steady-state short-circuit currents in rectifiers for dc traction supply," *Vehicular Technology, IEEE Transactions on*, vol. 47, no. 4, pp. 1390–1404, 1998.

[22] Y. Okada, T. Koseki, and K. Hisatomi, "Power management control in dc-electrified railways for the regenerative braking systems of electric trains," *Advances in Transport*, vol. 15, pp. 919–929, 2004.

[23] Y. Okada, T. Koseki, and S. Sone, "Energy management for regenerative brakes on a dc feeding system," in *International Symposium on Speed-up and Service Technology for Railway and Maglev Systems 2003 (STECH03), Tokyo, JAPAN*, August 2003.

[24] C. Mayet, L. Horrein, A. Bouscayrol, P. Delarue, J.-N. Verhille, E. Chattot, and B. Lemaire-Semail, "Comparison of different models and simulation approaches for the energetic study of a subway," *Vehicular Technology, IEEE Transactions on*, vol. 63, no. 2, pp. 556–565, Feb 2014.

[25] T. Steihaug, "The conjugate gradient method and trust regions in large scale optimization," *SIAM Journal on Numerical Analysis*, vol. 20, no. 3, pp. 626–637, 1983.

[26] M. Abdelaziz, H. Farag, E. El-Saadany, and Y.-R. Mohamed, "A novel and generalized three-phase power flow algorithm for islanded microgrids using a newton trust region method," *Power Systems, IEEE Transactions on*, vol. 28, no. 1, pp. 190–201, Feb 2013.

[27] O. Alizadeh Mousavi and R. Cherkaoui, "Maximum voltage stability margin problem with complementarity constraints for multi-area power systems," *Power Systems, IEEE Transactions on*, vol. 29, no. 6, pp. 2993–3002, Nov 2014.

[28] H.-D. Chiang, T.-Q. Zhao, J.-J. Deng, and K. Koyanagi, "Homotopy-enhanced power flow methods for general distribution networks with distributed generators," *Power Systems, IEEE Transactions on*, vol. 29, no. 1, pp. 93–100, Jan 2014.

[29] S. Li, "Power flow modeling to doubly-fed induction generators (dfigs) under power regulation," *Power Systems, IEEE Transactions on*, vol. 28, no. 3, pp. 3292–3301, Aug 2013.

[30] C. Opathella, B. Singh, D. Cheng, and B. Venkatesh, "Intelligent wind generator models for power flow studies in pss," *Power Systems, IEEE Transactions on*, vol. 28, no. 2, pp. 1149–1159, May 2013.

[31] Y. Phulpin, M. Begovic, M. Petit, J.-B. Heyberger, and D. Ernst, "Evaluation of network equivalents for voltage optimization in multi-area power systems," *Power Systems, IEEE Transactions on*, vol. 24, no. 2, pp. 729–743, May 2009.

**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 (with tenure since 2010). Presently his main research interests are focused in the micro-grid and smartgrid modelling and operation techniques, railway traction networks simulation and combined AC/DC power flow algorithms.

**Bassam Mohamed** received the M.Sc degree from the University of Oviedo, Gijón, Spain, in 2014. He is now pursuing his Ph.D studies in the Department of Electrical Engineering at the University of Oviedo. His master thesis was focused on implementing power flow and optimal power solver for transmission networks. Now, he is working on modelling and simulation of AC and DC micro-grid and railway traction networks.

**Cristina González-Morán** (SM' 15) received the M.Sc. and Ph.D. degrees from the University of Oviedo, Gijón, Spain, in 2003 and 2010, respectively, both in Electrical Engineering. She is currently an Associate Professor in the Department of Electrical Engineering at the University of Oviedo. Her areas of interest include renewable energies, distributed generation and microgrids modelling, simulation, design and optimisation.

**Islam El-Sayed** was born in Zagazig, Egypt, in 1984. He received the B.Sc. degree in electrical engineering from Zagazig University in 2006. He received the Ph.D. degree in the Department of Electrical, Computer, and Systems Engineering, University of Oviedo, Gijón, Spain, in 2012. His ongoing research is focused on sensor-less control of active magnetic bearings and implementation and optimisation of network analysis algorithms.