Load Flow Analysis in a Radial Distribution Systems with DGs

Sonu Koli
2022EEB1215

Nakul
2022EEB1192

Aman Bhagat
2022EEB1152

***Abstract*—This report presents the MATLAB implementation of a novel algorithm for solving the power flow (PF) problem in distributed generation (DG) systems integrated into three-phase distribution networks. The approach combines backward-forward sweep techniques, graph theory, and matrix algebra to effectively address PF challenges. Central to this method is the formulation of two key matrices: one representing the system's structure (connectivity matrix and bus-branch topology matrix) and the other directly involved in PF parameter computation. Notably, the proposed architecture excels in handling matrix size and sparsity issues, enhancing robustness and efficiency compared to existing methods.**

Distributed Generators (DGs) are modelled as PV buses, by creating a matrix that facilitates the computation of required reactive power injections at the required bus.

***Index Terms*—Newton-Raphson Method, Backward Forward Sweep(BFS), Distributed Generators (DGs), Power Flow**

I. Introduction

Power flow analysis is one of the most important tools required for proper planning, analysis and operation of distribution networks. The power flow solution of distribution networks requires special methods to compute the system power flow parameters as it involves calculating the solutions for various sets of non-linear equations. Traditional power flow algorithms such as the Newton-Raphson method, Gauss-Seidel method, and fast decoupled method are used in solving power flow problems for transmission system load flow problems, however they display poor convergence when applied to distribution system load flow problems. This is mainly due to the fact that distribution level networks have a number of distinct features - unbalanced nature of loads, high R/X ratio, presence of mutual impedance in the distribution lines and a large number of buses. One method commonly used in distributed system power flow solutions is the Backward-Forward Sweep (BFS) that iteratively computes the bus voltages and branch currents by sequentially applying KVL and KCL at various nodes on the distribution system. One issue with BFS is its complications in integrating various devices viz., regulators, transformers, and capacitors. Also, BFS involves time consuming and complex sequential calculations that are computationally heavy.

We focus on the algorithm proposed in [1] that combines Backward-Forward sweep with matrix algebra and graph theory to reduce computational complexity and handle all load types (constant power, constant current, constant impedance) and configurations (star, delta). For handling PV bus model of DGs, a sensitivity matrix has been formulated which helps to compute the net reactive current injections for compensating the voltage deviation of the PV bus.

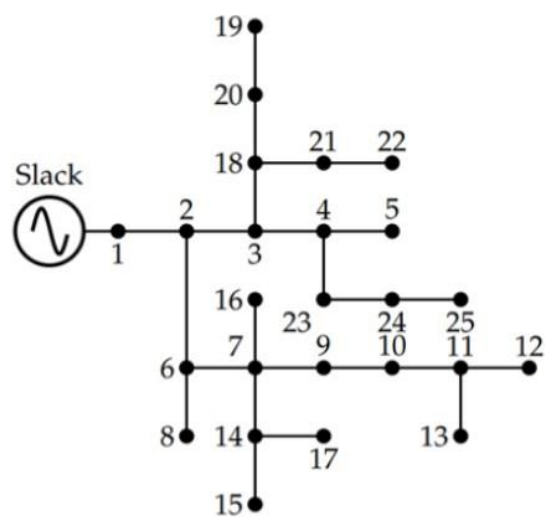


Fig. 1: IEEE 25-BUS Single Line Diagram

II. Designing the Matrices

The topological representation of the 25-bus system which has been modelled in MATLAB has been shown in Fig. 1. For the design, two categories of matrices have been formulated. First, matrices that depict the configuration or structure of the distribution system have been developed. These matrices are the connectivity matrix and bus-branch topology matrix. Second, matrices that are directly involved in computing PF parameters are designed. These matrices are weighted connectivity matrix, bus current beyond branch matrix, branch current matrix, voltage drop matrix, source bus to other bus drop matrix, and voltage matrix. [1]

A. Connectivity Matrix (E) and Weighted Connectivity Matrix (W)

The connectivity matrix is required for identifying the topology of the distribution system, which reveals the number of paths and also the branches involved in those paths. The connectivity matrix defines the connectivity from the source bus to the end bus of the radial distribution network. Its dimensions are p X M, where p represents the number of paths from sources to all leaf nodes, and M is the total number of branches. For our implementation, p is 10 and M is 24. There are in general three E matrices, $[E_a, E_b, E_c]$ (one for each phase), however since our system is balanced, i.e., every branch has three phases, all these matrices will be the same.

$$E_{ij} = \begin{cases} 1 & \text{if phase } \phi \text{ exists in branch } j, \text{ and } j \in i^{th} \text{ path} \\ 0 & \text{otherwise} \end{cases}$$

Here, E_{ij} represents elements of the matrix E at j_{th} branch of the i_{th} path.

As each phase has three components, one self-impedance and mutual impedance because of other phases. These coupling impedances are shown by the weighted connectivity matrix W or dimension p X M (10 x 24 in our case).

$$W_{ij}^{\phi\sigma} = (E_{ij}^{\phi} \times E_{ij}^{\sigma}) \times z_j^{\phi\sigma} \quad (1)$$

Here, ϕ and σ are the phases [a,b,c], i,j represent the path and branch numbers respectively and z represents impedence between phases. We will then have nine such matrices $[W_{aa}, W_{ab}, W_{ac}, W_{ba}, W_{bb}, W_{bc}, W_{ca}, W_{cb}, W_{cc}]$ depicting the relation between all phases taken two at a time.

B. Bus-Branch Topology Matrix (T)

This matrix represents the topology of the network and shows all the buses that come after each line section or branch. It represents the network topology matrix corresponding to all the phases, and is thus helpful in finding the branch phase current. T consists of three bus-branch topology matrices of the order M X N each, which in our case is of dimension 24 x 25.

$$T_{ij} = \begin{cases} 1 & \text{if phase } \phi \text{ of } j_{th} \text{ bus exist beyond branch } i \\ 0 & \text{if phase } \phi \text{ of } j_{th} \text{ bus does exist beyond } i \end{cases}$$

Here, T_{ij} is the element of T Matrix

C. Bus Current Beyond Branch Matrix (I) Computation

The bus current matrix stores the load current existing beyond each branch in the network and is calculated corresponding to each phase, and depends on whether the loads are delta or star connected and the type of load (ZIP). Thus, we get three I matrices, each of dimension 24 x 25 (*number of branches(M)* x *number of buses(N)*).

$$I_{ij}^{\phi} = -\frac{(T_{ij}^{\phi} \times (S_j^{\phi})*)}{(V_j^{\phi})*} \times |V_j^{\phi}|^{\eta} \quad (2)$$

Here, this is used for Star Load, where, ϕ is the phase, and η is the type of load (ZIP) $[\eta = 0, 1, 2]$.

D. Branch Current Matrix (J) Computation

The J matrix represents the total current flowing in a certain path, and is found by the summation of the branch currents of all branches belonging to a certain path. Thus, J_i is the current flowing through phase of branch i and in total we have three branch current matrix i.e., J_a, J_b, J_c of the order $(M \times 1)$ each.

$$J_i^{\phi} = \sum_{i=1} I_{ij} \quad (3)$$

E. Voltage Drop Matrix (D) Computation

This matrix provides the voltage drop in each phase of all the branches existing in the path connecting the slack bus to the leaf bus due to the presence of self and mutual coupling impedance of the branches. Thus, we have nine voltage drop matrices of the order p X M, which is 10 x 24 in our case, each representing the drops between the different phases. The D matrix is used to calculate the total voltage drop from the source to any bus.

$$D_{ij}^{\phi\sigma} = W_{ij}^{\phi\sigma} \times J_i^{\sigma} \quad (4)$$

F. Source to Other Bus Drop Matrix (L) Computation

The L matrix determines the voltage drop in a phase ϕ from the source bus to all other buses. Thus, we have three L matrices for each phase, where L_{ij} for a particular phase is calculated by summing all the branch voltage drops from the source bus to the particular bus. Thus, this matrix is of order p X M, which is 10 x 24 in our case. This matrix is essential in the calculation of the bus voltages in each iteration and thus in the voltage updation.

$$L_{ij}^{\varphi} = \begin{cases} \sum_{j=1}^{j} D_{ij}^{\varphi a} + D_{ij}^{\varphi b} + D_{ij}^{\varphi c}, & \text{if } \hat{E}_{ij}^{\varphi} = 1 \\ 0 & \text{if } \hat{E}_{ij}^{\varphi} = 0 \end{cases}$$

G. Bus Voltage (V) Computation

The bus voltage matrix (V_{ϕ}) is of the order $(N \times 1)$ for each phase , and it depends upon two significant matrices formulated for the respective phases i.e., L_{ϕ} and E_{ϕ}. We essentially find the max drop of each bus from the slack bus, and thus compute the voltage at that particular bus per phase for that iteration. This is the matrix that is used to test for convergence. Other than these defined matrices, we defined

$$V_{i+1}^{a} = Max\{(1\angle 0^{\circ}(\hat{E}_{1i}^{a}) - L_{1i}^{a}), (1\angle 0^{\circ}(\hat{E}_{2i}^{a}) - L_{2i}^{a}), \ldots \\ \ldots\ldots (1\angle 0^{\circ}(\hat{E}_{\rho i}^{a}) - L_{\rho i}^{a})\}$$

two more matrices- the S and Z matrices, which are the power and impedance matrices that are specific to the network and are used in the computations of these pre-defined matrices.

III. Backward Forward Sweep Using Matrices

In this algorithm, we first initialised the connectivity, weighted connectivity and bus branch topology matrix and the branch voltage matrix was initialised at 1 pu with the corresponding phase (0, -120deg, 120deg). Then, we started the loop which would iterate through the network matrices. Inside this loop, our defined S matrix which is specific to the 25-bus system, along with the initial branch voltage matrix were used to calculate the bus current beyond branch matrix, which was consequently used to calculate the branch current matrix. It should be noted here that since a 25-bus system contains star connected loads and all the loads are constant power loads, we use the appropriate value of η . Also, the S matrix and the Z matrix which are specific to the 25-bus system, were converted to per unit values based on the data given in the paper (300MVA and 4.16KV base). Next, the J matrix along with the weighted connectivity matrix are used to calculate the voltage drop matrix which is further used to calculate the source to other bus drop matrix. Now, based on the source to other bus drop matrix, the new bus voltage values are calculated and compared to the previous bus voltage values. Here, we have checked whether the change in the bus voltage of each bus is smaller than a threshold, and updated only if the difference is larger than the threshold. Once all the buses have minimal change in their bus voltages (lesser than the set threshold value), the loop ends and the system has converged.

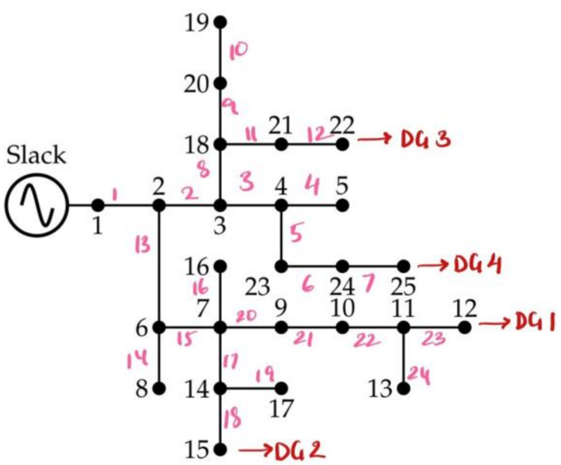


Fig. 2: IEEE 25-BUS System with DGs Included and Branch Details

IV. Current Injection

Within the domain of power flow analysis, a crucial aspect of power system engineering, current injection presents itself as a valuable technique for solving for voltage and current within the system, particularly advantageous for unbalanced three-phase systems. This approach deviates from the traditional methodology that treats specific grid points,

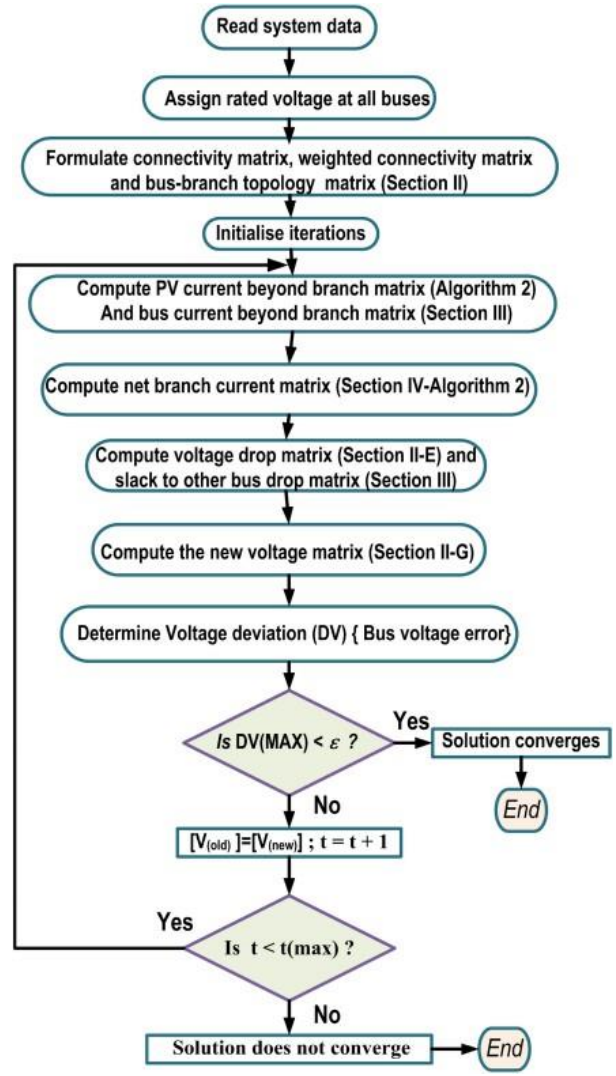


Fig. 3: Flowchart of Power Flow Algorithm

such as generators and loads, as injecting either real or reactive power.

Current injection algorithms leverage current injections as unknowns instead. In the context of Distributed Generators (DGs) – which in reality are PV buses maintaining constant voltage – the algorithm employs a strategic approach. Initially, for the backward-forward sweep calculations, DGs are treated as PQ buses (constant power injection). This allows the iterative process to converge and determine initial voltage magnitudes for all buses in the system.

Following convergence, the algorithm meticulously compares the converged voltage magnitudes of these (assumed) PQ buses (originally DGs) with their designated voltage setpoints as PV buses. This comparison reveals the voltage difference that necessitates correction. By leveraging the well-established

Branch	Sending end	Receiving end	Conductor type	Length (ft)	Receiving end load (kVA)		
					Phase a	Phase b	Phase c
1	1	2	1	1000	0	0	0
2	2	3	1	500	$35 + j25$	$40 + j30$	$45 + j32$
3	2	6	2	500	$40 + j30$	$45 + j32$	$35 + j25$
4	3	4	1	500	$50 + j40$	$60 + j45$	$50 + j35$
5	3	18	2	500	$40 + j30$	$40 + j30$	$40 + j30$
6	4	5	2	500	$40 + j30$	$40 + j30$	$40 + j30$
7	4	23	2	400	$60 + j45$	$50 + j40$	$50 + j35$
8	6	7	2	500	0	0	0
9	6	8	2	1000	$40 + j30$	$40 + j30$	$40 + j30$
10	7	9	2	500	$60 + j45$	$50 + j40$	$50 + j35$
11	7	14	2	500	$50 + j35$	$50 + j40$	$60 + j45$
12	7	16	2	500	$40 + j30$	$40 + j30$	$40 + j30$
13	9	10	2	500	$35 + j25$	$40 + j30$	$45 + j32$
14	10	11	2	300	$45 + j32$	$35 + j25$	$40 + j30$
15	11	12	3	200	$50 + j35$	$60 + j45$	$50 + j40$
16	11	13	3	200	$35 + j25$	$45 + j32$	$40 + j30$
17	14	15	2	300	$133.3 + j100$	$133.3 + j100$	$133.3 + j100$
18	14	17	3	300	$40 + j30$	$35 + j25$	$45 + j32$
19	18	20	2	500	$35 + j25$	$40 + j30$	$45 + j32$
20	18	21	3	400	$40 + j30$	$35 + j25$	$45 + j32$
21	20	19	3	400	$60 + j45$	$50 + j35$	$50 + j40$
22	21	22	3	400	$50 + j35$	$60 + j45$	$50 + j40$
23	23	24	2	400	$35 + j25$	$45 + j32$	$40 + j30$
24	24	25	3	400	$60 + j45$	$50 + j30$	$50 + j35$

Type	Impedance (ohms/mile)		
	a	b	c
1a	$0.3686 + j0.6852$	$0.0169 + j0.1515$	$0.0155 + j0.1098$
1b	$0.0169 + j0.1515$	$0.3757 + j0.6715$	$0.0188 + j0.2072$
1c	$0.0155 + j0.1098$	$0.0188 + j0.2072$	$0.3723 + j0.6782$
2a	$0.9775 + j0.8717$	$0.0167 + j0.1697$	$0.0152 + j0.1264$
2b	$0.0167 + j0.1697$	$0.9844 + j0.8654$	$0.0186 + j0.2275$
2c	$0.0152 + j0.1264$	$0.0186 + j0.2275$	$0.9810 + j0.8648$
3a	$1.9280 + j1.4194$	$0.0161 + j0.1183$	$0.0161 + j0.1183$
3b	$0.0161 + j0.1183$	$1.9308 + j1.4215$	$0.0161 + j0.1183$
3c	$0.0161 + j0.1183$	$0.0161 + j0.1183$	$1.9337 + j1.4236$

Fig. 4: IEEE 25-BUS System Data

formula I = YV (current equals admittance times voltage), the algorithm calculates the precise amount of reactive current that needs to be injected at these specific DGs. Since the objective is to regulate voltage, the injected currents are predominantly reactive in nature, influencing the imaginary component of power. This strategic injection of reactive current allows the algorithm to achieve the desired voltage magnitudes at the DGs without affecting their real power output. The current injection approach offers enhanced efficiency for complex systems and greater flexibility when dealing with unknown power factors, making it a valuable tool for power flow analysis in modern distribution networks.

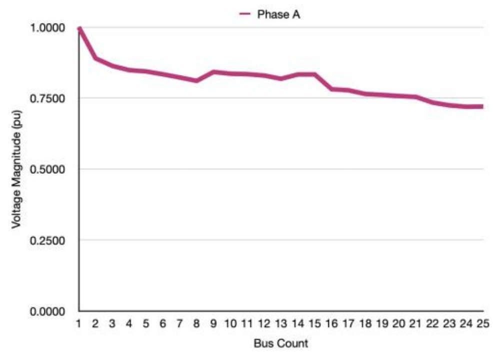
V. Future Scope

Distributed generation (DG) introduces several challenges that demand attention. Primarily, integrating DG units into the distribution network transforms it from a passive to an active system, characterized by multiple energy sources and bidirectional power flow. Consequently, this shift impacts various operational parameters such as voltage levels, fault currents, system losses, reliability, and efficiency. To address these complexities, novel solution algorithms are imperative, ones that are both simple and robust, especially in the context of renewable DG resources. These algorithms must efficiently manage the dynamic nature of the network while ensuring accurate analysis and swift adaptation to changing conditions.

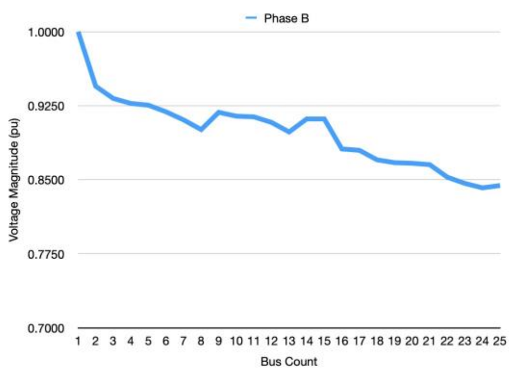
Another critical consideration stems from the fluctuating nature of output power supplied to the network, dictated by continuous variations in wind speeds and solar illumination throughout the day. This variability necessitates frequent and precise load flow analyses to maintain grid stability and performance. The methodology proposed in our research offers a swift and efficient solution, laying the groundwork for future adaptations to accommodate various faults and evolving load flow equations attributed to renewable DGs. Thus, leveraging our approach holds promise for advancing load flow solutions in the face of increasingly complex distributed generation scenarios.

VI. Results and Conclusions

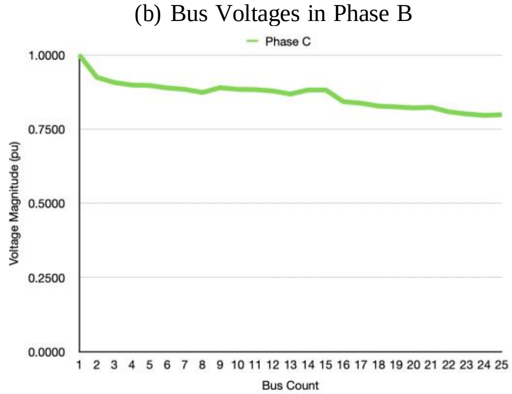
The 25 bus data was taken from an IEEE paper [2]
We iterated the BFS loop several times and kept the error threshold as 0.01%. We observed that the bus voltages converged and plotted the bus voltages per phase as follows:



(a) Bus Voltages in Phase A



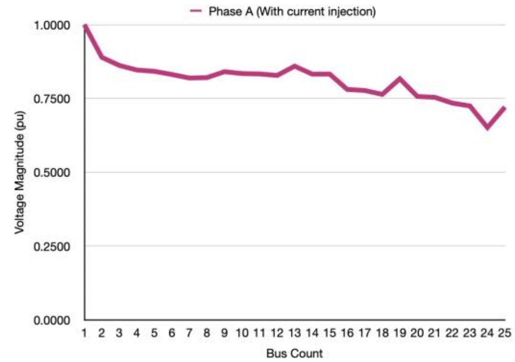
(b) Bus Voltages in Phase B



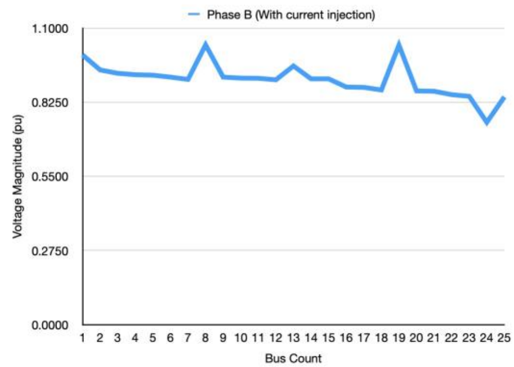
(c) Bus Voltages in Phase C

Fig. 5: Voltage vs Bus Number after BFS Implementation

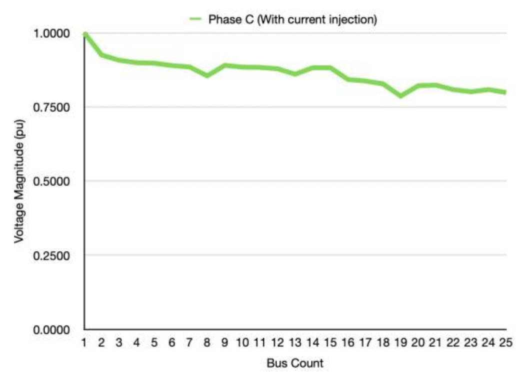
We observed that this method is computationally very effective and fast due to the matrix multiplications of many matrices that just contain binary values.This method, effectively uses information about the topology or structure



(a) Bus Voltages in Phase A



(b) Bus Voltages in Phase B



(c) Bus Voltages in Phase C

Fig. 6: Voltage vs Bus Number after Current Injection

of the network and the computations involved to come up with an efficient and quick load flow solution.
Thus, to conclude, in this paper, an algorithm for three phase power flow solutions for a network integrated with DGs has been derived using graph theory approach. It has used two classes of matrices i.e., the configuration depiction matrices (connectivity matrix and bus-branch topology matrix), and the power flow computation matrices (bus current beyond branch matrix, branch current matrix, voltage drop matrix, source bus to other bus drop matrix, and voltage matrix). The first category of matrices reveals the system's topology or structure while the second category is used in the computation

of the power flow parameters. These matrices have been formulated such that we can treat the three-phase network as a single phase load flow problem instead. The size of all the matrices are similar to the size of the network's single-phase equivalent. Hence, the size of matrices and sparsity problems can be handled in a better way than contemporary approaches, which makes this algorithm very robust and efficient

REFERENCES

[1] Krishna Murari and Sukumar Kamalasadan. "Graph Based Power Flow Approach for Single Phase Distribution System with Distributed Generators(DGs) Considering All Load Types". In: (2021), pp. 1–8. DOI: 10.1109/IAS48185.2021.9677418.

[2] Padarbinda Samal and Sanjib Ganguly. "A modified forward backward sweep load flow algorithm for unbalanced radial distribution systems". In: (2015), pp. 1–5. DOI: 10.1109/PESGM.2015.7286413.