Object Oriented State Estimation for Radial Distribution Systems

Giovanni M. Casolino, *Member, IEEE*, Arturo Losi, *Member, IEEE*

Abstract— **Object–oriented (OO) state estimation (SE) is presented for radial distribution systems. The SE problem is formulated as usual; an OO modeling of the distribution systems is presented, where classes yield the equations that describe the SE problem. The OO implementation of the method is presented, and its most relevant features are discussed. The application of the proposed method and OO implementation to a 69-branch system shows the viability of the approach.**

Keywords— **Power distribution, state estimation, object–oriented methods.**

I. Introduction

STATE estimation allows to optimally estimate the current static operating point of a power system, starting from a set of redundant real–time measurements [1–3]. State estimation is a system-wide optimization problem; decentralized two-level methods have been proposed for transmission systems [4–7] and also for distribution systems [8].

Developed either for transmission systems or for distribution systems (with methods proposed that account for their peculiar characteristics [9–14]) the main effort of the proposals has usually been to enhance the computational efficiency of the methods. On the other hand, modern large–scale computer–based management systems, such as Distribution Management Systems (DMSs), adopt the open systems approach, since it offers significant features such as flexibility, expansibility, easy maintenance and upgrade [15–17]. In open architectures, distributed computation can be adopted; it can significantly increase the computing capacity, with less demanding requirements on computational efficiency.

Within the open systems approach, the software is developed upon the object–oriented (OO) programming paradigm (OOP) [18]. The OOP makes it possible to obtain a direct correspondence between real objects (system components) and programming objects; it eases the adoption of a single component/object database for all the DMS functions (with no need of conversion of names and numbering of components), the treatment of system topology changes, and the introduction of new components.

The full exploitation of the OOP can not be obtained by simply recoding the application software [19]; it calls for a deep revision of both the description of the distribution systems and the methods that realize the functionalities of the DMS.

This work was supported by Ministero dell'Istruzione, Università e Ricerca, Italy.

The authors are with the Dipartimento di Ingegneria Industriale, Università degli Studi di Cassino, Via G. Di Biasio 43, 03043 Cassino (FR), Italy (e-mail {casolino, losi}@unicas.it).

The focus on OOP within a DMS has been the key point of work on the load flow for radial distribution systems [20], extended to weakly meshed topologies [21] with the inclusion of dispersed generation [22], and on the State Estimation (SE) for radial distribution systems [23]. In this paper, attention is again on the SE application for radial topologies, extended to include bada data processing.

The SE problem is formulated as usual, with equality constraints where appropriate. An OO modeling of the distribution system is proposed; classes are characterized by variables and equations that describe the SE problem. The SE problem is solved with the widely adopted Gauss-Newton method, implemented in a OO algorithm; incorporation of new devices and computational issues are discussed. The results obtained for a 69-branch distribution system are presented and commented upon.

II. State estimation problem

The analytical relationships between state variables and measurements,

$$\mathbf{z} = \mathbf{h}(\mathbf{x}) + \mathbf{e}, \tag{1}$$

are the basis of the SE problem. In (1), $\mathbf{z}$ is the m-vector of measurements (all vectors are column vectors), $\mathbf{x}$ is the n-vector of state variables, $\mathbf{h}$ is the m-vector of nonlinear functions which relate measurements to state variables, and $\mathbf{e}$ is the m-vector of measurement errors, assumed to be independent random variables with normal distribution and zero mean.

A. Problem formulation

The aim of the SE is obtaining the maximum likelihood (ML) estimate of the state for a given set of measurements. With the above assumptions on the measurement errors, the ML estimate becomes a Weighted Least Square (WLS) one. Equalities are added to the WLS problem to handle exact measurements (such as zero injections), to represent parts of the system for which Ohm's law is not appropriate, to constrain the describing variables if they are not a minimal set, etc. [2, 3, 8]. The result is the constrained nonlinear minimization problem:

$$\begin{aligned} &\min\ J(\mathbf{x}) = [\mathbf{z} - \mathbf{h}(\mathbf{x})]'\,\mathbf{W}[\mathbf{z} - \mathbf{h}(\mathbf{x})], \\ &\text{subject to } \mathbf{c}(\mathbf{x}) = \mathbf{0}. \end{aligned} \tag{2}$$

In (2), symbol $'$ represents transposition, $\mathbf{W}$ is the $m \times m$ diagonal matrix of the weights squared associated with the measurements, equal to the inverse of the diagonal covariance matrix of the measurement errors, $\mathbf{x}$ is the n-vector of describing variables, and $\mathbf{c}(\mathbf{x})$ is the r-vector of equality constraint nonlinear functions ($r \le n$, $n \le m + r$).

A.1 First-order optimality conditions

Let the Lagrangian function for problem (2) be:

$$L(\mathbf{x}, \boldsymbol{\lambda}) := \frac{1}{2} J(\mathbf{x}) + \boldsymbol{\lambda}' \mathbf{c}(\mathbf{x}), \tag{3}$$

where $\boldsymbol{\lambda}$ is the r-vector of Lagrange multipliers associated with the constraints (2.2); factor 1/2 eases the subsequent development (while halving the value of $\boldsymbol{\lambda}$). The solution satisfies the first–order (F–O) necessary conditions:

$$\begin{aligned} L_{\mathbf{x}}(\mathbf{x}, \boldsymbol{\lambda}) &= -\mathbf{h}_{\mathbf{x}}(\mathbf{x}) \mathbf{W}[\mathbf{z} - \mathbf{h}(\mathbf{x})] + \mathbf{c}_{\mathbf{x}}(\mathbf{x}) \boldsymbol{\lambda} &= \mathbf{0}, \\ L_{\boldsymbol{\lambda}}(\mathbf{x}, \boldsymbol{\lambda}) &= \mathbf{c}(\mathbf{x}) &= \mathbf{0}. \end{aligned} \tag{4}$$

In Eq. (4) the symbols $L_{\mathbf{x}}(\mathbf{x}, \boldsymbol{\lambda})$ and $L_{\boldsymbol{\lambda}}(\mathbf{x}, \boldsymbol{\lambda})$ denote the n-vector and the r-vector of the derivatives of the scalar functions $L(\mathbf{x}, \boldsymbol{\lambda})$ with respect to $\mathbf{x}$ and $\boldsymbol{\lambda}$, respectively; the symbols $\mathbf{h}_{\mathbf{x}}(\mathbf{x})$ and $\mathbf{c}_{\mathbf{x}}(\mathbf{x})$ denote the $n \times m$ and $n \times r$ matrices of the derivatives of the m-vector function $\mathbf{h}(\mathbf{x})$ and r-vector function $\mathbf{c}(\mathbf{x})$ with respect to $\mathbf{x}$, respectively.

B. Bad data processing

Large measurement errors or bad data, due for example to meter failure, have to be detected and identified. It can be based on the test on the performance index and on the largest absolute value of the normalized residuals.

With the χ^2 test on the value of the performance index, $J(\widehat{\mathbf{x}})$, detection (only) of a bad data can be attained [3]. Bad data is detected if

$$\begin{aligned} \nu &= m + r - n, \\ J(\widehat{\mathbf{x}}) &\geq \chi^2_{\nu, 1-\alpha}, \end{aligned} \tag{5}$$

where ν is the number of degrees of freedom, $\widehat{\mathbf{x}}$ represents the estimated describing variable vector, and α (for example, 5%) is the assumed probability of being wrong (rejecting the hypothesis of all good data while it is actually true).

Detection and identification of a bad data (if it is not a critical measurement) can be based on the method of normalized residuals: for non-interacting bad data, the largest absolute value of normalized residual corresponds to a bad data [24, 25]. The m-vector of normalized measurement residuals, $\mathbf{r}_N$, is given by:

$$\begin{aligned} \mathbf{r} &= \mathbf{z} - \mathbf{h}(\widehat{\mathbf{x}}), \\ \mathbf{r}_N &= (diag\, \mathbf{R})^{-1/2} \mathbf{r}, \end{aligned} \tag{6}$$

where $\mathbf{R}$ is the covariance matrix of $\mathbf{r}$, and $diag\mathbf{R}$ is its diagonal. Bad data is detected and identified in the j^*-th measurement if

$$\begin{aligned} & j^* = \arg\max_j |\mathbf{r}_N^j|, \\ & |\mathbf{r}_N^{j^*}| > c, \quad \text{Prob}\left(|\mathbf{r}_N^{j^*}| > c \right) = \alpha. \end{aligned} \tag{7}$$

Once detected and identified, bad data is removed, or it is made ineffective by appropriately modifying the value of the measurement [25].

III. Object–oriented distribution system modeling and state estimation problem

The key concept of the object–oriented (OO) modeling of a system is the *class*, a programming entity that represents the set of objects with similar properties and behaviour. Classes refer to concepts recognized and understood in the real world [26]; base class (or classes) capture aspects common to all (or to large sets of) objects/concepts, from which other classes are derived with refinements and specifications, to form a class hierarchy useful to describe the system at different levels of abstraction.

Our base class is the abstract class *connection*; it has ports, port variables, and computational methods; a port is said to be either ingoing or outgoing, depending on the conventional direction of powers assumed at the port, inwards or outwards. Class *connection* and the derived classes have been proposed in [21, 22] with computational methods for the solution of the load-flow problem. Here, the methods are extended to solve the state-estimation problem. Abstract class *connection* is further generalized: the ingoing/outgoing nature of a given port is not anymore an embedded characteristic but is made dependent on the configuration of the system the *connection* belongs to; an appropriate method of *connection* sets the nature of each port based on the direction in which the connection is traversed in the oriented graph of the network.

Ingoing and outgoing port variables are identified with superscript i and o, respectively. For balanced distribution systems (in steady-state), an incoming port is characterized by four variables, the real and imaginary component of the voltage and the active and reactive powers at the port, $V_{R,i}, V_{I,i}, P_i, Q_i$; the set of the outgoing ports (either one or many) is characterized by the four variables $V_{R,o}, V_{I,o}, P_o, Q_o$.

From *connection*, the following classes are derived:

- *branch*, an abstract class with one ingoing port and one outgoing port (such as a line or a transformer); from it, concrete classes are derived that model specific cases of power injection;
- *root*, a concrete connection with only one outgoing port (the supplying system);
- *fork*, a concrete class with one ingoing port and many outgoing ports (a zero impedance busbar);
- *terminal*, a concrete class with only one ingoing port (a termination of the system);
- *tie-switch*, a connection with two ports and two states: open or closed.

The class hierarchy is shown in Fig. 1. The model of a radial network is an oriented graph of object instances appropriately connected. It origins at root and ends at terminals and open tie-switches; the connected ports are ingoing/outgoing pairs, and and no port is unconnected.

For the subsequent development, some notation related to the oriented graph is useful. Let $p(j)$ represent the parent of the j–th connection, the one that precedes it in the oriented graph; $p(j)$ is empty for the root. Let $C(j)$ represent the children of the j–th connection, the set of connec-

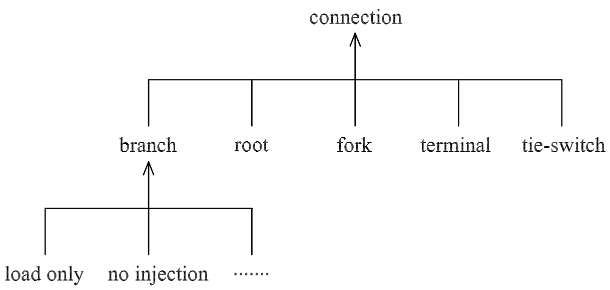


Fig. 1. Class hierarchy for distribution system modeling.

tions that follow it in the graph; $C(j)$ is empty for terminals.

In the following, the equations that characterize each class are shown, for the representation of the SE problem in steady-state balanced operation of the system.

A. Class connection

The abstract class *connection* is characterized by the weighted error function; for the generic j-th *connection*, it is:

$$J^j(\mathbf{x}^j) = \left[\mathbf{z}^j - \mathbf{h}^j(\mathbf{x}^j)\right]' \mathbf{W}^j \left[\mathbf{z}^j - \mathbf{h}^j(\mathbf{x}^j)\right], \tag{8}$$

where $\mathbf{x}^j$ represents the port variables pertaining to the connection. Derived classes provide the specification of $\mathbf{z}^j$, $\mathbf{h}^j$ and $\mathbf{W}^j$, with the details of what measurements can be carried out for that specific *connection*; a zero weight accounts for the lack of a certain measurement.

The abstract class *connection* is also characterized by equality constraints, $\mathbf{c}^j = 0$, again specified by the derived classes.

B. Class root

Class *root* represents the supplying system, possibly at higher voltage level. It has only one outgoing port.

For the *root*, we assume there can be only the measurement of the voltage amplitude at the outgoing port (the uppercase superscript denotes the measurement type)

$$\mathbf{h}^{V\,j} = \sqrt{(V^j_{R,o})^2 + (V^j_{I,o})^2}, \tag{9}$$

with $j = root$.

The constraints of *root* express:

– the equality to zero of the imaginary part of the root outgoing voltage

$$\mathbf{c}^{D_I\,root}(\mathbf{x}^{root}) \equiv V^{root}_{I,o}; \tag{10}$$

– the active and reactive balances between the outgoing powers and the ingoing powers of the subsequent connection,

$$\begin{aligned} \mathbf{c}^{P\,j}(\mathbf{x}) &\equiv P^j_o - P^{C(j)}_i, \\ \mathbf{c}^{Q\,j}(\mathbf{x}) &\equiv Q^j_o - Q^{C(j)}_i. \end{aligned} \tag{11}$$

with $j = root$.

C. Class branch and some derived classes

A *branch* models lines and transformers, and power injection (such as loads and/or distributed generation) at one busbar. Class *branch* has one ingoing port and one outgoing port; its circuit representation is one of the two depicted in Fig. 2, depending on the direction the branch is traversed in the oriented graph. In writing equations for *branch*, we refer to the case of Fig. 2.a; similar expressions can be easily derived in the case of Fig. 2.b.

The measurements at a *branch* are :

– voltage amplitude at the load busbar (9);

– current amplitude

$$\mathbf{h}^{I_b\,j} = \sqrt{\frac{(P^j_i)^2 + (Q^j_i)^2}{(V^j_{R,i})^2 + (V^j_{I,i})^2}}; \tag{12}$$

– branch active and reactive powers

$$\begin{aligned} \mathbf{h}^{P_b\,j} &= P^j_i, \\ \mathbf{h}^{Q_b\,j} &= Q^j_i; \end{aligned} \tag{13}$$

– active and reactive powers injections

$$\begin{aligned} \mathbf{h}^{P_{in}\,j} &= P^j_o - P^j_i + \mathrm{R}^j \frac{(P^j_i)^2 + (Q^j_i)^2}{(V^j_{R,i})^2 + (V^j_{I,i})^2}, \\ \mathbf{h}^{Q_{in}\,j} &= Q^j_o - Q^j_i + \mathrm{X}^j \frac{(P^j_i)^2 + (Q^j_i)^2}{(V^j_{R,i})^2 + (V^j_{I,i})^2}; \end{aligned} \tag{14}$$

Equality constraints assigned to a *branch* express:

– the equality between the voltage (real and imaginary parts) at the ingoing port and the voltage at the outgoing port of the parent

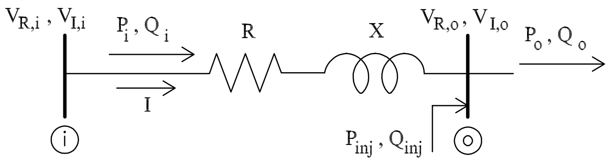
$$\begin{aligned} \mathbf{c}^{V_R\,j}(\mathbf{x}) &\equiv V^{p(j)}_{R,o} - V^j_{R,i}, \\ \mathbf{c}^{V_I\,j}(\mathbf{x}) &\equiv V^{p(j)}_{I,o} - V^j_{I,i}; \end{aligned} \tag{15}$$

– the voltage drop (real and imaginary parts)

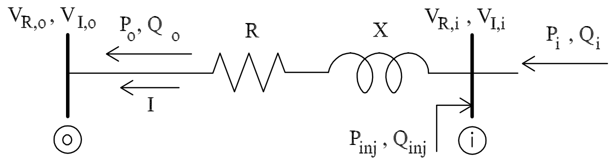
$$\begin{aligned} \mathbf{c}^{D_R\,j}(\mathbf{x}^j) \equiv{}& V^j_{R,i} - V^j_{R,o} - \\ & \frac{V^j_{R,i}(\mathrm{R}^j P^j_i + \mathrm{X}^j Q^j_i) - V^k_{I,i}(\mathrm{X}^j P^j_i - \mathrm{R}^j Q^j_i)}{(V^j_{R,i})^2 + (V^j_{I,i})^2}, \\ \mathbf{c}^{D_I\,j}(\mathbf{x}^j) \equiv{}& V^j_{I,i} - V^j_{I,o} - \\ & \frac{V^j_{R,i}(\mathrm{X}^j P^j_i - \mathrm{R}^j Q^j_i) + V^j_{I,i}(\mathrm{R}^j P^j_i + \mathrm{X}^j Q^j_i)}{(V^j_{R,i})^2 + (V^j_{I,i})^2}; \end{aligned} \tag{16}$$

– the active and reactive balances between the outgoing powers and the ingoing powers of the child connection (11).

From abstract class *branch*, concrete classes are derived that models specific cases of power injection, in particular loads and distributed generation (DG) devices. In the following examples, we describe two basic cases; in Sect. V we illustrate how to model other configurations.



(a)Traversed from left to right



(b)Traversed from right to left

Fig. 2. Circuit representation of abstract class *branch*.

Load only – The usual case of a *branch* feeding a load is modeled by specifying that the measured load powers enter with the minus sign in the evaluation of the terms related to measurements (14).

No injection – For a zero injection *branch* (no load–no DG), two exact measurements constraints are added, and the weights associated with measurements (14) are set to zero:

$$\begin{aligned}\mathbf{c}^{P_{in}\,j} &= P_o^j - P_i^j + \mathrm{R}^j \frac{(P_i^j)^2 + (Q_i^j)^2}{(V_{R,i}^j)^2 + (V_{I,i}^j)^2},\\ \mathbf{c}^{Q_{in}\,j} &= Q_o^j - Q_i^j + \mathrm{X}^j \frac{(P_i^j)^2 + (Q_i^j)^2}{(V_{R,i}^j)^2 + (V_{I,i}^j)^2}.\end{aligned} \tag{17}$$

D. Class fork

Class *fork* describes a zero-impedance busbar; it connects one ingoing port to two or more outgoing ports.

For the *fork*, we assume there can be only the measurement of the voltage amplitude at the outgoing ports (9).

The constraints assigned to *fork* are:
- the voltage equality with the parent (15)
- the voltage equality inside the *fork*

$$\begin{aligned}\mathbf{c}^{D_R\,j}(\mathbf{x}^j) &\equiv V_{R,i}^j - V_{R,o}^j,\\ \mathbf{c}^{D_I\,j}(\mathbf{x}^j) &\equiv V_{I,i}^j - V_{I,o}^j;\end{aligned} \tag{18}$$

- the active and reactive power balances inside the fork

$$\begin{aligned}\mathbf{c}^{D_P\,j}(\mathbf{x}^j) &\equiv P_i^j - P_o^j,\\ \mathbf{c}^{D_Q\,j}(\mathbf{x}^j) &\equiv Q_i^j - Q_o^j;\end{aligned} \tag{19}$$

- the active and reactive balances between the outgoing powers and the ingoing powers of the children connections

$$\begin{aligned}\mathbf{c}^{P\,j}(\mathbf{x}) &\equiv P_o^j - \sum_{w\in C(j)} P_i^w,\\ \mathbf{c}^{Q\,j}(\mathbf{x}) &\equiv Q_o^j - \sum_{w\in C(j)} Q_i^w.\end{aligned} \tag{20}$$

E. Class terminal

The *terminal* class describes a "dead" end of the distribution system; it has only one ingoing port.

We assume there is no measurement at the *terminal*.

The constraints of the *terminal* express:
- the voltage equality with the parent (15)
- the null value of the ingoing active and reactive powers:

$$\begin{aligned}\mathbf{c}^{D_P\,j}(\mathbf{x}^j) &\equiv P_i^j,\\ \mathbf{c}^{D_Q\,j}(\mathbf{x}^j) &\equiv Q_i^j.\end{aligned} \tag{21}$$

F. Tie-switch

Class *tie-switch* models a switch which can change the configuration of the radial system; it is a connection with two ports and two states: open or closed. In the open state, it represents an infinite impedance between the two ports, both ingoing; in the closed status, it represents a zero impedance connecting the ports, one ingoing and one outgoing. We assume there are no measurements at a *tie-switch.*

In the closed status, the ports are one ingoing and one outgoing. The constraints of the *tie-switch* in the closed status express:
- the voltage equality with the parent (15)
- the voltage equality inside the switch (18)
- the active and reactive power balances inside the tie-switch (19)
- the power balances with the child connection (11)

In the open state, the ports are both ingoing; a *tie-switch* has two parents, referred to as $p_1(j)$ and $p_2(j)$, no children, and represents two ends of the oriented graph. The constraints of the *tie-switch* in the open state express:
- the voltage equality with the parent, for each port

$$\begin{aligned}\mathbf{c}^{V_{R_1}\,j}(\mathbf{x}) &\equiv V_{R,o}^{p_1(j)} - V_{R,i_1}^j, & \mathbf{c}^{V_{R_2}\,j}(\mathbf{x}) &\equiv V_{R,o}^{p_2(j)} - V_{R,i_2}^j,\\ \mathbf{c}^{V_{I_1}\,j}(\mathbf{x}) &\equiv V_{I,o}^{p_1(j)} - V_{I,i_1}^j, & \mathbf{c}^{V_{I_2}\,j}(\mathbf{x}) &\equiv V_{I,o}^{p_2(j)} - V_{I,i_2}^j;\end{aligned} \tag{22}$$

- the null value of the active and reactive powers at the two ports

$$\begin{aligned}\mathbf{c}^{D_{P_1}\,j}(\mathbf{x}) &\equiv P_{i_1}^j, & \mathbf{c}^{D_{P_2}\,j}(\mathbf{x}) &\equiv P_{i_2}^j,\\ \mathbf{c}^{D_{Q_1}\,j}(\mathbf{x}) &\equiv Q_{i_1}^j, & \mathbf{c}^{D_{Q_2}\,j}(\mathbf{x}) &\equiv Q_{i_2}^j.\end{aligned} \tag{23}$$

G. Problem characteristics

The SE problem, with the above description of the classes, has the following characteristics.

G.1 Describing variables

Root and *terminal*s contribute four variables each, for they have only one port; *branch*es, *tie-switch*es and *fork*s contribute eight variables, four for each port. For a system composed of the *root*, b *branch*es, f *fork*s, s *tie-switch*es and t *terminal*s, the SE problem has

$$n = \dim\{\mathbf{x}\} = 4 + 8b + 8f + 8s + 4t. \tag{24}$$

G.2 Objective function

Let q be total number of *connection*s in the system:

$$q = 1 + b + f + s + t; \tag{25}$$

the objective function of the SE problem is the sum of the contributions of all *connections*:

$$J(\mathbf{x}) = \sum_{j=1}^{q} J^j(\mathbf{x}^j). \tag{26}$$

G.3 Constraints and measurements

The total number of structural equality constraints, r_{str}, contributed by all *connection*s in a network is

$$r_{str} = 3 + 6b + 8f + 8s + 4t; \tag{27}$$

it does not count for the exact measurement constraints, if any.

The difference between the number of describing variables, n, and of the number of structural equality constraints, r_{str}, is:

$$n_{min} = n - r_{str} = 2b + 1; \tag{28}$$

it is the dimension of a state variable set, a minimal set of describing variables. The $2b+1$ state variables could be, for example, the substation voltage amplitude, with the imaginary part equal to zero, and the active and reactive powers flowing in the branches. The sum of the number of measurements actually carried out and of the number of exact measurements (if any) has to be not less than n_{min}.

IV. OO distribution state estimation algorithm

The OO implementation of the Distribution State Estimation (OODSE) is based on local processing steps and message passing. In each processing step, the methods of only one object are applied to the data available to that object; thanks to message passing between neighboring objects, data exchange takes places in between two such steps.

The Gauss-Newton method is implemented; the partial derivatives of the equations that characterize each class are required.

A. Optimization problem solution

The Gauss-Newton method is the Newton's method [27] where the second order derivatives of measurement functions (as well as of equality constraints) are neglected [3,28].

At the k–th iteration, the following linear system is solved for given values of $\mathbf{x}_k$ and $\boldsymbol{\lambda}_k$:

$$\begin{bmatrix} \mathbf{h_x W h'_x} & \mathbf{c_x} \\ \mathbf{c'_x} & \mathbf{0} \end{bmatrix}_k \begin{bmatrix} \boldsymbol{\Delta}\mathbf{x}_k \\ \boldsymbol{\lambda}_{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{h_x W (z - h)} \\ -\mathbf{c} \end{bmatrix}_k, \tag{29}$$

and the variables $\mathbf{x}$ are accordingly updated.

With row–column permutations preserving symmetry, Eq. (29) can be re-written as

$$\begin{bmatrix} \mathbf{A}^{11} & \cdots & \mathbf{A}^{1q} \\ \vdots & \vdots & \vdots \\ \mathbf{A}^{q1} & \cdots & \mathbf{A}^{qq} \end{bmatrix}_k \begin{bmatrix} \mathbf{y}^1 \\ \vdots \\ \mathbf{y}^q \end{bmatrix}_k = \begin{bmatrix} \mathbf{a}^1 \\ \vdots \\ \mathbf{a}^q \end{bmatrix}_k, \tag{30}$$

where vector $\mathbf{y}_h^j$ for the j–th connection represents variation of port variables and Lagrange multipliers (see Appendix VIII-A).

It can be shown that (see Appendix VIII-A)

$$\begin{aligned} &\mathbf{A}^{ij} = \mathbf{0} \Longleftrightarrow \ i \neq j \text{ and } i \notin p(j) \text{ and } i \notin C(j), \\ &\mathbf{A}^{p(j)\,j} = \mathbf{A}^{j\,p(j)\,\prime} = \boldsymbol{\Theta}, \end{aligned} \tag{31}$$

where $\boldsymbol{\Theta}$ is a constant matrix. The solution of the linear system (30) can be obtained with the following recursive equations, for $j = 1, \ldots, q$:

$$\begin{aligned} \widetilde{\mathbf{A}}_k^{j\,j} &= \mathbf{A}_k^{j\,j} - \boldsymbol{\Theta}\Big(\sum_{w \in C(j)} \widetilde{\mathbf{A}}_k^{w\,w^{-1}}\Big)\boldsymbol{\Theta}', \\ \widetilde{\mathbf{a}}_k^{j} &= \mathbf{a}_k^{j} - \boldsymbol{\Theta} \sum_{w \in C(j)} \widetilde{\mathbf{A}}_k^{w\,w^{-1}} \widetilde{\mathbf{a}}_k^{w}, \\ \mathbf{y}_k^{j} &= \widetilde{\mathbf{A}}_k^{j\,j^{-1}}\big(\widetilde{\mathbf{a}}_k^{j} - \boldsymbol{\Theta}'\mathbf{y}_k^{p(j)}\big). \end{aligned} \tag{32}$$

B. Bad data processing

For the Gauss-Newton approach, the covariance matrix $\mathbf{R}$ of the residual vector $\mathbf{r}$ is [29]

$$\mathbf{R} = \mathbf{W}^{-1} - \mathbf{h'_x}\ \mathbf{E}\ \mathbf{h_x}, \tag{33}$$

where $\mathbf{E}$ is the upper left $n \times n$ submatrix of the inverse of the matrix appearing in the lhs of (29).

We are interested only in the diagonal elements of $\mathbf{R}$ [see (6)], which is to say in the diagonal elements of the submatrices $\mathbf{R}^{jj}$ along the principal diagonal. Based on our modeling, it can be shown that (see Appendix VIII-B), for $j = 1, \ldots, q$:

$$\mathbf{R}^{jj} = \mathbf{W}^{j^{-1}} - \mathbf{h}_{\mathbf{x}^j}^{j\,\prime}\ \mathbf{E}^{jj}\ \mathbf{h}_{\mathbf{x}^j}^{j}, \tag{34}$$

where $\mathbf{E}^{jj}$ is the $n^j \times n^j$ submatrix of $\mathbf{E}$ along the principal diagonal, n^j being the number of describing variables of the j-th connection.

The inverse, $\mathbf{F}$, of the the lhs matrix of (30),

$$\begin{bmatrix} \mathbf{F}^{11} & \cdots & \mathbf{F}^{1q} \\ \vdots & \vdots & \vdots \\ \mathbf{F}^{q1} & \cdots & \mathbf{F}^{qq} \end{bmatrix} = \begin{bmatrix} \mathbf{A}^{11} & \cdots & \mathbf{A}^{1q} \\ \vdots & \vdots & \vdots \\ \mathbf{A}^{q1} & \cdots & \mathbf{A}^{qq} \end{bmatrix}^{-1}, \tag{35}$$

could be obtained from the inverse of the lhs matrix of (29) with the same row–column permutations used to get lhs matrix of (30) from (29); so:

$$\mathbf{E}^{jj} = \mathbf{F}_1^{jj} \tag{36}$$

where $\mathbf{F}_1^{jj}$ is the upper left $n^j \times n^j$ submatrix of $\mathbf{F}^{jj}$. In Appendix VIII-B it is shown that matrices $\mathbf{F}^{jj}$ can be recursively computed:

$$\mathbf{F}^{j\,j} = \widetilde{\mathbf{A}}^{j\,j\,^{-1}} \Big(\mathbf{I} + \mathbf{\Theta}' \, \mathbf{F}^{p(j)\,p(j)} \mathbf{\Theta} \, \widetilde{\mathbf{A}}^{j\,j\,^{-1}} \Big), \qquad (37)$$

where matrices $\widetilde{\mathbf{A}}^{j\,j}$ are the same as in (32).

C. OODSE algorithm

The OODSE algorithm can be implemented, based on (32), (37), and on downstream and upstream graph tree traversing and message passing. Matrix $\mathbf{\Theta}$ has many zero rows and columns (see Appendix VIII-A); it can be exploited to simplify the message passing and to reduce the computational cost.

Let ε represent the mismatch on the F–O optimality conditions (4), η the maximum absolute value of normalized residuals (7), θ an 'upstream/downstream' flag. The algorithm can be summarized as follows:

part I – estimation of describing variables

- I.1 root sets θ ='downstream', and initiates a downstream graph traversing;
- I.2 after receiving data from its parent, the j–th connection:
  1. updates its copy of the outgoing voltage (real and imaginary parts) of the parent and stores $\mathbf{y}^{p(j)}$ for the subsequent upstream graph traversing,
  2. updates its variables $\mathbf{y}^j$ with (32.3),
  3. sends downstream $\mathbf{y}^j$, together with θ;
- I.3 at the end of the downstream traversing, each ending connection (terminals and open switches) sets θ ='upstream' and initiates an upstream graph traversing;
- I.4 after receiving data from all its children, the j–th connection (with the exception of the root):
  1. updates ε,
  2. evaluates matrix $\widetilde{\mathbf{A}}^{j\,j\,^{-1}}$ and vector $\widetilde{\mathbf{A}}^{j\,j\,^{-1}} \tilde{\mathbf{a}}^j$, with (32.1) and (32.2),
  3. sends upstream the above quantities, together with the scalars P_i^j, Q_i^j, $\lambda^{V_R j}$, $\lambda^{V_I j}$, and θ;
- I.5 at the end of the upstream traversing, the root:
  1. updates ε,
  2. if ε is not within the prescribed accuracy, updates its variable $\mathbf{y}^{root}$ with (32.3), and starts a new step I.1,
  3. if ε is within the prescribed accuracy, starts part II;

part II.a – bad data processing: detection

- II.1 root sets θ ='downstream' and initiates a downstream graph traversing;
- II.2 after receiving data from its parent, the j–th connection:
  1. evaluates matrix $\mathbf{F}^{j\,j}$ with (37) and vector $|\mathbf{r}_N^j|$ with (6), (7), (34), (36), and stores $|\mathbf{r}_N^j|$ for the subsequent upstream graph traversing;
  2. sends downstream matrix $\mathbf{F}^{j\,j}$;
- II.3 at the end of the downstream traversing, each ending connection sets θ ='upstream' and initiates an upstream graph traversing;
- II.4 after receiving data from all its children, the j–th connection:
  1. sums its contribution to ν (5);
  2. sums its contribution $J^j(\widehat{\mathbf{x}}^j)$ (8) to the overall value of $J(\widehat{\mathbf{x}})$,
  3. updates the value of η,
  4. sends upstream ν, $J(\widehat{\mathbf{x}})$ and η, with the exception of *root* ;
- II.5 at the end of the upstream traversing, the *root*
  1. if no bad data is detected (based on the value of ν, $J(\widehat{\mathbf{x}})$ and η) terminates computations,
  2. if bad data is detected, starts part II.b.

part II.b – bad data processing: removal

- II.6 root sets θ ='downstream' and initiates a downstream graph traversing;
- II.7 after receiving data from its parent, the j–th connection:
  1. if bad data is recognized based on the value of η, removes it, sets θ='upstream' and initiates an upstream traversing,
  2. if no bad data is recognized, sends downstream η;
- II.8 ending connections reached by the downstream traversing set θ='upstream' and initiate an upstream traversing;
- II.9 after receiving data from all its children, the j–th connection continues the upstream traversing;
- II.10 at the end of the upstream traversing, the *root* starts a new estimation of describing variables with I.1.

V. Characteristics of the OODSE algorithm

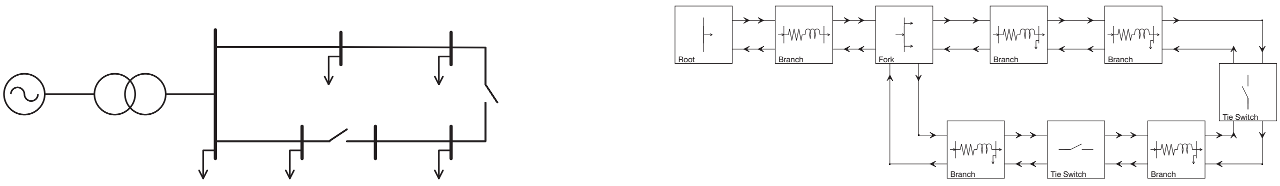
As seen in the previous section, the development of the OODSE algorithm is demanding for it requires the understanding of the message passing with a clear definition of the interface among *connections*, a clever intermediate data storage, and complex computations.

Nevertheless, its features make it attractive. We illustrate the main two: the ease of incorporating new components or models, and the easily obtained limitation of computational cost.

A. Incorporating new components

The introduction of new components or models into the OO algorithm is straightforward. As an example, we show this feature for the introduction of a DG device; the reader is referred to Sect[22] for more details. It is apparent that introducing the new component does not affect the OODSE algorithm in any way.

DG by asynchronous generator – To model a *branch* with power injection from a DG asynchronous generator, the classical relationships (for example in [30]) between active and reactive generated powers, P_g^j and Q_g^j, voltage amplitude, V^j, slip, s^j, and mechanical power provided by



(a) (b)

Fig. 3. A simple distribution system (a) and its graphical OO representation (b).

the prime mover, P_m^j, are assigned to the derived class [22]:

$$\begin{aligned} \mathbf{c}^{P_g^j} &\equiv P_g^j - P_g^j(V^j, s^j), \\ \mathbf{c}^{Q_g^j} &\equiv Q_g^j - Q_g^j(V^j, s^j), \end{aligned} \tag{38}$$

where

$$\begin{aligned} V^j &= \sqrt{(V_{R,o}^j)^2 + (V_{I,o}^j)^2}, \\ s^j &= s^j(V^j, P_m^j). \end{aligned}$$

Electrical powers P_g^j and Q_g^j and prime mover power P_m^j are additional describing variable, to be estimated and possibly measured; the measurement of the injected powers (14) is indeed the measurement of P_g^j and Q_g^j.

B. Computational aspects

The computational efficiency is not the main concern in the OO approach; nevertheless, some considerations about computational issues are of interest.

Computations for the instances of objects are carried out one at a time, with low computing memory requirements.

As regards the computing effort, we focus on the computational cost per iteration, which is the computational cost for all connections per iteration due to (32). Branches outnumber other connections in the network; we can approximate the overall cost by the cost due to the branches. Assuming that the computational cost is measured by the number of multiply-add pairs, in Appendix VIII-C it is shown that for each branch the cost is almost $1.33 \cdot 10^3$; so, the overall cost C_{oo} is

$$C_{oo} \simeq \; 1.33 \cdot 10^3 \times b, \tag{39}$$

where b is the number of branches.

To compare it with the one of a classical method, at first we note that the number of describing variables can be kept as low as $2b + 1$, the dimension of a state variable set for a radial system composed of b branches. Indeed, a modeling with all nodal voltages and branch powers would be much more useful; it would be made of $4b + 1$ describing variables and $2b$ constraints (not counting exact measurements constrains), which sum up to almost $6b$ quantities to be computed. Second, in a classical method the inverse is not required; in addition, exploiting the sparsity the relevant term in the cost C_{sp} of the solution of system (29) can be kept at almost [31]

$$C_{sp} \simeq (6\, b)^2. \tag{40}$$

From (39)– (40), as far as computational cost is considered, it is apparent that for up to a few tens of branches a sparsity oriented classical method is preferable, while the contrary is true for systems with many branches. It has to be noticed that, while both (39) and (40) derive from the exploitation of the sparsity, in classical methods it is system specific and not very flexible for configuration changes, while in the OO algorithm it is easily obtained, and configuration changes are easily accounted for.

Note that the above discussion only gives a rough estimation of the actual computational effort. In particular, neither the issues related to the message passing in the proposed method have been afforded, nor the possibly relevant computational cost of the matrix reordering for sparsity exploitation [31] in classical methods has been considered.

VI. Numerical application

The OODSE application has been developed in the Ptolemy environment [32], an extensible open CAD environment based on the C++ language; any other OO platform would have been equally useful. To implement the OODSE, a library of blocks has been developed, based on the classes defined in Sect. III. The representation of a distribution system is obtained by connecting pairs of ports of the blocks (Fig. 3).

The 69–branch distribution system whose data are reported in [33] has been studied; its graphical representation is reported in Fig. 4. The numerical tests have been conducted with the power measurements at the branches (either actually carried out or taken into account with zero-injection constraints) and the voltage measurement at the root and at the forks; the measurement redundancy (the ratio of the number of measurements to the number of state variables) is 2.04. The measurement standard deviation is 2% for all measurements, and the starting point is the load flow solution for the measured root voltage and load powers. Bad data are introduced in two measurements: a reactive power and a voltage amplitude, as reported in Fig. 4; for bad data processing, α is set to 5%, with a corresponding value of c equal to 1.960 [see (7)].

Convergence is achieved at $k = 3$, 4, 6; bad data is found and measurements are removed at $k = 3$, 4. Figure 5 reports the graph of objective function, $J(\widehat{\mathbf{x}})$, and of the maximum absolute value of normalized residuals $|\mathbf{r}_N^{j^*}|$, versus the iteration count, k, together with the values of $\chi^2_{\nu,1-\alpha}$ and c (the change of the value of $\chi^2_{\nu,1-\alpha}$ at $k = 3$, 4 is not graphically appreciable).

As for computation times, each iteration has required

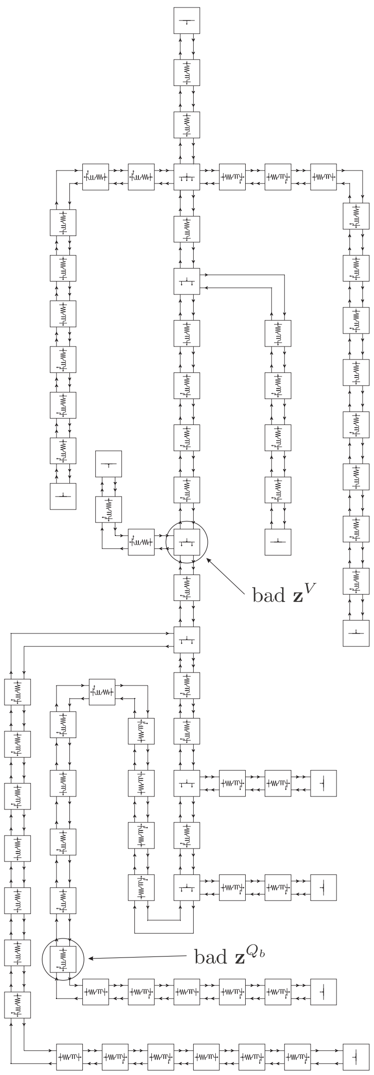


Fig. 4. Graphical OO representation of the 69-branch system.

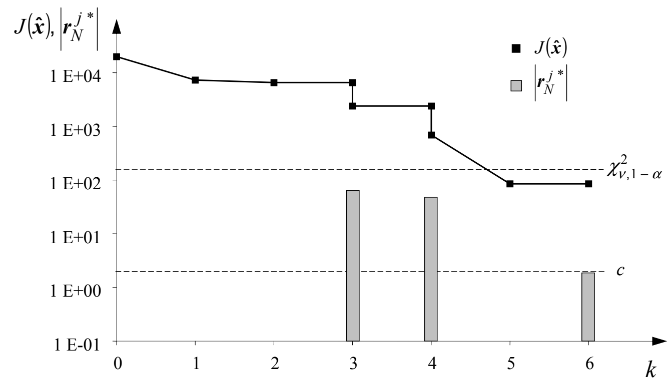


Fig. 5. Objective function, $J(\hat{\mathbf{x}})$, and max_abs value of normalized residuals, $|\mathbf{r}_N^{j*}|$.

about 0.040 s; the application has been developed with a 32-bit library and the tests have been done in double precision on a Workstation Sun-Blade-1000 Ultra-SPARC-III+, run under the SunOS 5.8.

VII. Conclusion

The state estimation problem is formulated for radial distribution networks. The OO approach is adopted for the modeling of the distribution system and for the SE problem description.

A solution method is implemented in a OO algorithm, together with bad data processing.

The proposed method allows a painless introduction of new components and models. Computational cost grows linearly with the dimension of the network, and it can be less than the one of classical method; sparsity exploitation is easily attained also for configuration changes.

In future work the possibility of extending the OO approach to state estimation for weakly meshed topologies will be investigated.

VIII. Appendix

A. Vectors and matrices

Referring to *branch*es (B), in (30) vector $\mathbf{y}_k^j$ is the 14-vector

$$\mathbf{y}_{\substack{k\\ j\in B}}^j = \left[\boldsymbol{\Delta}\mathbf{x}_k^{j\,\prime}\ \boldsymbol{\lambda}_{k+1}^{V_R\,j}\ \boldsymbol{\lambda}_{k+1}^{V_I\,j}\ \boldsymbol{\lambda}_{k+1}^{D_R\,j}\ \boldsymbol{\lambda}_{k+1}^{D_I\,j}\ \boldsymbol{\lambda}_{k+1}^{P\,j}\ \boldsymbol{\lambda}_{k+1}^{Q\,j}\right]'; \qquad (41)$$

similar expressions hold for the other connection types.

Equation (30) is obtained from (29) with row-column permutations which preserve symmetry:

$$\mathbf{A}_k^{ij} = \mathbf{A}_k^{j\,i\,\prime}\ \forall\ i,j; \qquad (42)$$

the form of vector $\mathbf{a}_k^j$can be recognized by looking at vector $\mathbf{y}^j$; we note that measurement functions $\mathbf{h}^j$ depend only on variables $\mathbf{x}^j$. Examples of a matrix along the diagonal, $\mathbf{A}_k^{jj}$, and of $\mathbf{a}_k^j$ for the case of *branch*es are in (43).

Out of diagonal matrices, $\mathbf{A}_k^{ij}$ $i \neq j$, are constant matrices, for they are zero or they contain derivatives of linear constraints. Matrix $\mathbf{A}^{p(j)j}$ [and then $\mathbf{A}^{j\,p(j)}$] has only four nonzero terms, equal to $+1$ or -1: in the $p(j)$–th part of system (30) they catch $\lambda^{V_R\,j}$, $\lambda^{V_I\,j}$, ΔP_i^j, ΔQ_i^j of child j, while in the j–th part they catch $\lambda^{P\,p(j)}$, $\lambda^{Q\,p(j)}$, $\Delta V_{R,o}^{p(j)}$, $\Delta V_{I,o}^{p(j)}$ of parent $p(j)$. It greatly simplifies the message passing, and it reduces the computational burden involved in (32) (see VIII-C).

B. Covariance

Let covariance matrix $\mathbf{R}$ in (33) be written as:

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}^{11} & \cdots & \mathbf{R}^{1q} \\ \vdots & \vdots & \vdots \\ \mathbf{R}^{q1} & \cdots & \mathbf{R}^{qq} \end{bmatrix} = \begin{bmatrix} \mathbf{W}^{1^{-1}} & & \\ & \ddots & \\ & & \mathbf{W}^{q^{-1}} \end{bmatrix} -$$

$$\begin{bmatrix} \mathbf{h}_{\mathbf{x}^1}^{1\,\prime} & & \\ & \ddots & \\ & & \mathbf{h}_{\mathbf{x}^q}^{q\,\prime} \end{bmatrix} \begin{bmatrix} \mathbf{E}^{11} & \cdots & \mathbf{E}^{1q} \\ \vdots & \vdots & \vdots \\ \mathbf{E}^{q1} & \cdots & \mathbf{E}^{qq} \end{bmatrix} \begin{bmatrix} \mathbf{h}_{\mathbf{x}^1}^{1} & & \\ & \ddots & \\ & & \mathbf{h}_{\mathbf{x}^q}^{q} \end{bmatrix}; \qquad (44)$$

$$\mathbf{A}_{\substack{k\\ j\in B}}^{jj} = \begin{bmatrix} \mathbf{h}_{\mathbf{x}^j}^{j}\mathbf{W}^j\mathbf{h}_{\mathbf{x}^j}^{j}{}' & \mathbf{c}_{\mathbf{x}^j}^{V_R\,j} & \mathbf{c}_{\mathbf{x}^j}^{V_I\,j} & \mathbf{c}_{\mathbf{x}^j}^{V_D\,j} & \mathbf{c}_{\mathbf{x}^j}^{D_I\,j} & \mathbf{c}_{\mathbf{x}^j}^{P\,j} & \mathbf{c}_{\mathbf{x}^j}^{Q\,j} \\ \mathbf{c}_{\mathbf{x}^j}^{V_R\,j}{}' & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{c}_{\mathbf{x}^j}^{V_I\,j}{}' & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{c}_{\mathbf{x}^j}^{D_R\,j}{}' & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{c}_{\mathbf{x}^j}^{D_I\,j}{}' & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{c}_{\mathbf{x}^j}^{P\,j}{}' & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{c}_{\mathbf{x}^j}^{Q\,j}{}' & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}_k \qquad \mathbf{a}_{\substack{h\\ j\in B}}^{j} = -\begin{bmatrix} \mathbf{h}_{\mathbf{x}^j}^{j}\mathbf{W}^j[\mathbf{h}^j-\mathbf{z}^j] \\ \mathbf{c}^{V_Rj} \\ \mathbf{c}^{V_Ij} \\ \mathbf{c}^{D_Rj} \\ \mathbf{c}^{D_Ij} \\ \mathbf{c}^{Pj} \\ \mathbf{c}^{Qj} \end{bmatrix}_k . \tag{43}$$

Eq. (34), here reported, is easily obtained:

$$\mathbf{R}^{jj} = \mathbf{W}^{j^{-1}} - \mathbf{h}_{\mathbf{x}^j}^{j\,'}\ \mathbf{E}^{jj}\ \mathbf{h}_{\mathbf{x}^j}^{j}.$$

Matrices $\mathbf{F}^{jj}$ in (35), whose upper left $n^j \times n^j$ submatrix is equal to $\mathbf{E}^{jj}$, can be computed as follows. Let us write the definition of inverse, $\mathbf{A}\,\mathbf{F} = \mathbf{I}$, as:

$$\begin{bmatrix} \mathbf{A}^{11} & \cdots & \mathbf{A}^{1q} \\ \vdots & \vdots & \vdots \\ \mathbf{A}^{q1} & \cdots & \mathbf{A}^{qq} \end{bmatrix} \begin{bmatrix} \mathbf{F}^1 \\ \vdots \\ \mathbf{F}^q \end{bmatrix} = \begin{bmatrix} \mathbf{U}^1 \\ \vdots \\ \mathbf{U}^q \end{bmatrix}, \tag{45}$$

where, for $j = 1, \ldots, q$,

$$\begin{aligned} \mathbf{F}^j &= \big[\mathbf{F}^{j\,1}\cdots\mathbf{F}^{j\,j}\cdots\mathbf{F}^{j\,q}\big], \\ \mathbf{U}^j &= \big[\mathbf{0}\cdots\cdots\mathbf{I}\cdots\cdots\mathbf{0}\big]. \end{aligned} \tag{46}$$

As in (32), we can write:

$$\begin{aligned} \widetilde{\mathbf{A}}^{j\,j} &= \mathbf{A}^{j\,j} - \mathbf{\Theta}\Big(\sum_{w\in C(j)} \widetilde{\mathbf{A}}^{w\,w^{-1}}\Big)\mathbf{\Theta}', \\ \widetilde{\mathbf{U}}^j &= \mathbf{U}^j - \mathbf{\Theta}\sum_{w\in C(j)} \widetilde{\mathbf{A}}^{w\,w^{-1}}\widetilde{\mathbf{U}}^w, \\ \mathbf{F}^j &= \widetilde{\mathbf{A}}^{j\,j^{-1}}\big(\widetilde{\mathbf{U}}^j - \mathbf{\Theta}'\,\mathbf{F}^{p(j)}\big). \end{aligned} \tag{47}$$

Matrices $\widetilde{\mathbf{U}}^j$ are such that:

$$\begin{aligned} \widetilde{\mathbf{U}}^{jj} &= \mathbf{I}, \\ \widetilde{\mathbf{U}}_{i\neq j}^{j\,i} &= \begin{cases} \neq \mathbf{0}, & \text{if } i \text{ is a descendant of } j, \\ \mathbf{0}, & \text{if } i \text{ is not a descendant of } j; \end{cases} \end{aligned} \tag{48}$$

it follows that

$$\begin{aligned} \mathbf{F}^{jj} &= \widetilde{\mathbf{A}}^{j\,j^{-1}}\big(\mathbf{I} - \mathbf{\Theta}'\,\mathbf{F}^{p(j)\,j}\big), \\ \mathbf{F}^{j\,p(j)} &= \widetilde{\mathbf{A}}^{j\,j^{-1}}\big(\mathbf{0} - \mathbf{\Theta}'\,\mathbf{F}^{p(j)\,p(j)}\big). \end{aligned} \tag{49}$$

Matrix $\mathbf{F}$ is symmetric, for it is the inverse of a symmetric matrix,

$$\mathbf{F}^{p(j)\,j} = \mathbf{F}^{j\,p(j)\,'}, \tag{50}$$

which, combined with (49), yields

$$\mathbf{F}^{j\,j} = \widetilde{\mathbf{A}}^{j\,j^{-1}}\Big(\mathbf{I} + \mathbf{\Theta}'\,\mathbf{F}^{p(j)\,p(j)}\mathbf{\Theta}\,\widetilde{\mathbf{A}}^{j\,j^{-1}}\Big),$$

as in (37).

C. Computational cost

Let us consider (32), here reported for an easy reference,

$$\begin{aligned} \widetilde{\mathbf{A}}_h^{j\,j} &= \mathbf{A}_h^{j\,j} - \mathbf{\Theta}\Big(\sum_{w\in C(j)} \widetilde{\mathbf{A}}_h^{w\,w^{-1}}\Big)\mathbf{\Theta}', \\ \widetilde{\mathbf{a}}_h^{j} &= \mathbf{a}_h^{j} - \mathbf{\Theta}\sum_{w\in C(j)} \widetilde{\mathbf{A}}_h^{w\,w^{-1}}\widetilde{\mathbf{a}}_h^{w}, \\ \mathbf{y}_h^{j} &= \widetilde{\mathbf{A}}_h^{j\,j^{-1}}\big(\widetilde{\mathbf{a}}_h^{j} - \mathbf{\Theta}'\,\mathbf{y}_h^{p(j)}\big). \end{aligned} \tag{32}$$

Matrix $\mathbf{\Theta}$ has only four non-zero, unitary terms (see VIII-A), in four different row/columns. Matrix operations involved in (32.1) reduce to a few additions, and only four terms of $\widetilde{\mathbf{A}}_h^{w\,w^{-1}}\widetilde{\mathbf{a}}_h^{w}$ are actually needed in (32.2); in the usual case of a *branch* feeding a load, these four terms require $4 \times 14 = 56$ multiply-add pairs.

Matrix by vector multiplication in (32.3) has no sparsity characteristic; for the usual *branch*, it requires $14\times 14 = 196$ multiply-add pairs.

Matrix inversion in (32) is the most demanding; exploiting the symmetry of matrix $\widetilde{\mathbf{A}}_h^{j\,j}$ and its static characteristics (zero and unitary terms), the number of multiply-add pairs for usual *branch*es is about 1.07×10^3.

Summing up the computational cost for the above partial operations, the cost incurred in by all computation for a *branch* is about 1.33×10^3.

References

[1] F. C. Schweppe, J. Wildes, and D. B. Rom, "Power System Static State Estimation – Parts i, ii and iii," *IEEE Trans. on Power Apparatus and Systems*, vol. PAS-89, pp. 120–135, 1970.

[2] F. F. Wu, "Power system state estimation: a survey," *Int. J. Electrical Power and Energy Systems*, vol. 12, no. 2, pp. 80–87, April 1990.

[3] A. Monticelli, "Electric Power System State Estimation," *Proceedings of the IEEE*, vol. 88, no. 2, pp. 262–282, February 2000.

[4] T. Van Cutsem, J. L. Howard, and M. Ribbens-Pavella, "A Two-level Static Estimator for Electric Power Systems," *IEEE Trans. on Power Apparatus and Systems*, vol. PAS–100, no. 8, pp. 3722–3732, August 1981.

[5] K. L. Lo, M. M. Salem, R. D. McColl, and A. M. Moffatt, "Two-level state estimation for large power system – Parts i and ii," *IEE Proceedings – C*, vol. 135, no. 4, pp. 299–318, July 1988.

[6] R. Ebrahimian and R. Baldick, "State Estimation Distributed Processing," *IEEE Trans. on Power Systems*, vol. 15, no. 4, pp. 1240–1246, November 2000.

[7] L. Zhao and A. Abur, "Multi Area State Estimation Using Synchronized Phasor Measurements," *IEEE Trans. on Power Systems*, vol. 20, no. 2, pp. 611–617, May 2005.

[8] F. F. Wu and A. F. Neyer, "Asynchronous Distributed State Estimation for Power Distribution Systems," *Proc. of 10th Power Systems Computation Conference*, pp. 439–446, August 1990.

[9] W.-M. Lin, J.-H. Teng, and S.-J. Chen, "A Highly Efficient Algorithm in Treating Current Measurements for the Branch-Current-Based Distribution State Estimation," *IEEE Trans. on Power Systems*, vol. 16, no. 3, pp. 433–439, July 2001.

[10] I. Roytelman and S. M. Shahidehpour, "State Estimation for Electric Power Distribution Systems in Quasi Real-Time Conditions," *IEEE Trans. on Power Delivery*, vol. 8, no. 4, pp. 2009–2015, October 1993.

[11] M. E. Baran and A. W. Kelley, "State Estimation for Real-Time Monitoring of Distribution Systems," *IEEE Trans. on Power Systems*, vol. 9, no. 3, pp. 1601–1609, August 1994.

[12] C. N. Lu, J. H. Teng, and W.-H. E. Liu, "Distribution System State Estimation," *IEEE Trans. on Power Systems*, vol. 10, no. 1, pp. 229–240, February 1995.

[13] M. E. Baran and A. W. Kelley, "A Branch-Current-Based State Estimation Method for Distribution Systems," *IEEE Trans. on Power Systems*, vol. 10, no. 1, pp. 483–491, February 1995.

[14] W.-M. Lin and J. H. Teng, "Distribution System State Estimation," *IEEE Trans. on Power Systems*, vol. 11, no. 1, pp. 518–524, February 1996.

[15] L. Murphy and F. F. Wu, "An Open Design Approach for Distributed Energy Management Systems," *IEEE Trans. on Power Systems*, vol. 8, no. 3, pp. 1172–1179, August 1993.

[16] T. E. Dy-Liacco, "Modern Control Centers and Computer Networking," *IEEE Computer Applications in Power*, pp. 17–22, October 1994.

[17] IEEE Task Force on Open Systems, "Benefits, Problems, and Issues in Open Systems Architectures," *IEEE Trans. on Power Systems*, vol. 9, no. 1, pp. 191–197, February 1994.

[18] A. F. Neyer, F. F. Wu, and K. Imhof, "Object Oriented Programming for Flexible Software: Example of a Load Flow," *IEEE Trans. on Power Systems*, vol. 5, no. 3, pp. 689–696, August 1990.

[19] J. Britton, "An Open, Object-Based Model as the Basis of an Architecture for Distribution Control Centers," *IEEE Trans. on Power Systems*, vol. 7, no. 4, pp. 1500–1508, November 1992.

[20] A. Losi and M. Russo, "An Object Oriented Approach to Load Flow in Distribution Systems," *Proc. of IEEE-PES Summer Meeting*, vol. 4, pp. 2332–2337, July 2000.

[21] A. Losi and M. Russo, "Object Oriented Load Flow for Radial and Weakly Meshed Distribution Systems," *IEEE Trans. on Power Systems*, vol. 18, no. 4, pp. 1265–1274, November 2003.

[22] A. Losi and M. Russo, "Dispersed Generation Modeling for Object–Oriented Distribution Load Flow," *IEEE Trans. on Power Delivery*, vol. 20, no. 2-II, pp. 1532–1540, April 2005.

[23] G. M. Casolino and A. Losi, "Object-oriented distribution state estimation," *Engineering Intelligent Software*, vol. 11, no. 4, pp. 193–200, December 2003.

[24] E. Handschin, F. C. Schweppe, J. Kohlas, and A. Fietcher, "Bad Data Analysis for Power Systems State Estimation," *IEEE Trans. on Power Apparatus and Systems*, vol. PAS-94, pp. 329–337, March/April 1975.

[25] A. Monticelli and A. Garcia, "Reliable Bad Data Processing for Real-Time State Estimation," *IEEE Trans. on Power Apparatus and Systems*, vol. PAS–102, pp. 1126–1139, May 1983.

[26] J. Zhu and D. L. Lubkeman, "Object Oriented Development of Software Systems for Power System Simulations," *IEEE Trans. on Power Systems*, vol. 12, no. 2, pp. 1002–1007, May 1997.

[27] D. G. Luenberger, *Linear and nonlinear programming*, Addison-Wesley Publishing Company, 2nd edition, 1984.

[28] J. E. Dennis Jr. and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, vol. 16 of *Classics in Applied Mathematics*, SIAM, Philadelphia, 1996.

[29] F. F. Wu, W.-H. E. Liu, and S.M. Lun, "Observability analysis and bad data processing for state estimation with equality constraints," *IEEE Trans. on Power Systems*, vol. 3, no. 2, pp. 541–548, May 1988.

[30] N. Jenkins, R. Allan, P. Crossley, D. Kirschen, and G. Strbac, *Embedded Generation*, Number 31 in IEE Power and Energy. The Institute of Electrical Engineers, London, UK, 2000.

[31] I. S. Duff, A. M. Erisman, and J. K. Reid, *Direct Methods for Sparse Matrices*, Oxford Science Publications, 1986.

[32] E. A. Lee et al., *Ptolemy Reference Manual – V. 0.7*, EECS Dept. University of California, Berkeley, CA, 1997.

[33] M. E. Baran and F. F. Wu, "Optimal capacitor placement on radial distribution systems," *IEEE Trans. on Power Delivery*, vol. 4, no. 1, pp. 725–734, January 1989.

Giovanni M. Casolino received his Laurea degree and his Ph.D. in Electrical Engineering both from the Università degli Studi di Cassino, Italy, in 1999, 2004. Currently, he is Assistant Professor of Electrical Power System at the Università degli Studi di Cassino.

Arturo Losi received his Laurea degree and his Ph.D. in Electrical Engineering both from the Università degli Studi di Napoli, Italy, in 1980, 1987. Currently, he is Professor of Electrical Power Systems at the Università degli Studi di Cassino, Italy.