Object Oriented State Estimation for Radial and Weakly Meshed Distribution Systems

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

Abstract— **Object–oriented (OO) state estimation (SE) is presented, for radial and weakly meshed distribution systems. For radial topologies, the SE problem is formulated as usual; for weakly meshed topologies, closed switches are appropriately modeled in a SE problem formulation suitable for a newly proposed two-cycle method. 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. In the case of weakly meshed topologies, it requires some approximations to be introduced, whose influence on the convergence is investigated. 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, convergence of numerical 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.

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 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.

The focus on OOP within a DMS has been the key point of recent 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, extended to treat also weakly meshed topologies. We adopt a system-wide formulation; a decomposition-coordination approach requires that each subsystem is observable on its own, a too demanding requirement for our modeling where every single connection would be a subsystem.

For radial systems, the SE problem is formulated as usual, with equality constraints where appropriate; for weakly meshed systems, a suitable modeling of the closed switches allows to formulate an equivalent SE problem, solved with a two-cycle method.

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. For weakly meshed topologies, the OO implementation requires some approximations to be introduced, whose influence is analyzed and numerically assessed on a small test network.

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:

$$\begin{aligned} \min\ & J(\mathbf{x}) = \frac{1}{2}\ [\mathbf{z}-\mathbf{h}(\mathbf{x})]'\,\mathbf{W}[\mathbf{z}-\mathbf{h}(\mathbf{x})], \\ \text{subject to}\ & \mathbf{c}(\mathbf{x}) = \mathbf{0}. \end{aligned} \tag{2}$$

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 ($r \leq n,\ n \leq m + r$).

A.1 First-order optimality conditions

The Lagrangian function for problem (2) is:

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

where $\boldsymbol{\lambda}$ is the r-vector of Lagrange multipliers associated with the constraints (2.2). The solution satisfies the first–order (F–O) necessary conditions:

$$\begin{aligned} L_{\mathbf{x}}(\mathbf{x}, \boldsymbol{\lambda}) = & J_{\mathbf{x}}(\mathbf{x}) + \mathbf{c}_{\mathbf{x}}(\mathbf{x})\boldsymbol{\lambda} \quad = \mathbf{0}, \\ & \mathbf{c}(\mathbf{x}) \qquad\qquad\quad = \mathbf{0}. \end{aligned} \tag{4}$$

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

B. Closed switches

A closed switch is a zero impedance series branch within the system. It has been recognized that, instead of representing it in the usual way with approximations (i.e., with a small series impedance), it is convenient to include the active and reactive powers flowing along the branch in the set of the describing variables, and the equality of the voltages at the two ends in the equality constraint set [24].

A suitable representation of the closed switch has power injections at the two ends (as in [25] - Fig. 1); with the explicit description of the power injections, it has to be:

$$\begin{aligned} &\mathrm{c}^{D_{P_1}} \equiv P_1 - \alpha = 0, && \mathrm{c}^{D_{P_2}} \equiv P_2 + \alpha = 0, \\ &\mathrm{c}^{D_{Q_1}} \equiv Q_1 - \beta = 0, && \mathrm{c}^{D_{Q_2}} \equiv Q_2 + \beta = 0, \\ &\mathrm{c}^{D_R} \equiv V_{R,1} - V_{R,2} = 0, && \mathrm{c}^{D_I} \equiv V_{I,1} - V_{I,2} = 0. \end{aligned} \tag{5}$$

The number of degrees of freedom with the above modeling does not change with respect to a modeling in which the powers are not within the describing variable set.

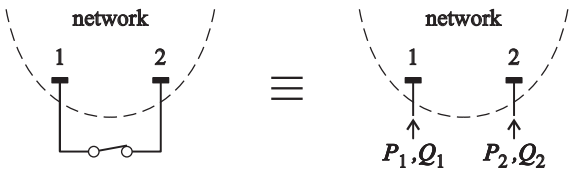

Fig. 1. Modeling of closed switches (see text).

III. A two–cycle method for weakly meshed systems

Problem (2) including constraints (5) can be solved with existing methods for constrained SE (for example, see [3]). In the method we propose, an equivalent SE problem for an equivalent radial system is iteratively solved. For an easy notation, the method is illustrated for the case of only one closed switch, and only the related constraints are explicitly considered; the extension to the general case is straightforward.

It is known that the solution of problem (2)–(5) can be obtained with the augmented Lagrangian method (or multiplier method) [26]. With it, some (even all) constraints are dropped from the constraint set, appropriately weighted and added to the objective function to obtain a new objective function, which is minimized subject to the remaining constraints (if any); iteratively, the weights (Lagrange multipliers) are updated, and a new minimization problem is solved, until convergence is reached. It is a two-cycle method: in the outer cycle the Lagrange multipliers are updated, while in the inner cycle an optimization problem is solved.

We propose to treat with augmented Lagrangian the constraints on voltages (5.3); at the k-th iteration of the outer cycle, the following optimization problem

$$\begin{aligned} \min\ J_k^a(\mathbf{x}) = & J(\mathbf{x}) + \\ & \widetilde{\lambda}_k^{D_R}(V_{R,1} - V_{R,2}) + \widetilde{\lambda}_k^{D_I}(V_{I,1} - V_{I,2}) + \\ & \frac{1}{2}\ \xi\Big((V_{R,1} - V_{R,2})^2 + (V_{I,1} - V_{I,2})^2\Big), \end{aligned} \tag{6}$$

subject to

$$\begin{aligned} &\mathrm{c}^{D_{P_1}} \equiv P_1 - \alpha = 0, && \mathrm{c}^{D_{P_2}} \equiv P_2 + \alpha = 0, \\ &\mathrm{c}^{D_{Q_1}} \equiv Q_1 - \beta = 0, && \mathrm{c}^{D_{Q_2}} \equiv Q_2 + \beta = 0, \end{aligned}$$

is solved to obtain $\mathbf{x}_k$ for the given values of Lagrange multipliers $\widetilde{\lambda}_k^{D_R}$, $\widetilde{\lambda}_k^{D_I}$; ξ is a constant. In problem (6) variables $\mathbf{x}$ include P_1, Q_1, P_2, Q_2, α and β.

For the subsequent development, it is worth noting that at the solution of problem (6) the zero of the derivative of the Lagrangian function with respect to α and β is such that:

$$\begin{aligned} -\lambda^{D_{P_1}} + \lambda^{D_{P_2}} &= 0, \\ -\lambda^{D_{Q_1}} + \lambda^{D_{Q_2}} &= 0, \end{aligned} \tag{7}$$

where the superscript of the Lagrange multipliers is the same as the constraints they are associated with.

A. Inner cycle

In the modified problem (6) the weakly-meshed topology is found in the objective function and in the constraints. The problem can be further modified to obtain an optimization problem whose constraints represent a radial system; in the inner cycle power injections are kept constant at the values $\widetilde{\alpha}_k$ and $\widetilde{\beta}_k$, and they are updated in the outer cycle together with $\widetilde{\lambda}_k^{D_R}$, $\widetilde{\lambda}_k^{D_I}$. Problem (6) becomes

$$\begin{aligned}\min\ J_k^a(\mathbf{x}) =& J(\mathbf{x})+\\ & \widetilde{\lambda}_k^{D_R}(V_{R,1}-V_{R,2})+\widetilde{\lambda}_k^{D_I}(V_{I,1}-V_{I,2})+\\ & \frac{1}{2}\,\xi\Big((V_{R,1}-V_{R,2})^2+(V_{I,1}-V_{I,2})^2\Big),\end{aligned} \tag{8}$$

subject to

$$\begin{aligned}\mathrm{c}^{D_{P_1}} \equiv& P_1-\widetilde{\alpha}_k=0, \qquad \mathrm{c}^{D_{P_2}} \equiv P_2+\widetilde{\alpha}_k=0,\\ \mathrm{c}^{D_{Q_1}} \equiv& Q_1-\widetilde{\beta}_k=0, \qquad \mathrm{c}^{D_{Q_2}} \equiv Q_2+\widetilde{\beta}_k=0,\end{aligned}$$

and it is solved to obtain $\mathbf{x}_k$; in problem (8) variables $\mathbf{x}$ include P_1, Q_1, P_2, Q_2 and do not include α and β. Note that the constraints of problem (8) describe a radial system with power injections at some terminals, treated as exact measurements.

B. Outer cycle

With reference to one outer cycle iteration, let

$$\begin{aligned}\Delta\widetilde{\lambda}_k^{D_R}=\widetilde{\lambda}_{k+1}^{D_R}-\widetilde{\lambda}_k^{D_R}, \quad \Delta\widetilde{\lambda}_k^{D_I}=\widetilde{\lambda}_{k+1}^{D_I}-\widetilde{\lambda}_k^{D_I},\\ \Delta\widetilde{\alpha}_k=\widetilde{\alpha}_{k+1}-\widetilde{\alpha}_k, \qquad \Delta\widetilde{\beta}_k=\widetilde{\beta}_{k+1}-\widetilde{\beta}_k,\end{aligned} \tag{9}$$

denote the variations of the values of the four quantities related to the closed switch, and

$$\boldsymbol{\Delta x}_k=\mathbf{x}_{k+1}-\mathbf{x}_k, \qquad \boldsymbol{\Delta\lambda}_k=\boldsymbol{\lambda}_{k+1}-\boldsymbol{\lambda}_k, \tag{10}$$

denote the consequent variation of the solution of the inner cycle optimization problem.

If variations (9)-(10) are sufficiently small, the inner cycle F–O conditions at $k+1$ can be expressed by the first–order Taylor expansion of conditions at k; it results [functional dependence is omitted – the symbol $f\big|_k$ stands for $f(\mathbf{x}_k)$]:

$$\begin{aligned}L^a_{\mathbf{xx}}\big|_k\boldsymbol{\Delta x}_k+\mathbf{c}_{\mathbf{x}}\big|_k\boldsymbol{\Delta\lambda}_k+\mathrm{c}^{D_R}_{\mathbf{x}}\Delta\widetilde{\lambda}_k^{D_R}+\mathrm{c}^{D_I}_{\mathbf{x}}\Delta\widetilde{\lambda}_k^{D_I}=0,\\ \mathbf{c}'_{\mathbf{x}}\big|_k\boldsymbol{\Delta x}_k+\Big[-\Delta\widetilde{\alpha}_k\ +\Delta\widetilde{\alpha}_k\ -\Delta\widetilde{\beta}_k\ +\Delta\widetilde{\beta}_k\Big]'=0,\end{aligned} \tag{11}$$

where symbol $L^a_{\mathbf{xx}}$ represents the Hessian of the Lagrangian function of problem (8).

We would like that, after variations (9) are imposed, the new solution of problem (8) of the inner cycle is such that the voltages at the two sides of closed switch are equal, as well as the multipliers associated with the equalities on the powers injected at the two sides [see (5.3) and (7)]:

$$\begin{aligned}(V_{R,1\,k}+\Delta V_{R,1\,k})-(V_{R,2\,k}+\Delta V_{R,2\,k})=0\\ (V_{I,1\,k}+\Delta V_{I,1\,k})-(V_{I,2\,k}+\Delta V_{I,2\,k})=0\\ -(\lambda_k^{D_{P_1}}+\Delta\lambda_k^{D_{P_1}})+(\lambda_k^{D_{P_2}}+\Delta\lambda_k^{D_{P_2}})=0\\ -(\lambda_k^{D_{Q_1}}+\Delta\lambda_k^{D_{Q_1}})+(\lambda_k^{D_{Q_2}}+\Delta\lambda_k^{D_{Q_2}})=0.\end{aligned} \tag{12}$$

There are four conditions in (12), as many as the parameters related to the closed switch in problem (8), $\widetilde{\alpha}_k, \widetilde{\beta}_k, \widetilde{\lambda}_k^{D_R}$, and $\widetilde{\lambda}_k^{D_I}$.

Conditions (12) together with (11) form a squared linear system: variations (9) can be computed to have the new optimal solution of the inner cycle optimization problem to comply with conditions on the overall solution (at least, in an approximated way).

The convergence of the updating method represented by (11)–(12) has to be assessed; in Appendix A, numerical results evidence that convergence is easily obtained.

To summarize, in a weakly meshed system a closed switch is considered open to obtain a radial system; its closed status is taken into account partly by adding a term to the original objective function, partly by imposing power injections at the two ends of the (now) open switches. The term added to the objective function is zero if (and only if) the voltages at the two ends of the switch are equal; the term is driven to zero. The power injections at the ends sum up to zero, as it is required by closed switches. To some extent, but with many differences, the proposed method of breaking the loops at the closed switches and injecting powers of opposite sign at the break points resembles the equivalent power injection method adopted in backward/forward sweep methods in power flow algorithms [25].

IV. 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 [27]; 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 ingoing or outgoing depending on the conventional direction of powers at the port, inwards or outwards; the port variables are accordingly 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*, concrete classes are derived [21]:

- *branch*, a connection with one incoming port and one outgoing port (such as a line or a transformer);
- *root*, a connection with only one outgoing port (the supplying system);
- *fork*, a connection with one incoming port and many outgoing ports (a zero impedance busbar);

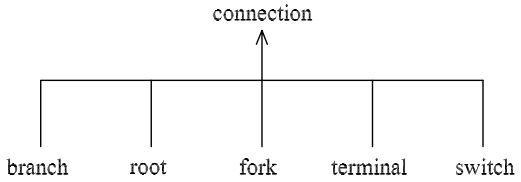

Fig. 2. Class hierarchy for distribution system modeling.

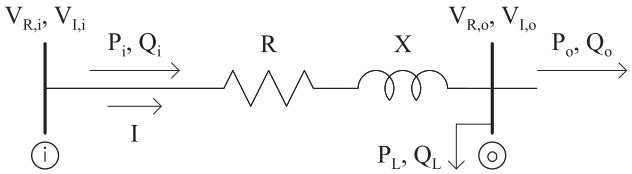

Fig. 3. Circuit representation of class *branch*.

– *terminal*, a connection with only one incoming port (a termination of the system);
– *switch*, a connection with two incoming ports (a looping switch), and two states: open or closed; in the open state, it represents an infinite impedance connection between the two ports, while the impedance is zero in the closed status.

The class hierarchy is shown in Fig. 2 [21]. The model of a distribution system is an oriented graph of object instances, appropriately connected: one outgoing port is connected to only one ingoing port and viceversa, and no port is unconnected. Only root can be the origin of the graph, and only switches or terminals can be the ends of the graph.

For the subsequent development, it is useful to introduce some notation related to the graph. Let $p(j)$ represent the parent(s) of the j–th connection, the one(s) that precedes it in the oriented graph. $p(j)$ is made of only one element, but for switches whose $p(j)$ contains two elements, referred to as $p_1(j)$ and $p_2(j)$; $p(j)$ is empty for the root. Let $C(j)$ represent the children of the j–th connection, the set of connections that follow it in the graph; $C(j)$ is empty for terminals and switches.

In the following, the equations that characterize each class are shown, for the representation of 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:

$$J^j(\mathbf{x}^j) = \frac{1}{2}\left[\mathbf{z}^j - \mathbf{h}^j(\mathbf{x}^j)\right]' \mathbf{W}^j \left[\mathbf{z}^j - \mathbf{h}^j(\mathbf{x}^j)\right], \tag{13}$$

where $\mathbf{x}^j$ represent 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 actual 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 branch

Branches model physical lines and transformers; they include also loads and shunts devices at the ending busbar. Class *branch* has one ingoing port and one outgoing port; Fig. 3 depicts its circuit representation.

The measurements at a *branch* are (the uppercase superscript denotes the measurement type):

– voltage amplitude at the output port

$$h^{V_o\,j} = \sqrt{(V^j_{R,o})^2 + (V^j_{I,o})^2}; \tag{14}$$

– ingoing current amplitude

$$h^{I_i\,j} = \sqrt{\frac{(P^j_i)^2 + (Q^j_i)^2}{(V^j_{R,i})^2 + (V^j_{I,i})^2}}; \tag{15}$$

– ingoing active and reactive powers

$$\begin{aligned} h^{P_i\,j} &= P^j_i, \\ h^{Q_i\,j} &= Q^j_i; \end{aligned} \tag{16}$$

– active and reactive power injections

$$\begin{aligned} h^{P_L\,j} &= P^j_i - P^j_o - \mathrm{R}^j \frac{(P^j_i)^2 + (Q^j_i)^2}{(V^j_{R,i})^2 + (V^j_{I,i})^2}, \\ h^{Q_L\,j} &= Q^j_i - Q^j_o - \mathrm{X}^j \frac{(P^j_i)^2 + (Q^j_i)^2}{(V^j_{R,i})^2 + (V^j_{I,i})^2}. \end{aligned} \tag{17}$$

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

$$\begin{aligned} \mathbf{c}^{V_R\,j}(\mathbf{x}) &\equiv V^{p(j)}_{R,o} - V^j_{R,i} = 0, \\ \mathbf{c}^{V_I\,j}(\mathbf{x}) &\equiv V^{p(j)}_{I,o} - V^j_{I,i} = 0; \end{aligned} \tag{18}$$

– the voltage drop (real and imaginary parts)

$$\begin{aligned} &\mathbf{c}^{D_R\,j}(\mathbf{x}^j) \equiv V^j_{R,i} - V^j_{R,o} - \\ &\quad \frac{V^j_{R,i}(\mathrm{R}^j P^j_i + \mathrm{X}^j Q^j_i) - V^k_{I,i}(\mathrm{X}^j P^j_i - \mathrm{R}^j Q^j_i)}{(V^j_{R,i})^2 + (V^j_{I,i})^2} = 0, \\ &\mathbf{c}^{D_I\,j}(\mathbf{x}^j) \equiv V^j_{I,i} - V^j_{I,o} - \\ &\quad \frac{V^j_{R,i}(\mathrm{X}^j P^j_i - \mathrm{R}^j Q^j_i) + V^j_{I,i}(\mathrm{R}^j P^j_i + \mathrm{X}^j Q^j_i)}{(V^j_{R,i})^2 + (V^j_{I,i})^2} = 0; \end{aligned} \tag{19}$$

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

$$\begin{aligned} \mathbf{c}^{P\,j}(\mathbf{x}) &\equiv P^j_o - \sum_{w \in C(j)} P^w_i = 0, \\ \mathbf{c}^{Q\,j}(\mathbf{x}) &\equiv Q^j_o - \sum_{w \in C(j)} Q^w_i = 0; \end{aligned} \tag{20}$$

– in the case there are zero injection exact measurements, the following additional constraints are added and the weights associated with measurements (17) are set to zero

$$\begin{aligned}\mathbf{c}^{P_L\,j} &= P_i^j - P_o^j - \mathrm{R}^j \frac{(P_i^j)^2 + (Q_i^j)^2}{(V_{R,i}^j)^2 + (V_{I,i}^j)^2} = 0,\\ \mathbf{c}^{Q_L\,j} &= Q_i^j - Q_o^j - \mathrm{X}^j \frac{(P_i^j)^2 + (Q_i^j)^2}{(V_{R,i}^j)^2 + (V_{I,i}^j)^2} = 0.\end{aligned} \tag{21}$$

C. 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 is only the measurement of the voltage amplitude at the outgoing port, as (14).

The constraints of *root* express:

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

$$\mathbf{c}^{D_I\,root}(\mathbf{x}^{root}) \equiv V_{I,o}^{root} = 0; \tag{22}$$

– the active and reactive balances between the outgoing powers and the ingoing powers of the subsequent connection, as (20).

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 is only the measurement of the voltage amplitude at the outgoing ports, as (14).

The constraints assigned to *fork* are:

– the voltage equality with the parent (18)
– the voltage equality inside the *fork*

$$\begin{aligned}\mathbf{c}^{D_R\,j}(\mathbf{x}^j) &\equiv V_{R,i}^j - V_{R,o}^j = 0,\\ \mathbf{c}^{D_I\,j}(\mathbf{x}^j) &\equiv V_{I,i}^j - V_{I,o}^j = 0;\end{aligned} \tag{23}$$

– the active and reactive power balances inside the fork

$$\begin{aligned}\mathbf{c}^{D_P\,j}(\mathbf{x}^j) &\equiv P_i^j - P_o^j,\\ \mathbf{c}^{D_Q\,j}(\mathbf{x}^j) &\equiv Q_i^j - Q_o^j;\end{aligned} \tag{24}$$

– the active and reactive balances between the outgoing powers and the ingoing powers of the subsequent connection, as (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 (18)
– the null value of the ingoing active and reactive powers:

$$\begin{aligned}\mathbf{c}^{D_P\,j}(\mathbf{x}^j) &\equiv P_i^j = 0,\\ \mathbf{c}^{D_Q\,j}(\mathbf{x}^j) &\equiv Q_i^j = 0.\end{aligned} \tag{25}$$

F. Class switch

The *switch* class describes a looping switch; it has two incoming ports and two states: closed or open.

In the closed status, we assume there are the following measurements:

– ingoing current amplitude at one of the two ports

$$h^{I_{i,w}\,j} = \sqrt{\frac{(P_{i_w}^j)^2 + (Q_{i_w}^j)^2}{(V_{R,i_w}^j)^2 + (V_{I,i_w}^j)^2}}; \quad w = 1 \text{ or } 2 \tag{26}$$

– ingoing active and reactive powers at one of the two ports

$$\left.\begin{aligned}h^{P_{i,w}\,j} &= P_{i_w}^j\\ h^{Q_{i,w}\,j} &= Q_{i_w}^j\end{aligned}\right\} \quad w = 1 \text{ or } 2. \tag{27}$$

The constraints of the *switch* in the closed status express:

– the voltage equality with the parent (18), for each port
– the equality between the voltage (real and imaginary parts) at the two ingoing ports

$$\begin{aligned}\mathbf{c}^{D_R\,j}(\mathbf{x}^j) &\equiv V_{R,i_1}^j - V_{R,i_2}^j = 0,\\ \mathbf{c}^{D_I\,j}(\mathbf{x}^j) &\equiv V_{I,i_1}^j - V_{I,i_2}^j = 0;\end{aligned} \tag{28}$$

– the active and reactive power injections at the two ingoing ports [see (5)]

$$\begin{aligned}\mathbf{c}^{D_{P_1}\,j}(\mathbf{x}) &\equiv P_{i_1}^j - \alpha^j = 0, \quad \mathbf{c}^{D_{P_2}\,j}(\mathbf{x}) \equiv P_{i_2}^j + \alpha^j = 0,\\ \mathbf{c}^{D_{Q_1}\,j}(\mathbf{x}) &\equiv Q_{i_1}^j - \beta^j = 0, \quad \mathbf{c}^{D_{Q_2}\,j}(\mathbf{x}) \equiv Q_{i_2}^j + \beta^j = 0,\end{aligned} \tag{29}$$

where α and β are two additional variables that characterize the *switch*.

In the open state, there is no measurement at the *switch*. In this state, the constraints of the *switch* express:

– the voltage equality with the parent (18), for each port
– the active and reactive power injections at the two ingoing ports (29)
– the null value of the active and reactive power injections

$$\alpha^j = 0, \qquad \beta^j = 0. \tag{30}$$

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; each *branch* and each *fork* contributes eight variables (four for each port), and *switch*es contribute ten variables each (four for each port and active and reactive power injections). For a system composed of the *root*, b *branch*es, f *fork*s, s *switch*es and t *terminal*s, the SE problem has

$$n = \dim\{\mathbf{x}\} = 4 + 8b + 8f + 10s + 4t. \tag{31}$$

G.2 Objective function

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

$$q = 1 + b + f + t + s; \tag{32}$$

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

$$J(\mathbf{x}) = \sum_{j=1}^{q} J^j(\mathbf{x}^j). \tag{33}$$

G.3 Constraints and measurements

The total number of equality constraints, r, contributed by all *connection*s in a network, apart from the ones necessary to treat exact measurements, is:

$$r = 3 + 6b + 8f + 10s + 4t. \tag{34}$$

The difference between the dimensions of the describing variables set, n, and of the equality constraints set, r, is:

$$n_{min} = n - r = 2b + 1; \tag{35}$$

it is the dimension of a state variable set, a minimal set of describing variables. The $2b + 1$ state variables can be, for example, the substation voltage amplitude, with the imaginary part equal to zero, and the ingoing active and reactive powers in all 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}.

V. 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.

For radial topologies, the Gauss-Newton method usually adopted for the solution of the SE problem is implemented. For weakly meshed systems, the two-cycle method of Sect. III is implemented, again relying on the Gauss-Newton method to solve the inner cycle optimization problem. The partial derivatives of the equations that characterize each class are required.

A. OO implementation of the optimization problem solution (inner cycle)

The values of $\mathbf{x}$ and $\boldsymbol{\lambda}$ that satisfy the F–O necessary conditions of the SE optimization problem can be obtained by the Gauss-Newton method; it is the Newton's method [26] where the second order derivatives of measurement functions and of equality constraints (appearing in the Hessian of the Lagrangian function) are neglected [3, 28].

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

$$\begin{bmatrix} \mathbf{H} & \mathbf{c_x} \\ \mathbf{c}'_{\mathbf{x}} & \mathbf{0} \end{bmatrix}_h \begin{bmatrix} \boldsymbol{\Delta}\mathbf{x}_h \\ \boldsymbol{\lambda}_{h+1} \end{bmatrix} = - \begin{bmatrix} J^a_{\mathbf{x}} \\ \mathbf{c} \end{bmatrix}_h , \tag{36}$$

and the variables $\mathbf{x}$ are accordingly updated; $\mathbf{H}$ represents the approximated Hessian. Equation (36) is referred to problem (8) for weakly meshed systems; for radial systems, it is $J^a \equiv J$, and the Hessian is consequently obtained.

Equation (36) can be written as

$$\begin{bmatrix} \mathbf{A}^{11} & \mathbf{A}^{12} & \cdots & \mathbf{A}^{1q} \\ \mathbf{A}^{21} & \mathbf{A}^{22} & \cdots & \mathbf{A}^{2q} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{A}^{j1} & \mathbf{A}^{j2} & \cdots & \mathbf{A}^{jq} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{A}^{q1} & \mathbf{A}^{q2} & \cdots & \mathbf{A}^{qq} \end{bmatrix}_h \begin{bmatrix} \mathbf{y}^1 \\ \mathbf{y}^2 \\ \vdots \\ \mathbf{y}^j \\ \vdots \\ \mathbf{y}^q \end{bmatrix}_h = \begin{bmatrix} \mathbf{a}^1 \\ \mathbf{a}^2 \\ \vdots \\ \mathbf{a}^j \\ \vdots \\ \mathbf{a}^q \end{bmatrix}_h , \tag{37}$$

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

It can be shown that, since

$$\mathbf{A}^{ij} = \mathbf{0} \iff i \neq j \text{ and } i \notin p(j) \text{ and } i \notin C(j), \tag{38}$$

the solution of the linear system (37) can be obtained with the following recursive equations, for $j = 1, \ldots, q$:

$$\begin{aligned} \tilde{\mathbf{A}}^{j\,j}_h &= \mathbf{A}^{j\,j}_h - \sum_{w \in C(j)} \mathbf{A}^{j\,w} \tilde{\mathbf{A}}^{w\,w^{-1}}_h \mathbf{A}^{w\,j}, \\ \tilde{\mathbf{a}}^{j}_h &= \mathbf{a}^{j}_h - \sum_{w \in C(j)} \mathbf{A}^{j\,w} \tilde{\mathbf{A}}^{w\,w^{-1}}_h \tilde{\mathbf{a}}^{w}_h, \\ \mathbf{y}^j_h &= \begin{cases} \tilde{\mathbf{A}}^{j\,j^{-1}}_h \big(\tilde{\mathbf{a}}^j_h - \mathbf{A}^{j\,p(j)} \mathbf{y}^{p(j)}_h\big), \forall j \notin S_c \\ \mathbf{A}^{j\,j^{-1}}_h \big(\mathbf{a}^j_h - \sum\limits_{w=1,2} \mathbf{A}^{j\,p_w} \mathbf{y}^{p_w}_h\big), \forall j \in S_c \end{cases} , \end{aligned} \tag{39}$$

where S_c represents the set of closed switches.

In weakly meshed systems, closed switches couple variables $\mathbf{y}^j_h$ of the sections of the network constituted by all connections of the loop created by the closed switches. These dependencies are hard treating in a OODSE algorithm, due to the characteristic we seek of exchanging information only between neighboring objects. To avoid these dependencies, the terms related to ξ lying outside the principal diagonal of $\mathbf{A}^{j\,j}$, $j \in S_c$, are neglected; the convergence of the resulting approximated algorithm is discussed in Appendix C.

B. OO implementation of the outer cycle update (weakly meshed systems)

For the general case of many closed switches, neglecting the second order derivatives the linear system (11)–(12) can be put as:

$$\begin{bmatrix} \mathbf{H} & \mathbf{c_x} & \mathbf{c}_{\mathbf{x}}^{D_R} & \mathbf{c}_{\mathbf{x}}^{D_I} & \mathbf{0} & \mathbf{0} \\ \mathbf{c}'_{\mathbf{x}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{c}_{\alpha}^{D_P\prime} & \mathbf{c}_{\beta}^{D_Q\prime} \\ \mathbf{c}_{\mathbf{x}}^{D_R\prime} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{c}_{\mathbf{x}}^{D_I\prime} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{c}_{\alpha}^{D_P} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{c}_{\beta}^{D_Q} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}_k \begin{bmatrix} \boldsymbol{\Delta}\mathbf{x} \\ \boldsymbol{\Delta}\boldsymbol{\lambda} \\ \boldsymbol{\Delta}\tilde{\boldsymbol{\lambda}}^{D_R} \\ \boldsymbol{\Delta}\tilde{\boldsymbol{\lambda}}^{D_I} \\ \boldsymbol{\Delta}\tilde{\boldsymbol{\alpha}} \\ \boldsymbol{\Delta}\tilde{\boldsymbol{\beta}} \end{bmatrix}_k = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{V}_{R,i_2} - \mathbf{V}_{R,i_1} \\ \mathbf{V}_{I,i_2} - \mathbf{V}_{I,i_1} \\ \boldsymbol{\lambda}^{D_{P_1}} - \boldsymbol{\lambda}^{D_{P_2}} \\ \boldsymbol{\lambda}^{D_{Q_1}} - \boldsymbol{\lambda}^{D_{Q_2}} \end{bmatrix}_k . \tag{40}$$

for S_c (first block) · for S_c (second block) · for S_c (third block)

Linear system (40) can be written as

$$\begin{bmatrix} \mathbf{B}^{11} & \cdots & \mathbf{B}^{1q} \\ \vdots & \vdots & \vdots \\ \mathbf{B}^{j1} & \cdots & \mathbf{B}^{jq} \\ \vdots & \vdots & \vdots \\ \mathbf{B}^{q1} & \cdots & \mathbf{B}^{qq} \end{bmatrix}_k \begin{bmatrix} \mathbf{u}^1 \\ \vdots \\ \mathbf{u}^j \\ \vdots \\ \mathbf{u}^q \end{bmatrix}_k = \begin{bmatrix} \mathbf{b}^1 \\ \vdots \\ \mathbf{b}^j \\ \vdots \\ \mathbf{b}^q \end{bmatrix}_k , \tag{41}$$

where vector $\mathbf{u}_k^j$ for the j–th connection represents variation of port variables and of Lagrange multipliers; for closed switches, it also represents $\mathbf{\Delta}\widetilde{\boldsymbol{\lambda}}_k^{D_R}$, $\mathbf{\Delta}\widetilde{\boldsymbol{\lambda}}_k^{D_I}$, $\mathbf{\Delta}\widetilde{\boldsymbol{\alpha}}_k$, and $\mathbf{\Delta}\widetilde{\boldsymbol{\beta}}_k$ (see Appendix B).

It is

$$\mathbf{B}^{ij} = \mathbf{0} \iff i \neq j \text{ and } i \notin p(j) \text{ and } i \notin C(j), \tag{42}$$

and

$$\mathbf{b}^j = \mathbf{0}, \quad \forall j \notin S_c. \tag{43}$$

Due to (42)-(43), for all connections but closed switches it can be written , $\forall j \notin S_c$:

$$\begin{aligned} \mathbf{u}_k^j &= -\widetilde{\mathbf{B}}_k^{j\,j\,-1} \sum_{i \in C(j)} \mathbf{B}^{j\,i} \mathbf{u}_k^i + \widetilde{\mathbf{B}}_k^{j\,j\,-1} \sum_{\substack{i \,\in\, C(p_j) \\ i \neq j}} \widetilde{\mathbf{B}}_k^{j\,i} \mathbf{u}_k^i, \\ \widetilde{\mathbf{B}}_k^{j\,j} &= \mathbf{B}_k^{j\,j} - \mathbf{B}^{j\,p_j} \widetilde{\mathbf{B}}_k^{p_j\,p_j\,-1} \mathbf{B}^{p_j\,j}, \\ \widetilde{\mathbf{B}}_k^{j\,i} &= \mathbf{B}^{j\,p_j} \widetilde{\mathbf{B}}_k^{p_j\,p_j\,-1} \mathbf{B}^{p_j\,i}, \end{aligned} \tag{44}$$

where p_j stands for $p(j)$; for closed switches, it can be written, $j \in S_c$:

$$\begin{aligned} \mathbf{u}_k^j &= \widetilde{\mathbf{B}}_k^{j\,j\,-1} \mathbf{b}_k^j, \\ \widetilde{\mathbf{B}}_k^{j\,j} &= \mathbf{B}_k^{j\,j} - \sum_{w=1,2} \mathbf{B}^{j\,p_w} \widetilde{\mathbf{B}}_k^{p_w\,p_w\,-1} \mathbf{B}^{p_w\,j}, \end{aligned} \tag{45}$$

where p_w, $w = 1, 2$, represent the two parents of the switch. We note that (44) are necessary only to derive the expression of the matrices appearing in (45); indeed, we are interested in actually computing only four components of $\mathbf{u}^j$ for every closed switch.

The sums in the rhs of (44.1) determine a cross dependence of matrices $\widetilde{\mathbf{B}}_k^{j\,j}$ of all the descendants of forks. As before, these dependencies are hard treating in the OODSE algorithm. To avoid them, approximation is introduced neglecting the second term in the rhs of (44.1); it means assuming that each branch stemming from a fork sees itself as the only child of it. In Appendix C, convergence conditions for the approximated resulting algorithm are illustrated, with examples on a small network.

With this understanding, equations (44)–(45) can be implemented in the OODSE algorithm.

C. OODSE algorithm

For the (inner cycle) optimization problem, let ε_i represent the mismatch on the F–O optimality conditions, and ψ_i a "convergence achieved" flag, respectively; moreover, let ε_o represent the mismatch on the outer cycle update for weakly meshed systems (see Appendix D). The OODSE algorithm can be implemented, based on (39), (44), (45), and on downstream and upstream graph tree traversing:

i. all variables for all connections are set to their initial value;
ii. the root initiates a downstream graph traversing, in which any connection sends data to its children;
iii. after receiving data from its parent, the j–th connection:
   1. computes matrix $\widetilde{\mathbf{B}}^{j\,j\,-1}$ with (44.2),
   2. updates its copy of the receiving end voltage (real and imaginary parts) of the parent and stores $\mathbf{y}^{p(j)}$ for the subsequent upstream graph traversing,
   3. updates its variables $\mathbf{y}^j$ with (39.3),
   4. sends downstream $\widetilde{\mathbf{B}}^{j\,j\,-1}$, $\mathbf{y}^j$, together with ψ_i;
iv. at the end of the downstream graph traversing, all ending connections (terminal and switches) initiate an upstream graph traversing, in which every connection sends data to its parent, and every closed switch computes and sends upstream its part of ε_o; moreover, if ψ_i is high, closed switches compute the new values $\widetilde{\boldsymbol{\lambda}}_{k+1}^{D_R}$, $\widetilde{\boldsymbol{\lambda}}_{k+1}^{D_I}$, $\widetilde{\boldsymbol{\alpha}}_{k+1}$, and $\widetilde{\boldsymbol{\beta}}_{k+1}$;
v. after receiving data from all its children, the j–th connection (with the exception of the root):
   1. updates ε_i – every fork updates also ε_o,
   2. evaluates the matrix $\tilde{\mathbf{A}}^{j\,j\,-1}$ and the vector $\tilde{\mathbf{A}}^{j\,j\,-1}\tilde{\mathbf{a}}^j$ [see (39.1) and (39.2)],
   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 ε_o;
vi. at the end of the upstream graph traversing, the root:
   1. updates ε_i, and sets ψ_i high if ε_i is within the prescribed accuracy,
   2. if ψ_i is high and ε_o too is within the required accuracy, stops the computations,
   3. otherwise, updates its variable $\mathbf{y}^{root}$ with (39.3), and sends them downstream together with ψ_i, initiating a new graph traversing.

Steps iii. and v. define two different methods for each connection, to be applied during the downstream and the upstream graph traversing respectively.

Matrices $\mathbf{B}^{p(j)\,j}$ and $\mathbf{B}^{j\,p(j)}$, as well as $\mathbf{A}^{p(j)\,j}$ and $\mathbf{A}^{j\,p(j)}$, have many zero rows and columns. It can be exploited to simplify the message passing and to reduce the computational requirements (see Appendix B).

D. 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, the computational cost per iteration grows linearly with the number of *connection*s because of the local processing, while the growth is cubic in classical methods.

The exploitation of the sparsity of the matrices in clas-

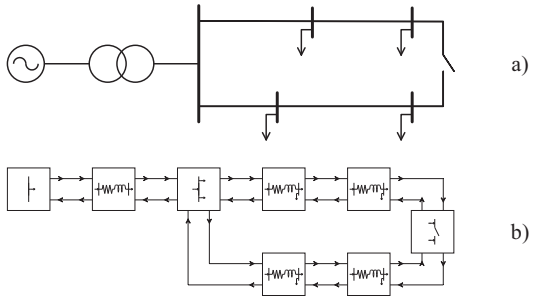

Fig. 4. A simple distribution system (a) and its graphical OO representation (b).

sical methods is system specific and not flexible for configuration changes, while in the OO algorithm it is easily obtained, and configuration changes are easily treated.

VI. Numerical applications

The OODSE application has been developed in the Ptolemy environment [29], 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. IV. The representation of a distribution system is obtained by connecting the input and the output portholes of the blocks (Fig. 4).

The 69–branch distribution system whose data are reported in [30] has been studied; four switches have been added, as shown in the graphical representation of Fig. 5. 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 measurement to the number of state variables) is 2.04. The measurement standard deviation is 2% for all measurements, and the starting point was the load flow solution for the measured root voltage and load powers.

First, we considered the radial configuration; Fig. 6 reports the graph of the (inner cycle) F–O mismatch, ε_i, versus the iteration count, h. The convergence is quite speedy; indeed, for radial topologies the OODSE does not introduce any approximation to the usual Gauss-Newton method.

The results for two weakly meshed configurations are reported in Fig. 7 and Fig. 8, for the case of only one switch closed (♯1) and for the case of all switches closed, respectively. The graphs of the outer cycle and the inner cycle mismatches, ε_o and ε_i, versus the overall iteration, count, h_t, and the outer cycle iteration count, k, are reported. It can be seen that the more the closed switches, the slower the convergence; it depends on the number of forks with descendant(s) ending on closed switches (see Appendix C).

Further cases (not reported for the sake of space) have been run with starting points different from the load flow solution; convergence has always been reached.

As for computation times, each iteration has required about 0.040 s; the application has been developed with a 32-bit library and the tests have been done in double

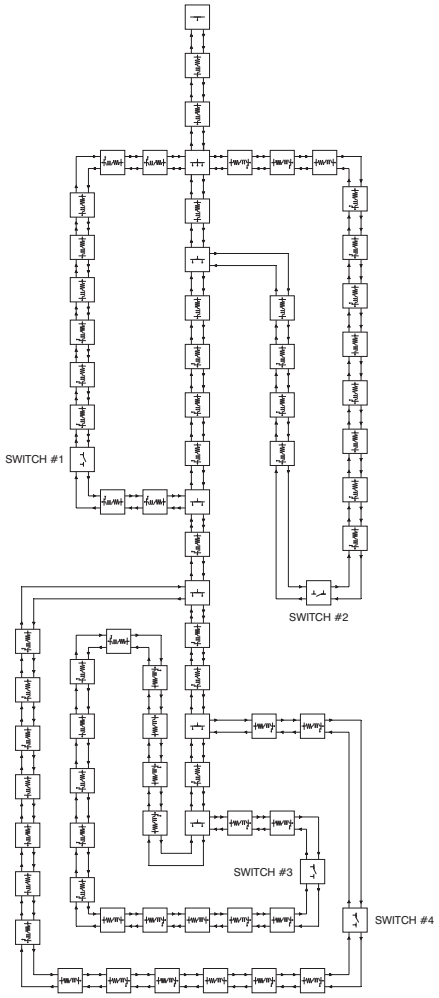

Fig. 5. Graphical OO representation of the 69-branch system.

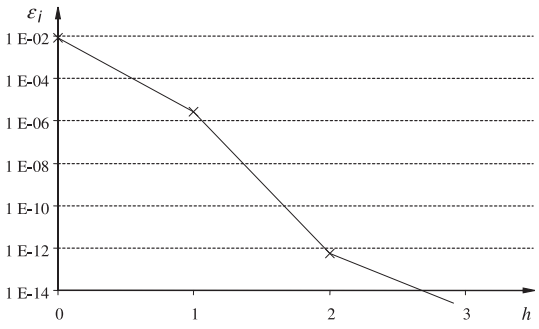

Fig. 6. Convergence of the OODSE – radial configuration.

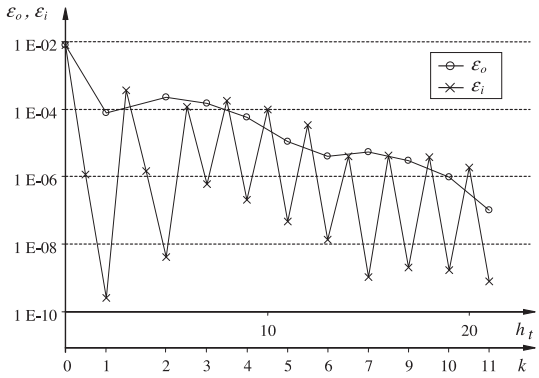

Fig. 7. Convergence of the OODSE – switch ♯1 closed.

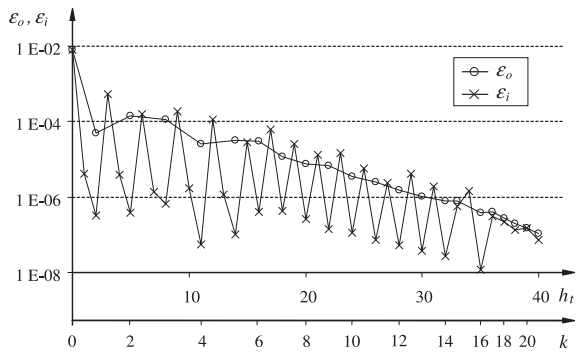

Fig. 8. Convergence of the OODSE – all switches closed.

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 and weakly meshed topologies, with a suitable modeling of closed switches. 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, which requires some approximation to be introduced to treat weakly meshed systems.

Radial systems require few iterations to converge; weakly meshed systems require more iterations, whose number grows as the number of closed switches grows.

Future work will concern other issues of the state estimation, such as bad the data analysis and the influence of measurement redundancy, and the extension to unbalanced systems.

VIII. Appendix

A. Convergence of the outer cycle update

To have an insight on the convergence of the updating method represented by (11)–(12) and on the influence of constant ξ, we have carried out numerical experiments on the simple system of Fig. 4 by the widely-used computing environment Matlab [31]. The starting point was the solution of the load flow corresponding to the measured values of root voltage and load powers; there were five measurements at each branch (one voltage and four powers) and the voltage measurement at the root.

The second order derivatives of measurement functions and of equality constraints appearing in $L^a_{\mathbf{xx}}$ in (11) have been neglected. Three examples of convergence behavior are reported in Fig. 10 (dotted lines, full markers), for three values of constant ξ ($\xi_a = 2\,\xi_b = 20\,\xi_c$). It can be observed that, in practice, the convergence is linear, and the value of ξ can be chosen to have a good (small) convergence ratio [26].

B. Vectors and matrices

In (41), for branches (B) and closed switches (S_c), it is:

$$\underset{j\in B}{\mathbf{u}^j_k} = \left[\boldsymbol{\Delta}\mathbf{x}^{j\,\prime}\ \boldsymbol{\Delta}\boldsymbol{\lambda}^{V_R\,j}\ \boldsymbol{\Delta}\boldsymbol{\lambda}^{V_I\,j}\ \boldsymbol{\Delta}\boldsymbol{\lambda}^{D_R\,j}\ \boldsymbol{\Delta}\boldsymbol{\lambda}^{D_I\,j}\ \boldsymbol{\Delta}\boldsymbol{\lambda}^{P\,j}\ \boldsymbol{\Delta}\boldsymbol{\lambda}^{Q\,j}\right]'_k,$$

$$\underset{j\in S_c}{\mathbf{u}^j_k} = \Big[\boldsymbol{\Delta}\mathbf{x}^{j\,\prime}\ \boldsymbol{\Delta}\boldsymbol{\lambda}^{V_{R_w}\,j}\ \boldsymbol{\Delta}\boldsymbol{\lambda}^{V_{I_w}\,j}\ \boldsymbol{\Delta}\boldsymbol{\lambda}^{P_w\,j}\ \boldsymbol{\Delta}\boldsymbol{\lambda}^{Q_w\,j}\ \boldsymbol{\Delta}\tilde{\boldsymbol{\lambda}}^{D_R\,j}\ \boldsymbol{\Delta}\tilde{\boldsymbol{\lambda}}^{D_I\,j}\ \boldsymbol{\Delta}\tilde{\boldsymbol{\alpha}}^{j}\ \boldsymbol{\Delta}\tilde{\boldsymbol{\beta}}^{j}\ \Big]'_k\ \ w=1,2; \tag{46}$$

similar expressions hold for the other connection types.

In (37), for branches and switches (S), it is:

$$\begin{aligned}\underset{j\in B}{\mathbf{y}^j_h} &= \left[\boldsymbol{\Delta}\mathbf{x}^{j\,\prime}_h\ \boldsymbol{\lambda}^{V_R\,j}_{h+1}\ \boldsymbol{\lambda}^{V_I\,j}_{h+1}\ \boldsymbol{\lambda}^{D_R\,j}_{h+1}\ \boldsymbol{\lambda}^{D_I\,j}_{h+1}\ \boldsymbol{\lambda}^{P\,j}_{h+1}\ \boldsymbol{\lambda}^{Q\,j}_{h+1}\right]',\\ \underset{j\in S}{\mathbf{y}^j_h} &= \left[\boldsymbol{\Delta}\mathbf{x}^{j\,\prime}_h\ \boldsymbol{\lambda}^{V_{R_w}\,j}_{h+1}\ \boldsymbol{\lambda}^{V_{I_w}\,j}_{h+1}\ \boldsymbol{\lambda}^{D_{P_w}\,j}_{h+1}\ \boldsymbol{\lambda}^{D_{Q_w}\,j}_{h+1}\right]'\ w{=}1{,}2.\end{aligned} \tag{47}$$

From (41) and (37), we note that

$$\begin{aligned}\mathbf{A}^{ij} &= \mathbf{A}^{j\,i\,\prime}, && i\neq j,\\ \mathbf{B}^{ij} &= \mathbf{B}^{j\,i\,\prime}, && i\neq j,\\ \mathbf{A}^{ij} &= \mathbf{B}^{ij}, && i\neq j,\ i,j\notin S_c;\end{aligned} \tag{48}$$

they are constants matrices, for they are zero or they contain derivatives of linear constraints.

Matrices $\mathbf{A}^{jj}_h$, $\mathbf{B}^{jj}_k$ $\forall j$, are symmetrical (49); second order derivatives of measurement functions and of equality constraints are neglected.

Matrices $\mathbf{A}^{ij}$, $i\neq j$, depend on which type connections i and j are. But, whichever the parent/child are, matrix $\mathbf{A}^{p(j)j}$ [and then $\mathbf{A}^{j\,p(j)}$ – see (48)] has only four nonzero terms, equal to $+1$ or -1: in the $p(j)$–th part of system (37) they catch $\lambda^{V_R\,j}$, $\lambda^{V_I\,j}$, ΔP^j_i, ΔQ^j_i of child j, while in the j–th part they catch $\lambda^{P\,p(j)}$, $\lambda^{Q\,p(j)}$, $\Delta V^{p(j)}_{R,o}$, $\Delta V^{p(j)}_{I,o}$ of parent $p(j)$. It greatly simplifies the message passing for the inner cycle, and it reduces the computational burden involved in (39). The same goes for matrix $\mathbf{B}^{ij}$, $i\neq j$, and for the outer cycle update message passing and computations (44)-(45).

The form of vector $\mathbf{a}^j_h$ can be recognized by looking at vector $\mathbf{y}^j$ (47); for example [see (2), (36)]:

$$\underset{j\in B}{\mathbf{a}^j_h} = -\begin{bmatrix}\mathbf{h}^j_{\mathbf{x}^j}\mathbf{W}^j[\mathbf{h}^j-\mathbf{z}^j]\\ \mathbf{c}^{V_R\,j}\\ \mathbf{c}^{V_I\,j}\\ \mathbf{c}^{D_R\,j}\\ \mathbf{c}^{D_I\,j}\\ \mathbf{c}^{P\,j}\\ \mathbf{c}^{Q\,j}\end{bmatrix}_h. \tag{50}$$

C. Approximations

At first, we note that the solution does not depend on the value of ξ; it only has to be greater than zero. So, it is possible to adopt for ξ different values in the inner cycle and in the outer cycle, and they can be chosen to obtain the best convergence behaviour.

Inner cycle – A measure of the approximation in the inner cycle is the spectral radius of the neglected part of the matrix governing the iteration:

$$\varrho_i(h) = \sup\ singvalue\big(\boldsymbol{\Delta}\mathbf{N}(h)\big), \tag{51}$$

where $\boldsymbol{\Delta}\mathbf{N}(h)$ is the difference between the complete matrix in the lhs of (37) and the approximated one.

$$\mathbf{A}^{jj}_{\substack{h\\ j\in S_c}} = \begin{bmatrix} \left(\begin{array}{c} \mathbf{h}^j_{\mathbf{x}^j}\mathbf{W}^j\mathbf{h}^{j\,\prime}_{\mathbf{x}^j}+ \\ \xi\left(\begin{array}{c}\mathbf{c}^{D_Rj}_{\mathbf{x}^j}\mathbf{c}^{D_Rj\,\prime}_{\mathbf{x}^j}+\\ \mathbf{c}^{D_Ij}_{\mathbf{x}^j}\mathbf{c}^{D_Ij\,\prime}_{\mathbf{x}^j}\end{array}\right)\end{array}\right)_h & \mathbf{c}^{V_{R_w}{}^j}_{\mathbf{x}^j} & \mathbf{c}^{V_{I_w}{}^j}_{\mathbf{x}^j} & \mathbf{c}^{P_wj}_{\mathbf{x}^j} & \mathbf{c}^{Q_wj}_{\mathbf{x}^j} \\ \mathbf{c}^{V_{R_w}{}^{j\,\prime}}_{\mathbf{x}^j} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{c}^{V_{I_w}{}^{j\,\prime}}_{\mathbf{x}^j} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{c}^{P_w\,j\,\prime}_{\mathbf{x}^j} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{c}^{Q_w\,j\,\prime}_{\mathbf{x}^j} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}_{w=1,2} \qquad \mathbf{B}^{jj}_{\substack{k\\ j\in S_c}} = \begin{bmatrix} \mathbf{A}^{jj}_k & \mathbf{E}^{jj} \\ \mathbf{E}^{jj\,\prime} & \mathbf{0} \end{bmatrix}$$

$$\mathbf{E}^{jj}_{\substack{\\ j\in S_c}} = \begin{bmatrix} \mathbf{c}^{D_Rj}_{\mathbf{x}^j} & \mathbf{c}^{D_Ij}_{\mathbf{x}^j} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{c}^{P_wj\,\prime}_{\alpha^j} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{c}^{Q_wj\,\prime}_{\beta^j} \end{bmatrix}_{w=1,2} \tag{49}$$

The value of $\varrho_i(h)$ is not influenced by the number of closed switches; it is constant and equal to ξ. Numerical experiments on the system of Fig. 4 have shown that also in the inner cycle the convergence is obtained with a wide range of values of ξ. The graph of ε_i against the inner cycle iteration count, h, in the first outer cycle iteration for two values of ξ ($\xi^{\mathrm{a}} = 10\,\xi^{\mathrm{b}}$) is shown in Fig. 9, for both the exact method (dotted line, full markers) and the approximated method (continuous line, empty markers).

It is apparent that the convergence with the exact method does not depend on the value of ξ; it does so with the approximated method, and the bigger the value of ξ, the worse the convergence. On the other hand, the value of ξ can not be arbitrarily low; in particular, it can not be arbitrarily different from the one used in the outer cycle, since the consequent error in the F–O optimality conditions could hamper the convergence of the outer cycle.

Outer cycle – Let $\mathbf{M}_k$ be the matrix in lhs of (40) reduced through the rhs zeros to the space of the four quantities per closed switch updated in the outer cycle. The approximated updating method obtained by neglecting sums in (44) can be represented by matrix $\mathbf{M}^*(k)$:

$$\mathbf{M}^*(k) = \mathbf{M}(k) - \mathbf{\Delta M}(k). \tag{52}$$

As a measure of the approximation due to $\mathbf{\Delta M}_k$, let us consider its spectral radius, $\varrho_o(k)$:

$$\varrho_o(k) = \sup\, singvalue\big(\mathbf{\Delta M}(k)\big), \tag{53}$$

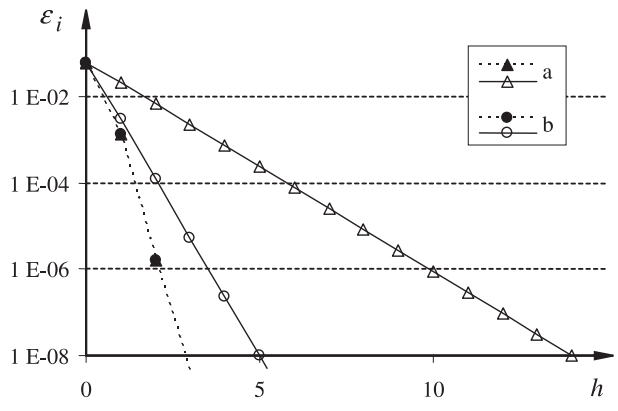

Fig. 9. Convergence of the inner cycle (study network – in the first outer cycle iteration); $\xi^{\mathrm{a}} = \varrho_i^{\mathrm{a}} > \xi^{\mathrm{b}} = \varrho_i^{\mathrm{b}}$.

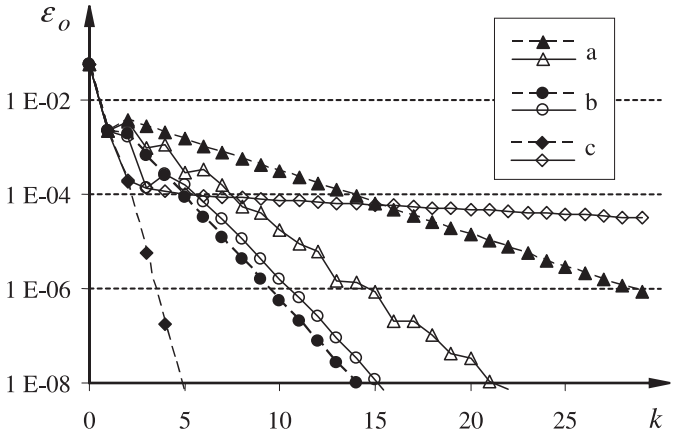

Fig. 10. Convergence at the outer cycle (study network); $\xi^{\mathrm{a}} > \xi^{\mathrm{b}} > \xi^{\mathrm{c}}$ – $\varrho_o^{\mathrm{a}} > \varrho_o^{\mathrm{b}} < \varrho_o^{\mathrm{c}}$.

where $singvalue(\cdot)$ represents the singular values of a matrix; we can expect that the smaller $\varrho_o(k)$, the smaller the errors due to the approximation [28].

In general, the value of $\varrho_o(k)$ depends on the number of times sums in (44) are neglected, and on the value of ξ.

The sums in the rhs of (44) only happen in forks. Analytical development suggests (and numerical experiments on the 69-branch system confirm) that the value of ϱ_o grows as the number of forks whose descendant(s) end in closed switches increases. A rough explanation is that neglecting sums in a fork puts terms in the part of $\mathbf{\Delta M}(k)$ regarding the connections upstream of the fork; these terms do actually contribute to $\mathbf{\Delta M}(k)$ if they are related to non-null terms $\mathbf{b}^j$, namely if there is some closed switch downstream from the fork. The more the forks with some descendant(s) in a loop, the more the terms added up to the same part of $\mathbf{\Delta M}(k)$, the bigger ϱ_o.

The dependence of ϱ_o on ξ is difficult to ascertain with analytical development. Numerical experiments on the simple network of Fig. 4 have been carried out, with different values of ξ. As an example of convergence behavior, in Fig. 10 the graph (continuous line, empty markers) of ε_o is plotted against the outer cycle iteration count, k, for three values of ξ. A big value of ξ is preferable for the convergence speed, while a too low value of ξ slows it down; in the examined case, the best convergence is obtained for the lowest value of $\varrho_o(k)$ (computed in the first iteration). The choice of a good value of ξ requires a few trials.

D. Mismatches

Inner cycle – The inner cycle mismatch, ε_i, is

$$\varepsilon_i = \max_j \{\varepsilon_i^j\}$$

$$\underset{j\in B}{\varepsilon_i^j} = \left\| \begin{bmatrix} \begin{pmatrix} \mathbf{c}_{\mathbf{x}^j}^{V_R\,j}\boldsymbol{\lambda}^{V_R\,j} + \mathbf{c}_{\mathbf{x}^j}^{V_I\,j}\boldsymbol{\lambda}^{V_I\,j} + \\ \mathbf{c}_{\mathbf{x}^j}^{V_R\,C(j)}\boldsymbol{\lambda}^{V_R\,C(j)} + \mathbf{c}_{\mathbf{x}^j}^{V_I\,C(j)}\boldsymbol{\lambda}^{V_I\,C(j)} + \\ \mathbf{c}_{\mathbf{x}^j}^{D_R\,j}\boldsymbol{\lambda}^{V_R\,j} + \mathbf{c}_{\mathbf{x}^j}^{D_I\,j}\boldsymbol{\lambda}^{D_I\,j} + \\ \mathbf{c}_{\mathbf{x}^j}^{P\,p(j)}\boldsymbol{\lambda}^{P\,p(j)} + \mathbf{c}_{\mathbf{x}^j}^{Q\,p(j)}\boldsymbol{\lambda}^{Q\,p(j)} + \\ \mathbf{c}_{\mathbf{x}^j}^{P\,j}\boldsymbol{\lambda}^{P\,j} + \mathbf{c}_{\mathbf{x}^j}^{Q\,j}\boldsymbol{\lambda}^{Q\,j} \end{pmatrix} \\ \mathbf{0} \end{bmatrix} - \mathbf{a}^j \right\|_\infty \tag{54}$$

similar expression hold for ε_i^j for other types of connection. In the OODSE, during the upstream graph tree traversing, every connection computes its ε_i^j and sends upstream the maximum among its ε_i^j and the one(s) received from its child(ren). At the end of the upstream traversing, the root is aware of ε_i.

Outer cycle – The outer cycle mismatch, ε_o, is the maximum absolute value of the rhs term in (40):

$$\varepsilon_o = \max\{\varepsilon_o^j\}, \qquad \varepsilon_o^j = \begin{cases} \left\| \begin{matrix} \mathbf{V}_{R,i_2}^j - \mathbf{V}_{R,i_1}^j \\ \mathbf{V}_{I,i_2}^j - \mathbf{V}_{I,i_1}^j \\ \boldsymbol{\lambda}^{D_{P_1}j} - \boldsymbol{\lambda}^{D_{P_2}j} \\ \boldsymbol{\lambda}^{D_{Q_1}j} - \boldsymbol{\lambda}^{D_{Q_2}j} \end{matrix} \right\|_\infty, & j \in S_c \\ 0, & j \in \{T, S_o\} \end{cases} \tag{55}$$

where $\|\omega\|_\infty$ is the maximum of the absolute values of the components of vector ω (ℓ_∞-norm of vector). In the OODSE, at the beginning of the upstream graph tree traversing, every terminal/switch computes its ε_o^j and sends it upstream. Branches let them flow, while forks send upstream the maximum of the ε_o^j sent by their children. At the end of the upstream traversing, the root is aware of ε_o.

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] A. Monticelli and A. Garcia, "Modeling Zero Impedance Branches in Power System State Estimation," *IEEE Trans. on Power Systems*, vol. 6, no. 4, pp. 1561–1570, November 1991.

[25] G. X. Luo and A. Semlyen, "Efficient Load Flow for Large Weakly Meshed Networks," *IEEE Trans. on Power Systems*, vol. 5, no. 4, pp. 1309–1316, November 1990.

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

[27] 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.

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

[30] 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.

[31] The Mathworks, *MATLAB Reference Guide*, The Mathworks Inc., Natick, MA, USA, 1999.

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.