Robust Optimal Power Flow Solution Using Trust Region and Interior-Point Methods

Andréa A. Sousa, Geraldo L. Torres, *Member IEEE*, Claudio A. Cañizares, *Fellow IEEE*

***Abstract*—A globally convergent optimization algorithm for solving large nonlinear optimal power flow (OPF) problems is presented. As power systems become heavily loaded there is an increasing need for globally convergent OPF algorithms. By global convergence one means the optimization algorithm being able to converge to an OPF solution, if at least one exists, for any choice of initial point. The globally convergent OPF presented is based on an infinity-norm trust region approach, using interior-point methods to solve the trust region subproblems. The performance of the proposed trust region interior-point OPF algorithm, when applied to the IEEE 30-, 57-, 118- and 300-bus systems, and to an actual 1211-bus system, is compared with that of two widely used nonlinear interior-point methods, namely, a pure primal-dual and its predictor-corrector variant.**

***Index Terms*—Optimal Power Flow, Trust Region Method, Interior Point Method, Global Convergence.**

Notation

x	n-vector of optimization variables.
$\underline{x}$	n-vector of lower bounds to x.
$\overline{x}$	n-vector of upper bounds to x.
$f(x)$	scalar nonlinear objective function in n unknowns.
$g(x)$	m-vector of equality constraints functions in (4).
$\nabla f(x)$	n-vector of the gradient of $f(x)$.
$\nabla g(x)$	$n \times m$ gradient matrix (transposed Jacobian) of $g(x)$.
λ	m-vector of Lagrange multipliers associated with the equality constraints $g_i(x) = 0$, for $i = 1, \ldots, m$.
$\nabla^2 f(x)$	$n \times n$ Hessian matrix of $f(x)$.
$\nabla^2 g_i(x)$	$n \times n$ Hessian matrix of constraint function $g_i(x)$.
H_k	$n \times n$ Hessian matrix of Lagrange function associated with problem (4): $H_k = \nabla^2 f(x) + \sum_{i=1}^{m} \lambda_i \nabla^2 g_i(x)$.
m_k	quadratic model function of $f(x)$ around point x_k.
Δ_k	trust region size (radius) at iteration k.
$\bar{\Delta}$	maximum allowed trust region size.
d	n-vector of unknowns in a trust region subproblem.
v	n-vector of unknowns in vertical subproblem (10).
ξ	the contraction factor of the trust region.
Z_k	$n \times (n-m)$ matrix that spans the null space of the gradient matrix $\nabla g(x_k)$, i.e., $Z_k^T \nabla g(x_k) = 0$.
u	$(n-m)$-vector of unknowns in the horizontal subproblem (13).
A_k	Hessian matrix in the QP problem (15).
C_k	constraint matrix in the QP problem (15).
b_k	coeficients of the linear term in the QP problem (15).
w	vector of unknowns in the QP problem (15).
r	vector of slack variables that transform lower bounds in (15) into equalities: $\underline{w} + r - C_k w = 0$.
R	diagonal matrix with $R_{ii} = r_i$.
s	vector of slack variables that transform upper bounds in (15) into equalities: $C_k w + s - \overline{w} = 0$.
S	diagonal matrix with $S_{ii} = s_i$.
π	vector of Lagrange multipliers associated with the lower bound constraints in (15).
υ	vector of Lagrange multipliers associated with the upper bound constraints in (15).
τ	vector of Lagrange multipliers associated with the linear equality constraints in (11).
μ_k	barrier parameter in the interior-point algorithm.
α_k	step length parameter in the interior-point algorithm.
γ	step length reduction factor ($\gamma = 0.9995$).
σ	centering parameter ($\sigma = 0.2$).
$\psi(x, \eta)$	merit function associated with the problem (4).
$\tilde{\psi}(d, \eta)$	merit function model around the point x_k.
pr(d_k)	predicted reduction in the merit function.
ar(d_k)	actual reduction in the merit function.
η	penalty parameter in the merit function.
η^+	a trial value for the penalty parameter η.
d_{soc}	second order correction term to avoid Maratos effect.
OPF	Optimal Power Flow.
NLP	Nonlinear Programming.
QP	Quadratic Programming.
SQP	Sequential Quadratic Programming.
SLP	Sequential Linear Programming.
PDIP	Primal-Dual Interior-Point algorithm.
PCIP	Predictor-Corrector Interior-Point algorithm.
TRIP	Trust Region Interior-Point algorithm.

Manuscript submitted January 2009; accepted August 2010.
This work was supported by the Brazilian research funding agencies CNPq and CAPES.
A. A. Sousa and G. L. Torres are with the Universidade Federal de Pernambuco, Recife, Pernambuco, Brazil (e-mail: gltorres@ufpe.br).
C. A. Cañizares is with the University of Waterloo, Waterloo, Ontario, Canada (e-mail: ccanizar@uwaterloo.ca).

I. Introduction

POWER SYSTEMS in general are currently operating closer to their security limits. This increases the nonlinear behavior of power systems' mathematical models, such as *optimal power flow* (OPF) problems. The restructuring of the power industry has also lead to new complex OPF models [1]–[3], which require robust and reliable solution techniques. Thus, this paper aims at presenting a globally convergent optimization algorithm for solving large-scale nonlinear OPF problems.

Global convergence is a desirable property for any nonlinear optimization algorithm, so that one is able to find a solution, if at least one exists, for any choice of initial point. There are two classical approaches for globalizing a locally convergent algorithm: *line search* procedures, or use of *trust regions* [4],

[5]. Motivated by [6], [7], this paper concentrates on trust region methods due to their significant success in globalizing algorithms for solving unconstrained optimization problems and nonlinear systems of equations. Specially, the application of the trust region technique proposed in [8] and [9] to obtain robust and reliable solutions of OPFs is described here.

Trust region algorithms are a relatively new class of optimization algorithms that minimize a quadratic approximation of a nonlinear objective function within a closed region (a constraint on the size of the step) called the *trust region*; within this region the quadratic model can be trusted to be a good fit to the nonlinear objective function. The achieved reduction in the quadratic approximation should correspond to a reduction in the original objective; if that is not the case, then the size of the trust region is reduced and the approximation model is solved again. There is a broad family of trust region methods [4], [5], and they differ from each other mainly in the way they model the objective function and handle the constraints; the classical approach is based on a quadratic approximation for unconstrained minimization. Applications with equality constraints, functional inequality constraints and simple bound constraints are described in [10], [11]. The Byrd-Omojokun technique proposed in [8] and [9] can be seen as a sequential quadratic programming (SQP) method with a trust region. It decomposes each constrained SQP subproblem into two smaller unconstrained subproblems which are easier to solve, thus making the method attractive for large-scale optimization applications such as nonlinear OPF problems.

Some applications of trust region methods in power system optimization can be found in the literature. The authors in [12] and [13] are the first to use a trust region algorithm to solve state estimation problems. In [14], another application of a trust region method to state estimation is presented. The authors in [15] use a trust region algorithm to solve a reactive power dispatch OPF problem. In [1], a trust region algorithm is employed to deal with convergence issues in non-smooth market-based OPFs. More recently, in [16], a trust region interior-point OPF algorithm is presented, where the trust region constraint is defined using the inifinity-norm, leading to quadratic programming (QP) trust region subproblems that are solved by an interior-point method. The present paper expands on the work presented in [16], improving the algorithms and demonstrating their successful application to large test systems.

Globally convergent optimization algorithms are by nature computationally costly, since code complexness and high processing times are usually the cost for convergence robustness. A trust region method can be associated with interior-point methods to improve efficiency [16]. This can be accomplished in two main ways: (i) applying a trust region method to the interior-point logarithmic barrier problem to globalize its solution, or (ii) applying an interior-point method to solve the trust region subproblems. In this paper, the latter approach is followed; thus, the Byrd-Omojokun trust region technique is first used to decompose each trust region subproblem into two smaller subproblems which are easier to solve, and second an efficient primal-dual interior-point algorithm such as the one described in [17] is employed to solve the generated QP subproblems.

Several standard methods for solving trust region subproblems are discussed in [4], [5], [18], [19]. However, solving the subproblems via interior-point methods, as is done in this paper, allows easy interfacing of the trust region OPF algorithm with interior-point based OPF programs, which is an interesting feature. Practical OPF applications require fast optimization algorithms, such as primal-dual interior-point methods. However, a well-known problem with these algorithms is the lack of global convergence, especially for large system applications. A trust region algorithm allows to address this issue, and can be used when regular OPF solution techniques fail, in spite of the higher computational costs.

The major computational implementation issues of the proposed trust region interior-point algorithm are extensively discussed in the paper, and its computational performance is studied using the IEEE 30-, 57-, 118- and 300-bus systems, and an actual 1211-bus system. Its performance is compared with that of two widely used interior-point algorithms for direct nonlinear OPF solution [20]–[22], i.e., a pure primal-dual and its predictor-corrector variant, bearing in mind that the main focus of the trust region OPF algorithm is on convergence robustness rather than rivaling for processing time.

The paper is organized as follows: In Section II, the basics of trust region methods for unconstrained minimization are briefly described, followed by the mathematical development of the trust region algorithm for solving a general standard form of OPF problems with nonlinear equality constraints and simple bounds. Section III presents the proposed solution of the trust region subproblems by a primal-dual interior-point method and discusses the major implementation issues. In Section IV, numerical results for a power loss minimization OPF problem for various test systems are presented and extensively discussed. Section V summarizes the main contributions of the paper.

II. Trust Region Methods

A. Unconstrained Optimization

Trust region methods were initially developed to solve unconstrained minimization problems of the form [4]:

$$\begin{array}{ll} \min & f(x) \\ \text{s. t.:} & x \in \mathbb{R}^n. \end{array} \tag{1}$$

The information gathered about $f(x)$ is used to construct a *model function* m_k whose behavior near the approximation point x_k should be similar to that of $f(x)$. The function m_k is usually defined as a quadratic function that is derived from the truncated Taylor series expansion of $f(x)$ around x_k, in the form:

$$m_k(x_k + d) = f(x_k) + \nabla f(x_k)^T d + \frac{1}{2} d^T \nabla^2 f(x_k) d \tag{2}$$

where $\nabla f(x_k) \in \mathbb{R}^n$ and $\nabla^2 f(x_k) \in \mathbb{R}^{n\times n}$ are the gradient and the Hessian of $f(x)$ evaluated at x_k, respectively.

If a point $x = x_k + d$ is far from x_k, then model function $m_k(x_k + d)$ may not be a good approximation to $f(x)$; thus, the solution $d_* = \arg\min\ m_k(x_k + d)$ does not always make sense as a minimization step for $f(x)$. For instance, the model function Hessian $\nabla^2 f(x_k)$ may be indefinite, and there are directions along which $m_k(x_k + d)$ is unbounded from below; in this case, $\|d\|$ is infinite. Therefore, to globalize the algorithm, the search for a minimizer of m_k is restricted to some region around x_k, called the trust region. This trust region is usually a ball defined by the Euclidean norm $\|d\|_2 \leq \Delta_k$, where the scalar Δ_k is called the *trust region radius*. In principle, any other norm can be used [4]. The trust region subproblem around the point x_k for the unconstrained minimization problem (1) is then defined as:

$$\begin{aligned} \min \quad & f(x_k) + \nabla f(x_k)^T d + \frac{1}{2} d^T \nabla^2 f(x_k) d \\ \text{s. t.:} \quad & \|d\|_2 \leq \Delta_k. \end{aligned} \tag{3}$$

If the candidate solution $x_{k+1} = x_k + d_k$ does not produce a sufficient decrease in $f(x)$, according to a merit function, then the trust region as given by the radius Δ_k is too large. Hence, to proceed, the radius Δ_k is reduced and the trust region subproblem (3) is solved again.

B. The Byrd-Omojokun Trust Region Method

Trust region methods for general constrained optimization can be also used to solve highly nonlinear OPF problems of the general form:

$$\min \quad f(x) \tag{4a}$$
$$\text{s. t.:} \quad g(x) = 0 \tag{4b}$$
$$\underline{x} \leq x \leq \overline{x} \tag{4c}$$

where $x \in \mathbb{R}^n$ is a vector of decision variables, including the control and the state variables; $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is a nonlinear vector function including conventional power flow equations and other equality constraints such as power balance across boundaries in a pool operation; and $\underline{x}$ and $\overline{x}$ are lower and upper bounds on the variables x, corresponding to physical and operating limits on the system.

Trust region methods to solve the constrained problem (4) usually employ a QP approximation along with a trust region constraint [6], as follows:

$$\min \quad f(x_k) + \nabla f(x_k)^T d + \frac{1}{2} d^T H_k d \tag{5a}$$
$$\text{s. t.:} \quad g(x_k) + \nabla g(x_k)^T d = 0 \tag{5b}$$
$$\underline{x} \leq x_k + d \leq \overline{x} \tag{5c}$$
$$\|d\| \leq \Delta_k \tag{5d}$$

where H_k is the Hessian with respect to x of the Lagrange function associated with (4), i.e.,

$$H_k = \nabla^2 f(x_k) + \sum_{i=1}^{m} \lambda_i^k \nabla^2 g_i(x_k), \tag{6}$$

and λ_i is the Lagrange multiplier associated with the equality constraint $g_i(x) = 0$.

The Byrd-Omojokun trust region technique [8], [9] was originally developed to solve equality constrained problems, and later it was adapted by Plantenga [6] to solve problems with equality constraints and simple bound constraints, such as (4). As with most nonlinear optimization methods, handling inequality constraints in trust region methods is not as simple as handling equalities. Motivated by successful procedures used in interior-point methods to handle inequalities, mainly in the dual affine scaling variant [17], Plantenga proposed an affine scaling transformation for handling the simple bound constraints. The affine scaling allows for the bound constraints to be dropped as long as the trust region size is properly chosen. However, the scaled trust region subproblem becomes badly ill-conditioned whenever x_k approaches the boundary. To deal with this ill-conditioning, Plantenga then proposed the inclusion of a logarithmic barrier term into $f(x)$, similarly to the approach used in the primal-dual interior-point method. The performance of this approach, however, was not adequate.

Based on [23], in this paper, the trust region is defined using the infinity norm as opposed to the Euclidean norm, so that the trust region subproblem can be expressed as:

$$\min \quad f(x_k) + \nabla f(x_k)^T d + \frac{1}{2} d^T H_k d \tag{7a}$$
$$\text{s. t.:} \quad g(x_k) + \nabla g(x_k)^T d = 0 \tag{7b}$$
$$\underline{x} \leq x_k + d \leq \overline{x} \tag{7c}$$
$$\|d\|_\infty \leq \Delta_k. \tag{7d}$$

Since $\|d\|_\infty = \max_i |d_i|$, the constraints (7c) and (7d) on the step d can be combined into a single simple bound constraint, so that the subproblem (7) can be rewritten as:

$$\min \quad f(x_k) + \nabla f(x_k)^T d + \frac{1}{2} d^T H_k d \tag{8a}$$
$$\text{s. t.:} \quad g(x_k) + \nabla g(x_k)^T d = 0 \tag{8b}$$
$$\max\{\underline{\delta}, -\Delta_k\} \leq d \leq \min\{\overline{\delta}, \Delta_k\} \tag{8c}$$

where $\underline{\delta} = \underline{x} - x_k$ and $\overline{\delta} = \overline{x} - x_k$; the "max" and "min" operators on vectors are applied componentwise.

Because the trust region constraint (7d) is defined using the infinity norm, all constraints in (8) are linear, so that the corresponding trust region subproblem is a box-constrained QP problem. However, regardless the norm used, the trust region constraint (8c) could limit the size of the step d in such a way that the equality constraints (8b) become infeasible, i.e., a step d that satisfies (8b) may not intercept the trust region (8c), so that the feasible set defined by (8b) and (8c) may be empty.

If x_k is far from satisfying one or more of the linearized constraints in (8b) and the trust region size Δ_k is small, then the constraints (8b) and (8c) may be inconsistent, i.e., there is no step d that satisfies both the linear equality constraint (8b) and the trust region constraint (8c). On the other hand, the global convergence of the trust region method depends on being able to reduce Δ_k until the model subproblem accurately represents the actual problem. Therefore, inconsistent constraints may be unavoidable in a trust region subproblem. Several approaches have been proposed for applying a trust region constraint in a manner that does not generate inconsistent subproblems. The Byrd-Omojokun approach is used here.

In order to resolve the possible conflict between satisfying the constraints (8b) and (8c), the solution of (8) by the Byrd-Omojokun trust region technique is divided into two new subproblems, called the *vertical* (or *normal*) subproblem and the *horizontal* (or *tangential*) subproblem. It is the ability of trust region methods to deal with inconsistent linearized constraints that makes them globally convergent algorithms, as opposed to some popular sequential optimization approaches such as SLP and SQP.

1) The Vertical Subproblem: The vertical subproblem is defined as

$$\min \quad \frac{1}{2}\|g(x_k)+\nabla g(x_k)^T v\|_2^2 \tag{9a}$$

$$\text{s. t.:} \quad \max\{\underline{\delta}, -\xi\Delta_k\} \le v \le \min\{\overline{\delta}, \xi\Delta_k\} \tag{9b}$$

where $\xi \in (0,1)$ is a contraction factor to obtain a tighter version of the trust region (typically, $\xi = 0.8$). By ignoring the constant term $\frac{1}{2}g(x_k)^T g(x_k)$ in the quadratic objective, the vertical subproblem (9) can be rewritten as:

$$\min \quad (\nabla g(x_k)g(x_k))^T v + \frac{1}{2}v^T \nabla g(x_k)\nabla g(x_k)^T v \tag{10a}$$

$$\text{s. t.:} \quad \max\{\underline{\delta}, -\xi\Delta_k\} \le v \le \min\{\overline{\delta}, \xi\Delta_k\}. \tag{10b}$$

The role of the vertical subproblem is to find a *vertical step* v that lies well inside the trust region (according to the contraction factor ξ) and that attempts to satisfy the equality constraints (8b) as much as possible (minimizing the squared Euclidean norm of the constraints residuals). Notably, the vertical subproblem involves the minimization of a convex quadratic function over a box, so that it always has a solution that can be obtained by well-known algorithms for this kind of problem. Standard methods for solving trust region subproblems are discussed in [4], [5], [18], [19].

2) The Horizontal Subproblem: After solving (10) for the vertical step v_k, the full step d_k is obtained by solving the horizontal subproblem

$$\min \quad \nabla f(x_k)^T d + \frac{1}{2}d^T H_k d \tag{11a}$$

$$\text{s. t.:} \quad \nabla g(x_k)^T d = \nabla g(x_k)^T v_k \tag{11b}$$

$$\max\{\underline{\delta}, -\Delta_k\} \le d \le \min\{\overline{\delta}, \Delta_k\}. \tag{11c}$$

Note that equality constraint (11b) is a relaxation of equality constraint (8b), so that (11b) and (11c) are always consistent; e.g., $d = v_k$ is a feasible solution. However, it is the vertical subproblem (10) that relaxes the original constraints enough to allow for consistency [6]. The constraint (11b) simply has the effect of forcing the final step d_k to make the same progress as v_k does towards satisfying the linearized equality constraints while minimizing the quadratic objective function.

The only constraints in the vertical subproblem are simple bound constraints, whereas the horizontal subproblem (11) has both simple bounds and linear equality constraints. But one may also reformulate (11) without equality constraints. To this end, Byrd and Omojokun proposed to compute d_k by means of a step complementary to v_k. Such a step is orthogonal do v_k and belongs to the null space of $\nabla g(x_k)$. Thus, let Z_k be an $n \times (n-m)$ matrix that spans the null space of $\nabla g(x_k)$, i.e., $\nabla g(x_k)^T Z_k = 0$. Then, the full step d can be computed from

$$d = v_k + Z_k u \tag{12}$$

where $u \in \mathbb{R}^{n-m}$ becomes the unknown vector. Using (12) in (11), observing that v_k and $Z_k u$ are orthogonal, i.e., $v_k^T Z_k u = 0$, and ignoring constant terms in the objective function, the horizontal subproblem (11) can be rewritten without equality constraints, in the form:

$$\min \ \nabla \bar{f}_k^T Z_k u + \frac{1}{2}u^T Z_k^T H_k Z_k u \tag{13a}$$

$$\text{s. t.:} \ \max\{\underline{\delta}, -\Delta_k\} - v_k \le Z_k u \le \min\{\overline{\delta}, \Delta_k\} - v_k \tag{13b}$$

where, for shortness, $\nabla \bar{f}_k = \nabla f(x_k) + H_k v_k$. The subproblem (13) is much smaller than (11) but it requires computing the orthogonal complement Z_k.

The basic steps of the Byrd-Omojokun technique for solving the trust region subproblem (8) are as follows: First, (10) is solved for the vertical step v_k. Then, (13) is solved for the horizontal step $Z_k u_k$, and then using (12) the full step d_k is obtained. Finally, a trial solution is determined as follows:

$$x_{k+1} = x_k + d_k \tag{14}$$

as long as x_{k+1} provides a sufficient decrease in a merit function (see Section III-E); otherwise, the size of the trust region is reduced and a new candidate solution is computed.

III. Implementation Issues

In this section, the solutions of the vertical and horizontal subproblems are described, and important implementation issues are discussed. In [6]–[9], the trust region is defined using the Euclidean norm and the subproblems are approximately solved using the Powell's *dogleg* method and the Steihaug's conjugate gradient method. In this paper the infinity-norm is used, and the subproblems are then solved by a primal-dual interior-point method for QP.

A. Single Standard Form for the Subproblems

A single notation and standard form for the vertical and horizontal subproblems can be defined, so that they can be solved by the same interior-point algorithm, as follows:

$$\min \quad b_k^T w + \frac{1}{2}w^T A_k w \tag{15a}$$

$$\text{s. t.:} \quad \underline{w} \le C_k w \le \overline{w} \tag{15b}$$

where the index k stands for a subproblem at iteration k. To model the vertical subproblem (10), in (15), $w = v$ and

$$b_k = \nabla g(x_k) g(x_k) \tag{16a}$$

$$A_k = \nabla g(x_k)\nabla g(x_k)^T \tag{16b}$$

$$C_k = I \tag{16c}$$

$$\underline{w} = \max\{\underline{\delta}, -\xi\Delta_k\} \tag{16d}$$

$$\overline{w} = \min\{\overline{\delta}, \xi\Delta_k\}. \tag{16e}$$

To model the horizontal subproblem (13), in (15), $w = u$ and

$$b_k = Z_k^T \nabla \bar{f}_k \tag{17a}$$

$$A_k = Z_k^T H_k Z_k \tag{17b}$$

$$C_k = Z_k \tag{17c}$$

$$\underline{w} = \max\{\underline{\delta},\ -\Delta_k\} - v_k \tag{17d}$$

$$\overline{w} = \min\{\overline{\delta},\ \Delta_k\} - v_k. \tag{17e}$$

1. Set $k_j = 0$, choose $\mu_0 > 0$ and a starting point y_0 such that $(r_0, s_0, \pi_0, \upsilon_0) > 0$.
2. Obtain the Newton system to (21) at the current point y_{k_j}, and compute the search direction Δy from
$$\nabla^2_{yy} L(y_{k_j}, \mu_{k_j})\Delta y = -\nabla_y L(y_{k_j}, \mu_{k_j}).$$
3. Compute a new solution estimate from
$$y_{k_j+1} = y_{k_j} + \alpha_{k_j}\Delta y$$
where α_{k_j} is the smallest of the primal and dual steps
$$\alpha^P_{k_j} = \min\left\{\gamma \min_{\Delta_i<0}\left\{\frac{-r_i^{k_j}}{\Delta r_i}, \frac{-s_i^{k_j}}{\Delta s_i}\right\}, 1\right\} \tag{19a}$$
$$\alpha^D_{k_j} = \min\left\{\gamma \min_{\Delta_i<0}\left\{\frac{-\pi_i^{k_j}}{\Delta \pi_i}, \frac{-\upsilon_i^{k_j}}{\Delta \upsilon_i}\right\}, 1\right\} \tag{19b}$$
4. If $\|\nabla_y L(y_{k_j+1}, \mu_{k_j})\| \leq \epsilon$, then stop. Otherwise, obtain
$$\mu_{k_j+1} = \sigma \frac{r^T_{k_j+1}\pi_{k_j+1} + s^T_{k_j+1}\upsilon_{k_j+1}}{2p}, \tag{20}$$
set $k_j = k_j + 1$, and return to Step 2.

Alg. 1: Interior-point algorithm for solving (15).

Since the gradient matrix $\nabla g(x_k)$ is of size $n \times m$ with $n > m$, the $n \times n$ matrix A_k in (16b) is symmetric positive semidefinite, with $n - m$ linearly independent null vectors.

B. The Primal-Dual Interior-Point Solver

The primal-dual interior-point method to solve the bound constrained QP problem (15) operates on the modified problem

$$\begin{aligned} \min \quad & b_k^T w + \frac{1}{2} w^T A_k w - \mu_{k_j} \sum_{i=1}^{p} (\ln r_i + \ln s_i) \\ \text{s. t.:} \quad & \underline{w} + r - C_k w = 0, \\ & C_k w + s - \overline{w} = 0, \end{aligned} \tag{18}$$

where r and s are p-vectors of slack variables that transform the simple bounds into equalities, and $\mu_{k_j} > 0$ is a *barrier parameter* that is monotonically reduced to zero as iterations progress. Strict positivity conditions on the slacks, $r > 0$ and $s > 0$, are handled implicitly through step length control.

If w_* is a local minimizer of (18), then there exist vectors of Lagrange multipliers, (π_*, υ_*), that satisfy the Karush-Kuhn-Tucker (KKT) first-order optimality conditions:

$$R\pi - \mu_{k_j} e = 0, \tag{21a}$$
$$S\upsilon - \mu_{k_j} e = 0, \tag{21b}$$
$$\underline{w} + r - C_k w = 0, \tag{21c}$$
$$C_k w + s - \overline{w} = 0, \tag{21d}$$
$$b_k + A_k w - C_k^T \pi + C_k^T \upsilon = 0, \tag{21e}$$

where R and S are diagonal matrices with $R_{ii} = r_i$ and $S_{ii} = s_i$, and e is a p-vector of ones. For compactness of presentation, one can rewrite (21) in the compact form:

$$\nabla_y L(y, \mu_{k_j}) = 0 \tag{22}$$

where $y = (r, s, \pi, \upsilon, w)$. The main steps of the interior-point algorithm for solving (15) are shown in Alg. 1. The large linear indefinite system solved in Step 2 is:

$$\begin{bmatrix} \Pi & 0 & R & 0 & 0 \\ 0 & \Upsilon & 0 & S & 0 \\ I & 0 & 0 & 0 & -C_k \\ 0 & I & 0 & 0 & C_k \\ 0 & 0 & -C_k^T & C_k^T & A_k \end{bmatrix} \begin{pmatrix} \Delta r \\ \Delta s \\ \Delta \pi \\ \Delta \upsilon \\ \Delta w \end{pmatrix} = - \begin{pmatrix} R\pi - \mu_{k_j} e \\ S\upsilon - \mu_{k_j} e \\ \underline{w} + r - C_k w \\ C_k w + s - \overline{w} \\ b_k + A_k w - C_k^T \pi + C_k^T \upsilon \end{pmatrix} \tag{23}$$

where Π and Υ are diagonal matrices with $\Pi_{ii} = \pi_i$, and $\Upsilon_{ii} = \upsilon_i$. In Step 3, $\gamma = 0.9995$ is a safety factor to ensure that the next point will satisfy the strict positivity conditions on slacks and dual variables. In Step 4, $\sigma = 0.2$ is the expected, not necessarily realized, cut in complementarity residual.

C. Solving the Horizontal Subproblem Without Z_k

Computing the orthogonal complement Z_k may be time consuming, and thus d_k should be directly computed from (11). In this case the problem is not in the standard form (15), requiring a modification of the Newton system in order to accommodate the linear equality constraint and corresponding Lagrange multiplier vector τ. To this end, in (15), $w = d$ and

$$b_k = \nabla f(x_k) \tag{24a}$$
$$A_k = H_k \tag{24b}$$
$$C_k = I \tag{24c}$$
$$\underline{w} = \max\{\underline{\delta}, -\Delta_k\} \tag{24d}$$
$$\overline{w} = \min\{\overline{\delta}, \Delta_k\}. \tag{24e}$$

The Newton system in Step 2 of the IP algorithm becomes:

$$\begin{bmatrix} \Pi & 0 & R & 0 & 0 & 0 \\ 0 & \Upsilon & 0 & S & 0 & 0 \\ I & 0 & 0 & 0 & 0 & -C_k \\ 0 & I & 0 & 0 & 0 & C_k \\ 0 & 0 & 0 & 0 & 0 & \nabla g(x_k)^T \\ 0 & 0 & -C_k^T & C_k^T & \nabla g(x_k) & A_k \end{bmatrix} \begin{pmatrix} \Delta r \\ \Delta s \\ \Delta \pi \\ \Delta \upsilon \\ \Delta \tau \\ \Delta w \end{pmatrix} = - \begin{pmatrix} R\pi - \mu_{k_j} e \\ S\upsilon - \mu_{k_j} e \\ \underline{w} + r - C_k w \\ C_k w + s - \overline{w} \\ \nabla g(x_k)^T (w - v_k) \\ b_k + A_k w + \nabla g(x_k)\tau - C_k^T \pi + C_k^T \upsilon \end{pmatrix}. \tag{25}$$

D. Obtaining the Lagrange Multipliers to Evaluate H_k

The Lagrange multipliers associated with the nonlinear OPF problem (4) are not explicitly computed by the trust region method. Then, to evaluate the Hessian matrix H_k using (6), in each trust region iteration, one can use a least-squares estimate for λ_k [6], computed from:

$$(\nabla g(x_k)^T \nabla g(x_k))\lambda_k = -\nabla g(x_k)^T \nabla f(x_k). \tag{26}$$

Under the regularity assumption, $\nabla g(x_k)$ is full rank and the positive definite system (26) can be efficiently solved by Cholesky factorization or a conjugate gradient method.

The system (26) is derived from the dual KKT equation of an NLP problem without bound constraints, i.e., it is suitable for equality constrained problems only. In the existence of the bound constraint (4c), the dual KKT equation is

$$\nabla f(x) + \nabla g(x)\lambda - \widetilde{\pi} + \widetilde{\upsilon} = 0. \tag{27}$$

Due to the sign conditions on the Lagrange multipliers $\widetilde{\pi}$ and $\widetilde{\upsilon}$ associated with the lower and upper bounds, respectively, a special approach to compute λ is described in [6]. Thus, assume that the variables x are reordered and partitioned as $x = (x_L, x_I, x_U)$, where L is the index set of variables binding at lower bounds, I of inactive variables, and U of variables binding at upper bounds; this same index partition convention is applied to $\nabla f(x)$, $\nabla g(x)$, $\widetilde{\pi}$ and $\widetilde{\upsilon}$. The complementarity conditions imply that $\widetilde{\pi}_I = \widetilde{\pi}_U = 0$ and $\widetilde{\upsilon}_L = \widetilde{\upsilon}_I = 0$, so that the KKT equation (27) becomes

$$\begin{bmatrix} \nabla g(x)_L \\ \nabla g(x)_I \\ \nabla g(x)_U \end{bmatrix} \lambda = - \begin{bmatrix} \nabla f(x)_L \\ \nabla f(x)_I \\ \nabla f(x)_U \end{bmatrix} + \begin{bmatrix} \widetilde{\pi}_L \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ \widetilde{\upsilon}_U \end{bmatrix}. \tag{28}$$

Assuming that $\nabla g(x)_I$ is full rank, the nonzero Lagrange multipliers are given by

$$\lambda = -(\nabla g(x)_I^T \nabla g(x)_I)^{-1} \nabla g(x)_I^T \nabla f(x)_I \tag{29a}$$
$$\widetilde{\pi}_L = \nabla f(x)_L + \nabla g(x)_L \lambda \tag{29b}$$
$$\widetilde{\upsilon}_U = -\nabla f(x)_U - \nabla g(x)_U \lambda \tag{29c}$$

Initially, the sets L and U are obtained as $L = \{i \mid x_i - \underline{x}_i < \epsilon\}$ and $U = \{i \mid \overline{x}_i - x_i < \epsilon\}$. Next, an estimate λ is computed by (29a), followed by a check on $\widetilde{\pi}_L$ and $\widetilde{\upsilon}_U$ given by (29b) and (29c) to see if they have the correct positive sign. If a sign error occurs, then one or more of the indices in L and U do not belong there. The bound with the greatest sign violation is removed, yielding new index sets. This process is repeated until no sign errors occur (or sets L and U become empty).

In this paper, advantage is taken of solving the horizontal subproblem by interior-point methods to devise a very simple, accurate and inexpensive estimate of the Lagrange multipliers λ_k. Thus, simply set $\lambda_k = \tau_*$, where τ_* is the optimal Lagrange multiplier vector associated with the linear equality constraint (11b) from the horizontal subproblem in trust region iteration k (see Section III-C). As shown in Section IV-A, Table XII, this estimate for λ_k is accurate and readily available after solving the horizontal subproblem (11) in each trust region iteration.

E. Merit Function

Having computed d_k, a *merit function* is used to decide if this step sufficiently decreases or not the objective $f(x)$. Based on [6] and [7], the following merit function is used:

$$\psi(x, \eta) = f(x) + \eta \|g(x)\| \tag{30}$$

where $\eta > 0$ is a *penalty parameter* that weights constraint satisfaction relative to objective minimization. A procedure for choosing and updating η is described in [7]. Thus, given a step d_k, η should be chosen large enough so that d_k results in a reduction of a merit function model around x_k:

$$\tilde{\psi}(d, \eta) = f(x_k) + \nabla f(x_k)^T d + \frac{1}{2} d^T H_k d + \eta \|g(x_k) + \nabla g(x_k)^T d\|. \tag{31}$$

This merit function model is used to obtain the *predicted reduction* in the merit function (30), which is defined as:

$$\mathrm{pr}(d_k) = \tilde{\psi}(0, \eta) - \tilde{\psi}(d_k, \eta) = -\nabla f(x_k)^T d_k - \frac{1}{2} d_k^T H_k d_k + \eta(\|g(x_k)\| - \|g(x_k) + \nabla g(x_k)^T v_k\|) \tag{32}$$

where the relations $d_k = v_k + Z_k u_k$ and $\nabla g(x_k)^T Z_k = 0$ are used. Since v_k reduces $\|g(x_k) + \nabla g(x_k)^T v_k\|$ with respect to $\|g(x_k)\|$, the term within parenthesis is positive. The predicted reduction $\mathrm{pr}(d_k)$ is used as a measure for accepting or not the step d_k, and for updating the trust region, by comparing it to the *actual reduction*, defined as:

$$\begin{aligned} \mathrm{ar}(d_k) &= \psi(x_k, \eta) - \psi(x_k + d_k, \eta) \\ &= f(x_k) - f(x_k + d_k) + \eta(\|g(x_k)\| - \|g(x_k + d_k)\|) \end{aligned} \tag{33}$$

If the reduction ratio $\mathrm{ar}(d_k)/\mathrm{pr}(d_k)$ is close to 1, then there is good agreement between the model $\tilde{\psi}_k$ and the function ψ over this step, and thus it is safe to expand the trust region for the next iteration [4]. If $\mathrm{ar}(d_k)/\mathrm{pr}(d_k)$ is positive but not close to 1, then the trust region is not altered, but if it is close to zero or negative the trust region is reduced. The trust region update is explained in detail at the end of this section.

A base condition for choosing and updating the penalty η is that it should be large enough for d_k to provide a reduction in $\tilde{\psi}(d, \eta)$. First, a trial value η^+ is obtained as:

$$\eta^+ = \max\left\{\eta_k,\ 0.1 + \frac{\nabla f(x_k)^T d_k + \frac{1}{2} d_k^T H_k d_k}{\|g(x_k)\| - \|g(x_k) + \nabla g(x_k)^T v_k\|}\right\} \tag{34}$$

where η_k is the penalty used in the previous iteration. Next, the trial value is improved using Alg. 2, proposed in [7]. The first *if* condition in Alg. 2 allows for an additional increase in η^+, if little progress is made towards feasibility ($\|g(x_k)\| > \frac{1}{5}\|g(x_{k-l})\|$) and the last two steps were not both successful, suggesting that the linearized model of the constraints is not a good fit. Safeguards are included to prevent η^+ from changing too fast ($\eta^+ < 5\eta_k$) and to allow increases only when the parameter is already on the rise ($\eta_k > \eta_{k-l}$). The second *if* condition allows for reductions in η^+ only when the iterations are close to satisfying the equalities, and the reduced η^+ is neither lower than $\bar{\eta}$ (the value that ensures a descent step for the merit function) nor lower than $\|\lambda_k\|$.

The merit function (30) is non-smooth and suffers from the *Maratos effect* [24], performing poorly on some problems. This effect can be avoided by computing a *second order correction* term that is added to d_k [6], yielding the trial step

$$d_{\text{soc}} = d_k - \nabla g(x_k)(\nabla g(x_k)^T \nabla g(x_k))^{-1} g(x_k + d_k). \tag{35}$$

Since computing d_{soc} increases the computational costs, d_{soc} is computed only if the step d_k is to be rejected, x_k is nearly feasible, and $\|v_k\|$ is small compared to $\|Z_k u_k\|$; these conditions try to identify the Maratos effect in the proximity of a feasible point. Based on [6], the introduction of the second order correction and updating of the trust region and penalty parameter are described in Alg. 3. The basic steps of the Byrd-Omojokun technique are summarized in Alg. 4.

1) If $\|g(x_k)\| > \frac{1}{5}\|g(x_{k-l})\|$, $\eta^+ > \eta_k$, $\eta^+ < 5\eta_k$, $\eta_k > \eta_{k-l}$, and last two steps were not both successful, then
$$\eta^+ = \min\left\{5\eta_k,\ \eta^+ + 25(\eta^+ - \eta_{k-l})\right\}$$
where $k-l$ is the index of the last successful step.
2) If $\eta^+ = \eta_k$, $\|v_k\| < \dfrac{\xi\Delta_k}{10}$ and $\|g(x_k)\|_\infty < 10^4\epsilon$, then
$$\eta^+ = \max\{\eta_0,\ \bar{\eta},\ \|\lambda_k\|\}$$
where $\bar{\eta}$ is the second term within braces in (34).

Alg. 2: Heuristic for updating the penalty parameter.

1) Compute η^+ by Alg. 2, actual reduction $\mathrm{ar}(d_k)$ by (33) and predicted reduction $\mathrm{pr}(d_k)$ by (32).
2) If $\dfrac{\mathrm{ar}(d_k)}{\mathrm{pr}(d_k)} \geq \bar{\rho}$, then set $x_{k+1} = x_k + d_k$, $\eta_{k+1} = \eta^+$, $\Delta_{k+1} \geq \Delta_k$ and return.
3) If $\|v_k\| \leq 0.8\xi\Delta_k$ and $\|v_k\| \leq 0.1\|Z_k u_k\|$, then obtain d_{soc} by (35), compute $\mathrm{ar}(d_{\text{soc}})$ and go to step 4. If not, set $x_{k+1} = x_k$, $\eta_{k+1} = \eta_k$, $\Delta_{k+1} \leq \gamma\|d_k\|$ and return.
4) If $\dfrac{\mathrm{ar}(d_{\text{soc}})}{\mathrm{pr}(d_k)} \geq \bar{\rho}$, then set $x_{k+1} = x_k + d_{\text{soc}}$, $\eta_{k+1} = \eta^+$, $\Delta_{k+1} \geq \Delta_k$ and return. If not, set $x_{k+1} = x_k$, $\eta_{k+1} = \eta_k$, $\Delta_{k+1} \leq \gamma\|d_k\|$ and return.

Alg. 3: Updating the trust region and the solution estimate.

1) Set $k = 0$, choose x_0 within bounds, set maximum trust region size $\bar{\Delta} > 0$, choose $\Delta_0 \in (0, \bar{\Delta})$, $\epsilon \in \left(0, \frac{1}{2}\right]$ and $\bar{\rho} \in \left(0, \frac{1}{2}\right]$.
2) If $\|\nabla f(x_k) + \nabla g(x_k)^T \lambda_k\| < \epsilon$ and $\|g(x_k)\|_\infty < \epsilon$, then stop.
3) Solve the vertical subproblem (10) for the step v_k.
4) Solve the horizontal subproblem (13) for the step u_k.
5) Set the solution of subproblem (5) as $d_k = v_k + Z_k u_k$.
6) Compute $\mathrm{pr}(d_k)$, $\mathrm{ar}(d_k)$ and η^+ from (32) to (34).
7) Compute x_{k+1}, η_{k+1} and Δ_{k+1} by Alg. 3. Set $k = k+1$ and go back to step *2*.

Alg. 4: The trust region algorithm.

The update of the trust region size shown in Alg. 3 as either $\Delta_{k+1} \geq \Delta_k$ or $\Delta_{k+1} \leq \gamma\|d_k\|$, with $\gamma \in (0, 1)$, is as follows: Based on [6], when a step d_k (or d_{soc}) is accepted (in step 2 or step 4 of Alg. 3), the trust region size is increased as:

$$\Delta_{k+1} = \begin{cases} \max\{2.0\|d_k\|, \Delta_k\} & \text{if } \frac{\mathrm{ar}(d_k)}{\mathrm{pr}(d_k)} \geq 0.9 \\ \max\{1.2\|d_k\|, \Delta_k\} & \text{if } 0.3 \leq \frac{\mathrm{ar}(d_k)}{\mathrm{pr}(d_k)} < 0.9 \\ \Delta_k & \text{otherwise} \end{cases} \quad (36)$$

When a step d_k is rejected (second *if* condition in step 2 or step 4 of Alg. 3), Δ_{k+1} is reduced to a fraction $\gamma \in [0.1, 0.5]$ of the failed step length, with γ given by

$$\gamma = \begin{cases} \dfrac{1-\bar{\rho}}{1 - \frac{\mathrm{ar}(d_k)}{\mathrm{pr}(d_k)}} & \text{if } \|d_k\| = \Delta_k \\ \dfrac{\dot{p}(0)}{2\big(\dot{p}(0) - \frac{\mathrm{ar}(d_k)}{\mathrm{pr}(d_k)}\big)} & \text{otherwise} \end{cases} \quad (37)$$

where $\dot{p}(0) = -\nabla f(x_k)^T d_k + \eta_{k+1}\|\nabla g(x_k)^T d_k\|$. In both cases the value of γ is kept within the interval $[0.1, 0.5]$.

IV. Numerical Results

The proposed trust region interior-point OPF algorithm (TRIP) was tested with the IEEE 30-, 57-, 118- and 300-bus systems, and an actual 1211-bus system. Its performance can be compared with that of two successful and widely used interior-point (IP) algorithms for direct nonlinear OPF solutions: a pure primal-dual (PDIP) and its predictor-corrector variant (PCIP) [20]–[22]. The OPF problem solved here, without loss of generality, is the classical active power loss minimization. The constraints sets include the active and reactive power balance equations for all buses, and lower and upper bounds on voltage magnitudes, tap ratios, shunt susceptances, generators reactive power output, and selected branch flows. The corresponding mathematical formulation of this OPF is as follows:

$$\begin{aligned} \min \quad & \sum_{(i,j)\in\mathcal{B}} g_{ij}(V_i^2 + V_j^2 - 2V_iV_j\cos\theta_{ij}) \\ \text{s. t.:} \quad & P_i(\theta,V,t) - P_{G_i} + P_{D_i} = 0, && \forall i \in \mathcal{N} \\ & Q_i(\theta,V,t) - Q_{G_i} + Q_{D_i} = 0, && \forall i \in \mathcal{G} \\ & Q_i(\theta,V,t) - Q_{G_i} + Q_{D_i} = 0, && \forall i \in \mathcal{F} \\ & Q_i(\theta,V,t) + Q_{D_i} + b_i^{\text{sh}}V_i^2 = 0, && \forall i \in \mathcal{E} \\ & F_{ij}(\theta,V,t) - f_{ij} = 0, && \{(i,j)\} \subset \mathcal{B} \\ & V_i^{\min} \leq V_i \leq V_i^{\max}, && \forall i \in \mathcal{N} \\ & t_{ij}^{\min} \leq t_{ij} \leq t_{ij}^{\max}, && \{(i,j)\} \subset \mathcal{T} \\ & b_i^{\min} \leq b_i^{\text{sh}} \leq b_i^{\max}, && \forall i \in \mathcal{E} \\ & Q_i^{\min} \leq Q_{G_i} \leq Q_i^{\max}, && \forall i \in \mathcal{G} \\ & f_{ij}^{\min} \leq f_{ij} \leq f_{ij}^{\max}, && \{(i,j)\} \subset \mathcal{B} \end{aligned} \quad (38)$$

where

$$P_i(\theta,V,t) = V_i \sum_{j\in\mathcal{N}_i} V_j(G_{ij}\cos\theta_{ij} + B_{ij}\sin\theta_{ij}) \quad (39)$$

$$Q_i(\theta,V,t) = V_i \sum_{j\in\mathcal{N}_i} V_j(G_{ij}\sin\theta_{ij} - B_{ij}\cos\theta_{ij}) \quad (40)$$

V_i and θ_i are the voltage magnitude and phase angle ($\theta_{ij} = \theta_i - \theta_j$); P_{G_i} and P_{D_i} are active power generation and demand; Q_{G_i} and Q_{D_i} are reactive power generation and demand, and b_i^{sh} is the shunt susceptance, all at bus i; g_{ij} is the conductance of a circuit connecting the bus i to bus j; $Y_{ij} = G_{ij} + jB_{ij}$ is the ij-th element of the bus admittance matrix; $\mathcal{N}$ is the index set of all system buses, $\mathcal{G}$ of generator buses, $\mathcal{F}$ of load buses with fixed shunt var control, $\mathcal{E}$ of load buses eligible for shunt var control, $\mathcal{B}$ of branches (lines and transformers), and $\mathcal{T}$ of

TABLE I
SIZES OF THE POWER SYSTEMS AND THE NONLINEAR OPF PROBLEM (4)

Test System	$\|\mathcal{N}\|$	$\|\mathcal{G}\|$	$\|\mathcal{E}\|$	$\|\mathcal{B}\|$	$\|\mathcal{T}\|$	n	m	p
IEEE 30	30	6	5	41	4	75	60	46
IEEE 57	57	7	5	80	17	143	114	87
IEEE 118	118	54	12	186	9	311	236	194
IEEE 300	300	69	23	411	35	727	600	428
Actual 1211	1,211	312	0	1,567	553	3,165	2,422	1,955

TABLE II
SIZES OF THE TRUST REGION SUBPROBLEMS AND THE NEWTON SYSTEMS

Test System	Number of variables			Size of Z_k	Size of Newton system		
	v	d	u		(23)-V	(23)-H	(25)
IEEE 30	75	75	15	75×15	375	315	435
IEEE 57	143	143	29	143×29	715	601	829
IEEE 118	311	311	75	311×75	1,555	1,319	1,791
IEEE 300	727	727	127	727×127	3,635	3,035	4,235
Actual 1211	3,165	3,165	1,743	3,165×1,743	15,825	13,403	18,247

TABLE III
INITIAL LOADING AND LOSSES

Test System	MW Load	MVAr Load	MW Loss
IEEE 30	283.40	126.20	17.63
IEEE 57	1,250.80	336.40	27.86
IEEE 118	3,668.00	1,438.00	132.48
IEEE 300	23,246.87	7,787.97	408.32
Actual 1211	153,206.80	42,243.10	897.59

TABLE IV
FINAL LOSSES AND NUMBER OF ITERATIONS USING INITIALIZATION I

Test System	Number of Iterations			Minimum Active Losses	
	PDIP	PCIP	TRIP	(MW)	(Reduction %)
IEEE 30	10	7	2	17.79	-0.91
IEEE 57	11	8	3	24.34	12.63
IEEE 118	14	9	3	118.77	10.36
IEEE 300	15	9	4	378.37	7.33
Actual 1211	18	14	6	881.32	1.81

TABLE V
NUMBER OF ITERATIONS BY TRIP ALGORITHM USING INITIALIZATION I

Test System	Δ_0	Number of Iterations			Average Iterations	
		TRIP	IP-Vert	IP-Horz	IPav-V	IPav-H
IEEE 30	1	2	12	12	6.0	6.0
IEEE 57	1	3	18	19	6.0	6.3
IEEE 118	1	3	19	22	6.3	7.3
IEEE 300	1	4	31	38	7.7	9.5
Actual 1211	2	6	36	81	6.0	13.5

transformers with LTCs. Table I shows the size of each set for the various test systems as well as the number of primal variables n, number of equality constraints m, and number of simple bound constraints p of the NLP problem (4).

All algorithms were fully implemented in MATLAB. The initial slack variables and Lagrange multipliers for the PDIP and PCIP algorithms are determined as in [22]. For comparison purposes, in all algorithms the initial x_0 is chosen in four different ways: (i) an initial power flow solution, (ii) a flat start (i.e., $V_i^0 = 1$ and $\theta_i^0 = 0$), (iii) the middle point between limits (e.g., $V_i^0 = (V_i^{\min} + V_i^{\max})/2$), and (iv) random points within limits generated by the MATLAB function `rand()`. The initializations (ii), (iii) and (iv) surely satisfy the bound constraints but violate the equality constraints. The initialization (i) satisfies the equality constraints but most likely violates bounds. However, note from (8c) that a starting point should not violate a variable limit by an amount larger than the trust region size Δ_k, otherwise the lower bound in (8c) will be greater than the upper bound resulting in an ill-posed trust region subproblem; hence, when an initial variable given by a power flow solution violates its limit, it must be reset to a value within limits. In summary, x_0 needs not satisfy the equality constraints but it must be within the interval $(\underline{x} - \Delta_k, \overline{x} + \Delta_k)$.

The initialization of the IP algorithm for QP when solving the vertical and horizontal subproblems is simpler, with $v_0 = 0$ and $d_0 = v_*$, where v_* is the optimal solution of the vertical subproblem solved ahead of the horizontal subproblem. The IP algorithms for QP and NLP use the same parameters: $\mu_0 = 0.01$, $\gamma = 0.9995$, $\sigma = 0.2$ and $\epsilon_1 = 10^{-4}$. The parameters of the trust region algorithm are: $\Delta_0 = 1$, $\bar{\Delta} = 5$, $\xi = 0.8$, $\eta_0 = 2$ and $\bar{\rho} = 0.1$. Table II shows the number of variables v in (10), variables d in (11), variables u in (13), the dimension of the orthogonal complement Z_k, the size of the Newton system (23) when solving either the vertical or the horizontal subproblem, and the size of the Newton system (25) when solving the horizontal subproblem without computing Z_k. Table III shows the total system active and reactive power loads, and the active power losses for the base condition, i.e., prior to the application of the optimization.

A. Comparisons

For initialization (i) (power flow solution), Table IV shows the number of iterations for the PDIP and PCIP algorithms, the number of *outer* iterations by the TRIP algorithm, and the minimum active losses after optimization. The operating points obtained by all algorithms are exactly the same, and all algorithms were successful in yielding a solution. The PCIP algorithm performs better than the PDIP algorithm, as expected. Note that the number of TRIP iterations is low (an average of 4 outer iterations per test system); however, because the cost (processing time) of a single trust region iteration is nearly 1.5 times the cost of a PDIP solution, the TRIP algorithm is nearly 6 times slower than the PDIP algorithm, which is to be expected.

Table V shows the solution details of the TRIP algorithm. The column Δ_0 is the initial trust region size, and the columns `IP-Vert` and `IP-Horz` are the total number of IP iterations for QP to solve the vertical and the horizontal subproblems, respectively, in all trust region iterations. The column `TRIP` is the number of trust region (outer) iterations; in each outer iteration one vertical and one horizontal subproblem is formed and solved. Also shown in Table IV are the average numbers of IP iterations to solve the vertical (`IPav-V`) and horizontal (`IPav-H`) subproblems. The average number of IP iterations is reasonably low, and from these numbers one can infer that the vertical subproblems are easier to solve than the horizontal subproblems. The IP algorithm for QP performed quite well solving both subproblems, with the Byrd-Omojokun technique ensuring the feasibility of the two subproblems.

For the IEEE 300-bus system, the solution by the nonlinear PDIP algorithm takes 18 iterations (see Table IV). For each IP iteration for NLP a linear system of size 4,235 (see Table II) is formed and solved. On the other hand, the solution by the

TABLE VI
NUMBER OF ITERATIONS USING INITIALIZATIONS (II) AND (III)

Test System	Number Iterations - (ii) PDIP	PCIP	TRIP	Number Iterations - (iii) PDIP	PCIP	TRIP
IEEE 30	11	9	3	11	9	3
IEEE 57	13	12	5	12	10	5
IEEE 118	14	11	4	13	9	4
IEEE 300	19	12	5	18	12	5
Actual 1211	Fail	Fail	6	Fail	Fail	9

TABLE VII
NUMBER OF ITERATIONS TRIP ALGORITHM USING INITIALIZATION (II)

Test System	Δ_0	Number of Iterations - (ii) TRIP	IP-Vert	IP-Horz	Average Iterations IPav-V	IPav-H
IEEE 30	1	3	18	18	6.0	6.0
IEEE 57	1	5	30	30	6.0	6.0
IEEE 118	1	4	31	31	7.7	7.7
IEEE 300	1	5	45	44	9.0	8.8
Actual 1211	2	6	55	86	9.2	14.3

TABLE VIII
NUMBER OF ITERATIONS TRIP ALGORITHM USING INITIALIZATION (III)

Test System	Δ_0	Number of Iterations - (iii) TRIP	IP-Vert	IP-Horz	Average Iterations IPav-V	IPav-H
IEEE 30	1	3	18	18	6.0	6.0
IEEE 57	1	5	33	31	6.6	6.2
IEEE 118	1	4	31	30	7.7	7.5
IEEE 300	1	5	45	44	9.0	8.8
Actual 1211	2	9	74	106	8.2	12.9

TRIP algorithm takes 4 outer iterations, which means that 4 vertical and 4 horizontal subproblems are formed and solved, requiring a total of 31 IP iterations for QP to solve the 4 vertical subproblems, and 38 IP iterations for QP to solve the 4 horizontal subproblems (see Table V). Linear systems of size 3,635 are formed and solved in each of the 31 IP iterations to solve the vertical subproblems, and of size 4,235 in each of the 38 IP iterations to solve the horizontal subproblems. Clearly, the trust region algorithm is slower than the direct nonlinear IP algorithm. However, the trust region OPF algorithm is not designed to compete in speed with popular OPF solvers, since its main characteristic is convergence robustness.

To test the performance of the TRIP algorithm for a large system an actual 1211-bus system is used. The optimization of this system is challenging because voltage magnitudes in some buses are very low, the voltage limit ranges are tight, and no shunt compensation is considered in the optimization. All algorithms converge when the starting point is an initial power flow solution (see Table IV), but the PDIP and PCIP algorithms fail to converge when using a flat start or a middle point initialization (see Table VI). Poor centrality of the IP iterates results in step lengths too close to zero early in the iterations, leading to convergence failure. The TRIP algorithm converged in all instances, and the solution is further detailed in Table VII (flat start) and Table VIII (middle point). The number of outer iterations for the TRIP algorithm is similar for the three initializations (6 to 9 iterations). The most noticeable change is the number of IP iterations to solve the vertical subproblem in the initial trust region iterations when x_0 is far from the optimal solution (see Table IX), e.g., a flat start. This is probably due to the fact that the objective of the vertical subproblem is to minimize infeasibility (squared residuals) of the equality constraints, and the higher the initial infeasibility, the higher the number of IP iterations to solve the vertical subproblem. However, after 2 or 3 outer iterations the equality constraints are nearly satisfied and the number of IP iterations to solve the vertical subproblem becomes very small. The performance of the algorithms using random initial points is discussed in Section IV-B.

The convergence of the TRIP algorithm was studied when the Lagrange multiplier vector λ_k needed to evaluate H_k is set to the least-squares estimate (26), as usually done in the literature, or to the optimal multiplier τ_* from the horizontal subproblem (11), as proposed in this paper. The computation

TABLE IX
CONVERGENCE PROCESS FOR IEEE 300 BUS USING FLAT START

k	IP iterations Vert	Horz	Infeasibility Residuals Primal	Dual ($\approx$)	ar/pr	Δ_k	η_k	$f(x_k)$
0	-	-	8.64x10^{-1}	4.95x10^{-4}	-	1	2	8.20
1	13	9	1.17x10^{-1}	5.43x10^{-2}	0.967	5	2	7.38
2	14	9	2.81x10^{-2}	1.74x10^{-2}	0.964	5	2	4.38
3	6	8	4.13x10^{-3}	2.17x10^{-2}	0.903	5	2	4.25
4	6	9	7.92x10^{-4}	2.22x10^{-2}	0.706	5	2	4.26
5	6	9	2.15x10^{-5}	2.22x10^{-2}	0.978	5	2	4.26

TABLE X
IP CONVERGENCE, VERTICAL SUBPROBLEM, THE FIRST TRUST REGION ITERATION, IEEE 300 BUS

Iter	α_k	Infeasibility Residuals Primal	Dual	Complem.	Objective in (10a)
0	-	0.0000	6.53x10^{+3}	1.45x10^{+1}	1.79x10^{+3}
1	0.183	0.0000	5.21x10^{+3}	5.13x10^{+0}	1.20x10^{+3}
2	0.558	0.0000	2.09x10^{+3}	1.15x10^{+0}	2.34x10^{+2}
3	0.413	1.88x10^{-4}	1.00x10^{+3}	6.24x10^{-1}	8.08x10^{+1}
4	0.303	1.48x10^{-4}	3.28x10^{+2}	4.42x10^{-1}	3.97x10^{+1}
10	0.746	3.84x10^{-8}	4.93x10^{-2}	6.32x10^{-4}	5.74x10^{-1}
11	0.827	6.17x10^{-9}	8.51x10^{-3}	1.31x10^{-4}	5.73x10^{-1}
12	0.880	0.00x10^{+0}	1.02x10^{-3}	2.20x10^{-5}	5.73x10^{-1}
13	1.000	0.00x10^{+0}	9.04x10^{-9}	3.15x10^{-6}	5.73x10^{-1}

TABLE XI
IP CONVERGENCE, HORIZONTAL SUBPROBLEM, THE FIRST TRUST REGION ITERATION, IEEE 300 BUS

Iter	α_k	Infeasibility Residuals Primal	Dual	Complem.	Objective in (11a)
0	-	0.00x10^{+00}	8.20x10^{-2}	1.34x10^{+0}	1.19x10^{+1}
1	0.835	5.69x10^{-03}	1.47x10^{-2}	3.48x10^{-1}	1.18x10^{+1}
2	0.602	2.42x10^{-03}	5.86x10^{-3}	1.59x10^{-1}	1.18x10^{+1}
3	0.900	1.12x10^{-14}	5.86x10^{-4}	3.16x10^{-2}	1.18x10^{+1}
4	0.847	7.34x10^{-15}	9.23x10^{-5}	6.27x10^{-3}	1.18x10^{+1}
6	0.707	5.25x10^{-15}	7.70x10^{-6}	8.48x10^{-4}	1.18x10^{+1}
7	0.971	6.42x10^{-15}	2.27x10^{-7}	8.21x10^{-5}	1.18x10^{+1}
8	0.838	1.06x10^{-14}	3.67x10^{-8}	2.23x10^{-5}	1.18x10^{+1}
9	0.906	9.71x10^{-11}	3.47x10^{-9}	2.16x10^{-6}	1.18x10^{+1}

TABLE XII
COMPARISON OF LAGRANGE MULTIPLIER ESTIMATES, IEEE 300-BUS

Bus Number Largest	Smallest	PDIP for NLP Largest	Smallest	Estimate by (26) Largest	Smallest	Estimate by τ_* Largest	Smallest
528	176	1.2948	0.9360	0.8438	0.6511	1.2948	0.9360
70	7166	1.2805	0.9616	0.8426	0.6745	1.2805	0.9616
112	166	1.2288	0.9616	0.8385	0.6745	1.2288	0.9616
114	165	1.2240	0.9635	0.8426	0.6761	1.2240	0.9636
204	7023	1.2218	0.9758	0.8438	0.6784	1.2217	0.9758
$\Vert\lambda_k\Vert$		18.9477		12.9563		18.9477	

of λ_k by (29) is not tested due to its cost. For the cases tested, the OPF solutions obtained were the same, notwithstanding different sequences of iterates $\{x_k\}$. This result is expected, since the trust region approach still should work when H_k is only a quasi-Newton approximation of the actual Lagrange Hessian. Nevertheless, the higher the number of binding inequalities in the optimal solution is, the less accurate the least-squares estimate is (26), since the terms $\widetilde{\pi}_L$ and $\widetilde{v}_U$ present in (29) and ignored in (26) become more relevant. This happens, for instance, with the IEEE 300-bus system. For this system, Table XII shows the five largest and five smallest λ_i's associated with the active power balance constraints, as computed by the nonlinear PDIP algorithm (assumed as reference values), and the λ_i's for the same constraints as obtained by the least-squares estimate (26) and by the optimal solution of the horizontal subproblem (11); also shown are the norms of the Lagrange multiplier vectors provided by the three approaches. Clearly, the most accurate estimate for λ_k is the one provided by the horizontal subproblem solution τ_*.

B. Convergence Characteristics and Robustness

Table IX illustrates the convergence processes for the IEEE 300-bus system. The column $f(x_k)$ is the nonlinear objective function in (4); `Vert` and `Horz` are the number of IP iterations to solve the vertical and horizontal subproblems, respectively, in each outer iteration; `Primal` and `Dual` are the primal and dual infeasibilities, respectively; `ar/pr` is the reduction ratio $\mathrm{ar}(d_k)/\mathrm{pr}(d_k)$; Δ_k is the trust region size; and η_k is the penalty parameter in the merit function. Note that the objective function may increase in some iterations, since the merit function makes a trade-off between reducing the objective function and satisfying the constraints, and infeasible points during the iterates may provide lower objective values. During the iterations, the constraints residual norm has a penalty η_k forcing the algorithm to find a feasible point, at the expense of degrading the objective function value. The size of the trust region usually increases during the iterations; this should be expected for most of the cases, since close to the solution the generated points are more likely to be feasible and the approximation model to be trusted. The dual infeasibility in the outer iterations (shown in Tables IX and XIV) is only an approximation, since the Lagrange multipliers from the bound constraints of the nonlinear OPF problem are not computed by the trust region approach; the displayed dual values take into account only the equality constraints yet using a Lagrange multiplier estimate.

The number of IP iterations to solve the vertical and the horizontal subproblems are nearly constant, changing very little during the outer iterations. The highest number of IP iterations usually occurs in the solution of the vertical subproblem in the initial outer iterations when the starting point x_0 is far from the optimal solution. Tables X and XI demonstrate the convergence processes of the IP solver applied to the first vertical and horizontal subproblems in the optimization of IEEE 300-bus system. In these tables, the column α_k refers to the (common) primal and dual step sizes; columns `Primal`, `Dual` and `Complem` refer to the primal infeasibility, dual infeasibility and complementarity residual, respectively; and column `Objective` displays the objective function values (including the constant terms dropped from (10) and (11)).

The primal infeasibility shown in Table X is always zero because vertical subproblem (10) has no equality constraint, unlike horizontal subproblem (11), and the bound constraints are strictly satisfied during the IP iterations. Note that the complementarity residual is the optimality condition that takes longer to converge; thus, the number of IP iterations may be reduced if the barrier parameter is more sharply reduced. Recall that the objective function from the vertical subproblem is the Euclidean norm of the linear equality constraints from the trust region subproblem (8), and note that in trust region iterations 1 and 2 this objective is nonzero, which means that the feasible set of the trust region subproblem (8) in iterations 1 and 2 is empty. This example demonstrates the effectiveness of the Byrd-Omorjokun technique to overcome inconsistency in the constraints of the approximation model.

One can conclude, from the previous discussion, that OPF infeasibility may be detected from the vertical subproblem solution, because the vertical subproblem objective will never be zero when the OPF problem (4) is infeasible, and, when (4) is feasible, the vertical objective becomes zero as soon as the linearized equality constraints are satisfied. Thus, infeasibility may be detected by monitoring the vertical subproblem solutions. Furthermore, one can argue that the cost of a trust region iteration may be cut by nearly half by not solving the vertical subproblem after its objective becomes zero, since from that iteration on, the linearized equality constraints are compatible with the trust region constraint, and thus the full step d_k can be obtained by directly solving the trust region subproblem.

To assess the robustness of the trust region OPF algorithm

TABLE XIII
NUMBER OF ITERATIONS AND FINAL LOSSES: INCREASED LOAD

Test System	Total Number of Iterations			Minimum Active Losses	
	TRIP	IP-Vert	IP-Horz	(MW)	(Reduction %)
IEEE 30	3	18	20	27.74	-0.91
IEEE 57	4	24	28	57.88	7.41
IEEE 118	4	31	30	137.97	9.52
IEEE 300	4	28	31	499.04	8.51

TABLE XIV
CONVERGENCE PROCESS FOR IEEE 300 BUS: INCREASED LOAD

k	IP iterations		Infeasibility Residuals		ar/pr	Δ_k	η_k	$f(x_k)$
	Vert	Horz	Primal	Dual (≈)				
0	-	-	5.34x10^{-1}	4.77x10^{-3}	-	2	10	4.56
1	9	9	1.31x10^{-2}	4.24x10^{-2}	0.974	5	10	6.19
2	8	7	1.22x10^{-2}	4.25x10^{-2}	0.452	5	10	10.3
3	6	7	5.24x10^{-4}	7.96x10^{-2}	0.938	5	10	11.0
4	5	8	7.12x10^{-6}	8.24x10^{-2}	1.000	5	10	11.0

TABLE XV
FIFTY (50) RUNS OF TRIP ALGORITHM USING RANDOM STARTING POINTS

Test System	Number of Outer Iterations				Initial Point Infeasibility		
	Total	Avg	Max	Min	Max-v	Min-v	Avg-v
IEEE 30	206	4.12	6	3	12.7	4.7	8.4
IEEE 57	264	5.28	9	3	20.0	12.7	15.4
IEEE 118	208	4.16	5	4	72.4	54.2	61.9
IEEE 300	273	5.46	9	4	524.2	154.7	307.1

two sets of simulations are carried out. One regards the choice of the initial point, where a set of 50 random starting points is generated for each test system, and the other regards the shape/size of the feasible set. For the latter case, the loading of the system is increased to stress the system [25], so that the solutions of the OPF problems become more difficult to obtain with the specified controls and operating limits. To generate difficult but guaranteed to be feasible cases, the maximum loading of the systems satisfying the specified operating limits is obtained by solving the maximum loading OPF problem [25], where a loading parameter is maximized with the power balance constraints set as:

$$\begin{aligned} P_i(\theta,V,t) - P_{G_i} + (1+\sigma)P_{D_i} = 0, \quad & i \in \mathcal{N} \\ Q_i(\theta,V,t) - Q_{G_i} + (1+\sigma)Q_{D_i} = 0, \quad & i \in \mathcal{G} \\ Q_i(\theta,V,t) - Q_{G_i} + (1+\sigma)Q_{D_i} = 0, \quad & i \in \mathcal{F} \\ Q_i(\theta,V,t) + (1+\sigma)Q_{D_i} + b_i^{\text{sh}}V_i^2 = 0, \quad & i \in \mathcal{E} \end{aligned} \tag{41}$$

After computing the maximum loading parameter σ_*, the power system load is set to a point close to the maximum loading, i.e., $(1+0.9\sigma_*)P_{D_i}$ and $(1+0.9\sigma_*)Q_{D_i}$, so that the system is now significantly stressed, which would force the associated active loss minimization solution process to face convergence problems due to the reduction of the feasible set and some inequalities becoming strongly active. The active power losses (before the minimization) are 27.49 MW, 62.51 MW, 152.49 MW and 545.43 MW for IEEE 30-, 57-, 118- and 300-bus systems, respectively. The trust region OPF algorithm converged in all test runs; Table XIII displays the corresponding number of iterations for convergence and the final losses. The convergence process for the IEEE 300-bus test system is shown in Table XIV. Note that the number of trust region iterations has not increased with the complexity of the OPF problems.

Table XV displays the performance of the TRIP algorithm using random starting points x_0. The column `Total` is the total number of trust region iterations in the set of 50 test runs; `avg` is the average number of trust region iterations; `max` and `min` are the maximum and minimum number of trust region iterations observed in a single test run out of the 50 runs for each system; `max-v`, `min-v` and `avg-v` are the maximum, minimum and average objective values, respectively, of the vertical subproblem in the first trust region iteration. Note that the number of trust region iterations changes very little, for the same test system and among the different test systems. Also notice from the initial point infeasibility (`max-v` and `min-v`) that the random starting points range from medium to highly infeasible initial trust region subproblems. Indeed, the infeasibility of x_0 is higher than shown in Table XV, since the constraints residuals shown in this table are the minimal ones after performing the first trust region iteration.

The performance (accuracy versus number of iterations) of the trust region algorithm when using very tight convergence tolerance has been analyzed. Table XVI displays, for each IEEE test system and for each trust region iteration, the primal feasibility residual

$$\max\{\|g(x_k)\|_\infty,\ \max_i\{x_i^k - \overline{x}_i\},\ \max_i\{\underline{x}_i - x_i^k\}\}.$$

TABLE XVI
OUTER ITERATIONS VS. PRIMAL INFEASIBILITY: INITIALIZATION (I)

Iter.	Primal Infeasibility			
	IEEE 30	IEEE 57	IEEE 118	IEEE 300
0	5.132x10^{-02}	1.356x10^{-01}	2.036x10^{-01}	5.333x10^{-01}
1	6.094x10^{-03}	3.165x10^{-03}	1.449x10^{-02}	1.766x10^{-02}
2	3.447x10^{-05}	5.875x10^{-04}	4.095x10^{-04}	2.566x10^{-03}
3	1.477x10^{-09}	2.608x10^{-09}	2.802x10^{-07}	4.860x10^{-04}
4	-	-	2.215x10^{-13}	2.936x10^{-06}
5	-	-	-	6.414x10^{-10}

Note from Table V (for $\epsilon = 10^{-4}$, which is a typical tolerance for OPF applications) and Table XVI that, for the test runs performed, the accuracy of the solution almost doubles at the cost of one additional trust region iteration. Thus, the trust region algorithm handles tight convergence tolerances with little increase in the number of iterations.

C. Parameter Selection

According to [4], trust region algorithms in general are not very sensitive to the choice of their parameters. Reference [26] presents a detailed analysis of the sensitivity of trust region algorithms for unconstrained optimization to their parameters, involving 3,940 combinations of parameters (testing 4 parameters) which resulted in a grand total of 95,040 test runs. It is concluded that commonly used “standard” values for the parameters appear not to be the best choice, and alternative values are suggested. The most expressive finding is that the parameter $\bar{\rho}$ (borderline of $\text{ar}(d_k)/\text{pr}(d_k)$ to accept or discard the computed step d_k) should be set to 0.001 instead of the usual value of 0.1.

In this paper, parameters are chosen based on standard values reported in the literature (e.g., [4], [7]) and on characteristics of the OPF problem. For example, trust region constraint (8c) suggests that the initial trust region Δ_0 can be chosen based on the ranges of the variables. As the trust region algorithm suitably can increase or decrease Δ_k in each iteration, according to the ratio $\text{ar}(d_k)/\text{pr}(d_k)$, the choice of Δ_0 is not a critical issue. With regard to η_0, in theory, η should be larger than the largest absolute value of the Lagrange multipliers, i.e., $\eta > \max_i |\lambda_i|$. Given that when minimizing losses the multipliers associated with active power balance constraints are close to unity, and those associated with reactive power balance constraints are close to zero, $\eta_0 = 2$ is chosen. Observe that the trust region algorithm updates this parameter during the iterative procedure.

The contraction factor ξ (typically, $\xi = 0.8$), the lower ξ is, the tighter the trust region constraint $\|v\|_\infty \leq \xi\Delta_k$ is in the vertical subproblem, and the higher the optimal objective $\|g(x_k) + \nabla g(x_k)v_*\|^2$ shall be. By considering a tighter trust region in the vertical subproblem ($\xi\Delta_k$), one allows more room ($(1-\xi)\Delta_k$) to reduce the model objective function in the horizontal subproblem. Thus, the role of the parameter ξ is to balance satisfying the linearized equality constraints against minimizing the model objective. Test runs performed with $\xi \in \{0.7,\ 0.75,\ 0.8,\ 0.85,\ 0.9,\ 0.95\}$ and remaining parameters using standard values showed no impact on the number of outer iterations. However, $\xi = 0.9$ resulted in a smoother sequence of the ratios $\text{ar}(d_k)/\text{pr}(d_k)$.

The parameter $\bar{\rho} = 0.1$ is also typical in the literature and thus it was used here. This is probably the parameter that influences the most the performance of the algorithm, as it is used in deciding whether the computed step d_k should be accepted or not and, consequently, whether the trust region size should be increased or reduced. In [26], the authors suggest a smaller value for this parameter, but in the context of unconstrained optimization; however, for the test runs presented here, the ratios $\text{ar}(d_k)/\text{pr}(d_k)$ are all greater than 0.1, and thus choosing $\bar{\rho} < 0.1$ would have no effect on the results presented. A detailed analysis of this parameter in combination with all other parameters involves the consideration of a rather large number of sets of parameters and test runs (hundreds of thousands), which is beyond the scope of the present paper.

V. Conclusions

This paper has presented in detail an application of the Byrd-Omojokun trust region technique to solve nonlinear OPF problems. The proposed OPF algorithms are shown to be globally convergent and robust, as it is based on the Byrd-Omojokun technique which is efficient in dealing with inconsistent constraints that may appear in trust region subproblems. In the numerical tests performed, considering highly nonlinear OPF problems and random starting points, the proposed technique demonstrated the expected convergence robustness.

The proposed method is roughly 6 times slower than the PDIP algorithm. However, its main application is to obtain OPF solutions when standard IP methods fail. Currently, a number of issues to improve the performance of the proposed algorithm are being studied, in particular, considering new merit functions, new rules to update the trust region radius and using fast approximate solutions of the vertical and horizontal subproblems.

References

[1] H. Wang, C. E. Murillo-Sánchez, R. D. Zimmerman, and R. J. Thomas, "On computational issues of market-based optimal power flow," *IEEE Trans. on Power Systems*, vol. 22, pp. 1185–1193, Aug. 2007.

[2] W. Rosehart, C. Roman, and A. Schellenberg, "Optimal power flow with complementarity constraints," *IEEE Trans. on Power Systems*, vol. 20, pp. 813–822, May 2005.

[3] S. K. M. Kodsi and C. A. Cañizares, "Application of a stability-constrained optimal power flow to tunning of oscillation controls in competitive electricity markets," *IEEE Trans. on Power Systems*, vol. 22, pp. 1944–1954, Nov. 2007.

[4] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer, 1999.

[5] A. R. Conn, N. I. M. Gould, and P. L. Toint, *Trust-Region Methods*. Philadelphia: MPS-SIAM Series on Optimization, 2000.

[6] T. D. Plantenga, *Large-scale nonlinear constrained optimization using trust regions*. PhD thesis, Northwestern University, Evanston, IL, USA, 1994.

[7] M. Lalee, J. Nocedal, and T. Plantenga, "On the implementation of an algorithm for large-scale equality constrained optimization," *SIAM J. on Optimization*, vol. 8, no. 3, pp. 682–706, 1998.

[8] R. H. Byrd, "Robust trust region methods for constrained optimization," in *Proceedings of Third SIAM Conference on Optimization*, (Houston, TX), may 1987.

[9] O. Omojokun, *Trust Region Algorithms for Optimization with Nonlinear Equality and Inequality Constraints*. PhD thesis, Dept. of Computer Science, University of Colorado, Boulder, Colorado, 1989.

[10] X. Liu and Y. Yuan, "A robust trust region algorithm for solving general nonlinear programming," *Journal of Computational Mathematics*, vol. 19, pp. 309–322, 2001.

[11] J. Zhang, "Robust trust region method for general constrained optimization," *Mathematical Methods of Operations Research*, vol. 60, pp. 73–85, Sept. 2004.

[12] S. Pajic and K. A. Clements, "Globally convergent state estimation via the trust region method," in *2003 IEEE Bologna PowerTech Conference*, (Bologna, Italy), June 2003.

[13] S. Pajic and K. A. Clements, "Power system state estimation via globally convergent methods," *IEEE Trans. on Power Systems*, vol. 20, pp. 1683–1689, Nov. 2005.

[14] A. S. Costa, R. Salgado, and P. Haas, "Globally convergent state estimation based on Givens rotations," in *Proc. of the 2007 IREP Symposium*, (Charleston SC, USA), Aug. 2007.

[15] R. Zhou, Y. Zhang, and H. Yang, "A trust-region algorithm based on global SQP for reactive power optimization," in *2nd International Conference on Electrical and Electronics Engineering and XI Conference on Electrical Engineering*, (Mexico City, Mexico), Sept. 2005.

[16] A. A. Sousa and G. L. Torres, "Globally convergent optimal power flow by trust-region interior-point methods," in *IEEE PowerTech 2007*, (Lausanne, Switzerland), pp. 1–6, July 2007.

[17] S. J. Wright, *Primal-Dual Interior-Point Methods*. SIAM, 1997.

[18] J. J. Moré, "Recent developments in algorithms and software for trust region methods," in *Mathematical Programming: The State of the Art*, pp. 258–287, Springer Verlag, 1983.

[19] D. C. Sorensen, "Newton's method with a model trust region modification," *SIAM J. on Numerical Analysis*, vol. 19, pp. 409–426, 1982.

[20] Y. Wu, A. S. Debs, and R. E. Marsten, "A direct nonlinear predictor-corrector primal-dual interior point algorithm for optimal power flow," *IEEE Trans. on Power Systems*, vol. 9, pp. 876–883, May 1994.

[21] S. Granville, "Optimal reactive dispatch through interior point methods," *IEEE Trans. on Power Systems*, vol. 9, pp. 136–146, Feb. 1994.

[22] G. L. Torres and V. H. Quintana, "An interior point method for nonlinear optimal power flow using voltage rectangular coordinates," *IEEE Trans. on Power Systems*, vol. 13, pp. 1211–1218, Nov. 1998.

[23] F. A. M. Gomes, M. C. Maciel, and J. M. Martínez, "Nonlinear programming algorithms using trust regions and augmented lagrangians with nonmonotone penalty parameters," *Mathematical Programming*, vol. 84, no. 1, pp. 161–200, 1999.

[24] N. Maratos, *Exact Penalty Function Algorithms for Finite Dimensional and Control Optimization Problems*. PhD thesis, University of London, London, UK, 1978.

[25] R. J. Avalos, C. A. Cañizares, F. Milano, and A. Conejo, "Equivalency of continuation and optimization methods to determine saddle-node and limit-induced bifurcations in power systems," *IEEE Trans. on Circuit and Systems I*, vol. 56, no. 1, pp. 201–223, Jan. 2009.

[26] N. I. M. Gould, D. Orban, A. Sartenaer, and P. L. Toint, "Sensitivity of trust-region algorithms to their parameters," *4OR A Quarterly Journal of Operations Research*, vol. 3, pp. 227–241, 2005.

Andréa A. Sousa received the B.Eng. (1999), M.Sc. (2005) and D.Sc. (2008) degrees, all in Electrical Engineering, from the Universidade Federal da Paraíba (Brazil), the Universidade Federal de Campina Grande (Brazil) and the Universidade Federal de Pernambuco (Brazil), respectively.

Geraldo L. Torres received the B.Eng. (1987), M.Sc. (1991) and Ph.D. (1998) degrees, all in Electrical Engineering, from the Universidade de Pernambuco (Brazil), the Universidade Federal de Pernambuco (Brazil) and the University of Waterloo (Canada), respectively. Since 1992 he has been with the Universidade Federal de Pernambuco, where he is a Full Professor. His research concentrates mainly on optimization applications to power systems.

Claudio A. Cañizares (S85, M91, SM00, F07) received in April 1984 the Electrical Engineer diploma from the Escuela Politécnica Nacional (EPN), Quito-Ecuador, where he held different teaching and administrative positions from 1983 to 1993. His MS (1988) and PhD (1991) degrees in Electrical Engineering are from the University of Wisconsin-Madison. Dr. Cañizares has held various academic and administrative positions at the E&CE Department of the University of Waterloo since 1993, where he is currently a Full Professor. His research activities concentrate on the study of modeling, simulation, control, stability and computational issues in power systems in the context of competitive electricity markets.