Distributed Model Predictive Volt/Var Optimization of Active Distribution Networks

I. INTRODUCTION

Model based volt/var optimization is envisioned as the major objective of smart distribution network to facilitate higher integration of renewables [1]. Volt/var optimization has the primary objective of economic operation (minimize loss or minimize power purchase cost) of network while complying with the technical requirements ensuring iota of voltage and line current violations. The volt/var optimization framework proposed in [1] helps to optimally coordinate the traditional voltage controllers such as OLTC or switched capacitors, and trending resources from Distributed Energy Resources (DERs) by taking advantage of the communication infrastructure.

Traditionally volt/var optimization is performed in a centralized architecture, where the distribution system operator (DSO) monitored and controlled all the DERs [2], [3]. Additionally, the centralized controller also had to ensure secure operation of critical resources like storage systems for prolonged life-time. With the growing interconnection of DERs from private sectors, the centralized architecture is going to be computationally burdensome and unscalable [4]. As it is highly dependent on the communication from all the resources to a central optimization entity, the reliability of the centralized architecture is weakened. Moreover, in many cases, the private DERs would be reluctant to share all their resource information, other than the trading variables. Considering these issues in centralized architecture such as computation inefficiency, less communication reliability and information insecurity, distributed optimization could be a potential alternative for volt/var optimization.

Volt/var optimization is a NP-hard problem which involves both integer (e.g., from tap changers or switching capacitor) and real (e.g, from inverter based DERs) optimization variables. The nonconvex formulation of this problem can be efficiently solved to find a global minimum by utilizing Mixed Integer Quadratic Programming (MIQP) solvers such as Gurobi, MOSEK, etc.

Why ADMM?

However, when the volt/var optimization is implemented in distributed architecture utilizing Alternative Direction of Method of Multipliers (ADMM), there occurs a convergence problem mainly due to integer variables [5].

Elaborate more

Consistently, the previous literature works [6], [**?**] on distributed optimization of power networks have excluded the discrete control resources to avoid integer variables in their problem formulations.

The distributed optimization proposed in this study allows to include discrete control resources in the distributed volt/var optimization while improving the convergence of ADMM. It is achieved firstly by relaxing the integer variable to continuous variables, secondly applying ADMM for resulting optimization problem without integer/discrete variable, and finally applying branch-and-bound method to fine tune the optimal solution including integer variables.

The proposed distributed optimization architecture for volt/var optimization allows any number of DERs to trade their resources with the DSO, as shown in Fig. **??**. Due to local optimization initially, the DSO does not have any information of the DERs' capability however, it simply requests for purchase of certain amount of resources. And, the DERs responds to the DSO with the quantity of resources they can contribute. This process of resource arbitrage finally reaches a consensus among the DERs and DSO in a few iterations. The DSO also has its local resources however, purchases the resources from DERs for techno-economic operation of the network. This study has tested the proposed distributed optimization to achieve volt/var optimization of Cigre MV distribution network, which has tap changers and switched capacitor banks as its local control resources. The PVs and Battery Storage Units (BSU) are considered as the participating DERs. The objective of the DSO is to minimize the power purchase cost from the wholesale market, whereas PV and BSU have the objective to minimize the PV curtailment cost and battery degradation cost. The proposed optimization problem is modeled in Pyomo [7], whereas the load-flow computation is carried on PandaPower[8]. The following case studies were conducted in this study.

- Firstly, the centralized based volt/var optimization is formulated and its solutions are used for benchmarking the proposed distributed optimization technique.
- The convergence speed of the proposed distributed optimization technique is compared with and without integer relaxation and Branch-and-Bound method.
- The impact of communication link failure is also analyzed to determine the reliability of the proposed technique.

II. MODELING OF ACTIVE DISTRIBUTION NETWORK

A. PV source

A power injection model is used for modeling PV sources. All the PV sources are assumed to be curtailable. The average power of PV sources along the forecasting horizon ($\mathcal{T} = [t, t+1, t+2, \ldots, t+N_p]^T$) are predicted by point predictors. In this regard, the power injection from a PV source at j^{th} bus is constrained by:

$$0 \leq p_j^t \leq \hat{p}_j^t, \quad \forall j \in \mathcal{I}^{PV}, t \in \mathcal{T} \tag{1}$$

where $\hat{p}_j(t)$ is the point prediction of PV. The uncertainty of prediction is neglected in this paper, assuming it is fairly small for short-term prediction at 5-15 minute time-steps [**?**]. Additionally, the PV sources are obliged to have a reactive power margin equivalent to 44% of its rated inverter capacity (S_j^{rat}) all the time, as per DER integration rule "IEEE 1547-2018" [9]. Following this rule, the reactive power from PV inverters limited by following constraint.

$$-0.44S_j^{rat} \leq q_j^t \leq 0.44S_j^{rat}, \quad \forall j \in \mathcal{I}^{PV}, t \in \mathcal{T} \quad (2)$$

B. Battery Energy storage system

The battery energy storage system (BESS) mainly comprises of battery and inverter system for controlled charging/discharging. The battery is characterized by state of charge (SOC), which defines the percentage of energy available in it. For the sake of convex model, the paper neglects the small losses in the charging/discharging process as in [10]. In this regard, the SOC dynamic of j^{th} BESS is expressed as:

$$SoC_j^t = SoC_j^{t-1} + \frac{\Delta t\, p_j^t}{B_{cap}} \quad \forall j \in \mathcal{I}^{ES}, t \in \mathcal{T} \quad (3)$$

where B_{cap} is the battery capacity and p_j^t is the power exchanged by BESS at time t. The operating constraints for (p_j^t, q_j^t, SoC_j^t) of ESS are listed below:

$$-P_j^{rat} \leq p_j^t \leq P_j^{rat} \quad (4)$$
$$-0.443S_j^{rat} \leq q_j^t \leq 0.44S_j^{rat} \quad (5)$$
$$SOC_j^{min} \leq SOC_j^t \leq SOC_j^{max} \quad (6)$$

where P_j^{rat} is the power rating of BESS which limits the maximum and minimum value of power exchanged by it. The constraint (5) comes from DER interconnection rule [9] which imposes 44% of reactive power margin from inverter rating. In addition, the battery SOC is limited by lower and upper limits for its safe operation as in (6).

The ageing process of battery energy storage is highly nonlinear [11]. However, for operational optimization, the degradation cost can be modelled fairly in quadratic form by only considering the significant loss factor, which is due to charging/discharging, as in [12] as:

$$J_j^t = \mu_j^2(SoC_j^t - SoC_j^{t-1})^2 + \sigma_j^2(SoC_j^t - SoC_j^{ref})^2 \quad (7)$$

where, μ_j and σ_j are the cost of degradation related to charging/discharging and high/low SoC respectively.

C. Switched Capacitor Bank (SCB)

Let q_j^{step} and N_j^{max} be incremental Q-injection at rated voltage (V_{rat}) and maximum step of a SCB. At any step N_j, the Q-injection of SCB when the terminal voltage $\sqrt{v_j}$ is given by:

$$q_j^t = N_j^t\, q_{step}\frac{v_j^t}{V_{rat}^2} \quad \forall j \in \mathcal{I}^{SCB}, \quad \forall t \in \mathcal{T}, \quad (8)$$
$$0 \leq N_j^t \leq N_j^{max}. \quad (9)$$

Although Equations (17) and (18) mathematically models the SCB, they cannot be directly used in Mixed-Integer convex Programming (MIQP) due to quadratic equality expression in (17). One way to convexify (17) is to neglect its dependence on v_j as in [13]; however, the equation (17) is converted into set of mixed integer linear constraints as shown below.

Firstly, the integer variable N_j^t is expressed linear sum of binary variable b_i^t using binary expansion [14] as:

$$N_j^t = \sum_{i=0}^{(M_j-1)} 2^i b_{ij}^t. \quad (10)$$

Here, M_j is the number of binary variable required to express N_j^{max} (has integer value). The value of M_j is given byceil$\{log_2(N_j^{max})\}$. Replacing N_j in (17) using (10), we get:

$$q_j^t = \frac{q_{step}}{V_{rat}^2}\sum_{i=0}^{(M_j-1)} 2^i b_{ij}^t v_j^t. \quad (11)$$

The auxiliary variable $z_{ij}(t)$ is introduced such that, (11) can be written as linear sum of $z_{ij}(t)$, as:

$$q_j^t = \frac{q_{step}}{V_{rat}^2}\sum_{i=0}^{(M_j-1)} 2^i z_{ij}^t \quad (12)$$
$$z_{ij}^t = b_{ij}^t v_j^t \quad (13)$$

The expression (13) has the product of binary and continuous variable, which is non-convex and not accepted by MIQP solvers like gurobi 8.0. It can be replaced by set of linear mixed-integer constraint using the big-**M** method [15], as:

$$z_{ij}^t \leq b_{ij}^t \lceil v_j \rceil, \quad z_{ij}^t \leq v_j^t \quad (14)$$
$$z_{ij}^t \geq 0, \qquad z_{ij}^t \geq v_j^t - (1 - b_{ij}^t)\lceil v_j \rceil \quad (15)$$

where $\lceil v \rceil$ is the upper bound of voltage at a node. Using (18) and (10), the box constraints over the binary variables are written as:

$$0 \leq \sum_{i=0}^{(M_j-1)} 2^i b_{ij}^t \leq N_j^{max} \quad (16)$$

Finally the model of SCB suitable for MIQP is given by (12), and (14-16).

D. Switched Capacitor Bank

$$q_j^t = N_j^t\, q_{step} \quad \forall j \in \mathcal{I}^{SCB}, \quad \forall t \in \mathcal{T}, \quad (17)$$
$$0 \leq N_j^t \leq N_j^{max}. \quad (18)$$

E. Distribution Network

Let's define a radial distribution network by a connected graph $G = (\mathcal{I}^b, \mathcal{E})$, where each node in $\mathcal{I}^b$ represents a bus and each link in $\mathcal{E}$ represents a line. Such network can be modeled using Branch Flow Model (BFM) and solved exactly using using convex optimization [16]. The corresponding BFM is as follows:

$$p_j^t = \sum_{k:j\to k} P_{jk}^t - \sum_{i:i\to j} (P_{ij}^t - r_{ij}\ell_{ij}^t) + g_j v_j^t,\ \forall j \in \mathcal{I}^b \quad (19)$$

$$q_j^t = \sum_{k:j\to k} Q_{jk}^t - \sum_{i:i\to j} (Q_{ij}^t - \mathsf{x}_{ij}\ell_{ij}^t) + b_j v_j^t,\ \forall j \in \mathcal{I}^b \quad (20)$$

$$v_j^t = v_i^t - 2(r_{ij}P_{ij}^t + \mathsf{x}_{ij}Q_{ij}^t) + (r_{ij}^2 + \mathsf{x}_{ij}^2)\ell_{ij}^t,\ \forall (i,j) \in \mathcal{E} \quad (21)$$

$$\ell_{ij}^t \geq \frac{(P_{ij}^t)^2 + (Q_{ij}^t)^2}{v_i^t}, \quad 0 \leq \ell_{ij}^t \leq \lceil \ell_{ij} \rceil \quad \forall (i,j) \in \mathcal{E} \quad (22)$$

$$\lfloor v \rfloor \leq v_j^t \leq \lceil v \rceil, \quad \forall j \in \mathcal{I}^b, \quad \forall t \in \mathcal{T} \quad (23)$$

Here, i, j, and k are the indices for nodes in $\mathcal{I}^b$. $i : i \to j$ and $k : j \to k$ denote the set of links starting from i^th node to all its successor nodes and the set of links ending to k^{th} node from all its predecessor nodes. It is to be noted that P_{jk} and Q_{jk} represent the sending end power from j^{th} to k^{th} node.

F. Linearized DistFlow Model

Lets define a radial distribution network by a connected graph $G = (\mathcal{I}^b, \mathcal{E})$, where each node in $\mathcal{I}^b$ represents a bus and each link in $\mathcal{E}$ represents a line/transformer. A linearized Branch Flow Model (BFM) [17] of distribution network is used in this paper. The linearized BFM is as follows:

$$p_j^t = \sum_{k:j\to k} P_{jk}^t - \sum_{i:i\to j} P_{ij}^t + g_j v_j^t,\ \forall j \in \mathcal{I}^b \quad (24)$$

$$q_j^t = \sum_{k:j\to k} Q_{jk}^t - \sum_{i:i\to j} Q_{ij}^t + b_j v_j^t,\ \forall j \in \mathcal{I}^b \quad (25)$$

$$v_j^t = v_i^t - 2(r_{ij}P_{ij}^t + \mathsf{x}_{ij}Q_{ij}^t),\ \forall (i,j) \in \mathcal{E} \quad (26)$$

Here, i, j, and k are the indices for nodes in $\mathcal{I}^b$. $i : i \to j$ and $k : j \to k$ denote the set of links starting from i^th node to all its successor nodes and the set of links ending to k^{th} node from all its predecessor nodes. It is to be noted that P_{jk} and Q_{jk} represent the sending end power from j^{th} to k^{th} node. The operational constraints (27) and (28) are defined as per the German grid code [18]. The constraint (27) defines the range of reactive power exchange at the grid connection point and (28) defines the operating voltage limits of nodes.

$$Q_{jk}^t \leq P_{jk}^t tan(cos^{-1}(0.95)),\ j \in \mathcal{S}, k : j \to k \quad (27)$$

$$\lfloor v \rfloor \leq v_j^t \leq \lceil v \rceil, \quad \forall j \in \mathcal{I}^b, \quad \forall t \in \mathcal{T} \quad (28)$$

III. Problem Formulation

A distribution network with local and distributed control resources has been considered for the study in this paper. Control assets owned by DSO and prosumers/private-entities are defined as local and distributed resources, respectively. For demonstration of the proposed Distributed-MPC scheme, the paper considers SCB as local resources of DSO whereas PVs and BSUs as distributed resources. Firstly, a centralized control is formulated for VVO and then a distributed control scheme is designed based on the principle of ADMM. The associated MPC has same control and prediction horizon of length (N_p) throughout this paper and is denoted by $\mathcal{T} = [t+1, t+2, \ldots, t+N_p]$.

A. Centralized Controller

The centralized VVO is designed to minimize (a) power purchase from transmission network, (b) cost of control resources (c) PV power curtailment, (d) Battery degradation cost. All these cost are squared and lumped in a central objective function (29), which is minimized on a rolling horizon basis.

$$\mathbf{min} \sum_{t\in\mathcal{T}} \Big(\sum_{j\in\mathcal{I}^{PV}} f_j(\boldsymbol{x}_{pv,j}^t) + \sum_{j\in\mathcal{I}^{ES}} g_j(\boldsymbol{x}_{es,j}^t) + h(\boldsymbol{x}^t) \Big) \quad (29)$$

Here, $f(\cdot)$, $g(\cdot)$ and $h(\cdot)$ are cost function associated with PV power curtailment, degradation of BSUs, and (cost of power purchase and control resources) respectively, and are defined as:

$$f_j(\boldsymbol{x}_{pv,j}^t) = \eta_j^2(\bar{p}_j^t - p_j^t)^2(\Delta t)^2 \qquad \forall j \in \mathcal{I}^{PV}, \quad (30)$$

$$g_j(\boldsymbol{x}_{es,j}^t) = J_j^t \qquad \forall j \in \mathcal{I}^{ES}, \quad (31)$$

$$h(\boldsymbol{x}^t) = (\Delta t)^2 \Big(\sum_{(i,j)\in\mathcal{S}}^{k:j\to k} (\alpha^t P_{ij}^t)^2 + (\boldsymbol{x}^t)^\top \boldsymbol{R}\boldsymbol{x}^t \Big). \quad (32)$$

The J_j^t in (31) is expressed in (7). The cost function of control resources $((\boldsymbol{x}^t)^\top \boldsymbol{R}\boldsymbol{x}^t \Delta t^2)$ is expressed in quadratic form in (32) to ensure it has positive value despite the injection or absorption of resources. For the sake of consistency, all other cost functions have been expressed in quadratic form. The control input ($\boldsymbol{x}^t$) comprises of both local ($\boldsymbol{x}_{lr}^t$) and distributed resources ($\boldsymbol{x}_{dr}^t$), which are detailed below.

$$\boldsymbol{x}^t = [\boldsymbol{x}_{lr}^t, \boldsymbol{x}_{dr}^t]^\top, \boldsymbol{x}_{lr}^t = [q_j^t : j \in \mathcal{I}^{SC}], \boldsymbol{x}_{dr}^t = [\boldsymbol{x}_{pv}^t, \boldsymbol{x}_{es}^t]^\top \quad (33)$$

The control inputs ($\boldsymbol{x}_{pv}^t$, $\boldsymbol{x}_{es}^t$) are the vectorized form of inputs from all PVs and BSUs respectively. They are expressed as:

$$\boldsymbol{x}_{pv}^t = [\boldsymbol{x}_{pv,j}^t : j \in \mathcal{I}^{PV}]^\top \quad \text{where, } \boldsymbol{x}_{pv,j}^t = [p_j^t, q_j^t] \quad (34)$$

$$\boldsymbol{x}_{es}^t = [\boldsymbol{x}_{es,j}^t : j \in \mathcal{I}^{ES}]^\top \quad \text{where, } \boldsymbol{x}_{es,j}^t = [p_j^t, q_j^t] \quad (35)$$

The unit cost of control inputs are denoted by the diagonal matrix, which is $\boldsymbol{R} = diag\{\boldsymbol{R}_{lr}, \boldsymbol{R}_{dr}\}$. Along with $\boldsymbol{R}$, other unit costs/penalties such as α/ η are expressed in **€**/MWh or **€**/MVARh. Hence, the associated cost functions are multiplied by Δt as the resources are updated at every time-step (Δt) rather than on hourly basis. The operational constraints of PV, ES, and SCB are given in (1-2), (3-6), and (12-16) respectively. Furthermore, the network constraint are listed in (24-28). The formulated centralized optimal controller takes the form of Mixed-Integer Quadratic Programming (MIQP).

B. Centralized to Distributed Control

In this work, the centralized objective (29) and its constraints are segregated among PV owners, BSU owners and DSO, where each entity would only require to solve their own local optimization problem. However, the coupling of control inputs pertaining to objective function of PV ($f(\cdot)$) and BSU ($g(\cdot)$) with DSO's objective function ($h(\cdot)$) in (29) do not allow for direct segregation. Hence, for separation of control inputs of PVs and BSUs, their copied variable ($\boldsymbol{z}$) is introduced to form a new objective function as:

$$\textbf{min} \sum_{t\in\mathcal{T}} \Big(\sum_{j\in\mathcal{I}^{PV}} f_j(\boldsymbol{x}_{pv,j}^t) + \sum_{j\in\mathcal{I}^{ES}} g_j(\boldsymbol{x}_{es,j}^t) + h(\boldsymbol{x}_{lr}^t, \boldsymbol{z}_{pv}^t, \boldsymbol{z}_{\boldsymbol{es}}^{\boldsymbol{t}}) \Big) \tag{36}$$

$$\text{subject to: } \boldsymbol{x}_{pv,j}^t - \boldsymbol{z}_{pv,j}^t = 0 \quad \forall j \in \mathcal{I}^{PV},\ t \in \mathcal{T} \text{ and} \tag{37}$$

$$\boldsymbol{x}_{es,j}^t - \boldsymbol{z}_{es,j}^t = 0 \quad \forall j \in \mathcal{I}^{ES},\ t \in \mathcal{T}. \tag{38}$$

The constraints pertaining to PV owner, BSU owner, and DSO remain intact, which are defined in short as:

$$\Psi_{pv,j}(\boldsymbol{x}_{pv,j}^t) \leq 0 \qquad \forall j \in \mathcal{I}^{PV} \text{(PV constraints)} \tag{39}$$

$$\Psi_{es,j}(\boldsymbol{x}_{es,j}^t) \leq 0 \qquad \forall j \in \mathcal{I}^{ES} \text{(BSU constraints)} \tag{40}$$

$$\Phi(\boldsymbol{x}_{lr}^t, \boldsymbol{z}_{pv}^t, \boldsymbol{z}_{es}^t) \leq 0 \qquad \text{(DSO constraints)} \tag{41}$$

The equivalent centralized optimization problem (36) is in the required form to apply consensus ADMM [5] for distributed solution. However, the optimization problem (36)-(41) in not convex due to presence of binary variables in DSO constraints pertaining to SCBs ((12), and (14-16)). As the convergence of consensus ADMM is not guaranteed [5], [19] for non-convex problems, it is not directly applicable for distributed optimization of the problem (36)-(41).

As the non-convexity is primarily due to binary variables in DSO's constraints, the paper proposes strategic application of binary-relaxation, consensus-ADMM and branch-and-bound (BnB) method to obtain the optimal solution via distributed optimization. Each of these strategies are detailed below.

1) Binary relaxation: All the binary variables in (36)-(41) are relaxed to continuous variables. This relaxation would convert the non-convex optimization problem to second-order-conic problem (convex), which is then used for distributed optimization via consensus-ADMM. The Integer variables are absent in this paper; however same relaxation principle could be utilized for formulation of relaxed problem which is convex.

2) Consensus-ADMM: The binary relaxed version of the centralized problem (36) can be solved in distributed manner by applying consensus-ADMM. To begin, the equality constraints in (37)-(38) are used to form a augmented Lagrangian cost function as:

$$\begin{aligned} L_{\rho_{pv},\rho_{es}} &= \sum_{j\in\mathcal{I}^{PV}} \sum_{t\in\mathcal{T}} f_j(\boldsymbol{x}_{pv}^t) + \sum_{j\in\mathcal{I}^{ES}} \sum_{t\in\mathcal{T}} g_j(\boldsymbol{x}_{es}^t) + \\ &\sum_{t\in\mathcal{T}} h(\boldsymbol{x}_{lr}^t, \boldsymbol{z}_{pv}^t, \boldsymbol{z}_{\boldsymbol{es}}^{\boldsymbol{t}}) + \sum_{j\in\mathcal{I}^{PV}} \sum_{t\in\mathcal{T}} (\boldsymbol{y}_{pv,j}^t)^\top (\boldsymbol{x}_{pv,j}^t - \boldsymbol{z}_{pv,j}^t) + \\ &\sum_{j\in\mathcal{I}^{ES}} \sum_{t\in\mathcal{T}} (\boldsymbol{y}_{es}^t)^\top (\boldsymbol{x}_{es,j}^t - \boldsymbol{z}_{es,j}^t) + \frac{\rho_{pv}}{2} \sum_{j\in\mathcal{I}^{PV}} \sum_{t\in\mathcal{T}} ||\boldsymbol{x}_{pv,j}^t - \boldsymbol{z}_{pv,j}^t||^2 \\ &+ \frac{\rho_{es}}{2} \sum_{j\in\mathcal{I}^{ES}} \sum_{t\in\mathcal{T}} ||\boldsymbol{x}_{es,j}^t - \boldsymbol{z}_{es,j}^t||^2 \end{aligned} \tag{42}$$

It is to be noted that the order of summations are interchanged in each terms of (42) as compared to (36), which is valid for summing operation. Here $\boldsymbol{y}_{pv}^t$ and $\boldsymbol{y}_{es}^t$ are the vectors containing Lagrangian multipliers. The augmented Lagrangian ($L_{\rho_{pv},\rho_{es}}$) could be optimized with decision variables pertaining to PVs, BSUs and DSOs, where the non-local decision variables are hold constant to their respective updated value. This mechanism breaks $L_{\rho_{pv},\rho_{es}}$ into DERs and DSOs local optimization problem. One could refer to [5] for more detail understanding of segregating the larger optimization problem. The scaled dual variables, $\boldsymbol{u}_{pv}^t = \boldsymbol{y}_{pv}^t/\rho_{pv}$ and $\boldsymbol{u}_{es}^t = \boldsymbol{y}_{es}^t/\rho_{es}$ where ρ_{pv} and ρ_{es} are the penalty parameters, are defined to implement the scaled version of consensus-ADMM. Finally the consensus-ADMM could be solved iteratively by segregating Lagrangian cost function (42) among the PVs, BSUs and DSO. For faster convergence, the over-relaxation parameter (α) (as described in [20]) have been associated in the designed consensus-ADMM method, and it is detailed in Algorithm 1.

The consensus-ADMM performs iterative optimization until the desired degree of consensus is achieved between the DERs (PVs/BSUs) and DSO. The minimum thresholds of consensus are defined by primal and dual residues (ϵ^{pr} and ϵ^{du}). In our design, the DSO is not only a local optimization entity but also a fusion center, where the local optimal decision value from DERs ($\boldsymbol{x}_{pv}^t$, $\boldsymbol{x}_{es}^t$) are collected, dual scaled variables are updated, primal/dual residue are computed, penalty parameters are adaptively adjusted and finally dispatch the required information to DERs. Furthermore, the fusion center determines the termination condition of consensus-ADMM. The consensus-ADMM begins with initialization of scaled ($\boldsymbol{u}$) and copied variables ($\boldsymbol{z}$) to zero and residues (r and s) to larger values. Each PV owners minimize the active power curtailment (f_j) while exchanging active/reactive power to DSO, and ensure its local constraint ($\Psi_{pv,j} \leq 0$) are not violated. After solving a local optimization problem, the PVs communicate $\boldsymbol{x}_{pv,j}^t$ to fusion center (DSO). Concurrently, BSU owners minimize battery degradation cost while exchanging active/reactive power with DSO, and ensuring BSU operation under its set of constraints ($\Psi_{es,j} \leq 0$). Like PV owners, BSU owners also send its decision variables ($\boldsymbol{x}_{es,j}^t$) to DSO. Utilizing the received information on decision variables from DERs, DSO performs its local optimization to minimize loss and cost of local/distributed resources (h) while satisfying it constraints ($\phi \leq 0$). The functions of a fusion center performed by DSO are indicated by line 9 to 12 in Algorithm 1, and the whole algorithm is reiterate until the residues are lower than required degree of consensus.

3) Branch-and-Bound Optimization: The cost function of DSO was formulated with binary relaxation to convexify and implement consensus-ADMM. Hence, the consensus-ADMM would not produce discrete values of N_{sc}^t, which bring ambiguity when dispatching SCBs. The paper utilizes Least Cost Branch-and-Bound (BnB) method to determine the optimal discrete value of N_{sc}^t utilizing the consensus-ADMM for solving the DSO and DER's optimization problem. The flow of least cost BnB is shown in Algorithm 2.

The BnB starts with creation of the root node and its successive branch nodes while prioritizing nodes having lower cost until the node with lowest cost and negligible difference between lower and upper bounds is determined. Each node in the BnB is an optimization problem where the lower bound is determined by consensus-ADMM method whereas upper bound is computed by a suitable heuristic function. The general heuristic is to evaluate a cost function determined by rounding-up the relaxed variables. In context of this paper,

Algorithm 1: Consensus-ADMM: Coordination between multiple local optimizing entities

Set: Set $\rho_{pv},\ \rho_{es},\ \alpha,\ \epsilon^{pr},\ \epsilon^{du}$ to designed value

Initialize: $k=0,\ \boldsymbol{u}_{pv}^{t,k}=0,\ \boldsymbol{u}_{es}^{t,k}=0, \boldsymbol{z}_{pv}^{t}=0,$ $\boldsymbol{z}_{es}^{t}=0, r_{pv}^{k}=\infty,\ r_{es}^{k}=\infty,\ s_{pv}^{k}=\infty,\ s_{es}^{k}=\infty$

while $r_{pv}^{k}, r_{es}^{k} \geq\ \epsilon_{pv}^{pr}, \epsilon_{es}^{pr}$ ***and*** $s_{pv}^{k}, s_{es}^{k} \geq \epsilon_{pv}^{du}, \epsilon_{es}^{du}$ **do**

Optimize all PVs' local variables:

$$\boldsymbol{x}_{pv,j}^{t,k+1} = \underset{x_{pv,j}^{t}}{\text{argmin}} \sum_{t\in\mathcal{T}} f_j(x_{pv,j}^{t}) + \sum_{t\in\mathcal{T}} \frac{\rho_{pv}}{2}\|\boldsymbol{x}_{pv,j}^{t} - \boldsymbol{z}_{pv,j}^{t,k} + \boldsymbol{u}_{pv,j}^{t,k}\|^2 \quad \forall j\in\mathcal{I}^{PV}$$

$$\text{s.t.: } \Psi_{pv,j}(\boldsymbol{x}_{pv,j}^{t}) \leq 0$$

Communicate $\boldsymbol{x}_{pv,j}^{t,k+1}$ to DSO.

Optimize all BSU's local variables:

$$\boldsymbol{x}_{es,j}^{t,k+1} = \underset{x_{es,j}^{t}}{\text{argmin}} \sum_{t\in\mathcal{T}} g_j(x_{es,j}^{t}) + \sum_{t\in\mathcal{T}} \frac{\rho_{es}}{2}\|\boldsymbol{x}_{es,j}^{t} - \boldsymbol{z}_{es,j}^{t,k} + \boldsymbol{u}_{es,j}^{t,k}\|^2 \quad \forall j\in\mathcal{I}^{ES}$$

$$\text{s.t.: } \Psi_{es,j}(\boldsymbol{x}_{es,j}^{t}) \leq 0$$

Communicate $\boldsymbol{x}_{es,j}^{t,k+1}$ to DSO.

Optimize DSO's local and copied variables:

$$\boldsymbol{w}_{pv,j}^{t,k+1} = \alpha_{pv}\boldsymbol{x}_{pv,j}^{t,k+1} - (1-\alpha_{pv})\boldsymbol{z}_{pv,j}^{t,k} \quad \forall j\in\mathcal{I}^{PV}$$

$$\boldsymbol{w}_{es,j}^{t,k+1} = \alpha_{es}\boldsymbol{x}_{es,j}^{t,k+1} - (1-\alpha_{es})\boldsymbol{z}_{es,j}^{t,k} \quad \forall j\in\mathcal{I}^{ES}$$

$$\boldsymbol{x}_{lr}^{t,k+1}, \boldsymbol{z}_{pv}^{t,k+1}, \boldsymbol{z}_{es}^{t,k+1} = \underset{\boldsymbol{x}_{lr,t}^{t}, \boldsymbol{z}_{pv}^{t}, \boldsymbol{z}_{es}^{t}}{\text{argmin}} \sum_{t\in\mathcal{T}} h(\boldsymbol{x}_{lr}, \boldsymbol{z}_{pv}^{t}, \boldsymbol{z}_{es}^{t}) + \frac{\rho_{pv}}{2}\sum_{j\in\mathcal{I}^{PV}}\sum_{t\in\mathcal{T}}\|\boldsymbol{w}_{pv,j}^{t,k+1} - \boldsymbol{z}_{pv,j}^{t} + \boldsymbol{u}_{pv,j}^{t,k}\|^2 + \frac{\rho_{es}}{2}\sum_{j\in\mathcal{I}^{ES}}\sum_{t\in\mathcal{T}}\|\boldsymbol{w}_{es,j}^{t,k+1} - \boldsymbol{z}_{es,j}^{t} + \boldsymbol{u}_{es,j}^{t,k}\|^2$$

$$\text{s.t.: } \Phi(\boldsymbol{x}_{lr}^{t}, \boldsymbol{z}_{pv}^{t}, \boldsymbol{z}_{es}^{t}) \leq 0$$

Update dual scaled variables:

$$\boldsymbol{u}_{pv,j}^{t,k+1} = \boldsymbol{u}_{pv,j}^{t,k} + \boldsymbol{w}_{pv,j}^{t,k+1} - \boldsymbol{z}_{pv,j}^{t,k+1}$$

$$\boldsymbol{u}_{es,j}^{t,k+1} = \boldsymbol{u}_{es,j}^{t,k} + \boldsymbol{w}_{es,j}^{t,k+1} - \boldsymbol{z}_{es,j}^{t,k+1}$$

Compute primal and dual residue:

$$\boldsymbol{r}_{pv}^{k+1}, \boldsymbol{r}_{es}^{k+1} = \|\boldsymbol{x}_{pv}^{t,k+1} - \boldsymbol{z}_{pv}^{t,k+1}, \boldsymbol{x}_{es}^{t,k+1} - \boldsymbol{z}_{es}^{t,k+1}\|$$

$$\boldsymbol{s}_{pv}^{k+1}, \boldsymbol{s}_{es}^{k+1} = \| -\rho_{pv}(\boldsymbol{z}_{pv}^{t,k+1} - \boldsymbol{z}_{pv}^{t,k}), -\rho_{es}(\boldsymbol{z}_{es}^{t,k+1} - \boldsymbol{z}_{es}^{t,k})\|$$

Self adaptive adjustment of ρ_{pv} and ρ_{es}:

if $r_{pv}^{k+1} \geq 4s_{pv}^{k+1}$ **then**
 $\rho_{pv} = 2\rho_{pv}, \quad \boldsymbol{u}_{pv}^{t,k+1} = \boldsymbol{u}_{pv}^{t,k+1}/2$
else if $s_{pv}^{k+1} \geq 4r_{pv}^{k+1}$ **then**
 $\rho_{pv} = \rho_{pv}/2, \quad \boldsymbol{u}_{pv}^{t,k+1} = 2\boldsymbol{u}_{pv}^{t,k+1}$
end

if $r_{es}^{k+1} \geq 4s_{es}^{k+1}$ **then**
 $\rho_{es} = 2\rho_{es}, \quad \boldsymbol{u}_{es}^{t,k+1} = \boldsymbol{u}_{es}^{t,k+1}/2$
else if $s_{es}^{k+1} \geq 4r_{es}^{k+1}$ **then**
 $\rho_{es} = \rho_{es}/2, \quad \boldsymbol{u}_{es}^{t,k+1} = 2\boldsymbol{u}_{es}^{t,k+1}$
end

Communicate $(\rho_{pv}, \boldsymbol{u}_{pv,j}^{t,k+1})$ and $(\rho_{es}, \boldsymbol{u}_{es,j}^{t,k+1})$ to corresponding PV and BSU.

k= k+1

end

the nodes are formulated on the basis of DSO's cost function in search of binary solution of b_{ij}^{t}. The BnB is initialized with best node to be the root node at first and the all the binary relaxed variables are arranged in set $\mathcal{B}$. The root node is pushed into a stack $\mathcal{S}$ before the search of optimal solution is started. A node is poped from a stack and checked if it has a feasible solution. If the a feasible solution do not exist for the node, it is dropped out. Otherwise, it is again checked for updating the best lower bound. The node is further processed for killing, discrete solution or branching as indicated in line 10, 11, and 12 of Algorithm 2. Left and right branch nodes are created for a node which requires branching by adding a constraints over its parent node. After then the lower and upper bounds of branch nodes are determined by consensus-ADMM strategy defined in Algorithm 1. At the end the branch nodes are pushed to stack in ascending order of their cost. The set of relaxed variables ($\mathcal{B}$) is rotated left for adding the constraints for next relaxed variable in the successive branch nodes if required. Finally the best node would contain the discrete optimal solution of DSO's cost function.

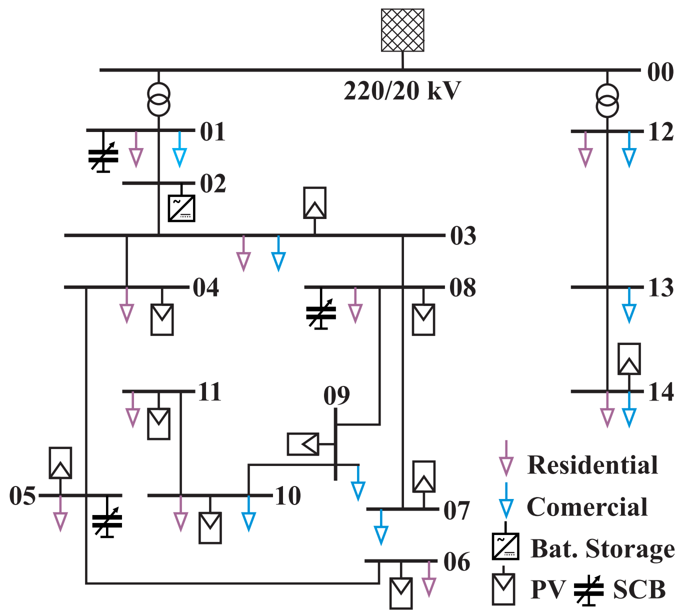

Fig. 1. Cigre MV distribution network [8], [21]

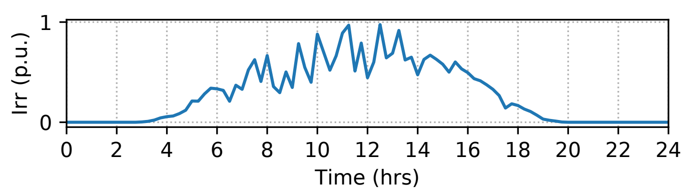

Fig. 2. 15-minute averaged solar irradiation received on 20^{th} August [22].

IV. Simulation Results

A. Simulation with only local control of SCBs

B. Simulation with the proposed centralized control scheme

C. Simulation with the proposed distributed control

V. Conclusion

Algorithm 2: Least Cost BnB embedding consensus-ADMM

Set: Set the binary-relaxed optimization model of DSO as root node ($\mathcal{N}^{root}$).
Initialize: $\lfloor \mathcal{N}^{root} \rfloor = 0$, $\lceil \mathcal{N}^{root} \rceil = \infty$, $\mathcal{N}^{best} = \mathcal{N}^{root}$, $\mathcal{N}^{root}$ has a feasible solution.
Store the binary-relaxed variables of $\mathcal{N}^{root}$ in set $\mathcal{B}$.
Push $\mathcal{N}^{root}$ in a stack $\mathcal{S}$.
while *$\mathcal{S}$ is not empty* **do**
    $\mathcal{N}$ = pop a node from $\mathcal{S}$.
    if *$\mathcal{N}$ has feasible solution:* **then**
        if $\lceil \mathcal{N} \rceil < \lceil \mathcal{N}^{best} \rceil$ **then**
            $\lceil \mathcal{N}^{best} \rceil = \lceil \mathcal{N} \rceil$
        end
        if $\lfloor \mathcal{N} \rfloor > \lceil \mathcal{N}^{best} \rceil$ **then**
            Kill $\mathcal{N}$
        else if $|(\lfloor \mathcal{N} \rfloor - \lceil \mathcal{N} \rceil| < 10^{-5}$ **then**
            $\mathcal{N}^{best} = \mathcal{N}$
        else
            Create left and right branch nodes:
            Left Node ($\mathcal{N}^L$): Create by adding a constraint $\mathcal{B}[0] = 1$ in its parent node $\mathcal{N}$.
            Evaluate $\mathcal{N}^L$ using Consensus-ADMM.
            Collect the optimal value of objective functions of PVs (f_j^*) and BSUs (g_j^*).
            Set $\lfloor \mathcal{N}^L \rfloor = h^* + \sum_{j \in \mathcal{I}^{PV}} f_j^* + \sum_{j \in \mathcal{I}^{ES}} g_j^*$
            Roundup the binary relaxed variables to integers to evaluate $\lceil h^* \rceil$.
            Set $\lceil \mathcal{N}^L \rceil = \lceil h^* \rceil + \sum_{j \in \mathcal{I}^{PV}} f_j^* + \sum_{j \in \mathcal{I}^{ES}} g_j^*$
            Right Node ($\mathcal{N}^R$): Create by adding a constraint $\mathcal{B}[0] = 0$ in its parent node $\mathcal{N}$.
            Evaluate $\mathcal{N}^R$ using Consensus-ADMM.
            Collect the optimal value of objective functions of PVs (f_j^*) and BSUs (g_j^*).
            Set $\lfloor \mathcal{N}^R \rfloor = h^* + \sum_{j \in \mathcal{I}^{PV}} f_j^* + \sum_{j \in \mathcal{I}^{ES}} g_j^*$
            Roundup the binary relaxed variables to integers to evaluate $\lceil h^* \rceil$.
            Set $\lceil \mathcal{N}^R \rceil = \lceil h^* \rceil + \sum_{j \in \mathcal{I}^{PV}} f_j^* + \sum_{j \in \mathcal{I}^{ES}} g_j^*$
            Stack left and right nodes as per their cost:
            if $\lfloor \mathcal{N}^L \rfloor \leq \lfloor \mathcal{N}^R \rfloor$ **then**
                Push $\mathcal{N}^R$ and then $\mathcal{N}^L$ in $\mathcal{S}$
            else
                Stack $\mathcal{N}^L$ and then $\mathcal{N}^R$ in $\mathcal{S}$
            end
            Left rotate the set $\mathcal{B}$.
        end
    end
end
$\mathcal{N}^{best}$ contains the optimal solution.

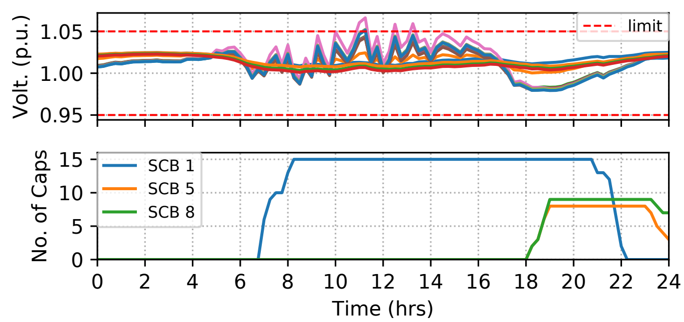

Fig. 3. Simulation with only local control of SCB

VI. Mathematical problem to be solved using ADMM and verify its solution with centralized optimization

Consider a following optimization problem:

$$\textbf{min} \sum_{t=1}^{5} \Big[5x^t + 3y^t + 50(\Delta n^t)^2 + 100(\hat{y} - y^t) \Big] \tag{43}$$

subject to :

$$x^t \geq 0 \tag{44}$$

$$x^t + y^t + 5n^t = \hat{l} \tag{45}$$

$$0 \leq y^t \leq \hat{y} \tag{46}$$

$$\Delta n^t = n^t - n^{t-1}; \quad n^0 = 0 \tag{47}$$

$$0 \leq n \leq 8, \quad \text{(n is an integer variable)} \tag{48}$$

$$\hat{y}[1:5] = [2, 4, 8, 7, 10] \tag{49}$$

$$\hat{l}[1:5] = [25, 40, 35, 50, 55] \tag{50}$$

We will separate the objective function (54) in two parts, such as:

$$\textbf{min} \sum_{t=0}^{5} \Big[5x^t + 3y^t + 50(\Delta n^t)^2 \Big] + \textbf{min} \quad \sum_{t=0}^{5} 100(\hat{y} - y^t) \tag{51}$$

For applying ADMM, we shall separate the common variable y from both objective by introducing an auxiliary variable z, such as:

$$\textbf{min} \sum_{t=0}^{5} \Big[5x^t + 3y^t + 50(\Delta n^t)^2 \Big] + \textbf{min} \sum_{t=0}^{5} 100(\hat{y} - z^t) \tag{52}$$

subject to:

$$y^t - z^t = 0 \tag{53}$$

VII. Things to perform

1) First solve the above optimization with centralized optimization.
2) Solve the same problem with integer relaxation using ADMM.
3) Solve the same problem with Branch and Bound integrated ADMM.

VIII. Mathematical problem to be solved using ADMM and verify its solution with centralized optimization

Consider a following optimization problem:

$$\textbf{min} \sum_{t=1}^{5} \Big[5x^t + 3y^t + 4w_d^t + 50(\Delta n^t)^2 + 100(\hat{y} - y^t) + 200(0.8 - q^t)^2\Big] \tag{54}$$

subject to :

$$x^t \geq 0 \tag{55}$$

$$x^t + y^t + 5n^t + w_d^t = \hat{l} + w_c^t \tag{56}$$

$$0 \leq y^t \leq \hat{y} \tag{57}$$

$$\Delta n^t = n^t - n^{t-1}; \quad n^0 = 0 \tag{58}$$

$$0 \leq n \leq 8, \quad \text{(n is an integer variable)} \tag{59}$$

$$\hat{y}[1:5] = [2, 4, 8, 7, 10] \tag{60}$$

$$\hat{l}[1:5] = [25, 40, 35, 50, 55] \tag{61}$$

$$q^t = q^{t-1} + 0.05(w_c^t - 5w_d^t); \quad q^0 = 0.8 \tag{62}$$

$$0 \leq w_c^t \leq 10\beta_c^t \tag{63}$$

$$0 \leq w_d^t \leq 10\beta_d^t \tag{64}$$

$$\beta_c^t + \beta_d^t \leq 1.5; \quad \beta_c \text{ and } \beta_d \text{ are binary variables} \tag{65}$$

$$0.4 \leq q^t \leq 1 \tag{66}$$

We will separate the objective function (54) in three parts, such as:

$$\textbf{min} \sum_{t=0}^{5} \Big[5x^t + 3y^t + 4w_d^t + 50(\Delta n^t)^2\Big] + \textbf{min} \quad \sum_{t=0}^{5} 100(\hat{y} - y^t) + \textbf{min} \sum_{t=0}^{5} 200(0.8 - q^t)^2 \tag{67}$$

For applying ADMM, we shall separate the common variable y between first and second objectives, also w_d and w_c between first and third objectives. We introduce auxiliary variables z, u_d, and u_c for separation of variables, such as:

$$\textbf{min} \sum_{t=0}^{5} \Big[5x^t + 3y^t + 4w_d^t + 50(\Delta n^t)^2\Big] + \textbf{min} \sum_{t=0}^{5} 100(\hat{y} - z^t) + \textbf{min} \sum_{t=0}^{5} 200(0.8 - q^t)^2 \tag{68}$$

subject to:

$$y^t - z^t = 0 \tag{69}$$

$$w_d^t - u_d^t = 0 \tag{70}$$

$$w_c^t - u_c^t = 0 \tag{71}$$

IX. Things to perform

1) First solve the above optimization with centralized optimization.
2) Solve the same problem with integer relaxation using ADMM.
3) Solve the same problem with Branch and Bound integrated ADMM.

References

[1] L. Miller, L. Cibulka, M. Brown, A. v. Meier, and R. Arghandeh, "Distribution System Voltage Management and Optimization for Integration of Renewables and Electric Vehicles," California Institute for Energy and Environment, Tech. Rep., 2013.

[2] S. Singh, V. B. Pamshetti, and S. P. Singh, "Time Horizon-based Model Predictive Volt/VAR Optimization for Smart Grid Enabled CVR in Presence of Electric Vehicle Charging Loads," *IEEE Transactions on Industry Applications*, vol. PP, no. c, pp. 1–1, 2019.

[3] Y. Xu, Z. Y. Dong, R. Zhang, and D. J. Hill, "Multi-Timescale Coordinated Voltage/Var Control of High Renewable-Penetrated Distribution Systems," *IEEE Transactions on Power Systems*, vol. 32, no. 6, pp. 4398–4408, 11 2017.

[4] A. Saad, T. Youssef, A. T. Elsayed, A. Amin, O. H. Abdalla, and O. Mohammed, "Data-Centric Hierarchical Distributed Model Predictive Control for Smart Grid Energy Management," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 7, pp. 4086–4098, 2019.

[5] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2010.

[6] W. Zheng, W. Wu, B. Zhang, H. Sun, and Y. Liu, "A Fully Distributed Reactive Power Optimization and Control Method for Active Distribution Networks," *IEEE Transactions on Smart Grid*, vol. 7, no. 2, pp. 1021–1033, 2016.

[7] W. E. Hart, C. D. Laird, J.-P. Watson, D. L. Woodruff, G. A. Hackebeil, B. L. Nicholson, and J. D. Siirola, *Pyomo — Optimization Modeling in Python*, 2nd ed., ser. Springer Optimization and Its Applications. Cham: Springer International Publishing, 2017, vol. 67.

[8] L. Thurner, A. Scheidler, F. Schafer, J. H. Menke, J. Dollichon, F. Meier, S. Meinecke, and M. Braun, "Pandapower - An Open-Source Python Tool for Convenient Modeling, Analysis, and Optimization of Electric Power Systems," *IEEE Transactions on Power Systems*, vol. 33, no. 6, pp. 6510–6521, 2018.

[9] IEEE Standard Association, *IEEE Std. 1547-2018. Standard for Interconnection and Interoperability of Distributed Energy Resources with Associated Electric Power Systems Interfaces*. IEEE, 2018.

[10] M. Kraning, Y. Wang, E. Akuiyibo, and S. Boyd, "Operation and configuration of a storage portfolio via convex optimization," *IFAC Proceedings Volumes (IFAC-PapersOnline)*, vol. 44, no. 1 PART 1, pp. 10 487–10 492, 2011.

[11] K. Smith, M. Earleywine, E. Wood, J. Neubauer, and A. Pesaran, "Comparison of plug-in hybrid electric vehicle battery life across geographies and drive cycles," *SAE Technical Papers*, 2012.

[12] M. Koller, T. Borsche, A. Ulbig, and G. Andersson, "Defining a degradation cost function for optimal control of a battery energy storage system," *2013 IEEE Grenoble Conference PowerTech, POWERTECH 2013*, 2013.

[13] Y. Jiang, C. Wan, J. Wang, Y. Song, and Z. Y. Dong, "Stochastic receding horizon control of active distribution networks with distributed renewables," *IEEE Transactions on Power Systems*, vol. 34, no. 2, pp. 1325–1341, 2019.

[14] L. Liberti, *Mathematical Progamming*. [Online]. Available: https://www.lix.polytechnique.fr/ liberti/teaching/dix/inf580-15/mathprog.pdf

[15] D. G. Luenberger and Y. Ye, *Linear and nonlinear programming*. Springer, 2008.

[16] M. Farivar and S. H. Low, "Branch flow model: Relaxations and convexification-part i," *IEEE Transactions on Power Systems*, vol. 28, no. 3, pp. 2554–2564, 2013.

[17] H. G. Yeh, D. F. Gayme, and S. H. Low, "Adaptive VAR control for distribution circuits with photovoltaic generators," *IEEE Transactions on Power Systems*, vol. 27, no. 3, pp. 1656–1663, 2012.

[18] E.ON Netz GmBH, "Grid Connection Regulations for High and Extra High Voltage," pp. 1–46, 2006. [Online]. Available: https://www.nerc.com/docs/pc/ivgtf/German$_{E}ON_{G}rid_{C}ode.pdf$

[19] R. Takapoui, N. Moehle, S. Boyd, and A. Bemporad, "A simple effective heuristic for embedded mixed-integer quadratic programming," *International Journal of Control*, no. 1, pp. 1–11, 2017. [Online]. Available: http://dx.doi.org/10.1080/00207179.2017.1316016

[20] E. Ghadimi, A. Teixeira, I. Shames, and M. Johansson, "Optimal parameter selection for the Alternating Direction Method of Multipliers (ADMM): Quadratic problems," *IEEE Transactions on Automatic Control*, vol. 60, no. 3, pp. 644–658, 2015.

[21] K. Rudion, A. Orths, Z. A. Styczynski, and K. Strunz, "Design of benchmark of medium voltage distribution network for investigation of DG integration," *2006 IEEE Power Engineering Society General Meeting, PES*, p. 6 pp., 2006.

[22] J. Kalisch, D. Schmidt, Thomas Heinemann, and E. Lorenz, "Continuous meteorological observations in high-resolution (1Hz) at University of Oldenburg," 2015. [Online]. Available: https://doi.pangaea.de/10.1594/PANGAEA.847476