Coordination and Control of Distributed Energy Resources for Provision of Ancillary Services

Alejandro D. Domínguez-García†, Christoforos N. Hadjicostis‡†, Brett. A. Robbins†

†Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
‡Department of Electrical and Computer Engineering
University of Cyprus

Los Alamos National Laboratory
Los Alamos, NM
June 20, 2011
D-G Group Research Overview

- **Renewable Resources Integration**
  - Impact of wind integration on system dynamic performance
  - Stochastic models of PV systems for quantification of reliability/energy-yield

- **Health Monitoring and Fault Diagnosis of Electrical Energy Systems**
  - Fault detection and isolation algorithms for FACTS devices
  - Health monitoring of micro-grids and other small-footprint power systems

- **Reliability Models for Next Generation Electric Power Grids**
  - Impacts of coupling between the system cyber and physical layers
  - Impacts of coupling between system dynamics and component stress

- **Coordination and Control of Distributed Energy Resources**
  - Distributed reactive power support for voltage control
  - Distributed energy storage for frequency and peak-shaving control
Outline

Introduction

Mathematical Preliminaries

Distributed Control with Unconstrained Node Capacity

Distributed Control with Constrained Node Capacity

Voltage Control in Subtransmission Networks

Concluding Remarks
Outline

Introduction

Mathematical Preliminaries

Distributed Control with Unconstrained Node Capacity

Distributed Control with Constrained Node Capacity

Voltage Control in Subtransmission Networks

Concluding Remarks
The Potential of Distributed Energy Resources (DERs)

- On the power grid, there exist many DERs that can be potentially used to provide ancillary services
- Power electronics grid interfaces commonly used in DERs can be utilized to provide reactive power support
  - Voltage control in subtransmission and distribution
- Plug-in-hybrid vehicles (PHEV) can be utilized for providing active power for up and down regulation
  - Energy peak-shaving during peak hours and load-leveling at night
Control and Coordination of DERs

- Proper coordination and control of these DERs is key for enabling their utilization for ancillary services.

- One solution can be achieved through a centralized control strategy where each DER is commanded from a central controller:
  - It is necessary to overlay a communication network connecting the central controller with each distributed resource.
  - It requires knowledge of the distributed resources that are available on the distribution side at any given time.

- We propose an alternative approach that utilizes distributed strategies for control and coordination of DERs.

- These strategies offer several advantages, including the following:
  - More economical as they do not require communication infrastructure between a centralized controller and the various devices.
  - They do not require complete knowledge of the distributed resources available.
  - Potentially more resilient to faults and/or unpredictable behavioral patterns by the distributed resources.
Basic Setup

- The DERs can be thought of as nodes in a network, where each node can exchange information with neighboring nodes.
- Through an iterative process with neighboring nodes (presumable nodes in close proximity)
  - Each DER in the network will compute the amount of active or reactive power that it needs to provide.
- Collectively, the local control decisions made by the resources should have the same effect as the centralized control strategy (or, if multiple solutions are possible, be one of the feasible solutions).
- Such a solution could rely on inexpensive and simple communication protocols, e.g., ZigBee technology.
- We discuss several algorithms to solve this coordination/cooperation problem under various conditions.
Outline

Introduction

Mathematical Preliminaries

Distributed Control with Unconstrained Node Capacity

Distributed Control with Constrained Node Capacity

Voltage Control in Subtransmission Networks

Concluding Remarks
Graph-Theoretic Notions

- The exchange of information between nodes where resources are located can be described by a directed graph $G = \{V, E\}$:
  - $V = \{1, 2, \ldots, n\}$ is the vertex set (each vertex corresponds to a node)
  - $E \subseteq V \times V$ is the set of directed edges, where $(j, i) \in E$ if node $j$ can receive information from node $i$

- The graph is undirected if and only if whenever $(j, i) \in E$, then also $(i, j) \in E$, i.e., if node $j$ can receive information from node $i$, then node $i$ can also receive information from node $j$

- All nodes that can transmit information to node $j$ are said to be neighbors of node $j$ and are represented by the set $N_j = \{i \in V : (j, i) \in E\}$

- The number of neighbors of $j$ is called the in-degree of $j$ and denoted by $D_j^- = |N_j|$

- The number of nodes that have $j$ as neighbor, i.e., $j$ can transmit information to these nodes, is called the out-degree of $j$ and is denoted by $D_j^+$

- In undirected graphs, $D_j^- = D_j^+$ for all nodes $j$
Distributed Algorithms Formulation

- Let $\pi_j[k]$ be the amount of active or reactive power demanded from the distributed resource located in node $j$ at the $k$ round of information exchange between nodes.

- The proposed distributed algorithms determine the amount of resource that will be contributed by node $j$ by performing linear iterations of the form

$$
\pi_j[k + 1] = p_{jj}[k]\pi_j[k] + \sum_{i \in N_j} p_{ji}[k]\pi_i[k],
$$

(1)

where the $p_{ji}[k]$’s are a set of (potentially time-varying) weights, and $\pi_j[k]$’s are non-negative quantities.

- Each node updates its demanded amount to be a linear combination of its own demanded amount and the demanded amount of its neighbors.

- The choice of $p_{ji}[k]$’s will depend on the problem constraints.

- We provide algorithms when
  - There is no limit on the amount of active or reactive power that each resource can provide.
  - The maximum (or minimum) amount of active or reactive power each resource can provide is limited.
Outline

Introduction

Mathematical Preliminaries

Distributed Control with Unconstrained Node Capacity

Distributed Control with Constrained Node Capacity

Voltage Control in Subtransmission Networks

Concluding Remarks
The Basic Setup

- There is a leading node that knows the total amount of active or reactive power $\rho_d$ that needs to be provided by the remaining $n$ nodes.
- This leader can communicate with $l \geq 1$ nodes, and initially sends a command demanding $\frac{\rho_d}{l}$ units of active or reactive power from each of them.
- Unless $\rho_d$ changes, the leader will not subsequently communicate with the nodes.
- Let $\pi_j[k]$ be the active or reactive power demanded from node $j$ at step $k$, and define the corresponding active or reactive power demand vector as $\pi[k] = [\pi_1[k], \pi_2[k], \ldots, \pi_j[k], \ldots, \pi_n[k]]'$.
- Define the collective active or reactive power demand as $\rho[k] = \sum_{j=1}^{n} \pi_j[k]$.
- The objective is to design a distributed iterative algorithm that, at step $k$, updates the active or reactive power demand from node $j$ based on:
  - Its own current active or reactive power demand $\pi_j[k]$.
  - Current active or reactive power demanded from neighbors of $j$.
- Such that after $m$ steps the collective active or reactive power demand equals the total active or reactive power demanded by the leader: $\rho[m] = \sum_{j=1}^{n} \pi_j[m] = \rho_d$. 

The simplest solution, which results in constant weights $p_{ji}$, is for each node $j$ to equally split its current value among itself and the nodes that have $j$ as neighbor

$$\pi_j[k + 1] = \frac{1}{1 + D_j^+} \pi_j[k] + \sum_{i \in N_j} \frac{1}{1 + D_i^+} \pi_i[k]$$

where $D_i^+$ the number of nodes that $i$ can transmit information to (the out-degree of node $i$)

Algorithm (2) does not necessarily split the total active or reactive power demand $\rho_d$ evenly among all the nodes, but it ensures that

$$\sum_{j=1}^{N} \pi_j[k] = \rho_d, \ \forall k \geq 0$$

Provided the directed graph describing the exchanges between nodes has a single recurrent class, which necessarily makes it aperiodic by construction due to the fact that $\frac{1}{1 + D_j^+} \neq 0$, the steady state solution provided by (2) is unique
Sketch of Convergence Proof

- The splitting algorithm (2) can be rewritten in matrix form as

\[
\pi[k + 1] = P_c \pi[k], \\
\pi[0] = \pi_0,
\]

(3)

where \( \pi_0 = [\pi_1[0], \pi_2[0], \ldots, \pi_j[0], \ldots, \pi_n[0]]' \) with \( \pi_i[0] = \rho_d/l \) if \( i \) is a neighbor of the leader node and \( \pi_i[0] = 0 \) otherwise

- By construction, matrix \( P_c \) is column stochastic and also primitive

- The Perron-Frobenius theorem for non-negative matrices states that \( P_c \) has a unique eigenvalue with largest modulus at \( \lambda_1 = 1 \)

- Let \( x \) be a right eigenvector of \( P_c \) associated with \( \lambda_1 \) and let \( y \) be a left eigenvector of \( P_c \) associated with \( \lambda_1 \) such that \( x' y = 1 \)

- Again, from the fact that \( P_c \) is column stochastic, the entries of vector \( y \) must be all equal

- Without loss of generality, let \( y = [1, 1, \ldots, 1]' \), and since \( x' y = 1 \), the entries of \( x \) must add up to one

- Then the steady-state solution of (3) is given by

\[
\pi^{ss} = xy' \pi_0 = (\sum_{j=1}^{N} \pi_j[0])x.
\]

(4)

- Since \( \sum_{j=1}^{N} \pi_j[0] = \rho_d \) and the entries of \( x \) are nonnegative and add up to one, it follows that entries of \( \pi^{ss} \) are nonnegative and add up to \( \rho_d \).
Uneven Splitting Example

(a) Network topology.  
(b) Node demanded capacity evolution.

Figure: Four-node network implementing uneven splitting strategy.

- The leader initially splits $\rho_d = 1$ in half and passes it to nodes 1 and 2.
- Each node updates its value as follows

\[
\begin{align*}
\pi_1[k+1] &= \frac{1}{3}(\pi_1[k] + \pi_2[k] + \pi_3[k]), \\
\pi_2[k+1] &= \frac{1}{3}(\pi_1[k] + \pi_2[k]) + \frac{1}{2}\pi_4[k], \\
\pi_3[k+1] &= \frac{1}{3}(\pi_1[k] + \pi_3[k]), \\
\pi_4[k+1] &= \frac{1}{3}(\pi_2[k] + \pi_3[k]) + \frac{1}{2}\pi_4[k],
\end{align*}
\]

with $\pi_1[0] = \pi_2[0] = \frac{1}{2}$, and $\pi_3[0] = \pi_4[0] = 0$.

- The steady-state solution is given by $\pi^{ss} = [0.23, 0.35, 0.11, 0.31]'$.  

\[\sum_{i=1}^{n} \pi_i^{ss} = 1\]
A solution to reach even splitting can be easily obtained when
\[ \sum_{i=1}^{n} p_{ji} = \sum_{i=1}^{n} p_{ij} = 1 \] for all \( j = 1, 2, \ldots, n \)

The simplest realization of such algorithm is obtained when the graph describing the exchanges of information is undirected, which results in
\[ D_j^- = D_j^+ := D_j, \; \forall j = 1, \ldots, n \]

Define the maximum degree of the network as \( D = \max_j \{D_j\} \)

One way even splitting can be achieved is by having each node \( j \) update its value as follows
\[
\pi_j[k+1] = (1 - \frac{|N_j|}{1 + D})\pi_j[k] + \sum_{i \in N_j} \frac{1}{1 + D} \pi_i[k],
\] (6)

where \( |N_j| = D_j^+ \) denotes the number of elements in the set \( N_j \)

Even splitting can also be achieved if instead of \( D \), we use any upper bound \( D' \geq D \) in (6)
Even Splitting Example

(a) Network topology.  
(b) Node demanded capacity evolution.

**Figure:** Four-node network implementing even splitting strategy.

- Each node updates its value as follows
  
  \[
  \pi_1[k + 1] = \frac{1}{3}(\pi_1[k] + \pi_2[k] + \pi_3[k]), \\
  \pi_2[k + 1] = \frac{1}{3}(\pi_1[k] + \pi_2[k]) + \frac{1}{2}\pi_4[k], \\
  \pi_3[k + 1] = \frac{1}{3}\pi_1[k] + \frac{1}{3}\pi_3[k], \\
  \pi_4[k + 1] = \frac{1}{3}\pi_2[k] + \frac{2}{3}\pi_4[k],
  \]

  with \(\pi_1[0] = \pi_2[0] = 1/2\), and \(\pi_3[0] = \pi_4[0] = 0\).

- The steady-state solution is \(\pi^{ss} = [0.25, 0.25, 0.25, 0.25]'\).
- The key is that the corresponding transition matrix is doubly stochastic.
Outline

Introduction

Mathematical Preliminaries

Distributed Control with Unconstrained Node Capacity

Distributed Control with Constrained Node Capacity

Voltage Control in Subtransmission Networks

Concluding Remarks
The Setup

- We now address the case where nodes have limits on the amounts of active or reactive power they can provide.
- Let \( \pi_j^{\text{max}} \), for \( j = 1, 2, \ldots, n \), be the maximum active or reactive power that node \( j \) can provide (its maximum capacity).
- Define the corresponding maximum active or reactive power capacity vector as \( \pi^{\text{max}} = [\pi_1^{\text{max}}, \pi_2^{\text{max}}, \ldots, \pi_n^{\text{max}}]'. \)
- Let \( \rho[k] = \sum_{j=1}^{n} \pi_j[k] \) be the collective active or reactive power capacity demanded from the nodes at instant \( k \), and \( \rho_d \) be the collective active or reactive power demand.
  - We assume that \( \rho_d \leq \sum_{j=1}^{n} \pi_j^{\text{max}} := \chi^{\text{max}} \).
- The objective is to design a distributed iterative algorithm that, at step \( k \), updates the active or reactive power demanded from node \( j \) based on:
  - Current active or reactive power demand \( \pi_j[k] \)
  - Current active or reactive power demanded by neighbors of \( j \)
  such that after \( m \) steps:
    - \( \pi_j[m] \) reaches a steady state value \( \pi_j^s \leq \pi_j^{\text{max}}, \forall j \)
    - \( \sum_{j=1}^{n} \pi_j^{ss} = \rho_d \leq \chi^{\text{max}} \).
Fair Splitting Algorithm

- A simple solution to the constrained problem can be obtained if each node could compute (or knows) the ratio $\rho_d/\chi^{\max}$
- Then, the total active or reactive power demand can be collectively provided by having each node $j$ provide

$$\pi_j := \frac{\rho_d}{\chi^{\max}} \pi_j^{\max} \leq \pi_j^{\max} \quad (8)$$

- This can be achieved if the nodes run twice (with appropriate initial conditions) the following algorithm

$$\pi_j[k + 1] = \frac{1}{1 + D_j^+} \pi_j[k] + \sum_{i \in N_j} \frac{1}{1 + D_i^+} \pi_i[k]. \quad (9)$$

- Let $\hat{\pi}[k]$ denote the solution to (9) achieved with initial conditions set to initially demanded capacity
- Let $\tilde{\pi}[k]$ denote the solution to (9) achieved with initial conditions set to maximum capacities
- At each iteration step, each node $j$ computes

$$\pi_j[k] = \frac{\hat{\pi}_j[k]}{\tilde{\pi}_j[k]} \pi_j^{\max} \quad (10)$$

- Then

$$\lim_{k \to \infty} \pi_j[k] = \lim_{k \to \infty} \frac{\hat{\pi}_j[k]}{\tilde{\pi}_j[k]} \pi_j^{\max} = \frac{\rho_d}{\chi^{\max}} \pi_j^{\max} = \pi_j \quad (11)$$
Sketch of Convergence Proof

- As before, the linear iteration in (9) can be rewritten in matrix form as
  \[ \pi[k + 1] = P_c \pi[k], \]
  \( (12) \)

- By construction, matrix \( P_c \) is column stochastic and also primitive (assuming the graph is strongly connected)

- Since \( P_c \) is column stochastic, it follows that
  \[ \sum_{j=1}^{N} \hat{\pi}_j[k] = \rho_d, \quad \sum_{j=1}^{N} \tilde{\pi}_j[k] = \chi_{\text{max}}, \quad \forall k. \]
  \( (13) \)

- Since \( P_c \) is primitive, Perron-Frobenius ensures that
  \[ \hat{\pi}_j^{ss} := \lim_{k \to \infty} \hat{\pi}_j[k] = \alpha \lim_{k \to \infty} \tilde{\pi}_j[k] =: \alpha \tilde{\pi}_j^{ss}, \quad \forall j, \text{ for some } \alpha > 0 \]

- Thus, \( \rho_d = \sum_{j=1}^{N} \hat{\pi}_j^{ss} = \alpha \sum_{j=1}^{N} \tilde{\pi}_j^{ss} = \alpha \chi_{\text{max}}, \) and therefore
  \[ \alpha = \frac{\rho_d}{\chi_{\text{max}}}. \]
  \( (14) \)

- Then, the contribution \( \pi[k] \) of each will asymptotically converge to the desired value \( \pi_j \)

  \[ \pi[k] = \lim_{k \to \infty} \frac{\hat{\pi}_j[k]}{\hat{\pi}_j[k]} \pi_j^{max} = \frac{\hat{\pi}_j^{ss}}{\tilde{\pi}_j^{ss}} \pi_j^{max} = \alpha \pi_j^{max} = \frac{\rho_d}{\chi_{\text{max}}} \pi_j^{max} = \pi_j. \]
  \( (15) \)
Fair Splitting Example

(a) Network topology. (b) Evolution of \( \hat{\pi}_j \) in the algorithm second run.

Figure: Four-node network implementing even splitting strategy.

- Take \( \rho_d = 1 \) and \( \pi^{\text{max}} = [0.4, 0.2, 0.4, 0.1] \) so that \( \chi^{\text{max}} = 1.1 > \rho_d \)
- Each node runs twice (in parallel) the same algorithm used in the even splitting example
  - For the first run, initial conditions are \( \hat{\pi}[0] = [0.5, 0.5, 0, 0]' \)
  - For the second run, initial conditions are \( \hat{\pi}[0] = \pi^{\text{max}} = [0.4, 0.2, 0.4, 0.1]' \)
- The solution of the first run is equivalent to the solution of the even splitting algorithm example, i.e., each node computes \( \rho_d/4 = 0.25 \)
- On the second run, each node computes \( \sum_{j=1}^{4} \pi_{j}^{\text{max}}/4 = 0.275 \)
- The contributions are \( \pi = \alpha[0.4, 0.2, 0.4, 0.1]' \), where \( \alpha = 0.25/0.275 \)
Outline

Introduction

Mathematical Preliminaries

Distributed Control with Unconstrained Node Capacity

Distributed Control with Constrained Node Capacity

Voltage Control in Subtransmission Networks

Concluding Remarks
Operational conditions require the voltage at bus $s$ of the electrical network to be maintained at a reference voltage $V_s^{ref}$.

In order to achieve this requirement, we control the total reactive power demand at bus $s$, which is given by $Q_s + Q_s^d$, where

- $Q_s$ is the reactive power injected in bus $s$ provided by the network, and
- $Q_s^d$ is the reactive power injected in bus $s$ provided by DERs.

By controlling $Q_s^d$, the total demand of reactive power at bus $s$ can be effectively controlled.
We implement two control architectures:

- All load buses can provide reactive power through coordination of DERs
- Only bus 6 can provide reactive power through coordination of DERs

For each of the architectures, we implement distributed coordination algorithms with and without constraints on DER capacity

- For simplicity, we assume the same DER network with four nodes showed in previous slides

We compare the performance of control architectures and distributed algorithms for a contingency in generator 2
Base Case: No Reactive Power Support

Figure: Voltages before and after contingency.

Prior to contingency: Generator 5 is at maximum capacity

After contingency:

- Generator 5 becomes a $PQ$ bus, and Generator 1 picks up all the reactive power demand
- Voltages on buses 5 and 6 fall outside ±5% of the nominal value (1 p.u.)

Table: Pre-contingency power flow solution

<table>
<thead>
<tr>
<th>Bus #</th>
<th>$P_g$</th>
<th>$Q_g$</th>
<th>$P_L$</th>
<th>$Q_L$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.5840</td>
<td>0.5388</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0.8500</td>
<td>0.3458</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0.7160</td>
<td>0.5500</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>1.00</td>
<td>0.35</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>1.25</td>
<td>0.50</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0.90</td>
<td>0.30</td>
</tr>
</tbody>
</table>
With no constraints on DERs (unrealistic), system voltage profile is recovered in 10s.

With constraints on DERs reactive power, system voltage profile is recovered to acceptable limits:

- Voltages on all buses are within $\pm 5\%$ of the nominal value (1 p.u.)
Through sensitivity analysis, it can be shown that bus 6 voltage is the most sensitive to loss of generator 2
  - DER-based reactive power support is only available in this bus
With no constraints on DERs
  - Voltages on buses 3, 5, and 6 are very close to pre-contingency values
  - Voltages on buses 2 and 4 are restored to values within ±5% of the nominal value (1 p.u.)
With constraints on DERs
  - All bus Voltages are restored to values within ±5% of the nominal value (1 p.u.)
The non-adaptive algorithm converges much faster than the adaptive algorithm (not discussed in the presentation).

Both algorithms converge fast enough so the effect on the system is not noticeable.

The adaptive algorithm has advantages if the DER network topology changes.

For larger DER networks, convergence speed of the distributed algorithms might play an important role.

Careful analysis will be needed to justify the algorithm chosen.
Outline

Introduction

Mathematical Preliminaries

Distributed Control with Unconstrained Node Capacity

Distributed Control with Constrained Node Capacity

Voltage Control in Subtransmission Networks

Concluding Remarks
Summary and Path Ahead

- We have presented distributed control algorithms that can be used to determine (in a distributed fashion) the amount of active or reactive power that needs to be provided by distributed active and reactive power resources.
- These strategies have the potential to enable assets already present in distribution systems as active and reactive power support resources.
- We showed how these algorithms can be used to provide reactive power support for voltage control.
- Ongoing work is investigating convergence speed issues in different algorithms for the constrained case.
- Ongoing work is conducting case studies in larger networks, including distribution networks.
- Further work will investigate the algorithms performance in the presence of faults, e.g., broken communication links, and nodes not updating their values.