

Module PE.PAS.U22.5



Composite system reliability

U22.1 Introduction

We briefly introduced composite system reliability in module U17, (also known as HL-II or G&T adequacy analysis), where we identified two attributes that characterize most of the approaches developed to date. These attributes are:

- Method of representing stochastic nature of the operating conditions: By “operating conditions,” we are referring to the basecase network configuration (topology and unit commitment) together with the loading and dispatch. The methods include nonsequential and sequential enumeration.
- Method of representing stochastic nature of contingencies: The methods include contingency enumeration and Monte Carlo.

These methods are summarized in Table U22.1a.

Table U22.1a: HL-II Evaluation approaches

Contingency selection	Operating Conditions	
	Non-sequential	Sequential
Enumeration	Non-sequential, with contingency enumeration	Sequential, with contingency enumeration
Monte-Carlo	Non-sequential, with Monte-Carlo contingency selection	Sequential, with Monte-Carlo contingency selection

GE-MARS simulates operating conditions sequentially; it selects contingencies via Monte Carlo (Table U21.1a lower right-hand cell).

We also provided a generic algorithm for composite system reliability analysis that applies independent of the particular approach taken. We repeat this algorithm here, in Fig. U22.1.

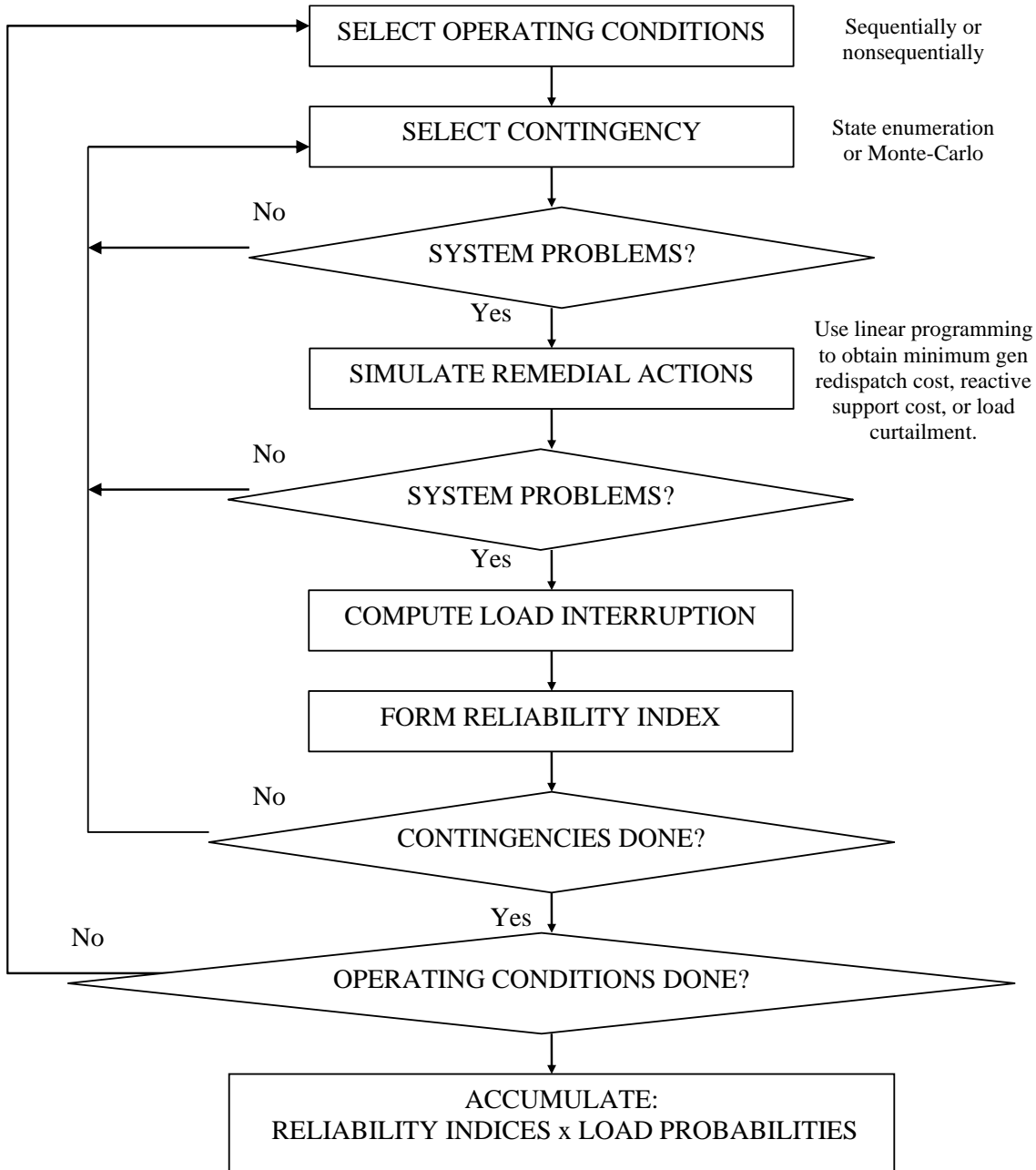


Fig. U22.1: Generic HL-II Evaluation Algorithm

In this module, we begin in Section U22.2 by introducing linear sensitivities which are useful in decreasing computations for

composite reliability analysis. In Section U22.3, we summarize the common contingency enumeration approaches. In Section U22.4, we describe a remedial action algorithm for simulating operator action for overload relief. Section U22.5 describes how to determine bounds on the reliability indices.

The approaches are applicable independent of whether the operating conditions are represented sequentially or nonsequentially. All of the developments in this module focus on line overload problems. Similar algorithms may be developed for low voltage and voltage instability.

U22.2 Linear sensitivities

There are 2 basic linear sensitivities that we will use. Generation shift factors provide us with a fast method of computing changes to circuit flows for changes in real power bus injection. Line outage distribution factors provide a fast method of computing changes to circuit flows when circuits are removed from service.

U22.2.1 DC power flow

The so-called DC power flow is an approximation of the standard AC power flow equations based on the assumptions that circuit resistance is negligible, all bus voltages have magnitudes of 1.0 pu, and the angular separation across a circuit is small enough so that $\cos\delta_{jk}\approx 1$, $\sin\delta_{jk}\approx\delta_{jk}$ when δ_{jk} is measured in radians. With these approximations, and assuming an N bus power system, it is possible to show that $\underline{P}=\underline{B}'\underline{\theta}$ where \underline{P} is the vector of real power injections at N-1 buses, $\underline{\theta}$ is the vector of angles at N-1 buses, and \underline{B}' is the “B-prime” version of the bus admittance matrix for the network being analyzed. Thus, given the network topology (enabling formation of \underline{B}') and the bus injections (enabling formation of \underline{P}), it is possible to compute the bus angles $\underline{\theta}$ from one application of LU decomposition to the linear set of equations represented by $\underline{P}=\underline{B}'\underline{\theta}$. Dimensions of these are:

for \underline{P} , $N-1\times 1$; for $\underline{\theta}$, $N-1\times 1$; and for \underline{B}' ; $N-1\times N-1$

The \underline{B}' matrix may be formed from the Y-bus in the following fashion. Note: For an inductive circuit, the susceptance b is positive based on a *defined* admittance expression of $y=g-jb$.

1. Eliminate real part: Take the imaginary part of all elements in the Y-bus, i.e., for each element, set the real part to 0 and remove the “j” from the imaginary part. This results in matrix \underline{B} .
2. Remove shunt elements: Replace diagonal element \underline{B}_{kk} with the sum of the non-diagonal elements in row k .
3. Negate: Multiply all elements by -1.
4. Remove slack bus: Remove row k and column k for any k .

For example, below is the Y-bus and the \underline{B}' matrix for a network having no resistance and no shunt elements, where bus 1 is assumed as the slack bus.

$$\underline{Y} = j \begin{bmatrix} -30 & 10 & 10 & 10 \\ 10 & -20 & 10 & 0 \\ 10 & 10 & -30 & 10 \\ 10 & 0 & 10 & -20 \end{bmatrix}$$

$$\underline{B}' = \begin{bmatrix} 20 & -10 & 0 \\ -10 & 30 & -10 \\ 0 & -10 & 20 \end{bmatrix}$$

Although the DC power flow equation $\underline{P}=\underline{B}'\underline{\theta}$ provides the ability to compute the bus angles, it does not directly provide the line flows. A systematic method of computing the line flows is:

$$\underline{P}_B = (\underline{D} \times \underline{A}) \times \underline{\theta}$$

where:

- \underline{P}_B is the vector of branch flows. It has dimension of $M \times 1$, where M is the number of branches in the network. Branches are ordered arbitrarily, but whatever order is chosen must also be used in \underline{D} and \underline{A} .

- $\underline{\theta}$ is (as before) the vector of nodal phase angles for buses 2,...N, where N is the number of buses in the network.
- \underline{D} is an $M \times M$ matrix having non-diagonal elements of zeros; the diagonal element in position row k, column k contains the negative of the susceptance of the k^{th} branch.
- \underline{A} is the $M \times N-1$ *node-arc incidence matrix*. It is also called the adjacency matrix, or the connection matrix. Its development requires a few comments.

Development of the node-arc incidence matrix A:

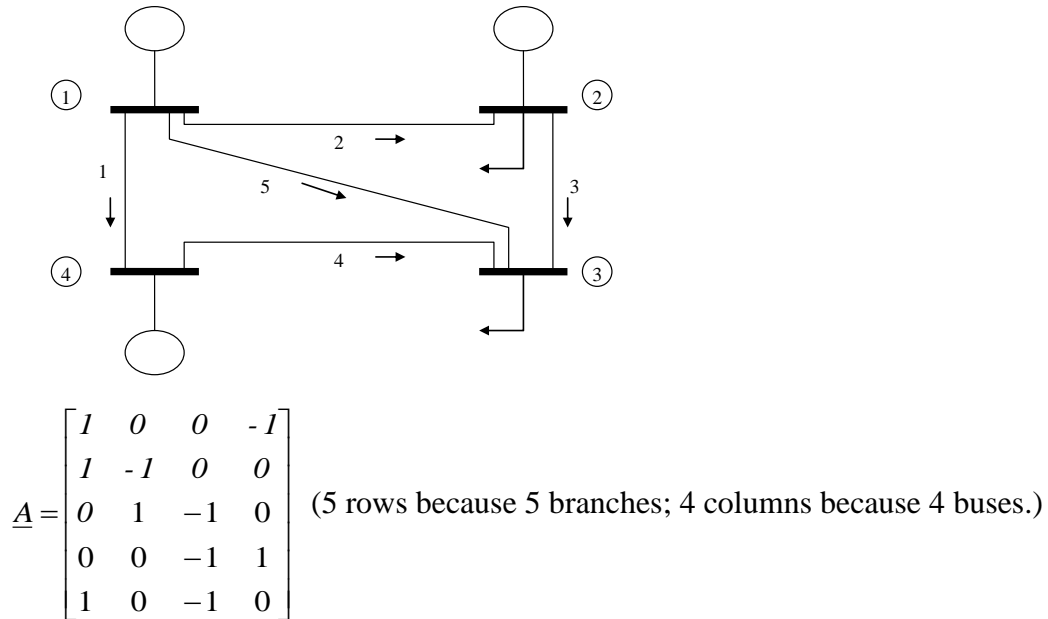
This matrix is well known in any discipline that has reason to structure its problems using a network of nodes and “arcs” (or branches or edges). Any type of transportation engineering is typical of such a discipline.

The node-arc incidence matrix contains a number of rows equal to the number of arcs and a number of columns equal to the number of nodes.

Element (k,j) of \underline{A} is 1 if the k^{th} branch begins at node j, -1 if the k^{th} branch terminates at node j, and 0 otherwise.

A branch is said to “begin” at node j if the power flowing across branch k is defined positive for a direction *from* node j to the other node.

A branch is said to “terminate” at node j if the power flowing across branch k is defined positive for a direction *to* node j from the other node. Below is an illustration which is used in the LP-OPF example given in Appendix 1.



Note that, for DC power flow calculations, i.e., when computing

$$\underline{P} = \underline{B}' \underline{\theta}$$

$$\underline{P}_B = (\underline{D} \times \underline{A}) \times \underline{\theta},$$

matrix \underline{A} is of dimension $M \times N-1$, where M is number of branches and N is number of buses. Thus, it has only $N-1$ columns. This is because we do not form a column for the reference bus, in order to conform to the vector $\underline{\theta}$, which is of dimension $(N-1) \times 1$. This works because the angle being excluded, θ_1 , is zero.

A comprehensive example in Appendix 1 illustrates how to develop \underline{P}_B , \underline{D} , and \underline{A} for the LP-OPF (in which case we use an $M \times N$ dimensional matrix \underline{A}).

U22.2.2 Generation shift factors

The generation shift factor $C_{k,j}$ provides the change in flow on circuit k for a unit change in injection at bus j , i.e.,

$$C_{k,j} = \left. \frac{\Delta P_k}{\Delta P_j} \right|_{\text{Reallocation Policy}} \quad (\text{U22.1})$$

It is assumed that the change in injection at bus j is compensated by a defined “reallocation policy,” the simplest of which is to make an equal and opposite change in injection at a designated swing bus. An expression for obtaining the generation shift factors is derived in Appendix 2 (where $C_{k,j}$ is denoted as $t_{b,k}$).

U22.2.3 Line outage distribution factors

The line outage distribution factor $d_{k,j}$, provides the change in flow on circuit k , denoted by Δf_k , due to outage of circuit j having flow f_j^0 , and is given by

$$d_{k,j} = \frac{\Delta f_k}{f_j^0} \quad (\text{U22.2})$$

The development of the generation shift factor and the line outage distribution factor are based on the DC power flow approximation and are given in [1].

Reference [5] shows how to include ramp rates into the above sensitivity factors.

An expression for obtaining line outage distribution factors is derived in Appendix 3.

U22.3 Contingency enumeration

The simplest approach to contingency selection is to assess all contingencies; in this approach, one needs only to develop an appropriate scheme to enumerate them so that none are missed. This approach is, however, computationally expensive, except for small systems.

A second approach is to simply limit the contingencies. Here, one can think of the following methods:

- Use a list: Provide a contingency list that prescribes exactly the contingencies of interest. This approach has merit in some cases where study objectives so dictate.

- Limit the order: For example, we may require analysis for all N-1 contingencies, or we may go one deeper and require analysis for all N-1 and N-2 contingencies. The problem with this approach is that it may spend a great deal of computation on contingencies that have little effect on the resulting indices, or, on the other hand, it may miss some higher-order contingencies that do significantly contribute to the reliability indices.
- Limit the probability: Here, one analyzes all contingencies for which their probability exceeds a certain threshold. This is slightly more refined than limiting the order, and it is subject to the same problems.
- Based on severity: The principle here is to avoid analyzing contingencies that do not result in a problem. Some type of screening approach is normally involved, using either the DC load flow or the linear sensitivities. It is convenient to also associate with each contingency k screened a performance index PI_k given by:

$$PI_k = \sum_{j=1}^{N_b} \left[\frac{P_j}{P_{j,\max}} \right]^n \quad (\text{U22.3})$$

where n is positive, even, and relatively large (e.g., n=10). This means that any individual term in the summation of (U22.3), $[P_j/P_{j,\max}]^n$, will be very small if $P_j < P_{j,\max}$ and very large if $P_j > P_{j,\max}$. Thus, PI_k is guaranteed to be at least 1.0 if there is even one overload following contingency k. More important, PI_k is a very good metric for comparing the severity of one contingency against another.

The so-called wind-chime scheme [2,3] combines one of the first three approaches with the last approach. In the wind-chime scheme, two sets of contingencies are selected – the primary set and the secondary set. These two sets need not be different and in fact, most often, they are the same. For purposes of generality,

however, we will assume in our analysis that they are different. Any of the first three contingency selection methods may be used to make these selections.

Then, each set of contingencies is screened, and the performance indicator PI_k is computed for each contingency in each set. Then the contingencies are ranked from most severe to least severe.

- Denote the primary contingencies as $P1, P2, \dots$, with $P1$ more severe than $P2$.
- Denote the secondary contingencies as $S1, S2, \dots$, with $S1$ more severe than $S2$.

Then $N-2$ contingencies $(P1, S) S=S1, \dots$, are analyzed until we find three consecutive contingencies with no violation.

Then $N-2$ contingencies $(P2, S), S=S1, \dots$, are analyzed until we find three consecutive contingencies with no violation.

And we continue analyzing each set of $N-2$ contingencies (P,S) , with primary contingency P and secondary contingency $S=S1, \dots$, until we have three consecutive sets of contingencies $(PN,S), (PN+1,S), (PN+2,S), S=S1, S2, S3$, with no violation. The algorithm stops at this point.

Fig. U22.2 illustrates the windchime algorithm [4]; we observe that

- for $P3$, we stop testing secondary contingencies after $S8$;
- for $P4$, we stop testing secondary contingencies after $S8$;
- for $P5$, we stop testing secondary contingencies after $S4$;
- for $P6$, we stop testing secondary contingencies after $S5$;
- for $P7$, we stop testing secondary contingencies after $S3$;
- for $P8$, we stop testing secondary contingencies after $S3$;
- for $P9$, we stop testing secondary contingencies after $S3$;
- Because we had three consecutive no-violation primary contingencies in a row $(P7, P8, P9)$, we are done.

The identified violating contingencies are the ones of concern. In Fig U22.2, it is the lavender colored ones; we performed 56 tests (green & lavender) out of a possible 132 tests to find 24 violations.

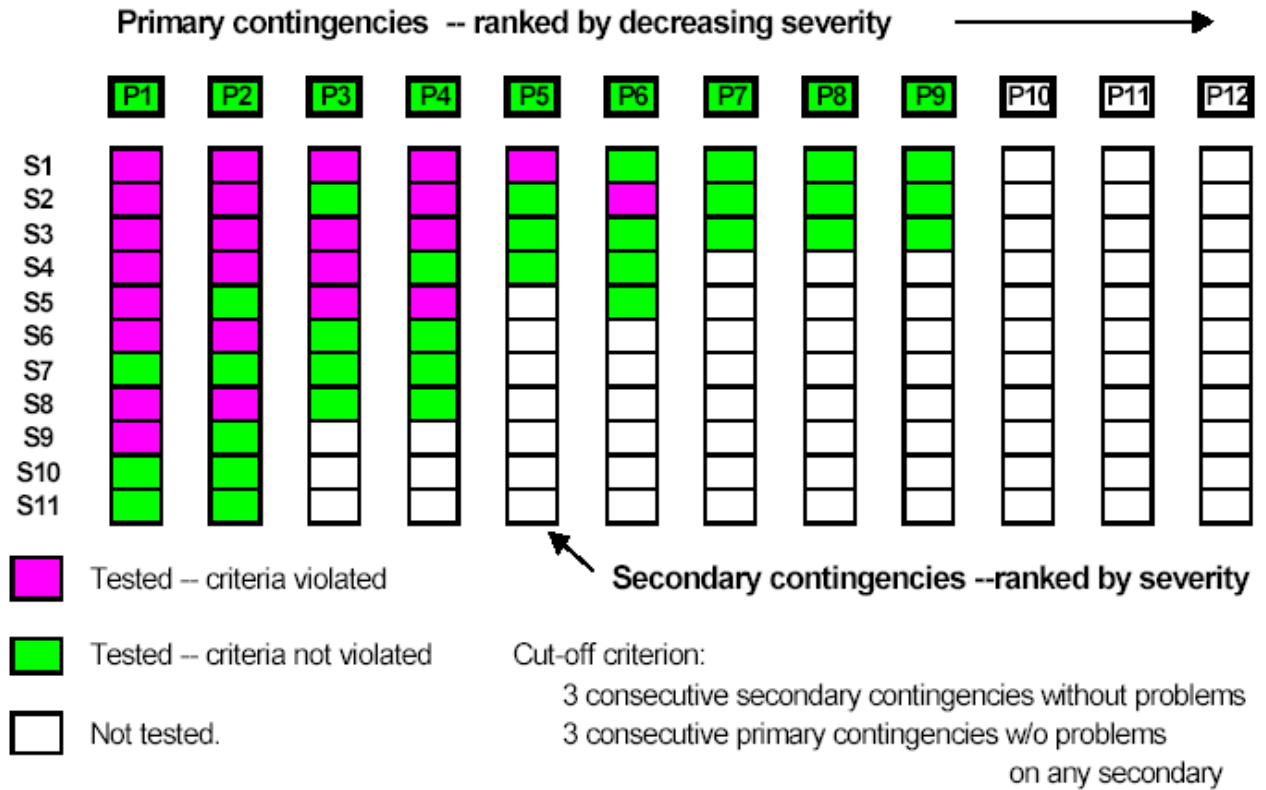


Fig. U22.2: Illustration of the windchime algorithm [4]

It is reasonable to use this algorithm for purposes of N-3 contingency screening as well, where in addition to primary and secondary contingency sets, one would also identify a tertiary contingency set ranked in order of decreasing severity T1, T2, T3,..., where,

- for T1, one repeats the above N-2 contingency approach;
- for T2, one repeats the above N-2 contingency approach;
- ...

until one performs three sets of TN contingencies without seeing a violation. This essentially establishes a 3-dimensional volume comprised of multiple slices each of which appear as Fig. U22.2. One could perform N-2 or N-3 contingency screening for planning or for on-line security assessment. This approach can be combined with the approach of N-k contingency selection based on probability, given in [6] and/or in [7].

One should realize that not all N-k ($k \geq 2$) have performance criteria associated with them, and so the windchime approach may find contingencies which result in flow violations, but (although they are “good to know”), because they are not specified according to the criteria of the North American Electric Corporation (NERC) [8], they do not motivate operational or planning action. From [8], summarized in Table U22.1b, N-k ($k \geq 2$) contingencies that do motivate operational or planning actions are identified as:

- P3 and P6, N-2 contingencies, but note that these require “system adjustment” between contingencies;
- P4, N-k ($k \geq 2$) contingencies, but note that these must occur as a result of a faulted component followed by a stuck breaker;
- P5 N-k ($k \geq 2$) contingencies, but note that these must occur as a result of a faulted component followed by a relay failure;
- P7 N-k ($k \geq 2$) contingencies, but note that these must occur as a result of a common structure failure.

However, we will include these “out of criteria” events into our estimation of the resource adequacy indices.

Table U22.1b: Summary of Planning Events P0-P7 from NERC Standard TPL-001-5 [8]

Category	Initial Condition	Event
P0	Normal System	None
P1, Single Contingency	Normal System	Loss of one of the following:
		1. Generator
		2. Transmission Circuit
		3. Transformer
		4. Shunt Device
P2, Single Contingency	Normal System	1. Opening of a line section without a fault
		2. Bus section fault
		3. Internal Breaker Fault (non-bus-tie breaker)
		4. Internal Breaker Fault (bus-tie breaker)
P3, Multiple Contingency	Loss of generator unit followed by system adjustments	Loss of one of the following:
		1. Generator
		2. Transmission Circuit
		3. Transformer
		4. Shunt Device
P4, Multiple Contingency (Fault plus stuck breaker)	Normal System	5. Single pole of a DC line
		Loss of multiple elements caused by a stuck breaker (non-bus-tie breaker) attempting to clear a fault on one of the following:
		1. Generator
		2. Transmission circuit
		3. Transformer
		4. Shunt device
		5. Bus section
6. Loss of multiple elements caused by a stuck breaker (bus-tie breaker) attempting to clear a fault on the associated bus		

Category	Initial Condition	Event
P5, Multiple Contingency, (Fault plus relay failure to operate)	Normal System	Delayed fault clearing due to the failure of a non-redundant relay protecting the faulted element to operate as designed, for one of the following:
		1. Generator
		2. Transmission Circuit
		3. Transformer
		4. Shunt Device
P6, Multiple Contingency, (Two overlapping singles)	Loss of one of the following followed by system adjustments	Loss of one of the following:
		1. Transmission circuit
		2. Transformer
	3. Shunt Device	
4. Single pole of a DC line	4. Single pole of a DC line	
P7, Multiple Contingency, (Common structure)	Normal System	The loss of:
		1. Any two adjacent (vertically or horizontally) circuits on common structure
		2. Loss of bipolar DC line

U22.4 Remedial actions

Corrective or remedial actions performed automatically should be built into the simulation of the contingency. Examples of such actions include direct or automatic load tripping, generation rejection, and turbine-governor response to load imbalance. For these actions, it is generally known what action will take place, for what conditions, and at what time.

Corrective or remedial actions performed by the operator are quite uncertain, as they may depend on human decision-making and responsiveness. We can, however, make certain assumptions regarding operator action, and based on these assumptions, develop appropriate simulation algorithms.

The fundamental assumptions we make regarding operator action are that the operator will make the minimum control effort in order to alleviate any overloads. (One may also formulate based on the assumption that the operator will make the least-cost control adjustments to alleviate any overloads; such an assumption can be accommodated in the formulation to follow.)

The most comprehensive algorithm available for implementing remedial actions is the optimal power flow (OPF). The advantage of the OPF is that it provides the ability to simulate operator action for overload, low voltage, and voltage instability. However, it is computationally expensive, although computing power today via combination of high-end and parallel processing makes it quite practical, even for large systems. We provide a simpler approach which only allows remedial actions for overloads. One may consider it to be an approximation to the OPF.

Define ΔP_{gi}^+ and ΔP_{gi}^- to be an increase or decrease, respectively, in the generation at bus i , and ΔP_{dj} to be the load curtailment at bus j , where all three variables are always nonnegative.

Also define K_i and K_j to be weights associated with the bus i generation change and the bus j load curtailment, respectively.

- The bus i generation change weight K_i should be chosen based on dispatchability (infinity, i.e., very large, if not dispatchable) and ramp rates (the higher the ramp rate, the smaller the K_i) for each plant, indicating that the faster the plant can move, the more we will rely on it to implement the desired remedial action as soon as possible.
- The bus j load curtailment weight K_j should be chosen based on whatever contracts are in place with the loads. It is typical that $K_j \gg K_i$ indicating that load curtailment is typically much less desirable than generation redispatch.

We can then write a linear program as follows:

$$\min \sum_{i=1}^{n_g} K_i \Delta P_{gi}^+ + \sum_{i=1}^{n_g} K_i \Delta P_{gi}^- + \sum_{j=1}^{n_d} K_j \Delta P_{dj} \quad (\text{U22.4a})$$

subject to:

$$\sum_{i=1}^{n_g} \Delta P_{gi}^+ - \sum_{i=1}^{n_g} \Delta P_{gi}^- + \sum_{j=1}^{n_d} \Delta P_{dj} = 0 \quad (\text{U22.4b})$$

$$P_k^0 + \sum_{i=1}^{n_g} C_{k,i} \Delta P_{gi}^+ - \sum_{i=1}^{n_g} C_{k,i} \Delta P_{gi}^- + \sum_{j=1}^{n_d} C_{k,i} \Delta P_{dj} \leq P_k^{\max} \quad \forall k \quad (\text{U22.4c})$$

$$0 \leq \Delta P_{dj} \leq P_{dj}^0 \quad \forall j \quad (\text{U22.4d})$$

$$0 \leq \Delta P_{gi}^+ \leq P_{gj}^{\max} - P_{gi}^0 \quad \forall i \quad (\text{U22.4e})$$

$$0 \leq \Delta P_{gi}^- \leq P_{gi}^0 \quad \forall i \quad (\text{U22.4f})$$

where ΔP_k^0 is the flow on circuit k after the contingency, and $C_{k,i}$ is the generation shift factor giving the sensitivity of the circuit k flow to the bus j injection.

The formulation above will identify the load interruption necessary to relieve all overloads. Notice that the above formulation is not a security-constrained OPF, i.e.,

- it operates on the *post-contingency state* to *correct* post-contingency overloads,
- whereas the security-constrained OPF operates on the *pre-contingency state* to *prevent* post-contingency overloads.

An approach is provided in App. 1 that implements an LP-OPF. This approach minimizes redispatch cost rather than control effort.

U22.5 Bounding the failure probability

Given that we are capable of performing the contingency selection and then analyzing each contingency selected to determine whether it is a failure state or not, the remaining issue is to determine the reliability indices. Specifically, we desire to compute the LOLP.

However, we face the basic problem that we know whether a state is a failure state or not for only the states that we have information. Based on the windchime contingency selection approach described previously, this consists of all N-1 and N-2 contingency states. (Actually, we only analyze a subset of N-2 contingency states, but the windchime approach enables one to conclude with high confidence that all unanalyzed N-2 contingency states are success states (i.e., no loss of load), and so we can say that we have information for all N-1 and N-2 states).

However, we still must consider the N-k states for $k > 2$. Although the probability of most of these states is quite small, the fact that there are so many of them suggests that they may comprise a significant percentage of the desired LOLP index.

We may determine how close our LOLP estimate is to the actual LOLP by using a bounding approach. In this approach, we will determine a lower bound together with an upper bound on the failure probability. The width of the bounded interval provides an indication of how close our LOLP estimate is.

U22.5.1 The binary tree

We illustrate the method based on understanding of a binary tree. We motivate the concept of a binary tree using a simple example. Consider the 3-component system of Fig. U22.3.

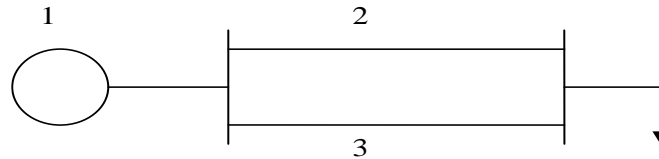


Fig. U22.3: 3 component system for illustrating bounding approach

We denote the availability and the failure probability (FOR) of each component as p_j and q_j , respectively. Thus the total number of states of this system is $2^3=8$. These states are summarized in Table U22.2.

Table U22.2: Summary of states for example system

State s	Component In (1) or Out (0)			Pr[s]
	1	2	3	
1	1	1	1	$p_1p_2p_3$
2	1	1	0	$p_1p_2q_3$
3	1	0	1	$p_1q_2p_3$
4	1	0	0	$p_1q_2q_3$
5	0	1	1	$q_1p_2p_3$
6	0	1	0	$q_1p_2q_3$
7	0	0	1	$q_1q_2p_3$
8	0	0	0	$q_1q_2q_3$

The binary tree for this system represents the different states of Table U22.2 and is illustrated in Fig. U22.3.

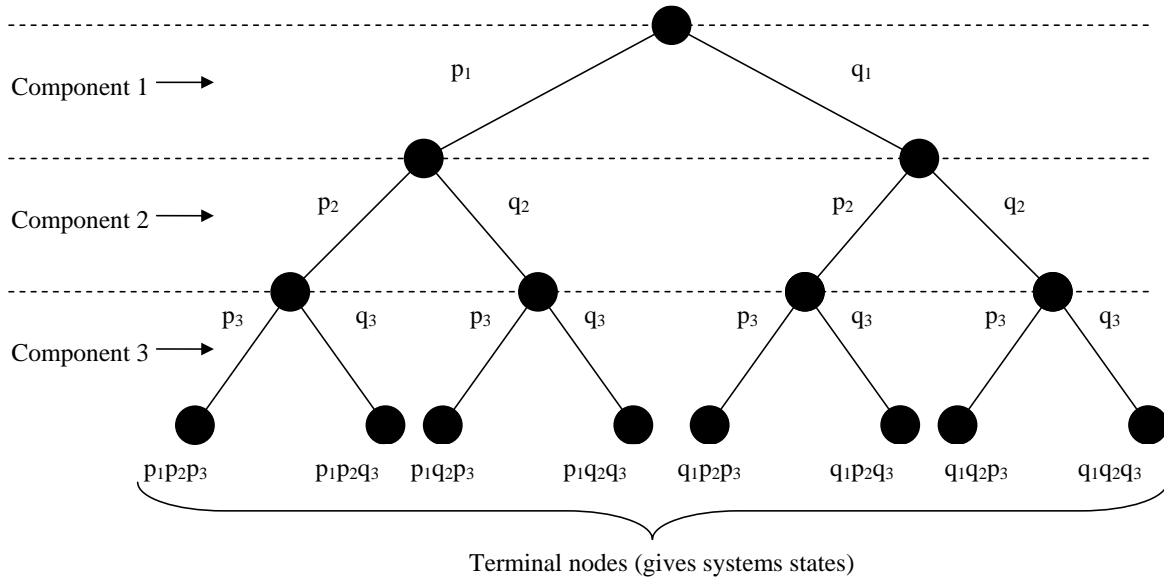


Fig. U22.3: Binary tree for illustrating system states

Our goal is to identify which terminal nodes correspond to the failed states. If we can do that, then LOLP is computed simply as:

$$LOLP = \sum_{s \in \text{FailedStates}} \text{Pr}[s] \quad (\text{U22.5})$$

The problem is, of course, that the tree can become very large. For example, if we have just 100 components, which would of course be a very small system, we would have $2^{100} = 1.27 \times 10^{20}$ terminal nodes (or states).

U22.5.2 Obtaining a lower bound

We turn now to a general case corresponding to a system with N components; we assume that each one is modeled with 2 states.

We define $\Delta \text{Pr}_j[\text{F}]$ as the additional (and previously uncounted) failure probability we can add due to knowledge that contingency j fails (results in loss of load).

Let's assume that contingency j corresponds to loss of circuit k and that we know it is a failed contingency, i.e., it results in loss of load.

We assume a mapping between the contingency identification j and the outaged component k ; this might not be necessary (i.e., we could assume contingency j implies loss of component j), but it maintaining this mapping may allow broader contingency definition; e.g., contingency j might be a P7, loss of multiple circuits on the same tower. Key is that each contingency j must have a specific probability couple (q_k, p_k) corresponding to *outaged* and *not outaged*. The state corresponding to all other ccts in (besides cct k) is the most probable state of those which stem from the node corresponding to q_k . Then the additional failure probability due to knowledge that contingency j fails, $\Delta Pr_j[F]$, can be expressed as

$$\Delta Pr_j[F] = q_k \prod_{\substack{i=1 \\ i \neq k}}^N p_i + x \tag{U22.6}$$

where x is the probability of all other failed states stemming from the q_k branch. *Since each state probability in x will have at least 2 terms in q (one for q_k and at least one other one, since the state corresponding to all other terms in p is the first expression in U22.6), x should be rather small.*

Now represent cct k at the top of the tree, as shown in Fig. U22.4.

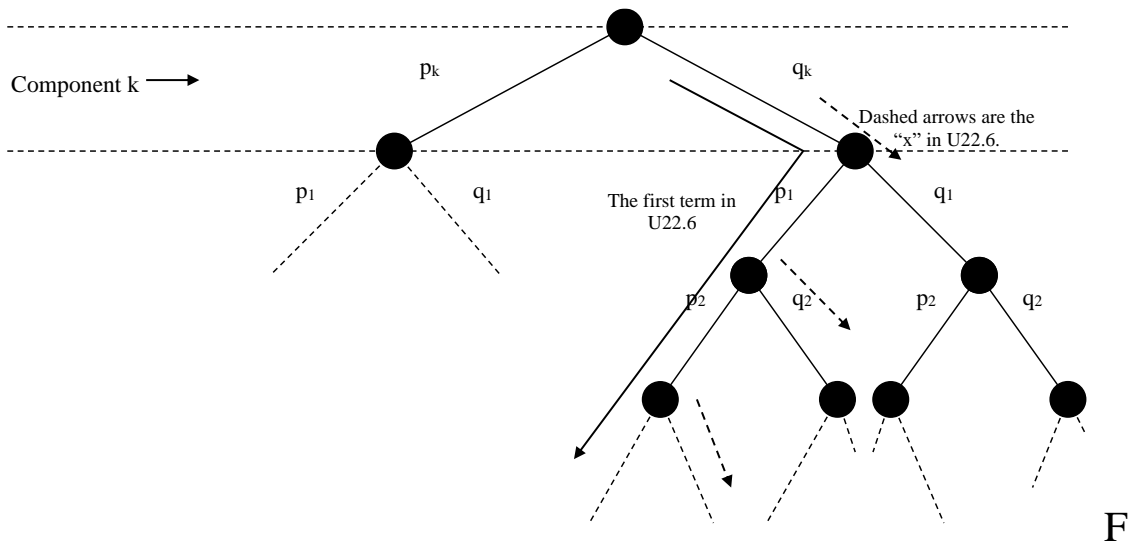


fig. U22.4: Binary tree for illustrating system states with circuit k at the tree-top

If (U22.6) is true, then it must be true that the right-hand-side of

$$\Delta Pr_j[F] \geq q_k \prod_{\substack{i=1, \\ i \neq k}}^N p_i = \Delta_{APP} Pr_j[F], \quad (\text{U22.7})$$

denoted as $\Delta_{APP} Pr_j[F]$ (“APP” for “approximate”) is a lower bound on the actual $\Delta Pr_j[F]$.

If we compute (U22.7) for all failed contingencies, then we may obtain a lower bound on the LOLP according to:

$$LOLP \geq \sum_{j \in \text{Failed Contingencies}} \Delta_{APP} Pr_j[F] \quad (\text{U22.8})$$

U22.5.3 Obtaining an upper bound

Obtaining a lower bound does us little good unless we can also obtain an upper bound. To do this, consider that we are able to identify contingency j , corresponding to loss of circuit k , is a success, i.e., no loss of load. We again represent circuit k at the top of the tree, as in Fig. U22.4.

Here, we know that at least one of the terminal nodes will be a success – the one with all components in except for circuit k . Therefore we may write that:

$$\Delta Pr_j[S] \geq q_k \prod_{\substack{i=1, \\ i \neq k}}^N p_i = \Delta_{APP} Pr_j[S] \quad (\text{U22.9})$$

This provides us with a lower bound on the probability of not losing load (in contrast to the lower bound on the probability of losing load which we obtained in (U22.8)). However, we want an upper bound on the probability of losing load!

To get this, multiplying both sides of (U22.9) by -1 (and reversing the sense of the inequality), and then add 1 to both sides results in:

$$1 - \Delta \Pr_j[S] \leq 1 - q_k \prod_{\substack{i=1, \\ i \neq k}}^N p_i = 1 - \Delta \Pr_j[S]_{APP} \quad (\text{U22.10})$$

Now consider that $1 - \Delta \Pr_j[S]$ is actually the contingency j probability of failure. Thus, we see that the right-hand-side of (U22.10) provides an *upper bound* on the probability of failure for all contingency states stemming from q_k . Thus, we have that

$$LOLP \leq \sum_{j \in \text{Success States}} \Delta \Pr_j[S]_{APP} \quad (\text{U22.11})$$

U22.5.4 Algorithm for computing lower and upper LOLP bounds

The following is an algorithm for computing the lower and upper bounds on LOLP for a given system, with N components.

1. $\Delta \Pr[F]=0$, $\Delta \Pr[S]=0$
2. For $j=1, N$
 - a. If j =failed, then
 - i. Compute failure probability approximation

$$\Delta \Pr_j[F]_{APP} = q_k \prod_{\substack{i=1, \\ i \neq k}}^N p_i$$

- ii. Sum to total failure probability approximation:

$$\Delta \Pr[F] = \Delta \Pr[F] + \Delta \Pr_j[F]_{APP}$$

- b. ElseIf j =success, then

- i. Compute success probability approximation

$$\Delta \Pr_{APP}^j[S] = q_k \prod_{\substack{i=1, \\ i \neq k}}^N p_i$$

- ii. Sum to total failure probability approximation:

$$\Delta \Pr[S] = \Delta \Pr[S] + \Delta \Pr_{APP}^j[S]$$

Then we have that the actual LOLP is bounded as illustrated in Fig. U22.5.

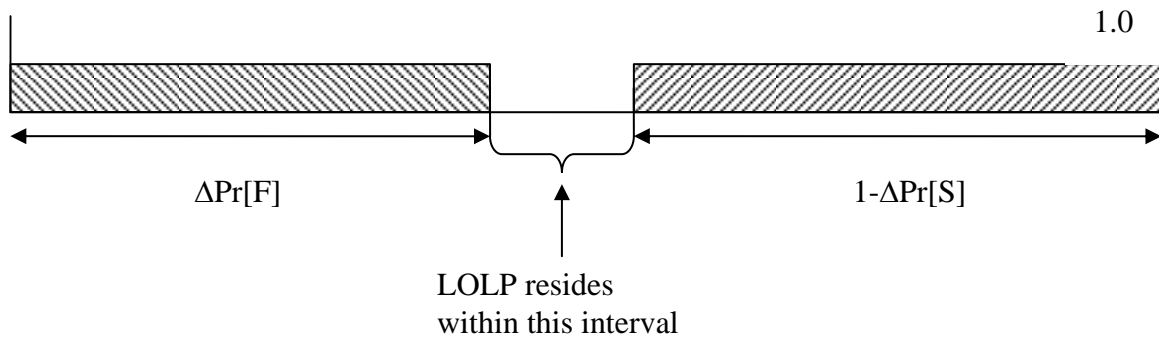


Fig. U22.5: Illustration of bounds on LOLP

Appendix 1: LP-OPF Example

This appendix provides an LP-OPF formulation which minimizes a linearized cost function subject to a set of equality constraints (DC load flow equations and line flow equations) and a set of inequality constraints (on generation limits and on branch flows). This appendix also illustrates how to form the \underline{P}_B vector and the \underline{D} and \underline{A} matrices.

$$\min \sum_{k=1}^{N_g} \sum_{j=1}^{M_j} s_{kj} P_{kj} = \underline{s}^T \underline{P}'$$

s.t.

$$\underline{B}'\underline{\theta} + \underline{P}_D = \underline{T}\underline{P}' + \underline{P}_{\min} \quad \text{DC Power Flow Equation}$$

$$\underline{P}_B = (\underline{D} \times \underline{A}) \times \underline{\theta} \quad \text{Line Flow Equation}$$

$$\underline{P}'_{\min} \leq \underline{P}' \leq \underline{P}'_{\max} \quad \text{Generator Constraints}$$

$$-\underline{P}_{B \max} \leq \underline{P}_B \leq \underline{P}_{B \max} \quad \text{Line Flow Constraints}$$

$$-\pi \leq \underline{\theta} \leq \pi \quad \text{Bus Angle Constraints}$$

\underline{D} is an $M \times M$ matrix that will have its diagonal as the admittances of the lines.

M = number of branches

\underline{A} is an $M \times (N-1)$ node-arc incidence matrix that describe the flow from bus i to bus j .

N = number of busses

\underline{B}' is the DC power flow matrix of dimension $(N-1) \times (N-1)$, where N is the number of buses in the network, obtained as follows:

1. Replace diagonal element \underline{B}'_{kk} with the sum of the non-diagonal elements in row k . Alternatively, subtract b_k (the shunt term) from B_{kk} , and multiply by -1.
2. Multiply all off-diagonals by -1.

$\underline{\theta}$ is the vector of nodal phase angles for buses $2, \dots, N$

\underline{P}_D is the Power demanded at each bus.

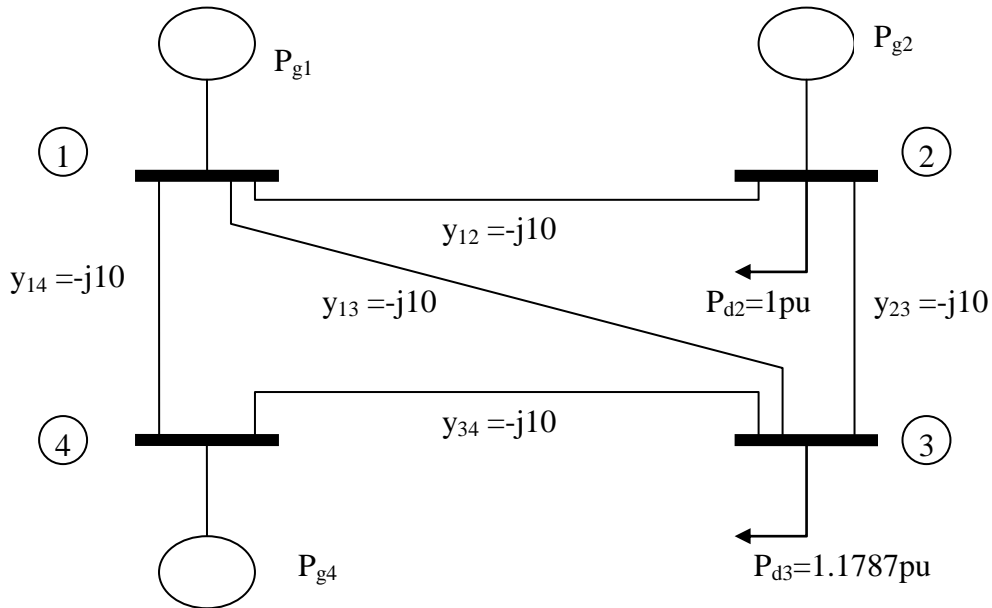
We formulate the above into the following:

$\underline{A}_{eq}\underline{x}=\underline{b}_{eq}$ is the equation that will be used to do this.

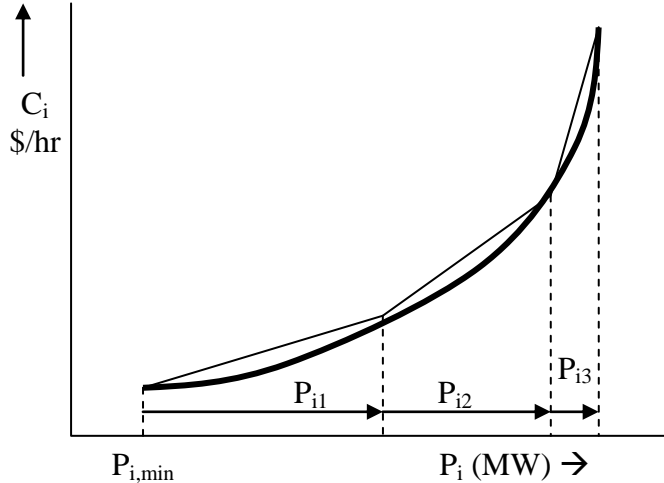
Aeq matrix – The number of columns is the sum of the number of generators, lines, and buses. The number of rows is the sum of the number of lines and buses.

Example:

This is the basic system that will be used for the example.



This graph gives the basic representation of the cost curves division for a generator



We will use linearized representation, but to begin with, we assume quadratic representation for the three cost curves of the example.

$$C_1(P_1) = 0.00533P_1^2 + 11.669P_1 + 213.1$$

$$C_2(P_2) = 0.00889P_2^2 + 10.333P_2 + 200$$

$$C_3(P_3) = 0.00741P_3^2 + 10.833P_3 + 240$$

In the implementation, we expect to receive the curves in the form of break points.

Break points for PW linear approximation of cost curves given as MW, Cost

Unit	BP#1 (unit min)	BP#2	BP#3	BP#4 (unit max)
1	50.0, 809.9	100, 1433	160, 2217	200, 2760
2	37.5, 5325	70, 9699	130, 1693	150, 1950
3	45.0, 742.5	90, 1275	140, 1902	180, 2430

The slope is then computed for each of the segments (1-2), (2-3), and (3-4) for each curve. This is done by the following equation:

$$s_{kj} = \frac{Cost_j - Cost_k}{MW_j - MW_k} \quad \text{and substituting numbers: } s_{11} = \frac{1433 - 809.9}{100 - 50} = 12.46$$

This process is done for each section and each generator. The slopes of the piecewise linear approximations of the cost curves are shown below:

Unit, i	s_{i1}	s_{i2}	s_{i3}
1	12.46	13.07	13.58
2	11.29	12.11	12.82
3	11.83	12.54	13.20

$$K_1(P_{11}, P_{11}, P_{11}) = C_1(P_{1,\min}) + s_{11}P_{11} + s_{12}P_{12} + s_{13}P_{13}$$

$$K_2(P_{21}, P_{22}, P_{23}) = C_2(P_{2,\min}) + s_{21}P_{21} + s_{22}P_{22} + s_{23}P_{23}$$

$$K_3(P_{31}, P_{32}, P_{33}) = C_3(P_{3,\min}) + s_{31}P_{31} + s_{32}P_{32} + s_{33}P_{33}$$

The limits on generation for each section must be calculated. This is given by:

$$P_{kj} = MW_j - MW_k, \quad \text{for example: } P_{11} = 100 - 50 = 50MW$$

This process is done for each section of the curve for each generator. Representing the vector of piecewise linear generation values as P' , we have:

$$\underline{P}' = \begin{bmatrix} P_{11} \\ P_{12} \\ P_{13} \\ P_{21} \\ P_{22} \\ P_{23} \\ P_{41} \\ P_{42} \\ P_{43} \end{bmatrix} \begin{array}{l} \left. \begin{array}{l} \\ \\ \\ \end{array} \right\} \text{Unit 1} \\ \left. \begin{array}{l} \\ \\ \end{array} \right\} \text{Unit 2} \\ \left. \begin{array}{l} \\ \\ \\ \end{array} \right\} \text{Unit 4} \end{array} \rightarrow \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \leq \underline{P}' \leq \begin{bmatrix} 0.50 \\ 0.60 \\ 0.40 \\ 0.325 \\ 0.60 \\ 0.20 \\ 0.45 \\ 0.50 \\ 0.40 \end{bmatrix}$$

In the above, the generation is represented in per-unit on 100 MVA base.

Next, the T matrix must be built. This is a matrix that operates on the piecewise linear generation vector P' to obtain generation injections, i.e., it sums the generation variables for each unit to obtain the total generation at that unit. For this case, T will be 4 rows high, and 9 columns long.

$$T = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

$$\underbrace{\begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{bmatrix}}_{\# \text{ of buses}} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} P_{11} \\ P_{12} \\ P_{13} \\ P_{21} \\ P_{22} \\ P_{23} \\ P_{41} \\ P_{42} \\ P_{43} \end{bmatrix}$$

The next step is to get the respective parts for the DC Power Flow ($\underline{B}'\theta + \underline{P}_D = \underline{T}\underline{P}' + \underline{P}_{\min}$). We first obtain the B' matrix.

Neglecting resistance, the Y-bus is:

$$Y = j \begin{bmatrix} B_{11} & B_{12} & B_{13} & B_{14} \\ B_{21} & B_{22} & B_{23} & B_{24} \\ B_{31} & B_{32} & B_{33} & B_{34} \\ B_{41} & B_{42} & B_{43} & B_{44} \end{bmatrix}$$

$$= j \begin{bmatrix} b_1 + b_{12} + b_{13} + b_{14} & -b_{12} & -b_{13} & -b_{14} \\ -b_{21} & b_2 + b_{21} + b_{23} + b_{24} & -b_{23} & -b_{24} \\ -b_{31} & -b_{32} & b_3 + b_{31} + b_{32} + b_{34} & b_{34} \\ -b_{41} & -b_{42} & -b_{43} & b_4 + b_{41} + b_{42} + b_{43} \end{bmatrix}$$

From the above, we obtain the B' matrix from the Y-bus, as follows:

5. Remove the "j" from the Y-bus.
6. Replace diagonal element \underline{B}'_{kk} with the sum of the non-diagonal elements in row k. Alternatively, subtract b_k (the shunt term) from B_{kk} , and multiply by -1.
7. Multiply all off-diagonals by -1.

Comparison of the numerical values of the Y-bus with the numerical values of the B' matrix for our example will confirm the above procedure:

$$\underline{Y} = j \begin{bmatrix} -30 & 10 & 10 & 10 \\ 10 & -20 & 10 & 0 \\ 10 & 10 & -30 & 10 \\ 10 & 0 & 10 & -20 \end{bmatrix} \quad \underline{B}' = \begin{bmatrix} 30 & -10 & -10 & -10 \\ -10 & 20 & -10 & 0 \\ -10 & -10 & 30 & -10 \\ -10 & 0 & -10 & 20 \end{bmatrix}$$

Another way to remember the B' matrix is to observe that since its non-diagonal elements are the negative of the Y-bus matrix, the B' non-diagonal elements are susceptances..

Limits on bus angles are: $-\pi \leq \theta \leq \pi$

\underline{P}_D is the vector of load values, in per unit: $\underline{P}_D = \begin{bmatrix} P_{D1} \\ P_{D2} \\ P_{D3} \\ P_{D4} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1.1787 \\ 0 \end{bmatrix} p.u.$

\underline{P}_{\min} is the minimum amount of power that each generator can produce.

\underline{P}_{\max} is the maximum amount of power that each generator can produce.

$$\underbrace{\begin{bmatrix} 0.50 \\ 0.375 \\ 0 \\ 0.45 \end{bmatrix}}_{P_{\min}} \leq \begin{bmatrix} P_{g1} \\ P_{g2} \\ P_{g3} \\ P_{g4} \end{bmatrix} \leq \underbrace{\begin{bmatrix} 0.200 \\ 0.150 \\ 0 \\ 0.180 \end{bmatrix}}_{P_{\max}}$$

We account for the lower limit in the DC power flow equation through P_{\min} . Expressing the DC Power Flow Equation ($\underline{B}'\underline{\theta} + \underline{P}_D = \underline{T}\underline{P}' + \underline{P}_{\min}$).

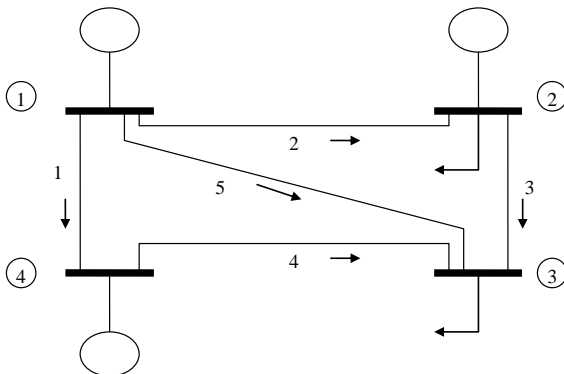
$$\begin{bmatrix} 30 & -10 & -10 & -10 \\ -10 & 20 & -10 & 0 \\ -10 & -10 & 30 & -10 \\ -10 & 0 & -10 & 20 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 1.0 \\ 1.1787 \\ 0 \end{bmatrix} = \underline{TP}' + \begin{bmatrix} 0.5 \\ 0.375 \\ 0 \\ 0.45 \end{bmatrix}$$

The Line Flow Equations are represented next.

The D-matrix is formed by placing the admittances of each branch along the diagonal of an $M \times M$ matrix, where $M=5$.

$$\underline{D} = \begin{bmatrix} 10 & 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 10 & 0 & 0 \\ 0 & 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 0 & 10 \end{bmatrix}$$

The node-arc incidence matrix is found by first defining a direction of power flow in the system.



A branch is said to “begin” at node j if the power flowing across branch k is defined positive for a direction *from* node j to the other node.

A branch is said to “terminate” at node j if the power flowing across branch k is defined positive for a direction to node j from the other node.

With this definition, we can express the node-arc-incidence matrix as:

$$\underline{A} = \begin{bmatrix} 1 & 0 & 0 & -1 \\ 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & -1 & 1 \\ 1 & 0 & -1 & 0 \end{bmatrix} \quad (\text{Has four columns because there are four bus angles})$$

We will next find $\underline{D} \times \underline{A}$.

$$\underline{D} \times \underline{A} = \begin{bmatrix} 10 & 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 10 & 0 & 0 \\ 0 & 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 0 & 10 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -1 \\ 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & -1 & 1 \\ 1 & 0 & -1 & 0 \end{bmatrix} = \begin{bmatrix} 10 & 0 & 0 & -10 \\ 10 & -10 & 0 & 0 \\ 0 & 10 & -10 & 0 \\ 0 & 0 & -10 & 10 \\ 10 & 0 & -10 & 0 \end{bmatrix}$$

Next, the line flow constraints will be stated. ($-\underline{P}_{B \max} \leq \underline{P}_B \leq \underline{P}_{B \max}$)

To begin with, we will represent unconstrained transmission using very large limits.

$$\begin{bmatrix} -500 \\ -500 \\ -500 \\ -500 \\ -500 \end{bmatrix} \leq \underline{P}_B \leq \begin{bmatrix} 500 \\ 500 \\ 500 \\ 500 \\ 500 \end{bmatrix} \quad \text{per-unit.}$$

Bus angle (theta) constraints are $-\pi$ to π (360 degrees)

The A_{eq} matrix will now be exemplified and built.

In this example, we have 4 buses, 5 branches, and 9 generation sections. So that sums up to 18, so that is the number of columns in the A_{eq} matrix. For the number of rows, we have 5 branches and 4 buses, which sums up to 9 and is the number of rows in this matrix.

- The last 4 columns on the first 5 rows is the $D \times A$ matrix.
- The first 14 columns on the first 5 rows are the elements in the line flow equations that multiply the variables $P_{11}, P_{12}, P_{13}, P_{21}, P_{22}, P_{23}, P_{41}, P_{42}, P_{43}, P_{B1}, P_{B2}, P_{B3}, P_{B4}$, and P_{B5} . $P_{11}, P_{12}, P_{13}, P_{21}, P_{22}, P_{23}, P_{41}, P_{42}, P_{43}$ are set to zero because they are the

generations not the line flow. $P_{B1}, P_{B2}, P_{B3}, P_{B4},$ and P_{B5} are set to -1 to say that there is a branch in a specific location.

- The first 9 columns of the last 4 rows multiply with the generation variables $P_{11}, P_{12}, P_{13}, P_{21}, P_{22}, P_{23}, P_{41}, P_{42}, P_{43},$. The demand (P_D) will be given to us in the \underline{b} matrix and $P_{gk}-P_{dk}$ is in the \underline{b} matrix.
- The 10-14 columns of the last 4 rows correspond to the line flow variables, but these are zeros because they are not the DC power flow equation.
- The last 4 columns of the last 4 rows correspond to the DC power flows that include all the angles.

We can see that these elements will occupy the upper right hand corner of \underline{A}_{eq} . So that will take care of the last 4 columns in the first 5 rows.

What about the first 14 columns? These elements are the equations that are multiplied by the variables $P_{11}, P_{12}, P_{13}, P_{21}, P_{22}, P_{23}, P_{41}, P_{42}, P_{43}, P_{B1}, P_{B2}, P_{B3}, P_{B4},$ and P_{B5} . Since we do not use the generation variables within the line flow equations, the first 9 columns of these top 5 rows will be zeros. The last 5 columns in these top 5 rows will also be zeros, except the one element in each of these rows that multiply the corresponding line flow variable, and that element will be -1.

The right hand side equations will be set to zero

$$\underline{A}_{eq} \underline{x} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 10 & 0 & 0 & -10 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 10 & -10 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 10 & -10 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -10 & 10 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 10 & 0 & -10 & 0 \\ - & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - \end{bmatrix} \begin{bmatrix} P_{11} \\ P_{12} \\ P_{13} \\ P_{21} \\ P_{22} \\ P_{23} \\ P_{41} \\ P_{42} \\ P_{43} \\ P_{B1} \\ P_{B2} \\ P_{B3} \\ P_{B4} \\ P_{B5} \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -1 \\ -1.1787 \\ 0 \end{bmatrix}$$

The B' matrix deal with the bus angles. This is inserted into the last 4 columns and the bottom 4 rows. The resulting matrix appears as:

$$\underline{A}_{eq} \underline{x} = \begin{bmatrix}
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 10 & 0 & 0 & -10 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 10 & -10 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 10 & -10 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -10 & 10 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 10 & 0 & -10 & 0 \\
 - & - & - & - & - & - & - & - & - & - & - & - & - & - & 30 & -10 & -10 & -10 \\
 - & - & - & - & - & - & - & - & - & - & - & - & - & - & -10 & 20 & -10 & 0 \\
 - & - & - & - & - & - & - & - & - & - & - & - & - & - & -10 & -10 & 30 & -10 \\
 - & - & - & - & - & - & - & - & - & - & - & - & - & - & -10 & 0 & -10 & 20
 \end{bmatrix}
 \begin{bmatrix}
 P_{11} \\
 P_{12} \\
 P_{13} \\
 P_{21} \\
 P_{22} \\
 P_{23} \\
 P_{41} \\
 P_{42} \\
 P_{43} \\
 P_{B1} \\
 P_{B2} \\
 P_{B3} \\
 P_{B4} \\
 P_{B5} \\
 \theta_1 \\
 \theta_2 \\
 \theta_3 \\
 \theta_4
 \end{bmatrix}
 = \begin{bmatrix}
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 -1.1787 \\
 0
 \end{bmatrix}$$

Once again, we need to consider the first eight columns. Columns 4-8 correspond to the line flow variables, which do not appear in the DC power flow equations, so these will be zero.

Given that the demand for Generator 2 = 1.0 per unit and Generator 3 = 1.178 per unit
 $P_{d1}=0, P_{d2}=1.0, P_{d3}=1.1787, P_{d4}=0, P_{g3}=0$

Since these are constants, they can go to the right-hand side. This means on the left-hand side, the injection, which is defined as $P_{gk}-P_{Dk}$ should be negative. That is, we should see on the left-hand-side $-P_{gk}+P_{Dk}$. But now we will take the load term onto the right-hand-side by subtracting it from both sides.

Thus, we see that the load term should show up on the right-hand-side as a negative number

$$\underline{A}_{eq} \underline{x} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 10 & 0 & 0 & -10 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 10 & -10 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 10 & -10 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -10 & 10 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 10 & 0 & -10 & 0 \\ -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 30 & -10 & -10 & -10 \\ 0 & 0 & 0 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -10 & 20 & -10 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -10 & -10 & 30 & -10 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & -10 & 0 & -10 & 20 \end{bmatrix} \begin{bmatrix} P_{11} \\ P_{12} \\ P_{13} \\ P_{21} \\ P_{22} \\ P_{23} \\ P_{41} \\ P_{42} \\ P_{43} \\ P_{B1} \\ P_{B2} \\ P_{B3} \\ P_{B4} \\ P_{B5} \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -1 \\ -1.1787 \\ 0 \end{bmatrix}$$

Putting all of the constraints together, and converting the generator constraints to per unit, the following vector of constraints is given.

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -5 \\ -5 \\ -5 \\ -5 \\ -5 \\ -\pi \\ -\pi \\ -\pi \\ -\pi \end{bmatrix} \leq \begin{bmatrix} P_{11} \\ P_{12} \\ P_{13} \\ P_{21} \\ P_{22} \\ P_{23} \\ P_{41} \\ P_{42} \\ P_{43} \\ P_{B1} \\ P_{B2} \\ P_{B3} \\ P_{B4} \\ P_{B5} \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix} \leq \begin{bmatrix} .5 \\ .6 \\ .4 \\ .325 \\ .6 \\ .2 \\ .45 \\ .5 \\ .4 \\ 5 \\ 5 \\ 5 \\ 5 \\ 5 \\ \pi \\ \pi \\ \pi \\ \pi \end{bmatrix}$$

We then solve for these in Matlab using linear programming:

```
%Load is system load plus losses
```

```
Load=2.1787;
```

```
%Build objective function vector.
```

```
c=[1246 1307 1358 1129 1211 1282 1183 1254 1320 0 0 0 0 0 0 0 0];
```

```
%Build A matrix for inequality constraints  $Ax < b$ .
```

```
A=[];
```

```
%Build b, the right-hand-side of inequality constraints.
```

```
b=[];
```

```
%Build Aeq matrix for equality constraints.
```

```
Aeq=[0 0 0 0 0 0 0 0 0 -1 0 0 0 0 10 0 0 -10;  
      0 0 0 0 0 0 0 0 0 0 -1 0 0 0 10 -10 0 0;  
      0 0 0 0 0 0 0 0 0 0 -1 0 0 0 10 -10 0;  
      0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 -10 10;  
      0 0 0 0 0 0 0 0 0 0 0 0 -1 10 0 -10 0;  
      -1 -1 -1 0 0 0 0 0 0 0 0 0 0 0 30 -10 -10 -10;  
      0 0 0 -1 -1 -1 0 0 0 0 0 0 0 0 -10 20 -10 0;  
      0 0 0 0 0 0 0 0 0 0 0 0 0 0 -10 -10 30 -10;  
      0 0 0 0 0 0 -1 -1 -1 0 0 0 0 0 -10 0 -10 20];
```

```
%Build right-hand side of equality constraint.
```

```
beq=zeros(9,1);
```

```
beq(7)=-1;
```

```
beq(8)=-1.1787;
```

```
%Build upper and lower bounds on decision variables.
```

```
LB=[0 0 0 0 0 0 0 0 0 -5 -5 -5 -5 -5 -pi -pi -pi -pi];
```

```
UB=[.5 .5 .4 .325 .6 .2 .45 .5 .4 5 5 5 5 5 pi pi pi pi];
```

```
[X,FVAL,EXITFLAG,OUTPUT,LAMBDA]=LINPROG(c,A,b,Aeq,beq,LB,UB);
```

The solution is then:

$$\underline{x} = \begin{bmatrix} P_{11} \\ P_{12} \\ P_{13} \\ P_{21} \\ P_{22} \\ P_{23} \\ P_{41} \\ P_{42} \\ P_{43} \\ P_{B1} \\ P_{B2} \\ P_{B3} \\ P_{B4} \\ P_{B5} \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix} = \begin{bmatrix} 0.5000 \\ 0.0000 \\ 0.0000 \\ 0.3250 \\ 0.6000 \\ 0.0000 \\ 0.4500 \\ 0.3037 \\ 0.0000 \\ -0.1670 \\ 0.2473 \\ 0.1723 \\ 0.5867 \\ 0.4197 \\ 0.0126 \\ -0.0122 \\ -0.0294 \\ 0.0293 \end{bmatrix}$$

With the objective function of :

$$Z(\underline{x}) = [1246 \quad 1307 \quad 1358 \quad 1129 \quad 1211 \quad 1282 \quad 1183 \quad 1254 \quad 1320 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0] \begin{bmatrix} P_{11} \\ P_{12} \\ P_{13} \\ P_{21} \\ P_{22} \\ P_{23} \\ P_{41} \\ P_{42} \\ P_{43} \\ P_{B1} \\ P_{B2} \\ P_{B3} \\ P_{B4} \\ P_{B5} \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix}$$

We find that $Z = 2629.7$ \$/hr

But we need to add to this solution for the following for each unit:

$$\Delta C_k = C(P_{gk,\min}) - S_k P_{gk,\min}$$

$$\underline{x} = \begin{bmatrix} P_{11} \\ P_{12} \\ P_{13} \\ P_{21} \\ P_{22} \\ P_{23} \\ P_{41} \\ P_{42} \\ P_{43} \\ P_{B1} \\ P_{B2} \\ P_{B3} \\ P_{B4} \\ P_{B5} \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix} = \begin{bmatrix} 0.5000 \\ 0.0000 \\ 0.0000 \\ 0.3250 \\ 0.6000 \\ 0.0000 \\ 0.4500 \\ 0.3037 \\ 0.0000 \\ -0.1670 \\ 0.2473 \\ 0.1723 \\ 0.5867 \\ 0.4197 \\ 0.0126 \\ -0.0122 \\ -0.0294 \\ 0.0293 \end{bmatrix}$$

$$\Delta C_1 = 0.00533(50)^2 + 11.669(50) + 213.1 - 12.46(50) = 186.88$$

$$\Delta C_2 = 0.00889(95.5)^2 + 10.333(92.5) + 200 - 11.29(32.5) - 12.11(60) = 143.36$$

$$\Delta C_4 = 0.00741(75.37)^2 + 10.833(75.37) + 240 - 11.83(45) - 12.54(30.37) = 185.39$$

So the true objective function is:

$$2629.7 + 186.88 + 143.36 + 185.39 = 3145.33 \text{ $/hr}$$

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -5 \\ -0.15 \\ -5 \\ -5 \\ -5 \\ -\pi \\ -\pi \\ -\pi \\ -\pi \end{bmatrix} \leq \begin{bmatrix} P_{11} \\ P_{12} \\ P_{13} \\ P_{21} \\ P_{22} \\ P_{23} \\ P_{41} \\ P_{42} \\ P_{43} \\ P_{B1} \\ P_{B2} \\ P_{B3} \\ P_{B4} \\ P_{B5} \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix} \leq \begin{bmatrix} .5 \\ .6 \\ .4 \\ .325 \\ .6 \\ .2 \\ .45 \\ .5 \\ .4 \\ 5 \\ 0.15 \\ 5 \\ 5 \\ 5 \\ \pi \\ \pi \\ \pi \\ \pi \end{bmatrix}$$

We then solve for these in Matlab using linear programming:

```
%Load is system load plus losses
```

```
Load=2.1787;
```

```
%Build objective function vector.
```

```
c=[1246 1307 1358 1129 1211 1282 1183 1254 1320 0 0 0 0 0 0 0 0];
```

```
%Build A matrix for inequality constraints Ax<b.
```

```
A=[];
```

```
%Build b, the right-hand-side of inequality constraints.
```

```
b=[];
```

```
%Build Aeq matrix for equality constraints.
```

```
Aeq=[0 0 0 0 0 0 0 0 0 -1 0 0 0 0 10 0 0 -10;
      0 0 0 0 0 0 0 0 0 0 -1 0 0 0 10 -10 0 0;
      0 0 0 0 0 0 0 0 0 0 -1 0 0 0 10 -10 0;
      0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 -10 10;
      0 0 0 0 0 0 0 0 0 0 0 0 -1 10 0 -10 0;
      -1 -1 -1 0 0 0 0 0 0 0 0 0 0 0 30 -10 -10 -10;
      0 0 0 -1 -1 -1 0 0 0 0 0 0 0 0 -10 20 -10 0;
      0 0 0 0 0 0 0 0 0 0 0 0 0 0 -10 -10 30 -10;
      0 0 0 0 0 0 -1 -1 -1 0 0 0 0 0 -10 0 -10 20;];
```

%Build right-hand side of equality constraint.

```
beq=zeros(9,1);
```

```
beq(7)=-1;
```

```
beq(8)=-1.1787;
```

%Build upper and lower bounds on decision variables.

```
LB=[0 0 0 0 0 0 0 0 -5 -0.15 -5 -5 -5 -pi -pi -pi]';
```

```
UB=[.5 .5 .4 .325 .6 .2 .45 .5 .4 5 0.15 5 5 5 pi pi pi pi]';
```

```
[X,FVAL,EXITFLAG,OUTPUT,LAMBDA]=LINPROG(c,A,b,Aeq,beq,LB,UB);
```

$$\underline{x} = \begin{bmatrix} P_{11} \\ P_{12} \\ P_{13} \\ P_{21} \\ P_{22} \\ P_{23} \\ P_{41} \\ P_{42} \\ P_{43} \\ P_{B1} \\ P_{B2} \\ P_{B3} \\ P_{B4} \\ P_{B5} \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0 \\ 0 \\ 0.325 \\ 0.6 \\ 0.1946 \\ 0.45 \\ 0.1090 \\ 0 \\ -0.0697 \\ 0.15 \\ 0.26967 \\ 0.4894 \\ 0.4197 \\ 0.0125 \\ -0.0025 \\ -0.0295 \\ 0.0195 \end{bmatrix}$$

We find that $Z = 2635.2$

$$\Delta C_1 = 0.00533(50)^2 + 11.669(50) + 213.1 - 12.46(50) = 186.88$$

$$\Delta C_2 = 0.00889(111.96)^2 + 10.333(111.96) + 200 - 11.29(32.5) - 12.11(60) - 12.82(10.9) = 235.06$$

$$\Delta C_4 = 0.00741(55.9)^2 + 10.833(55.9) + 240 - 11.83(45) - 12.54(10.9) = 199.6835$$

So the true objective function is:

$$2635.2 + 186.88 + 235.06 + 199.68 = 3256.82 \text{ \$/hr}$$

The most noticeable change in the system is how the generation shifted at each generator to account for the constraint on Bus 2.

$$\begin{bmatrix} P_{G1} \\ P_{G2} \\ P_{G3} \\ P_{G4} \\ P_{B1} \\ P_{B2} \\ P_{B3} \\ P_{B4} \\ P_{B5} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \underbrace{\begin{bmatrix} 0.5 \\ 0 \\ 0 \\ 0.325 \\ 0.6 \\ 0.1946 \\ 0.45 \\ 0.1090 \\ 0 \\ -0.0697 \\ 0.15 \\ 0.26967 \\ 0.4894 \\ 0.4197 \end{bmatrix}}_{\text{Part of } \underline{x}} = \begin{bmatrix} 0.5 \\ 1.1197 \\ 0 \\ 0.55903 \\ -0.069675 \\ 0.15 \\ 0.26967 \\ 0.48935 \\ 0.41967 \end{bmatrix} = \begin{bmatrix} P_{G1} \\ P_{G2} \\ P_{G3} \\ P_{G4} \\ P_{B1} \\ P_{B2} \\ P_{B3} \\ P_{B4} \\ P_{B5} \end{bmatrix}$$

The Change in the P vector that was solved for is:

$$\Delta P = \underbrace{\begin{bmatrix} 0.5 \\ 0.925 \\ 0 \\ 0.7537 \\ -0.16701 \\ 0.24734 \\ 0.17234 \\ 0.58669 \\ 0.41967 \end{bmatrix}}_{P_{original}} - \underbrace{\begin{bmatrix} 0.5 \\ 1.1197 \\ 0 \\ 0.55903 \\ -0.069675 \\ 0.15 \\ 0.26967 \\ 0.48935 \\ 0.41967 \end{bmatrix}}_{P_{constrain}} = \begin{bmatrix} 0 \\ -0.1947 \\ 0 \\ 0.19467 \\ -0.0974 \\ 0.09734 \\ -0.0973 \\ 0.09734 \\ 0 \end{bmatrix}$$

Summary algorithm for the steps above

Using the above steps we determine the necessary variables to enter into a linear programming equation that produces the solution vector. The variables required by the equation are:

- Objective function vector
- A matrix
- RHS of inequality constraints

-
- A equivalent matrix
 - RHS equality constraints
 - Lower bounds
 - Upper bounds

We find the objective function, lower bounds, and the upper bounds from the cost curve data. The A matrix, RHS of inequality constraints, RHS of equality constraints, and the A equivalent matrix are found using MatLab code using data from PSSE.

Appendix 2: Derivation of Generation Shift Factors

We desire to compute the generation shift factor, $t_{b,k}$.

We can obtain the flows on all circuits given the injections at all buses, as shown below:

$$\underline{P} = \underline{B}' \underline{\theta} \quad (1)$$

$$\underline{P}_B = (\underline{D} \times \underline{A}) \times \underline{\theta} \quad (2)$$

Inverting eq (1) yields:

$$\underline{\theta} = [\underline{B}']^{-1} \underline{P} \quad (3)$$

Substitution of (3) into (2) yields:

$$\underline{P}_B = (\underline{D} \times \underline{A}) [\underline{B}']^{-1} \underline{P} \quad (4)$$

But this is not what we want. What we want is the change in flow on circuit b given a change in injection at bus k .

Here is a “change in injection vector,” $\Delta \underline{P}$:

$$\Delta \underline{P} = \begin{bmatrix} P_2 \\ P_3 \\ \vdots \\ P_k \\ \vdots \\ P_N \end{bmatrix} - \begin{bmatrix} P_2^0 \\ P_3^0 \\ \vdots \\ P_k^0 \\ \vdots \\ P_N^0 \end{bmatrix} = \begin{bmatrix} \Delta P_2 \\ \Delta P_3 \\ \vdots \\ \Delta P_k \\ \vdots \\ \Delta P_N \end{bmatrix} = \underline{P} - \underline{P}^0 \quad (5)$$

The change in circuit flows can then be expressed as

$$\begin{aligned}
\Delta \underline{P}_B &= \underline{P}_B - \underline{P}_B^0 \\
&= (\underline{D} \times \underline{A}) [\underline{B}']^{-1} \underline{P} - (\underline{D} \times \underline{A}) [\underline{B}']^{-1} \underline{P}^0 \\
&= (\underline{D} \times \underline{A}) [\underline{B}']^{-1} (\underline{P} - \underline{P}^0) \\
&= (\underline{D} \times \underline{A}) [\underline{B}']^{-1} \Delta \underline{P}
\end{aligned} \tag{6}$$

Now let the $\Delta \underline{P}$ vector be all zeros except for the element corresponding to the k^{th} bus, and assign this bus an injection change of 1.

$$\Delta \underline{P} = \begin{bmatrix} \Delta P_2 \\ \Delta P_3 \\ \vdots \\ \Delta P_k \\ \vdots \\ \Delta P_N \end{bmatrix} = \begin{bmatrix} \Delta P_2 \\ \Delta P_3 \\ \vdots \\ \Delta P_k \\ \vdots \\ \Delta P_N \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} \tag{7}$$

Then

$$\Delta \underline{P}_B = \begin{bmatrix} \Delta P_{B1} \\ \Delta P_{B2} \\ \vdots \\ \Delta P_{Bb} \\ \vdots \\ \Delta P_{BM} \end{bmatrix} = \begin{bmatrix} t_{1,k} \\ t_{2,k} \\ \vdots \\ t_{b,k} \\ \vdots \\ t_{M,k} \end{bmatrix} = (\underline{D} \times \underline{A}) [\underline{B}']^{-1} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} \leftarrow \Delta P_k \tag{8}$$

Question: Does the above equation imply the injection is changed at only one bus? **Answer:** No. There is an equal and opposite change to the injection at the swing bus, in this case, bus 1.

Definition: The generation shift factor $t_{b,k}$ is defined as

$$t_{b,k} = \frac{\Delta P_{Bb}}{\Delta P_k} \Bigg|_{\text{Reallocation Policy}}$$

This is denoted as a_{δ} in the W&W (ref [1]) text (see eq. 11.1).

Example 1:

We consider an example used in Appendix 1, illustrated below. Compute the generation shift factors for all branches corresponding to an increase in bus 2 injection and a decrease in bus 3 injection.

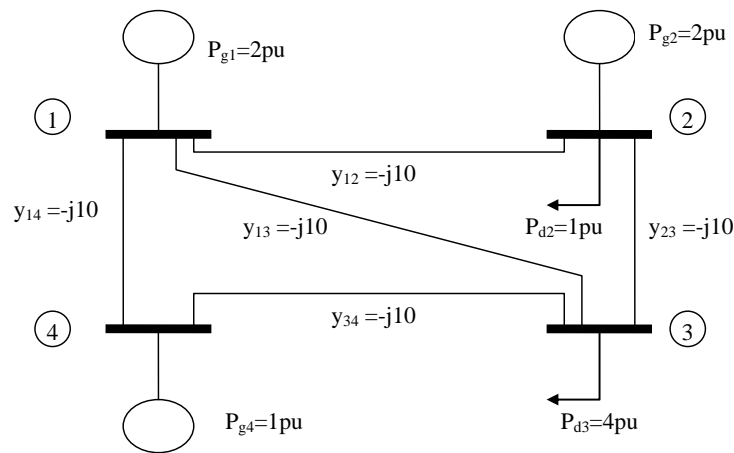


Fig. 2: Four-bus network used in example

$$\begin{aligned}
\begin{bmatrix} t_{1,23} \\ t_{2,23} \\ t_{3,23} \\ t_{4,23} \\ t_{5,23} \end{bmatrix} &= \begin{bmatrix} 10 & 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 10 & 0 & 0 \\ 0 & 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 0 & 10 \end{bmatrix} \begin{bmatrix} 0 & 0 & -1 \\ -1 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & -1 & 1 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} 20 & -10 & 0 \\ -10 & 30 & -10 \\ 0 & -10 & 20 \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix} \\
&= \begin{bmatrix} 0 & 0 & -10 \\ -10 & 0 & 0 \\ 10 & -10 & 0 \\ 0 & -10 & 10 \\ 0 & -10 & 0 \end{bmatrix} \begin{bmatrix} 0.0625 & 0.025 & 0.0125 \\ 0.025 & 0.05 & 0.025 \\ 0.0125 & 0.025 & 0.0625 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix} \\
&= \begin{bmatrix} 0 & 0 & -10 \\ -10 & 0 & 0 \\ 10 & -10 & 0 \\ 0 & -10 & 10 \\ 0 & -10 & 0 \end{bmatrix} \begin{bmatrix} 0.0375 \\ -0.025 \\ -0.0125 \end{bmatrix} = \begin{bmatrix} 0.125 \\ -0.375 \\ 0.625 \\ 0.125 \\ 0.25 \end{bmatrix}
\end{aligned}$$

Note that the above generation shift factors are for a “double shift.”

You can think of it like this. A generation shift factor for branch b , bus k would be $t_{b,k}$ and another generation shift factor for branch b , bus j would be $t_{b,j}$. If we have an injection increase at bus k of ΔP_k and an injection increase at bus j of ΔP_j (negative), then

$$\Delta P_b = t_{b,k} \Delta P_k + t_{b,j} \Delta P_j \quad (9)$$



Therefore, if $\Delta P_k = -\Delta P_j$, then

$$\Delta P_b = (t_{b,k} - t_{b,j}) \Delta P_k \quad (10)$$

Note that once ΔP_b is obtained, then it must be added to the original flow on branch b to get the resulting total flow following the generation shift, i.e.,

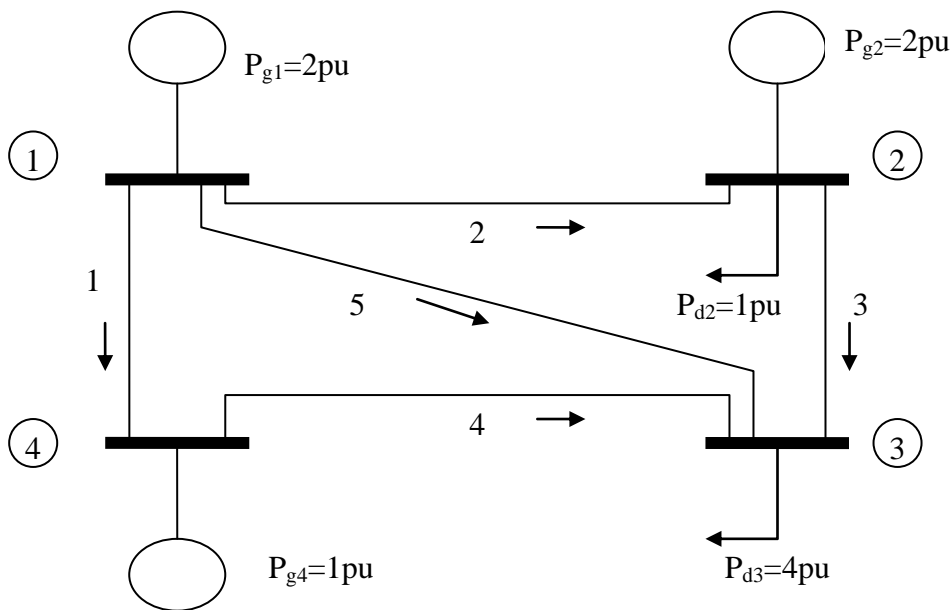
$$\hat{P}_b = P_b^0 + \Delta P_b$$

The last equation is the same as eq. (11.2) in W&W text.

Appendix 3: Derivation of Line Outage Distribution Factors

A significant problem with W&W's method given in [1] of obtaining the LODFs is that it requires $\underline{X}=(\underline{B}')^{-1}$, and if the system is very large, then inverting the matrix can be a computationally intense problem. We provide another method in this section. Our treatment is adapted from [1].

Let's reconsider our familiar 4-bus, 5-branch example problem.



The \underline{B}' matrix for this system is

$$\underline{B}' = \begin{bmatrix} 20 & -10 & 0 \\ -10 & 30 & -10 \\ 0 & -10 & 20 \end{bmatrix}$$

What happens to \underline{B}' if we lose the circuit #3 (from bus 2 to bus 3)?

We could re-develop the new \underline{B}' from the one-line diagram as we are accustomed to doing now. Another way is to discern how the circuit #3 affects the \underline{B}' matrix, in that it will affect exactly 4 elements, as indicated with the underlines below, corresponding to elements in bus numbered positions (2,2), (2,3), (3,2), and (3,3).

$$\underline{B}' = \begin{array}{ccc} & \begin{array}{ccc} 2 & 3 & 4 \end{array} \\ \begin{array}{c} 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} \underline{20} & \underline{-10} & 0 \\ \underline{-10} & \underline{30} & -10 \\ 0 & -10 & 20 \end{bmatrix} \end{array}$$

Recalling that all branch admittances of our network are $-j10$, what would these four elements be if branch #3 (between buses 2 and 3) were not there?

$$\underline{B}'^{out} = \begin{array}{ccc} & \begin{array}{ccc} 2 & 3 & 4 \end{array} \\ \begin{array}{c} 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} \underline{10} & \underline{0} & 0 \\ \underline{0} & \underline{20} & -10 \\ 0 & -10 & 20 \end{bmatrix} \end{array}$$

What is the difference between \underline{B}' and \underline{B}'^{out} ?

$$\Delta \underline{B}' = \underline{B}'^{out} - \underline{B}' = \begin{bmatrix} \underline{10} & \underline{0} & 0 \\ \underline{0} & \underline{20} & -10 \\ 0 & -10 & 20 \end{bmatrix} - \begin{bmatrix} \underline{20} & \underline{-10} & 0 \\ \underline{-10} & \underline{30} & -10 \\ 0 & -10 & 20 \end{bmatrix} = \begin{bmatrix} \underline{-10} & \underline{10} & 0 \\ \underline{10} & \underline{-10} & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Notice that the elements in $\Delta \underline{B}'$ are all multiples of $B'_{23} = -10$, i.e.,

$$\Delta \underline{B}' = -10 \begin{bmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

The above matrix can be expressed as

$$\begin{bmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix} \begin{bmatrix} 1 & -1 & 0 \end{bmatrix}$$

From this simple illustration, we can see a generalization, that whenever we remove a branch between buses i and j , with corresponding \underline{B}' matrix element B'_{ij} , the \underline{B}' matrix will change as indicated below.

Inconsistency: In Section 6.0, we used (i,j) to indicate terminals of the circuit to be loaded (l) and (m,n) to indicate terminals of the circuit to be outaged (k). In the development of this section, the nomenclature on terminal number has been reversed, i.e., (i,j) becomes the terminals of the circuit to be outaged (k) and (m,n) becomes the terminals of the circuit to be loaded (l).

$$\Delta \underline{B}' = \underline{B}'^{out} - \underline{B}' = b_{ij} \begin{bmatrix} & & \begin{matrix} i \\ \downarrow \end{matrix} & & \begin{matrix} j \\ \downarrow \end{matrix} & & \\ 0 & \dots & 0 & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \\ 0 & \dots & 1 & \dots & -1 & \dots & 0 \leftarrow i \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \dots & -1 & \dots & 1 & \dots & 0 \leftarrow j \\ \vdots & & \vdots & & \vdots & & \vdots \\ 0 & \dots & 0 & \dots & 0 & \dots & 0 \end{bmatrix} \quad (22)$$

is the susceptance of branch i-j (and is a negative number for any standard circuit, which is inductive). We use b_{ij} instead of $-B'_{ij}$ in order to ensure we have a defined term even when i or j are the swing bus.

The previous relation may be expressed as

Caution: The designated positions in the row and column vectors correspond to buses i and j, i.e., they are not the i^{th} and j^{th} positions.

$$\Delta \underline{B}' = b_{ij} \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ -1 \\ \vdots \\ 0 \end{bmatrix} \leftarrow i \times [0 \dots 1 \dots -1 \dots 0] \begin{matrix} \uparrow & \uparrow \\ i & j \end{matrix} \quad (23)$$

If we define

$$\underline{e}_{ij} = \begin{bmatrix} 0 \\ \vdots \\ 1 \leftarrow i \\ \vdots \\ -1 \leftarrow j \\ \vdots \\ 0 \end{bmatrix} \quad (24)$$

then (23) becomes

$$\Delta \underline{B}' = b_{ij} \underline{e}_{ij} \underline{e}_{ij}^T \quad (25)$$

Special case: If the branch to be outaged is connected to the swing bus (in our case, it is bus #1), then,

$$\begin{array}{l} \text{if } i=1, \\ \underline{e}_{1j} = \begin{bmatrix} 0 \\ \vdots \\ \vdots \\ \vdots \\ 1 \leftarrow j \\ \vdots \\ 0 \end{bmatrix} \end{array} \quad \begin{array}{l} \text{if } j=1, \\ \underline{e}_{i1} = \begin{bmatrix} 0 \\ \vdots \\ -1 \leftarrow i \\ \vdots \\ \vdots \\ \vdots \\ 0 \end{bmatrix} \end{array}$$

From (22), and using (25), we have that

$$\Delta \underline{B}' = \underline{B}'^{out} - \underline{B}' = b_{ij} \underline{e}_{ij} \underline{e}_{ij}^T \quad (26)$$

Therefore the post-contingency \underline{B}' matrix can be expressed as

$$\underline{B}'^{out} = \underline{B}' + \Delta \underline{B}' = \underline{B}' + b_{ij} \underline{e}_{ij} \underline{e}_{ij}^T \quad (27)$$

From (1), we recall the DC power flow relation as

$$\underline{P} = \underline{B}' \underline{\theta} \quad (1)$$

If, when we remove the branch connected between buses i and j , the angles change by $\Delta\theta$, then the new (post-contingency) angles will be $\theta + \Delta\theta$, and (1) becomes

$$\underline{P} = \underline{B}'^{out} (\underline{\theta} + \Delta\underline{\theta}) \quad (28)$$

(We are assessing the effects of only a lone outage, so the injection vector \underline{P} does not change). Substituting (27) into (28), we obtain

$$\underline{P} = \left(\underline{B}' + b_{ij} \underline{e}_{ij} \underline{e}_{ij}^T \right) (\underline{\theta} + \Delta\underline{\theta}) \quad (29)$$

We can solve for the new angles according to

$$\underline{\theta} + \Delta\underline{\theta} = \left(\underline{B}' + b_{ij} \underline{e}_{ij} \underline{e}_{ij}^T \right)^{-1} \underline{P} \quad (30)$$

We do not seem to have made much progress, because we still have to take an inverse...

However, there is a significant benefit to writing the new matrix in the way that we have written it, and that benefit becomes apparent if we learn a certain matrix relation. This relation is generally referred to as a lemma.

Matrix Inversion Lemma (MIL): Assume \underline{B}' is a nonsingular $n \times n$ matrix, and let \underline{c} and \underline{d} be $n \times M$ matrices with $M < n$. Then:

$$\left(\underline{B}' + \underline{c} \underline{d}^T \right)^{-1} = \underline{B}'^{-1} - \underline{B}'^{-1} \underline{c} \left[\underline{I}^{(M)} + \underline{d}^T \underline{B}'^{-1} \underline{c} \right]^{-1} \underline{d}^T \underline{B}'^{-1}$$

where $\underline{I}^{(M)}$ is the $M \times M$ identity matrix.

We neglect the proof but mention that it is proved in [i, p. 100] by simply multiplying the right-hand-side of MIL by the expression inside the brackets of the left-hand-side, and showing that the product is the $n \times n$ identity matrix.

We also mention that MIL is *derived* in [ii, pp. 138-140].

It may not be very obvious at this point that MIL will help us, since we see 4 different inverses on the right-hand-side of MIL. Let's apply MIL to the inverted term of (30) to see what happens.

Observing that we can define

$$\begin{aligned}\underline{c} &= b_{ij} \underline{e}_{ij} \\ \underline{d}^T &= \underline{e}_{ij}^T\end{aligned}\quad (31)$$

we can apply MIL according to

$$\begin{aligned}\left(\underline{B}' + b_{ij} \underline{e}_{ij} \underline{e}_{ij}^T\right)^{-1} &= \\ \underline{B}'^{-1} - \underline{B}'^{-1} b_{ij} \underline{e}_{ij} \left[\underline{I}^{(M)} + \underline{e}_{ij}^T \underline{B}'^{-1} b_{ij} \underline{e}_{ij}\right]^{-1} \underline{e}_{ij}^T \underline{B}'^{-1}\end{aligned}\quad (32)$$

One of the inverses on the right-hand-side can be addressed right away, however, by identifying the dimensionality of the expression inside the right-hand-side brackets, $[\underline{I}^{(M)} + \underline{d}^T \underline{B}'^{-1} \underline{c}]$. Observing from the MIL that M is the number of columns in \underline{c} and \underline{d} , and noting from (31) that in our case, \underline{c} and \underline{d} have only M=1 column, we see that what is inside the right-hand-side brackets is a scalar quantity! So that inverse we can take, and accordingly, we express (32) as:

$$\left(\underline{B}' + b_{ij} \underline{e}_{ij} \underline{e}_{ij}^T\right)^{-1} = \underline{B}'^{-1} - \frac{\underline{B}'^{-1} b_{ij} \underline{e}_{ij} \underline{e}_{ij}^T \underline{B}'^{-1}}{1 + \underline{e}_{ij}^T \underline{B}'^{-1} b_{ij} \underline{e}_{ij}} \quad (33)$$

Pulling out the scalar multiplier b_{ij} from where it appears in both the numerator and denominator, we have

$$\left(\underline{B}' + b_{ij} \underline{e}_{ij} \underline{e}_{ij}^T\right)^{-1} = \underline{B}'^{-1} - \frac{b_{ij} \underline{B}'^{-1} \underline{e}_{ij} \underline{e}_{ij}^T \underline{B}'^{-1}}{1 + b_{ij} \underline{e}_{ij}^T \underline{B}'^{-1} \underline{e}_{ij}} \quad (34)$$

Now we can isolate b_{ij} to only one appearance in the expression by dividing top and bottom by it, resulting in:

$$\left(\underline{B}' + b_{ij} \underline{e}_{ij} \underline{e}_{ij}^T\right)^{-1} = \underline{B}'^{-1} - \frac{\underline{B}'^{-1} \underline{e}_{ij} \underline{e}_{ij}^T \underline{B}'^{-1}}{\frac{1}{b_{ij}} + \underline{e}_{ij}^T \underline{B}'^{-1} \underline{e}_{ij}} \quad (35)$$

What we have just expressed in (35) is the inverted term on the right-hand-side of (30), repeated below for convenience:

$$\underline{\theta} + \Delta \underline{\theta} = \left(\underline{B}' + b_{ij} \underline{e}_{ij} \underline{e}_{ij}^T\right)^{-1} \underline{P} \quad (30)$$

Substituting (35) into (30), we obtain:

$$\underline{\theta} + \Delta \underline{\theta} = \left(\underline{B}'^{-1} - \frac{\underline{B}'^{-1} \underline{e}_{ij} \underline{e}_{ij}^T \underline{B}'^{-1}}{\frac{1}{b_{ij}} + \underline{e}_{ij}^T \underline{B}'^{-1} \underline{e}_{ij}} \right) \underline{P} \quad (36)$$

Distributing the injection vector \underline{P} results in

$$\underline{\theta} + \Delta \underline{\theta} = \underline{B}'^{-1} \underline{P} - \frac{\underline{B}'^{-1} \underline{e}_{ij} \underline{e}_{ij}^T \underline{B}'^{-1} \underline{P}}{\frac{1}{b_{ij}} + \underline{e}_{ij}^T \underline{B}'^{-1} \underline{e}_{ij}} \quad (37)$$

But $\underline{\theta} = \underline{B}'^{-1} \underline{P}$, and therefore we can replace the corresponding expressions in both right-hand-side terms to obtain:

$$\underline{\theta} + \Delta \underline{\theta} = \underline{\theta} - \frac{\underline{B}'^{-1} \underline{e}_{ij} \underline{e}_{ij}^T \underline{\theta}}{\frac{1}{b_{ij}} + \underline{e}_{ij}^T \underline{B}'^{-1} \underline{e}_{ij}} \quad (38)$$

We can simplify a little more by investigating $\underline{e}_{ij}^T \underline{\theta}$ in the numerator. This would be:

$$\underline{e}_{ij}^T \underline{\theta} = \begin{bmatrix} 0 & \cdots & \underset{\uparrow}{1} & \cdots & \underset{\uparrow}{-1} & \cdots & 0 \\ & & i & & j & & \end{bmatrix} \begin{bmatrix} \theta_2 \\ \vdots \\ \theta_i \\ \vdots \\ \theta_j \\ \vdots \\ \theta_{n+1} \end{bmatrix} = \theta_i - \theta_j \quad (39)$$

Substituting (39) into (38) results in:

$$\underline{\theta} + \Delta \underline{\theta} = \underline{\theta} - \frac{\underline{B}'^{-1} \underline{e}_{ij} (\theta_i - \theta_j)}{\frac{1}{b_{ij}} + \underline{e}_{ij}^T \underline{B}'^{-1} \underline{e}_{ij}} \quad (40)$$

Now we have only two inverses left. Interestingly, they both pre-multiply \underline{e}_{ij} . That is, we observe that both inverses appear in $\underline{B}'^{-1} \underline{e}_{ij}$, an $n \times 1$ vector.

Question: Besides inverting \underline{B}'^{-1} , how might we evaluate this term?

Advice: When you don't know how to evaluate something, just name it. Then, if things don't get better right away, you can at least move on with a sort of indicator of where your problem lies.

So let's name this $n \times 1$ vector as \underline{g}^{ij} , i.e.,

$$\underline{g}^{ij} = \underline{B}'^{-1} \underline{e}_{ij} \quad (41)$$

Not sure if that helps much but it does indicate that

$$\underline{B}' \underline{g}^{ij} = \underline{e}_{ij} \quad (42)$$

Equation (42) should stimulate a very good idea within your mind. Since we very well know \underline{B}' and \underline{e}_{ij} , we can obtain \underline{g}^{ij} through LU factorization. Doing so will give us everything we need to evaluate (40), which, when we substitute \underline{g}^{ij} for $\underline{B}'^{-1}\underline{e}_{ij}$, becomes:

$$\underline{\theta} + \Delta \underline{\theta} = \underline{\theta} - \frac{(\theta_i - \theta_j)}{\frac{1}{b_{ij}} + \underline{e}_{ij}^T \underline{g}^{ij}} \underline{g}^{ij} \quad (43)$$

One last small change should be made to (43), and that is to recognize that the term in the denominator $\underline{e}_{ij}^T \underline{g}^{ij}$ can be expressed as

$$\underline{e}_{ij}^T \underline{g}^{ij} = \begin{bmatrix} 0 & \dots & \underset{i}{1} & \dots & \underset{j}{-1} & \dots & 0 \end{bmatrix} \begin{bmatrix} g_2^{ij} \\ \vdots \\ g_i^{ij} \\ \vdots \\ g_j^{ij} \\ \vdots \\ g_{n+1}^{ij} \end{bmatrix} = g_i^{ij} - g_j^{ij} \quad (44)$$

Therefore, (43) becomes

$$\underline{\theta} + \Delta \underline{\theta} = \underline{\theta} - \frac{(\theta_i - \theta_j)}{\frac{1}{b_{ij}} + (g_i^{ij} - g_j^{ij})} \underline{g}^{ij} \quad (45)$$

Now what is the LODF? Recall the definition of the LODF is

$$d_{\ell,k} = \Delta f_{\ell} / f_{k0} \quad (16)$$

where we recall that

- k designates the outaged circuit, terminated by buses i and j ;
- ℓ designates the circuit for which we want to compute the new flow, terminated by buses m and n .

First, let's express the denominator of (16) f_{k0} , which is

$$f_{k0} = -b_{ij}(\theta_i - \theta_j) = -b_{ij} \underline{e}_{ij}^T \underline{\theta} \quad (46)$$

Now let's express the numerator of (16) Δf_ℓ , which is

$$\Delta f_\ell = -b_{mn}(\Delta\theta_m - \Delta\theta_n) = -b_{mn} \underline{e}_{mn}^T \Delta \underline{\theta} \quad (47)$$

Caution: The ordering of m, n must be consistent with the direction of positive flow as defined by the node-arc incidence matrix.

But note that $\Delta \underline{\theta}$ in (47) can be expressed using the second term of (45), i.e.,

$$\Delta \underline{\theta} = -\frac{(\theta_i - \theta_j)}{\frac{1}{b_{ij}} + (g_i^{ij} - g_j^{ij})} \underline{g}^{ij} \quad (48)$$

Substituting (48) into (47) results in

$$\Delta f_\ell = b_{mn} \underline{e}_{mn}^T \frac{(\theta_i - \theta_j)}{\frac{1}{b_{ij}} + (g_i^{ij} - g_j^{ij})} \underline{g}^{ij} \quad (49)$$

It is helpful at this point to rearrange (49) according to

$$\Delta f_\ell = b_{mn} \frac{b_{ij}(\theta_i - \theta_j)}{1 + b_{ij}(g_i^{ij} - g_j^{ij})} \underline{e}_{mn}^T \underline{g}^{ij} \quad (50)$$

We recognize in (50) that

$$f_{k0} = -b_{ij}(\theta_i - \theta_j) \quad (51)$$

and

Inconsistency:
Recall the note on p. 22, which indicates an inconsistency in nomenclature with Sec 6.0.

$$\underline{e}_{mn}^T \underline{g}^{ij} = g_m^{ij} - g_n^{ij} \quad (52)$$

Substituting (51) and (52) into (50) results in

$$\Delta f_\ell = -b_{mn} \frac{f_{k0}(g_m^{ij} - g_n^{ij})}{1 + b_{ij}(g_i^{ij} - g_j^{ij})} \quad (53)$$

So (53) can be used to obtain the change in flow on circuit ℓ (terminated by buses m and n) due to outage of circuit k (terminated by buses i and j).

To get the LODF, we divide (53) by f_{k0} , resulting in

$$d_{\ell,k} = \frac{\Delta f_\ell}{f_{k0}} = -b_{mn} \frac{(g_m^{ij} - g_n^{ij})}{1 + b_{ij}(g_i^{ij} - g_j^{ij})} \quad (54)$$

The approach, then, to using (54), is to factorize \underline{B}' into the \underline{L} and \underline{U} factors *once*. Then, for each contingency $k=1, \dots, N_C$, (per (42)), we use forward and backwards substitution to obtain the vector \underline{g}^{ij} . The LODFs for every branch ℓ (terminated by buses m and n), are then computed from (54).

Example 5:

Consider our 4-bus, 5-branch example problem again. Compute \underline{g}^{ij} for a line 2-3 outage. Then use it to compute the post-contingency flow on circuit 3-4.

Solution:

Recall (42):

$$\underline{B}' \underline{g}^{ij} = \underline{e}_{ij}$$

where \underline{B}' is given by:

$$\underline{B}' = \begin{bmatrix} 20 & -10 & 0 \\ -10 & 30 & -10 \\ 0 & -10 & 20 \end{bmatrix}$$

and \underline{e}_{23} is given by

$$\underline{e}_{23} = \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix} \begin{matrix} \leftarrow i = 2 \\ \leftarrow j = 3 \end{matrix}$$

And so our equation is:

$$\begin{bmatrix} 20 & -10 & 0 \\ -10 & 30 & -10 \\ 0 & -10 & 20 \end{bmatrix} \begin{bmatrix} g_2^{23} \\ g_3^{23} \\ g_4^{23} \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix}$$

Performing LU decomposition, we obtain

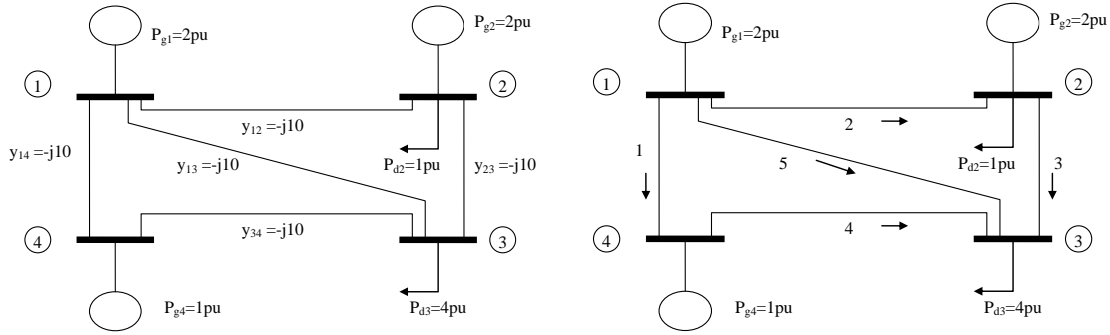
$$\underline{L} = \begin{bmatrix} 20 & 0 & 0 \\ -10 & 25 & 0 \\ 0 & -10 & 16 \end{bmatrix} \quad \underline{U} = \begin{bmatrix} 1 & -0.5 & 0 \\ 0 & 1 & -0.4 \\ 0 & 0 & 1 \end{bmatrix}$$

Notice that the above factors need be computed only one time; they may subsequently be applied to obtain the g-vector for outage of any circuit. In this case, we are interested in outage of the line from bus 2 to bus 3, therefore we write

$$\underline{L}\underline{w} = \underline{e}_{23} \Rightarrow \begin{bmatrix} 20 & 0 & 0 \\ -10 & 25 & 0 \\ 0 & -10 & 16 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix} \Rightarrow \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = \begin{bmatrix} 0.05 \\ -0.02 \\ -0.0125 \end{bmatrix}$$

$$\underline{U}\underline{g}^{23} = \underline{w} \Rightarrow \begin{bmatrix} 1 & -0.5 & 0 \\ 0 & 1 & -0.4 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} g_2^{23} \\ g_3^{23} \\ g_4^{23} \end{bmatrix} = \begin{bmatrix} 0.05 \\ -0.02 \\ -0.0125 \end{bmatrix} \Rightarrow \begin{bmatrix} g_2^{23} \\ g_3^{23} \\ g_4^{23} \end{bmatrix} = \begin{bmatrix} 0.0375 \\ -0.025 \\ -0.0125 \end{bmatrix}$$

Then we can compute the LODF for the circuit 3-4 after outage of circuit 2-3. First, however, note that the direction of positive flow on circuit 3-4 is defined by the node-arc incidence matrix as being from bus 4 to bus 3 (see below right-hand figure).



Therefore we need to designate $m=4$ and $n=3$. Then:

$$\begin{aligned}
 d_{\ell,k} &= \frac{\Delta f_{\ell}}{f_{k0}} = -b_{mn} \frac{(g_m^{ij} - g_n^{ij})}{1 + b_{ij}(g_i^{ij} - g_j^{ij})} \\
 &= -b_{43} \frac{(g_4^{23} - g_3^{23})}{1 + b_{23}(g_2^{23} - g_3^{23})} = 10 \frac{(-0.0125 - -0.025)}{1 + -10(0.0375 - -0.025)} \\
 &= 10 \frac{0.0125}{0.375} = 0.3333333
 \end{aligned}$$

If circuit 2-3 has flow of 1.25, then the change in flow on circuit 4-3, following outage of circuit 2-3 becomes

$$\Delta f_{\ell} = d_{\ell,k} f_{k0} = 0.3333333 * 1.25 = 0.4167$$

If the pre-contingency flow on circuit 3-4 was 1.25, then then new flow, following outage of circuit 2-3, will be

$$f_{\ell}^{out} = f_{\ell}^0 + \Delta f_{\ell} = 1.25 + 0.4167 = 1.6667$$

Now let's check our result with the DC power flow.

With all lines in we obtain

$$\underline{P}_B = (\underline{D} \times \underline{A}) [\underline{B}']^{-1} \underline{P}$$

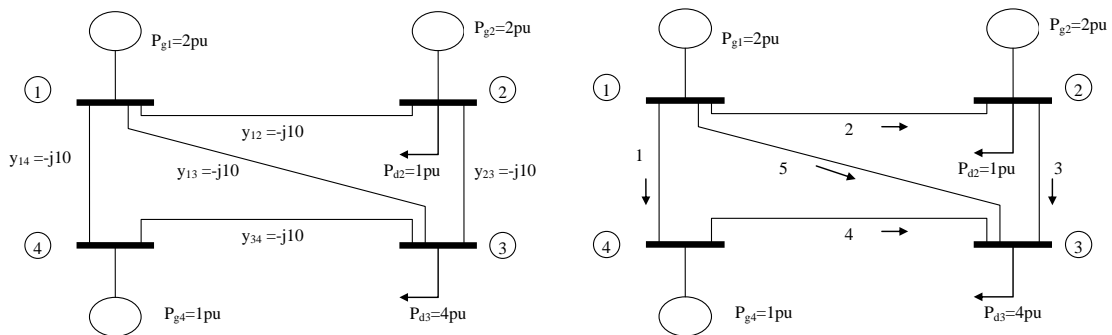
$$\begin{bmatrix} P_{B1} \\ P_{B2} \\ P_{B3} \\ P_{B4} \\ P_{B5} \end{bmatrix} = \begin{bmatrix} 10 & 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 10 & 0 & 0 \\ 0 & 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 0 & 10 \end{bmatrix} \begin{bmatrix} 0 & 0 & -1 \\ -1 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & -1 & 1 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} 20 & -10 & 0 \\ -10 & 30 & -10 \\ 0 & -10 & 20 \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ -4 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} P_{B1} \\ P_{B2} \\ P_{B3} \\ P_{B4} \\ P_{B5} \end{bmatrix} = \begin{bmatrix} 0.25 \\ 0.25 \\ 1.25 \\ 1.25 \\ 1.5 \end{bmatrix}$$

Observe that the flow on circuit 3-4 is 1.25 pu.

With circuit 2-3 out, we obtain:

$$\begin{bmatrix} P_{B1} \\ P_{B2} \\ P_{B3} \\ P_{B4} \\ P_{B5} \end{bmatrix} = \begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 10 \\ 0 & 0 & 0 & 10 \end{bmatrix} \begin{bmatrix} 0 & 0 & -1 \\ -1 & 0 & 0 \\ 0 & -1 & 1 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} 10 & 0 & 0 \\ 0 & 20 & -10 \\ 0 & -10 & 20 \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ -4 \\ 1 \end{bmatrix}$$



$$\begin{bmatrix} P_{B1} \\ P_{B3} \\ P_{B4} \\ P_{B5} \end{bmatrix} = \begin{bmatrix} 0.6667 \\ -1.0 \\ 1.6667 \\ 2.3333 \end{bmatrix}$$

Therefore the change in flows due to outage of circuit 2 (connected between buses 2 and 3) is

$$\Delta \underline{P}_b = \underline{P}_b^{out} - \underline{P}_b = \begin{bmatrix} 0.6667 \\ 0 \\ -1.0 \\ 1.6667 \\ 2.3333 \end{bmatrix} - \begin{bmatrix} 0.25 \\ 0.25 \\ 1.25 \\ 1.25 \\ 1.5 \end{bmatrix} = \begin{bmatrix} 0.4167 \\ -0.25 \\ -2.25 \\ 0.4167 \\ 0.8333 \end{bmatrix}$$

where we observe that $\Delta P_{b4}=0.4167$, as indicated in the calculation of Δf_l on pg. 33.

...Also should use W&W's method here.

Example 6:

For outage of branch connected to swing.

Do it three ways as in Example 6:

- a. Use the above "fast" method
- b. Use DC flow with and without outage;
- c. Use W&W's method.

References

- [1] A. Wood and B. Wollenberg, "Power generation, operation, and control," Second edition, John Wiley & Sons, 1996.
- [2] "Transmission system reliability methods," project 1530-1, prepared by Power Technologies Incorporated (PTI), EPRI report EL-2526, 1982.
- [3] Feng Xia; Meliopoulos, A.P.S, "A methodology for probabilistic simultaneous transfer capability analysis"; Power Systems, IEEE Transactions on, Volume 11, Issue 3, Aug. 1996 Page(s):1269 – 1278.
- [4] N. Dag Reppen, presentation slides given at EPRI Workshop on Reliability Analysis, San Diego, California, Feb 27, 2001.

-
- [5] R. A. Leon Candela, "Sensor network design for a secure electric energy infrastructure," M.S., thesis, Iowa State University, April 2005.
 - [6] Q. Chen and J. McCalley, "Identifying High-Risk N-k Contingencies for On-line Security Assessment," IEEE Transactions on Power Systems, 2005.
 - [7] Y. Jiang, A. Kalair, J. McCalley, P. Mitra, A. Gaikwad, D. Pratt, and J. Norris, "Contingency probability estimation for risk-based planning studies using NERC's outage data and Standard TPL-001-4," Proc. of the North American Power Symposium, 2021.
 - [8] NERC Standard TPL-001-5, "Transmission system planning performance requirements," 2020, www.nerc.com/pa/Stand/Reliability%20Standards/TPL-001-5.pdf.

[ⁱ] A. Debs, "Modern Power Systems Control and Operation," Kluwer, 1988.

[ⁱⁱ] A. Monticelli, "State estimation in electric power systems, a generalized approach," Kluwer, 1999.