

도시철도의 DC급전시스템 해석 알고리즘

Computer Algorithm for the Loadflow of the DC Traction Power Supply System

정상기* 홍재승**
Chung, Sang-Gi Hong, Jae-Seung

ABSTRACT

Computer algorithms for the loadflow of the DC traction power supply system are examined. Algorithms to solve the nodal equation are reviewed and the two iterative methods to solve the nonlinear nature of the loadflow are analyzed and tested, which are so called conductance matrix method and current vector iterative method. The result of the analysis tells that the current vector iterative method makes faster convergency and needs less computing time, and it is verified by the test running of the programs based on each of the iterative methods.

1. 서론

도시 전기철도 전력공급시스템의 설계 시 필요한 기초자료 및 판단은 전력공급시스템의 시뮬레이션을 통해서 얻어질 수 있고 시스템의 성능 입증 또한 시뮬레이션을 통해서 이루어진다. 이렇듯 전력공급시스템의 시뮬레이터는 전력공급시스템의 설계 시에나 운영 시에 반드시 필요한 중요한 Tool이다. 시뮬레이터는 차량의 운전곡선을 결정해주는 TPS 프로그램, 운전곡선에 의한 차량의 움직임을 결정하는 프로그램과 현재 차량의 상태 및 위치에서의 Loadflow 해석을 위한 프로그램으로 구성된다. 전력공급시스템은 크게 AC 방식과 DC 방식으로 나눌 수 있는 데 본문에서는 DC 전력공급시스템의 Loadflow 해석 알고리즘을 분석하고 Test program을 통해서 결과를 비교하였다.

2. Loadflow 해석

일반적으로 회로망의 해석은 Loop equation 혹은 Node equation을 연립으로 풀어서 가능해진다. 도시의 전기철도 전력공급시스템을 해석할 경우에도 시스템을 등가회로로 표현한 후 Loop나 Node equation을 세워서 연립으로 풀면 되는 것이다. 이 중 Loop 방정식보다는 Node 방정식을 세우는 게 더 체계적이기 때문에 컴퓨터를 이용한 해석의 경우에는 Node 방정식이 흔히 이용된다. 예를 들어 그림1의 급전시스템은 그림2의 등가회로로 표현할 수 있다(lumped parameter 사용).

* 한국철도기술연구원, 책임연구원, 경량전철연구팀, 비회원

** 한국철도기술연구원, 연구원, 경량전철연구팀, 비회원

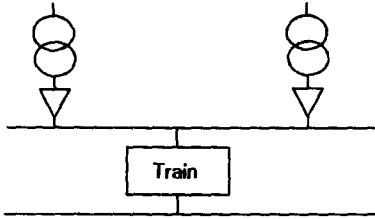


그림 1 간단한 전차선 전력공급

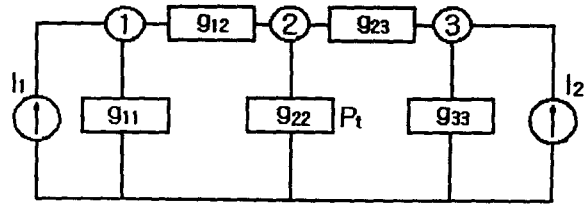


그림 2 Norton 등가회로

그림 2의 등가회로에 대해 node equation을 세우면

$$\begin{pmatrix} g_{11} + g_{12} & -g_{12} & 0 \\ -g_{12} & g_{12} + g_{22} + g_{23} & -g_{23} \\ 0 & -g_{23} & g_{23} + g_{33} \end{pmatrix} \begin{pmatrix} V_1 \\ V_2 \\ V_3 \end{pmatrix} = \begin{pmatrix} I_1 \\ I_2 \\ I_3 \end{pmatrix} \quad (1)$$

일반적으로

$$[G] \cdot [V] = [J] \quad (2)$$

여기서 회로의 node 수가 n개 일 때

[G] : n×n conductance matrix

[V] : 1×n voltage vector

[J] : 1×n current injection vector

식(2)는 v나 I에 대해 1차식 같아 보이지만 Loadflow의 문제에 있어서는 항상 P(power)에 관한 조건이 붙기 때문에 2차식(p=v*i)의 nonlinear equation이 된다. 이러한 nonlinear equation을 computer를 이용해 풀 때는 소위 iterative method를 사용하는 데 이것은 v를 가정하여([V_assumption]) 식 (2)를 풀고 그 결과([V_results])값을 이용하여 새로운 [V_assumption]을 만들고 새로운 [V_assumption]를 이용하여 다시 식 (2)를 푼다. 이런 과정을 반복하여 [V_results]의 값이 바로 전 계산에서 사용된 [V_assumption]과 비교하여 그 차이가 허용범위 이내에 들어 올 때까지 같은 과정을 반복하는 것이다. 따라서 여기에는 크게 두 개의 과정이 있다. 첫째는 일반적인 연립 1차방정식 [G][V]=[I]의 해를 구하는 과정과 첫 번째 과정에서 구해진 [V]의 해를 평가하여 [V]가 허용범위 이내로 들어오지 않는 경우 새로 구해진 [V]를 이용하여 [G] 및 [I]를 수정하여 수정된 [G][V]=[I]의 식을 만드는 것이 두 번째의 과정이다. 따라서 본문에서도 첫 번째 과정과 두 번째 과정을 나누어 검토해보기로 한다.

2.1 1차 연립방정식 [G]·[V]=[I]의 해를 위한 컴퓨터 알고리즘

전차선의 전력공급시스템 해석에 사용되는 알고리즘은 다음의 4단계로 수행되는 것이 보편화 되어있다. (i)sparse matrix formation (ii) optimal ordering (iii) matrix factorization (iv) solution by forward & backward substitution. (i)의 단계는 식 (2)의 [G] matrix를 효율적으로 구성하는 단계이다. 전차선의 전력공급시스템의 node는 대개 좌우의 node와만 연결되어 있다. 그러므로 matrix의 element 수(총 n×n) 중 3×n개만 nonzero값이고 나머지는 모두 0이다. 실제로 node의 개수가 300개 이상이면

[G]matrix의 99%이상이 0이 된다. $((n \times n) - 3 \times n) / (n \times n)$ 이러한 0의 저장은 memory의 비효율적 활용은 물론 matrix 연산 시 불필요한 계산을 하게되기 때문에 [G]를 1차원 배열에 저장하는 여러 가지 sparse technique가 사용되고 있다. (ii)의 과정은 (i)에서 형성된 [G]의 sparseness가 matrix 연산과정에서 생기는 소위 'Fill-in' 때문에 없어지는 것을 최소화하기 위해 row 혹은 column operation의 순서를 정하는 것이다. (iii)의 과정은 [G]를 [L][L]^t로 변환하는 과정이다. (iii)에 의해 식(2)는 식(3)으로 다시 표현된다. (iv)는 식(4)에서 forward substitution에 의해 [x]를 구하고 다시 식(5)에서 backward substitution에 의해 [V]구하는 과정이다.

$$[L] \cdot [L]^t \cdot [V] = [I] \quad (3)$$

식 (3)에서 [G]=[L] · [L]^t 이며 [L]은 low triangular matrix이다.

$$[L] \cdot [X] = [I] \quad (4)$$

$$[L]^t \cdot [V] = [X] \quad (5)$$

그림 3은 이상을 종합한 알고리즘을 보여준다.

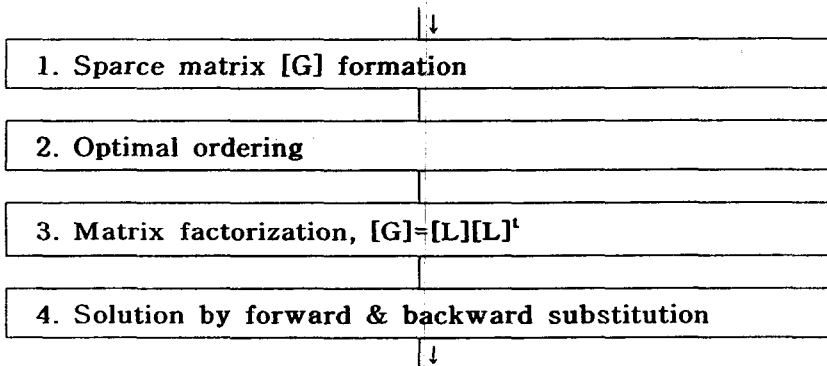


그림 3 [G] · [V]=[I]의 해를 위한 일반적인 컴퓨터 알고리즘

2.2 Iteration 알고리즘

Iterative method는 Loadflow의 조건이 P(여기서는 train의 부하)로 주어지기 때문에 필요하게 된다. 그러므로 P의 조건을 식 (2)의 Nodal equation에 어떻게 집어넣는가에 따라 iterative 방법도 달라지게 된다. 본문에서는 DC Loadflow 해석에 가장 많이 사용되는 두 가지 방법에 대하여 분석 비교하기로 한다. 두 가지 방법은 (i) conductance matrix iterative method와 (ii) current vector iterative method이다. 그림 4에서 P_t는 Train의 Demand Power이며 I₁, I₃은 정류기변전소의 Norton equivalent source이다. (i)의 방법은 그 이름이 암시하듯이 train의 부하를 conductance로 모의하는 것이다(식 6). 이 방법을 사용했을 때 식(2)는 식 (7)과 같이 된다.

$$g_t = +P_t/V^2 \quad (6)$$

$$\begin{pmatrix} g_{11} + g_{12} & -g_{12} \\ -g_{12} & g_{12} + g_{23} + \frac{g_t}{V} \\ & -g_{23} & g_{23} + g_{33} \end{pmatrix} \cdot \begin{pmatrix} V_1 \\ V_2 \\ V_3 \end{pmatrix} = \begin{pmatrix} I_1 \\ 0 \\ I_3 \end{pmatrix} \quad (7)$$

(ii)의 방법에서는 train의 부하를 injection current로 모의하는 것이다(식(8)). 이 방법을 사용하였을 때 식(2)는 식(9)와 같이 표현된다.

$$I_2 = -P_t / V \quad (8)$$

$$\begin{pmatrix} g_{11} + g_{12} & -g_{12} \\ -g_{12} & g_{12} + g_{23} \\ & -g_{23} & g_{23} + g_{33} \end{pmatrix} \cdot \begin{pmatrix} V_1 \\ V_2 \\ V_3 \end{pmatrix} = \begin{pmatrix} I_1 \\ I_2 \\ I_3 \end{pmatrix} \quad (9)$$

식(6)이나 식(8)은 사실 부정확한 모의이다. 식(6)에서 보면 V_2 의 가정에 10%의 error가 있을 때 g_t 는 $1/0.9^2=1.23$, 즉 g_t 가정이 23% 틀렸음을 의미한다. 같은 방법으로 식(8)에서는 I_2 의 가정이 $1/0.9=1.11$, 즉 I_2 의 가정에 11% error를 의미한다. 다시 말하면 부하를 식(8)로 모의하는 것이 식(6)으로 모의하는 것보다 error가 작고 그러므로 좀 더 빨리 converge됨을 의미한다. 이 내용을 간단한 계통(그림4, 5)을 예로 그래프로 표현하여 살펴보자.

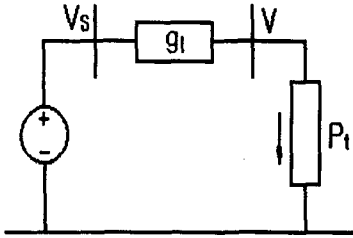


그림 4

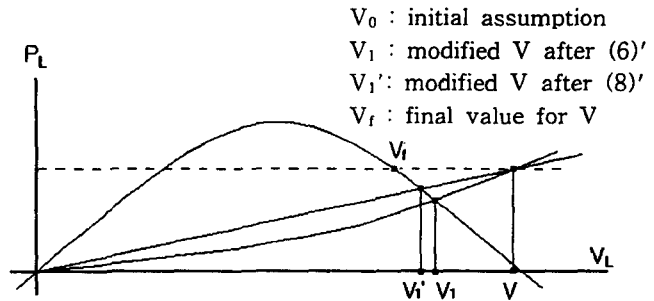


그림 5 convergence의 그래프 표현

$$P_t = V \cdot (V_s - V) \cdot g_t \quad (10)$$

$$\text{또한 } P_t = g_t \cdot V^2 \quad (6')$$

$$P_t = I_t \cdot V \quad (8')$$

그림 5에서 보듯이 (8')에 의한 해가 (6')에 의한 해보다 정답 V_f 에 빠르게 접근하고 있음을 알 수 있다. 또 다른 측면, computer time 면에서 검토하여 보자. 그림 3의 알고리즘에서 conductance matrix iteration 방법을 사용할 경우는 매 iteration 때마다 $[G]$ 를 수정하여야 하기 때문에 step 3 및 4를 반복 수행한다. 반면에 current vector iterative 방법을 사용하였을 때는 $[G]$ 에 변화가 없기 때문에 step 4만 반복하여 수행하면 된다. 즉 current vector iterative 방법을 사용하였을 때 훨씬 작은 computer time이 소요된다고 예상할 수 있다. 그림 6과 그림 7은 이상을 종합한 두 가지 방법에 대한 알고리즘을 보여준다.

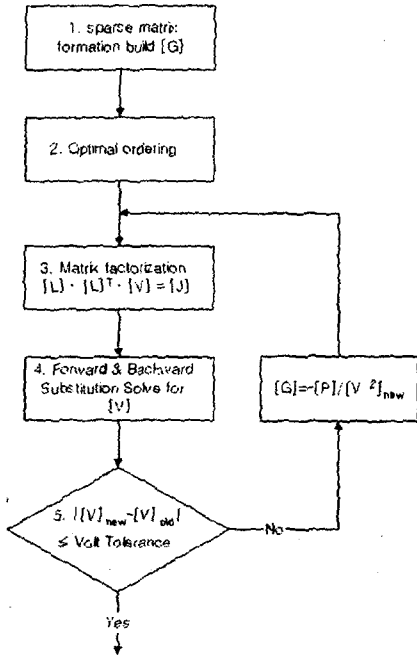


그림 6 Conductance-matrix Iterative Method .

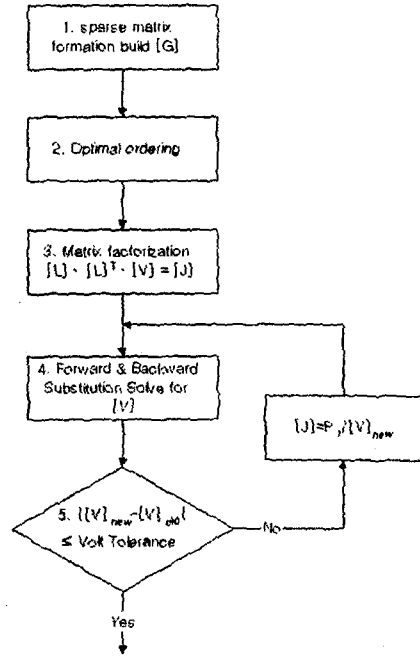


그림 7 Current-vector Iterative Method

3. Test

3.1 Test 노선 및 Data

본 연구에서 Test 노선은 현재 계획되고 있는 부산 반송선을 참조하여 구성하였다. 반송선의 변전소에 대한 Data를 표 3에 나타내고 있다. 변전소의 용량은 모의를 위한 가정치이며 실제값과는 무관하다. 그림 8은 Test 노선을 모델링한 그림이다.

표 1. 변전소 선정 및 용량

번호	변전소명	노선상 위치	용량[p.u]
SS1	고속터미널 변전소	0.000km	40
SS2	안락 변전소	시작-3.472km	70
SS3	동천교 변전소	시작-6.006km	70
SS4	하반송 변전소	시작-9.372km	60
SS5	기지 변전소	시작-12.297km	40

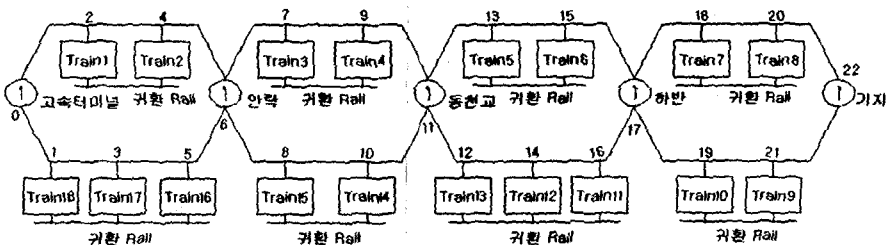


그림 8 시뮬레이션을 위한 모델링회로

각 node number는 고속터미널 변전소를 기준으로 거리가 가까운 순위로 하였다. 본 회로에서 총 node의 개수는 23개가 된다.

전차선로의 1[Km]당 저항은 0.048[Ω/Km]이다. 급전선의 저항이나 귀환선의 저항이 거리에 비례하므로 전차선로의 저항에는 급전선의 저항과 귀환선의 저항이 합하여져 있다. 그러므로 Test 결과에서 나타나는 node의 전압은 차량의 집전장치의 전압이 아니라 집전장치와 귀환레일 사이의 전압임을 유의할 필요가 있다. Data는 750V, 1MW를 각각 전압 base, MW base로 한 Per Unit값이다.

3.2 Test 결과

Test는 Current-vector iteration method와 Conductance-matrix iteration method를 이용하여 각 node의 전압을 구하고 CPU time과 Iteration 회수에 대하여 비교하였다. Test에서 사용된 Tolerance는 각각 0.0001[p.u.]으로 하였다. 이 Tolerance들은 각 node전압의 오차범위이다. 또한 사례연구 1에서는 전차가 Motoring인 경우만 취급하고, 사례연구 2에서는 Breaking인 경우도 고려하여 계산결과를 살펴보겠다.

표 2. 사례연구 1을 위한 입력자료

전차 번호	구분	소비 전력	운행 상태	전차의 위치[m]	전차 번호	구분	소비 전력	운행 상태	전차의 위치[m]
1	상행선	0.5	M	20	10	하행선	0.45	M	9500
2	상행선	0.41	M	2000	11	하행선	0.50	M	9000
3	상행선	0.49	M	4500	12	하행선	0.46	M	7500
4	상행선	0.47	M	5500	13	하행선	0.47	M	6100
5	상행선	0.46	M	7000	14	하행선	0.49	M	5000
6	상행선	0.49	M	8500	15	하행선	0.46	M	3900
7	상행선	0.48	M	11000	16	하행선	0.44	M	3300
8	상행선	0.42	M	12000	17	하행선	0.50	M	1000
9	하행선	0.43	M	11500	18	하행선	0.50	M	300

단 M: Motoring인 경우

표 2는 운행중인 전차가 모두 Motoring인 경우에 해당한다. 표 1과 2를 이용한 전차선로의 node전압을 계산한 결과는 표 3과 같다. Current-vector iteration method와 Conductance-matrix iteration method의 계산결과는 같게 나온다.

표 3. 표 1, 2를 이용한 계산결과(전압)

Node 번호	이름	전압[p.u.]	Node 번호	이름	전압[p.u.]	Node 번호	이름	전압[p.u.]
0	Sub1	0.9616	8	Train3	0.9336	16	Train11	0.9439
1	Train1	0.9605	9	Train14	0.9347	17	Sub4	0.9662
2	Train18	0.9395	10	Train4	0.9417	18	Train10	0.9602
3	Train17	0.9197	11	Sub3	0.9674	19	Train7	0.9320
4	Train2	0.9350	12	Train13	0.9609	20	Train9	0.9472
5	Train16	0.9614	13	Train5	0.9257	21	Train8	0.9549
6	Sub2	0.9712	14	Train12	0.9218	22	Sub5	0.9729
7	Train15	0.9482	15	Train6	0.9264			

이 값들을 그래프로 표현하면 다음과 같다.

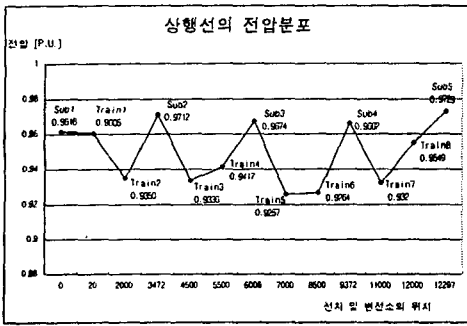


그림 9 상행선의 전압분포

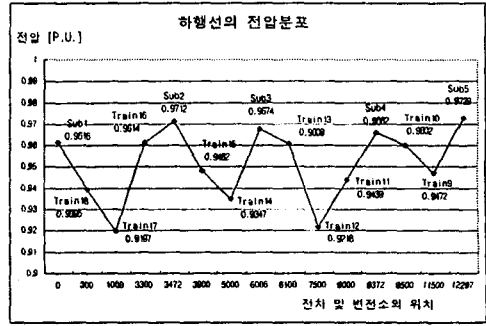


그림 10 하행선의 전압분포

계산시간을 비교하면 Current-vector Iteration Method가 Conductance-matrix Iteration Method보다 약 1.33배 빠르며 또한 iteration의 횟수도 적은 것을 알 수 있다.

표 4. 사례연구 1의 계산시간 및 iteration 횟수

계산방법	계산시간[Sec]	Iteration[횟수]
Current-vector iteration method	0.164835	4
Conductance-matrix iteration method	0.219780	5

두 번째 사례연구에서는 전차의 위치는 같지만, 회생제동인 경우도 포함하여 계산을 수행하였다.

표 5. 사례연구 2를 위한 입력자료

전차 번호	구분	소비 전력	운행 상태	전차의 위치[m]
1	상행선	-0.19	B	20
2	상행선	-0.18	B	2000
3	상행선	0.45	M	4500
4	상행선	-0.20	B	5500
5	상행선	-0.16	B	7000
6	상행선	0.49	M	8500
7	상행선	0.46	M	11000
8	상행선	-0.17	B	12000
9	하행선	-0.19	B	11500
10	하행선	0.50	M	9500
11	하행선	-0.17	B	9000
12	하행선	0.41	M	7500
13	하행선	-0.14	B	6100
14	하행선	0.47	M	5000
15	하행선	0.44	M	3900
16	하행선	-0.18	B	3300
17	하행선	-0.17	B	1000
18	하행선	0.46	M	300

단 M: Motoring인 경우, B: Braking인 경우

표 5의 입력을 이용한 계산결과는 표 6과 같이 나타난다.

표 6. 표 1, 5를 이용한 계산결과(전압)

Node 번호	이름	전압[p.u.]	Node 번호	이름	전압[p.u.]	Node 번호	이름	전압[p.u.]
0	Sub1	0.9984	8	Train3	0.9721	16	Train11	0.9856
1	Train1	0.9988	9	Train14	0.9607	17	Sub4	0.9858
2	Train18	0.9904	10	Train4	0.9913	18	Train10	0.9816
3	Train17	0.9994	11	Sub3	0.9923	19	Train7	0.9655
4	Train2	1.0083	12	Train13	0.9915	20	Train9	1.0023
5	Train16	0.9959	13	Train5	0.9889	21	Train8	0.9937
6	Sub2	0.9930	14	Train12	0.9623	22	Sub5	0.9977
7	Train15	0.9721	15	Train6	0.9630			

이 값들을 그래프로 표현하면 다음과 같다. 계산값들을 살펴보면 Motoring인 경우와는 달리 상행선의 Train2, 하행선의 Train9에서는 회생제동 시 발생하는 전력으로 인하여 전압이 상승하는 것을 살펴볼 수 있다.

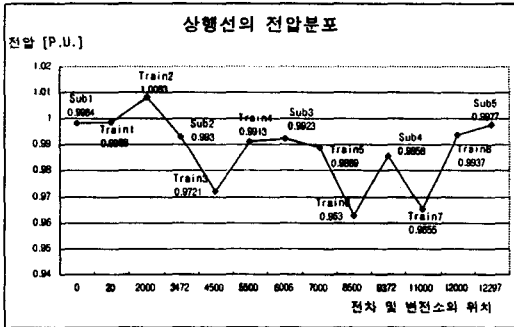


그림 11 상행선의 전압분포

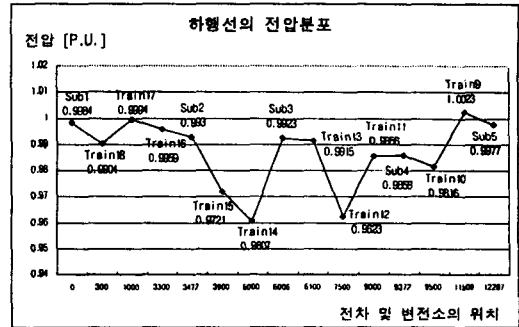


그림 12 하행선의 전압분포

계산시간을 비교하면 Current-vector Iteration Method가 Conductance-matrix Iteration Method보다 약 1.33배 빠르다는 것을 알 수 있다. 또한 iteration의 횟수를 비교하면 Current-vector Iteration method가 적은 것을 알 수 있다.

표 7. 사례연구 2의 계산시간 및 iteration 횟수

계산방법	계산시간[Sec]	Iteration[횟수]
Current-vector iteration method	0.109890	3
Conductance-matrix iteration method	0.164835	4

4. 결론

DC 전력공급시스템의 loadflow 해석은 일반적으로 계통을 Norton의 등가회로로 표현한 후 nodal equation을 해를 구함으로써 수행된다. 그러나 loadflow의 조건이 power로 주어지기 때문에 nodal equation이 nonlinear하게 되고 nonlinear equation을 풀기 위해 iterative 방법이 사용된다. 본문에서는 nodal equation을 푸는 방법과 iteration 방법에 대한 효과적인 알고리즘을 검토하였다. nodal equation을 푸는 방법은 그림 의 알고리즘이 보편화되어 있으며 iterative 방법에서는 current vector iterative 방법이 conductance matrix iterative 방법보다 converging time에서나 computer time에서 훨씬 더 효율적인 것으로 나타났다.

참고문헌

- [1] Alan Jennings, "Matrix Computation for Engineers and Scientists", John Wiley & Sons, Ltd, 1977
- [2] Y. Cai, M.R. Irving, S.H. Case, "Iterative techniques for the solution of Complex DC-rail-traction systems including regenerative braking", IEE Proc. Vol.142, No.5, 1995.