Losses Minimization In Power Systems Using Unified Power Flow Controller

***P. Yella Reddy[1], B. Rama Bhupal Reddy[2]**

[1]Research Scholar, Department of Mathematics, Rayalaseema University, Kurnool-518007, A.P.

Professor & Head, Department of Mathematics, K.S.R.M. College of Engineering (Autonomous), Kadapa-516003.

Email: reddyyella31@gmail.com[1], reddybrb@gmail.com[2]

Abstract

In this paper, the losses in the power system are get optimized and improved the system performance. Initially based on Newton- Raphson method, the losses in the system are calculated for an 5-bus system. These losses are minimized by place a compensating device named Unified Power Flow Controller (UPFC). By reducing the losses an optimization can be achieved which improves the performance. In this Matlab is coded for an IEEE 5-bus system to get optimized results using UPFC.

Key words: Newton-Raphson method, Newton-Raphson Algorithm, series and shunt controllers, loss minimization, optimal control.

Introduction: With the increased loading in existing power transmission systems due to increased demand, the problem of voltage stability along with voltage collapse has become a major concern in power system operation, control and planning. When the voltages at the system buses are low, the losses will also be increased. This study is devoted to develop a technique for improving the voltage and minimizing the loss and hence eliminate voltage instability in a power system. In addition more than 99% of the operating time, the focus of the control system is on loss minimization, which can be done through the installation of controllable devices in the transmission system. The Flexible AC Transmission System (FACTS) devices can be effectively used for power flow control voltage regulation, improvement of the power system stability, minimization of losses and reduction of harmonics. There are two main objectives of FACTS devices which are increasing the power transfer capability of transmission system and restricting power flow over designated lines. Most of the power generation companies has increased the need for enhanced secured operation of power systems, which are facing the threat of voltage stability leading to voltage collapse and also for minimization of active power loss leading to reduction in electricity cost. It is well known that the power flow through transmission line is a function of line impendence, magnitude and face angle of bus voltage. If these parameters can be controlled, the power flow through the transmission line can be controlled in a predetermined manner. Controlling power flow in modern power systems can be made more flexible by the use of power electronics. This is associated with the use of FACTS technology. The UPFC is the FACTS devices in terms of its ability to control our system quantities. It can control the active and reactive power flow through the lines and also bus voltage. UPFC was proposed for real time control and dynamic compensation of AC transmission systems, providing the necessary functional flexibility require to solve many of the problems, such as minimization of system loss, elimination of line over loads and low voltage profiles. UPFC has a series voltage source and a shunt voltage source, allowing independent control of the voltage magnitude and real and reactive power flows along a given transmission line.

The Newton-Raphson Algorithm

In large scale power flow studies the Newton-Raphson method has proved most successful owing its strong convergence characteristics (Peterson and Scott Meyer, 1971: Tinney & Hart 1967). The approach uses iteration to solve the following set of non-linear algebraic equations:

$$\left.\begin{array}{l} f_1(x_1,\ x_2,.....,x_N)=0 \\ f_2(x_1,\ x_2,.....,x_N)=0 \\ \vdots \\ f_N(x_1,\ x_2,.....,x_N)=0 \end{array}\right\} (or)\ F(x)=0 \tag{1}$$

Where F = The set of n non-linear equations
X = Vector of n unknown state variables

This method consists of determining the vector of state variables X by performing a Taylor series expansion of F(X) about an initial estimate $X^{(o)}$.:

$$F(X)=F\left(X^{(0)}\right)+J\left(X^{(0)}\right)\left(X-X^{(0)}\right)+\text{higher order terms} \tag{2}$$

Where, $J(X^{(0)})$ is a matrix of first order partial derivatives of F(X) with respect to X and it is termed as Jacobian, evaluated at $X=X^{(0)}$.

For calculating the vector of state variables X by assuming that $X^{(1)}$ is the value computed by algorithm at iteration 1 and it is close to the $X^{(0)}$ which is initial estimate.

Based on this, all higher order derivative terms in equation (2) can be neglected, hence

$$\underbrace{\begin{bmatrix} f_1\left(X^{(1)}\right) \\ f_2\left(X^{(1)}\right) \\ \vdots \\ f_n\left(X^{(1)}\right) \end{bmatrix}}_{F\left(X^{(1)}\right)} \approx \underbrace{\begin{bmatrix} f_1\left(X^{(0)}\right) \\ f_2\left(X^{(0)}\right) \\ \vdots \\ f_n\left(X^{(0)}\right) \end{bmatrix}}_{F\left(X^{(0)}\right)} + \underbrace{\left.\begin{bmatrix} \frac{\partial f_1(X)}{\partial x_1} & \frac{\partial f_1(X)}{\partial x_2} & \cdots & \frac{\partial f_1(X)}{\partial x_n} \\ \frac{\partial f_2(X)}{\partial x_1} & \frac{\partial f_2(X)}{\partial x_2} & \cdots & \frac{\partial f_2(X)}{\partial x_n} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial f_n(X)}{\partial x_1} & \frac{\partial f_n(X)}{\partial x_2} & \cdots & \frac{\partial f_n(X)}{\partial x_n} \end{bmatrix}\right|_{X=X^{(0)}}}_{J\left(X^{(0)}\right)} \underbrace{\begin{bmatrix} X_1^{(1)}-X_1^{(0)} \\ X_2^{(1)}-X_2^{(0)} \\ \vdots \\ X_n^{(1)}-X_n^{(0)} \end{bmatrix}}_{X^{(1)}-X^{(0)}} \tag{3}$$

In compact form, and generalizing above expression for iteration (i)

$$F\left(X^{(i)}\right)\simeq F\left(X^{(i-1)}\right)+J\left(X^{(i-1)}\right)\cdot\left(X^{(i)}-X^{(i-1)}\right) \tag{4}$$

Where i=1, 2,…., furthermore, it is assumed as $X^{(i)}$ is sufficiently close to solution $X^{(*)}$, then

$$F\left(X^{(i)}\right)\approx F\left(X^{(*)}\right)$$

Hence equation (4) becomes

$$F\left(X^{(i-1)}\right)+J\left(X^{(i-1)}\right)\cdot\left(X^{(i)}-X^{(i-1)}\right)=0 \tag{5}$$

And solving for $X^{(i)}$

$$X^{(i)}=X^{(i-1)}-J^{-1}\left(X^{(i-1)}\right)F\left(X^{(i-1)}\right) \tag{6}$$

The iterative solution can be expressed as a function of the correction vector

$$\Delta X^{(i)}=X^{(i)}-X^{(i-1)}$$

$$\Delta X^{(i)}=-J^{-1}\left(X^{(i-1)}\right)\cdot F\left(X^{(i-1)}\right) \tag{7}$$

And initial estimates are updated using following relations

$$X^{(i)} = X^{(i-1)} + \Delta X^{(i)} \tag{8}$$

The calculations are repeated as many times as required using the most up-to-date values of X in equation (7), this is done until the mismatches ΔX are within a prescribed small tolerance.

In order to apply the NR method to power flow problems, the relevant equations must be expressed inform of equation (7) where X is set of unknown nodal voltage magnitudes and phase angles.

The power mismatch equations Δp and ΔQ are expanded around a base point ($\theta^{(0)}$, $V^{(o)}$), and hence power flow NR algorithm is expressed by following relationship

$$\underbrace{\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix}^{(i)}}_{F\left(X^{(i-1)}\right)} = -\underbrace{\begin{bmatrix} \dfrac{\partial P}{\partial \theta} & \dfrac{\partial P}{\partial V}V \\ \dfrac{\partial Q}{\partial \theta} & \dfrac{\partial Q}{\partial V}V \end{bmatrix}^{(i)}}_{J\left(X^{(i-1)}\right)} \cdot \underbrace{\begin{bmatrix} \Delta \theta \\ \dfrac{\Delta V}{V} \end{bmatrix}^{(i)}}_{\Delta X^{(i)}} \tag{9}$$

The various matrices in the Jacobian may consists of upto (n_b-1) X (n_b-1) elements in the form

$$\left.\begin{matrix} \dfrac{\partial P_k}{\partial \theta_m}, & \dfrac{\partial P_k}{\partial V_m}V_m, \\ \\ \dfrac{\partial Q_k}{\partial \theta_m}, & \dfrac{\partial Q_k}{\partial V_m}V_m, \end{matrix}\right\} \tag{10}$$

Where k=1, ….n_b & m=1, …., n_b but omitting the slack bus entries.

Also, the rows and columns corresponding to reactive power and voltage magnitude for PV buses are discarded when buses k and m are not directly linked by transmission element the corresponding k-m entry in Jacobian is null.

It must be pointed out that the connection terms ΔV_m are divided by V_m to compensate for the fact that Jacobian terms $\left(\dfrac{\partial P_k}{\partial V_m}\right)V_m$ and $\left(\dfrac{\partial Q_k}{\partial V_m}\right)V_m$ are multiplied by V_m.

Consider l^{th} element is connected between buses k and m in figure shown below, for which self and mutual Jacobian term are given as

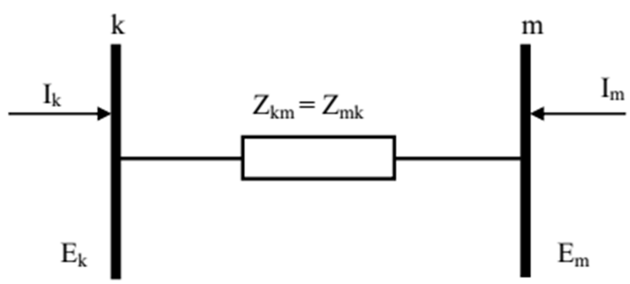


Figure 1 : Equivalent Impedance

Case (i): For k ≠ m:

$$\frac{\partial P_{k,l}}{\partial \theta_{m,l}} = V_k V_m \left[G_{km} Sin(\theta_k - \theta_m) - B_{km} Cos(\theta_k - \theta_m)\right] \tag{11}$$

$$\frac{\partial P_{k,l}}{\partial V_{m,l}} V_{m,l} = V_k V_m \left[G_{km} Cos(\theta_k - \theta_m) + B_{km} Sin(\theta_k - \theta_m) \right] \tag{12}$$

$$\frac{\partial Q_{k,l}}{\partial \theta_{m,l}} = -\frac{\partial P_{k,l}}{\partial \theta_{m,l}} V_{m,l} \tag{13}$$

$$\frac{\partial Q_{k,l}}{\partial V_{m,l}} V_{m,l} = -\frac{\partial P_{k,l}}{\partial \theta_{m,l}} \tag{14}$$

Case (ii): For k = m :

$$\frac{\partial P_{k,l}}{\partial \theta_{k,l}} = -Q_k^{Cal} - V_k^2 B_{kk} \tag{15}$$

$$\frac{\partial P_{k,l}}{\partial V_{k,l}} V_{k,l} = P_k^{Cal} + V_k^2 G_{kk} \tag{16}$$

$$\frac{\partial Q_{k,l}}{\partial \theta_{k,l}} = P_k^{Cal} - V_k^2 G_{kk} \tag{17}$$

$$\frac{\partial Q_{k,l}}{\partial V_{k,l}} V_{k,l} = Q_k^{Cal} - V_k^2 B_{kk} \tag{18}$$

In general, for a bus k containing n transmission elements l, the bus self elements take the following form:

$$\frac{\partial P_k}{\partial \theta_k} = \sum_{l=1}^{n} \frac{\partial P_{k,l}}{\partial \theta_{k,l}} \tag{19}$$

$$\frac{\partial P_k}{\partial V_k} V_k = \sum_{l=1}^{n} \frac{\partial P_{k,l}}{\partial V_{k,l}} V_{k,l} \tag{20}$$

$$\frac{\partial Q_k}{\partial \theta_k} = \sum_{l=1}^{n} \frac{\partial Q_{k,l}}{\partial \theta_{k,l}} \tag{21}$$

$$\frac{\partial Q_k}{\partial V_k} V_k = \sum_{l=1}^{n} \frac{\partial Q_{k,l}}{\partial V_{k,l}} V_{k,l} \tag{22}$$

The mutual elements given by equations (11) to (14) remain the same where we have one transmission element.

After the voltage magnitudes and phase angles have been calculated by iteration, active and reactive power flows thought the transmission system are determined quite straight forwardly.

An important is that, the mismatch power equations ΔP and ΔQ of the slack bus are not included in equation (9) and unknown variables P_{slack} and Q_{slack} are computed and system power loss and power flows have been determined.

One of the main strength of NR method is reliability towards convergence. The method is said to exhibit a quadratic convergence characteristics, for example :

$$f\left(X^{(1)}\right) = 1e-1 \ ,$$

$$f\left(X^{(2)}\right) = 1e-2 \ ,$$

$$f\left(X^{(3)}\right)=1e-4\ ,$$
$$f\left(X^{(4)}\right)=1e-8$$

For maximum mismatch, the solutions are of characteristic is independent of size of the network being solved and the number and kinds of control equipment present in power system.

UNIFIED POWER FLOW CONTROLLER (UPFC)

An equivalent circuit consisting of two co-ordinated synchronous voltage sources should represents the UPFC adequately for the purpose of fundamental frequency steady state analysis such an equivalent circuit is shown in figure (1). The synchronous voltage sources represent the fundamental Fourier series component of the switched voltage waveforms at the converter terminals of the UPFC.

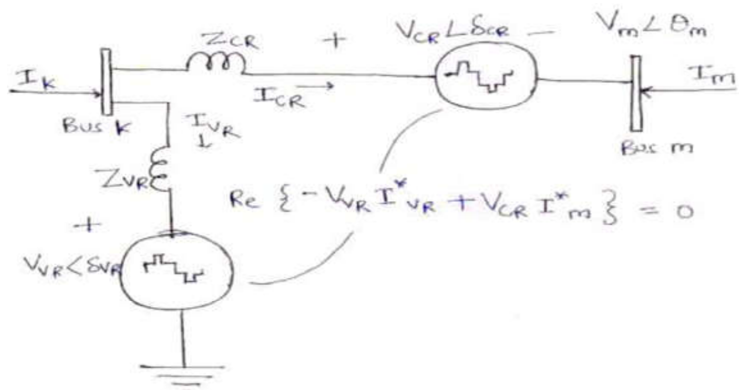


Figure 2: Shows UPFC equivalent circuit

The UPFC voltage sources are

$$E_{VR}=V_{VR}\left(Cos\,\delta_{VR}+j\,Sin\,\delta_{VR}\right) \tag{23}$$

$$E_{CR}=V_{CR}\left(Cos\,\delta_{CR}+j\,Sin\,\delta_{CR}\right) \tag{24}$$

Where V_{VR} and δ_{VR} are the controllable magnitude $\left(V_{VR}\min\leq V_{VR}\leq V_{VR}\max\right)$ and phase angle $\left(0\leq\delta_{VR}\leq 2\pi\right)$ of the voltage source representing the shunt converter.

The magnitude V_{CR} and phase angle δ_{CR} of the voltage source representing the series converter are controlled between limits $\left(V_{CR}\min\leq V_{CR}\leq V_{CR}\max\right)$ and $\left(0\leq\delta_{CR}\leq 2\pi\right)$ respectively.

The phase angle of series injected voltage determines the mode of power flow control. If δ_{CR} is in phase with the nodal voltage angle θ_k, the UPFC regulates the terminal voltage. If δ_{CR} is in quadature with respect to θ_k, it controls active power flow, acting as a phase shifter. If δ_{CR} is in quadature with line current angle then it control active power flow, acting as a variable series compensator. At any other value of δ_{CR}, the UPFC operates as a combination of voltage regulator, variable series compensator, and phase shifter. The magnitude of the series injected voltage determines the amount of power flow to be controlled.

Power Flow model

Based on the equivalent circuit shown in figure (2) and equation (23) and equation (24), the active and reactive power equations are (Fuerte-esquivel and Acha, 1997; Fuerte-esquivel, Acha and Ambriz-perez, 200C), at bus K:

$$\begin{aligned} P_k = V_k^2 G_{kk} &+ V_k V_m \left[G_{km} Cos(\theta_k - \theta_m) + B_{km} Sin(\theta_k - \theta_m)\right] \\ &+ V_k V_{CR}\left[G_{km} Cos(\theta_k - \delta_{cR}) + B_{km} Sin(\theta_k - \delta_{cR})\right] \\ &+ V_k V_{VR}\left[G_{vR} Cos(\theta_k - \delta_{VR}) + B_{VR} Sin(\theta_k - \delta_{VR})\right] \end{aligned} \tag{25}$$

$$\begin{aligned} Q_k = -V_k^2 B_{kk} &+ V_k V_m \left[G_{km} Sin(\theta_k - \theta_m) - B_{km} Cos(\theta_k - \theta_m)\right] \\ &+ V_k V_{CR}\left[G_{km} Sin(\theta_k - \delta_{cR}) - B_{km} Cos(\theta_k - \delta_{cR})\right] \\ &+ V_k V_{VR}\left[G_{vR} Sin(\theta_k - \delta_{VR}) - B_{VR} Cos(\theta_k - \delta_{VR})\right] \end{aligned} \tag{26}$$

At bus m:

$$\begin{aligned} P_m = V_m^2 G_{mm} &+ V_m V_k \left[G_{mk} Cos(\theta_m - \theta_k) + B_{mk} Sin(\theta_m - \theta_k)\right] \\ &+ V_m V_{CR}\left[G_{mm} Cos(\theta_m - \delta_{cR}) + B_{mm} Sin(\theta_m - \delta_{cR})\right] \end{aligned} \tag{27}$$

$$\begin{aligned} Q_m = -V_m^2 B_{mm} &+ V_m V_k \left[G_{mk} Sin(\theta_m - \theta_k) - B_{mk} Cos(\theta_m - \theta_k)\right] \\ &+ V_m V_{CR}\left[G_{mm} Sin(\theta_m - \delta_{cR}) - B_{mm} Cos(\theta_m - \delta_{cR})\right] \end{aligned} \tag{28}$$

Series converter:

$$\begin{aligned} P_{cR} = V_{cR}^2 G_{mm} &+ V_{cR} V_k \left[G_{km} Cos(\delta_{cR} - \theta_k) + B_{km} Sin(\delta_{cR} - \theta_k)\right] \\ &+ V_{cR} V_m\left[G_{mm} Cos(\delta_{cR} - \theta_m) + B_{mm} Sin(\delta_{cR} - \theta_m)\right] \end{aligned} \tag{29}$$

$$\begin{aligned} Q_{cR} = -V_{cR}^2 B_{mm} &+ V_{cR} V_k \left[G_{km} Sin(\delta_{cR} - \theta_k) - B_{km} Cos(\delta_{cR} - \theta_k)\right] \\ &+ V_{cR} V_m\left[G_{mm} Sin(\delta_{cR} - \theta_m) - B_{mm} Cos(\delta_{cR} - \theta_m)\right] \end{aligned} \tag{30}$$

Shunt Converter:

$$P_{VR} = V_{VR}^2 G_{VR} + V_{VR} V_k \left[G_{VR} Cos(\delta_{VR} - \theta_k) + B_{VR} Sin(\delta_{VR} - \theta_k)\right] \tag{31}$$

$$Q_{VR} = V_{VR}^2 B_{VR} + V_{VR} V_k \left[G_{VR} Sin(\delta_{VR} - \theta_k) - B_{VR} Cos(\delta_{VR} - \theta_k)\right] \tag{32}$$

Assuming loss less converter values, the active power supplied to the shunt converter, P_{VR} equals the active power demanded by the series converter P_{CR}: that is

$$P_{VR} + P_{CR} = 0 \tag{33}$$

Further more, if the coupling transformer are assumed to contain no resistance then the active power at bus k matches the active power at bus m, accordingly,

$$P_{VR} + P_{CR} = P_K + P_m = 0 \tag{34}$$

The UPFC power equations, in linearised form, are combined with those of the AC network. For the case when the UPFC controls the following parameters:

1. Voltage magnitudes at the shunt converter (bus K),
2. Active power flows from m to bus k, and
3. Reactive power injected at bus m, taking bus m to be a P_Q bus the linearised system equation is as follows:

$$\begin{bmatrix} \Delta P_k \\ \Delta P_m \\ \Delta Q_k \\ \Delta Q_m \\ \Delta P_{mk} \\ \Delta Q_{mk} \\ \Delta P_{bb} \end{bmatrix} = \begin{bmatrix} \frac{\partial P_k}{\partial \theta_k} & \frac{\partial P_k}{\partial \theta_m} & \frac{\partial P_k}{\partial V_{VR}} V_{VR} & \frac{\partial P_k}{\partial V_m} V_m & \frac{\partial P_k}{\partial \delta_{CR}} & \frac{\partial P_k}{\partial V_{CR}} V_{CR} & \frac{\partial P_k}{\partial \delta_{VR}} \\ \frac{\partial P_m}{\partial \theta_k} & \frac{\partial P_m}{\partial \theta_m} & 0 & \frac{\partial P_m}{\partial V_m} V_m & \frac{\partial P_m}{\partial \delta_{CR}} & \frac{\partial P_m}{\partial V_{CR}} V_{CR} & 0 \\ \frac{\partial Q_k}{\partial \theta_k} & \frac{\partial Q_k}{\partial \theta_m} & \frac{\partial Q_k}{\partial V_{VR}} V_{VR} & \frac{\partial Q_k}{\partial V_m} V_m & \frac{\partial Q_k}{\partial \delta_{CR}} & \frac{\partial Q_k}{\partial V_{CR}} V_{CR} & \frac{\partial Q_k}{\partial \delta_{VR}} \\ \frac{\partial Q_m}{\partial \theta_k} & \frac{\partial Q_m}{\partial \theta_m} & 0 & \frac{\partial Q_m}{\partial V_m} V_m & \frac{\partial Q_m}{\partial \delta_{CR}} & \frac{\partial Q_m}{\partial V_{CR}} V_{CR} & 0 \\ \frac{\partial P_{mk}}{\partial \theta_k} & \frac{\partial P_{mk}}{\partial \theta_m} & 0 & \frac{\partial P_{mk}}{\partial V_m} V_m & \frac{\partial P_{mk}}{\partial \delta_{CR}} & \frac{\partial P_{mk}}{\partial V_{CR}} V_{CR} & 0 \\ \frac{\partial Q_{mk}}{\partial \theta_k} & \frac{\partial Q_{mk}}{\partial \theta_m} & 0 & \frac{\partial Q_{mk}}{\partial V_m} V_m & \frac{\partial Q_{mk}}{\partial \delta_{CR}} & \frac{\partial Q_{mk}}{\partial V_{CR}} V_{CR} & 0 \\ \frac{\partial P_{bb}}{\partial \theta_k} & \frac{\partial P_{bb}}{\partial \theta_m} & \frac{\partial P_{bb}}{\partial V_{VR}} V_{VR} & \frac{\partial P_{bb}}{\partial V_m} V_m & \frac{\partial P_{bb}}{\partial \delta_{CR}} & \frac{\partial P_{bb}}{\partial V_{CR}} & \frac{\partial P_{bb}}{\partial \delta_{VR}} \end{bmatrix} \cdot \begin{bmatrix} \Delta \theta_k \\ \Delta \theta_m \\ \frac{\Delta V_{VR}}{V_{VR}} \\ \frac{\Delta V_m}{V_m} \\ \Delta \delta_{CR} \\ \frac{\Delta V_{CR}}{V_{CR}} \\ \Delta \delta_{VR} \end{bmatrix} \quad (35)$$

Where ΔP_{bb} is the power mismatch given by equation (33)

If voltage control at bus k is deactivated, the third column of equation (35) is placed by partial derivatives of the bus and UPFC mismatch powers with respect to the bus voltage magnitude V_k. Moreover, the voltage magnitude increment of the shunt source $\frac{\Delta V_{VR}}{V_{VR}}$ is replaced by the voltage magnitude increment t bus k, $\frac{\Delta V_k}{V_k}$.

If both buses, k and m are P_Q the linearised system of equations is as follows:

$$\begin{bmatrix} \Delta P_k \\ \Delta P_m \\ \Delta Q_k \\ \Delta Q_m \\ \Delta P_{mk} \\ \Delta Q_{mk} \\ \Delta P_{bb} \end{bmatrix} = \begin{bmatrix} \frac{\partial P_k}{\partial \theta_k} & \frac{\partial P_k}{\partial \theta_m} & \frac{\partial P_k}{\partial V_k} V_k & \frac{\partial P_k}{\partial V_m} V_m & \frac{\partial P_k}{\partial \delta_{CR}} & \frac{\partial P_k}{\partial V_{CR}} V_{CR} & \frac{\partial P_k}{\partial \delta_{VR}} \\ \frac{\partial P_m}{\partial \theta_k} & \frac{\partial P_m}{\partial \theta_m} & \frac{\partial P_m}{\partial V_k} V_k & \frac{\partial P_m}{\partial V_m} V_m & \frac{\partial P_m}{\partial \delta_{CR}} & \frac{\partial P_m}{\partial V_{CR}} V_{CR} & 0 \\ \frac{\partial Q_k}{\partial \theta_k} & \frac{\partial Q_k}{\partial \theta_m} & \frac{\partial Q_k}{\partial V_k} V_k & \frac{\partial Q_k}{\partial V_m} V_m & \frac{\partial Q_k}{\partial \delta_{CR}} & \frac{\partial Q_k}{\partial V_{CR}} V_{CR} & \frac{\partial Q_k}{\partial \delta_{VR}} \\ \frac{\partial Q_m}{\partial \theta_k} & \frac{\partial Q_m}{\partial \theta_m} & \frac{\partial Q_m}{\partial V_k} V_k & \frac{\partial Q_m}{\partial V_m} V_m & \frac{\partial Q_m}{\partial \delta_{CR}} & \frac{\partial Q_m}{\partial V_{CR}} V_{CR} & 0 \\ \frac{\partial P_{mk}}{\partial \theta_k} & \frac{\partial P_{mk}}{\partial \theta_m} & \frac{\partial P_{mk}}{\partial V_k} V_k & \frac{\partial P_{mk}}{\partial V_m} V_m & \frac{\partial P_{mk}}{\partial \delta_{CR}} & \frac{\partial P_{mk}}{\partial V_{CR}} V_{CR} & 0 \\ \frac{\partial Q_{mk}}{\partial \theta_k} & \frac{\partial Q_{mk}}{\partial \theta_m} & \frac{\partial Q_{mk}}{\partial V_k} V_k & \frac{\partial Q_{mk}}{\partial V_m} V_m & \frac{\partial Q_{mk}}{\partial \delta_{CR}} & \frac{\partial Q_{mk}}{\partial V_{CR}} V_{CR} & 0 \\ \frac{\partial P_{bb}}{\partial \theta_k} & \frac{\partial P_{bb}}{\partial \theta_m} & \frac{\partial P_{bb}}{\partial V_k} V_k & \frac{\partial P_{bb}}{\partial V_m} V_m & \frac{\partial P_{bb}}{\partial \delta_{CR}} & \frac{\partial P_{bb}}{\partial V_{CR}} V_{CR} & \frac{\partial P_{bb}}{\partial \delta_{VR}} \end{bmatrix} \cdot \begin{bmatrix} \Delta \theta_k \\ \Delta \theta_m \\ \frac{\Delta V_k}{V_k} \\ \frac{\Delta V_m}{V_m} \\ \Delta \delta_{CR} \\ \frac{\Delta V_{CR}}{V_{CR}} \\ \Delta \delta_{CR} \end{bmatrix} \quad (36)$$

In this case, V_{VR} is maintained at a fixed value within prescribed limits $V_{VR} \min \leq V_{VR} \leq V_{VR} \max$.

Unified Power Flow Controller Computer Program in Matlab Code

This Program incorporates the UPFC model within the Newton–Raphson power flow program. The functions Power Flows Data, Y Bus, and PQ flows are also used here. In the main UPFC Newton–Raphson program, the function UPFC Data is added to read the UPFC data, UPFC Newton Raphson replaces Newton Raphson, and UPFCPQ flows is used to calculate power flows and losses in the UPFC.

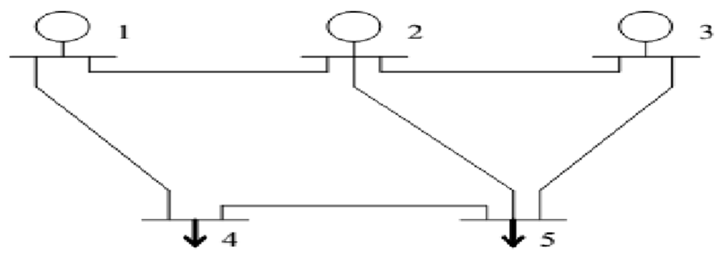


Figure 3: Five-bus test network with one unified power flow controller, and power flow result

IEEE 5 – BUS SYSTEM DATA

Table 1. BUS DATA FOR IEEE 5-BUS SYSTEM

BUS Code P	Assumed Bus Voltage	Generation		Load	
		Megawatts	Megavars	Megawatts	Megavars
1	1.06 + j0.0	0	0	0	0
2	1.0 + j0.0	40	30	20	10
3	1.0 + j0.0	0	0	45	15
4	1.0 + j0.0	0	0	40	05
5	1.0 + j0.0	0	0	60	10

Table .2 :LINE DATA FOR IEEE 5-BUS SYSTEM

Bus Code p-q	Line Impedance Zpq		Line Charging $\sqrt{Y_{pq}/2}$
	R Per Unit	X Per unit	X Per unit
1-2	0.02	0.06	
1-3	0.08	0.24	0.0 + j0.025
2-3	0.06	0.25	0.0 + j0.020
2-4	0.06	0.18	0.0 + j0.020
2-5	0.04	0.12	0.0 + j0.015
3-4	0.01	0.03	0.0 + j0.010
4-5	0.08	0.24	0.0 + j0.025

Table 3: MW Limits for branches in IEEE -5 Bus System

Line	MW Limit (P.u)
1-2	0.8
1-3	0.3
2-3	0.2
2-4	0.2
2-5	0.6
3-4	0.1
4-5	0.1

PROGRAM: Program written in Matlab to incorporate the unified power flow controller (UPFC) model within the Newton–Raphson power flow algorithm

```
% - - - Main UPFC Program
PowerFlowsData; %Function to read network data
UPFCdata; %Function to read the UPFC data
[YR,YI] = YBus(tlsend,tlrec,tlresis,tlreac,tlsuscep,tlcond,ntl,nbb);
[VM,VA,it,Vcr,Tcr,Vvr,Tvr] = UPFCNewtonRaphson(tol,itmax,ngn,nld,...
nbb,bustype,genbus,loadbus,PGEN,QGEN,QMAX,QMIN,PLOAD,QLOAD,YR,YI,...
VM,VA,NUPFC,UPFCsend,UPFCrec,Xcr,Xvr,Flow,Psp,PSta,Qsp,QSta,Vcr,...
Tcr,VcrLo,VcrHi,Vvr, Tvr,VvrLo,VvrHi,VvrTar,VvrSta);
[PQsend,PQrec,PQloss,PQbus] = PQflows(nbb,ngn,ntl,nld,genbus,...
loadbus,tlsend,tlrec,tlresis,tlreac,tlcond,tlsuscep,PLOAD,QLOAD,... VM,VA);
[UPFC_PQsend,UPFC_PQrec,PQcr,PQvr] = PQUPFCpower(nbb,VA,VM,NUPFC,...
UPFCsend,UPFCrec,Xcr,Xvr,Vcr,Tcr,Vvr,Tvr); %Print results
it                  %Number of iterations
VM                  %Nodal voltage magnitude (p.u.)
VA=VA*180/pi %Nodal voltage phase angles (deg)
Sources=[Vcr,Tcr*180/pi,Vvr,Tvr*180/pi]       %Final source voltage para-
% meters
UPFC_PQsend   %Active and reactive powers in sending bus (p.u.)
UPFC_PQrec    %Active and reactive powers in receiving bus (p.u.)
%End of MAIN UPFC PROGRAM
%Carry out iterative solution using the Newton–Raphson method
function [VM,VA,it,Vcr,Tcr,Vvr,Tvr] = UPFCNewtonRaphson(tol,itmax,...
ngn,nld, nbb,bustype,genbus,loadbus,PGEN,QGEN,QMAX,QMIN,PLOAD,...
QLOAD,YR,YI,VM,VA, NUPFC,UPFCsend,UPFCrec,Xcr,Xvr,Flow,Psp,PSta,...
Qsp,QSta,Vcr,Tcr,VcrLo,VcrHi,Vvr,Tvr,VvrLo,VvrHi,VvrTar,VvrSta);
%       GENERAL SETTINGS flag = 0;
it = 1;
%       CALCULATE NET POWERS
[PNET,QNET] = NetPowers(nbb,ngn,nld,genbus,loadbus,PGEN,QGEN,...
PLOAD,QLOAD);
while ( it < itmax & flag==0 )
  % CALCULATED POWERS
  [PCAL,QCAL] = CalculatedPowers(nbb,VM,VA,YR,YI);
  % CALCULATED UPFC POWERS
  [PspQsend,PspQrec,PQcr,PQvr,PCAL,QCAL] = UPFCCalculatedpower...
  (nbb,VA, VM,NUPFC,UPFCsend,UPFCrec,Xcr,Xvr,Vcr,Tcr,Vvr,Tvr,PCAL,...
  QCAL);
  % POWER MISMATCHES
  [DPQ,DP,DQ,flag] = PowerMismatches(nbb,tol,bustype,flag,PNET,QNET,...
  PCAL,QCAL);
  % UPFC POWER MISMATCHES
  [DPQ,flag] = UPFCPowerMismatches(flag,tol,nbb,DPQ,VM,VA,NUPFC,Flow,...
  Psp,PSta,Qsp,QSta,PspQsend,PspQrec,PQcr,PQvr); if flag == 1
    break
  end
  % JACOBIAN FORMATION
  [JAC] = NewtonRaphsonJacobian(nbb,bustype,PCAL,QCAL,DPQ,VM,VA,YR,...
  YI);
  % MODIFICATION OF THE JACOBIAN FOR UPFC
  [JAC] = UPFCJacobian(nbb,JAC,VM,VA,NUPFC,UPFCsend,UPFCrec,Xcr,...
  Xvr,Flow,PSta,QSta,Vcr,Tcr,Vvr,Tvr,VvrSta);
  %     SOLVE JOCOBIAN D = JAC\DPQ';
  %     UPDATE THE STATE VARIABLES VALUES
```

```
  [VA,VM] = StateVariablesUpdating(nbb,D,VA,VM,it);
  % UPDATE THE TCSC VARIABLES
  [VM,Vcr,Tcr,Vvr,Tvr] = UPFCUpdating(nbb,VM,D,NUPFC,UPFCsend,PSta,...
  QSta,Vcr,Tcr,Vvr,Tvr,VvrTar,VvrSta);
  %CHECK VOLTAGE LIMITS IN THE CONVERTERS
  [Vcr,Vvr] = UPFCLimits(NUPFC,Vcr,VcrLo,VcrHi,Vvr,VvrLo,VvrHi);
  it = it + 1;
end
%Function to calculate injected bus powers by the UPFC
function [UPFC_PQsend,UPFC_PQrec,PQcr,PQvr,PCAL,QCAL] = UPFCCalcula...
tedpower(nbb,VA,VM,NUPFC,UPFCsend,UPFCrec,Xcr,Xvr,Vcr,Tcr,Vvr,...
Tvr,PCAL,QCAL);
for ii = 1 : NUPFC
  Bkk = - 1/Xcr(ii)-1/Xvr(ii);
  Bmm = -1/Xcr(ii);
  Bmk = 1/Xcr(ii);
  Bvr = 1/Xvr(ii);
  for kk = 1 : 2
    A1 = VA(UPFCsend(ii))-VA(UPFCrec(ii));
    A2 = VA(UPFCsend(ii))-Tcr(ii);
    A3 = VA(UPFCsend(ii))-Tvr(ii);
    % Computation of Conventional Terms
    Pkm = VM(UPFCsend(ii))*VM(UPFCrec(ii))*Bmk*sin(A1);
    Qkm = - VM(UPFCsend(ii))^2*Bkk - VM(UPFCsend(ii))*VM(UPFCrec(ii))...
    *Bmk*cos(A1);
    % Computation of Shunt Converters Terms
    Pvrk = VM(UPFCsend(ii))*Vvr(ii)*Bvr*sin(A3); Qvrk = -
    VM(UPFCsend(ii))*Vvr(ii)*Bvr*cos(A3); if kk == 1
      % Computation of Series Converters Terms
      Pcrk = VM(UPFCsend(ii))*Vcr(ii)*Bmk*sin(A2); Qcrk = -
      VM(UPFCsend(ii))*Vcr(ii)*Bmk*cos(A2); %Power in bus k
      Pk = Pkm + Pcrk + Pvrk;
      Qk = Qkm + Qcrk + Qvrk;
      UPFC_PQsend(ii) = Pk + Qk*i;
      PCAL(UPFCsend(ii)) = PCAL(UPFCsend(ii)) + Pk;
      QCAL(UPFCsend(ii)) = QCAL(UPFCsend(ii)) + Qk;
      %Power in Series Converter
      Pcr = Vcr(ii)*VM(UPFCsend(ii))*Bmk*sin(-A2);
      Qcr = - Vcr(ii)^2*Bmm - Vcr(ii)*VM(UPFCsend(ii))*Bmk*cos(-A2); %Power in
      Shunt Converter
      Pvr = Vvr(ii)*VM(UPFCsend(ii))*Bvr*sin(-A3);
      Qvr = Vvr(ii)^2*Bvr - Vvr(ii)*VM(UPFCsend(ii))*Bvr*cos(-A3); PQvr(ii) = Pvr +
      Qvr*i;
    else
      % Computation of Series Converters Terms
      Pcrk = VM(UPFCsend(ii))*Vcr(ii)*Bkk*sin(A2);
      Qcrk = - VM(UPFCsend(ii))*Vcr(ii)*Bkk*cos(A2);
      %Power in bus m
      Pcal = Pkm + Pcrk;
      Qcal = Qkm + Qcrk;
      UPFC_PQrec(ii) = Pcal + Qcal*i;
      PCAL(UPFCsend(ii)) = PCAL(UPFCsend(ii)) + Pcal;
      QCAL(UPFCsend(ii)) = QCAL(UPFCsend(ii)) + Qcal;
      %Power in Series Converter
      Pcr = Pcr + Vcr(ii)*VM(UPFCsend(ii))*Bkk*sin(-A2);
```

```
        Qcr = Qcr - VM(UPFCsend(ii))*Vcr(ii)*Bkk*cos(-A2);
        PQcr(ii) = Pcr + Qcr*i;
      end
      send = UPFCsend(ii);
      UPFCsend(ii) = UPFCrec(ii);
      UPFCrec(ii) = send;
      Beq = Bmm;
      Bmm = Bkk;
      Bkk = Beq;
    end
end
%Function to compute power mismatches with UPFC
function [DPQ,flag] = UPFCPowerMismatches(flag,tol,nbb,DPQ,VM,VA,...
NUPFC,Flow,Psp,PSta,Qsp,QSta,UPFC_PQsend,UPFC_PQrec,PQcr,PQvr); iii = 0;
for ii = 1 : NUPFC
    index = 2*(nbb + ii) + iii;
    if PSta(ii) == 1
      if Flow(ii) == 1
        DPQ(index-1) = Psp(ii) - real(UPFC_PQsend(ii)); else
        DPQ(index-1) = - Psp(ii) - real(UPFC_PQrec(ii)); end
    else
      DPQ(index-1) = 0;
    end
    if QSta(ii) == 1
      if Flow(ii) == 1
        DPQ(index) = Qsp(ii) - imag(UPFC_PQrec(ii)); else
        DPQ(index) = - Qsp(ii) - imag(UPFC_PQrec(ii)); end
    else
      DPQ(index) = 0;
    end
    DPQ(index + 1) = - real(PQcr(ii) + PQvr(ii)); iii=iii+1;
end
%       Check for convergence if ( abs(DPQ) < tol )
    flag = 1; end
%Function to add the UPFC elements to the Jacobian matrix function [JAC] =
UPFCJacobian(nbb,JAC,VM,VA,NUPFC,UPFCsend,...
UPFCrece,Xcr,Xvr,Flow,PSta,QSta,Vcr,Tcr,Vvr,Tvr,VvrSta); iii = 0;
for ii = 1 : NUPFC
    indexQ=2*(nbb + ii) + iii;
    indexP=indexQ-1;
    indexL=indexQ + 1;
    if VvrSta(ii) == 1
      JAC(:,2*UPFCsend(ii)) = 0.0;
    end
    Bmm = - 1/Xcr(ii)-1/Xvr(ii);
    Bkk = -1/Xcr(ii);
    Bmk = 1/Xcr(ii);
    Bvr = 1/Xvr(ii);
    for kk = 1 : 2
      A1 = VA(UPFCsend(ii))-VA(UPFCrece(ii));
      A2 = VA(UPFCsend(ii))-Tcr(ii);
      A3 = VA(UPFCsend(ii))-Tvr(ii);
      % Computation of Conventional Terms
      Hkm = - VM(UPFCsend(ii))*VM(UPFCrece(ii))*Bmk*cos(A1);
      Nkm = VM(UPFCsend(ii))*VM(UPFCrece(ii))*Bmk*sin(A1);
```

```
% Computation of Shunt Converters Terms
Hvr = -VM(UPFCsend(ii))*Vvr(ii)*Bvr*cos(A3); Nvr =
VM(UPFCsend(ii))*Vvr(ii)*Bvr*sin(A3);
%   Computation of Series Converters Terms if kk == 1
  Hcr = - VM(UPFCsend(ii))*Vcr(ii)*Bmk*cos(A2); Ncr =
  VM(UPFCsend(ii))*Vcr(ii)*Bmk*sin(A2);
else
  Hcr = - VM(UPFCsend(ii))*Vcr(ii)*Bkk*cos(A2);
  Ncr = VM(UPFCsend(ii))*Vcr(ii)*Bkk*sin(A2); end
if kk == 1
  JAC(2*UPFCsend(ii)-1,2*UPFCsend(ii)-1) = JAC(2*UPFCsend...
  (ii)-1, 2*UPFCsend(ii)-1) - VM(UPFCsend(ii))^2*Bmm; if VvrSta(ii) == 1
    JAC(2*UPFCsend(ii)-1,2*UPFCsend(ii)) = Nvr;
    JAC(2*UPFCsend(ii),2*UPFCsend(ii)) = Hvr; else
    JAC(2*UPFCsend(ii)-1,2*UPFCsend(ii)) = JAC(2*UPFCsend(ii)-1,...
    2*UPFCsend(ii)) - Nkm + Nvr;
    JAC(2*UPFCsend(ii),2*UPFCsend(ii)) = JAC(2*UPFCsend...
    (ii),2*UPFCsend(ii)) - Hkm + Hvr + 2*VM(UPFCsend(ii))^2*Bmk;
  end
  JAC(2*UPFCsend(ii)-1,indexL) = Hvr;
  JAC(2*UPFCsend(ii),indexL) = - Nvr; else
  JAC(2*UPFCsend(ii)-1,2*UPFCsend(ii)-1) = JAC(2*UPFCsend... (ii)-
  1,2*UPFCsend(ii)-1) + VM(UPFCsend(ii))^2*Bmk;
  JAC(2*UPFCsend(ii),2*UPFCsend(ii)) = JAC(2*UPFCsend(ii),... 2*UPFCsend(ii))
  + VM(UPFCsend(ii))^2*Bmk;
  JAC(2*UPFCsend(ii)-1,indexL) = 0.0;
  JAC(2*UPFCsend(ii),indexL) = 0.0;
end
JAC(2*UPFCsend(ii)-1,2*UPFCrece(ii)-1) = JAC(2*UPFCsend(ii)-1,...
2*UPFCrece(ii)-1) + Hkm;
JAC(2*UPFCsend(ii),2*UPFCrece(ii)-1) = JAC(2*UPFCsend(ii),...
2*UPFCrece(ii)-1) - Nkm;
if VvrSta(ii) == 1 & kk == 2
  JAC(2*UPFCsend(ii)-1,2*UPFCrece(ii)) = 0.0;
  JAC(2*UPFCsend(ii),2*UPFCrece(ii)) = 0.0; else
  JAC(2*UPFCsend(ii)-1,2*UPFCrece(ii)) = JAC(2*UPFCsend(ii)-1,...
  2*UPFCrece(ii)) + Nkm;
  JAC(2*UPFCsend(ii),2*UPFCrece(ii)) = JAC(2*UPFCsend(ii),...
  2*UPFCrece(ii)) + Hkm;
end
%   Computation of Active Power Controlled Jacobian's Terms if PSta(ii) == 1

  if (Flow(ii) == 1 & kk == 1) j (Flow(ii) == -1 & kk == 2) if kk == 1
      JAC(indexP, 2*UPFCsend(ii)-1) = - Hkm - Hcr - Hvr; JAC(indexP,
      2*UPFCsend(ii)) = - Nkm + Ncr;
  JAC(indexP, 2*UPFCrece(ii)-1) = - Hkm; if VvrSta(ii) == 1
    JAC(indexP, 2*UPFCrece(ii)) = 0.0; else
    JAC(indexP, 2*UPFCrece(ii)) = Nkm; end
  JAC(indexP, indexP) = Hcr;
  if QSta(ii) == 1
    JAC(indexP, indexQ) = Ncr;
  else
    JAC(indexP, indexQ) = 0.0;
  end
else
```

```
        JAC(indexP, 2*UPFCsend(ii)-1) = - Hkm - Hcr;
        JAC(indexP, 2*UPFCsend(ii)) = Nkm + Ncr;
        JAC(indexP, 2*UPFCrece(ii)-1) = Hkm;
        if VvrSta(ii) == 1
          JAC(indexP, 2*UPFCrece(ii)) = 0.0; else

          JAC(indexP, 2*UPFCrece(ii)) = Nkm; end
        JAC(indexP, indexP) = Hcr;
        if QSta(ii) == 1
          JAC(indexP, indexQ) = Ncr;
        else
          JAC(indexP, indexQ) = 0.0;
        end
        JAC(indexP, indexL) = 0.0;
      end
    end
    JAC(2*UPFCsend(ii)-1, indexP) = + Hcr;
    JAC(2*UPFCsend(ii), indexP) = - Ncr;
  else
    JAC(indexP, indexP) = 1.0;
  end
  %       Computation of Rective Power Controlled Jacobian's Terms if QSta(ii) == 1

    if (Flow(ii) == 1 & kk == 1) j (Flow(ii) == -1 & kk == 2) if kk == 1
        JAC(indexQ, 2*UPFCsend(ii)-1) = - Nkm + Ncr;
        JAC(indexQ, 2*UPFCsend(ii)) = - 2*VM(UPFCsend(ii))^2*Bmm - ...
        Hkm + Hcr;
        JAC(indexQ, 2*UPFCrece(ii)-1) = Nkm; JAC(indexQ, indexQ) = Hcr;
        if VvrSta(ii) == 1
          JAC(indexQ, 2*UPFCrece(ii)) = 0.0;
            else
              JAC(indexQ, 2*UPFCrece(ii)) = Hkm; end
            if PSta(ii) == 1
              JAC(indexQ, indexP) = - Ncr;
            else
              JAC(indexQ, indexP) = 0.0;
            end
          else
            JAC(indexQ, 2*UPFCsend(ii)-1) = Nkm + Ncr;
                                JAC(indexQ, 2*UPFCsend(ii)) = - 2*VM(UPFCsend(ii)) ...
            ^2*Bkk + Hkm + Hcr;
            JAC(indexQ, 2*UPFCrece(ii)-1) = - Nkm; JAC(indexQ, indexQ) = Hcr; if
            VvrSta(ii) == 1
              JAC(indexQ, 2*UPFCrece(ii)) = 0.0; else
              JAC(indexQ, 2*UPFCrece(ii)) = Hkm; end
            if PSta(ii) == 1
              JAC(indexQ, indexP) = - Ncr;
            else
              JAC(indexQ, indexP) = 0.0;
            end
          end
        end
        JAC(2*UPFCsend(ii)-1, indexQ) = Ncr;
        JAC(2*UPFCsend(ii), indexQ) = Hcr;
      else
```

```
      JAC(indexQ, indexQ) = 1.0;
    end
    temp = UPFCsend(ii);
    UPFCsend(ii) = UPFCrece(ii);
    UPFCrece(ii) = temp;
  end
  A1 = Tcr(ii) - VA(UPFCsend(ii));
  A2 = Tcr(ii) - VA(UPFCrece(ii));
  A3 = Tvr(ii) - VA(UPFCsend(ii));
  Hcrk = - Vcr(ii)*VM(UPFCsend(ii))*Bmk*cos(A1); Ncrk =
  Vcr(ii)*VM(UPFCsend(ii))*Bmk*sin(A1); Hcrm =
  Vcr(ii)*VM(UPFCrece(ii))*Bmk*cos(A2); Ncrm = -
  Vcr(ii)*VM(UPFCrece(ii))*Bmk*sin(A2); Hvrk = -
  Vvr(ii)*VM(UPFCsend(ii))*Bvr*cos(A3); Nvrk =
  Vvr(ii)*VM(UPFCsend(ii))*Bvr*sin(A3);
  JAC(indexL, 2*UPFCsend(ii)-1) = Hcrk + Hvrk; if VvrSta == 1
    JAC(indexL, 2*UPFCsend(ii)) = Nvrk; else
    JAC(indexL, 2*UPFCsend(ii)) = Nvrk + Ncrk; end
  JAC(indexL, 2*UPFCrece(ii)-1) = Hcrm; JAC(indexL, 2*UPFCrece(ii)) = Ncrm;
  JAC(indexL, indexL) = - Hvrk; if PSta == 1
    JAC(indexL, indexP) = - Hcrk - Hcrm; else
    JAC(indexL, indexP) = 0.0;
  end
  if QSta == 1
    JAC(indexL, indexQ) = Ncrk + Ncrm;
  else
    JAC(indexL, indexP) = 0.0;
  end
  iii   = iii + 1;
end
%Function to update the UPFC state variables
function [VM,Vcr,Tcr,Vvr,Tvr] = UPFCUpdating(nbb,VM,D,NUPFC,...
UPFCsend,PSta, QSta,Vcr,Tcr,Vvr,Tvr,VvrTar,VvrSta); iii = 0;
for ii = 1 : NUPFC
  indexQ=2*(nbb + ii) + iii;
  indexP=indexQ-1;
  indexL=indexQ + 1;
  if PSta(ii) == 1
    Tcr(ii) = Tcr(ii) + D(indexP);
  end
  if QSta(ii) == 1
    Vcr(ii) = Vcr(ii) + D(indexQ)*Vcr(ii); end
  if VvrSta(ii) == 1
    Vvr(ii) = Vvr(ii) + D(2*UPFCsend(ii),1)*Vvr(ii);
    VM(UPFCsend(ii)) = VvrTar(ii);
  end
  Tvr(ii) = Tvr(ii) + D(indexL);
  iii   = iii +1;
end
%Function to check the voltage sources limits in the UPFC
function [Vcr,Vvr] = UPFCLimits(NUPFC,Vcr,VcrLo,VcrHi,Vvr,VvrLo,...
VvrHi);
for ii = 1 : NUPFC
  % Check Magnitude Voltage Limits
  if abs(Vcr(ii)) < VcrLo(ii) j abs(Vcr(ii)) > VcrHi(ii)
```

```
    if abs(Vcr(ii)) < VcrLo(ii)
      Vcr(ii) = VcrLo(ii);
    elseif abs(Vcr(ii)) > VcrHi(ii)
      Vcr(ii) = VcrHi(ii);
    end
  end
  if abs(Vvr(ii)) < VvrLo(ii) j abs(Vvr(ii)) > VvrHi(ii)
    if abs(Vvr(ii)) < VvrLo(ii)
      Vvr(ii) = VvrLo(ii);
    elseif abs(Vvr(ii)) > VvrHi(ii)
      Vvr(ii) = VvrHi(ii);
    end
  end
end
%Function to calculate the power flows in the UPFC controller function
[UPFC_PQsend,UPFC_PQrec,PQcr,PQvr] = PQUPFCpower(nbb,...
VA,VM, NUPFC,UPFCsend,UPFCrec,Xcr,Xvr,Vcr,Tcr,Vvr,Tvr); for ii = 1 : NUPFC
  Bkk = - 1/Xcr(ii)-1/Xvr(ii);
  Bmm = -1/Xcr(ii);
  Bmk = 1/Xcr(ii);
  Bvr = 1/Xvr(ii);
  for kk = 1 : 2
    A1 = VA(UPFCsend(ii))-VA(UPFCrec(ii));
    A2 = VA(UPFCsend(ii))-Tcr(ii);
    A3 = VA(UPFCsend(ii))-Tvr(ii);
 % Computation of Conventional Terms
    Pkm = VM(UPFCsend(ii))*VM(UPFCrec(ii))*Bmk*sin(A1);
    Qkm = - VM(UPFCsend(ii))^2*Bkk - VM(UPFCsend(ii))...
    *VM(UPFCrec(ii))*Bmk*cos(A1);
% Computation of Shunt Converters Terms
    Pvrk = VM(UPFCsend(ii))*Vvr(ii)*Bvr*sin(A3); Qvrk = -
    VM(UPFCsend(ii))*Vvr(ii)*Bvr*cos(A3); if kk == 1
% Computation of Series Converters Terms
      Pcrk = VM(UPFCsend(ii))*Vcr(ii)*Bmk*sin(A2);
      Qcrk = - VM(UPFCsend(ii))*Vcr(ii)*Bmk*cos(A2); %Power in bus k
      Pk = Pkm + Pcrk + Pvrk;
      Qk = Qkm + Qcrk + Qvrk;
      UPFC_PQsend(ii) = Pk + Qk*i;
%Power in Series Converter
      Pcr = Vcr(ii)*VM(UPFCsend(ii))*Bmk*sin(-A2);
      Qcr = - Vcr(i)^2*Bmm - Vcr(ii)*VM(UPFCsend(ii))*Bmk*cos(-A2); %Power in
 Shunt Converter
      Pvr = Vvr(ii)*VM(UPFCsend(ii))*Bvr*sin(-A3);
      Qvr = Vvr(ii)^2*Bvr - Vvr(ii)*VM(UPFCsend(ii))*Bvr*cos(-A3); PQvr(ii) = Pvr +
      Qvr*i;
    else
% Computation of Series Converters Terms
      Pcrk = VM(UPFCsend(ii))*Vcr(ii)*Bkk*sin(A2);
      Qcrk = - VM(UPFCsend(ii))*Vcr(ii)*Bkk*cos(A2); %Power in bus m
      Pcal = Pkm + Pcrk;
      Qcal = Qkm + Qcrk;
      UPFC_PQrec(ii) = Pcal + Qcal*i; %Power in Series Converter
      Pcr = Pcr + Vcr(ii)*VM(UPFCsend(ii))*Bkk*sin(-A2); Qcr = Qcr -
      VM(UPFCsend(ii))*Vcr(ii)*Bkk*cos(-A2); PQcr(ii) = Pcr + Qcr*i;
    end
```

```
    send = UPFCsend(ii);
    UPFCsend(ii) = UPFCrec(ii);
    UPFCrec(ii) = send;
    Beq = Bmm;
    Bmm = Bkk;
    Bkk = Beq;
  end
end
```

RESULT:- (Without UPSC):

VOLTAGE PROFILE

BUS	VOLTAGE	ANGLE(rad)	ANGLE(deg)
1	1.060000	0.000000	0.000000
2	0.992621	-0.033974	-1.946578
3	0.981487	-0.080021	-4.584891
4	0.977982	-0.085585	-4.903633
5	0.964484	-0.099667	-5.710501

POWER FLOW BETWEEN LINES

Bus code	Real power	Reactive power	Loss
1-2	0.896246	0.868027	0.028772
1-3	0.420055	0.192529	0.016028
2-3	0.244368	-0.034026	0.003649
2-4	0.276769	-0.024054	0.004666
2-5	0.546338	0.053060	0.012304
3-4	0.194745	0.040617	0.000420
4-5	0.066428	0.009338	0.000462

RESULT:- (With UPSC):

```
a =
   1.0600
a =
   1.0600
a =
   1.0600
a =
   1.0600
a =
   1.0600
```

VOLTAGE PROFILE

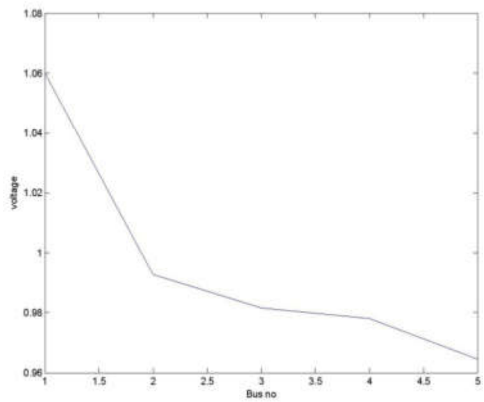
BUS	VOLTAGE	ANGLE(rad)	ANGLE(deg)
1	1.060000	0.000000	0.000000
2	1.010313	-0.018443	-1.056683
3	1.001060	-0.061380	-3.516792
4	0.999897	-0.064197	-3.678243
5	1.009032	-0.052078	-2.983876

POWER FLOW BETWEEN LINES

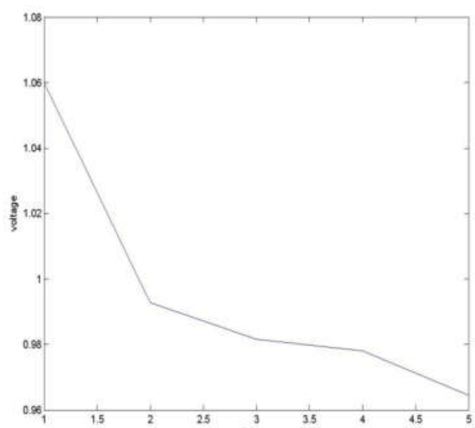
Bus code	Real power	Reactive power	Loss
1-2	0.560495	0.660294	0.014165
1-3	0.324683	0.132328	0.009338
2-3	0.234197	-0.041363	0.003250
2-4	0.250330	-0.039522	0.003705
2-5	0.261802	-0.086987	0.002887
3-4	0.096292	-0.003199	0.000093
4-5	-0.057175	-0.043685	0.000290

WAVEFORMS:

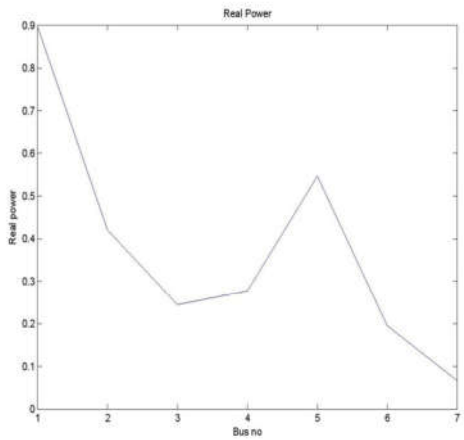
WITHOUT UPFC :



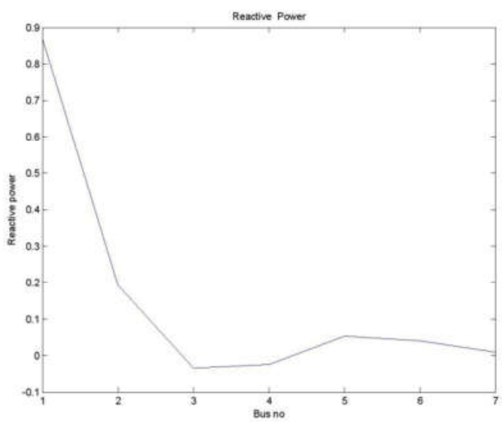
Graph 1:



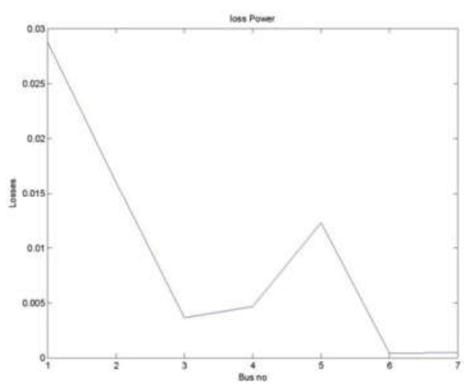
Graph 2:



Graph 3:

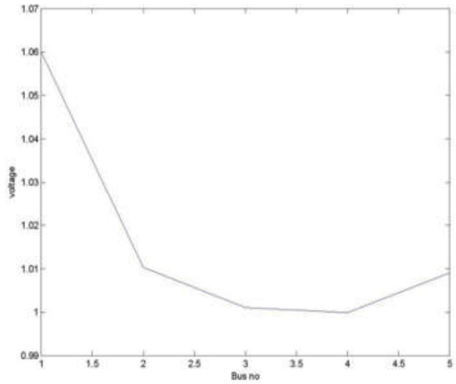


Graph 4:

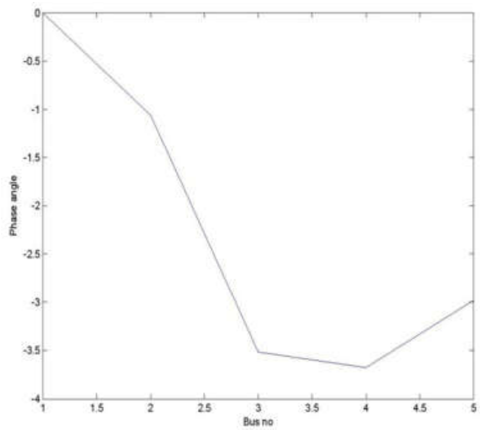


Graph 5:

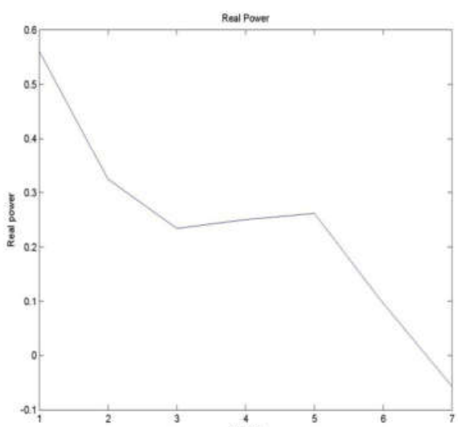
WITH UPFC :



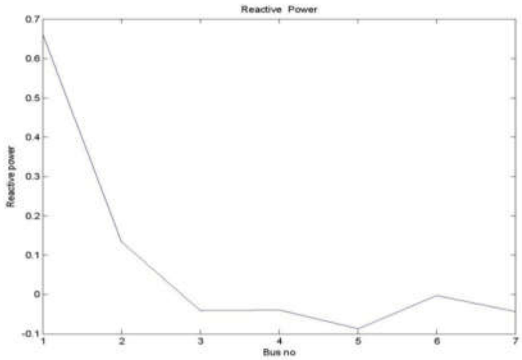
Graph 1:



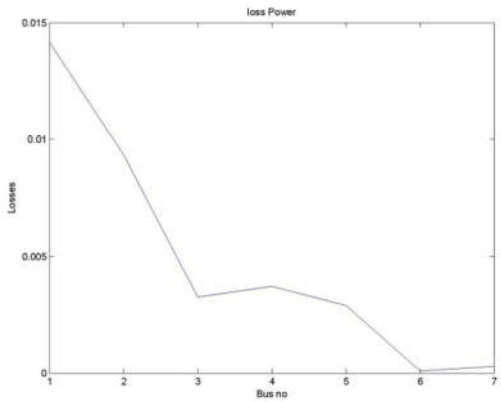
Graph 2:



Graph 3:



Graph 4:



Graph 5:

CONCLUSION

In this paper the UPFC is used to investigate the performance of the Unified Power Flow Controller(UPFC) and thereby the load flow studies are done by incorpating the Voltage Source Model of UPFC in the Newton Raphson (N-R) algorithm.

The N-R algorithm is able to control the flow of power and voltage individually as well as simultaneously. The result for a IEEE-5 Bus system has been presented above without and with UPFC and are compared in terms of Real and Reactive power flow and the Voltage magnitude. Hence it was observed that the UPFC regulates the real and reactive power of the buses and the lines and it also controls the voltage of the bus within specified limits, thereby reduces the total losses in the lines.

References:

1. Acha, E., 1993, 'A Quasi-Newton Algorithm for the Load Flow Solution of Large Networks with FACTS Controlled Branches', in Proceedings of the 28th Universities Power Engineering Conference 1993, Staffordshire University, Volume 1, 21–23 September, Stafford, UK, pp. 153–156.
2. Ambriz-Pe´rez, H., Acha, E., and Fuerte-Esquivel, C.R., 2000, 'Advanced SVC Models for Newton–Raphson Load Flow and Newton Optimal Power Flow Studies', IEEE Trans. Power Systems 15(1) 129–136.
3. Arrillaga, J., Arnold, C.P., 1990, Computer Analysis of Power Systems, John Wiley & Sons, Chichester. Chang, S.K., Brandwajn V., 1988, 'Adjusted Solutions in Fast Decoupled Load Flow', IEEE Trans. Power Systems 3(2) 726–733.
4. Erinmez, I.A. (Ed.), 1986, 'Static Var Compensators', Working Group 38-01, Task Force No. 2 on SVC, Conseil International des Grands Reseaux Electriques (CIGRE´).

5. Fuerte-Esquivel, C.R., Acha, E., 1996, ‘Newton–Raphson Algorithm for the Reliable Solution of Large Power Networks with Embedded FACTS Devices’, IEE Proceedings : Generation, Transmission and Distribution 143(5) 447– 454.
6. Fuerte-Esquivel, C.R., Acha, E., 1997, ‘The Unified Power Flow Controller: A Critical Comparison of Newton–Raphson UPFC Algorithms in Power Flow Studies’ IEE Proceedings : Generation, Transmission and Distribution 144(5) 437– 444.
7. Fuerte-Esquivel, C.R., Acha, E., Ambriz-Pe´rez, H., 2000a, ‘Integrated SVC and Step-down Transformer Model for Newton–Raphson Load Flow Studies’, IEEE Power Engineering Review 20(2) 45–46.
8. Fuerte-Esquivel, C.R., Acha, E., Ambriz-Pe´rez, H., 2000b, ‘AThyristor Controlled Series Compensator Model for the Power Flow Solution of Practical Power Networks’, IEEE Trans. Power Systems 15(1) 58–64.
9. Fuerte-Esquivel, C.R., Acha, E., Ambriz-Pe´rez, H., 2000c, ‘A Comprehensive UPFC Model for the Quadratic Load Flow Solution of Power Networks’, IEEE Trans. Power Systems 15(1) 102–109.
10. Fuerte-Esquivel, C.R., Acha, E., Tan, S.G., Rico, J.J., 1998, ‘Efficient Object Oriented Power System Software for the Analysis of Large-scale Networks Containing FACTS Controlled Branches’, IEEE Trans. Power System 3(2) 464–472.
11. Grainger, J.J., Stevenson W.D. Jr., 1994, Power System Analysis, McGraw-Hill, New York.
12. Hingorani, N.G., Gyugyi, L., 2000, Understanding FACTS Concepts and Technology of Flexible AC Transmission Systems, Institute of Electrical and Electronic Engineers, New York.
13. IEEE/CIGRE´: (Institute of Electrical and Electronic Engineers/Conseil International des Grands Reseaux E´lectriques), 1995, ‘FACTS Overview’, Special Issue, 95TP108, IEEE Service Center, Piscataway, NJ.
14. IEEE SSCWG (Special Stability Controls Working Group), 1995, ‘Static Var Compensator Models for Power Flow and Dynamic Performance Simulation’, IEEE Trans. Power Systems 9(1) 229–240.
15. Kundur P.P., 1994, Power System Stability and Control, McGraw-Hill, New York.
16. Nabavi-Niaki, A., Iravani, M.R., 1996, ‘Steady-state and Dynamic Models of Unified Power Flow Controller (UPFC) for Power System Studies’, IEEE Trans. Power Delivery 11(4) 1937–1943.
17. Noroozian M., Andersson G., 1993, ‘Power Flow Control by Use of Controllable Series Components’, IEEE Trans. on Power Delivery 8(3) 1420–1429.
18. Song, Y.H., Johns, A.T., 1999, Flexible AC Transmission Systems (FACTS), Institution of Electric Engineers, London.
19. Tinney W.F., Hart C.E., 1967, ‘Power Flow Solution by Newton’s Method’, IEEE Trans. Power Apparatus and Systems PAS-96(11) 1449–1460.