KATHMANDU UNIVERSITY JOURNAL OF SCIENCE, ENGINEERING AND TECHNOLOGY
VOL.I, No.III, JANUARY, 2007.

MATLAB SYMBOLIC COMPUTATION FOR THE STEADY STATE MODELING OF SYMMETRICALLY LOADED SELF EXCITED INDUCTION GENERATOR

Gurung K., Freere P.

Department of Electrical and Electronics Engineering
Kathmandu University, P.O.Box: 6250, Kathmandu, Nepal

Corresponding Author E-Mail: krishna@ku.edu.np

ABSTRACT
This paper presents the use of Matlab symbolic computation technique to model and simulate self excited induction generator. In this technique, the computer itself carries out both the tedious job of deriving the complex coefficients of the polynomial equations and solving them. Hence the modeling and programming becomes very simple yet versatile. Good agreement between the results obtained from the conventional method and that obtained using symbolic computation validates the effectiveness of this new technique.

Key words: Symbolic computation, induction generator, self excitation.

INTRODUCTION
Self excited induction generators (SEIG) have become very popular in micro and pico hydro systems of Nepal. This is mainly because they are robust, easily available and inexpensive. They require little maintenance and hence are very much suitable for remote area applications.

Both the frequency and magnetizing reactance of SEIG vary with load in order to maintain an exact balance of active and reactive power across the air gap. Hence, it is a crucial step in the steady state analysis of a SEIG to determine the per unit frequency F and the magnetizing reactance X_m for given machine parameters, speed, excitation capacitance and load impedance [1]. The balance of active and reactive power across the air gap can be realized by equating the real and imaginary terms of total admittances, connected across the terminal representing air gap, respectively to zero.

The usual practice is to derive the complex coefficients of the non-linear equations manually and solve them using numerical methods. The mathematical manipulations required are tedious, time consuming and liable to human error. It requires tremendous human effort for accurate programming and debugging. The model lacks flexibility as the coefficients are valid only for a given circuit configuration. Inclusion of the core loss resistance or load inductance will increase the order of the equations.

The above shortcomings can be overcome by the use of symbolic computation technique in Matlab. The symbolic computation technique allows one to solve the SEIG governing equations without having to derive the complex coefficients of polynomial equations manually. A single command ‘solve’ can be used to solve multiple equations and the user does not need to bother about the numerical methods involved. This makes the modeling and simulation of SEIG very simple yet versatile. Good agreement between the results obtained

from the conventional method and that obtained using symbolic computation validates the effectiveness of this new technique.

Steady state modeling of self excited induction generator

Most of the steady state models of SEIG developed by different researchers are based on per phase equivalent circuit. These models use the following two basic methods; i) Loop impedance method and ii) Nodal admittance method. The steady state model based on nodal admittance method and used in [2] is presented here. This model makes assumptions that the load is purely resistive, core loss component is neglected and the machine parameters (except for magnetizing reactance) remain constant.

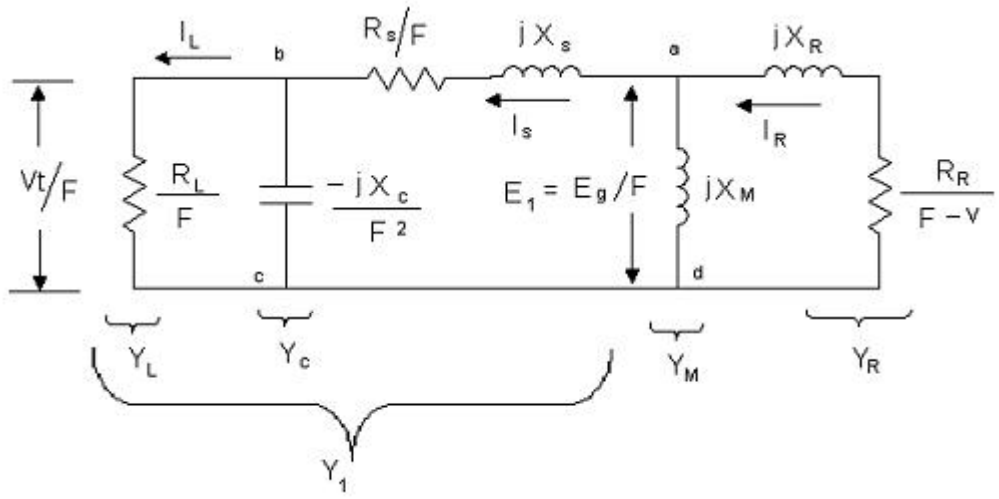

Figure 1. Per phase equivalent circuit of SEIG

Figure 1. above shows the per phase equivalent circuit of SEIG, where the different symbols represent:

Rs, R_R, R_L	→	p.u. stator, rotor and load resistances respectively
Xs, X_R, X_M, X_C	→	p.u stator, rotor, magnetizing and excitation reactances respectively
Ys, Y_R, Y_M, Y_L , Y_C	→	p.u stator, rotor, magnetizing, load and excitation admittances respectively.
F	→	p.u.frequency
v	→	p.u. rotor speed
Is, I_R, I_L	→	p.u. stator, rotor and load currents respectively
Eg, Vt	→	p.u. air gap and load terminal voltage respectively

The total current at node ‘a’ in the above figure can be written as

$$E_1(Y_1+Y_M+Y_R) = 0 \qquad \text{……..(1)}$$

Where

$$Y_1 = \frac{(Y_C + Y_L)Y_S}{Y_C + Y_L + Y_S} \qquad Y_C = \frac{1}{-(jX_C / F^2)}$$

$$Y_L = \frac{1}{(R_L / F\ )} \qquad Y_S = \frac{1}{(R_S / F) + jX_S} \qquad \text{…….(2)}$$

$$Y_M = -\frac{1}{jX_M} \qquad Y_R = \frac{1}{\frac{R_R}{F - v} + jX_R}$$

Under self-excitation $E_1 \neq 0$, therefore sum of total admittance connected across the air gap must be zero i.e.

$$Y_1+Y_M+Y_R = 0 \qquad \text{……..(3)}$$

As the admittances are complex quantities, the real and imaginary parts of equation 3 can be equated to zero.

Therefore,

$$\text{Real}(Y_1+Y_M+Y_R) = 0 \qquad \text{……..(4)}$$

$$\text{Imag}(Y_1+Y_M+Y_R) = 0 \qquad \text{……..(5)}$$

For given value of shaft speed, generator parameters, excitation capacitance and load impedance, solution of equation 4 gives the p.u. output frequency F. The corresponding value of magnetizing reactance X_M can then be found from equation 5 using the value of F obtained from 4. After determining the values of F and X_M, the air gap voltage Eg can be determined using the experimentally obtained magnetization curve, which relates Eg/F and X_M. Now, different quantities can be calculated using the following relations describing figure 1.

$$I_S = \frac{E_g / F}{\frac{R_S}{F} + jX_S - \frac{jX_C R_L}{F^2 R_L - jFX_C}} \qquad I_R = \frac{-E_g / F}{\frac{R_R}{F - v} + jX_R}$$

$$I_L = \frac{-jX_C I_S}{R_L F - jX_C} \qquad V_t = I_L R_L \qquad \text{……..(6)}$$

$$P_{in} = \frac{-3R_R |I_R|^2}{F - v} \qquad P_{out} = 3|I_L|^2 R_L$$

Conventional solution methods

In the conventional methods, complex coefficients of equations 4 and 5 are manually derived and then solved using numerical methods. In the above example, equations 4 and 5 can be simplified to equations 6 and 7 respectively.

$$A_5F^5 + A_4F^4 + A_3F^3 + A_2F^2 + A_1F + A_0 = 0 \qquad \text{.....(7)}$$

$$X_M = \frac{-1}{\frac{X_R}{[R_R/(F-v)]^2 + X_R^2} + \frac{X_{ac}}{R_{ac}^2 + X_{ac}^2}} \qquad \text{.....(8)}$$

where

$$A_0 = -vR_R\left(\frac{R_3}{R_L}\right)^2 \qquad A_1 = R_R\left(\frac{R_3}{R_L}\right)^2 + R_3\left(\frac{R_R}{R_L}\right)^2 + v^2R_3\left(\frac{X_R}{R_L}\right)^2$$

$$A_2 = -2vR_3\left(\frac{X_R}{R_L}\right)^2 - vR_R\left\{\left(\frac{R_S}{X_C}\right)^2 + \left(\frac{X_S}{R_L}\right)^2 - 2\left(\frac{X_S}{X_C}\right)\right\}$$

$$A_3 = R_R\left[\left(\frac{X_S}{R_L}\right)^2 + \left(\frac{R_S}{X_C}\right)^2 - 2\left(\frac{X_S}{X_C}\right)\right] + R_3\left(\frac{X_R}{R_L}\right)^2 + R_S\left(\frac{R_R}{X_C}\right)^2 + v^2R_S\left(\frac{X_R}{X_C}\right)^2$$

$$A_4 = -v\left[R_R\left(\frac{X_S}{X_C}\right) + 2R_S\left(\frac{X_R}{X_C}\right)^2\right] \qquad A_5 = R_R\left(\frac{X_S}{X_C}\right)^2 + R_S\left(\frac{X_R}{X_C}\right)^2 \qquad \text{.....(9)}$$

$$R_{bc} = \frac{R_L X_C^2}{\left[F\left(F^2R_L^2 + X_C^2\right)\right]} \qquad X_{bc} = \frac{R_L^2 X_C}{\left(F^2R_L^2 + X_C^2\right)}$$

$$R_{ac} = \frac{R_S}{F} + R_{bc} \qquad X_{ac} = X_S - X_{bc} \qquad R_3 = R_S + R_L$$

Disadvantages of conventional solution methods

Although the conventional methods are effective in simulating the SEIG performance, they have common disadvantages as correctly pointed out by T.F. Chan in [1] and [3]. They can be listed out as:

1. All the coefficients of the non-linear equations or a higher order polynomial need to be derived manually. The mathematical manipulations are tedious, time-consuming and prone to human errors.
2. The expressions for the coefficients are very long and complicated, which require tremendous human effort for accurate programming and debugging.

3. The model lacks flexibility as the coefficients are valid only for a given circuit configuration. For example, inclusion of the core-loss resistance or the addition of compensation capacitive reactance will change the order of the equations .

Many researchers have proposed different techniques to tackle these problems recently [1-5]. However they still require some degree of manual manipulation of the mathematical equations before these techniques are applied. This paper proposes a novel technique using MATLAB symbolic computation, which eliminates all the above problems.

Symbolic computation in Matlab

MATLAB can compute on symbolic variables just as on constants. This exempts one from the tedious job of manual manipulation of the complex equations to obtain the final two equations 7 and 8. It can also solve several simultaneous equations hence the user does not need to use numerical methods to solve the complex equations 7 and 8 obtained after the manipulation. Since MATLAB performs both the jobs, steady state modeling and simulation of SEIG becomes very simple and effective [6]. Few simple examples below illustrate the symbolic computation that can be done in MATLAB (version 6.5 or above).

Command	Returns
syms a b	creates two symbolic variables a & b.
x = (a + b)^2	$x = (a + b)^2$
x = expand(x)	$x = a^2 + 2ab + b^2$
y = (a - b)^2	$y = (a - b)^2$
y = expand(y)	$y = a^2 - 2ab + b^2$
z = x^2 + y^2	$z = (a^2 + 2ab + b^2)^2+(a^2 - 2ab + b^2)^2$
z = simple (z)	$z = 2a^4 + 12a^2b^2 + 2b^4$

Equation 3 can be written down in the similar fashion.

Similarly, the following commands can be used to solve the Z for a and b if the Z happens to be a complex quantity which is true for equation 3.

Command	Returns
zreal = real(z)	- Equates real terms of Z to zero. - assigns the equation name as ‘zreal’.
zimag = imag(z)	- Equates imaginary terms of Z to zero. - Assigns the equation name as ‘zimag’.

[a,b] = solve(zreal, zimag)	Solves zreal and zimag for a and b
a = double(a) b = double(b)	Returns the numeric values of a and b

In this way equation 3 can be solved directly using Matlab Symbolic Computation technique. This exempts one from having to manually derive equations 7 and 8 and using numerical methods to solve them.

RESULTS AND DISCUSSIONS

In order to verify the validity of the new technique the generator equivalent circuit used in [2] was simulated using Matlab symbolic computation and the results were compared with the one obtained from the conventional method. The SEIG used for this purpose is a 3-phase, 4-pole, 60 Hz, 1 kW, 380V, 2.27 A, Y-connected squirrel cage induction machine whose per phase equivalent circuit parameters in pu are:

$$R_s = 0.1 \qquad X_s = 0.2 \qquad R_r = 0.06 \qquad X_r = 0.2.$$

And the magnetization curve is represented mathematically as

$$\frac{E_g}{F} = 1.12 + 0.078X_M - 0.146X_M^2 \quad ; \; 0 < X_M < 3. \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(10)$$

The voltage regulation curve of the SEIG for different loading at constant speed (rated) from both the methods are plotted in the figure below.

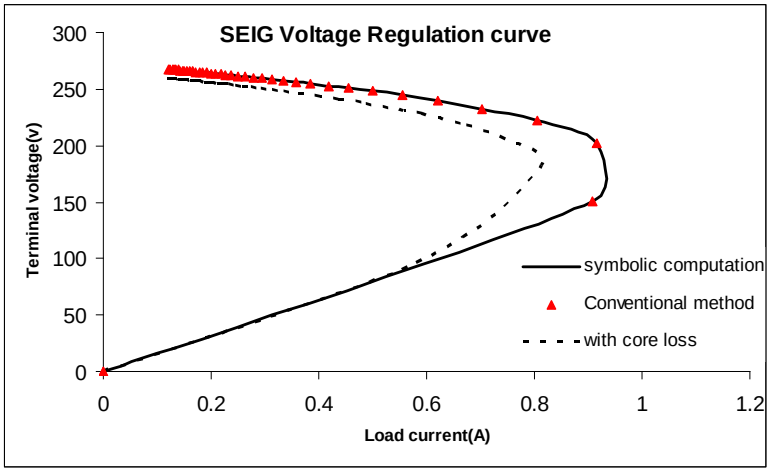

Figure 2. Voltage regulation curve of SEIG

From the figure 2, it can be seen that the results obtained using the symbolic computation technique is in good agreement with that obtained from the conventional method. It also demonstrates the effect of adding a core loss component to the SEIG equivalent circuit. While the effect of core loss can be significant for more accurate analysis, it cannot be included in the conventional method without significantly increasing the complexity in the mathematical manipulation required. On the other hand it can be included in the symbolic computation technique with much ease.

CONCLUSION

From the above results it can be concluded that the Matlab Symbolic computation technique is very effective for the simulation of SEIG. This technique has the advantage that there is no need to manually derive the complex coefficients of the polynomial equations. It also exempts one from using complex numerical methods to solve the polynomial equations. Modeling becomes very simple yet versatile. Core loss and other component can be included easily. The programming and debugging become very easy.

REFERENCES

1. Chan T.F., 1995. Analysis of Self-excited Induction Generators Using an Iterative Method. IEEE transaction on Energy Conversion, 10(3), 502-507.

2. Anagreh Y.N. 2003. Teaching the Self Excited Induction Generator using MATLAB, International Journal of Electrical Engineering Education, 40(1). 55-65.

3. Chan T.F., 1994. Steady State Analysis of Self-excited Induction Generators, IEEE transaction on Energy Conversion, 9(2, 288-296.

4 Sandhu K.S., 2003. Iterative Model for the Analysis of Self-Excited Induction Generators, Electric Power Components and Systems, 31, 925–939.

5. Nigim K.A., Salama M.M.A., Kazerani M. 2003. Solving polynomial algebraic equations of the stand alone induction generator. International Journal of Electrical Engineering Education, 40(1), 45-54.

6 Gurung K., Freere P., 2006. Three phase self excited induction generator with a single phase load. M.S. Research Thesis, Kathmandu University.