Driving Point Impedance Computation Applying Parallel Processing Techniques

A. MEDINA, A. RAMOS-PAZ
Facultad de Ingeniería Eléctrica, División de Estudios de Posgrado
Universidad Michoacana de San Nicolás de Hidalgo
Edificio de la División del Posgrado de Ingeniería Eléctrica, Ciudad Universitaria, 58030
Morelia, Michoacán, MEXICO

Abstract: - This contribution deals with the application of parallel processing techniques based on Parallel Processing Machine (PVM) and Multithreading to the Driving Point Impedances (DPI) of Electric Power Systems. It is demonstrated that the application of parallel processing techniques dramatically reduces the intensive computation effort required for the determination of the power system response represented by driving point impedances.

Key-Words: - Driving Point Impedance, Parallel Processing, Parallel Virtual Machine (PVM), Multithreading.

1 Introduction

In general, an intensive computation effort is required to reproduce the system response for a practical range of frequencies for useful transient and harmonic analysis. The frequency step used is directly proportional to the accuracy obtained. For practical applications, this information is further processed so that the system response obtained with the driving point impedance is adequately reproduced at resonance frequencies by a frequency dependent equivalent, e.g. [1-3].

In this contribution two parallel processing platforms based on Multithreads [4] and PVM [5] are applied to the efficient computation of the system frequency response. It is shown that as the complexity of the system increases, the application of parallel processing significantly reduces the computational effort required by a conventional sequential process to obtain the network frequency response. This is achieved by increasing the process relative efficiency of the parallel processing.

2 Driving Point Impedance

The power network frequency response can be obtained by assembling, at any particular frequency, its respective admittance or impedance matrix from the individual system components. The transfer function between nodal currents and voltages appearing throughout the system busbars is represented, for any frequency \( f \), by the matrix equation,

\[
\bar{I}_f = Y_f \bar{V}_f
\]

The inverse of the frequency admittance matrix \( Z_f \) is the frequency impedance matrix \( Z_f \), where each diagonal element \( Z_{f,j} \) is known as driving point impedance of node \( j \).

The combination of inductive and capacitive elements as seen from a particular bus, can result in either series resonance or a parallel resonance. The result of a series resonance may be the presence of unexpected large amounts of harmonic currents flowing through certain network elements, whereas the result of parallel resonance may be the presence of excessive harmonic voltages across network elements [6][7].

3 Parallel Processing Techniques

Parallel processing can be defined as a form of information processing where two or more processors in combination with some form of inter-processor communication system, cooperate on the concurrent solution of a large problem [8]. The emergence of massive parallel processors and distributed computation have paved the way to the wide acceptance and application of parallel processing for the solution of problems of considerable magnitude in diverse fields.
3.1 Parallel Virtual Machine

PVM (Parallel Virtual Machine) [5] is a platform that allows an heterogeneous network computer set to work as a large computer of multiple processors. Thus, a low cost and powerful virtual computer of multiple supercomputers can be created. PVM has several important advantages, e.g.

- Easy to set up.
- Many virtual machines can co-exist with the same hardware.
- The development of programs is based on message passing libraries.
- Supports C and Fortran.
- The software is very portable.
- The code source is available for free.
- PVM enables users to exploit their existing computer hardware to solve much larger problems with minimal additional cost.

It supports operating platforms such as Windows or LINUX. In PVM the information is transferred by mean a zip-send-reception-unzip protocol. This protocol is based on message passing libraries.

3.2 Multithreading

Multithreading [4] is the application of lightweight subprocesses executed within a process sharing code and data segments, but with their own program counter, machine registers and stack. Global and static variables are common to all threads.

4 Parallel DPI calculation Scheme Proposed

Conventionally, the computation of $Z_f$ is carried-out by means of a sequential process where $Z_{ij}$ is obtained at each frequency. An accurate computation of $Z_{ij}$ requires a small enough frequency step size to be used, so that parallel and series systems resonances are appropriately reproduced. However, the frequency step size is inversely proportional to the computational effort needed to obtain $Z_{ij}$ over the entire frequency range of analysis. However, the computation of $Z_j$, for $f = f_0, f_1, f_2, \ldots, f_n$ can take advantage from the fact that $Z_{jn}$ is independent from $Z_{jn}$, which in turn is independent from $Z_{jn}$, etc. The above process independence makes ideal the application of parallel processing, since concurrent processes are used for the computation of $Z_f$ over the frequency range of interest. Figure 1 illustrates the parallel scheme proposed for the computation of $Z_f$.

Here, a main process element collects all the information related to the electric network such as topology, elements and involved electric variables and parameters. This information is stored in a linked list from a defined class, who contains the relevant information associated with each individual element.

$$\text{No. of frequencies} = \frac{\text{No. of process elements}}{\text{Frequency step}}$$

If the frequency number is larger than the number of process elements in (2), then each process element computes more than one inverse of $Y_j$. However, if the number of threads is equal to the number of frequencies, then each element process calculates one inverse of $Y_j$. On the other hand, if the number of element process is larger than the number frequencies, then the additional threads are not used. Once the electric network information has been collected, the main process element determines the frequency range assigned to each thread for the computation of the system frequency behavior, e.g.

![Figure 1. Proposed parallel scheme](image-url)
processor (in the case of PVM) to be stored in a common variable. This variable is protected against overwriting using the mutual exclusivity mechanism, which prevents simultaneous information access from multiple process elements.

Every single process element builds-up the $Y_f$ matrix using the relevant information of the stored circuit in a linked list of structures, which contains information on the reception and sending nodes, the resistance, inductance and capacitance values and finally a number representing the element arrangement. This number is used to build-up the impedance value on function of the resistor, capacitor and inductance arrangement. Figure 4 illustrates the storage scheme used in this investigation. This scheme consists on an arrangement of pointers to a basic storage structure which contains three elements; an integer variable that represents the receiving node, a complex variable that stores the admittance value and a pointer to this structure. This pointer is used to link the different elements connected to a specific busbar. Based on the storage scheme each thread builds-up the $Y_f$ at the different frequencies assigned.

The process of bifactorization is based on a LU procedure, where, a matrix is converted into the product of two matrices, a lower and an upper matrix respectively, e.g.

$$Y_f = LU$$

where the elements of the matrices $L$ and $U$ are calculated with (4) and (5) respectively.

$$l_{ij} = a_{ij} - \sum_{k<j} l_{ik} u_{kj} \quad i \geq j$$

(4)

$$u_{ij} = \left( a_{ij} - \sum_{k<i} l_{kj} u_{ki} \right) / l_{ii} \quad i < j$$

(5)

Once the matrix is bifactorized, a forward – backward substitution of (6) and (7) is used to obtain, by columns, the elements of the matrix inverse of $Y_f$.

During this process a zero-valued vector is used, except for a 1 in the position of the column to be obtained.

$$Y_f = b_i + \sum_{j=0}^n l_{ij} Y_j \quad \text{forward substitution}$$

(6)

$$X_i = l_{ij} Y_j + \sum_{j=0}^n r_{ij} X_j \quad \text{backward substitution}$$

(7)
Before starting the bifactorization process an ordering scheme is used to generate the minimum number of non-zero elements.

## 5 Test Case

Figure 5 illustrates the test case to be analyzed. It contains five nodes, seven transmission lines, two generation units, represented by voltage injections of 1.0 p.u. The test system data are \( r_1=r_2=r_3=r_4=r_5=r_6=10\,\text{m}\Omega \), \( l_1=l_2=l_3=l_4=l_5=l_6=20\,\text{mH} \) for the transmission lines, \( r_7=20\,\text{m}\Omega \), \( l_7=2\,\text{mH}, \) \( C_1=200\,\mu\text{F} \), \( l_8=l_9=0.69\,\text{mH}, \) \( C_2=C_3=300\,\mu\text{F} \), \( l_{10}=11.5\,\text{mH} \) and \( C_4=100\,\mu\text{F} \).

The programming code was developed in C++ with multithreading [4]. For the case with PVM, the PVM3 library [10] was used. Two processors 794.675 Mhz computer was used to executed the code associated with multithreads. The operative system used was Linux Ubuntu. The developed code reads a data file containing the system configuration and their parameters, the simulation data such as number of nodes, branches, harmonics, frequency step and the number of threads to be used in the simulation. The parallel virtual machine used in this investigation was build using 3 two processors 794.675 Mhz computers.

Appendix I illustrates the models used for representation of the synchronous machine and the transformer [6]. GNUPLOT [9] was used for the graphical representation of the driving point impedance.

For the case with PVM, the PVM3 library [5] was used whereas for the Multithreading case the PTHREAD library was applied [10]. The mutual exclusivity mechanism is applied to appropriately control the access from multiple threads [10][11].

Figure 6 illustrates the impedance system seen from node 2. It can be noticed that there are four points of parallel resonance, e.g. at 120, 180, 230 and 360 hz respectively.

Figure 7 illustrates the system frequency response, represented by the DPI, given as the impedance magnitude versus frequency, as seen from node 4. Two parallel resonances take place at 120 and 350 Hz, respectively. A 0.1 Hz frequency step size was used with a 60 Hz base frequency.
Figure 8 illustrates the effect of the frequency step in the driving point calculation. It can be observed that frequency steps of $1.0\text{fund}$ ($\text{fund} = \text{fundamental}$) and $0.5\text{fund}$ do not identify the parallel resonances associated with the analyzed electric power system, whereas the use of $0.1\text{fund}$ and $0.01\text{fund}$ identify all resonances points associated with the electric power systems analyzed.

![Figure 8. Effect of the frequency step in the DPI evaluation](image)

The parallel processing relative efficiency is computed as [12],

$$E_{relative} = \frac{T_i}{T_p}$$  \hspace{1cm} (8)

where,

$T_i$  \quad execution time with $I$ process element.

$T_p$  \quad execution time with $P$ process elements.

Tables 1-2 illustrate the relative efficiency achieved with the application of parallel processing based on multithreading to the computation of the frequency system response using two different frequency steps.

With $\Delta f = f_{\text{fund}} (=60 \text{ Hz})$ no improvement is obtained in the relative efficiency for the DPI computation, since it remains in 1.0. With $\Delta f = 0.1f_{\text{fund}}$ the relative efficiency increases from 1.0 to 1.2727 with ten harmonics (times the fundamental frequency) and to 1.5238 with forty harmonics, see Table I. For $\Delta f = 0.01f_{\text{fund}}$ the relative efficiency increases from 1.0 to 1.7631 with ten harmonics and to 1.9140 with forty harmonics, see Table II. The application of a third thread does not increase the relative efficiency, since for this investigation a two processors computer was used. The relative efficiency increases in direct proportion to the size of the problem to be solved and the number of threads and processors used, as seen from Tables 1-2.

Table 1. Relative Efficiency with $\Delta f=0.1\text{fund}$

<table>
<thead>
<tr>
<th>Number of threads</th>
<th>Number of harmonics</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10</td>
</tr>
<tr>
<td>1</td>
<td>1.0000</td>
</tr>
<tr>
<td>2</td>
<td>1.2727</td>
</tr>
<tr>
<td>3</td>
<td>1.2727</td>
</tr>
</tbody>
</table>

Table 2. Relative Efficiency with $\Delta f=0.01\text{fund}$

<table>
<thead>
<tr>
<th>Number of threads</th>
<th>Number of harmonics</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10</td>
</tr>
<tr>
<td>1</td>
<td>1.0000</td>
</tr>
<tr>
<td>2</td>
<td>1.7631</td>
</tr>
<tr>
<td>3</td>
<td>1.7631</td>
</tr>
</tbody>
</table>

Table 3 illustrates the relative efficiency obtained with the use of 1-3 slave processors. It can be noted that relative efficiency increases with the increase of slave processors and frequency step used. The maximum relative efficiency obtained is 2.97 for a $\Delta f=0.001$ and with the use of 3 slave processors.

Table 3. Relative Efficiency obtained with PVM and 3 slave processors

<table>
<thead>
<tr>
<th>Number of slave processors</th>
<th>Relative Efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.0000</td>
</tr>
<tr>
<td>2</td>
<td>1.0556</td>
</tr>
<tr>
<td>3</td>
<td>1.0000</td>
</tr>
</tbody>
</table>

6 Conclusions

This contribution has introduced the application of parallel processing based on multithreading, to the fast calculation of driving point impedances in electric networks.

In particular, this investigation has demonstrated that the application of parallel processing techniques significantly increases the relative efficiency for the computation of the frequency dependent system response in the form of driving point impedances, as seen by any system busbar. The efficiency will increase in direct proportion with the system dimension, the size of problem and the number of process elements used.
Acknowledgements

The authors acknowledge the Universidad Michoacana de San Nicolás de Hidalgo (UMSNH) through the División de Estudios de Posgrado of the Facultad de Ingeniería Eléctrica the facilities granted to carry-out this investigation. A. Ramos-Paz acknowledges financial support received by the UMSNH and CONACYT to carry-out his Doctoral studies.

References:


Appendix I

Power System Component Models

Synchronous Machine

\[ Z_{\text{generator}} = r \sqrt{h} + jX^* \]

where

- \( X^* \) generator subtransient reactance.
- \( r \) resistance
- \( h \) harmonic order

Transformer

\[ Z_{\text{generator}} = r \sqrt{h} + jX_{r} \]

where

- \( X_{r} \) transformer short circuit reactance.
- \( r \) resistance
- \( h \) harmonic order