



The Load Flow Calculation in Harmonic Polluted Radial Electric Networks with Distributed Generation

Ciprian Bud, Bogdan Tomoiaga
National Power Grid Company - Transelectrica
Transport Branch Cluj
Cluj-Napoca, Romania
e-mail: ciprian.bud@transelectrica.ro
bogdan.tomoiaga@yahoo.fr

Mircea Chindriş
Technical University of Cluj-Napoca
Cluj-Napoca, Romania
e-mail: mircea.chindris@eps.utcluj.ro

Antoni Sudrià i Anderu
Universitat Politècnica de Catalunya
Barcelona, Spain
e-mail: sudria@citcea.upc.edu

Abstract—Analyse of harmonic distortion in power systems consists in the determination of steady state quantities for each harmonic component at a time. The existing theoretical relationships are difficult to be implemented in complex systems containing many elements, in various combinations. In the case of radial power systems operated in sinusoidal conditions, the load flow calculation can be performed using a particular method known as the *backward/forward sweep*. For harmonic polluted radial electrical networks, we can consider the backward/forward sweep algorithm with some specific adaptations. In this paper, the authors propose to model the electrical harmonic complex quantities through abstract data types with complex parameters. To illustrate the proposed methodology, authors have developed original software that allows performing all necessary operations with electrical harmonic quantities.

Keywords—harmonic distortion, abstract data types, objects oriented programming, power flow, distributed generation.

I. INTRODUCTION

The survey of harmonic distortion in power systems consists in the determination of steady state quantities for each harmonic component at a time. The existing theoretical relationships are difficult to be implemented in complex systems containing many elements, in various combinations.

Besides the data types existing in the numbers theory, the high level programming languages allow the definition of new artificial data types, for instance *abstract data types*. The term *type of data* designates a *set of values* (the domain of type) and a *set of operations* that can be performed with these values. The set of operations can be divided in three subsets: (i) operations among the same type of data; (ii) operations among the specified type of data and another type of data and (iii) operations performed on the data itself. As an example, we can consider data belonging to the real numbers set (the domain of type). In this case, they can perform arithmetical operations with another real number (operations among the same data types), arithmetical operations with an integer number (operations among the specified type of data and another type), and the extraction of integer part (operation applied to the data type itself) [1].

In this paper, the authors propose to model the harmonic complex quantities (a set of complex quantities corresponding to each harmonic component) through *abstract data types* with complex parameters ($\underline{x}_1, \underline{x}_2, \dots, \underline{x}_n$); among these parameters, some operations are defined. By implementing this model, all harmonic quantities will be considered as “harmonic complex” objects. As a result, electrical engineering laws, as Ohm’s or Kirchhoff’s, are reduced to simplified expressions corresponding to fundamental harmonic component case. The paper presents an original solution (and dedicated software) to

perform load flow calculus in harmonic polluted radial electric networks with distributed generation using *abstract data types*.

II. ELECTRIC QUANTITIES AS ABSTRACT DATA TYPE

A. Complex Quantities as Abstract Data Type

Complex numbers are regularly used in electrical engineering. They have the following characteristics:

1) *Domain*: The domain is indicated by C and is specified as:

$$C = \{(a,b) \mid a,b \in R\} \quad (1)$$

2) *Operations Set*: In the complex numbers set, the following operations are defined:

- for any two quantities $v_1, v_2 \in C$, it is possible to determine the quantity $v \in C$, with parameters a_e and b_e , defined through:

Addition: $v = v_1 + v_2$, where:

$$a_e = a_1 + a_2, \quad b_e = b_1 + b_2 \quad (2)$$

In a same manner, it is possible to define the operations: subtraction (-), multiplication (*) and division (/).

- for any quantity $v \in C$, it is possible to determine the subsequent quantities:

Conjugate: $v_1 \in C$ defined through $v_1 = v^*$, with parameters a_e and b_e calculated as follows:

$$a_e = a_1 \quad b_e = -b_1 \quad (3)$$

Module: $r \in R$ defined by $r = |v|$, calculated as:

$$r = |v| = \sqrt{a^2 + b^2} \quad (4)$$

Angle: $\theta \in R$ defined by $\theta = \text{Angle}(v)$, calculated

$$\text{as: } \theta = \text{Angle}(v) = \text{atg} \frac{b}{a} \quad (5)$$

B. Harmonic complex quantities as abstract data types

A “harmonic complex” number may represent many non-sinusoidal physical quantities, namely tensions, currents, impedances etc [1].

1) *Domain*: The domain of the harmonic complex numbers is denoted by HC and specified as:

$$HC = \{(v_1, v_2, \dots, v_i) \mid v_i \in C; \quad i = \overline{1, n}; \quad n \in N^* \quad (6)$$

2) *Operations Set*: The following operations are defined in this set:

- for any two quantities $A, B \in HC$, it is possible to determine the quantity $C \in HC$, with parameters (c_1, c_2, \dots, c_n) , defined through:

Addition: $C = A + B$, where:

$$c_i = a_i + b_i; \quad i = \overline{1, n}; \quad n \in N^* \quad (7)$$

In a same manner, it is possible to define the operations: subtraction (-), multiplication (*) and division (/).

- for any quantity $A \in HC$, it is possible to determine the quantity $r \in R$ defined as:

Module: $r = |A|$, calculated as:

$$r = |A| = \sqrt{\sum_{i=1}^n |a_i|^2}; \quad n \in N^* \quad (8)$$

Dross: $r = \text{Dross}(A)$, calculated as:

$$r = \text{Dross}(A) = \sqrt{\sum_{i=2}^n |a_i|^2}; \quad n \in N^* \quad (9)$$

Total harmonic distortion: $r = \text{THD}(A)$, calculated as:

$$r = \text{THD}(A) = \frac{\text{Dross}(A)}{|a_1|} \cdot 100 \quad (10)$$

- for any quantity $A \in HC$, it is possible to determine the quantity $B \in HC$, with parameters (b_1, b_2, \dots, b_n) , defined through:

Conjugate: $B = A^*$, calculated according to:

$$b_i = a_i^*; \quad i = \overline{1, n}; \quad n \in N^* \quad (11)$$

Weight: $B = \text{Weight}(A)$, calculated according to:

$$|b_i| = \frac{|a_i|}{|a_1|} \quad i = \overline{1, n}; \quad n \in N^* \quad (12)$$

$$\text{Angle}(b_i) = \text{Angle}(a_i)$$

Equivalent vector: $B = \bar{A}$, calculated according to:

$$B = A + A \cdot a^2 + A \cdot a \quad (13)$$

where $a = (a_1, a_1^2, \dots, a_1^i)$; $i = \overline{1, n}$; $n \in N^*$ is defined as

“harmonic rotation factor” and $a_1 \left(-\frac{1}{2}, \frac{\sqrt{3}}{2} \right)$ is the rotation factor for the fundamental harmonic component case;

- for any two quantities $C = A + B$, $|B| \in R$ it is possible to determine the quantity $B \in HC$, with parameters (b_1, b_2, \dots, b_n) , defined through:

RMS (root mean squares): $B = \text{RMS}(A)$, calculated according to:

$$|b_1| = \frac{|B|}{|A|} \quad (14)$$

$$|b_i| = |b_1| \cdot |a_i| \quad i = \overline{1, n}; \quad n \in N^*$$

$$\text{Angle}(b_i) = \text{Angle}(a_i)$$

III. LOAD FLOW CALCULATION FOR RADIAL ELECTRIC NETWORKS

The load flow calculation for electric networks consists in the determination of steady state quantities associated to its elements.

A. Adaptation of backward/forward sweep algorithm for harmonic polluted radial electric networks

In this case, we have two types of nodes:

- one source node (infinite bus), to which the specified quantities are the components of voltage, and the unknown quantities are the components of loads (powers, currents);
- some consumer nodes, to which the specified quantities are loads (constant powers, constant currents, constant impedances/admittances or a combination of the above), and the unknown quantities are the components of voltage. The capacitor banks can be represented in a same manner: as constant harmonic impedances in a combination with the specified characteristics of loads.

The electric lines are considered as harmonic impedances.

The load flow calculation can be performed using a specific method, known as the *backward/forward sweep*. This method consists of two steps [8]:

- *Backward sweep*, where, starting from the end nodes and going towards the source node, and using the Kirchhoff's current law, the current at each load node, as well as the current flowing through its ingoing branch, are calculated;
- *Forward sweep*, where, starting in the opposite direction, from the source node S (whose constant voltage is taken as reference) and going towards the end nodes, using the Ohm's law, the voltage drop on each branch, as well as the voltage at each load node, are calculated.

In harmonic polluted power systems, the above-proposed models can be introduced. Accordingly, the load flow calculation algorithm using the backward/forward sweep consists in the following steps:

- Ordering the network (indexing the ingoing node and branch for each load node) and setting the voltages at the load nodes to the value of the source node (S) phase voltage – in this first step, the voltage system is considered sinusoidal:

$$U_k^{(0)} = U_S, k=1,2,\dots,n, k \neq S \quad (15)$$

where: $U_S, U_k \in HC$ with complex parameters (u_1, u_2, \dots, u_n) calculated according to:

$$u_1 = u_{Source}; \quad u_i = 0; \quad i = \overline{2, n}; \quad n \in N^* \quad (16)$$

- Setting the initial iteration index: $p = 1$;
- Backward sweep*: crossing the network from the end nodes towards the source node and performing the following operations:

- Calculation of the current at the node k by using the expression of the load power given by (concordant with relationships 14):

$$I_k^{(p)} = RMS(IP_k) \quad (17)$$

where $I_k^{(p)}, IP_k \in HC$;

$$|I_k^{(p)}| = \frac{|S_k^*|}{\sqrt{3} \cdot |U_k^{(p)*}|} - \text{r.m.s. value of the current (in}$$

this case loads are represented only through constant powers);

IP_k - represent weights of harmonic components;

- Calculation of the current flowing through the branch ingoing to node k :

$$I_{ik}^{(p)} = I_k^{(p)} + \sum_{j \in Next(k)} I_{kj}^{(p)} \quad (18)$$

$$\text{if } connection = Y \rightarrow I_{ikN}^{(p)} = I_{ik}^{(p)} \quad (19)$$

$$\text{if } connection = \Delta \rightarrow I_{ikN}^{(p)} = \sum_{j \in Next(k)} I_{kjN}^{(p)} \quad (20)$$

where: i is the index of the node up stream to the node k ;

$Next(k)$ - the set of nodes next to the node k ;

- Forward sweep*: the calculation of voltages at the nodes, crossing the network from the source node towards the end nodes. For the in progress iteration p , considering the crossing direction of a branch from the node i towards node k , the calculation is performed in the following manner:

- Calculation of the voltage drop on the $i - k$ branch:

$$\Delta U_{ik}^{(p)} = Z_{ik} \cdot I_{ik}^{(p)} + Z_{ikN} \cdot I_{ikN}^{(p)} \quad (21)$$

where: $Z_{ik} \in HC$;

$Z_{ikN} \in HC$ - harmonic impedance of neutral conductor;

- Calculation of the voltage at the node k :

$$U_k^{(p)} = U_i^{(p)} - \Delta U_{ik}^{(p)} \quad (22)$$

- Calculation of the power injected into the network by the source node:

$$S_S^{(p)} = U_S \cdot \sum_{i \in Next(S)} I_{Si}^{*(p)} \quad (23)$$

- If $p > 1$ and $|\overline{S_S^{(p)}} - \overline{S_S^{(p-1)}}| \leq \varepsilon_s$ then go to the next step, else update $p = p + 1$ and go to step c.
- Calculation of power losses through the network branches.

B. Adaptation of backward/forward sweep algorithm for harmonic polluted radial electric networks with distributed generators

Distributed generators can be classified:

- generators which provide *constant power* ($P_i Q_i$ nodes), to which the specified quantities are the constant powers and the unknown quantities are the components of voltage;
- generators which provide *constant active power and constant module of voltage* ($P_i U_i$ nodes), to which the specified quantities are the active power and module of voltage, and the unknown quantities are the reactive power and the angle (phase) of voltage. These sources can generate active power and sometimes can generate or consume reactive power, having the possibility to maintain the nodal voltage at a set value by means of an automatic voltage regulator.

For harmonic polluted radial electric networks that include distributed generators, we can consider the

backward/forward sweep with some specifications and adaptations. The algorithm is the following [2]:

1. Initialise the iterative step $p = 0$ and establish the initial value of the (generated) reactive power for every $P_I U_I$ nodes: $Q_{1g,k}^{(0)} = 0$;
2. Update the iterative step $p = p + 1$;
3. Perform load flow calculation by *backward/forward sweep*;
4. For each node k ($P_I U_I$ type), if $\|U_{1k}^{(p)} - |U_{1k}^{specified}\| \leq \epsilon_U$ or $Q_{1g,k}^{calc} = Q_{1g,k}^{min}$ or

$Q_{1g,k}^{calc} = Q_{1g,k}^{max}$, the iterative process stops;

5. Calculate the generated reactive power $Q_{1g,k}^{calc}$ necessary to achieve the specified voltage $U_{1k}^{specified}$ at a node k (the $P_I U_I$ nodes), e.g. using the secant method:

$$Q_{1g,k}^{calc} = Q_{1g,k}^{(p-1)} + \frac{Q_{1g,k}^{(p-1)} - Q_{1g,k}^{(p-2)}}{|U_{1k}^{(p-1)}| - |U_{1k}^{(p-2)}|} (|U_{1k}^{(p)}| - |U_{1k}^{specified}|) \quad (24)$$

$$\begin{cases} \text{if } (Q_{1g,k}^{min} \leq Q_{1g,k}^{calc} \leq Q_{1g,k}^{max}) & \text{then } Q_{1g,k}^{(p)} = Q_{1g,k}^{calc} \\ \text{if } (Q_{1g,k}^{calc} < Q_{1g,k}^{min}) & \text{then } Q_{1g,k}^{(p)} = Q_{1g,k}^{min} \\ \text{if } (Q_{1g,k}^{calc} > Q_{1g,k}^{max}) & \text{then } Q_{1g,k}^{(p)} = Q_{1g,k}^{max} \end{cases} \quad (25)$$

6. Go to step 2.

By introducing the proposed objects in the existing algorithm for networks operated in sinusoidal regime (with demonstrated correctness and convergence) [2], we obtained a novel algorithm for load-flow calculation in harmonic polluted states. It is important to specify that the number of iterations is a function of two constants, which represent the precision of calculus: ϵ_S (the deviation of injected power from the source node – infinite bus) and ϵ_U (the deviation of voltage in PU node types).

IV. APPLICATION

The high level programming languages (*Object Pascal*, *C++*, *Java* etc.) allow the definition of *abstract data types* as objects and the implementation of the above defined models. For carrying out, the authors have selected the *C++* language program because it admits the overcharge of existing arithmetical operators (they are redefined in the program).

Figure 1 presents the object (*class*) C (complex). The member functions implement the relationship presented in chapter II.A, and the unit can be used in any program written in *C++* language for all calculus with complex numbers [3, 5]. Figure 2 presents the object HC (harmonic complex), the member functions implementing the relationship presented in chapter II.B. This unit can also be used in any program written in *C++* language. Based on these two units (C and HC), authors have developed an original program, named *OBmac*, dedicated to mathematical operations with three-phase quantities. Realized with *GUI* (*Graphic User Interface*) technology, the program ensures an extreme easiness in the use.

In the program are introduced eleven harmonic complex objects, namely A, B, C, \dots, K . For each object, the following are defined:

- one location for writing relationships among the harmonic complex quantities (objects);
- a table with modules of 50 harmonic components;
- a table with angles of 50 harmonic components.

```
typedef double REAL;           //define a real number
class C                         { //class complex
protected:                     //the protected part
    REAL a, b;                  //real and imaginary data
public:                          //the public part
    C();                        //constructor
    ~C();                       //destructor
    virtual C operator + (C& c); //operation of addition
    virtual C operator - (C& c); //operation of subtraction
    virtual C operator * (C& c); //operation of multiplication
    virtual C operator / (C& c); //operation of division
    virtual C operator * (REAL r); //operation of multiplication
                                   //with a real number
    virtual C conjugate();       //operation by conjugate
    virtual REAL module();      //module operation
};
```

Figure 1. Complex object implementation

Any harmonic complex quantity is introduced (as input data) through *Input/Output* panel and is attached to an object making a “click” on „<<” button concordant to symbol of object. – figure 3. It is possible to rotate (electrical rotation) the object with 120° through multiplication with *harmonic rotation object* (a).

```
#include "TAD_C.cpp"           //include C object
const n = 51;
class HC                         { //class three complex
private:                         //the private part
    C ARM[n];                    //harmonic complex data
public:                           //the public part
    HC();                        //constructor
    ~HC();                       //destructor
    HC operator + (HC& hc);       //operation of addition
    HC operator - (HC& hc);       //operation of subtraction
    HC operator * (HC& hc);       //operation of multiplication
    HC operator / (HC& hc);       //operation of division
    HC operator * (REAL r);       //operation of multiplication
                                   //with a real number
    HC conjugate();              //operation by conjugate
    REAL module();               //module operation
    REAL dross();                //dross operation
    HC THD();                    //THD operation
    HC weight();                 //weight operation
    HC RMS(REAL r);              //r.m.s. operation
    HC equivalent_v();           //equivalent vector operation
};
```

Figure 2. Harmonic complex object implementation

The results are obtained making a “click” on “*RUN*” button and appear in the two tables with module and angles of harmonic components. Any object can be also graphic viewed in *Input/Output* panel making a “click” on “>>” button concordant to symbol of object.

A numerical example (figure 4) is given for an electric circuit connected to the point of common coupling (PCC) of a non-linear consumer, where [4]:

- $L_{sc} = 3.39 \cdot 10^{-5}$ [H] is the equivalent inductance of power system;
- $R = 444$ [mΩ] – the equivalent resistance of the linear consumer;

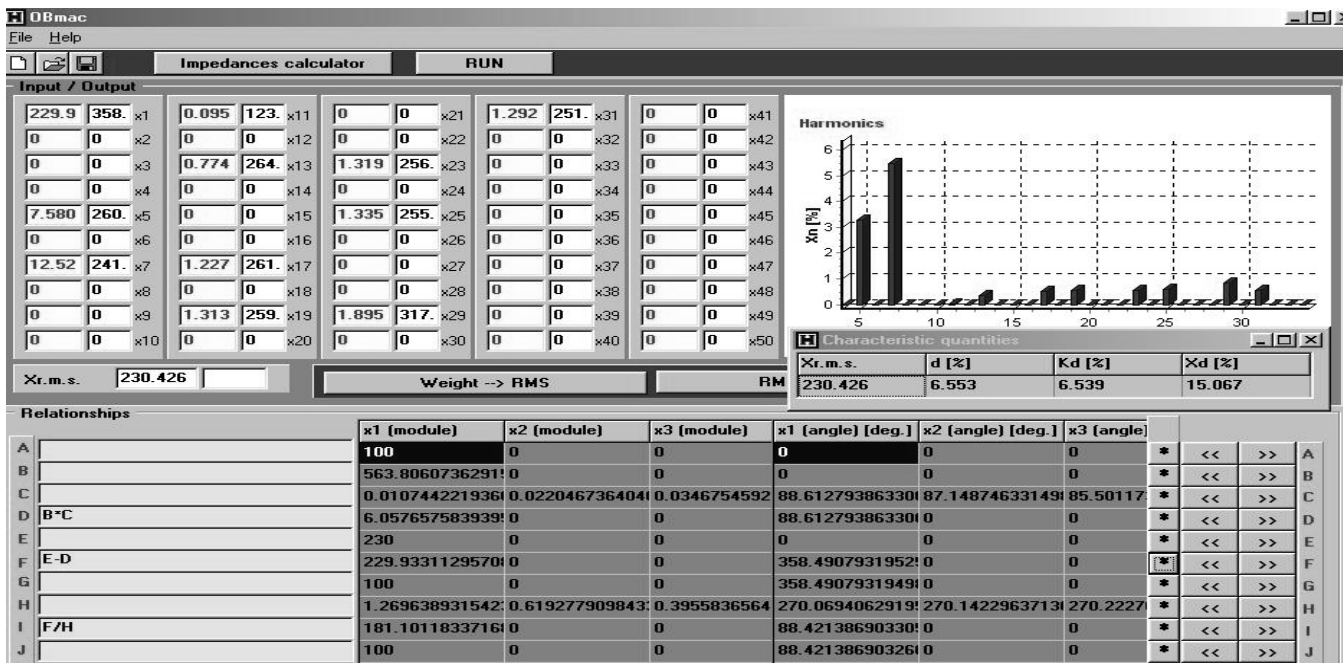


Figure 3. Obmac program

- $r = 1.538 [m\Omega]$ – the resistance of the parallel filter;
- $L = 0.033 [mH]$ – the inductance of the parallel filter;
- $C = 2.49[mF]$ – the capacitance of the capacitor bank.

To solve the problem, the following objects are defined in the program:

- **A** – spectrum with weights of harmonic currents of non-linear consumer;
- **B** – spectrum with harmonic currents (of non-linear consumer) obtained through “RMS” operation (making a “click” on “Weight --> RMS” button), considering the r.m.s. value of harmonic current ($579.710 [A]$);
- **C** – harmonic equivalent impedance at PCC;
- **D** – harmonic voltage drops at PCC;
- **E** – source voltage which contains only the fundamental component ($230 [V]$);
- **F** – harmonic voltage at PCC (making a “click” on “*” button concordant to symbol of object, we view in *characteristic quantities* panel: r.m.s. value, THD and the dross - X_d);
- **G** – spectrum with weights of harmonic voltages at PCC;
- **H** – harmonic equivalent impedance of parallel filter;
- **I** – harmonic current through capacitors bank.

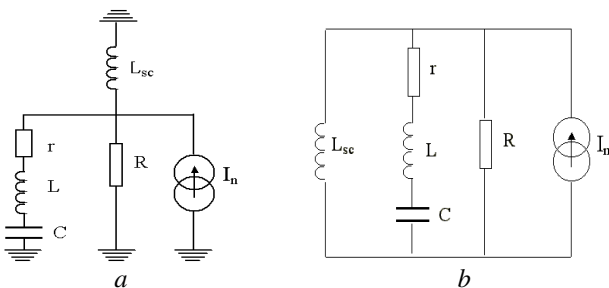


Figure 4. Electric scheme viewed in common coupled point
a – simplified electric scheme
b – equivalent electric scheme

V. CONCLUSIONS

The paper presents an original solution to perform calculus with complex and harmonic complex quantities by using *abstract data types*. Related mathematical models for *complex abstract data type* and *harmonic complex abstract data type* are developed. These structures were implemented as objects in an original software which allows to manipulate complex and harmonic complex quantities by simply using common operators like “+”, “-”, “*” and “/”.

By implementing these objects in a backward/forward sweep algorithm for networks with distributed generation, a new algorithm for harmonic polluted regime was proposed in the paper. A numerical example is given to confirm the computer implementation of proposed objects; the complete algorithm for the power-flow computation is under a test process for validation and will be presented in future works. The authors intend to develop an algorithm and an associated program for load-flow analysis in harmonic polluted and unbalanced electric networks with distributed generation.

REFERENCES

- [1] M. Chindriș, B. Tomoiagă and C. Bud, “A Novel Model and Object Oriented Program to Analyse the Harmonic Distortion in Power Systems”, in *1st International Conference on Modern Power Systems*, Cluj-Napoca, November 2006, pp. 141-146.
- [2] M. Eremia et alii, *Electric Power Systems – Electric Networks*, vol. I. Bucuresti: Academiei, 2006.
- [3] V. Cioban, Z. Darvai, *The programming evolutes methods – C and C++ languages (in romanian)*. Cluj-Napoca: Litho. Univ. “Babes-Bolyai”, 2001.
- [4] M. Chindriș, et alii, *Reducerea poluării armonice a rețelelor electrice industriale*. Cluj-Napoca: Mediamira, 2003.
- [5] B. Stroustrup, *C++ Programming Language – Special Edition (in romanian)*. Bucuresti: Teora, 2000



OBphasor - Calculator for electrical engineering

- Useful tool for each electrical engineer
- It doesn't require installation on computer
- It works on all versions of Windows operating system

[Click here for more details >>](#)

Download
for free