MEĐUNARODNO REGIONALNO SAVETOVANJE O ELEKTRODISTRIBUTIVNIM MREŽAMA

REGIONAL CONFERENCE AND EXHIBITION ON ELECTRICITY DISTRIBUTION

Crna Gora, Herceg Novi, 5 - 8. oktobar 2004. / ***Montenegro, Herceg Novi, October 5 - 8, 2004***

MICROPROCESSOR PROTECTION ALGORITHMS OF DISTRIBUTIVE LEADS MPZ-ZIM

Milenko Đurić, ***Faculty of Electrotechnics, Belgrade, Serbia***
Goran Đukić, ***Faculty of Electrotechnics, Belgrade, Serbia***
Željko Đurišić, ***Faculty of Electrotechnics, Belgrade, Serbia***

INTRODUCTION

Basic protection functions of distributive leads have been present and used for a long time, both with electromechanical and statical protections; they are also used in microprocessor protections for this purpose. They are, simply put, based on various types of over current protection functions, directional protection functions, automatic switching functions. These protections can also contain thermal protection functions, as well as frequency protection. All of the protection functions are based on phase currents, and phase and line voltages; they also depend on direct, inverse and zero component values of currents and voltages. An adequate digital signal processing (of currents, voltages and their symmetrical components) is needed for the practical realization of all the protection functions mentioned, as well as for measurements, which make an integral part of multifunctional digital terminals (installed in same devices with protection functions). This means that an adequate signal processing of currents and voltages (sometimes of directly measured zero components) is essential for determination of symmetrical components (of current and voltage), power (active and/or reactive), frequency, temperature. Algorithms for determination of effective (current and voltage) values, symmetrical components (current and voltage), frequency, power (active and reactive) and thermal protection, have been presented in this paper.

1.CURRENT AND VOLTAGE MEASUREMENT ALGORITHM

This algorithm is based on recursive Fourier's method, which is well known and widely used in signal processing. This is a very robust method which has some good filtering capabilities, but also, has some drawbacks. One of the main drawbacks is necessity of knowing the signal frequency, before processing it. If the frequency of the real signal differs from the frequency assumed in Fourier's method, algorithm makes an error when calculating the harmonic component's amplitude of the signal. In this work, recursive Fourier's algorithm has been modified, so changing of signal frequency in terms of few *Hz* (which is enough for measurements of industrial frequency signals), has a very small influence on determination of basic harmonic's amplitude. Let's observe the signal (of voltage or current) in the following form:

$$x(t) = C\cos(\omega t + \phi) \qquad (1).$$

Parameters of the signal are: C -maximum value, $\omega = 2\pi f$ -angle frequency, ϕ -starting phase angle.

If the exact value of angle frequency (ω) is unknown and approximated with assumed value ω_f , parameters can be easily calculated with the use of discrete Fourier series :

$$\underline{X} \cong \frac{2}{m}\left[\sum_{n=1}^{m} x_n \cos\left(\frac{\omega_f T_f}{m} n\right) - j\sum_{n=1}^{m} x_n \sin\left(\frac{\omega_f T_f}{m} n\right)\right] = A_f + jB_f \ , \tag{2}$$

given symbols represent: $C_f^2 = A_f^2 + B_f^2$, $tg\phi_f = \frac{B_f}{A_f}$, m -number of samples in the period T_f (period which corresponds with frequency ω_f), ω_f -assumed frequency in Fourier's series (if we want (2) to give a good result ω_f must equal ω , $\omega_f = \omega$), x_n - n-th signal sample. Lets observe two sequences of samples. First one indexed with $n = 1,2,3,...,m$, and second one with $n = 2,3,4,...(m+1)$. Both of these correspond with the period of signal being processed. Second sequence is "younger" cause it holds a sample indexed with $(m+1)$, while the oldest sample indexed with (1) is rejected. Each of the sequences defines phasors of harmonics (of the processed signal):

$$\underline{X}_1 \cong \frac{2}{m}\left[\sum_{n=1}^{m} x_n \cos\left(\frac{\omega_f T_f}{m} n\right) - j\sum_{n=1}^{m} x_n \sin\left(\frac{\omega_f T_f}{m} n\right)\right] = A_{1f} + jB_{1f} \ , \tag{3}$$

and:

$$\underline{X}_2 \cong \frac{2}{m}\left[\sum_{n=2}^{m+1} x_n \cos\left(\frac{\omega_f T_f}{m} n\right) - j\sum_{n=2}^{m+1} x_n \sin\left(\frac{\omega_f T_f}{m} n\right)\right] = A_{2f} + jB_{2f} \ , \quad \omega_f T_f = 2\pi \ . \tag{4}$$

Lets subtract them:

$$\left(A_{2f} - A_{1f}\right)\frac{m}{2} = \sum_{n=2}^{m+1} x_n \cos\left(\frac{2\pi}{m} n\right) - \sum_{n=1}^{m} x_n \cos\left(\frac{2\pi}{m} n\right) = x_2 \cos\left(\frac{2\pi}{m} 2\right) + x_3 \cos\left(\frac{2\pi}{m} 3\right) + ... +$$

$$+ x_{(m+1)} \cos\left[\frac{2\pi}{m}(m+1)\right] - x_1 \cos\left(\frac{2\pi}{m}\right) - x_2 \cos\left(\frac{2\pi}{m} 2\right) - ..... - x_m \cos\left(\frac{2\pi}{m} m\right) =$$

$$= x_{(m+1)} \cos\left[\frac{2\pi}{m}(m+1)\right] - x_1 \cos\left(\frac{2\pi}{m}\right) = \left(x_{(m+1)} - x_1\right)\cos\left(\frac{2\pi}{m}\right)$$

which gives:

$$A_{2f} = A_{1f} + \frac{2}{m}\left(x_{(m+1)} - x_1\right)\cos\left(\frac{2\pi}{m}\right) ,$$

and:

$$B_{2f} = B_{1f} + \frac{2}{m}\left(x_{(m+1)} - x_1\right)\sin\left(\frac{2\pi}{m}\right) .$$

Based on this analysis, we can write relations for recursive calculation of phasor components, in $(n+1)$-th step of iteration cyclus:

$$A_{(n+1)f} = A_{nf} + \frac{2}{m}\left(x_{(m+n)} - x_n\right)\cos\left(\frac{2\pi}{m} n\right) \quad \text{and} \quad B_{(n+1)f} = B_{nf} + \frac{2}{m}\left(x_{(m+n)} - x_n\right)\sin\left(\frac{2\pi}{m} n\right) , \tag{5}$$

given symbols represent: n-sample number ($n = 1,2,3,4,.....$, in continual signal analysis, n can become a huge number), m-number of samples in a period $T_f = \frac{1}{f_f} = \frac{2\pi}{\omega_f}$.

In practical applications of discrete Fourier's series, with continual processing of long signals, n would become huge and it would lead to cpu owerflow error. We can solve this, knowing that sinus and cosines functions are periodical. That means that (n) doesn't have to be larger then (m), it can take values $n = 1,2,3,...,m,1,2,3,...,m,1,2,3,...,m,1,2,3,...,m,1,2$ etc.. This implies that we have to have one buffer (cpu register) of (m) in length; members of this register have to be updated in each iteration step, after calculation of signal phasor's components. Lets mark signal samples with (x) and buffer members with $X(i)$, where i takes values $i = 1,2,3,\ldots,m$. Phasor components of the first harmonic can be calculated as:

$$A_f = A_f + \frac{2}{m}\left(x - X(1)\right)\cos\left(\frac{2\pi}{m}n\right) \quad \text{and} \quad B_f = B_f + \frac{2}{m}\left(x - X(1)\right)\sin\left(\frac{2\pi}{m}n\right), \tag{6}$$

where $n = 1,2,3,\ldots,m,1,2,3,\ldots,m,1,2,3,\ldots,m,1,2,3,\ldots,m,1,2$ etc..

Equation (6) is given in a form of computer instruction, which means that we get a new value of A_f by adding $\frac{2}{m}\left(x - X(1)\right)\cos\left(\frac{2\pi}{m}n\right)$ to an old one. After calculation of A_f and B_f, register member indexed with (*1*) is rejected, and other sample indexes are decreased by (*1*). New sample (taken from A/D converter) becomes a sample indexed with (*m*), $X(m) = x$. Calculation is repeated in the same way. For extensive calculations, vector X needs to be *(m)* in length. All other units are scalars. There is no need to calculate sinus and cosinus functions (which we have in expressions for A_f and B_f) in each iteration step. It's better to calculate them in advance, for one complete period, and then store in two vectors, each of *(m)* in length. These vectors are:

$$CO = \left[\cos\left(\frac{2\pi}{m}\right), \cos\left(\frac{2\pi}{m}2\right), \cos\left(\frac{2\pi}{m}3\right), \ldots\ldots, 1\right]^T \quad \text{and} \tag{7}$$

$$SI = \left[\sin\left(\frac{2\pi}{m}\right), \sin\left(\frac{2\pi}{m}2\right), \sin\left(\frac{2\pi}{m}3\right), \ldots\ldots, 0\right]^T . \tag{8}$$

Members of these vectors can be multiplied with $\left(\frac{2}{m}\right)$ (for phasor amplitudes) or with $\left(\frac{\sqrt{2}}{m}\right)$ (for effective values). Using this procedure, only two additions and one multiplication is needed in each iteration step, which is minimal cpu time consuming. This is main advantage of recursive algorithm, when compared to non-recursive. Signal which is being processed with (6), should be sampled with sampling period $T_{odab} = \frac{T_f}{m}$, and frequency $f_{odab} = \frac{m}{T_f}$. Knowing that, $\omega_f T_f = 2\pi$, means that $\frac{\omega_f T_f}{m} = \alpha_f$ is *m*-th part of a full circle. That means that assumed frequency in Fourier's method is defined with period of sampling T_{odab} and with number of samples (m) in one period. When a sampling period is fixed, assumed signal frequency ω_f can be changed only by changing the number of samples in a period. However, this makes a very rough difference for a relatively small number of samples in a period. That is the reason why fixed assumed signal frequency is being used in most cases. There are cpu's where one can change sampling period, but these changes are not continual, so it is not possible to observe continual changes of the real signal frequency, while processing it with Fourier transform.

When a long signal is processed, real (A_f) and imaginary (B_f) components of phasor $\underline{X}$ are calculated after each sample. If $\omega = \omega_f$, components A_f and B_f are constant, because x and $X(1)$ are shifted for one period exactly, which means they have the same values, so their difference is $x - X(1) = 0$, and there's no correction of calculated component. Phasor argument depends on starting sample's position (in a signal's wave), or, in another words- it depends on starting phase (phase angle) of the first sample. Amplitude of the signal's main harmonic is constant, because its components are constant also.

If $\omega < \omega_f$ or $\omega > \omega_f$, components A_f and B_f are not constant, and they change according to deformed sinus and cosines functions, with frequency $\Delta\omega = \omega - \omega_f$. Deformations of these curves have wave-like characteristics, and they increase with larger $\Delta\omega$. Frequency of wave-like deformation is (2ω). Because of wave deformation of components A_f and B_f, the main harmonic's amplitude also has frequency (2ω). Components A_f and B_f are shifted for $\frac{\pi}{2}$ or $\left(-\frac{\pi}{2}\right)$, depending on the sign of $\Delta\omega = \omega - \omega_f$.

Based on analysis given here previously, it can be concluded that recursive Fourier's algorithm, expressions (6), gives incorrect values of main harmonic's amplitude, when signal frequency (ω) differs from assumed frequency (ω_f). When the signal is continually processed, main harmonic's amplitude oscillates with frequency 2ω. This fact is used for modification of Fourier's algorithm. Knowing that main harmonic's amplitude oscillates with frequency 2ω, better amplitude values can be accomplished by averaging results, after each half period of main harmonic's signal. In other words, if a signal is processed with (m) samples per period, it would be sufficient to calculate average of $\left(\frac{m}{2}\right)$ results. If averaging is applied, final processing results will have a time delay of half a period (of the main harmonic). However, any signal filtration always includes time delay, so one has to compromise between degree of filtration and amount of time delay. When signal frequency differs from assumed frequency in discrete Fourier series, $\left(\frac{m}{2}\right)$ samples does not define half period of the main harmonic exactly. That is the reason why averaging result is not constant, it also oscillates with 2ω, but with far smaller amplitude then algorithm result that doesn't use averaging. Oscillating of average value can be decreased if we apply averaging once more. It means that after this, algorithm has a full period delay of a basic harmonic. After applying averaging twice, we get a very "steady" value of the main harmonic's amplitude with a very small deviation (error), if the frequency is within *(47-53)Hz* range; that is acceptable when having in mind that frequency rarely deviates for more then *3Hz*, in power systems with nominal frequency of *50Hz*. Delay is acceptable for all types of current and voltage relays, so this algorithm is acceptable as well.

For practical usage of expressions (6), additional vectors of sinus (7), cosines (8) and samples ((m) in length), results and average values ($\left(\frac{m}{2}\right)$ in length), need to be formed. Vector of samples is:

$$X = [x_1, x_2, x_3, \ldots, x_m]^T \ . \qquad (9)$$

Using additional vectors (7) and (8), calculations in (6) include only two multiplications and four additions, without having to calculate trigonometrical functions and divisions, which enormously_speeds up the whole process. After fetching of each new sample x_{novo}, vector (9) should be updated by changing its indexes, according to the following:

$$x_1 = x_2, \quad x_2 = x_3, \quad \ldots, \quad x_m = x_{novo} \, .$$

In this way, we get a "sliding" window and each sample is treated as a scalar value. For each new sample, expression (6) gives A_f and B_f. We get:

$$C_f = \sqrt{A_f^2 + B_f^2} \ . \qquad (10)$$

In order to calculate amplitude's average or first harmonic's effective value average, and additional vector (which is $\left(\frac{m}{2}\right)$ in length) needs to be formed:

$$CAB = \left[C_{f1}, C_{f2}, C_{f3}, \ldots, C_{f\left(\frac{m}{2}\right)} \right]^T \ . \qquad (11)$$

After calculating new value of the amplitude C_{fnovo} using (6) and (10), vector (11) should be updated, by rejecting its first member and adding new one:

$$C_{f1} = C_{f2},\quad C_{f2} = C_{f3},\quad \ldots,\quad C_{f\left(\frac{m}{2}\right)} = C_{fnovo}.$$

After this procedure, we get:

$$CS = \frac{2}{m}\left(C_{f1} + C_{f2} + \ldots + C_{f\left(\frac{m}{2}\right)}\right). \tag{12}$$

Procedure goes on in the same way for each new amplitude value. If we want to find an average of the main harmonic's amplitude averages, we need to form additional vector of average values (which is $\left(\frac{m}{2}\right)$ in length):

$$CSS = \left[CS_1, CS_2, CS_3, \ldots, CS_{\left(\frac{m}{2}\right)}\right]^T. \tag{13}$$

After calculating new amplitude's average value CSS_{novo} using (13), vector (9) should be updated, by rejecting its first member and adding new one:

$$CS_1 = CS_2,\quad CS_2 = CS_3,\quad \ldots,\quad CS_{\left(\frac{m}{2}\right)} = CS_{novo}.$$

After this procedure, we get:

$$CSS = \frac{2}{m}\left(CS_1 + CS_2 + \ldots + CS_{\left(\frac{m}{2}\right)}\right).$$

Procedure goes on in the same way for each new amplitude's average value. For practical usage of modified recursive Fourier's algorithm, we need three vectors (of (m) in length), and two vectors (of $\left(\frac{m}{2}\right)$ in length), which means that very few memory resources are needed. Algorithm is simple, very fast and excellent for applying in relay protection.

2.ALGORITHM FOR MEASUREMENT OF SYMMETRICAL COMPONENTS

Method of symmetrical components is based on the fact that non symmetrical three phased phasor system can be presented using two three phased phasor symmetrical systems with opposite phase angles and one single phased phasor system. Having in mind that this method is based on decomposition of electrical units, it also includes the superposition method, which can only be used in linear circuits. This method is also known as Fortesque's transformation, named after its author. Non symmetrical three phased system of voltages in phases *A*, *B* and *C* (same goes for the currents), can be expressed using direct (*d*), inverse (*i*) and zero (*0*) component system, shown in the following relation:

$$[\underline{U}] = [\underline{F}][\underline{U}_K] = \begin{bmatrix} \underline{U}_A \\ \underline{U}_B \\ \underline{U}_C \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & \underline{a}^2 & \underline{a} \\ 1 & \underline{a} & \underline{a}^2 \end{bmatrix} \begin{bmatrix} \underline{U}_0 \\ \underline{U}_d \\ \underline{U}_i \end{bmatrix}, \tag{14}$$

given symbols represent: $\underline{a} = e^{j2\pi/3}$; $[\underline{U}] = [\underline{U}_A \quad \underline{U}_B \quad \underline{U}_C]^T$ -vector of phase voltages, phases *A*, *B* and *C*; $[\underline{U}_K] = [\underline{U}_0 \quad \underline{U}_d \quad \underline{U}_i]^T$ -vector of component voltages (zero, direct and inverse voltage component).

Three phased asymmetrical system of voltages, can be decomposed by symmetrical components using the inverse relation of (14):

$$\left[\underline{U}_K\right]=\left[\underline{F}\right]^{-1}\left[\underline{U}\right]=\begin{bmatrix}\underline{U}_0\\ \underline{U}_d\\ \underline{U}_i\end{bmatrix}=\frac{1}{3}\begin{bmatrix}1 & 1 & 1\\ 1 & \underline{a} & \underline{a}^2\\ 1 & \underline{a}^2 & \underline{a}\end{bmatrix}\begin{bmatrix}\underline{U}_A\\ \underline{U}_B\\ \underline{U}_C\end{bmatrix}. \qquad (15)$$

Using (15), we get vectors of symmetrical components:

$$\underline{U}_0=\frac{\underline{U}_A+\underline{U}_B+\underline{U}_C}{3}=\frac{AC+BC+CC}{3}+j\frac{AS+BS+CS}{3}=C_0+jS_0, \quad U_0=\sqrt{C_0^2+S_0^2}\,, \qquad (16)$$

$$\underline{U}_d=\frac{\underline{U}_A+\underline{a}\underline{U}_B+\underline{a}^2\underline{U}_C}{3}=\frac{2AC-BC-CC-\sqrt{3}(BS-CS)}{6}+j\frac{2AS-BS-CS+\sqrt{3}(BC-CC)}{6}=$$
$$=C_d+jS_d, \quad U_d=\sqrt{C_d^2+S_d^2} \qquad ,(17)$$

$$\underline{U}_i=\frac{\underline{U}_A+\underline{a}^2\underline{U}_B+\underline{a}\underline{U}_C}{3}=\frac{2AC-BC-CC+\sqrt{3}(BS-CS)}{6}+j\frac{2AS-BS-CS-\sqrt{3}(BC-CC)}{6}=$$
$$=C_i+jS_i, \quad U_i=\sqrt{C_i^2+S_i^2} \qquad ,(18)$$

where vectors of phase values are defined as in (2):

$$\underline{U}_A=AC+jAS, \quad \underline{U}_B=BC+jBS, \quad \underline{U}_C=CC+jCS\,,$$

and vectors of symmetrical components as:

$$\underline{U}_0=C_0+jS_0, \quad \underline{U}_d=C_d+jS_d, \quad \underline{U}_i=C_i+jS_i\,.$$

Second letter (*C*) in components of phase vectors, marks the cosines component (real) while first letter (*S*) in components of symmetrical components' vectors, marks the sinus component (imaginary). Working with cosinus and sinus components is better, because real component can be either cosines or sinus, depending on the coordinate system chosen for vector representation.

It has been emphasized that Fourier's algorithm gives incorrect values of cosines component's amplitude and sinus component's amplitude, of processed signal's main harmonic, when signal's frequency differs from assumed frequency. Having in mind that symmetrical components are sums of phase vector components, they will be calculated incorrectly as well. When three phased signals are being continually processed, symmetrical components' amplitudes (of the main harmonic), oscillate between maximal and minimal values, with frequency 2ω. This fact was used for modification of Fourier's algorithm, and for measurement of symmetrical components. Knowing that symmetrical components' amplitudes (of the main harmonic) oscillate with 2ω, better values can be accomplished by averaging results, after each half period of main harmonic's signal. In other words, if a signal is processed with (m) samples per period, it would be sufficient to calculate average of $\left(\frac{m}{2}\right)$ results. If averaging is applied, final processing results will have a time delay of half a period (of the main harmonic). However, any signal filtration always includes time delay, so one has to compromise between degree of filtration and amount of time delay. When signal frequency differs from assumed frequency in discrete Fourier series, $\left(\frac{m}{2}\right)$ samples does not define half period of the main harmonic exactly. That is the reason why averaging result is not constant, it also oscillates with 2ω, but with far smaller amplitude then algorithm result that doesn't use averaging. Oscillating of average value can be decreased if we apply averaging once more. It means that after this, algorithm has a full period delay of a basic harmonic. After applying averaging twice, we get a very "steady" value of the main harmonic's amplitude with a very small deviation (error), if the frequency is within *(47-53)Hz* range; that is acceptable when having in mind that frequency rarely deviates for more then

3Hz, in power systems with nominal frequency of *50Hz*. Delay is acceptable for all types of relays based on symmetrical components, so this algorithm is acceptable as well.

For calculating symmetrical components of three phased signals, relation (2) should be used on signals in each of three phases. For practical calculation of symmetrical voltage components, additional vectors of sinus and cosines are needed, defined in (7) and (8). Also two more additional vectors, both of $\left(\frac{m}{2}\right)$ in length are needed: one is vector of symmetrical components and the other one contains symmetrical components' averages. Vectors of samples are:

$$AOD = [a_1, a_2, a_3, ..., a_m]^T, \quad BOD = [b_1, b_2, b_3, ..., b_m]^T, \quad COD = [c_1, c_2, c_3, ..., c_m]^T. \tag{19}$$

This method includes only additions and multiplications, without having to calculate trigonometrical functions and divisions, which speeds up the procedure. After taking new samples a_{novo}, b_{novo}, c_{novo}, vectors (19) should be updated in this way:

$$a_1 = a_2, a_2 = a_3, ..., a_m = a_{novo}; \; b_1 = b_2, b_2 = b_3, ..., b_m = b_{novo}; \; c_1 = c_2, c_2 = c_3, ..., c_m = c_{novo}.$$

In this case also, we have a sliding window, and samples are treated as scalars. Relations (16), (17) and (18) give us U_{0f}, U_{df} and U_{if} for each window. For calculation of symmetrical components amplitude averages (of the main harmonic), additional vectors of $\left(\frac{m}{2}\right)$ in length need to be formed:

$$U0 = \left[U_{0f1}, U_{0f2}, U_{0f3}, ..., U_{0f\left(\frac{m}{2}\right)}\right]^T, \; UD = \left[U_{df1}, U_{df2}, U_{df3}, ..., U_{df\left(\frac{m}{2}\right)}\right]^T, \; UI = \left[U_{if1}, U_{if2}, U_{if3}, ..., U_{if\left(\frac{m}{2}\right)}\right]^T. \tag{20}$$

After calculation of new values of amplitudes U_{0fnovo}, U_{dfnovo} and U_{ifnovo}, using (2), (16), (17) and (18), vectors (20) should be updated by rejecting their first members and adding new ones:

$$U_{0f1} = U_{0f2}, \; U_{0f2} = U_{0f3}, ..., U_{0f\left(\frac{m}{2}\right)} = U_{0fnovo}; \; U_{df1} = U_{df2}, \; U_{df2} = U_{df3}, ..., U_{df\left(\frac{m}{2}\right)} = U_{dfnovo};$$

$$U_{if1} = U_{if2}, \; U_{if2} = U_{if3}, ..., U_{if\left(\frac{m}{2}\right)} = U_{ifnovo}.$$

After this procedure, we get:

$$US_0 = \frac{2}{m}\left(U_{0f1} + U_{0f2} + ... + U_{0f\left(\frac{m}{2}\right)}\right), US_d = \frac{2}{m}\left(U_{df1} + U_{df2} + ... + U_{df\left(\frac{m}{2}\right)}\right), US_i = \frac{2}{m}\left(U_{if1} + U_{if2} + ... + U_{if\left(\frac{m}{2}\right)}\right). \tag{21}$$

Procedure goes on in the same way for each new data window. For averaging of symmetrical components' amplitude averages (of the main harmonic), we need additional vectors of $\left(\frac{m}{2}\right)$ in length:

$$USS_0 = \left[US_{01}, US_{02}, ..., US_{0\left(\frac{m}{2}\right)}\right]^T, \; USS_d = \left[US_{d1}, US_{d2}, ..., US_{d\left(\frac{m}{2}\right)}\right]^T, \; USS_i = \left[US_{i1}, US_{i2}, ..., US_{i\left(\frac{m}{2}\right)}\right]^T. \tag{22}$$

After calculation of new amplitude averages US_{0novo}, US_{dnovo} and US_{inovo} using (21), vectors (22) should be updated by rejecting their first members and by adding new ones:

$$US_{01}=US_{02},\ US_{02}=US_{03},\ldots\ldots,US_{0\left(\frac{m}{2}\right)}=US_{0novo};\ US_{d1}=US_{d2},\ US_{d2}=US_{d3},\ldots\ldots,US_{d\left(\frac{m}{2}\right)}=US_{dnovo};$$

$$US_{i1}=US_{i2},\ US_{i2}=US_{i3},\ldots\ldots,US_{i\left(\frac{m}{2}\right)}=US_{inovo}.$$

After this procedure, we get:

$$USS_0=\frac{2}{m}\left(US_{01}+US_{02}+\ldots+US_{0\left(\frac{m}{2}\right)}\right),USS_d=\frac{2}{m}\left(US_{d1}+US_{d2}+\ldots+US_{d\left(\frac{m}{2}\right)}\right),$$

$$USS_i=\frac{2}{m}\left(US_{i1}+US_{i2}+\ldots+US_{i\left(\frac{m}{2}\right)}\right).(23)$$

Procedure goes on in the same way for each new data window. For practical application of Fourier's algorithm for symmetrical components, we need five vectors (of (m) in length) and six vectors (of $\left(\frac{m}{2}\right)$ in length). Maximal number of samples in data window is 50, so this algorithm is not resource hungry. Algorithm is simple (it uses scalar calculations) and fast enough for usage in relay protection.

3.POWER MEASUREMENT ALGORITHM

Power measurement algorithm requires processing of current and voltage samples, as described previously in current an voltage measurement algorithm. Using relations:

$$\underline{U}_f=U_{r_f}+jU_{i_f},\quad \underline{I}_f=I_{r_f}+jI_{i_f}\quad \text{and}\quad \underline{S}_f=P_f+jQ_f=\underline{U}_f\underline{I}_f^*=(U_{r_f}+jU_{i_f})\cdot(I_{r_f}-jI_{i_f}),$$

we get expressions for active and reactive power:

$$P_f=U_{r_f}I_{r_f}+U_{i_f}I_{i_f}\quad \text{i}\quad Q_f=U_{i_f}I_{r_f}-U_{r_f}I_{i_f}. \tag{24}$$

Based on analysis given previously, it can be concluded that Fourier's algorithm gives incorrect value of main harmonic's amplitude, when signal frequency differs from assumed frequency. When the signal is continually processed, main harmonic's current amplitude and main harmonic's voltage amplitude oscillate with frequency 2ω. This fact is used for modification of Fourier's algorithm for evaluation active and reactive power measurement of main current and voltage harmonics. Knowing that (P_f) and (Q_f) amplitudes oscillate with 2ω, better values can be accomplished by averaging results, after each half period of main harmonic's signal. In other words, if a signal is processed with (m) samples per period, it would be sufficient to calculate average of $\left(\frac{m}{2}\right)$ results. If averaging is applied, final processing results will have a time delay of half a period (of the main harmonic). However, any signal filtration always includes time delay, so one has to compromise between degree of filtration and amount of time delay. When signal frequency differs from assumed frequency in discrete Fourier series, $\left(\frac{m}{2}\right)$ samples does not define half period of the main harmonic exactly. That is the reason why averaging result is not constant, it also oscillates with 2ω, but with far smaller amplitude then algorithm result that doesn't use averaging. Oscillating of average value can be decreased if we apply averaging once more. It means that after this, algorithm has a full period delay of a basic harmonic. After applying averaging twice, we get a very "steady" value of the main harmonic's amplitude with a very small deviation, if the frequency is within *(47-53)Hz* range; that is acceptable when having in mind that frequency rarely deviates for more then *3Hz*, in power systems with nominal frequency of *50Hz*. One period delay is acceptable for all types of directional relays, so this algorithm can be used in relay protection for calculating direction. For practical usage of these relations, additional vectors CO, SI, voltage and current samples ((m) in length) and vector of powers and power averages($\left(\frac{m}{2}\right)$ in legth).

Vectors of voltage and current samples are:

$$UODB=[u_1,u_2,u_3,...,u_m]^T \text{ i } IODB=[i_1,i_2,i_3,...,i_m]^T . \tag{25}$$

After fetching new samples u_{novo} and i_{novo}, vectors (25) are updated as shown:

$$u_1=u_2,\ \ u_2=u_3,\ \ ...,\ \ u_m=u_{novo} \text{ i } i_1=i_2,\ \ i_2=i_3,\ \ ...,\ \ i_m=i_{novo} .$$

In this way we get sliding window, and signal samples are treated as scalars. For each data window, relations (5) give ($U_{r_f}, U_{i_f}, I_{r_f}$ and I_{i_f}). Then we use relation (24) and get (P_f) and (Q_f). For calculation of (P_f) and (Q_f) average (of the main harmonic), additional vectors of $\left(\frac{m}{2}\right)$ in length need to be formed:

$$PP=\left[P_{f1},P_{f2},P_{f3},...,P_{f\left(\frac{m}{2}\right)}\right]^T \text{ and } QP=\left[Q_{f1},Q_{f2},Q_{f3},...,Q_{f\left(\frac{m}{2}\right)}\right]^T . \tag{26}$$

After calculation of new P_{fnovo} and Q_{fnovo} using (5), vectors (26) should be updated by rejecting their first members and adding new ones:

$$P_{f1}=P_{f2},\ \ P_{f2}=P_{f3},\ \ ...,\ \ P_{f\left(\frac{m}{2}\right)}=P_{fnovo} \text{ and } Q_{f1}=Q_{f2},\ \ Q_{f2}=Q_{f3},\ \ ...,\ \ Q_{f\left(\frac{m}{2}\right)}=Q_{fnovo} .$$

After this procedure, we get:

$$PS=\frac{2}{m}\left(P_{f1}+P_{f2}+...+P_{f\left(\frac{m}{2}\right)}\right) \text{ and } QS=\frac{2}{m}\left(Q_{f1}+Q_{f2}+...+Q_{f\left(\frac{m}{2}\right)}\right) . \tag{27}$$

Procedure goes on in the same way for each new pair of (P_f) and (Q_f). For averaging of power averages, we need to form additional vectors of power averages (of $\left(\frac{m}{2}\right)$ in length):

$$PSS=\left[PS_1,PS_2,PS_3,...,PS_{\left(\frac{m}{2}\right)}\right]^T \text{ and } QSS=\left[QS_1,QS_2,QS_3,...,QS_{\left(\frac{m}{2}\right)}\right]^T . \tag{28}$$

After calculation of new power averages PS_{novo} and QS_{novo} using (27), vectors (28) need to be updated by rejecting their first members and adding new ones:

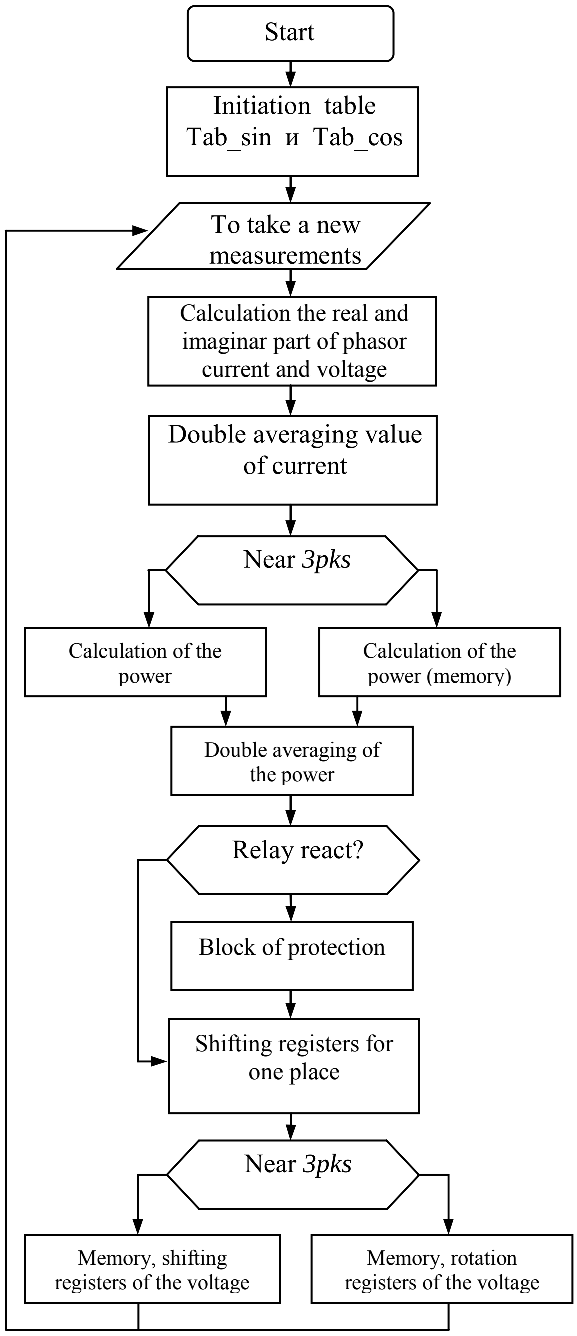
$$PS_1=PS_2,\ \ PS_2=PS_3,\ \ ...,\ \ PS_{\left(\frac{m}{2}\right)}=PS_{novo} \text{ and}$$

$$QS_1=QS_2,\ \ QS_2=QS_3,\ \ ...,\ \ QS_{\left(\frac{m}{2}\right)}=QS_{novo} .$$

After this procedure, we get:

$$PSS = \frac{2}{m}\left(PS_1 + PS_2 + \ldots + PS_{\left(\frac{m}{2}\right)}\right) \text{ and } QSS = \frac{2}{m}\left(QS_1 + QS_2 + \ldots + QS_{\left(\frac{m}{2}\right)}\right).$$

Procedure goes on in the same way for each new pair of power averages. For practical application of modifiedFourier's algorithm, we need four vectors of (m) in length, and four vectors of $\left(\frac{m}{2}\right)$ in length, which means, very few memory resources. Algorithm is simple and fast enough for usage in relay protection. Algorithm of directional protection (with measurement of line voltages and phase currents), based on power measurement, is shown in picture 1.



Picture 1. *Algorithm od directional protection function MPZ - ZIM*

4. ALGORITHM OF DIGITAL THERMAL PROTECTION

Thermal protection is one of the basic protection functions of all elements in power systems, and its algorithm is implemented as a stand-alone function in most of today's multifunctional relays. Indirect digital thermal protection of general type is implemented inside of microprocessor MPZ-ZIM protection. Its algorithm is mainly based on measurement of amplitudes of phase currents (of the main harmonic), while protected element, in thermal sense, is approximated as a homogenous body with time constants for heating and cooling. Algorithm needs more resources when it comes to cpu time.

In algorithm given, the heating model is based on assumption that protected object is thermally homogenous body, which can be modelled with time constants for heating and cooling, in processes of heating and cooling, respectively. The heating power is modelled with Joule's losses on three phased symmetrical element, with constant electrical phase resistance R and phase currents of alternate values. We assume that these losses are equally spread trough the element volume. Temperature of the surrounding is modelled as constant temperature which can be tuned as needed. Under the conditions given, differential equation of thermal balance (equation of heating) is used:

$$R\left(I_A^2+I_B^2+I_C^2\right)dt = Cd\theta + K\theta dt\,, \tag{29}$$

symbols given represent: I_A, I_B, I_C-effective values of phase currents calculated with the use of recursive Fourier's method; $R\left(I_A^2+I_B^2+I_C^2\right)dt$- energy of heating, generated in three-phased element during a short interval of time dt; $Cd\theta$-increase in accumulated energy of heating, which is manifested by temperature increase in amount of $d\theta$ (C is heating capacity); $K\theta dt$-heating energy emitted into the surrounding or the cooling medium, in time interval dt (K is cooling constant and θ is overtemperature of object).

Possibility of protected object having asymmetrical current load is assumed in equation (29). That is, for one thing, a real possibility (cuts in phase lines, non symmetrical voltage excitation, non symmetrical load etc), and, on the other hand, that doesn't at all cause any technical problems with thermal protection function realisations , because multifunctional relay processes all of three phase currents. In steady state ($d\theta = 0$) there is a balance between heating power and colling power. If the heating is caused by nominal phase currents, which means $I_A = I_B = I_C = I_n$ (nominal phase current is defined with nominal apparent power and nominal voltage, and for specific cooling conditions), then stationary overtemperature corresponds with nominal overtemperature:

$$\theta_n = \frac{3RI_n^2}{K}\,. \tag{30}$$

Dividing this with nominal over temperature θ_n, equation (29) can be normalized, and it results in:

$$T_Z\frac{d\left(\theta(\%)\right)}{dt} + \theta(\%) = 100\left(\frac{I_A^2+I_B^2+I_C^2}{3I_n^2}\right), \tag{31}$$

symbols given represent: $T_Z = \dfrac{C}{K}$-heating time constant of protected element, $\theta(\%) = 100\dfrac{\theta}{\theta_n}$-percentage of element over temperature. In cases of large current overloads ($I_A \vee I_B \vee I_C > 2I_n$), heating power is much larger then cooling power. That causes local overheating, which makes starting assumption about equal heating very rough. For the sake of more efficient thermal protection, in cases of large current overloads, the heating process can be treated as adiabatic. The adiabatic heating equation is :

$$T_Z\frac{d\left(\theta(\%)\right)}{dt} = 100\left(\frac{I_A^2+I_B^2+I_C^2}{3I_n^2}\right). \tag{32}$$

In this analysis , the heating process is treated as adiabatic with three phased heating generation , if one of phase currents at least , is at least two times larger then nominal. This ensures more secure approach and some effects not treated in this analysis are compensated (increase of electrical resistance with the temperature inerease, for example). It can also be assumed that heating time counstant, in cases of big overloads, is smaller then the time constant in usual working conditions. When the element is disconnected from its power supply, thermal process of cooling is taking place inside of it; generally speaking, cooling constant T_H can be different from heating constant T_Z. The difference usually exists in rotating electrical machines, where conditions change when the machine is switched off; cooling conditions change because the rotor and its cooling fan stop, so ventilation conditions change also. For that reason, it is very important to assume different values for time constants during cooling, (these constants are basiclly the same in all statical components, that have no rotating parts). Taking this into consideration, thermal process of cooling is given in the next expression:

$$T_H \frac{d(\theta(\%))}{dt} = -\theta(\%)\,. \tag{33}$$

One of specific things related to thermal protection function and its algorithm, is the fact that entire algorithm needs to be permanently processed. So, an optimized algorithm for temperature measurement is needed, which ensures fast and reliable processing. Besides, opposite of other protection types, thermal protection function must be active even in cases when element loses its power supply, in order to track its cooling and to continue temperature tracking when it gets its power supply back online. For that reason, continual work of thermal protection function must be achieved. Relatively small value of sampling period T_S (of the order of few ms) compared to heating time constant T_Z (which is, for most elements in power systems, from order of few minutes to few tens of minutes) makes it possible to solve the differential equation of heating numerically. Knowing this, differential temperature increase $d\theta$ in (29) can be replaced with finite temperature increase $\Delta\theta$, made in time $\Delta t = T_S = 1/f_S$, equation (29) then becomes:

$$\Delta\theta_{(i+1)}(\%) = \frac{T_S}{T_Z}\left[100\left(\frac{I^2_{A(i+1)} + I^2_{B(i+1)} + I^2_{C(i+1)}}{3I_n^2}\right) - \theta_i(\%)\right], \tag{34}$$

where symbols given represent: $I_{A(i+1)}, I_{B(i+1)}, I_{C(i+1)}$ -estimated phase currents in $(i+1)$-th sampling step; $\Delta\theta_{(i+1)}(\%)$ -percentage temperature addition in time interval T_S, which corresponds with $(i+1)$-th sampling step (temperature base value corresponds with nominal overtemperature of protected element) ; $\theta_i(\%)$ - percentage overtemperature value of the element, at the end of i-th sampling step. Similar to previous analysis and equation (32),expression for temperature addition during adiabatic heating can be given as:

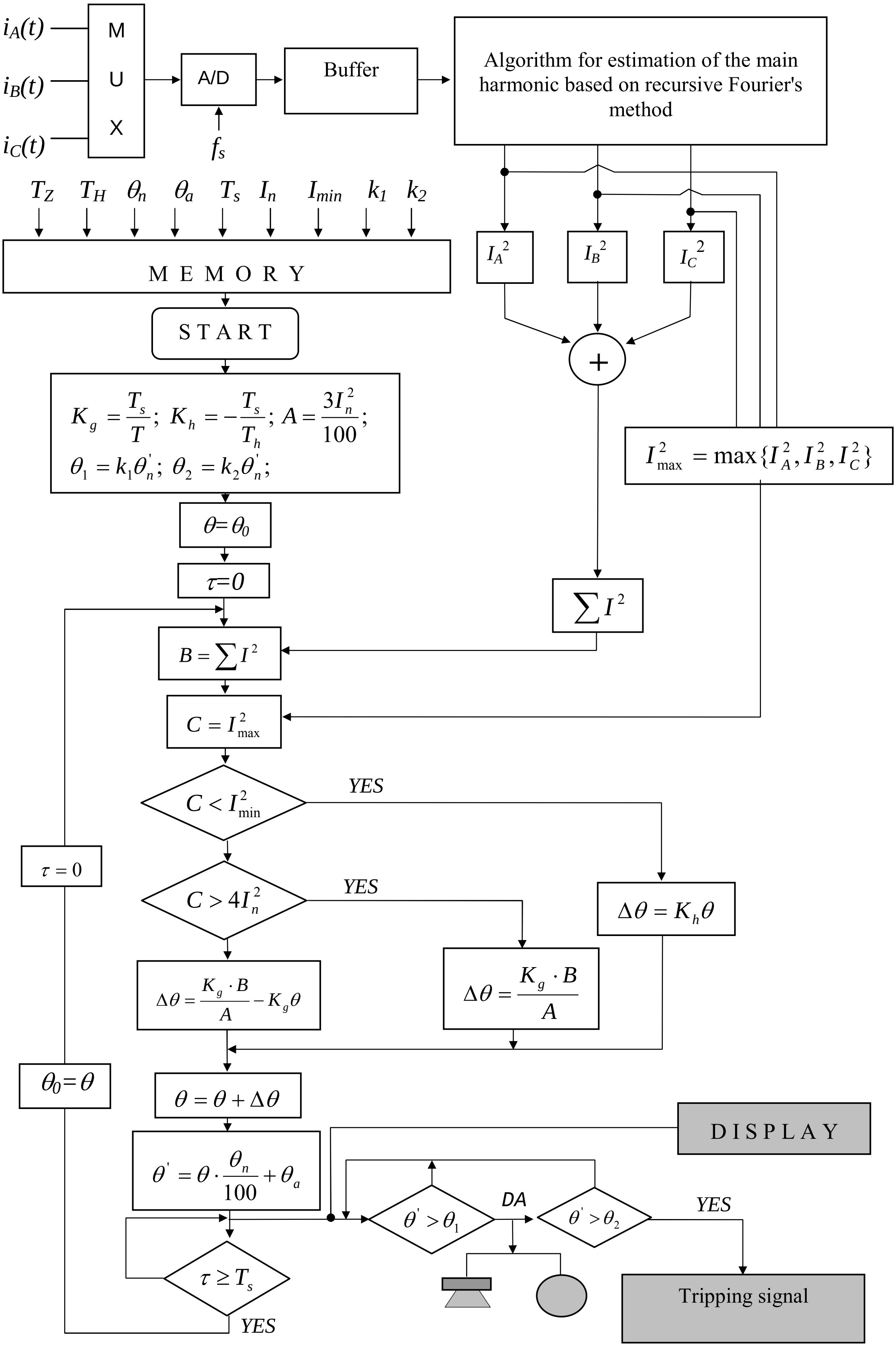
$$\Delta\theta_{(i+1)}(\%) = 100\frac{T_S}{T_Z}\left(\frac{I^2_{A(i+1)} + I^2_{B(i+1)} + I^2_{C(i+1)}}{3I_n^2}\right). \tag{35}$$

In cases when protected element is disconnected from its power supply, cooling conditions can, as explained before, take place with time constant that is not the same. This possibility must also be included in algorithm. Detection of this state is done by comparing the amplitude of phase current with some specified minimal current value ($I_{\min}$). This current value ($I_{\min}$), can occur even when element is offline and it can be caused by some parasite effects, such as capacitive currents for example. So, if the condition is met:

$$I_{A(i+1)}, I_{B(i+1)}, I_{C(i+1)} < I_{\min}\,, \tag{36}$$

temperature addition is calculated by the use of next expression:

$$\Delta\theta_{(i+1)}(\%) = -\frac{T_S}{T_H}\theta_i(\%)\,. \tag{37}$$



Picture 2. *Logical block diagram of digital thermal protection*

Percentage overtemperature of protected object, at the end of $(i+1)$-th step, is:

$$\theta_{(i+1)}(\%) = \theta_i(\%) + \Delta\theta_{(i+1)}(\%) . \tag{38}$$

Knowing that relay can also show absolute temperature on display, this option should be also included in algorithm. Absolute temperature at the end of $(i+1)$-th sampling step ($\theta'_{(i+1)}$) is:

$$\theta'_{(i+1)} = \theta_{(i+1)} + \theta_a , \tag{39}$$

where is θ_a -surrounding temperature (minimal temperature that element can have when it is offline, for the given temperature conditions). Thermal protection is usually two level protection. First level is the alarm and second one is operational. Algorithm of thermal protection is shown in picture 2.

5.FREQUENCY PROTECTION ALGORITHM

Frequency protection algorithm is based on least error squares error (LES) method. Implemented algorithm shows improvements compared to iterative non recursive LES method, when it comes to processing speed. Complete mathematical method for minimization of sums of squares, is presented as a fixed coefficient matrix in relay's memory buffer. Determination of frequency, then becomes a multiplication of vectors of samples (of current data window) with corresponding data rows. Algorithm is optimized in terms of coefficient matrix forming, for practical implementation. Voltage signal (its frequency is being measured), generally is a composite -periodical function of time; it can contain a DC voltage component along with higher harmonics, both in normal and disturbed working states. Therefore, the input voltage signal (in A/D converter), can in mathematical terms, be shown as:

$$u(t) = U_0 + \sum_{k=1}^{M} U_k \sin(k\omega t + \theta_k) + e(t) . \tag{40}$$

Using basic trigonometric relations, previous relation can also be written as:

$$u(t) = U_0 + \sum_{k=1}^{M} U_{rk} \sin(k\omega t) + \sum_{k=1}^{M} U_{ik} \cos(k\omega t) + e(t) , \tag{41}$$

given symbols are: $u(t)$ -voltage magnitude in time moment t; U_0 -DC voltage component; $\omega = 2\pi f$ -angle frequency of the main harmonic; M -highest harmonic order in voltage signal; U_k -amplitude of k-th harmonic; θ_k -phase of k-th harmonic; $U_{rk} = U_k \cos\theta_k$, $U_{ik} = U_k \sin\theta_k$ -real and imaginary component of k-th harmonic, respectively; $e(t)$ -random noise signal.

By expanding trigonometric functions $\sin(k\omega t)$ and $\cos(k\omega t)$ in Taylor's series, near nominal (assumed) frequency ω_0, expression (41) can be linearized. Linearized analytical voltage signal form, near its nominal frequency ω_0, can be given as:

$$u(t) = U_0 + \sum_{k=1}^{M} \left[U_{rk} \sin(k\omega_0 t) + U_{rk} \Delta\omega k t \cos(k\omega_0 t) \right] + \sum_{k=1}^{M} \left[U_{ik} \cos(k\omega_0 t) - U_{ik} \Delta\omega k t \sin(k\omega_0 t) \right] + e(t) . \tag{42}$$

Relation (42) can be simplified, and shown in a form:

$$u(t) = \sum_{j=1}^{4M} a_j(t) x_j + e(t) , \tag{43}$$

given symbols are: $a_j(t)$ -coefficients; x_j -variables defined in the following way:

$$a_1 = 1,\ \ a_{1+k} = \sin(k\omega_0 t),\ \ a_{M+1+k} = kt\cos(k\omega_0 t),\ \ a_{2M+1+k} = \cos(k\omega_0 t),\ \ a_{3M+1+k} = -kt\sin(k\omega_0 t)$$
$$x_1 = U_0,\ \ x_{1+k} = U_{rk},\ \ x_{M+1+k} = U_{rk}\Delta\omega,\ \ x_{2M+1+k} = U_{ik},\ \ x_{3M+1+k} = U_{ik}\Delta\omega,\ \ k = 1,2,\ldots,M. \tag{44}$$

Each of the voltage signals on A/D converter output, can be described with (43). If we watch only (m) consecutive voltage samples (which are also called - a data window), then, by writing relation (43) for each of the samples - we can form a system of (m) relations. It can be shown in a matrix form as:

$$[u] = [a][x] + [e], \tag{44}$$

given symbols are: $[u] = [u_1, u_2, \ldots, u_m]^T$ -vector of samples; $[e]$-vector of random noise (sample deviation from assumed signal model); $[a]$-coefficient matrix, with a dimension of $(m \cdot (4M+1))$; $[x]$-variable vector defined in (44). Sampling frequency $f_s (f_s = 1/T)$ is considered to be constant in this analysis, so all samples have equal time period T between them. Knowing this, matrix coefficients in $[a]$ are calculated according to the following relations:

$$a_1(n) = 1,\ \ a_{1+k}(n) = \sin(k\omega_0 nT),\ \ a_{M+1+k}(n) = knT\cos(k\omega_0 nT),\ \ a_{2M+1+k}(n) = \cos(k\omega_0 nT),$$
$$a_{3M+1+k}(n) = -knT\sin(k\omega_0 nT)\ \ \ n = 1,2,\ldots,m. \tag{45}$$

Algorithm considers that variables $x_j (j = 1,2,\ldots,4M+1)$, (elements of vector $[x]$ in (44)) do not change their values during one data window. This assumption is correct, knowing that frequency is related with generator's speed change in the system. Since these machines have a relatively large inertia, frequency usually won't change during one data window even with bigger disturbances of active power (data window width usually ranges from one to three periods of measured signal $(0.02 - 0.06)s$). So, system of equations (44), has $(4M+1)$ unknown variables. Minimal data window width should be at least $m_{\min} = 4M+1$, in order to have a single solution. Usually, data window width is larger then minimal, which means that system is redundant ($m > 4M+1$) -that ensures more stability in the cases when signal contains much noise. Knowing that each of the equations in (44) contains noise signal, it is most appropriate to use least error squares method. Essential thing in this method would be to find an optimal estimation of vector of variables $[x]$, so that random noise is minimal. According to this method, optimal estimation of vector $[x]^*$ would be:

$$[x]^* = \left[[a]^T [a]\right]^{-1} [a]^T [u] = [A][u], \tag{46}$$

$[A]$ is left pseudoinverse matrix of $[a]$. Elements of this matrix depend on sampling period and assumed frequency ω_0, and they can be calculated in advance, which is good because it saves cpu time. After calculation of vector $[x]$, we can also determine: DC voltage component, effective values and starting phases (calculated in relation to first sample in data window) of basic harmonic and higher harmonics included in the model, frequency deviation from nominal value. Calculation of these parameters is done according to following relations :

$$U_0 = x_1,\ \ U_k = \sqrt{U_{rk}^2 + U_{ik}^2} = \sqrt{x_{1+k}^2 + x_{2M+1+k}^2},\ \ \Delta\omega = \frac{U_{rk}\Delta\omega}{U_{rk}} = \frac{x_{M+1+k}}{x_{1+k}},$$
$$\Delta\omega = \frac{U_{ik}\Delta\omega}{U_{ik}} = \frac{x_{3M+1+k}}{x_{2M\,1+k}},\ \ |\Delta\omega| = \sqrt{\frac{x_{M+1+k}^2 + x_{3M+1+k}^2}{x_{1+k}^2 + x_{2M+1+k}^2}},\ \ \omega = \omega_0 + \Delta\omega. \tag{47}$$

Using second and third relation in (47) (usually relations are applied to first harmonic cause its dominant) frequency deviation in current data window, which means its sign and modulus, can be determined. Accuracy of frequency determination is mostly satisfying. More advanced approach in least error squares method would include frequency feedback. Basic idea is iterative correction of frequency ω_0, which results in bringing linearization point closer to real frequency in each iteration step (equation (43)). This way, we can achieve desired accuracy of frequency, in certain number of steps. Iterative algorithm can be given with following equations :

$$[x_i]=[A_i(\omega_{i-1})][u_i], \quad [x_{i+1}]=[A_{i+1}(\omega_i)][u_i], \quad \omega_i=\omega_{i-1}+\Delta\omega_i, \tag{48}$$

given symbols are: ω_{i-1},ω_i - estimated frequency in $(i-1)$-th and i-th iteration respectively; $[A_i(\omega_{i-1})],[A_i(\omega_i)]$-left pseudoinverse matrices of coefficients, calculated for ω_{i-1},ω_i, recpectively; $[x_i],[x_{i+1}]$-optimal estimation of vector of variables in $(i-1)$-th and i-th iteration, respectively; $\Delta\omega_i$-correction of frequency astimated in $(i-1)$-th iteration. Condition needed for iteration stop is that current frequency correction is smaller of some value, given in advance. Iterative algorithm has proven to have good frequency estimation qualities. Drawback of this method lies in the fact that coefficients of matrix $[a]$ (its left pseudoinverse matrix) need to be calculated in every iteration. This results in increased cpu time, especially when inverting $[a]$. That is the main reason for rare usage of this method in relay protection. In order to speed up the process, way to avoid calculation of $[A]$ in each iteration step needs to be found. Therefore, $[a]$ needs to be analysed; $[A]$ as its pseudoinverse matrix also; relations (47) which define a process of extraction of measured frequency from vector of variables $[x]$, are also of interest. Following relation shows spread form of matrix $[a_i]$; it corresponds with linearized model of measured voltage signal, near current frequency ω_{0i}, estimated in i-th iteration:

$$[a_i]=\begin{bmatrix}
1 & 1 & \dots & 1 \\
\sin(\omega_{0i}T) & \sin(\omega_{0i}2T) & \dots & \sin(\omega_{0i}mT) \\
\sin(2\omega_{0i}T) & \sin(2\omega_{0i}2T) & \dots & \sin(2\omega_{0i}mT) \\
\dots\dots\dots & \dots\dots\dots & \dots & \dots\dots\dots \\
\sin(M\omega_{0i}T) & \sin(M\omega_{0i}2T) & \dots & \sin(M\omega_{0i}mT) \\
T\cos(\omega_{0i}T) & 2T\cos(\omega_{0i}2T) & \dots & nT\cos(\omega_{0i}nT) \\
T\cos(2\omega_{0i}T) & 2T\cos(2\omega_{0i}2T) & \dots & nT\cos(2\omega_{0i}nT) \\
\dots\dots\dots & \dots\dots\dots & \dots & \dots\dots\dots \\
T\cos(M\omega_{0i}T) & 2T\cos(M\omega_{0i}2T) & \dots & nT\cos(M\omega_{0i}nT) \\
\cos(\omega_{0i}T) & \cos(\omega_{0i}2T) & \dots & \cos(\omega_{0i}nT) \\
\cos(2\omega_{0i}T) & \cos(2\omega_{0i}2T) & \dots & \cos(2\omega_{0i}nT) \\
\dots\dots\dots & \dots\dots\dots & \dots & \dots\dots\dots \\
\cos(M\omega_{0i}T) & \cos(M\omega_{0i}2T) & \dots & \cos(M\omega_{0i}nT) \\
-T\cos(\omega_{0i}T) & -2T\cos(\omega_{0i}2T) & \dots & -nT\cos(\omega_{0i}nT) \\
-T\cos(2\omega_{0i}T) & -2T\cos(2\omega_{0i}2T) & \dots & -nT\cos(2\omega_{0i}nT) \\
\dots\dots\dots & \dots\dots\dots & \dots & \dots\dots\dots \\
-T\cos(M\omega_{0i}T) & -2T\cos(M\omega_{0i}T) & \dots & -nT\cos(M\omega_{0i}T)
\end{bmatrix}. \tag{49}$$

Matrix $[a_i]_{mx(4M+1)}$ in (49) is shown in general form, which means that signal includes DC voltage component, main harmonic and higher harmonics up to order of M. Number of matrix rows can be decreased, if we do not take into count some of them (the even ones, for example). Number of columns in matrix corresponds with number of samples m. In order to form an optimal estimation of vector $[x_i]$, it is needed according to (46), to calculate matrix $[A_i]$. Spread form of matrix $[A_i]$ is shown in the following relation:

$$[A_i]=\left[[a_i]^T[a_i]\right]^{-1}[a_i]^T=\begin{bmatrix} A_{11}(\omega_{0i}) & A_{12}(\omega_{0i}) & \dots & A_{1m}(\omega_{0i}) \\ A_{21}(\omega_{0i}) & A_{22}(\omega_{0i}) & \dots & A_{2m}(\omega_{0i}) \\ \dots\dots\dots & \dots\dots\dots & \dots & \dots\dots\dots \\ A_{(4M+1)1}(\omega_{0i}) & A_{(4M+1)2}(\omega_{0i}) & \dots & A_{(4M+1)m}(\omega_{0i}) \end{bmatrix}_{(4M+1)xm} . \tag{50}$$

Using this relation, equation (46) can be shown in spread form, given with (51). Calculation of frequency correction $\Delta\omega_i$, in comparison to current frequency ω_{0i}, can be done by using second and third expression in (47). In matrix equation (51), rows (except the first and the last one) that also exist in calculation of frequency correction, are marked; they are based on phasor components of voltage's main harmonic (second and third expression in (47), already mentioned). Second and third equation in (47) define frequency deviation, both in value and in sign. That means that only two rows of current matrix $[A_i]$ are active, in frequency calculation. However, for reliable frequency estimation, four rows in matrix $[A_i]$ need to be monitored (two of them correspond with first relation in (47) and the other two with second relation in (47)). To be more precise, during signal processing, it can occur that first sample in data window (referent sample) has similar value (or close value) with the value of voltage signal at its peak or at zero; that means that one of the current phasor components (real or imaginary) becomes equal (or near) to zero.

$$\begin{bmatrix} x_{1i} \\ x_{2i} \\ \dots \\ x_{(M+2)i} \\ \dots \\ x_{(2M+2)i} \\ \dots \\ x_{(3M+2)i} \\ \dots \\ x_{(4M+1)i} \end{bmatrix}^* = \begin{bmatrix} A_{11}(\omega_{0i}) & A_{12}(\omega_{0i}) & \dots & A_{1m}(\omega_{0i}) \\ A_{21}(\omega_{0i}) & A_{22}(\omega_{0i}) & \dots & A_{2m}(\omega_{0i}) \\ \dots\dots\dots & \dots\dots\dots & \dots & \dots\dots\dots \\ A_{(M+2)1}(\omega_{0i}) & A_{(M+2)2}(\omega_{0i}) & \dots & A_{(M+2)m}(\omega_{0i}) \\ \dots\dots\dots & \dots\dots\dots & \dots & \dots\dots\dots \\ A_{(2M+2)1}(\omega_{0i}) & A_{(2M+2)2}(\omega_{0i}) & \dots & A_{(2M+2)m}(\omega_{0i}) \\ \dots\dots\dots & \dots\dots\dots & \dots & \dots\dots\dots \\ A_{(3M+2)1}(\omega_{0i}) & A_{(3M+2)2}(\omega_{0i}) & \dots & A_{(3M+2)m}(\omega_{0i}) \\ \dots\dots\dots & \dots\dots\dots & \dots & \dots\dots\dots \\ A_{(4M+1)1}(\omega_{0i}) & A_{(4M+1)2}(\omega_{0i}) & \dots & A_{(4M+1)m}(\omega_{0i}) \end{bmatrix} \begin{bmatrix} u_{1i} \\ u_{2i} \\ \dots \\ \\ \\ u_{ki} \\ \\ \\ \dots \\ u_{mi} \end{bmatrix} \tag{51}$$

In this case, one of the two relations in (47) becomes numerically undefined (division with zero), and results in calculation error. Using fourth relation in (47) this can be avoided, but this relation can't give deviation sign, so its practically useless. The only solution is to monitor both components and to use one of them that is larger by modulus value, for current data window. Matrix elements of identified rows (in $[A_i]$) as well as other matrix elements which correspond with frequency ω_{0i}, have to be calculated in each iteration. That means more cpu time, when having in mind relatively complicated matrix $[a_i]$ and large matrix computations. Basic idea is to calculate elements of matrix $[A_i]$ in advance, for sequence of equally distant frequency values, within the defined range around nominal value. For each of the $[A_i]$ matrices calculated, submatrices that contain only four needed rows would be formed; those rows are enough for frequency estimation around working point. Submatrices together, form one matrix $[A^*]$, which includes complete measurement range. This matrix is put into memory of digital relay and addressed in appropriate way. Only the identification of corresponding submatrix (whose base frequency is nearest to current process' frequency) is done in iterative process. Rest of the process includes simple computations that do not include much cpu time. Number of submatrices N, which form matrix $[A^*]$ depend measurement range $(\omega_{\max}-\omega_{\min})$ and computation step $\Delta\omega_0$, which means-with range frequency. If matrices have equally distant frequency, then N is calculated with:

$$N=\frac{\omega_{\max}-\omega_{\min}}{\Delta\omega_0}, \tag{52}$$

parameters $\omega_{\max}, \omega_{\min}$ and $\Delta\omega_0$ need to be chosen, so that N is integer. Relation (53) is general form of matrix $[A^*]$. According to things said, $[A^*]$ has dimension of $(4Nxm)$. It is important to notice that matrix dimensions don't depend on harmonic level included in the signal. According to mathematical model, relatively simple frequency (and voltage) estimation algorithm is formed. Algorithm is shown in picture number 3. Symbols have the following meaning :

$\omega_{\min}, \omega_{\max}$ -lower and upper frequency range (which is covered with coefficient matrix $[A^*]$) boundary; $\Delta\omega_0$ -frequency step in $[A^*]$ matrix; ω_n -nominal frequency; i -iteration counter; $i_{\max}$ -maximal allowed number of iterations in frequency calculation; j -current row pointer in $[A^*]$; $j_{\max}$ -maximal pointer value ($j_{\max} = N - 3 = \dfrac{\omega_{\max} - \omega_{\min}}{\Delta\omega_0} - 3$); R, p -working registers; u -register for storage of samples from current data window; u_r, u_q -registers for storage of voltage's real and imaginary component (of the main harmonic); ω -actual frequency; $\Delta\omega$ -actual correction of frequency, calculated in previous step (iteration).

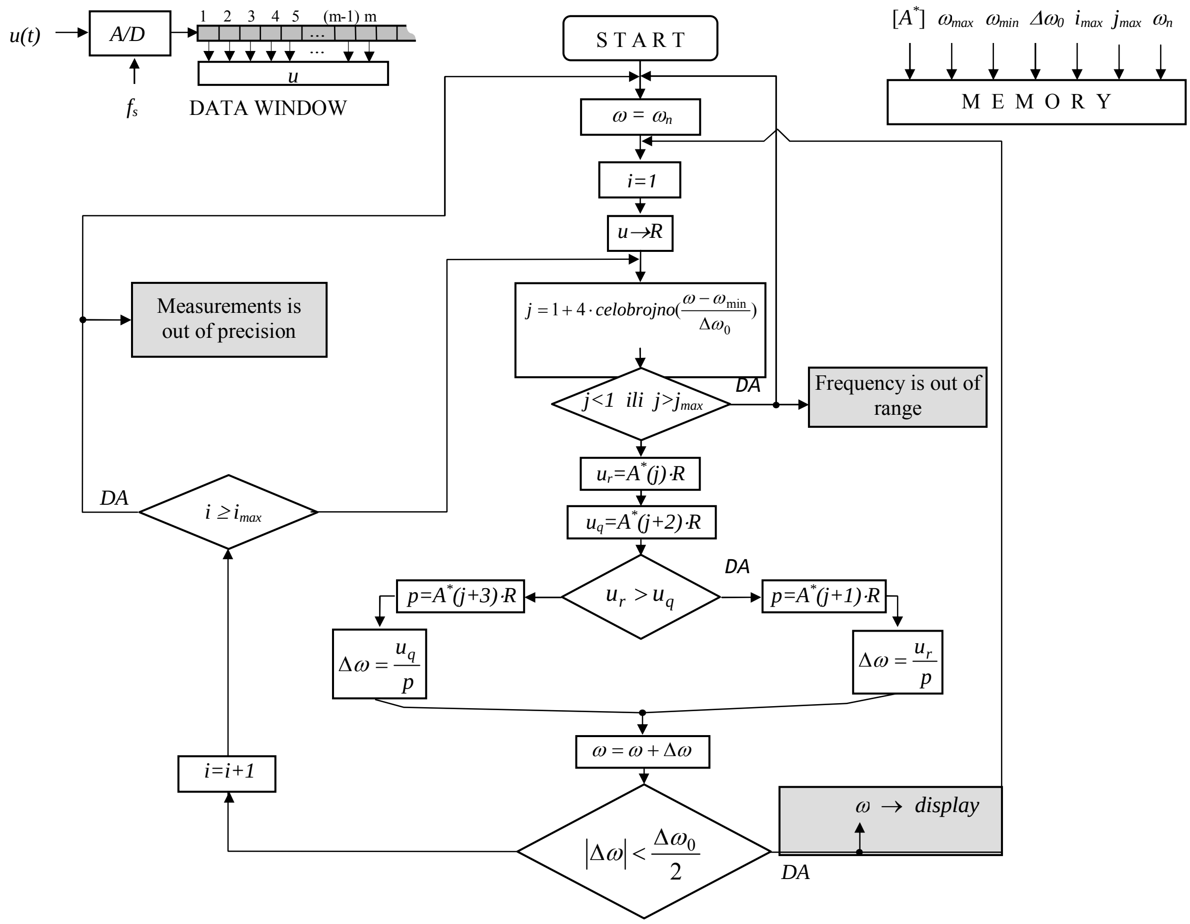
$$[A^*] = \begin{bmatrix}
A_{21}(\omega_{\min}) & A_{22}(\omega_{\min}) & \cdots & A_{2m}(\omega_{\min}) \\
A_{(M+2)1}(\omega_{\min}) & A_{(M+2)2}(\omega_{\min}) & \cdots & A_{(M+2)m}(\omega_{\min}) \\
A_{(2M+2)1}(\omega_{\min}) & A_{(2M+2)2}(\omega_{\min}) & \cdots & A_{(2M+2)m}(\omega_{\min}) \\
A_{(3M+2)1}(\omega_{\min}) & A_{(3M+2)2}(\omega_{\min}) & \cdots & A_{(3M+2)m}(\omega_{\min}) \\
\cdots\cdots & \cdots\cdots & \cdots & \cdots\cdots \\
A_{21}(\omega_{\min}+\Delta\omega_0) & A_{22}(\omega_{\min}+\Delta\omega_0) & \cdots & A_{2m}(\omega_{\min}+\Delta\omega_0) \\
A_{(M+2)1}(\omega_{\min}+\Delta\omega_0) & A_{(M+2)2}(\omega_{\min}+\Delta\omega_0) & \cdots & A_{(M+2)m}(\omega_{\min}+\Delta\omega_0) \\
A_{(2M+2)1}(\omega_{\min}+\Delta\omega_0) & A_{(2M+2)2}(\omega_{\min}+\Delta\omega_0) & \cdots & A_{(2M+2)m}(\omega_{\min}+\Delta\omega_0) \\
A_{(3M+2)1}(\omega_{\min}+\Delta\omega_0) & A_{(3M+2)2}(\omega_{\min}+\Delta\omega_0) & \cdots & A_{(3M+2)m}(\omega_{\min}+\Delta\omega_0) \\
\cdots\cdots & \cdots\cdots & \cdots & \cdots\cdots \\
..\; A_{21}(\omega_{\min}+2\Delta\omega_0) & A_{22}(\omega_{\min}+2\Delta\omega_0) & \cdots & A_{2m}(\omega_{\min}+2\Delta\omega_0) \\
A_{(M+2)1}(\omega_{\min}+2\Delta\omega_0) & A_{(M+2)2}(\omega_{\min}+2\Delta\omega_0) & \cdots & A_{(M+2)m}(\omega_{\min}+2\Delta\omega_0) \\
A_{(2M+2)1}(\omega_{\min}+2\Delta\omega_0) & A_{(2M+2)2}(\omega_{\min}+2\Delta\omega_0) & \cdots & A_{(2M+2)m}(\omega_{\min}+2\Delta\omega_0) \\
A_{(3M+2)1}(\omega_{\min}+2\Delta\omega_0) & A_{(3M+2)2}(\omega_{\min}+2\Delta\omega_0) & \cdots & A_{(3M+2)m}(\omega_{\min}+2\Delta\omega_0) \\
\cdots\cdots & \cdots\cdots & \cdots & \cdots\cdots \\
\cdots\cdots & \cdots\cdots & \cdots & \cdots\cdots \\
A_{21}(\omega_{\max}) & A_{22}(\omega_{\max}) & \cdots & A_{2m}(\omega_{\max}) \\
A_{(M+2)1}(\omega_{\max}) & A_{(M+2)2}(\omega_{\max}) & \cdots & A_{(M+2)m}(\omega_{\max}) \\
A_{(2M+2)1}(\omega_{\max}) & A_{(2M+2)2}(\omega_{\max}) & \cdots & A_{(2M+2)m}(\omega_{\max}) \\
A_{(3M+2)1}(\omega_{\max}) & A_{(3M+2)2}(\omega_{\max}) & \cdots & A_{(3M+2)m}(\omega_{\max})
\end{bmatrix} \tag{53}$$

Suggested algorithm is based on assumption that signal's frequency can be considered constant during one data window. Sampling frequency and data width are non changeable and defined in advance. Initialization of frequency measurement sets working frequency to value ω_n. According to that, pointer j is automatically being set to first row of coefficients' submatrix $[A^*]$ that corresponds with nominal frequency ω_n. Iterative process starts with succesive multiplication of submatrix rows and current vector of samples. Iteration process stops in one of three cases:

1)Calculated frequency is smaller then one half of matrix step $\Delta\omega_0$. In that case, frequency correction cannot be continued anymore. Result of measurement is correct and it is a sum of frequency calculated in the previous step and actual correction $\Delta\omega$. New data window is taken, working frequency stays on value that has been calculated, and new iterative process is started.

2)Current frequency ω is out of measurement range covered by matrix $[A^*]$. "Breaktrough" of measurement range is signalled. New data window is taken and frequency set as ($\omega = \omega_n$) . Estimation is continued.

3)Number of iteration exceeds maximum allowed. This limit was included to prevent possibility to "stuck" and to prevent slow convergence. Way to solve this is the same as restart described in 2).



Picture 3.*Logical block diagram of frequency algorithm*

Complete algorithm suggested here is based on relatively small number of mathematical relations which do not require much computer hardware resources; therefore it is very fast. Suggested algorithm does estimation of phasor components (of voltage's main harmonic) with intention of frequency determination, but it is also a voltage estimator. Therefore, besides frequency measurement, algorithm also does voltage's effective value measurement (of the signal's main harmonic). Voltage measured in this way is easy for showing on display, cause it represents mean effective value (of the main voltage harmonic) on an interval of data window width.

6.CONCLUSION

Protection functions' algorithms, in microprocessor protection of distributive leads MPZ - ZIM, have been thoroughly shown trough this work. Common to all of them is their simplicity and accuracy in frequent range that's of interest. Algorithms for measurement of currents, voltages and symmetrical components are based on recursive Fourier's method, which has been modified for this purpose, so frequency change within few *Hz* has very small influence on their accuracy. Then, on the basis of these processed signals (of currents, voltages and symmetrical components), algorithm for measurement of power has been thoroughly shown; also, one option of directional protection has also been given (in the form of block diagram). On the basis of digitally processed phase currents, algorithm for thermal protection has been shown (as well as its block diagram). In the end, frequency protection algorithm, based on least error squares method, has been presented.

7.LITERATURE

[1]M.Đurić, *Tehnika zaštite u elektroenergetici*, Beopres, Beograd, 1988.
[2]V.Terzija, M.Đurić, *Digitalne metode za merenje frekvencije niskofrekventnih signala,* Beopres, Beograd, 1988.
[3]J.Z.Yang, C.W.Liu, *A Precise Calculation of Power System Frequency*, IEEE Transactions on Power Delivery, Vol. 16., No.3, July 2001.