September 10 - 12, 2003
Pilsen, Czech Republic

CONTRIBUTION TO THE MINIMIZATION OF LOOSES IN THREE-PHASE LINE

DANIEL MAYER, PETR KROPÍK[1]

Abstract: In the article is developed the approach for minimizing losses in line by passive compensator. Given calculation method is illustrated by three numerical problems. It is possible to modify this method even for design of filters enabling increased quality of transmitted electric energy by suppressing unwanted higher harmonics in network.

Keywords: looses in three-phase line, Matlab, objective function, optimization.

1 Defining the solved problem

Three-phase non-linear load of inductive character is connected to balanced three-phase network, whose voltages are sinusoidal functions with period T, Fig. 1. The load draws currents $i_1(t)$, $i_2(t)$, $i_3(t)$ that are periodical, generally unbalanced and nonsinusoidal. To the load terminals shunt compensators are attached that contain two-poles RLC. The inductance of reactance coils is chosen so that resonance frequency f_r of the two-poles is distanced from the frequency of higher harmonics generated by the nonlinear load, usually $f_r = 189$ Hz or $f_r = 134$ Hz. The network is connected with the load through line with currents $i_{\ell 1}(t)$, $i_{\ell 2}(t)$, $i_{\ell 3}(t)$. Let us consider that the terminal voltage-rigidity of load can be considered sufficient. Time course of voltage on load terminal is known and we define parameters R, L, C of compensation two-poles, for which the losses in line are minimal. We minimize the functional, which is objective function

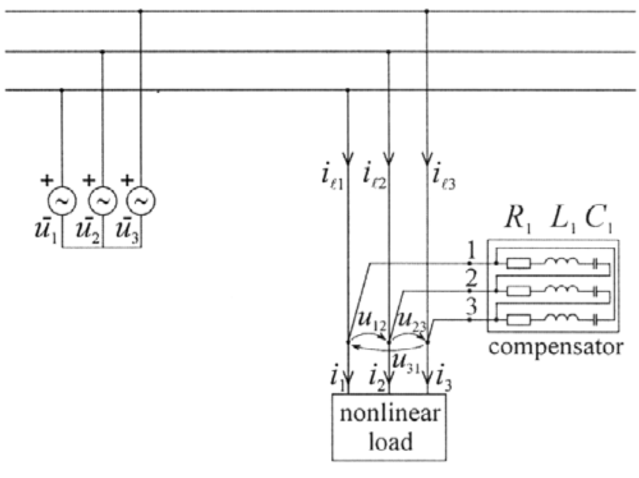

Fig. 1. Three-phase circuit structure.

$$F = \frac{1}{T}\int_0^T (i_{\ell 1}^2 + i_{\ell 2}^2 + i_{\ell 3}^2)\,\mathrm{d}t \qquad (1)$$

[1] Faculty of Electrical Engineering, University of West Bohemia Pilsen, Sady Pětatřicátníků 14, 30614 Plzeň, tel..: {377634633, 377634639}, e-mail: {mayer@kte.zcu.cz, pkropik@kte.zcu.cz }

2 Calculation of compensation of two-poles parameters

Instantaneous values of phase voltages and line voltages of a balanced network are

$$u_1 = U\sin\omega t \qquad u_2 = U\sin(\omega t - 2\pi/3) \qquad u_2 = U\sin(\omega t + 2\pi/3) \qquad (2)$$

$$u_{12} = \sqrt{3}\,U\sin(\omega t + \pi/6) \qquad u_{23} = \sqrt{3}\,U\sin(\omega t - \pi/2) \qquad u_{12} = \sqrt{3}\,U\sin(\omega t + 5\pi/6) \qquad (3)$$

Currents in wye-connected compensation two-poles $R_i L_i C_i$ $(i = 1,2,3)$ are

$$i_{12} = I_{12}\sin(\omega t + \pi/6 - \psi_1) \qquad i_{23} = I_{23}\sin(\omega t - \pi/2 - \psi_2) \qquad i_{31} = I_{31}\sin(\omega t + 5\pi/6 - \psi_3) \qquad (4)$$

where $$I_{ij} = \sqrt{3}\,U\left[R_i + (\omega L_i - 1/\omega C_i\right]^{-\frac{1}{2}} \qquad i,j = 1,2,3;\ i \neq j \qquad (5)$$

The reactance coil has inductance L_i , which is defined so that the two-pole has the chosen resonance frequency f_r , thus

$$L_i = \frac{1}{\omega_0^2\,C_i}, \qquad \text{where} \quad \omega_0 = 2\pi f_r \qquad (6)$$

Let its resistance be k-multiple of inductive reactance, thus

$$R_i = k\,\omega\,L_i = \frac{k\,\omega}{\omega_0^2\,C_i} \qquad (7)$$

Then phase angle $\psi_1 = \psi_2 = \psi_3 = \psi \in \langle 0, \pi/2 \rangle$, when

$$\tan\psi = \frac{1}{R_i}\left(\omega L_i - \frac{1}{\omega C_i}\right) = \frac{1}{k}\left(1 - \frac{\omega_0^2}{\omega^2}\right) \qquad (8)$$

It is possible to express equation (5) using equations (6) and (7) in the form

$$I_{ij} = \frac{C_i U_{ij}}{A} \qquad (9) \qquad \text{where} \quad A^2 = \frac{\omega^2}{k^2\,\omega_0^4} + \left(\frac{\omega}{\omega_0^2} - \frac{1}{\omega}\right)^2 \qquad (10)$$

instantaneous line-currents in eq. (1) are calculated from equations

$$i_{\ell 1} = i_1 + i_{12} - i_{31}, \quad i_{\ell 2} = i_2 + i_{23} - i_{12}, \quad i_{\ell 3} = i_3 + i_{31} - i_{23} \qquad (11)$$

So the optimization problem is formulated. The solutions are the parameters of compensation two-poles.

If the load is linear, unbalanced and of inductance character, it draws currents

$$i_1 = I_1\sin(\omega t - \varphi_1), \quad i_2 = I_2\sin(\omega t - \varphi_2 - 2\pi/3), \quad i_3 = I_3\sin(\omega t - \varphi_3 + 2\pi/3) \qquad \varphi_1, \varphi_2, \varphi_3 \geq 0 \qquad (12)$$

3 Numerical minimization of the objective function (1)

All above-mentioned formulas were implemented using programming language of computational system MATLAB and MATLAB Optimization Toolbox. At the beginning current amplitudes I_{12}, I_{23} and I_{31} had been computed – see equations (5). Constants definitions and auxiliary computations are not given here, as it is mentioned above. In the second step variables A_1, A_2 a A_3 were computed using equations (10), i.e. for example

```
A1=(0.1.^2).*((omg.^2)./(omg0.^4))+(((omg./(omg0^2))-(1./omg)).^2);
```

Computation followed with calculating of relevant amplitudes according to equation (5), i.e. for example

```
I12=C1.*(U12./sqrt(A1));
```

and currents in compensation two-poles using equation (4)

```
i12 = I12.*sin(omg.*t+(pi./6)-psi1);
i23 = I23.*sin(omg.*t-(pi./2)-psi2);
i31 = I31.*sin(omg.*t+(5.*pi./6)-psi3);
```

Before current in the load were calculated, we had computed following auxiliary variables, which represents final angles. We need to calculate these angles to make program code more transparent and we need to know it in the next part of computation.

```
ang_i1 = omg.*t-(pi./3);
ang_i2 = omg.*t+((-52.*(2.*pi./360))-((2./3).*pi));
ang_i3 = omg.*t+((-68.*(2.*pi./360))+((2./3).*pi));
```

Calculation of current in the load according to equation (12), i.e. for example

```
i1=I1.*sin(ang_i1);
```

The part of program code shown above generated a course of currents in case of linear load. In case of non-linear load this course must been adjusted, i.e. for example

```
i1 = ~((mod(ang_i1,pi) < angle_4_t) & (mod(ang_i1,pi) > 0)).*i1;
```

Some parts of the currents i_1, i_2 and i_3 courses had been levelled with the zero by this part of program code, according to value of variable *angle_4_t*. This method produced required course of currents. At the end of computation the courses of currents $i_{\ell 1}$, $i_{\ell 2}$ and $i_{\ell 3}$ were calculated with help of conditions (11)

```
il1=i1+i12-i31;        il2=i2+i23-i12;        il3=i3+i31-i23;
```

Final sum of squares of these currents was computed

```
y=(il1.^2)+(il2.^2)+(il3.^2);
```

The numerical integration was based on equation (1). A standard MATLAB functions *quad* and *quadl* can be used. These functions used recursive adaptive Simpson quadrature algorithm. Function quad(*fun*, *a*, *b*) approximates the integral of function *fun* from *a* to *b* within an error of 10^{-6}. Function *fun* accepts vector *x* and returns vector *y*. Using form quad(*fun*, *a*, *b*, *tol*) uses an absolute error tolerance *tol* instead of the default (10^{-6}). In our calculations it was needed to set this tolerance usually between 10^{-7} and 10^{-9} to reach an adequate accuracy of integration. For this reason we used function *quadl* instead of *quad*. The function *quadl* should be more efficient with high accuracies and smooth integrands.

Finally we used this function in the following form

```
quadl('fun',0,T,1e-8,[],C1,C2,C3) / T;
```

Additional arguments *C1*, *C2* and *C3* were passed directly to function *fun*(*t*,*C1*,*C2*,*C3*).

Result of this integration represents our objective function. To solve optimization problem, we applied standard MATLAB functions *fminsearch*, *fminunc* and *fmincon* included in MATLAB Optimization Toolbox.

Function *fminsearch* is generally referred to as unconstrained non-linear optimization. We used it in form

```
options = optimset('fminsearch');
options.TolFun=1e-15; options.TolX=1e-15; options.MaxFunEvals=1000;
```

```
[min,fval,exitflag,output]=fminsearch(@objective_f,input,options);
```

The variable *options* represent set of initial parameters of this function. Useful parameters are

Display – Level of display. `'off'` displays no output; `'iter'` displays output at each iteration; `'final'` displays just the final output; `'notify'` (default) displays output only if the function does not converge.
MaxFunEvals – Maximum number of function evaluations allowed.
MaxIter – Maximum number of iterations allowed.
TolFun – Termination tolerance on the function value.
TolX – Termination tolerance on *x*.

Function *fminsearch* uses algorithm based on the Nelder-Mead simplex direct search method. This is a method that does not use numerical or analytic gradients as in *fminunc* or *fmincon* (see below). When the solving problem is highly discontinuous, *fminsearch* may be more robust than *fminunc*.

Function *fminunc* is generally referred to as unconstrained non-linear optimization of multivariable function. We used it in form

```
options=optimset('fminunc'); options.TolFun=1e-15;
options.TolX=1e-15; options.MaxFunEvals=1200; options.GradObj='on';
[min,fval,exitflg,output,grad,hessian]=fminunc(@objective_f,input,options);
```

The variable *options* represents set of initial parameters of this function as above. Many parameters are same as parameters of the function *fminsearch*. We used special parameter GradObj sets '*on*'

GradObj – gradient for the objective function. User in objective function defines it. The gradient must be provided to use the large-scale method. We used it as an optional parameter for the medium-scale method.

Function *fminunc* uses algorithm based on the BFGS (Broyden, Fletcher, Goldfarb, Shanno) Quasi-Newton method with a mixed quadratic and cubic line search procedure (in case of medium-scale optimization). The DFP (Davidon, Fletcher, Powell) formula is used to approximate the inverse Hessian matrix. In case of Large-Scale Optimization an algorithm subspace trust region method based on the interior-reflective Newton method is used. Each iteration involves the approximate solution of a large linear system using the method of preconditioned conjugate gradients (PCG).

When it is needed to eliminate improper values of variables (e.g. negative values of capacitance) we implement function *fmincon*. This function finds a minimum of a constrained non-linear multivariable function. We used it in form

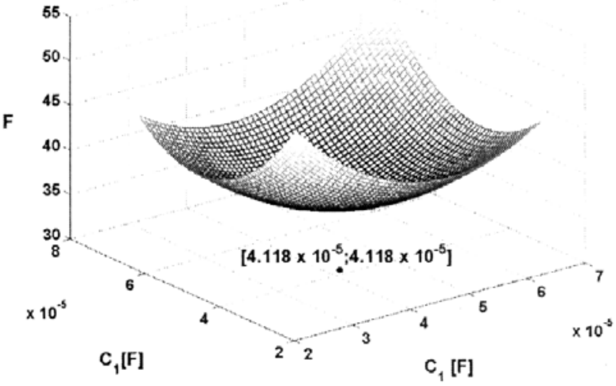

Fig. 2. Objective function for problem 4.1.

```
mat_A=[-1,0,0;0,-1,0;0,0,-1]; vec_b=[0;0;0];
options=optimset('fmincon');options.TolFun=1e-15;options.TolX=1e-20;
options.TolCon=1e-15; options.MaxFunEvals=800; options.GradObj='on';
[min,fval,exitflag,output,lambda_v,grad_v,hessian_v]=fmincon(@criteria_f,
input,mat_A,vec_b,[],[],[],[],[],options);
```

Variable *mat_A* represents the matrix **A** of the coefficients of the linear inequality constraints and *vec_b* represents corresponding right side vector **b** (i.e. $\mathbf{Ax} \le \mathbf{b}$).

Function *fmincon* uses algorithm based on the Sequential Quadratic Programming (SQP) method (in case of medium-scale optimization). Quadratic Programming (QP) subproblem is solved at each iteration. An estimate of the Hessian of the Lagrangian is updated at each iteration using the BFGS formula (see *fminunc* above). A line search is performed using a merit function. The QP subproblem is solved using an active set strategy.

4 Numerical problems

4.1 Linear balanced load

For comparison of the results obtained in numerical calculation with analytical solution we use the following simple problem. Line voltages of balanced network are $\sqrt{3}\,U = 380\,\text{V}$, $\omega = 2\pi f = 100\pi$. The load is linear (i. e. it draws harmonic currents) and symmetrical. Drawn currents are expressed in equation (12), where

$$I_1 = I_2 = I_3 = I = 10\text{A},\ \varphi_1 = \varphi_2 = \varphi_3 = \varphi = 60^\circ.$$

Compensation is done using static condenser in wye connection. Currents i_{12}, i_{23}, i_{31} acc. Eq. (4), where acc. Eq. (9)

$$I_{ij} = 3{,}8.10^4\,\pi\, C_i \tag{13}$$

Substituting for i_{t1}, i_{t2}, i_{t3} in eq. (1) and solving optimization task we get

$$C_1 = C_2 = C_3 = 4{,}188.10^{-5}\,\text{F},\ \text{for } F_{\min} = 37{,}5 \tag{14}$$

For judging the environment of the found minimum of functional F according eq. (1) there is shown in Fig. 2 function $F = F(C_1, C_2)$ for $C_3 = 4{,}188.10^{-5}\,\text{F}$ in 3D representation. Optimization was done using three above-mentioned methods and the same results were achieved with the difference that function *fmincon* and *fminunc* showed higher accuracy of the result, but only in higher orders, which does not have any practical meaning.

It is possible to solve this symmetrical problem also analytically (see e.g. [1]).

Obviously $C_1 = C_2 = C_3 = C$, where
$$C = \frac{I \sin\varphi}{3\,\omega U} = 4{,}188.10^{-5}\ \text{F} \tag{15}$$

4.2 Linear unbalanced load

Network is the same as in the previous example: $\sqrt{3}\,U = 380\,\text{V}$, $\omega = 100\pi$. The load is linear, unbalanced; drawn currents are expressed by equations (12), where

$$I_1 = 10\,\text{A},\quad I_1 = 8\,\text{A},\quad I_1 = 12\,\text{A},\quad \varphi_1 = 60^\circ,\quad \varphi_1 = 52^\circ,\quad \varphi_1 = 68^\circ \tag{16}$$

Compensation is done using two-poles $R_i\,L_i\,C_i$ $(i = 1,2,3)$ for the following values:

$$f_r = 189\,\text{Hz},\ \omega_0 = 2\pi 189\,\text{s}^{-1},\ k = 0{,}1. \tag{17}$$

Through minimization of the functional (1) we obtain:

$$C_1 = 2{,}826.10^{-5}\,\text{F},\quad C_2 = 4{,}015.10^{-5}\,\text{F}$$
$$C_3 = 4{,}846.10^{-5}\,\text{F}$$

According eq. (6) is

$$L_1 = 0{,}1577\ \text{H},\quad L_2 = 0{,}1110\ \text{H}$$
$$L_3 = 0{,}0919\ \text{H}$$

And according eq. (7) is

$$R_1 = 4{,}954\ \Omega,\quad R_2 = 3{,}486\ \Omega$$
$$R_3 = 2{,}8889\ \Omega$$

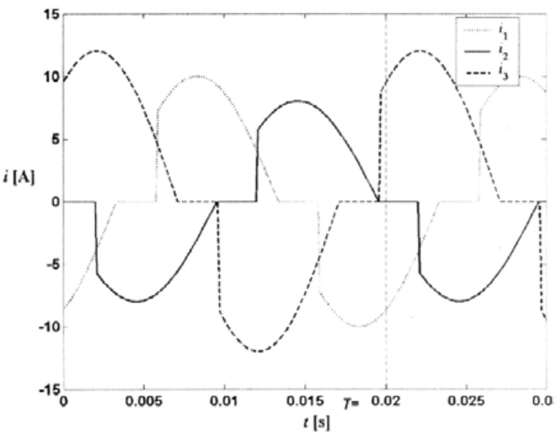

Fig. 3. Time-dependency of the current of the load.

This case was solved again using all three methods. The best effect was achieved using function *fminsearch*. In case of function *fminunc* and *fmincon* the calculation got much longer and taking into consideration the character of the objective function course some numerical instabilities occurred as well.

4.3 Nonlinear unbalanced load

The network is the same as in the previous problems. Instantaneous values of currents drawn by the load are (Fig. 3)

$$i_i = \begin{cases} 0 & \text{for } 0 < t < \alpha \\ I_i \sin(\omega t - \psi_i) & \text{for } \alpha < t < 2\pi,\ i = 1,2,3 \end{cases}$$

where $\psi_1 = -\varphi_1$, $\psi_2 = -\varphi_2 - 2\pi/3$, $\psi_3 = -\varphi_3 + 2\pi/3$

Calculation is done for $\alpha = 45^\circ$ and for values I_1, I_2, I_3, φ_1, φ_2, φ_3 according eq. (16).

Compensation is done using two-poles $R_i\,L_i\,C_i$ $(i = 1,2,3)$, for which eq. (17) is valid. Minimazing functional (1) we get

$$C_1 = 2{,}841.10^{-5}\,\text{F},\quad C_2 = 4{,}184.10^{-5}\,\text{F},\quad C_3 = 4{,}645.10^{-5}\,\text{F}$$
$$L_1 = 0{,}1568\ \text{H},\quad L_2 = 0{,}1065\ \text{H},\quad L_3 = 0{,}0959\ \text{H}$$
$$R_1 = 4{,}926\ \Omega,\quad R_2 = 3{,}345\ \Omega,\quad R_3 = 3{,}013\ \Omega$$

Application of the three above-mentioned methods had the same effect as the previous cases.

Acknowledgement. Financial support of the Grant Agency of the Czech Republic (project No. 102/01/1401) is highly acknowledged.

References

[1] Mayer D.: Introduction to the Electrical Circuit Theory. SNTL/ALFA, Praha 1981. (In Czech.)
[2] Optimization Toolbox User's Guide, The MathWorks, Inc., Natick, U.S.A., 2002.