

CORREÇÃO AUTOMATIZADA DO FATOR DE POTÊNCIA E ATENUAÇÃO DAS COMPONENTES HARMÔNICAS

Diego Garcia Borges¹

Marcelo Semensato²

¹Instituto Federal de Educação, Ciência e Tecnologia/Campus Jataí/Engenharia Elétrica/ PIBITI-IFG,
diegoifg@ymail.com

²Instituto Federal de Educação, Ciência e Tecnologia/Campus Jataí/Área de Indústria/
semensato@ymail.com

Resumo

O projeto consiste em um controlador automático para a correção do fator de potência e atenuação de Dht's para um conjunto de cargas genéricas de até 30 A. Esse controle é feito utilizando um microcontrolador da família PIC, sendo o modelo PIC 16F877A.

O microcontrolador analisa as formas de onda da tensão e corrente de entrada e seleciona o banco de capacitor adequado para a correção do fator de potência. Mas como as indústrias possuem diversas cargas não lineares existem várias componentes harmônicas no sistema que impedem esta correção precisa do fator potência. Para uma correção mais eficaz é preciso selecionar filtros passivos para filtrar as harmônicas de maior amplitude. O microcontrolador é responsável por efetuar os cálculos do espectro harmônico e selecionar o melhor filtro passivo.

Palavras-chave:

Fator de potência, distorções harmônicas, microcontrolador, banco de capacitores.

INTRODUÇÃO

A qualidade de energia vem sendo discutida com maior ênfase a cada novo ano, com o objetivo de reduzir anomalias na rede. Dentre elas, o fator de potência e as distorções harmônicas consistem nos assuntos que mais recebem atenção das indústrias e concessionárias energéticas, e a correção dessas anomalias economizam custos, e o controle proposto neste trabalho, procura tornar essa economia de energia mais evidente.

Acontece, que esse novo tipo de carga, traz consigo uma característica de seus componentes, que é a não-linearidade entre tensão e corrente, provocando com isso, uma distorção na forma de sinal (tensão ou/e corrente).

Essa nova topologia de cargas no sistema energético, altera a qualidade de energia, e provoca perturbações no sistema como um todo, ocasionando baixa eficiência.

O projeto, através de seu algoritmo, analisa as distorções do sinal elétrico, e o microcontrolador realiza o papel de acionar ou não, um filtro passivo passa-baixa na rede de energia, para atenuar as distorções harmônicas e posteriormente realizar os cálculos para a correção do fator de potência.

► MATERIAIS E MÉTODOS

• MATERIAIS

- ✓ Software Proteus Design Suite version 8.2 – Labcenter Eletronics
- ✓ Software PIC C Compiler - CCS Compilers
- ✓ Software MPLAB® XC Compilers - Microchip

• MÉTODOS

O projeto em questão é continuidade e aprimoramento da pesquisa anterior para correção do fator de potência. Na pesquisa atual, o foco foi a correção de distorções harmônicas. As harmônicas provocam uma distorção na onda senoidal transmitida na rede de distribuição, visto que elas são compostas por ondas múltiplas da frequência fundamental do sistema, no caso, 60hz.

O indicador essencial é o DHT (Distorção Harmônica Total) que traduz através de uma só grandeza a deformação da onda em tensão ou em corrente.

Quanto mais significativa a presença de harmônicos na rede, mais difícil se torna a correção do fator de potência, uma vez que o baixo fator de potência deixa de ser causado somente pela presença de cargas indutivas, mais agora ocorrendo a existência adicional de potência reativa necessária para sustentar a distorção causada no sinal.

A fase atual da pesquisa se encontra nos testes do programa em C para correção do fator de potência e adequação da FFT (transformada rápida de Fourier) para decomposição do espectro harmônico presente.

O microcontrolador é o responsável pelo cálculo da FFT, e a partir dos cálculos é possível determinar a harmônica de maior amplitude, enviando um sinal de comando para a saída correspondente ao filtro passivo que atenua a harmônica.

As distorções harmônicas produzidas por equipamentos elétricos dependem das suas características construtivas, das suas condições operacionais e das interações entre estes e o sistema elétrico. Portanto, tanto os harmônicos gerados por equipamentos, como os harmônicos presentes no sistema, apresentam variações de amplitude, fase e frequência ao longo do tempo.

A análise harmônica consiste no processo de obtenção da amplitude e fase de cada componente múltipla da onda fundamental, por decomposição do sinal original.

Através de conversores analógico-digitais os sinais podem ser discretizados, passando a ser representados por séries finitas de amostras espaçadas no tempo. Para estes sinais discretos, usa-se a Transformada Discreta de Fourier (DFT). Se o sinal discreto a ser transformado é uma série de amostras periódicas e igualmente espaçadas, a DFT pode ser calculada através de algoritmos rápidos e eficientes, conhecidos como Transformada Rápida de Fourier (FFT), que formam a base da análise deste projeto.

A transformada de Fourier é um método matemático que converte o sinal do domínio do tempo para o domínio da frequência, explicitando assim suas várias componentes harmônicas.

Segue a seguir, em apêndice 1, a atual fase em que a pesquisa se encontra em relação ao desenvolvimento do algoritmo de controle. Relembrando que a transformada rápida de Fourier (FFT) estava a ser desenvolvida nos próximos meses.

► REFERÊNCIAS BIBLIOGRÁFICAS

- [1] (EPCOS(3), 2007) EPCOS AG. Power Quality Solutions – Overview 2008: Power Factor Correction. Germany: EPCOS, 2007. 12 p.
- [2] (COTRIM(1), 2008) COTRIM, Ademaro A. M. B.; MORENO, Hilton (revisão e atualização técnica); GRIMONI, José Aquiles Baesso (revisão e atualização técnica). Compensação da energia reativa; STAROSTA, José (autoria). In: Instalações Elétricas. 5ª edição. São Paulo: Prentice Hall, 2008. cap. 14, pp. 419-434.
- [3] Costa, F. F., Naidu, S. R. e Costa, E. G. (2004). Estudo comparativo entre algoritmos recursivos para rastreamento de harmônicos, XV Congresso Brasileiro de Automática (XV CBA), Gramado – Rs, 2004.
- [4] ANDRADE, Adriano O. et al. Técnicas de Janelamento de Sinais. Universidade Federal de Uberlândia. In: SEMINÁRIO DOS ESTUDANTES DE ENGENHARIA ELÉTRICA DA UFU, 3, 2000, Uberlândia, MG, Brasil, 2000.
Disponível em:
< <http://www.aoandrade.eletrica.ufu.br/Documents/T%E9cnicas%20de%20Janelamento%20de%20Sinais.pdf> >. Acesso em: 10 nov. 2014.
- [5] COOLEY, J.W., and TUKEY, J.W. An algorithm for the Machine Calculation of Complex Fourier Series. Mathematics of Computation, abr. 1965. Disponível em:
< www.utdallas.edu/~cpb021000/EE%204361/Great%20DSP%20Papers/Cooley%20FFT.pdf >. Acesso em: 16 Jan 2015.
- [6] SAMPAIO, Rubens; CATALDO, Edson; BRANDÃO, Alexandre de S. Análise e Processamento de Sinais (Notas em Matemática Aplicada). Rio de Janeiro: SBMAC, 2006.
Disponível em:
< <http://www.sbmac.org.br/boletim/arquivos2006/minicurso-22.pdf> >.
Acesso em: 15 Jan. 2015.

APÊNDICE 1 –CÓDIGO DE CONTROLE

```
1: /*****
2:                                     CORRECAO AUTOMATICA DO FATOR DE POTENCIA E
                                     ATENUAÇÃO DAS COMPONENTES HARMONICAS
3:
4: Pesquisador: DIEGO GARCIA BORGES
5: Orientador: MsC. MARCELO SEMENSATO
6:                                     IFG/JATAÍ
7: *****/
8:
9: # include < 16f877a. h >      // identifica microcontrolador alvo
10: # device adc= 1 0
11: # use delay(clock= 20000000)
12: # include < MATH.H >
13: # include < STDIO.H >
14: # include < LCD.c >
15: # fuses xt, noprotect, put, brownout, nolvp, nocpd, nowrt // configuração dos fusíveis
16: # use fast_io(a)
17: # use fast_io(b)
18: # use fast_io(b)
19:
20: void main( )    // função principal
21: {
22:     long int x;
23:     int resistor= 10, parada= 0 , fp= 0 , capa= 0 , capb= 0 , capc= 0 ;
24:     float tensao= 0 , corrente= 0 , potencia= 0 , tensaoc= 0 , tensaos= 0 , correntes= 0 , total= 0 ,
    reativa=0, tensao_anterior=0, corrente_anterior=0;
25:     set_tris_a( 0 b00000011 ) ;
26:     set_tris_b( 0 b000000000 ) ;
27:     set_tris_c( 0 b000000000 ) ;
28:     output_b( 0 ) ;
29:     output_c( 0 ) ;
30:     output_d( 0 ) ;
31:     lcd_init( ) ;
32:     lcd_putc( " Controlador FP e Atenuador de DHT's" ) ;
33:     delay_ms(5000) ;
34:     setup_adc(ADC_CLOCK_INTERNAL) ;
35:     setup_adc_ports( A LL_ANALOG ) ;
36:     while( 1 )
37:     {
38:         for ( x = 0 ; x< 4 320; x+ +)
39:         {
40:             set_adc_channel( 0 ) ;
41:             delay_us(20);
42:             tensao=read_adc( ) ;
43:
44:             set_adc_channel( 1 ) ;
45:             delay_us( 20 ) ;
```

```
46: corrente= read_adc( );
47:
48: tensao= (tensao- 512) ;
49: tensaoc= ((5 * t ensao) /189.44) ;
50: tensao= (tensaoc* resistor) ;
51: tensaos= (tensaos+ (tensao* tensao) );
52:
53: corrente= (corrente- 512) ;
54: corrente= ((5 * corrente) /67.584) ;
55: correntes= (correntes+ (corrente* corrente) );
56:
57: potencia= (potencia+ (tensao* corrente) );
58:
59:
60: if ( tensao< 0 . 0 1 & & tensao> -0.0 1 & & parada= =0 & & ( tensao- t ensao_anterior) >0 &
& x> 10)
61: {
62:   parada= 1 ;
63: }
64: if ( corrente< 0 . 0 1 & & corrente> -0.0 1 & & parada= =1 & & ( corrente- corrente_anterior) >0 )
65: {
66:   fp= 1 ; parada= 2 ;
67: }
68: if ( corrente< 0 . 0 1 & & corrente> -0.0 1 & & parada= =1 & & ( corrente- corrente_anterior) <0 )
69: {
70:   fp= 2 ; parada= 2 ;
71: }
72: tensao_anterior= tensao;
73: corrente_anterior= c orrente;
74: }
75: tensao= sqrt( tensaos/ 4320) ;
76: corrente= sqrt( correntes/ 4320) ;
77: potencia= (potencia/ 4320) ;
78: total= (tensao* corrente) ;
79: reativa= (t otal* total) -(potencia* potencia) ;
80: reativa= (sqrt( reativa) );
81: printf( 1 cd_putc, " \fReativa:\n %1.2f", reativa) ;
82: delay_ms( 5000) ;
83: tensaos= 0 ; correntes= 0 ; potencia= 0 ; parada= 0 ;
84:
85: if ( f p= =1 )
86: {
87:   if ( reativa> 1 000 & & capa= =0 )
88:   {
89:     output_high( pin_b0) ;
90:     delay_ms( 200) ;
91:     capa= 1 ;
92:     reativa= reativa- 1 500;
93:   }
```

```
94: if ( reativa > 1000 & & capb == 0 )
95: {
96: output_high( pin_b1 );
97: delay_ms( 200 );
98: capb = 1 ;
99: reativa = reativa - 1500;
100: }
101: if ( reativa > 1 000 & & capc == 0 )
102: {
103: output_high( pin_b2 );
104: delay_ms( 200 );
105: capc = 1 ;
106: reativa = reativa - 1500;
107: }
108: }
109: if ( fp == 2 )
110: {
111: if ( reativa > 1 000 & & capc == 1 )
112: {
113: output_low( pin_b2 );
114: delay_ms( 200 );
115: capc = 0 ;
116: reativa = reativa - 1 500;
117: }
118: if ( reativa > 1000 & & capb == 1 )
119: {
120: output_low( pin_b1 );
121: delay_ms( 200 );
122: capb = 0 ;
123: reativa = reativa - 1 500;
124: }
125: if ( reativa > 1000 & & capa == 1 )
126: {
127: output_low( pin_b0 );
128: delay_ms( 200 );
129: capa = 0 ;
130: reativa = reativa - 1500;
131: }
132: }
133: }
```