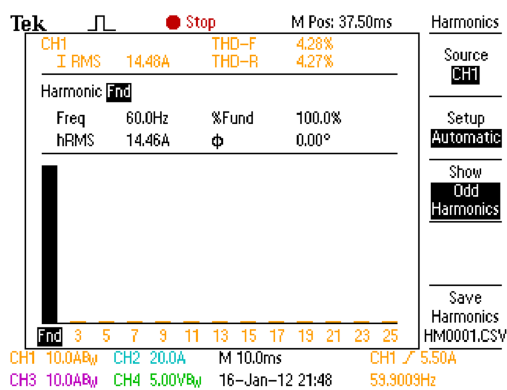
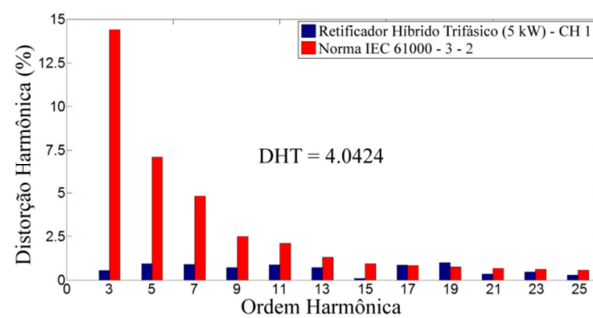


Como pode ser observado pelas Fig. 5.13 a 5.15, a análise do DHT_1 para cada fase do sistema CA fornece valores ligeiramente diferentes, mas nunca superiores a 5%, significando que tal DHT é adequado em comparação à norma IEC 61000-3-2 e está em conformidade com o que foi exposto na modelagem matemática apresentada no Capítulo 3.

A análise do espectro de frequência das correntes de entrada foi realizada utilizando o *Tektronix Power Analysis Application Software (TPS2PWR1)* em conjunto com o *software Matlab®*, a fim de proporcionar uma comparação quanto aos limites harmônicos impostos pela norma IEC 61000-3-2. A referida análise foi possível utilizando-se o vetor de pontos de cada sinal obtido com o osciloscópio e calculando-se seu DHT através do *software Matlab®* considerando até a 50ª ordem.

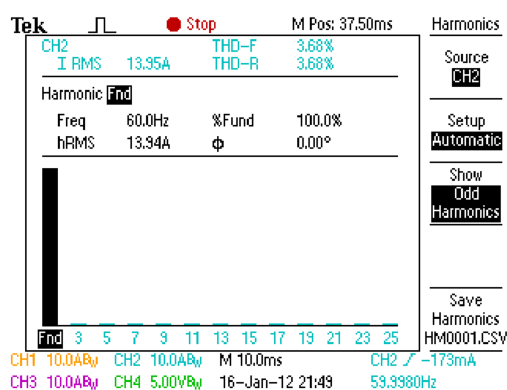


(a)

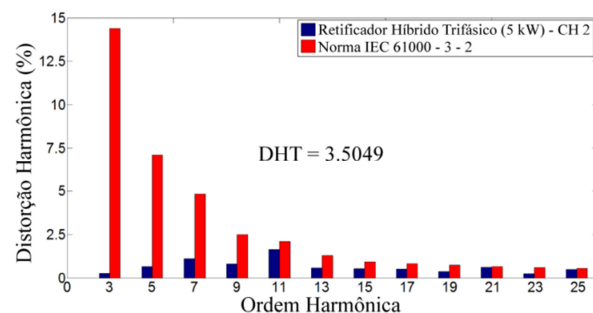


(b)

Fig. 5.13 – Espectro harmônico da corrente $i_{a(in)}$ determinado pelos *software* (a) *TPS2PWR1* e (b) *Matlab®*.



(a)



(b)

Fig. 5.14 – Espectro harmônico da corrente $i_{b(in)}$ determinado pelos *software* (a) *TPS2PWR1* e (b) *Matlab®*.

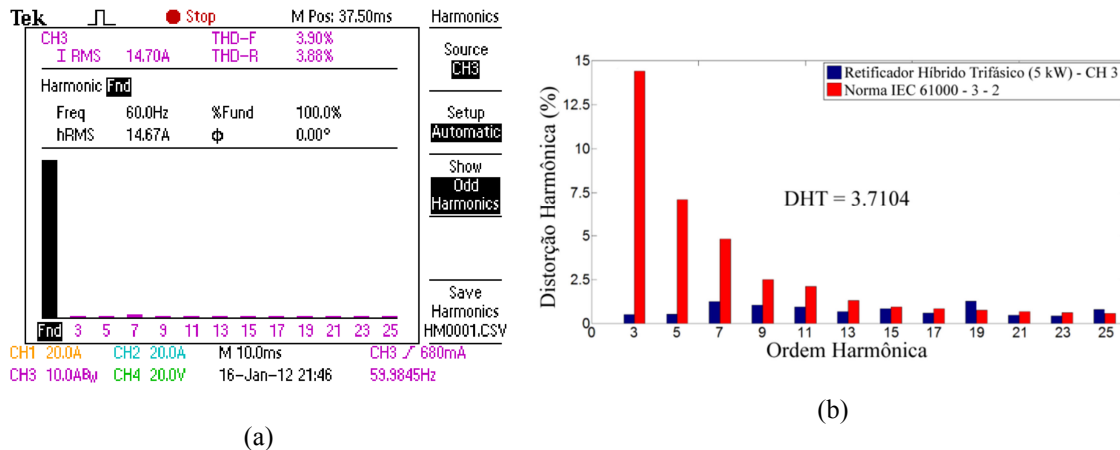


Fig. 5.15 – Espectro harmônico da corrente $i_{c(in)}$ determinado pelos *software* (a) *TPS2PWR1* e (b) *Matlab*[®].

O valor final da DHT de $i_{b(in)}$ revelou-se o mais reduzido, sendo que essa pequena diferença entre as DHTs de $i_{a(in)}$, $i_{b(in)}$, e $i_{c(in)}$ visualizada através do *software TPS2PWR1* também foi confirmada pelo *Matlab*[®], uma vez que foi considerada a mesma ordem harmônica (até a 50ª ordem). Em comparação com a análise de simulação apresentada no Capítulo 3, é importante salientar que a principal diferença entre os valores de DHT_1 pode ser atribuída à indutância de linha do sistema de fornecimento de energia, a qual não foi considerada na solução do modelo matemático. Portanto, uma vez que a reatância indutiva da linha funciona como filtros que contribuem para a atenuação tanto de *spikes* como de *ripples* de alta frequência presentes nas correntes, menores valores de DHT_1 são passíveis de serem alcançados em testes experimentais.

Na Fig. 5.16 são ilustradas as formas de onda da tensão e corrente de carga e a parcela de corrente fornecida através do Ret-1 (após a ponte retificadora). Através do valor médio da corrente no lado CC de Ret-1 ($I_{Ret-1(med)} = 10,1$ A), da tensão e da corrente na carga ($V_{0(med)} = 290$ V e $I_{0(med)} = 17$ A), da potência ativa média total fornecida à carga ($P_{0(med)} = 4,84$ kW) é possível calcular as parcelas de potência processadas por cada grupo de retificador. Pode-se verificar que o Ret-1 processa uma potência média de 2,929 kW, o que corresponde a 60,516% da potência ativa média total. O conversores chaveados, por sua vez, processam em torno de 1,911 kW, correspondendo a 39,484% da potência ativa média total. Portanto, é possível confirmar que a contribuição de potência de cada grupo retificador obtida experimentalmente está de acordo com os resultados de simulação demonstrados no Capítulo 3, provando a eficácia da análise teórica anteriormente descrita.

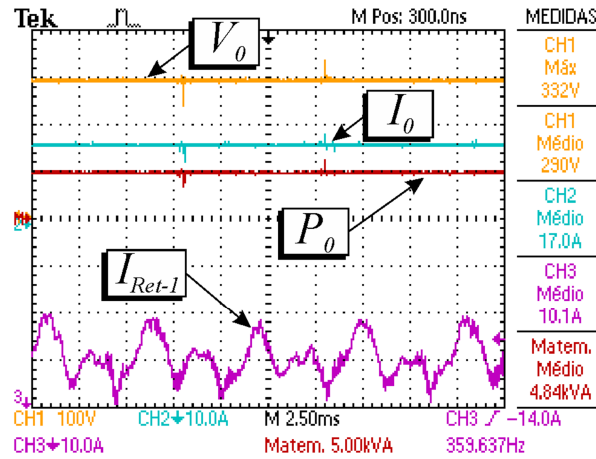


Fig. 5.16 – Tensão na carga (V_o), corrente na carga (I_o), corrente de Ret-1, após a ponte retificadora (I_{Ret-1}), e potência ativa total processada pelo retificador híbrido trifásico (P_o).

Utilizando o medidor digital de potência *Yokogawa*[®] WT230 como mostrado na Fig. 5.17, foi possível realizar as medidas das potências processadas pelo protótipo.

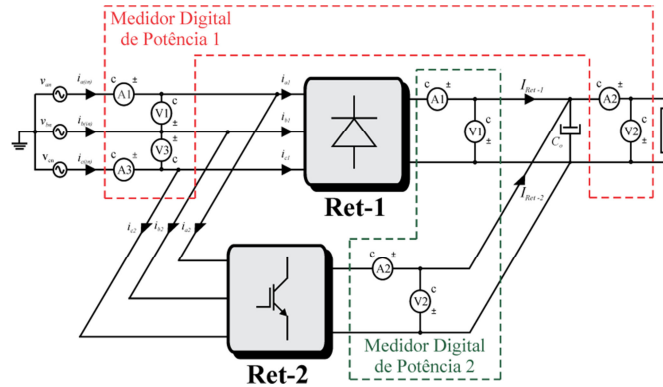
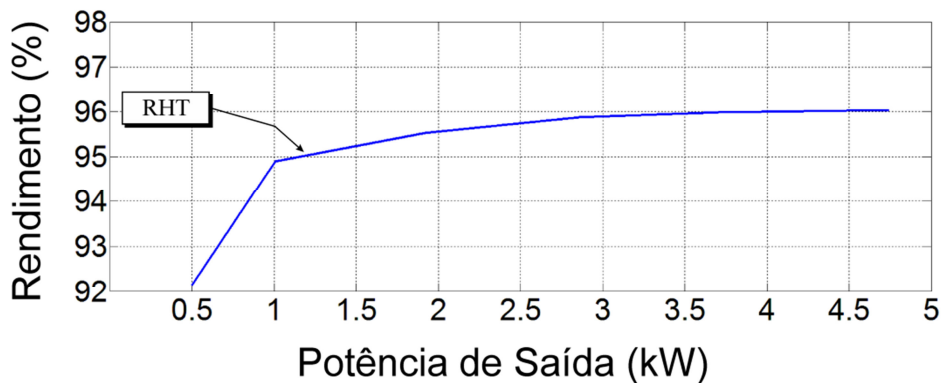


Fig. 5.17 – Arranjo para medição e determinação do rendimento global da estrutura e do processamento de potência de cada grupo retificador usando equipamentos *Yokogawa* WT230[®].

A curva de rendimento em função da potência de saída obtida através de ensaios sob diferentes condições de carga está ilustrada na Fig. 5.18 (a). Pela Fig. 5.18 observa-se que um rendimento de aproximadamente 96% foi alcançado para uma ampla faixa de variação de carga.



(a)

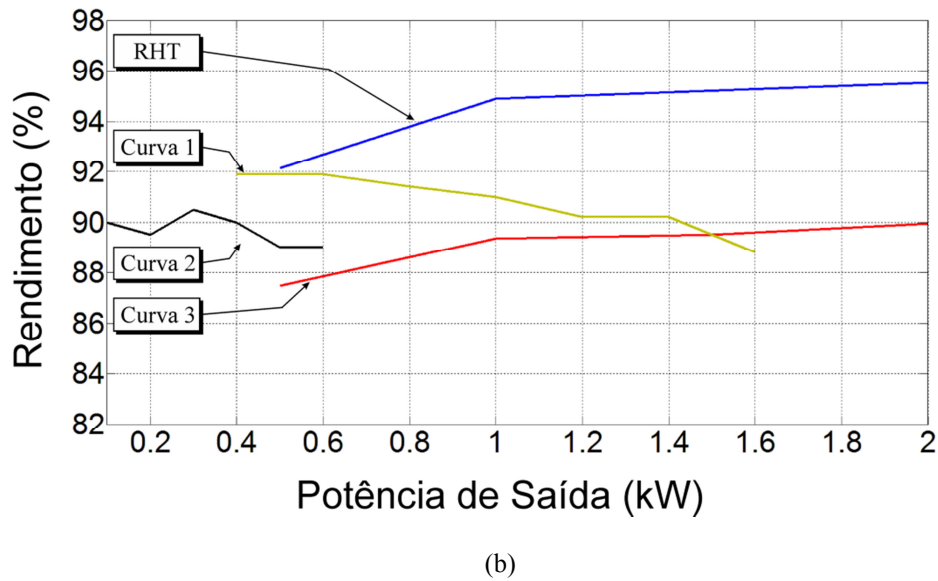


Fig. 5.18 – (a) Curva de rendimento do RHT em função da potência de saída e (b) comparação entre o rendimento do RHT em relação aos conversores PWM apresentados em [11] (curva 1), [12] (curva 2) e [28] (curva 3).

Comparou-se, conforme demonstrado na Fig. 5.18 (b), o resultado de rendimento do RHT com o de duas diferentes estruturas topológicas de retificadores PWM encontrados na literatura. O primeiro é uma aplicação de um conversor BOOST monofásico para correção de fator de potência utilizado em aplicações trifásicas [11] (curva 1) e o segundo é um retificador trifásico de alto fator de potência utilizando três conversores SEPIC isolados [12] (curva 2). Observa-se que, apesar do bom desempenho em relação à imposição da corrente senoidal de entrada e de controle da tensão no barramento CC, estes retificadores PWM trifásicos usuais são limitados a aplicações de média potência (abaixo de 1 kW) devido à drástica diminuição do rendimento em níveis de potência elevados (acima de 1 kW). Dessa forma, pode-se concluir que, apesar do fato de a tensão no barramento CC não ser regulada, a estrutura do retificador híbrido trifásico proposto oferece maior rendimento em aplicações de alta potência, que são difíceis de serem alcançados com retificadores PWM usuais. Assim, esta característica do RHT pode ser muito atrativa para muitas aplicações em que um segundo estágio de conversão CC-CC ou CC-CA seja necessário para alimentar outros equipamentos eletrônicos e, conseqüentemente, uma tensão no barramento CC fixa não seja obrigatória.

É apresentada também na Fig. 5.18 (b) uma comparação com o resultado de rendimento obtido através da estrutura do RHT operando somente com os três conversores SEPIC atuando com controle de fator de potência, mitigação de conteúdo

harmônico das correntes de linha de entrada e controle da tensão de saída em 300 V [28] (curva 3). Percebe-se que houve um ganho de mais de 5% no rendimento quando da utilização do RHT com o estágio não controlado em detrimento do controle da tensão de saída, promovendo consequentemente uma maior robustez da estrutura, o que torna o RHT bastante confiável em termos de eficiência e bastante atrativo para aplicações em que há a necessidade de um estágio PFC para o fornecimento de um barramento CC para a alimentação de outros conversores CC-CC ou outras cargas eletrônicas.

Utilizando o osciloscópio de quatro canais isolados *Tektronix*[®] *TPS 2024B* para a medição do fator de potência e o *software Matlab*[®] para determinação da DHT_I das correntes de linha de entrada para diferentes cargas, foi possível construir, através de interpolação dos valores obtidos, as curvas da distorção harmônica total de corrente e do fator de potência em função da potência de saída, conforme visualizado na Fig. 5.19. Pode-se observar que para cargas mais baixas, o nível de DHT das correntes de linha de entrada é maior, o que determina um baixo fator de potência. Para o RHT operando com carga de 5 kW, o fator de potência geral da estrutura demonstrou-se próximo de 0,98 em consequência da baixa distorção harmônica total de corrente. Com isto comprova-se que o fator de potência está intimamente ligado ao nível de distorção harmônica das correntes de linha de entrada, pois, quanto menor o DHT_I , menor é o valor eficaz das correntes e consequentemente menor é a potência aparente drenada da rede. Se a potência aparente drenada da rede diminui para uma mesma quantidade de potência ativa processada, tem-se um aumento considerável do fator de potência.

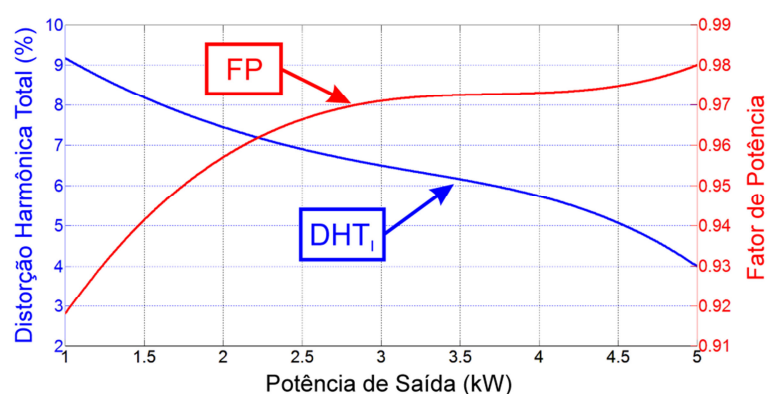


Fig. 5.19 – Curvas de distorção harmônica total de corrente e fator de potência em função da potência de saída.

Para verificar o desempenho dinâmico do retificador híbrido trifásico, aplicou-se um degrau de carga de 2,5 kW para 5 kW e o resultado sobre o comportamento das correntes de linha de entrada e da tensão no barramento CC durante o transitório é

apresentado na Fig. 5.20 (a). Pela Fig. 5.20 (b), tomando a fase A como exemplo, é possível notar que, durante o período transitório, as magnitudes da corrente imposta pelo SEPIC e da corrente drenada pelo retificador trifásico não controlado de seis pulsos aumentam instantaneamente. Assim, a divisão de potência ativa processada entre as unidades retificadoras é mantida inalterada, mostrando a eficácia da estratégia de controle proposta.

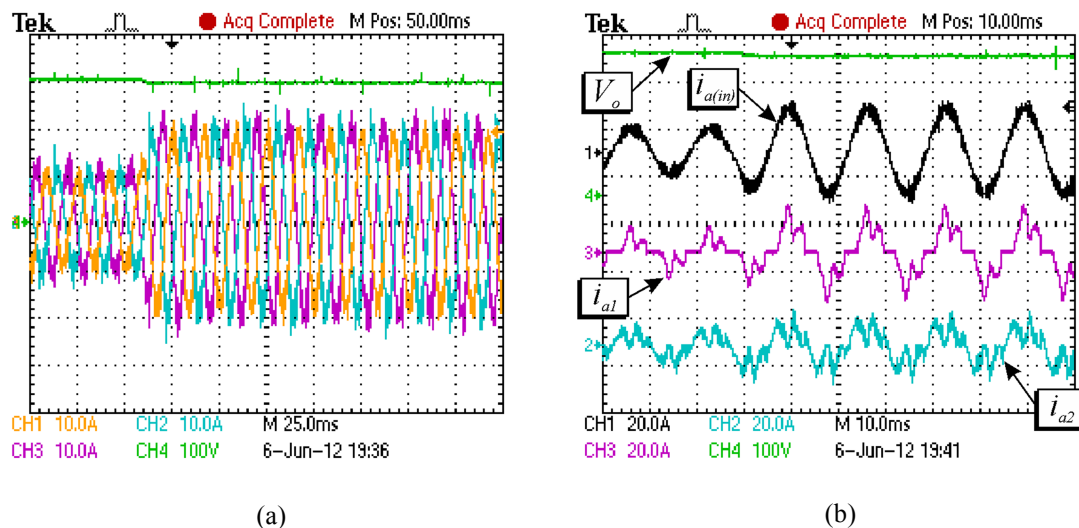


Fig. 5.20 – Resposta dinâmica durante um degrau de carga de 2,5 kW para 5 kW: (a) tensão no barramento CC e correntes de linha de entrada $i_{a(in)}$, $i_{b(in)}$ e $i_{c(in)}$; (b) tensão no barramento CC e correntes $i_{a(in)}$, i_{a1} e i_{a2} .

Finalmente, um inversor de frequência utilizado para o acionamento de um motor de indução de 2 CV foi conectado ao barramento CC fornecido pelo RHT. Foi utilizado um freio eletromagnético para estabelecer um conjugado mecânico no motor. Nas Tab. 5.14 e 5.15 são apresentados as características técnicas do motor de indução e do inversor, respectivamente.

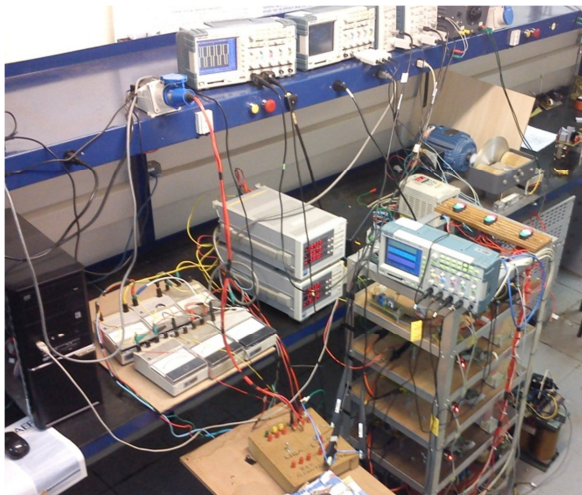
Tab. 5.14 – Motor de indução W21 da WEG®.

	Tipo	Gaiola
	Potência	1,5 kW (2,0 CV)
	Rotação nominal	1740 RPM
	Fator de serviço	1,15
	Grau de proteção	IP55
	Categoria	N
	Tensão de linha Δ/Y	220/380 V
	Corrente de linha Δ/Y	6,07/3,51 A
	Corrente de partida	6,4 In
	Fator de potência	0,78
	Rendimento	83,1 %
	Frequência	60 Hz
	Máxima temperatura ambiente	40 °C

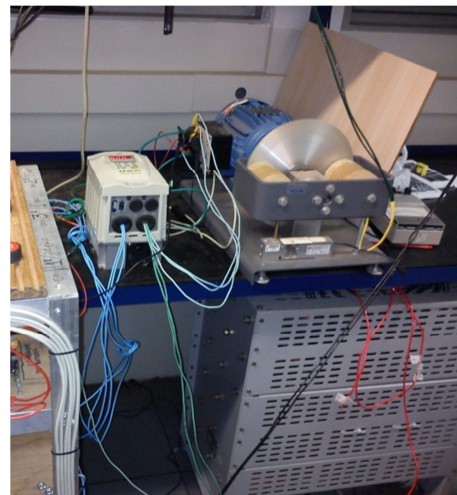
Tab. 5.15 – Inversor de frequência *CFW-09* da *WEG*®.

	Corrente nominal de saída	7 A
	Tensão	220-230 V
	Potência	2,7 kVA
	Tipo de carga	Torque constante e torque variável
	Corrente de saída máxima	10,5 A
	Corrente nominal de entrada	8,4/19 A
	Frequência de chaveamento nominal	5 kHz
	Motor máximo	2 CV/1,5 kW
	Potência dissipada nominal	80 W

A Fig. 5.21 ilustra a bancada de ensaio do RHT conectado ao sistema composto pelo inversor de frequência, o motor de indução e o freio eletromagnético.



(a)



(b)

Fig. 5.21 – (a) Bancada com equipamentos de medição e RHT conectado ao (b) sistema composto pelo inversor de frequência, motor e freio eletromagnético.

Na Fig. 5.22 são demonstradas as formas de onda das correntes de linha de entrada ($i_{a(in)}$, $i_{b(in)}$ e $i_{c(in)}$), das correntes de linha no motor (i_{Ma} , i_{Mb} e i_{Mc}), das tensões de linha no motor (v_{Mab} , v_{Mbc} e v_{Mca}) e da tensão no barramento CC. A Fig. 5.23 ilustra a forma de onda das correntes de linha de entrada no início da atuação dos conversores SEPIC.

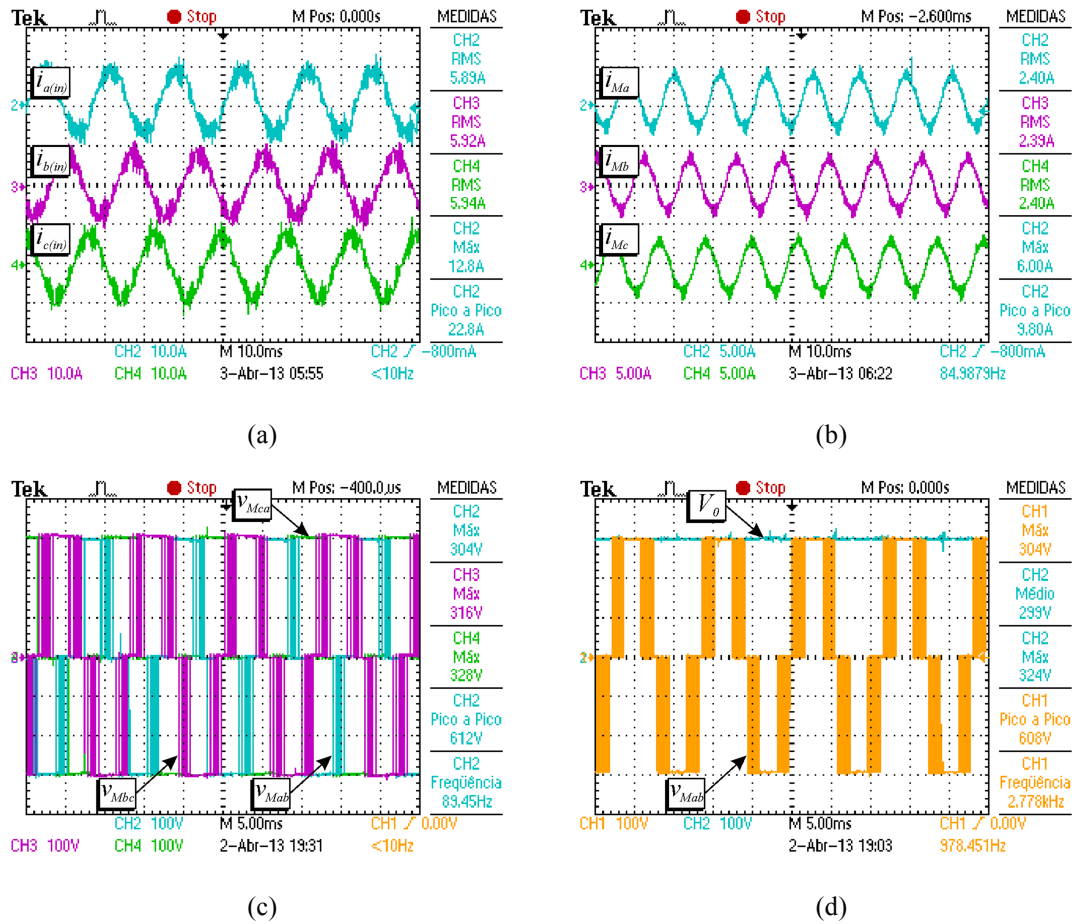


Fig. 5.22 – Formas de onda (a) das correntes de linha de entrada $i_{a(in)}$, $i_{b(in)}$ e $i_{c(in)}$; (b) das correntes de linha i_{Ma} , i_{Mb} e i_{Mc} no motor; (c) das tensões de linha v_{Mab} , v_{Mbc} e v_{Mca} no motor; e (d) da tensão no barramento CC e da tensão de linha v_{Mab} do motor, obtidas através da conexão do sistema composto pelo inversor de frequência, motor e freio eletromagnético ao barramento CC fornecido pelo RHT.

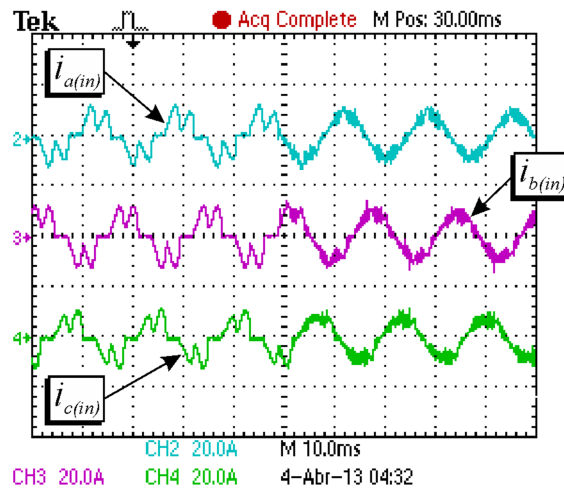


Fig. 5.23 – Formas de onda das correntes de linha de entrada $i_{a(in)}$, $i_{b(in)}$ e $i_{c(in)}$ no início da atuação dos conversores SEPIC, obtidas através da conexão do sistema composto pelo inversor de frequência, motor e freio eletromagnético ao barramento CC fornecido pelo RHT.

Observa-se através das Fig. 5.22 e 5.23 o bom funcionamento do RHT em termos de mitigação de conteúdo harmônico de corrente e correção de fator de potência diante do fornecimento de um barramento CC para a conexão também de cargas não lineares.

5.4 Considerações Finais

Neste capítulo foram apresentados os resultados experimentais obtidos a partir de um protótipo do RHT de 5 kW desenvolvido em laboratório.

Em relação às simulações apresentadas no Capítulo 3, os resultados experimentais obtidos foram bastante satisfatórios, comprovando que os conversores chaveados realmente são capazes de determinar correntes de linha de entrada senoidais, garantindo elevado fator de potência e reduzida distorção harmônica de corrente, tal como nos conversores pré-reguladores convencionais. É importante ressaltar que a estratégia de controle digital utilizada mostrou-se muito eficaz, permitindo uma maior facilidade durante a implementação do controle e um excelente tempo de resposta quanto a variações de carga, uma vez que a técnica de controle de corrente por histerese é muito robusta quanto a estas variações e proporciona um desempenho muito bom durante condições transitórias.

Uma análise das potências processadas pelos conversores envolvidos na estrutura foi mostrada nos resultados experimentais. Verificou-se que em condições normais de operação e suprimento de tensão, os retificadores chaveados processam aproximadamente 39.484 % da potência total (13.16% para cada SEPIC), sendo o restante, 60.516%, processada pelo retificador trifásico não controlado de seis pulsos, assegurando o alto rendimento da estrutura.

Em conformidade ao que foi descrito nos Capítulos 3 e 4, o ganho k igual a 1,113, utilizado no algoritmo da estratégia de controle do RHT desenvolvida no DSP foi capaz de manter o nível de DHT_1 das correntes de linha de entrada em conformidade com a norma IEC 61000-3-2 e determinar que o percentual de processamento de potência por parte dos conversores chaveados seja de 40% em relação à potência ativa média total de saída, mesmo quando da ocorrência de variações de carga.

Uma análise da eficiência do RHT diante da atuação sob diferentes condições de carga foi realizada, o que determinou um rendimento global da estrutura de aproximadamente 96% para cargas superiores a 3 kW, tornando o retificador híbrido



trifásico proposto bastante atrativo para aplicações em instalações industriais de elevada potência.

Capítulo 6

Conclusões Gerais

Na introdução deste trabalho foram apresentados os problemas aos quais o sistema elétrico tem sido submetido devido ao aumento do conteúdo harmônico das correntes drenadas pelas diversas cargas não lineares presentes na rede de alimentação. O aumento deste conteúdo harmônico está diretamente ligado ao aumento da potência aparente requerida por estas cargas, o que resulta na redução considerável do fator de potência. Foi apresentada então uma estrutura retificadora híbrida composta pela associação em paralelo de conversores SEPIC monofásicos ligados a cada braço de um retificador trifásico não controlado de seis pulsos, capaz de impor correntes de linha de entrada senoidais em fase com as tensões trifásicas de fase, capaz de promover baixo conteúdo harmônico destas correntes e elevado fator de potência.

No Capítulo 2 foi descrito o princípio de funcionamento do retificador híbrido trifásico (RHT) proposto, destacando que as correntes de linha de entrada drenadas da rede resultam da soma, para cada fase, entre as correntes drenadas pelo retificador trifásico não controlado de seis pulsos e as correntes impostas pelos conversores chaveados. Este capítulo teve como ênfase a apresentação do equacionamento utilizado para a determinação do valor das indutâncias e das capacitâncias presentes no circuito de potência do RHT. Adicionalmente, a nova técnica de controle desenvolvida neste trabalho foi apresentada. Esta foi implementada digitalmente utilizando-se o DSP *TMS320F28335* da *Texas Instruments*[®] e consistiu em estabelecer a imposição das correntes nos indutores de entrada dos conversores SEPIC seguindo uma referência senoidal, de forma a garantir que as correntes de linha de entrada apresentassem um conteúdo harmônico em conformidade com a norma IEC 61000-3-2, além de garantir que o retificador trifásico não controlado de seis pulsos processasse 60% da potência ativa total entregue à carga, promovendo maior confiabilidade operacional e robustez à estrutura proposta. Para tanto se utilizou o controlador de corrente por histerese, devido, principalmente, à sua robustez diante de variações de carga, assim como pela excelente

resposta dinâmica durante condições transitórias, fato este comprovado na análises computacionais e experimentais.

Utilizando-se a técnica de espaço de estados, foi apresentado no Capítulo 3 todo o equacionamento necessário para a determinação das matrizes de sistema e de entrada utilizadas na solução do modelo matemático desenvolvido para o RHT. Com base na estratégia de controle apresentada no Capítulo 2, foi elaborado um código matemático utilizando o *software Matlab*[®] para o desenvolvimento da modelagem matemática e para a consequente obtenção das formas de ondas das correntes e das tensões nos elementos que compõem o circuito de potência, determinando-se, desta forma, os valores de pico, eficazes e médios das tensões e correntes nos dispositivos semicondutores, os quais foram utilizados para a especificação dos componentes utilizados no protótipo. Para fins de validação do modelo matemático desenvolvido, resultados de simulação obtidos na plataforma computacional *PSim*[®] foram confrontados com os resultados obtidos com o *Matlab*[®], comprovando a eficácia da modelagem matemática desenvolvida.

O Capítulo 4, por sua vez, destacou-se pela explicação da transformação da técnica de controle desenvolvida em linguagem digital para implementação no DSP. Foram apresentadas as configurações dos registros do DSP responsáveis pelo funcionamento do conversor analógico-digital utilizado para a aquisição dos sinais de corrente e tensão sensorizados e pelos pinos de I/O utilizados para o envio dos pulsos para os interruptores dos conversores chaveados.

Por fim, o Capítulo 5 apresentou os resultados experimentais obtidos com a construção de protótipo de 5 kW do RHT nas dependências do Núcleo de Pesquisa em Eletrônica de Potência (NUPEP) da Faculdade de Engenharia Elétrica (FEELT) da Universidade Federal de Uberlândia (UFU), destacando também os resultados experimentais das correntes de linha de entrada e das correntes drenadas por Ret-1 e Ret-2, comprovando a eficácia da estratégia de controle no que tange a mitigação do conteúdo harmônico de corrente e a consequente correção do fator de potência, determinando um DHT das correntes CA de alimentação em conformidade com a norma IEC 61000-3-2. Demonstrou-se que a técnica de controle implementada no DSP foi realmente capaz de estabelecer a divisão de potência entre as unidades retificadoras de tal maneira que um simples e robusto retificador trifásicos não controlado processe a maior parte da potência ativa requerida pela carga, o que garantiu um rendimento global da estrutura em torno de 96% para ampla faixa de carga.

Como sugestão para trabalhos futuros, em complemento ao trabalho apresentado em [23], propõe-se o desenvolvimento de um PLL digital para sincronismo entre os sinais de referências senoidais utilizados para imposição das correntes nos conversores chaveados e as respectivas tensões de fase para que, em caso de distúrbios na rede CA de alimentação em que ocorra deslocamento de fase entre as tensões, conforme observado em alguns casos de afundamentos temporários de tensão, de tal sorte que, mesmo durante a ocorrência dos distúrbios, a imposição de corrente com magnitude e forma de onda desejadas seja alcançada. Além disso, sugere-se desenvolver um compensador digital de potência para garantir divisão igualitária de potência entre os conversores chaveados de modo que o processamento de potência entre as estruturas retificadoras que compõem o RHT seja mantido mesmo diante da ocorrência de afundamentos de tensão.

Referências Bibliográficas

[1] Martins A. S.; Bonan G.; Flores G. C. Entendendo o Fator de Potência. Departamento de Pesquisa e Desenvolvimento - CP Eletrônica S.A. Disponível em: <http://www.cp.com.br/upl/artigo_4.pdf>. Acesso em: 06 maio 2012.

[2] Redl R. Reducing distortion in boost rectifiers with automatic control. Applied Power Electronics Conference and Exposition, pp. 74-80, 1997.

[3] Melo P. F.; Gules R.; Romaneli E. F. R.; Annunziato R. C. A. Modified SEPIC converter for high-power-factor rectifier and universal input voltage applications. IEEE Transactions on Power Electronics, vol. 25, no. 2, Feb. 2010.

[4] Kolar J. W.; Friedli T. The essence of three-phase PFC rectifier systems. IEEE 33rd International Telecommunications Energy Conference, pp. 1-27, 2011.

[5] Alves R. L.; Barbi I. Analysis and implementation of a hybrid high-power-factor three-phase unidirectional rectifier. IEEE Transactions on Power Electronics, vol. 24, no. 3, pp. 632-640, Mar. 2009.

[6] Font C. H. I.; Barbi I. A high power hybrid three-phase rectifier with regenerate capability. Eletrônica de Potência, vol. 17, no. 1, pp. 419-428, Feb. 2012.

[7] Freitas L. C. G.; Simoes M. G.; Canesin C. A.; Freitas L. C. A novel programmable PFC based hybrid rectifier for ultra clean power application. Power Electronics Specialist Conference, pp. 2172-2177, 2004.

[8] Freitas L. C. G.; Simoes M. G.; Canesin C. A.; Freitas L. C. Programmable PFC based hybrid multipulse power rectifier for ultra clean power application. IEEE Transactions on Power Electronics, vol. 21, no. 4, pp. 959-966, Jul. 2006.

[9] Freitas L. C. G.; Simoes M. G.; Canesin C. A.; Freitas L. C. Performance evaluation of a novel hybrid multipulse rectifier for utility interface of power electronic

converters. IEEE Transactions on Industrial Electronics, vol. 54, no. 6, pp. 3030-3041, Dez. 2007.

[10] Hahn J.; Enjeti P.N.; Pitel I. J. A new three-phase power-factor correction (pfc) scheme using two single-phase pfc modules. IEEE Transactions on Industrial Electronics, vol. 38, no. 1, pp. 123-130, Jan./Fev. 2002.

[11] Spiazzi G.; Lee F. C. Implementation of single-phase boost power-factor-correction circuits in three-phase applications. IEEE Transactions on Industrial Electronics, vol. 44, no. 3, pp. 365-371, 1997.

[12] Kamnarn U.; Chunkag V. Nearly unity power-factor of the modular three-phase AC to DC converter with minimized DC bus capacitor. IEEE 4th Power Conversion Conference, pp. 1636-1643, 2007.

[13] Zhang W.; Feng G.; Liu Y.; Wu B. A digital power factor correction (PFC) control strategy optimized for DSP. IEEE Transactions on Power Electronics, vol. 19, no. 6, pp. 1474-1485, Nov. 2004.

[14] Shao-long L.; Ai-juan J.; Hang-tian L. An improved and digital current control strategy for one cycle control based three-phase boost rectifier under unbalanced system. Power Electronics and Motion Control Conference, pp. 1-5, 2006.

[15] Sing B.; Sing B. N.; Chandra A.; et al. A review of three-phase improved power quality AC-DC converters. IEEE Transactions on Industrial Electronics, vol. 51, no. 3, pp. 641-660, Jun. 2004.

[16] Barreto L. H. S. C.; Vieira Júnior J. B.; Coelho E. A. A.; Farias V. J.; de Freitas L. C. The Bang-Bang Hysteresis Current Waveshaping Control Technique Used to Implement a High Power Factor Power Supply. IEEE Transactions on Power Electronics, vol. 19, no. 1, pp. 160-168, Jan. 2004.

[17] Toffano R.; Pereira A. A.; Farias V. J.; Vieira Júnior J. B.; de Freitas L. C. A lossless Commutation PWM Boost Converter with Unity Power Factor Operation. IEE European Conference on Power Electronics and Applications, pp. 4454-4458, Set. 1997.

[18] Rasoanarivo I.; Tehrani K. A.; Sargos F. M. Fractional Order PID and Modulated hysteresis for High Performance Current control in Multilevel Inverters. IAS Annual Meeting 2011, pp. 1-7.

[19] Rasoanarivo I.; Bréchet S.; Battiston A.; Nahid-Mobarakeh B. Behavioral Analysis of a Boost Converter with High Performance Source Filter and a Fractional Order PID Controller. IAS Annual Meeting 2012, pp. 1-7.

[20] Stefanutti W.; Mattaveli P. Fully Digital Hysteresis Modulation With Switching-Time Prediction. IEEE Transactions on Industry Applications, vol. 42, no. 3, pp. 763-769, Mai/Jun. 2006.

[21] Costa A. V. ; Rodrigues D. B. ; Lima, G. B. ; Freitas L. C. ; Coelho E. A. A. ; Farias V. J. ; Freitas L. C. G. Retificador Híbrido De Alta Potência Com Reduzida Dhti Utilizando Conversor Boost Para Promover Alta Suportabilidade A Afundamentos De Tensão. Eletrônica de Potência, vol. 17, pp. 609-622, 2012.

[22] Rodrigues D. B. ; Costa A. V. ; Lima G. B. ; Freitas L. C. ; Coelho E. A. A. ; Farias V. J. ; Freitas L. C. G. DSP-based Implementation of Control Strategy for Sinusoidal Input Line Current Imposition for a Hybrid Three-phase Rectifier. IEEE Transactions on Industrial Informatics, v. PP, pp. 1, 2012.

[23] Costa A. V. ; Rodrigues D. B. ; Lima G. B. ; Freitas L. C. ; Coelho E. A. A. ; Freitas L. C. G. Retrofitting Technique to Improve Voltage Sags Ride-Through Capability of ASD Using Sepic Rectifiers. In: IEEE Applied Power Electronics Conference and Exposition, APEC, 2012, Orlando. IEEE Applied Power Electronics Conference and Exposition, 2012.

[24] Rodrigues D. B. ; Freitas L. C. ; Coelho E. A. A. ; Vieira Junior J. B. ; Freitas L. C. G. ; Farias V. J. ; Costa A. V. Hybrid three-phase rectifier with high power factor and voltage sags ride-through capability for utility interface of adjustable speed drives. In: XI Congresso Brasileiro de Eletrônica de Potência - XI COBEP, 2011, Natal - RN. Power Electronics Conference (COBEP), 2011 Brazilian, 2011. pp. 582-588.

[25] Costa A. V. ; Rodrigues D. B. ; Freitas L. C. ; Vieira Junior J. B. ; Coelho E. A. A. ; Farias V. J. ; Freitas L. C. G. Hybrid high-power rectifier with reduced THDI and voltage sag ride-through capability. In: IAS Annual Meeting, 2011, Orlando, Florida USA. Industry Applications Society Annual Meeting (IAS), 2011 IEEE, 2011. pp. 1-6.

[26] Costa A. V. ; Rodrigues D. B. ; Freitas L. C. ; Vieira Junior J. B. ; Coelho E. A. A. ; Farias V. J. ; Freitas L. C. G. Proposta de Técnica para Aumento de Suportabilidade a Afundamentos Temporários de Tensão em Acionamentos de Velocidade Variável. In: IX Conferência Brasileira sobre Qualidade da Energia Elétrica (IX CBQEE), 2011, Cuiabá - MT. Conferência Brasileira sobre Qualidade da Energia Elétrica (CBQEE), 2011. pp. 1-6.

[27] Rodrigues D. B. ; Freitas L. C. G. Retificador híbrido trifásico de alta potência e elevado fator de potência com controle digital utilizando DSP. In: I Encontro de Iniciação Científica e Tecnológica da UFU, 2011, Uberlândia, MG. I Encontro de Iniciação Científica e Tecnológica da UFU, 2011.

[28] Rodrigues D. B. ; Costa A. V. ; Lima G. B. ; Freitas L. C. ; Coelho E. A. A. ; Farias V. J. ; Freitas L. C. G. Implementação de Conversores SEPIC Não Isolados em Sistemas Trifásicos para Obtenção de Elevado Fator de Potência e Suportabilidade a Afundamentos de Tensão. In: 10th IEEE/IAS International Conference on Industry Applications, 2012. INDUSCON, 2012. pp. 1-9.

[29] Freitas, L. C. G. Um Novo Retificador Híbrido MultiPulsos sem a Utilização de Transformadores Defasadores e/ou Transformadores de Interfase. 2006. 180 f. Tese (Doutorado) - Universidade Federal de Uberlândia, Uberlândia, 2006.

[30] Barbi I. Eletrônica de Potência: Projeto de Fontes Chaveadas. 1ª Edição, Florianópolis: Edição do Autor, 2001. 334 p.

[31] Martins D. C.; Barbi I. Eletrônica de Potência: Conversores CC-CC Básicos Não Isolados. 2ª Edição Revisada, Florianópolis: Edição dos Autores, 2006. 377 p.

[32] Texas Instruments - TMS320F28335, Digital Signal Controllers (DSCs), SPRS439F–June 2007–Revised April 2009.

[33] Lima G. B. Análise e desenvolvimento de um novo conversor CA-CC híbrido monofásico com controle digital baseado em DSP. 2011. 148 f. Dissertação (Mestrado) - Faculdade de Engenharia Elétrica, Universidade Federal de Uberlândia, Uberlândia, 2011. Cap. 4.

[34] Nise N. S. Engenharia de Sistemas de Controle. 3ª Edição, Rio de Janeiro: LTC, 2002. 695 p.

Apêndice A – Código Matemático para Solução do Modelo

```
clc; clear
clear all
close all
fprintf('PREPARANDO A SIMULAÇÃO. AGUARDE...');
%% Parâmetros do Circuito de Potência
Vp = 180;
f = 60;
Leq = 2/1000;
L4 = 1/1000;
L5 = 0.5/1000;
L6 = 1/1000;
L7 = 0.5/1000;
L8 = 1/1000;
L9 = 0.5/1000;
C0 = 3000/1000000;
C1 = 22/1000000;
C2 = 22/1000000;
C3 = 22/1000000;
C4 = 22/1000000;
C5 = 22/1000000;
C6 = 22/1000000;
R = 17;
k = 1.113; %Ganho da Tensão de Referência

%% Parametros de Integração
el = 14;
f = 60;
ciclos = 5;
T = 1/60;
tmax = ciclos*T; %Tempo máximo de simulação
deltaT = 4e-6; %Passo de integração
t = 0:deltaT:tmax; %vetor tempo com intervalo deltat

%% Tensão de Entrada
Van = Vp*sin(2*pi*f*t);
Vbn = Vp*sin(2*pi*f*t - ((2*pi)/3));
Vcn = Vp*sin(2*pi*f*t + ((2*pi)/3));
Va = abs(Van);
Vb = abs(Vbn);
Vc = abs(Vcn);
isen_A = abs(Van)/Vp;
isen_B = abs(Vbn)/Vp;
isen_C = abs(Vcn)/Vp;
for i=1:length(t)
    Vab(i) = Van(i) - Vbn(i);
    Vac(i) = Van(i) - Vcn(i);
    Vbc(i) = Vbn(i) - Vcn(i);
    Vba(i) = Vbn(i) - Van(i);
    Vca(i) = Vcn(i) - Van(i);
    Vcb(i) = Vcn(i) - Vbn(i);
end

%% Matriz de Constantes
A1 = [0 0 0 0 0 0 0 -1/(Leq) 0 0 0 0 0 0;0 0 0 0 0 0 0 0 0 0 0 0 0 0;0 0 0 0 0 0 0 0 0 0 0 0 0 1/L5 1/L5 0 0 0
0;0 0 0 0 0 0 0 0 0 0 0 0 0 0;0 0 0 0 0 0 0 0 0 0 0 0 0 1/L7 1/L7 0 0;0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;0 0 0 0 0
0 0 0 0 0 1/L9 1/L9;1/C0 0 0 0 0 0 0 -1/(R*C0) 0 0 0 0 0;0 0 -1/C1 0 0 0 0 0 0 0 0 0 0;0 0 0 -
1/C2 0 0 0 0 0 0 0 0 0 0 -1/C3 0 0 0 0 0 0 0 0;0 0 0 0 -1/C4 0 0 0 0 0 0 0 0;0 0 0 0 0
0 -1/C5 0 0 0 0 0 0 0 0 0 0 0 -1/C6 0 0 0 0 0 0];
B1 = [1/(Leq) 0 0 0;0 1/L4 0 0;0 0 0 0;0 0 1/L6 0;0 0 0 0;0 0 0 1/L8;0 0 0 0;0 0 0 0;0 0 0 0
0;0 0 0 0;0 0 0 0;0 0 0 0;0 0 0];
A2 = [0 0 0 0 0 0 0 -1/(Leq) 0 0 0 0 0;0 0 0 0 0 0 0 0 0 0 0 0 0 0;0 0 0 0 0 0 0 0 0 0 0 0 0 1/L5 1/L5 0 0 0
0;0 0 0 0 0 0 0 0 0 0 0 0 0 0;0 0 0 0 0 0 0 0 0 0 0 0 0 1/L7 1/L7 0 0;0 0 0 0 0 0 0 0 -1/L8 0 0 0 0 -1/L8 -
1/L8;0 0 0 0 0 0 0 -1/L9 0 0 0 0 0;1/C0 0 0 0 0 0 1/C0 1/C0 -1/(R*C0) 0 0 0 0 0;0 0 -1/C1 0 0 0 0 0
0 0 0 0;0 0 -1/C2 0 0 0 0 0 0 0 0 0 0 0 0;0 0 0 0 -1/C3 0 0 0 0 0 0 0 0;0 0 0 0 -1/C4 0 0 0 0 0 0 0
0 0 0;0 0 0 0 1/C5 0 0 0 0 0 0 0 0;0 0 0 0 0 1/C6 0 0 0 0 0 0 0];
B2 = [1/(Leq) 0 0 0;0 1/L4 0 0;0 0 0 0;0 0 0 1/L6 0;0 0 0 0;0 0 0 1/L8;0 0 0 0;0 0 0 0;0 0 0 0;0 0 0
0;0 0 0 0;0 0 0 0;0 0 0 0;0 0 0];
A3 = [0 0 0 0 0 0 0 -1/(Leq) 0 0 0 0 0;0 0 0 0 0 0 0 0 0 0 0 0 0 0;0 0 0 0 0 0 0 0 0 0 0 0 0 1/L5 1/L5 0 0 0
0;0 0 0 0 0 0 0 -1/L6 0 0 -1/L6 -1/L6 0 0;0 0 0 0 0 0 0 -1/L7 0 0 0 0 0;0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0;0 0 0 0 0 0 0 0 0 0 0 0 0 1/L9 1/L9;1/C0 0 0 1/C0 1/C0 0 0 -1/(R*C0) 0 0 0 0 0;0 0 0 0;0 0 -1/C1 0 0 0 0 0
0 0 0]
```

```

0 0 0 0 0 0;0 0 -1/C2 0 0 0 0 0 0 0 0 0 0;0 0 0 0 1/C3 0 0 0 0 0 0 0 0;0 0 0 0 1/C4 0 0 0 0 0 0 0 0
0 0;0 0 0 0 0 0 -1/C5 0 0 0 0 0 0 0 0;0 0 0 0 0 0 -1/C6 0 0 0 0 0 0 0 0];
B3 = [1/(Leq) 0 0 0;0 1/L4 0 0;0 0 0 0;0 0 1/L6 0;0 0 0 0;0 0 0 1/L8;0 0 0 0;0 0 0 0;0 0 0 0
0;0 0 0 0;0 0 0 0;0 0 0 0];
A4 = [0 0 0 0 0 0 -1/(Leq) 0 0 0 0 0 0;0 0 0 0 0 0 0 0 0 0 0 0 0 0;0 0 0 0 0 0 0 0 1/L5 1/L5 0 0 0
0;0 0 0 0 0 0 -1/L6 0 0 -1/L6 -1/L6 0 0;0 0 0 0 0 0 0 -1/L7 0 0 0 0 0 0;0 0 0 0 0 0 0 -1/L8 0 0 0 0
-1/L8 -1/L8;0 0 0 0 0 0 0 -1/L9 0 0 0 0 0 0;1/C0 0 0 1/C0 1/C0 1/C0 1/C0 -1/(R*C0) 0 0 0 0 0 0;0 0 0 -
1/C1 0 0 0 0 0 0 0 0 0 0 0 0;0 0 -1/C2 0 0 0 0 0 0 0 0 0 0 0 0;0 0 0 1/C3 0 0 0 0 0 0 0 0;0 0 0 1/C4
0 0 0 0 0 0 0 0;0 0 0 0 0 1/C5 0 0 0 0 0 0 0;0 0 0 0 0 1/C6 0 0 0 0 0 0 0 0];
B4 = [1/(Leq) 0 0 0;0 1/L4 0 0;0 0 0 0;0 0 1/L6 0;0 0 0 0;0 0 0 1/L8;0 0 0 0;0 0 0 0;0 0 0 0;0 0 0
0;0 0 0 0;0 0 0 0;0 0 0 0];
A5 = [0 0 0 0 0 0 -1/(Leq) 0 0 0 0 0 0;0 0 0 0 0 0 0 -1/L4 -1/L4 -1/L4 0 0 0 0;0 0 0 0 0 0 0 -1/L5
0 0 0 0 0 0;0 0 0 0 0 0 0 0 0 0 0 0;0 0 0 0 0 0 0 0 1/L7 1/L7 0 0;0 0 0 0 0 0 0 0 0 0 0 0
0;0 0 0 0 0 0 0 0 0 0 0 0 1/L9 1/L9;1/C0 1/C0 1/C0 1/C0 0 0 0 -1/(R*C0) 0 0 0 0 0;0 1/C1 0 0 0 0 0 0
0 0 0 0;0 1/C2 0 0 0 0 0 0 0 0 0 0;0 0 0 0 -1/C3 0 0 0 0 0 0 0 0;0 0 0 0 -1/C4 0 0 0 0 0 0 0 0
0 0;0 0 0 0 0 -1/C5 0 0 0 0 0 0 0;0 0 0 0 0 0 -1/C6 0 0 0 0 0 0 0 0];
B5 = [1/(Leq) 0 0 0;0 1/L4 0 0;0 0 0 0;0 0 1/L6 0;0 0 0 0;0 0 0 1/L8;0 0 0 0;0 0 0 0;0 0 0 0;0 0 0
0;0 0 0 0;0 0 0 0;0 0 0 0];
A6 = [0 0 0 0 0 0 -1/(Leq) 0 0 0 0 0 0;0 0 0 0 0 0 0 -1/L4 -1/L4 -1/L4 0 0 0 0;0 0 0 0 0 0 0 -1/L5
0 0 0 0 0 0;0 0 0 0 0 0 0 0 0 0 0 0;0 0 0 0 0 0 0 0 1/L7 1/L7 0 0;0 0 0 0 0 0 0 -1/L8 0 0 0 0
-1/L8 -1/L8;0 0 0 0 0 0 0 -1/L9 0 0 0 0 0 0;1/C0 1/C0 1/C0 0 0 1/C0 1/C0 -1/(R*C0) 0 0 0 0 0 0;0 1/C1
0 0 0 0 0 0 0 0 0 0;0 1/C2 0 0 0 0 0 0 0 0 0 0;0 0 0 0 -1/C3 0 0 0 0 0 0 0 0;0 0 0 0 -1/C4
0 0 0 0 0 0 0 0;0 0 0 0 1/C5 0 0 0 0 0 0 0;0 0 0 0 1/C6 0 0 0 0 0 0 0 0];
B6 = [1/(Leq) 0 0 0;0 1/L4 0 0;0 0 0 0;0 0 1/L6 0;0 0 0 0;0 0 0 1/L8;0 0 0 0;0 0 0 0;0 0 0 0;0 0 0
0;0 0 0 0;0 0 0 0;0 0 0 0];
A7 = [0 0 0 0 0 0 -1/(Leq) 0 0 0 0 0 0;0 0 0 0 0 0 0 -1/L4 -1/L4 -1/L4 0 0 0 0;0 0 0 0 0 0 0 -1/L5
0 0 0 0 0 0;0 0 0 0 0 0 0 -1/L6 0 0 -1/L6 -1/L6 0 0;0 0 0 0 0 0 0 -1/L7 0 0 0 0 0 0;0 0 0 0 0 0 0 0
0 0 0 0;0 0 0 0 0 0 0 0 0 0 0 1/L9 1/L9;1/C0 1/C0 1/C0 1/C0 1/C0 0 0 -1/(R*C0) 0 0 0 0 0;0 1/C1
0 0 0 0 0 0 0 0 0 0;0 1/C2 0 0 0 0 0 0 0 0 0 0;0 0 0 0 1/C3 0 0 0 0 0 0 0 0;0 0 0 1/C4 0 0
0 0 0 0 0 0 0;0 0 0 0 0 -1/C5 0 0 0 0 0 0 0;0 0 0 0 0 -1/C6 0 0 0 0 0 0 0 0];
B7 = [1/(Leq) 0 0 0;0 1/L4 0 0;0 0 0 0;0 0 1/L6 0;0 0 0 0;0 0 0 1/L8;0 0 0 0;0 0 0 0;0 0 0 0;0 0 0
0;0 0 0 0;0 0 0 0;0 0 0 0];
A8 = [0 0 0 0 0 0 -1/(Leq) 0 0 0 0 0 0;0 0 0 0 0 0 0 -1/L4 -1/L4 -1/L4 0 0 0 0;0 0 0 0 0 0 0 -1/L5
0 0 0 0 0 0;0 0 0 0 0 0 -1/L6 0 0 -1/L6 -1/L6 0 0;0 0 0 0 0 0 0 -1/L7 0 0 0 0 0 0;0 0 0 0 0 0 0 -
1/L8 0 0 0 0 -1/L8 -1/L8;0 0 0 0 0 0 0 -1/L9 0 0 0 0 0 0;1/C0 1/C0 1/C0 1/C0 1/C0 1/C0 -1/(R*C0) 0 0 0 0 0
0 0 0 0;0 1/C1 0 0 0 0 0 0 0 0 0 0;0 1/C2 0 0 0 0 0 0 0 0 0 0;0 0 0 1/C3 0 0 0 0 0 0 0 0
0 0;0 0 0 1/C4 0 0 0 0 0 0 0 0;0 0 0 0 1/C5 0 0 0 0 0 0 0;0 0 0 0 1/C6 0 0 0 0 0 0 0 0];
B8 = [1/(Leq) 0 0 0;0 1/L4 0 0;0 0 0 0;0 0 1/L6 0;0 0 0 0;0 0 0 1/L8;0 0 0 0;0 0 0 0;0 0 0 0;0 0 0
0;0 0 0 0;0 0 0 0;0 0 0 0];

%% Condições Iniciais
vC0(1) = 297.7;vC0_area = 0;vC0_area2 = 0;vC0_pico = 0;vC0_avg = 297.9427;
vC1(1) = -14.63;vC1_area = 0;vC1_area2 = 0;vC1_pico = 0;
vC2(1) = -14.63;vC2_area = 0;vC2_area2 = 0;vC2_pico = 0;
vC3(1) = 73.47;vC3_area = 0;vC3_area2 = 0;vC3_pico = 0;
vC4(1) = 73.47;vC4_area = 0;vC4_area2 = 0;vC4_pico = 0;
vC5(1) = 80.87;vC5_area = 0;vC5_area2 = 0;vC5_pico = 0;
vC6(1) = 80.87;vC6_area = 0;vC6_area2 = 0;vC6_pico = 0;
vS1(1) = 0;vS1_area = 0;vS1_area2 = 0;vS1_pico = 0;
vS2(1) = 0;vS2_area = 0;vS2_area2 = 0;vS2_pico = 0;
vS3(1) = 0;vS3_area = 0;vS3_area2 = 0;vS3_pico = 0;
vD1(1) = 0;vD1_area = 0;vD1_area2 = 0;vD1_pico = 0;
vD2(1) = 0;vD2_area = 0;vD2_area2 = 0;vD2_pico = 0;
vD3(1) = 0;vD3_area = 0;vD3_area2 = 0;vD3_pico = 0;
vD4(1) = 0;vD4_area = 0;vD4_area2 = 0;vD4_pico = 0;
vD5(1) = 0;vD5_area = 0;vD5_area2 = 0;vD5_pico = 0;
vD6(1) = 0;vD6_area = 0;vD6_area2 = 0;vD6_pico = 0;
vD7(1) = 0;vD7_area = 0;vD7_area2 = 0;vD7_pico = 0;
vD8(1) = 0;vD8_area = 0;vD8_area2 = 0;vD8_pico = 0;
vD9(1) = 0;vD9_area = 0;vD9_area2 = 0;vD9_pico = 0;
Vret_area = 0;Vret_area2 = 0;Vret_pico = 0;
Va_area = 0;Va_area2 = 0;Va_pico = 0;
Vb_area = 0;Vb_area2 = 0;Vb_pico = 0;
Vc_area = 0;Vc_area2 = 0;Vc_pico = 0;
iLeq(1) = 10.5;iLeq_area = 0;iLeq_area2 = 0;iLeq_pico = 0;iLeq_avg = 10.4;
iL4(1) = 0.06;iL4_area = 0;iL4_area2 = 0;iL4_pico = 0;
iL5(1) = 0.01;iL5_area = 0;iL5_area2 = 0;iL5_pico = 0;
iL6(1) = 5.328;iL6_area = 0;iL6_area2 = 0;iL6_pico = 0;
iL7(1) = 3.221;iL7_area = 0;iL7_area2 = 0;iL7_pico = 0;
iL8(1) = 6.413;iL8_area = 0;iL8_area2 = 0;iL8_pico = 0;
iL9(1) = 5.634;iL9_area = 0;iL9_area2 = 0;iL9_pico = 0;
iS1(1) = 0;iS1_area = 0;iS1_area2 = 0;iS1_pico = 0;
iS2(1) = 0;iS2_area = 0;iS2_area2 = 0;iS2_pico = 0;
iS3(1) = 0;iS3_area = 0;iS3_area2 = 0;iS3_pico = 0;
iD1(1) = 0;iD1_area = 0;iD1_area2 = 0;iD1_pico = 0;
iD2(1) = 0;iD2_area = 0;iD2_area2 = 0;iD2_pico = 0;
iD3(1) = 0;iD3_area = 0;iD3_area2 = 0;iD3_pico = 0;
iD4(1) = 0;iD4_area = 0;iD4_area2 = 0;iD4_pico = 0;
iD5(1) = 0;iD5_area = 0;iD5_area2 = 0;iD5_pico = 0;
iD6(1) = 0;iD6_area = 0;iD6_area2 = 0;iD6_pico = 0;
iD7(1) = 0;iD7_area = 0;iD7_area2 = 0;iD7_pico = 0;
iD8(1) = 0;iD8_area = 0;iD8_area2 = 0;iD8_pico = 0;
iD9(1) = 0;iD9_area = 0;iD9_area2 = 0;iD9_pico = 0;
Io_area = 0;Io_area2 = 0;Io_avg = 17.7306;
iref_A(1) = 0; % Referência senoidal realimentada
iref_B(1) = 0; % Referência senoidal realimentada
iref_C(1) = 0; % Referência senoidal realimentada
ial(1) = 0;ial_area2 = 0;ial_pico = 0;

```

```

ia2(1) = 0; ia2_area2 = 0; ia2_pico = 0;
ia1n(1) = 0; ia1n_area2 = 0; ia1n_pico = 0;
ib1(1) = 0; ib1_area2 = 0; ib1_pico = 0;
ib2(1) = 0; ib2_area2 = 0; ib2_pico = 0;
ibin(1) = 0; ibin_area2 = 0; ibin_pico = 0;
icl(1) = 0; icl_area2 = 0; icl_pico = 0;
ic2(1) = 0; ic2_area2 = 0; ic2_pico = 0;
icin(1) = 0; icin_area2 = 0; icin_pico = 0;
ic0(1) = 0; ic0_area2 = 0;

%% Expressão da integração trapezoidal
%-----
%[dx/dt] = [A]*[x] + [B]*f(t) ---- Equação de Estado

% [x(t) = [M]*[x(t-deltaT)] + [N]*{f(t-deltaT) + f(t)} --- Solução da Equação de Estado

% [M] = {[I] - 0.5*deltaT*[A]}^-1*[I]+0.5*deltaT*[A]}
% [N] = {[I] - 0.5*deltaT*[A]}^-1*[0.5*deltaT*[B]}
%-----

%% Tensões de Atuação
for i=2:length(t)
    clc;
    fprintf('SIMULANDO...');
    fprintf('\n %.1f', ((i/(tmax/deltaT))*100);
    fprintf('%s');

    % Atuação da tensão Vab
    if ((Vab(i) > Vcb(i)) && (Vab(i) >= Vac(i)))
        Vret(i) = Vab(i);
        ia1n(i) = iLeq(i-1)+iL4(i-1);
        ibin(i) = iLeq(i-1)+iL6(i-1);
        icin(i) = iL8(i-1);

    % Atuação da tensão Vac
    elseif ((Vac(i) > Vab(i)) && (Vac(i) >= Vbc(i)))
        Vret(i) = Vac(i);
        ia1n(i) = iLeq(i-1)+iL4(i-1);
        ibin(i) = iL6(i-1);
        icin(i) = iLeq(i-1)+iL8(i-1);

    % Atuação da tensão Vbc
    elseif ((Vbc(i) > Vac(i)) && (Vbc(i) >= Vba(i)))
        Vret(i) = Vbc(i);
        ia1n(i) = iL4(i-1);
        ibin(i) = iLeq(i-1)+iL6(i-1);
        icin(i) = iLeq(i-1)+iL8(i-1);

    % Atuação da tensão Vba
    elseif ((Vba(i) > Vbc(i)) && (Vba(i) >= Vca(i)))
        Vret(i) = Vba(i);
        ia1n(i) = iLeq(i-1)+iL4(i-1);
        ibin(i) = iLeq(i-1)+iL6(i-1);
        icin(i) = iL8(i-1);

    % Atuação da tensão Vca
    elseif ((Vca(i) > Vba(i)) && (Vca(i) >= Vcb(i)))
        Vret(i) = Vca(i);
        ia1n(i) = iLeq(i-1)+iL4(i-1);
        ibin(i) = iL6(i-1);
        icin(i) = iLeq(i-1)+iL8(i-1);

    % Atuação da tensão Vcb
    elseif ((Vcb(i) > Vca(i)) && (Vcb(i) >= Vab(i)))
        Vret(i) = Vcb(i);
        ia1n(i) = iL4(i-1);
        ibin(i) = iLeq(i-1)+iL6(i-1);
        icin(i) = iLeq(i-1)+iL8(i-1);

    end
    iref_A(i) = isen_A(i)*k*Io_avg;
    iref_B(i) = isen_B(i)*k*Io_avg;
    iref_C(i) = isen_C(i)*k*Io_avg;

%% Condições das Chaves
%% 1ª) Sa(ON); Sb(ON); Sc(ON)
    if ((iref_A(i) > ia1n(i)) && (iref_B(i) > ibin(i)) && (iref_C(i) > icin(i)))
        I = eye(1);
        a = (I - 0.5*deltaT*A1);
        M = inv(a)*(I + 0.5*deltaT*A1);
        b = (I - 0.5*deltaT*A1);
        N = inv(b)*(0.5*deltaT*B1);

        %Integração Trapezoidal_1

```

```

x = M*[iLeq(i-1);iL4(i-1);iL5(i-1);iL6(i-1);iL7(i-1);iL8(i-1);iL9(i-1);vC0(i-1);vC1(i-
1);vC2(i-1);vC3(i-1);vC4(i-1);vC5(i-1);vC6(i-1)] + N*[Vret(i-1) + Vret(i);Va(i-1) + Va(i);Vb(i-1) +
Vb(i);Vc(i-1) + Vc(i)];
iLeq(i) = x(1);
iL4(i) = x(2);
iL5(i) = x(3);
iL6(i) = x(4);
iL7(i) = x(5);
iL8(i) = x(6);
iL9(i) = x(7);
vC0(i) = x(8);
vC1(i) = x(9);
vC2(i) = x(10);
vC3(i) = x(11);
vC4(i) = x(12);
vC5(i) = x(13);
vC6(i) = x(14);
iS1(i) = iL4(i)+iL5(i);
iS2(i) = iL6(i)+iL7(i);
iS3(i) = iL8(i)+iL9(i);
vS1(i) = 0;
vS2(i) = 0;
vS3(i) = 0;
iD1(i) = 0;
iD2(i) = 0;
iD3(i) = 0;
iD4(i) = 0;
iD5(i) = 0;
iD6(i) = 0;
iD7(i) = 0;
iD8(i) = 0;
iD9(i) = 0;
vD1(i) = (Van(i)+vC0(i))/2;
vD2(i) = 0;
vD3(i) = (Van(i)+vC0(i))/2;
vD4(i) = (Vbn(i)+vC0(i))/2;
vD5(i) = 0;
vD6(i) = (Vbn(i)+vC0(i))/2;
vD7(i) = (Vcn(i)+vC0(i))/2;
vD8(i) = 0;
vD9(i) = (Vcn(i)+vC0(i))/2;
iC0(i) = iLeq(i) - (vC0(i)/R);

end

%% 2ª) Sa(ON); Sb(ON); Sc(OFF)
if ((iref_A(i) > iain(i)) && (iref_B(i) > ibin(i)) && (iref_C(i) < icin(i)))
    I = eye(el);
    a = (I - 0.5*deltaT*A2);
    M = inv(a)*(I + 0.5*deltaT*A2);
    b = (I - 0.5*deltaT*A2);
    N = inv(b)*(0.5*deltaT*B2);

    %Integração Trapezoidal_1

    x = M*[iLeq(i-1);iL4(i-1);iL5(i-1);iL6(i-1);iL7(i-1);iL8(i-1);iL9(i-1);vC0(i-1);vC1(i-
1);vC2(i-1);vC3(i-1);vC4(i-1);vC5(i-1);vC6(i-1)] + N*[Vret(i-1) + Vret(i);Va(i-1) + Va(i);Vb(i-1) +
Vb(i);Vc(i-1) + Vc(i)];
    iLeq(i) = x(1);
    iL4(i) = x(2);
    iL5(i) = x(3);
    iL6(i) = x(4);
    iL7(i) = x(5);
    iL8(i) = x(6);
    iL9(i) = x(7);
    vC0(i) = x(8);
    vC1(i) = x(9);
    vC2(i) = x(10);
    vC3(i) = x(11);
    vC4(i) = x(12);
    vC5(i) = x(13);
    vC6(i) = x(14);
    iS1(i) = iL4(i)+iL5(i);
    iS2(i) = iL6(i)+iL7(i);
    vS1(i) = 0;
    vS2(i) = 0;
    vS3(i) = Vcn(i)+vC0(i);
    iS3(i) = 0;
    iD1(i) = 0;
    iD2(i) = iL5(i);
    iD3(i) = 0;
    iD4(i) = 0;
    iD5(i) = iL7(i);
    iD6(i) = 0;
    iD7(i) = iL8(i)+iL9(i);
    iD8(i) = iL9(i);
    iD9(i) = iL8(i)+iL9(i);
    vD1(i) = (Van(i)+vC0(i))/2;

```

```

vD2(i) = 0;
vD3(i) = (Van(i)+vC0(i))/2;
vD4(i) = (Vbn(i)+vC0(i))/2;
vD5(i) = 0;
vD6(i) = (Vbn(i)+vC0(i))/2;
vD7(i) = 0;
vD8(i) = 0;
vD9(i) = 0;
iC0(i) = iLeq(i) + iL8(i) + iL9(i) - (vC0(i)/R);
end

%% 3ª) Sa(ON); Sb(OFF); Sc(ON)
if ((iref_A(i) > iain(i)) && (iref_B(i) < ibin(i)) && (iref_C(i) > icin(i)))
    I = eye(el);
    a = (I - 0.5*deltaT*A3);
    M = inv(a)*(I + 0.5*deltaT*A3);
    b = (I - 0.5*deltaT*A3);
    N = inv(b)*(0.5*deltaT*B3);

    %Integração Trapezoidal_1

    x = M*[iLeq(i-1);iL4(i-1);iL5(i-1);iL6(i-1);iL7(i-1);iL8(i-1);iL9(i-1);vC0(i-1);vC1(i-1);vC2(i-1);vC3(i-1);vC4(i-1);vC5(i-1);vC6(i-1)] + N*[Vret(i-1) + Vret(i);Va(i-1) + Va(i);Vb(i-1) + Vb(i);Vc(i-1) + Vc(i)];
    iLeq(i) = x(1);
    iL4(i) = x(2);
    iL5(i) = x(3);
    iL6(i) = x(4);
    iL7(i) = x(5);
    iL8(i) = x(6);
    iL9(i) = x(7);
    vC0(i) = x(8);
    vC1(i) = x(9);
    vC2(i) = x(10);
    vC3(i) = x(11);
    vC4(i) = x(12);
    vC5(i) = x(13);
    vC6(i) = x(14);
    iS1(i) = iL4(i)+iL5(i);
    iS2(i) = 0;
    iS3(i) = iL8(i)+iL9(i);
    vS1(i) = 0;
    vS2(i) = Vbn(i)+vC0(i);
    vS3(i) = 0;
    iD1(i) = 0;
    iD2(i) = iL5(i);
    iD3(i) = 0;
    iD4(i) = iL6(i)+iL7(i);
    iD5(i) = iL7(i);
    iD6(i) = iL6(i)+iL7(i);
    iD7(i) = 0;
    iD8(i) = iL9(i);
    iD9(i) = 0;
    vD1(i) = (Van(i)+vC0(i))/2;
    vD2(i) = 0;
    vD3(i) = (Van(i)+vC0(i))/2;
    vD4(i) = 0;
    vD5(i) = 0;
    vD6(i) = 0;
    vD7(i) = (Vcn(i)+vC0(i))/2;
    vD8(i) = 0;
    vD9(i) = (Vcn(i)+vC0(i))/2;
    iC0(i) = iLeq(i) + iL6(i) + iL7(i) - (vC0(i)/R);
end

%% 4ª) Sa(ON); Sb(OFF); Sc(OFF)
if ((iref_A(i) > iain(i)) && (iref_B(i) < ibin(i)) && (iref_C(i) < icin(i)))
    I = eye(el);
    a = (I - 0.5*deltaT*A4);
    M = inv(a)*(I + 0.5*deltaT*A4);
    b = (I - 0.5*deltaT*A4);
    N = inv(b)*(0.5*deltaT*B4);

    %Integração Trapezoidal_1

    x = M*[iLeq(i-1);iL4(i-1);iL5(i-1);iL6(i-1);iL7(i-1);iL8(i-1);iL9(i-1);vC0(i-1);vC1(i-1);vC2(i-1);vC3(i-1);vC4(i-1);vC5(i-1);vC6(i-1)] + N*[Vret(i-1) + Vret(i);Va(i-1) + Va(i);Vb(i-1) + Vb(i);Vc(i-1) + Vc(i)];
    iLeq(i) = x(1);
    iL4(i) = x(2);
    iL5(i) = x(3);
    iL6(i) = x(4);
    iL7(i) = x(5);
    iL8(i) = x(6);
    iL9(i) = x(7);
    vC0(i) = x(8);
    vC1(i) = x(9);
    vC2(i) = x(10);

```

```

vC3(i) = x(11);
vC4(i) = x(12);
vC5(i) = x(13);
vC6(i) = x(14);
iS1(i) = iL4(i)+iL5(i);
iS2(i) = 0;
iS3(i) = 0;
vS1(i) = 0;
vS2(i) = Vbn(i)+vC0(i);
vS3(i) = Vcn(i)+vC0(i);
iD1(i) = 0;
iD2(i) = iL5(i);
iD3(i) = 0;
iD4(i) = iL6(i)+iL7(i);
iD5(i) = iL7(i);
iD6(i) = iL6(i)+iL7(i);
iD7(i) = iL8(i)+iL9(i);
iD8(i) = iL9(i);
iD9(i) = iL8(i)+iL9(i);
vD1(i) = (Van(i)+vC0(i))/2;
vD2(i) = 0;
vD3(i) = (Van(i)+vC0(i))/2;
vD4(i) = 0;
vD5(i) = 0;
vD6(i) = 0;
vD7(i) = 0;
vD8(i) = 0;
vD9(i) = 0;
iC0(i) = iLeq(i) + iL6(i) + iL7(i) + iL8(i) + iL9(i) - (vC0(i)/R);
end

%% 5ª) Sa(OFF); Sb(ON); Sc(ON)
if ((iref_A(i) < iain(i)) && (iref_B(i) > ibin(i)) && (iref_C(i) > icin(i)))
    I = eye(el);
    a = (I - 0.5*deltaT*A5);
    M = inv(a)*(I + 0.5*deltaT*A5);
    b = (I - 0.5*deltaT*A5);
    N = inv(b)*(0.5*deltaT*B5);

    %Integração Trapezoidal_1
    x = M*[iLeq(i-1);iL4(i-1);iL5(i-1);iL6(i-1);iL7(i-1);iL8(i-1);iL9(i-1);vC0(i-1);vC1(i-1);vC2(i-1);vC3(i-1);vC4(i-1);vC5(i-1);vC6(i-1)] + N*[Vret(i-1) + Vret(i);Va(i-1) + Va(i);Vb(i-1) + Vb(i);Vc(i-1) + Vc(i)];
    iLeq(i) = x(1);
    iL4(i) = x(2);
    iL5(i) = x(3);
    iL6(i) = x(4);
    iL7(i) = x(5);
    iL8(i) = x(6);
    iL9(i) = x(7);
    vC0(i) = x(8);
    vC1(i) = x(9);
    vC2(i) = x(10);
    vC3(i) = x(11);
    vC4(i) = x(12);
    vC5(i) = x(13);
    vC6(i) = x(14);
    iS1(i) = 0;
    iS2(i) = iL6(i)+iL7(i);
    iS3(i) = iL8(i)+iL9(i);
    vS1(i) = Van(i)+vC0(i);
    vS2(i) = 0;
    vS3(i) = 0;
    iD1(i) = iL4(i)+iL5(i);
    iD2(i) = iL5(i);
    iD3(i) = iL4(i)+iL5(i);
    iD4(i) = 0;
    iD5(i) = iL7(i);
    iD6(i) = 0;
    iD7(i) = 0;
    iD8(i) = iL9(i);
    iD9(i) = 0;
    vD1(i) = 0;
    vD2(i) = 0;
    vD3(i) = 0;
    vD4(i) = (Vbn(i)+vC0(i))/2;
    vD5(i) = 0;
    vD6(i) = (Vbn(i)+vC0(i))/2;
    vD7(i) = (Vcn(i)+vC0(i))/2;
    vD8(i) = 0;
    vD9(i) = (Vcn(i)+vC0(i))/2;
    iC0(i) = iLeq(i) + iL4(i) + iL5(i) - (vC0(i)/R);
end

%% 6ª) Sa(OFF); Sb(ON); Sc(OFF)
if ((iref_A(i) < iain(i)) && (iref_B(i) > ibin(i)) && (iref_C(i) < icin(i)))
    I = eye(el);

```

```

a = (I - 0.5*deltaT*A6);
M = inv(a)*(I + 0.5*deltaT*A6);
b = (I - 0.5*deltaT*A6);
N = inv(b)*(0.5*deltaT*B6);

%Integração Trapezoidal_1

x = M*[iLeq(i-1);iL4(i-1);iL5(i-1);iL6(i-1);iL7(i-1);iL8(i-1);iL9(i-1);vC0(i-1);vC1(i-1);vC2(i-1);vC3(i-1);vC4(i-1);vC5(i-1);vC6(i-1)] + N*[Vret(i-1) + Vret(i);Va(i-1) + Va(i);Vb(i-1) + Vb(i);Vc(i-1) + Vc(i)];
iLeq(i) = x(1);
iL4(i) = x(2);
iL5(i) = x(3);
iL6(i) = x(4);
iL7(i) = x(5);
iL8(i) = x(6);
iL9(i) = x(7);
vC0(i) = x(8);
vC1(i) = x(9);
vC2(i) = x(10);
vC3(i) = x(11);
vC4(i) = x(12);
vC5(i) = x(13);
vC6(i) = x(14);
iS1(i) = 0;
iS2(i) = iL6(i)+iL7(i);
iS3(i) = 0;
vS1(i) = Van(i)+vC0(i);
vS2(i) = 0;
vS3(i) = Vcn(i)+vC0(i);
iD1(i) = iL4(i)+iL5(i);
iD2(i) = iL5(i);
iD3(i) = iL4(i)+iL5(i);
iD4(i) = 0;
iD5(i) = iL7(i);
iD6(i) = 0;
iD7(i) = iL8(i)+iL9(i);
iD8(i) = iL9(i);
iD9(i) = iL8(i)+iL9(i);
vD1(i) = 0;
vD2(i) = 0;
vD3(i) = 0;
vD4(i) = (Vbn(i)+vC0(i))/2;
vD5(i) = 0;
vD6(i) = (Vbn(i)+vC0(i))/2;
vD7(i) = 0;
vD8(i) = 0;
vD9(i) = 0;
iC0(i) = iLeq(i) + iL4(i) + iL5(i) + iL8(i) + iL9(i) - (vC0(i)/R);
end

%% 7ª) Sa(OFF); Sb(OFF); Sc(ON)
if ((iref_A(i) < iain(i)) && (iref_B(i) < ibin(i)) && (iref_C(i) > icin(i)))
I = eye(e1);
a = (I - 0.5*deltaT*A7);
M = inv(a)*(I + 0.5*deltaT*A7);
b = (I - 0.5*deltaT*A7);
N = inv(b)*(0.5*deltaT*B7);

%Integração Trapezoidal_1

x = M*[iLeq(i-1);iL4(i-1);iL5(i-1);iL6(i-1);iL7(i-1);iL8(i-1);iL9(i-1);vC0(i-1);vC1(i-1);vC2(i-1);vC3(i-1);vC4(i-1);vC5(i-1);vC6(i-1)] + N*[Vret(i-1) + Vret(i);Va(i-1) + Va(i);Vb(i-1) + Vb(i);Vc(i-1) + Vc(i)];
iLeq(i) = x(1);
iL4(i) = x(2);
iL5(i) = x(3);
iL6(i) = x(4);
iL7(i) = x(5);
iL8(i) = x(6);
iL9(i) = x(7);
vC0(i) = x(8);
vC1(i) = x(9);
vC2(i) = x(10);
vC3(i) = x(11);
vC4(i) = x(12);
vC5(i) = x(13);
vC6(i) = x(14);
iS1(i) = 0;
iS2(i) = 0;
vS1(i) = Van(i)+vC0(i);
vS2(i) = Vbn(i)+vC0(i);
vS3(i) = 0;
iS3(i) = iL8(i)+iL9(i);
iD1(i) = iL4(i)+iL5(i);
iD2(i) = iL5(i);
iD3(i) = iL4(i)+iL5(i);
iD4(i) = iL6(i)+iL7(i);

```

```

        iD5(i) = iL7(i);
        iD6(i) = iL6(i)+iL7(i);
        iD7(i) = 0;
        iD8(i) = iL9(i);
        iD9(i) = 0;
        vD1(i) = 0;
        vD2(i) = 0;
        vD3(i) = 0;
        vD4(i) = 0;
        vD5(i) = 0;
        vD6(i) = 0;
        vD7(i) = (Vcn(i)+vC0(i))/2;
        vD8(i) = 0;
        vD9(i) = (Vcn(i)+vC0(i))/2;
        iC0(i) = iLeq(i) + iL4(i) + iL5(i) + iL6(i) + iL7(i) - (vC0(i)/R);
    end

%% 8ª) Sa(OFF); Sb(OFF); Sc(OFF)
if ((iref_A(i) < iain(i)) && (iref_B(i) < ibin(i)) && (iref_C(i) < icin(i)))
    I = eye(e1);
    a = (I - 0.5*deltaT*A8);
    M = inv(a)*(I + 0.5*deltaT*A8);
    b = (I - 0.5*deltaT*A8);
    N = inv(b)*(0.5*deltaT*B8);

    %Integração Trapezoidal_1

    x = M*[iLeq(i-1);iL4(i-1);iL5(i-1);iL6(i-1);iL7(i-1);iL8(i-1);iL9(i-1);vC0(i-1);vC1(i-1);vC2(i-1);vC3(i-1);vC4(i-1);vC5(i-1);vC6(i-1)] + N*[Vret(i-1) + Vret(i);Va(i-1) + Va(i);Vb(i-1) + Vb(i);Vc(i-1) + Vc(i)];
    iLeq(i) = x(1);
    iL4(i) = x(2);
    iL5(i) = x(3);
    iL6(i) = x(4);
    iL7(i) = x(5);
    iL8(i) = x(6);
    iL9(i) = x(7);
    vC0(i) = x(8);
    vC1(i) = x(9);
    vC2(i) = x(10);
    vC3(i) = x(11);
    vC4(i) = x(12);
    vC5(i) = x(13);
    vC6(i) = x(14);
    iS1(i) = 0;
    iS2(i) = 0;
    iS3(i) = 0;
    vS1(i) = Van(i)+vC0(i);
    vS2(i) = Vbn(i)+vC0(i);
    vS3(i) = Vcn(i)+vC0(i);
    iD1(i) = iL4(i)+iL5(i);
    iD2(i) = iL5(i);
    iD3(i) = iL4(i)+iL5(i);
    iD4(i) = iL6(i)+iL7(i);
    iD5(i) = iL7(i);
    iD6(i) = iL6(i)+iL7(i);
    iD7(i) = iL8(i)+iL9(i);
    iD8(i) = iL9(i);
    iD9(i) = iL8(i)+iL9(i);
    vD1(i) = 0;
    vD2(i) = 0;
    vD3(i) = 0;
    vD4(i) = 0;
    vD5(i) = 0;
    vD6(i) = 0;
    vD7(i) = 0;
    vD8(i) = 0;
    vD9(i) = 0;
    iC0(i) = iLeq(i) + iL4(i) + iL5(i) + iL6(i) + iL7(i) + iL8(i) + iL9(i) - (vC0(i)/R);
end

%% Retificação das Correntes
if (iLeq(i) < 0)
    iLeq(i) = -iLeq(i);
end
if (iL4(i) < 0)
    iL4(i) = -iL4(i);
end
if (iL5(i) < 0)
    iL5(i) = -iL5(i);
end
if (iL6(i) < 0)
    iL6(i) = -iL6(i);
end
if (iL7(i) < 0)
    iL7(i) = -iL7(i);
end
if (iL8(i) < 0)

```

```

        iL8(i) = -iL8(i);
    end
    if (iL9(i) < 0)
        iL9(i) = -iL9(i);
    end
%% Determinação das Formas de Onda das Correntes

% Atuação da tensão Vab
    if ((Vab(i) > Vcb(i)) && (Vab(i) >= Vac(i)))
        ia1(i) = iLeq(i);
        ia2(i) = iL4(i);
        iain(i) = iain(i);
        ib1(i) = -iLeq(i);
        ib2(i) = -iL6(i);
        ibin(i) = -ibin(i);
        if (Vcn(i) >= 0)
            ic1(i) = 0;
            ic2(i) = iL8(i);
            icin(i) = icin(i);
        else
            ic1(i) = 0;
            ic2(i) = -iL8(i);
            icin(i) = -icin(i);
        end

% Atuação da tensão Vac
    elseif ((Vac(i) > Vab(i)) && (Vac(i) >= Vbc(i)))
        ia1(i) = iLeq(i);
        ia2(i) = iL4(i);
        iain(i) = iain(i);
        if (Vbn(i) >= 0)
            ib1(i) = 0;
            ib2(i) = iL6(i);
            ibin(i) = ibin(i);
        else
            ib1(i) = 0;
            ib2(i) = -iL6(i);
            ibin(i) = -ibin(i);
        end
        ic1(i) = -iLeq(i);
        ic2(i) = -iL8(i);
        icin(i) = -icin(i);

% Atuação da tensão Vbc
    elseif ((Vbc(i) > Vac(i)) && (Vbc(i) >= Vba(i)))
        if (Van(i) >= 0)
            ia1(i) = 0;
            ia2(i) = iL4(i);
            iain(i) = iain(i);
        else
            ia1(i) = 0;
            ia2(i) = -iL4(i);
            iain(i) = -iain(i);
        end
        ib1(i) = iLeq(i);
        ib2(i) = iL6(i);
        ibin(i) = ibin(i);
        ic1(i) = -iLeq(i);
        ic2(i) = -iL8(i);
        icin(i) = -icin(i);

% Atuação da tensão Vba
    elseif ((Vba(i) > Vbc(i)) && (Vba(i) >= Vca(i)))
        ia1(i) = -iLeq(i);
        ia2(i) = -iL4(i);
        iain(i) = -iain(i);
        ib1(i) = iLeq(i);
        ib2(i) = iL6(i);
        ibin(i) = ibin(i);
        if (Vcn(i) >= 0)
            ic1(i) = 0;
            ic2(i) = iL8(i);
            icin(i) = icin(i);
        else
            ic1(i) = 0;
            ic2(i) = -iL8(i);
            icin(i) = -icin(i);
        end

% Atuação da tensão Vca
    elseif ((Vca(i) > Vba(i)) && (Vca(i) >= Vcb(i)))
        ia1(i) = -iLeq(i);
        ia2(i) = -iL4(i);
        iain(i) = -iain(i);
        if (Vbn(i) >= 0)
            ib1(i) = 0;
            ib2(i) = iL6(i);
            ibin(i) = ibin(i);

```

```

else
    ib1(i) = 0;
    ib2(i) = -iL6(i);
    ibin(i) = -ibin(i);
end
ic1(i) = iLeq(i);
ic2(i) = iL8(i);
icin(i) = icin(i);

% Atuação da tensão Vcb
elseif ((Vcb(i) > Vca(i)) && (Vcb(i) >= Vab(i)))
    if (Van(i) >= 0)
        ia1(i) = 0;
        ia2(i) = iL4(i);
        iain(i) = iain(i);
    else
        ia1(i) = 0;
        ia2(i) = -iL4(i);
        iain(i) = -iain(i);
    end
    ib1(i) = -iLeq(i);
    ib2(i) = -iL6(i);
    ibin(i) = -ibin(i);
    ic1(i) = iLeq(i);
    ic2(i) = iL8(i);
    icin(i) = icin(i);
end
Io(i) = vC0(i)/R;

%% Valores Médios Para Cálculo de Potências

Io_area = Io(i) + Io_area;
Io_avg = Io_area/i;
vC0_area = vC0(i) + vC0_area;
vC0_avg = vC0_area/i;
iLeq_area = iLeq(i) + iLeq_area;
iLeq_avg = iLeq_area/i;

%% Cálculos de Potência Instantânea

Ptotal(i) = (vC0_avg^2)/R;
Pret1(i) = iLeq_avg * vC0_avg;
Pret2(i) = Ptotal(i) - Pret1(i);
Pret1_Percent(i) = (Pret1(i)*100)/Ptotal(i);
Pret2_Percent(i) = (Pret2(i)*100)/Ptotal(i);
Ptotal_Percent(i) = Pret1_Percent(i) + Pret2_Percent(i);

end

%% Valores Médios das Tensões

vC0_area = 0;

for i = 1:length(t)
    vC0_area = vC0(i) + vC0_area;
    vC1_area = vC1(i) + vC1_area;
    vC2_area = vC2(i) + vC2_area;
    vC3_area = vC3(i) + vC3_area;
    vC4_area = vC4(i) + vC4_area;
    vC5_area = vC5(i) + vC5_area;
    vC6_area = vC6(i) + vC6_area;
    vS1_area = vS1(i) + vS1_area;
    vS2_area = vS2(i) + vS2_area;
    vS3_area = vS3(i) + vS3_area;
    vD1_area = vD1(i) + vD1_area;
    vD2_area = vD2(i) + vD2_area;
    vD3_area = vD3(i) + vD3_area;
    vD4_area = vD4(i) + vD4_area;
    vD5_area = vD5(i) + vD5_area;
    vD6_area = vD6(i) + vD6_area;
    vD7_area = vD7(i) + vD7_area;
    vD8_area = vD8(i) + vD8_area;
    vD9_area = vD9(i) + vD9_area;
    Vret_area = Vret(i) + Vret_area;
    Va_area = Va(i) + Va_area;
    Vb_area = Vb(i) + Vb_area;
    Vc_area = Vc(i) + Vc_area;
end

vC0_avg = vC0_area/length(t);
vC1_avg = vC1_area/length(t);
vC2_avg = vC2_area/length(t);
vC3_avg = vC3_area/length(t);
vC4_avg = vC4_area/length(t);
vC5_avg = vC5_area/length(t);
vC6_avg = vC6_area/length(t);
vS1_avg = vS1_area/length(t);

```

```

vS2_avg = vS2_area/length(t);
vS3_avg = vS3_area/length(t);
vD1_avg = vD1_area/length(t);
vD2_avg = vD2_area/length(t);
vD3_avg = vD3_area/length(t);
vD4_avg = vD4_area/length(t);
vD5_avg = vD5_area/length(t);
vD6_avg = vD6_area/length(t);
vD7_avg = vD7_area/length(t);
vD8_avg = vD8_area/length(t);
vD9_avg = vD9_area/length(t);
Vret_avg = Vret_area/length(t);
Va_avg = Va_area/length(t);
Vb_avg = Vb_area/length(t);
Vc_avg = Vc_area/length(t);

```

```
%% Valores Médios das Correntes
```

```
Io_area = 0;
```

```

for i = 1:length(t)
    iLeq_area = iLeq(i)+ iLeq_area;
    iL4_area = iL4(i)+ iL4_area;
    iL5_area = iL5(i)+ iL5_area;
    iL6_area = iL6(i)+ iL6_area;
    iL7_area = iL7(i)+ iL7_area;
    iL8_area = iL8(i)+ iL8_area;
    iL9_area = iL9(i)+ iL9_area;
    Io_area = Io(i)+ Io_area;
    iS1_area = iS1(i) + iS1_area;
    iS2_area = iS2(i) + iS2_area;
    iS3_area = iS3(i) + iS3_area;
    iD1_area = iD1(i) + iD1_area;
    iD2_area = iD2(i) + iD2_area;
    iD3_area = iD3(i) + iD3_area;
    iD4_area = iD4(i) + iD4_area;
    iD5_area = iD5(i) + iD5_area;
    iD6_area = iD6(i) + iD6_area;
    iD7_area = iD7(i) + iD7_area;
    iD8_area = iD8(i) + iD8_area;
    iD9_area = iD9(i) + iD9_area;
end

```

```

iLeq_avg = iLeq_area/length(t);
iL4_avg = iL4_area/length(t);
iL5_avg = iL5_area/length(t);
iL6_avg = iL6_area/length(t);
iL7_avg = iL7_area/length(t);
iL8_avg = iL8_area/length(t);
iL9_avg = iL9_area/length(t);
Io_avg = Io_area/length(t);
iS1_avg = iS1_area/length(t);
iS2_avg = iS2_area/length(t);
iS3_avg = iS3_area/length(t);
iD1_avg = iD1_area/length(t);
iD2_avg = iD2_area/length(t);
iD3_avg = iD3_area/length(t);
iD4_avg = iD4_area/length(t);
iD5_avg = iD5_area/length(t);
iD6_avg = iD6_area/length(t);
iD7_avg = iD7_area/length(t);
iD8_avg = iD8_area/length(t);
iD9_avg = iD9_area/length(t);

```

```
%% Valores de Pico das Tensões
```

```

for i = 1:length(t)
    if (vC0(i) >= vC0_pico)
        vC0_pico = vC0(i);
    end
    if (vC1(i) >= vC1_pico)
        vC1_pico = vC1(i);
    end
    if (vC2(i) >= vC2_pico)
        vC2_pico = vC2(i);
    end
    if (vC3(i) >= vC3_pico)
        vC3_pico = vC3(i);
    end
    if (vC4(i) >= vC4_pico)
        vC4_pico = vC4(i);
    end
    if (vC5(i) >= vC5_pico)
        vC5_pico = vC5(i);
    end
    if (vC6(i) >= vC6_pico)
        vC6_pico = vC6(i);
    end
end

```

```
end
if (vS1(i) >= vS1_pico)
    vS1_pico = vS1(i);
end
if (vS2(i) >= vS2_pico)
    vS2_pico = vS2(i);
end
if (vS3(i) >= vS3_pico)
    vS3_pico = vS3(i);
end
if (vD1(i) >= vD1_pico)
    vD1_pico = vD1(i);
end
if (vD2(i) >= vD2_pico)
    vD2_pico = vD2(i);
end
if (vD3(i) >= vD3_pico)
    vD3_pico = vD3(i);
end
if (vD4(i) >= vD4_pico)
    vD4_pico = vD4(i);
end
if (vD5(i) >= vD5_pico)
    vD5_pico = vD5(i);
end
if (vD6(i) >= vD6_pico)
    vD6_pico = vD6(i);
end
if (vD7(i) >= vD7_pico)
    vD7_pico = vD7(i);
end
if (Vret(i) >= Vret_pico)%
    Vret_pico = Vret(i);%
end
if (Va(i) >= Va_pico)%
    Va_pico = Va(i);%
end
if (Vb(i) >= Vb_pico)%
    Vb_pico = Vb(i);%
end
if (Vc(i) >= Vc_pico)%
    Vc_pico = Vc(i);%
end

end

%% Valores de Pico das Correntes

for i = 1:length(t)
    if (iLeq(i) >= iLeq_pico)
        iLeq_pico = iLeq(i);
    end
    if (iL4(i) >= iL4_pico)
        iL4_pico = iL4(i);
    end
    if (iL5(i) >= iL5_pico)
        iL5_pico = iL5(i);
    end
    if (iL6(i) >= iL6_pico)
        iL6_pico = iL6(i);
    end
    if (iL7(i) >= iL7_pico)
        iL7_pico = iL7(i);
    end
    if (iL8(i) >= iL8_pico)
        iL8_pico = iL8(i);
    end
    if (iL9(i) >= iL9_pico)
        iL9_pico = iL9(i);
    end
    if (iS1(i) >= iS1_pico)
        iS1_pico = iS1(i);
    end
    if (iS2(i) >= iS2_pico)
        iS2_pico = iS2(i);
    end
    if (iS3(i) >= iS3_pico)
        iS3_pico = iS3(i);
    end
    if (iD1(i) >= iD1_pico)
        iD1_pico = iD1(i);
    end
    if (iD2(i) >= iD2_pico)
        iD2_pico = iD2(i);
    end
    if (iD3(i) >= iD3_pico)
        iD3_pico = iD3(i);
    end
end
```

```

if (iD4(i) >= iD4_pico)
    iD4_pico = iD4(i);
end
if (iD5(i) >= iD5_pico)
    iD5_pico = iD5(i);
end
if (iD6(i) >= iD6_pico)
    iD6_pico = iD6(i);
end
if (iD7(i) >= iD7_pico)
    iD7_pico = iD7(i);
end
if (ia1(i) >= ia1_pico)
    ia1_pico = ia1(i);
end
if (ia2(i) >= ia2_pico)
    ia2_pico = ia2(i);
end
if (iain(i) >= iain_pico)
    iain_pico = iain(i);
end
if (ib1(i) >= ib1_pico)
    ib1_pico = ib1(i);
end
if (ib2(i) >= ib2_pico)
    ib2_pico = ib2(i);
end
if (ibin(i) >= ibin_pico)
    ibin_pico = ibin(i);
end
if (icl(i) >= icl_pico)
    icl_pico = icl(i);
end
if (ic2(i) >= ic2_pico)
    ic2_pico = ic2(i);
end
if (icin(i) >= icin_pico)
    icin_pico = icin(i);
end
end

%% Valores Eficazes das Tensões

for i = 1:length(t)
    vC0_area2 = vC0(i)^2+ vC0_area2;
    vC1_area2 = vC1(i)^2+ vC1_area2;
    vC2_area2 = vC2(i)^2+ vC2_area2;
    vC3_area2 = vC3(i)^2+ vC3_area2;
    vC4_area2 = vC4(i)^2+ vC4_area2;
    vC5_area2 = vC5(i)^2+ vC5_area2;
    vC6_area2 = vC6(i)^2+ vC6_area2;
    vS1_area2 = vS1(i)^2 + vS1_area2;
    vS2_area2 = vS2(i)^2 + vS2_area2;
    vS3_area2 = vS3(i)^2 + vS3_area2;
    vD1_area2 = vD1(i)^2 + vD1_area2;
    vD2_area2 = vD2(i)^2 + vD2_area2;
    vD3_area2 = vD3(i)^2 + vD3_area2;
    vD4_area2 = vD4(i)^2 + vD4_area2;
    vD5_area2 = vD5(i)^2 + vD5_area2;
    vD6_area2 = vD6(i)^2 + vD6_area2;
    vD7_area2 = vD7(i)^2 + vD7_area2;
    vD8_area2 = vD8(i)^2 + vD8_area2;
    vD9_area2 = vD9(i)^2 + vD9_area2;
    Vret_area2 = Vret(i)^2 + Vret_area2;
    Va_area2 = Va(i)^2 + Va_area2;
    Vb_area2 = Vb(i)^2 + Vb_area2;
    Vc_area2 = Vc(i)^2 + Vc_area2;
end

vC0_rms = (2*vC0_area2/length(t))^0.5;
vC1_rms = (2*vC1_area2/length(t))^0.5;
vC2_rms = (2*vC2_area2/length(t))^0.5;
vC3_rms = (2*vC3_area2/length(t))^0.5;
vC4_rms = (2*vC4_area2/length(t))^0.5;
vC5_rms = (2*vC5_area2/length(t))^0.5;
vC6_rms = (2*vC6_area2/length(t))^0.5;
vS1_rms = (2*vS1_area2/length(t))^0.5;
vS2_rms = (2*vS2_area2/length(t))^0.5;
vS3_rms = (2*vS3_area2/length(t))^0.5;
vD1_rms = (2*vD1_area2/length(t))^0.5;
vD2_rms = (2*vD2_area2/length(t))^0.5;
vD3_rms = (2*vD3_area2/length(t))^0.5;
vD4_rms = (2*vD4_area2/length(t))^0.5;
vD5_rms = (2*vD5_area2/length(t))^0.5;
vD6_rms = (2*vD6_area2/length(t))^0.5;
vD7_rms = (2*vD7_area2/length(t))^0.5;
vD8_rms = (2*vD8_area2/length(t))^0.5;

```



```
vD9_rms = (2*vD9_area2/length(t))^0.5;
Vret_rms = (2*Vret_area2/length(t))^0.5;
Va_rms = (2*Va_area2/length(t))^0.5;
Vb_rms = (2*Vb_area2/length(t))^0.5;
Vc_rms = (2*Vc_area2/length(t))^0.5;

%% Valores Eficazes das Correntes

for i = 1:length(t)
    iLeq_area2 = iLeq(i)^2+ iLeq_area2;
    iL4_area2 = iL4(i)^2+ iL4_area2;
    iL5_area2 = iL5(i)^2+ iL5_area2;
    iL6_area2 = iL6(i)^2+ iL6_area2;
    iL7_area2 = iL7(i)^2+ iL7_area2;
    iL8_area2 = iL8(i)^2+ iL8_area2;
    iL9_area2 = iL9(i)^2+ iL9_area2;
    Io_area2 = Io(i)^2+ Io_area2;
    iS1_area2 = iS1(i)^2 + iS1_area2;
    iS2_area2 = iS2(i)^2 + iS2_area2;
    iS3_area2 = iS3(i)^2 + iS3_area2;
    iD1_area2 = iD1(i)^2 + iD1_area2;
    iD2_area2 = iD2(i)^2 + iD2_area2;
    iD3_area2 = iD3(i)^2 + iD3_area2;
    iD4_area2 = iD4(i)^2 + iD4_area2;
    iD5_area2 = iD5(i)^2 + iD5_area2;
    iD6_area2 = iD6(i)^2 + iD6_area2;
    iD7_area2 = iD7(i)^2 + iD7_area2;
    iD8_area2 = iD8(i)^2 + iD8_area2;
    iD9_area2 = iD9(i)^2 + iD9_area2;
    ia1_area2 = ia1(i)^2 + ia1_area2;
    ia2_area2 = ia2(i)^2 + ia2_area2;
    iain_area2 = iain(i)^2 + iain_area2;
    ib1_area2 = ib1(i)^2 + ib1_area2;
    ib2_area2 = ib2(i)^2 + ib2_area2;
    ibin_area2 = ibin(i)^2 + ibin_area2;
    ic1_area2 = ic1(i)^2 + ic1_area2;
    ic2_area2 = ic2(i)^2 + ic2_area2;
    icin_area2 = icin(i)^2 + icin_area2;
    iC0_area2 = iC0(i)^2 + iC0_area2;%
end

iLeq_rms = (2*iLeq_area2/length(t))^0.5;
iL4_rms = (2*iL4_area2/length(t))^0.5;
iL5_rms = (2*iL5_area2/length(t))^0.5;
iL6_rms = (2*iL6_area2/length(t))^0.5;
iL7_rms = (2*iL7_area2/length(t))^0.5;
iL8_rms = (2*iL8_area2/length(t))^0.5;
iL9_rms = (2*iL9_area2/length(t))^0.5;
Io_rms = (2*Io_area2/length(t))^0.5;
iS1_rms = (2*iS1_area2/length(t))^0.5;
iS2_rms = (2*iS2_area2/length(t))^0.5;
iS3_rms = (2*iS3_area2/length(t))^0.5;
iD1_rms = (2*iD1_area2/length(t))^0.5;
iD2_rms = (2*iD2_area2/length(t))^0.5;
iD3_rms = (2*iD3_area2/length(t))^0.5;
iD4_rms = (2*iD4_area2/length(t))^0.5;
iD5_rms = (2*iD5_area2/length(t))^0.5;
iD6_rms = (2*iD6_area2/length(t))^0.5;
iD7_rms = (2*iD7_area2/length(t))^0.5;
iD8_rms = (2*iD8_area2/length(t))^0.5;
iD9_rms = (2*iD9_area2/length(t))^0.5;
ia1_rms = (2*ia1_area2/length(t))^0.5;
ia2_rms = (2*ia2_area2/length(t))^0.5;
iain_rms = (2*iain_area2/length(t))^0.5;
ib1_rms = (2*ib1_area2/length(t))^0.5;
ib2_rms = (2*ib2_area2/length(t))^0.5;
ibin_rms = (2*ibin_area2/length(t))^0.5;
ic1_rms = (2*ic1_area2/length(t))^0.5;
ic2_rms = (2*ic2_area2/length(t))^0.5;
icin_rms = (2*icin_area2/length(t))^0.5;
iC0_rms = (2*iC0_area2/length(t))^0.5;%

%% Cálculos de Potência Média

Ptotalmed = (vC0_avg^2)/R;
Pretlmed = iLeq_avg * vC0_avg;
Pret2med = Ptotalmed - Pretlmed;
Pretlmed_Percent = (Pretlmed*100)/Ptotalmed;
Pret2med_Percent = (Pret2med*100)/Ptotalmed;
Ptotalmed_Percent = Pretlmed_Percent + Pret2med_Percent;

%% Cálculo THD

H = 25;
W = 2*pi*f;
An_norma = [0 14.4 0 7.1 0 4.8 0 2.5 0 2.1 0 1.3 0 0.94 0 0.82 0 0.74 0 0.67 0 0.61 0 0.56]; %3-2
num_ciclos = 1;
tam_ciclo = roundn(((num_ciclos*T)/(deltaT)),0)
```

```

Pontos_total = roundn((tmax/deltaT),0)

% ----- THD para Iain-----
ondaA = iain;

a0a = 1/T*trapz(t(1:length(ondaA)),ondaA);

%Coefs. an e bn para n > 0
aa = [];
ba = [];
na = 1; % fundamental pra frente e somente as ímparares
while na<=H
    aa(na) = 1/T*trapz(t(1:length(ondaA)),ondaA.*cos(na*2*pi*f*t(1:length(ondaA))));
    ba(na) = 1/T*trapz(t(1:length(ondaA)),ondaA.*sin(na*2*pi*f*t(1:length(ondaA))));
    na = na + 2;
end

AnA = sqrt(aa.^2 + ba.^2);
An_percentagem_A = (AnA(2:H)*100)/AnA(1);
FinA = (-atan(ba/aa) + pi/2)*180/pi; %+pi/2 para passar para seno
thd_Iain = sqrt(sum(AnA(2:length(AnA)).^2))/AnA(1)*100;

% ----- THD para Ibin -----
ondaB = ibin;

a0b = 1/T*trapz(t(1:length(ondaB)),ondaB);

%Coefs. an e bn para n > 0
ab = [];
bb = [];
nb = 1; % fundamental pra frente e somente as ímparares
while nb<=H
    ab(nb) = 1/T*trapz(t(1:length(ondaB)),ondaB.*cos(nb*2*pi*f*t(1:length(ondaB))));
    bb(nb) = 1/T*trapz(t(1:length(ondaB)),ondaB.*sin(nb*2*pi*f*t(1:length(ondaB))));
    nb = nb + 2;
end

AnB = sqrt(ab.^2 + bb.^2);
An_percentagem_B = (AnB(2:H)*100)/AnB(1);
FinB = (-atan(bb/ab) + pi/2)*180/pi; %+pi/2 para passar para seno
thd_Ibin = sqrt(sum(AnB(2:length(AnB)).^2))/AnB(1)*100;

% ----- THD para Icin -----
ondaC = icin;

a0c = 1/T*trapz(t(1:length(ondaC)),ondaC);

%Coefs. an e bn para n > 0
ac = [];
bc = [];
nc = 1; % fundamental pra frente e somente as ímparares
while nc<=H
    ac(nc) = 1/T*trapz(t(1:length(ondaC)),ondaC.*cos(nc*2*pi*f*t(1:length(ondaC))));
    bc(nc) = 1/T*trapz(t(1:length(ondaC)),ondaC.*sin(nc*2*pi*f*t(1:length(ondaC))));
    nc = nc + 2;
end

AnC = sqrt(ac.^2 + bc.^2);
An_percentagem_C = (AnC(2:H)*100)/AnC(1);
FinC = (-atan(bc/ac) + pi/2)*180/pi; %+pi/2 para passar para seno
thd_Icin = sqrt(sum(AnC(2:length(AnC)).^2))/AnC(1)*100;

%% Mostra valores na tela
clc
fprintf('A Matriz de Sistema para Sa(ON), Sb(ON) e Sc(ON) é:\n')
A1
fprintf('\n\nA Matriz de Sistema para Sa(ON), Sb(ON) e Sc(OFF) é:\n')
A2
fprintf('\n\nA Matriz de Sistema para Sa(ON), Sb(OFF) e Sc(ON) é:\n')
A3
fprintf('\n\nA Matriz de Sistema para Sa(ON), Sb(OFF) e Sc(OFF) é:\n')
A4
fprintf('\n\nA Matriz de Sistema para Sa(OFF), Sb(ON) e Sc(ON) é:\n')
A5
fprintf('\n\nA Matriz de Sistema para Sa(OFF), Sb(ON) e Sc(OFF) é:\n')
A6
fprintf('\n\nA Matriz de Sistema para Sa(OFF), Sb(OFF) e Sc(ON) é:\n')
A7
fprintf('\n\nA Matriz de Sistema para Sa(OFF), Sb(OFF) e Sc(OFF) é:\n')
A8
fprintf('\n\nA Matriz de Entrada para todas as condições de funcionamento das chaves dos conversores
SEPIC é:\n')
B = B1
fprintf('\n\nValor médio da corrente em Leq: iLeq(med) = %.4f A\n',iLeq_avg)
fprintf('\n\nValor médio da corrente em L4: iL4(med) = %.4f A\n',iL4_avg)
fprintf('\n\nValor médio da corrente em L5: iL5(med) = %.4f A\n',iL5_avg)
fprintf('\n\nValor médio da corrente em L6: iL6(med) = %.4f A\n',iL6_avg)
fprintf('\n\nValor médio da corrente em L7: iL7(med) = %.4f A\n',iL7_avg)

```

```

fprintf('\n\nValor médio da corrente em L8: iL8(med) = %.4f A\n', iL8_avg)
fprintf('\n\nValor médio da corrente em L9: iL9(med) = %.4f A\n', iL9_avg)
fprintf('\n\nValor médio da corrente em S1: iS1(med) = %.4f A\n', iS1_avg)
fprintf('\n\nValor médio da corrente em S1: iS1(med) = %.4f A\n', iS1_avg)
fprintf('\n\nValor médio da corrente em S2: iS2(med) = %.4f A\n', iS2_avg)
fprintf('\n\nValor médio da corrente em S3: iS3(med) = %.4f A\n', iS3_avg)
fprintf('\n\nValor médio da corrente em D1: iD1(med) = %.4f A\n', iD1_avg)
fprintf('\n\nValor médio da corrente em D2: iD2(med) = %.4f A\n', iD2_avg)
fprintf('\n\nValor médio da corrente em D3: iD3(med) = %.4f A\n', iD3_avg)
fprintf('\n\nValor médio da corrente em D4: iD4(med) = %.4f A\n', iD4_avg)
fprintf('\n\nValor médio da corrente em D5: iD5(med) = %.4f A\n', iD5_avg)
fprintf('\n\nValor médio da corrente em D6: iD6(med) = %.4f A\n', iD6_avg)
fprintf('\n\nValor médio da corrente em D7: iD7(med) = %.4f A\n', iD7_avg)
fprintf('\n\nValor médio da corrente em D8: iD8(med) = %.4f A\n', iD8_avg)
fprintf('\n\nValor médio da corrente em D9: iD9(med) = %.4f A\n', iD9_avg)
fprintf('\n\nValor médio da corrente na Carga: Io = %.4f A\n', Io_avg)
fprintf('\n\nValor de pico da corrente em Leq: iLeq(pico) = %.4f A\n', iLeq_pico)
fprintf('\n\nValor de pico da corrente em L4: iL4(pico) = %.4f A\n', iL4_pico)
fprintf('\n\nValor de pico da corrente em L5: iL5(pico) = %.4f A\n', iL5_pico)
fprintf('\n\nValor de pico da corrente em L6: iL6(pico) = %.4f A\n', iL6_pico)
fprintf('\n\nValor de pico da corrente em L7: iL7(pico) = %.4f A\n', iL7_pico)
fprintf('\n\nValor de pico da corrente em L8: iL8(pico) = %.4f A\n', iL8_pico)
fprintf('\n\nValor de pico da corrente em L9: iL9(pico) = %.4f A\n', iL9_pico)
fprintf('\n\nValor de pico da corrente em S1: iS1(pico) = %.4f A\n', iS1_pico)
fprintf('\n\nValor de pico da corrente em S1: iS1(pico) = %.4f A\n', iS1_pico)
fprintf('\n\nValor de pico da corrente em S2: iS2(pico) = %.4f A\n', iS2_pico)
fprintf('\n\nValor de pico da corrente em S3: iS3(pico) = %.4f A\n', iS3_pico)
fprintf('\n\nValor de pico da corrente em D1: iD1(pico) = %.4f A\n', iD1_pico)
fprintf('\n\nValor de pico da corrente em D2: iD2(pico) = %.4f A\n', iD2_pico)
fprintf('\n\nValor de pico da corrente em D3: iD3(pico) = %.4f A\n', iD3_pico)
fprintf('\n\nValor de pico da corrente em D4: iD4(pico) = %.4f A\n', iD4_pico)
fprintf('\n\nValor de pico da corrente em D5: iD5(pico) = %.4f A\n', iD5_pico)
fprintf('\n\nValor de pico da corrente em D6: iD6(pico) = %.4f A\n', iD6_pico)
fprintf('\n\nValor de pico da corrente em D7: iD7(pico) = %.4f A\n', iD7_pico)
fprintf('\n\nValor de pico da corrente em D8: iD8(pico) = %.4f A\n', iD8_pico)
fprintf('\n\nValor de pico da corrente em D9: iD9(pico) = %.4f A\n', iD9_pico)
fprintf('\n\nValor de pico da corrente na fase A do Ret-1: ial(pico) = %.4f A\n', ial_pico)
fprintf('\n\nValor de pico da corrente no SEPIC da fase A: ia2(pico) = %.4f A\n', ia2_pico)
fprintf('\n\nValor de pico da corrente de linha de entrada da fase A: iain(pico) = %.4f A\n', iain_pico)
fprintf('\n\nValor de pico da corrente na fase B do Ret-1: ib1(pico) = %.4f A\n', ib1_pico)
fprintf('\n\nValor de pico da corrente no SEPIC da fase B: ib2(pico) = %.4f A\n', ib2_pico)
fprintf('\n\nValor de pico da corrente de linha de entrada da fase B: ibin(pico) = %.4f A\n', ibin_pico)
fprintf('\n\nValor de pico da corrente na fase C do Ret-1: ic1(pico) = %.4f A\n', ic1_pico)
fprintf('\n\nValor de pico da corrente no SEPIC da fase C: ic2(pico) = %.4f A\n', ic2_pico)
fprintf('\n\nValor de pico da corrente de linha de entrada da fase C: icin(pico) = %.4f A\n', icin_pico)
fprintf('\n\nValor eficaz da corrente em Leq: iLeq(rms) = %.4f A\n', iLeq_rms)
fprintf('\n\nValor eficaz da corrente em L4: iL4(rms) = %.4f A\n', iL4_rms)
fprintf('\n\nValor eficaz da corrente em L5: iL5(rms) = %.4f A\n', iL5_rms)
fprintf('\n\nValor eficaz da corrente em L6: iL6(rms) = %.4f A\n', iL6_rms)
fprintf('\n\nValor eficaz da corrente em L7: iL7(rms) = %.4f A\n', iL7_rms)
fprintf('\n\nValor eficaz da corrente em L8: iL8(rms) = %.4f A\n', iL8_rms)
fprintf('\n\nValor eficaz da corrente em L9: iL9(rms) = %.4f A\n', iL9_rms)
fprintf('\n\nValor eficaz da corrente em S1: iS1(rms) = %.4f A\n', iS1_rms)
fprintf('\n\nValor eficaz da corrente em S1: iS1(rms) = %.4f A\n', iS1_rms)
fprintf('\n\nValor eficaz da corrente em S2: iS2(rms) = %.4f A\n', iS2_rms)
fprintf('\n\nValor eficaz da corrente em S3: iS3(rms) = %.4f A\n', iS3_rms)
fprintf('\n\nValor eficaz da corrente em D1: iD1(rms) = %.4f A\n', iD1_rms)
fprintf('\n\nValor eficaz da corrente em D2: iD2(rms) = %.4f A\n', iD2_rms)
fprintf('\n\nValor eficaz da corrente em D3: iD3(rms) = %.4f A\n', iD3_rms)
fprintf('\n\nValor eficaz da corrente em D4: iD4(rms) = %.4f A\n', iD4_rms)
fprintf('\n\nValor eficaz da corrente em D5: iD5(rms) = %.4f A\n', iD5_rms)
fprintf('\n\nValor eficaz da corrente em D6: iD6(rms) = %.4f A\n', iD6_rms)
fprintf('\n\nValor eficaz da corrente em D7: iD7(rms) = %.4f A\n', iD7_rms)
fprintf('\n\nValor eficaz da corrente em D8: iD8(rms) = %.4f A\n', iD8_rms)
fprintf('\n\nValor eficaz da corrente em D9: iD9(rms) = %.4f A\n', iD9_rms)
fprintf('\n\nValor de eficaz da corrente na fase A do Ret-1: ial(rms) = %.4f A\n', ial_rms)
fprintf('\n\nValor de eficaz da corrente no SEPIC da fase A: ia2(rms) = %.4f A\n', ia2_rms)
fprintf('\n\nValor de eficaz da corrente de linha de entrada da fase A: iain(rms) = %.4f A\n', iain_rms)
fprintf('\n\nValor de eficaz da corrente na fase B do Ret-1: ib1(rms) = %.4f A\n', ib1_rms)
fprintf('\n\nValor de eficaz da corrente no SEPIC da fase B: ib2(rms) = %.4f A\n', ib2_rms)
fprintf('\n\nValor de eficaz da corrente de linha de entrada da fase B: ibin(rms) = %.4f A\n', ibin_rms)
fprintf('\n\nValor de eficaz da corrente na fase C do Ret-1: ic1(rms) = %.4f A\n', ic1_rms)
fprintf('\n\nValor de eficaz da corrente no SEPIC da fase C: ic2(rms) = %.4f A\n', ic2_rms)
fprintf('\n\nValor de eficaz da corrente de linha de entrada da fase C: icin(rms) = %.4f A\n', icin_rms)
fprintf('\n\nValor eficaz da corrente no capacitor C0: iC0(rms) = %.4f A\n', iC0_rms)
fprintf('\n\nValor médio da tensão em C0: vC0(med) = %.4f V\n', vC0_avg)
fprintf('\n\nValor médio da tensão em C1: vC1(med) = %.4f V\n', vC1_avg)
fprintf('\n\nValor médio da tensão em C2: vC2(med) = %.4f V\n', vC2_avg)
fprintf('\n\nValor médio da tensão em C3: vC3(med) = %.4f V\n', vC3_avg)
fprintf('\n\nValor médio da tensão em C4: vC4(med) = %.4f V\n', vC4_avg)
fprintf('\n\nValor médio da tensão em C5: vC5(med) = %.4f V\n', vC5_avg)
fprintf('\n\nValor médio da tensão em C6: vC6(med) = %.4f V\n', vC6_avg)

```

```

fprintf('\n\nValor médio da tensão em S1: vS1( med) = %.4f V\n',vS1_avg)
fprintf('\n\nValor médio da tensão em S1: vS1( med) = %.4f V\n',vS1_avg)
fprintf('\n\nValor médio da tensão em S2: vS2( med) = %.4f V\n',vS2_avg)
fprintf('\n\nValor médio da tensão em S3: vS3( med) = %.4f V\n',vS3_avg)
fprintf('\n\nValor médio da tensão em D1: vD1( med) = %.4f V\n',vD1_avg)
fprintf('\n\nValor médio da tensão em D2: vD2( med) = %.4f V\n',vD2_avg)
fprintf('\n\nValor médio da tensão em D3: vD3( med) = %.4f V\n',vD3_avg)
fprintf('\n\nValor médio da tensão em D4: vD4( med) = %.4f V\n',vD4_avg)
fprintf('\n\nValor médio da tensão em D5: vD5( med) = %.4f V\n',vD5_avg)
fprintf('\n\nValor médio da tensão em D6: vD6( med) = %.4f V\n',vD6_avg)
fprintf('\n\nValor médio da tensão em D7: vD7( med) = %.4f V\n',vD7_avg)
fprintf('\n\nValor médio da tensão em D8: vD8( med) = %.4f V\n',vD8_avg)
fprintf('\n\nValor médio da tensão em D9: vD9( med) = %.4f V\n',vD9_avg)
fprintf('\n\nValor médio da tensão na ponte RT: Vret( med) = %.4f A\n',Vret_avg)
fprintf('\n\nValor médio da tensão na ponte RMa: Va( med) = %.4f A\n',Va_avg)
fprintf('\n\nValor médio da tensão na ponte RMb: Vb( med) = %.4f A\n',Vb_avg)
fprintf('\n\nValor médio da tensão na ponte RMc: Vc( med) = %.4f A\n',Vc_avg)
fprintf('\n\nValor de pico da tensão em C0: vC0( pico) = %.4f V\n',vC0_pico)
fprintf('\n\nValor de pico da tensão em C1: vC1( pico) = %.4f V\n',vC1_pico)
fprintf('\n\nValor de pico da tensão em C2: vC2( pico) = %.4f V\n',vC2_pico)
fprintf('\n\nValor de pico da tensão em C3: vC3( pico) = %.4f V\n',vC3_pico)
fprintf('\n\nValor de pico da tensão em C4: vC4( pico) = %.4f V\n',vC4_pico)
fprintf('\n\nValor de pico da tensão em C5: vC5( pico) = %.4f V\n',vC5_pico)
fprintf('\n\nValor de pico da tensão em C6: vC6( pico) = %.4f V\n',vC6_pico)
fprintf('\n\nValor de pico da tensão em S1: vS1( pico) = %.4f V\n',vS1_pico)
fprintf('\n\nValor de pico da tensão em S1: vS1( pico) = %.4f V\n',vS1_pico)
fprintf('\n\nValor de pico da tensão em S2: vS2( pico) = %.4f V\n',vS2_pico)
fprintf('\n\nValor de pico da tensão em S3: vS3( pico) = %.4f V\n',vS3_pico)
fprintf('\n\nValor de pico da tensão em D1: vD1( pico) = %.4f V\n',vD1_pico)
fprintf('\n\nValor de pico da tensão em D2: vD2( pico) = %.4f V\n',vD2_pico)
fprintf('\n\nValor de pico da tensão em D3: vD3( pico) = %.4f V\n',vD3_pico)
fprintf('\n\nValor de pico da tensão em D4: vD4( pico) = %.4f V\n',vD4_pico)
fprintf('\n\nValor de pico da tensão em D5: vD5( pico) = %.4f V\n',vD5_pico)
fprintf('\n\nValor de pico da tensão em D6: vD6( pico) = %.4f V\n',vD6_pico)
fprintf('\n\nValor de pico da tensão em D7: vD7( pico) = %.4f V\n',vD7_pico)
fprintf('\n\nValor de pico da tensão em D8: vD8( pico) = %.4f V\n',vD8_pico)
fprintf('\n\nValor de pico da tensão em D9: vD9( pico) = %.4f V\n',vD9_pico)
fprintf('\n\nValor de pico da tensão na ponte RT: Vret( pico) = %.4f A\n',Vret_pico)
fprintf('\n\nValor de pico da tensão na ponte RMa: Va( pico) = %.4f A\n',Va_pico)
fprintf('\n\nValor de pico da tensão na ponte RMb: Vb( pico) = %.4f A\n',Vb_pico)
fprintf('\n\nValor de pico da tensão na ponte RMc: Vc( pico) = %.4f A\n',Vc_pico)
fprintf('\n\nValor eficaz da tensão em C0: vC0( rms) = %.4f V\n',vC0_rms)
fprintf('\n\nValor eficaz da tensão em C1: vC1( rms) = %.4f V\n',vC1_rms)
fprintf('\n\nValor eficaz da tensão em C2: vC2( rms) = %.4f V\n',vC2_rms)
fprintf('\n\nValor eficaz da tensão em C3: vC3( rms) = %.4f V\n',vC3_rms)
fprintf('\n\nValor eficaz da tensão em C4: vC4( rms) = %.4f V\n',vC4_rms)
fprintf('\n\nValor eficaz da tensão em C5: vC5( rms) = %.4f V\n',vC5_rms)
fprintf('\n\nValor eficaz da tensão em C6: vC6( rms) = %.4f V\n',vC6_rms)
fprintf('\n\nValor eficaz da tensão em S1: vS1( rms) = %.4f V\n',vS1_rms)
fprintf('\n\nValor eficaz da tensão em S1: vS1( rms) = %.4f V\n',vS1_rms)
fprintf('\n\nValor eficaz da tensão em S2: vS2( rms) = %.4f V\n',vS2_rms)
fprintf('\n\nValor eficaz da tensão em S3: vS3( rms) = %.4f V\n',vS3_rms)
fprintf('\n\nValor eficaz da tensão em D1: vD1( rms) = %.4f V\n',vD1_rms)
fprintf('\n\nValor eficaz da tensão em D2: vD2( rms) = %.4f V\n',vD2_rms)
fprintf('\n\nValor eficaz da tensão em D3: vD3( rms) = %.4f V\n',vD3_rms)
fprintf('\n\nValor eficaz da tensão em D4: vD4( rms) = %.4f V\n',vD4_rms)
fprintf('\n\nValor eficaz da tensão em D5: vD5( rms) = %.4f V\n',vD5_rms)
fprintf('\n\nValor eficaz da tensão em D6: vD6( rms) = %.4f V\n',vD6_rms)
fprintf('\n\nValor eficaz da tensão em D7: vD7( rms) = %.4f V\n',vD7_rms)
fprintf('\n\nValor eficaz da tensão em D8: vD8( rms) = %.4f V\n',vD8_rms)
fprintf('\n\nValor eficaz da tensão em D9: vD9( rms) = %.4f V\n',vD9_rms)
fprintf('\n\nValor eficaz da tensão na ponte RT: Vret( rms) = %.4f A\n',Vret_rms)
fprintf('\n\nValor eficaz da tensão na ponte RMa: Va( rms) = %.4f A\n',Va_rms)
fprintf('\n\nValor eficaz da tensão na ponte RMb: Vb( rms) = %.4f A\n',Vb_rms)
fprintf('\n\nValor eficaz da tensão na ponte RMc: Vc( rms) = %.4f A\n',Vc_rms)
fprintf('\n\nPotência ativa média total processada pelo Retificador Híbrido Trifásico: Prht = %.4f W\n',Ptotalmed)
fprintf('\n\nPotência ativa média processada pelo Retificador Não Controlado: Pret-1 = %.4f W\n',Pretlmed)
fprintf('\n\nPotência ativa média processada pelos conversores SEPIC: Pret-2 = %.4f W\n',Pret2med)
fprintf('\n\nPercentual da potência ativa média processada pelo Retificador Não Controlado em relação o Retificador Híbrido Trifásico: Pret-1[%%] = %.4f %%\n',Pretlmed_Percent)
fprintf('\n\nPercentual da potência ativa média processada pelos conversores SEPIC em relação o Retificador Híbrido Trifásico: Pret-2[%%] = %.4f %%\n',Pret2med_Percent)
fprintf('\n\nDistorção harmônica da corrente de linha de entrada da Fase A: DHTIa = %.4f %%\n',thd_Iain)
fprintf('\n\nDistorção harmônica da corrente de linha de entrada da Fase B: DHTIb = %.4f %%\n',thd_Ibin)
fprintf('\n\nDistorção harmônica da corrente de linha de entrada da Fase C: DHTIc = %.4f %%\n',thd_Icin)
%% Gráficos

figure
subplot(3,1,1);
plot( t,iain,'blue');
legend('Ia(in)');
title('Fase A - Corrente de Linha de Entrada');
xlabel('Tempo');

```

```

ylabel('Corrente [A]');
grid on;
subplot(3,1,2);
plot (t,ial,'blue');
legend('Ia1');
title('Fase A - Corrente do Retificador Não Controlado');
xlabel('Tempo');
ylabel('Corrente [A]');
grid on;
subplot(3,1,3);
plot (t,ia2,'blue');
legend('Ia2');
title('Fase A - Corrente no Conversor SEPIC');
xlabel('Tempo');
ylabel('Corrente [A]');
grid on;

figure
subplot(3,1,1);
plot (t,ibin,'red');
legend('Ib(in)');
title('Fase B - Corrente de Linha de Entrada');
xlabel('Tempo');
ylabel('Corrente [A]');
grid on;
subplot(3,1,2);
plot (t,ibl,'red');
legend('Ib1');
title('Fase B - Corrente do Retificador Não Controlado');
xlabel('Tempo');
ylabel('Corrente [A]');
grid on;
subplot(3,1,3);
plot (t,ib2,'red');
legend('Ib2');
title('Fase B - Corrente no Conversor SEPIC');
xlabel('Tempo');
ylabel('Corrente [A]');
grid on;

figure
subplot(3,1,1);
plot (t,icin,'green');
legend('Ic(in)');
title('Fase C - Corrente de Linha de Entrada');
xlabel('Tempo');
ylabel('Corrente [A]');
grid on;
subplot(3,1,2);
plot (t,icl,'green');
legend('Ic1');
title('Fase C - Corrente do Retificador Não Controlado');
xlabel('Tempo');
ylabel('Corrente [A]');
grid on;
subplot(3,1,3);
plot (t,ic2,'green');
legend('Ic2');
title('Fase C - Corrente no Conversor SEPIC');
xlabel('Tempo');
ylabel('Corrente [A]');
grid on;

figure
plot(t,iain,t,Van/5);
legend('Ia(in)', 'Van/5');
title('Fase A - Corrente de Linha de Entrada e Tensão de Fase');
xlabel('Tempo');
grid on;

figure
plot(t,ibin,t,Vbn/5);
legend('Ib(in)', 'Vbn/5');
title('Fase B - Corrente de Linha de Entrada e Tensão de Fase');
xlabel('Tempo');
grid on;

figure
plot(t,icin,t,Vcn/5);
legend('Ic(in)', 'Vcn/5');
title('Fase C - Corrente de Linha de Entrada e Tensão de Fase');
xlabel('Tempo');
grid on;

figure
plot(t,Vab,t,Vba,t,Vac,t,Vca,t,Vcb,t,Vbc,t,vC0);
legend('Vab', 'Vba', 'Vac', 'Vca', 'Vcb', 'Vbc', 'Vo');
title('Tensões de Linha e Tensão na Carga');

```

```

xlabel('Tempo');
grid on;

figure
plot(t,iLeq,t,Io);
legend('iRet-1','Io');
title('Correntes em Ret-1 e na Carga');
xlabel('Tempo');
grid on;

figure
plot(t,Ptotal,t,Pret1,t,Pret2);
legend('Potência Total','Potência Ret-1','Potência Ret-2');
title('Processamento de Potência Ativa');
xlabel('Tempo');
ylabel('Potência [W]');
grid on;

figure
plot(t,Ptotal_Percent,t,Pret1_Percent,t,Pret2_Percent);
legend('[%]Potência Total','[%]Potência Ret-1','[%]Potência Ret-2');
title('Percentual de Processamento de Potência Ativa');
xlabel('Tempo');
ylabel('Percentual da Potência Total [%]');
grid on;

figure
hold off
bar(An_percentagem_A,1.3);
hold on;
bar(An_norma,'r');
legend('RH Trifásico Proposto - Fase A','Norma IEC 61000 - 3 - 2')
title('Espectro de Frequência');
xlabel('Ordem Harmônica');
ylabel('Amplitude Harmônica em Relação à Fundamental [%]');
La = num2str(thd_Iain);
La = strcat('THD = ',La);
text(length(An_percentagem_A)/2,max(An_percentagem_A),La);

figure
hold off
bar(An_percentagem_B,1.3);
hold on;
bar(An_norma,'r');
legend('RH Trifásico Proposto - Fase B','Norma IEC 61000 - 3 - 2')
title('Espectro de Frequência');
xlabel('Ordem Harmônica');
ylabel('Amplitude Harmônica em Relação à Fundamental [%]');
Lb = num2str(thd_Ibin);
Lb = strcat('THD = ',Lb);
text(length(An_percentagem_B)/2,max(An_percentagem_B),Lb);

figure
hold off
bar(An_percentagem_C,1.3);
hold on;
bar(An_norma,'r');
legend('RH Trifásico Proposto - Fase C','Norma IEC 61000 - 3 - 2')
title('Espectro de Frequência');
xlabel('Ordem Harmônica');
ylabel('Amplitude Harmônica em Relação à Fundamental [%]');
Lc = num2str(thd_Icin);
Lc = strcat('THD = ',Lc);
text(length(An_percentagem_C)/2,max(An_percentagem_C),Lc);

```

Apêndice B – Algoritmo de Controle

```
1. //Inclusão das bibliotecas
   #include "DSP28x_Project.h"

2. // Funções carregadas.
   interrupt void adc_isr(void);

   //Definições de Valores Fixos.
3. #define ADC_CKPS 0x0 // ADC module clock = HSPCLK/2*ADC_CKPS = 25.0MHz/(1*2) = 12.5MHz.
4. #define BUFSIZE 1667 //833 para meio ciclo.
5. #define k 1.113 //Ganho da referência.

   //Declaração de Variáveis Globais.
6. Uint32 cont=0; //Contador Geral.
7. int16 lo =0; //Corrente de carga adquirida.
8. float lo_area=0;
9. float lo_med=1; //Valor médio da corrente de carga.
10. int16 linA=0; //Corrente de entrada da fase A adquirida.
11. int16 linB=0; //Corrente de entrada da fase B adquirida.
12. int16 linC=0; //Corrente de entrada da fase C adquirida.
13. int16 VinA=0; //Variável provida de aquisição da tensão de entrada da fase A utilizada para detecção da passagem por zero.
14. float VinA_pass; //Variável provida de aquisição da tensão de entrada da fase A no período de amostragem passado.
   // Senoide gerada no Matlab®.
15. float Produto=1; //Multiplicação de VinA*VinA_ant.
16. int16 cond=0;
17. float IRefA=0; //Referência da fase A após produtos (Isen*lo_avg*K).
18. float IRefB=0; //Referência da fase B após produtos (Isen*lo_avg*K).
19. float IRefC=0; //Referência da fase C após produtos (Isen*lo_avg*K)..
20. float Isen[] = {0, 3.769902e-003, 7.539751e-003, 1.130949e-002, 1.507907e-002, 1.884844e-002, 2.261754e-002, 2.638632e-002, 3.015472e-002, 3.392269e-002, 3.769018e-002, 4.145714e-002, 4.522351e-002, 4.898923e-002, 5.275426e-002, 5.651853e-002, 6.028201e-002, 6.404463e-002, 6.780633e-002, 7.156708e-002, 7.532681e-002, 7.908546e-002, 8.284299e-002, 8.659935e-002, 9.035447e-002, 9.410831e-002, 9.786082e-002, 1.016119e-001, 1.053616e-001, 1.091098e-001, 1.128564e-001, 1.166014e-001, 1.203448e-001, 1.240864e-001, 1.278263e-001, 1.315644e-001, 1.353006e-001, 1.390348e-001, 1.427671e-001, 1.464974e-001, 1.502256e-001, 1.539516e-001, 1.576755e-001, 1.613971e-001, 1.651165e-001, 1.688334e-001, 1.725480e-001, 1.762602e-001, 1.799698e-001, 1.836769e-001, 1.873813e-001, 1.910831e-001, 1.947822e-001, 1.984785e-001, 2.021720e-001, 2.058626e-001, 2.095503e-001, 2.132350e-001, 2.169167e-001, 2.205953e-001, 2.242708e-001, 2.279430e-001, 2.316121e-001, 2.352778e-001, 2.389402e-001, 2.425992e-001, 2.462548e-001, 2.499068e-001, 2.535554e-001, 2.572003e-001, 2.608415e-001, 2.644790e-001, 2.681128e-001, 2.717428e-001, 2.753689e-001, 2.789911e-001, 2.826093e-001, 2.862236e-001, 2.898337e-001, 2.934397e-001, 2.970416e-001, 3.006392e-001, 3.042326e-001, 3.078216e-001, 3.114063e-001, 3.149865e-001, 3.185623e-001, 3.221335e-001, 3.257002e-001, 3.292622e-001, 3.328195e-001, 3.363722e-001, 3.399200e-001, 3.434630e-001, 3.470011e-001, 3.505343e-001, 3.540625e-001, 3.575857e-001, 3.611038e-001, 3.646168e-001, 3.681246e-001, 3.716271e-001, 3.751244e-001, 3.786163e-001, 3.821029e-001, 3.855840e-001, 3.890596e-001, 3.925298e-001, 3.959943e-001, 3.994532e-001, 4.029064e-001, 4.063539e-001, 4.097957e-001, 4.132316e-001, 4.166616e-001, 4.200857e-001, 4.235039e-001, 4.269160e-001, 4.303221e-001, 4.337220e-001, 4.371158e-001, 4.405033e-001, 4.438846e-001, 4.472596e-001, 4.506283e-001, 4.539905e-001, 4.573463e-001, 4.606956e-001, 4.640383e-001, 4.673744e-001, 4.707039e-001, 4.740267e-001, 4.773428e-001, 4.806521e-001, 4.839545e-001, 4.872501e-001, 4.905388e-001, 4.938205e-001, 4.970951e-001, 5.003627e-001, 5.036232e-001, 5.068765e-001, 5.101227e-001, 5.133615e-001, 5.165931e-001, 5.198173e-001, 5.230342e-001, 5.262436e-001, 5.294455e-001, 5.326400e-001, 5.358268e-001, 5.390060e-001, 5.421776e-001, 5.453414e-001, 5.484976e-001, 5.516459e-001, 5.547863e-001, 5.579189e-001, 5.610436e-001, 5.641603e-001, 5.672689e-001, 5.703696e-001, 5.734621e-001, 5.765464e-001, 5.796226e-001, 5.826905e-001, 5.857501e-001, 5.888014e-001, 5.918444e-001, 5.948789e-001, 5.979050e-001, 6.009226e-001, 6.039316e-001, 6.069321e-001, 6.099239e-001, 6.129071e-001, 6.158815e-001, 6.188472e-001, 6.218041e-001, 6.247522e-001, 6.276914e-001, 6.306216e-001, 6.335429e-001, 6.364552e-001, 6.393585e-001, 6.422527e-001, 6.451377e-001, 6.480136e-001, 6.508802e-001, 6.537376e-001, 6.565858e-001, 6.594245e-001, 6.622540e-001, 6.650740e-001, 6.678845e-001, 6.706856e-001, 6.734771e-001, 6.762591e-001, 6.790314e-001, 6.817941e-001, 6.845471e-001, 6.872904e-001, 6.900239e-001, 6.927476e-001, 6.954614e-001, 6.981654e-001, 7.008595e-001, 7.035436e-001, 7.062176e-001, 7.088817e-001, 7.115357e-001, 7.141795e-001, 7.168133e-001, 7.194368e-001, 7.220501e-001, 7.246531e-001, 7.272459e-001, 7.298283e-001, 7.324003e-001, 7.349619e-001, 7.375131e-001, 7.400538e-001, 7.425840e-001, 7.451036e-001, 7.476127e-001, 7.501111e-001, 7.525988e-001, 7.550759e-001, 7.575422e-001, 7.599978e-001, 7.624425e-001, 7.648764e-001, 7.672995e-001, 7.697116e-001, 7.721128e-001, 7.745031e-001, 7.768823e-001, 7.792505e-001, 7.816076e-001, 7.839536e-001, 7.862884e-001, 7.886121e-001, 7.909246e-001, 7.932258e-001, 7.955158e-001, 7.977944e-001, 8.000618e-001, 8.023177e-001, 8.045622e-001, 8.067954e-001, 8.090170e-001, 8.112271e-001, 8.134258e-001, 8.156128e-001, 8.177883e-001, 8.199521e-001, 8.221043e-001, 8.242448e-001, 8.263736e-001, 8.284906e-001, 8.305959e-001, 8.326894e-001, 8.347710e-001, 8.368408e-001, 8.388986e-001, 8.409446e-001, 8.429786e-001, 8.450006e-001, 8.470106e-001, 8.490086e-001, 8.509945e-001, 8.529683e-001, 8.549300e-001, 8.568795e-001, 8.588169e-001, 8.607420e-001, 8.626549e-001, 8.645556e-001, 8.664440e-001, 8.683200e-001, 8.701838e-001, 8.720351e-001, 8.738741e-001, 8.757006e-001, 8.775147e-001, 8.793163e-001, 8.811054e-001, 8.828820e-001, 8.846461e-001, 8.863976e-001, 8.881364e-001, 8.898627e-001, 8.915763e-001, 8.932773e-001, 8.949655e-001, 8.966410e-001, 8.983038e-001, 8.999538e-001, 9.015911e-001, 9.032155e-001, 9.048271e-001, 9.064258e-001, 9.080116e-001, 9.095845e-001, 9.111445e-001, 9.126916e-001, 9.142257e-001, 9.157468e-001, 9.172548e-001, 9.187499e-001, 9.202318e-001, 9.217007e-001, 9.231566e-001, 9.245992e-001, 9.260288e-001, 9.274452e-001, 9.288484e-001, 9.302384e-001, 9.316151e-001, 9.329787e-001, 9.343289e-001, 9.356659e-001, 9.369896e-001, 9.383000e-001, 9.395971e-001, 9.408808e-001, 9.421511e-001, 9.434080e-001, 9.446515e-001, 9.458816e-001, 9.470983e-001, 9.483015e-001, 9.494912e-001, 9.506674e-001, 9.518302e-001, 9.529793e-001, 9.541150e-001, 9.552371e-001, 9.563456e-001, 9.574405e-001, 9.585218e-001, 9.595895e-001, 9.606435e-001,
```

9.616839e-001, 9.627106e-001, 9.637237e-001, 9.647230e-001, 9.657087e-001, 9.666806e-001, 9.676387e-001, 9.685832e-001, 9.695138e-001, 9.704307e-001, 9.713338e-001, 9.722231e-001, 9.730985e-001, 9.739601e-001, 9.748079e-001, 9.756419e-001, 9.764619e-001, 9.772681e-001, 9.780604e-001, 9.788388e-001, 9.796033e-001, 9.803539e-001, 9.810905e-001, 9.818132e-001, 9.825219e-001, 9.832167e-001, 9.838975e-001, 9.845643e-001, 9.852172e-001, 9.858560e-001, 9.864808e-001, 9.870916e-001, 9.876883e-001, 9.882711e-001, 9.888397e-001, 9.893944e-001, 9.899349e-001, 9.904614e-001, 9.909738e-001, 9.914722e-001, 9.919564e-001, 9.924266e-001, 9.928826e-001, 9.933245e-001, 9.937523e-001, 9.941660e-001, 9.945656e-001, 9.949510e-001, 9.953223e-001, 9.956794e-001, 9.960224e-001, 9.963513e-001, 9.966659e-001, 9.969664e-001, 9.972528e-001, 9.975249e-001, 9.977829e-001, 9.980267e-001, 9.982564e-001, 9.984718e-001, 9.986730e-001, 9.988601e-001, 9.990329e-001, 9.991916e-001, 9.993360e-001, 9.994663e-001, 9.995823e-001, 9.996842e-001, 9.997718e-001, 9.998452e-001, 9.999045e-001, 9.999495e-001, 9.999803e-001, 9.999968e-001, 9.999992e-001, 9.999874e-001, 9.999613e-001, 9.999210e-001, 9.998666e-001, 9.997979e-001, 9.997150e-001, 9.996179e-001, 9.995066e-001, 9.993810e-001, 9.992413e-001, 9.990874e-001, 9.989193e-001, 9.987370e-001, 9.985404e-001, 9.983297e-001, 9.981048e-001, 9.978658e-001, 9.976125e-001, 9.973451e-001, 9.970635e-001, 9.967677e-001, 9.964577e-001, 9.961336e-001, 9.957953e-001, 9.954429e-001, 9.950763e-001, 9.946956e-001, 9.943008e-001, 9.938918e-001, 9.934687e-001, 9.930315e-001, 9.925801e-001, 9.921147e-001, 9.916352e-001, 9.911415e-001, 9.906338e-001, 9.901120e-001, 9.895761e-001, 9.890262e-001, 9.884622e-001, 9.878841e-001, 9.872921e-001, 9.866859e-001, 9.860658e-001, 9.854317e-001, 9.847835e-001, 9.841213e-001, 9.834452e-001, 9.827551e-001, 9.820510e-001, 9.813330e-001, 9.806010e-001, 9.798551e-001, 9.790952e-001, 9.783214e-001, 9.775338e-001, 9.767322e-001, 9.759168e-001, 9.750874e-001, 9.742443e-001, 9.733873e-001, 9.725164e-001, 9.716317e-001, 9.707333e-001, 9.698210e-001, 9.688949e-001, 9.679551e-001, 9.670015e-001, 9.660342e-001, 9.650531e-001, 9.640583e-001, 9.630498e-001, 9.620277e-001, 9.609918e-001, 9.599423e-001, 9.588792e-001, 9.578024e-001, 9.567121e-001, 9.556081e-001, 9.544905e-001, 9.533594e-001, 9.522147e-001, 9.510565e-001, 9.498848e-001, 9.486996e-001, 9.475009e-001, 9.462887e-001, 9.450631e-001, 9.438240e-001, 9.425716e-001, 9.413057e-001, 9.400265e-001, 9.387339e-001, 9.374279e-001, 9.361087e-001, 9.347761e-001, 9.334302e-001, 9.320711e-001, 9.306987e-001, 9.293132e-001, 9.279144e-001, 9.265024e-001, 9.250772e-001, 9.236389e-001, 9.221875e-001, 9.207229e-001, 9.192453e-001, 9.177546e-001, 9.162509e-001, 9.147341e-001, 9.132044e-001, 9.116617e-001, 9.101060e-001, 9.085373e-001, 9.069558e-001, 9.053614e-001, 9.037541e-001, 9.021340e-001, 9.005010e-001, 8.988552e-001, 8.971967e-001, 8.955254e-001, 8.938414e-001, 8.921447e-001, 8.904353e-001, 8.887133e-001, 8.869786e-001, 8.852313e-001, 8.834714e-001, 8.816990e-001, 8.799141e-001, 8.781166e-001, 8.763067e-001, 8.744843e-001, 8.726595e-001, 8.708022e-001, 8.689426e-001, 8.670707e-001, 8.651864e-001, 8.632899e-001, 8.613810e-001, 8.594599e-001, 8.575267e-001, 8.555812e-001, 8.536235e-001, 8.516538e-001, 8.496719e-001, 8.476779e-001, 8.456719e-001, 8.436539e-001, 8.416239e-001, 8.395819e-001, 8.375280e-001, 8.354622e-001, 8.333846e-001, 8.312950e-001, 8.291937e-001, 8.270806e-001, 8.249557e-001, 8.228191e-001, 8.206708e-001, 8.185108e-001, 8.163393e-001, 8.141561e-001, 8.119613e-001, 8.097550e-001, 8.075372e-001, 8.053079e-001, 8.030672e-001, 8.008150e-001, 7.985151e-001, 7.962766e-001, 7.939904e-001, 7.916929e-001, 7.893842e-001, 7.870642e-001, 7.847331e-001, 7.823908e-001, 7.800374e-001, 7.776729e-001, 7.752974e-001, 7.729108e-001, 7.705132e-001, 7.681047e-001, 7.656853e-001, 7.632550e-001, 7.608139e-001, 7.583619e-001, 7.558992e-001, 7.534257e-001, 7.509415e-001, 7.484466e-001, 7.459411e-001, 7.434250e-001, 7.408984e-001, 7.383612e-001, 7.358135e-001, 7.332553e-001, 7.306868e-001, 7.281078e-001, 7.255185e-001, 7.229189e-001, 7.203090e-001, 7.176889e-001, 7.150586e-001, 7.124181e-001, 7.097675e-001, 7.071068e-001, 7.044360e-001, 7.017553e-001, 6.990645e-001, 6.963639e-001, 6.936533e-001, 6.909329e-001, 6.882026e-001, 6.854626e-001, 6.827128e-001, 6.799534e-001, 6.771842e-001, 6.744055e-001, 6.716171e-001, 6.688193e-001, 6.660119e-001, 6.631950e-001, 6.603687e-001, 6.575331e-001, 6.546880e-001, 6.518337e-001, 6.489701e-001, 6.460973e-001, 6.432153e-001, 6.403242e-001, 6.374240e-001, 6.345147e-001, 6.315964e-001, 6.286691e-001, 6.257329e-001, 6.227878e-001, 6.198338e-001, 6.168710e-001, 6.138995e-001, 6.109192e-001, 6.079303e-001, 6.049327e-001, 6.019265e-001, 5.989118e-001, 5.958885e-001, 5.928568e-001, 5.898167e-001, 5.867681e-001, 5.837113e-001, 5.806461e-001, 5.775727e-001, 5.744911e-001, 5.714013e-001, 5.683034e-001, 5.651974e-001, 5.620834e-001, 5.589614e-001, 5.558314e-001, 5.526936e-001, 5.495479e-001, 5.463943e-001, 5.432331e-001, 5.400641e-001, 5.368874e-001, 5.337031e-001, 5.305112e-001, 5.273118e-001, 5.241048e-001, 5.208904e-001, 5.176687e-001, 5.144395e-001, 5.112031e-001, 5.079594e-001, 5.047084e-001, 5.014503e-001, 4.981851e-001, 4.949128e-001, 4.916334e-001, 4.883471e-001, 4.850538e-001, 4.817537e-001, 4.784467e-001, 4.751328e-001, 4.718123e-001, 4.684850e-001, 4.651511e-001, 4.618105e-001, 4.584634e-001, 4.551098e-001, 4.517497e-001, 4.483832e-001, 4.450103e-001, 4.416311e-001, 4.382456e-001, 4.348539e-001, 4.314560e-001, 4.280520e-001, 4.246419e-001, 4.212258e-001, 4.178036e-001, 4.143756e-001, 4.109416e-001, 4.075018e-001, 4.040562e-001, 4.006049e-001, 3.971479e-001, 3.936852e-001, 3.902170e-001, 3.867432e-001, 3.832638e-001, 3.797791e-001, 3.762889e-001, 3.727934e-001, 3.692927e-001, 3.657866e-001, 3.622754e-001, 3.587590e-001, 3.552375e-001, 3.517109e-001, 3.481794e-001, 3.446429e-001, 3.411015e-001, 3.375553e-001, 3.340043e-001, 3.304485e-001, 3.268880e-001, 3.233229e-001, 3.197532e-001, 3.161789e-001, 3.126002e-001, 3.090170e-001, 3.054294e-001, 3.018375e-001, 2.982413e-001, 2.946408e-001, 2.910362e-001, 2.874274e-001, 2.838145e-001, 2.801976e-001, 2.765767e-001, 2.729519e-001, 2.693232e-001, 2.656907e-001, 2.620544e-001, 2.584144e-001, 2.547707e-001, 2.511234e-001, 2.474725e-001, 2.438181e-001, 2.401603e-001, 2.364990e-001, 2.328344e-001, 2.291664e-001, 2.254952e-001, 2.218208e-001, 2.181432e-001, 2.144626e-001, 2.107786e-001, 2.070922e-001, 2.034025e-001, 1.997100e-001, 1.960146e-001, 1.923164e-001, 1.886155e-001, 1.849120e-001, 1.812058e-001, 1.774970e-001, 1.737857e-001, 1.700719e-001, 1.663557e-001, 1.626372e-001, 1.589163e-001, 1.551932e-001, 1.514678e-001, 1.477404e-001, 1.440108e-001, 1.402792e-001, 1.365455e-001, 1.328100e-001, 1.290725e-001, 1.253332e-001, 1.215922e-001, 1.178494e-001, 1.141049e-001, 1.103588e-001, 1.066112e-001, 1.028620e-001, 9.911134e-002, 9.535930e-002, 9.160590e-002, 8.785120e-002, 8.409525e-002, 8.033810e-002, 7.657981e-002, 7.282044e-002, 6.906003e-002, 6.529863e-002, 6.153631e-002, 5.777312e-002, 5.400910e-002, 5.024432e-002, 4.647882e-002, 4.271266e-002, 3.894590e-002, 3.517858e-002, 3.141076e-002, 2.764249e-002, 2.387384e-002, 2.010484e-002, 1.633556e-002, 1.256604e-002, 8.796346e-003, 5.026527e-003, 1.256637e-003, 2.513271e-003, 6.283144e-003, 1.005293e-002, 1.382257e-002, 1.759201e-002, 2.136121e-002, 2.513010e-002, 2.889863e-002, 3.266675e-002, 3.643441e-002, 4.020155e-002, 4.396812e-002, 4.773406e-002, 5.149933e-002, 5.526386e-002, 5.902761e-002, 6.279052e-002, 6.655254e-002, 7.031361e-002, 7.407368e-002, 7.783270e-002, 8.159061e-002, 8.534736e-002, 8.910291e-002, 9.285718e-002, 9.661013e-002, 1.003617e-001, 1.041119e-001, 1.078605e-001, 1.116077e-001, 1.153532e-001, 1.190972e-001, 1.228394e-001, 1.265799e-001, 1.303185e-001, 1.340554e-001, 1.377903e-001, 1.415233e-001, 1.452542e-001, 1.489831e-001, 1.527099e-001, 1.564345e-001, 1.601568e-001, 1.638769e-001, 1.675947e-001, 1.713101e-001, 1.750231e-001, 1.787335e-001, 1.824415e-001, 1.861468e-001, 1.898495e-001, 1.935495e-001, 1.972467e-001, 2.009411e-001, 2.046327e-001, 2.083214e-001, 2.120071e-001, 2.156898e-001, 2.193694e-001, 2.230460e-001, 2.267193e-001, 2.303894e-001, 2.340563e-001, 2.377198e-001, 2.413799e-001, 2.450367e-001, 2.486899e-001, 2.523396e-001, 2.559857e-001, 2.596282e-001, 2.632669e-001, 2.669020e-001, 2.705332e-001, 2.741606e-001, 2.777841e-001, 2.814037e-001, 2.850193e-001, 2.886308e-001, 2.922382e-001, 2.958414e-001, 2.994405e-001, 3.030353e-001, 3.066258e-001, 3.102119e-001, 3.137936e-001, 3.173709e-001, 3.209436e-001, 3.245118e-001, 3.280754e-001, 3.316343e-001, 3.351885e-001, 3.387379e-001, 3.422825e-001, 3.458223e-001, 3.493571e-001, 3.528870e-001, 3.564119e-001, 3.599317e-001, 3.634464e-001, 3.669559e-001, 3.704602e-001, 3.739592e-001, 3.774529e-001, 3.809413e-001, 3.844242e-001, 3.879017e-001, 3.913737e-001, 3.948401e-001, 3.983009e-001, 4.017560e-001, 4.052054e-001,

0.086491e-001, 4.120869e-001, 4.155189e-001, 4.189450e-001, 4.223652e-001, 4.257793e-001, 4.291874e-001, 4.325894e-001, 4.359582e-001, 4.393748e-001, 4.427582e-001, 4.461353e-001, 4.495061e-001, 4.528705e-001, 4.562284e-001, 4.595799e-001, 4.629248e-001, 4.662631e-001, 4.695948e-001, 4.729199e-001, 4.762382e-001, 4.795498e-001, 4.828545e-001, 4.861524e-001, 4.894433e-001, 4.927273e-001, 4.960043e-001, 4.992743e-001, 5.025372e-001, 5.057929e-001, 5.090414e-001, 5.122827e-001, 5.155167e-001, 5.187434e-001, 5.219627e-001, 5.251746e-001, 5.283791e-001, 5.315760e-001, 5.347654e-001, 5.379471e-001, 5.411213e-001, 5.442877e-001, 5.474464e-001, 5.505973e-001, 5.537404e-001, 5.568756e-001, 5.600029e-001, 5.631223e-001, 5.662336e-001, 5.693369e-001, 5.724321e-001, 5.755192e-001, 5.785981e-001, 5.816688e-001, 5.847312e-001, 5.877853e-001, 5.908310e-001, 5.938683e-001, 5.968972e-001, 5.999177e-001, 6.029295e-001, 6.059329e-001, 6.089276e-001, 6.119136e-001, 6.148910e-001, 6.178596e-001, 6.208195e-001, 6.237705e-001, 6.267126e-001, 6.296459e-001, 6.325702e-001, 6.354855e-001, 6.383917e-001, 6.412899e-001, 6.441770e-001, 6.470560e-001, 6.499257e-001, 6.527862e-001, 6.556374e-001, 6.584793e-001, 6.613119e-001, 6.641350e-001, 6.669487e-001, 6.697529e-001, 6.725477e-001, 6.753328e-001, 6.781084e-001, 6.808743e-001, 6.836305e-001, 6.863770e-001, 6.891138e-001, 6.918408e-001, 6.945579e-001, 6.972652e-001, 6.999626e-001, 7.026500e-001, 7.053274e-001, 7.079948e-001, 7.106521e-001, 7.132994e-001, 7.159365e-001, 7.185634e-001, 7.211801e-001, 7.237866e-001, 7.263828e-001, 7.289686e-001, 7.315441e-001, 7.341092e-001, 7.366639e-001, 7.392081e-001, 7.417418e-001, 7.442649e-001, 7.467775e-001, 7.492794e-001, 7.517708e-001, 7.542514e-001, 7.567213e-001, 7.591804e-001, 7.616288e-001, 7.640663e-001, 7.664930e-001, 7.689088e-001, 7.713136e-001, 7.737075e-001, 7.760904e-001, 7.784623e-001, 7.808231e-001, 7.831728e-001, 7.855114e-001, 7.878388e-001, 7.901550e-001, 7.924600e-001, 7.947537e-001, 7.970361e-001, 7.993072e-001, 8.015670e-001, 8.038153e-001, 8.060523e-001, 8.082777e-001, 8.104917e-001, 8.126942e-001, 8.148851e-001, 8.170644e-001, 8.192321e-001, 8.213882e-001, 8.235326e-001, 8.256653e-001, 8.277863e-001, 8.298955e-001, 8.319929e-001, 8.340784e-001, 8.361522e-001, 8.382140e-001, 8.402639e-001, 8.423019e-001, 8.443279e-001, 8.463419e-001, 8.483439e-001, 8.503339e-001, 8.523117e-001, 8.542774e-001, 8.562310e-001, 8.581724e-001, 8.601017e-001, 8.620187e-001, 8.639234e-001, 8.658159e-001, 8.676961e-001, 8.695639e-001, 8.714194e-001, 8.732625e-001, 8.750931e-001, 8.769114e-001, 8.787172e-001, 8.805104e-001, 8.822912e-001, 8.840595e-001, 8.858151e-001, 8.875582e-001, 8.892887e-001, 8.910065e-001, 8.927117e-001, 8.944042e-001, 8.960839e-001, 8.977510e-001, 8.994053e-001, 9.010467e-001, 9.026754e-001, 9.042913e-001, 9.058943e-001, 9.074844e-001, 9.090617e-001, 9.106260e-001, 9.121773e-001, 9.137158e-001, 9.152412e-001, 9.167536e-001, 9.182530e-001, 9.197393e-001, 9.212126e-001, 9.226727e-001, 9.241198e-001, 9.255537e-001, 9.269745e-001, 9.283821e-001, 9.297765e-001, 9.311577e-001, 9.325256e-001, 9.338803e-001, 9.352218e-001, 9.365499e-001, 9.378647e-001, 9.391662e-001, 9.404544e-001, 9.417291e-001, 9.429905e-001, 9.442385e-001, 9.454731e-001, 9.466942e-001, 9.479019e-001, 9.490961e-001, 9.502769e-001, 9.514441e-001, 9.525978e-001, 9.537379e-001, 9.548645e-001, 9.559776e-001, 9.570770e-001, 9.581629e-001, 9.592351e-001, 9.602937e-001, 9.613386e-001, 9.623699e-001, 9.633875e-001, 9.643914e-001, 9.653816e-001, 9.663581e-001, 9.673209e-001, 9.682699e-001, 9.692051e-001, 9.701266e-001, 9.710343e-001, 9.719282e-001, 9.728082e-001, 9.736745e-001, 9.745269e-001, 9.753654e-001, 9.761901e-001, 9.770009e-001, 9.777979e-001, 9.785809e-001, 9.793500e-001, 9.801052e-001, 9.808465e-001, 9.815739e-001, 9.822873e-001, 9.829867e-001, 9.836721e-001, 9.843436e-001, 9.850011e-001, 9.856446e-001, 9.862741e-001, 9.868895e-001, 9.874910e-001, 9.880784e-001, 9.886517e-001, 9.892111e-001, 9.897563e-001, 9.902875e-001, 9.907322e-001, 9.912788e-001, 9.918113e-001, 9.923227e-001,

```

4.325894e-001, 4.291874e-001, 4.257793e-001, 4.223652e-001, 4.189450e-001, 4.155189e-001, 4.120869e-001,
4.086491e-001, 4.052054e-001, 4.017560e-001, 3.983009e-001, 3.948401e-001, 3.913737e-001, 3.879017e-001,
3.844242e-001, 3.809413e-001, 3.774529e-001, 3.739592e-001, 3.704602e-001, 3.669559e-001, 3.634464e-001,
3.599317e-001, 3.564119e-001, 3.528870e-001, 3.493571e-001, 3.458223e-001, 3.422825e-001, 3.387379e-001,
3.351885e-001, 3.316343e-001, 3.280754e-001, 3.245118e-001, 3.209436e-001, 3.173709e-001, 3.137936e-001,
3.102119e-001, 3.066258e-001, 3.030353e-001, 2.994405e-001, 2.958414e-001, 2.922382e-001, 2.886308e-001,
2.850193e-001, 2.814037e-001, 2.777841e-001, 2.741606e-001, 2.705332e-001, 2.669020e-001, 2.632669e-001,
2.596282e-001, 2.559857e-001, 2.523396e-001, 2.486899e-001, 2.450367e-001, 2.413799e-001, 2.377198e-001,
2.340563e-001, 2.303894e-001, 2.267193e-001, 2.230460e-001, 2.193694e-001, 2.156898e-001, 2.120071e-001,
2.083214e-001, 2.046327e-001, 2.009411e-001, 1.972467e-001, 1.935495e-001, 1.898495e-001, 1.861468e-001,
1.824415e-001, 1.787335e-001, 1.750231e-001, 1.713101e-001, 1.675947e-001, 1.638769e-001, 1.601568e-001,
1.564345e-001, 1.527099e-001, 1.489831e-001, 1.452542e-001, 1.415233e-001, 1.377903e-001, 1.340554e-001,
1.303185e-001, 1.265799e-001, 1.228394e-001, 1.190972e-001, 1.153532e-001, 1.116077e-001, 1.078605e-001,
1.041119e-001, 1.003617e-001, 9.661013e-002, 9.285718e-002, 8.910291e-002, 8.534736e-002, 8.159061e-002,
7.783270e-002, 7.407368e-002, 7.031361e-002, 6.655254e-002, 6.279052e-002, 5.902761e-002, 5.526386e-002,
5.149933e-002, 4.773406e-002, 4.396812e-002, 4.020155e-002, 3.643441e-002, 3.266675e-002, 2.889863e-002,
2.513010e-002, 2.136121e-002, 1.759201e-002, 1.382257e-002, 1.005293e-002, 6.283144e-003, 2.513271e-003};
21. main()
22. {
    // Step 1. Initialize System Control:
    // PLL, WatchDog, enable Peripheral Clocks.
    // This example function is found in the DSP2833x_SysCtrl.c file.
23.     InitSysCtrl();
24.     EALLOW;
25.     #if (CPU_FRQ_150MHZ) // Default - 150 MHz SYSCLKOUT
26.     #define ADC_MODCLK 0x1 // HSPCLK = SYSCLKOUT/2*ADC_MODCLK2 = 150/(2*3) = 25.0 MHz.
27.     #endif
28.     #if (CPU_FRQ_100MHZ)
29.     #define ADC_MODCLK 0x2 // HSPCLK = SYSCLKOUT/2*ADC_MODCLK2 = 100/(2*2) = 50.0 MHz.
30.     #endif
31.     EDIS;
32.     EALLOW;
33.     SysCtrlRegs.HISPCP.all = 0x3; // HSPCLK = SYSCLKOUT/(2*3).
34.     EDIS;

    // Step 2. Initialize GPIO:
    // This example function is found in the DSP2833x_Gpio.c file and
    // illustrates how to set the GPIO to it's default state.
    // InitGpio(); // Skipped for this example
35.     EALLOW;
    //Configuração da Saída dos Pulsos Chave S1.
36.     GpioCtrlRegs.GPAMUX1.bit.GPIO12 = 0x0; //Configura GPIO12 como pino de I/O.
37.     GpioCtrlRegs.GPADIR.bit.GPIO12 = 0x1; //Configura GPIO12 como OUTPUT.
38.     GpioDataRegs.GPACLEAR.bit.GPIO12 = 0x1; //Configura GPIO12 como LOW.

    //Configuração da Saída dos Pulsos Chave S2.
39.     GpioCtrlRegs.GPAMUX1.bit.GPIO13 = 0x0; //Configura GPIO13 como pino de I/O.
40.     GpioCtrlRegs.GPADIR.bit.GPIO13 = 0x1; //Configura GPIO13 como OUTPUT.
41.     GpioDataRegs.GPACLEAR.bit.GPIO13 = 0x1; //Configura GPIO13 como LOW.

    //Configuração da Saída dos Pulsos Chave S3.
42.     GpioCtrlRegs.GPBMUX1.bit.GPIO33 = 0x0; //Configura GPIO33 como pino de I/O.
43.     GpioCtrlRegs.GPBDIR.bit.GPIO33 = 0x1; //Configura GPIO33 como OUTPUT.
44.     GpioDataRegs.GPBCLEAR.bit.GPIO33 = 0x1; //Configura GPIO33 como LOW.
45.     EDIS;

    // Step 3. Clear all interrupts and initialize PIE vector table:
    // Disable CPU interrupts.
46.     DINT;

    // Initialize the PIE control registers to their default state.
    // The default state is all PIE interrupts disabled and flags
    // are cleared.
    // This function is found in the DSP2833x_PieCtrl.c file.
47.     InitPieCtrl();

    // Disable CPU interrupts and clear all CPU interrupt flags:
48.     IER = 0x0000;
49.     IFR = 0x0000;

    // Initialize the PIE vector table with pointers to the shell Interrupt
    // Service Routines (ISR).
    // This will populate the entire table, even if the interrupt
    // is not used in this example. This is useful for debug purposes.
    // The shell ISR routines are found in DSP2833x_DefaultIsr.c.
    // This function is found in DSP2833x_PieVect.c.
50.     InitPieVectTable();

    // Interrupts that are used in this example are re-mapped to
    // ISR functions found within this file.
51.     EALLOW; // This is needed to write to EALLOW protected register
52.     PieVectTable.ADCINT = &adc_isr;
53.     EDIS; // This is needed to disable write to EALLOW protected registers

    // Step 4. Initialize all the Device Peripherals:
    // This function is found in DSP2833x_InitPeripherals.c
54.     InitAdc(); // Inicia o ADC

    // Step 5. User specific code, enable interrupts:
    // Enable ADCINT in PIE
55.     PieCtrlRegs.PIEIER1.bit.INTx6 = 1;

```

```

56.     IER |= M_INT1; // Enable CPU Interrupt 1
57.     EINT; // Enable Global interrupt INTM
58.     ERTM; // Enable Global realtime interrupt DBGM
    // Configuração do ADC
59.     AdcRegs.ADCMAXCONV.all = 0x0003; // Setup 4conv's on SEQ1 and 1conv on SEQ2.
60.     AdcRegs.ADCCHSELSEQ1.bit.CONV00 = 0x0; // Setup ADCINA0 as 1st SEQ1 conv.
61.     AdcRegs.ADCCHSELSEQ1.bit.CONV01 = 0x1; // Setup ADCINA3 as 2nd SEQ1 conv.
62.     AdcRegs.ADCCHSELSEQ1.bit.CONV02 = 0x2; // Setup ADCINA4 as 3rd SEQ1 conv.
63.     AdcRegs.ADCCHSELSEQ1.bit.CONV03 = 0x3; // Setup ADCINA5 as 4nd SEQ1 conv.
64.     AdcRegs.ADCCHSELSEQ2.bit.CONV04 = 0x4; // Setup ADCINA6 as 5st SEQ2 conv.
65.     AdcRegs.ADCCTRL2.bit.EPWM_SOC_SEQ1 = 1; // Enable SOCA from ePWM to start SEQ1.
66.     AdcRegs.ADCCTRL2.bit.INT_ENA_SEQ1 = 1; // Enable SEQ1 interrupt (every EOS).
67.     AdcRegs.ADCCTRL3.bit.ADCCLKPS = ADC_CKPS;
    // Assumes ePWM1 clock is already enabled in InitSysCtrl();
68.     EPwm1Regs.ETSEL.bit.SOCASEN = 1; // Enable SOC on A group.
69.     EPwm1Regs.ETSEL.bit.SOCASEL = 2; // Select SOC from period TBCNT = CMPA.
70.     EPwm1Regs.ETPS.bit.SOCAPRD = 1; // Generate pulse on 1st event.
71.     EPwm1Regs.TBCTL.bit.CTRMODE = 0; // count up and start.
72.     EPwm1Regs.TBPRD = 750; // Set period for ePWM1 750.

/*
Ex: F_onda = 60 Hz
F_amostragem = 100 kHz
BUFFERSIZE = F_amostragem/F_onda = 100k/120 = 1667
TBPRD = HSPCLK / F_amostragem = 75MHz / 100kHz
TBPRD = 750
*/
// Espera pela interrupção do ADC
73.     while(1)
74.     {
75.     }

interrupt void adc_isr(void)
{
76.     if(cont <= BUFSIZE)
77.     {
78.         //----- Aquisições -----
80.         linA = (AdcRegs.ADCRESULT0 >>4) - 2011; //Armazena o valor de ADCRESULT0 na variável linA.
81.         VinA_pass = VinA;
82.         VinA = (AdcRegs.ADCRESULT1 >>4) - 2030;
83.         Produto = (float)(VinA) * (float)(VinA_ant); //Multiplicação da tensão da amostra pela amostra
84.         linB = (AdcRegs.ADCRESULT2 >>4) - 2022; //Armazena o valor de ADCRESULT0 na variável linB.
85.         linC = (AdcRegs.ADCRESULT3 >>4) - 2022; //Armazena o valor de ADCRESULT0 na variável linC.
86.         lo = (AdcRegs.ADCRESULT4 >>4) - 2024; //Armazena o valor de ADCRESULT0 na variável linC.

        //Retifica as ondas de corrente.
87.         if (linA<0)
88.         {linA = -linA;}
89.         if (linB<0)
90.         {linB = -linB;}
91.         if (linC<0)
92.         {linC = -linC;}

        //----- Tratamento de Sinal e Detecção de Zero-----
        //Detecção de zero de tensão.
93.         if ((Produto<= 0) &&(cond==1))
94.         {
95.             cont=0;
96.             cond=0;
97.             lo_area=0;
98.         }
99.         if (cont==800)
100.        {cond=1;}

        //-----Malha Fechada-----
101.        IRefA = (Isen[i] * k * lo_med); //Fase A Realimentada.
102.        IRefB = (Isen[i] + 278] * k * lo_med); //Fase B Realimentado.
103.        IRefC = (Isen[i] + 555] * k * lo_med); //Fase C Realimentado.

        //-----Pulsos de S1-----
104.        if(IRefA > ((float)linA))
105.        {GpioDataRegs.GPASET.bit.GPIO12 = 0x1;} //Configura GPIO12 como HIGH .
106.        else
107.        {GpioDataRegs.GPACLEAR.bit.GPIO12 = 0x1;} //Configura GPIO12 como LOW.

        //-----Pulsos de S2-----
108.        if(IRefB > ((float)linB))
109.        {GpioDataRegs.GPASET.bit.GPIO13 = 0x1;} //Configura GPIO13 como HIGH.
110.        else
111.        {GpioDataRegs.GPACLEAR.bit.GPIO13 = 0x1;} //Configura GPIO13 como LOW.

        //-----Pulsos de S3-----
112.        if(IRefC > ((float)linC))
113.        {GpioDataRegs.GPBSET.bit.GPIO33 = 0x1;} //Configura GPIO33 como HIGH.
114.        else
115.        {GpioDataRegs.GPBCLEAR.bit.GPIO33 = 0x1;} //Configura GPIO33 como LOW.

        //Incrementa contador geral.

```



```
116.             cont++;

                //Calcula valor médio da corrente de carga.
117.             lo_area = lo_area + (float)lo;
118.             lo_med = lo_area / (float)cont;
119.
120.         }
121.     else
122.     {
123.         cont = 0;
124.         GpioDataRegs.GPACLEAR.bit.GPIO12 = 0x1; //Configura GPIO12 como LOW.
125.         GpioDataRegs.GPACLEAR.bit.GPIO13 = 0x1; //Configura GPIO13 como LOW.
126.         GpioDataRegs.GPBCLEAR.bit.GPIO33 = 0x1; //Configura GPIO33 como LOW.
127.     }
    // Reinitialize for next ADC sequence.
128.    AdcRegs.ADCTRL2.bit.RST_SEQ1 = 1; // Reset SEQ1.
129.    AdcRegs.ADCST.bit.INT_SEQ1_CLR = 1; // Clear INT SEQ1 bit.
130.    PieCtrlRegs.PIEACK.all = PIEACK_GROUP1; // Acknowledge interrupt to PIE.
131.    return;
132. }
```