



## Exercício 03

### Programação Aplicada a Sistemas de Energia

#### Questão 1: p11.m

Em sistemas de geração fotovoltaicos a energia absorvida pelos arranjos de painéis é convertida em energia elétrica em corrente contínua. Como na maioria dos sistemas não há nenhuma forma de armazenamento, e como energia se conserva, toda energia absorvida pelos módulos deve ser prontamente injetada na rede. Para escoar a produção os inversores de frequência devem não apenas executar a conversão CC-CA, mas fazê-la de forma a controlar a injeção de potência na rede, garantindo o equilíbrio energético e consequentemente a estabilidade do barramento CC.

Para tal, a tensão imposta pelo inversor deve estar em sincronismo com a da rede. De fato, o controle da potência ativa do inversor depende da fase relativa entre a tensão da rede e a tensão imposta pelo inversor. Ainda, uma vez que a rede opera com frequência variável em torno dos 60 Hz, o inversor precisa a todo instante estimar a fase da tensão da rede para executar os ajustes necessários a fim de manter o sincronismo e o ângulo de carga desejado. Essa estimação não é trivial já que, além das variações da tensão e frequência em regime permanente, a rede ainda apresenta efeitos transitórios impulsivos e oscilatórios, distorções harmônicas, flutuações, ruídos, afundamentos, etc. Isso é ainda mais desafiador em sistemas monofásicos, onde amplitude, frequência e fase devem ser obtidas a partir de medições de um único sinal sujeito aos fenômenos já citados.

Na prática existem algumas estratégias para se obter uma estimativa precisa e instantânea da fase da tensão da rede, sendo que a maioria delas são baseadas na estrutura SOGI-PLL, cujo diagrama de blocos básico é apresentado na Figura 1.

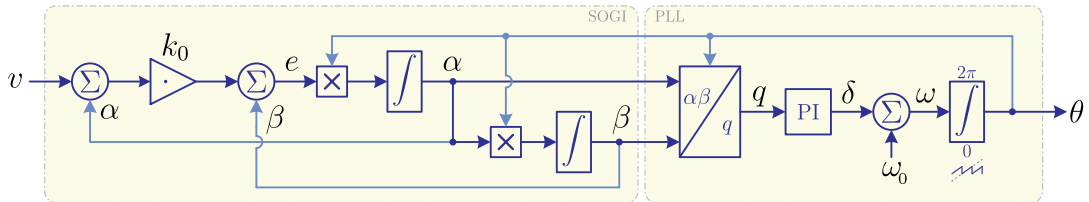


Figura 1: SOGI-PLL.

O estimador da Figura 1 pode ser discretizado e reescrito na forma de equações a diferenças, apresentadas nas equações (1–8), permitindo sua implementação em controladores digitais.

O algoritmo discretizado recebe um sinal amostrado da tensão de entrada (geralmente em algumas poucas dezenas de kHz, dependendo da frequência de comutação empregado no inversor). A partir dessas amostras, calcula a fase quase instantânea (com atraso de uma amostra). Esse sinal é utilizado nas malhas de controle das correntes e da potência injetadas na rede (não apresentados).

$$e_n = k_0 (v_n - \alpha_{n-1}) - \beta_{n-1} \quad (1)$$

$$a_n = w_{n-1} k_1 (e_n + e_{n-1}) + \alpha_{n-1} \quad (2)$$

$$b_n = w_{n-1} k_1 (a_n + a_{n-1}) + \beta_{n-1} \quad (3)$$

$$d_n = \alpha_n \sin(\theta_{n-1}) - \beta_n \cos(\theta_{n-1}) \quad (4)$$

$$q_n = \alpha_n \cos(\theta_{n-1}) + \beta_n \sin(\theta_{n-1}) \quad (5)$$

$$\delta_n = k_2 q_n + k_3 q_{n-1} + \delta_{n-1} \quad (6)$$

$$\omega_n = \omega_0 + \delta_n \quad (7)$$

$$\theta_n = k_1 (\omega_n + \omega_{n-1}) + \theta_{n-1} \pmod{2\pi} \quad (8)$$



Em [https://github.com/j-Lago/aulas\\_orcad](https://github.com/j-Lago/aulas_orcad) você encontrará um arquivo `rede.csv` com a tensão (coluna 2) amostrada de alguns ciclos da rede elétrica e um registro temporal (coluna 1) de cada amostra. Escreva um script que importe esse sinal para um vetor, e para cada amostra (cada elemento do vetor) execute os cálculos (1–8).

Guarde os resultados de cada amostra de  $v$  (tensão de entrada),  $\alpha$  (tensão filtrada),  $d$  (componente de eixo direto da tensão em base síncrona),  $q$  (componente em quadratura) e  $\theta$  (fase estimada pelo PLL). Gere e exporte uma figura apresentando a evolução temporal de cada uma desses sinais, bem como alguma indicação do instante que o PLL conseguiu se sincronizar (atracar) com a rede elétrica. De forma simplificada, assumamos que o sincronismo foi obtido no instante do primeiro cruzamento por zero do sinal  $\theta$  (saída do PLL) em que  $\arctan(q/d) < 0.035$  rad ( $\approx 2^\circ$ ) e  $d > 250$  V. A Figura 2 mostra um exemplo de como os resultados do script podem ser apresentados.

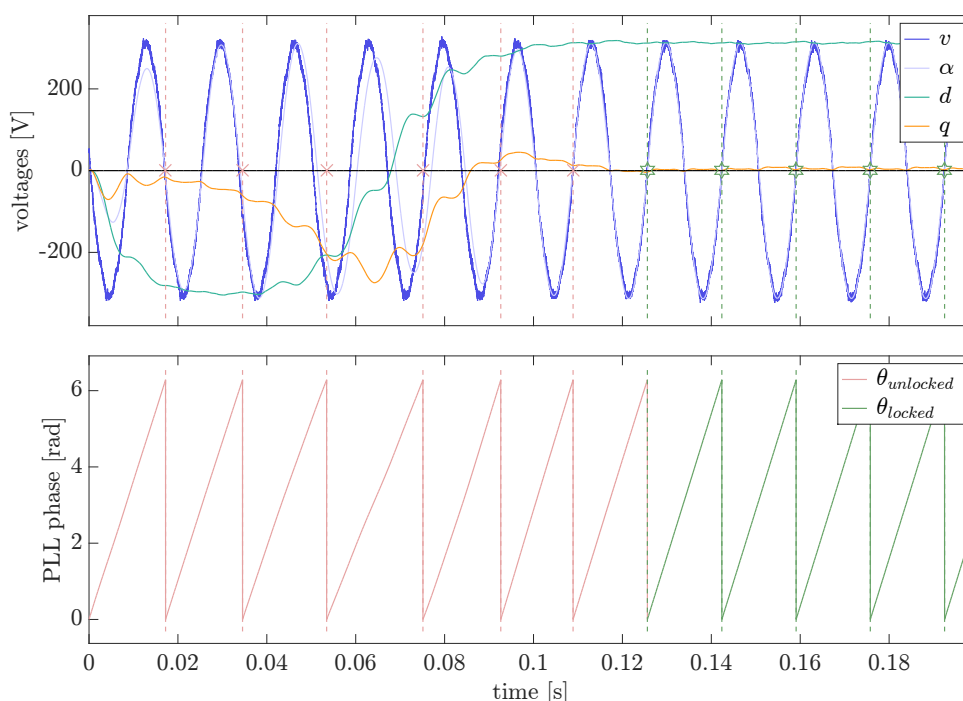


Figura 2: Resultado esperado do script.

Assuma as seguintes constantes para a implementação do controlador e das integrações:

$$\begin{aligned}k_0 &= 0.7 \\k_1 &= 2.840000000003951 \cdot 10^{-6} \\k_2 &= 0.385715277950311 \\k_3 &= -0.385713293478261 \\\omega_0 &= 376.9911184307752\end{aligned}\tag{9}$$

Ainda, exporte para um arquivo `.csv` a porção do sinal de tensão de entrada compreendida entre a detecção da sincronização até o último cruzamento por zero do ângulo  $\theta$  (período que compreende todos os ciclos inteiros em que o PLL estava sincronizado). Exporte juntamente com a tensão o período correspondente do vetor de tempo e do ângulo de saída do PLL.



time [s]	PLL angle [rad]	grid voltage [V]
0.12562	, 0.00158479	, 4.3073
0.12562	, 0.00370203	, 14.323
0.12563	, 0.005819309	, -0.32799
0.12564	, 0.007936621	, 14.62
0.12564	, 0.010053976	, 14.767
⋮	⋮	⋮

Tabela 1: Exemplo da tabela a ser exportada em formato `.csv`.

Questão 2: `rms_f.m`

Os dados exportados na questão anterior consistem num intervalo da tensão da rede com ciclos inteiros. Isso nos permite calcular o valor rms da tensão amostrada da rede através de (10):

$$V_{\text{rms}} = \sqrt{\frac{1}{N} \sum_{n=1}^N v_n^2} \tag{10}$$

Escreva uma função `[Vrms, fl] = rms(nome_arquivo)` que recebe como argumento um texto indicando o caminho do arquivo exportado na questão anterior e que calcule e retorne o valor rms e a frequência fundamental média do sinal.

<sup>†</sup>Note que (10) só é válida para um vetor  $\mathbf{x}$  contendo ciclos inteiros da componente fundamental.

Questão 3: `chuveiro.m`

Um chuveiro elétrico com controle de temperatura entrega à resistência de aquecimento uma tensão cíclica dada por:

$$v_r(\theta) = \begin{cases} 0 & , 0 \leq \theta < \alpha \\ \sqrt{2} \cdot 220 \sin(\theta) & , \alpha \leq \theta < \pi \\ 0 & , \pi \leq \theta < (\pi + \alpha) \\ \sqrt{2} \cdot 220 \sin(\theta) & , (\pi + \alpha) \leq \theta < 2\pi \end{cases} \tag{11}$$

onde o controle da temperatura é feito pelo um ajuste do ângulo de disparo  $\alpha$  de um TRIAC. Assuma frequência de 60 Hz.

Escreva uma função `chuveiro(alpha)` que recebe como argumento o ângulo  $\alpha$  (em radianos) e que a partir deste gera um gráfico da tensão e da potência na resistência de aquecimento, considerando uma resistência de 100  $\Omega$ . Tente não utilizar explicitamente laços `while` ou `for`.

Além de plotar a tensão e a corrente para o intervalo  $0 \leq \theta < 4\pi$ , o algoritmo deve exportar uma figura `chuveiro_{alpha}.png` contendo o gráfico gerado, onde `{alpha}` deve ser substituído e preenchido com o ângulo passado no argumento da função, para o qual os gráficos foram gerados.

Também crie e exporte uma tabela contendo como colunas o tempo, a fase da tensão da rede e a tensão na resistência (similar à Tabela 2) e utilize o algoritmo `rms_f.m`, gerado anteriormente, para verificar a variação da tensão rms na resistência de aquecimento para diferentes valores de  $\alpha$ .

A Figura 3 apresenta dois exemplos, um gerado quando é passado um ângulo de 0.8 rad e outro quando é passado um ângulo de 2.2 rad à função `chuveiro(alpha)`.

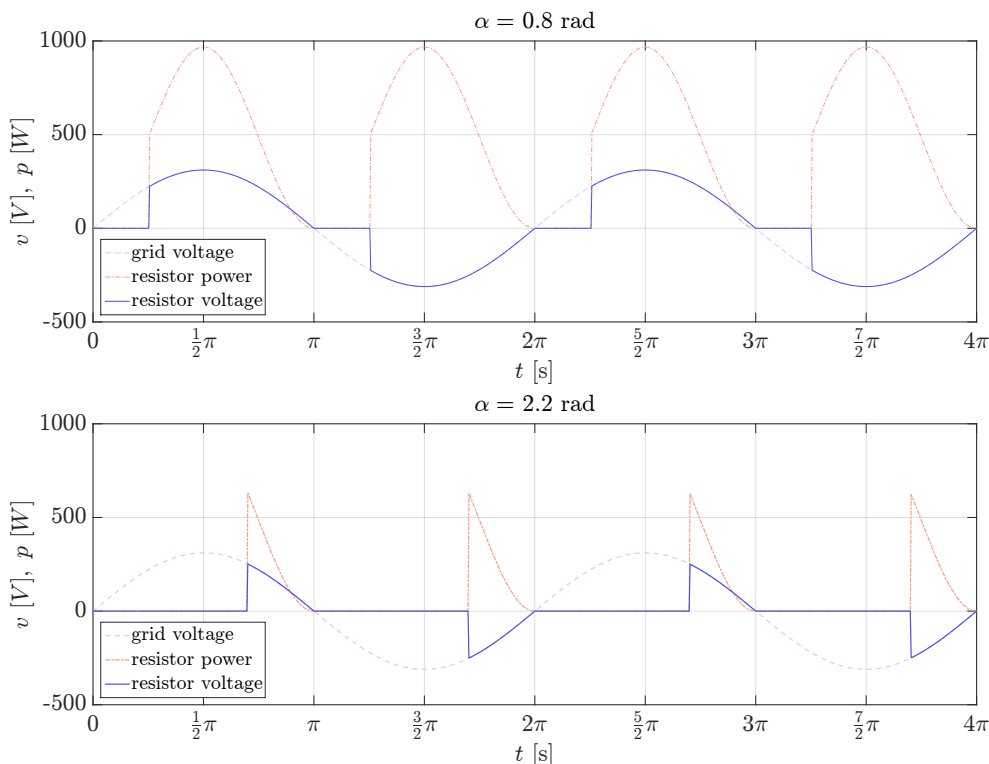


Figura 3: Exemplos dos gráficos gerados pela função quando essa recebe os ângulos de 0.8 rad e de 2.2 rad como argumento.

#### Questão 4: `grafico.m`

Escreva um **script** para plotar a função (12) para o intervalo  $-5 \leq x \leq 30$ .

$$y = \begin{cases} 15\sqrt{4x} + 10 & , x \geq 9 \\ 10x + 10 & , 0 \leq x < 9 \\ 10 & , x < 0 \end{cases} \quad (12)$$

#### Questão 5: `fourier.m`

Uma forma de onda quadrada de amplitude 1 e período  $2\pi$  pode ser aproximada por uma série de Fourier truncada na componente harmônica de ordem  $H$  através de:

$$s(\theta) = \sum_{h=1}^H a(h) \sin(h\theta) \quad (13)$$

onde

$$a(h) = \begin{cases} 4/(\pi h) & , \text{se } h \text{ for ímpar} \\ 0 & , \text{se } h \text{ for par} \end{cases} \quad (14)$$

Escreva um script que ao ser executado solicita ao usuário qual o valor de  $H$  e plota um gráfico da reconstituição da onda quadrada para componentes harmônicas até  $1, 2, 3 \dots H$ . Caso o usuário digite um valor negativo para  $H$  o programa deve apresentar uma mensagem indicando ao usuário a inconsistência e dando a oportunidade deste digitar novamente outro valor válido para  $H$ .