

Atividade H – Processamento digital de sinais aplicado a SEP

1 Instruções gerais

- Esse documento apresenta um exemplo de aplicação de processamento digital de sinais para uso no projeto temático da disciplina.
- Cada grupo de alunos deve utilizar os dados obtidos na simulação anterior realizada no Matlab/simulink para alimentar os algoritmos de processamento digital de sinais de seu dispositivo.

2 Roteiro

2.1 Arquivos para simulação

Faça download dos seguintes arquivos da atividade H a partir do moodle da disciplina:

- ExemploIED_com_simulacao.slx
- ExemploIED_com_sinais.slx
- exemplo_algoritmo.m

Salve ambos os arquivos em uma mesma pasta, em um diretório conhecido de seu computador.

2.2 ExemploIED_com_simulacao

Nesse exemplo, foi criado um bloco no Simulink para representar o algoritmo que opera em tempo de execução dentro de um dispositivo. O bloco é denominado IED (de *Intelligent Electronic Device*), e está programado para ser executado a uma taxa fixa de execução dentro da simulação. Nesse caso, foi escolhido 960 [Hz] para simular um dispositivo que opera com uma amostragem de 16 amostras por ciclo de 60,0 [Hz]. Essa taxa de execução pode ser configurada para outros valores, bastando clicar com o botão direito sobre o bloco e escolher “Block parameters (Subsystem)”, modificando a o parâmetro “Sample Time”.

Clicando duas vezes sobre o bloco, tem-se acesso ao algoritmo básico de funcionamento do dispositivo. Esse algoritmo está escrito em linguagem M (Matlab), mas poderia ter sido constituído em C e outras linguagens suportadas pelo MatLab.

Observe no código que a primeira linha declara a quantidade de entradas e saídas do bloco. Ela pode ser modificada conforme sua aplicação. Ao se alterar e salvar o código fonte, o ícone do bloco no Simulink reporta as mudanças realizadas, mostrando mais ou menos pontos de entrada e saída do bloco.

As linhas a seguir declaram variáveis persistentes, não voláteis, que preservam seu conteúdo entre cada iteração de execução do Simulink. Essas variáveis representam dados em variáveis globais ao código, que deveriam ser armazenados na memória do dispositivo físico real (microcontrolador), permitindo ter uma estimativa do tamanho de memória requerida pelo seu dispositivo para sua operação. Você pode declarar quantas variáveis precisar, mas é importante lembrar que, devido ao fato do Matlab não ter declaração explícita de tipos, essas

Atividade H – Processamento digital de sinais aplicado a SEP

variáveis persistentes devem ser inicializadas em algum ponto do código para que o Matlab compreenda o tipo de dado que será armazenado no seu interior.

Durante o funcionamento do código, alguns dados são inicializados em toda iteração do loop de execução do dispositivo. Entretanto, outros dados como constantes, coeficientes, etc. devem ser inicializados apenas uma vez. Para isso foi criada uma seção no código controlada pela variável 'FunctionSetup', que permite a inicialização, na primeira iteração do bloco, de uma série de constantes. Nas demais iterações esse trecho de código não é mais executado.

É importante notar que nessa inicialização, como exemplo, são computados os coeficientes de filtros, do tipo FIR, para computação da transformada discreta de Fourier, para a componente fundamental do sinal, como para sua terceira harmônica.

Os filtros FIR criados se tratam do filtro de coseno ' $X_C(z)$ ' e de seno ' $X_S(z)$ ' da TDF, dados a seguir:

$$X_C^h(z) = \frac{\sqrt{2}}{N} \sum_{i=1}^N \cos\left(\frac{2\pi \cdot i \cdot h}{N}\right) \cdot x(z) \cdot z^{i-1}$$

$$X_S^h(z) = \frac{\sqrt{2}}{N} \sum_{i=1}^N \text{sen}\left(\frac{2\pi \cdot i \cdot h}{N}\right) \cdot x(z) \cdot z^{i-1}$$

Onde 'N' é o tamanho em amostras da janela de cálculo e 'h' é o índice da harmônica desejada para cálculo da respectiva componente fasorial da TDF.

Esses filtros remetem às seguintes equações de diferenças, que fazem a convolução de 'N' amostras com 'N' coeficientes, para cada um dos filtros:

$$X_C^h(k) = b_0 \cdot x(k) + b_1 \cdot x(k-1) + b_2 \cdot x(k-2) + \dots + b_{N-1} \cdot x(k-N-1)$$

$$X_S^h(k) = d_0 \cdot x(k) + d_1 \cdot x(k-1) + d_2 \cdot x(k-2) + \dots + d_{N-1} \cdot x(k-N-1)$$

Onde 'k' é o índice da amostra mais recente do vetor de entrada 'x', 'k-1' o índice da amostra do instante anterior e assim por diante. Cada um dos 'i'ésimos coeficientes dessas equações dependem da ordem 'h' da harmônica desejada para cálculo, e são dados pelas sequências de cossenos e senos mostradas a seguir, considerando 'i' desde 0 até 'N-1':

$$\text{Coeficientes do filtro FIR coseno: } b_i^h = \frac{\sqrt{2}}{N} \cos\left(\frac{2\pi \cdot (i+1) \cdot h}{N}\right)$$

$$\text{Coeficientes do filtro FIR seno: } d_i^h = \frac{\sqrt{2}}{N} \text{sen}\left(\frac{2\pi \cdot (i+1) \cdot h}{N}\right)$$

Em cada janela de 'N' amostras em que esses filtros são calculados são obtidas as componentes real e imaginária de um fasor, com valores eficazes, ou seja, um fasor X a cada amostra 'k' do vetor de entradas 'x':

$$\dot{X}^h(k) = X_C^h(k) + j \cdot X_S^h(k)$$

Atividade H – Processamento digital de sinais aplicado a SEP

Onde 'h' é a ordem da harmônica calculada.

No código são calculados os coeficientes dos filtros FIR cosseno e seno, da primeira e da terceira harmônica, como exemplo. Cada filtro será usado tanto para processamento de um sinal de tensão como um de corrente nesse exemplo. Assim, como os filtros precisam ter uma janela de 'N' amostras de suas entradas, dois vetores 'LastI' e 'LastV' são criados e inicializados com esse propósito.

No código de exemplo também será computada a potência média ativa do sistema através da sua definição.

$$PMed(k) = \frac{1}{N} \sum_{i=1}^N v(k-i-1) \cdot i(k-i-1)$$

Para isso é necessário ter uma memória para armazenar um período de 'N' amostras da potência instantânea, no vetor 'LastPInst'.

Em seguida à rotina de inicialização controlada pela variável 'FunctionSetup' está o algoritmo de cálculo propriamente dito. Inicialmente os dados filtrados analogicamente, digitalizados, provenientes de conversores AD unipolares, com tensão entre 0 a 3,3V, 12 bits de resolução, são convertidos em dados internos ao IED para os respectivos valores de engenharia. A seguir, são executados os cálculos de parte real e imaginária de fasores para as componentes fundamentais e para uma terceira harmônica. Deve-se notar que, ao final do cálculo do fasor de uma grandeza, é executado um deslocamento no vetor de amostras anteriores (scroll), para dar lugar para uma nova amostra na próxima iteração.

Em sequência, é calculada a potência média pela sua definição, bem com o cálculo da potência complexa com o uso dos fasores de tensão e corrente calculados.

Um exemplo de algoritmo de processamento das medidas é apresentado, para acionar uma saída digital no caso da potência média ser superior a 250,0 [W].

Todas as medidas de interesse são então copiadas para as variáveis de retorno da função (declaradas no topo do programa, em sua primeira linha).

No final do algoritmo de exemplo, é feito um exemplo de exibição periódica de grandezas, com auxílio de um contador de iterações 'cont'.

Ao simular o código no simulink, o usuário terá uma série de grandezas criadas em sua área de trabalho, para posterior exibição, documentação e análise.

Os grupos que quiserem portar esse tipo de bloco de cálculo para seu modelo de Simulink, pode fazer isso através de um simples *copy & paste* do bloco do IED e dos blocos de filtragem, condicionamento e digitalização, para então realizar as devidas conexões aos sinais de TP's TC's e sinais para coleta dos vetores com os resultados de cálculo e *scopes* para exibições instantâneas.

Atividade H – Processamento digital de sinais aplicado a SEP

2.3 ExemploIED_com_sinais

Nesse exemplo, foi criado um bloco no Simulink para representar o algoritmo que opera em tempo de execução dentro de um dispositivo, que lê os dados provenientes de outra simulação realizada previamente no Simulink com o modelo de sua rede.

Para esse tipo de aplicação, os grupos de alunos devem ajustar o final de seus cálculos de filtragem analógica, condicionamento e digitalização para produzir várias estrutura de dados, como mostrado a seguir:

```
%Ch1_VAN, Ch4_IA são vetores produzidos pela simulação da atividade G
tdados = (1/960) * (1:size(Ch1_VAN,1));
Val_Ch1.time = tdados;
Val_Ch1.signals.values = Ch1_VAN;
Val_Ch1.dimensions = 1;

Val_Ch4.time = tdados;
Val_Ch4.signals.values = Ch4_IA;
Val_Ch4.dimensions = 1;
```

As estruturas criadas (Val_Ch1 e Val_Ch4) são consumidas pelos blocos 'From Workspace' nesse novo modelo de Simulink. Ao se clicar no 'play' nesse modelo, todos os dados da primeira simulação são colocadas nas entradas do bloco de simulação do IED, produzindo os resultados em tempo de execução do processamento do algoritmo programado no bloco do IED.

2.4 exemplo_algoritmo.m

Nesse exemplo é demonstrado como emular o funcionamento do algoritmo de um IED, diretamente através de um script, a ser executado em sequência daquele feito para a Atividade G.

O algoritmo é idêntico ao comentado anteriormente, mas deve ser modificado para permitir a iteração por todo o conjunto de dados produzidos na atividade G. Verifique no código as modificações necessárias para o funcionamento e para o registro de todas as quantidades e informações de interesse para documentação de seu projeto.