

Atividade G – Projeto e simulação de filtros analógicos e digitalização

1 Instruções gerais

- Esse documento apresenta um exemplo de projeto, síntese e simulação de sinais, sua aplicação com filtros analógicos *anti-aliasing* e a simulação do processo de digitalização, para uso no projeto temático da disciplina.
- Essas instruções também permitem uma introdução ao uso do Matlab/Simulink, com seu toolbox Simscape/SimPowerSystems, para produção de sinais particulares de um sistema de potência.
- Utilize o software Mathworks Matlab/Simulink a partir da versão 2015a. Ele está disponível nos computadores da sala Energia, na conta/usuário “alunos”.
- Sugere-se que os alunos editem um arquivo de roteiro `procsinais.m` para registrar todos os comandos utilizados para se obter os resultados apresentados. O conteúdo do arquivo pode ser colocado junto do projeto entregue, em um apêndice ao final do texto.

2 Roteiro

2.1 Arquivos para simulação

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

- `SEP_Exemplo.slx`
- `SEP_Exemplo2.slx`
- `procsinais.m`

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

2.2 Breve contato com o Matlab

Procure no “Menu Iniciar” de seu computador o programa Matlab. Execute-o e aguarde seu carregamento. Após ser carregado, o Matlab abre uma janela conhecida como “Command Window”, além de outras janelas de apoio, tais como “Workspace”, “Command history”, “Current folder”, como pode ser visto na Fig. 1. Na janela de “Command window” o usuário pode digitar uma série de comandos em um *prompt*, após o caractere `>`. Os comandos são digitados e executados ao final com a tecla ENTER, seja para a configuração do ambiente, seja para o desempenho de algum algoritmo ou programa criado pelo usuário, etc.

Em linhas gerais o Matlab é uma linguagem de programação interpretada, com vários conjuntos de rotinas de apoio (*toolboxes*) para diversas áreas da ciência, tais como engenharias, física, matemática, etc. Um de seus recursos mais interessantes, e talvez seu maior diferencial para outras linguagens, está na capacidade de tratar dados vetoriais e matrizes, como necessário na solução de sistemas lineares, com enorme facilidade para o usuário. O programa é bastante utilizado para a prototipagem e a simulação de algoritmos, uma vez que possui um amplo espectro de rotinas e funções de apoio, uma vasta gama de tipos de variáveis e estruturas de programação, além de não requerer a declaração ou a alocação de variáveis.

Atividade G – Projeto e simulação de filtros analógicos e digitalização

Além de ferramentas para programação, o *software* possui uma série de recursos para visualização científica e análise de dados, tais como gráficos 2D e 3D, tabelas, animações, etc.

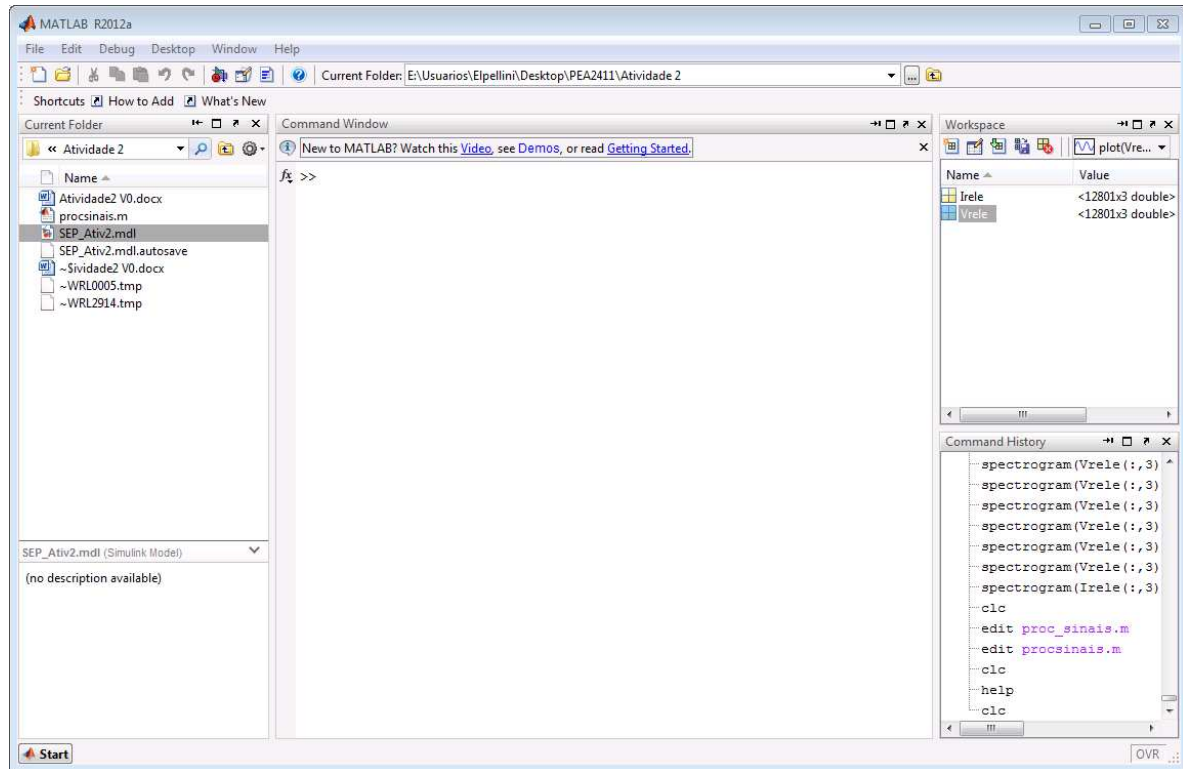


Fig. 1 – Janela de comandos do Matlab. Na parte da esquerda, a pasta de trabalho, ao centro a janela de comandos, e à direita as variáveis do ambiente de trabalho (workspace) e histórico de comandos.

Algumas informações básicas a respeito do programa são citadas a seguir:

- O Matlab possui uma série de funcionalidades, para inúmeras áreas da engenharia, disponibilizada através de *toolboxes*, comprados individualmente. Nem sempre todos os *toolboxes* estão instalados em um Matlab e, por conta disso, nem todos os comandos e funções podem ser reconhecidos.
- *Toolboxes* importantes para a área de automação de sistemas elétricos são aqueles que permitem a simulação de sistemas elétricos de potência (*Simulink Simscape\SimPowerSystems*), o *Control System Toolbox* e o *Signal Processing Toolbox*.
- O Matlab possui diversos tipos de dados primitivos, como números de ponto flutuante, números inteiros, *strings*, caracteres, etc. Também possui estruturas de dados, células, listas, vetores e matrizes. Vetores e matrizes utilizam como indexação o número 1 para o primeiro elemento e não o número 0, como é comum em outras linguagens de programação.
- O programa não requer a declaração prévia de variáveis. Ao se atribuir um valor a uma variável ela é automaticamente declarada, com seu tipo adequado ao dado associado. *Strings* ou vetores de caracteres são mencionados utilizando-se o caractere aspas simples ' como delimitador. Vetores e matrizes tem sua declaração feita com o uso dos caracteres colchetes [e]. Ex.: `a=1; b=1.0; c='teste'; A=[[11 12]; [21 22]]`; `A(1,2)=44`

Atividade G – Projeto e simulação de filtros analógicos e digitalização

- Colocando-se o ponto e vírgula ao final de um comando, seu resultado não ecoa na janela de comandos. Sem o ponto e vírgula, o Matlab exibe o resultado do comando na tela. Pode-se usar o ponto e vírgula para se separar vários comandos em uma mesma linha. Ex.: `clear all; a=1; b=a+i*a;`
- A parte imaginária de um número complexo é denotada pela letra minúscula “i”. Não use “i” como uma variável ou contador em seu programa. Um número complexo em coordenadas polares deve ser inserido em coordenadas retangulares. Ex.: `a=20*cos(pi/4)+i*20*sin(pi/4)`
- Para obter ajuda com o Matlab, utilize o comando `help`. Ele pode ser sucedido por um comando ao qual se deseja auxílio. Ex. `help matlab\general;` Verifique para que serve os comandos `clear`, `clc`, `who`, `whos`, `plot`, `demo`, `cd`, `dir`, `arith`, `lang`, `datatypes`.
- Podem ser descobertos outros comandos relacionados com alguma necessidade através do comando `lookfor <palavra_chave>`. O Matlab retornará uma lista de comandos relacionados com a palavra chave (ou palavras chave, se delimitadas pelas aspas simples) citada. A lista resultante pode ser bastante ampla. Por exemplo: `lookfor 'transfer function';`
- Uma lista de comandos digitada manualmente, um a um, pode ser também escrita em um arquivo texto, com extensão `.m`, para servir como *script* ou roteiro. Digitando o nome do arquivo *script* criado no *prompt* do Matlab, sem o `.m`, promove a execução da sequência de operações descrita em seu interior, de forma automática. Você pode editar um *script* de dentro do próprio Matlab, digitando `edit roteiro.m`
- Ao executar *scripts* e roteiros, verifique se você está operando com o Matlab no diretório correto onde estão localizados os arquivos. Isso pode ser visto ao se usar o comando `cd`

Um bom ponto de partida para se familiarizar com as diversas funcionalidades do Matlab é seguir a página “Getting Started with Matlab”, disponível em <http://www.mathworks.com/help/matlab/getting-started-with-matlab.html>. Veja as demonstrações (demos) e exemplos disponíveis.

Existem diversas outras referências, mesmo em língua portuguesa, de como utilizar o programa, basta procurar em algum mecanismo de busca na Internet.

2.3 Rodando uma simulação do Simulink

O Matlab possui um arcabouço (*framework*) de simulação para sistemas dinâmicos denominado Simulink. O Simulink pode ser acionado através da execução do comando `simulink` na janela de comandos do Matlab. Rode o Simulink e veja sua interface, procure por exemplos e demonstrações. Um bom ponto de partida é a página “Getting Started with Simulink”, disponível em: <http://www.mathworks.com/help/simulink/getting-started-with-simulink.html>. Procure pelos tutoriais e exemplos.

O Simulink permite a descrição de sistemas dinâmicos, e sua simulação numérica, através de diagramas de blocos funcionais, dos mais diversos. Em suas versões mais recentes, o Simulink conta com um simulador de redes elétricas, com elementos de sistemas de potência, em um toolbox denominado Simscape\SimPowerSystems. Podem ser criados sistemas elétricos de elevada complexidade, para simulação de transitórios eletromagnéticos, transitórios eletromecânicos, regime permanente, fluxo de potência, etc. Para isso, basta iniciar um novo modelo (File, New) arrastar e soltar os blocos disponíveis na paleta do programa para a área de trabalho, interligar as entradas e saídas, parametrizar os blocos internamente (clicando duas vezes sobre cada bloco), para finalmente

Atividade G – Projeto e simulação de filtros analógicos e digitalização

executar o sistema através do botão *Play* (ou no menu *Simulation, Start*), disponível na barra de menus do programa.

Os resultados do Simulink podem ser apresentados por meio de gráficos, geração de arquivos, ou a produção de variáveis dentro do ambiente de linha de comando do Matlab.

Toda a modelagem feita no Matlab Simulink deve ser gravada em um arquivo. A extensão utilizada pelo Matlab Simulink são os arquivos do tipo *.mdl* ou *.slx*.

2.4 Script para análise dos dados

O arquivo `procsinais.m` é um arquivo texto comum. Ele contém alguns exemplos de como processar as informações produzidas após a execução das simulações do Simulink. Sugere-se que os alunos editem esse arquivo e coloquem a listagem de comandos que produz os gráficos e análises solicitados nessa atividade. Esse arquivo pode ser chamado diretamente no prompt do Matlab, produzindo todos os gráficos e análises, de forma automática.

2.5 Executando uma simulação e um script para análise dos dados

Nos arquivos que você obteve para a atividade existe alguns arquivos com um modelo de um sistema elétrico de potência, denominado `SEP_ExemploX.slx`.

2.5.1 Mudando o diretório para o local de trabalho dos arquivos

Feche qualquer janela do programa que tenha aberto até o momento, a não ser a janela de linha de comando (*prompt*) do Matlab. Dado que deve ser conhecida a pasta onde estão localizados os arquivos baixados em seu computador, por exemplo:

```
E:\Usuarios\Elpellini\Desktop\PEA3411\AtividadeF
```

No *prompt* do Matlab digite o comando:

```
cd 'E:\Usuarios\Elpellini\Desktop\PEA3411\AtividadeF';
```

Atenção: Pode-se confirmar em que diretório o Matlab se encontra digitando o comando `cd` sem argumentos.

2.5.2 Abrindo um arquivo do Simulink e executando a simulação

O grupo de alunos pode usar uma das duas redes de exemplo para geração de sinais representativos de uma rede elétrica particular.

2.5.2.1 Linha de transmissão SEP_Exemplo

No *prompt* do Matlab, digite:

```
open 'SEP_Exemplo.slx'
```

Ou, se o Simulink já estiver aberto, em seu menu, escolha a opção *File, Open*, e procure o arquivo *.mdl* desejado. Isso abrirá o modelo do sistema elétrico de potência que será simulado.

Atividade G – Projeto e simulação de filtros analógicos e digitalização

O sistema em simulação se trata de duas fontes ideais, com seus equivalentes de *Thévenin*. As fontes estão interligadas por duas linhas de 138,0 [kV], em dois trechos: um trecho de 40,0 [km] de extensão e um trecho de 150,0 [km] de extensão, como pode ser visto na Fig. 1.

No ponto de conexão das duas linhas há uma barra denominada B1, onde existem transformadores de corrente e tensão ideais, capazes de medir as tensões de fase e as correntes de linha nesse ponto do sistema elétrico, em valores instantâneos, em [V] e [A] respectivamente.

É importante ressaltar que a linha 1 foi dividida ao meio, resultando em dois trechos de 20,0 [km] em série (L1R e L2R). A linha 2 também foi dividida em dois trechos, com o trecho L1F permanecendo com 50,0 [km] e o trecho L2F com 100,0 [km].

Essa divisão foi feita propositalmente para a inserção de blocos de simulação de faltas (FR e FF no sistema), para que possam ser representados defeitos do tipo curto-circuitos, entre fase-terra, dupla-fase terra, trifásicos, etc. Dessa forma, esses tipos de eventos podem ser simulados nesses pontos do sistema, para posterior estudo. O interessante desse caso de simulação é que, devido ao comprimento das linhas, os fenômenos de reflexão e transmissão de ondas viajantes que ocorrem nos eventos de curto-circuito ocasionam transitórios nas formas de onda de tensão e corrente, com componentes harmônicas diversas e distorções bastante perceptíveis. Nesse caso, será premente a filtragem analógica dos sinais para condicionamento de seu espectro, evitando recobrimentos (*aliasing*) no processo de digitalização.

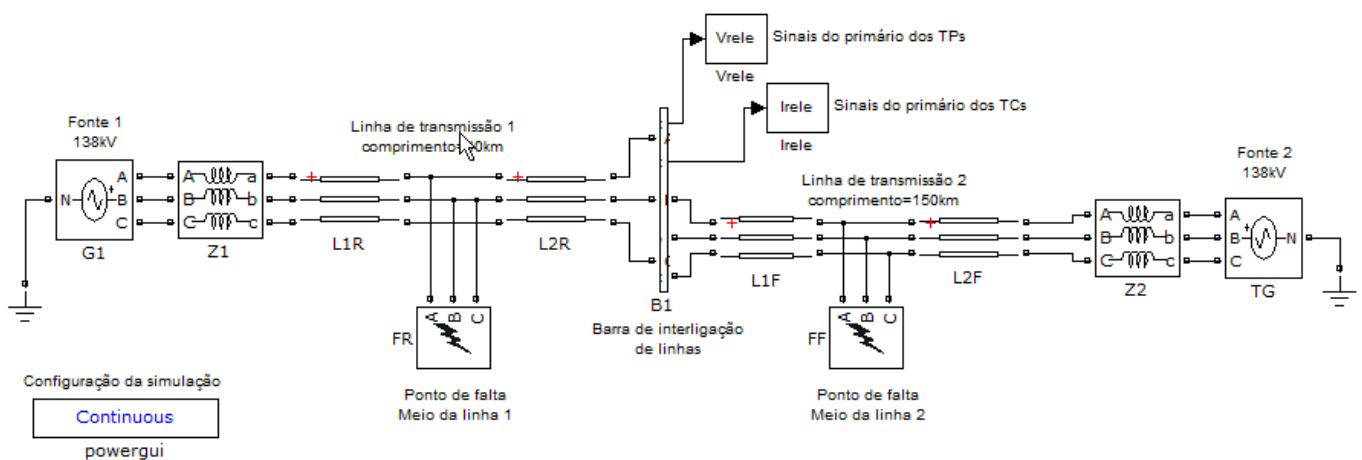


Fig. 2 - Sistema Exemplo para simulação.

Cada um dos blocos da Fig. 2 possui parâmetros e ajustes próprios. Para verificar ou alterar essas configurações, clique duas vezes em cada bloco do sistema.

Para executar a simulação, acione o botão *Play* (ou menu *Simulation, Start*) disponível na barra de menus do programa. O tempo padrão de simulação é de 10/60 ou 10 ciclos de 60,0 [Hz] = 166,66 [ms]. Para executar a simulação de 166,66 [ms] o Simulink leva cerca de 5 a 10 segundos, dependendo da capacidade do computador do usuário.

Atividade G – Projeto e simulação de filtros analógicos e digitalização

2.5.2.2 Rede de distribuição com geração distribuída SEP_Exemplo2

Abra o modelo do Simulink 'SEP_Exemplo2.slx':

```
open 'SEP_Exemplo2.slx'
```

O sistema em simulação se trata de um ramal de distribuição simplificado que alimenta, entre outras cargas, um hospital e uma empresa de ônibus elétricos. Do lado da concessionária de energia há uma fonte ideal com equivalente de Thévenin ligado a uma barra B1.

Na barra B2 existe uma carga local de consumidores residenciais e a empresa de ônibus elétrico, que possui um grande retificador trifásico não controlado, sem filtros harmônicos, alimentando vários carregadores de bateria, agrupados em uma carga RL equivalente, cuja corrente DC total é medida por um transdutor de corrente tipo resistor Shunt, com medida direta em [A]. Na barra B2 existe um IED que mede as tensões de fase e as correntes de linha que são aportados até um segundo trecho de rede elétrica.

Na barra B3 existe a conexão de um hospital à rede primária, além de um pequeno grupo de consumidores residenciais. Esse hospital possui um determinado consumo interno e um recurso de cogeração, que consiste em um turbogerador síncrono que pode utilizar o excedente de produção de vapor da caldeiraria do hospital para a geração de energia elétrica, seja para próprio consumo ou exportação à concessionária de energia.

O sistema é ilustrado em um modo de operação tipicamente noturno, de madrugada, quando a carga interna do hospital está bastante reduzida e sua cogeração está operando em quase 65% de potência. Enquanto isso, na empresa de ônibus, o retificador está funcionando a plena carga para recarregar as baterias dos veículos que serão utilizados durante o dia.

Essa rede possui uma forte distorção harmônica já em regime permanente senoidal devido às cargas não lineares do retificador ligado à rede elétrica.

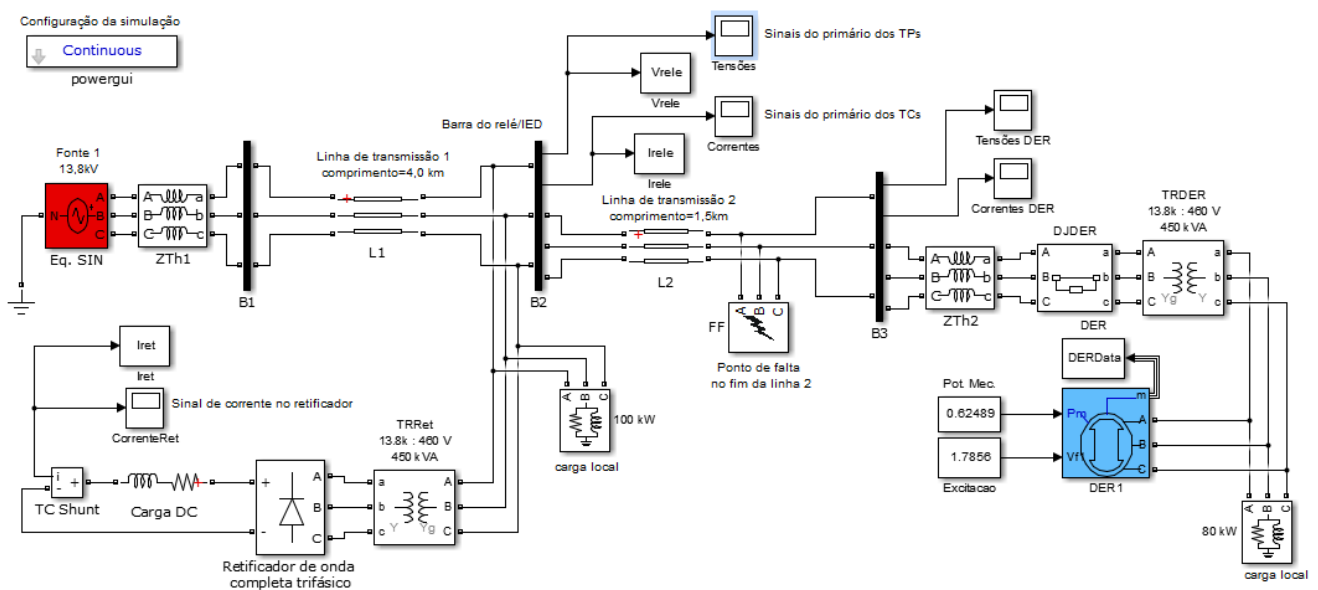


Fig. 3 - Sistema Exemplo2 para simulação.

Atividade G – Projeto e simulação de filtros analógicos e digitalização

Para simulação de alguns transitórios mais severos, como um curto-circuito, foi colocado um bloco de simulação de falta FF em um dos trechos de linha do sistema. Isso pode ser visto com mais detalhes na Fig. 3 a seguir.

Para executar a simulação, acione o botão *Play* (ou menu *Simulation, Start*) disponível na barra de menus do programa. O tempo padrão de simulação é de 10/60 ou 10 ciclos de 60,0 [Hz] = 166,66 [ms]. Trabalhando com os dados da simulação.

2.5.3 Trabalhando com os dados da simulação

Os alunos são estimulados a modificar as redes de exemplo apresentadas, para representar seus casos particulares de aplicações do dispositivo que está sendo projetado no âmbito da disciplina. O objetivo é ter vetores com os valores dos sinais da rede elétrica para tratamento por seus algoritmos.

Após a simulação, na janela principal do Matlab, na seção *Workspace*, devem ter sido criadas ao menos duas variáveis: *Vrele* e *Irele*. Detalhes sobre essas variáveis podem ser obtidos ao se digitar o comando *whos* no *prompt*. Alguns outros vetores como *IRet* são usados para demonstrar grandezas não necessariamente senoidais (corrente DC do retificador) que também podem ser objetivo de análise por um equipamento de medição, automação, controle ou proteção de redes elétricas.

Vrele e *Irele* são duas matrizes, cada uma com três colunas e diversas linhas contendo cada um dos valores instantâneos de tensão e corrente da rede na barra B1. São valores sintéticos obtidos na simulação que representam as quantidades analógicas envolvidas. Eles foram registrados com uma frequência de amostragem exageradamente alta para emular um sinal no domínio do tempo contínuo (1280 amostras por ciclo de 60,0 [Hz] ou $1280 \cdot 60 = 76,8$ [kHz]). As grandezas são registradas em valores de ponto flutuante de dupla precisão, nas variáveis de engenharia de tensão [V] e corrente [A]. A primeira coluna de cada variável corresponde às grandezas primárias da fase A, a segunda da fase B, e a terceira da fase C. O registro contém os sinais do sistema elétrico durante o transitório resultantes da aplicação de um curto circuito trifásico-terra, franco (com baixa resistência), no instante $t = 2,03/60 = 33,833$ [ms], no ponto de falta FF.

Para se visualizar os dados, crie um vetor com estampas de tempo, espaçados do período de amostragem de $1/(1280 \cdot 60)$, com valor mínimo entre 0 e o comprimento da simulação, de 10/60 [s]. Digite no *prompt*:

```
t = 0 : 1/(1280*60) : 10/60;
```

Este vetor tem sua primeira posição $t(1) = 0$, e última $t(\dots) = 10/60$, com uma grande quantidade de amostras, que pode ser conhecida através do comando `size(t)`. A seguir, faça o gráfico da tensão na fase A vista na barra B1 com o seguinte comando:

```
plot(t, Vrele(:,1));  
grid;  
title('Tensão na fase A');  
legend('VAN');  
xlabel('Tempo [s]');  
ylabel('[V]');
```

Atividade G – Projeto e simulação de filtros analógicos e digitalização

Na primeira linha acima, a matriz 'Vrele' é citada com sua primeira coordenada utilizando o caractere ':' e a segunda com o número '1'. Isso denota ao Matlab que o comando deve usar todas as linhas da primeira coluna da matriz 'Vrele' no desenho do gráfico. Para vetores unidimensionais, como o 't', não é necessário explicitar 't(:)' para que o Matlab considere todas as suas linhas.

O comando plot constrói o gráfico de todos os valores do vetor de tempo 't' versus todas as linhas da primeira coluna da matriz 'Vrele'. A segunda linha adiciona um reticulado sob o gráfico. Da terceira à última linha são adicionados um título, legenda da curva, legenda do eixo das abcissas e legenda para o eixo das ordenadas.

Explore as ferramentas da janela do gráfico criado e veja seus recursos de zoom, colocação de cursores, exportação e impressão.

Podem ser criados gráficos com múltiplas curvas, simplesmente adicionando-se os demais vetores de pontos x,y, ao comando plot, separados por vírgulas, como a seguir:

```
plot(t, Vrele(:,1), t, Vrele(:,2), t, Vrele(:,3));  
grid;  
title('Tensões na barra B1');  
legend('VAN', 'VBN', 'VCN');  
xlabel('Tempo [s]');  
ylabel('[V]');
```

Atenção: cada vez que o comando 'plot' é executado, o gráfico anterior é sobrescrito pelo novo gráfico. O comando figure pode auxiliar na criação de diversos gráficos em janelas distintas. Use help figure para maiores detalhes.

Podem ser criados outros sinais a partir do sinal original, como por exemplo:

```
vn = Vrele(:,1) + Vrele(:,2) + Vrele(:,3);  
plot(t,vn);
```

Isso irá gerar um novo vetor 'vn' que contém, em cada linha, a soma dos valores instantâneos das tensões da fase A, fase B e fase C do sistema, resultando na tensão de neutro.

É importante observar que apesar dos operandos 'vn', 'Vrele(:,1)', 'Vrele(:,2)', etc., usados na aritmética acima se tratarem de vetores, o Matlab compreende que o usuário deseja executar uma operação vetorial e não escalar, nesse caso, somando cada um dos elementos de cada linha, e armazenando em um novo vetor 'vn'.

Para se compreender melhor a simplificação feita nesse tipo de sintaxe "vetorial", um algoritmo equivalente para constituir um vetor 'vn', usando um *loop* iterativo *for*, seria algo como o mostrado a seguir.

```
for j = 1:size(Vrele(:,1))  
    vn(j) = Vrele(j,1) + Vrele(j,2) + Vrele(j,3);  
end
```


Atividade G – Projeto e simulação de filtros analógicos e digitalização

De forma similar podem ser feitas operações de multiplicação entre vetores. Entretanto, deve-se atentar que o Matlab interpreta as multiplicações feitas com o caractere '*' como multiplicações matriciais se seus operandos forem vetores, ou seja, um vetor $n \times m$ multiplicado por outro vetor $m \times n$ resulta em uma matriz $n \times n$. E nesses casos, devem-se verificar com cuidado as dimensões dos vetores envolvidos, uma vez que um vetor de 1000 linhas e 1 coluna, quando multiplicado por outro vetor de 1000 linhas e 1 coluna, resultara em erro, já que as dimensões não estão consistentes (a segunda matriz deveria ser transposta, com 1 linha e 1000 colunas, para que uma multiplicação matricial funcione corretamente). Para transpor um vetor no Matlab, basta utilizar o caractere de apóstrofo após o nome do mesmo (Ex.: `Vrele(:,1)'`).

Para explorar a multiplicação (e divisão) no Matlab em seu aspecto de operação escalar ou matricial, é demonstrado a seguir um exemplo para se calcular a potência instantânea dos sinais de uma fase, usando um método algo tradicional, com um *loop for* iterativo.

```
for j = 1:size(Vrele(:,1))
    pa(j) = Vrele(j,1) * Irele(j,1);
end
```

Ao invés de fazer o loop, pode-se fazer uma multiplicação automática de cada elemento de cada linha, gerando automaticamente o vetor de saída. Para esse tipo de álgebra o Matlab possui um operador de multiplicação especial, constituído por um caractere ponto seguido pelo sinal de multiplicação (`.*`). Ele permite o cálculo das multiplicações das grandezas, em cada linha dos vetores, de forma automática. Ou seja, as operações citadas acima podem ser substituídas simplesmente por:

```
pa = Vrele(:,1) .* Irele(:,1);
```

2.6 Exemplo de simulação, projeto e aplicação de filtros analógicos

Nesse exemplo, será simulado um processo de condicionamento, filtragem e digitalização de sinais de um sistema elétrico de potência. Para isso, serão utilizados os sinais de tensão e corrente vistos na barra B1 do sistema mostrado na Fig.2 ou Fig. 3, amostrados com uma frequência de aquisição bastante elevada, aproximando as respostas de um sistema em tempo contínuo por sinais em tempo discreto com um intervalo de tempo muito pequeno entre amostras.

2.6.1 Configuração e execução da simulação

Se simulados da forma como estão, os resultados desses modelos apresentam o comportamento das tensões e correntes em regime permanente senoidal. Entretanto, para análise de transitórios podem ser feitos diversos tipos de curto-circuitos nos blocos FF e FR das redes. Nesse exemplo mostrado a seguir, serão simuladas faltas entre a fase C e terra, com uma resistência de falta para a terra de $8/10 = 0,8 \text{ } [\Omega]$. O ponto de aplicação do curto-circuito será feito no bloco FF do modelo SEP_Exemplo da Fig. 1.

Deve-se clicar duas vezes sobre o bloco FF no modelo do Simulink para configurar corretamente o tipo e a resistência de falta, como desejado. Isso pode ser visto na Fig. 4.

Atividade G – Projeto e simulação de filtros analógicos e digitalização

Atenção: Ambos os blocos (FF e FR) devem ser configurados adequadamente. Apenas o bloco FF deve estar configurado para a falta. Se isso não for feito corretamente, o usuário poderá estar simulando um caso (exótico) de faltas simultâneas, quando dois eventos de curto-circuitos ocorrem em dois pontos diferentes do sistema de potência, ao mesmo tempo. O SEP_Exemplo2 não possui outro bloco de simulação de falta.

Na Fig. 4, devem-se observar os parâmetros das fases envolvidas, e do envolvimento e valor da resistência de falta para a terra. Os demais parâmetros devem permanecer inalterados.

Atenção: Os parâmetros numéricos de ponto flutuante devem ser indicados com ponto decimal. Ou seja, uma resistência de 0,8 [Ω] será configurada como 0.8

Os demais parâmetros de tensões, impedâncias e tempos de simulação devem ser mantidos inalterados. Simule o circuito sob falta para que sejam gerados os vetores de Vrele e Irele. Crie um vetor de tempo auxiliar como mostrado na seção 2.5.3.

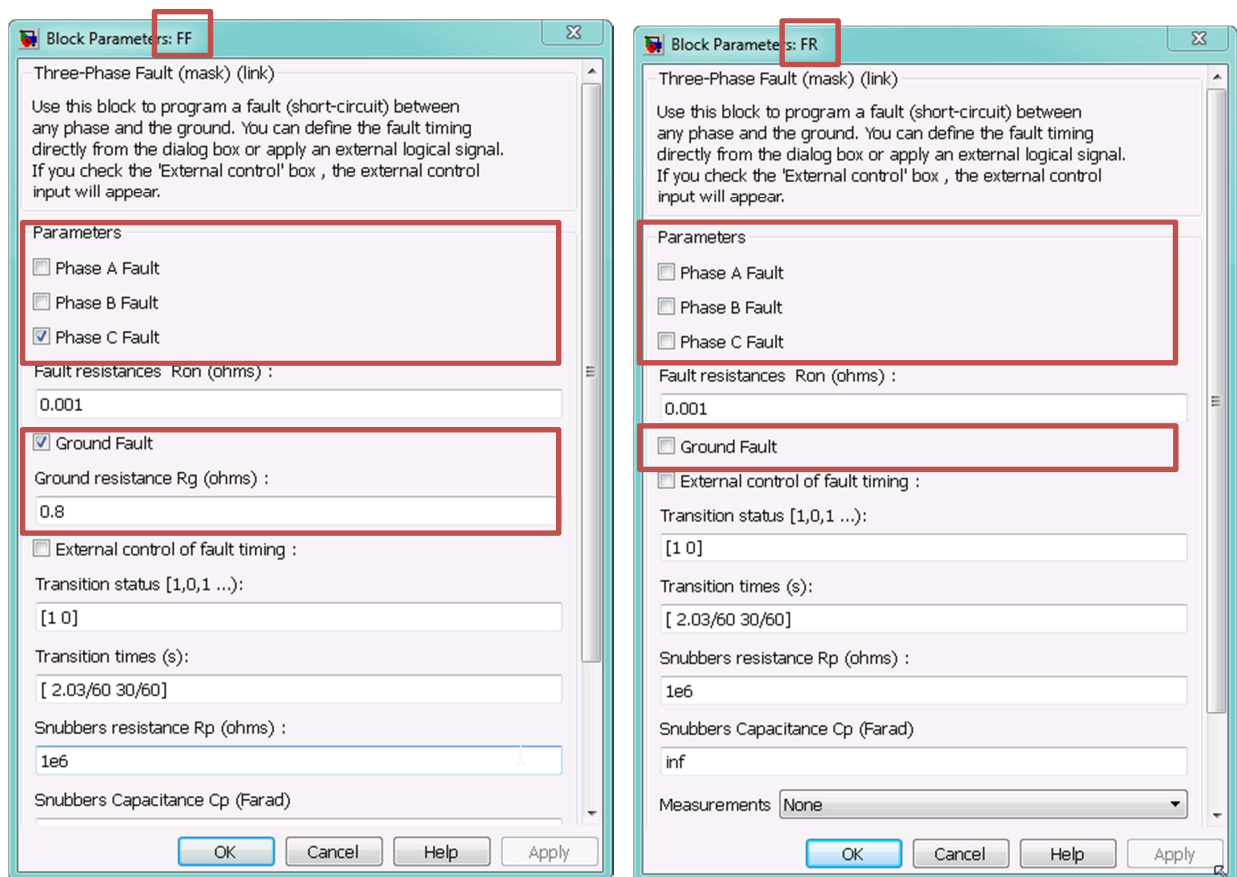


Fig. 4 – Configuração da falta. Se a falta deve ocorrer no bloco FF, então o bloco FR não deve ter nenhum envolvimento de fase ou terra.

Atividade G – Projeto e simulação de filtros analógicos e digitalização

2.6.2 Aplicação de TPs e TCs lineares

Os transformadores de potencial e corrente reais possuem, além de sua relação de transformação nominal, uma precisão associada a suas medidas (que pode variar entre 0.3, 0.6, 1.2 e 3.0 [%]) e um determinado grau de linearidade. Sua precisão e linearidade variam com a intensidade e natureza (indutiva ou capacitiva) da carga em seu secundário. Nesse exemplo, serão considerados TCs e TPs ideais, com respostas lineares, e sem erros de precisão.

No caso da rede SEP_Exemplo, considerando a existência de TPs com relação de transformação de tensão de 138000 : 115, constitua os sinais de tensão secundária de cada fase. Ex.:

```
VANs = Vrele(:,1) * (115/138000);
VBNs = Vrele(:,2) * (115/138000);
VCNs = Vrele(:,3) * (115/138000);
```

Considerando TCs em cada medida da corrente de linha, com relação de transformação de corrente de 800 : 5, constitua os sinais de corrente secundária IAs, IBs e ICs e a corrente de neutro INs. Ex.:

```
IAs = Irele(:,1) * (5/800);
IBs = Irele(:,2) * (5/800);
ICs = Irele(:,3) * (5/800);
INs = (Irele(:,1) + Irele(:,2) + Irele(:,3)) * (5/800);
```

Verifique:

- Plote os sinais de tensão e corrente em dois gráficos distintos e sinalize o instante de aplicação da falta nos gráficos.
- Identifique e meça quais as máximas amplitudes registradas nos sinais de tensão e corrente.
- Note a origem das oscilações de frequência elevada nos sinais de tensão do sistema. Procure analisar o porquê das oscilações de alta frequência serem menos pronunciadas nos sinais de corrente.
- O período antes da falta é denominado pré-falta. Verifique se possível identificar qualquer tipo de desequilíbrio nas correntes da rede na situação de pré-falta. Observe a corrente de neutro calculada para tirar essa conclusão.

2.6.3 Condicionamento dos sinais em um IED de medição

Considere que os sinais dos secundários dos TCs e TPs ligados à barra B1 são ligados, por exemplo, a um IED (*Intelligent Electronic Device*, ou um relé digital multifunção moderno) com a seguinte arquitetura de entradas analógicas e esquema de ligação, como mostrado nas Figs. 5 e 6.

Trata-se de um IED de medição com três canais de tensão de fase, três canais de corrente de linha, e um canal de corrente de neutro. O canal de corrente de neutro possui um fator de conversão diferente dos demais canais para que seja possível observar a corrente de neutro com maior resolução após o processo de digitalização. O equipamento possui um conversor AD por canal de entrada analógica, com aquisição simultânea.

Atividade G – Projeto e simulação de filtros analógicos e digitalização

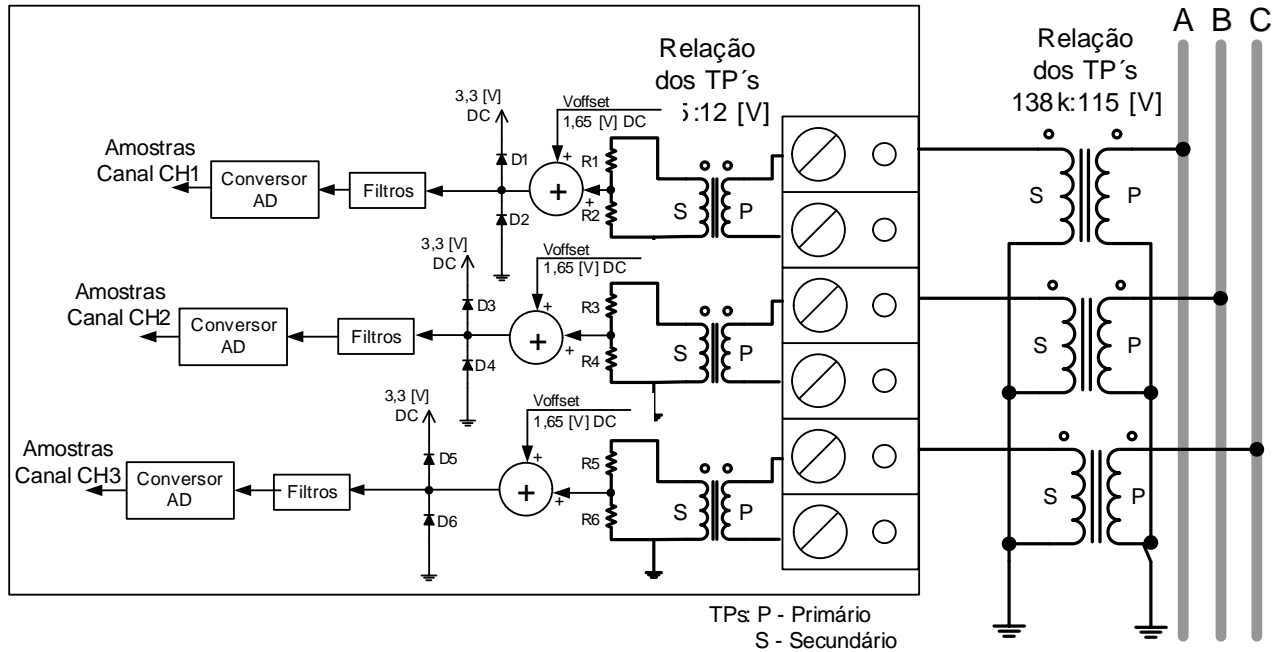


Fig. 5 – Entradas analógicas de tensão em um IED.

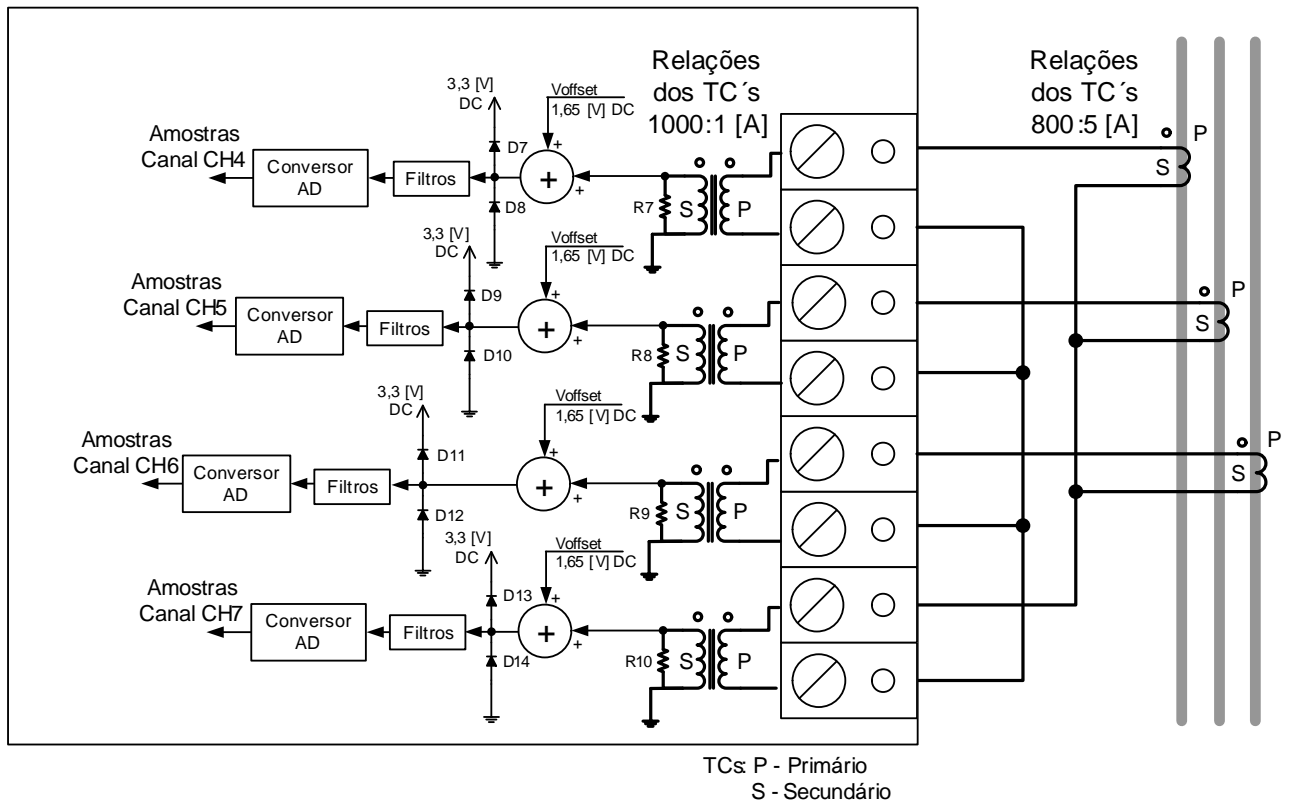


Fig. 6 – Entradas analógicas de corrente em um IED.

Atividade G – Projeto e simulação de filtros analógicos e digitalização

Sabe-se que:

- Os filtros citados nas Figs. 5 e 6 são passa-baixa com circuitos com alta impedância de entrada, não consumindo corrente considerável dos circuitos que estão em sua entrada. Esse filtro é praticamente um ganho unitário para baixas frequências.
- Os diodos mostrados (D1 a D14) são ideais, servindo para limitação do sinal analógico que será aplicado na entrada dos filtros e conversor AD, entre 0,0 [V] e 3,3 [V], nesse caso. Na prática esses diodos devem ter uma baixa tensão direta de condução para que seu efeito de grampeamento da tensão ocorra próximo das tensões limites. Foram omitidos resistores de limitação de corrente para esses diodos.
- Os conversores AD utilizados em cada canal são considerados do tipo unipolar, de 12 bits de resolução, tempo de conversão insignificante perto da frequência de amostragem utilizada, *sample-hold* interno, tensão de entrada entre 0,0 [V] a 3,3 [V], com saídas digitais codificadas em binário, de forma que a menor tensão de entrada é representada em decimal pelo número 0, e a maior tensão de entrada é representada em decimal pelo número $2^{12}-1 = 4096-1 = 4095$. O AD só permite representar em sua saída números inteiros entre tais valores extremos.

Exemplos de exercício:

- A) Projete os valores dos divisores resistivos das entradas de tensão (pares de resistores R1 e R2, R3 e R4, R5 e R6), de forma que para uma tensão de entrada eficaz de 1,5 [p.u.] na rede primária (tensão nominal de linha de 138,0 [kV]), o sinal que deverá estar presente na entrada do AD ocupe toda a faixa dinâmica de entrada, entre 0,0 e 3,3 [V]. Ou seja, para uma tensão de 1,5 [p.u.] eficaz na rede, o valor de pico positivo na saída do divisor resistivo deve resultar em 3,3 [V], enquanto que o valor de pico negativo na saída do divisor resistivo deve resultar em 0,0 [V]. Verifique que o sinal sofre um deslocamento de +1,65 [V] para se tornar unipolar. Considere que os resistores tem potência máxima dissipada de $1/8$ [W] = 0,125 [W]. Considere que todas as entradas analógicas de tensão são idênticas.
- B) Projete os valores dos resistores de shunt no secundário dos TC's das entradas de corrente de fase (resistores R7, R8 e R9), de forma que para uma corrente eficaz de entrada de 5,0 [p.u.] na rede primária (admitindo a corrente nominal como 800 [A]), o sinal de queda de tensão produzido no resistor de shunt (depois de deslocado de +1,65 [V]), esteja presente na entrada do AD ocupando toda sua faixa dinâmica de entrada, entre 0,0 e 3,3 [V]. Considere resistores com potência máxima de $1/8$ [W]. Considere que as entradas para as correntes de fase A, B e C sejam idênticas.
- C) Projete o valor do resistor de shunt no secundário do TC da entrada de corrente de neutro (resistor R10), de forma que para uma corrente eficaz de neutro na entrada de 0,1 [p.u.] na rede primária (80,0 [A]), o sinal de queda de tensão produzido no resistor de shunt (depois de deslocado de +1,65 [V]), esteja presente na entrada do AD ocupando toda sua faixa dinâmica de entrada, entre 0,0 e 3,3 [V]. Considere um resistor com potência máxima de $1/8$ [W].

Atenção: o limite superior das faixas dinâmicas dos canais de corrente e tensão podem ser feitos diferentes dos estipulados, conforme os requisitos de máxima tensão ou corrente que seu projeto temático está considerando. Tipicamente, para aplicações de medição e controle são usados de 1.5 a 2.0 [p.u.] de excursão para os sinais de tensão, e de 1.0 a 2.0 [p.u.] para os sinais de corrente. Para aplicações de proteção, os canais de corrente são

Atividade G – Projeto e simulação de filtros analógicos e digitalização

feitos para poder lidar com correntes máximas entre 15.0 a 50.0 [p.u.], dependendo da potência de curto circuito no local de aplicação.

2.6.4 Circuito de condicionamento das entradas analógicas

Considere que os sinais obtidos na simulação com o Matlab Simulink sejam aplicados às entradas de tensão e corrente mostradas e projetadas no item anterior. Utilize o Matlab para sintetizar os sinais internos dos circuitos de condicionamento analógico, incluindo o efeito dos circuitos grampeadores de proteção, até os estágios antes dos filtros analógicos.

Exemplos de cálculos:

- A) Obtenha um gráfico das tensões na saída do grampeador, para as entradas de tensão de fase A, B, C. Ex.:

```
VANinterno = VANS * (R2/(R1+R2)) * (12/115);
VBNinterno = VBNS * (R4/(R3+R4)) * (12/115);
VCNinterno = VCNS * (R6/(R5+R6)) * (12/115);
plot(t, VANinterno, t, VBNinterno, t, VCNinterno);
```

- B) Obtenha um gráfico das 3 tensões nas saídas dos circuitos limitadores dos canais de tensão (VANgr, VBNgr e VCNgr). Considere a limitação do circuito grampeador entre os valores de 0,0 e 3,3 [V].

```
VANgr = VANinterno + 1.65;
VANgr = max(0, VANgr);
VANgr = min(3.3, VANgr);
```

- C) Obtenha um gráfico das tensões produzidas pelas correntes de linha A, B, C, nas respectivas entradas analógicas de corrente, na saída dos resistores de *shunt* dos TCs. Ex.:

```
IAinterno = IAs * (1/1000);
VIAinterno = R7 * IAinterno;
```

- D) Obtenha um gráfico das 3 tensões nas saídas dos circuitos limitadores dos canais de corrente de linha A, B e C (VIAgr, VIBgr, VICgr). Considere a limitação entre os valores de 0,0 e 3,3 [V].

```
VIAgr = (VIAinterno + 1.65);
VIAgr = max(0, VIAgr);
VIAgr = min(3.3, VIAgr);
```

- E) De forma semelhante. obtenha um gráfico da tensão produzida pela corrente na entrada analógica de corrente de neutro, visto na saída de seu resistor de *shunt* R10.
- F) Obtenha um gráfico da tensão na saída do circuito limitador de corrente de neutro (VINgr). Considere o deslocamento de 1,65 [V] e a limitação entre os valores de 0,0 e 3,3 [V].
- G) Verifique nos seus sinais se há atuação do circuito limitador, com saturação analógica dos sinais de entrada.

Atividade G – Projeto e simulação de filtros analógicos e digitalização

2.6.5 Analisando o espectro de frequências nos sinais

Para de verificar quantitativamente o espectro dos sinais analógicos medidos de tensões e correntes pode-se fazer uso de rotinas de transformada discreta de Fourier do Matlab, tal como a função 'fft'.

Sabe-se que os sinais produzidos nas simulações 'Vrele', 'Irele', etc. foram coletados com uma quantidade de amostras por ciclo 'N', com um número de 1280 amostras por ciclo de 60,0 [Hz]. A simulação foi feita com uma duração de 10 ciclos. Para obter o espectro de Fourier de um ciclo completo de um dos sinais, deve-se digitar no *prompt* da janela de comandos do MatLab:

```
ini=1;  
fft_VANgr = fft(VANgr(ini:ini+1280), 1280);
```

O índice utilizado para VANgr na chamada à função 'fft' indica que queremos executar uma transformada discreta de Fourier do sinal VANgr, para as amostras desde 'ini' até 'ini+1280', onde 1280 corresponde ao tamanho da janela 'N' necessária para representar o sinal na fundamental de 60,0 [Hz]. Dessa forma, é retornado em 'fft_VANgr' um vetor com números complexos (parte real e imaginária) de cada componente harmônica do sinal, desde a componente DC até a maior harmônica possível de ser obtida com essa janela. Se por exemplo, além da componente DC, é interessante conhecer as primeiras 10 a 15 harmônicas, faça:

```
fft_VANgr(1:16)
```

A posição 1 do vetor retorna a componente DC (F_0), a posição 2 a componente da primeira harmônica (fundamental) do sinal (F_1), a posição 3 a segunda harmônica e assim por diante.

Para plotar a magnitude e fase das 15 primeiras harmônicas, incluindo o nível DC (harmônica de ordem '0'), pode-se fazer:

```
h = 0:15; %contador da ordem da harmonica  
subplot(211); stem(h, abs(fft_VANgr(1:16))); grid;  
title('Módulo das harmônicas na tensão na fase A');  
ylabel('magnitude (pico) [V]');  
subplot(212); stem(h, phase(fft_VANgr(1:16)) * 180/pi ); grid;  
title('Módulo das harmônicas na tensão na fase A');  
ylabel('Fase [graus]');  
xlabel('Ordem das harmonicas');
```

É interessante observar que o comando FFT retorna números complexos cujos módulos não estão normalizados com a energia do sinal. Deve-se dividir o resultado de todas as componentes obtidas por 'N/2' para que o módulo de cada harmônica seja condizente com o valor de pico do sinal original. Deve-se atentar ainda que a magnitude de cada componente harmônica será dada então em valor de pico e não eficaz. Para melhorar a representação, é costumeiro serem utilizados valores por unidade, bastando utilizar como tensão ou corrente de base os valores de pico adequados.

Atividade G – Projeto e simulação de filtros analógicos e digitalização

Assim, o gráfico anterior poderia estar normalizado com as seguintes operações com o vetor 'fft_VANgr':

```
fft_VANgr = fft_VANgr / (1280/2); %normaliza para valores de pico
fft_VANgr = fft_VANgr / ((13800/sqrt(3))*sqrt(2)); %normaliza para
p.u. usando uma tensão de fase, de pico
```

2.6.6 Avaliação do espectro de sinais AC

É interessante observar o espectro das tensões VANgr, VBNgr, VCNgr, VIgr, VIBgr e VICgr quanto às **magnitudes** de suas componentes espectrais. Podem-se apresentar os resultados na forma de dois gráficos, um para o espectro das tensões e outro para o espectro das correntes. Os sinais podem ser plotados usando múltiplos comandos 'stem', exibindo até a 15ª harmônica, utilizando a função 'hold on' para imprimir três fases no mesmo plano cartesiano.

Escolha um ponto inicial 'ini' distante do início da simulação, para realizar a análise espectral em algum instante quando o sistema dinâmico já está em regime permanente. Garanta que o sinal sob análise não está saturado pela atuação do circuito de grampeamento. Se estiver grampeado, o sinal deverá ter componentes harmônicas elevadas devido às distorções e não linearidades do grampeador. Deseja-se verificar apenas as harmônicas naturais dos sinais do sistema de potência simulado.

Verifique e identifique quais harmônicas estão presentes nos sinais de tensão e de corrente. Para as tensões:

```
figure(1);
stem(h, abs(fft_VANgr(1:16)), 'r'); grid; hold on;
stem(h, abs(fft_VBNgr(1:16)), 'g'); grid;
stem(h, abs(fft_VCNgr(1:16)), 'b'); grid;
legend('VANgr', 'VBNgr', 'VCNgr');
title('Módulo das harmônicas na tensão na fase A');
ylabel('magnitude (pico) [V]');
xlabel('Ordem das harmonicas');
```

Como o conversor AD considerado nessa atividade é de 12 bits, pode-se analisar qual o menor sinal que pode ser representado em sua saída e seu correspondente valor em cada entrada analógica. Componentes harmônicas abaixo desse valor não serão percebidos pelo conversor AD e não são relevantes. Componentes harmônicas acima desse valor, ao recobrirem no espectro outras componentes, podem alterar a espectro do sinal digitalizado. Uma forma aproximada de se estabelecer as componentes a serem filtradas é verificar, daquelas que serão recobertas no espectro, se seu valor é superior a $\frac{1}{2} \cdot \frac{1}{2^{12}-1}$ [p.u.].

2.6.7 Determinação do recobrimento dos sinais AC

Considerando, por exemplo, que os sinais VANgr, VBNgr, VCNgr e IAggr, IBgr, ICgr serão amostrados no conversor AD com 8 amostras por ciclo de 60,0 [Hz], o espectro originalmente presente nos sinais irá se repetir na frequência de amostragem escolhida (8 amostras por ciclo = oitava harmônica, $f = 8 \cdot 60 = 480$ Hz), recobrando algumas de suas componentes.

Atividade G – Projeto e simulação de filtros analógicos e digitalização

Verifique qual a ordem da componente harmônica que se recobre sobre a frequência fundamental de 60 Hz, por exemplo. Determine qual componente se recobre sobre a 3ª. Harmônica, 5ª e 7ª harmônica.

2.6.8 Projeto de um filtro anti-aliasing para os sinais AC

Especifique e projete um filtro passa-baixa que será aplicado aos sinais de tensão e corrente. Para a especificação, por exemplo, coloque o final da faixa de passagem entre a frequência fundamental de 60,0 [Hz] e a segunda harmônica, por exemplo. Admita como atenuação máxima o valor de -3,0 [dB] na frequência de corte, extremo na banda de passagem. Pode-se fazer o início da banda rejeição com frequência próxima a da 4ª harmônica. Admita como atenuação mínima na banda de rejeição, por exemplo, $20 \cdot \log\left(\frac{1}{2^{12}-1}\right) = -72,0$ [dB].

Projeto um filtro padrão Butterworth com a especificação designada, determinando sua ordem e amortecimento, além de sua expressão completa no domínio da frequência complexa de Laplace.

Para síntese automática do filtro de Butterworth no Matlab pode se utilizar o comando:

```
[num,den] = butter(<ordem>, <freq.decorte>, 'low', 's')
```

Onde <ordem> e <freq.decorte> são respectivamente a ordem calculada para o filtro e sua frequência de corte em [rad/s].

Para visualização da função de transferência obtida, crie um sistema dinâmico em 's' no Matlab com o comando:

```
fpb = tf(num,den) %sem ponto-e-virgula para o resultado vir na tela
```

Note que o filtro gerado pelo Matlab possui coeficientes de grande amplitude.

2.6.9 Estudo do filtro passa baixa

Usando o Matlab, obtenha a resposta em frequência do filtro passa-baixa com um diagrama de bode, com:

```
bode(fpb);
```

Verifique a resposta do filtro para um degrau com o comando 'step'. Obtenha o tempo de subida e estabilização do filtro.

Determine qual o ganho em dB e o atraso de fase em graus para um sinal na frequência fundamental de 60,0 [Hz] para filtro criado. Isso pode ser feito através do comando:

```
ganho= abs(freqresp(fpb, 2*pi*60))*180/pi;  
atraso = phase(freqresp(fpb, 2*pi*60))*180/pi;
```

2.6.10 Filtragem analógica dos sinais de tensão e corrente

Considerando o vetor de tempo auxiliar para as amostras dos vetores VANgr, VBNgr, VCNgr, IAgr, IBgr e ICgr, simule a filtragem analógica de cada um dos sinais originais com o filtro passa-baixa sintetizado com o comando:

Atividade G – Projeto e simulação de filtros analógicos e digitalização

```
VANfiltrado = lsim(fpb, VANgr, t);
```

Plote o vetor resultante do comando junto do sinal original. Verifique o atraso criado pelo filtro e a diminuição do conteúdo harmônico. Faça isso para todos os sinais analógicos de seu dispositivo.

2.6.11 Curiosidade: Comparação do conteúdo espectral após filtragem

Analise quantitativamente o conteúdo espectral, tanto de um sinal de tensão como de corrente, antes e depois da filtragem analógica. Note se alguma componente harmônica é reduzida.

2.6.12 Simulação da digitalização dos sinais no conversor

Até o momento foram constituídos vetores das tensões nas saídas dos filtros analógicos passa-baixa: VANfiltrado, VBNfiltrado, VCNfiltrado, VIAfiltrado, VIBfiltrado, VICfiltrado, VINfiltrado. Todos esses sinais estão superamostrados, com uma taxa elevada de $1280 \cdot 60 = 76,8$ [kHz], e possuem valores analógicos de tensão entre 0,0 a 3,3 [V].

Como deseja-se obter esses sinais, com taxa de amostragem de 16 amostras por ciclo de 60,0 [Hz], e pelo fato do sinal original e do sinal desejado possuírem frequências de amostragem múltiplas, para emular o processo de amostragem pode-se utilizar o método de “decimação simples” ou, com nome mais apropriado em inglês, *downsampling*.

A decimação simples consiste em varrer cada vetor de dados, coletar uma amostra a cada ‘n’, e armazenar em outro vetor de saída. Por exemplo, se coletarmos uma amostra a cada 80 dos vetores de 1280 amostras por ciclo estaremos obtendo 16 amostras por ciclo.

É importante notar que esse processo de decimação é, a rigor, impreciso, pois pode estar havendo algum recobrimento dos espectros de frequência ao se descartar as demais amostras (*aliasing*). Um outro método mais preciso é a aplicação de um processo de reamostragem do sinal com uso de filtros digitais.

Para emular o processo de amostragem, pode-se fazer:

```
k=1;
for j = 1:size(VANadc)
    if mod(j-1,80)==0
        VAN_sh(k) = VANfiltrado(j);
        VBN_sh(k) = VBNfiltrado(j);
        VCN_sh(k) = VCNfiltrado(j);
        VIA_sh(k) = VIAfiltrado(j);
        VIB_sh(k) = VIBfiltrado(j);
        VIC_sh(k) = VICfiltrado(j);
        VIN_sh(k) = VINfiltrado(j);
        k=k+1;
    end
end
```

Atividade G – Projeto e simulação de filtros analógicos e digitalização

O processo anterior imita o comportamento dos *sample hold* presentes nos conversores ADs. Tem-se sinais amostrados, com uma taxa mais baixa, mas ainda representando tensões analógicas na faixa entre 0,0 e 3,3 [V].

Para simular a digitalização do conversor AD, deve-se considerar o intervalo de quantização ou *quantum* do conversor AD, como sendo a relação entre sua faixa dinâmica de entrada e de saída, ou seja, $q = \frac{V_{FS}}{2^n - 1}$, sendo V_{FS} a tensão de fundo de escala de sua entrada e 'n' a resolução em bits do conversor AD. Adotando $V_{FS} = 3,3$ [V] e $n=12$ bits, ao dividir o sinal amostrado por 'q', e arredondando o resultado para um número inteiro, tem-se o valor desejado do sinal digitalizado. Deve-se atentar que por ocasião de truncamentos, deve-se checar se o valor está contido no intervalo de valores de saída permitidos para o conversor. Considerando um conversor unipolar, entre 0 e $2^n - 1$.

```
q = 3.3 / 4095;
Ch1_VAN = min(0, max(round(VAN_sh/q), 4095));
Ch2_VBN = min(0, max(round(VBN_sh/q), 4095));
...
```

Exemplos de atividades:

- A) Obtenha os vetores dos sinais digitalizados para todos os canais de seu dispositivo, no caso de exemplo, para Ch1_VAN, Ch2_VBN, Ch3_VCN, Ch4_IA, Ch5_IB, Ch6_IC e Ch7_IN. Plote o gráfico de apenas um canal de tensão e um canal de corrente de uma mesma fase, no mesmo plano x-y. Escolha uma tensão e corrente pertencentes a uma mesma fase que está envolvida por exemplo num transitório de curto-circuito. Utilize um vetor auxiliar 'n' como índice ou base de tempo para numerar as amostras. Use a função `stairs` (ou `stem` por curiosidade) ao invés da `plot` para exibir os resultados de forma diferente. Ex.:

```
n=1:1:size(Ch1_VAN,2);
stairs(n, Ch1_VAN, 'b'); hold on; grid;
stairs(n, Ch4_IA, 'r'); hold off;
```

- B) Escolha um dos sinais de corrente que apresente menores distorções harmônicas e oscilações de alta frequência durante um período de falta. Obtenha uma figura com dois gráficos sobrepostos, contendo na parte superior o sinal original e na parte inferior o sinal amostrado e digitalizado. Ex.:

```
close all; %fecha qualquer figura aberta
figure(1);
subplot(211);
plot(Irele(:,2)); grid;
subplot(212);
stairs(Ch5_IB); grid;
```

Atividade G – Projeto e simulação de filtros analógicos e digitalização

- C) Para cada canal, obtenha a expressão matemática que converte um valor decimal amostrado pelo conversor AD em seu valor de real, nas unidades de engenharia ([V] ou [A]), referidas ao primário dos TP's ou TC's conectados a rede elétrica. Use o formato:

$$VAN(x) = A \cdot x + B$$

Onde x é o valor amostrado pelo AD e $VAN(x)$ seu valor em Volts, no primário do TP ligado à rede elétrica.

- D) Para cada canal, qual seria a expressão que converte as amostras do conversor AD, de números inteiros na faixa de 0 a 4095, para valores em ponto flutuante em [p.u.]. Use o formato:

$$VAN_{pu}(x) = A \cdot x + B$$

Onde x é o valor amostrado pelo AD e $VAN_{pu}(x)$ seu valor em [p.u.].

- E) Verifique que a incerteza ou erro percentual do processo de digitalização para cada canal é $q/2$.

2.6.13 Para as próximas atividades

Uma vez digitalizados e possuindo equações que permitam converter os valores digitais para seus equivalentes em unidades de engenharia, pense nos algoritmos de processamento necessários para extração das informações pertinentes para cada tipo de aplicação.

Esses algoritmos são baseados em filtros e expressões algébricas, no domínio do tempo discreto, que permitem extrair dos sinais, individualmente, várias grandezas. Por exemplo:

- Valor eficaz
- Valor de pico
- Amplitude pico a pico
- Frequência e período da fundamental
- *Duty-cycle*
- Valor médio ou nível DC
- Fasores de cada componente harmônica do sinal (módulo e fase)
- Distorção harmônica total do sinal

Através de análise conjunta desses sinais, podem ser obtidas outras grandezas de interesse:

- Potência instantânea
- Potência ativa média, potência aparente com base em valores eficazes, potência reativa e fator de potência considerando fatores de distorção harmônica
- Potência ativa, reativa e aparente e fator de potência da componente fundamental
- Energia
- Defasagens entre sinais
- Componentes simétricas de tensões e corrente
- Impedâncias e admitâncias vistas por fase
- Componentes vetoriais e outras grandezas para observabilidade e controle de dispositivos