



MÉTODO DE ESTIMAÇÃO DE PARÂMETROS PARA CONVERSORES VSC CONECTADOS À REDE

Neimar Machado da Silva

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia Elétrica, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia Elétrica.

Orientador: Luis Guilherme Barbosa Rolim

Rio de Janeiro
Fevereiro de 2019

MÉTODO DE ESTIMAÇÃO DE PARÂMETROS PARA CONVERSORES VSC
CONECTADOS À REDE

Neimar Machado da Silva

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO
ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE
ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE
JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A
OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA
ELÉTRICA.

Examinada por:

Prof. Luis Guilherme Barbosa Rolim, Dr.Ing

Prof. Walter Issamu Suemitsu, Dr.Ing.

Prof. João Onofre Pereira Pinto, Ph.D.

RIO DE JANEIRO, RJ – BRASIL
FEVEREIRO DE 2019

Machado da Silva, Neimar

Método de Estimação de Parâmetros para Conversores VSC conectados à Rede/Neimar Machado da Silva. – Rio de Janeiro: UFRJ/COPPE, 2019.

XVIII, 75 p.: il.; 29, 7cm.

Orientador: Luis Guilherme Barbosa Rolim

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia Elétrica, 2019.

Referências Bibliográficas: p. 63 – 66.

1. Método de Estimação de Parâmetros. 2. VSC. 3. *Hardware-in-the-loop*. 4. Sistema Digital. 5. CHIRP. I. Rolim, Luis Guilherme Barbosa. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Elétrica. III. Título.

*Lutar sempre, vencer às vezes,
desistir jamais!*

Agradecimentos

Agradeço a Deus pela oportunidade de viver.

À minha mãe por me fornecer os meios necessários à minha formação.

Aos professores, os grandes proporcionadores do conhecimento. Especial agradecimento ao professor Luis Guilherme Barbosa Rolim pela orientação deste trabalho.

Aos amigos da COPPE que muito me ensinaram e que se dispuseram em ajudar em todos os momentos.

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelo apoio financeiro no primeiro ano de mestrado.

Aos irmãos em Cristo pelo inestimável apoio nas orações e conselhos.

À minha esposa por todo amor e carinho. A ela sou um eterno devedor de toda compreensão e companheirismo.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento Pessoal de Nível Superior do Brasil (CAPES).

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

MÉTODO DE ESTIMAÇÃO DE PARÂMETROS PARA CONVERSORES VSC CONECTADOS À REDE

Neimar Machado da Silva

Fevereiro/2019

Orientador: Luis Guilherme Barbosa Rolim

Programa: Engenharia Elétrica

Para que os conversores conectados a redes funcionem de forma estável e otimizada suas malhas de controle de corrente e de tensão devem sofrer ajustes específicos para aquele tipo de rede em que forem operar. Estes ajustes estão diretamente relacionados com a impedância da rede na qual se insere o conversor. Dessa forma, é desejável que o próprio conversor identifique a impedância da rede vista por ele e que possa ajustar os ganhos de controle sem que haja a necessidade de ajustes adicionais por parte do projetista. É proposto, então, um algoritmo capaz de calcular os parâmetros de impedância da rede através da injeção de um sinal de referência do tipo CHIRP (*Compressed High Intensity Radar Pulse*) na rede por um conversor alimentado em tensão (VSC, do inglês *Voltage Sourced Converter*) com saída controlada por corrente. Com estes parâmetros, o conversor pode dinamicamente ajustar os ganhos das malhas de controle.

Portanto, a principal contribuição desta dissertação consiste em avaliar um método que envolve a utilização de um algoritmo recursivo capaz de calcular os parâmetros de impedância da rede sem a necessidade de desconexão do conversor VSC da rede. Este algoritmo define a impedância da rede de forma rápida, consumindo pouco espaço de armazenamento e pouco tempo durante seu processamento.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

PARAMETER ESTIMATION METHOD FOR GRID-CONNECTED VSC CONVERTERS

Neimar Machado da Silva

February/2019

Advisor: Luis Guilherme Barbosa Rolim

Department: Electrical Engineering

Grid-connected converters work in a stable and optimized way if their voltage and current control loops are tuned specifically to that grid for which they were designed. This tuning is directly related to the grid impedance at the point the converter is connected to. So, it is desirable that the converter itself be capable to identify the grid impedance and be able to tune its control loop gains without further operator adjustments. Thinking on that, it is proposed an algorithm capable to calculate the parameters of grid impedance injecting a reference signal in CHIRP shape to the grid through current-controlled voltage sourced converted (VSC). With these parameters on hand, the converter can dynamically tune its control loop gains.

So, the main contribution of this dissertation consists on evaluating a method that involves the use of a recursive algorithm that is capable to calculate the grid impedance parameters without disconnecting the VSC converter from the grid. This algorithm defines the grid impedance in a fast way while consuming few memory and processing time.

Sumário

Lista de Figuras	x
Lista de Tabelas	xiii
Lista de Símbolos	xiv
Lista de Abreviaturas	xvii
1 Introdução	1
1.1 Motivação	2
1.2 Revisão Bibliográfica	3
1.3 Objetivo	4
1.4 Estrutura do Documento	4
2 Método de Medição de Impedância Recursivo	6
2.1 Diagrama em Blocos do Método	6
2.2 Sinal Sintetizado pelo Conversor	7
2.3 Análise do Sistema Digital	11
2.4 Influência da Rede nas Medições	15
2.5 Algoritmo Recursivo	19
3 Detalhamento da Simulação	25
3.1 Ambiente de Simulação	25
3.2 Características do Conversor	26
3.3 Parâmetros da Simulação	27
4 Resultados e Discussões	34
4.1 Rede de 1ª ordem	34
4.2 Rede de 1ª ordem com harmônicos na tensão	43
4.3 Rede de 2ª ordem	46
4.4 Rede de 4ª ordem	51
4.5 Rede Benchmark	55
4.6 Análise do Tempo de Processamento	58

5	Conclusões e Trabalhos Futuros	61
	Referências Bibliográficas	63
A	Phase-Locked-Loop	67
B	Código do Algoritmo Recursivo	69

Lista de Figuras

2.1	Diagrama de blocos do método de estimação de impedância da rede.	7
2.2	Circuito básico utilizado nas simulações contendo um conversor VSC conectado a uma rede no seu equivalente de Thévenin.	7
2.3	Gráfico de corrente sintetizada pelo método de comparação direta.	8
2.4	Diagrama simplificado do conversor VSC conectado à rede no seu equivalente de Thévenin.	8
2.5	Circuito caracterizando a injeção de um pulso CHIRP na rede através do VSC.	9
2.6	Sinal CHIRP com frequência variando de $10Hz$ a $100Hz$	10
2.7	Resposta em frequência do sinal CHIRP com $f_0 = 10Hz$ e $f_1 = 100Hz$	10
2.8	Sinal CHIRP dentro de uma janela Tukey com α igual a 50%.	11
2.9	Espectro do sinal CHIRP dentro de uma janela Tukey.	12
2.10	Circuito simplificado considerando o conversor VSC como uma fonte de corrente.	12
2.11	Diagrama em blocos do sistema digital considerando a rede como uma perturbação no sistema.	15
2.12	Diagrama de Blocos completo do Método Recursivo de Medição de Impedância da Rede testado neste trabalho.	24
3.1	Caixa DLL de interface com o MCU para uma rede de 2^a ordem.	26
3.2	(a) Tensão trifásica filtrada na saída do conversor e algoritmo de PLL em funcionamento; (b) Corrente de referência em formato CHIRP sendo injetada na rede.	29
3.3	Análise de corrente e tensão no elo CC em comparativo com a corrente de saída em uma fases do conversor.	30
3.4	Correntes e Tensões trifásicas na saída do VSC.	31
3.5	Comparativo entre o sinal de referência CHIRP e a corrente injetada pelo VSC medida em uma das fases e após ser filtrada.	32
4.1	Rede com impedância modelada de 1^a ordem.	35

4.2	Gráfico das curvas da variação de tensão no elo CC, e das correntes e tensões numa rede de 1ª ordem.	36
4.3	Gráfico das respostas em frequência da admitância da rede de 1ª ordem obtidas pelo método dos mínimos quadrados e pelo método recursivo.	40
4.4	Gráfico das curvas real e calculada pelo método com impedância da rede de 1ª ordem para diferentes valores de n.	41
4.5	Gráfico das curvas real e calculada pelo método com impedância da rede de 1ª ordem para diferentes valores de n ampliado entre as frequências de 1º e 10º harmônicos.	42
4.6	Gráfico mostrando o erro normalizado entre os coeficientes reais e aqueles obtidos a cada iteração.	43
4.7	Gráfico das curvas de variação de tensão no elo CC e da corrente e tensão da rede contendo harmônicos de tensão com THD em 11,6%.	44
4.8	Resposta do método considerando uma impedância <i>RL</i> e rede contendo harmônicos de tensão.	45
4.9	Impedância da Rede modelada de 2ª Ordem.	46
4.10	Gráfico das curvas da variação de tensão no elo CC, e das correntes e tensões numa rede de 2ª ordem.	47
4.11	Gráfico das respostas em frequência da impedância de uma rede de 2ª ordem obtidas pelo método dos mínimos quadrados, pelo método recursivo e a gerada pelo circuito real.	50
4.12	Gráfico das curvas real e calculada pelo método com impedância da rede de 2ª ordem para diferentes valores de n.	51
4.13	Impedância de 4ª ordem.	52
4.14	Gráfico das curvas da variação de tensão no elo CC, e das correntes e tensões numa rede de 4ª ordem.	53
4.15	Gráfico das respostas em frequência da impedância de uma rede de 4ª ordem obtidas pelo método dos mínimos quadrados, pelo método recursivo e a gerada pelo circuito real.	54
4.16	Gráfico das curvas real e calculada pelo método com impedância da rede de 4ª ordem para diferentes valores de ordem n.	55
4.17	Gráfico das curvas da variação de tensão no elo CC, e das correntes e tensões na rede <i>benchmark</i> 1.	56
4.18	Resposta do método recursivo sobre a rede <i>benchmark</i> 1 considerando diferentes valores de n.	57
4.19	Resposta em frequência da rede <i>benchmark</i> 1 utilizando o método dos mínimos quadrados aplicando diferentes valores de n.	58

4.20	Número de clocks por iteração necessários para atualizar as matrizes Γ e Ξ em função da ordem.	60
A.1	Diagrama em Blocos do PLL.	68

Lista de Tabelas

2.1	Vantagens e Desvantagens do MMQ e do Método Recursivo.	23
3.1	Parâmetros do conversor.	27
3.2	Parâmetros das simulações.	32
4.1	Comparativo entre os coeficientes obtidos pelo algoritmo recursivo com os coeficientes teóricos.	39
4.2	Comparativo entre os coeficientes obtidos pelo algoritmo recursivo com os coeficientes teóricos numa rede de 1 ^a ordem com harmônicos de tensão.	45
4.3	Comparativo entre os coeficientes obtidos pelo algoritmo recursivo com os coeficientes teóricos.	49
4.4	Tempo gasto pelo MCU na execução das operações intra-iterações em função da ordem do sistema de equações.	60

Lista de Símbolos

α	Razão entre a faixa cossenoidal e a faixa total da janela Tukey.
γ	Harmônico genérico presente na rede.
Γ	Vetor contendo elementos da linha da matriz A no instante k .
Θ	Vetor contendo os coeficientes da rede a serem estimados pelo conversor.
Θ_{exp}	Vetor Θ expandido contendo todos os coeficientes e amplitudes de cada harmônico da rede.
θ_k^0	Fase referente à componente fundamental da rede no instante de amostragem k .
θ_k^γ	Fase referente ao harmônico γ no instante de amostragem k .
ξ	Matriz quadrada gerada pelo produto dos vetores Γ transposto com o vetor Γ no instante de amostragem k .
Ξ	Matriz quadrada gerada pelo produto da matriz A transposta com a matriz A .
ϕ	Vetor coluna obtido pelo produto escalar entre o vetor coluna Γ e o valor amostrado de saída y_k no instante de amostragem k .
Φ	Vetor coluna obtido pelo produto da matriz A transposta pelo vetor coluna Y .
$\rho(t)$	Amplitude do sinal Tukey.
ω_γ	Frequência angular do harmônico γ .

a_i e b_i	Coefficientes da equação característica da impedância da rede.
A	Matriz contendo as medidas passadas de tensão e corrente.
A_{exp}	Matriz A expandida contendo cossenos e senos das fases de cada harmônico presente na rede.
f_a	Frequência de Amostragem.
f_{chav}	Frequência de Chaveamento do Conversor.
f_0	Frequência inicial do pulso CHIRP.
f_1	Frequência final do pulso CHIRP.
f_c	Frequência de corte do filtro.
h	Período de amostragem.
i_k	Corrente instantânea medida na iteração k .
I_C^*	Corrente no formato CHIRP sintetizada pelo VSC.
I_{base}	Corrente base.
I_{ref}	Amplitude da corrente no formato CHIRP aplicada.
k	Instante de amostragem.
L	Reatância equivalente da rede vista pelo VSC.
L_c	Indutância de saída do VSC.
M	Número de harmônicos presentes na rede.
m	Número de zeros da equação característica da impedância da rede.
N	Número Total de Amostras.
n	Ordem da equação característica da impedância da rede.
P_{conv}	Potência do Conversor.
R	Resistência equivalente da rede vista pelo VSC.
t_0	Instante inicial de aplicação do sinal CHIRP.
t_i	Instante final de aplicação do sinal CHIRP.

T	Tempo total simulado. Duração total de funcionamento do algoritmo recursivo.
v_k	Tensão instantânea medida na iteração k .
$V_{C\gamma}$	Amplitude da componente cossenoidal do harmônico γ .
$V_{S\gamma}$	Amplitude da componente senoidal do harmônico γ .
V_{PCC}	Tensão no Ponto de Conexão Comum.
V_R	Tensão trifásica da rede.
V_{base}	Tensão de linha base.
V_{CCbase}	Tensão base no elo CC.
V_{CC}	Tensão inicial no elo CC.
$x(t)$	Amplitude do sinal CHIRP.
Y	Vetor contendo as medidas de tensão atuais.
y_k	Amostra de saída na iteração k .
Z_c	Impedância do reator de saída do conversor.
Z_R	Impedância equivalente da rede vista pelo conversor VSC.
$Z(z)$	Transformada Z da impedância da rede vista pelo conversor VSC calculada pelo Algoritmo Recursivo.
Z_{base}	Impedância base.

Lista de Abreviaturas

ADC	Conversor Analógico-Digital
ANEEL	Agência Nacional de Energia Elétrica
CHIRP	<i>Compressed High Intensity Radar Pulse</i>
CPU	Unidade Central de Processamento
DFT	<i>Discrete Fourier Transform</i>
DLL	<i>Dynamic Link Library</i>
EMC	Compatibilidade Eletromagnética
FFT	<i>Fast Fourier Transform</i>
FT	Função de Transferência
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IGBT	Transistores Bipolares de Porta Isolada
MCU	Unidade Microcontroladora
MMQ	Método dos Mínimos Quadrados
PC	Computador Pessoal
PCC	Ponto de Conexão Comum
PI	Controlador Proporcional-Integral
PLL	<i>Phase-Locked Loop</i>

PRODIST	Procedimentos de Distribuição de Energia Elétrica no Sistema Elétrico Nacional
PRBS	<i>Pseudo Random Binary Sequence</i>
PWM	<i>Pulse Width Modulation</i>
SCI	<i>Serial Communication Interface</i>
SVM	<i>Space Vector Modulation</i>
THD	<i>Total Harmonic Distortion</i>
UART	<i>Universal Asynchronous Receiver-transmitter</i>
VSC	<i>Voltage-Sourced Converter</i>
ZOH	<i>Zero-Order Hold</i>

Capítulo 1

Introdução

Os incentivos por geração distribuída de energia elétrica fornecidos pelos governos de todo o mundo vêm aumentando devido à crescente demanda por energia aliada à pressão da sociedade e de diversos órgãos não governamentais por instalações de fontes renováveis de energia. Ainda hoje, as hidrelétricas assumem papel fundamental como fonte de energia renovável, porém, a sua baixa escalabilidade eleva a importância de outras fontes alternativas de energia como a eólica, solar e dos mares como destacado em [1] que mostra vários empreendimentos em energias renováveis em construção atualmente no país. A instalação destas fontes alternativas de energia de forma distribuída tem por consequência natural a inserção de novos conversores à rede. Estes conversores de características não lineares são fontes de harmônicos na rede o que faz decair a qualidade da energia fornecida. Se por um lado diminui-se a pressão por mais oferta, por outro, aumenta-se consideravelmente a poluição harmônica da rede decaindo a qualidade do funcionamento destes conversores.

Diversos estudos vêm sendo conduzidos visando à análise de desempenho de conversores conectados a redes que contenham alto percentual de harmônicos e sua influência nos conversores tanto em termos de corrente quanto em termos de tensão como destacado no artigo [2]. Por outro lado, o padrão IEEE 519-1992 [3] também orienta a THD máxima que os conversores podem injetar no PCC e a ANEEL, com o intuito de também preservar a rede, estabelece os Procedimentos de Distribuição de Energia Elétrica no Sistema Elétrico Nacional (PRODIST) determinando critérios para preservação da rede e assegurar a qualidade da energia. Estudos também vem sendo direcionados na análise dos conversores e em seus controles quando inseridos em redes cujas impedâncias se modificam utilizando, para isso, uma infinidade de tipos de controle extensamente estudados na literatura, como controles adaptativos. Toda esta análise visa cumprir, dentre outros requisitos, manter os conversores dentro dos seus limites de estabilidade. Um dos critérios mais utilizados de estabilidade entre conversores foi inicialmente estudado em [4]. O autor separa o sistema em duas partes, o conversor e a rede, e prova que o sistema é estável se o contorno de

Nyquist gerado pelo produto da impedância da rede, ou seja, a impedância vista pelo conversor, e a admitância de entrada do conversor permanece dentro do círculo unitário. Diversos outros critérios de estabilidade foram desenvolvidos baseados na impedância vista pelo conversor e pela rede utilizando o critério de Nyquist de estabilidade como demonstrado em [5], [6], [7] e [8].

Especificamente, a estabilidade de um conversor tipo fonte de tensão (VSC) conectado à rede depende da razão entre a impedância da rede e a impedância do conversor conforme demonstrado em [9]. Em [10] e [11] foi provado que altas impedâncias da rede podem levar o sistema conversor-rede à instabilidade. Assim, para garantir que os controles do conversor possam trabalhar dentro dos limites de estabilidade, estes devem ser parametrizados em função da impedância da rede vista pelo inversor no ponto em que foi conectado. Dessa forma, é de grande valia criar uma forma de monitorar a impedância da rede de tal forma que os parâmetros de controle do conversor possam ser ajustados dinamicamente de acordo com esta variação de impedância.

1.1 Motivação

Este trabalho foi inspirado na dissertação [12], onde foi proposto que o conversor tivesse a capacidade de medir por si próprio a impedância da rede. Foram levantados naquela dissertação os possíveis riscos de um conversor passar a operar na instabilidade devido a mudanças repentinas de impedância da rede. Atualmente, este risco é cada vez mais real devido a novas fontes de energia renovável integradas a redes de distribuição mediante inversores VSC com corrente controlada. Diversos conversores conectados na rede, variando a todo o tempo sua potência gerada entre zero e 100%, podem alterar substancialmente a impedância da rede acarretando nos problemas já levantados.

Um conversor VSC conectado à rede é controlado em forma de corrente, uma vez que a própria rede fornece a referência de tensão trifásica para o conversor. É usual que existam dentro do controlador do conversor algoritmos anti-ilhamento que desconectem o conversor em caso de ausência da rede. Este algoritmo se faz necessário por diversos motivos citados em [13] onde se destaca, por exemplo, proteção da rede e de pessoas e garantia da qualidade da energia fornecida. Uma das formas de implementar essa funcionalidade é por meio de detecção da impedância da rede. Também, dentro do controlador VSC, se faz necessário um algoritmo de controle de corrente injetada na rede. Uma forma amplamente utilizada é o controle por histerese, chamado também por controle de banda de tolerância ou de PWM adaptativo como foi apresentado em [14]. Este tipo de controle de corrente tem como vantagem a sua fácil implementação contudo seu princípio de operação resulta em grandes variações de

frequência de chaveamento causando problemas de compatibilidade eletromagnética (EMC)¹. Outro tipo de controle de corrente muito utilizado é o que emprega laços de realimentação clássicos com controladores lineares (e.g. PI), sendo o sinal de controle aplicado à planta (circuito externo) por meio de modulação PWM senoidal ou vetorial (SVM)². Dessa forma, a frequência da portadora determina a frequência de chaveamento, melhorando a compatibilidade eletromagnética. Se por um lado estes controladores podem ser facilmente projetados aplicando métodos bem conhecidos de controle clássico, por outro lado é necessário conhecer, ou pelo menos, estimar a função de transferência da planta, para projetar controladores que se mantenham estáveis diante de grandes variações de parâmetros da mesma.

Nota-se, portanto, que as características da rede são extremamente necessárias para que o controle de corrente de um conversor VSC conectado a rede seja executado corretamente.

1.2 Revisão Bibliográfica

Existem diversas formas de se realizar o monitoramento de impedância da rede, como por exemplo aquele descrito em [15] que realiza a medição injetando um sinal de banda larga na rede e medindo sua resposta ao impulso. Em [16], a impedância da rede foi medida utilizando a técnica de *Pseudo Random Binary Sequence* (PRBS) onde um sinal aleatório de resposta em frequência de amplitude constante, ou seja, com espectro uniformemente distribuído, é injetado na rede. Outros trabalhos propõem a medição da impedância da rede através da injeção de um pulso *Compressed High Intensity Radar Pulse* (CHIRP). A grande vantagem de se utilizar um pulso CHIRP segundo [17] provém do fato de o tempo de processamento ser bem reduzido já que é um sinal de curto espaço de tempo com largo espectro de frequência. Como destacado em [18] e [19], que tratam sobre o problema de identificação de sistemas, o sinal CHIRP é largamente utilizado como sinal de excitação por conter um conjunto infinito de senóides sendo, assim, utilizado para se estimar plantas de ordens desconhecidas.

O grande desafio enfrentado nos métodos é encontrar a impedância equivalente no PCC com maior aproximação possível, perturbando o mínimo possível a rede e da forma mais rápida e menos custosa. É interessante ressaltar neste ponto que a impedância da rede medida de forma *offline* não retrata fielmente a impedância

¹EMC é a capacidade de um dispositivo eletro-eletrônico de funcionar sem causar perturbações eletromagnéticas a outros dispositivos ou de ser influenciado por eles.

²A modulação espaço vetorial - SVM - é um algoritmo para o controle PWM que visa reduzir o número de chaveamentos do conversor limitando-se a chavear entre apenas dois estados complementares numa frequência de amostragem fixa o que reduz drasticamente as perdas de energia nas chaves e aumenta a EMC do conversor.

vista pela rede e sua influência nos conversores devido a forte influência do conversor sobre a rede dadas as suas características não lineares e mudanças em termos de impedância que ocorrem em tempo real. Dessa forma, dentre os vários métodos existentes, eles podem ser categorizados nos seguintes critérios como definidos em [20] e [21] e [22]: Invasivo ou Não-invasivo, *Online* ou *Offline*, Regime Permanente ou Transitório, Domínio da Frequência ou do Tempo.

O método de medição de impedância testado nesta dissertação será do tipo:

- Invasivo: o conversor irá perturbar forçosamente a rede elétrica;
- *Online*: o conversor permanecerá conectado durante o processo de medição.
- Regime Transitório: o conversor irá gerar perturbações na rede excitando um amplo espectro de frequências para análise no domínio da frequência da rede.
- Domínio do Tempo: utilizando técnicas de sistema digital, o método determinará no domínio do tempo os parâmetros da rede sem a necessidade de se calcular a FFT, DFT ou fasores complexos.

1.3 Objetivo

O objetivo desta dissertação é avaliar um método de medição de impedância da rede realizado por um VSC operando normalmente sem a necessidade de desconectá-lo da rede ou de acréscimo de qualquer outro equipamento externo ao conversor. Toda a dinâmica do método pode ser implementada dentro do microcontrolador do sistema de controle ocupando pouco espaço em memória. Este método de medição de impedância é realizado injetando um pulso tipo CHIRP na rede, porém, diferentemente do que é feito em [17], onde o conversor é desconectado da rede, nesta dissertação, o pulso CHIRP é injetado na rede em forma de corrente sem desconectá-lo dela.

De forma a avaliar se o método em forma de algoritmo recursivo é capaz de calcular a impedância da rede com razoável aproximação, ele foi testado em 5 diferentes tipos de redes: de 1^a, 2^a e 4^a ordens, rede de 1^a ordem com harmônicos, e um modelo de rede *benchmark* desenvolvido em [12] baseada em [23].

A medição da impedância é realizada utilizando técnicas de sistema digital num sistema *hardware-in-the-loop* e utilizando o método dos mínimos quadrados. O algoritmo foi testado em um microcontrolador (MCU) obtendo êxito nos seus resultados.

1.4 Estrutura do Documento

Esta dissertação está organizada da seguinte forma: no capítulo 2, descreve-se a teoria do método. No capítulo 3, os parâmetros das simulações onde o método foi

testado são detalhados. No capítulo 4, mostram-se os resultados alcançados e, por fim, no capítulo 5, são dadas as conclusões e propostos os trabalhos futuros.

Capítulo 2

Método de Medição de Impedância Recursivo

Neste capítulo, descreve-se o circuito teste onde o método de medição de impedância recursivo é aplicado, o conversor VSC utilizado, o tipo de sinal sintetizado pelo conversor que é injetado em formato de corrente na rede, e, por fim, o algoritmo recursivo.

2.1 Diagrama em Blocos do Método

O diagrama em blocos do sistema considerado está descrito na Figura 2.1. O bloco mais à esquerda é o conversor VSC que recebe potência de alguma fonte de energia, por exemplo, solar ou eólica, e a transfere para a rede. O bloco filtro normalmente é do tipo *choke*¹ que serve para filtrar parte dos harmônicos gerados pelo conversor. O bloco mais à direita é a rede que pode ser descrita, de uma forma simplificada, como uma fonte de tensão com uma impedância² em série, ou seja, está sendo considerado o modelo Thévenin da rede. O bloco mais abaixo é o microcontrolador responsável por receber a resposta da rede, proveniente deste estímulo do VSC e por gerar o pulso do tipo CHIRP a ser injetado na rede. É dele também a responsabilidade de calcular a impedância da rede a partir dessa resposta.

O foco desta dissertação é detalhar o algoritmo executado pelo microcontrolador desde a geração do pulso CHIRP até o processamento da resposta gerada pela rede.

¹Indutores utilizados como filtros passa-baixas na saída de cada uma das três fases do conversor responsáveis por filtrar as oscilações de tensão causadas pelos sucessivos chaveamentos do conversor. [24]

²O termo impedância nesta ocasião não se refere a apenas um único elemento do circuito e, sim, à impedância equivalente vista pelo conversor, ou seja, a uma função de transferência $Z(s)$ que representa o efeito combinado das impedâncias existentes na rede.

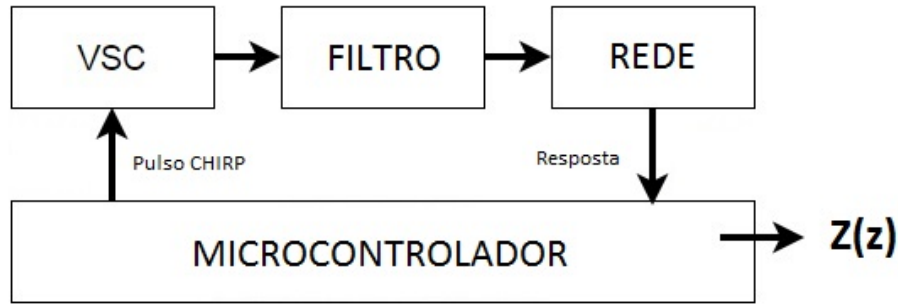


Figura 2.1: Diagrama de blocos do método de estimação de impedância da rede.

2.2 Sinal Sintetizado pelo Conversor

O VSC utilizado é similar ao descrito em [25] e mostrado na Figura 2.2. Este conversor é do tipo fonte de tensão com um elo CC modelado como sendo uma fonte de tensão CC independente e ideal. Ele é construído utilizando IGBTs com diodos em posição anti-paralelo permitindo, assim, o fluxo de corrente reversa da rede. O conversor VSC de dois níveis foi escolhido para testar o algoritmo de medição de impedância devido a sua utilização em larga escala e facilidade de implementação de sistemas de controle de alto desempenho conforme demonstrado em [10]. Ressalta-se, porém, que o método proposto poderia ser aplicado a outros conversores que sejam capazes de sintetizar corrente no formato CHIRP, porém, uma tal análise para diferentes topologias não está no escopo desta dissertação.

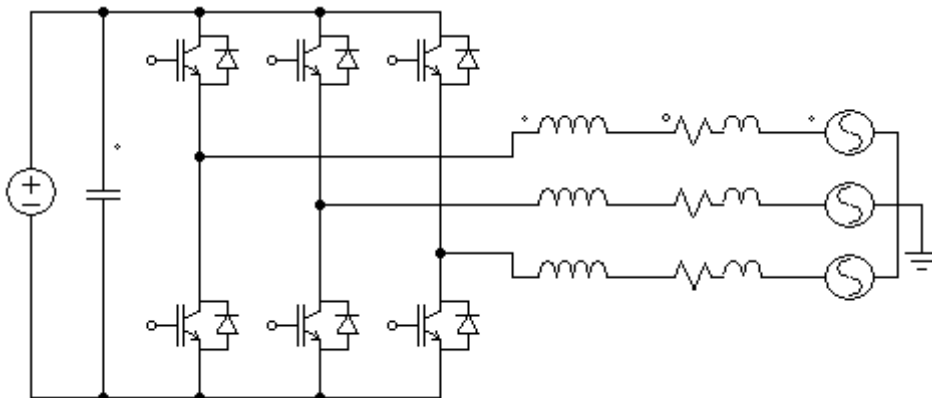


Figura 2.2: Circuito básico utilizado nas simulações contendo um conversor VSC conectado a uma rede no seu equivalente de Thévenin.

Como o método consiste em injetar uma corrente no formato CHIRP na rede, esta corrente deverá ser sintetizada pelo conversor utilizando a técnica de comparação direta entre o sinal amostrado e o formato de corrente desejado. Basicamente, o fechamento e abertura das chaves ocorrerá seguindo-se a corrente de referência como mostrada no gráfico da Figura 2.3

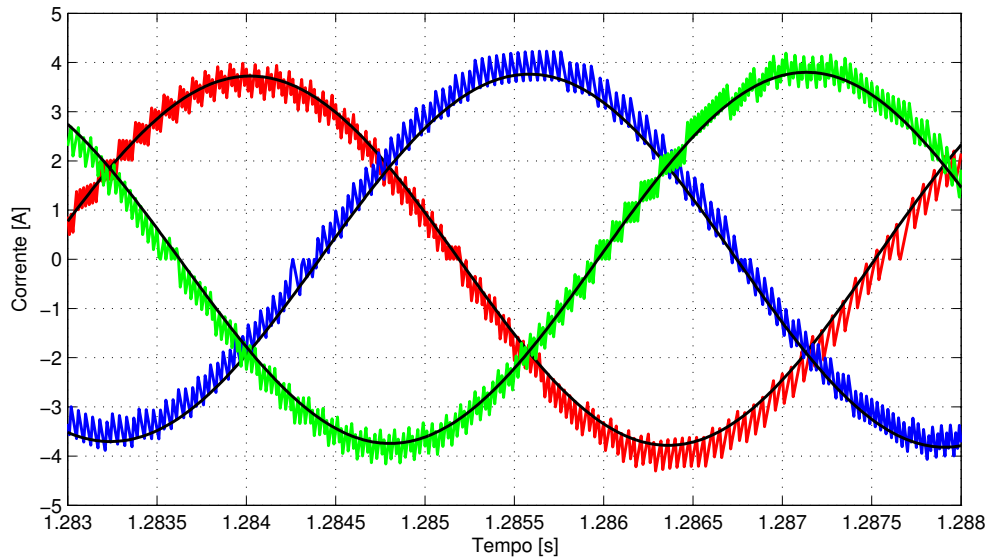


Figura 2.3: Gráfico de corrente sintetizada pelo método de comparação direta.

A fim de facilitar o entendimento do método, deste ponto em diante, a rede estará representada pelo seu equivalente de Thévenin, ou seja, a rede estará sendo representada por uma fonte de tensão e uma impedância em série. A impedância, neste caso, será do tipo indutiva, assim, o conversor se conecta à rede através de um resistor e indutor. A impedância da rede de 1ª ordem é mostrada na Figura 2.4.

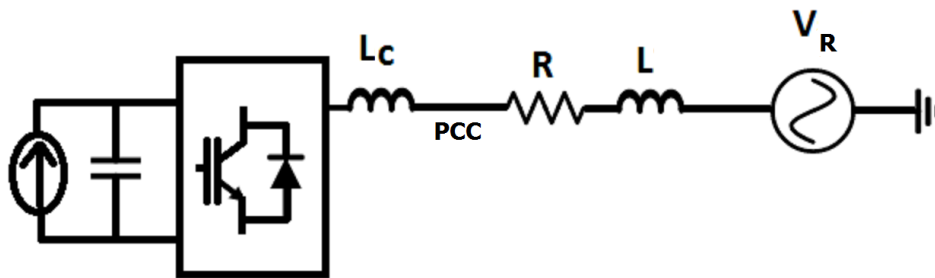


Figura 2.4: Diagrama simplificado do conversor VSC conectado à rede no seu equivalente de Thévenin.

O método consiste em injetar um sinal característico na rede de tal forma que, a partir de sua resposta, os coeficientes de impedância da rede possam ser estimados. Estes coeficientes serão calculados utilizando um algoritmo que é baseado no método dos mínimos quadrados, o mesmo utilizado em [26]. Nesta dissertação, o pulso a ser injetado é do tipo CHIRP conforme é detalhado em [17] e [25] e mostrado na Figura 2.5

A grande vantagem em se utilizar um pulso de corrente no formato CHIRP se deve ao fato do módulo da sua resposta em frequência ser um espectro com

valor constante entre f_0 , frequência inicial do pulso, e f_1 , frequência final do pulso, conforme detalhado em [17] e [25] o que faz excitar as frequências naturais da rede dentro deste espectro.

Nesta dissertação, será injetado um pulso CHIRP similar ao da Figura 2.6 em formato de corrente na rede através do conversor VSC utilizando o método de controle de corrente por comparação direta com uma referência. Este método também pode ser chamado de PWM adaptativo ou de controle por histerese utilizando banda de tolerância nula, uma vez que os limites de corrente superior e inferior são idênticos. A importância deste controle de corrente como a sua flexibilidade e dinamicidade está exposta em [27].

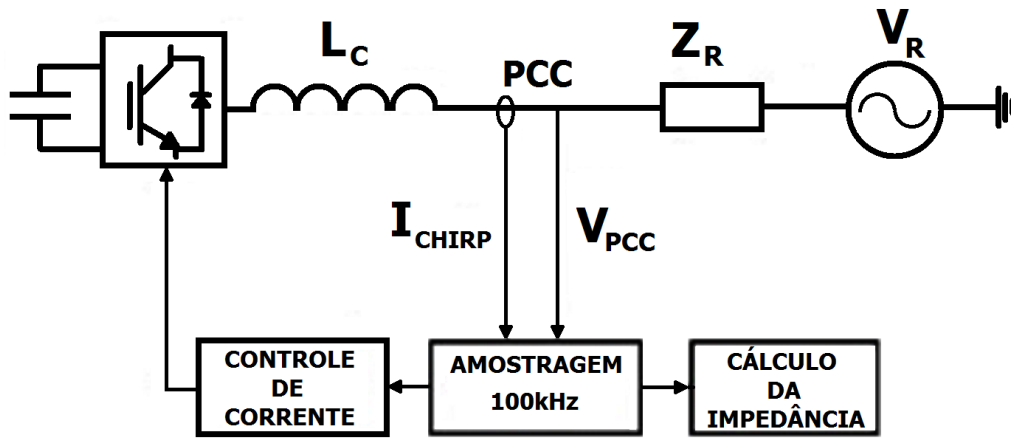


Figura 2.5: Circuito caracterizando a injeção de um pulso CHIRP na rede através do VSC.

Este pulso de corrente em formato CHIRP é um sinal de amplitude constante com uma variação linear de frequência no tempo iniciando de uma frequência inicial f_0 e terminando numa frequência final f_1 caracterizado pela fórmula:

$$x(t) = \sin\left(2\pi t\left(f_0 + \frac{(f_1 - f_0)t}{2T}\right)\right) \quad (2.1)$$

Onde:

x - Amplitude do sinal.

f_0 - Frequência inicial.

f_1 - Frequência final.

T - Diferença de tempo entre o início da aplicação da frequência f_0 no instante t_0 e o instante t_1 quando o sinal está aplicando a frequência f_1 .

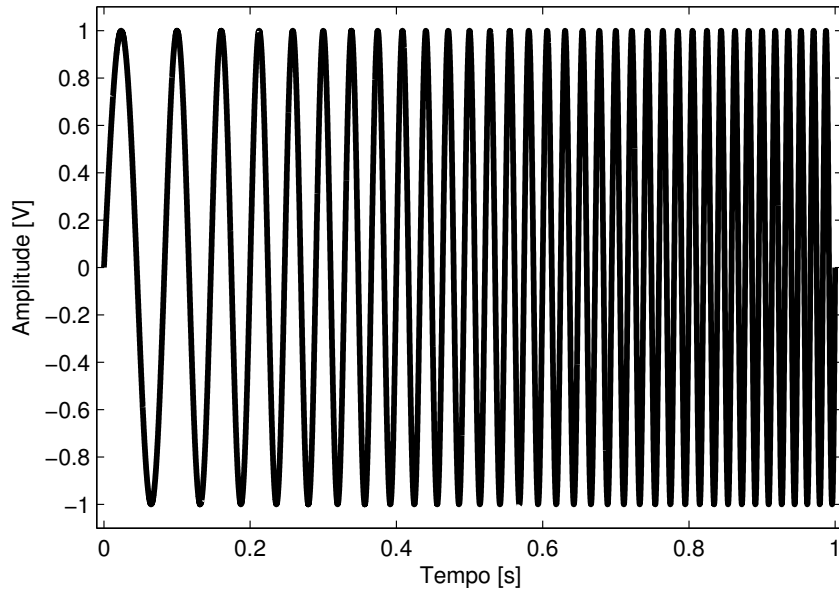


Figura 2.6: Sinal CHIRP com frequência variando de 10Hz a 100Hz .

Na Figura 2.6, pode-se visualizar um sinal CHIRP com frequência variando de 10Hz a 100Hz e na Figura 2.7 seu espectro de frequência. Note que o sinal CHIRP é um sinal cuja frequência varia linearmente com o tempo de uma frequência f_0 inicial a uma frequência final f_1 durante o período T , porém, seu espectro de frequência possui oscilações na banda passante.

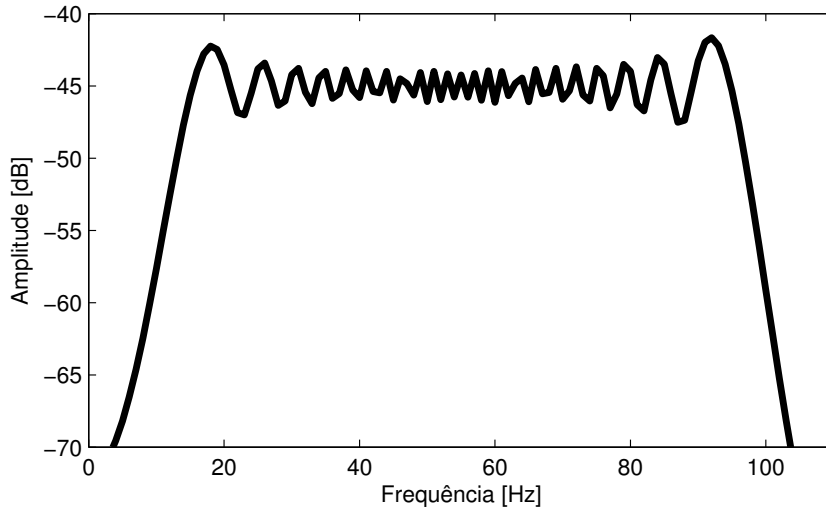


Figura 2.7: Resposta em frequência do sinal CHIRP com $f_0 = 10\text{Hz}$ e $f_1 = 100\text{Hz}$.

Para obter uma banda passante sem oscilações, uma janela Tukey[28] foi utilizada para suprimir estas oscilações indesejáveis no espectro. Segue modelo matemático do sinal CHIRP dentro de uma janela Tukey caracterizado pela equação 2.2 considerando um período de tempo de 1s de janela.

$$\rho(t) = \begin{cases} \frac{1}{2}[1 + \cos(\frac{2\pi}{\alpha} \cdot (t - \frac{\alpha}{2}))] & 0 \leq t < \frac{\alpha}{2} \\ 1 & \frac{\alpha}{2} \leq t < 1 - \frac{\alpha}{2} \\ \frac{1}{2}[1 + \cos(\frac{2\pi}{\alpha} \cdot (t - 1 + \frac{\alpha}{2}))] & 1 - \frac{\alpha}{2} \leq t \leq 1 \end{cases} \quad (2.2)$$

Onde:

ρ - Amplitude do sinal envolto pela janela Tukey.

α - Razão entre a faixa cossenoidal e a faixa total da janela, neste caso, de 1s.

Na Figura 2.8, plota-se o mesmo sinal CHIRP da Figura 2.6 dentro de uma janela Tukey com $\alpha = 0.5$, ou seja, durante o primeiro quarto de tempo de aplicação do sinal, há uma acréscimo cossenoidal de amplitude. Já no último quarto de aplicação, há um decréscimo cossenoidal de amplitude. O efeito disto é mostrado na Figura 2.9, onde é caracterizada a resposta em frequência deste novo sinal envolto por uma janela Tukey. Note que, quando o sinal CHIRP é envolvido por uma janela Tukey, obtém-se uma resposta em frequência plana na banda de interesse. A característica peculiar deste sinal será explorada no algoritmo, pois, se um pulso de corrente com este formato for injetado na rede, a resposta em frequência da tensão medida será proporcional à resposta em frequência da impedância da rede, caso não haja nenhum distúrbio adicional no circuito.

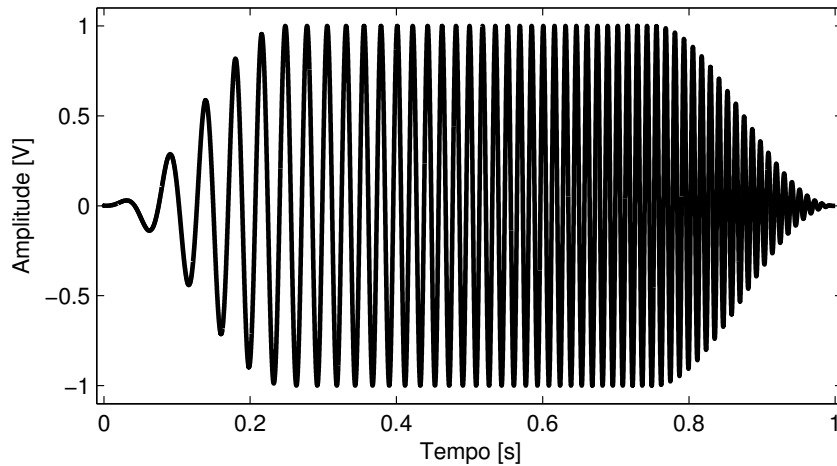


Figura 2.8: Sinal CHIRP dentro de uma janela Tukey com α igual a 50%.

2.3 Análise do Sistema Digital

Nesta seção, será aprofundada a análise da aplicação do sinal CHIRP na rede e do algoritmo modelado num sistema digital. Destaca-se a influência da rede nas

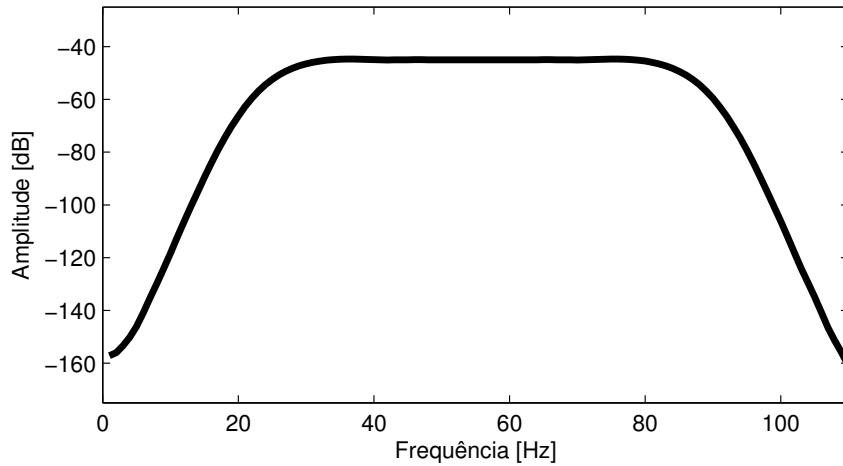


Figura 2.9: Espectro do sinal CHIRP dentro de uma janela Tukey.

medições e a necessidade de uso de um sincronizador de fase e frequência do tipo *phase-locked loop* (PLL) explicando-o resumidamente. Por fim, será detalhado o algoritmo recursivo passando pelo seu modelamento matemático.

Para melhor entendimento do algoritmo de medição de impedância da rede, deve-se explicar porque o sinal CHIRP será importante para esta medição. Para isto, considera-se o tipo mais simplificado de rede representado pelo seu equivalente Thévenin monofásico, como mostrado na Figura 2.10. Neste caso, a impedância da rede poderá ser calculada através da equação 2.3.

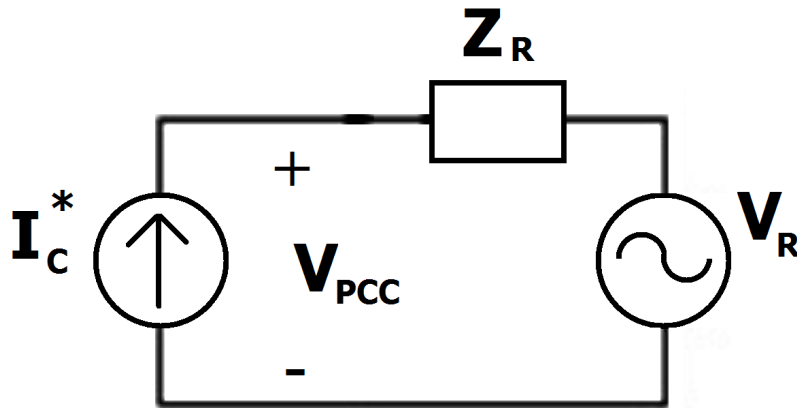


Figura 2.10: Circuito simplificado considerando o conversor VSC como uma fonte de corrente.

$$Z_R = \frac{V_{PCC} - V_R}{I_c^*} \quad (2.3)$$

Onde:

Z_R - Impedância equivalente da rede.

V_{PCC} - Tensão no PCC.

V_R - Tensão da rede.

I_c^* - Corrente em formato CHIRP injetada pelo conversor.

Como visto anteriormente, a resposta em frequência do sinal de corrente I_c^* em formato CHIRP dentro de uma janela Tukey é constante na banda de interesse. Ao analisar a equação (2.3), idealmente, se for aplicada uma fonte de corrente no formato CHIRP, desconsiderando a tensão da rede V_R , a resposta em frequência medida V_{PCC} tende a ser a própria impedância Z_R da rede. O algoritmo avaliado nesta dissertação leva em consideração a tensão na rede nos seus cálculos, porém, minimiza seus efeitos modelando-a como uma perturbação num sistema discretizado. Este modelo será detalhado mais adiante.

Como o algoritmo funcionará de forma discretizada dentro de um microcontrolador, cabe descrever as equações nessa forma. A função de transferência de um sistema digital genérico pode ser expresso na equação (2.4).

$$H(z) = \frac{Y(z)}{X(z)} \quad (2.4)$$

Para este caso de medição de impedância da rede, as variáveis da equação (2.4) se relacionam com as de interesse nesta dissertação:

$H(z) \rightarrow Z_R(z)$ - Impedância discretizada da rede.

$Y(z) \rightarrow V_{PCC}(z)$ - Tensão discretizada medida no PCC .

$X(z) \rightarrow I_c^*(z)$ - Corrente injetada discretizada no formato CHIRP.

Para o circuito descrito na Figura 2.10, caso seja desconsiderada a tensão da rede, a equação (2.3), reescrita como em (2.4), pode ser expandida em (2.5).

$$Z_R(z) = \frac{b_0 + b_1 z^{-1} + \dots + b_m z^{-m}}{1 + a_1 z^{-1} + \dots + a_n z^{-n}} \quad (2.5)$$

Onde:

n - Ordem do sistema e $m \leq n$ de forma a manter a causalidade³.

a_i, b_i - Coeficientes da rede.

Considerando a transformada Z inversa da expressão (2.5) com uma taxa de amostragem fixa no tempo, chega-se à equação (2.6):

³Considerando um sistema causal, o valor máximo de \mathbf{m} deverá ser menor que \mathbf{n} . Como \mathbf{m} também é o número de zeros da equação (2.5), e considerando que a rede possui todos os zeros possíveis na sua equação, o valor de \mathbf{m} , sem perda de generalidade, poderá assumir o valor de $\mathbf{n}-1$.

$$\begin{bmatrix} y_0 \\ \vdots \\ y_k \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} -y_{-1} & \cdots & -y_{-n} & x_0 & \cdots & x_{-m} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -y_{k-1} & \cdots & -y_{k-n} & x_k & \cdots & x_{k-m} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -y_{N-1} & \cdots & -y_{N-n} & x_N & \cdots & x_{N-m} \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_n \\ b_0 \\ \vdots \\ b_m \end{bmatrix}$$

$$y_k = -a_1 y_{k-1} - \dots - a_n y_{k-n} + b_0 x_k + b_1 x_{k-1} + \dots + b_m x_{k-m} \quad (2.6)$$

Ao adquirir $N + 1$ amostras, obtém-se um conjunto de equações discretas no tempo que podem ser agrupadas em forma de matriz como segue adiante:

As matrizes acima podem ser substituídas por uma equação reduzida descrita em (2.7):

$$Y = A.\Theta \quad (2.7)$$

Onde, considerando ($m = n - 1$), tem-se que:

Y - Vetor de dimensão $[(N + 1) \times 1]$ contendo as medidas atuais.

A - Matriz de dimensão $[(N + 1) \times (2n)]$ contendo as medidas passadas de tensão e corrente.

Θ - Vetor de dimensão $[(2n) \times 1]$ contendo os coeficientes da rede a serem estimados.

Utilizando o método dos mínimos quadrados [29], pode-se estimar a matriz Θ em (2.7) aplicando a fórmula em (2.8):

$$\Theta = ((A^T A)^{-1} A^T) Y \quad (2.8)$$

Isso significa dizer que, para estimar os coeficientes da matriz Θ em modo *offline*, todos os valores de tensão devem ser adquiridos a partir da injeção de corrente e as matrizes Y e A devem ser totalmente preenchidas antes de se iniciarem os cálculos. Em sequência, deve-se aplicar a equação em (2.8) obtendo, assim, a matriz Θ com os coeficientes estimados a partir do método dos mínimos quadrados.

O principal problema encontrado neste método se deve ao fato da necessidade de uma quantidade considerável de memória no sistema para o cálculo da impedância. A matriz A contém todas as amostras de corrente e tensão e o sistema deve possuir espaço suficiente em memória tanto para a matriz A quanto para a sua inversa e sua transposta. Além disso o tempo exigido de processamento é naturalmente grande.

Nesta dissertação, este sistema é resolvido fazendo-se uso de um método *online*, pois o conversor se mantém conectado à rede em todo o tempo, cujo processamento se dará através de matrizes recursivas em *tempo real*⁴ Este método economiza espaço no processamento e traz dinamicidade à estimação da impedância da rede.

2.4 Influência da Rede nas Medições

Antes de explicar o algoritmo recursivo, deve ser considerada a influência da rede nos cálculos da impedância. Conforme foi explicado no início desta seção, a impedância da rede pode ser calculada utilizando-se a expressão em (2.8) apenas se a tensão V_R referente à tensão da rede for ignorada. Porém, em casos reais, como o conversor VSC está conectado à rede, a sua tensão deve ser considerada no cálculo dos coeficientes de impedância. Portanto, de cada medição de tensão V_{PCC} deve ser subtraída a tensão V_R naquele instante como mostra a Figura(2.11) onde a tensão da rede é interpretada como sendo uma perturbação na saída do sistema que deve ser eliminada.

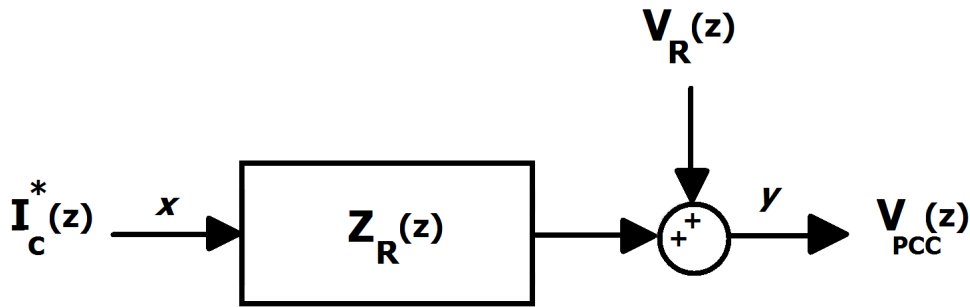


Figura 2.11: Diagrama em blocos do sistema digital considerando a rede como uma perturbação no sistema.

Como não é possível medir diretamente a tensão da rede, há a necessidade de estimá-la utilizando a expressão em (2.9). Ela diz que a tensão da rede é formada pelo somatório de harmônicos presentes na rede juntamente com sua componente fundamental.

$$V_R = \sum_{\gamma=0}^M V_{C\gamma} \cos(\omega_\gamma t) + V_{S\gamma} \sin(\omega_\gamma t) \quad (2.9)$$

Onde,

$V_{C\gamma}$ - Amplitude da componente cossenoidal do harmônico γ .

⁴Ou seja, o processamento ocorre em paralelo com a aquisição de dados e o resultado é gerado logo após o fim da aquisição dos mesmos.

$V_{S\gamma}$ - Amplitude da componente senoidal do harmônico γ .

ω_γ - Frequência angular do harmônico γ .

M - Total de harmônicos presentes na rede.

Assim, considerando que a rede esteja presente e que não possua harmônicos, a equação (2.7) deverá ser reescrita anexando-se a frequência da fundamental à matriz A conforme mostra a equação (2.10) onde se adiciona a tensão da rede inferida em cada medida:

$$\begin{bmatrix} y_0 \\ \vdots \\ y_k \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} -y_{-1} & \cdots & -y_{-n} & x_0 & \cdots & x_{-m} & \cos(\theta_0^0) & \sin(\theta_0^0) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -y_{k-1} & \cdots & -y_{k-n} & x_k & \cdots & x_{k-m} & \cos(\theta_k^0) & \sin(\theta_k^0) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -y_{N-1} & \cdots & -y_{N-n} & x_N & \cdots & x_{N-m} & \cos(\theta_N^0) & \sin(\theta_N^0) \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_n \\ b_0 \\ \vdots \\ b_m \\ V_{C_0} \\ V_{S_0} \end{bmatrix} \quad (2.10)$$

Onde:

θ_k^0 - Fase referente à componente fundamental da rede no instante de amostragem k .

Neste momento, vale ressaltar que, para completar a matriz A, serão necessárias algumas medidas de saída passadas, ou seja, anteriores ao estado zero, especificamente, \mathbf{n} medidas de saída passadas. São elas: $y_{-1}, y_{-2}, \dots, y_{-n}$. Também serão necessárias medidas de entrada no estado zero e algumas anteriores ao estado inicial, especificamente, \mathbf{m} medidas de entrada passadas. São elas: $x_{-1}, x_{-2}, \dots, x_{-m}$. Como a simulação inicia-se no estado zero, ou seja, o início da tomada de medidas não considera as medidas anteriores ao estado inicial, para contornar este problema, a matriz Y será preenchida a partir do passo de integração k igual a \mathbf{n} , consequentemente, as dimensões das matrizes diminuirão. Considerando esta restrição, a equação (2.10) poderá ser reescrita na equação 2.11

$$\begin{bmatrix} y_n \\ \vdots \\ y_k \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} -y_{n-1} & \cdots & -y_0 & x_n & \cdots & x_{n-m} & \cos(\theta_0^0) & \sin(\theta_0^0) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -y_{k-1} & \cdots & -y_{k-n} & x_k & \cdots & x_{k-m} & \cos(\theta_k^0) & \sin(\theta_k^0) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -y_{N-1} & \cdots & -y_{N-n} & x_N & \cdots & x_{N-m} & \cos(\theta_N^0) & \sin(\theta_N^0) \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_n \\ b_0 \\ \vdots \\ b_m \\ V_{C_0} \\ V_{S_0} \end{bmatrix} \quad (2.11)$$

De forma a generalizar ainda mais o resultado, caso a rede possua a sua componente fundamental e demais harmônicos, a equação matricial (2.11) poderá ser reescrita como em (2.12):

$$\begin{bmatrix} y_n \\ \vdots \\ y_k \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} & \cos(\theta_o^1) & \sin(\theta_o^1) & \cdots & \cos(\theta_o^\gamma) & \sin(\theta_o^\gamma) & \cdots & \cos(\theta_o^M) & \sin(\theta_o^M) \\ & \vdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ A_{ant} & \cos(\theta_k^1) & \sin(\theta_k^1) & \cdots & \cos(\theta_k^\gamma) & \sin(\theta_k^\gamma) & \cdots & \cos(\theta_k^M) & \sin(\theta_k^M) \\ & \vdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ & \cos(\theta_N^1) & \sin(\theta_N^1) & \cdots & \cos(\theta_N^\gamma) & \sin(\theta_N^\gamma) & \cdots & \cos(\theta_N^M) & \sin(\theta_N^M) \end{bmatrix} \begin{bmatrix} \Theta_{ant} \\ V_{C_1} \\ V_{S_1} \\ \vdots \\ V_{C_\gamma} \\ V_{S_\gamma} \\ \vdots \\ V_{C_M} \\ V_{S_M} \end{bmatrix} \quad (2.12)$$

Onde:

A_{ant} - Matriz A da equação (2.11).

Θ_{ant} - Vetor Θ da equação (2.11).

θ_k^γ - Fase referente ao harmônico γ no instante de amostragem k .

Como observado na equação (2.12), se forem considerados a rede e seus harmônicos, em comparativo com a equação (2.11), a matriz A irá expandir as suas colunas e a matriz Θ irá aumentar o número de elementos. A matriz A terá um acréscimo de duas vezes M de colunas e a matriz Θ em duas vezes M o número de elementos, sendo M o número de harmônicos presentes na rede.

Portanto, a equação matricial (2.12) poderá ser generalizada na equação (2.13) a mesma expressa em (2.7), porém, agora, considerando todos os fatores já mencionados:

$$Y = A_{exp} \cdot \Theta_{exp} \quad (2.13)$$

Onde:

Y - Vetor de dimensão $[(N - n + 1) \times 1]$ contendo as medidas de tensão atuais;

A_{exp} - Matriz A de dimensão $[(N - n + 1) \times (2n + 2 + 2M)]$ contendo cossenos e senos das fases de cada harmônico da rede;

Θ_{exp} - Vetor Θ de dimensão $[(2n + 2 + 2M) \times 1]$ contendo todos os coeficientes e amplitudes de cada harmônico da rede;

Assim, considerando a influência da rede, a equação matricial (2.13) poderá ser utilizada para calcular os coeficientes da impedância da rede. Caso se queira estimar a admitância da rede, de forma inversa e sem comprometer o cálculo final, deve-se aplicar tensão na rede e medir a corrente drenada pelo conversor. Portanto, da mesma forma que foi utilizado o método dos mínimos quadrados em [29] no caso sem a influência da rede, pode-se estimar também a matriz Θ_{exp} em (2.13) utilizando o mesmo método e, assim, obter os coeficientes da impedância da rede, agora considerando a influência da rede e de seus harmônicos no cálculo. A respectiva fórmula é mostrada em (2.14):

$$\Theta_{exp} = ((A_{exp}^T A_{exp})^{-1} A_{exp}^T) Y \quad (2.14)$$

Nesta dissertação, será injetada corrente na rede através do VSC e medida a tensão na saída do conversor fazendo-se a estimativa de impedância da rede. Como foi visto, uma grande vantagem deste método é a capacidade de rejeitar harmônicos de tensão da rede ao modelá-los como sendo perturbações no sistema.

A partir da tensão medida V_{PCC} , deve-se considerar, em cada passo, a contribuição da tensão da rede V_R de acordo com (2.9). Qualquer outro harmônico da rede poderá ser considerado adicionando-se mais colunas na matriz A expandida. Neste caso, mais colunas em cossenos e senos devem ser adicionadas à equação (2.13) correspondentes a cada harmônico da rede. Para definir quantos harmônicos a rede possui num dado instante, pode ser feita previamente uma análise harmônica em V_{PCC} , anexando-se, posteriormente, os harmônicos encontrados à expressão (2.13).

Assim, o método pode ser resumido da seguinte forma: injeta-se uma corrente no formato CHIRP envolta por uma janela Tukey e mede-se a tensão na saída do conversor considerando no cálculo, em cada instante, a fase de cada harmônico de tal forma que no final pode-se estimar os coeficientes da rede. Como o sinal injetado possui banda constante unitária, o sinal de tensão medido se confunde com a própria impedância da rede.

No apêndice A será descrito o uso do PLL no algoritmo recursivo. Ele se faz necessário pois, como deve ser encontrado o valor instantâneo do ângulo de fase no PCC em cada instante, um PLL deverá ser utilizado para se encontrar a frequência da fundamental e de seus harmônicos.

2.5 Algoritmo Recursivo

Nesta dissertação, foi empregado um algoritmo recursivo que calcula a impedância da rede utilizando a expressão em (2.13), ou seja, sem a necessidade de construção das matrizes Y , A_{exp} e Θ_{exp} pois o cálculo será efetuado dinamicamente em tempo real. Portanto, em vez de adquirir todas as medidas de resposta da rede, construir as matrizes de acordo com a equação matricial em (2.13) e calcular os coeficientes pelo método dos mínimos quadrados, o algoritmo realiza este cálculo de forma bem mais simples, rápida e compacta, ou seja, de forma recursiva. De forma a melhorar a visualização do algoritmo e considerando que todo o conversor considerado nesta dissertação estará conectado à rede e que, possivelmente, conterá harmônicos, as matrizes expandidas A_{exp} e Θ_{exp} da equação (2.13) serão chamadas apenas de matrizes A e Θ sem perda de generalidade e conteúdo.

Utilizando este algoritmo recursivo, o microcontrolador responsável por realizar o controle do conversor não necessita de uma memória de processamento grande o suficiente para processar todas as matrizes da equação (2.13). Pelo contrário, o microcontrolador irá precisar apenas de memória suficiente para calcular algumas instruções básicas utilizando matrizes de menor ordem proporcionais à ordem da rede.

Este algoritmo proposto é baseado no método dos mínimos quadrados e estima os coeficientes da equação característica da rede de forma recursiva, ou seja, o seu cálculo é obtido de forma iterativa, a cada passo de amostragem e se aproxima dos valores reais à medida que novas medidas são tomadas. Basicamente, o algoritmo funciona da seguinte forma: a cada passo de integração, é construído um vetor coluna Γ contendo medidas recém obtidas de corrente e tensão e também de medidas passadas. Este vetor é sempre atualizado com as novas medidas tomadas a cada iteração k . No final de cada iteração k , uma matriz ξ de tamanho proporcional ao quadrado da ordem n é construída a partir do vetor coluna Γ já atualizado. Esta matriz ξ , após construída no instante atual de iteração, servirá para atualizar a matriz Ξ contendo todo o somatório de matrizes ξ construídas até o instante atual. No mesmo passo de integração, também é atualizado, e sempre a cada instante de amostragem, um vetor coluna chamado Φ , formado por medidas passadas e atuais de corrente e tensão. Finalmente, após a última iteração, há a inversão da matriz Ξ e posterior multiplicação pelo vetor coluna Φ obtendo, assim, uma estimativa dos coeficientes de impedância da rede de tal forma que o cálculo final é construído ao longo das iterações, ou seja, de forma recursiva.

Para explicar o funcionamento do algoritmo, algumas variáveis precisam ser primeiramente definidas. O vetor coluna $\Gamma(k)$, como já dito, é modificado a cada instante k . Este vetor Γ é definido como sendo a linha transposta da matriz A da

equação (2.13) no instante k , ou seja, o vetor Γ é definido como:

$$\Gamma(k) \equiv \left[-y(k-1) \quad \cdots \quad -y(k-n) \quad x(k) \quad \cdots \quad x(k-m) \quad \cos(\theta_k^0) \quad \sin(\theta_k^0) \right]^T \quad (2.15)$$

Necessariamente, o vetor Γ deverá ser construído a partir do instante ($k = n$), pois serão iniciadas as tomadas de medidas a partir do instante ($k = 0$). Antes deste instante, os valores de resposta do sistema são indefinidos a menos que seja considerado que o sistema possua resposta nula (sistema relaxado) antes do instante inicial $y(0)$ quando não há corrente injetada pelo conversor na rede até este instante ($k = 0$) o que não se pode garantir já que o conversor está conectado à rede.

Com a definição do vetor coluna Γ , a matriz A poderá ser reescrita da seguinte forma:

$$A \equiv \left[\Gamma^T(n) \quad \Gamma^T(n+1) \quad \cdots \quad \Gamma^T(k) \quad \cdots \quad \Gamma^T(N-1) \quad \Gamma^T(N) \right]^T \quad (2.16)$$

Será definida também uma matriz Ξ como sendo:

$$\Xi \equiv A^T \cdot A \quad (2.17)$$

Com esta definição, obtém-se o produto:

$$\Xi \equiv \left[\Gamma(n) \quad \cdots \quad \Gamma(k) \quad \cdots \quad \Gamma(N) \right] \cdot \left[\Gamma^T(n) \quad \cdots \quad \Gamma^T(k) \quad \cdots \quad \Gamma^T(N) \right]^T \quad (2.18)$$

Observando a equação acima, nota-se que a matriz Ξ pode ser entendida como sendo um produto de dois vetores, chegando-se à expressão final:

$$\Xi = \Gamma(n) \cdot \Gamma^T(n) + \cdots + \Gamma(k) \cdot \Gamma^T(k) + \cdots + \Gamma(N) \cdot \Gamma^T(N) \quad (2.19)$$

A matriz Ξ poderá ser reescrita em formato de somatório como segue:

$$\Xi = \sum_{k=n}^N \Gamma(k) \cdot \Gamma^T(k) \quad (2.20)$$

Pode ser definida, neste momento também, a matriz ξ como sendo o produto dos vetores $\Gamma(k)$ e $\Gamma(k)^T$ no instante de amostragem k como segue:

$$\xi(k) \equiv \Gamma(k) \Gamma(k)^T \quad (2.21)$$

Portanto, é definida mais uma variável Ξ da seguinte forma:

$$\Xi = \sum_{k=n}^N \xi(k) \quad (2.22)$$

Onde, $\xi(k)$ é uma matriz quadrada de ordem $(2n + 2 + 2M)$ gerada pelo produto $\Gamma(k).\Gamma^T(k)$ no instante k que varia de n até N . Neste ponto, vale salientar que a matriz Ξ é incrementada a cada iteração k .

Agora, define-se uma matriz Φ como sendo o produto da matriz A transposta pelo vetor coluna Y contendo as medidas de tensão:

$$\Phi \equiv A^T.Y \quad (2.23)$$

Portanto:

$$\Phi = \begin{bmatrix} \Gamma(n) & \Gamma(n+1) & \cdots & \Gamma(k) \cdots & \Gamma(N-1) & \Gamma(N) \end{bmatrix}.Y \quad (2.24)$$

Como a matriz Y é o vetor coluna contendo todos os valores de saída, a matriz Φ pode ser reescrita da seguinte forma:

$$\Phi = \Gamma(n)y_n + \cdots + \Gamma(k)y_k \cdots + \Gamma(N)y_N \quad (2.25)$$

Reescrevendo a matriz Φ em formato de somatório:

$$\Phi = \sum_{k=n}^N \Gamma(k).y_k \quad (2.26)$$

Pode ser definida também a matriz ϕ como sendo o produto do vetor $\Gamma(k)$ pelo valor amostrado de saída atual y_k no instante de amostragem k como segue:

$$\phi(k) \equiv \Gamma(k)y_k \quad (2.27)$$

Portanto, a matriz Φ pode ser reescrita da seguinte forma:

$$\Phi = \sum_{k=n}^N \phi(k) \quad (2.28)$$

Onde, $\phi(k)$ é um vetor coluna gerado pelo produto escalar entre o vetor coluna $\Gamma(k)$ e o escalar y_k no instante k que varia de n até N . Vale ressaltar que a matriz Φ é incrementada a cada iteração k .

Resumidamente, são listadas as variáveis criadas para o método recursivo:

Γ - Vetor de dimensão $[(2n+2+2M) \times 1]$ contendo elementos da linha transposta da matriz A .

Ξ - Matriz quadrada de dimensão $[(2n + 2 + 2M) \times (2n + 2 + 2M)]$ gerada pelo produto da matriz A transposta com a matriz A.

ξ - Matriz quadrada de dimensão $[(2n + 2 + 2M) \times (2n + 2 + 2M)]$ gerada pelo produto dos vetores Γ transposto com o vetor Γ no instante de amostragem k .

Φ - Vetor de dimensão $[(2n + 2 + 2M) \times 1]$ obtido pelo produto da matriz A transposta pelo vetor coluna Y.

ϕ - Vetor de dimensão $[(2n + 2 + 2M) \times 1]$ obtido pelo produto escalar entre o vetor Γ e o valor amostrado de saída y_k no instante de amostragem k .

Dadas as definições acima, a expressão em (2.8) poderá ser reescrita em (2.31)

$$\Theta = ((A^T A)^{-1} A^T).Y \quad (2.29)$$

$$\Theta = (A^T A)^{-1}.(A^T Y) \quad (2.30)$$

$$\Theta = \Xi^{-1}\Phi \quad (2.31)$$

Após serem definidos estes novos termos, nota-se que tanto a matriz Ξ como a matriz Φ são atualizadas a cada iteração pois são somatórios de medidas passadas com a atual. No final da amostragem, é computada a matriz Θ utilizando a expressão em (2.31).

Portanto, para calcular a impedância da rede utilizando o algoritmo recursivo, a memória do microcontrolador será ocupada apenas com a matriz $\xi(k)$ quadrada de ordem $[(2n + 2 + 2M)]$ e com o vetor coluna $\phi(k)$ de tamanho $[(2n + 2 + 2M)]$ cujos tamanhos exigem bem menos espaço na memória que a matriz A construída para o cálculo através do método de mínimos quadrados de dimensões $[(N - n + 1) \times (2n + 2 + 2M)]$. Exemplificando, suponha-se que o número total de amostras, \mathbf{N} , seja igual a 100k, que a ordem da rede n seja igual a 2, que o valor de m seja igual a 1 e que o número de harmônicos presentes na rede seja zero. A memória inicial ocupada para o cálculo através do método dos mínimos quadrados deverá ser da ordem de 600k posições ocupada essencialmente pela matriz A, porém, lembrando que a matriz A deverá ser transposta ocupando o mesmo espaço em memória que a matriz A. Portanto, o total ocupado de memória será em torno de 1200k posições. Já no algoritmo recursivo o mesmo resultado é alcançado ocupando um espaço de 42 posições (= 6x6 posições da matriz ξ somado com 6 posições do vetor ϕ), ou seja, o espaço ocupado utilizando o algoritmo recursivo é da ordem de 28k vezes menor que o ocupado utilizando o método dos mínimos quadrados.

Na Tabela 2.1, foram resumidas as vantagens e desvantagens do Método Recursivo utilizado neste trabalho e do Método dos Mínimos Quadrados.

Processamento	MMQ	Método Recursivo
Tipo	<i>Offline</i>	Tempo Real
Memória Consumida	Grande	Pequena
Velocidade Exigida	Baixa	Alta

Tabela 2.1: Vantagens e Desvantagens do MMQ e do Método Recursivo.

Resumidamente, poderá ser escrito o método proposto em forma de algoritmo cujo começo se dá com a inicialização das variáveis: k , Ξ e Φ . Lembrando que elas serão atualizadas a cada iteração. Em seguida, se inicia a injeção de corrente no formato CHIRP e a aquisição de tensão da rede. A medição de tensão é feita a cada passo de iteração. A cada passo, também, é atualizado o vetor Γ com as novas medidas adquiridas, calculam-se os novos vetores ξ e ϕ e atualizam-se as matrizes Ξ e Φ . Finalmente, após as $(N - n)$ iterações, a matriz Θ é computada e os coeficientes da impedância da rede são obtidos. Neste trabalho, foi utilizado o Método de Eliminação de Gauss para se calcular a matriz Θ conforme programa do MCU transcrito no apêndice.

A seguir, foi transcrito o algoritmo recursivo utilizado nas simulações desta dissertação.

```

1:  $k \leftarrow n$ 
2:  $\Xi \leftarrow 0$ 
3:  $\Phi \leftarrow 0$ 
4:  $fase_n \leftarrow PLL$ 
5: while  $k \leq N$  do
6:    $\langle i_k \rangle \leftarrow chirp$ 
7:    $[y_k, x_k] \leftarrow rede$ 
8:    $fase_k \leftarrow fase_{k-1} + \Delta fase$ 
9:    $\Gamma \leftarrow [y_k, x_k, fase_k]$ 
10:   $\xi \leftarrow \Gamma \cdot \Gamma^T$ 
11:   $\phi \leftarrow \Gamma \cdot y$ 
12:   $\Xi \leftarrow \Xi + \xi$ 
13:   $\Phi \leftarrow \Phi + \phi$ 
14:   $k \leftarrow k + 1$ 
15: end while
16:  $\Theta = \Xi^{-1} \Phi$ 
17: return  $\Theta$ 

```

Na Figura 2.12, é mostrado o diagrama de blocos completo do algoritmo de medição de impedância da rede.

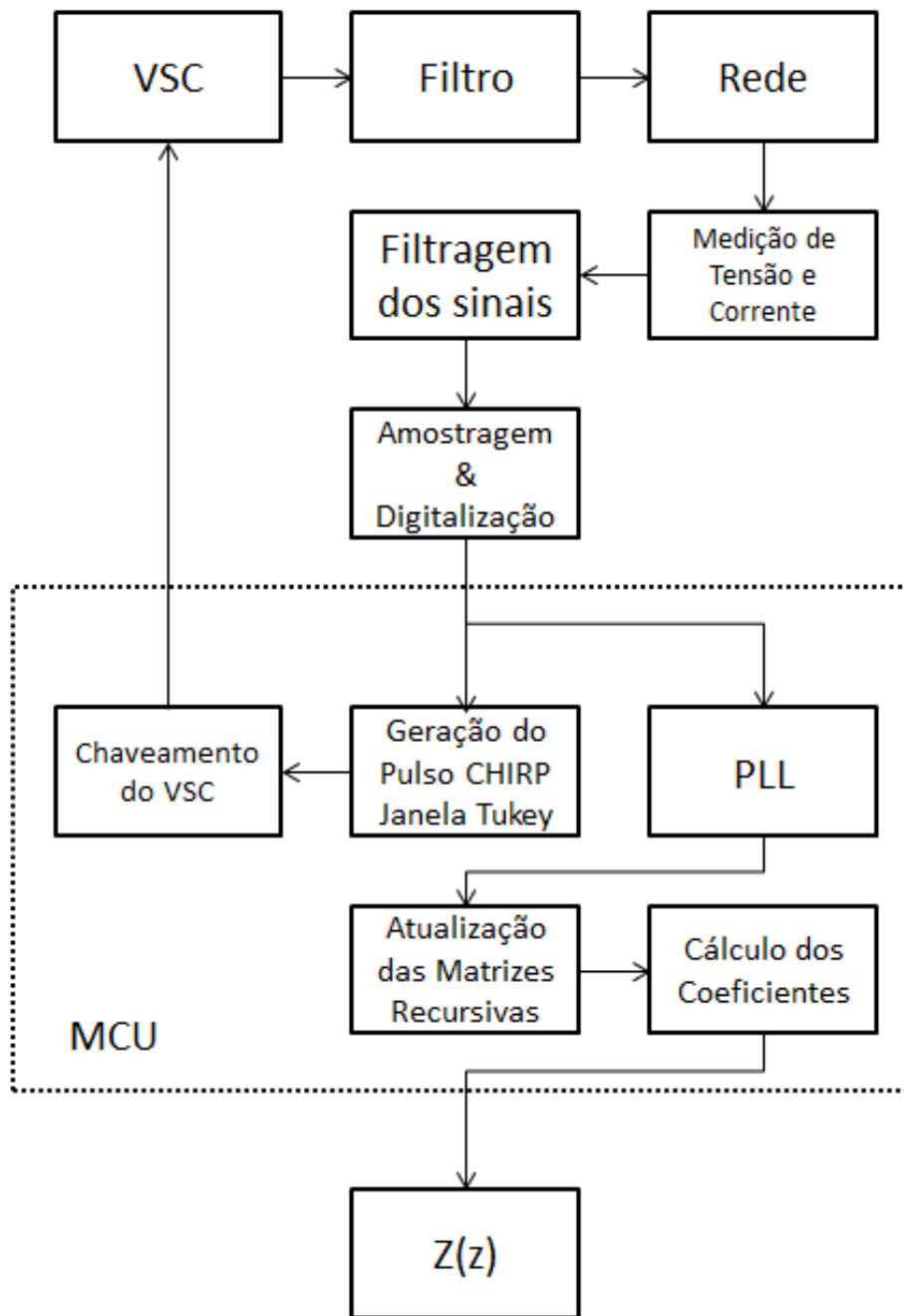


Figura 2.12: Diagrama de Blocos completo do Método Recursivo de Medição de Impedância da Rede testado neste trabalho.

Assim, foi concluída a explicação do algoritmo utilizado e, a partir deste ponto, são mostrados os parâmetros das simulações elaboradas e os resultados alcançados.

Capítulo 3

Detalhamento da Simulação

Neste capítulo, é descrito o ambiente de simulação, caracterizado o conversor VSC empregado e detalhados os parâmetros das simulações.

3.1 Ambiente de Simulação

O algoritmo foi testado num ambiente *hardware-in-the-loop* ou seja, um MCU foi empregado em conjunto com um simulador de circuitos. O MCU utilizado foi o baseado numa CPU de 32bits, com processamento em ponto flutuante de 200Mhz de velocidade, modelo *TMS320F28377S DelfinoTM* da *Texas Instruments*. Dessa forma, todo o processamento do algoritmo foi realizado dentro do MCU, ou seja, desde a obtenção dos sinais digitalizados correspondentes às entradas analógicas até o processamento dos sinais de controle a serem enviados ao conversor VSC. Além disso, o MCU foi o responsável por aplicar o algoritmo recursivo proposto em tempo real entregando, ao final do algoritmo, os coeficientes da função de transferência da impedância da rede. O processador estava embarcado numa placa de interface LAUNCHXL-F28377S com interface via portas seriais (SCI/UARTS) com o computador.

A simulação do circuito elétrico foi realizada dentro do software PSIM da empresa *Powersim INC*, com solução em passo fixo, projetado especificamente para simulação de circuitos de eletrônica de potência. Dentro do ambiente do PSIM, um bloco DLL foi criado e, através do mesmo, um arquivo de extensão (*.dll) é chamado o qual contém o programa de interface com o MCU. Este programa foi escrito na linguagem C++ onde estão contidas algumas bibliotecas responsáveis pela abertura e fechamento das portas seriais de acordo com o protocolo RS-232 para interface com o MCU externo ao computador.

Na Figura 3.1, é mostrado o bloco de DLL criado no PSIM responsável pela interface com o MCU para uma rede de ordem 2. Cada entrada de sinal do lado direito do bloco corresponde a uma variável de 32 bits que é enviada do PSIM

pela DLL para o MCU via porta serial. À esquerda, são posicionados os sinais de controle provenientes do MCU. Note que o bloco DLL recebe os sinais de corrente e de tensão do circuito no PLL e responde com os sinais de chaveamento dos IGBT's. Além desses sinais de controle, ao final da simulação, são enviados os coeficientes calculados pelo algoritmo.

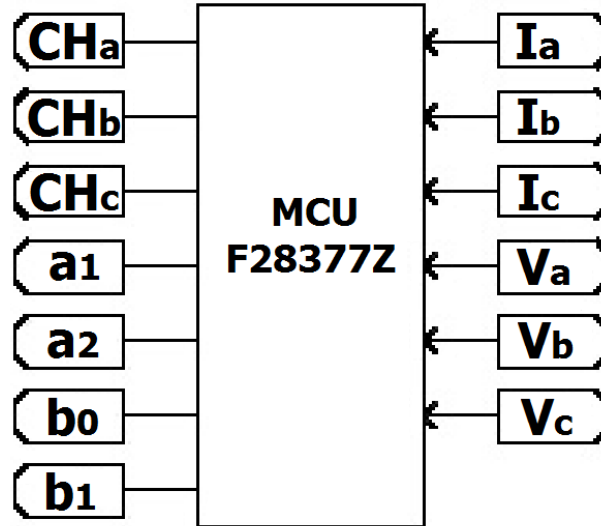


Figura 3.1: Caixa DLL de interface com o MCU para uma rede de 2ª ordem.

Resumidamente, o processo *hardware-in-the-loop* funciona da seguinte forma: o MCU recebe os sinais de corrente e de tensão provenientes do software de simulação PSIM, discretizando e digitalizando os mesmos. Depois, realiza o método de comparação direta de sintetização de corrente gerando os sinais de controle de chaveamento dos IGBT's. A saída de corrente do conversor VSC terá dessa maneira o formato de um pulso CHIRP. Durante a excitação da rede por esta corrente, o MCU adquire os sinais de resposta da rede, realizando o processamento do algoritmo e entregando os coeficientes da rede ao final da excitação.

3.2 Características do Conversor

O conversor utilizado é do tipo fonte de tensão (VSC) com seu controle de corrente sendo realizado pelo método de comparação direta. A corrente aplicada pelo conversor na rede é comparada com a corrente de referência em formato CHIRP a cada iteração, injetando-se assim, o tipo de corrente desejado na rede. Neste momento, vale ressaltar que este método aplica a corrente em formato CHIRP sem injetar potência na rede já que não é este o propósito. Neste algoritmo, o conversor será utilizado apenas para medição de impedância da rede. Tendo em mãos os coefici-

entes da rede, diversos ajustes podem ser feitos, posteriormente, no seu controle de corrente de forma a otimizar a transferência de potência para a rede. Esta segunda etapa de aplicação dos coeficientes obtidos não faz parte do escopo deste trabalho.

O conversor trifásico simulado é formado por seis conjuntos de IGBT's com diodos em anti-paralelo e três reatores na saída, um em cada fase. O conversor simulado possui 19,05kVA de potência com tensão de linha no lado AC da rede fixada em $220V_{rms}$. A tensão de linha base assumida é de $V_{base} = 220V_{rms}$ e a potência base é $P_{base} = 19,05kVA$. Com estes valores de potência e de tensão de linha base, a corrente base foi calculada em $I_{base}=50A$ e a impedância base em $Z_{base} = 2,54\Omega$.

Conforme mostra a literatura, é recomendável que o reator de saída do conversor tenha impedância (Z_c) fixada em 10% da impedância base Z_{base} . A tensão base no elo CC (V_{CCbase}) foi fixada em 1,35 vezes a tensão de linha base.

Um outro ponto importante a ressaltar a respeito do conversor utilizado é a necessidade do pré-carregamento do capacitor do elo CC antes da execução do método de medição de impedância da rede. Como já ressaltado, durante o método, o conversor deixa de cumprir o seu objetivo de transferir temporariamente energia para a rede e passa a executar somente o algoritmo de medição. Porém, no momento em que se inicia o método se faz necessário o pré-carregamento do elo CC para que haja diferença de pontencial necessária para a excursão da corrente sintetizada na saída.

Todos os valores dos parâmetros do conversor foram resumidos na Tabela 3.1:

Parâmetros do Conversor	Notação	Valor
Potência do conversor	P_{conv}	19,02kVA
Tensão de linha base	V_{base}	$220V_{rms}$
Impedância base	Z_{base}	1,27 Ω
Corrente base	I_{base}	50A
Impedância do reator	Z_c	0,1pu
Indutância do reator	L_c	337 μH
Tensão base no elo CC	V_{CCbase}	1,35 V_{base}
Tensão de pré-carregamento do Elo CC	V_{CC}	1,5pu

Tabela 3.1: Parâmetros do conversor.

3.3 Parâmetros da Simulação

Com o objetivo de excitar o maior intervalo de frequências possível da rede e, assim, obter o espectro de frequência da impedância da rede com maior resolução, foi empregado um sinal CHIRP variando de 0Hz até 3000Hz.

Como as variáveis internas do firmware do MCU são de 32 bits, elas possuem

apenas 7 algarismos significativos. Isto implica que se deve limitar os valores acumulados nas matrizes recursivas de tal forma que no cálculo dos coeficientes não ocorram operações que excedam o número de algarismos significativos causando uma estimação errônea dos coeficientes da rede. Considerando que as variáveis Ξ e Φ acumulam matrizes a cada iteração, deve-se limitar o incremento total dos valores, ou limitando-se o número de iterações ou diminuindo-se o valor a ser incrementado. O primeiro objetivo é alcançado diminuindo-se a frequência de amostragem e limitando o tempo de injeção do sinal CHIRP e, conseqüentemente, limitando-se do acúmulo dos valores. O segundo objetivo é atingido acumulando-se os valores por unidade (pu) de tensão e corrente nas matrizes recursivas.

Tendo em vista a necessidade de redução da taxa de amostragem, foi utilizada, portanto, a taxa de 20kHz. Como o passo de amostragem, ou intervalo de tempo entre amostras, é o inverso da taxa, este será de $50\mu s$. A corrente foi injetada na rede no formato CHIRP desde o instante $t = 50ms$ até o final da simulação em $t = 250ms$. O tempo total de funcionamento do algoritmo recursivo é, portanto, de apenas $200ms$, sendo que na primeira etapa da simulação, de $t = 0s$ a $t = 50ms$, o algoritmo é dedicado apenas ao PLL encontrar a fase do sistema trifásico de entrada do conversor, conforme mostra a Figura 3.2. A segunda etapa da simulação, de $t = 50ms$ a $t = 250ms$, o algoritmo é dedicado ao cálculo propriamente dito dos coeficientes. Note no primeiro gráfico da Figura 3.2 que a fase encontrada se mantém mesmo após a rede começar a gerar harmônicos devido à injeção de corrente no formato CHIRP na rede no segundo gráfico da Figura 3.2.

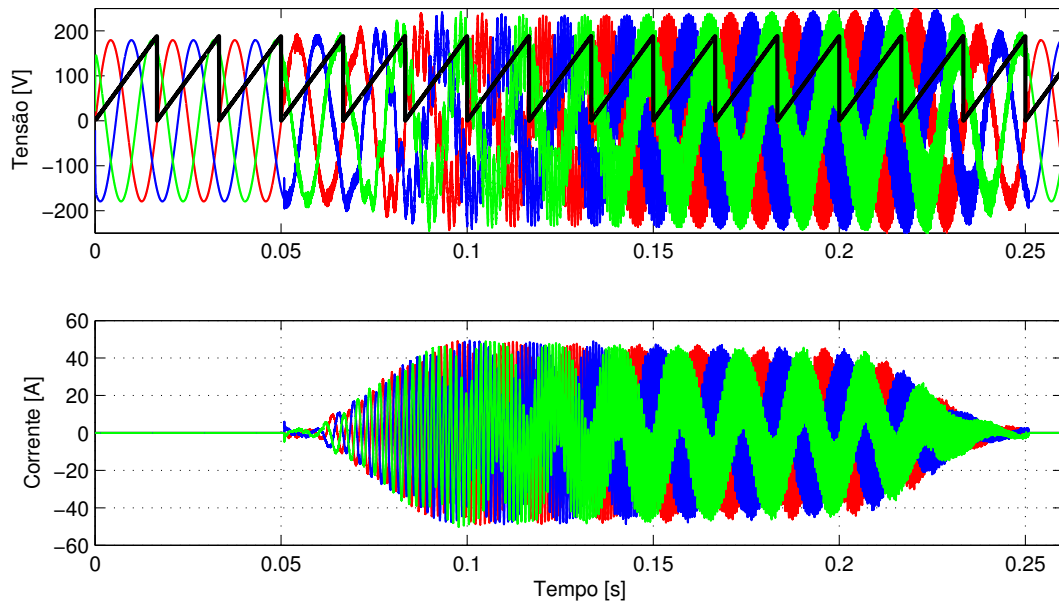


Figura 3.2: (a) Tensão trifásica filtrada na saída do conversor e algoritmo de PLL em funcionamento; (b) Corrente de referência em formato CHIRP sendo injetada na rede.

Na Figura 3.3, são apresentados três gráficos que mostram, respectivamente, a corrente no elo CC, a tensão no elo CC e a corrente medida na saída do conversor em uma das fases durante a simulação da rede de ordem 1 sem harmônicos de tensão. Foi utilizada uma corrente de referência de 1pu e com o elo inicialmente carregado também com 1,5pu de tensão DC. Note que à medida que se evolui a simulação, há um acréscimo da tensão no elo CC. Em termos práticos seria um problema, pois poderia elevar a tensão no capacitor do elo CC além das suas especificações. Esta elevação de tensão se deve à componente da fundamental de corrente estar ainda presente no controle de corrente causando uma injeção de potência da rede no conversor. Naturalmente, o controle de corrente utilizando o método de comparação direta não é o ideal, porém, atende ao que o trabalho se propõe em termos de injeção de corrente em formato CHIRP. A frequência de chaveamento do conversor utilizada foi de 100kHz, porém, uma das formas de contornar o problema de incremento de tensão no elo CC, seria aumentar a taxa de amostragem o que levaria a uma injeção de corrente mais próxima da referência sem a presença da fundamental. A solução definitiva é proposta como trabalho futuro. Vale reforçar que o conversor se propõe a injetar corrente no formato CHIRP sem a preocupação de transferir potência durante a execução do algoritmo.

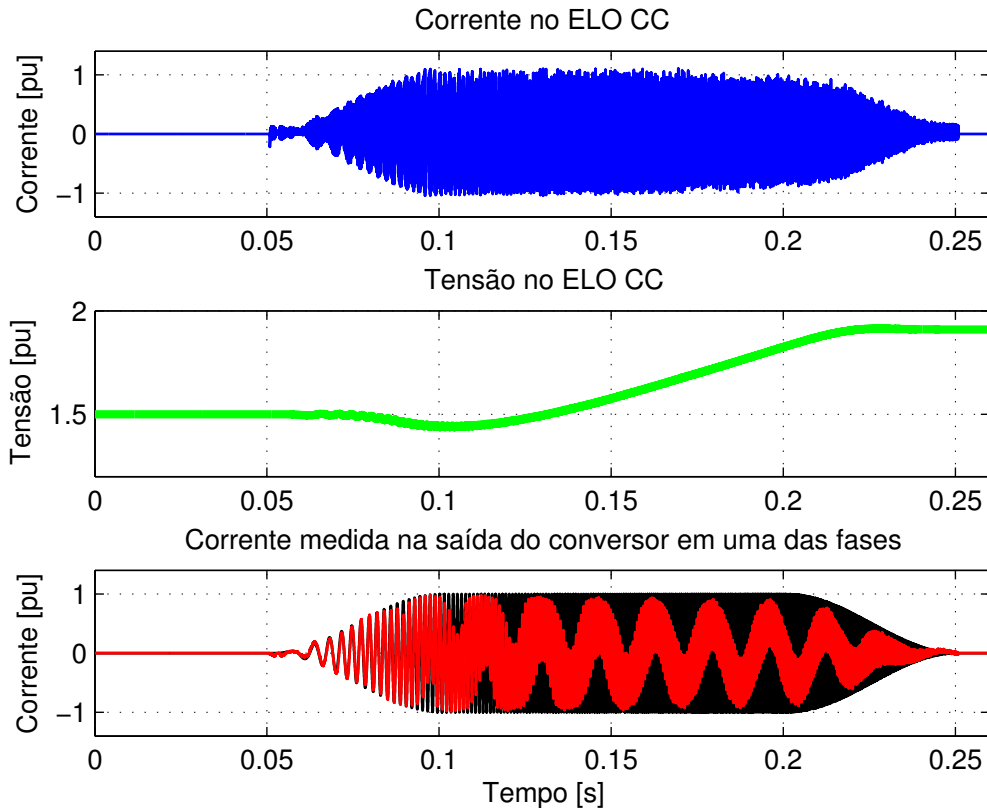


Figura 3.3: Análise de corrente e tensão no elo CC em comparativo com a corrente de saída em uma fases do conversor.

Em todas as simulações, assume-se que o sistema se encontra equilibrado, ou seja, a impedância vista por qualquer uma das fases é idêntica. Por isso, em todas as simulações, se considera a tensão e corrente em apenas uma das fases para o cálculo da impedância da rede. Outro ponto a ser considerado durante a simulação é a necessidade de utilização de um filtro na recepção do sinal de tensão da rede para filtrar os harmônicos gerados pelo próprio conversor durante o seu chaveamento. Para não prejudicar o método, o mesmo filtro foi aplicado na medição da corrente injetada pelo conversor na rede. Isto se fez necessário para que os atrasos introduzidos na medição pelo filtro não comprometam o cálculo dos coeficientes caso incidam em apenas em uma das variáveis medidas. Este filtro é do tipo passa-baixas de tal forma que elimine os ruídos de entrada e também não comprometa o sinal de corrente injetado.

Na Figura 3.4, são mostradas as correntes e as tensões trifásicas medidas e as filtradas com uma frequência de corte em 2,4kHz.

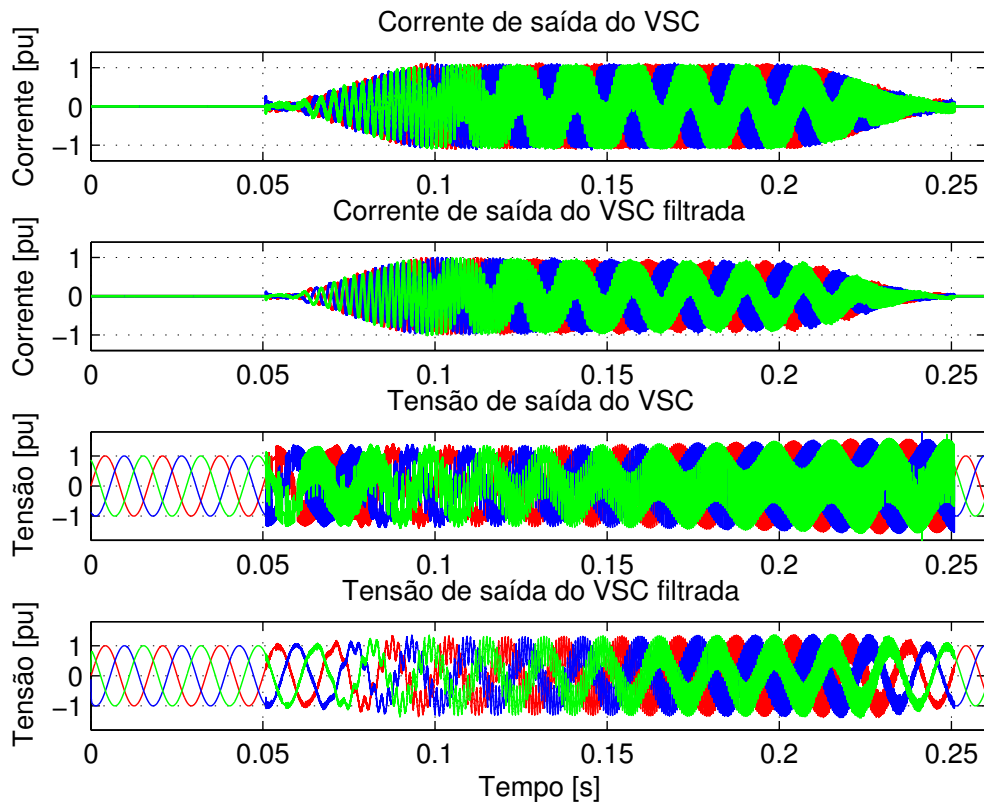


Figura 3.4: Correntes e Tensões trifásicas na saída do VSC.

Na Figura 3.5, são mostradas o sinal de referência CHIRP e a corrente de entrada no MCU. Note que a referência utilizada pelo VSC para injeção da corrente no formato CHIRP utilizando o método de comparação direta tem o seu formato ideal enquanto que a corrente de saída, medida em uma das fases da saída do conversor após passar pelo filtro de 2,4kHz, possui uma deterioração do seu sinal devido ao filtro e ao próprio método de geração de corrente.

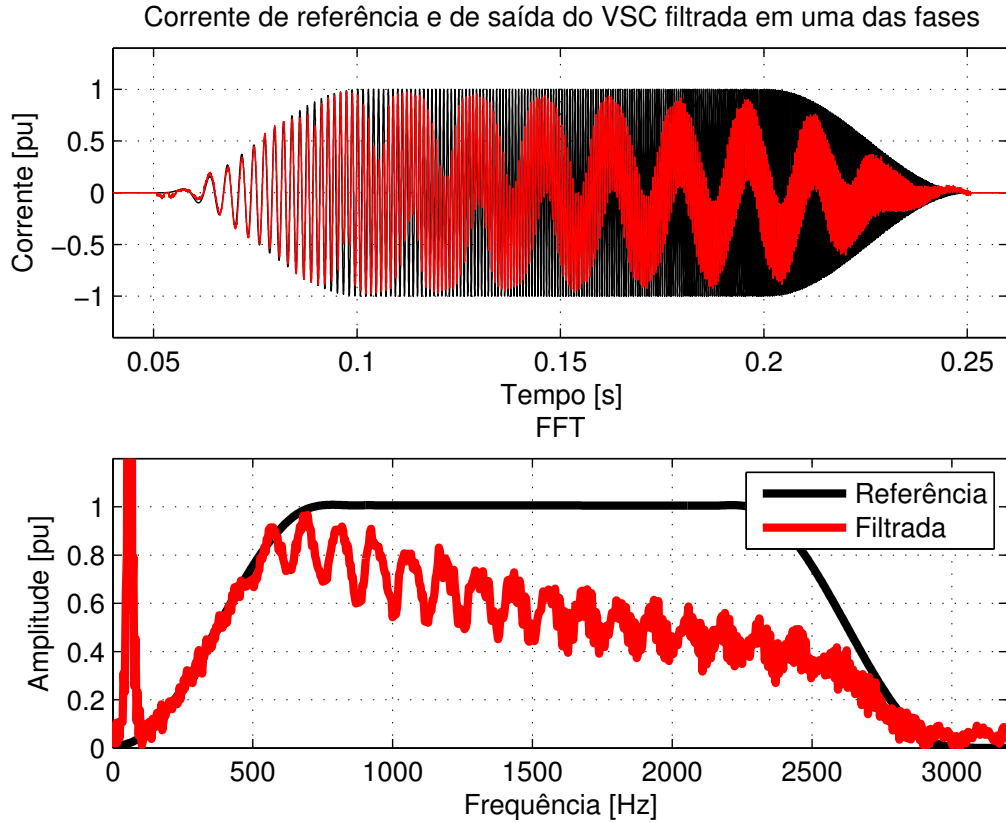


Figura 3.5: Comparativo entre o sinal de referência CHIRP e a corrente injetada pelo VSC medida em uma das fases e após ser filtrada.

Foram resumidos na Tabela 3.2 os parâmetros utilizados nas simulações.

Parâmetros	Notação	Valor
Instante inicial de aplicação do sinal	t_0	50ms
Instante final de aplicação do sinal	t_i	250ms
Tempo total simulado	T	200ms
Intervalo de tempo entre amostras	h	$50\mu\text{s}$
Frequência de Amostragem	f_a	20kHz
Frequência de Chaveamento do Conversor	f_{chav}	100kHz
Frequência inicial do pulso CHIRP	f_0	0Hz
Frequência final do pulso CHIRP	f_i	3kHz
Amplitude da corrente aplicada	I_{ref}	50A (1pu)
Frequência de corte do filtro	f_c	2,4kHz
Número Total de Amostras	N	4k

Tabela 3.2: Parâmetros das simulações.

O parâmetro instante inicial de aplicação do sinal é o instante em que o sinal

CHIRP começa a ser gerado pelo conversor e a ser injetado na rede. O instante final é o momento em que cessa a injeção do sinal CHIRP e o conversor retorna a sua operação normal de transferência de potência. O número total de amostras pode ser obtido multiplicando-se a taxa de amostragem utilizada na recepção dos sinais do VSC pelo tempo total de simulado.

Capítulo 4

Resultados e Discussões

Com o objetivo de testar o algoritmo recursivo num MCU, foram realizadas simulações num sistema *hardware-in-the-loop* onde o conversor VSC e a rede são simulados e o controle feito dentro de um MCU. Neste capítulo, são mostrados os resultados alcançados utilizando o algoritmo recursivo em redes com diferentes tipos de impedância. São elas:

1. Rede modelada de 1^a ordem.
2. Rede modelada de 1^a ordem com harmônicos.
3. Rede modelada de 2^a ordem.
4. Rede modelada de 4^a ordem.
5. Rede benchmark.

Na última seção deste capítulo, faz-se uma análise do tempo de processamento gasto pelo MCU durante as operações de atualização das matrizes recursivas intra-iterações.

Recordando que, em todas as simulações, foi aplicada uma corrente de referência no formato CHIRP numa janela Tukey. Este sinal foi gerado pelo MCU e transmitido ao conversor VSC para ser injetado na rede num período total de $200ms$, desde o instante $t = 50ms$ até o instante $t = 250ms$. Ao final da simulação, o MCU entrega imediatamente os valores dos coeficientes estimados da rede conforme algoritmo descrito anteriormente, ou seja, o processamento do MCU é realizado em *tempo real*.

4.1 Rede de 1^a ordem

Na primeira simulação, foi considerada uma impedância da rede de primeira ordem, caracterizada por uma carga *RL*. O circuito equivalente é mostrado na Figura 4.1.

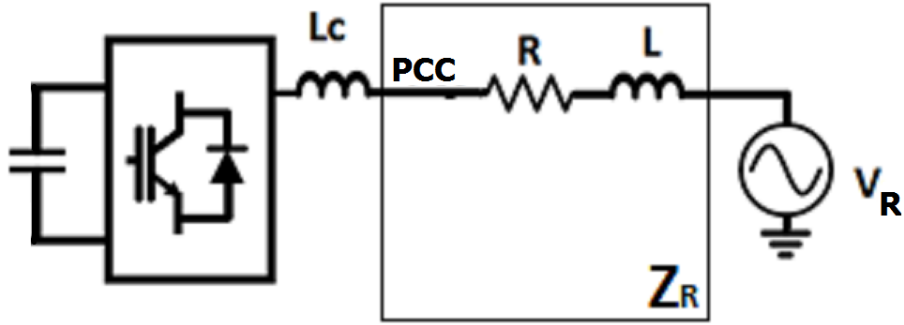


Figura 4.1: Rede com impedância modelada de 1ª ordem.

Em circuitos reais, a rede pode ser modelada como uma impedância com característica indutiva na frequência fundamental, como mostrado em [12]. Os valores de indutância podem variar desde algumas dezenas até centenas de microhenrys a depender do tipo de rede onde foi instalado o conversor. Para o exemplo que se segue, assume-se uma resistência de $R = 1\Omega$ e uma indutância de $L = 318\mu H$. Ou seja, a admitância desta rede de 1ª ordem deverá ter a F.T. dada pela equação (4.1) no sistema contínuo cuja frequência de corte fica localizada em torno de $500Hz$.

$$H(s) = \frac{1}{0,000318s + 1} \quad (4.1)$$

Como o MCU irá estimar a rede no seu modelo discreto, a equação (4.1) deverá ser discretizada. Considerando um passo de iteração fixo em $50\mu s$, de frequência de amostragem em $20kHz$, e utilizando o método de discretização do tipo ZOH , a rede (4.1) assume a F.T. no sistema discretizado mostrada na equação (4.2).

$$H_{REAL}(z) = \frac{0,1455}{z - 0,8545} \quad (4.2)$$

A Figura 4.2 mostra algumas curvas geradas durante a simulação quando há injeção do pulso CHIRP nesta rede de 1ª ordem. No primeiro gráfico, é mostrada a variação de tensão no elo CC durante a injeção do pulso CHIRP. No segundo e terceiro gráficos, são mostradas as correntes e as tensões trifásicas medidas na saída do conversor.

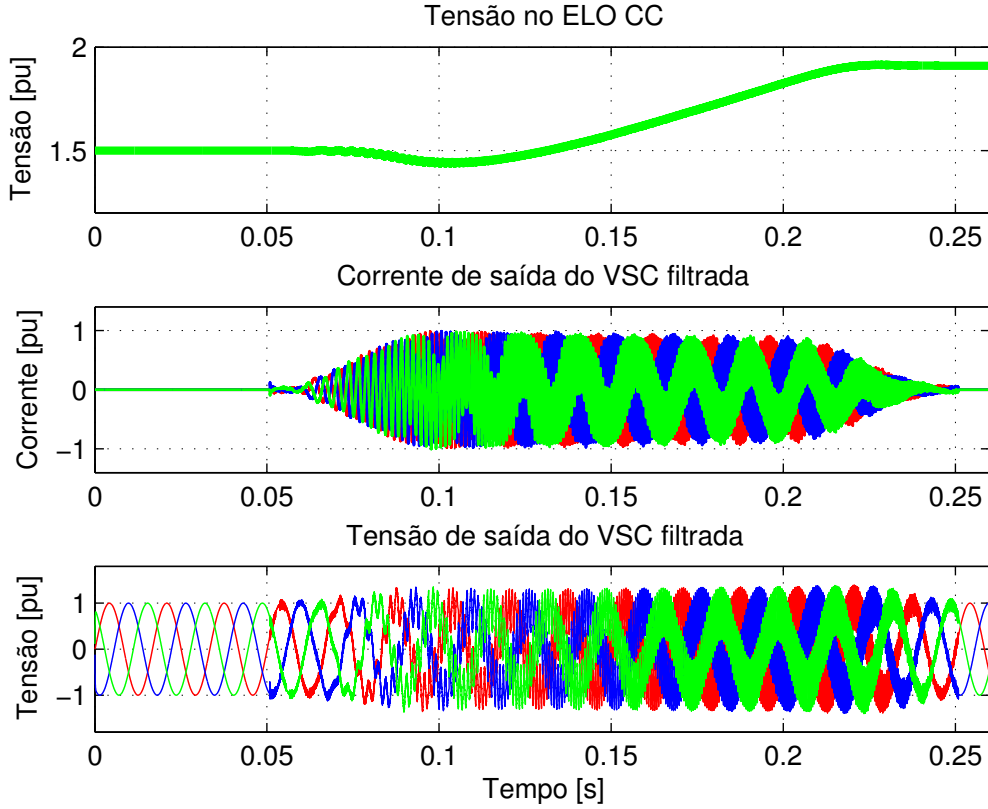


Figura 4.2: Gráfico das curvas da variação de tensão no elo CC, e das correntes e tensões numa rede de 1ª ordem.

A fim de comparação do método recursivo proposto com o método dos mínimos quadrados, primeiramente, foram coletados todos os dados da simulação, cujos parâmetros foram os descritos na Tabela 3.2, após isto, foram construídas as matrizes conforme a equação (2.13), sem considerar harmônicos na rede, e, por fim, foram calculados os coeficientes da rede utilizando a equação (2.14). Esta etapa é feita de forma *offline* utilizando o software *Matlab*® instalado num PC de 64 bits.

Para a construção das matrizes, cabe reforçar que, como a rede está sendo modelada de forma predominantemente indutiva na frequência fundamental, a função de transferência retrata não a impedância da rede e, sim, a admitância dela. Portanto, as medidas de saída a serem consideradas, aquelas posicionadas no vetor Y , serão as medidas de corrente; já as medidas de entrada, posicionadas na matriz A , serão as medidas de tensão.

Portanto, a matriz Y , contendo as medidas atuais, portará as medidas atuais de corrente que, neste caso, conforme demonstrado no capítulo 2, é um vetor coluna de dimensões $[(N - n + 1) \times 1]$. A matriz A , por sua vez, tem dimensões $[(N - n + 1) \times (2n + 2 + 2M)]$, e o vetor Θ tem dimensões $[(2n + 2 + 2M) \times 1]$.

Assim, para este caso específico, as variáveis terão os seguintes valores:

- $N = 4000$; pois a frequência de amostragem é de 20kHz num período de tomada de dados de 200ms;
- $n = 1$; ordem da rede considerada.
- $m = 0$; está sempre sendo considerado um valor igual a $n-1$; e
- $M = 0$; sem considerar harmônicos na rede neste exemplo;

Portanto, as dimensões das matrizes da equação (2.13) para este caso serão:

$$\begin{aligned} Y &= [4000 \times 1] \\ A &= [4000 \times 4] \\ \Theta &= [4 \times 1] \end{aligned}$$

A equação (2.13) poderá, então, ser reescrita substituindo os dados obtidos durante a simulação na equação (4.3) que se segue:

$$\begin{bmatrix} i_1 \\ \vdots \\ i_k \\ \vdots \\ i_{4000} \end{bmatrix} = \begin{bmatrix} -i_0 & v_1 & \cos(\theta_0) & \sin(\theta_0) \\ \vdots & \vdots & \vdots & \vdots \\ -i_{k-1} & v_k & \cos(\theta_k) & \sin(\theta_k) \\ \vdots & \vdots & \vdots & \vdots \\ -i_{4000-1} & v_{4000} & \cos(\theta_N) & \sin(\theta_N) \end{bmatrix} \begin{bmatrix} a_1 \\ b_0 \\ V_{C_0} \\ V_{S_0} \end{bmatrix} \quad (4.3)$$

Uma vez construídas as matrizes Y , A e Θ , obtém-se a resposta da rede na equação (4.4) através da solução de (2.14) que utiliza o método dos mínimos quadrados (MMQ) realizado de forma *offline*:

$$H_{MMQ}(z) = \frac{0,13}{z - 0,9276} \quad (4.4)$$

Para o caso do algoritmo recursivo, executado em tempo real dentro do MCU de 32 bits, o vetor Γ é construído a cada instante k , enquanto as matrizes Ξ e Φ são incrementadas também a cada iteração k utilizando atualizações das matrizes ξ e ϕ respectivamente. Assim, conforme já apresentado no capítulo 2, os vetores Γ , ϕ e Φ são de dimensão $(2n + 2 + 2M)$ enquanto as matrizes ξ e Ξ são quadradas de ordem $(2n + 2 + 2M)$. Para este exemplo, as matrizes recursivas terão as seguintes dimensões:

$$\Gamma = [4 \times 1]$$

$$\xi = [4 \times 4]$$

$$\Xi = [4 \times 4]$$

$$\phi = [4 \times 1]$$

$$\Phi = [4 \times 1]$$

Para o algoritmo recursivo, as matrizes devem ser construídas ou incrementadas a cada iteração. Segue o vetor Γ construído no instante k :

$$\Gamma(k) = \begin{bmatrix} -i_{k-1} & -v_k & \cos(\theta_k) & \sin(\theta_k) \end{bmatrix}^T \quad (4.5)$$

Como já dito, a matriz ξ é uma matriz quadrada resultante do produto do vetor Γ pelo seu transposto no instante k .

$$\xi(k) = \Gamma(k)\Gamma(k)^T \quad (4.6)$$

Substituindo o vetor Γ na equação, obtém-se:

$$\xi(k) = \begin{bmatrix} -i_{k-1} \\ -v_k \\ \cos(\theta_k) \\ \sin(\theta_k) \end{bmatrix} \begin{bmatrix} -i_{k-1} & -v_k & \cos(\theta_k) & \sin(\theta_k) \end{bmatrix} \quad (4.7)$$

Multiplicando os vetores acima chega-se à matriz ξ construída no instante k :

$$\xi(k) = \begin{bmatrix} i_{k-1}^2 & i_{k-1}v_k & -i_{k-1}\cos(\theta_k) & -i_{k-1}\sin(\theta_k) \\ i_{k-1}v_k & v_k^2 & -v_k\cos(\theta_k) & -v_k\sin(\theta_k) \\ -i_{k-1}\cos(\theta_k) & -v_k\cos(\theta_k) & \cos(\theta_k)^2 & \sin(\theta_k)^2 \\ -i_{k-1}\sin(\theta_k) & -v_k\sin(\theta_k) & \sin(\theta_k)\cos(\theta_k) & \sin(\theta_k)^2 \end{bmatrix} \quad (4.8)$$

A matriz ξ servirá para incrementar a matriz Ξ uma vez que a matriz Ξ é um somatório das matrizes ξ geradas em cada iteração:

$$\Xi = \sum_{k=1}^{4000} \xi(k) \quad (4.9)$$

O vetor ϕ também é modificado a cada iteração k . O vetor ϕ é um vetor coluna resultante do produto escalar entre o vetor coluna Γ e o valor de saída amostrado no instante k , y_k . Segue o seu desenvolvimento:

$$\phi(k) = \Gamma(k)y_k \quad (4.10)$$

Substituindo as variáveis acima, obtemos:

$$\phi(k) = \begin{bmatrix} -i_{k-1} & -v_k & \cos(\theta_k) & \sin(\theta_k) \end{bmatrix}^T i_k \quad (4.11)$$

Multiplicando os termos, chega-se ao vetor ϕ obtido no instante k:

$$\phi(k) = \begin{bmatrix} -i_k i_{k-1} & -i_k v_k & i_k \cos(\theta_k) & i_k \sin(\theta_k) \end{bmatrix}^T \quad (4.12)$$

O vetor ϕ servirá para incrementar o vetor Φ , uma vez que o vetor Φ é um somatório dos vetores ϕ gerados em cada iteração:

$$\Phi = \sum_{k=1}^{4000} \phi(k) \quad (4.13)$$

Dessa forma, a cada iteração, foram tomados os dados de medição, incrementadas as matrizes e vetores e, no fim da última iteração, foi executado o cálculo obtendo a estimativa dos coeficientes de impedância da rede. A resposta foi obtida através da solução da equação (2.31) que utiliza o algoritmo recursivo para estimar os parâmetros da rede em tempo real. A solução final é sempre encontrada no fim do algoritmo, no fim da última iteração, quando é calculada a matriz Θ contendo os coeficientes da rede conforme mostra a equação (2.31).

Deste resultado, foi obtida a F.T. (4.14) mostrada na equação (4.4). Ou seja, o método alcança o mesmo resultado porém de forma recursiva exigindo menos memória de processamento já que o resultado é construído a medida que avançam as iterações.

$$H_{REC}(z) = \frac{0,1354}{z - 0,9257} \quad (4.14)$$

Na Tabela 4.1, foram mostrados os coeficientes obtidos pelo algoritmo recursivo e sua comparação com os coeficientes reais. Note que os coeficientes obtidos pelo método recursivo estão com menos de 10% de erro se comparados com os teóricos.

Coefficientes	Reais	MMQ	Erro	Recursivo	Erro
b_0	0,1455	0,13	10,6%	0,1354	6,9%
a_1	-0,8545	-0,9276	8,5%	-0,9257	8,3%

Tabela 4.1: Comparativo entre os coeficientes obtidos pelo algoritmo recursivo com os coeficientes teóricos.

As curvas das respostas em frequência obtidas pelo método dos mínimos quadrados e pelo algoritmo recursivo foram exibidas num único gráfico como mostra a Figura 4.3. Destaca-se que a resposta obtida utilizando o método dos mínimos

quadrados é praticamente idêntica à resposta real da rede em baixas frequências enquanto a resposta do método recursivo se aproxima da real em altas frequências.

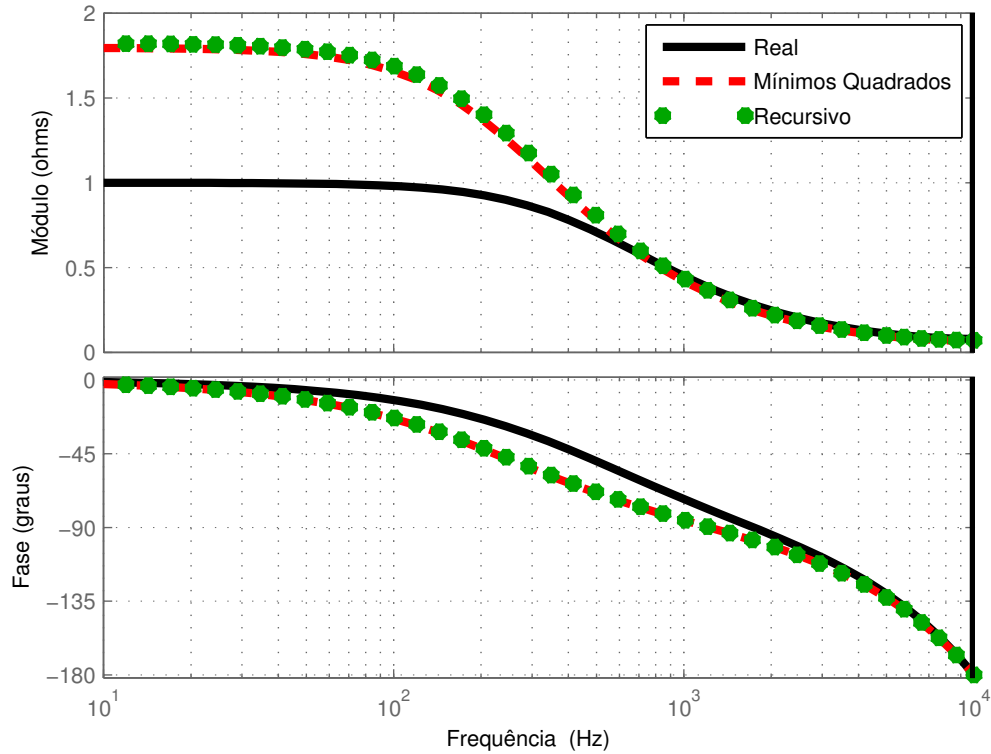


Figura 4.3: Gráfico das respostas em frequência da admitância da rede de 1ª ordem obtidas pelo método dos mínimos quadrados e pelo método recursivo.

A Figura 4.4 mostra as respostas em frequência obtidas pelo MCU para o mesmo circuito de 1ª ordem, porém definindo, antes do início da simulação, diferentes valores de n . Foram simulados valores de n igual a 2 e n igual a 6. Ou seja, no sistema de equações, foram definidos no algoritmo valores superiores de ordem da rede.

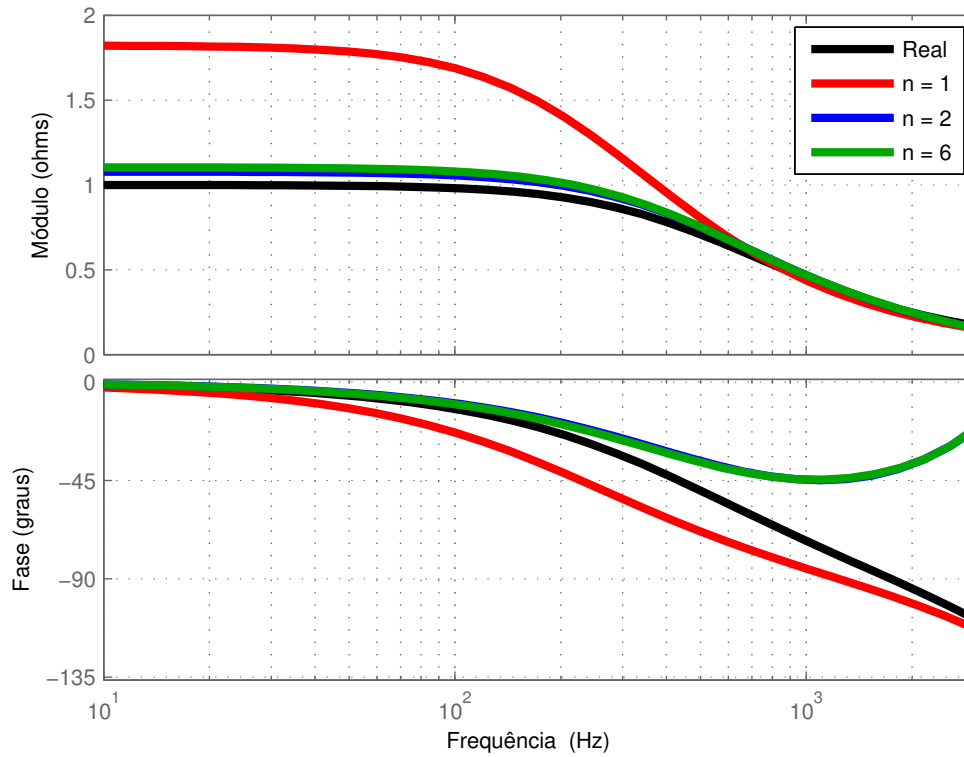


Figura 4.4: Gráfico das curvas real e calculada pelo método com impedância da rede de 1ª ordem para diferentes valores de n .

Observando o gráfico da Figura 4.4, nota-se que, quando se aumenta a ordem das matrizes recursivas, a curva calculada se aproxima da real em baixas frequências porém se afasta nas altas. Entretanto, entre a frequência de 1º harmônico e 10º harmônico, o erro é pequeno conforme demonstra o gráfico da Figura 4.5.

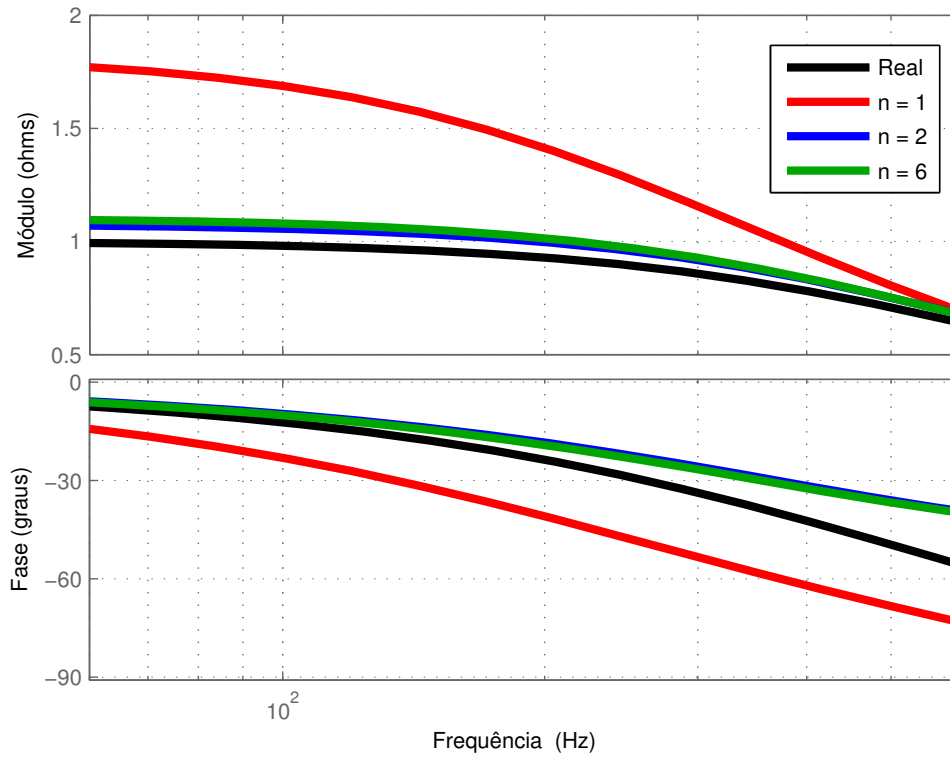


Figura 4.5: Gráfico das curvas real e calculada pelo método com impedância da rede de 1ª ordem para diferentes valores de n ampliado entre as frequências de 1º e 10º harmônicos.

Por fim, para demonstrar que os coeficientes são obtidos à medida que evoluem as iterações, é mostrado, no gráfico da Figura 4.6, o erro normalizado entre os coeficientes reais da impedância da rede e os coeficientes obtidos pelo algoritmo recursivo a cada iteração. O erro tende a estabilizar ao fim da simulação e mostra que o coeficiente b_0 tende ao valor real, porém, o erro aumenta para a_1 , mas ambos estabilizam ao fim do experimento.

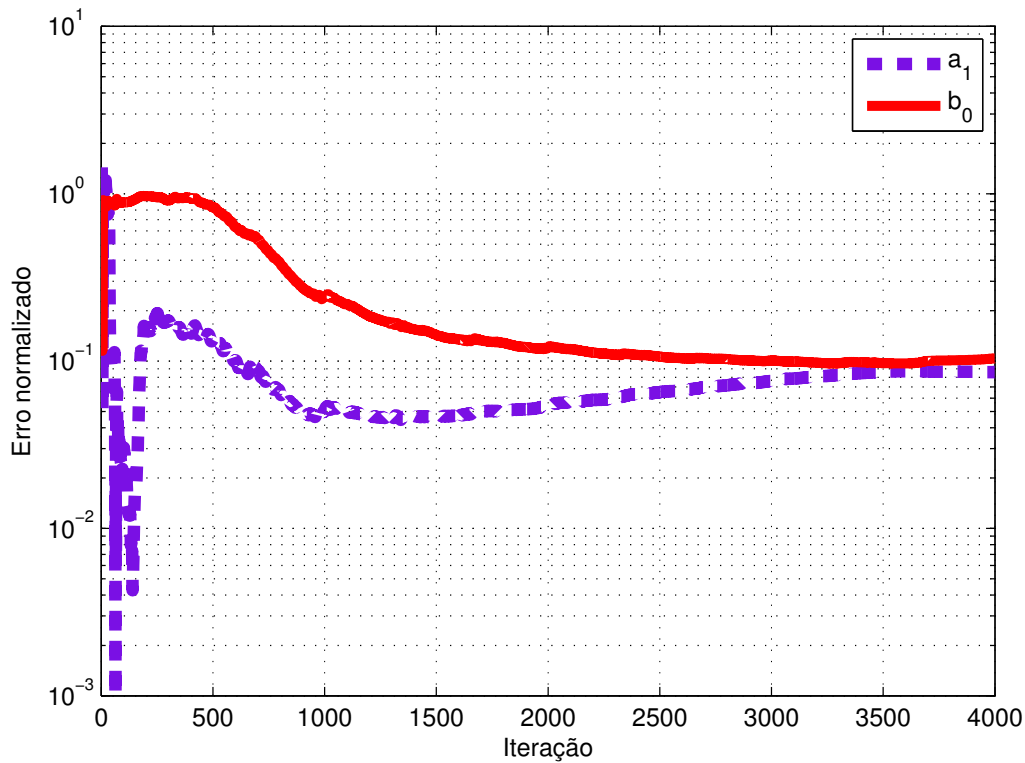


Figura 4.6: Gráfico mostrando o erro normalizado entre os coeficientes reais e aqueles obtidos a cada iteração.

4.2 Rede de 1^a ordem com harmônicos na tensão

Na segunda simulação, foi testada a mesma rede de 1^a ordem da seção anterior, porém, com harmônicos na tensão. Ela contém componentes de 5^o, 7^o, 11^o e 13^o harmônicos, com percentual harmônico de 7,5%, 6,5%, 4,5% e 4% respectivamente totalizando um THD de 11,6%.

A Figura 4.7 mostra as curvas de tensão no elo CC, das correntes trifásicas filtradas na saída do conversor e das tensões trifásicas também filtradas medidas na saída do conversor durante a injeção do pulso CHIRP na rede. Note que, diferentemente do gráfico de tensão de saída do conversor da Figura 4.2, as tensões trifásicas mostram alguma distorção devido ao acréscimo dos harmônicos.

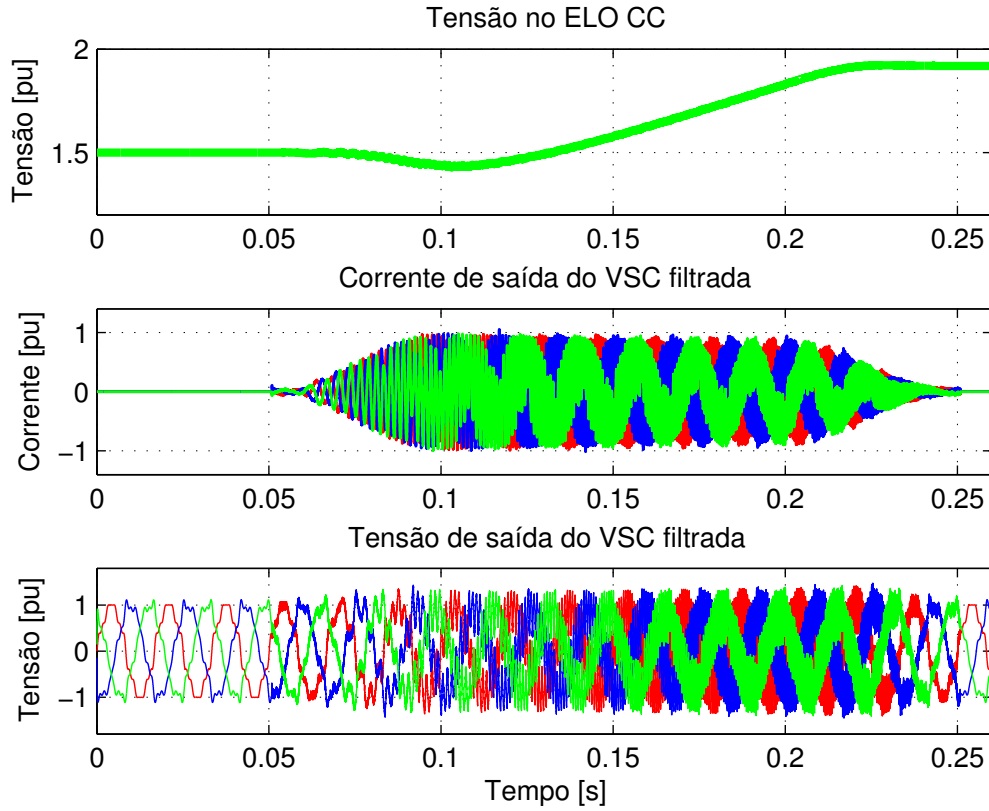


Figura 4.7: Gráfico das curvas de variação de tensão no elo CC e da corrente e tensão da rede contendo harmônicos de tensão com THD em 11,6%.

Considerando que M será igual a 4 pois a rede neste caso contém 4 harmônicos de tensão, as matrizes recursivas neste caso terão as seguintes dimensões:

$$\Gamma = [12 \times 1]$$

$$\xi = [12 \times 12]$$

$$\Xi = [12 \times 12]$$

$$\phi = [12 \times 1]$$

$$\Phi = [12 \times 1]$$

Deste resultado, foi obtida a F.T. (4.15) quando são considerados todos os harmônicos nos cálculos. Nota-se que os coeficientes se posicionam bem próximo daqueles mostrados na equação real (4.2).

$$H(z) = \frac{0,1352}{z - 0,9253} \quad (4.15)$$

Na Tabela 4.2, foram mostrados os coeficientes obtidos pelo algoritmo recursivo e sua comparação com os coeficientes obtidos teoricamente. Note que os coeficientes obtidos pelo método recursivo estão com menos de 10% de erro se comparados com os teóricos.

Coeficientes	Reais	Recursivo	Erro
b_0	0,1455	0,1352	7,1%
a_1	-0,8545	-0,9253	8,2%

Tabela 4.2: Comparativo entre os coeficientes obtidos pelo algoritmo recursivo com os coeficientes teóricos numa rede de 1ª ordem com harmônicos de tensão.

Na Figura 4.8, foram geradas cinco curvas de resposta em frequência obtidas pelo algoritmo recursivo considerando diferentes valores de M . Note que à medida que se adicionam harmônicos nos cálculos há uma aproximação da resposta obtida à real. Ou seja, está sendo demonstrado que é relevante o valor de M nos cálculos dos coeficientes quando há harmônicos de tensão na rede.

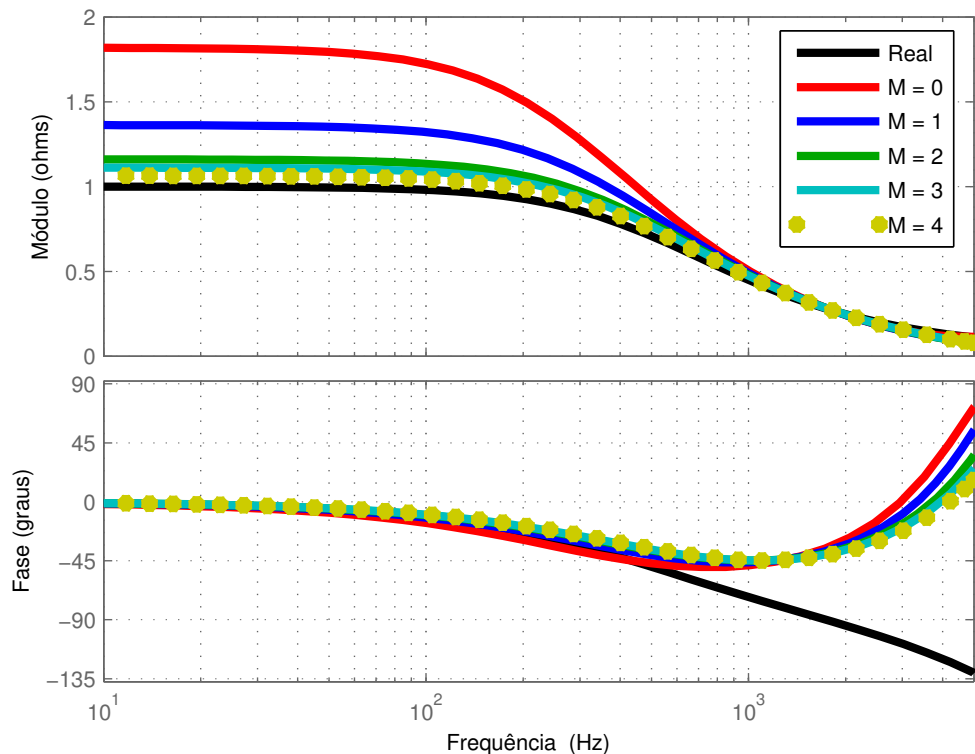


Figura 4.8: Resposta do método considerando uma impedância RL e rede contendo harmônicos de tensão.

Note que o algoritmo foi capaz de anular o efeito dos harmônicos da tensão mesmo

com elevada distorção na tensão. Lembrando que a anulação dos harmônicos é feita acrescentando-se duas colunas no vetor Γ , uma em seno e outra em cosseno, para cada harmônico de tensão.

4.3 Rede de 2ª ordem

Na terceira simulação, a rede foi modelada como sendo uma carga *RLC*. Neste caso, a rede pode ser entendida como sendo indutiva, como no primeiro caso, porém, conectada a um banco de capacitores para correção de fator de potência. Com isto, adiciona-se um pólo à função de transferência e um ponto de ressonância para checar a capacidade do algoritmo em identificá-lo. Assume-se, então, que a rede pode ser representada pelo circuito equivalente mostrado na Figura 4.9.

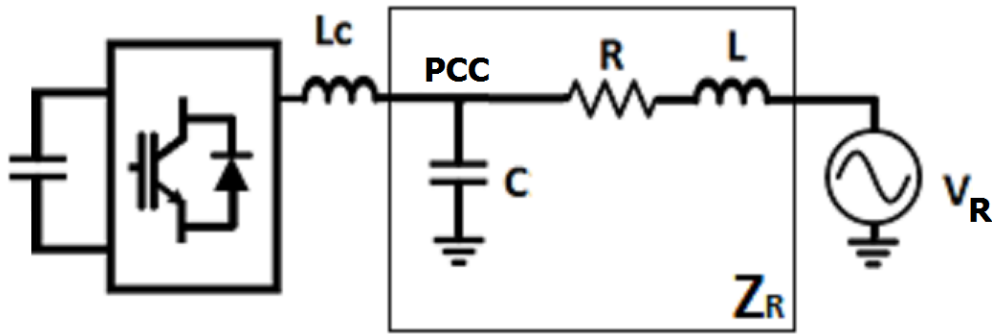


Figura 4.9: Impedância da Rede modelada de 2ª Ordem.

Considerando os valores de $R=1\Omega$, $L=451\mu\text{H}$ e $C=50\mu\text{F}$, sua F.T. é dada pela equação (4.16) contendo uma frequência de ressonância aproximadamente em 1kHz .

$$H(s) = \frac{20000s + 4,435 \cdot 10^7}{s^2 + 2217s + 4,435 \cdot 10^7} \quad (4.16)$$

Da mesma forma como foi feito na rede de 1ª ordem, como o MCU irá estimar a rede no seu modelo discreto, a equação (4.16) deverá ser discretizada. Considerando também um passo de iteração fixo em $50\mu\text{s}$, pois a frequência de amostragem está em 20kHz , e utilizando o método de discretização do tipo *ZOH*, a rede (4.16) assume a F.T. no sistema discretizado mostrada na equação (4.17).

$$H_{REAL}(z) = \frac{0,9821z - 0,8781}{z^2 - 1,791z + 0,8951} \quad (4.17)$$

A Figura 4.10 mostra algumas curvas geradas durante a simulação quando há injeção do pulso CHIRP na rede de 2ª ordem. No primeiro gráfico, é mostrada a variação de tensão no elo CC durante a injeção do pulso CHIRP. No segundo e

terceiro gráficos, são mostradas as correntes e as tensões trifásicas medidas na saída do conversor.

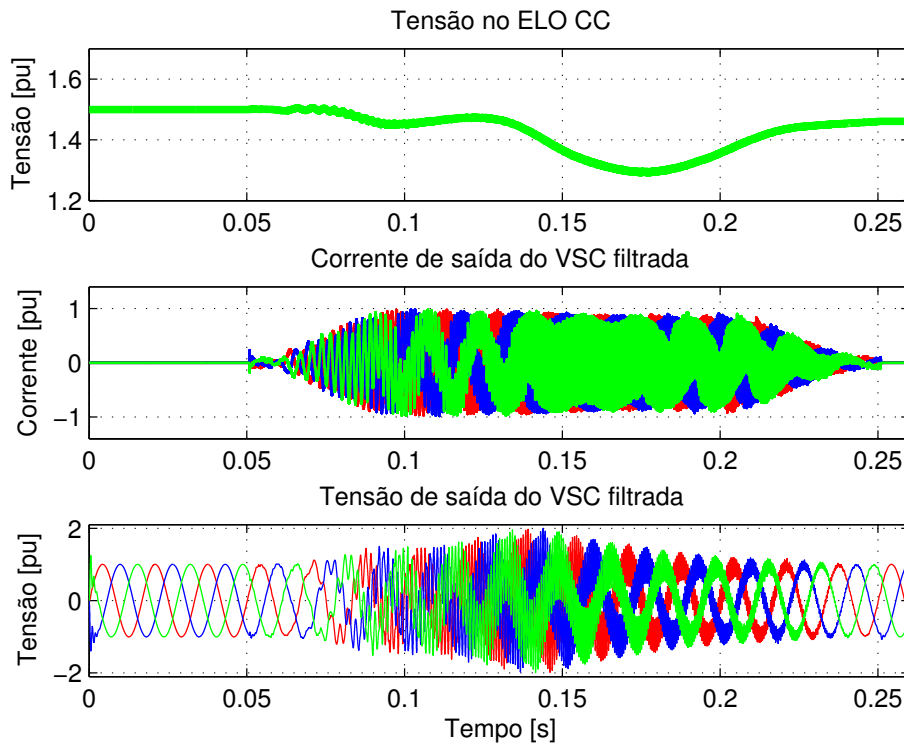


Figura 4.10: Gráfico das curvas da variação de tensão no elo CC, e das correntes e tensões numa rede de 2ª ordem.

Para esta simulação numa rede de 2ª ordem sem harmônicos, foram utilizados os mesmos parâmetros da Tabela 3.2. Da mesma forma como foi realizado na rede de 1ª ordem, a fim de comparação do método dos mínimos quadrados com o método recursivo proposto, primeiramente, foram coletados todos os dados da simulação. Após isto, foram construídas as matrizes conforme a equação (2.13) e, por fim, foram calculados os coeficientes da rede utilizando a equação (2.14) referente à solução pelo método dos mínimos quadrados.

Neste caso, a função de transferência retrata a impedância da rede. Portanto, as medidas de saída a serem consideradas, aquelas posicionadas no vetor Y , serão as medidas de tensão; já as medidas de entrada, posicionadas na matriz A , serão as medidas de corrente.

Assim, as dimensões das matrizes da equação (2.13), considerando que a ordem do sistema de equações é igual a 2 ($n = 2$) e que a rede está sem harmônicos ($M = 0$), serão:

$$Y = [3999 \times 1]$$

$$A = [3999 \times 6]$$

$$\Theta = [6 \times 1]$$

Uma vez construídas as matrizes Y , A e Θ , obtém-se a resposta da rede (4.18) através da solução da equação (2.14):

$$H_{MMQ}(z) = \frac{0,9119z - 0,7313}{z^2 - 1,802z + 0,9117} \quad (4.18)$$

Analisando as matrizes do algoritmo recursivo, as dimensões das matrizes recursivas Γ , Ξ e Φ para esta rede de 2^a ordem serão:

$$\Gamma = [6 \times 1]$$

$$\xi = [6 \times 6]$$

$$\Xi = [6 \times 6]$$

$$\phi = [6 \times 1]$$

$$\Phi = [6 \times 1]$$

Lembrando que, no algoritmo recursivo, as matrizes Γ , ξ e ϕ devem ser construídas e as matrizes Ξ e Φ devem ser incrementadas a cada iteração. Segue o vetor Γ construído no instante k :

$$\Gamma(k) = \left[-v_{k-1} \quad -v_{k-2} \quad i_k \quad i_{k-1} \quad \cos(\theta_k^0) \quad \sin(\theta_k^0) \right]^T \quad (4.19)$$

A matriz ξ será construída a partir do vetor Γ :

$$\xi(k) = \begin{bmatrix} -v_{k-1} \\ -v_{k-2} \\ i_k \\ i_{k-1} \\ \cos(\theta_k^0) \\ \sin(\theta_k^0) \end{bmatrix} \begin{bmatrix} -v_{k-1} & -v_{k-2} & i_k & i_{k-1} & \cos(\theta_k^0) & \sin(\theta_k^0) \end{bmatrix} \quad (4.20)$$

Multiplicando os vetores acima chega-se à matriz ξ construída no instante k :

$$\xi(k) = \begin{bmatrix} v_{k-1}^2 & v_{k-1}v_{k-2} & -v_{k-1}i_k & -v_{k-1}i_{k-1} & -v_{k-1}\cos(\theta_k^0) & -v_{k-1}\sin(\theta_k^0) \\ v_{k-2}v_{k-1} & v_{k-2}^2 & -v_{k-2}i_k & -v_{k-2}i_{k-1} & -v_{k-2}\cos(\theta_k^0) & -v_{k-2}\sin(\theta_k^0) \\ -i_kv_{k-1} & -i_kv_{k-2} & i_k^2 & i_ki_{k-1} & i_k\cos(\theta_k^0) & i_k\sin(\theta_k^0) \\ -i_{k-1}v_{k-1} & -i_{k-1}v_{k-2} & i_{k-1}i_k & i_{k-1}^2 & i_{k-1}\cos(\theta_k^0) & i_{k-1}\sin(\theta_k^0) \\ -v_{k-1}\cos(\theta_k^0) & -v_{k-2}\cos(\theta_k^0) & i_k\cos(\theta_k^0) & i_{k-1}\cos(\theta_k^0) & \cos^2(\theta_k^0) & \sin(\theta_k^0)\cos(\theta_k^0) \\ -v_{k-1}\sin(\theta_k^0) & -v_{k-2}\sin(\theta_k^0) & i_k\sin(\theta_k^0) & i_{k-1}\sin(\theta_k^0) & \sin(\theta_k^0)\cos(\theta_k^0) & \sin^2(\theta_k^0) \end{bmatrix} \quad (4.21)$$

A matriz ξ servirá para incrementar a matriz Ξ uma vez que a matriz Ξ é um somatório das matrizes ξ geradas em cada iteração:

$$\Xi = \sum_{k=2}^{4000} \xi(k) \quad (4.22)$$

O vetor ϕ também é modificado a cada iteração k . Ele é um vetor coluna resultante do produto escalar entre o vetor coluna Γ e o valor de saída y_k amostrado no instante k que neste caso é o valor de tensão v_k

$$\phi(k) = \left[-v_k v_{k-1} \quad -v_k v_{k-2} \quad v_k i_k \quad v_k i_{k-1} \quad v_k \cos(\theta_k^0) \quad v_k \sin(\theta_k^0) \right]^T \quad (4.23)$$

O vetor ϕ servirá para incrementar o vetor Φ , uma vez que este é um somatório dos vetores ϕ gerados em cada iteração:

$$\Phi = \sum_{k=2}^{4000} \phi(k) \quad (4.24)$$

Da mesma forma como foi realizada na rede de 1ª ordem, a resposta foi obtida através da solução da equação (2.31) que utiliza o algoritmo recursivo dentro do MCU. A solução final é sempre encontrada no fim do algoritmo, no fim da última iteração, quando é calculada a matriz Θ com os coeficientes utilizando a equação (2.31).

Deste resultado, foi obtida a F.T. (4.25) mostrada na equação (4.4):

$$H_{REC}(z) = \frac{0,8931z - 0,7052}{z^2 - 1,795z + 0,9094} \quad (4.25)$$

Na Tabela 4.3, foram agrupados os coeficientes obtidos pelo método recursivo no MCU, aqueles gerados pelo circuito real e os erros dos coeficientes do método recursivo com relação aos teóricos.

Coeficientes	Reais	MMQ	Erro	Recursivo	Erro
b_0	0,9821	0,9119	7,14%	0,8931	9,1%
b_1	-0,8781	-0,7313	16,7%	-0,7052	19,7%
a_1	-1,791	-1,802	0,61%	-1,795	0,22%
a_2	0,8951	0,9177	2,5%	0,9094	1,6%

Tabela 4.3: Comparativo entre os coeficientes obtidos pelo algoritmo recursivo com os coeficientes teóricos.

As curvas das respostas em frequência obtidas pelo método recursivo e pelo método dos mínimos quadrados foram exibidas num único gráfico como mostra a

Figura 4.11. Nota-se que há uma aproximação maior da curva calculada pelo método dos mínimos quadrados à curva real se comparada com a do método recursivo. Isto se deve ao fato do método dos mínimos quadrados ser calculado por um PC de 64 bits o que lhe garante mais algarismos significativos e, conseqüentemente, uma melhor resolução dos coeficientes.

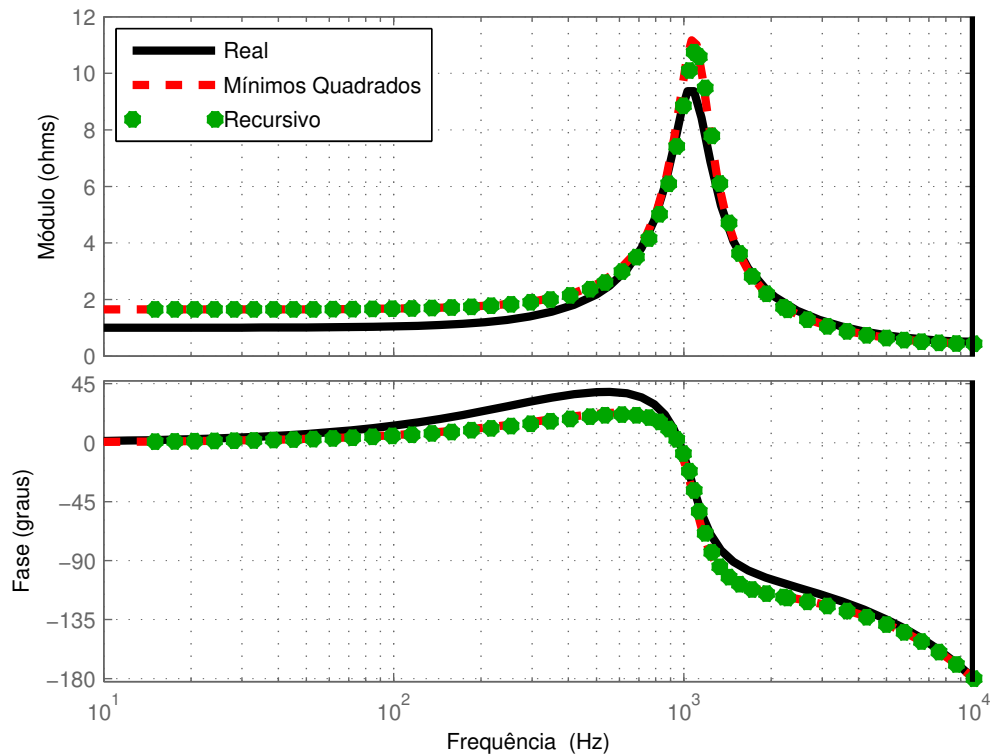


Figura 4.11: Gráfico das respostas em frequência da impedância de uma rede de 2ª ordem obtidas pelo método dos mínimos quadrados, pelo método recursivo e a gerada pelo circuito real.

O gráfico da Figura 4.11 e a Tabela 4.3 demonstram que o algoritmo foi capaz de calcular os coeficientes da F.T. da rede de 2ª ordem.

No gráfico da Figura 4.12 foram geradas as respostas em frequência da rede de 2ª ordem empregando diferentes valores de n no sistema de equações, ou seja, definindo previamente valores de n igual a 2, 4 e 6.

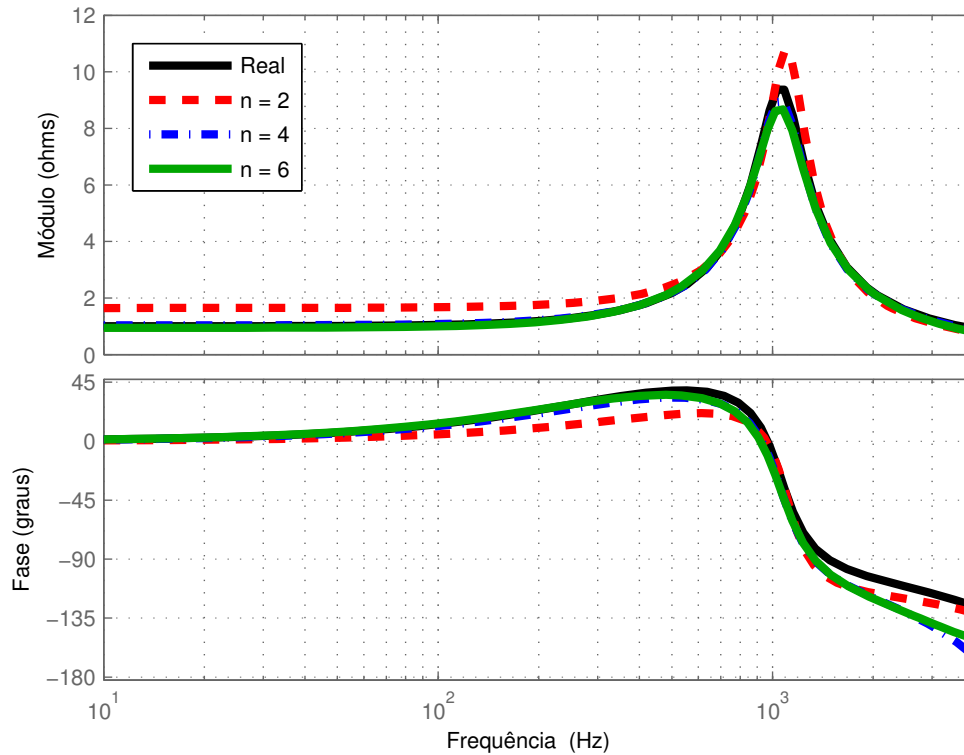


Figura 4.12: Gráfico das curvas real e calculada pelo método com impedância da rede de 2ª ordem para diferentes valores de n .

Observando o gráfico da Figura 4.12, observa-se que da mesma forma como ocorreu na rede de 1ª ordem quando se aumentou a ordem do sistema, a curva calculada se aproximou da real à medida que se aumentou o valor de n .

4.4 Rede de 4ª ordem

Na quarta simulação, o algoritmo foi testado numa rede de 4ª ordem colocando um filtro LCL na saída do conversor e uma impedância RCL em série conforme mostra o circuito equivalente da Figura 4.13.

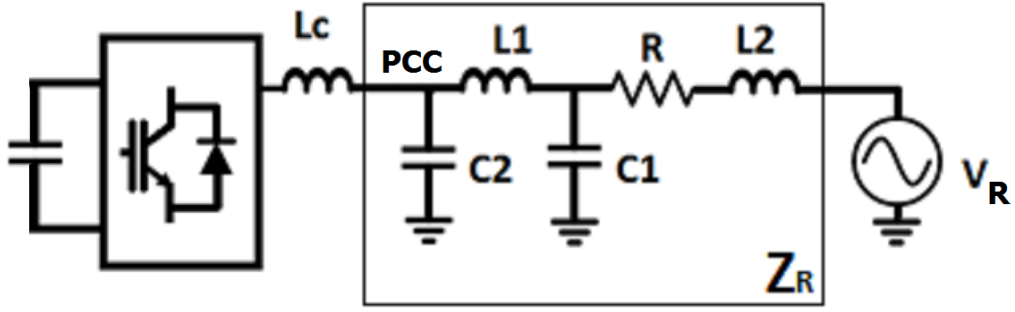


Figura 4.13: Impedância de 4ª ordem.

Considerando os valores de $R=1\Omega$, $L1=4\text{mH}$, $C1=10\mu\text{F}$, $L2=100\mu\text{H}$ e $C2=500\mu\text{F}$ sua F.T. é dada pela equação (4.26) com frequência de ressonância aproximada de 100Hz .

$$H(s) = \frac{0,005s^3 + 50s^2 + 5,1 \cdot 10^6 s + 10^9}{2,5 \cdot 10^{-6} s^4 + 0,025s^3 + 2551s^2 + 510000s + 10^9} \quad (4.26)$$

Da mesma forma como foi feito nos exemplos anteriores, como o MCU irá estimar a rede no seu modelo discreto, a equação (4.26) deverá ser discretizada. Considerando o passo de iteração fixo em $50\mu\text{s}$, pois a frequência de amostragem está em 20kHz , e utilizando o método de discretização do tipo *ZOH*, a rede (4.26) assume a F.T. no sistema discretizado mostrada na equação (4.27).

$$H_{REAL}(z) = \frac{0,09998z^3 - 0,09812z^2 + 0,06037z - 0,06064}{z^4 - 1,98z^3 + 1,585z^2 - 1,21z + 0,6065} \quad (4.27)$$

A Figura 4.14 mostra algumas curvas geradas durante a simulação quando há injeção do pulso CHIRP na rede de 4ª ordem. No primeiro gráfico, é mostrada a variação de tensão no elo CC durante a injeção do pulso CHIRP. No segundo e terceiro gráficos, são mostradas as correntes e as tensões trifásicas medidas na saída do conversor.

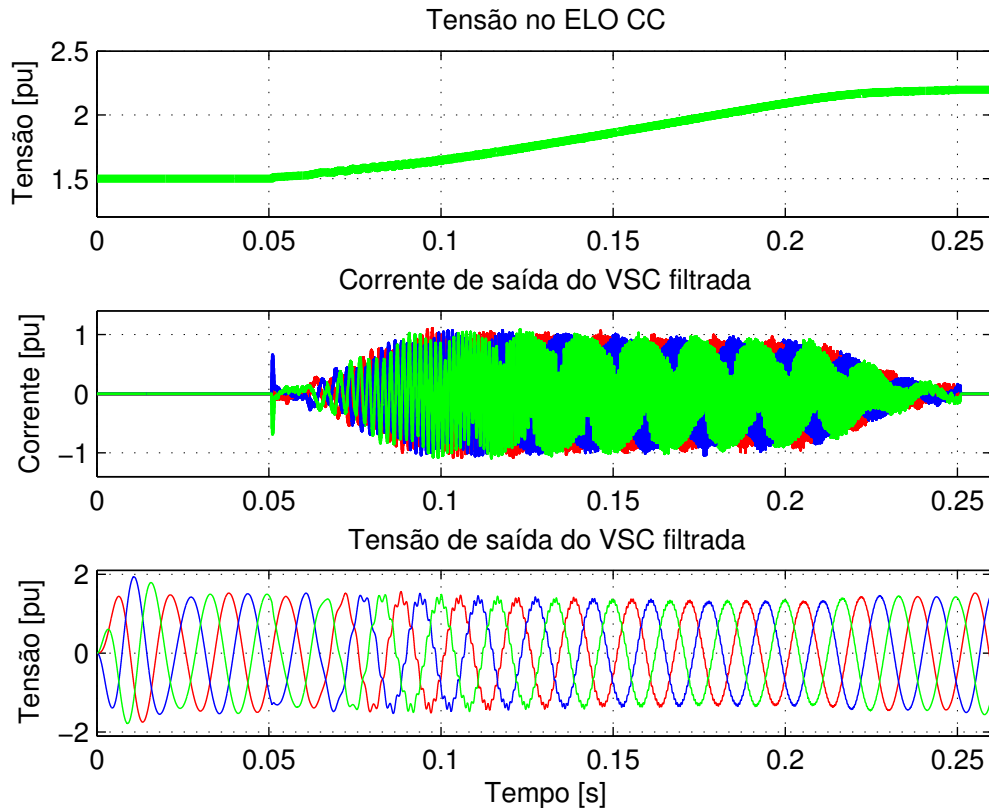


Figura 4.14: Gráfico das curvas da variação de tensão no elo CC, e das correntes e tensões numa rede de 4ª ordem.

As matrizes recursivas, considerando a ordem do sistema de equações igual a 4 ($n = 4$) e desconsiderando os harmônicos de tensão da rede ($M = 0$), terão as seguintes dimensões:

$$\Gamma = [10 \times 1]$$

$$\xi = [10 \times 10]$$

$$\Xi = [10 \times 10]$$

$$\phi = [10 \times 1]$$

$$\Phi = [10 \times 1]$$

Após obter os coeficientes pelo método recursivo e calcular os coeficientes pelo método dos mínimos quadrados são geradas as respostas em frequência conforme mostra a Figura 4.15. Nota-se que há uma maior aproximação da curva de resposta em frequência obtida pelo método dos mínimos quadrados da obtida teoricamente.

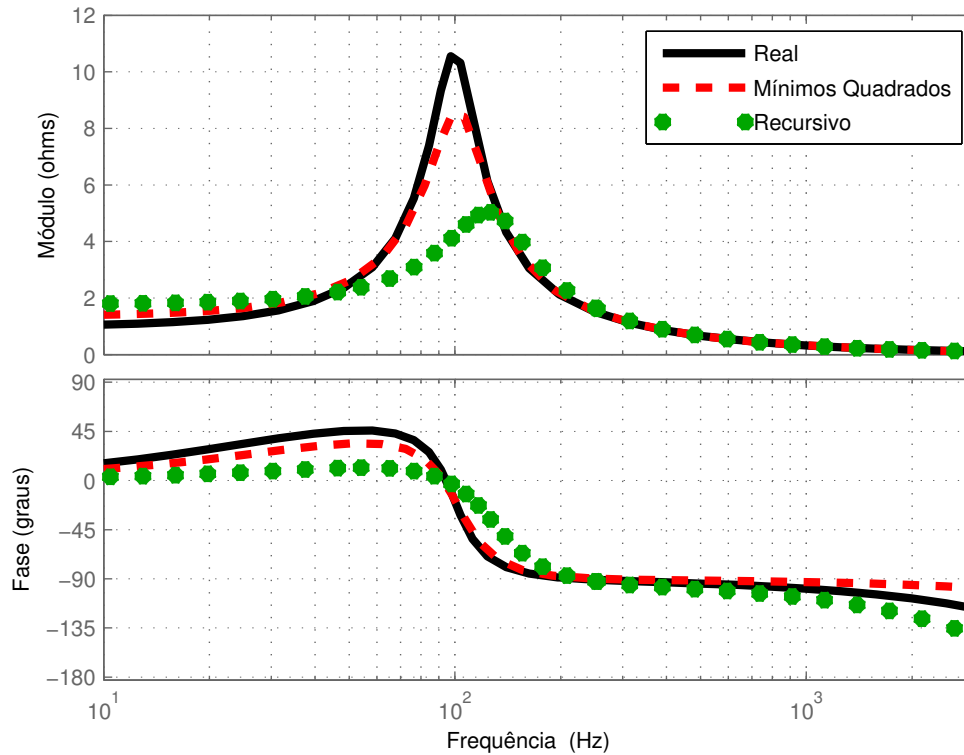


Figura 4.15: Gráfico das respostas em frequência da impedância de uma rede de 4ª ordem obtidas pelo método dos mínimos quadrados, pelo método recursivo e a gerada pelo circuito real.

Neste exemplo, já foi observado um distanciamento maior da curva obtida pelo circuito real da obtida pelo algoritmo recursivo. A frequência de ressonância em 100 Hz não foi encontrada pelo método recursivo. O algoritmo recursivo encontra a frequência de ressonância em 126 Hz com uma amplitude em torno de 50% do valor real.

De forma similar ao realizado nos exemplos anteriores, foi simulado o circuito de 4ª ordem utilizando diferentes valores para a ordem do sistema de equações. O resultado foi mostrado no gráfico da Figura 4.16.

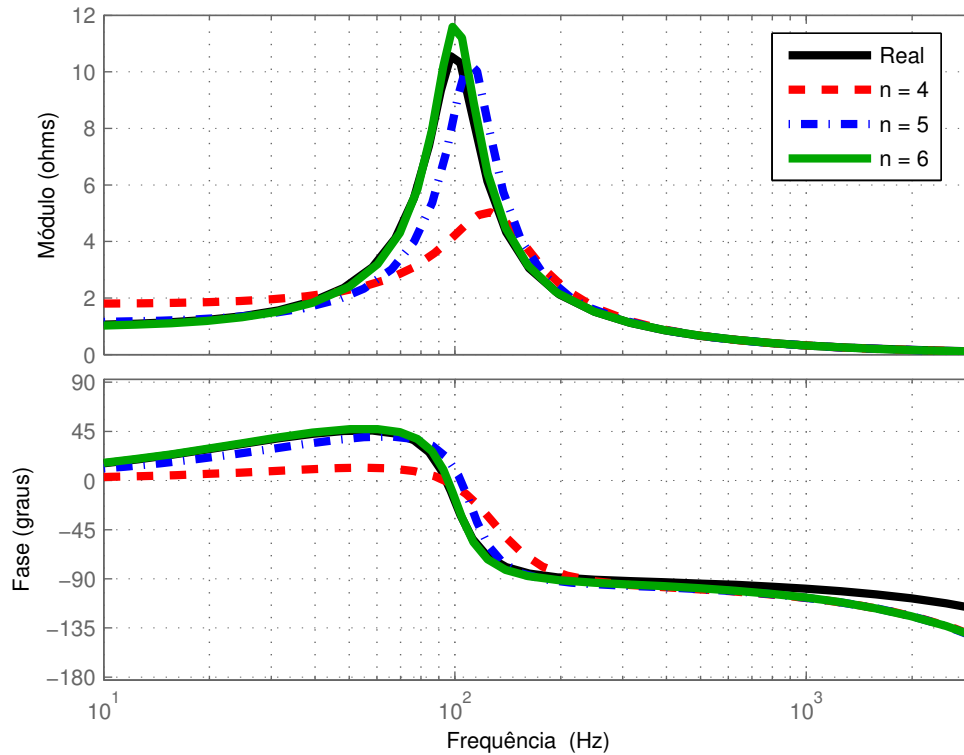


Figura 4.16: Gráfico das curvas real e calculada pelo método com impedância da rede de 4^a ordem para diferentes valores de ordem n.

Observando o gráfico da Figura 4.16 nota-se novamente que à medida que se aumenta a ordem do sistema há uma aproximação da curva simulada à curva real. Diferente da resposta em frequência obtida para n igual a 4, utilizando n igual a 6, o MCU foi capaz de encontrar o pico de ressonância em 100Hz.

4.5 Rede Benchmark

Na quinta e última simulação, o algoritmo foi testado sobre um modelo de rede padrão que melhor caracterize a rede de tensão trifásica contendo uma geração distribuída. A busca por um modelo padrão de rede, também conhecida por *benchmark*, visa se aproximar de um modelo real para melhor prever o funcionamento dos conversores conectados à rede. Dois modelos de *benchmark* foram aplicados em [12] onde buscava-se desenvolver um modelo de rede aérea de distribuição secundária¹. A estrutura da rede é do tipo radial e modelada com alimen-

¹Postes de concreto constituídos de seus condutores, transformadores, chaves e demais acessórios cuja fiação pode ser dividida em primária, em torno de 13,8kV de média tensão, e secundária, de

tadores, transformadores e bancos de capacitores. Dos modelos propostos em [12], aqui foi utilizado o *benchmark 1*, modelo mais simplificado de uma rede real que o *benchmark 2*, que possui cargas não-lineares. O modelo *benchmark 1* retrata com alto nível de detalhamento uma rede de baixa tensão e que pode ser utilizada como modelo de rede padrão visto pelos conversores VSC em geração distribuída. O modelo *benchmark 1* é um circuito de ordem igual a 40 que, além de conter capacitâncias e indutâncias, leva em consideração o modelo de cabeamento ac do PSIM.

A Figura 4.17 mostra algumas curvas geradas durante a simulação quando há injeção do pulso CHIRP na rede *benchmark 1*. No primeiro gráfico, é mostrada a variação de tensão no elo CC durante a injeção do pulso CHIRP. No segundo e terceiro gráficos, são mostradas as correntes e as tensões trifásicas medidas na saída do conversor.

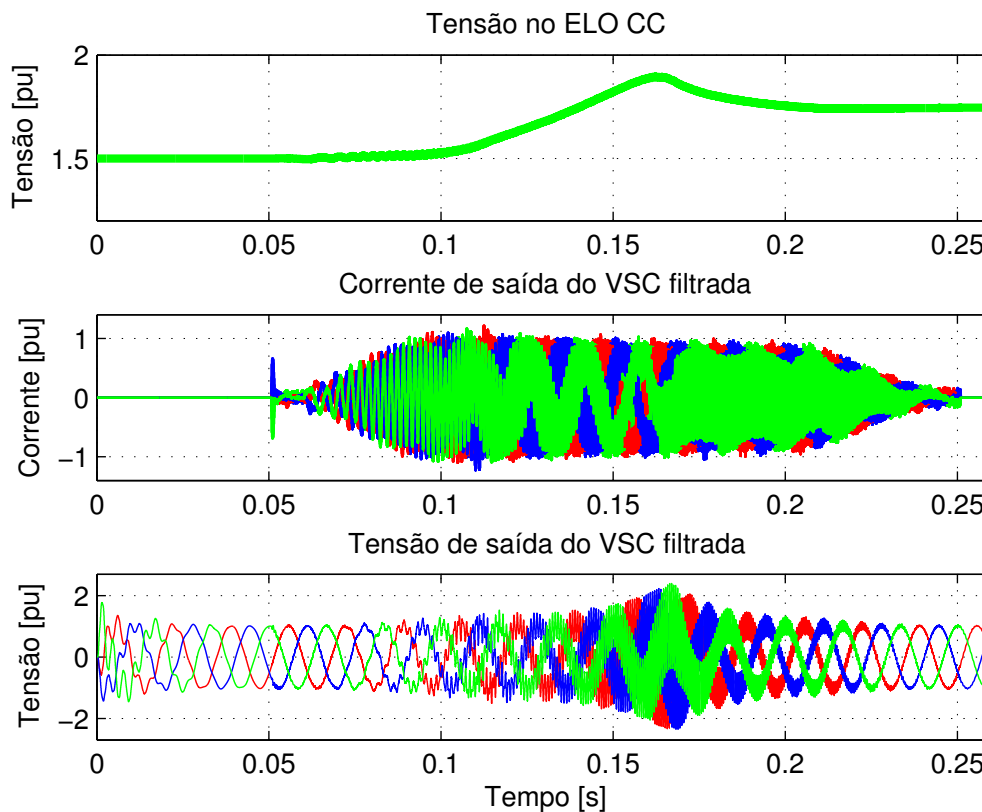


Figura 4.17: Gráfico das curvas da variação de tensão no elo CC, e das correntes e tensões na rede *benchmark 1*.

Na Figura 4.18, é mostrado o gráfico de resposta em frequência da impedância do modelo simplificado de rede utilizando o método recursivo para diferentes valores 220V/127V (fase-fase/fase-neutro) de baixa tensão.

de ordem n comparado com o do método utilizado em [12].

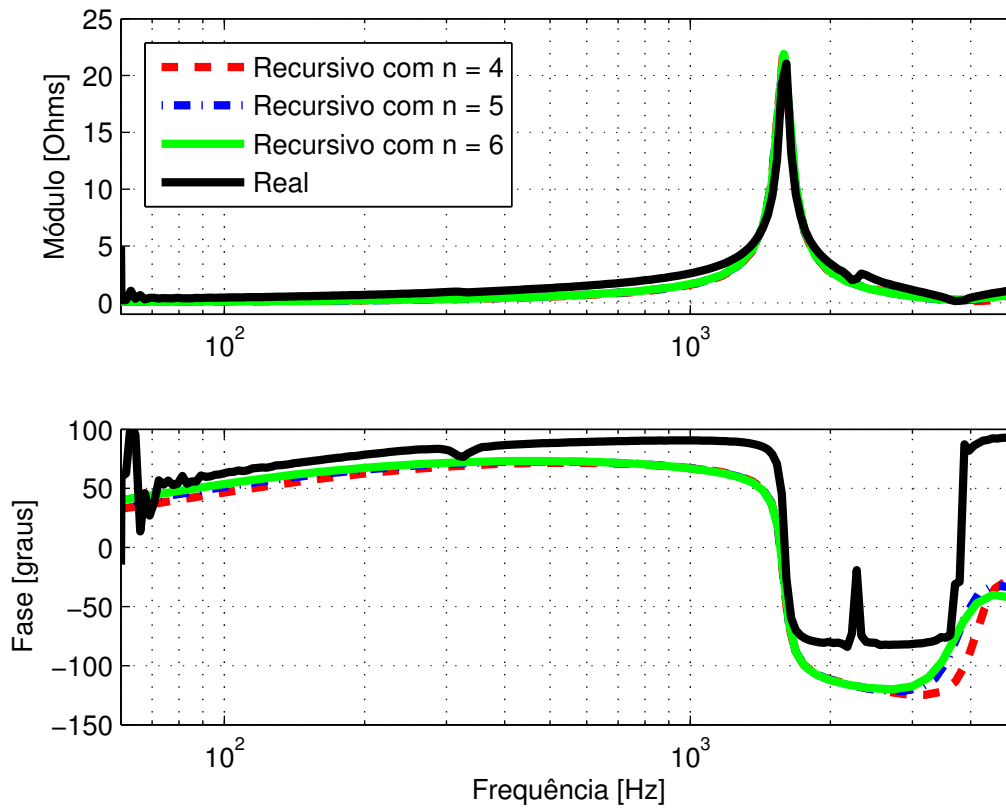


Figura 4.18: Resposta do método recursivo sobre a rede *benchmark* 1 considerando diferentes valores de n .

Em [12], a curva representa a impedância vista entre duas fases quando calcula-se a impedância injetando corrente entre duas fases e adquirindo a tensão de linha, ou seja, medindo-se a tensão entre duas fases. O método recursivo calcula a impedância da rede injetando corrente em uma das fases e medindo a tensão através de uma das tensões de fase. Portanto, para ser possível a comparação entre as duas curvas, a amplitude em [12] é dividida por raiz de 3. Note que o algoritmo foi capaz de encontrar o pico de ressonância da rede em $1200Hz$.

Os próximos gráficos da Figura 4.19 mostram a resposta em frequência da rede *benchmark* 1 a partir do método dos mínimos quadrados aplicando diferentes valores de n . Note que há uma maior aproximação da curva gerada pelo método com relação à curva real.

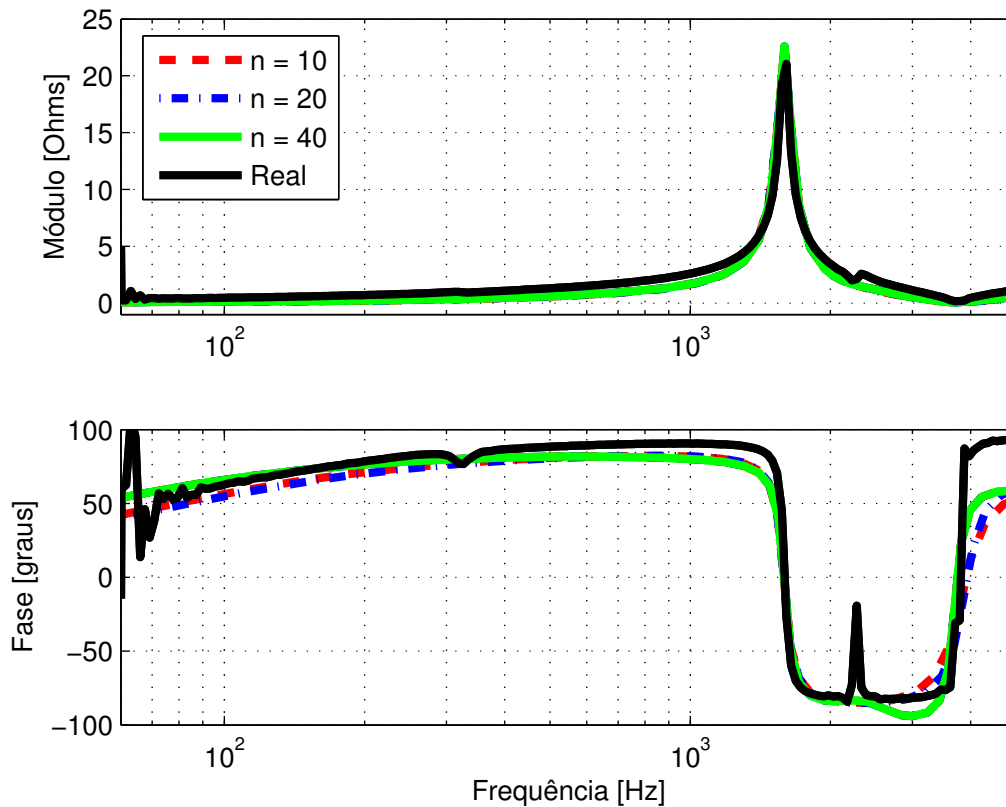


Figura 4.19: Resposta em frequência da rede *benchmark* 1 utilizando o método dos mínimos quadrados aplicando diferentes valores de n .

Uma importante conclusão parcial destes resultados é que o algoritmo mostra um desempenho superior à medida que se eleva a ordem do sistema de equações que é definida previamente antes do início da execução do algoritmo. Porém, como será analisado posteriormente, há um custo computacional nesta definição.

4.6 Análise do Tempo de Processamento

Nesta seção, foi feita a análise do tempo de processamento gasto pelo MCU na construção do vetor Γ e nas atualizações das matrizes Ξ e Φ em cada iteração. Lembrando que o algoritmo recursivo propõe reduzir o espaço de armazenamento em memória, porém, em contrapartida, há uma sobrecarga de operações do MCU durante cada iteração, principalmente, à medida em que se aumenta a ordem da rede. Portanto, é válido analisar o tempo de execução de cada iteração do algoritmo recursivo em função da ordem do sistema de equações. Como foi observado nos exemplos da seção anterior, a exatidão do modelo está diretamente relacionada à ordem definida previamente do sistema de equações. Porém, não se deve esquecer

do compromisso em obedecer a limitação do tempo de execução em cada iteração.

Quando se deseja determinar a impedância da rede vista pelo conversor, obviamente, não se conhece preliminarmente qual a ordem do sistema. E, também, observando os resultados obtidos, como o modelo se torna mais fiel à F.T. da rede à medida que se aumenta a ordem do sistema de equações, é desejável que a maior ordem possível seja definida antes da execução do modelo. Em contrapartida, isso acarreta num aumento quadrático do tempo de execução das operações como será visto mais adiante. Desta forma, o grande desafio do algoritmo recursivo, em termos práticos, será estender ao máximo o tempo entre iterações de tal forma que se possa escolher a maior ordem do sistema e, assim, obter a resposta em frequência da rede mais próxima da real possível. O tempo entre interações, por outro lado, é inversamente proporcional à taxa de amostragem dos sinais provenientes da rede. Ou seja, caso se deseje aumentar esta margem de tempo, deve-se diminuir a frequência de amostragem. Não se deve, porém, diminuir ilimitadamente a frequência de amostragem pois, pelo teorema de Nyquist-Shannon, a condição suficiente para a completa reconstrução do sinal analógico, sem que haja superposição dos sinais amostrados, a frequência de amostragem deve ser, no mínimo, o dobro da banda deste sinal.

Nestas simulações, como a banda do sinal de corrente em formato CHIRP é de 3kHz, a frequência de amostragem mínima, respeitando o Teorema de Nyquist-Shannon, é de 6kHz. Para esta frequência de amostragem, o tempo intra-iterações é dado por $166,7\mu s$. Se aumentar a frequência de amostragem para 100kHz, o tempo intra-iterações cai para $10\mu s$. Nota-se, portanto, que o grande fator limitante para definição da ordem do sistema de equações é a frequência de amostragem dos sinais adquiridos da rede.

Para se analisar a relação entre a ordem do sistema de equações e o tempo gasto pelo MCU em cada iteração, foi construída a Tabela 4.4. Os dados foram gerados pelo mesmo MCU utilizado nas simulações e descrito no capítulo 3. Foi contado o número de clocks consumidos pelo MCU apenas na atualização das matrizes recursivas para cada ordem do sistema de equações previamente estabelecida. O tempo gasto pelo MCU na execução deste número de clocks foi calculado considerando a velocidade do processador de 200MHz. Ou seja, para esta velocidade, cada ciclo completo de relógio tem um período de $5ns$.

Ordem	Número de Clocks	Tempo de Execução
1	377	1,885 μ s
2	1138	5,69 μ s
3	2278	11,39 μ s
4	3862	19,31 μ s
5	5814	29,07 μ s
6	8174	40,87 μ s
7	10923	54,61 μ s

Tabela 4.4: Tempo gasto pelo MCU na execução das operações intra-iterações em função da ordem do sistema de equações.

O gráfico da Figura 4.20 foi gerado utilizando os valores da Tabela 4.4. Nota-se uma curva exponencial à medida que se aumenta a ordem da rede. Isto se deve ao aumento de linhas e colunas que eleva o número de operações de forma quadrática.

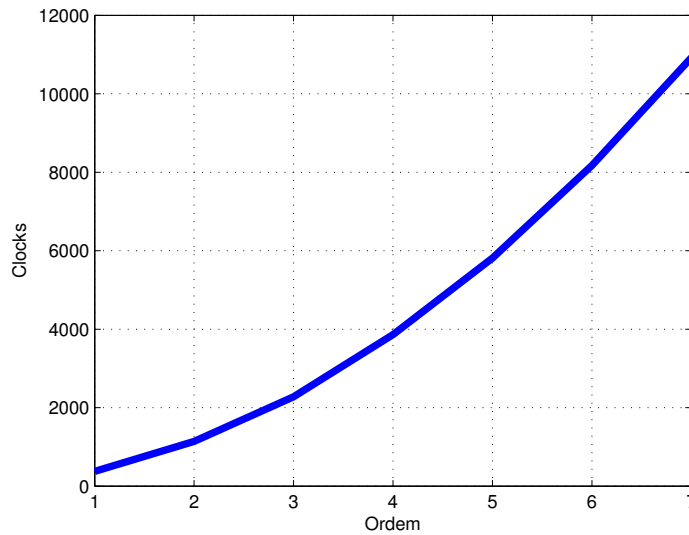


Figura 4.20: Número de clocks por iteração necessários para atualizar as matrizes Γ e Ξ em função da ordem.

Ajustando uma curva quadrática ao gráfico, gera-se a equação que melhor descreve a curva da Figura 4.20. Segue a equação gerada em (4.28):

$$CLK(n) = 200n^2 + 160n + 14 \quad (4.28)$$

Portanto, para o caso de uma frequência de amostragem de 100kHz, ou seja, um tempo entre iterações de 10 μ s, a ordem máxima do sistema de equações deve ser igual a 2. Por outro lado, se a frequência de amostragem for de 20kHz, a ordem do sistema de equações máxima a ser definida será igual a 6.

Capítulo 5

Conclusões e Trabalhos Futuros

A principal contribuição desta dissertação foi avaliar um método em forma de algoritmo recursivo para conversores VSC conectados à rede capaz de estimar os parâmetros da impedância da rede. Este algoritmo foi produzido a partir do Método dos Mínimos Quadrados com a vantagem de ser recursivo, ou seja, o resultado é construído cada iteração, e pode facilmente ser implementado no microcontrolador integrante do sistema de controle do conversor sem que haja necessidade de instalação de qualquer outro dispositivo na rede. O algoritmo também traz como vantagens o uso de pouco recurso de memória interna do controlador e rapidez nos seus resultados, pois é executado em tempo real. Além disso, a injeção de corrente num formato CHIRP traz consigo a vantagem de estimular um grande intervalo de harmônicos da rede num curto período de tempo. Como desvantagem, o método recursivo exige que o conversor, durante o processamento do algoritmo, deixe de trabalhar como transmissor de potência, tendo que se dedicar integralmente aos cálculos de impedância da rede ainda que por poucos centésimos de segundos. Um importante benefício trazido por este método será capacitar o conversor a realizar um auto-ajuste de seus parâmetros internos de controle com o fim de manter sua estabilidade conforme a impedância da rede se altere.

O algoritmo foi testado dentro do microcontrolador MCU *F28377S* da *Texas Instruments* num processo *Hardware-in-the-loop* onde a rede foi simulada dentro do software PSIM com interface via porta serial com o MCU. Uma técnica de identificação de funções de transferência discretas, por método de mínimos quadrados, foi utilizada para processar os sinais medidos no ponto de conexão entre o conversor e a rede. Equações matriciais foram construídas e um algoritmo recursivo foi utilizado a fim de calcular a impedância da rede. Foram alcançados bons resultados quando o método foi submetido a testes em redes de 1^a, 2^a e 4^a ordens e com redes de 1^a ordem possuindo harmônicos como observados no capítulo 4. Conseguiu-se também identificar uma rede *benchmark* utilizando valores elevados de ordem. Porém, num microprocessador real, obtém-se uma aproximação da curva *benchmark* devido à

necessidade de restringir a ordem do modelo por causa da limitação do tempo de processamento entre iterações. Para uma frequência de amostragem de 20kHz, a ordem do modelo escolhida previamente deve ser limitada à sexta ordem devido ao tempo de processamento gasto pelo MCU entre cada iteração.

Como sugestão de trabalhos futuros, o conversor VSC poderá ser testado com um controlador auto-ajustável em conjunto com este algoritmo. Neste caso, poderá ser observado o ajuste dinâmico dos ganhos das malhas de controle de tensão e de corrente quando ocorrer degradação do desempenho causada por variação na impedância da rede. Outro importante resultado a ser observado será o funcionamento otimizado do conversor independentemente do tipo de rede em que estiver conectado o que habilitará o conversor VSC a se conectar a qualquer rede sem a necessidade de ajustes adicionais por parte do operador. Desenvolver uma forma de otimizar o algoritmo recursivo proposto com o controle do elo CC evitando sobrecarga no capacitor também será desejável. A utilização do sinal CHIRP pode ser vantajosa no futuro por excitar um grande número de harmônicos da rede num curto espaço de tempo, porém, utilizar um outro método de injeção de corrente que produza menos ruídos e que não exija interrupção de transferência de potência seria interessante. E, por fim, a validação experimental num conversor real conectado à rede será de suma importância para o aperfeiçoamento deste algoritmo.

Referências Bibliográficas

- [1] ANEEL. “BIG - Banco de Informações de Geração”. Disponível em: <http://www.aneel.gov.br/aplicacoes/capacidadebrasil/capacidadebrasil.cfm>. Acesso em: 10 de Outubro de 2018.
- [2] BLOOMING, T. M., CARNOVALE, D. J. “Application of IEEE Std 519-1992 harmonic limits”. In: *Pulp and Paper Industry Technical Conference, 2006. Conference Record of Annual*, pp. 1–9. IEEE, 2006.
- [3] IEEE 519:1992. *IEEE Recommended Practices and Requirements for Harmonic Control in Electrical Power Systems*. Standard, Institute of Electrical and Electronics Engineers, April 1993.
- [4] MIDDLEBROOK, R. D. “Input filter considerations in design and application of switching regulators”, *IAS’76*, 1976.
- [5] FAMILIANT, Y. A., HUANG, J., CORZINE, K. A., et al. “New techniques for measuring impedance characteristics of three-phase AC power systems”, *IEEE Transactions on Power Electronics*, v. 24, n. 7, pp. 1802–1810, 2009.
- [6] SUDHOFF, S. D., GLOVER, S. F., LAMM, P. T., et al. “Admittance space stability analysis of power electronic systems”, *IEEE Transactions on Aerospace and Electronic Systems*, v. 36, n. 3, pp. 965–973, 2000.
- [7] BELKHAYAT, M., WILLIAMS, M. “Impedance extraction techniques for DC and AC systems”. In: *Proceedings of the Naval Symposium on Electric Machines*, 2000.
- [8] WILDRICK, C. M., LEE, F. C., CHO, B. H., et al. “A method of defining the load impedance specification for a stable distributed power system”, *IEEE Transactions on power Electronics*, v. 10, n. 3, pp. 280–285, 1995.
- [9] CESPEDES, M., SUN, J. “Adaptive control of grid-connected inverters based on online grid impedance measurements”, *IEEE Transactions on sustainable energy*, v. 5, n. 2, pp. 516–523, 2014.

- [10] MIDTSUND, T., SUUL, J., UNDELAND, T. “Evaluation of current controller performance and stability for voltage source converters connected to a weak grid”. In: *Power Electronics for Distributed Generation Systems (PEDG), 2010 2nd IEEE International Symposium on*, pp. 382–388. IEEE, 2010.
- [11] CHEN, X., ZHANG, Y., WANG, S., et al. “Impedance-Phased Dynamic Control Method for Grid-Connected Inverters in a Weak Grid”, *IEEE Transactions on Power Electronics*, v. 32, n. 1, pp. 274–283, 2017.
- [12] DE OLIVEIRA RODRIGUES, R. “Conversores VSC controlados por Corrente Conectados em Redes de Distribuição: Análise de Interações Adversas”. Dissertação de Mestrado UFRJ/COPPE, 2015.
- [13] VIEIRA JUNIOR, J. C. D. M., OTHERS. “Metodologias para ajuste e avaliação do desempenho de relés de proteção anti-ilhamento de geradores síncronos distribuídos”, 2006.
- [14] WATANABE, E. H. “Conversores de Corrente”, *Notas de Aula da Disciplina de Aplicações de Eletrônica de Potência*, Julho 2015.
- [15] CESPEDES, M., SUN, J. “Three-phase impedance measurement for system stability analysis”. In: *Control and Modeling for Power Electronics (COMPEL), 2013 IEEE 14th Workshop on*, pp. 1–6. IEEE, 2013.
- [16] RICCOBONO, A., NAQVI, S. K. A., MONTI, A., et al. “Online wideband identification of single-phase AC power grid impedances using an existing grid-tied power electronic inverter”. In: *Power Electronics for Distributed Generation Systems (PEDG), 2015 IEEE 6th International Symposium on*, pp. 1–8. IEEE, 2015.
- [17] SHEN, Z., JAKSIC, M., MATTAVELLI, P., et al. “Three-phase AC system impedance measurement unit (IMU) using chirp signal injection”. In: *Applied Power Electronics Conference and Exposition (APEC), 2013 Twenty-Eighth Annual IEEE*, pp. 2666–2673. IEEE, 2013.
- [18] LJUNG, L. *System Identification: Theory for the User*. Pearson Education, 1998. ISBN: 9780132440530. Disponível em: <<https://books.google.com.br/books?id=fYSrk4wDKPsc>>.
- [19] PINTELO, R., SCHOUKENS, J. *System identification: a frequency domain approach*. John Wiley & Sons, 2012.

- [20] CIOBOTARU, M., TEODORESCU, R., BLAABJERG, F. “On-line grid impedance estimation based on harmonic injection for grid-connected PV inverter”. In: *Industrial Electronics, 2007. ISIE 2007. IEEE International Symposium on*, pp. 2437–2442. IEEE, 2007.
- [21] ASIMINOAEI, L., TEODORESCU, R., BLAABJERG, F., et al. “A new method of on-line grid impedance estimation for PV inverter”. In: *Applied Power Electronics Conference and Exposition, 2004. APEC'04. Nineteenth Annual IEEE*, v. 3, pp. 1527–1533. IEEE, 2004.
- [22] KNOP, A., FUCHS, F. “High frequency grid impedance analysis by current injection”. In: *Industrial Electronics, 2009. IECON'09. 35th Annual Conference of IEEE*, pp. 536–541. IEEE, 2009.
- [23] NUNES, R. V. “Análise da penetração harmônica em redes de distribuição desequilibradas devido às cargas residenciais e comerciais com a utilização do ATP”. 2007.
- [24] SCHWARZ, A. “Use of an inductor within a full bridge dc-dc power converter”. nov. 17 1981. US Patent 4,301,496.
- [25] MASSING, J. R., STEFANELLO, M., GRUNDLING, H. A., et al. “Adaptive current control for grid-connected converters with LCL filter”, *IEEE Transactions on Industrial Electronics*, v. 59, n. 12, pp. 4681–4693, 2012.
- [26] COBRECES, S., RODRIGUEZ, P., PIZARRO, D., et al. “Complex-space recursive least squares power system identification”. In: *Power Electronics Specialists Conference, 2007. PESC 2007. IEEE*, pp. 2478–2484. IEEE, 2007.
- [27] BLAABJERG, F., TEODORESCU, R., LISERRE, M., et al. “Overview of control and grid synchronization for distributed power generation systems”, *IEEE Transactions on industrial electronics*, v. 53, n. 5, pp. 1398–1409, 2006.
- [28] KAUR, H., DHALIWAL, B. “Design of Low Pass FIR Filter Using Artificial NeuralNetwork”, *International Journal of Information and Electronics Engineering*, v. 3, n. 2, pp. 204, 2013.
- [29] STRANG, G. *Introduction to linear algebra*, v. 3. Wellesley-Cambridge Press Wellesley, MA, 1993.

- [30] COSTA, D., ROLIM, L. G., AREDES, M. “Analysis and software implementation of a robust synchronizing circuit PLL circuit”. In: *Industrial Electronics, 2003. ISIE'03. 2003 IEEE International Symposium on*, v. 1, pp. 292–297. IEEE, 2003.

Apêndice A

Phase-Locked-Loop

O PLL é um sistema de controle que serve para identificar o ângulo de fase da rede. Neste trabalho, ele é primordial pois a consideração do ângulo de fase da fundamental em cada instante reduz o efeito da rede nas medições. Uma vez estabilizado o algoritmo do PLL e obtido o ângulo de fase da rede, são preenchidas as componentes cossenoidal e senoidal da rede e de seus harmônicos a cada instante. Para o método dos mínimos quadrados, a fim de exemplificação, é preenchida a equação matricial (2.11) e, se forem considerados harmônicos da rede, a equação matricial (2.12) deverá ser preenchida, ou seja, encontra-se a fase da fundamental e multiplica-se pela ordem do harmônico presente. Para o método recursivo, que é o objetivo deste trabalho, atualiza-se o vetor Γ conforme mostra a equação (2.15).

Como descrito no capítulo 2, o PLL, em forma de algoritmo, entra em funcionamento apenas na primeira parte da simulação quando ainda não há injeção do pulso CHIRP na rede, portanto, sem interferências do sinal gerado pelo conversor. É realizado desta forma pois, na primeira parte da simulação, se deseja apenas medir a tensão e obter a fase da rede antes do início propriamente dito do algoritmo recursivo. Na segunda parte da simulação, o PLL é desconectado e a fase é incrementada e utilizada no decorrer do algoritmo.

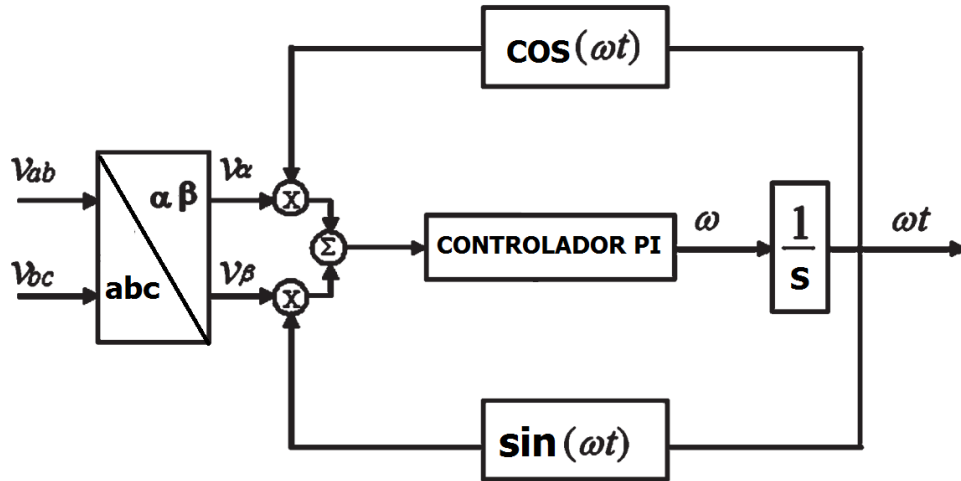


Figura A.1: Diagrama em Blocos do PLL.

O algoritmo do PLL apresentado em [30] foi utilizado conforme mostra o diagrama em blocos da Figura (A.1). Note que a saída ωt da Figura (A.1) é a fase θ da fundamental após estabilização do algoritmo. Após estabilização do sistema de controle do PLL, é encontrado o valor do ângulo de fase da componente fundamental da tensão no PCC. A partir deste ponto, o algoritmo é desconectado e o valor encontrado de ângulo de fase é incrementado a cada instante até o fim da injeção do pulso CHIRP. Em cada instante, a fase é incrementada considerando uma frequência da fundamental em 60Hz e o passo de integração de $50\mu s$. O código do PLL foi transcrito na seção seguinte juntamente com o restante do código do algoritmo recursivo de estimação de parâmetros de impedância da rede.

Apêndice B

Código do Algoritmo Recursivo

```
//-----  
// ----- Parâmetros da Função de Transferência -----  
//-----  
#define ORDEM 6 // Definida previamente  
#define HARMONICOS_GRID 1 // Número de harmônicos na rede  
#define TX_AMOST 20000 // Frequência de amostragem = 20kHz  
#define N 4000 // N = (t1 - t0)*TX_AMOST  
  
//----- Variáveis de Tempo -----  
double t0 = 0.05085;  
double t1 = 0.25085;  
double V_GRID_AMPLITUDE = 179.629248;  
double IDC_REF = 50;  
double h = (double)1/TX_AMOST;  
double pi = 3.1415926;  
double t = 0.0;  
  
// -----Varáveis utilizadas para Operações com Matrizes -----  
int m, n, o, i, j, k;  
double pivot, soma;  
unsigned int E = {2*ORDEM+2*HARMONICOS_GRID};  
  
// -----Variáveis do pulso CHIRP -----  
double fase;  
double f0 = 0;  
double f1 = 3000;  
double T;
```

```

// ----- Janela Tukey -----
double alfa = 0.5;
double tukey = 1.0;

// ----- Variáveis da Corrente de Referência -----
double Ia_ref;
double Ib_ref;
double Ic_ref;

// ----- Variáveis do PLL -----
double phasePLL = 0.0;
double Valpha = 0.0;
double Vbeta = 0.0;
double erro[2] = { 0.0 , 0.0 };
double freq = 376.99111843;
double Kp_pll = 0.1;
double Ki_pll = 0.1 * 20000;
float Va_PLL, Vb_PLL, Vc_PLL;

// ----- Função de Transferência -----
double ThetaR[2*ORDEM + 2*HARMONICOS_GRID][1];
double L[2*ORDEM + 2*HARMONICOS_GRID][1];
double Zeta[2*ORDEM + 2*HARMONICOS_GRID][2*ORDEM +
2*HARMONICOS_GRID];
double Phi[2*ORDEM + 2*HARMONICOS_GRID][1];
double Ytemp = {0.0};

//-----
static unsigned long int contador = 0;
static double Xatual;
static double Yatual;

//----- Variáveis de Entrada -----
double Ia = 0.0;
double Ib = 0.0;
double Ic = 0.0;
double Va = 0.0;
double Vb = 0.0;

```



```

double Vc = 0.0;
double Ia_filt;
double Va_filt;

// -----Variáveis de Saida -----
double chaveA, chaveB, chaveC;
double out[20];

// ----- Variável global de contagem de iterações -----
unsigned long int LoopCount;

void main(void)
{
// -----
//                               INICIANDO MICROCONTROLADOR
// -----
LoopCount = 0;
T = t1 - t0;
for(;;)
{

//-----
//-----PROCESSAMENTO-----
//-----
//----- Construindo Sinal de Corrente no formato CHIRP -----
if ((t >= t0) & (t <= t1)) {
    fase = (2 * pi*(f0 + (f1 - f0)*(t - t0)/(2*T)) * (t-t0) ) -
    - 2 * pi*( (int)((f0 + (f1 - f0)*(t - t0)/(2*T)) * (t-t0) ) );
}

//----- Construindo Janela TUKEY-----
if ((t >= t0) && ( t < (t1-t0)*alfa/2 + t0)) {
    tukey = (0.5)*(1 + cos( (2*pi/alfa)*( t/(t1-t0) - t0/(t1-t0) -
    - alfa/2 ) ));
}
else if ((t <= t1) && ( t >= t1 - (t1-t0)*(alfa/2) )){
    tukey = (0.5)*(1 + cos( (2*pi/alfa)*( t/(t1-t0) - t1/(t1-t0) +
    + alfa/2 )));
}
}

```

```

//-----
// Corrente de Referência para Controle Bang Bang de Corrente
//-----
Ia_ref = tukey*Idc_ref*sin(fase);
Ib_ref = tukey*Idc_ref*sin(fase - 2.0944);
Ic_ref = tukey*Idc_ref*sin(fase - 2*2.0944);

//----- PLL -----
if (t < t0) {
    Va_PLL = (float)Va/V_GRID_AMPLITUDE;
    Vb_PLL = (float)Vb/V_GRID_AMPLITUDE;
    Vc_PLL = (float)Vc/V_GRID_AMPLITUDE;
    Valpha = 0.8165*( Va_PLL - 0.5 * Vb_PLL - 0.5 * Vc_PLL );
    Vbeta = 0.8165*(0.8660* (Vb_PLL - Vc_PLL));
    erro[0] = Valpha * cos(phasePLL) + Vbeta * sin(phasePLL);
    freq = freq + Kp_pll * erro[0] + Ki_pll * (h/2) * (erro[0] +
        + erro[1]);
} else {
    freq = 2*pi*60;
}
phasePLL = phasePLL + 0.5 * h * (2 * freq);
if (phasePLL > 2*pi){
    phasePLL -= 2*pi;
}
erro[1] = erro[0];

//----- Gerando Chaveamento Bang Bang-----
if ((t >= t0) & (t <= t1)) {
    if (Ia > Ia_ref) {
        chaveA = 0;
    } else chaveA = 1;
    if (Ib > Ib_ref) {
        chaveB = 0;
    } else chaveB = 1;
    if (Ic > Ic_ref){
        chaveC = 0;
    } else chaveC = 1;
} else {

```

```

    chaveA = 1;
    chaveB = 1;
    chaveC = 1;
}

//----- Iniciando Processamento -----
if (ORDEM == 1) {
    Xatual = Va_filt/V_GRID_AMPLITUDE;
    Yatual = Ia_filt/IDC_REF;
}
else {
    Xatual = Ia_filt/IDC_REF;
    Yatual = Va_filt/V_GRID_AMPLITUDE;
}

//----- Calculando Parâmetros da FT Recursivamente -----
//-----Inicializando Variáveis-----
if (contador <= (N-1)) {
    if (contador==0){
        for (i=0; i<E; i++){
            Phi[i][0] = 0;
            ThetaR[i][0] = 0;
            for (j=0; j<E; j++){
                Zeta[i][j] = 0;
            }
        }
    }
    for (i=(2*ORDEM-1); i>0; i--){
        L[i][0] = L[i-1][0];
    }

    L[0][0] = -Ytemp; // L[0][0](k) = -Y(k-1)
    Ytemp = Yatual;
    L[ORDEM][0] = Xatual; // L[1][0] = X(k)
    L[E-2][0] = sin(phasePLL);
    L[E-1][0] = cos(phasePLL);
    for (i=HARMONICOS_GRID, j=1; j<=HARMONICOS_GRID; j++, i--){
        if (j - ((int)(j/2)*2)){
            L[E-2*i][0] = sin((3*j-2)*phasePLL);
        }
    }
}

```

```

        L[E-2*i+1][0] = cos((3*j-2)*phasePLL);
    } else {
        L[E-2*i][0] = sin((3*j-1)*phasePLL);
        L[E-2*i+1][0] = cos((3*j-1)*phasePLL);
    }
}

if (contador >= ORDEM){

//-----Atualizando Matriz Phi -----
    for (i=0; i<E; i++){
        Phi[i][0] += Yatual*L[i][0];
    }

//-----Adquirindo z(k) e Adicionando a Z -----
// Multiplicando L (E x 1) por Ltransp (1 x E) = Zeta (E x E)
// Algoritmo de Multiplicacao de Matrices
    m = E;
    o = E;
    for (i=0; i<m; i++){
        for (k=0; k<o; k++){
            Zeta[i][k] += L[i][0]*L[k][0];
        }
    }
}

if (contador== (N-1)){
    m = E;
    for (k=0; k<m; k++){
        for (i=(k+1), j=k; i<m; i++, j=k){
            pivot = Zeta[i][j]/Zeta[k][k];
            for (j=0; j<m; j++){
                Zeta[i][j] = Zeta[i][j] - pivot*Zeta[k][j];
            }
            Phi[i][0] = Phi[i][0] - pivot*Phi[k][0];
        }
    }
    ThetaR[m-1][0] = Phi[m-1][0]/Zeta[m-1][m-1];
    for (k = (m-2), soma=0; k>=0; k--){
        for (j=(m-1); j>k; j--){

```

```

        soma+= ThetaR[j] [0]*Zeta[k] [j];
    }
    ThetaR[k] [0] = (Phi[k] [0] - soma)/Zeta[k] [k];
    soma = 0;
}
}
if (t >= t0){
    contador++;
}

//-----Iniciando outra iteração -----
LoopCount++;

// -----Atualizando Variável Tempo -----
t = (double)LoopCount*h;
}
// -----
// -----
}

```