

UNIVERSIDADE FEDERAL DE JUIZ DE FORA
FACULDADE DE ENGENHARIA
CURSO DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

**Análise e Síntese de um Algoritmo “Phase-
Locked-Loop” Robusto para Estimação de
Amplitude, Fase e Frequência de Sinais
Elétricos**

Pedro Henrique de Castro Gomes

Dissertação de mestrado apresentada ao
Colegiado do Curso de Mestrado em Engenharia
Elétrica da Faculdade de Engenharia da
Universidade Federal de Juiz de Fora como parte
integrante dos requisitos necessários à obtenção
do título de Mestre em Engenharia Elétrica

Orientador: Prof. Carlos Augusto Duque, Dr.

Juiz de Fora, agosto de 2007

Análise e Síntese de um Algoritmo “Phase-Locked-Loop” Robusto para Estimação de Amplitude, Fase e Frequência de Sinais Elétricos

Pedro Henrique de Castro Gomes

Esta dissertação foi julgada adequada para a obtenção do título de Mestre em Engenharia Elétrica e aprovada em sua forma final pelo Colegiado do Curso de Mestrado em Engenharia Elétrica da Faculdade de Engenharia da Universidade Federal de Juiz de Fora.

Prof. Carlos Augusto Duque
Orientador, DCE/FENG-UFJF

Banca Examinadora:

Prof. Carlos Augusto Duque
Presidente, DCE/FENG-UFJF

Prof. Dr. Jacques Szczupak
DEE/PUC-RJ

Prof. Moisés Vidal Ribeiro
DEE/FENG-UFJF

Prof. Augusto Santiago Cerqueira
DCE/FENG-UFJF

RESUMO

A crescente demanda pelos consumidores por índices de qualidade de energia cada vez mais elevados e a recente desregulamentação do setor elétrico, vem exigindo uma demanda cada vez maior pelo monitoramento da qualidade da energia elétrica pelas concessionárias de uma maneira descentralizada. Aliado a esse fato, a alta proliferação de cargas não lineares conectadas à rede elétrica, entre outros, têm tornado a estimação de parâmetros dos sinais elétricos da rede uma tarefa cada vez mais desafiadora. Assim, o desenvolvimento de algoritmos de estimação eficientes e com baixa complexidade computacional, ou passíveis de implementação em sistemas (*hardwares*) de baixo custo, têm-se tornando uma prerrogativa importante.

Nesse escopo, essa dissertação apresenta a descrição de uma malha de PLL (*Phase-Locked-Loop*) robusta (ER-QPLL), capaz de estimar os parâmetros (fase, frequência e amplitude) da componente fundamental de um sinal de entrada qualquer. O desenvolvimento da estrutura baseou-se no aprimoramento de uma malha de PLL do tipo quadratura (QPLL), que estima os parâmetros da componente fundamental de um sinal de entrada através da aquisição das suas componentes em fase e em quadratura. As modificações da malha foram a introdução de um filtro *notch* adaptativo em sua entrada e a implementação de toda a estrutura utilizando o operador delta (δ), relacionado à Transformada Gama (γ). A introdução do filtro *notch* adaptativo na entrada da malha garante uma significativa melhoria na relação *SNR* do sinal de entrada, sem prejudicar demasiadamente a resposta dinâmica da estrutura. A característica adaptativa do filtro garante uma performance satisfatória da malha para sinais de entrada com parâmetros variantes no tempo. A implementação da malha utilizando o operador delta (δ) assegura uma performance ideal quando a mesma é implementada em sistemas de precisão limitada de, no mínimo, 16 bits. De acordo com os resultados demonstrados nesse trabalho, a performance da malha é satisfatória mesmo ao se utilizar altas taxas de amostragem relativas à frequência de operação da malha. Finalmente, foi proposta uma implementação da malha em um microprocessador (DSP) da família TMS320, o que comprova a viabilidade de implementação da mesma em sistemas (*hardware*) de ponto fixo.

ABSTRACT

The always more restrictive energy quality benchmarks, pushed on by consumers, associated with the electric sector deregulation has been imposing the necessity, for the concessionaries, of a better and decentralized monitoring of energy electric quality.

At the same time, the increase of nonlinear loads connected to the electric network, among other facts, has been increasing the complexities associated with this electric signals parameters estimation. So, the synthesis of efficient parameters estimation algorithms, with low computational effort and with easy implementation on low-cost hardware systems has becoming a priority for the energy quality area.

Based on these assumptions, this work deals with the design and synthesis of a robust Phase-Locked-Loop (PLL) structure, more specifically an Enhanced Quadrature Phase-Locked-Loop (ER-QPLL) with capacity of estimate several parameters, more specifically phase, frequency and amplitude, from any input signal. The synthesis of this ER-QPLL structure was based on the enhancement of a Quadrature Phase-Locked-Loop (QPLL) that can estimate the parameters of the fundamental component of any input signal through the information acquired with the acquisition of its phase and quadrature components.

The enhancements of this QPLL structure were, basically, the introduction of an adaptive notch filter on its input, associated with a delta operator (δ), a tool of the gamma transformer (γ), for modeling the whole structure. A significant improvement in the SNR of the input signal, without degradation of the dynamic structure output, was achieved with the introduction of the notch filter. The adaptive characteristics of this notch filter can deal, in a very good way, with the non-stationary properties of the input signals.

The structure implementation based on delta operator (δ) can assure an almost ideal performance for limited precision systems of, at least, 16 bits. According to the results obtained in this work, the performance of the proposed structure can be considered very good, even when dealing with high sampling rates relative to the network frequency operation. Finally, a structure based on a microprocessor DSP from TMS320 family was proposed and implemented showing its feasibility for fixed-point hardware.

A Petros Vinícius Araújo Damasceno
in memoriam

Mesmo que eu tivesse o dom da profecia, e conhecesse todos os mistérios e toda a ciência; mesmo que tivesse toda a fé, a ponto de transportar montanhas, se não tiver amor, não sou nada.
Coríntios - Cap.1, ver. 2, Bíblia Sagrada

Agradecimentos

Agradeço primeiramente a Deus, por ter criado todas as formas de vida que existem no universo e sem as quais o mesmo não teria sentido. Agradeço-o, também, por ter me dado saúde e inteligência para cumprir mais esse desafio na minha vida.

Gostaria de agradecer aos meus pais, Chico e Elizabete, por sempre terem me incentivado a nunca abandonar os estudos.

Agradeço a Vitor, Leandro e Mariana, pela amizade e companheirismo.

Agradeço imensamente aos meus orientadores Carlos Augusto Duque e Moisés Vidal Ribeiro, pelas idéias que contribuíram na conclusão desse trabalho e pela paciência de me orientar durante o período de redação da minha dissertação. Sem o seu apoio, esse trabalho certamente não teria sido concluído.

A todos os amigos do Laboratório de Sistemas Eletrônicos – LABSEL, pela amizade e horas de lazer compartilhadas durante a minha estadia no Laboratório, que foram fundamentais para a conclusão desse trabalho.

Aos amigos do mestrado, Cristiano, Danton, Fabrício e Iran, pelas idéias e amizade durante o período da dissertação.

A todas as pessoas que, de uma forma ou de outra, auxiliaram na conclusão desse trabalho.

SUMÁRIO

Resumo e Abstract	iii
Dedicatória	v
Agradecimentos	vii
Lista de Abreviaturas	x
Lista de Figuras	xi
Lista de Tabelas	xiv
Capítulo 1 – Estimação de Sinais Elétricos	1
1.1 Introdução	1
1.2 Objetivo do Trabalho	4
1.3 Divisão do Trabalho	6
Capítulo 2 – Phase-Locked Loops - PLLs	8
2.1 Introdução	8
2.2 PLL Analógicos	9
2.3 Comportamento dinâmico de uma malha PLL	11
2.3.1 Detector de Fase	11
2.3.2 Filtro Passa-baixas	12
2.3.3 VCO (Voltage Controlled Oscillator)	12
2.3.4 Análise da Malha de Controle do PLL	13
2.4 PLLs Digitais	16
2.4.1 Desenvolvimento de uma estrutura PLL totalmente digital	18
2.4.2 Ajuste dos Coeficientes da Malha PLL Digital	19
2.5 Quadrature Phase-Locked Loop - QPLL	22
2.6 Simulações Computacionais	28
2.7 Conclusão	33
Capítulo 3 – Enhanced Quadrature Phase-Locked Loop – E-QPLL	34
3.1 Introdução	34
3.2 O filtro <i>notch</i>	35
3.2.1 Introdução	35
3.2.2 Melhoria da Relação Sinal Ruído	38
3.3 O QPLL Modificado	41
3.3.1 Atualização da Frequência pela Média	41
3.3.2 Atualização instantânea	42
3.3.3 Comparação do desempenho	43
3.4 Comparação das estruturas E-QPLL e QPLL	45
3.4.1 Sinal Senoidal	47
3.4.2 Estimação em presença de harmônicos	48
3.4.3 Mudança de Frequência	49
3.4.3 Estimação do Terceiro Harmônico	50
3.4.5 Estimação com flicker	51
3.4.6 Estimação com variação senoidal da frequência	52

3.5	Estimação de Fase.....	54
3.5.1	Fase total e sinal de referência	56
3.5.2	Frequência variante em degrau.....	57
3.6	Conclusões	59
Capítulo 4 – Enhanced and Robust Quadrature Phase-Locked Loop – ER-QPLL.....		60
4.1	Introdução.....	61
4.2	Operador Delta	62
4.2.1	Equações a Diferença	63
4.2.2	Transformada Gama.....	65
4.3	E-QPLL	69
4.3.1	Malha QPLL Quantizada	70
4.3.1.1	Estratégia de implementação em ponto fixo	70
4.3.1.2	Resultados Computacionais	74
4.3.2	Malha E-QPLL quantizada	79
4.3.2.1	Filtro Notch Quantizado	79
4.4	ER-QPLL.....	83
4.4.1	Filtro <i>Notch</i> através da Transformada Gama	87
4.4.2	Otimização do parâmetro delta.....	89
4.4.3	Escolha do parâmetro delta.....	90
4.5	Resultados Comparativos	91
4.6	Implementação no TMS320F2812.....	94
4.7	Conclusões	97
Capítulo 5 – Conclusões Gerais.....		98
Referências Bibliográficas.....		103
Anexos....		106

LISTA DE ABREVIATURAS

- DFT** Transformada Discreta de Fourier - *Discrete Fourier Transform*
- FFT** Transformada Rápida de Fourier – *Fast Fourier Transform*
- PLL** *Phase-Locked-Loop*
- QPLL** *Quadrature Phase-Locked-Loop*
- E-QPLL** *Enhanced Quadrature Phase-Locked-Loop*
- ER-QPLL** *Enhanced and Robust Quadrature Phase-Locked-Loop*
- SNR** Relação Sinal Ruído - *Signal to Noise Ratio*
- DSP** Processador Digital de Sinais - *Digital Signal Processor*
- VCO** Oscilador controlado por voltagem – *Voltage Controlled Oscillator*
- SPLL** *Software PLL*
- PD** Detector de Fase – *Phase Detector*
- LF** Filtro passa-baixas
- MSE** Erro quadrático médio – *Mean square error*
- FPGA** *Field Programmable Gate Arrays*
- DFII** *Direct Form II*
- DFII_t** *Transposed Direct Form II*
- EES** *Error Spectrum Shaping*
- RAM** Random Access Memory
- JTAG** *Joint Test Action Group*
- MIPS** *Millions Instructions per second*

LISTA DE FIGURAS

2.1	Modelo simplificado de uma malha PLL.....	9
2.2	Modelo linear do detector de fase.....	12
2.3	Modelo linear do VCO.....	13
2.4	Modelo linear do VCO incluindo uma condição de contorno.....	13
2.5	Modelo linear do PLL.....	13
2.6	Modelo discretizado de um PLL.....	16
2.7	PLL digital.....	18
2.8	Resposta ao degrau da malha de PLL.....	20
2.9	Resposta em frequência da malha de PLL.....	20
2.10	Resposta ao degrau da malha de PLL.....	21
2.11	Resposta em frequência da malha de PLL.....	21
2.12	Malha QPLL.....	27
2.13	Malha QPLL simplificada.....	28
2.14	Desempenho inicial (a) Sinais de entrada e saída da malha (b) Erro de frequência.....	29
2.15	Parâmetros adquiridos pela malha de QPLL. (a) K_C . (b) K_V . (c) Amplitude.....	29
2.16	Sinal de entrada $u(n)$ e sinal de saída $y(n)$	30
2.17	Imunidade a ruídos. (a) Ruído de entrada. (b) Ruído de saída.....	30
2.18	Sinal de entrada $u(n)$ e sinal de saída $y(n)$	31
2.19	Imunidade a ruídos. (a) Ruído de entrada. (b) Ruído de saída.....	31
2.20	Variação da resposta dinâmica e rejeição a distúrbios da malha QPLL em função do ajuste dos parâmetros k_p e k_r . (a) Tempo para a amplitude da malha QPLL atingir 99% do seu valor final. (b) Relação Sinal/ruído de saída da malha QPLL.....	32
2.21	Resposta da malha a um distúrbio do tipo SAG seguido por um SWELL.....	33
3.1	Filtro passa-banda obtido a partir do filtro <i>notch</i>	36
3.2	Resposta ao degrau unitário ($\rho = 0,99$).....	37
3.3	Resposta em frequência do filtro <i>notch</i> : (a) Magnitude e (b) Fase para ($\rho_0 = 0,99$) (c) Magnitude e (d) Fase para $\rho_0 = 0,93$	38
3.4	Sinais de entrada (a) e saída (b) do filtro passa-banda.....	40
3.5	Sinais de entrada (a) e saída (b) do filtro passa-banda.....	40
3.6	Diagrama de blocos do QPLL modificado.....	41
3.7	Atualização do <i>notch</i> pela média das estimações da frequência.....	42
3.8	Algoritmo de atualização filtro <i>notch</i> : traço contínuo, atualização instantânea; traço tracejado, atualização pela média.....	43
3.9	Estratégia de cálculo do erro quadrático médio.....	44

3.10	Variação do erro médio quadrático para os valores $k_p = k_i$ variando desde $k_p = k_i = 1$ até $k_p = k_i = 250$	46
3.11	Estimação para o caso 1. (a) Amplitude; (b) Freqüência	48
3.12	Estimação para o caso 2. (a) Amplitude; (b) Freqüência	49
3.13	Estimação para o caso 3. (a) Amplitude; (b) Freqüência	50
3.14	Estimação para o caso 4. (a) Amplitude; (b) Freqüência	51
3.15	Estimação para o caso 5. (a) Amplitude; (b) Freqüência	52
3.16	Simulação para o caso 6 – Variação senoidal da freqüência	53
3.17	Estimação para a fase - caso 1	55
3.18	Estimação para a fase - caso 1 – usando conceito de fase total	57
3.19	Interpretação da fase para sinal com freqüência variante no tempo.....	58
3.20	Estimação da fase para o caso 3 analisado anteriormente.....	58
4.1	Operador δ^{-1}	65
4.2	DFIIt	65
4.3	Representação decimal de ponto fixo.....	70
4.4	Representação decimal no formato Q14.....	71
4.5	Fluxograma para soma de dois números binários com formatos Qn específicos....	73
4.6	Fluxograma para multiplicação de dois números binários com formatos Qn específicos	74
4.7	Resposta da malha quantizada em 16 bits, utilizando uma freqüência de amostragem igual a 1920 Hz	75
4.8	Resposta da malha quantizada em 16 bits, utilizando uma freqüência de amostragem igual a 7680 Hz.	76
4.9	Resposta da malha quantizada em 16 bits, utilizando uma freqüência de amostragem igual a 15000 Hz	76
4.10	Erro na quantização do coeficiente T_s	77
4.11	Erro na quantização do coeficiente T_s	78
4.12	Resposta da malha quantizada em 16 bits, utilizando uma freqüência de amostragem igual a 8.192 Hz.....	78
4.13	Resposta em freqüência e fase do filtro <i>notch</i> implementado com uma freqüência de amostragem igual a 1920 Hz.....	80
4.14	Resposta da malha E-QPLL quantizada em 16 bits, utilizando uma freqüência de amostragem igual a 1920 Hz.....	81
4.15	Resposta em freqüência e fase do filtro <i>notch</i> implementado com uma freqüência de amostragem igual a 15360 Hz	82
4.16	Resposta da malha E-QPLL quantizada em 16 bits, utilizando uma freqüência de amostragem igual a 15.360 Hz	82
4.17	Integrador no domínio δ (a) e Operação δ^{-1} (b)	84
4.18	Malha E-QPLL implementada no domínio do operador delta (δ)	85
4.19	Erro de quantização do coeficiente $\beta=C_1$	86
4.20	Erro de quantização do coeficiente β , em função do parâmetro $1/\Delta$, para uma taxa de amostragem igual a 7680 Hz	86

4.21 Resposta em frequência e fase do filtro <i>notch</i> implementado através do operador δ , utilizando uma taxa de amostragem igual a 15360 Hz.....	88
4.22 Resposta da malha ER-QPLL implementada através da Transformada γ	89
4.23 Erro na quantização dos coeficientes do filtro <i>notch</i> em função do parâmetro $1/\Delta$	90
4.24 Erro na quantização do coeficiente Δ por truncamento e arredondamento	91
4.25 Comparação das malhas com um sinal de entrada com um sinal com harmônicos.....	92
4.26 Respostas das malhas ER-QPLL e QPLL.....	93
4.27 Resposta das malhas QPLL e ER-QPLL.....	94
4.28 Amplitude estimada pelo ER-QPLL	95
4.29 Sinal de entrada e sinal sintetizado pela malha ER-QPLL.....	96
4.30 Estimação de frequência pela malha ER-QPLL.....	96

LISTA DE TABELAS

2.1	Parâmetros de desempenho da resposta transitória do PLL linear.....	15
3.1	Transitórios do filtro para diversos fatores <i>notch</i>	37
3.2	Melhoria na SNR do sinal após filtragem	39
3.3	Distúrbios no sinal de entrada	44
3.4	Erro médio quadrático obtido para diversos distúrbios e métodos de atualização do filtro <i>notch</i>	45
3.5	Casos Simulados	47
4.1	Relação de parâmetros	68
4.2	Transformadas Z e Gama	69
4.3	Formato das variáveis quantizadas da malha	72

Capítulo 1

A estimação de parâmetros de sinais elétricos

1.1 Introdução

A estimação dos parâmetros de um sinal elétrico é uma operação intrínseca em várias aplicações de sistemas de potência, tal como o monitoramento da qualidade da energia, sistemas de proteção e controle, entre outras. Técnicas variadas de processamento de sinais podem ser empregadas neste processo, dependendo das especificidades de cada aplicação, buscando-se sempre uma solução ótima para o trinômio velocidade de convergência, precisão e complexidade computacional. Pode-se, por vezes, relaxar as exigências sobre um destes fatores em detrimento das especificações colocadas para um outro; por exemplo, em proteção de sistemas de potência o fator primordial é a velocidade da estimação, podendo-se então relaxar as especificações necessárias à precisão, priorizando-se a velocidade como a característica mais importante [1]. Por outro lado, em aplicações de monitoramento, a ênfase poderá ser dada na precisão, relaxando-se as especificações concernentes à velocidade [2]. Pode-se citar como exemplo desta situação, a utilização de medidores de energia elétrica de alta precisão, geralmente empregado por consumidores de grande porte e concessionárias que necessitam medir grandes montantes de energia. Já a complexidade dos algoritmos está mais relacionada ao custo da implementação nas diversas plataformas de hardware e software disponíveis. Nesse caso, quanto menor a complexidade computacional, menor o custo total do sistema [3].

Convencionalmente, os parâmetros usuais estimados, para os sinais elétricos, são a amplitude e a fase da componente fundamental. Porém, com o advento da desregulamentação do setor elétrico, do aumento do número de cargas não lineares e da presença de geradores independentes no sistema, bem como sistemas isolados, a estimação de outros parâmetros se tornou uma exigência para a manutenção da qualidade da energia elétrica [6]. Nesta situação, novas informações, como a frequência dos sinais e os parâmetros das componentes harmônicas e inter-harmônicas (amplitude e fase), entre outros, passam a ocupar papel preponderante na análise dos sinais elétricos e necessitam ser estimados [2,4,5].

Este novo cenário trouxe consigo, porém, novas exigências e novas especificações que resultaram em condições não favoráveis para a estimação de parâmetros, quando se considera o uso de técnicas convencionais. De fato, muitas das técnicas digitais atualmente em uso nos equipamentos de proteção e medição foram desenvolvidas – e são utilizadas - baseadas na suposição que o sinal sendo processado é estacionário e de frequência constante. Esta consideração, contudo, não é válida se a estimação paramétrica está sendo efetuada em sistemas de energia cuja qualidade é baixa, fato este que pode conduzir a medidas erradas e conclusões equivocadas, ou até mesmo a funcionamentos indevidos de equipamentos de proteção e controle. É fato conhecido que medidores de qualidade de energia submetidos a diferentes distúrbios no sinal de entrada podem indicar sérios desvios nas medições efetuadas.

Os principais distúrbios na qualidade da energia estão relacionados com as distorções provocadas na forma de onda de tensão; esta onda, supostamente, deveria se comportar como uma senóide com amplitude, fase e frequência constantes, situação esta que não ocorre na prática. Os distúrbios mais comuns na forma de onda da tensão são [7,8]:

- Variação na amplitude do sinal, que pode durar de $\frac{1}{2}$ ciclo até alguns segundos. De fato, os “afundamentos de tensão” (*SAGs*) são os tipos de distúrbios mais comuns em qualidade de energia;
- Presença de componentes transitórias em decorrência de chaveamentos normais na rede (chaveamento de capacitores e outras cargas) ou à faltas no sistema;
- Variação na frequência do sistema devido às mudanças da rede e entrada e saída de grandes cargas ou unidades geradoras;

- Presença de componentes harmônicas e inter-harmônicas devido à existência de cargas eletrônicas chaveadas;
- Modulação da amplitude da componente fundamental (*"flicker"*) ocasionada pela presença de fornos a arcos e geradores eólicos, entre outros.

Assim, é comum utilizar-se, na literatura, o termo “baixa qualidade da energia”, quando o sinal de tensão apresenta algum tipo de distorção (amplitude, fase e frequência) ou quando a ele se encontram adicionadas outras componentes senoidais.

Esta realidade exige o emprego de técnicas mais complexas e elaboradas para estimação dos parâmetros dos sinais elétricos, que consigam efetuar medições confiáveis e garantir a confiabilidade dos dados adquiridos, a despeito de todos os distúrbios que podem estar contidos nos sinais.

Dentre estas diversas técnicas de estimação de parâmetros, as baseadas na Transformada Discreta de Fourier - DFT (*"Discrete Fourier Transform"*) são as mais empregadas. A DFT aparece em aplicações de proteção, controle e monitoramento, seja na sua forma de algoritmo rápido (*"Fast Fourier Transform"* - FFT) ou em forma de processamento recursivo [9,10]. Embora simples em sua estrutura, os algoritmos baseados em DFT requerem atenção quando aplicados a sinais não-estacionários, ou nos casos de amostragem não síncrona e de presença de inter-harmônicos. Os trabalhos [2] e [4] mostram a situações em que estes algoritmos apresentam erros consideráveis quando usados em sistemas com baixa qualidade de energia.

O Filtro de Kalman é outra técnica que tem sido extensivamente utilizada na estimação dos parâmetros de sinais elétricos. Enquanto os primeiros algoritmos de Filtros de Kalman consideravam a frequência constante e a estimavam apenas a amplitude e fase dos sinais, os algoritmos atuais estimam também a frequência do sinal de entrada. A grande restrição da aplicação desta técnica está no elevado esforço computacional requerido quando o algoritmo do filtro utilizado deve incluir a estimação de componentes harmônicas, uma vez que o processamento do Filtro de Kalman requer a inversão de matrizes de ordem elevada [31].

Outra técnica para a estimação de parâmetros é o uso de estruturas “*Phase-Locked Loop – PLL*”. As aplicações de PLL são amplamente conhecidas em sistemas de telecomunicações e eletrônica de potência. Nestas áreas, eles são utilizados como dispositivos rastreadores e extratores de uma componente de frequência. A utilização do PLL como estimador de parâmetros é, entretanto, bem recente [2,4].

Dentre as diversas estruturas de PLL sugeridas recentemente para a estimação de parâmetros, destaca-se a estrutura “*Quadrature Phase-Locked Loop – QPLL*” [11]. Esta estrutura estima a frequência e as componentes em fase e em quadratura da componente fundamental do sinal de entrada. Essas componentes estão relacionadas aos termos cossenoidal e senoidal, respectivamente, da componente fundamental do sinal de entrada. A amplitude e a fase do sinal são estimadas utilizando operações adicionais. Resultados apresentados na literatura [11,13] mostram que esta estrutura é robusta a variações dos parâmetros internos, sendo capaz de rastrear variações de frequência abruptas com velocidade razoável, bem como realizar estimações precisas para sinais de tensão contendo diversos tipos de distúrbios de qualidade de energia. Baseado nestas características apresentadas por este algoritmo, elegeu-se esta estrutura de estimação como a estrutura base para desenvolvimento deste trabalho.

1.2 Objetivo do Trabalho

O objetivo do presente trabalho foi a síntese de um algoritmo, baseado em uma estrutura QPLL modificada, para estimação de parâmetros da componente fundamental de sinais elétricos. As modificações propostas na estrutura do QPLL convencional foram a inclusão de um filtro passa-faixa adaptativo na entrada do QPLL e a implementação da estrutura resultante utilizando o operador delta (δ) [10].

O filtro passa-faixa introduzido na estrutura é obtido a partir de um filtro notch parametrizado de segunda ordem. Dois parâmetros, α e β , controlam o desempenho do filtro passa-banda. O parâmetro α controla a largura da faixa de passagem do filtro: quanto mais próximo da unidade for este parâmetro, mais seletivo ou sintonizado será o filtro e mais lenta será a sua resposta dinâmica. Já o parâmetro β controla a frequência central do filtro, sendo sua adaptabilidade obtida a partir de estimação da frequência proveniente do estimador QPLL, permitindo que o filtro possa ajustar a frequência

central para a frequência da componente fundamental, possibilitando que a estrutura possa operar em sistemas onde a frequência é variante com o tempo.

A inclusão do filtro passa-faixa na entrada do estimador melhora a precisão da estimação, basicamente, pelos seguintes motivos:

- Melhoria da relação sinal ruído (S/NR) na entrada do estimador;
- Atenuação das componentes de frequência fora da frequência central do filtro (harmônicas e inter-harmônicas).

Há que se ressaltar, contudo, que a inclusão do filtro aumenta o tempo de convergência do algoritmo devido ao seu comportamento transitório em situações de distúrbios. Entretanto, de acordo com simulações computacionais que serão apresentadas no Capítulo 4, é visto que a malha QPLL modificada apresenta uma resposta dinâmica mais rápida, em relação à malha QPLL original, se ambas forem ajustadas para apresentarem uma mesma rejeição a distúrbios em regime permanente.

Quanto mais próximo da unidade for o parâmetro α , mais próximo do círculo unitário estará o pólo do filtro notch, aumentando assim a sensibilidade dos coeficientes do filtro, o que traz dificuldade para sua implementação utilizando-se aritmética de ponto fixo [12]. Para contornar esta dificuldade, optou-se em utilizar a transformada Gama (γ) (ou operador delta (δ)) [12,14] para a implementação do filtro e do algoritmo QPLL, em detrimento da utilização da conhecida transformada Z. Os resultados alcançados, e discutidos ao longo deste trabalho, mostram que a utilização do operador delta (δ) tornou a estrutura altamente robusta à implementação em sistemas de precisão limitada de, no mínimo, 16 bits, permitindo sua implementação em processadores DSP de 16 bits.

Os resultados obtidos e as análises efetuadas permitem selecionar algumas contribuições deste trabalho à área de processamento de sinais, considerando-se especialmente sua utilização para os trabalhos no campo da qualidade da energia elétrica, dentre as quais podem ser destacadas:

- Utilização de filtro passa-faixa adaptativo para a melhoria do sinal de entrada no estimador QPLL;

- Utilização do operador delta (δ) para obtenção de estrutura robusta para implementação em ponto fixo;
- Implementação do algoritmo em processador digital de sinais da família TMS320F2800.

1.3 Divisão do Trabalho

O Capítulo 2 apresenta uma introdução às estruturas PLL. Uma breve discussão das diversas estruturas é apresentada com destaque para a estrutura do QPLL. Neste capítulo é apresentada a derivação das equações digitais do QPLL e resultados de simulação, implementado em linguagem Matlab, são apresentados mostrando o desempenho do estimador na presença de diversos distúrbios de qualidade de energia.

O Capítulo 3 apresenta estrutura E-QPLL (“*Enhanced QPLL*”). Esta estrutura corresponde à inclusão de um filtro passa-faixa adaptativo na entrada do estimador. O filtro passa-faixa é obtido a partir do filtro notch de segunda ordem cujo parâmetro relativo à frequência central é adaptado utilizando-se a frequência estimada pelo algoritmo. Duas estratégias para a adaptação da frequência central do filtro são discutidas. É mostrada neste capítulo a influência do filtro notch no tempo de convergência do estimador bem como a melhoria na relação sinal ruído do sinal de entrada do estimador. Finalmente alguns resultados de simulação em ponto-flutuante são apresentados e comparado com os gerados pela estrutura QPLL convencional.

O Capítulo 4 trata da implementação do E-QPLL em aritmética de ponto fixo. Inicialmente é apresentada uma revisão da transformada Gama (γ) e do operador delta (δ). A seguir mostra-se que o filtro notch, quando implementado com o operador delta (δ), apresenta uma estrutura mais robusta em relação à quantização dos coeficientes que a estrutura equivalente no domínio Z. Mostra-se que o QPLL também apresenta vantagens quando implementado com o mesmo operador. Segue-se a implementação das estruturas em aritmética de ponto fixo, com alguns resultados práticos implementados em processadores DSP da família TMS320F2800. Resultados comparativos entre o E-QPLL e o QPLL são apresentados.

O Capítulo 5 apresenta as conclusões gerais do trabalho.

Finalmente, no Capítulo 6, são anexadas as contribuições técnicas deste trabalho, que resultaram em 3 artigos técnicos, dois em congressos internacionais e um em congresso nacional.

Capítulo 2

Phase-Locked-Loops - PLLs

2.1 Introdução

Uma estrutura PLL (*Phased-Locked-Loop*) consiste, basicamente, em uma malha de controle realimentada cujo principal objetivo é a sintetização de uma senóide, geralmente de amplitude unitária, com frequência idêntica à frequência da componente fundamental de um sinal de entrada qualquer. Será efetuada, neste capítulo, uma revisão bibliográfica das malhas PLL. Será efetuada a descrição de um modelo linear de um PLL, conhecido como LPLL (*Linear Phased-Locked-Loop*) e suas respostas a uma entrada em degrau, bem como em frequência, serão analisadas.

Será desenvolvido, na seqüência, um modelo discreto de um PLL linear, com análise do processo de transição entre os espaços S e Z, ou seja, o mapeamento dos pólos do modelo contínuo para o modelo discreto. Será mostrado, logo a seguir, o exemplo completo do desenvolvimento de um PLL discreto, onde os seus parâmetros serão calculados e o seu desempenho será avaliado. Será efetuado, finalmente, o desenvolvimento de uma malha de *Quadrature Phase-Locked Loop – QPLL*. A utilização desta malha, como ressaltado anteriormente, torna possível realizar a aquisição de todos os parâmetros (amplitude, fase e frequência) da componente fundamental de um sinal de entrada qualquer.

2.2 PLL analógicos

As principais aplicações de malhas PLL são encontradas nos sistemas de telecomunicações, onde sua utilização é direcionada, basicamente, para as tarefas de modulação e/ou demodulação de sinais, bem como a sintetização de senóides com elevado grau de precisão [21]. De forma complementar às aplicações referidas, as estruturas PLL são também muito utilizadas na área de sistemas de potência, onde constituem parte integrante de procedimentos como controle de inversores de frequência e máquinas conectados à rede elétrica, detecção da frequência fundamental de um barramento elétrico, detecção e ou medição de harmônicas ou a leitura do fasor da componente fundamental de sinais elétricos, entre outros [16, 13, 2,11].

A Figura 2.1 mostra o esquema básico de uma estrutura de um PLL.

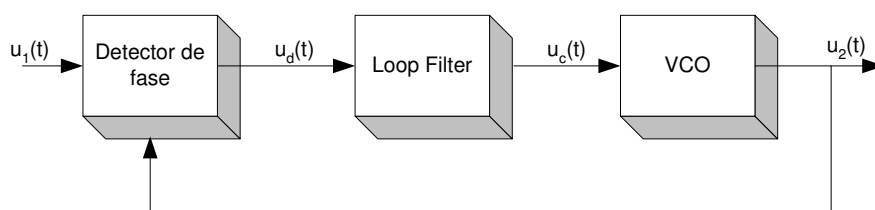


Figura 2.1 - Modelo simplificado de uma malha PLL

De acordo com a Figura 2.1, pode-se verificar que uma malha PLL é composta por basicamente três estruturas: um detector de fase, um filtro passa-baixas e um VCO (“*Voltage-controlled-oscillator*”). O funcionamento da malha PLL pode ser resumido da seguinte maneira: de acordo com a Figura 2.1, o detector de fase irá gerar um sinal de erro proporcional à diferença de fase entre a senóide gerada internamente pelo PLL e a componente fundamental do sinal de entrada. Esse sinal de erro é tipicamente uma correção de frequência, “ $\Delta\omega$ ”. Quanto maior a diferença de fase entre o sinal de entrada e a senóide gerada pelo PLL, maior será a correção de frequência. Em seguida, o sinal de correção de frequência passa pelo filtro passa-baixas (*Loop Filter*), onde são removidos os distúrbios indesejáveis que possam afetar o comportamento do VCO. Logo em seguida, o sinal $u_d(t)$, após ser filtrado pelo filtro passa-baixas, irá controlar a frequência do sinal gerado pelo VCO, que é, basicamente, um oscilador controlado por tensão e que efetua a síntese de uma senóide cuja frequência é diretamente proporcional a um sinal de

controle. Deste modo, após um tempo de *sincronização*, o PLL irá sintetizar, naturalmente, uma senóide sincronizada em fase e frequência com o sinal de entrada [21].

Dependendo do tipo do PLL utilizado, podem-se ter sinais de saída com fases idênticas à do sinal de entrada ou com fases deslocadas por uma constante qualquer. Assim, além de fornecer uma informação sobre a frequência do sinal de entrada, o PLL também pode ser utilizado para rastrear a fase do mesmo.

O desenvolvimento de estruturas PLL teve início por volta dos anos 30, quando a primeira estrutura PLL foi utilizada para sincronização dos sinais horizontal e vertical de um aparelho de televisão [23]. Nessa época, o uso de estruturas PLL era pouco difundido, principalmente por possuírem custos elevados. Entretanto, com a invenção dos circuitos analógicos integrados, tornou-se possível o encapsulamento de uma estrutura de PLL em um único “*chip*”. Com o advento desta nova tecnologia, o uso de estruturas de PLL se tornou mais difundido, havendo um grande desenvolvimento do mesmo e de suas aplicações. Mais recentemente, a popularização de microprocessadores e a conseqüente diminuição de seu preço, despertaram o interesse para o desenvolvimento de estruturas de PLL digitais, que se tornou uma área de pesquisas e desenvolvimento prioritários. A partir de 1980, o interesse na área se consolida, impulsionando os trabalhos de desenvolvimento e resultando, deste movimento, a síntese e implementação de diversas estruturas de PLL digitais [15,17].

As estruturas de PLL atualmente existentes podem ser divididas em dois grandes grupos: os PLL’s analógicos e os PLL’s digitais. Pode-se afirmar, em uma comparação sumária, que a principal desvantagem dos PLL’s analógicos em relação aos digitais é a necessidade da utilização de componentes externos para o ajuste da malha, o que sempre acarreta problemas e complexidades adicionais. Dentre estes problemas e complexidades podem ser destacados, por exemplo, a saturação dos componentes, erros de “*offset*”, envelhecimento dos componentes externos e a necessidade de ajuste inicial da estrutura do PLL analógico. Desta forma, as estruturas digitais de PLLs consolidam-se cada vez mais, aumentando sua aplicabilidade e possibilitando uma variedade de alterações estruturais, que resultam em resultados promissores e novas aplicabilidades, em um círculo tecnológico altamente virtuoso.

2.3 Comportamento dinâmico de uma malha de PLL

Para efetuar uma análise do comportamento dinâmico de uma malha de PLL é comum utilizar-se um modelo linear, construído sob a premissa que a frequência do sinal de entrada é praticamente igual à frequência da senóide sintetizada pelo PLL. Com base nessa suposição, será efetuada, a seguir, uma descrição detalhada de cada componente de um modelo linear de uma malha de PLL digital.

2.3.1 Detector de fase

A principal função do detector de fase é a geração de um sinal de erro proporcional à diferença de fase entre a senóide de entrada e a senóide gerada internamente pelo PLL. O detector de fase, em PLL digitais, é composto tipicamente por um multiplicador. Assim, considerando-se o sinal de entrada como $u(t) = A\text{sen}(\omega_1 t + \theta_1)$ e o sinal gerado pelo PLL como $v(t) = B\text{sen}(\omega_2 t + \theta_2)$, a saída do multiplicador será expressa por:

$$v(t)u(t) = AB\text{sen}(\omega_1 t + \theta_1)\text{sen}(\omega_2 t + \theta_2) \quad (2.1)$$

Simplificando (2.1), sob a consideração que $\omega_1 \cong \omega_2$, chega-se à:

$$v(t)u(t) = 2AB\text{sen}(2\omega t + \theta_e) + 2AB\text{sen}(\theta_e), \quad (2.2)$$

onde $\theta_e = \theta_2 - \theta_1$.

Quando a diferença de fase entre os sinais for reduzida, pode-se considerar o sinal de erro como expressão de uma relação linear da diferença de fase entre os sinais medidos, ou seja, $k_d\text{sen}(\theta_e) \cong k_d\theta_e$. Esta suposição pode ser considerada verdadeira durante o período de sincronização do PLL [22].

Dessa forma, ignorando-se os termos de alta frequência, verifica-se que o detector de fase irá fornecer um sinal contínuo proporcional à diferença de fases entre o sinal de entrada e o sinal gerado internamente pelo PLL. O modelo linear para o detector de fase está representado na Figura 2.2.

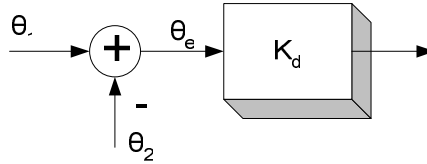


Figura 2.2 - Modelo linear do detector de fase

2.3.2 Filtro passa-baixas

De acordo com o modelo linear adotado para o detector de fase, nota-se que sua saída é composta por uma senóide e um termo contínuo proporcional à diferença de fase entre a senóide de entrada e a senóide sintetizada pelo PLL. Como o VCO irá sintetizar uma senóide de frequência proporcional a um sinal de controle, tem-se que é necessária a remoção de sinais de alta frequência presentes na saída do detector de fase. Assim, torna-se necessário o uso de um filtro passa-baixas na entrada do VCO.

Geralmente, em projetos de malhas de PLL, são utilizados filtros passa-baixas do tipo Butterworth de ordem igual a 1 ou superior [19]. A ordem do filtro passa-baixas e a sua banda de passagem são fatores cruciais que influenciam o comportamento dinâmico da malha de PLL e que serão analisados ainda nesse capítulo.

2.3.3 VCO (“Voltage Controlled Oscillator”)

A função do VCO é efetuar a síntese de uma senóide com frequência diretamente proporcional a um sinal de controle. Assim, pode-se afirmar que o VCO, quando excitado por uma entrada do controle $u_d(t)$, irá sintetizar um sinal igual a $G_{VCO} \text{sen}(k_v u_d(t)t)$, onde k_v é uma constante de proporcionalidade e G_{VCO} é um fator de ponderação, geralmente unitário. Isto permite concluir que a fase, ou o argumento da senóide sintetizada pelo PLL, será a integral do sinal de controle do VCO. O modelo linear do VCO, que fornece a fase do sinal sintetizado, está representado na Figura 2.3.

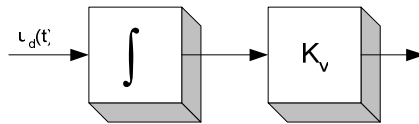


Figura 2.3 - Modelo linear do VCO

É comum utilizar-se uma condição de contorno para o sinal de controle do VCO com a finalidade de introduzir melhorias na resposta dinâmica do PLL. Utiliza-se como condição de contorno, tipicamente, um sinal de referência igual à frequência de oscilação central do PLL, ω_0 . Assim, tem-se que o modelo linear para o VCO pode ser representado também através da Figura 2.4, onde foi incluída uma condição de contorno igual à ω_0 .

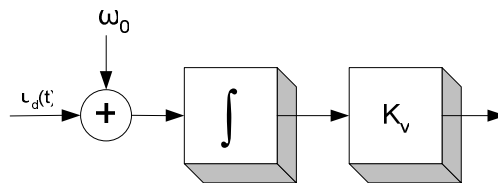


Figura 2.4 - Modelo linear do VCO incluindo uma condição de contorno

2.3.4 Análise da Malha de Controle do PLL

O modelo linear completo do PLL pode ser representado através da malha de controle da Figura 2.5.

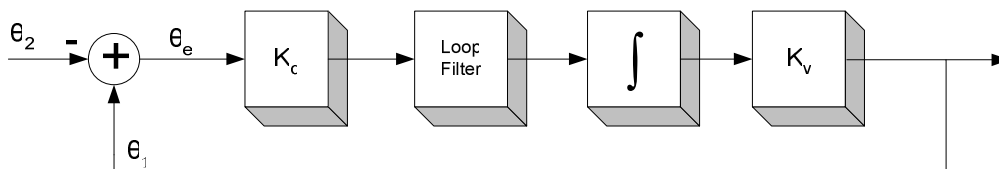


Figura 2.5 - Modelo linear do PLL

Para efeito de simplificação foi incluído, na malha do PLL, um filtro passa-baixas de primeira ordem e as condições de contorno foram consideradas nulas. A função de transferência do filtro passa-baixas de um pólo será então dada por [21]:

$$H_{FPB} = \frac{G_{1p}}{s + G_{1p}} \quad (2.3)$$

A função de transferência de malha fechada do modelo do PLL linear será dada por:

$$H_{mf} = \frac{H_{ma}}{1 + H_{ma}}, \quad (2.4)$$

onde a função de transferência de malha aberta é dada por $H_{ma} = \frac{K_d G_{1p} K_v}{s(G_{1p} + s)}$.

Desse modo, têm-se que:

$$H_{mf} = \frac{H_{ma}}{1 + H_{ma}} = \frac{K_d K_v G_{1p}}{s^2 + G_{1p}s + K_d K_v G_{1p}}. \quad (2.5)$$

Baseado na função de transferência do modelo linear do PLL nota-se que o mesmo é um sistema de segunda-ordem. Desse modo, a sua função de transferência também pode ser representada da seguinte forma:

$$H(s) = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2}, \quad (2.6)$$

onde ω_n é a frequência natural de oscilação e ξ é a taxa de amortecimento.

Comparando (2.5) e (2.6), chega-se à conclusão que:

$$\omega_n = \sqrt{K_d G_{1p} K_v}, \quad (2.7)$$

$$\xi = \frac{\sqrt{K_d G_{1p} K_v}}{2K_v K_d}. \quad (2.8)$$

O polinômio característico da função de transferência do modelo linear do PLL será expresso por:

$$\Delta(s) = s^2 + 2\xi\omega_n s + \omega_n^2. \quad (2.9)$$

As raízes da equação característica serão então:

$$\begin{aligned} s_0 &= -\xi\omega_n + j\omega_n\sqrt{1-\xi^2} = -\tau + j\omega, \\ s_1 &= -\xi\omega_n - j\omega_n\sqrt{1-\xi^2} = -\tau - j\omega, \end{aligned} \quad (2.10)$$

onde w é a frequência de amortecimento e τ é o fator de amortecimento.

Para assegurar a estabilidade da estrutura PLL, deve-se garantir que os pólos, que são as raízes da equação característica da função de transferência de malha fechada do PLL, estejam localizados no semi-plano esquerdo do plano complexo. Isso pode ser assegurado desde que:

$$\xi > 0. \quad (2.11)$$

Ou seja:

$$\frac{\sqrt{K_d K_v G_{1p}}}{2K_v K_d} > 0 \quad (2.12)$$

Deste modo, o ajuste dos ganhos do detector de fase e do VCO, bem como os coeficientes do filtro passa-baixas, devem ser sintonizados de modo a assegurar a estabilidade em malha fechada do PLL.

O desempenho da resposta transitória do PLL pode ser avaliado a partir de sua resposta ao degrau, pois uma entrada em degrau na malha de PLL simboliza uma variação brusca de fase, “ $\Delta\theta$ ”. A Tabela 2.1 indica os parâmetros utilizados na análise da resposta ao degrau da malha de PLL, expressos em função do coeficiente de amortecimento e da frequência natural de oscilação da malha de PLL [19].

Tabela 2.2 - Parâmetros de desempenho da resposta transitória do PLL linear

Parâmetro	Expressão
Tempo de acomodação	$t_s = \frac{4}{\xi \cdot w_n}$
Tempo máximo de sobre-sinal	$t_{\max} = \frac{\pi}{w_n \sqrt{1 - \xi^2}}$
Máximo sobre sinal	$M = 1 + e^{-\pi\xi / \sqrt{1 - \xi^2}}$
Máximo sobre sinal (%)	$M_{pct} = 100 \cdot e^{-\pi\xi / \sqrt{1 - \xi^2}}$

A frequência de corte da malha de PLL linear será dada por:

$$|H_{mf}(w_c)| = \frac{\sqrt{2}}{2}, \quad (2.13)$$

o que permite concluir que:

$$w_c = \sqrt{\frac{2w_n^4 - w_n^2}{-1 + 4\xi^2 w_n^2}}. \quad (2.14)$$

Comparando a Equação 2.14 com a Tabela 2.1, verifica-se que, quanto menor a frequência de corte do PLL, maior será a sua imunidade a ruídos presentes no sinal de entrada. Entretanto, de acordo com os parâmetros da Tabela 2.1, isto acarretaria uma resposta transitória mais lenta, o que provocaria um tempo de sincronização mais lento do PLL. Caso a frequência de corte do PLL seja ajustada em um valor mais alto, a sua imunidade a ruídos será diminuída. Em contrapartida, a sua resposta transitória será mais rápida, o que acarretará um tempo de sincronização mais baixo. Desse modo, nota-se que os ajustes dos parâmetros da malha do PLL devem ser cuidadosamente escolhidos, de modo a se obter um desempenho satisfatório do PLL para cada aplicação de interesse.

2.4 PLL digitais

A utilização de estruturas PLL digitais, conforme discutido em [15], tem ganhado força nos últimos 20 anos. Nesse contexto, destaca-se um sub-grupo de PLL digitais, os SPLL's, ou “*softwares-PLL*”. Esse tipo de malha de PLL encontra aplicações principalmente em microprocessadores, aonde a sua implementação é efetuada através de equações a diferenças, representando uma função de transferência discreta.

O ajuste dos parâmetros de estruturas de SPLL's requer a discretização da malha PLL representada na Figura 2.1, que pode ser expresso, no espaço Z, pela estrutura apresentada na Figura 2.6.

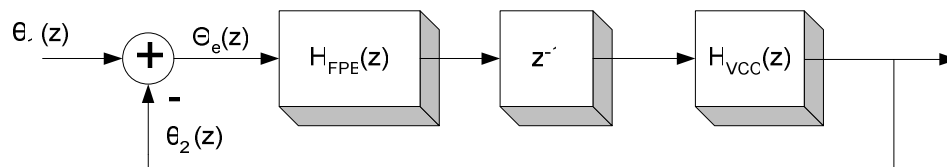


Figura 2.6 - Modelo discretizado de um PLL

A função de transferência de malha fechada do PLL digital será dada pela Equação 2.15,

$$H(z) = \frac{acz - c}{z^2 + (ac - 2)z + (1 - c)}, \quad (2.15)$$

onde: $H_{FPB} = \frac{az - 1}{z - 1}$ é a função de transferência do filtro passa-baixas;

$H_{VCO} = \frac{cz}{z - 1}$ é a função do VCO e

z^{-1} representa o atraso entre a conversão analógico-digital e a saída do PLL.

A função de transferência representada em (2.15) pode ser representada da seguinte forma:

$$H(z) = \frac{N(z)}{\Delta(z)} = \frac{N(z)}{(z - z_0)(z - z_1)}, \quad (2.16)$$

onde $\Delta(z)$ é a equação característica da função de transferência discreta e z_1 e z_2 são os pólos do PLL digital no domínio discreto.

A equação característica do PLL digital pode ser representada da seguinte forma:

$$\Delta(z) = z^2 + G_1z + G_0 \quad (2.17)$$

onde: $G_1 = -(z_1 + z_0)$

$G_0 = z_1 \cdot z_0$

Os pólos da malha de PLL no domínio discreto são obtidos através da seguinte relação:

$$\begin{aligned} z_0 &= e^{S_0 T_s} = e^{(-\zeta w_n T_s + j w_n T_s \sqrt{1 - \zeta^2})}, \\ z_1 &= e^{S_1 T_s} = e^{(-\zeta w_n T_s - j w_n T_s \sqrt{1 - \zeta^2})}, \end{aligned} \quad (2.18)$$

onde S_0 e S_1 são os pólos da função de transferência do PLL contínuo e T_s é o período de amostragem utilizado pelo modelo discreto.

Finalmente, os coeficientes G_0 e G_1 serão dados por:

$$\begin{aligned} G_0 &= e^{-2\zeta w_n T_s} \\ G_1 &= -2e^{-\zeta w_n T_s} \cos(w_n T_s \sqrt{1 - \zeta^2}). \end{aligned} \quad (2.19)$$

Para assegurar a estabilidade da malha PLL digital, os pólos da função de transferência discreta devem estar localizados no interior do círculo unitário $|z| \leq 1$. Isto significa dizer que $|z_1| \leq 1$ e $|z_2| \geq 1$. Desse modo, como a localização dos pólos da

função de transferência do PLL digital são dependentes da taxa de amostragem utilizada, deve-se assegurar a utilização de uma frequência de amostragem adequada.

2.4.1 Desenvolvimento de uma estrutura PLL inteiramente digital

A Figura 2.7 mostra uma estrutura de PLL inteiramente digital, conforme proposto por [18]. Notar que todas as estruturas desse PLL são discretas, o que permite a sua implementação em microprocessadores através da utilização de equações a diferenças.

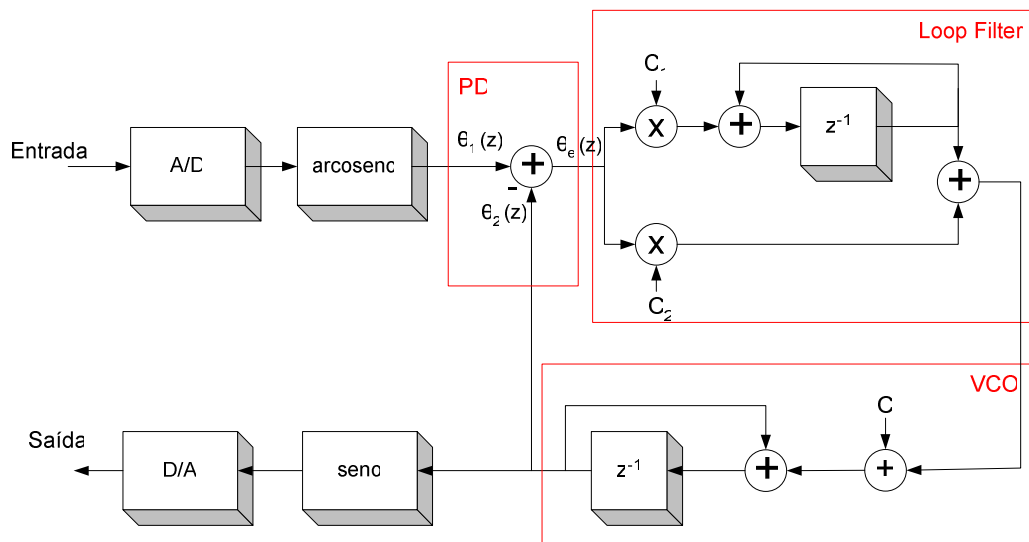


Figura 2.7 - PLL digital

O funcionamento do PLL da Figura 2.7 pode ser resumido da seguinte maneira: a fase do sinal de entrada é detectada pelo bloco “arco-seno”. Para assegurar o correto funcionamento desse bloco, nesta estrutura específica, é necessário que a amplitude do sinal de entrada seja igual à unidade. Em seguida, a fase detectada do sinal de entrada é subtraída da fase da senoide sintetizada pelo VCO da malha de PLL. Esse sinal de erro é filtrado por um filtro passa-baixas de 1ª. ordem. Em seguida, o sinal de erro é integrado pelo VCO para gerar a fase da senoide que será sintetizada pelo PLL. Os parâmetros de ajuste do PLL são os coeficientes do filtro passa-baixas “ C_1 ” e “ C_2 ” e a frequência central do PLL, “ C ”.

A função de transferência do PLL digital é dada pela equação:

$$H(z) = \frac{\phi(z)}{\theta(z)} = \frac{C_2(z-1) + C_1}{(z-1)^2 + C_2(z-1) + C_1}; \quad (2.20)$$

A equação característica do PLL é dada por:

$$\Delta(z) = z^2 + (-2 + C_2)z + 1 + C_1 - C_2; \quad (2.21)$$

Comparando a Equação 2.21 com a 2.17, os coeficientes C_1 e C_2 serão dados por:

$$\begin{aligned} G_1 &= -2 + C_2 \\ G_0 &= 1 + C_1 - C_2; \end{aligned} \quad (2.22)$$

Desse modo:

$$\begin{aligned} 1 + C_1 - C_2 &= e^{-2\xi w_n T_s} \\ -2 + C_2 &= -2e^{-\xi w_n T_s} \cdot \cos(w_n T_s \sqrt{1 - \xi^2}); \end{aligned} \quad (2.23)$$

Finalmente:

$$\begin{aligned} C_2 &= 2 - 2e^{-\xi w_n T_s} \cdot \cos(w_n T_s \sqrt{1 - \xi^2}) \\ C_1 &= e^{-2\xi w_n T_s} + C_2 - 1. \end{aligned} \quad (2.24)$$

2.4.2 Ajuste dos coeficientes da malha de PLL digital

O ajuste dos parâmetros C_1 e C_2 deve ser efetuado considerando-se a aplicação de interesse da malha de PLL. Para um sinal de entrada com uma baixa relação sinal/ruído, a frequência natural de oscilação do PLL deve ser ajustada em um baixo valor, de modo que o PLL filtre os ruídos presentes no sinal de entrada. Entretanto, nota-se que isto acarretaria uma resposta mais lenta da malha de PLL. Caso o sinal de entrada possua uma relação sinal/ruído elevada, recomenda-se que o ajuste da frequência natural de oscilação do PLL seja maior. Desse modo, obtém-se um menor tempo de sincronização da malha PLL.

Considerando-se uma aplicação em que a malha de PLL digital deve realizar a aquisição do argumento da tensão de um barramento elétrico de 60 Hz, que possui uma relação sinal/ruído baixa, os seguintes parâmetros foram utilizados:

$$T_s = 1/15360s$$

$$\xi = 0,707$$

$$w_n = 200rad / s.$$

Notar que foi definida uma frequência natural com um baixo valor, com a finalidade de filtragem dos ruídos presentes no sinal de entrada. Os coeficientes do filtro passa-baixas serão dados por:

$$C_1 = 1.6799e-004$$

$$C_2 = 0.0184.$$

A resposta ao degrau da malha de PLL pode ser observada na Figura 2.8. A resposta em frequência está representada na Figura 2.9.

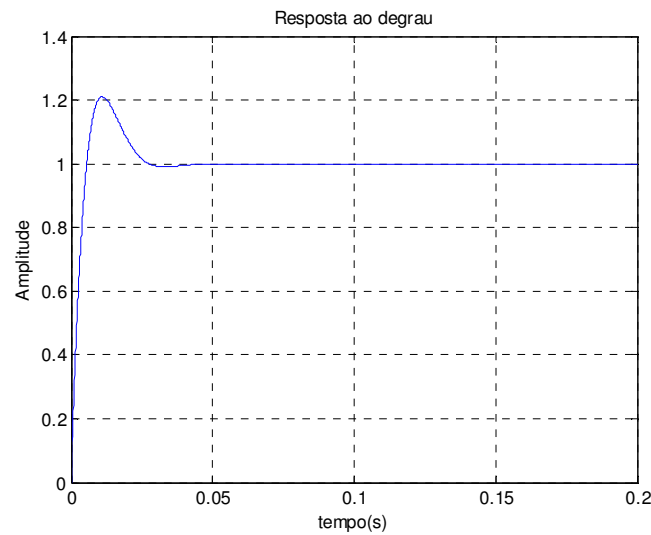


Figura 2.8 - Resposta ao degrau da malha de PLL

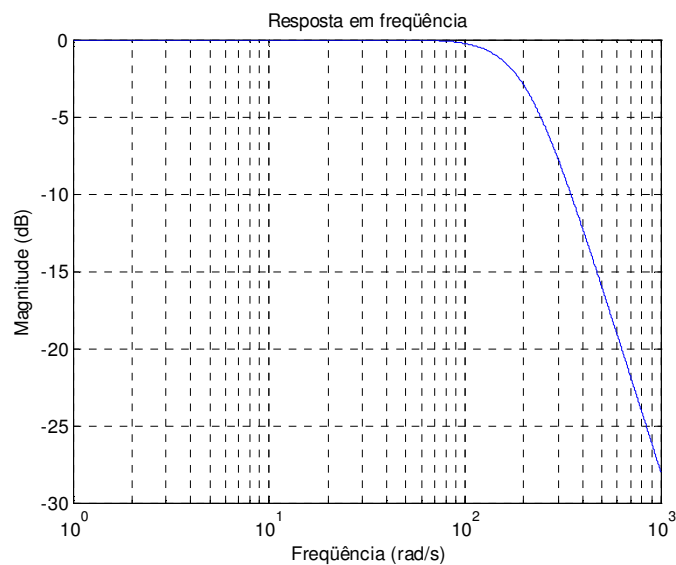


Figura 2.9 - Resposta em frequência da malha de PLL

Os parâmetros serão agora ajustados buscando obter-se um menor tempo de sincronização, mas proporcionando menor imunidade a ruídos. Utilizando-se os fatores:

$$T_s = 1/15360s$$

$$\xi = 0,707$$

$$w_n = 300rad / s.$$

os coeficientes do filtro serão dados por:

$$C_1 = 3.7624e-004$$

$$C_2 = 0.0276.$$

As respostas (tempo e frequência) estão nas Figuras 2.10 e 2.11, respectivamente.

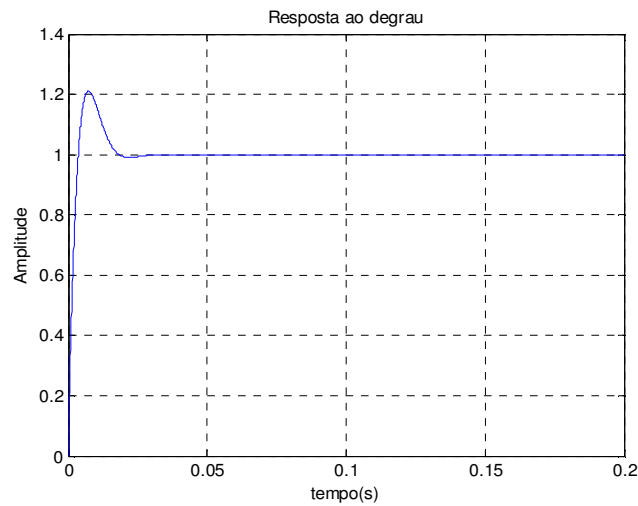


Figura 2.10 - Resposta ao degrau da malha de PLL

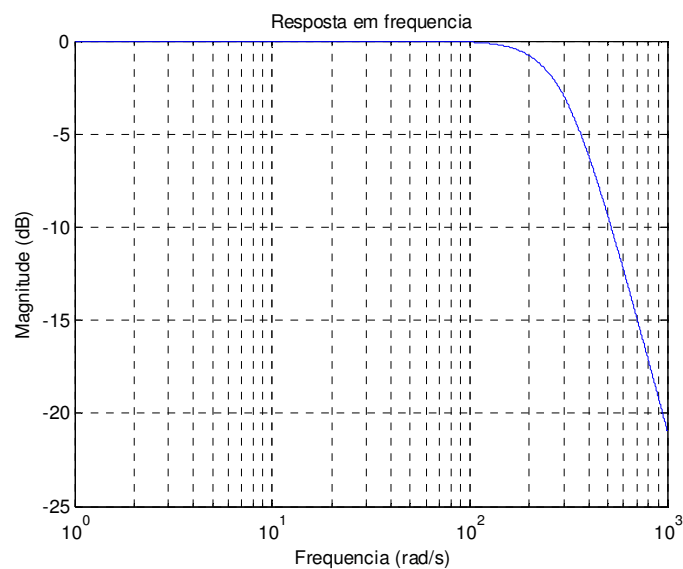


Figura 2.11 - Resposta em frequência da malha de PLL

Notar que nos dois casos apresentados, a resposta em frequência da malha PLL apresentou o comportamento de um filtro passa-baixas, indicando que a malha PLL atenua ruídos presentes no sinal de entrada.

Ao se utilizar uma frequência natural de oscilação igual a 200 rad/s, a atenuação do PLL, na frequência de 120 Hz, foi igual a 23 db. Entretanto, ao se utilizar uma frequência natural de oscilação igual a 300 rad/s, a atenuação se reduziu para 16 dB, indicando que a imunidade a ruídos presentes no sinal de entrada foi diminuída. Em contrapartida, ao se analisar a resposta ao degrau do PLL para as duas configurações, verificou-se que a primeira configuração, apesar de apresentar uma maior imunidade a ruídos, possui uma resposta transitória mais lenta.

2.5 “*Quadrature Phase-Locked-Loop*”

A malha PLL demonstrada anteriormente é baseada na função *arco-seno*, que fornece diretamente a fase do sinal de entrada e o bloco somador, que se encontra ligado em série com o bloco *arco-seno* e que fornece a diferença de fase entre o sinal de entrada e a senóide sintetizada pela malha PLL. Esta estrutura PLL demonstrada, entretanto, só é factível quando o sinal de entrada possuir amplitude unitária. Essa seção descreve uma malha PLL proposta em [11], também conhecida como QPLL (“*Quadrature-Phase-Locked-Loop*”), em função da malha realizar a aquisição das componentes em fase e em quadratura da componente fundamental de um sinal de entrada qualquer. Essa estrutura é capaz de detectar a frequência, fase e amplitude da componente fundamental de entrada de um sinal qualquer.

Considerar um sinal periódico $u(t)$, composto por harmônicas e ruídos aditivos, tipicamente presentes em sistemas de potência:

$$u(t) = u_1(t) + \sum_{i \neq 1} A_i \text{sen}(i\omega_0 t + \varphi_i) + n(t), \quad (2.25)$$

onde $u_1(t)$ é a sua componente fundamental e $n(t)$ representa um ruído aditivo.

O problema consiste em se determinar os parâmetros (amplitude, fase e frequência) da componente fundamental do sinal $u(t)$, dada por:

$$u_1(t) = A \sin(\omega t + \varphi) = A \sin(\psi(t)), \quad (2.26)$$

onde A é o valor de pico da senóide, ω é a sua frequência (rad/s), φ é a sua fase (rad) e $\psi(t)$ é o argumento total da senóide.

Assim, através da utilização de um processo que permita determinar todos os parâmetros desejados da componente fundamental de entrada, pode-se sintetizar uma senóide *sincronizada* com a componente fundamental de um sinal de entrada, através da utilização da Equação 2.26. Essa é a idéia básica de estruturas PLL.

Considere um conjunto M formado por todos os sinais senoidais periódicos em função do tempo e em função de um conjunto de parâmetros, também em função do tempo, denominados “ $\theta(\mathbf{t})$ ”. Desse modo, o conjunto M será dado por:

$$M = \{y(t, \theta(t)), t \in \mathfrak{R}, y : \text{senóide}\}, \quad (2.27)$$

onde:

$$\theta(t) = [\theta_1(t) \ \theta_2(t) \ \dots \ \theta_n(t)]^T \quad (2.28)$$

representa o vetor de parâmetros, o qual resulta no espaço de parâmetros:

$$\Theta(t) = \{[\theta_1(t) \ \theta_2(t) \ \dots \ \theta_n(t)], |\theta_i \in [\theta_i^{\min}, \theta_i^{\max}], i = 1, \dots, n\}. \quad (2.29)$$

Nesse caso, o objetivo é a obtenção de uma senóide $y(t, \theta(\mathbf{t}))$, pertencente ao conjunto M , que esteja mais “próxima” do sinal de entrada, $u(t)$. Como em sistemas de potência a componente fundamental do sinal elétrico possui tipicamente mais energia que as suas componentes harmônicas, a determinação da senóide $y(t, \theta(\mathbf{t}))$ mais próxima do sinal de entrada $u(t)$ será consequentemente a componente fundamental do mesmo.

Assim, a senóide $y(t, \theta(\mathbf{t}))$ deve estar associada a um vetor “ θ ” ótimo que minimize a função erro entre o sinal de entrada e a senóide $y(t, \theta(\mathbf{t}))$, ou seja:

$$\theta_{\text{ótimo}} = \arg \min_{\theta \in \Theta} e[y(t, \theta(t)), u(t)]. \quad (2.30)$$

Obviamente, o vetor $\theta_{\text{ótimo}}$ deve pertencer ao conjunto Θ definido na Equação 2.29, ou seja, $\theta_{\text{ótimo}} \in \Theta$.

A função erro entre a senóide $y(t, \boldsymbol{\theta}(t))$ procurada e o sinal de entrada $u(t)$ é dada por:

$$d(t, \boldsymbol{\theta}(t)) = [u(t) - y(t, \boldsymbol{\theta}(t))] = e(t). \quad (2.31)$$

Defini-se uma função *custo*, associada à função erro, dada por:

$$J(t, \boldsymbol{\theta}(t)) = d^2(t, \boldsymbol{\theta}(t)) \quad (2.32)$$

O procedimento seguinte é a estimação do vetor $\theta_{\text{ótimo}}$ que minimize a distância entre o sinal de entrada e a senóide procurada, $y(t, \theta(t))$, ou seja, minimize a função custo.

Utilizando-se o método *gradiente-descendente* como procedimento para minimização da função custo [20], têm-se:

$$\frac{d\boldsymbol{\theta}(t)}{dt} = -\boldsymbol{\mu} \cdot \frac{\partial [J(t, \boldsymbol{\theta}(t))]}{\partial \boldsymbol{\theta}(t)}, \quad (2.33)$$

onde:

$$\boldsymbol{\mu} = \begin{bmatrix} \mu_1 & & & \\ & \mu_2 & & \\ & & \ddots & \\ & & & \mu_n \end{bmatrix} \quad (2.34)$$

é a matriz que determina a velocidade de convergência do algoritmo, bem como sua estabilidade, chamada de *matriz de regulação*. O sinal negativo é usado na Equação 2.33 em consequência de o processo gradiente-descendente buscar a *minimização* da função custo.

Discretizando-se a Equação 2.33, através do método *Euler em atraso*, tem-se:

$$\boldsymbol{\theta}[k] = \boldsymbol{\theta}[k-1] - \mu T_s \frac{\partial J(t, \boldsymbol{\theta}[k-1])}{\partial \boldsymbol{\theta}[k-1]}, \quad (2.35)$$

onde T_s é a taxa de amostragem utilizada no processo de discretização. Desse modo, após efetuar-se a escolha dos parâmetros da componente fundamental a serem determinados, utiliza-se a Equação 2.35 para sua estimação.

Considerar, agora, a determinação dos parâmetros da componente fundamental do sinal $u(t)$ dado pela Equação 2.25. Tem-se que a componente fundamental, $u_1(t)$, pode ser representada através da seguinte equação:

$$u_1(t) = A \cdot \text{sen}(\psi(t)) = K_s \cdot \text{sen}(\phi(t)) + K_c \cdot \text{cos}(\phi(t)), \quad (2.36)$$

onde $A = \sqrt{K_c^2 + K_s^2}$ e $\psi(t) = \phi(t) + \tan^{-1} \frac{K_c}{K_s}$.

Nesse caso, os parâmetros a serem determinados são as componentes em fase e em quadratura do sinal, dados por K_s e K_c , bem como a fase $\phi(t)$. A fase $\phi(t)$ pode ser representada por $\phi(t) = \int_0^t (\omega_0 + \Delta\omega(\zeta)) d\zeta$, onde ω_0 é uma frequência previamente definida, próxima à frequência do sinal de entrada $u_1(t)$ e $\Delta\omega$ representa o desvio de frequência do sinal em torno da frequência central ω_0 . Desse modo, deseja-se estimar uma senóide $y(t, \theta(t))$ pertencente ao conjunto M descrito em (2.27), que esteja em função do tempo e em função do vetor de parâmetros $\theta(t)$ definido por:

$$\theta(t) = [K_s(t) \ K_c(t) \ \Delta\omega(t)] \quad (2.37)$$

O vetor de parâmetros $\theta(t)$ a ser encontrado deve minimizar a função custo definida por:

$$J(t, \theta(t)) = e^2(t, \theta(t)) = (u(t) - y(t))^2. \quad (2.38)$$

A função erro é expressa por:

$$e(t) = u(t) - [K_s \cdot \text{sen}(\phi(t)) + K_c \cdot \text{cos}(\phi(t))]. \quad (2.39)$$

O próximo passo é a estimação do vetor de parâmetros ótimo $\theta_{\text{ótimo}}$ que minimiza a função custo definida pela Equação 2.38. A minimização da função custo é efetuada através da utilização do método gradiente-descendente, ou seja:

$$\frac{d\boldsymbol{\theta}(t)}{dt} = -\boldsymbol{\mu} \frac{\partial[J(t, \boldsymbol{\theta}(t))]}{\partial \boldsymbol{\theta}(t)} = -\boldsymbol{\mu} \frac{\partial[e^2(t, \boldsymbol{\theta}(t))]}{\partial \boldsymbol{\theta}(t)}. \quad (2.41)$$

A equação anterior pode ser simplificada através da utilização da seguinte relação:

$$\frac{df^2(t)}{dt} = 2 \cdot f(t) \cdot \frac{df(t)}{dt}. \quad (2.42)$$

Assim:

$$\frac{d\boldsymbol{\theta}(t)}{dt} = -2 \cdot e(t, \boldsymbol{\theta}(t)) \cdot \boldsymbol{\mu} \frac{\partial[e(t, \boldsymbol{\theta}(t))]}{\partial \boldsymbol{\theta}(t)}. \quad (2.43)$$

Substituindo-se a Equação 2.37 e 2.39 na Equação 2.43, têm-se:

$$\begin{bmatrix} \dot{K}_s(t) \\ \dot{K}_c(t) \\ \dot{\Delta \boldsymbol{\omega}}(t) \end{bmatrix} = 2e(t) \begin{bmatrix} -\boldsymbol{\mu}_s & 0 & 0 \\ 0 & -\boldsymbol{\mu}_c & 0 \\ 0 & 0 & -\boldsymbol{\mu}_f \end{bmatrix} \cdot \begin{bmatrix} -\text{sen}(\phi(t)) \\ -\text{cos}(\phi(t)) \\ t(-K_s \text{cos}(\phi(t)) + K_c \text{sen}(\phi(t))) \end{bmatrix} \quad (2.44)$$

Nas equações anteriores, o ponto sobre as variáveis representa as derivadas das funções em relação ao tempo. Notar que a Equação 2.44 é variante no tempo; deste modo, admitindo-se que o sinal de entrada $u(t)$ seja aproximadamente periódico, pode-se esperar que o conjunto de soluções das equações diferenciais também seja periódico. Desse modo, duas soluções são possíveis [11]:

- Considerar a variável t como sendo igual à mod $(2\pi/w_0)$;
- Igualar a variável t a um valor contido no intervalo pertencente ao intervalo $(0, 2\pi)$.

A segunda opção é escolhida. Desse modo, a variável t é “absorvida” pela constante $\boldsymbol{\mu}_f$ e o conjunto de equações do processo de estimação da senóide $y(t)$ será dado por:

$$\begin{aligned}
\dot{K}_s(t) &= 2 \cdot \mu_s \cdot e(t) \cdot \text{sen}(\phi(t)) \\
\dot{K}_c(t) &= 2 \cdot \mu_c \cdot e(t) \cdot \cos(\phi(t)) \\
\dot{\Delta \omega}(t) &= 2 \cdot \mu_f \cdot e(t) [K_s \cdot \cos(\phi(t)) - K_c \cdot \text{sen}(\phi(t))]. \\
\phi(t) &= \omega_0 + \Delta \omega(t) \\
y(t) &= K_s \cdot \text{sen}(\phi(t)) + K_c \cdot \cos(\phi(t)) \\
e(t) &= u(t) - y(t)
\end{aligned} \tag{2.45}$$

Discretizando-se o conjunto de equações representado em (2.45), através do método *Euler em atraso*, têm-se:

$$\begin{aligned}
K_s[n+1] &= K_s[n] + 2 \cdot T_s \cdot \mu_s \cdot e[n] \cdot \text{sen}(\phi[n]) \\
K_c[n+1] &= K_c[n] + 2 \cdot T_s \cdot \mu_c \cdot e[n] \cdot \cos(\phi[n]) \\
\Delta \omega[n+1] &= \Delta \omega[n] + 2 \cdot T_s \cdot \mu_f \cdot e[n] [K_s \cdot \cos(\phi[n]) - K_c \cdot \text{sen}(\phi[n])] \\
\phi[n+1] &= \phi[n] + T_s \cdot (\omega_0 + \Delta \omega[n]) \\
y[n] &= K_s[n] \cdot \text{sen}(\phi[n]) + K_c[n] \cdot \cos(\phi[n]) \\
e[n] &= u[n] - y[n],
\end{aligned} \tag{2.46}$$

onde T_s é a taxa de amostragem utilizada pelo processo. O conjunto de equações descrito em (2.46) dá origem à malha PLL representada pela Figura 2.12:

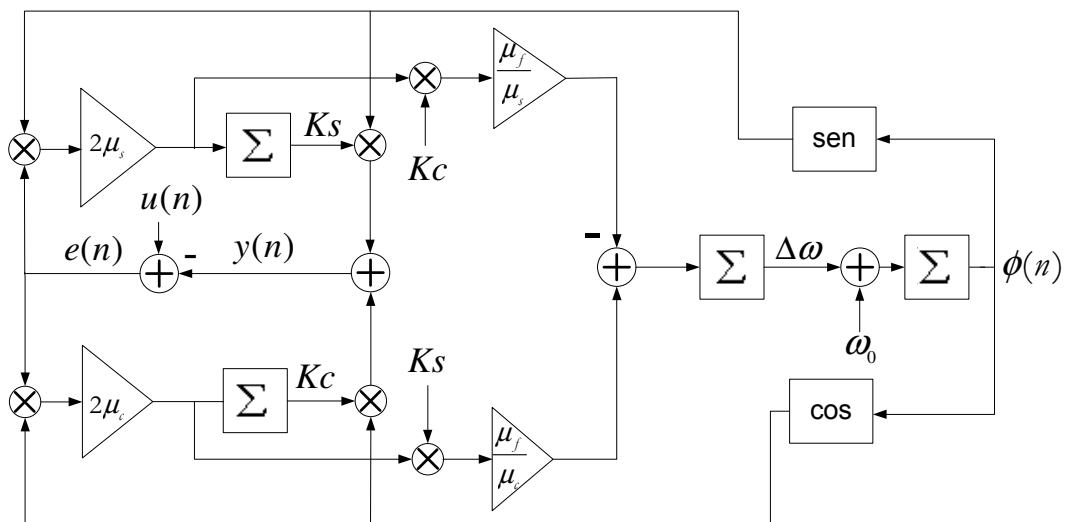


Figura 2.12 - Malha QPLL

A entrada da malha é dada por $u(n)$, o sinal de erro está representado por $e(n)$ e a senóide sintetizada pela malha é dada por $y(n)$. Com o propósito de simplificação da

estrutura, pode-se fazer com que $2\mu_s = 2\mu_c = k_p$, $\mu_s = \mu_c$ e $\frac{\mu_f}{\mu_c} = \frac{\mu_f}{\mu_c} = k_i$. Assim, a estrutura descrita na Figura 2.12 resultará no modelo simplificado representado pela Figura 2.13. As estruturas correspondentes ao PD (detector de fase), LF (filtro passa-baixas) e VCO (oscilador controlado por voltagem) estão indicadas na mesma figura.

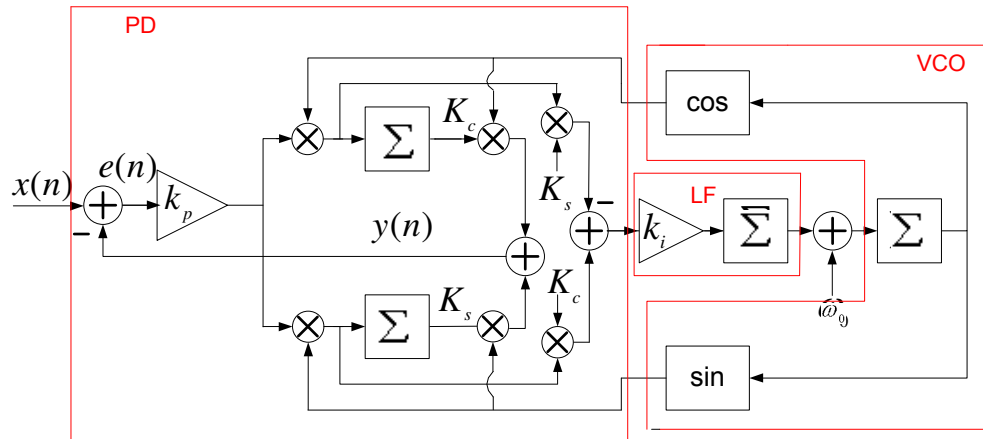


Figura 2.23 - Malha QPLL simplificada

A malha de PLL descrita na Figura 2.13 permite estimar diretamente as componentes em quadratura e em fase da componente fundamental de entrada, bem como sua frequência. O argumento e a amplitude podem ser estimados indiretamente através das equações:

$$A = \sqrt{K_c^2 + K_s^2}$$

$$\psi(t) = \phi(t) + \tan^{-1} \frac{K_c}{K_s} \cdot \quad (2.48)$$

2.6 Simulações Computacionais

No intuito de analisar o comportamento dinâmico da malha QPLL proposta, as equações descritas em (2.46) foram implementadas no software MATLAB® e diversas simulações computacionais foram realizadas. Em todas as simulações, foi utilizada uma taxa de amostragem da malha igual a 7680 Hz. A Figura 2.14 descreve o sinal $y(n)$ sintetizado pela malha QPLL quando excitada por um sinal de amplitude unitária e

freqüência igual a 60 Hz. Na mesma figura são mostrados o sinal de entrada e o erro de freqüência detectado pela malha QPLL. A freqüência central da malha QPLL, ω_0 , é considerada, inicialmente, como 59 Hz.

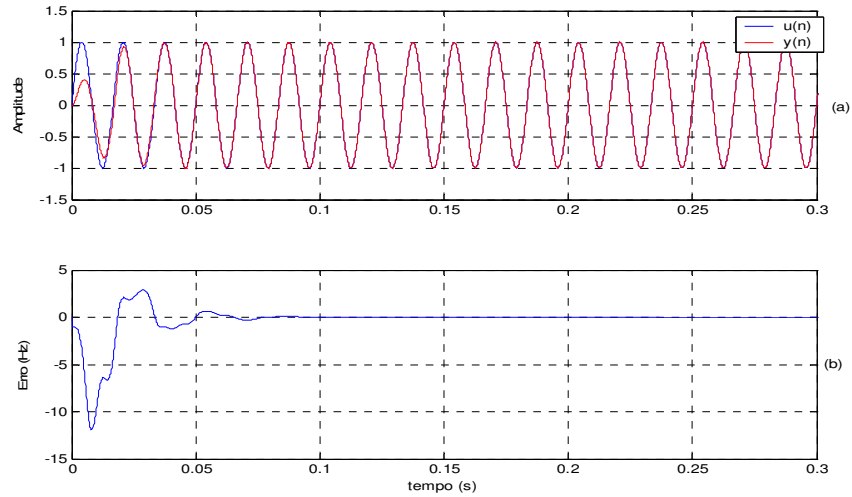


Figura 2.14 – Desempenho inicial (a) Sinais de entrada e saída da malha. (b) Erro de freqüência

De acordo com os resultados, a malha converge em um tempo igual a 2 ciclos da componente fundamental de entrada e o erro de freqüência, $\Delta\omega$, se estabiliza com um erro menor que 1% antes de 0,10 segundos. A Figura 2.15 mostra os parâmetros K_c e K_s estimados pela malha proposta, bem como a amplitude da senóide de entrada.

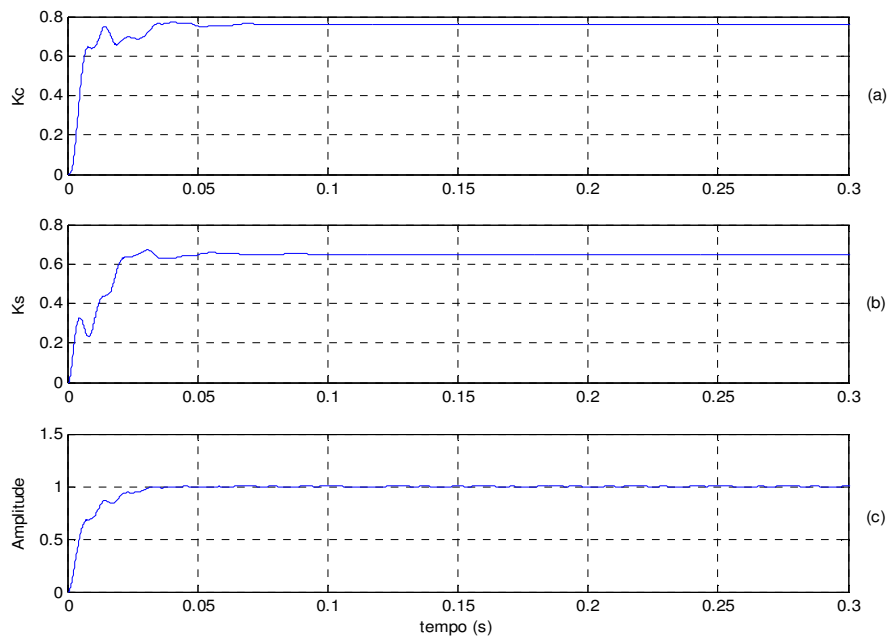


Figura 2.15 - Parâmetros adquiridos pela malha de QPLL. (a) K_c (b) K_s (c) Amplitude

Para analisar-se a eficácia da malha proposta com relação à rejeição de distúrbios, o sinal de entrada foi poluído com um ruído gaussiano com uma variância igual a $\sigma^2 = (0,1)^2$. Em seguida, os valores de k_p e k_i foram inicializados com $k_p = k_i = 150$. A senóide sintetizada pela malha, bem como o sinal de entrada, e a imunidade a ruídos são mostrados nas Figuras 2.16 e 2.17.

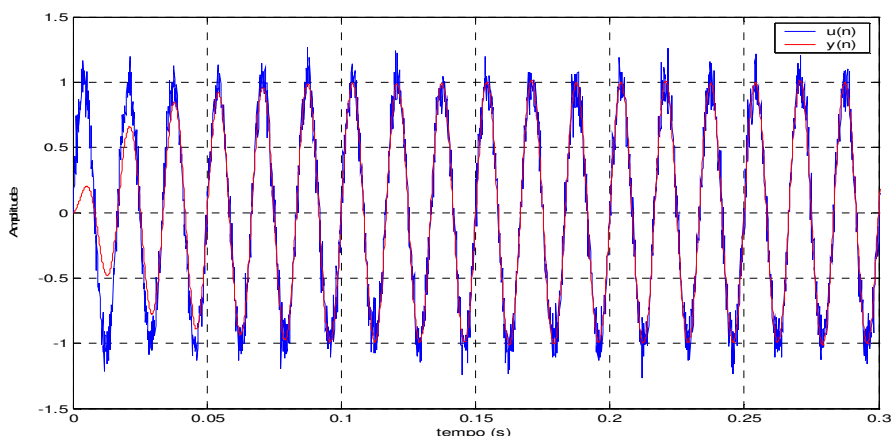


Figura 2.16 - Sinal de entrada $u(n)$ e sinal de saída $y(n)$

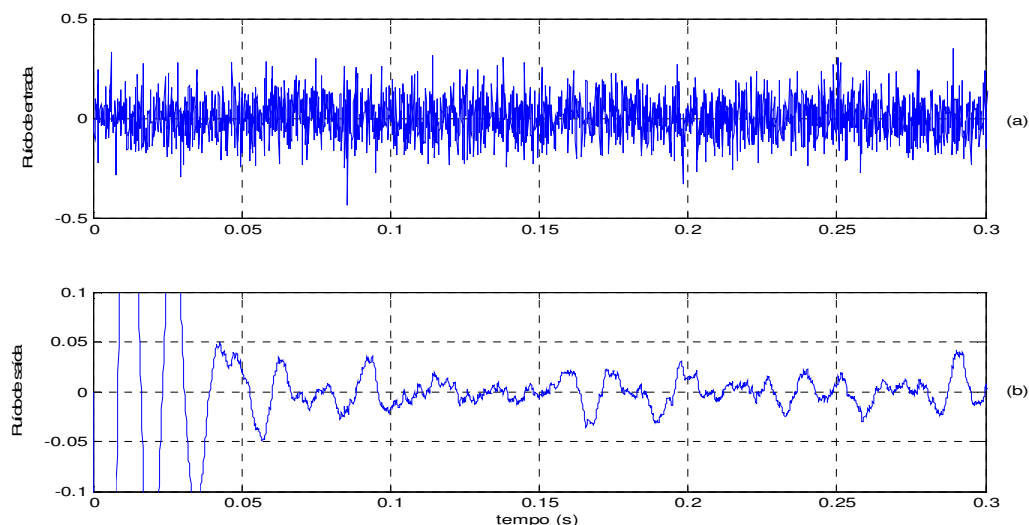


Figura 2.17 - Imunidade a ruídos. (a) Ruído de entrada. (b) Ruído de saída.

No intuito de promover uma resposta mais rápida da malha QPLL, os valores k_p e k_i foram inicializados com $k_p = k_i = 350$. A utilização de valores mais elevados para essas constantes, ou seja, a utilização de valores mais elevados para a matriz de regulação definida em (2.34), tem como objetivo fazer com que a solução das equações descritas

em (2.46), através do método gradiente-descendente, tenha uma convergência mais rápida. Entretanto, de acordo com as simulações apresentadas, comprovou-se que o uso de valores elevados para os coeficientes da matriz de regulação acaba por acarretar uma menor rejeição a distúrbios da malha de QPLL proposta. As Figuras 2.18 e 2.19 mostram os resultados das simulações realizadas, em que o sinal de entrada foi novamente poluído com um ruído gaussiano com uma variância $\sigma^2 = (0,1)^2$.

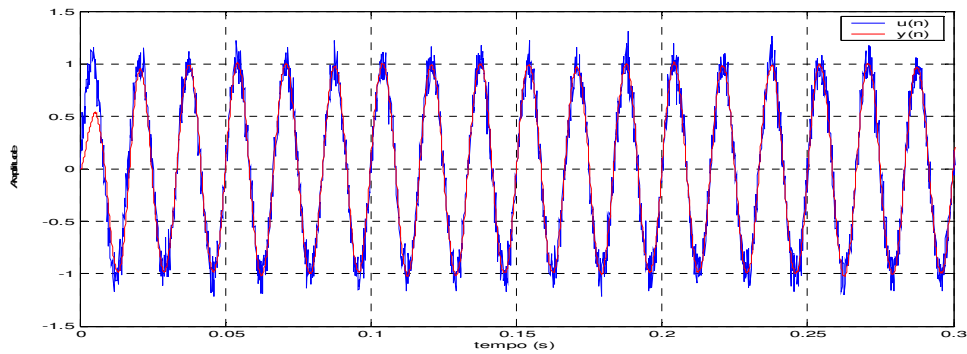


Figura 2.18 - Sinal de entrada $u(n)$ e sinal de saída $y(n)$

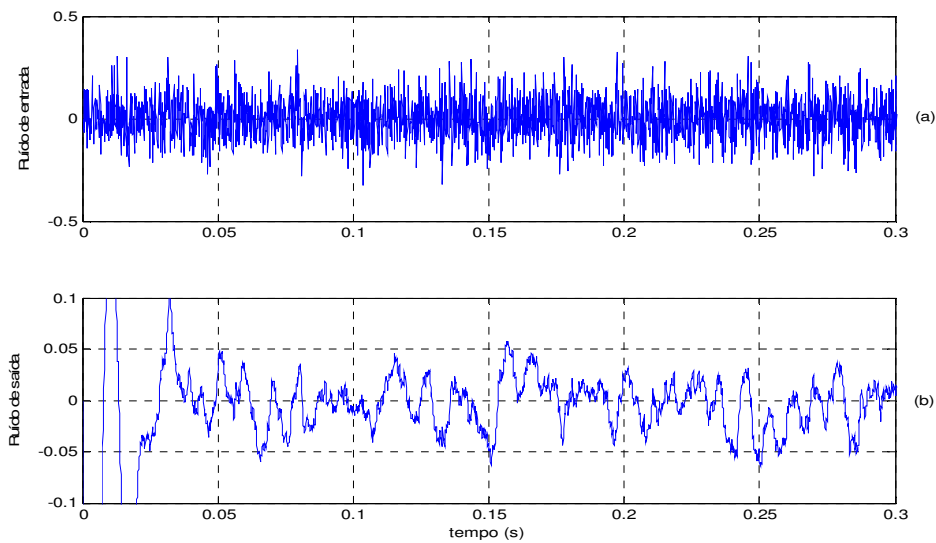


Figura 2.19 - Imunidade a ruídos. (a) Ruído de entrada. (b) Ruído de saída.

Nesse caso, nota-se que a rejeição a distúrbios da malha de QPLL mostrou-se mais elevada quando os coeficientes da matriz de regulação foram inicializados com valores menores. Entretanto, nesse caso, a sua resposta dinâmica se degradou, fazendo com que a malha se tornasse mais “lenta”. Assim, nota-se que os coeficientes da malha devem ser ajustados adequadamente de acordo com a aplicação de interesse.

A Figura 2.20a mostra o tempo de convergência da malha QPLL, quando os parâmetros k_p e k_i variam desde $k_p = k_i = 150$ até $k_p = k_i = 250$. Nesse caso, foi introduzido na malha um sinal de amplitude unitária e frequência igual a 60 Hz poluído com um ruído com uma variância $\sigma^2 = (0,1)^2$. A malha de QPLL foi inicializada com $\omega_0 = 59H_{\mathcal{Z}}$. O tempo de convergência foi calculado como sendo o tempo em que a malha proposta leva para atingir 99% do valor da amplitude do sinal de entrada. A Figura 2.20b mostra a relação sinal/ruído (SNR) de saída da malha QPLL, em função do ajuste dos parâmetros k_p e k_i .

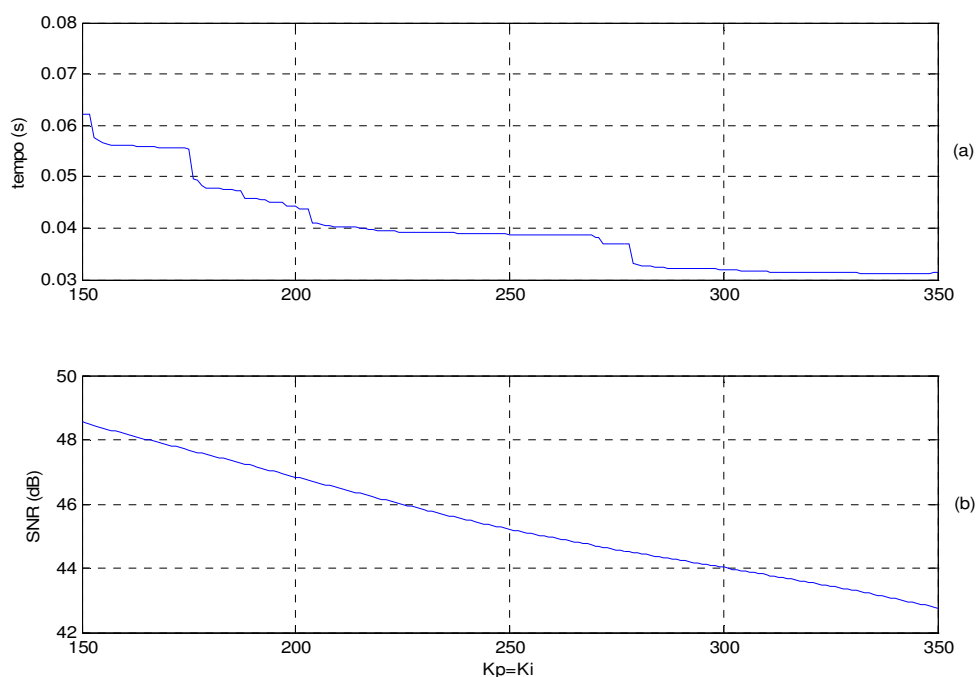


Figura 2.20 – Variação da resposta dinâmica e rejeição à distúrbios da malha QPLL em função do ajuste dos parâmetros k_p e k_i . (a) Tempo para a amplitude da malha QPLL atingir 99% do seu valor final. (b) Relação Sinal/ruído de saída da malha QPLL

Nota-se, claramente, que ao se ajustar os valores k_p e k_i em valores mais elevados, resultam um tempo mais rápido de convergência da malha. Isto pode ser visto na Figura 2.20a. Entretanto, ao se aumentar esses parâmetros, a malha apresenta uma menor imunidade a ruídos, conforme mostra a Figura 2.20b.

A Figura 2.21 mostra o sinal sintetizado pela malha QPLL, $y(t)$, quando um sinal de entrada $u(t)$, poluído por um distúrbio transitório do tipo *SAG* (afundamento de tensão) seguido por um do tipo *SWELL* (elevação de tensão) é introduzido na malha. De

acordo com a simulação, nota-se claramente que a malha de QPLL proposta responde rapidamente aos distúrbios de entrada.

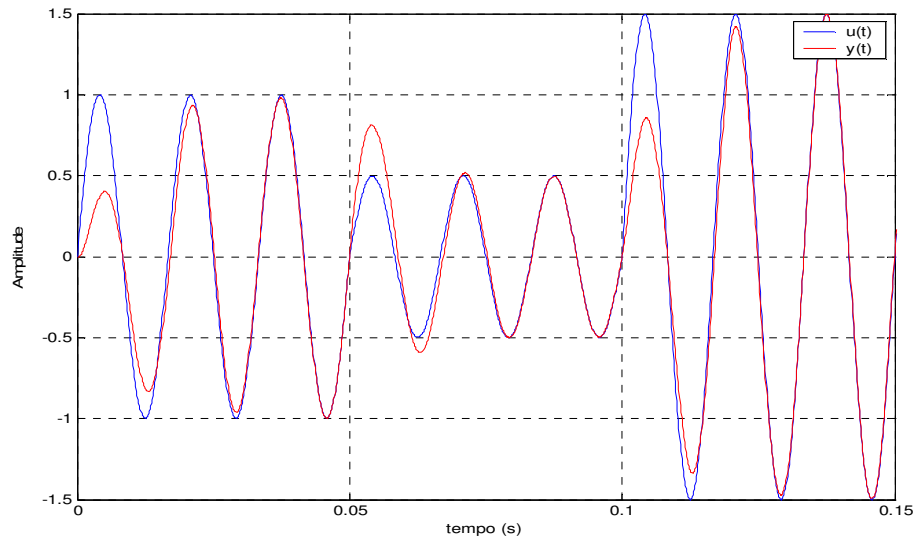


Figura 2.21 - Resposta da malha a um distúrbio do tipo SAG seguido por um SWELL

2.7 Conclusão

Esse capítulo apresentou uma revisão bibliográfica de malhas PLL. Baseado em modelos lineares foi apresentada a função de transferência que descreve as malhas de PLL de um modo geral. Através do estudo da função de transferência da malha, é possível realizar os ajustes dos coeficientes que controlam o comportamento dinâmico do PLL, de modo a adequá-lo de uma maneira ótima para cada aplicação de interesse. Em seguida, foi mostrado o mapeamento da função de transferência de um PLL representado no domínio contínuo para o domínio discreto, onde se viu as condições que o PLL digital deve possuir para se localizar na região de estabilidade do plano z . Finalmente, foi mostrado o modelo de um PLL chamado QPLL, que realiza a aquisição de todos os parâmetros da componente fundamental de um sinal de entrada qualquer. De acordo com as análises efetuadas no QPLL, concluiu-se que o seu comportamento dinâmico segue o mesmo padrão descrito para o modelo de PLL linear apresentado no início do capítulo.

Capítulo 3

Enhanced Quadrature Phase-Locked-Loop

3.1 Introdução

Foi apresentada, no capítulo anterior, a descrição de uma malha PLL do tipo quadratura (QPLL), com capacidade para estimar os parâmetros da componente fundamental de um sinal de entrada qualquer (amplitude, fase e frequência), através de um método de minimização de erro chamado *gradiente-descendente*. A velocidade de convergência e o erro em regime permanente da malha podem ser ajustados através de parâmetros presentes na sua *matriz de regulação*. De acordo com as simulações apresentadas no capítulo anterior, comprovou-se a alta imunidade a ruídos da malha, bem como sua rápida resposta transitória para diversos valores dos parâmetros da *matriz de regulação*. Entretanto, para diversas situações comuns em sistemas de potência, como a presença de componentes harmônicas e inter-harmônicas de baixa frequência no sinal de entrada, a resposta em regime permanente da malha não se mostra satisfatória. A solução mais simples para esse caso é o ajuste dos parâmetros da *matriz de regulação* da malha QPLL, de modo que se obtenha uma maior rejeição a ruídos em regime permanente. Entretanto, o ajuste desses parâmetros acaba prejudicando a resposta dinâmica da malha, que acaba se tornando lenta. Baseado no exposto, a proposta deste capítulo é a apresentação de estratégias de *filtragem* do sinal de entrada da malha, que ocasionem a

redução do erro em regime permanente, sem sacrificar demasiadamente a resposta dinâmica da estrutura.

3.2 O Filtro *Notch*

3.2.1. Introdução

Será analisada, nesta secção, uma estratégia de filtragem do sinal de entrada da malha QPLL através da utilização de um filtro passa banda, centrado na frequência da componente fundamental do sinal de entrada (tipicamente 60 Hz). A filtragem tem como objetivo eliminar todas as componentes harmônicas e reduzir os ruídos aditivos do sinal de entrada, permitindo somente a passagem de sua componente fundamental. O filtro passa-banda proposto neste trabalho é obtido a partir da estrutura de um filtro *notch*, ou rejeita-faixa, sintonizado na frequência da componente fundamental do sinal de entrada, com a finalidade de eliminá-la e, posteriormente, identificá-la através de uma equação algébrica.

Considere o sinal de entrada da malha de QPLL dado por:

$$v = v_1 + v_{res} , \quad (3.1)$$

onde v_1 representa sua componente fundamental e v_{res} as componentes harmônicas, ruídos aditivos e, também, componentes de frequência inferiores a 60 Hz (“*flickers*”). Deseja-se, neste caso, obter-se a componente fundamental, v_1 . Como a saída do filtro rejeita faixa, que está centrado na frequência da componente fundamental do sinal de entrada, será expressa aproximadamente por:

$$v_{filtro} \approx v_{res} , \quad (3.2)$$

têm-se que a obtenção da componente fundamental do sinal de entrada pode ser obtida através da equação algébrica simples:

$$v_1 = v_{filtro} - v . \quad (3.3)$$

A Figura 3.1 mostra como o filtro passa-banda pode ser obtido a partir de um filtro *notch* IIR. Conforme será discutido mais adiante, o sinal de saída conterá a componente fundamental do sinal, com o mínimo de distorção. Quanto mais estreita for a resposta em magnitude do filtro notch, menor será a distorção do sinal de saída. Além de eliminar as componentes harmônicas, ou inter-harmônicas, contidas no sinal de entrada, a estrutura também é capaz de melhorar a relação sinal ruído do sinal de saída, ou seja, a *SNR* do sinal de saída do filtro é aumentada.

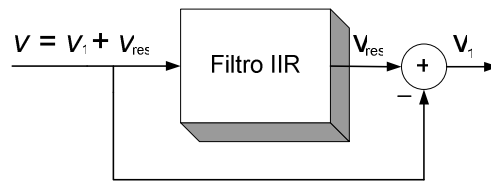


Figura 3.1 – Filtro passa-banda obtido a partir do filtro notch

A função de transferência de um filtro rejeita-faixa de segunda ordem, no domínio do tempo discreto, pode ser dada pela seguinte equação [26,32]:

$$H_0(z) = \frac{Y(z)}{X(z)} = \frac{1 + a_0 z^{-1} + z^{-2}}{1 + \rho_0 a_0 z^{-1} + \rho_0^2 z^{-2}}, \quad (3.4)$$

onde $0 << \rho_0 < 1$ é o fator *notch* e $a_0 = -2 \cos(\omega_0)$. A banda de passagem do filtro rejeita-faixa é definida através do parâmetro ρ_0 (fator do *notch*) e, quanto mais próximo da unidade, mais seletivo ou sintonizado será o filtro. Por outro lado, a resposta transitória do filtro é dada pela fórmula geral:

$$y_t(n) = k_1 \lambda^n + k_2 \lambda^{*n}, \quad (3.5)$$

onde k_1 e k_2 são constantes determinadas a partir das condições iniciais especificadas e λ é uma das raízes do polinômio característico dado por $p(n) = \lambda^2 + \rho_0 a_0 \lambda + 1$. É fácil verificar-se que $|\lambda| = \rho_0$. Portanto, pode-se concluir que, quanto mais próximo da unidade for o fator *notch*, mais lenta será a resposta dinâmica do filtro.

A Figura 3.2 mostra a resposta ao degrau unitário de um filtro notch com $\rho_0 = 0,99$. Na mesma figura, são marcadas as linhas de $\pm 1\%$ do valor final. Se for

considerado, para efeitos práticos, a duração do transitório como o número de ciclos de 60 Hz em que o sinal permanece fora da faixa de 1%, pode-se afirmar que o transitório do filtro em análise dura cerca de 4,4 ciclos da componente fundamental. O resultado apresentado na Figura 3.2 foi obtido utilizando-se uma frequência de amostragem $f_s = 7680$ Hz.

A Tabela 3.1 mostra os valores transitórios para diferentes fatores *notch*. A segunda coluna mostra o tempo de convergência em número de ciclos, ao passo que a terceira coluna apresenta a convergência em relação ao número de amostras.

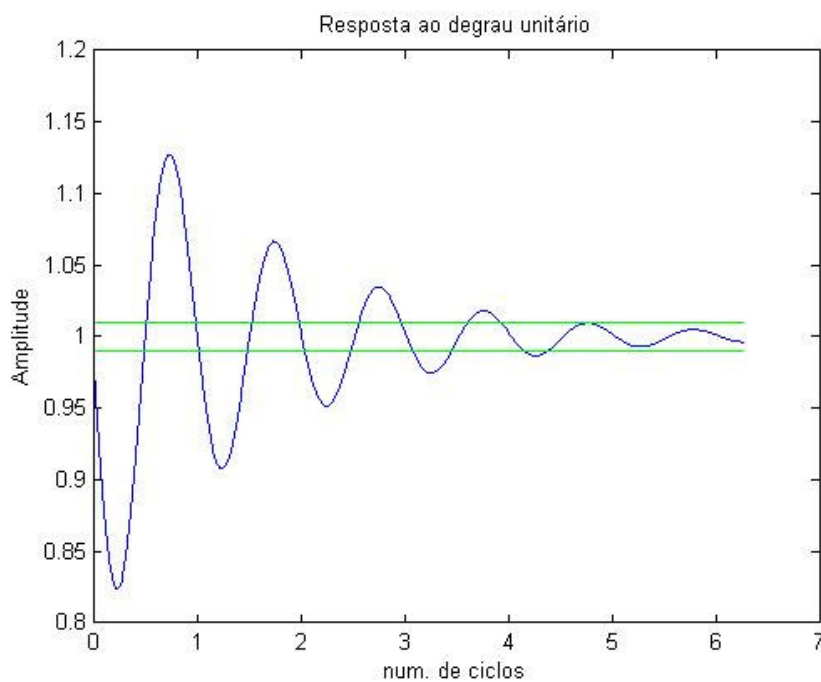


Figura 3.2- Resposta ao degrau unitário ($\rho_0 = 0.99$)

Tabela 3.1- Transitórios do filtro para diversos fatores notch

Fator notch	Duração do transitório	
	# ciclos	# amostras
0,99	4,4	563
0,98	2,5	320
0,96	1,5	192
0,90	1,3	166

As Figuras 3.3a e 3.3b mostram a resposta, em frequência e fase, de um filtro rejeita-faixa sintonizado na frequência de 60 Hz, ajustado com um fator notch $\rho_0 = 0,99$. A taxa de amostragem utilizada em sua implementação foi igual a $f_s = 7680$ Hz. Do mesmo modo, as Figuras 3.3c e 3.3d mostram a resposta, em frequência e fase, do filtro projetado com um fator notch $\rho_0 = 0,93$. Notar que o gráfico da resposta em fase do filtro *notch* apresenta uma descontinuidade na frequência central de corte (60 Hz). Entretanto, de acordo com (3.1), o atraso nessa frequência será nulo, o que torna a utilização do filtro *notch* apropriada para aplicações em que a fase da componente fundamental do sinal de entrada é uma variável a ser adquirida.

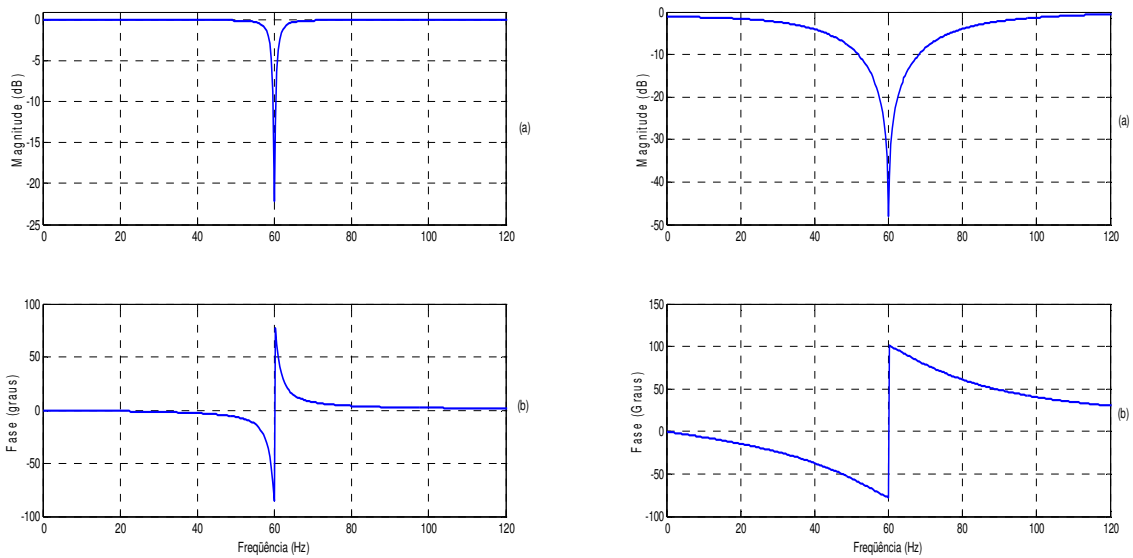


Figura 3.3- Resposta em frequência do filtro notch: (a) Magnitude e (b) Fase para ($\rho_0 = 0.99$); (c) Magnitude e (d) Fase para $\rho_0 = 0.93$

3.2.2. Melhoria da Relação Sinal-Ruído (*SNR*)

A filtragem do sinal de entrada pelo filtro passa-banda, construído a partir do filtro *notch*, produz um sinal de saída mais adequado às aplicações de estimação de parâmetros. Além de atenuar as componentes harmônicas ou inter-harmônicas presentes no sinal de entrada, ele também melhora a relação *SNR*. O aumento da *SNR* está diretamente ligado ao fator *notch* do filtro, ρ . A Equação 3.6 mostra a relação entre a variância do sinal de saída e a do sinal de entrada, se um ruído branco é apresentado na entrada do filtro [27]:

$$\sigma_s^2 = \frac{\sigma_e^2}{2\pi} \int_{-\pi}^{+\pi} |H_{PB}(e^{jw})|^2 dw, \quad (3.6)$$

onde $H_{PB}(e^{jw})$ é a resposta em frequência do filtro passa-banda obtido a partir do filtro *notch* e σ_e^2 e σ_s^2 representam as variâncias do sinal de entrada e de saída, respectivamente. A relação de Parseval, expressa pela Equação 3.6, pode ser escrita no domínio do tempo, como:

$$\sigma_s^2 = \sigma_e^2 \sum_{n=0}^{\infty} |h_{PB}(n)|^2, \quad (3.7)$$

onde $h_{PB}(n)$ é a resposta ao impulso do filtro passa-banda. A Equação 3.7 fornece uma maneira prática de se obter uma aproximação razoável da variância do sinal de saída. Como a resposta ao impulso do filtro decai para zero assintoticamente, o somatório efetuado utilizando um número finito de termos fornecerá aproximação razoável de (3.7). A Tabela 3.2 mostra a melhoria na *SNR*; observar que a *SNR* do sinal após a filtragem é aumentada de 20 dB para um fator notch de 0,99 e 10 dB para um fator de 0,99.

Tabela 3.2- Melhoria na *SNR* do sinal após filtragem

Fator notch	Aumento da <i>SNR</i> (dB)
0,99	20
0,98	17
0,96	14
0,90	10

Uma validação da implementação do filtro notch no domínio discreto foi então efetuada, utilizando-se, para isto, o software MATLAB®. A Figura 3.4 mostra a resposta do filtro *notch* excitado com um sinal expresso por:

$$y(n) = A \cos(2\pi n f_0 T_s) + h(n), \quad (3.8)$$

onde $A=1$, $f_0=60$ Hz, $T_s = 1/7680$ Hz e $h(n)$ representa um ruído aditivo gaussiano com média zero e variância $\sigma^2 = (0,1)^2$. Neste primeiro caso, foi utilizado um fator *notch* $\rho = 0,99$ para a síntese do filtro. A Figura 3.4a mostra o sinal de entrada e a Figura 3.4b

o sinal de saída do filtro passa banda, onde se pode observar a significativa melhoria na relação sinal ruído e um transitório de aproximadamente 4,4 ciclos.

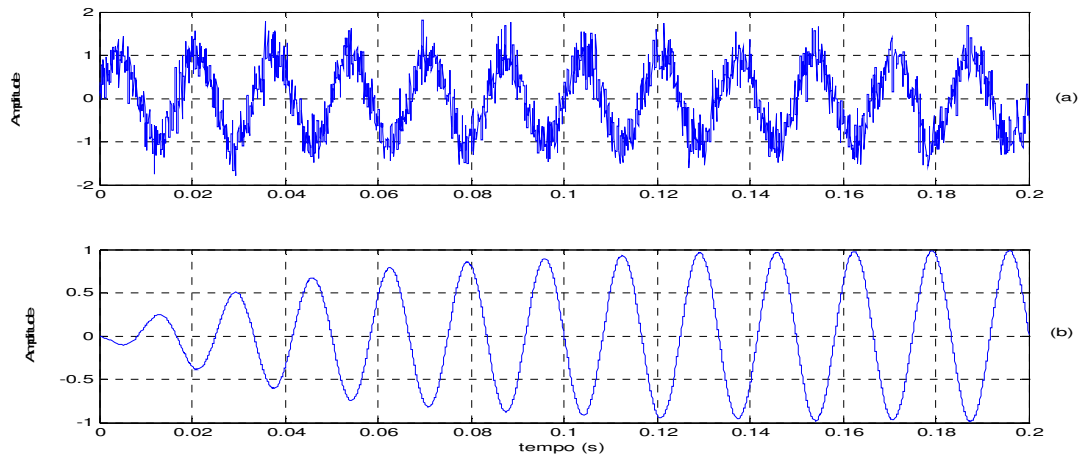


Figura 3.4 - Sinais de entrada (a) e saída (b) do filtro passa-banda

A Figura 3.5 mostra a resposta do filtro *notch* excitado pelo mesmo sinal anterior, porém adotando-se um fator *notch* igual a 0,93. Nota-se, de acordo com os resultados obtidos, que o tempo de transitório do filtro *notch* foi inferior ao primeiro caso; entretanto, no segundo caso, a melhoria da *SNR* foi menor, conforme já podia ser esperado.

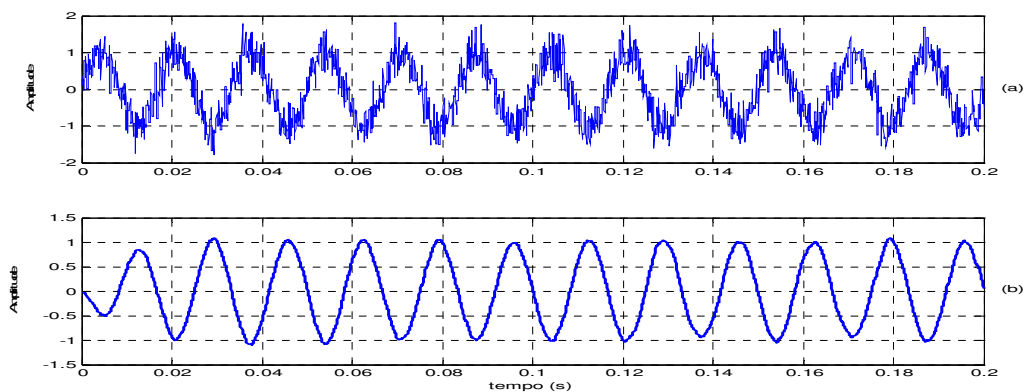


Figura 3.5 - Sinais de entrada (a) e saída (b) do filtro passa-banda

3.3 - O QPLL modificado

A Figura 3.6 mostra o diagrama de blocos da estrutura QPLL modificada, o E-QPLL, derivado do inglês “*Enhanced Quadrature PLL*”. O E-QPLL é formado por três blocos básicos: (a) o filtro passa-faixa, (b) o estimador QPLL e (c) o algoritmo de atualização da frequência central do filtro *notch*.

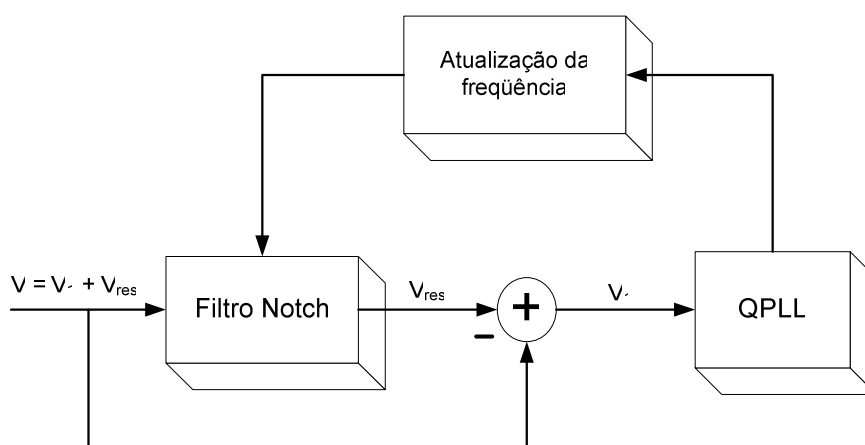


Figura 3.6- Diagrama de blocos do QPLL modificado

Os blocos do filtro passa-faixa e do estimador QPLL já foram previamente discutidos, razão pela qual será abordado, nesta seção, mais especificamente o bloco de atualização da frequência central do filtro *notch*. Em situação práticas, a frequência da componente fundamental pode variar, razão pela qual é necessário adotar-se uma estratégia de ajuste dos coeficientes do filtro *notch*.

Baseado nestas considerações, serão analisadas, a seguir, duas estratégias distintas para atualização da frequência central, bem como será efetuado um procedimento de análise de seu desempenho.

3.3.1 - Atualização da frequência pela média

Uma estratégia simples e eficiente de ajuste do filtro *notch* consiste em utilizar a frequência estimada do sinal de entrada pela malha de QPLL e, a partir de um valor médio integralizado em um intervalo considerado, realizar a atualização do filtro *notch*. Por este procedimento, a frequência detectada pela malha QPLL deve passar por um

filtro média móvel de ordem N , que estimará a média da frequência detectada pela malha QPLL dos últimos N pontos. Entretanto, com o objetivo de diminuir a complexidade computacional do algoritmo, o período de atualização do filtro notch será igual ao tamanho da janela do filtro média móvel. A Figura 3.7 mostra o diagrama do procedimento para atualização do filtro *notch*.

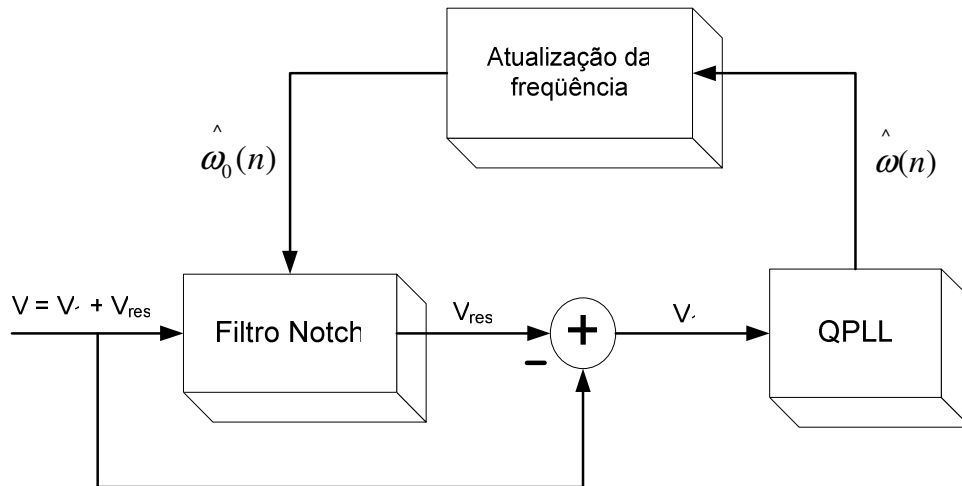


Figura 3.7- Atualização do notch pela média das estimações da frequência.

O algoritmo de atualização da frequência *notch*, pela média, será obtido pela expressão:

$$\hat{\omega}_0(n) = \begin{cases} \frac{1}{N} \sum_{i=0}^{N-1} \hat{\omega}(n-i), & \text{se } n = N, 2N, 3N \dots \\ \hat{\omega}_0(n-1) & , \text{ caso contrário} \end{cases} \quad (3.9)$$

onde $\hat{\omega}(n)$ é o valor estimado da frequência obtido a partir do QPLL e $\hat{\omega}_0(n)$ é a frequência central do filtro *notch*. Observar que a frequência é atualizada apenas para múltiplos inteiros de N .

3.3.2 - Atualização instantânea

No processo de atualização instantânea, o valor da frequência estimada é diretamente realimentado para o filtro *notch* para a atualização de sua frequência central. A Figura 3.8 mostra o desempenho dos dois processos de atualização para um sinal senoidal cuja frequência sofre uma variação em degrau de 60 Hz para 63 Hz. Os dois procedimentos descritos são mostrados na figura: a curva contínua representa o método

de atualização instantânea e a curva tracejada o método de atualização pela média. Pode-se observar, mediante uma inspeção meramente visual, que o desempenho dos dois métodos é muito parecido.

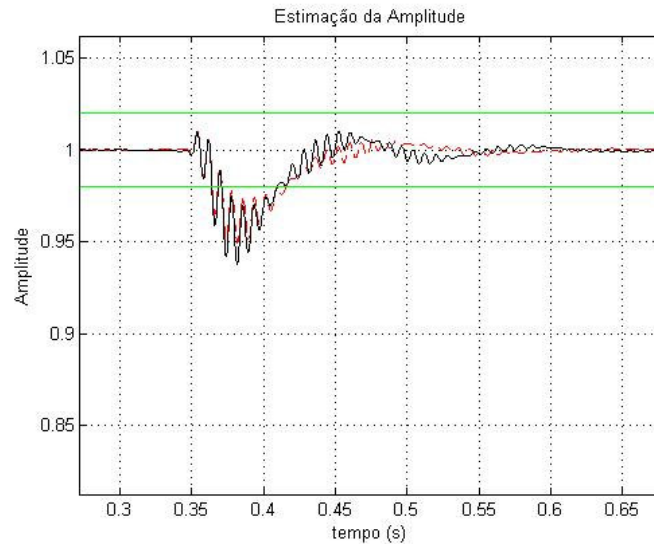


Figura 3.8- Algoritmo de atualização filtro notch: traço contínuo, atualização instantânea; traço tracejado, atualização pela média.

3.3.3 Comparação do desempenho dos métodos de ajuste dos coeficientes do filtro notch

No intuito de analisar o desempenho dos dois métodos propostos de atualização do filtro notch - instantâneo e pela média - é proposta, nesta seção, uma estratégia de análise de desempenho de cada método de atualização. Primeiramente, foi introduzido na malha E-QPLL um sinal dado pela equação:

$$u(n) = \cos\left(\frac{2\pi n f_0}{f_s}\right), \quad (3.10)$$

onde $f_0 = 60$ Hz e $f_s = 15360$ Hz.

Em seguida, foram adicionados diversos distúrbios no sinal considerado, de acordo com a Tabela 3.3 que se segue.

Tabela 3.3 – Distúrbios no sinal de entrada

Distúrbio	Magnitude	Instante inicial	Duração
Ruído gaussiano	Variância $\sigma^2 = (0,1)^2$	t = 0 segundos	1 segundo
SAG	20%	t = 0,5 segundos	0,05 segundos
SWELL	20%	t = 0,5 segundos	0,05 segundos
Harmônicas (3. ^a , 5. ^a , 9. ^a)	20%, 10% e 8%	t = 0 segundos	1 segundo
Degrau na frequência	+ 5%	t = 0,5 segundos	0,5 segundos
Degrau na fase	+5%	t = 0,5 segundos	0,5 segundos

Para cada distúrbio foi então calculado o erro quadrático médio entre a componente fundamental do sinal de entrada e a senóide sintetizada pela malha E-QPLL, de acordo com a Figura 3.9.

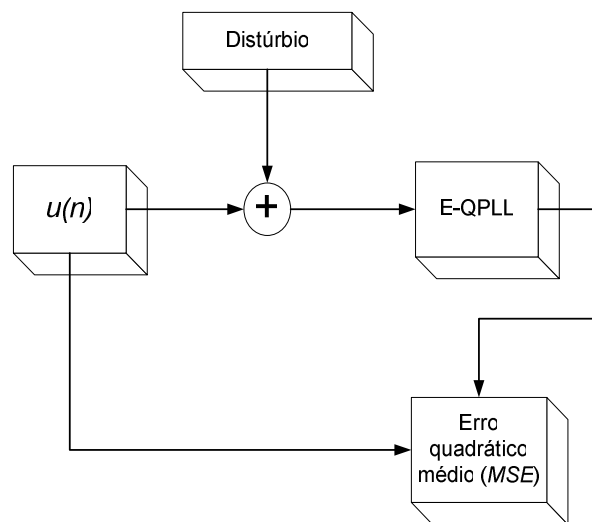


Figura 3.9 – Estratégia de cálculo do erro médio quadrático

O erro quadrático médio foi calculado considerando-se um intervalo de integração igual a 1 segundo. Os parâmetros da malha E-QPLL foram ajustados com os valores $k_p = k_i = 75$ e a frequência central de operação foi considerada inicialmente como 58 Hz. O filtro notch foi ajustado com $\rho = 0,996$ e a frequência de amostragem utilizada foi igual a 15.360 Hz.

Para cada situação, a atualização do filtro notch foi realizada através do método instantâneo ou através da média. Para o método de atualização pela média, diversos tamanhos de janelas foram utilizados. A Tabela 3.4 mostra o erro médio quadrático para

diversos tipos de sinais de entrada e métodos de atualização dos coeficientes do filtro notch.

Tabela 3.4 – Erro médio quadrático obtido para diversos distúrbios e métodos de atualização do filtro notch

Método Atualização Distúrbio	Instantâneo	Média – 1 Janela	Média – 2 Janelas	Média – 3 Janelas
Ruído gaussiano	0,131034	0,12745	0,124676	0,12223
SAG	0,131426	0,128304	0,125789	0,123831
SWELL	0,132011	0,128327	0,125789	0,123833
Harmônicos	0,130117	0,126831	0,124303	0,122338
Desvio Freqüência (3 Hz)	0,305085	0,292974	0,288851	0,284112
Desvio fase (0,45 rad.)	0,168495	0,143501	0,137082	0,132801

De acordo com os resultados obtidos, nota-se que a estratégia que apresentou o menor erro médio quadrático, em todos os casos, foi a da atualização pela média, utilizando 3 (três) janelas do filtro média móvel. Nesse caso, o tamanho de cada janela foi igual a 256 pontos, que é igual ao período da componente fundamental do sinal de entrada, considerando-se uma freqüência de 60 Hz e uma taxa de amostragem igual a 15360 Hz.

Considerando estes resultados preliminares será utilizado, para atualização dos coeficientes do filtro notch acoplado à malha E-QPLL, nas simulações mostradas a seguir, o método de atualização pela média, utilizando 3 (três) janelas.

3.4 Comparação de desempenho entre as estruturas E-QPLL e QPLL

Serão apresentados, nesta seção, resultados comparativos entre o desempenho das malhas E-QPLL e QPLL. Nas simulações a seguir, os parâmetros das malhas foram ajustados de modo que ambas obtenham uma mesma rejeição a ruídos em regime permanente, aproximadamente. Assim, espera-se obter uma comparação do desempenho

das malhas através da análise do seu tempo de convergência. Será efetuada, a seguir, uma descrição mais pormenorizada da estratégia de ajuste dos parâmetros das malhas.

Inicialmente, a malha E-QPLL foi ajustada com os parâmetros $k_p = k_i = 75$. O filtro-notch acoplado à sua entrada foi ajustado com um fator notch $\rho = 0,996$ e foi utilizada uma taxa de amostragem igual a 15360 Hz. O método de atualização do filtro notch foi pela média, utilizando 3 (três) janelas como período de atualização. Em seguida, foi introduzido na malha E-QPLL um sinal descrito pela seguinte equação:

$$y(n) = \cos\left(\frac{2\pi n f_0}{f_s}\right) + b(n), \quad (3.11)$$

onde $f_0 = 60$ Hz, $f_s = 15360$ Hz e $b(n)$ é um ruído aditivo gaussiano com variância igual a $\sigma^2 = (0,1)^2$. Calculou-se, a seguir, o erro médio quadrático entre o sinal sintetizado pela malha E-QPLL e a componente fundamental da entrada, com o erro calculado entre $t = 0,5$ s e $t = 1,0$ s. Nesse caso, com a malha E-QPLL inicializada com os parâmetros descritos, o erro médio quadrático obtido será igual a 0,00007. Ajustou-se então a malha QPLL com os parâmetros $k_p = k_i$ variando desde $k_p = k_i = 20$ até $k_p = k_i = 250$ e calculou-se, para cada valor, o erro médio quadrático entre a componente fundamental da entrada e o sinal sintetizado pela malha. A Figura 3.10 mostra a variação do erro quadrático médio para cada valor de ajuste dos parâmetros da malha.

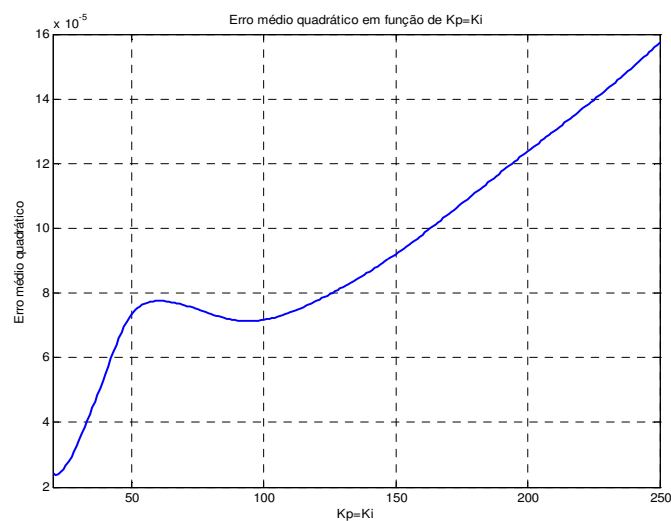


Figura 3.10 – Variação do erro médio quadrático para os valores $k_p = k_i$ variando desde $k_p = k_i = 1$ até $k_p = k_i = 250$.

Nota-se que, para o valor $k_p = k_i = 35$, a malha QPLL atinge aproximadamente uma mesma rejeição a ruídos que a malha E-QPLL ajustada com os valores descritos anteriormente. Desse modo, nas simulações a seguir, os valores $k_p = k_i = 35$ serão utilizados no ajuste da malha QPLL. A Tabela 3.5 mostra as diferentes situações simuladas para efeitos de comparação entre as duas versões do QPLL.

Tabela 3.5 - Casos Simulados

Caso	Descrição
1	Sinal senoidal com ruído gaussiano aditivo ($SNR= 15dB$). A amplitude varia de 1 pu para 0.8 pu
2	Harmônicos conforme descrito por (3.12). A amplitude da fundamental varia de 1.2 pu para 0.8 pu
3	Mudança na frequência em degrau, mais os harmônicos do caso 2. A frequência sofre uma variação de 3 Hz
4	Estimação do 3º harmônico na presença da componente fundamental.
5	Sinal com flicker (modulação em amplitude)
6	Variação senoidal da frequência

3.4.1 - Sinal senoidal com ruído gaussiano (Caso 1)

Neste primeiro caso um sinal senoidal com ruído gaussiano aditivo sofre uma variação na amplitude de 1,0 p.u. para 0,8 p.u., em $t = 0,5$ segundos. A SNR do sinal de entrada é de 15 dB para a situação de 0,8 p.u. de amplitude. A Figura 3.11 mostra as estimações de amplitude e frequência das malhas E-QPLL e QPLL. De acordo com as simulações nota-se que, para uma mesma rejeição a ruídos em regime permanente, a malha E-QPLL possui uma resposta transitória aproximadamente 0,15 segundos mais rápida, para a aquisição de amplitude e 0,05 segundos mais rápida, para a aquisição de frequência, comprovando a sua superioridade em relação à malha QPLL. Na mesma figura, podem-se ver as linhas que delimitam os valores de estimação de frequência e amplitude que estão dentro de uma faixa de erro menor que 1%.

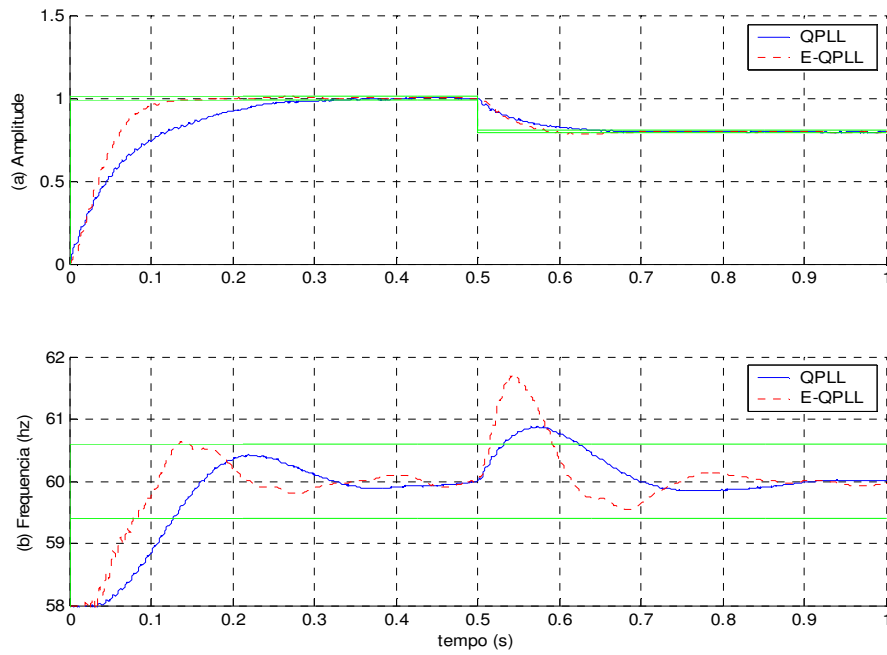


Figura 3.11 - Estimação para o caso 1. (a) Amplitude; (b) Frequência

3.4.2 - Estimação em presença de harmônicos (Caso 2)

Para este caso, o sinal de entrada é poluído por componentes harmônicos. A Figura 3.12 mostra a resposta das malhas *E-QPLL* e *QPLL*, para um sinal de entrada dado por:

$$\begin{aligned}
 u(n) = & \cos\left(\frac{2.\pi.n.f_0}{f_s}\right) + 0.20.\text{sen}\left(\frac{3.2.\pi.n.f_0}{f_s}\right) + 0.10.\text{sen}\left(\frac{5.2.\pi.n.f_0}{f_s}\right) + 0.08.\text{sen}\left(\frac{9.2.\pi.n.f_0}{f_s}\right) \\
 & + 0.06.\text{sen}\left(\frac{11.2.\pi.n.f_0}{f_s}\right),
 \end{aligned} \tag{3.10}$$

onde $f_0 = 60$ Hz e $f_s = 15360$ Hz. No instante $t = 0,5$ segundos, a amplitude da componente fundamental do sinal de entrada varia de 1,0 p.u. para 0,8 p.u. De acordo com os resultados das simulações, mostrados na Figura 3.12, pode-se comprovar, mediante inspeção visual, a superioridade da malha *E-QPLL* em relação a *QPLL*, haja vista que esta alcança um estado de regime permanente mais rapidamente, para os dois casos (aquisição de amplitude e frequência) e apresentando ainda, em regime permanente, uma mesma rejeição a ruídos.

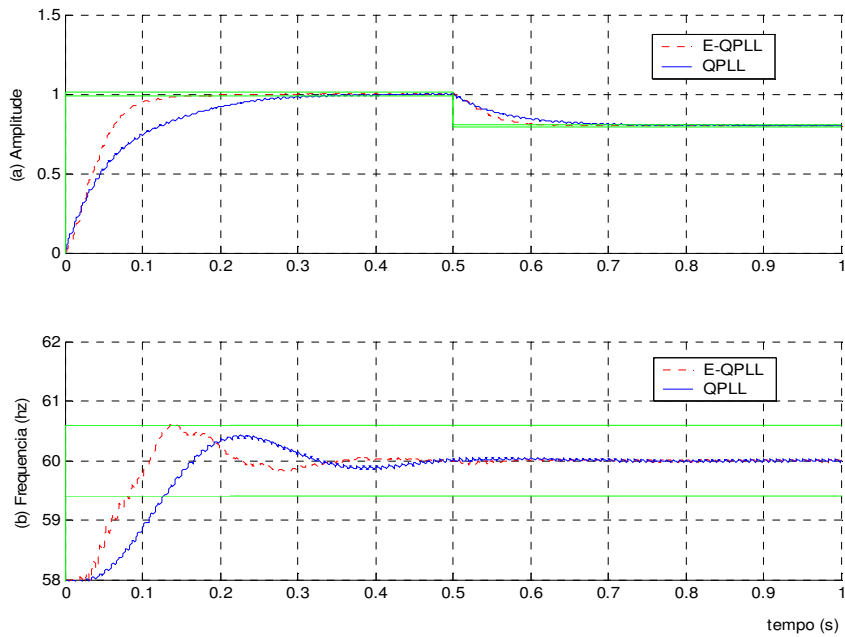


Figura 3.12 - Estimação para o caso 2. (a) Amplitude; (b) Freqüência

3.4.3 - Mudança da Freqüência em degrau (caso 3)

Para este terceiro caso, a freqüência da componente fundamental do sinal de entrada sofre uma mudança em degrau de 60 Hz para 63 Hz e as componentes harmônicas do sinal são as mesmas do caso 2 anterior. A Figura 3.13 mostra a estimação da amplitude e da freqüência das malhas QPLL e E-QPLL. Vale ressaltar que é possível observar-se, novamente, a ocorrência de um maior tempo de convergência para o algoritmo QPLL, para o mesmo erro em regime permanente. Quando se considera o desempenho na aquisição de freqüência, pode-se também notar que, na inicialização das malhas, a estrutura QPLL apresentou um desempenho superior. Apesar deste desempenho superior inicial, deve ser ressaltado, contudo, que quando ocorre uma variação na forma de um degrau, na freqüência de 3 Hz, no instante de tempo igual a $t = 0,3469$ segundos, isto resulta um maior tempo de acomodação da resposta para a malha E-QPLL.

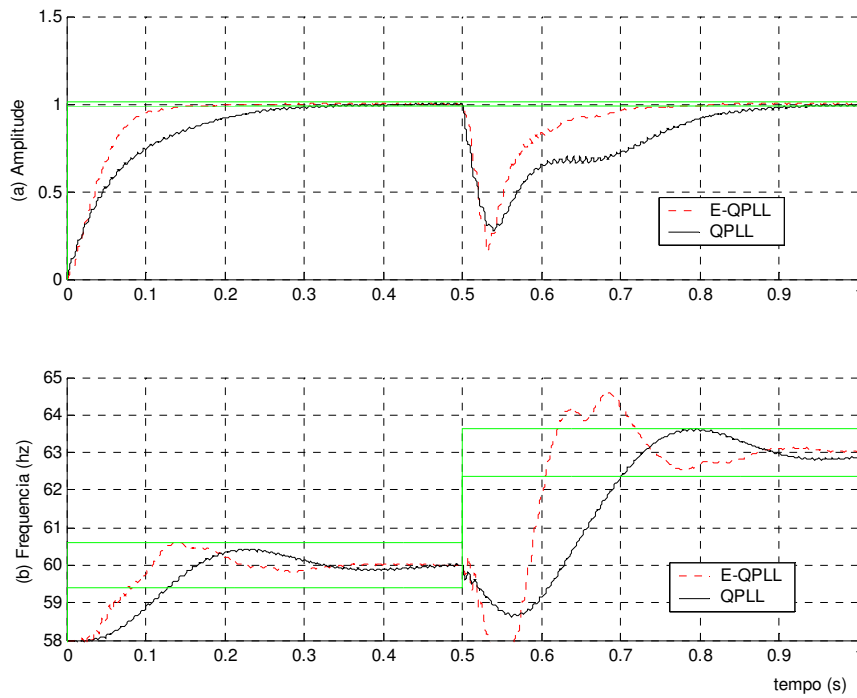


Figura 3.13 - Estimação para o caso 3. (a) Amplitude; (b) Frequência

3.4.4 - Estimação do 3º harmônico (caso 4)

Deseja-se, neste caso, estimar o 3º harmônico na presença da componente fundamental. O sinal utilizado para a estimação é expresso por:

$$u(n) = \text{sen}\left(\frac{2\pi n f_0}{f_s}\right) + 0.2 \cdot \text{sen}\left(\frac{3 \cdot 2\pi n f_0}{f_s}\right) \quad (3.13)$$

As malhas QPLL e E-QPLL foram inicializadas com uma frequência igual a $3\omega_0$, com os resultados da estimação mostrados na Figura 3.14. Como a componente fundamental não é filtrada no QPLL, o algoritmo estima a componente fundamental ao invés do terceiro harmônico. No caso do E-QPLL, o filtro *notch* reduz a energia da componente fundamental, fazendo com que o resultado da estimação seja a amplitude do 3º harmônico, como desejado. Isso indica a possibilidade da malha E-QPLL atuar como uma estrutura detectora de componentes harmônicos de um sinal de entrada qualquer, através da correta inicialização do filtro notch.

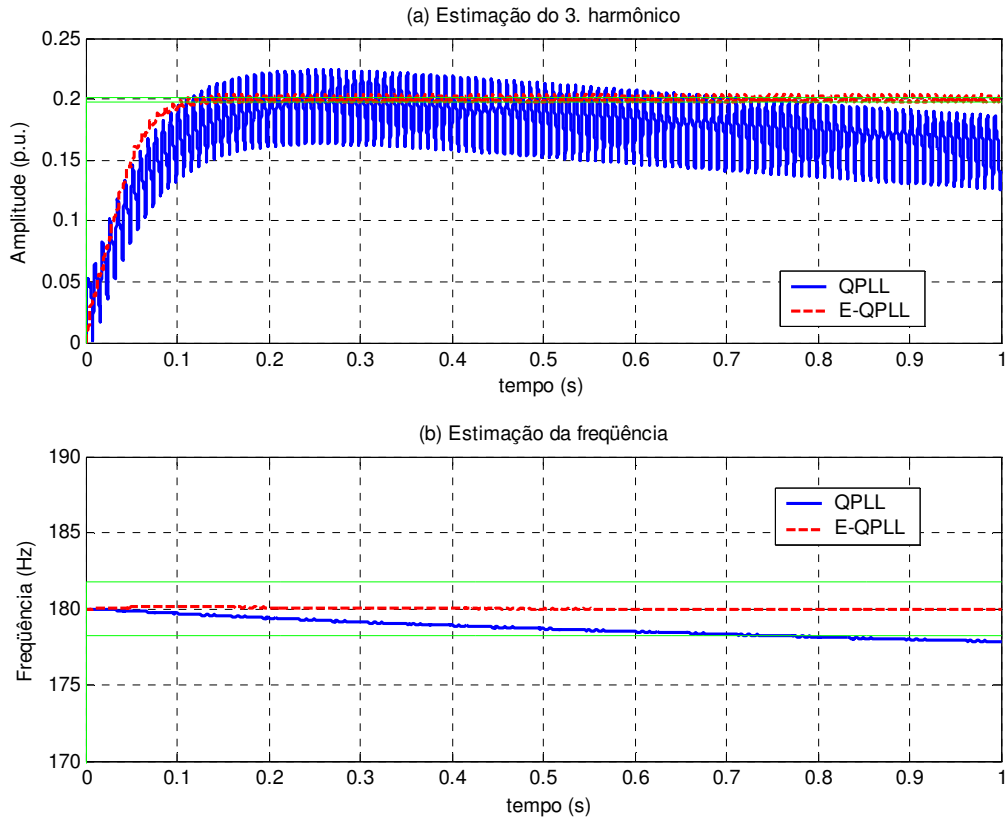


Figura 3.14- Estimação para o caso 4. (a) Amplitude; (b) Freqüência

3.4.5 - Estimação de sinal com “flicker” (caso 5)

Grandes cargas não lineares como fornos a arco geram no sinal de tensão uma modulação em amplitude, onde a componente fundamental (60 Hz) representa a portadora e a carga não linear produz a componente modulante. Estas oscilações de baixa freqüência são conhecidas como “flicker” ou cintilação luminosa [24]. No presente exemplo será avaliado o desempenho das duas estruturas PLL quando operando em presença de um sinal contaminado com “flicker”. A modelagem matemática do processo será efetuada de acordo com a sugerida por [25], ou seja, o sinal de entrada da malha será dado por:

$$u(n) = [1 + a(n)] \text{sen} \left(\frac{2\pi n f_0}{f_s} + \delta \right), \quad (3.14)$$

onde

$$a(n) = 0,05 \text{sen} \left(\frac{2,5\pi n}{f_s} \right) + 0,02 \text{sen} \left(\frac{10\pi n}{f_s} \right) \quad (3.15)$$

A Figura 3.15 mostra a estimação da amplitude e frequência para os dois algoritmos sob análise. Notar que a malha E-QPLL apresenta uma convergência mais rápida que a malha QPLL, provando que a presença do filtro notch não interfere no funcionamento da malha, se o sinal de entrada estiver contaminado por *flickers*. Entretanto, é possível notar, mediante inspeção visual, que ambas as malhas apresentaram erros na estimação com amplitudes superiores a 1%. Uma possível solução para este problema poderia ser tentada na forma de ajuste dos coeficientes k_p e k_i da malha com valores mais elevados, propiciando uma resposta dinâmica mais rápida para as malhas.

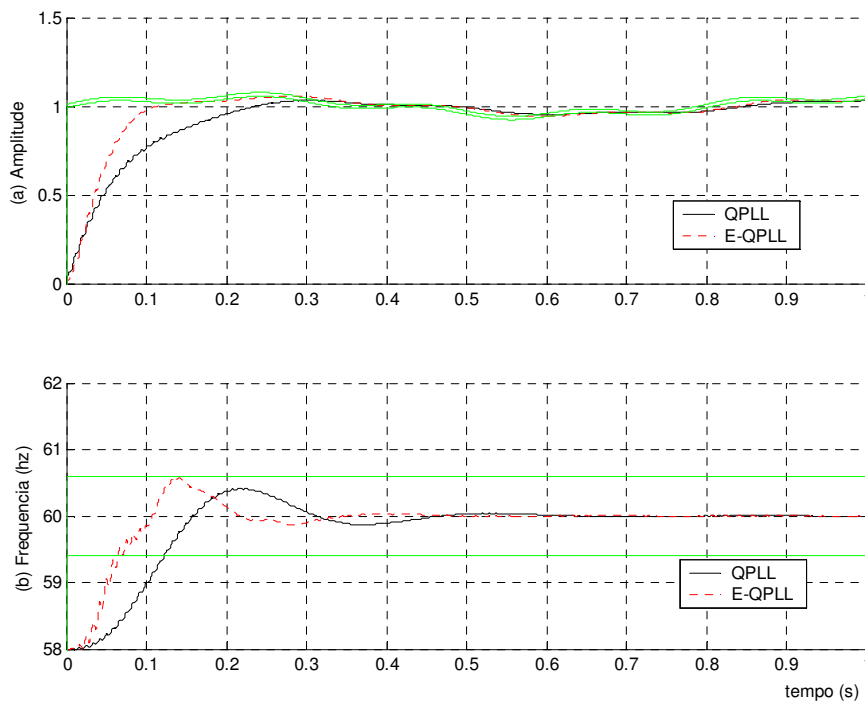


Figura 3.15- Estimação para o caso 5. (a) Amplitude (b) Frequência

3.4.6 - Estimação com variação senoidal da frequência (caso 6)

Neste exemplo será considerado o caso no qual a frequência do sinal apresenta uma variação senoidal. Trata-se do tipo de variação mais comum, encontrada geralmente na prática, e decorre, geralmente, de oscilações eletromecânicas existentes nos sistemas. O sinal usado para esta simulação é expresso por

$$u(t) = A_0 \cdot \sin \left(\int_0^t f(\tau) d\tau + \delta \right) + \eta(t) \quad (3.16)$$

onde a frequência é assumida como $f(t) = (1 + 0.05 \sin(4\pi t)) f_0$, o que significa uma taxa de alteração igual a 2 Hz, com o valor da frequência variando na faixa $f_0(1 \pm 0.05)$ Hz. Para esse caso, os parâmetros das malhas foram reajustados de forma a obter-se uma resposta transitória mais rápida pela natureza oscilante da frequência da componente fundamental do sinal de entrada. A Figura 3.16 mostra o resultado desta simulação.

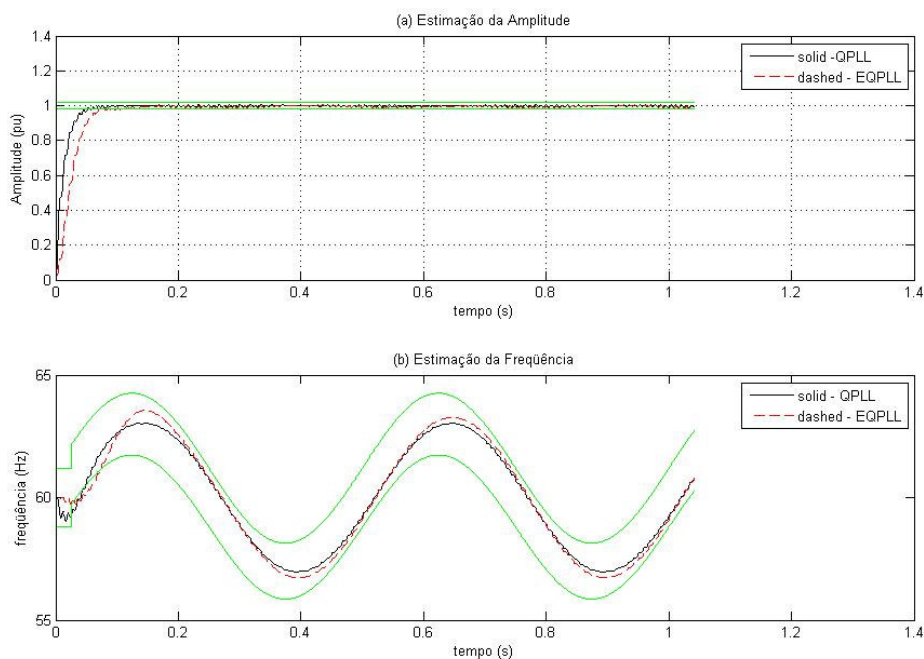


Figura 3.16 - Simulação para o caso 6 – Variação senoidal da frequência

Pode-se afirmar, pela inspeção visual dos resultados das estimações de frequência e amplitude das malhas, que os resultados são satisfatórios, com ambas as simulações obtendo resultados dentro de uma faixa de variabilidade abaixo de 1%. Isto indica a viabilidade de implementação da malha E-QPLL proposta nesse trabalho, em ambientes

onde a freqüência da componente fundamental do sinal de entrada apresente uma comportamento do tipo oscilatório.

3.5 - Estimação da Fase

Nos casos simulados anteriormente não foi apresentada a estimação da fase do fasor da componente fundamental do sinal de entrada. Esta estimação da fase requer cuidados especiais, especialmente quando a técnica de estimação utilizar estruturas QPLL ou E-QPLL, razão pela qual será tratada separadamente nesta seção.

Supor que um sinal senoidal de freqüência constante é inserido na entrada de uma estrutura QPLL,

$$u(t) = A \sin(\omega_0 t + \delta) = A \sin(\psi(t)) \quad (3.17)$$

Conforme apresentado no Capítulo 2, Seção 2.4, o modelo matemático para o QPLL é dado pela Equação 2.26, repetida a seguir por conveniência:

$$y(t) = K_s \cdot \text{sen}(\phi(t)) + K_c \cdot \text{cos}(\phi(t)), \quad (3.18)$$

onde:

$$A = \sqrt{K_c^2 + K_s^2}, \quad (3.19\text{-a})$$

$$\psi(t) = \phi(t) + \tan^{-1} \frac{-K_s}{K_c}$$

(3.19-b)

$$\phi(t) = \int_0^t \omega(\tau) d\tau \quad .$$

(3.19-c)

Para o caso de freqüência constante ω_0 , tem-se que:

$$\psi(t) = \omega_0 t + \tan^{-1} \frac{-K_s}{K_c}, \quad (3.20)$$

Para este caso, tem-se ainda que:

$$\delta = \tan^{-1} \frac{-K_s}{K_c}. \quad (3.21)$$

Considerar agora a estimação da fase para o caso 1 da Tabela 3.5 anterior e considerar ainda que o sinal possui uma fase inicial de 0,4 rad. A Figura 3.17 mostra a estimação da fase utilizando (3.19). Fica claro, por inspeção visual, que o valor estimado está longe do valor ideal, embora o modelo pareça correto.

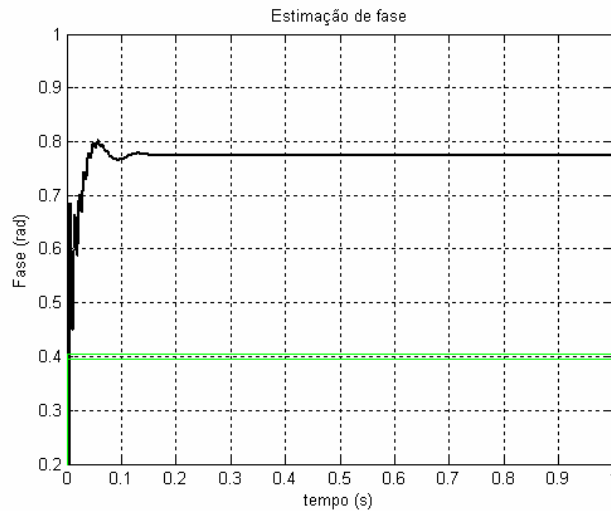


Figura 3.17- Estimação para a fase - caso 1

O erro na estimação da fase pode ser explicado da seguinte maneira: embora a frequência do sinal de entrada apresente um valor constante, a frequência estimada não é constante e passa por um período transitório, conforme mostrado na Figura 3.10. Este transitório, ou esta variação na estimação da frequência, provoca uma defasagem no sinal que precisa ser compensada pela malha PLL para reduzir a energia do erro. Esta variação é incorporada nas estimções de K_c e K_s . Para melhor entender este problema, imaginar que a frequência estimada apresente a seguinte expressão:

$$\omega(t) = \omega_0 + \Delta\omega \cdot (\mu(t) - \mu(t - \tau)) \quad (3.22)$$

onde $\mu(t)$ é a função degrau unitário. A partir de (3.17-c) encontra-se que,

$$\phi(t) = \int_0^t \omega(\tau) d\tau = \omega_0 t + \Delta\omega \cdot \tau \quad (3.23)$$

Chamando $\Delta\omega \cdot \tau = \delta_0$, o sinal estimado passa a ter a seguinte forma:

$$y(t) = A \cdot \text{sen}(\omega_0 t + \delta_0 + \delta) = K_s' \text{sen}(\omega_0 t) + K_c' \cos(\omega_0 t) \quad (3.24\text{-a})$$

$$K_s' = A \cdot \cos(\delta + \delta_0) \quad (3.24\text{-b})$$

$$K_c' = A \cdot \sin(\delta + \delta_0) \quad (3.24\text{-c})$$

Notar que, para esta situação, o cálculo da amplitude, Equação 3.19a, não se altera, porém para determinar a fase original deve-se utilizar a seguinte equação:

$$\delta = \tan^{-1}(-K_s' / K_c') - \delta_0 \quad (3.25)$$

A partir da Equação 3.25 pode-se compreender porque a fase estimada pela Equação 3.21 apresenta um deslocamento positivo: a dificuldade está na determinação do valor de δ_0 para o caso geral. Esta dificuldade impõe a necessidade de se modificar a maneira de medir a fase. O método sugerido a seguir utiliza o conceito de fase total e um sinal de referência.

3.5.1 - Fase total e sinal de referência

Considere o sinal de referência definido na Equação 3.26,

$$u_r(t) = \sin(\phi_r(t)) \quad (3.26)$$

onde $\phi_r(t) = \omega_0 t$ para o caso de frequência constante. A Equação 3.19b, repetida a seguir por conveniência,

$$\psi(t) = \phi(t) + \tan^{-1} \frac{-K_s}{K_c}$$

representa a fase total do sinal estimado. A fase do sinal estimado pode ser encontrada simplesmente tomando-se a Equação 3.26, que conduzirá a

$$\delta = \hat{\phi}(t) - \phi_r(t) \quad (3.27)$$

A Figura 3.18 mostra a fase obtida a partir de (3.27) para o sinal do caso 1 analisado anteriormente:

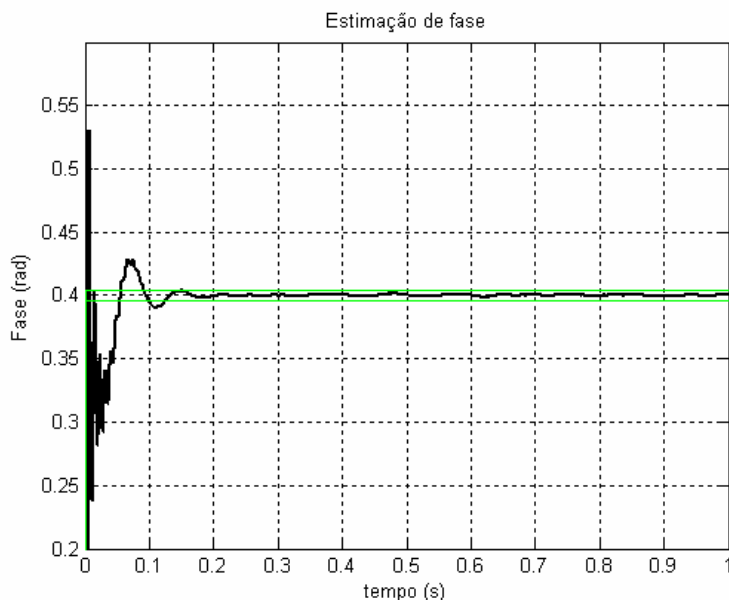


Figura 3.18 - Estimação para a fase - caso 1 – usando conceito de fase total

3.5.2 - Frequência variante em degrau

O método anterior pode agora ser estendido para o caso de frequência variante no tempo. Será considerado o caso simples em que a frequência apresenta uma variação em degrau no instante t_0 . A fase total para o sinal de referência é expressa por (3.28),

$$\phi_r(t) = \int_0^t \omega(\tau) d\tau = \begin{cases} \omega_0 t, & t \leq t_0 \\ -\Delta\omega t_0 + (\omega_0 + \Delta\omega)t, & t > t_0. \end{cases} \quad (3.28)$$

Notar que, após a mudança de frequência o sinal de referência apresenta uma nova fase dada pelo termo $-\Delta\omega t_0$. Para o caso em que o sinal a ser medido é da forma (3.15) com fase inicial δ_0 , após a mudança de frequência a nova fase será $\delta_n = \delta_0 - \Delta\omega t_0$. Isto está ilustrado na Figura 3.19. Notar que a inclinação da fase total é modificada com a frequência. Nesta figura está ilustrado o caso que a frequência sofre um aumento em t_0 .

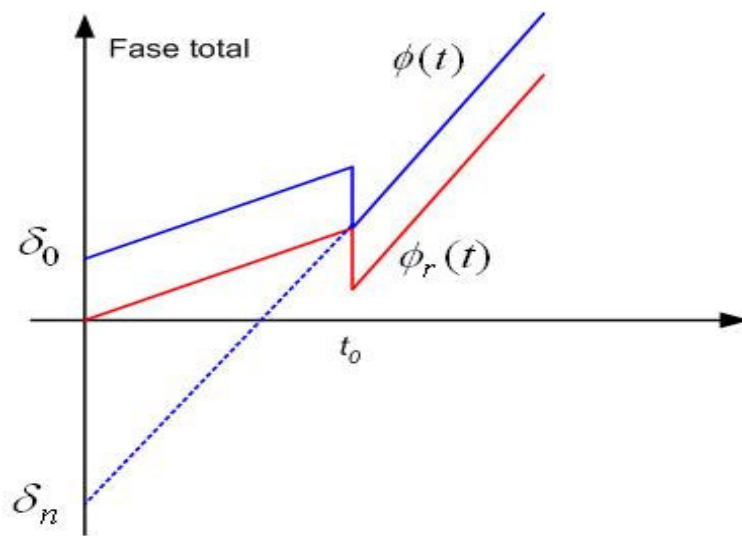


Figura 3.19 - Interpretação da fase para sinal com freqüência variante no tempo.

A Figura 3.20, finalmente, ilustra o resultado da simulação para o caso 3 da Tabela 3.1. Notar que a referência para a fase ideal é modificada conforme discutido anteriormente. A fase inicial considerada neste exemplo foi de -1 radiano e a variação de freqüência de 5%. A mudança de freqüência aconteceu no instante $t = 0,3469$ segundos.

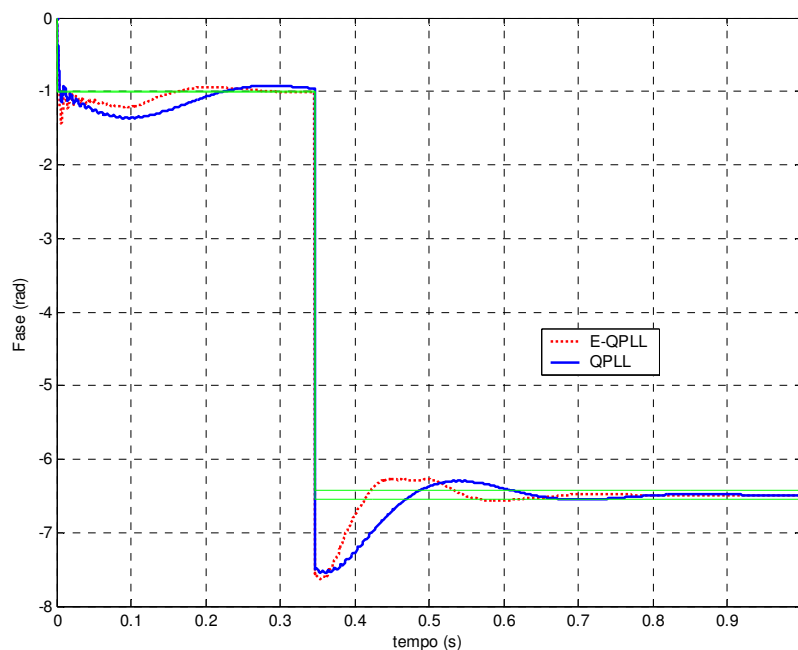


Figura 3.20- Estimação da fase para o caso 3 analisado anteriormente

Notar que, na estimação de fase, a malha E-QPLL atinge um estado permanente de aquisição da fase, com um erro menor que 1%, com um menor tempo de convergência que a malha QPLL original.

3.6- Conclusões

Esse capítulo apresentou a descrição de uma malha PLL do tipo Quadratura (QPLL), acrescida de um filtro passa-banda adaptativo acoplado à sua entrada. A filtragem passa-banda é obtida através da utilização de um filtro do tipo *notch*, centrado na frequência da componente fundamental do sinal de entrada, com o objetivo de eliminar essa componente para, logo após, identificá-la através de uma simples equação algébrica. A filtragem passa-banda tem como objetivo aumentar a relação *SNR* do sinal de entrada, bem como atenuar as componentes harmônicas e inter-harmônicas de baixa frequência que estejam presentes no mesmo.

Sabendo-se que o valor da frequência da componente fundamental do sinal de entrada não apresenta um valor fixo, foi necessária uma estratégia de ajuste dos coeficientes do filtro, de modo que o mesmo sempre elimine a componente fundamental do sinal de entrada. Desse modo, foram propostos dois esquemas de atualização dos coeficientes do filtro; atualização pelo valor instantâneo da frequência detectada pela malha QPLL ou por uma média desse valor, integralizada por um período de tempo pré-especificado. De acordo com as simulações realizadas, comprovou-se a superioridade do esquema de atualização pela média. Logo após, foram realizadas diversas simulações computacionais, comparando o desempenho da malha proposta nesse capítulo com a malha descrita no capítulo anterior. De acordo com os resultados obtidos, comprovou-se a superioridade da malha proposta, pois a mesma alcançou uma resposta transitória mais rápida que a malha QPLL original em diversas situações, para uma mesma rejeição a distúrbios, em regime permanente.

Capítulo 4

Estrutura Melhorada e Robusta do Algoritmo PLL – “*Enhanced and Robust Quadrature Phase-Locked-Loop*”

4.1 Introdução

Foi apresentada, no Capítulo 2 deste trabalho, a descrição de uma malha PLL do tipo quadratura (“QPLL”), que pode efetuar a aquisição dos parâmetros da componente fundamental de um sinal de entrada qualquer (amplitude, fase e frequência), utilizando, para isto, um procedimento algorítmico baseado no método de minimização denominado *gradiente-descendente*. Foi realizada, no Capítulo 3, a demonstração de uma malha PLL denominada *E-QPLL* (“*Enhanced-Quadrature-Phase-Locked-Loop*”), cujo desenvolvimento foi baseado na introdução de um filtro-notch adaptativo em sua entrada, com o objetivo de aumentar a rejeição a ruídos da mesma, sem sacrificar demasiadamente a velocidade da sua resposta dinâmica.

A implementação das malhas de QPLL ou *E-QPLL* em sistemas digitais, como processadores digitais de sinais (DSPs) ou FPGAs (*Field Programmable Gate Arrays*), requer a discretização das equações da malha e, também, do filtro notch adaptativo presente em sua entrada, conforme visto nos capítulos anteriores. As simulações das malhas PLL digitais realizadas nos capítulos anteriores foram efetuadas através da utilização do ambiente MATLAB®, que utiliza em sua estrutura interna uma aritmética de ponto flutuante, equivalente a um sistema de aritmética de ponto fixo de, pelo menos, 64 bits [28]. De acordo com os resultados das simulações, comprovou-se um funcionamento satisfatório das malhas descritas.

Entretanto, conforme será demonstrado no decorrer desse capítulo, ao se realizar a implementação da malha discreta utilizando um sistema de aritmética de ponto fixo

com um número reduzido de bits como, por exemplo, 16 bits, há uma degradação no seu desempenho. Este fato ocorre principalmente devido aos erros de arredondamento na quantização dos coeficientes da malha e dos filtros a ela relacionados. Sabe-se que quando um sistema contínuo é mapeado para o domínio discreto os seus pólos e zeros são mapeados do plano complexo para o plano z . Entretanto, ao se efetuar a quantização dos coeficientes do sistema considerado, há um deslocamento desses pólos e zeros, que se afastam da localização ideal no interior do círculo unitário, fazendo com que o sistema não se comporte exatamente como o seu equivalente no sistema contínuo, podendo até se tornar instável. Além desses erros, também podem também ser citados os ruídos de quantização presente em operações de multiplicação e divisão em ponto fixo. Estes erros podem ser reduzidos se a estrutura de implementação do sistema for adequadamente escolhida. Também pode ser provado que, conforme se aumenta a taxa de amostragem utilizada no sistema, maior é o deslocamento dos pólos e zeros do seu lugar ideal, aumentando o erro da malha discreta [12,14,29].

Considerando os problemas expostos, associados à discretização e implementação da malha em sistemas digitais de ponto fixo com número reduzido de bits, propõe-se, neste capítulo, a síntese de uma malha QPLL robusta, doravante denominada *ER-QPLL* (“*Enhanced and Robust Quadrature Phase-Locked-Loop*”), cujo objetivo principal é a redução dos erros mencionados. A síntese desta malha baseia-se na estrutura *E-QPLL* (“*Enhanced Quadrature Phase-Locked-Loop*”) descrita no capítulo anterior, mas seu desenvolvimento será efetuado a partir de uma discretização baseada na Transformada Gama (γ) [14]. Espera-se, com a utilização da Transformada Gama (γ), garantir um desempenho satisfatório da malha, mesmo quando esta é implementada em sistemas digitais com número reduzido de bits e utilizando elevadas taxas de amostragem.

A principal motivação do desenvolvimento de uma malha robusta de QPLL, factível de implementação em sistemas de aritmética de ponto fixo com número reduzido de bits, deve-se ao fato que sistemas que utilizam em seu “*hardware*” interno microprocessadores de ponto fixo, apresentam custo financeiro significativamente menor, se comparados a sistemas com microprocessadores de ponto flutuante ou microprocessadores com estrutura robusta de, no mínimo, 32 bits. Convém lembrar que é possível a emulação de aritmética de ponto flutuante em microprocessadores de ponto fixo. Entretanto, tal emulação acarreta enormes prejuízos ao sistema, no que diz respeito

à velocidade de processamento do mesmo, o que torna tal estratégia muitas vezes inviável.

4.2 O operador delta (δ)

No intuito de otimizar o desempenho da implementação da malha de E -QPLL descrita no capítulo anterior, em sistemas discretos de aritmética de ponto fixo, é introduzido nessa secção o operador delta (δ) [14], definido pela equação:

$$\delta\{x[k]\} = \frac{x[k+1] - x[k]}{\Delta}, \quad (4.1)$$

onde $x[k]$ é uma seqüência representando um sinal $x(n)$ discretizado e Δ , contido no intervalo $(0,1]$, é um parâmetro de otimização, geralmente relacionado à taxa de amostragem do sistema. É importante ressaltar que o parâmetro Δ não está necessariamente relacionado à taxa de amostragem, mas deve ser visto como um parâmetro de otimização do sistema.

Uma análise da equação (4.1) mostra que o operador delta (δ) é semelhante à derivada da seqüência $x[k]$, se for considerado que o valor de Δ é igual à taxa de amostragem utilizada pelo sistema. Ou seja:

$$\lim_{\Delta \rightarrow 0} \delta(x[k]) = \left. \frac{dx(t)}{dt} \right|_{x=x(k\Delta)}. \quad (4.2)$$

A última relação é mais uma vantagem do uso do operador delta (δ) em detrimento do uso do tradicional operador deslocador q (tradicionalmente relacionado ao operador z), uma vez que é possível estabelecer uma relação direta entre um sistema contínuo descrito por uma equação diferencial e o sistema no domínio discreto equivalente, simplesmente substituindo a derivada de uma função contínua pelo operador delta (δ). A partir daí é possível resolver o sistema discreto resultante utilizando equações a diferenças. Entretanto, como foge ao escopo dessa dissertação, essa propriedade do operador delta (δ) não será demonstrada nesse trabalho.

4.2.1 Equações a diferenças baseada no uso do operador δ

Considere uma função de transferência discreta dada por:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b_{n-1} \cdot z^{n-1} + b_{n-1} \cdot z^{n-2} + \dots + b_0}{a_n \cdot z^n + a_{n-1} \cdot z^{n-1} + \dots + a_0}, \quad (4.3)$$

A equação a diferenças referente a esta função de transferência é dada por:

$$a_n y[k+n] + a_{n-1} y[k+n-1] + \dots + a_0 y[k] = b_{n-1} u[k+n-1] + \dots + b_0 u[k], \quad (4.4)$$

Reescrevendo-se a última equação utilizando o operador deslocamento q^n , que tem como objetivo deslocar o sinal discretizado por n amostras, obtém-se:

$$a_n \cdot q^n \cdot y[k] + a_{n-1} \cdot q^{n-1} \cdot y[k] + \dots + a_0 \cdot y[k] = b_n \cdot q^n \cdot u[k] + b_{n-1} \cdot q^{n-1} \cdot u[k] + \dots + b_0 \cdot u[k], \quad (4.5)$$

A obtenção da equação a diferenças no domínio do operador delta(δ) pode ser conseguida diretamente, bastando para isto lembrar que :

$$\delta\{x[k]\} = \frac{q \cdot x[k] - x[k]}{\Delta} \leftrightarrow \delta = \frac{q-1}{\Delta}, \quad (4.6)$$

o que conduz a:

$$q = \Delta \delta + 1, \quad (4.7)$$

Deste modo, a representação da equação a diferenças no domínio do operador delta (δ) será dada por:

$$\alpha_n \delta^n y[k] + \alpha_{n-1} \delta^{n-1} y[k] + \dots + \alpha_0 y[k] = \beta_n \delta^n u[k] + \beta_{n-1} \delta^{n-1} u[k] + \dots + \beta_0 u[k], \quad (4.8)$$

e a relação entre os coeficientes será obtida a partir de:

$$\begin{bmatrix} \alpha_n \\ \alpha_{n-1} \\ \vdots \\ \alpha_2 \\ \alpha_1 \\ \alpha_0 \end{bmatrix} = \begin{bmatrix} C_n \Delta^n & 0 & 0 & \cdots & 0 & 0 \\ C_{n-1} \Delta^{n-1} & C_{n-1} \Delta^{n-1} & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ C_2 \Delta^2 & C_2 \Delta^2 & C_2 \Delta^2 & & 0 & 0 \\ C_1 \Delta & C_1 \Delta & C_1 \Delta & & C_1 \Delta & 0 \\ 1 & 1 & 1 & & 1 & 1 \end{bmatrix} \begin{bmatrix} a_n \\ a_{n-1} \\ \vdots \\ a_2 \\ a_1 \\ a_0 \end{bmatrix}, \quad (4.9)$$

onde $C_j^k = \frac{k!}{j!(k-j)!}$.

Para se resolver a equação a diferenças no domínio do operador delta (δ) é necessário, primeiramente, definir o operador causal *delta inverso*, definido por δ^{-1} . Considere que:

$$\delta^{-1}(\delta x[k]) = x[k], \quad (4.10)$$

Assim:

$$\delta^{-1} \left\{ \frac{x[k+1] - x[k]}{\Delta} \right\} = x[k], \quad (4.11)$$

Logo:

$$\frac{\delta^{-1}\{x(k+1)\} - \delta^{-1}\{x(k)\}}{\Delta} = x[k], \quad (4.12)$$

Finalmente:

$$\delta^{-1}\{x[k+1]\} = \Delta x[k] + \delta^{-1}\{x[k]\}, \quad (4.13)$$

Chamando $\delta^{-1}\{x[k]\} = y[k]$, obtém-se:

$$y[k] = \Delta x[k-1] + y[k-1], \quad (4.14)$$

A última equação pode ser representada graficamente através da Figura 4.1.

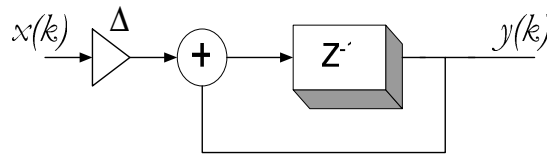


Figura 4.1 - Operador δ^{-1}

Desse modo, após a definição do operador causal δ^{-1} , torna-se possível resolver a equação a diferenças descrita em (4.8) através do uso de diversas estruturas, como a DFII, DFII, etc. [12]. Entretanto, de acordo com [12], a estrutura que produz melhores resultados referentes a erros de quantização é a estrutura DFII (Direct Transposed Form II). No caso da resolução de uma equação a diferenças de segunda ordem, a estrutura é dada pela Figura 4.2.

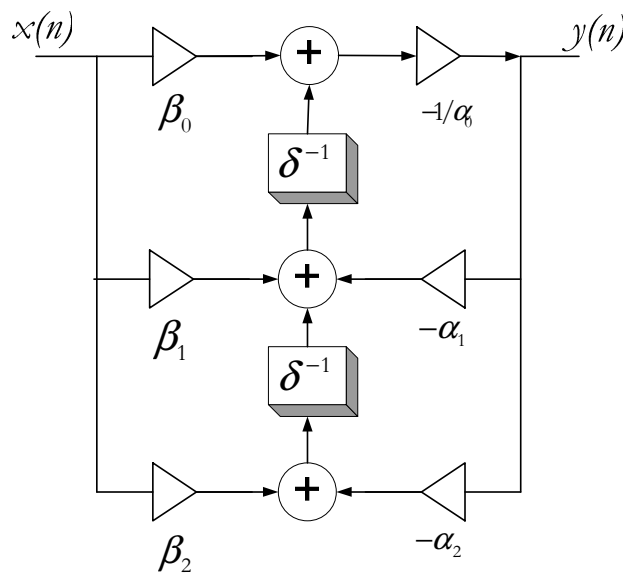


Figura 4.2 - DFII

4.2.2 A Transformada Gama (γ)

Será descrita, nesta secção, o desenvolvimento da Transformada Gama(γ) de uma seqüência $x[k]$ qualquer. A partir desta transformada, torna-se então possível obter a função de transferência de um sistema discreto, no domínio do operador delta (δ) e, a partir daí, implementar a equação de diferenças resultante através dos métodos descritos anteriormente.

A transformada gama(γ) de uma seqüência discreta, $x[k]$, é dada pela equação:

$$D\{x[k]\} = X_\gamma(\gamma) = \sum_{k=-\infty}^{\infty} x[k](\Delta\gamma+1)^{-k}, \quad (4.15)$$

onde γ é uma variável complexa e a sua relação com o operador z será descrita adiante.

Sabendo-se que a transformada z da mesma seqüência $x[k]$ é dada por:

$$X_z(z) = \sum_{k=-\infty}^{\infty} x[k].z^{-k}, \quad (4.16)$$

pode-se então estabelecer a relação direta entre a transformada gama de uma seqüência e a sua transformada z através da seguinte relação:

$$\gamma = \frac{z-1}{\Delta}, \quad (4.17)$$

Desse modo, considere uma função de transferência discreta de segunda ordem, no domínio do operador z , expressa por:

$$H(z) = \frac{b_2.z^{-2} + b_1.z^{-1} + b_0}{a_2.z^{-2} + a_1.z^{-1} + a_0}, \quad (4.18)$$

A função de transferência discreta, no domínio do operador delta (δ), será expressa por:

$$H_\delta(\gamma) = H(z)\Big|_{z=\Delta\gamma+1}, \quad (4.19)$$

A aplicação da última equação leva a função de transferência discreta da função $H(z)$ para o domínio discreto:

$$H_\delta(\gamma) = \frac{\beta_0 + \beta_1.\gamma^{-1} + \beta_2.\gamma^{-2}}{\alpha_0 + \alpha_1.\gamma^{-1} + \alpha_2.\gamma^{-2}}, \quad (4.20)$$

Para se obter a equação a diferenças referente à função de transferência no domínio do operador delta (δ) será considerada, primeiramente, a transformada gama da seqüência $\delta\{y[k]\}$. Desse modo, têm-se:

$$D\{\delta y[k]\} = \sum_{k=0}^{\infty} (1 + \Delta\gamma)^{-k} \left[\frac{y[k+1] - y[k]}{\Delta} \right]; \quad (4.21)$$

A partir da última equação, obtém-se:

$$D\{\delta y[k]\} = \left\{ (1 + \Delta\gamma) \sum_{k=1}^{\infty} (1 + \Delta\gamma)^{-k} y[k] - \sum_{k=0}^{\infty} (1 + \Delta\gamma)^{-k} y[k] \right\}, \quad (4.22)$$

Uma simplificação da última equação fornece:

$$D\{\delta y[k]\} = \mathcal{Y}_{\delta}(\gamma) - (1 + \Delta\gamma) \cdot y[0], \quad (4.23)$$

Considerando como condição inicial $y[0]=0$, obtém-se:

$$D\{\delta y[k]\} = \mathcal{Y}_{\delta}(\gamma), \quad (4.24)$$

Do mesmo modo, é possível provar que:

$$D\{\delta^n \{x[k]\}\} = \gamma^n \cdot Y_{\delta}(\gamma), \quad (4.25)$$

Verifica-se então que a equação a diferenças referente à função de transferência no domínio do operador delta (δ) será dada por:

$$Y_{\gamma}(\gamma) = 1/\alpha_0 (\gamma^{-2} X_{\gamma}(\gamma) \alpha_2 + \gamma^{-1} X_{\gamma}(\gamma) \alpha_1 - \gamma^{-1} Y_{\gamma}(\gamma) \alpha_1 - \gamma^{-2} Y_{\gamma}(\gamma) \alpha_2), \quad (4.26)$$

Aplicando-se a transformada gama inversa em ambos os lados da equação (4.26), obtém-se:

$$y[k] = 1/\alpha_0 (\delta^{-2} x[k] \alpha_2 + \delta^{-1} x[k] \alpha_1 - \delta^{-1} y[k] \alpha_1 - \delta^{-2} y[k] \alpha_2), \quad (4.27)$$

A relação entre os coeficientes de um polinômio de segundo grau, no domínio do operador z e no domínio do operador delta (δ) é dada através da tabela a seguir:

Tabela 4.1 – Relação de parâmetros

β	b_0	α_0	a_0
β_1	$\frac{2b_0 + b_1}{\Delta}$	α_1	$\frac{2a_0 + a_1}{\Delta}$
β_2	$\frac{b_0 + b_1 + b_2}{\Delta^2}$	α_2	$\frac{a_0 + a_1 + a_2}{\Delta^2}$

É importante notar que existe uma relação direta entre a transformada de Laplace de uma seqüência e a sua transformada Gama, se for considerado o parâmetro delta como a taxa de amostragem do sistema considerado. A transformada de Laplace de uma função $y(t)$ é dada por:

$$Y(s) = \int_0^{\infty} e^{-st} y(t) dt, \quad (4.28)$$

Em sistemas discretos, a integral da última equação se transforma em um somatório e a transformada de Laplace da seqüência $x(k)$ se transforma na sua transformada Gama, que também é expressa por:

$$Y'(s) = \sum_{k=0}^{\infty} e^{-sk\Delta} y(k\Delta), \quad (4.29)$$

A última equação é equivalente a equação (4.28). A equivalência entre estas equações pode ser estabelecida lembrando que a variável $e^{s\Delta}$ pode ser substituída pela equação:

$$e^{s\Delta} = 1 + \Delta\gamma, \quad (4.30)$$

No limite, têm-se:

$$\lim_{\Delta \rightarrow 0} e^{-s\Delta} = 1 + \Delta s \quad (4.31)$$

Ou seja, pode-se afirmar que quando a taxa de amostragem de um sistema é suficientemente alta em relação à banda de frequência do sinal, a transformada gama de uma seqüência é equivalente à sua transformada de Laplace.

Existem referências à Transformada Gama desde 1940, quando foi utilizada para implementação de sistemas discretos utilizando-se altas taxas de amostragem. A tabela a seguir apresenta a relação de várias transformadas gamas e transformadas Z de seqüências conhecidas:

Tabela 4.2 – Transformadas Z e Gama

Seqüência	Descrição	Transformada Z	Transformada Gama
Degrau	$U[k]$	$\frac{z}{z-1}$	$\frac{1+\Delta\gamma}{\gamma}$
Rampa	$k.u[k]$	$\frac{z}{(z-1)^2}$	$\frac{1+\Delta\gamma}{\gamma^2}$
Parábola	$k^2.u[k]$	$\frac{z.(z+1)}{(z-1)^3}$	$\frac{(1+\Delta\gamma)(2+\Delta\gamma)}{\gamma^3}$
Exponencial	$k^a.u[k]$	$\frac{z}{(z-a)}$	$\frac{1+\Delta\gamma}{\gamma - \left(\frac{e^{a\Delta} - 1}{\Delta}\right)}$

4.3 – Enhanced and Robust Quadrature PLL (ER-QPLL)

Será apresentada, nesta secção, a descrição da malha robusta de QPLL, doravante denominada ER-QPLL (“*Enhanced and Robust Quadrature-PLL*”). A implementação dessa malha é baseada na malha E-QPLL (“*Enhanced Quadrature-PLL*”) descrita no capítulo anterior, exceto que toda a sua estrutura é implementada utilizando o operador delta (δ), relacionado à transformada gama (γ). Isso fará com que a malha fique otimizada para uma implementação em sistemas de precisão finita, utilizando um número reduzido de bits para representação das variáveis envolvidas. A seguir, será demonstrada a degradação

das malhas QPLL e E-QPLL, implementadas utilizando precisão finita para, em seguida, ser apresentada a demonstração da implementação da malha de ER-QPLL.

4.3.1 Malha QPLL quantizada

Será demonstrada, nesta secção, a degradação do desempenho da malha de QPLL descrita no Capítulo 2, quando implementada utilizando-se precisão finita. Neste caso, todas as variáveis, coeficientes das estruturas de controle e integradores da malha são quantizados em 16 bits, onde o bit mais significativo é representado como sendo o bit de sinal. Assim, tem-se que o intervalo dinâmico de representação das variáveis varia entre +32767 e -32767. Para simular o conversor A/D do microprocessador TMS320F283, que será utilizado para uma implementação prática do algoritmo ER-QPLL mais adiante, o sinal de entrada foi quantizado em apenas 12 bits, o que garante uma variação entre +2048 e -2048.

4.3.1.1 Estratégia de implementação da malha em ponto fixo

Geralmente, ao se representar um número decimal qualquer utilizando-se aritmética de ponto fixo, admite-se que a variável de entrada é um número fracionário variando entre -1 e 1, acarretando que a “vírgula binária” fique localizada à esquerda do bit mais significativo da palavra de dados. Isso exige que o sinal de entrada seja normalizado de modo a permanecer nesse intervalo, entre -1 e 1. Desse modo, considerando-se um tamanho de palavra de 16 bits ($b+1=16$), onde o bit mais significativo representa o bit de sinal, têm-se que o menor número inteiro que pode ser representado nessa situação seria igual a 2^{-15} e o maior $1-2^{-15}$. A Figura 4.3 ilustra essa situação, onde também está indicada a localização da vírgula binária (Δ).

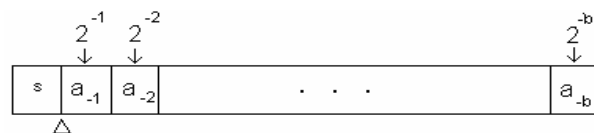


Figura 4.3 – Representação decimal de ponto fixo

Essa representação também é conhecida como Q15, onde o número 15 indica que foram utilizados 15 (quinze) bits para a representação da parte fracionária do número de entrada. Ao se utilizar essa representação, toda a estrutura da malha de QPLL teria que ser adaptada de modo a fazer com que as variáveis das suas equações permanecessem no intervalo entre -1 e 1. Para contornar esse problema, optou-se por fazer com que cada nó ou variável da malha do QPLL tenha uma vírgula binária localizada em um bit específico da palavra de dados, o que faz com que as variáveis decimais a serem quantizadas não estejam obrigatoriamente no intervalo entre 1 e -1.

A vírgula binária é assim otimizada de modo a garantir que cada variável ocupe o maior número de bits possível da palavra de dados. Assim, o maior número decimal positivo passível de representação seria igual a $2^{15}=32767$. Nesse caso, a vírgula binária ficaria à direita do bit menos significativo da palavra de dados, o que faria com que o número ficasse representado através do formato Q0. Do mesmo modo, o menor número inteiro passível de representação seria igual a 2^{-15} , onde a vírgula binária estaria localizada à esquerda do bit mais significativo da palavra de dados, fornecendo uma representação do tipo Q15. Como exemplo, a Figura 4.4 mostra o diagrama de representação de um número através do formato Q14.

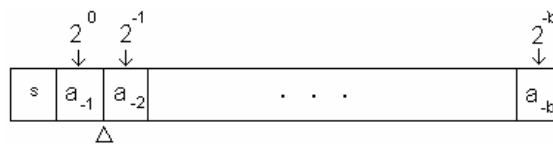


Figura 4.4 – Representação decimal no formato Q14

Deve ser lembrado que os algoritmos de multiplicação/adição da malha devem deslocar a vírgula binária das variáveis de entrada de modo a ficarem na mesma posição, para que essas operações possam ser efetuadas corretamente. Deve-se notar ainda que, nas operações de multiplicação e adição, o tamanho da palavra de dados resultante é igual a $2b$, onde b é o tamanho total da palavra de dados das variáveis de entrada. Assim, deve-se quantizar novamente o resultado da operação, de modo que o seu tamanho seja igual ao tamanho da palavra de dados original. Nesse caso, a quantização pode ser feita através dos métodos truncamento e arredondamento [27]. A implementação de uma quantização por truncamento é mais simples de ser implementada por um algoritmo, pois esse método simplesmente descarta os bits menos significativos da palavra de dados a ser

quantizada, restando apenas os bits de interesse. A quantização por arredondamento é semelhante à quantização por truncamento, exceto que o bit menos significativo da variável quantizada é modificado de modo que o número quantizado permaneça o mais próximo do número original.

Como exemplo, considere um sinal quantizado $Q(x)$ e a variável de entrada igual à x . O erro de quantização é dado por $\epsilon = x - Q(x)$. Pode-se provar que, nesse caso, a quantização por truncamento sempre produz um erro $0 \leq \epsilon \leq v$, onde $v = 2^{-b}$. No caso da quantização por arredondamento, o erro ficaria entre $-v/2 \leq \epsilon \leq v/2$ [27]. Notar que, neste caso, para implementação da malha QPLL, seria preferível utilizar uma quantização por arredondamento, pois o resultado seria um erro médio de quantização nulo. Entretanto, como esse método requer um algoritmo mais complexo, o método de quantização por truncamento foi utilizado no algoritmo da malha. No caso das quantizações dos coeficientes das estruturas de controle da malha e integradores, em que a mesma é feita *offline*, foi utilizada a quantização por arredondamento, o que garante um erro menor ou igual à quantização por truncamento.

A Tabela 4.3 mostra o formato ou a localização das vírgulas binárias das variáveis das equações da malha de QPLL.

Tabela 4.3 – Formato das variáveis quantizadas da malha

Variável	Formato	Variável	Formato
$Kc[n]$	Q14	$e[n]$	Q14
$Ks[n]$	Q14	$y[n]$	Q14
$\Delta\omega[n]$	Q14	$u[n]$	Q12
$\omega[n]$	Q6	T_s	Q14

A Figura 4.5 mostra o fluxograma realizado pelo algoritmo da malha QPLL, quando se deseja obter a soma de dois números representados por lógica binária, no caso em que cada número utiliza um formato do tipo Q_n específico. De acordo com o fluxograma, as variáveis α e β indicam a localização da vírgula binária dos números de entrada, A e B, respectivamente. Assim, para se efetuar corretamente a soma dos dois números, o algoritmo desloca a vírgula binária da variável A, ou B, de um modo que a soma entre as variáveis possa ser direta. Logo após, o algoritmo desloca novamente a

vírgula binária do resultado, de modo a normalizar a saída do algoritmo para um valor Q_n específico. Finalmente, o algoritmo checa se a operação de soma não resultou em um *overflow*. Caso tenha ocorrido, a variável de saída é limitada para o valor +32767 ou -32767.

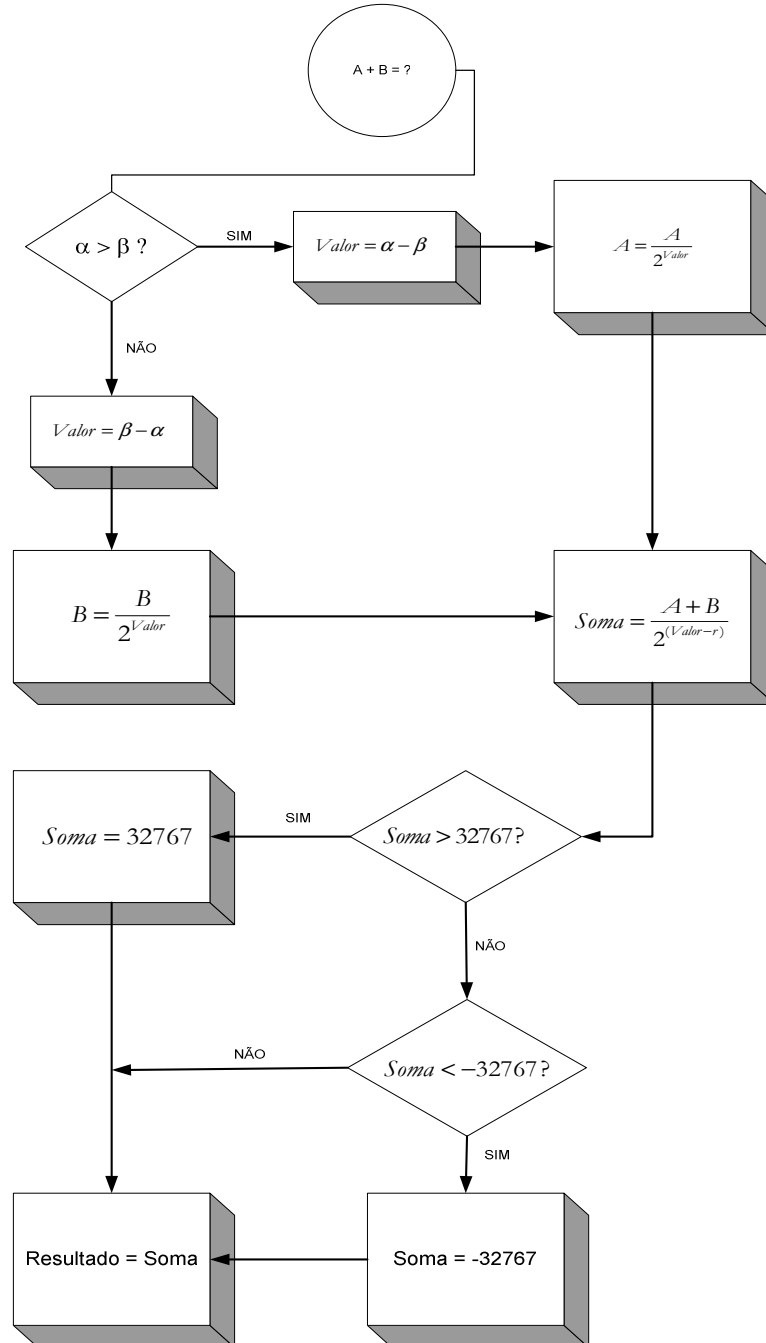


Figura 4.5 – Fluxograma para soma de dois números binários com formatos Q_n específicos

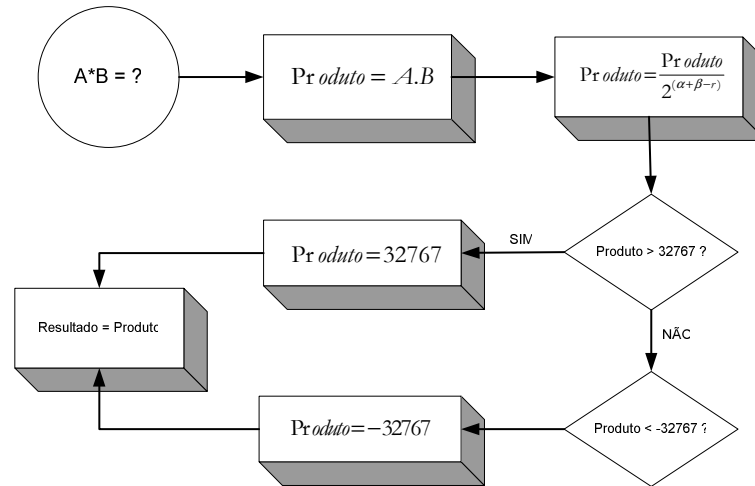


Figura 4.6 – Fluxograma para multiplicação de dois números binários com formatos Qn específicos

A Figura 4.6 mostra o fluxograma utilizado pelo algoritmo da malha QPLL quando se deseja obter a multiplicação de dois números representados por lógica binária, no caso em que cada número utiliza um formato do tipo Qn específico. Nesse caso, a operação é bem mais simples. Inicialmente, o algoritmo simplesmente multiplica as variáveis A e B. Logo em seguida, o resultado é dividido por $2^{\alpha+\beta-r}$, onde α é a localização da vírgula binária da variável A, β é a localização da vírgula binária da variável B e r é a localização desejada da vírgula binária do resultado da multiplicação. Finalmente, o algoritmo checa se a operação de soma não resultou em um *overflow*. Caso tenha ocorrido, a variável de saída é limitada para o valor +32767 ou -32767.

4.3.1.2 Resultados computacionais

Primeiramente, a implementação da malha foi realizada utilizando-se uma baixa taxa de amostragem, igual a 1920 Hz, o que gerou um comportamento satisfatório, de acordo com os resultados que se seguem.

Considere o sinal descrito pela equação:

$$u[n] = A \cdot \cos(\omega[n] + \phi) + b(n) \quad (4.32)$$

a ser injetado na malha QPLL quantizado, onde $A=0,75$ p.u., $\omega[n] = 2 \cdot \pi \cdot n \cdot (60,6)/1920$ rad/amostra, $\phi = -1,45$ rad e $b(n)$ é um ruído aditivo com variância $\sigma^2 = (0,025)^2$. Os

coeficientes k_p e k_i são inicializados com os valores $k_p = 100$ e $k_i = 150$. A frequência central da malha de QPLL é inicializada em 60 Hz. A Figura 4.7 mostra as respostas de amplitude, frequência e fase da malha.

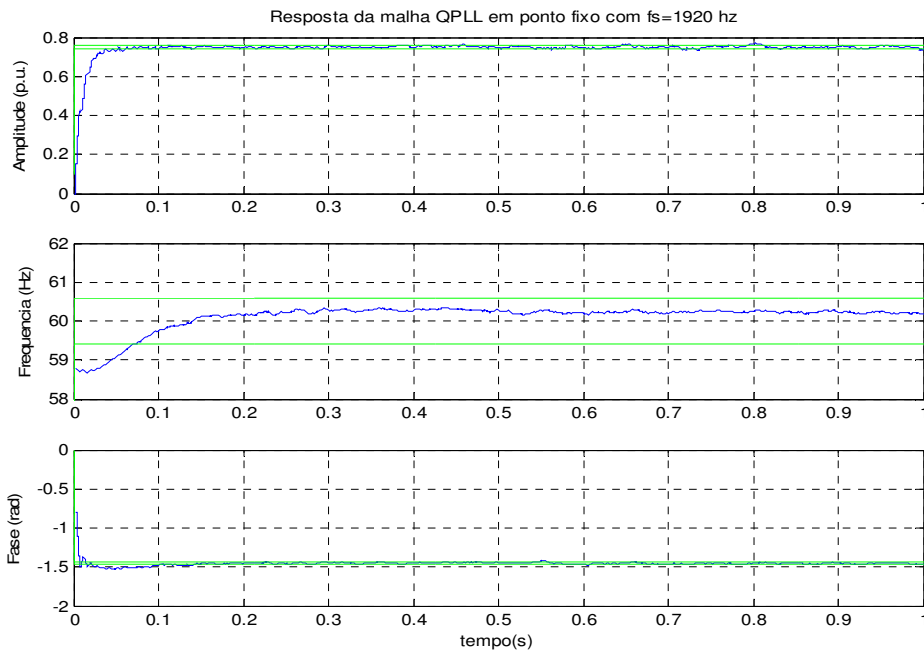


Figura 4.7 – Resposta da malha quantizada em 16 bits, utilizando uma frequência de amostragem igual a 1920 Hz.

De acordo com os resultados obtidos, nota-se que a malha apresentou comportamento satisfatório, apresentando um erro menor que 1% em todas as suas respostas. Será verificada, a seguir, a implementação da malha utilizando uma taxa de amostragem maior, igual a 7680 Hz. Os gráficos da Figura 4.8 mostram os resultados obtidos quando o mesmo sinal $u[n]$ descrito anteriormente é introduzido na malha. Pode-se notar que ocorreu um significativo desvio de frequência na resposta da malha, reportando-se um erro maior que 8%. No caso da resposta em fase da malha, a resposta da estrutura QPLL se deslocou π radianos da fase total do sinal de entrada. Isso indica que a frequência de amostragem da malha interfere diretamente no seu comportamento, quando implementada em aritmética de ponto fixo. A resposta em amplitude obteve um erro praticamente nulo: isto se deve ao fato de que as equações da malha buscam minimizar o erro entre o sinal de entrada e o sinal de saída sintetizado

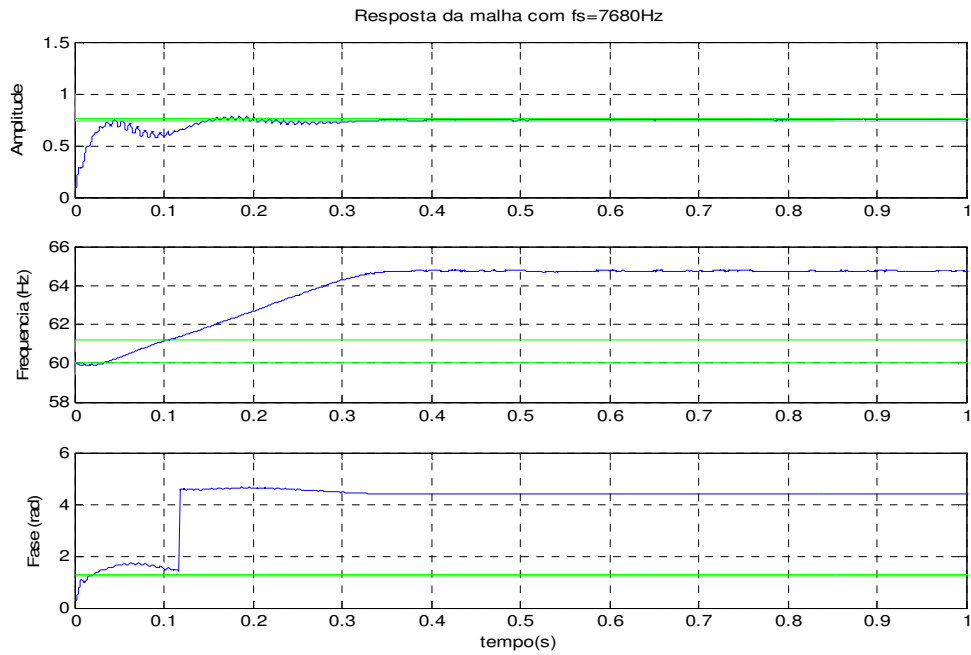


Figura 4.8 – Resposta da malha quantizada em 16 bits, utilizando uma frequência de amostragem igual a 7680 Hz.

Será considerada agora a implementação da malha QPLL com taxa de amostragem ainda maior, igual a 15.000 Hz. A Figura 4.9 mostra os resultados obtidos.

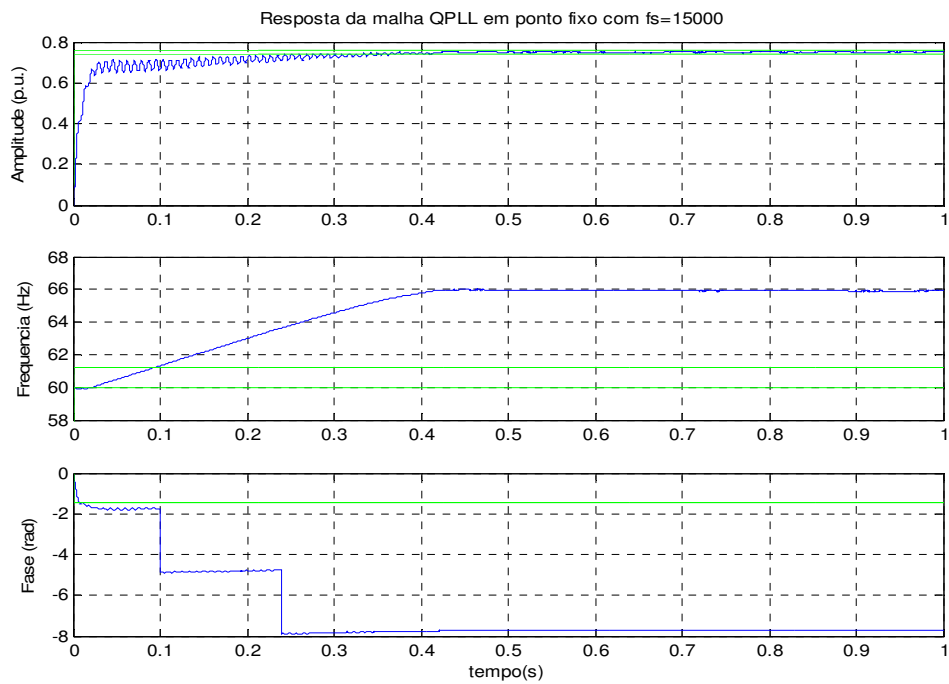


Figura 4.9 – Resposta da malha quantizada em 16 bits, utilizando uma frequência de amostragem igual a 15000 Hz.

Nota-se que, neste caso, o erro na detecção de frequência ficou abaixo dos 8%, enquanto o erro na detecção da fase ficou igual a 2π , o que acarreta um erro nulo total na detecção de fase da malha. A justificativa para um erro menor na resposta da malha, quando foi utilizada uma taxa de amostragem maior que 7680 Hz, pode ser obtida quando se plota o erro de quantização do coeficiente T_s das estruturas integradoras da malha de QPLL. A Figura 4.10 mostra o erro de quantização do coeficiente T_s quando o mesmo é quantizado por truncamento, bem como o erro final quando é utilizada uma quantização por arredondamento.

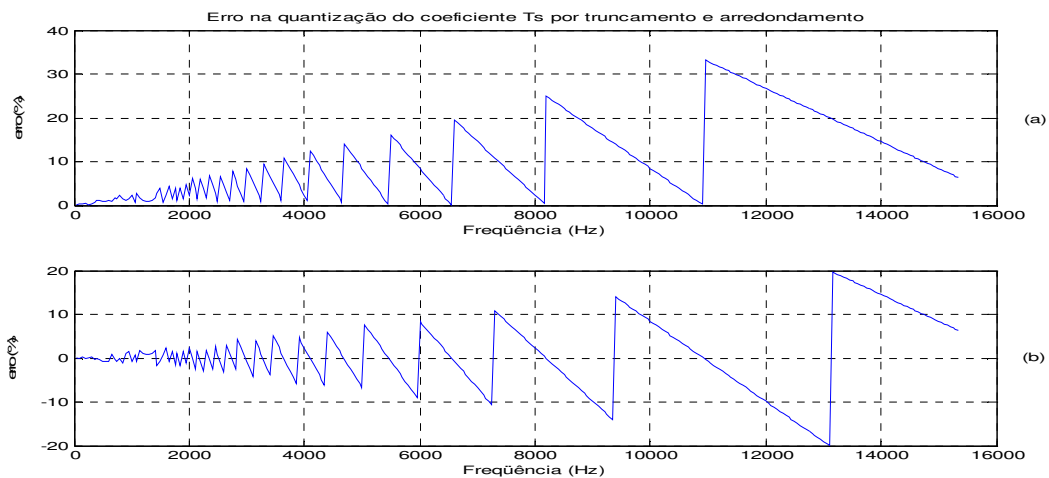


Figura 4.10 – Erro na quantização do coeficiente T_s

De acordo com os resultados exibidos, nota-se que o erro percentual na quantização do coeficiente T_s apresenta valores elevados, cujo valor aumenta conforme se aumenta a taxa de amostragem, podendo chegar até 30%. Deve-se lembrar que as equações dos integradores da malha QPLL quantizada foram implementadas através do método *Euler em atraso*, que forneceu um erro de quantização do coeficiente T_s indicado nas figuras anteriores. Caso o método de implementação fosse o método *Euler em avanço*, pode-se provar que o erro de quantização seria idêntico ao descrito anteriormente. A implementação das estruturas integradoras através do método de integração *Bilinear* forneceria um erro duas vezes menor que as outras estruturas [30]. Entretanto, mesmo através da implementação da malha utilizando esse método de integração, pode-se verificar a ocorrência de erros superiores a 10%, de acordo com a Figura 4.11.

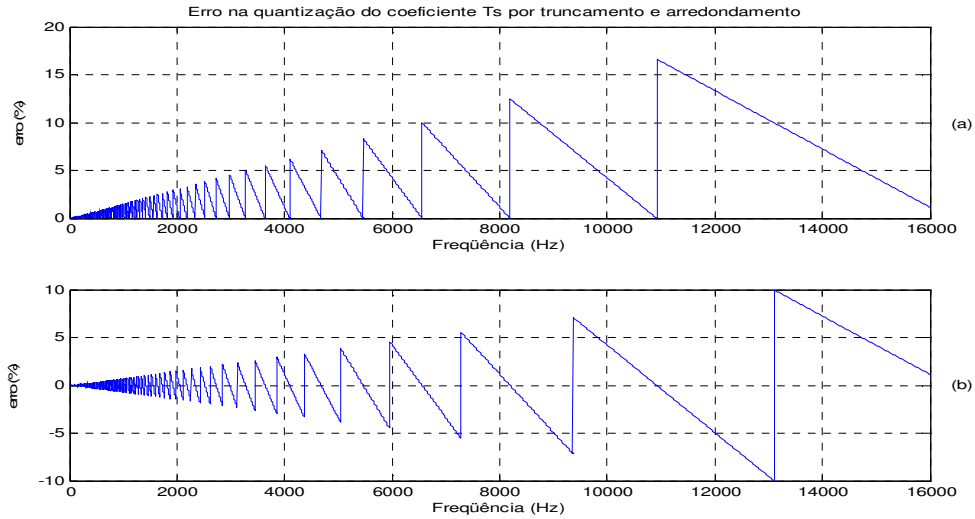


Figura 4.11 – Erro na quantização do coeficiente T_s

É interessante notar que, em determinadas frequências, o coeficiente quantizado T_s apresenta erro nulo. Esses pontos são frequências dadas pela fórmula $f_s = 2^n$, onde n é um número natural inteiro. Nesse caso, o erro é nulo, pois a representação dos números é efetuada através de lógica binária. Tem-se ainda que, utilizando um tamanho de palavra de dados igual a 16 bits, a maior frequência de amostragem que poderia ser representada, com um erro nulo, seria igual a $1/2^{-15} = 32767$ Hz. A partir daí, seria necessário um tamanho de palavra de dados maior para a representação do coeficiente T_s .

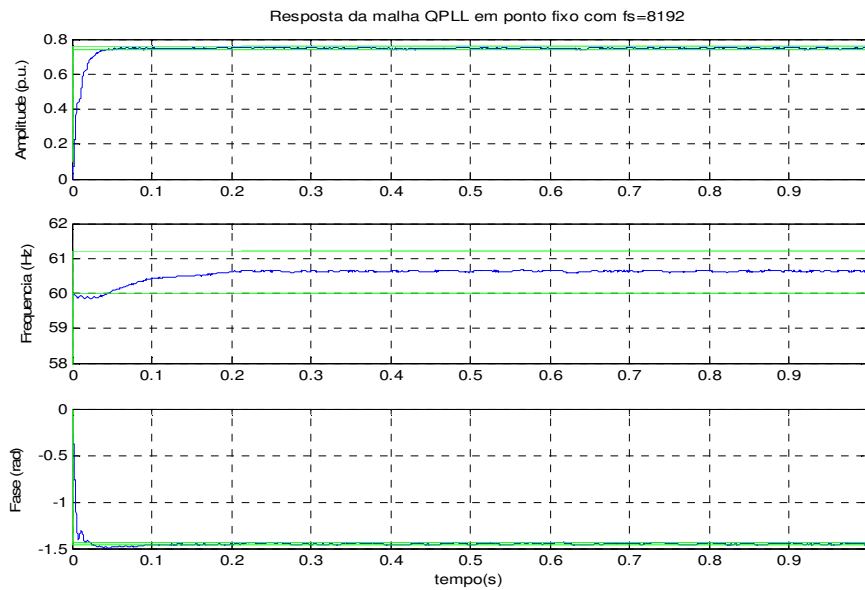


Figura 4.12 – Resposta da malha quantizada em 16 bits, utilizando uma frequência de amostragem igual a 8192 Hz.

A Figura 4.12 mostra a resposta da malha quando é utilizada uma frequência de amostragem igual a 8192 Hz. Note-se que essa frequência é da forma $f_s = 2^n$, sendo n um número inteiro positivo. O sinal utilizado na entrada da malha é o expresso pela equação (4.32). De acordo com os resultados obtidos, verifica-se que a malha apresentou erro inferior a 1% em todas as suas respostas. Isso garante que é possível uma implementação satisfatória da malha QPLL utilizando precisão finita, desde que a taxa de amostragem a ser utilizada pelo sistema seja adequadamente selecionada.

4.3.2 Malha E-QPLL quantizada

Considerar, agora, a implementação da malha E-QPLL descrita no Capítulo 3 utilizando aritmética de ponto fixo. Convém lembrar que a diferença da malha E-QPLL descrita no Capítulo 3, em relação à malha QPLL descrita no Capítulo 2, é a presença de um filtro notch adaptativo em sua entrada. Nas simulações do Capítulo 3, a implementação da malha foi efetuada através da utilização do ambiente MATLAB®, que utiliza em sua estrutura interna uma aritmética de ponto flutuante equivalente a um sistema de aritmética de ponto fixo de, no mínimo, 64 bits. Nesse caso, todas as simulações da malha apresentaram resultados satisfatórios, para diversos tipos de sinais de entrada. Entretanto, essa seção irá analisar o comportamento da malha quando a mesma é implementada utilizando precisão finita, onde o tamanho da palavra de dados será igual a 16 bits.

4.3.2.1 Filtro Notch Quantizado

Será analisado, nesta seção, o comportamento do filtro notch acoplado à entrada da malha E-QPLL, quantizado em 16 bits. Para esta análise, serão utilizadas as suas respostas em frequência e em fase, juntamente com a resposta em frequência e fase do filtro notch implementado em ponto flutuante.

Primeiramente, é utilizada uma baixa taxa de amostragem, igual a 1920 Hz e um fator notch $\rho = 0,991$, o que garante uma resposta do filtro quantizado bem próxima à resposta ideal, conforme mostra a Figura 4.13 a seguir.

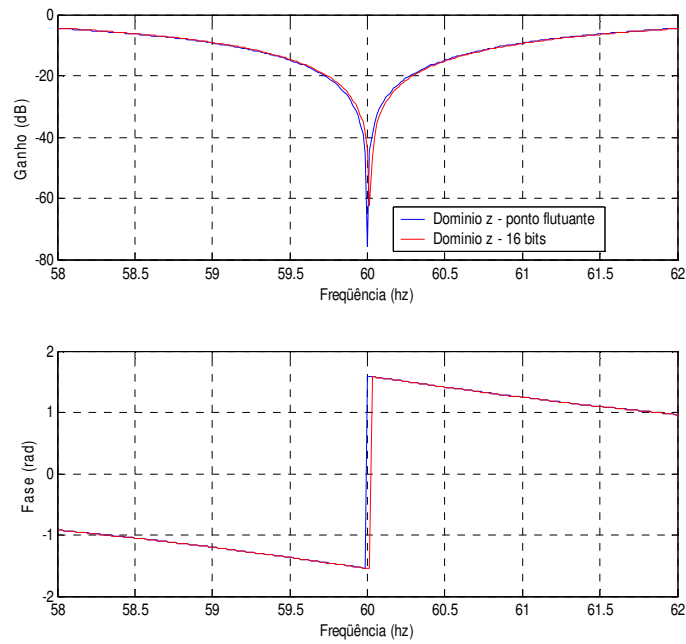


Figura 4.13 – Resposta em frequência e fase do filtro notch implementado com uma frequência de amostragem igual a 1920 Hz.

De acordo com os resultados obtidos, a rejeição do filtro notch na frequência de interesse (60 Hz), quando implementado utilizando aritmética de ponto flutuante, ficou igual a 75,67dB. No caso da implementação em ponto fixo, a rejeição alcançada nesse caso foi igual a 44 dB, o que implica em um ganho linear da ordem de 0,0059, condição suficiente para um funcionamento satisfatório do filtro. No caso da resposta em fase (rad), nota-se que há um desvio na resposta do filtro notch quantizado, o que seguramente pode provocar distorções na aquisição da fase da componente fundamental de um sinal de entrada qualquer. Caso o sinal de entrada esteja contaminado com componentes harmônicas de baixa frequência, provavelmente o desempenho do filtro notch ficaria comprometido.

A Figura 4.14 mostra a resposta da malha E-QPLL quantizada em 16 bits, quando um sinal de entrada descrito pela equação (4.31) é utilizado como sua entrada. O filtro notch é implementado através da forma transposta II [12].

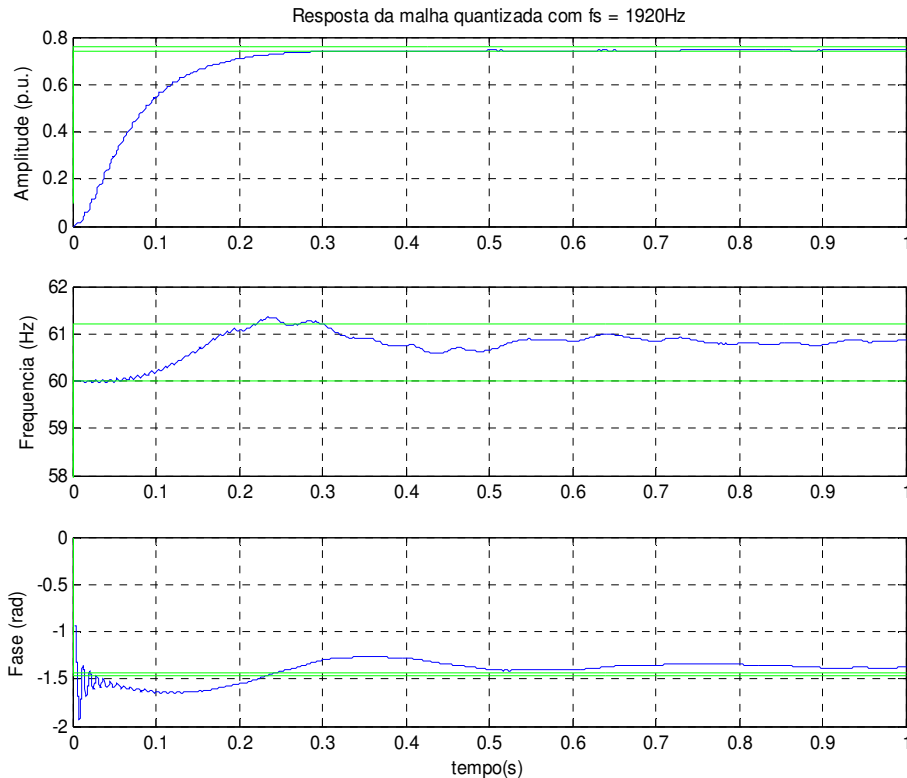


Figura 4.14 – Resposta da malha E-QPLL quantizada em 16 bits, utilizando uma frequência de amostragem igual a 1920 Hz.

Notar que houve um desvio na resposta em fase da malha QPLL quantizada, indicando que a quantização no filtro notch não foi eficiente, para a taxa de amostragem utilizada (1920 Hz). Entretanto, pode-se notar que a resposta em amplitude da malha foi satisfatória, resultado que provavelmente não teria ocorrido caso o sinal de entrada estivesse contaminado com alguma componente harmônica de baixa frequência. Nesse caso, se a aplicação envolver apenas a aquisição da amplitude e frequência do sinal de entrada, a implementação da malha através da transformada z , utilizando uma frequência de amostragem igual a 1920 Hz, poderá produzir resultados satisfatórios.

Será considerada agora a implementação do filtro notch com uma taxa de amostragem ainda maior, igual a 15360 Hz. Nesse caso, a rejeição do filtro quantizado na frequência de interesse (60 Hz) ficou igual a 24 dB, o que resulta em um elevado ganho linear nessa frequência, igual a 0,063, o que acaba inutilizando o filtro notch.

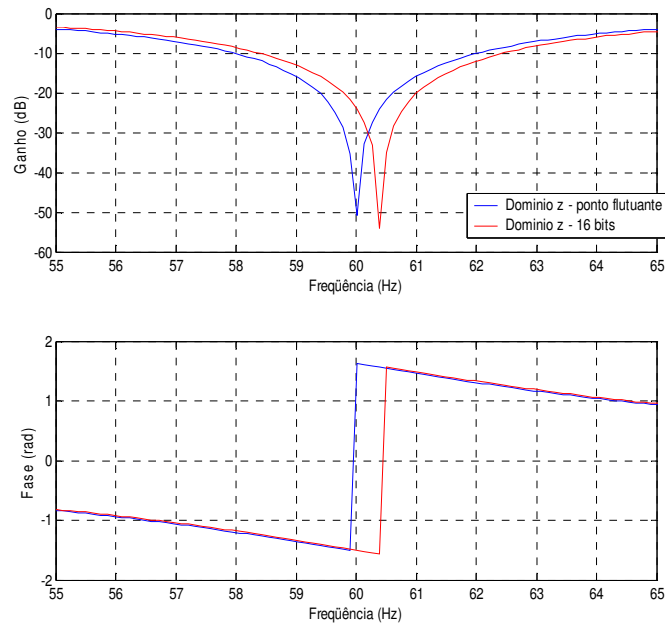


Figura 4.15 – Resposta em frequência e fase do filtro notch implementado com uma frequência de amostragem igual a 15360 Hz.

A Figura 4.16 mostra o resultado da implementação efetuada para a malha E-QPLL, onde se pode assinalar o aumento significativo no desvio da resposta do filtro quantizado em relação à resposta considerada ideal.

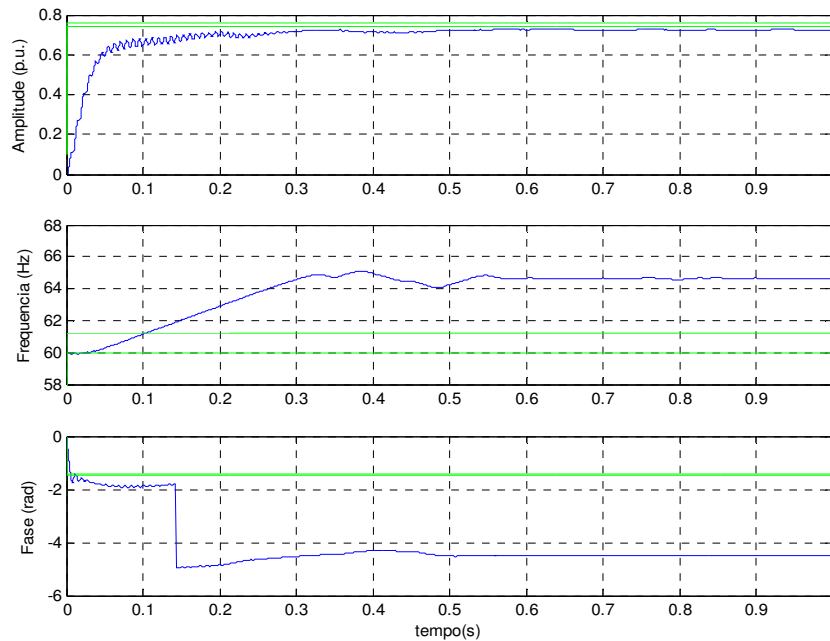


Figura 4.16 – Resposta da malha E-QPLL quantizada em 16 bits, utilizando uma frequência de amostragem igual a 15.360 Hz

Há de se observar, entretanto, que a resposta em amplitude da malha sofreu essa distorção por dois motivos: o erro de quantização do filtro notch e o erro na detecção da frequência do sinal de entrada da malha. Como a atualização do filtro notch é feita a partir da frequência adquirida pela malha, tem-se que, quando esta indica uma frequência errada de atualização para o filtro, a sua frequência de corte irá se afastar ainda mais da frequência de corte desejada.

Pode-se provar que quanto maior a frequência de implementação do filtro notch, mais a sua resposta em frequência se afastará da ideal. Isso ocorre principalmente devido à proximidade dos pólos do filtro discretizado da região $z = 1$, o que faz com que a sua resposta em frequência se torne mais sensível em relação à quantização de seus coeficientes [12,14]. Esse efeito é mais severo caso a frequência de sintonia do filtro notch seja bem menor que a frequência de amostragem do sistema ou se a largura de banda do filtro for muito reduzida. Estas situações levam os pólos do filtro notch a se agruparem ainda mais em torno da região $z = 1$. Além desse problema, pode ocorrer que um pólo do filtro se desloque para fora do círculo unitário, tornando o mesmo instável.

4.4 - Enhanced and Robust QPLL (ER-QPLL)

No intuito de superar os erros apresentados na implementação da malha E-QPLL quantizada em 16 bits, é apresentada nessa secção a descrição da malha ER-QPLL (*“Enhanced and Robust QPLL”*). A implementação de toda a malha, bem como do filtro notch adaptativo presente em sua entrada, será feita através da utilização do operador delta (δ), relacionado à transformada Gama (γ). Espera-se que isto garanta um funcionamento satisfatório da malha, utilizando uma implementação com precisão finita de, no mínimo, 16 bits. Conforme será visto a seguir, mesmo utilizando altas taxas de amostragem, a malha apresentará resultados satisfatórios.

A implementação da malha QPLL quantizada, no domínio do operador delta (δ), ocorre através da substituição de suas estruturas integradoras representadas no domínio do operador z por suas estruturas equivalentes implementadas no domínio do operador delta (δ).

Considerar a função de transferência de um integrador do tipo *Euler em atraso*, utilizada nas equações da malha, que é dada por:

$$\sum_z = \frac{T_s}{z-1}, \quad (4.33)$$

onde T_s é a taxa de amostragem da estrutura integradora.

A função de transferência equivalente implementada no domínio do operador delta (δ) pode ser obtida através da seguinte relação:

$$\delta = \frac{z-1}{\Delta}, \quad (4.34)$$

Substituindo-se a última equação na equação do integrador da malha, obtém-se a estrutura integradora representada no domínio do operador delta (δ), que será utilizada na malha ER-QPLL:

$$\sum_{\delta} = \frac{T_s}{\Delta\delta} = \frac{\beta \cdot \delta^{-1}}{1}, \quad (4.35)$$

onde $\beta = T_s / (1/\Delta)$.

A implementação da estrutura integradora no domínio do operador delta (δ) pode ser efetuada através do diagrama da Figura 4.17a. Na Figura 4.17b está representada a operação δ^{-1} .

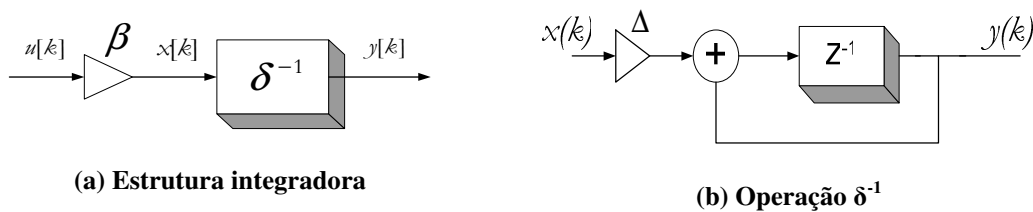


Figura 4.17 – Integrador no domínio δ (a) e Operação δ^{-1} (b)

Desse modo, o diagrama da malha ER-QPLL pode ser visto na Figura 4.18, onde se nota que a principal modificação da estrutura é a substituição das estruturas integradoras implementadas através do operador z por suas equivalente no domínio do operador delta (δ).

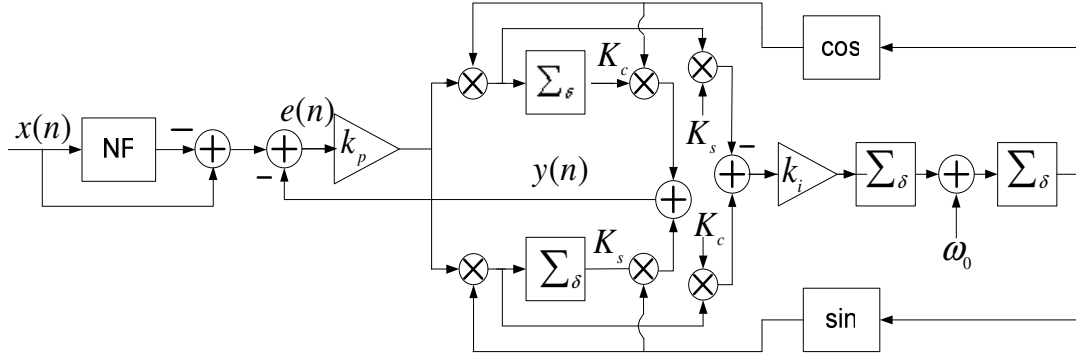


Figura 4.18 – Malha E-QPLL implementada no domínio do operador delta (δ)

Desse modo, as equações discretizadas da malha QPLL, implementadas no domínio do operador delta (δ), podem ser representadas através das seguintes equações:

$$\begin{aligned}
 K_s[n+1] &= K_s[n] + 2 \cdot \beta \cdot \Delta \cdot \mu_s \cdot e(t) \cdot \text{sen}(\phi(t)) \\
 K_c[n+1] &= K_c[n] + 2 \cdot \beta \cdot \Delta \cdot \mu_c \cdot e(t) \cdot \text{cos}(\phi(t)) \\
 \Delta \omega[n+1] &= \Delta \omega[n] + 2 \cdot \beta \cdot \Delta \cdot \mu_f \cdot e(t) [K_s \cdot \text{cos}(\phi(t)) - K_c \cdot \text{sen}(\phi(t))] \\
 \phi[n+1] &= \phi[n] + \beta \cdot \Delta \cdot (\omega_0 + \Delta \omega(t)) \\
 y[n] &= K_s[n] \cdot \text{sen}(\phi[n]) + K_c[n] \cdot \text{cos}(\phi[n]) \\
 e[n] &= u[n] - y[n],
 \end{aligned} \tag{4.36}$$

Nota-se que a principal diferença entre as equações da malha QPLL implementada através da Transformada γ e as implementadas através da Transformada Z é a substituição do coeficiente de integração T_s pelo coeficiente $\beta \cdot \Delta$, nas estruturas integradoras da malha. Entretanto, ao se realizar a quantização desses coeficientes nota-se, de acordo com o diagrama de um integrador representado na Figura 4.17a, que a quantização dos coeficientes β e Δ são efetuadas separadamente. Desse modo, sabendo que $\beta = T_s/\Delta$, esse parâmetro pode ser ajustado de acordo com o valor da variável Δ , sendo que esta não está necessariamente relacionada à taxa de amostragem do sistema, podendo ser escolhida livremente, tal que $0 < \Delta < 1$. Por exemplo, escolhendo-se o parâmetro $\Delta=1/2048$, a Figura 4.7 mostra o erro de quantização do parâmetro β ($\beta = C_1$), para diversas taxas de amostragem. Nota-se que, para taxas de amostragem de valor acima de 16.000 Hz, o erro na quantização do parâmetro apresenta valores menores do que 0,01%. Comparando a Figura 4.19 com a Figura 4.10 e 4.11, nota-se claramente a grande diminuição do erro de quantização obtido na implementação da estrutura, utilizando a Transformada Gama(γ).

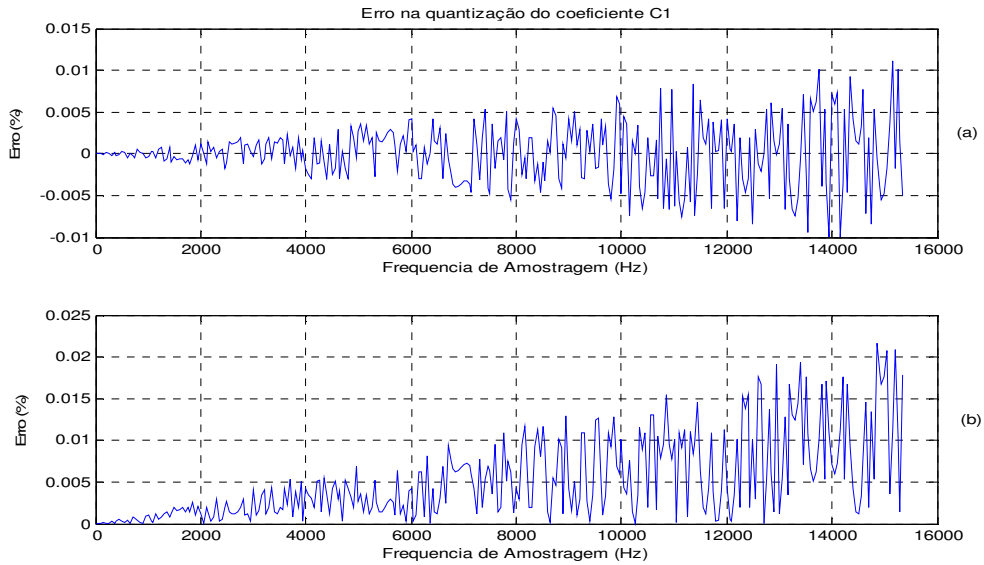


Figura 4.19 – Erro de quantização do coeficiente $\beta=C_1$

Nota-se que é possível, para cada escolha do parâmetro Δ , realizar uma nova plotagem, como a da Figura 4.19, o que indica um possível valor ótimo da variável Δ , que minimiza o erro de quantização da variável β , para uma dada frequência de amostragem. Desse modo, utilizando-se uma frequência de amostragem igual a 7680 Hz, a Figura 4.20 mostra o erro de quantização do coeficiente β , para o valor $1/\Delta$ variando desde a unidade até $1/\Delta = 32767$. De acordo com a Figura 4.20, nota-se que o erro de quantização do coeficiente β apresenta valores decrescentes, de acordo com o aumento da variável $1/\Delta$.

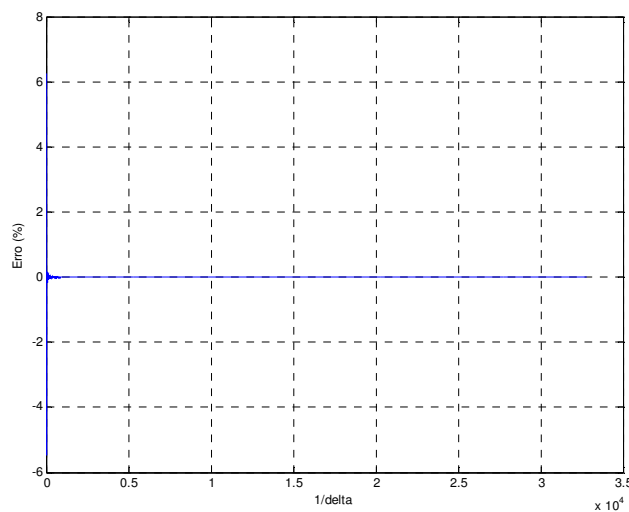


Figura 4.20 – Erro de quantização do coeficiente β , em função do parâmetro $1/\Delta$, para uma taxa de amostragem igual a 7680 Hz.

4.4.1 Filtro Notch implementado através da Transformada Gama (γ)

Conforme descrito anteriormente, a implementação do filtro notch quantizado com 16 bits, através da transformada Z, utilizando altas taxas de amostragens, não produziu resultados satisfatórios. Isso se deve principalmente à proximidade dos pólos da função de transferência do filtro à região do plano complexo $z = 1$, o que aumenta a sensibilidade da resposta em frequência do filtro em relação à quantização de seus coeficientes.

As estratégias de resolução desses problemas são, entre outras: utilizar um tamanho de palavra de dados maior para a representação dos coeficientes do filtro, escolher a melhor forma de implementação da estrutura (Forma Direta I, II, Transposta I ou II) ou ainda utilizar técnicas como EES (*Error Spectrum Shaping*), etc. Será proposta, nesta secção, uma estratégia de implementação do filtro notch através da utilização da transformada gama (γ). Essa transformada realiza o mapeamento dos pólos e zeros da função de transferência discreta do filtro para uma nova região, onde os efeitos de suas quantizações serão menos severos, conforme será demonstrado a seguir. O mapeamento dos pólos e zeros do filtro é feito através da escolha apropriada do parâmetro delta (Δ), que deve ser escolhido adequadamente no intuito de diminuir os erros de quantização dos coeficientes do filtro, bem como das operações de multiplicação e soma da estrutura implementada em ponto fixo.

Sabendo que a função de transferência de um filtro notch de segunda ordem no domínio do operador z , é dada pela Equação (3.4), a função de transferência do filtro $H_1(\delta)$, no domínio do operador delta (δ), utilizando a Equação (4.17), será dada por [32]:

$$H_1(\delta) = H_0(z) \Big|_{z=1+\Delta\delta} = \frac{\alpha_0 + \alpha_1 \cdot \delta^{-1} + \alpha_2 \cdot \delta^{-2}}{\beta_0 + \beta_1 \cdot \delta^{-1} + \beta_2 \cdot \delta^{-2}}, \quad (4.37)$$

Os coeficientes a_1 , a_2 , b_1 e b_2 da função de transferência do filtro notch original foram mapeados para o domínio do operador delta (δ), para os coeficientes α_0 , α_1 , α_2 , β_0 , β_1 e β_2 . A relação entre os coeficientes será obtida a partir da Tabela 4.1. Nesse caso, $\alpha_0 = \beta_0 = 1$.

Assim, considere a implementação de um filtro com um fator notch $\rho = 0,9975$, uma frequência de amostragem $f_s = 15360$ Hz e uma frequência de sintonia $f_0 = 60$ Hz. A Figura 4.21 mostra a resposta em frequência do filtro notch quando os coeficientes do mesmo são quantizados em 16 bits, juntamente com a sua resposta ideal. Note-se que foi utilizado um fator notch bem próximo à unidade, o que fará que o filtro fique bastante sensível à quantização de seus coeficientes. É utilizado o fator $\Delta = 1/4096$.

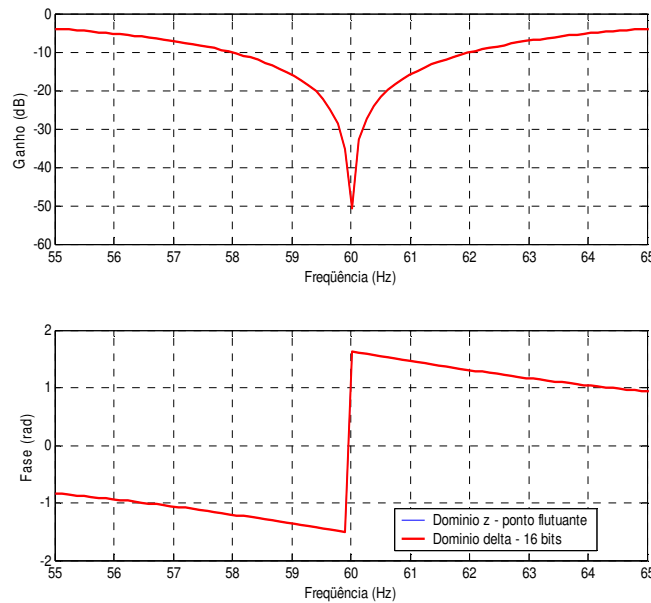


Figura 4.21 – Resposta em frequência e fase do filtro notch implementado através do operador δ , utilizando uma taxa de amostragem igual a 15360 Hz.

De acordo com os resultados obtidos, pode ser verificado que a resposta em frequência do filtro notch quantizado em 16 bits, implementado através da transformada Gama (γ), apresentou um resultado extremamente próximo à resposta em frequência considerada ideal.

Desse modo, a Figura 4.22 mostra o resultado da implementação da malha ER-QPLL, que possui a estrutura QPLL e o filtro notch adaptativo implementados através do operador delta (δ), quando o sinal dado pela Equação 4.31 é introduzido em sua entrada. Na implementação da malha, foi utilizada uma taxa de amostragem de 15.360 Hz, os parâmetros k_p e k_i foram inicializados com $k_p = 100$ e $k_i = 150$ e a frequência central de operação da malha foi inicializada em 58 Hz. Nota-se que a resposta da malha se comporta de um modo ideal, apresentando um erro menor que 1% em todas as suas respostas, comprovando a eficiência da sua implementação através da utilização da Transformada γ .

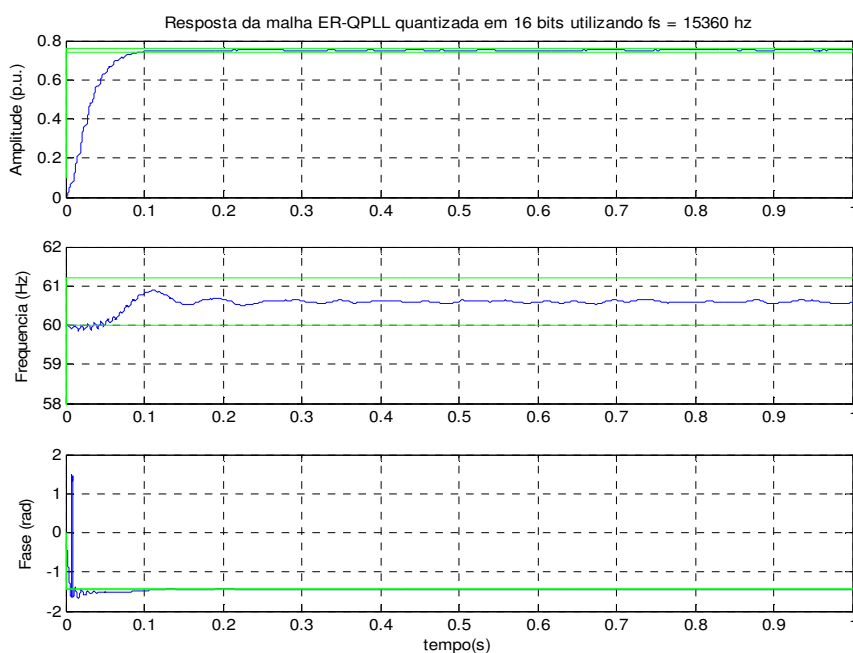


Figura 4.22 –Resposta da malha ER-QPLL implementada através da Transformada γ

4.4.2 Otimização do parâmetro Delta na implementação do filtro notch

O mapeamento dos coeficientes da função de transferência discreta do filtro notch para o domínio do operador delta (δ) deve ser feita através da escolha apropriada do parâmetro Δ , que deve ser escolhido de modo a reduzir os erros de quantização de uma implementação do filtro utilizando precisão finita. Uma estratégia eficiente de otimização do parâmetro Δ foi proposta por [12], onde o mesmo é otimizado de modo a reduzir os erros de quantizações e saturações nas operações de soma e multiplicação da estrutura implementada em ponto fixo. Essa estratégia pressupõe que o parâmetro delta (Δ) possa assumir qualquer valor entre 0 e 1. Entretanto, conforme será demonstrado a seguir, é desejável que a variável $1/\Delta$ assuma um valor do tipo $1/\Delta=2^n$ o que essa estratégia não pode garantir; assim, pode surgir a necessidade de se utilizar um tamanho de palavra de até 32 bits para a correta representação da variável Δ .

Calculada nestas observações, esta seção propõe uma nova estratégia de ajuste do parâmetro Δ . A estratégia proposta analisa os erros de quantização dos coeficientes do filtro notch quantizado, no domínio do operador delta (δ), para diversos valores do parâmetro Δ . A Figura 4.23 mostra o erro de quantização dos coeficientes do filtro notch, inicializado com os parâmetros descritos na Seção 4.4.1, mapeados para o

domínio do operador delta (δ), onde o eixo das abscissas mostra a variação do parâmetro $1/\Delta$.

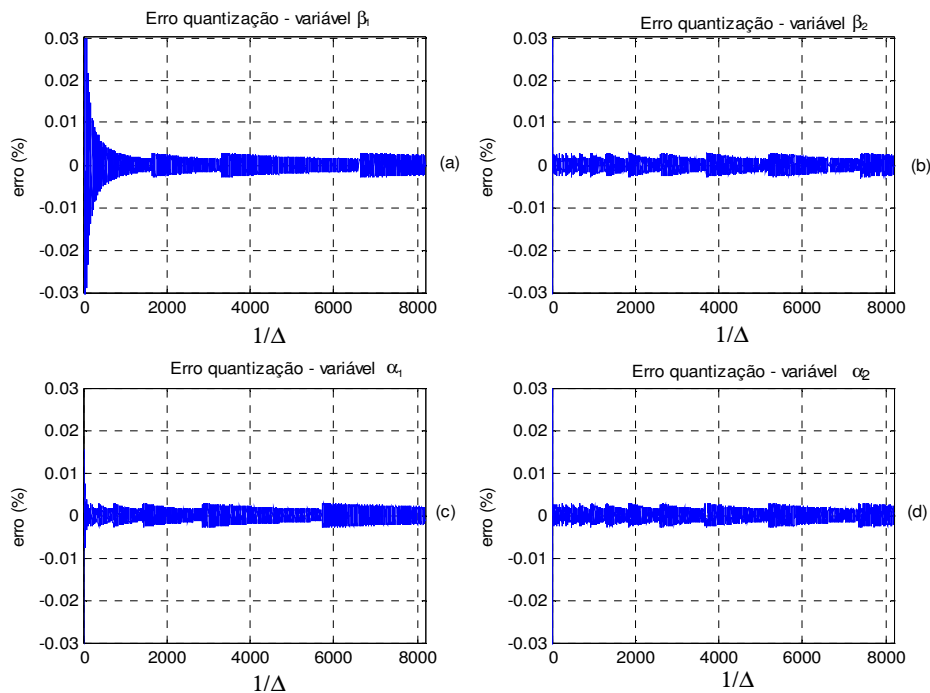


Figura 4.23 – Erro na quantização dos coeficientes do filtro notch em função do parâmetro $1/\Delta$

De acordo com os resultados obtidos nota-se que, para uma ampla variedade de valores de $1/\Delta$, os erros de quantização dos coeficientes do filtro notch apresentam valores menores que 0,1%, comprovando a eficiência de implementação deste filtro através da Transformada Gama(γ). É claro que, para diferentes valores de inicialização dos parâmetros do filtro notch, os gráficos exibidos anteriormente podem apresentar valores diferentes, mas esse exemplo permite uma boa generalização.

4.4.3 Escolha do parâmetro Δ

É importante notar que a escolha do parâmetro Δ deve ser feita cuidadosamente, de modo que o mesmo apresente um erro nulo quando quantizado utilizando precisão finita. Isso pode ser feito escolhendo esse parâmetro com um valor tal que $1/\Delta = 2^n$, sendo n um natural inteiro. A Figura 4.24 mostra o erro de quantização do parâmetro $1/\Delta$, onde se comprova que, em valores do tipo $1/\Delta = 2^n$, a quantização deste parâmetro apresenta um erro nulo.

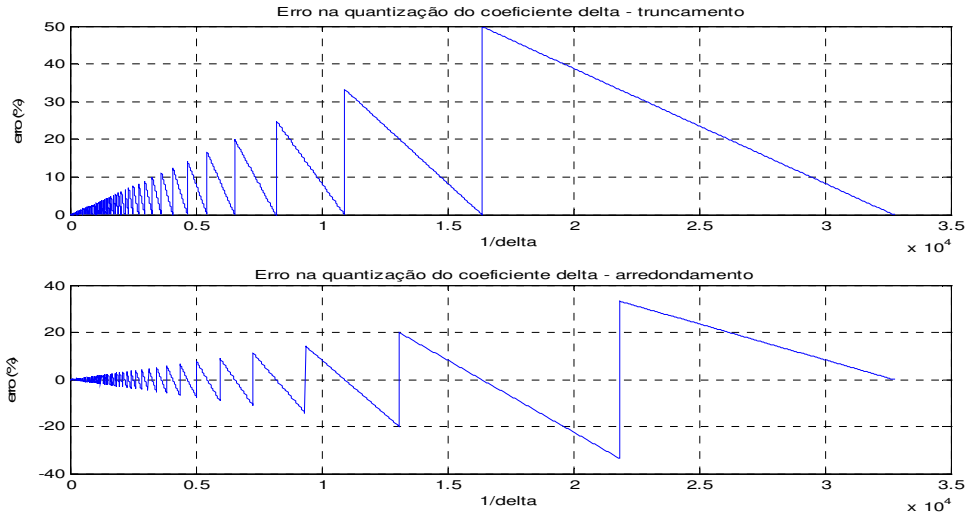


Figura 4.24 – Erro na quantização do coeficiente Δ por truncamento e arredondamento

Desse modo, ao se escolher o valor do parâmetro Δ a ser utilizado no mapeamento dos coeficientes do filtros notch e das estruturas integradoras para o domínio do operador delta (δ), recomenda-se sempre escolher valores de Δ do tipo $1/\Delta=2^n$, ou próximo à isso.

4.5 Resultados comparativos entre a malha ER-QPLL e QPLL

No intuito de comprovar a robustez da malha ER-QPLL proposta nesse capítulo, foram efetuadas diversas simulações comparando o desempenho da malha ER-QPLL quantizada em 16 bits com a malha QPLL implementada em ponto flutuante, descrita inicialmente no Capítulo 2. De acordo com a estratégia adotada no capítulo anterior, os parâmetros das malhas QPLL e ER-QPLL foram ajustados de um modo que ambas as malhas obtenham uma mesma rejeição a ruídos em regime permanente. Assim, serão utilizados os valores $k_p = k_i = 75$ para a malha ER-QPLL e $k_p = k_i = 30$, para a malha QPLL. A taxa de amostragem nas implementações a seguir será igual a 15.360 Hz. A frequência de operação será igual a 58 Hz.

1) Harmônicos

A Figura 4.25 mostra a resposta da malha ER-QPLL quantizada em 16 bits, juntamente com a resposta da malha QPLL implementada em ponto flutuante, quando o sinal de entrada descrito pela equação:

$$\begin{aligned}
 x(n) = & 1 \cdot \cos\left(\frac{2 \cdot \pi \cdot 60 \cdot n}{15360} + 1,45\right) + 0,2 \cdot \text{sen}\left(\frac{2 \cdot 2 \cdot \pi \cdot 60 \cdot n}{15360}\right) + 0,1 \cdot \text{sen}\left(\frac{5 \cdot 2 \cdot \pi \cdot 60 \cdot n}{15360}\right) \\
 & + 0,08 \cdot \text{sen}\left(\frac{9 \cdot 2 \cdot \pi \cdot 60 \cdot n}{15360}\right) + 0,06 \cdot \text{sin}\left(\frac{11 \cdot 2 \cdot \pi \cdot 60 \cdot n}{15360}\right) + 0,04 \cdot \text{sen}\left(\frac{13 \cdot 2 \cdot \pi \cdot 60 \cdot n}{15360}\right)
 \end{aligned}
 \tag{4.38}$$

é utilizado como sinal de entradas para as malhas.

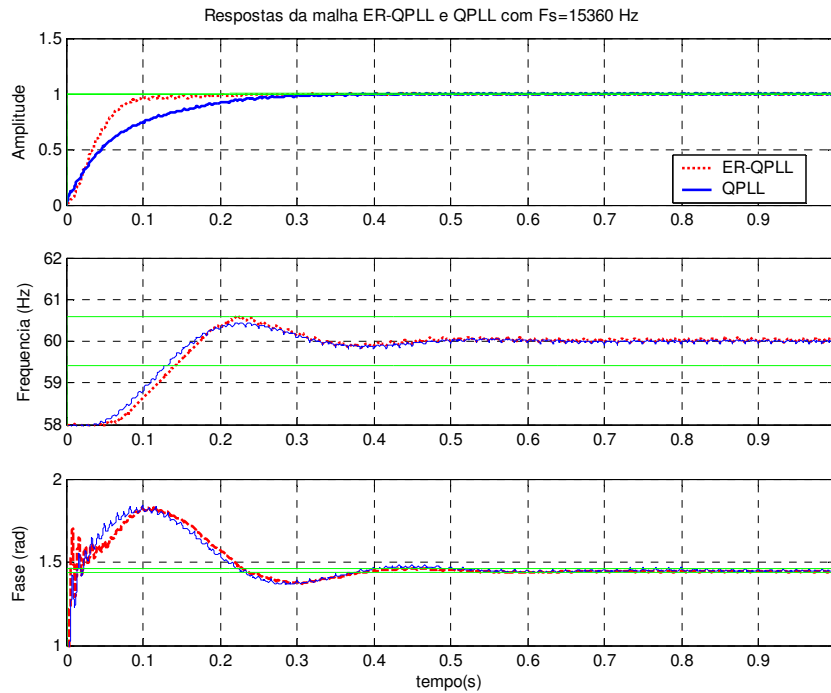


Figura 4.25 – Comparação das malhas com um sinal de entrada com um sinal com harmônicos

Nota-se que ambas as malhas apresentaram comportamentos semelhantes, exceto no caso da aquisição de amplitude, em que a malha ER-QPLL apresentou um transiente mais rápido. No tocante ao erro de regime permanente, pode-se verificar que todas as malhas apresentaram erros inferiores a 1%.

2) Ruído Gaussiano

Nesse caso, é introduzido um ruído gaussiano aditivo com variância $\sigma^2 = (0,1)^2$ no sinal de entrada descrito pela equação 4.38 e são removidas as componentes harmônicas no mesmo. É introduzido um degrau negativo na amplitude de entrada, no instante $t = 0,5$ segundos, de magnitude 20%, ou seja, a amplitude adquire um valor igual a 0,8 em $t=0,5$ segundos. Os resultados são mostrados na Figura 4.26.

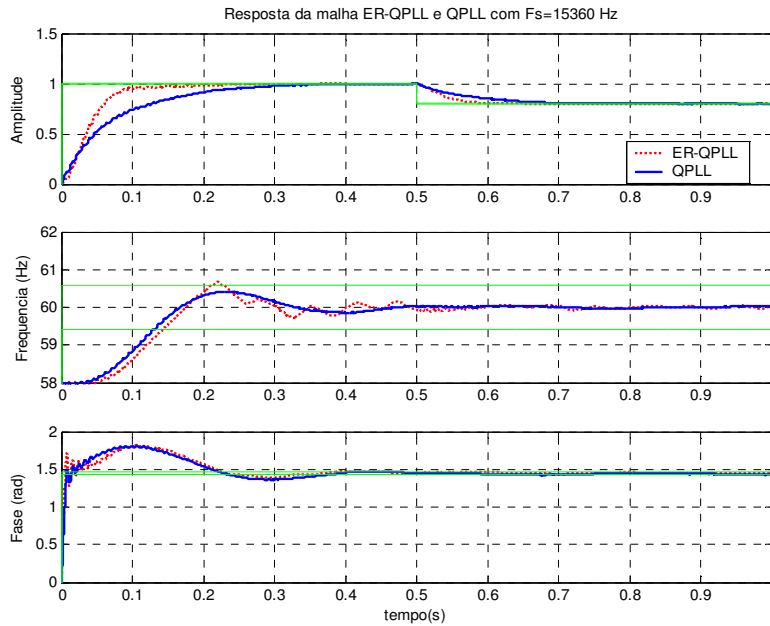


Figura 4.26 - Respostas das malhas ER-QPLL e QPLL

Notar que o desempenho das malhas foi semelhante ao caso anterior, com a malha ER-QPLL apresentando um transiente mais rápido que a malha QPLL.

3) Mudança de freqüência

Nesse caso, foi introduzido nas duas malhas, QPLL e ER-QPLL, o mesmo sinal do Caso 2 descrito anteriormente, apenas com uma ligeira modificação: nesta nova simulação, foi introduzido uma perturbação em degrau de 3 Hz na freqüência do sinal de entrada, no instante $t = 0,5$ segundos.

Ambas as malhas apresentaram um comportamento satisfatório e, novamente, a malha ER-QPLL apresentou um transitório mais rápido. Nota-se que há uma descontinuidade na resposta da fase do sinal de entrada em $t = 0,5$ segundos. Isso ocorre devido à mudança do “sinal de referência”, descrito no capítulo anterior, que permite estimar o erro da aquisição de fase da malha QPLL. A Figura 4.27 mostra os resultados obtidos para este caso.

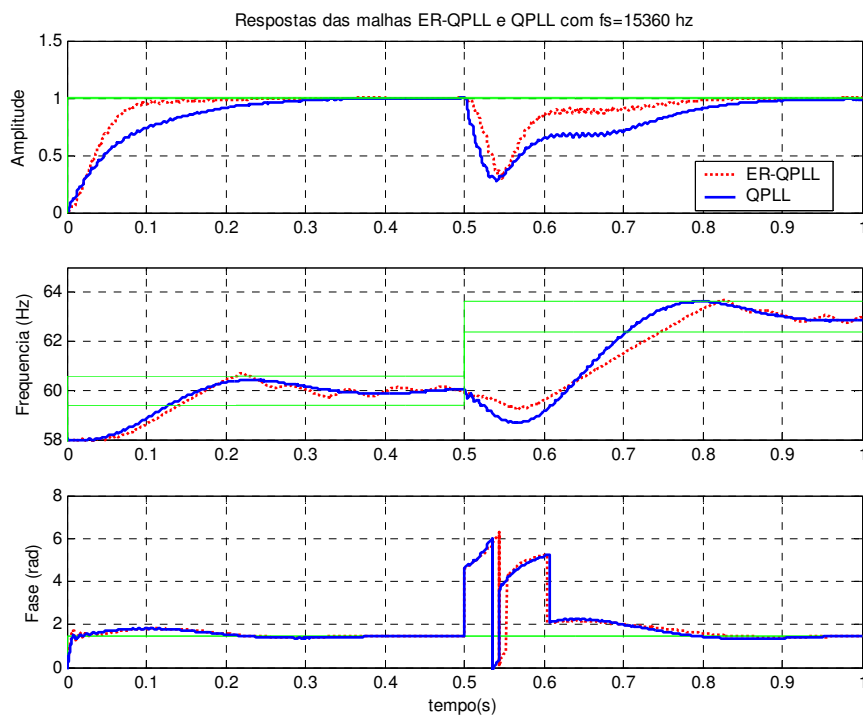


Figura 4.27 – Resposta das malhas QPLL e ER-QPLL

4.6 Implementação da malha ER-QPLL no microprocessador TMS320F2812

Essa secção descreve a implementação do algoritmo ER-QPLL, proposto neste trabalho, no microprocessador TMS320F2812, da *Texas Instruments*®. O microprocessador TMS320F2812 é um DSP (*Digital Signal Processor*) de 32 bits e com uma velocidade de operação de 150 MHz, fabricado pela *Texas Instruments*.

A implementação do algoritmo ER-QPLL foi efetuada diretamente através das equações descritas em (4.38), utilizando a linguagem de programação C++. O filtro notch foi implementado utilizando a estrutura representada através da Figura 4.2 (*DFIII*). A compilação do programa e o envio do código compilado, para a memória RAM do DSP, foram efetuadas através da utilização do software *Code Composer* e do hardware denominado *JTAG*.

Todas as variáveis utilizadas na implementação do algoritmo foram representadas utilizando a variável *integer*, que é uma variável com tamanho total de 16 bits. Nas

operações de soma e multiplicação, utilizou-se uma variável auxiliar do tipo *long int*, de 32 bits, para armazenamento das operações intermediárias. Desse modo, para efetuar a validação do algoritmo implementado no microprocessador, foram gerados diversos sinais através do software MATLAB®. Em seguida, foram quantizados em 12 bits e enviados para a memória RAM do microprocessador.

Primeiramente, foi introduzido na malha ER-QPLL um sinal expresso por:

$$x(n) = \cos(\omega_0 \cdot n) + 0,3 \cdot \cos(3 \cdot \omega_0 \cdot n + \pi / 3) + 0,2 \cdot \cos(5 \cdot \omega_0 \cdot n + 2 \cdot \pi / 7) + 0,1 \cdot \cos(7 \cdot \omega_0 \cdot n - \pi) + b(n), \quad (4.38)$$

onde $\omega_0 = 2 \cdot \pi \cdot f_0 / f_s$, $f_0 = 60$ Hz, $f_s = 7680$ Hz e $b(n)$ é um ruído aditivo com uma variância $\sigma^2 = (0,1)^2$.

A Figura 4.28 mostra a amplitude estimada pelo algoritmo ER-QPLL, quando o sinal descrito em (4.38) foi introduzido na malha.

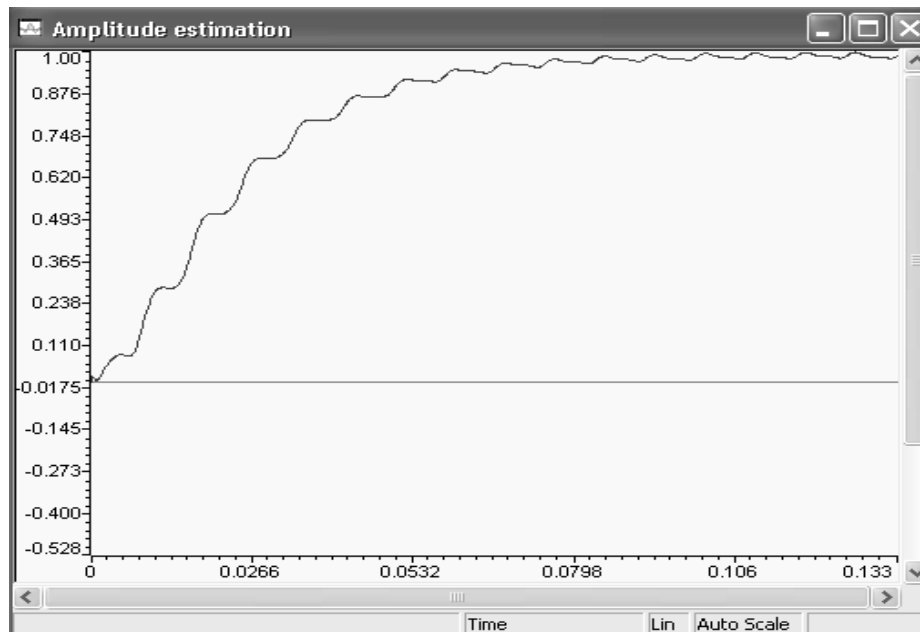


Figura 4.28 - Amplitude estimada pelo ER-QPLL

Nota-se que, em menos de 0,1 segundo, o algoritmo ER-QPLL realiza a estimação da amplitude do sinal de entrada com um erro menor do que 1%. A Figura 4.29 mostra o sinal de entrada – Equação 4.38-, bem como o sinal sintetizado pelo algoritmo ER-QPLL.

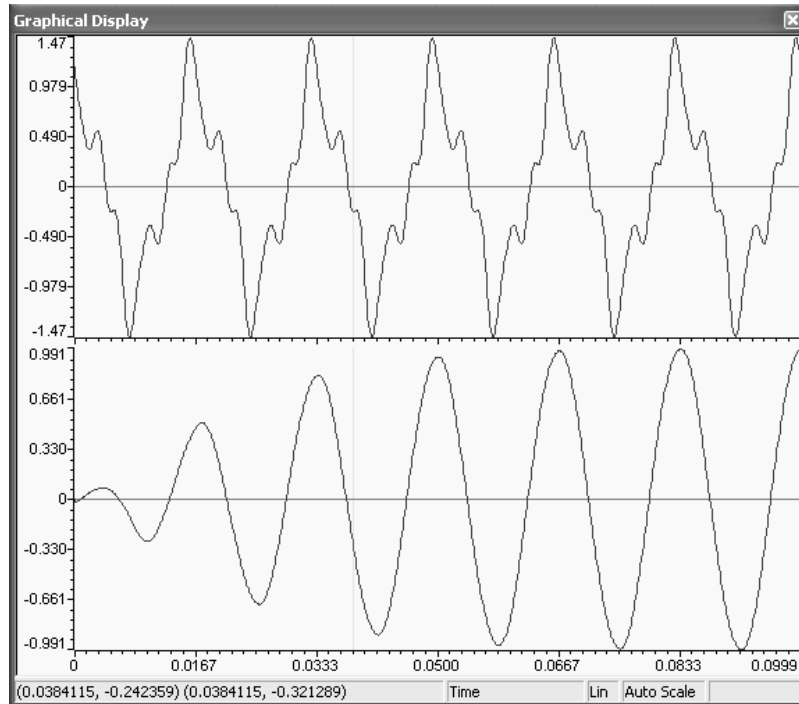


Figura 4.29 - Sinal de entrada e sinal sintetizado pela malha ER-QPLL

Finalmente, é introduzido novamente o sinal descrito pela equação (4.38) na malha ER-QPLL, exceto que, nesse caso, a frequência da componente fundamental de entrada é deslocada para 62 Hz em $t = 0$ segundos. A Figura 4.30 mostra a estimação de frequência realizada pelo algoritmo ER-QPLL

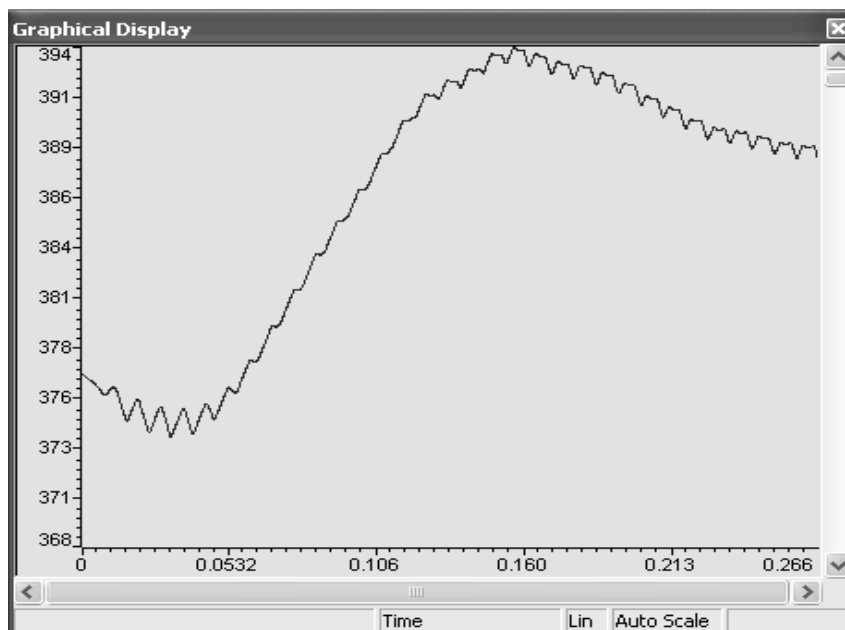


Figura 4.30 – Estimação de frequência pela malha ER-QPLL

Nota-se que o algoritmo demora aproximadamente 200ms para rastrear corretamente o valor da frequência da componente fundamental do sinal de entrada.

É importante lembrar que, caso se deseje realizar uma implementação em tempo real do algoritmo da malha ER-QPLL no microprocessador proposto nessa secção, seria necessário utilizar uma taxa de amostragem de tal modo que, entre duas amostras consecutivas do sinal de entrada, o microprocessador seja capaz de realizar todas as operações necessárias à implementação da malha ER-QPLL. Desse modo, através da utilização do software *Code Composer*, foi calculado o número de ciclos de máquina (número de instruções) gastos pelo microprocessador, na implementação das equações da malha ER-QPLL, no intervalo entre duas amostras consecutivas do sinal de entrada. Chegou-se à conclusão que o número de ciclos máximo gasto pelo microprocessador, na realização das instruções, foi igual a 2.950. Desse modo, sabendo que o parâmetro MIPS (Milhões de Instruções por segundo) do microprocessador utilizado é igual a 150, ou seja, sua velocidade de operação é igual a 150 MHz, tem-se que é possível realizar a implementação do algoritmo ER-QPLL proposto utilizando uma taxa de amostragem de até 50 KHz.

4.7 Conclusão

Esse capítulo apresentou a descrição de uma malha robusta de QPLL, denominada ER-QPLL. A implementação da malha foi efetuada através do mapeamento dos coeficientes do filtro notch e dos integradores da malha E-QPLL, descritas no capítulo anterior, para o domínio do operador delta (δ). O mapeamento para o novo domínio garantiu um funcionamento satisfatório da malha, quando a mesma foi implementada utilizando-se aritmética de precisão finita de, no mínimo, 16 bits. De acordo com os resultados obtidos, comprovou-se o funcionamento da malha utilizando taxas de amostragens superiores a 15.000 Hz. Finalmente, foi proposta a implementação da malha em um microprocessador de ponto fixo. De acordo com os resultados obtidos, comprovou-se a viabilidade de implementação do algoritmo em microprocessadores de ponto fixo de, no mínimo, 16 bits. A simplicidade do algoritmo ER-QPLL também permite a sua implementação em aplicações de tempo real utilizando-se altas taxas de amostragem.

Capítulo 5

Conclusões Gerais

O objetivo do presente trabalho, como destacado na Introdução, foi verificar a factibilidade de se efetuar síntese de um algoritmo proposto, baseado em uma estrutura QPLL modificada, para estimação de parâmetros da componente fundamental de sinais elétricos, denominada ER-QPLL, ou “*Enhanced and Robust Quadrature Phase-Locked-Loop*”. As modificações propostas na estrutura do QPLL convencional foram a inclusão de um filtro passa-faixa adaptativo na entrada do QPLL e a implementação da estrutura resultante utilizando o operador delta (δ).

Para se chegar à síntese do algoritmo proposto, o trabalho se iniciou com uma revisão bibliográfica de malhas PLL baseada em modelos lineares, sendo apresentadas as funções de transferência que descrevem as malhas de PLL de um modo geral. Através do estudo da função de transferência da malha, foi possível realizar os ajustes dos coeficientes que controlam o comportamento dinâmico do PLL, de modo a adequá-lo de uma maneira ótima para cada aplicação de interesse. Mostrou-se, na seqüência, que o mapeamento da função de transferência de um PLL, representado no domínio contínuo para o domínio discreto, deve satisfazer determinadas condições, que foram discutidas e analisadas. Ainda dentro desta revisão inicial foi mostrado o modelo de um PLL denominado QPLL, que permite adquirir todos os parâmetros da componente fundamental de um sinal de entrada de caráter geral. De acordo com as análises efetuadas nesta estrutura denominada QPLL, pode-se concluir que o seu comportamento dinâmico segue o mesmo padrão descrito para o modelo de PLL linear apresentado também discutido.

Passou-se então à descrição de uma estrutura composta por uma malha PLL do tipo Quadratura (QPLL), porém acrescida de um filtro passa-banda adaptativo acoplado à sua entrada. Este processo de filtragem passa-banda foi obtido pela utilização de um filtro do tipo *notch*, centrado na frequência da componente fundamental do sinal de entrada. Por este procedimento, ocorre a eliminação dessa componente que, logo após, é

identificada através de uma simples equação algébrica. A filtragem passa-banda, portanto, aparece com o objetivo de aumentar a relação SNR do sinal de entrada, bem como atenuar as componentes harmônicas e inter-harmônicas de baixa frequência que, porventura, estejam presentes no mesmo.

O filtro passa-faixa introduzido na estrutura é obtido a partir de um filtro notch parametrizado de segunda ordem. Dois parâmetros, α e β , controlam o desempenho do filtro passa-banda. O parâmetro α controla a largura da faixa de passagem do filtro: quanto mais próximo da unidade for este fator, mais seletivo ou sintonizado será o filtro e mais lenta será a sua resposta dinâmica. Já o parâmetro β controla a frequência central do filtro, sendo sua adaptabilidade obtida a partir de estimação da frequência proveniente do estimador QPLL, permitindo que o filtro possa ajustar a frequência central para a frequência da componente fundamental, possibilitando que a estrutura possa operar em sistemas onde a frequência é variante com o tempo.

A inclusão do filtro passa-faixa na entrada do estimador melhora a precisão da estimação, basicamente, pelos seguintes motivos:

- Melhoria da relação sinal ruído (SNR) na entrada do estimador;
- Atenuação das componentes de frequência fora da frequência central do filtro (harmônicas e inter-harmônicas).

Há que se ressaltar, contudo, que a inclusão do filtro aumenta o tempo de convergência do algoritmo devido ao seu comportamento transitório em situações de distúrbios. Entretanto, de acordo com as diversas simulações computacionais realizadas pode-se concluir que a malha QPLL modificada apresenta uma resposta dinâmica mais rápida, em relação à malha QPLL original, se ambas forem ajustadas para apresentarem uma mesma rejeição a distúrbios em regime permanente.

Sabendo-se que o valor da frequência da componente fundamental do sinal de entrada não apresenta um valor fixo, foi necessária adotar-se uma estratégia de ajuste dos coeficientes do filtro, de modo que o mesmo sempre elimine a componente fundamental do sinal de entrada. Neste quesito, duas propostas para atualização dos coeficientes do filtro foram implementadas e analisadas: atualização pelo valor instantâneo da frequência

detectada pela malha QPLL e atualização através de uma média desse valor, integralizada por um período de tempo pré-especificado. De acordo com as análises efetuadas a partir das diversas simulações realizadas, podê-se comprovar que a estratégia de atualizar os coeficientes pela média dos valores revelou-se superior à que utiliza o valor instantâneo.

Diversas simulações computacionais foram então realizadas buscando extrair conclusões sobre os desempenhos comparativos da malha QPLL proposta e da estrutura QPLL convencional. Os resultados obtidos mostraram, categoricamente, a superioridade da malha proposta sobre a convencional, pois a mesma alcançou respostas transitórias mais rápida que a malha QPLL original, para diversas situações analisadas e para uma mesma rejeição a distúrbios, em regime permanente.

A etapa seguinte do trabalho tratou da proposição de uma melhoria na estrutura QPLL utilizada com alterações para aumento de sua robustez, chegando-se à síntese do que se denominou ER-QPLL, ou seja, uma malha robusta de QPLL - ou "*Enhanced and Robust Quadrature Phase-Locked-Loop*". Os procedimentos adotados para melhoria da robustez da estrutura envolveram o mapeamento dos coeficientes do filtro notch e dos integradores da malha E-QPLL, como descrito anteriormente, para o domínio do operador delta (δ). O mapeamento para o novo domínio garantiu um funcionamento satisfatório da malha, quando a mesma foi implementada utilizando-se aritmética de precisão finita de, no mínimo, 16 bits.

Os resultados obtidos comprovaram o bom desempenho da malha, mesmo utilizando taxas de amostragens superiores a 15.000 Hz. Finalmente, foi proposta a implementação da malha em um microprocessador de ponto fixo. De acordo com os resultados obtidos, comprovou-se a viabilidade de implementação do algoritmo em microprocessadores de ponto fixo de, no mínimo, 16 bits. A simplicidade do algoritmo ER-QPLL também permite a sua implementação em aplicações de tempo real utilizando-se altas taxas de amostragem.

O algoritmo ER-QPLL proposto neste trabalho, foi então implementado para atuação em tempo real utilizando, para isto, o microprocessador TMS320F2812, da *Texas Instruments*®, um DSP (*Digital Signal Processor*) de 32 bits e com uma velocidade de operação de 150 MHz, fabricado pela *Texas Instruments*, e utilizando a linguagem de programação C++. O filtro notch foi implementado utilizando a estrutura *DFII*. A

compilação do programa e o envio do código compilado, para a memória RAM do DSP, foram efetuadas através da utilização do software *Code Composer* e do hardware denominado *JTAG*.

Os resultados obtidos permitiram tirar importantes conclusões sobre as condições de implementação, em tempo real, do algoritmo proposto. Chegou-se à conclusão que o número de ciclos máximo (número de operações) gasto pelo microprocessador, na realização do algoritmo, foi igual a 2.950. Desse modo, sabendo que o parâmetro MIPS (Milhões de Instruções por segundo) do microprocessador utilizado é igual a 150, ou seja, sua velocidade de operação é igual à 150 MHz, têm-se que é possível realizar a implementação do algoritmo ER-QPLL proposto utilizando uma taxa de amostragem de até 50 KHz.

Os resultados obtidos e as análises efetuadas permitem selecionar algumas contribuições deste trabalho à área de processamento de sinais, considerando-se especialmente sua utilização para os trabalhos no campo da qualidade da energia elétrica, dentre as quais podem ser destacadas:

- Utilização de filtro passa-faixa adaptativo para a melhoria do sinal de entrada no estimador QPLL;
- Utilização do operador delta (δ) para obtenção de estrutura robusta para implementação em ponto fixo;
- Implementação do algoritmo em processador digital de sinais da família TMS320F2800.

Trabalhos Futuros

São as seguintes as sugestões para a continuidade deste trabalho:

- 1- Utilização da estrutura para estimação de harmônicos e inter-harmônicos.

Pode-se observar que a estrutura ER-QPLL pode ser ajustada para a estimação dos parâmetros das componentes harmônicas. Os principais ajustes encontram-se na inicialização da frequência central do filtro notch e nos coeficientes internos do QPLL. Alguns trabalhos na literatura mostram que uma estrutura em cascata pode ser utilizada para estimação das componentes harmônicas do sinal de entrada, porém

resultados preliminares mostram que uma estrutura do tipo banco de filtros pode apresentar desempenho superior em relação à velocidade de convergência.

A estimação dos inter-harmônicos é possível de se obter com a mesma estrutura, porém se a componente do inter-harmônico estiver próxima de um harmônico de alta energia a tendência do ER-QPLL é estimar a componente de mais alta energia. Este ponto merece mais atenção, haja visto que a estimação de inter-harmônicos ainda é um problema em aberto na literatura.

2- Melhoria da estimação da componente fundamental em presença de flicker.

A presença de flicker ou outra forma de modulação que produza componentes em frequência próxima do 60 Hz, corresponde a uma situação que merece ser melhor estudada. Isto porque estas componentes podem estar dentro da banda de passagem do filtro passa-banda. Nesta situação o sinal apresentado ao estimador EQ-PLL apresenta além da componente de 60 Hz outras componentes que tendem a piorar os resultados da estimação. Uma das proposições para contornar este problemas é o uso da técnica da decimação, de modo que as componentes em torno do 60 Hz apareçam o mais distante possível uma da outra no sinal digital e deste modo a estrutura EQ-PLL poderá ser utilizada com mais eficiência.

3- Melhoria no processo de atualização da frequência central do filtro notch

Neste trabalho foram estudadas duas estratégias de atualização da frequência central do filtro notch. Uma das sugestões para trabalho futuro é a de estudar novas estratégias de atualização do filtro notch, bem como novas técnicas de implementação do mesmo de modo a reduzir o tempo total de convergência do algoritmo. Isto porque existem aplicações em sistemas de potência que requerem tempos de convergência menores que 2 ciclos.

4- Implementação em tempo real

Este trabalho mostrou uma implementação em tempo real utilizando a plataforma DSP da Texas Instruments. Nesta implementação utilizou-se sinais armazenados previamente na memória interna do DSP. Portanto o próximo passo é a construção do sistema de aquisição para a estimação possa ser feita em cima de sinais reais.

Referências Bibliográficas

- [1] JOVCIC, D., “Phase Locked Loop System for FACTS”, IEEE Trans. on Power Systems, v. 18, n. 3, pp. 1116-1124, Agosto 2003.
- [2] KARIMI-GHARTEMANI, M., IRAVANI, M. R., “Measurement of Harmonics/Inter-harmonics of Time-Varying Frequencies”, IEEE Trans. on Power Delivery, v. 20, n. 1, pp. 23-31, Janeiro 2005.
- [3] CARVALHO, J. R., DUQUE, C. A., RIBEIRO, M. V., SZCZUPAL, J. CERQUEIRA, A. S., GOMES, P. H. C., “PLL Based Multirate Harmonic Estimation”, IEEE PES general meeting, v.1 pp. 1-8, 2007.
- [4] KARIMI-GHARTEMANI, M., IRAVANI, M. R., “A Signal Processing Module for Power System Applications”, IEEE Trans. on Power Delivery, v. 18, n. 4, pp. 1118-1126, Outubro 2003.
- [5] POISSON, P., MEUNIER, M., “New signal processing tools applied to power quality analysis”, IEEE Trans. on Power Delivery, v.14 n.2, pp. 561-566, Abril 1999.
- [6] HEYDT, G. T., “Electric power quality: a tutorial introduction”, IEEE Comput. Applicat. Power, v. 11, pp. 15–19, Janeiro 1998.
- [7] ARRILLAGA, J., BOLLEN, M. H. J., WATSON, N. R., “Power quality following deregulation”, em Proc. 2000 of the IEEE, v. 88, n. 2, pp. 246–261.
- [8] DUGAN, R. C., MCGRANAGHAN, M. F., BEATY, H. W., “Electrical Power Systems Quality”, McGraw-Hill, 1996.
- [9] GOMES, P. H.C., DUQUE, C. A., CERQUEIRA, A. S. “Sistema de Monitoramento para Rede Elétrica baseado em DSP”, em: XV Congresso Brasileiro de Automática, 2004, Gramado, RS.
- [10] GEORGE, T. A. e BONES, D., “Harmonic power flow determination using the fast Fourier transform,” *IEEE Trans. Power Del.*, vol 6, pp. 530-535, Abril 1991.
- [11] KARIMI-GHARTEMANI, M. KARIMI, H. e IRAVANI, M. R., “A Magnitude/Phase-Locked Loop System Based on Estimation of Frequency and In-Phase/Quadrature-Phase Amplitudes”, IEEE Trans. on Ind. Electronics, V. 51, N. 2, ABRIL 2004.
- [12] KAURANIEMI, J., LAAKSO, T. I., HARTIMO, I., OVASKA, J., “Delta Operator Realizations of Direct-Form IIR Filters”, IEEE Trans. on circuits and systems – II Analog and digital signal processing, v. 45, n.1 Janeiro 1998.
- [13] KARIMI, H., KARIMI-GHARTEMANI, M., IRAVANI, M. R., “Estimation of Frequency and its Rate of Change for Applications in Power Systems”, IEEE Trans. on power delivery, v. 19, n.2, Abril 2004.

- [14] GOODWIN, G. C., MIDDLETON, R. H., POOR, H. V., “High-speed Digital Signal Processing and Control”, Proceedings of the IEEE, v. 80, n. 2, pp. 240-259, Fevereiro 1992.
- [15] LINDSEY, W. C., CHIE, C. M., “A Survey of Digital Phase-Locked loops”, Proceedings of the IEEE, v. 69, n.4, pp. 410-431, Abril 1981.
- [16] CHUNG, S., “A Phase Tracking System for Three Phase Utility Interface Inverters”, IEEE Transactions on Power Electronics, v. 15, n.3 Maio 2000.
- [17] GUPTA, S.C., “Phase-locked loops”, Proceedings of the IEEE, v. 63, n.2, Fevereiro 1975.
- [18] SHAYAN, Y. R. e LE-NGOC, T., “All digital phase-locked loop: concepts, design and applications”, Proceedings of the IEEE, v. 136, n.1, pp. 53-56, Fevereiro 1989.
- [19] LI, W. e MEINERS, J., “Introduction to Phase-Lock Loop System Modeling”
- [20] SNYMAN, J. A., Practical mathematical optimization: an introduction to basic optimization theory and classical and new gradient-based algorithms, Springer, Nova York, 2005.
- [21] BEST, R. E, Phase-Locked Loops: Design, Simulations and Applications, McGraw-Hill, 2003.
- [22] REEHAL, G., A Digital Frequency Synthesizer Using Phase Locked Loop Technique, Tese de Mestrado, Universidade de Ohio, 1998.
- [23] WANG, M. e SUN, Y., “A practical, precise method for frequency tracking on phasor estimation”, IEEE Trans. on Power Delivery, v. 19, n. 4, pp. 1547-1552, Outubro 2004.
- [24] GIRGIS, A. A., STEPHENS, J. W., e MAKRAM, E. B., “Measurement and prediction of voltage flicker magnitude and frequency”, IEEE Trans. On Power Delivery, v. 10, pp. 1600-1604, Julho 1995.
- [25] KARIMI-GHARTEMANI, M. e IRAVANI, M. R., “Nonlinear Adaptive Filter for online signal analysis in Power system applications”, IEEE Trans. On Power Delivery, v. 17, n. 2, pp. 617-622, Abril 2002.
- [26] HIRANO, K., NISHIMURA, S. e MITRA, S. K., “Design of Digital Notch Filters”, IEEE Trans. on Communications, v.22, n. 7, pp. 964-970, Julho 1974.
- [27] MITRA, S. K., Digital Signal Processing - A computer Based Approach., 2. ed. NJ, McGraw-Hill, 2001.
- [28] MARAFÃO, P. M., Análise e Controle da Energia Elétrica Através de Técnicas de Processamento Digital de Sinais, Tese de Doutorado, UNICAMP, 2004.
- [29] ROLF, I., Digital Control Systems, Springer Verlag, Berlin, 1981.

- [30] OGATA, K., Discrete-time Control Systems, Prentice-Hall International Editions, Englewood Cliffs, 1987.
- [31] EYKOFF, P. "System identification Process Parameter and State Estimation" J. Wiley, 1974.
- [32] RIBEIRO, M. V., Técnicas de Processamento de Sinais Aplicadas na Transmissão de Dados via Rede Elétrica e ao Monitoramento da Qualidade de Energia, Tese de Doutorado, UNICAMP, 2005.

Anexo – Equações da malha ER-QPLL implementada no microprocessador TMS320F2812

```

for (j=0;j<7680;j++)
{

ux= data_in;

/*Filtro notch no domínio do operador delta
-----*/

//o2(i)=o2(i-1) + delta*(Bd(3)*x(i-1) - Ad(3)*k(i-1));
aux1 = produto(10104,x_antes,0,14,1);
aux2 = produto(10353,k_antes,0,14,1);
aux3 = soma(aux1,-aux2,1,1,1);
aux4 = produto(aux3,delta,1,15,11);
o2 = soma(o2_antes,aux4,7,11,7);

//o1(i)=o1(i-1) + delta*(Bd(2)*x(i-1) - Ad(2)*k(i-1) + o2(i-1));

aux1 = produto(5,x_antes,0,14,12);
aux2 = produto(42,k_antes,0,14,8);
aux3 = soma(aux1,-aux2,12,8,7);
aux4 = soma(aux3,o2_antes,7,7,6);
aux5 = produto(aux4,delta,6,15,15);
o1 = soma(o1_antes,aux5,14,15,14);

//k(i) = (Bd(1)*x(i) + o1(i))/Ad(1);

k = soma(x,o1,14,14,14);

//u(i) = x(i) - k(i);

u2 = soma(x,-k,14,14,14);

//Fim da secção do filtro notch
//-----

//Início da malha de PLL
//-----

//Primeiro integrador no dominio delta

int1_antes = produto(int1_antes,C1,8,15,10);
Ks = produto(int1_antes,delta,10,15,14);
Ks = soma(Ks,Ks_antes,14,14,14);

```



```

//Segundo integrador no dominio delta

int2_antes = produto(int2_antes,C1,8,15,10);
Kc = produto(int2_antes,delta,10,15,14);
Kc = soma(Kc,Kc_antes,14,14,14);

//Terceiro integrador no dominio

int3_antes = produto(int3_antes,C1,6,15,8);
dw = produto(int3_antes,delta,8,15,12);
dw = soma(dw,dw_antes,12,6,6);

//Quarto integrador

dw_antes = produto(dw_antes,C1,6,15,8);
fase = produto(dw_antes,delta,8,15,12);
fase = soma(fase,fase_antes,12,12,12);

//Normalização da fase para permanecer no intervalo
//entre -pi e pi;

if (fase > 25736/2)
    fase = -(25736 - fase);

//Normalização da fase para permanecer no intervalo -1 e +1

fasex = produto(fase,10430,12,15,15);

a=qsin(fasex);
b=qcos(fasex);

c = produto(Ks,a,14,15,14);
d = produto(Kc,b,14,15,14);
y = soma(c,d,14,14,14);
e = soma(x,-y_antes,14,14,13);
e = produto(e,100,13,0,6);
int1 = produto(e,a,6,15,8);
int2 = produto(e,b,6,15,8);
n = produto(int2,Ks,8,14,11);
m = produto(Kc,int1,14,8,11);
int3 = soma(-m,n,11,11,10);
int3 = produto(int3,50,10,0,6);

//Delay das variáveis

delta[0]=dw;

int3_antes = int3;
int2_antes = int2;
int1_antes = int1;
fase_antes = fase;

```

```

Kc_antes = Kc;
Ks_antes = Ks;
dw_antes = dw;
o1_antes = o1;
o2_antes = o2;
k_antes = k;
x_antes = x;
y_antes = y;

//Somatório da frequência média

W2 = dw;
W = W + W2;

//Rotina de atualização do FILTRO NOTCH!
//*****

if (i==127)
{
W = W >> 7; //Divide a frequência por 128
w0 = W; //Joga-a para um número de ponto flutuante
w0 = w0/64; //Normaliza para o valor Q0
w0 = w0/7680; //Acha o seu valor normalizado
a0 = -2*cos(w0); //Acha o parâmetro a0;

//Acha os parâmetros no domínio Z

B2 = a0;
A2 = a0*.991;
A3 = .991*.991;

//Converte os parâmetros para o domínio delta

Bd2 = (2 + B2)*2048;
Bd3 = (1 + B2 + 1)*(2048.0*2048.0);

Ad2 = (2 + A2)*2048;
Ad3 = (1 + A2 + A3)*(2048.0*2048.0);

BD2 = round(Bd2);
BD3 = round(Bd3);

AD2 = round(Ad2);
AD3 = round(Ad3);

i=0;

}

//*****

```

```
i++;
```

```
}
```

```
while(1)
```

```
j=j;
```

```
}
```