



ISEL
INSTITUTO SUPERIOR DE
ENGENHARIA DE LISBOA

INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

Área Departamental de Engenharia Eletrotécnica de Energia e Automação

Análise e Implementação Digital de uma Malha de Captura de Fase para Sincronização de Conversores à Rede Elétrica

SÉRGIO ALEXANDRE ALVES ANDRÉ

Licenciado em Engenharia Eletrotécnica e Automação

Dissertação para a obtenção do grau de Mestre em Engenharia Eletrotécnica Ramo de
Automação e Eletrónica Industrial

Orientador:

Professor Doutor Vasco Emanuel Anjos Soares

Júri:

Presidente: Professor Doutor João Hermínio Ninitas Lagarto

Vogais:

Professor Doutor Vasco Emanuel Anjos Soares

Professor Doutor Fernando Manuel Fernandes Melício

Dezembro de 2015



ISEL
INSTITUTO SUPERIOR DE
ENGENHARIA DE LISBOA

INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

Área Departamental de Engenharia Eletrotécnica de Energia e Automação

**Análise e Implementação Digital de uma Malha de
Captura de Fase para Sincronização de Conversores à
Rede Elétrica**

Dezembro de 2015

Resumo

O presente trabalho focou-se na simulação e implementação de uma malha de captura de fase (PLL) para obtenção do ângulo respeitante à tensão da rede elétrica. Este método é implementado em conversores estáticos de potência, sendo indispensável à sincronização dos mesmos à rede elétrica.

Ao longo desta dissertação foram descritos diferentes tipos de malhas de capturas de fase com características específicas, podendo ser aplicadas numa grande diversidade de áreas.

Nos estudos realizados foi possível constatar que não existem parâmetros de dimensionamento ideais, uma vez que, para a aplicação pretendida requer-se um erro estático, velocidade resposta e amortecimento próprios para atender às características da rede elétrica e ligação dos conversores à mesma.

Partindo das soluções analógicas clássicas, foi realizada uma implementação digital equivalente com um microcontrolador de 32 bits, tendo sido comparados resultados de simulações realizados em MATLAB com resultados experimentais.

Abstract

This document focused on the simulation and implementation of a phase locked loop (PLL) to obtain the angle of the electrical grid. This method can be implemented in static power converters and it is indispensable to synchronize the converter to the grid.

Along the present work are described different phase locked loops with specific characteristics. The PLL has a great capacity to be used in other areas of engineering.

Developed studies were able to verify that no ideal design parameters are possible, since the intended application required a static error, a response speed and a damping considering the power grid characteristics and converters connected to them.

Starting from the classic analog solutions, a digital implementation was performed with an 32 bit microcontroller and the results were compared with the simulations performed in MATLAB.

Agradecimentos

A realização de um trabalho como o desenvolvido engloba não só o autor mas também inúmeras pessoas que contribuíram não só com conhecimento mas também com envolvimento pessoal. Desta forma, gostaria de realizar alguns agradecimentos e pedir desculpa a quem eventualmente me tenha esquecido de mencionar.

Gostaria em primeiro lugar de fazer um especial agradecimento ao meu orientador, Professor Doutor Vasco Soares, por toda a disponibilidade inesgotável, capacidade de ajuda e partilha de conhecimento em relação ao assunto.

Agradeço também à minha namorada, Andrea Santos, por toda a paciência e animo que me foi transmitindo ao longo do desenvolvimento deste trabalho, aproveitando aqui para agradecer também à minha família, mãe, pai, irmã e madrinha, pelos mesmos motivos.

Aos meus colegas do ISEL, Alexandre Bento, Miguel Rhodes e Mykhaylo Zahyka, aqui apresento o meu agradecimento, por alguns conselhos técnicos e companheirismo.

Por fim, mas não menos importantes, agradeço a toda a equipa do ISEL Formula Student, por toda a paciência e compreensão ao longo deste percurso.

Sérgio André,
Dezembro de 2015

Índice

| | |
|---|------|
| Resumo | i |
| Abstract | iii |
| Agradecimentos | v |
| Índice de figuras | ix |
| Índice de tabelas..... | xiii |
| Símbolos e variáveis..... | xiv |
| Acrónimos e abreviaturas | xvi |
| Capítulo I: Introdução..... | 1 |
| 1.1 – Enquadramento e motivação | 2 |
| 1.2 – Objetivos da dissertação..... | 2 |
| 1.3 – Estrutura da dissertação | 3 |
| Capítulo II: Malha de captura de fase (PLL) | 5 |
| 2.1 – Introdução ao PLL | 6 |
| 2.2 – Detetores de fase | 6 |
| 2.3 – Filtros..... | 17 |
| 2.4 – Osciladores controlados por tensão (VCO)..... | 22 |
| 2.5 – Divisor de frequência | 25 |
| Capítulo III: Dimensionamento e Modelização do PLL | 27 |
| 3.1 – Introdução..... | 28 |
| 3.2 – Escolha do detetor de fase | 29 |
| 3.3 – Escolha do filtro | 29 |
| 3.4 – Estudo do sistema através da análise de funções transferência | 29 |
| Capítulo IV: <i>Software</i> PLL | 35 |
| 4.1 – <i>Software</i> PLL | 36 |
| 4.2 – Discretização | 40 |
| 4.3 – SPLL com tensão de realimentação quadrada..... | 43 |

| | |
|---|----|
| 4.4 – SPLL com tensão de realimentação sinusoidal..... | 44 |
| 4.5 – SPLL com tensão de realimentação sinusoidal e filtro rejeita-banda..... | 47 |
| 4.6 – SPLL com tensão de realimentação sinusoidal e gerador de sinal ortogonal..... | 51 |
| Capitulo V: Implementação e Resultados..... | 55 |
| 5.1 – Esquema de ligações da implementação prática | 56 |
| 5.2 – Resposta do sistema digital para uma tensão de entrada sem conteúdo harmónico (sinusoidal) | 57 |
| 5.3 – Resposta do sistema digital para uma tensão de entrada com conteúdo harmónico | 64 |
| 5.4 – Resultados obtidos através de implementação | 68 |
| Capitulo VI: Conclusões e propostas de trabalho futuras | 75 |
| 6.1 – Conclusões..... | 76 |
| 6.2 – Propostas de trabalho futuro..... | 77 |
| Referências bibliográficas | 78 |
| Anexos..... | 79 |
| Anexo 1 – Verificação matemática do aparecimento da 2ª harmónica na saída do detetor de fase multiplicador..... | 80 |
| Anexo 2 – Código para cálculo e obtenção de figuras do sistema analógico | 82 |
| Anexo 3 – Código utilizado para simulação do processamento do PIC..... | 85 |
| Anexo 4 – Código exemplo utilizado na implementação com microcontrolador | 97 |

Índice de figuras

| | |
|---|----|
| Figura 1 – Diagrama de blocos representativo de um PLL..... | 6 |
| Figura 2 – Diagrama de blocos de um PLL do tipo Multiplicador..... | 7 |
| Figura 3 – Gráfico da tensão de entrada e saída do PLL multiplicador. | 7 |
| Figura 4 – Evolução do ganho K_d em função do valor eficaz da tensão de entrada (U_1). | 8 |
| Figura 5 – Símbolo digital do OU exclusivo..... | 8 |
| Figura 6 – Gráfico da tensão de entrada e saída do PLL OU exclusivo. | 10 |
| Figura 7 – Evolução da tensão U_d com a variação do erro no ângulo. | 10 |
| Figura 8 – Símbolo de um flip-flop JK. | 11 |
| Figura 9 – Gráfico da tensão de entrada e saída do PLL flip-flop JK..... | 12 |
| Figura 10 – Evolução da tensão U_d com a variação do erro no ângulo. | 12 |
| Figura 11 – Circuito de um PFD com saída em tensão. | 13 |
| Figura 12 – Diagrama representativo dos três níveis do PFD. | 13 |
| Figura 13 – Circuito de um PFD com saída em corrente..... | 14 |
| Figura 14 – Gráfico da tensão de entrada e saída do PFD. | 15 |
| Figura 15 – Evolução da tensão U_d com a variação do erro no ângulo. | 15 |
| Figura 16 – Circuito de um filtro leadlag passivo (entrada em tensão). | 17 |
| Figura 17 – Resposta em frequência de um filtro leadlag passivo (entrada em tensão). | 18 |
| Figura 18 – Circuito de um filtro leadlag passivo (entrada em corrente). | 18 |
| Figura 19 – Resposta em frequência de um filtro leadlag passivo (entrada em corrente). . | 19 |
| Figura 20 – Circuito de um filtro leadlag ativo (entrada em tensão). | 19 |
| Figura 21 – Resposta em frequência de um filtro leadlag ativo. | 20 |
| Figura 22 – Circuito de um filtro leadlag ativo (entrada em corrente). | 20 |
| Figura 23 – Circuito de um filtro PI ativo (entrada em tensão). | 21 |
| Figura 24 – Circuito de um filtro PI ativo (entrada em corrente). | 21 |
| Figura 25 – Gráfico da evolução da frequência de saída do VCO em função da tensão de entrada. | 23 |
| Figura 26 – Circuito de um VCO de relaxamento utilizado no circuito integrado 74HC/HCT4046. | 24 |
| Figura 27 – Diagrama de blocos com funções transferência de um PLL..... | 30 |
| Figura 28 – Gráfico da evolução do sistema ao longo do tempo e frequência natural para diferentes valores de ζ | 31 |
| Figura 29 – Resposta temporal a uma variação do tipo escalão no ângulo de entrada. | 32 |
| Figura 30 – Resposta em frequência da função de transferência em cadeia fechada do sistema simplificado. | 33 |

| | |
|--|----|
| Figura 31 – Gráfico da estabilidade estática e dinâmica de um LPLL de 2ª Ordem. | 34 |
| Figura 32 – Fluxograma representativo do algoritmo do PLL implementado com multiplicador. | 37 |
| Figura 33 – Fluxograma representativo do algoritmo do PLL implementado com PFD. | 39 |
| Figura 34 – Diagrama dos três tipos de integrações de Euler. | 40 |
| Figura 35 – Diagrama de Bode do filtro PI no domínio contínuo e discreto. | 41 |
| Figura 36 – Diagrama do PLL com tensão de retroação quadrada. | 43 |
| Figura 37 – Diagrama do PLL com tensão de retroação sinusoidal. | 44 |
| Figura 38 – Sinais de entrada e saída do detetor de fase multiplicador. | 45 |
| Figura 39 – Resultado da simulação para 50 Hz para o valor do erro no ângulo do PLL com retroação sinusoidal. | 46 |
| Figura 40 – Diagrama simplificado do PLL com filtro rejeita-banda. | 47 |
| Figura 41 – Diagrama simplificado do PLL com gerador de sinal ortogonal. | 47 |
| Figura 42 – Diagrama do PLL com filtro rejeita-banda. | 47 |
| Figura 43 – Diagrama de bode de filtro rejeita-banda. | 49 |
| Figura 44 – Diagrama de Bode do filtro rejeita-banda no domínio contínuo e discreto. | 50 |
| Figura 45 – Diagrama de Bode do filtro rejeita-banda variando o fator de qualidade. | 51 |
| Figura 46 – Diagrama do PLL com gerador de sinal ortogonal. | 51 |
| Figura 47 – Diagrama do gerador de sinal ortogonal. | 52 |
| Figura 48 – Esquema de ligações do PIC32MX230F256B, utilizado para implementação. | 56 |
| Figura 49 – Tensões de entrada e saídas do SPLL para uma frequência de 47 Hz para as várias implementações a) Simulação em regime contínuo b) Simulação em regime discreto. | 58 |
| Figura 50 – Tensões de entrada e saídas do SPLL para uma frequência de 50 Hz para as várias implementações a) Simulação em regime contínuo b) Simulação em regime discreto. | 59 |
| Figura 51 – Tensões de entrada e saídas do SPLL para uma frequência de 52 Hz para as várias implementações a) Simulação em regime contínuo b) Simulação em regime discreto. | 59 |
| Figura 52 – Diferença entre o ângulo de entrada e saída do SPLL para uma frequência de 47 Hz para as diferentes implementações. | 61 |
| Figura 53 – Diferença entre o ângulo de entrada e saída do SPLL para uma frequência de 50 Hz para as diferentes implementações. | 61 |
| Figura 54 – Diferença entre o ângulo de entrada e saída do SPLL para uma frequência de 52 Hz para as diferentes implementações. | 61 |
| Figura 55 – Diferença entre o ângulo de entrada e saída do SPLL para uma frequência de 47 Hz para as diferentes implementações (em regime estacionário). | 62 |
| Figura 56 – Diferença entre o ângulo de entrada e saída do SPLL para uma frequência de 50 Hz para as diferentes implementações (em regime estacionário). | 63 |

| | |
|--|-----------|
| Figura 57 – Diferença entre o ângulo de entrada e saída do SPLL para uma frequência de 52 Hz para as diferentes implementações (em regime estacionário). | 63 |
| Figura 58 – Tensões de entrada e saídas do SPLL para uma frequência de 47 Hz para as várias implementações em regime contínuo. | 65 |
| Figura 59 – Tensões de entrada e saídas do SPLL para uma frequência de 50 Hz para as várias implementações em regime contínuo. | 65 |
| Figura 60 – Tensões de entrada e saídas do SPLL para uma frequência de 52 Hz para as várias implementações em regime contínuo. | 66 |
| Figura 61 – Diferença entre o ângulo de entrada e saída do SPLL para uma frequência de 47 Hz para as diferentes implementações (em regime estacionário). | 67 |
| Figura 62 – Diferença entre o ângulo de entrada e saída do SPLL para uma frequência de 50 Hz para as diferentes implementações (em regime estacionário). | 67 |
| Figura 63 – Diferença entre o ângulo de entrada e saída do SPLL para uma frequência de 52 Hz para as diferentes implementações (em regime estacionário). | 68 |
| <i>Figura 64 – Tensões de entrada e saídas do com sinal de retroação quadrado para um sinal de entrada com 47 Hz a)Simulação b)Experimental.</i> | <i>69</i> |
| <i>Figura 65 – Tensões de entrada e saídas do SPLL com sinal de retroação sinusoidal para um sinal de entrada com 47 Hz a)Simulação b)Experimental.</i> | <i>69</i> |
| <i>Figura 66 – Tensões de entrada e saídas do SPLL com sinal de retroação sinusoidal e filtro rejeita-banda para um sinal de entrada com 47 Hz a)Simulação b)Experimental.....</i> | <i>70</i> |
| <i>Figura 67 – Tensões de entrada e saídas do SPLL com gerador de sinal ortogonal para um sinal de entrada com 47 Hz a)Simulação b)Experimental.</i> | <i>70</i> |
| <i>Figura 68 – Tensões de entrada e saídas do com sinal de retroação quadrado para um sinal de entrada com 50 Hz a)Simulação b)Experimental.</i> | <i>71</i> |
| <i>Figura 69 – Tensões de entrada e saídas do SPLL com sinal de retroação sinusoidal para um sinal de entrada com 50 Hz a)Simulação b)Experimental.</i> | <i>71</i> |
| <i>Figura 70 – Tensões de entrada e saídas do SPLL com sinal de retroação sinusoidal e filtro rejeita-banda para um sinal de entrada com 50 Hz a)Simulação b)Experimental.....</i> | <i>72</i> |
| <i>Figura 71 – Tensões de entrada e saídas do SPLL com gerador de sinal ortogonal para um sinal de entrada com 50 Hz a)Simulação b)Experimental.</i> | <i>72</i> |
| <i>Figura 72 – Tensões de entrada e saídas do com sinal de retroação quadrado para um sinal de entrada com 52 Hz a)Simulação b)Experimental.</i> | <i>73</i> |
| <i>Figura 73 – Tensões de entrada e saídas do SPLL com sinal de retroação sinusoidal para um sinal de entrada com 52 Hz a)Simulação b)Experimental.</i> | <i>73</i> |
| <i>Figura 74 – Tensões de entrada e saídas do SPLL com sinal de retroação sinusoidal e filtro rejeita-banda para um sinal de entrada com 52 Hz a)Simulação b)Experimental.....</i> | <i>74</i> |

Figura 75 – Tensões de entrada e saídas do SPLL com gerador de sinal ortogonal para um sinal de entrada com 52 Hz a) Simulação b) Experimental.74

Índice de tabelas

| | |
|--|----|
| Tabela 1 – Tabela de estados lógicos de entradas e saídas num detetor de fase do tipo OU exclusivo. | 9 |
| Tabela 2 – Tabela com estados lógicos do flip-flop J-K. | 11 |
| Tabela 2 – Tabela com resumo dos vários detetores de fase. | 16 |
| Tabela 3 – Tabela com resumo dos filtros normalmente utilizados com o PLL. | 22 |
| Tabela 4 – Tabela com valores harmónicos de tensão permitidos pela normalização. | 28 |
| Tabela 5 – Tabela com descrição da legenda dos resultados obtidos (Tensão). | 58 |
| Tabela 6 – Tabela com descrição da legenda dos resultados obtidos (erro no ângulo). | 60 |

Símbolos e variáveis

θ_1, θ_i – Ângulo de entrada da malha de captura de fase (*Phase Locked Loop* – PLL) – (rad)

θ_e – Ângulo de saída do detetor de fase (ângulo de erro) – (rad)

θ_2 – Ângulo de saída do oscilador controlado por tensão (*Voltage Controlled Oscillator* – VCO) ou do oscilador controlado digitalmente (*Digital Controlled Oscillator* - DCO) – (rad)

θ'_2 – Ângulo de saída do divisor de frequência – (rad)

θ_o – Ângulo de saída do PLL – (rad)

C, C_1, C_2 – Condensadores utilizados nos filtros descritos – (F)

f_c – Frequência central do filtro rejeita-banda – (Hz)

f_{min} – Frequência mínima do VCO – (Hz)

f_{med} – Frequência média do VCO – (Hz)

f_{max} – Frequência máxima do VCO – (Hz)

ω_0 – Frequência angular média do VCO – (rad/s)

ω_1 – Frequência angular do sinal de entrada – (rad/s)

ω_2 – Frequência angular do sinal de saída – (rad/s)

ω_n – Frequência angular natural do sistema – (rad/s)

ω_c – Frequência angular média do filtro rejeita-banda – (rad/s)

ω_s – Frequência angular superior do filtro rejeita-banda – (rad/s)

ω_i – Frequência angular inferior do filtro rejeita-banda – (rad/s)

ω_{3dB} – Frequência de Corte – (rad/s)

$\Delta\omega_p$ – Margem de pull-in – (rad/s)

$\Delta\omega_H$ – Margem de manutenção – (rad/s)

$\Delta\omega_{po}$ – Margem de pull-out – (rad/s)

$\Delta\omega_L$ – Margem de captura – (rad/s)

$\Delta\omega_{max}$ – Margem máxima de frequência – (rad/s)

- K_d – Ganho do detetor de fase
- K_0 – Ganho do VCO – (rad/s/V)
- G_0 – Ganho do filtro rejeita-banda
- R_1, R_2 – Resistências utilizadas nos filtros descritos – (Ω)
- t – Tempo – (seg)
- t_{est} – Tempo de estabilização do PLL – (s)
- T_s – Tempo de amostragem – (s)
- τ_1, τ_2, τ_3 – Constantes de tempo dos filtros descritos – (s)
- u_1 – Tensão de entrada do PLL – (V)
- u_d – Tensão de saída do detetor de fase – (V)
- u_f – Tensão de saída do filtro – (V)
- u_2 – Tensão de saída do VCO ou DCO – (V)
- u'_2 – Tensão de saída do divisor de frequência – (V)
- U_{10} – Valor máximo da tensão de entrada – (V)
- U_{20} – Valor máximo da tensão de retroação – (V)
- U_B – Diferença de potencial da saída do detetor de fase – (V)
- U_{sat+} – Tensão máxima de saída do detetor de fase – (V)
- U_{sat-} – Tensão mínima de saída do detetor de fase – (V)
- U_{cc} – Tensão de alimentação do circuito (normalmente 5 V) – (V)
- Q – Fator de qualidade do filtro rejeita-banda
- N – Divisor de frequência
- ζ – Parâmetro de controlo da resposta do sistema
- LB – Largura de banda do filtro rejeita-banda – (Hz)
- $F(s), H(s)$ – Funções de transferência dos filtros
- $G(s)$ – Funções de transferência de cadeia aberta do sistema

Acrónimos e abreviaturas

ADC – Conversor de sinal analógico para digital (*Analog to Digital Converter*)

DCO – Oscilador controlado por sinal digital (*Digital Controlled Oscillator*)

DPLL – Malha de captura de fase digital (*Digital Phase Locked Loop*)

DSP – Processador de sinal digital (*Digital Signal Processor*)

FPGA – Arranjo de Portas Programável em Campo (*Field-programmable gate arrays*)

ITAE – Integral do erro absoluto multiplicado pelo tempo (*Integral Time Absolute Error*)

LPLL – Malha de captura de fase linear (*Linear Phase Locked Loop*)

PFD – Detetor de fase e frequência (*Phase Frequency Detector*)

PLL – Malha de captura de fase (*Phase Locked Loop*)

PI – Controlador ou filtro proporcional integral (*Proportional Integrative*)

SPLL – Malha de captura de fase implementado em *software* (*Software Phase Locked Loop*)

VCO – Oscilador controlado por tensão (*Voltage Controlled Oscillator*)

Capítulo I: Introdução

Resumo:

Neste Capítulo será feita uma pequena introdução ao trabalho desenvolvido, englobando assim o enquadramento e motivação, onde é explicado o porquê da seleção do tema, os objetivos que seriam apropriados dado este trabalho e uma explicação da estrutura da dissertação.

1.1 – Enquadramento e motivação

Nos dias de hoje cada vez mais são utilizadas fontes eletrónicas comutadas que introduzem harmónicas na rede eléctrica, seja devido a equipamentos de eletrónica de consumo, seja na interligação na rede de dispositivos de energias renováveis. O excesso de harmónicas na rede pode provocar danos nos componentes eletrónicos ligados a esta, surgindo desta forma a necessidade de compensar estas perturbações. Um parâmetro indispensável para a compensação de harmónicas na rede eléctrica consiste no desenvolvimento de sistemas de sincronização tendo dado origem a diversos estudos [1] [2], onde se conclui que este pode ser determinado quer em malha aberta quer em malha fechada.

Com esta dissertação pretende-se estudar e conceber um sistema que determine o ângulo da tensão da rede eléctrica, ou seja uma malha de captura de fase (PLL – Phase Locked Loop) que possibilite a sincronização de conversores eletrónicos. Estes circuitos apresentam algumas vantagens, rápida resposta a variações de fase, frequência e tensão e capacidade de rejeição harmónica, que não são normalmente possibilitados pelos controlos em malha aberta. Além destas vantagens existem ainda melhorias que podem ser adicionadas aos PLLs de forma a reduzir o erro estacionário do ângulo. Atualmente grande parte dos estudos foca-se na implementação de PLLs com geradores de sinal ortogonal ou na adição de filtros à cadeia fechada de forma a reduzir este erro em ondas sinusoidais [3] [4].

Em aplicações industriais, estes algoritmos são implementados em processadores digitais, tais como microcontroladores, Digital Signal Processors (DSPs) e Field-programmable gate arrays (FPGAs). Para um funcionamento correto destas implementações é necessária uma correta discretização e uma boa seleção do processador utilizado.

1.2 – Objetivos da dissertação

Este trabalho tem como objetivo o estudo, o projeto e a implementação de uma malha de captura de fase (PLL) que permita a sincronização de conversores à rede eléctrica.

A malha de captura de fase dimensionada deve ter a capacidade de ser suficientemente robusta perante as harmónicas presentes na forma de onda da tensão da rede eléctrica. Deve ainda possuir a capacidade de obter uma imagem do ângulo da tensão da rede, ou seja, deve produzir um sinal sincronizado com a 1.^a harmónica da rede eléctrica. Além disto, deverá ter uma dinâmica suficientemente lenta para compensar/insensibilizar a alguns tipos de perturbação, tais como cavas de tensão.

Além do descrito acima é de referir que a malha de captura de fase implementada neste trabalho deverá ser realizada na forma discreta (em *Software*), através de um microcontrolador.

1.3 – Estrutura da dissertação

Este trabalho foi estruturado em 6 capítulos iniciando-se com uma introdução ao trabalho desenvolvido, no Capítulo I: Introdução.

Depois de feita uma introdução ao trabalho desenvolvido, no Capítulo II: Malha de captura de fase (PLL) é feita uma introdução ao PLL, são fornecidas algumas informações sobre este tipo de circuitos, algumas configurações possíveis e uma descrição dos seus blocos constituintes.

Após este capítulo, dá-se início ao dimensionamento de um PLL no Capítulo III: Dimensionamento e Modelização do PLL. Neste capítulo, é realizado um dimensionamento genérico sendo o dimensionamento específico à implementação feito no capítulo seguinte. No final deste capítulo são verificadas as margens de frequência do PLL.

No capítulo seguinte, Capítulo IV: *Software* PLL, são mencionadas algumas informações deste tipo de PLL e qual a metodologia que deve ser adotada para o seu projeto. É ainda neste capítulo que está presente o dimensionamento SPLL, do filtro rejeita-banda e do gerador de sinal ortogonal.

O Capítulo V: Implementação e Resultados apresenta os circuitos desenhados no programa de simulação Altium tal como foram implementados e todos os resultados obtidos através do Matlab e através de implementação prática.

Por fim, finaliza-se o trabalho com Capítulo VI: Conclusões e propostas de trabalho futuras onde são apresentadas todas as conclusões retiradas ao longo deste trabalho para as diferentes implementações, deixando-se em aberto uma possível proposta de estudo futuro.

Capítulo II: Malha de captura de fase (PLL)

Resumo:

Durante este capítulo são fornecidas todas as informações relevantes relacionadas com os PLL, como são estudados e todos os seus constituintes.

Começar-se-á por falar dos diferentes detetores de fase existentes e suas características. Seguidamente falar-se-á de alguns filtros que normalmente são mais utilizados para este tipo de aplicações, sendo depois mencionados os dois tipos de VCO existentes e por fim o divisor de frequências.

Todos estes constituintes são explicados detalhadamente, sendo possível depois deste capítulo seleccionar cada um deles averiguando os prós e contras de cada um, tendo em conta a aplicação do PLL.

2.1 – Introdução ao PLL

Uma malha de captura de fase é um circuito eletrónico que pode ter várias aplicações, tais como multiplicador ou divisor de frequências, sincronização de sinais, entre outras.

Na imagem seguinte, Figura 1, está a representação de um PLL em diagrama de blocos segundo [5].

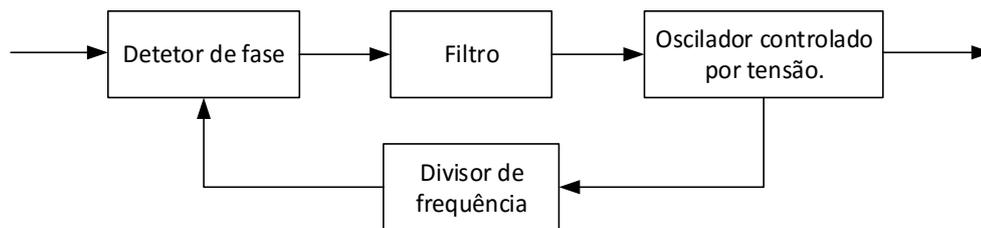


Figura 1 – Diagrama de blocos representativo de um PLL.

Cada bloco deste diagrama pode ser representado por uma função matemática, no entanto existem vários tipos de detetores de fase, de filtros e de osciladores controlados que serão descritos.

2.2 – Detetores de fase

A solução clássica do PLL poderá ser implementada com diferentes tipos de detetores de fase que seguidamente se descrevem.

- **Tipo Multiplicador**

Este tipo de detetor de fase é maioritariamente utilizado na malha de captura de fase linear (*Linear Phase Locked Loop* – LPLL).

A sua tensão de entrada ($u_1(t)$) é normalmente sinusoidal e a sua tensão de retroação ($u_2(t)$) é normalmente quadrada ou sinusoidal.

A equação (1) mostra o ganho deste detetor de fase (K_d) quando submetido a uma tensão de entrada sinusoidal e quadrada na retroação (demonstração no anexo 1).

$$K_d = \frac{2 U_{10} U_{20}}{\pi} \quad (1)$$

Onde U_{10} é o valor máximo da tensão de entrada e U_{20} o valor máximo da tensão de retroação.

O modelo matemático deste detetor de fase é apenas uma multiplicação dos dois sinais (u_1 – tensão de entrada do PLL e u_2' - tensão de retroação do PLL). Devido aos limites de alimentação do circuito multiplicador para obter um modelo mais realista, deve ser adicionado um bloco de saturação tal como mostra a Figura 2. O parâmetro de saída do detetor de fase é a tensão u_d .

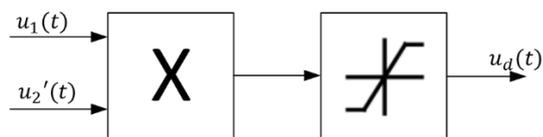


Figura 2 – Diagrama de blocos de um PLL do tipo Multiplicador.

Este tipo de detetor de fase tem uma característica que consiste na fase da tensão de saída estar avançada 90° em relação à onda de entrada, tal como ilustra a Figura 3.

Esta figura representa as ondas de tensão de entrada (u_1) e saída (u_2) quando o PLL está perfeitamente sincronizado.

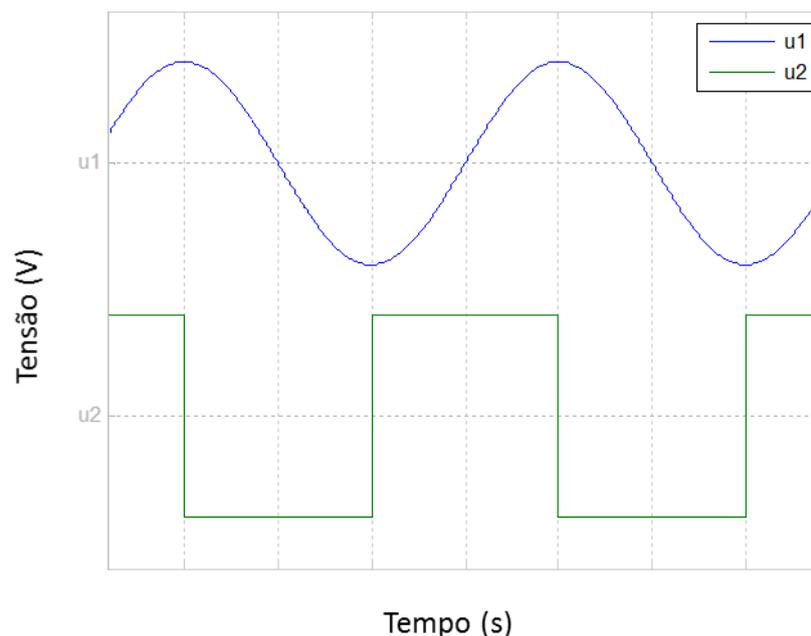


Figura 3 – Gráfico da tensão de entrada e saída do PLL multiplicador.

Apesar de existir um valor de ganho K_d (ganho do detetor de fase) que deveria ser constante, devido ao efeito da saturação do multiplicador, o crescimento deste valor é reduzido para tensões U_{10} mais elevadas. Este acontecimento é mostrado na Figura 4.

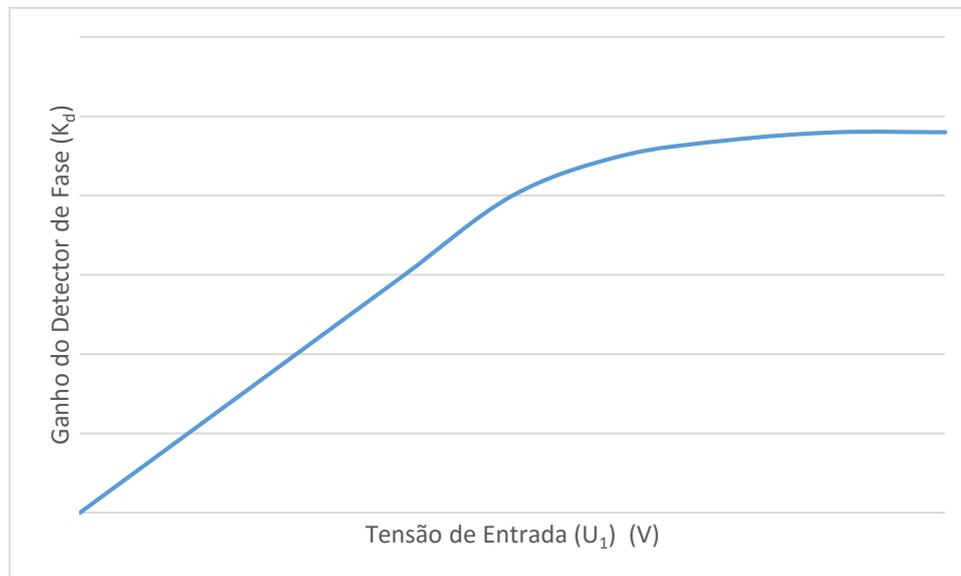


Figura 4 – Evolução do ganho K_d em função do valor eficaz da tensão de entrada (U_1).

- **Tipo OU exclusivo**

Este tipo de detetor de fase é maioritariamente utilizado na malha de captura de fase digital (*Digital Phase Locked Loop – DPLL*).

O seu modo de funcionamento só permite sinais digitais (binários) e o seu tempo no estado lógico 1 tem de ser igual ao tempo no estado lógico 0, ou seja, o *duty cycle* tem de ser 50%, para que este PLL funcione corretamente.

O diagrama deste tipo de PLL não é dado por uma ou um conjunto de equações, mas sim pelo seu próprio símbolo lógico, representado na Figura 5.

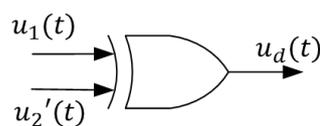


Figura 5 – Símbolo digital do OU exclusivo.

Não existe uma equação para descrever o comportamento deste tipo de PLL, no entanto existe uma tabela com os estados lógicos de entradas e respetiva saída.

| $u_1(t)$ | $u_2'(t)$ | $u_d(t)$ |
|----------|-----------|----------|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Tabela 1 – Tabela de estados lógicos de entradas e saídas num detetor de fase do tipo OU exclusivo.

Para este detetor de fase, segundo [5], a tensão de saída do detetor de fase (u_d) é dada pela equação (2).

$$u_d = K_d \theta_e, \text{ com } K_d = \frac{U_B}{\pi} = \frac{U_{sat+} - U_{sat-}}{\pi} \quad (2)$$

Onde θ_e representa ângulo de saída do detetor de fase, U_B a diferença de potencial da saída do detetor de fase e U_{sat+} e U_{sat-} a tensão máxima e mínima de saída do detetor de fase.

É comum neste tipo de sistemas o sinal lógico “0” ser considerado massa e o sinal lógico “1” a tensão de alimentação do circuito (V_{CC}), logo a equação simplifica-se para a equação (3).

$$K_d = \frac{V_{CC}}{\pi} \quad (3)$$

Através da montagem com o PLL do tipo OU exclusivo é possível obter uma tensão de saída com 90° de atraso em relação à onda de origem.

Para certas aplicações este atraso não é um problema, como por exemplo os multiplicadores ou divisores de frequência, mas para a aplicação de sincronismo com a rede elétrica pode não ser o único resultado pretendido, pois normalmente se requer as componentes direta e em quadratura na determinação das componentes de um sinal.

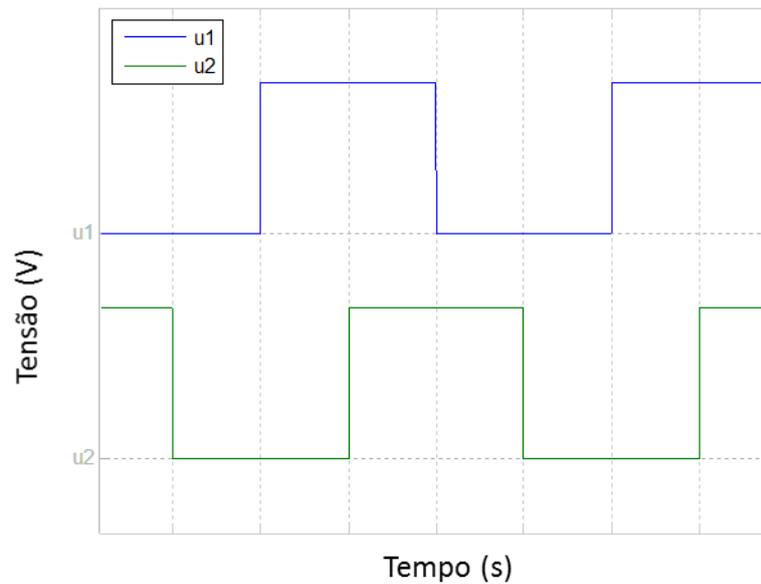


Figura 6 – Gráfico da tensão de entrada e saída do PLL OU exclusivo.

Realizando uma simulação para vários ângulos de defasagem entre a onda de entrada e de saída obtém-se o gráfico seguinte que nos dá a tensão média de saída do detetor de fase (U_d) em função do desfasamento existente entre o ângulo de entrada e de saída do PLL (θ_e).

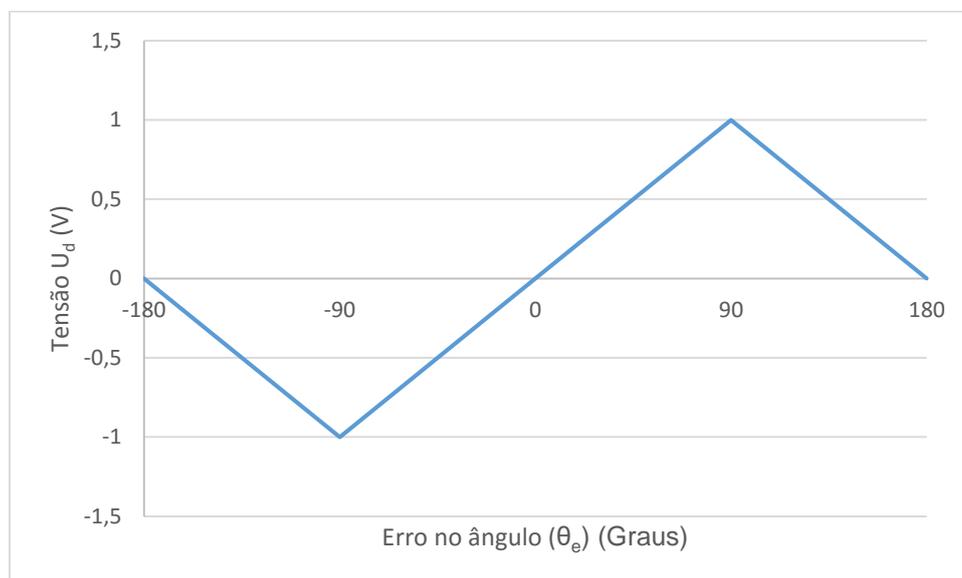


Figura 7 – Evolução da tensão U_d com a variação do erro no ângulo.

É de salientar que o gráfico da figura anterior corresponde não ao valor instantâneo da saída do detetor de fase mas sim ao valor médio desta tensão, sendo o sinal de saída ao longo do tempo um sinal digital (onda quadrada).

• **Tipo Flip-flop JK**

Este tipo de PLL também tem o inconveniente do PLL anterior sendo o seu funcionamento apenas para sinais digitais, no entanto quando este PLL está dessincronizado da entrada, comporta-se melhor que o PLL de OU exclusivo ou até mesmo que o multiplicador, segundo [5].

O seu símbolo é ilustrado na figura abaixo, não sendo mais que o símbolo do flip-flop conhecido e a sua tabela de verdade é representada na Tabela 2.

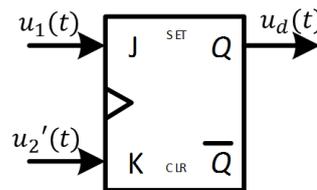


Figura 8 – Símbolo de um flip-flop JK.

Tal como anteriormente seguidamente é apresentada uma tabela com os estados lógicos de entradas e respetiva saída.

| J | K | Q |
|---|---|-----------------|
| 0 | 0 | Mantém o estado |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | Muda o estado |

Tabela 2 – Tabela com estados lógicos do flip-flop J-K.

A tensão de saída do detetor de fase (u_d) é dada pela equação abaixo, segundo [5].

$$u_d = K_d \theta_e, \text{ com } K_d = \frac{U_B}{2\pi} = \frac{U_{sat+} - U_{sat-}}{2\pi} \quad (4)$$

Pela mesma razão apresentada no detetor de fase Tipo OU exclusivo, o parâmetro K_d pode ser simplificado sendo o resultado mostrado na equação (5).

$$K_d = \frac{V_{cc}}{\pi} \quad (5)$$

Para o PLL do flip-flop JK as ondas de entrada e de saída estão desfasadas 180° , ou seja, encontram-se em oposição de fase, tal como é mostrado na Figura 9.

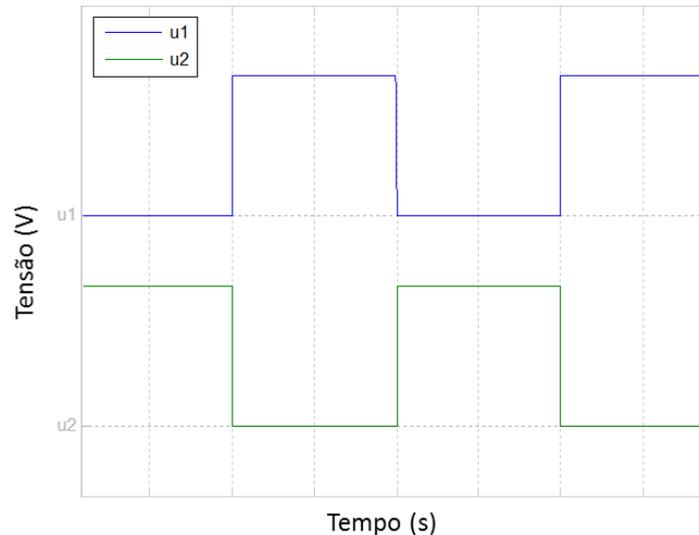


Figura 9 – Gráfico da tensão de entrada e saída do PLL flip-flop JK.

Da mesma forma que acima, é possível obter o valor da tensão média de saída do detetor de fase (U_d) com o crescimento do erro do ângulo (θ_e), sendo o gráfico resultante mostrado na Figura 10.

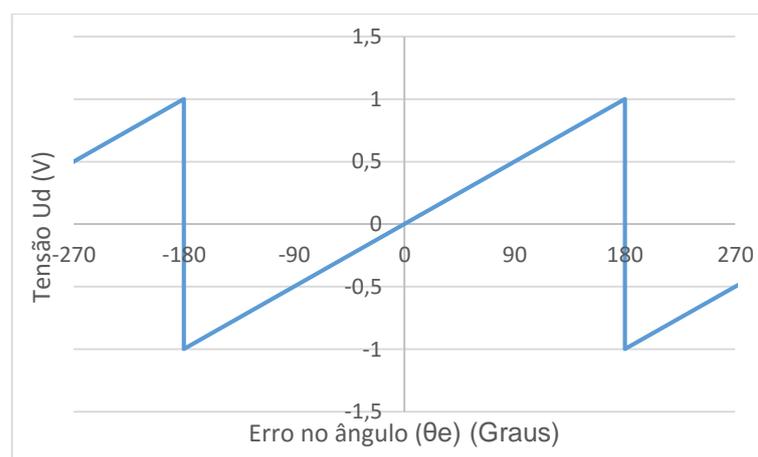


Figura 10 – Evolução da tensão U_d com a variação do erro no ângulo.

Na Figura 10 é mostrado o valor médio da tensão de saída do detetor de fase em função da variação da diferença entre os ângulos de entrada e saída, ou seja, o desvio de fase entre estas ondas.

- **Tipo detetor de fase e frequência**

Este detetor de fase ao contrário dos outros além de sincronizar as frequências sincroniza também a fase.

É o detetor de fase mais utilizado e pode ser utilizado com dois tipos de saídas, em corrente e em tensão.

A grande desvantagem deste detetor de fase é ter pouca imunidade ao ruído [5].

Seguidamente serão mostrados os circuitos normalmente utilizados para este tipo de detetores de fase com saída em tensão, Figura 11, e com saída em corrente, Figura 13.

- **Detetor de fase e frequência com saída em tensão**

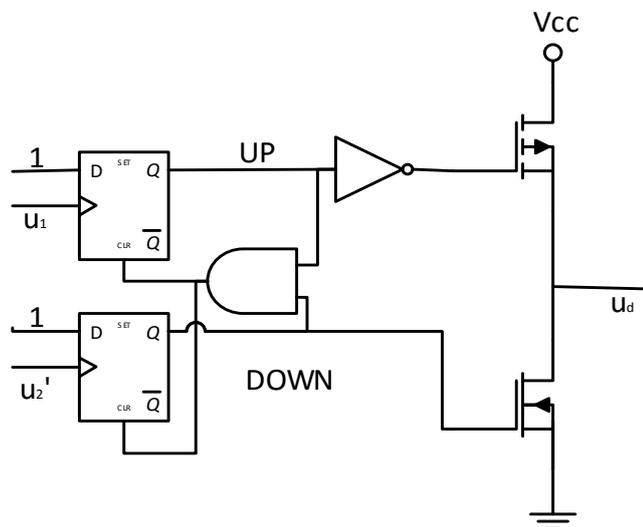


Figura 11 – Circuito de um PFD com saída em tensão.

Sabendo que este detetor de fase possui três estados, é mostrado abaixo um diagrama indicativo de quando existem estas mudanças de estado.

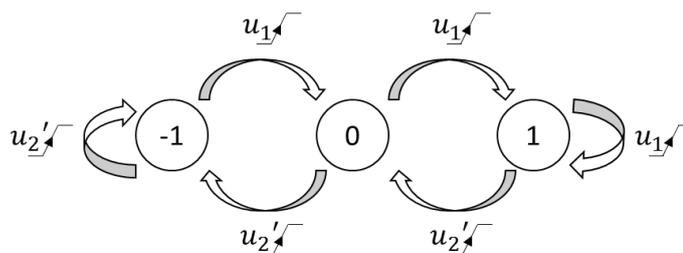


Figura 12 – Diagrama representativo dos três níveis do PFD.

A tensão u_d segundo [5] é dada pela equação (6) sendo a simplificação do K_d feita na equação (7).

$$u_d = K_d \theta_e, \text{ com } K_d = \frac{U_B}{4\pi} = \frac{U_{sat+} - U_{sat-}}{4\pi} \quad (6)$$

$$K_d = \frac{V_{CC}}{4\pi} \quad (7)$$

○ **Detetor de fase e frequência com saída em corrente**

A principal vantagem deste detetor de fase é a facilidade de utilização de um filtro RC passivo, que coloca um polo na origem da cadeia aberta do sistema quando a saída é de corrente, aumentando assim o tipo do sistema.

Tal como no funcionamento em tensão existem 3 estados para o funcionamento deste detetor de fase sendo o estado intermédio a ausência de corrente e os outros estados com corrente a circular no sentido direto ou inverso.

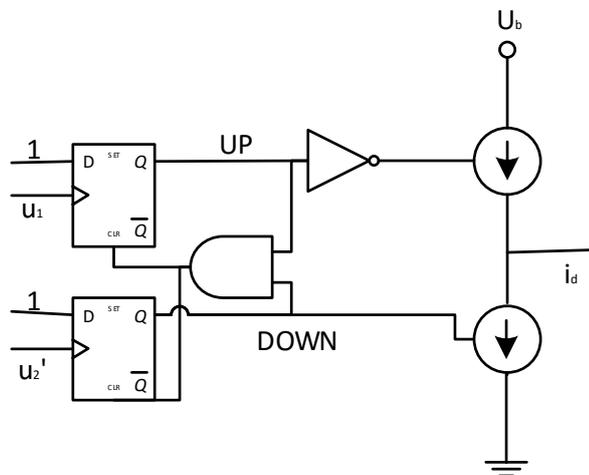


Figura 13 – Circuito de um PFD com saída em corrente.

A corrente i_d é dada pela equação (8), segundo [5].

$$i_d = K_d \theta_e, \text{ com } K_d = \frac{I_B}{2\pi} \quad (8)$$

Como é possível observar na Figura 14, a tensão de entrada e saída não possuem defasamento. Isto é outra das vantagens deste PLL quando o objetivo é sincronizar a onda de entrada e saída.

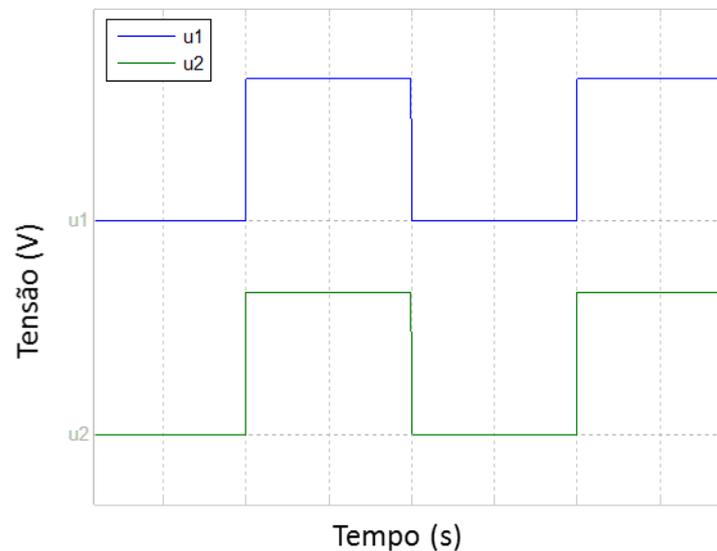


Figura 14 – Gráfico da tensão de entrada e saída do PFD.

Simulando o comportamento do PLL, tal como já foi feito anteriormente, pode verificar-se que o gráfico seguinte que representa o valor médio da tensão de saída do detetor de fase em função do erro entre o ângulo de entrada e de retroação do PLL.

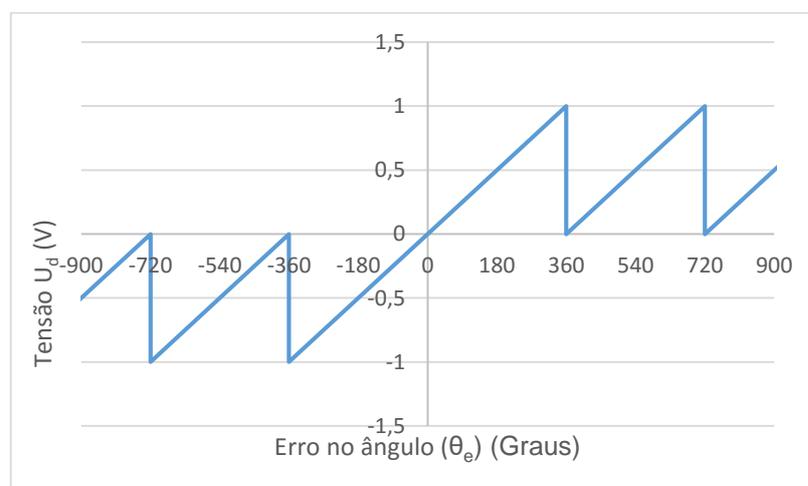


Figura 15 – Evolução da tensão U_d com a variação do erro no ângulo.

- **Resumo dos detetores de fase**

Como resumo dos diferentes detetores de fase é apresentada a tabela abaixo com algumas informações mais relevantes.

| Detetor de Fase | Tensão de Entrada | Ganho | Desfasagem entre a entrada e saída | Observações |
|----------------------|-------------------|-----------------------------|------------------------------------|---|
| Multiplicador | Análogica | Depende da forma de onda | -90 | Ganho depende da saturação Aparecimento de uma 2ª harmónica na saída deste detetor |
| OU Exclusivo | Digital | $K_d = \frac{V_{cc}}{\pi}$ | +90 | Duty cycle tem de ser 50% |
| Flip-flop J-K | Digital | $K_d = \frac{V_{cc}}{\pi}$ | 180 | Sincronização melhor que os anteriores |
| PFD Tensão | Digital | $K_d = \frac{V_{cc}}{4\pi}$ | 0 | Pouca imunidade a ruído e o seu funcionamento é equivalente a um polo na origem |
| PFD Corrente | | $K_d = \frac{I_B}{2\pi}$ | | Fácil utilização de filtro passa baixo, pouca imunidade a ruído e o seu funcionamento é equivalente a um polo na origem |

Tabela 3 – Tabela com resumo dos vários detetores de fase.

2.3 – Filtros

Neste subcapítulo serão estudados vários tipos de filtros. Apesar de alguns não serem aplicados no presente trabalho foram descritos para que fosse possível a sua implementação mediante outros requisitos de dimensionamento. Os circuitos analógicos ilustrados não serão implementados visto o intuito da dissertação ser a implementação digital, desta forma posteriormente é realizada a discretização do filtro utilizado.

Segundo [5], os filtros normalmente utilizados para aplicação no PLL, são os seguintes:

- **Montagem passiva de compensação de avanço-atraso (Passive Lead-lag)**

Existem dois modos de funcionamento destes filtros, com entrada em corrente e com entrada em tensão.

O filtro de entrada em tensão é o apresentado abaixo, onde entra a tensão de saída do detetor de fase (u_d) e na saída do filtro tem-se a tensão (u_f).

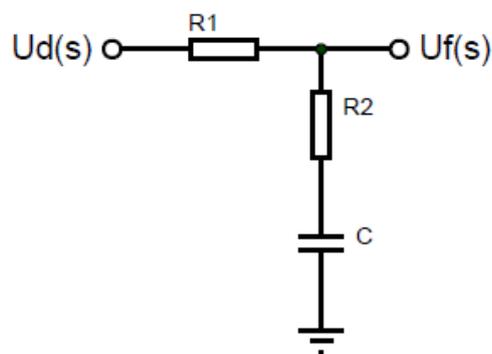


Figura 16 – Circuito de um filtro leadlag passivo (entrada em tensão).

Retirando a função de transferência deste filtro e simplificando obtém-se a seguinte expressão.

$$F(s) = \frac{U_f(s)}{U_d(s)} = \frac{1 + s \tau_2}{1 + s (\tau_1 + \tau_2)}, \text{ com } \tau_1 = R_1 C \text{ e } \tau_2 = R_2 C \quad (9)$$

Note-se que o parâmetro R_1 e R_2 são os valores das resistências em Ohm e o parâmetro C o valor da capacitância do condensador em Faraday.

Utilizando o Matlab obteve-se o diagrama de bode da função transferência do filtro com entrada em tensão.

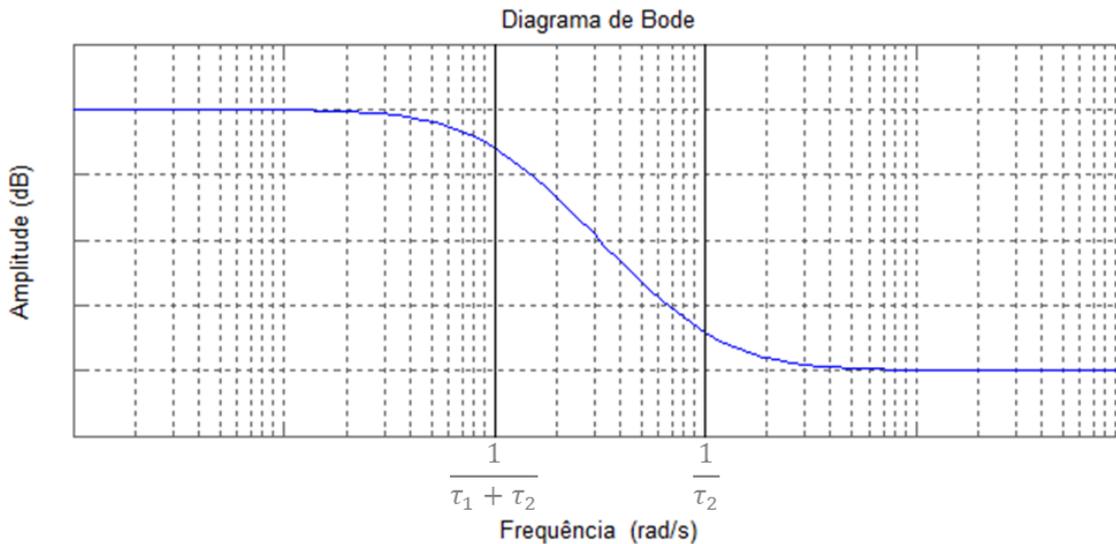


Figura 17 – Resposta em frequência de um filtro leadlag passivo (entrada em tensão).

Para a entrada de corrente não é necessária a resistência R_1 devido a esta funcionar como uma transformação de corrente para tensão.

No circuito com entrada em corrente o sinal de entrada é a corrente de saída do detetor de fase (i_d), como saída mantém-se a tensão do filtro (u_f).

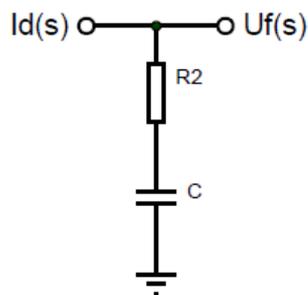


Figura 18 – Circuito de um filtro leadlag passivo (entrada em corrente).

Como não existe a resistência tal como já foi mencionado, a função transferência simplifica-se apresentando um polo na origem que será uma vantagem devido ao aumento do tipo do sistema.

$$F(s) = \frac{U_f(s)}{I_d(s)} = \frac{1 + s \tau_2}{C s}, \text{ com } \tau_2 = R_2 C \quad (10)$$

Recorrendo ao Matlab obteve-se o diagrama de bode correspondente à função transferência do filtro com entrada em corrente

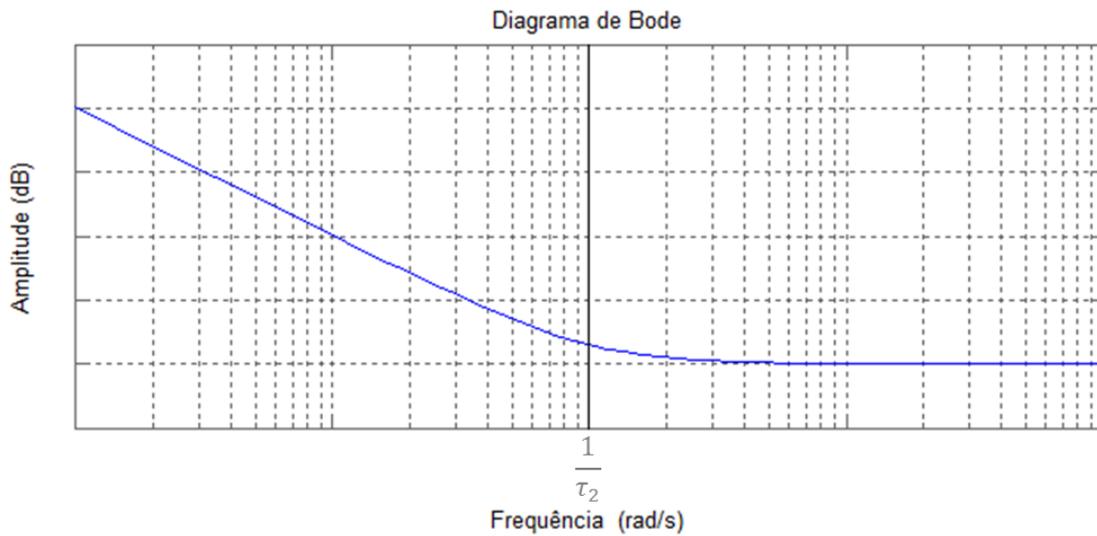


Figura 19 – Resposta em frequência de um filtro leadlag passivo (entrada em corrente).

- **Montagem ativa de compensação de avanço-atraso (Active Lead-lag)**

Tal como no ponto acima foram estudados os mesmos dois tipos de entradas para os filtros, começando-se pelo filtro com entrada em tensão representado pelo diagrama da Figura 20.

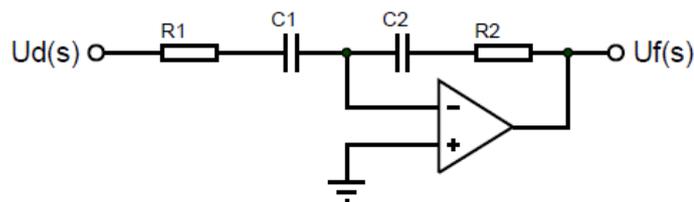


Figura 20 – Circuito de um filtro leadlag ativo (entrada em tensão).

A função de transferência destes filtros é dada pela equação seguinte:

$$F(s) = \frac{U_f(s)}{U_d(s)} = -K_a \cdot \frac{1 + s \tau_2}{1 + s \tau_1}, \text{ com } K_a = \frac{C_1}{C_2}, \tau_1 = R_1 C_1 \text{ e } \tau_2 = R_2 C_2 \quad (11)$$

Tal como acima obteve-se o diagrama de bode para esta função transferência, sendo o mesmo apresentado na Figura 21.

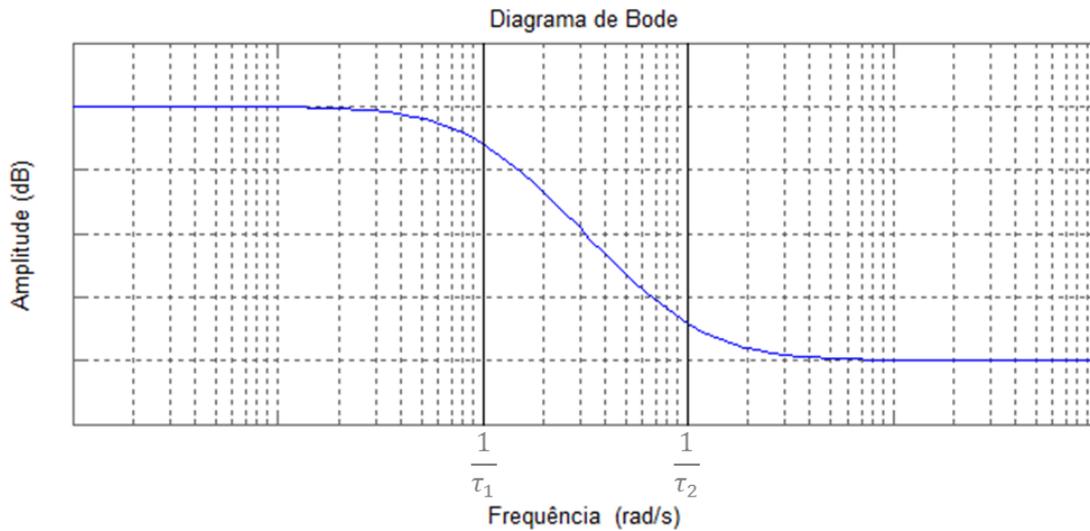


Figura 21 – Resposta em frequência de um filtro leadlag ativo.

O circuito do filtro com entrada de corrente é representado pelo diagrama da Figura 22. Sendo a sua função transferência apresentada logo de seguida pela equação (12).

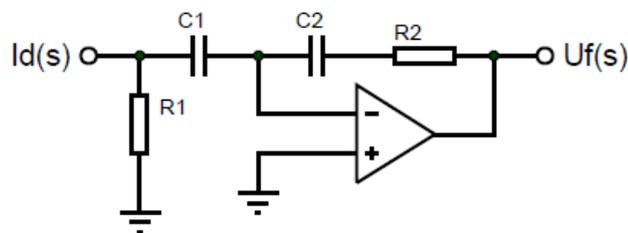


Figura 22 – Circuito de um filtro leadlag ativo (entrada em corrente).

$$F(s) = \frac{U_f(s)}{I_d(s)} = -K_a \cdot \frac{1 + s \tau_2}{1 + s \tau_1}, \text{ com } K_a = \frac{C_1}{C_2}, \tau_1 = R_1 C_1 \text{ e } \tau_2 = R_2 C_2 \quad (12)$$

Como as funções de transferência do filtro com entrada de tensão e de corrente são iguais, o diagrama de bode correspondente é igual sendo mostrado na Figura 21.

- **Montagem ativa de compensação proporcional integral de 1.^a Ordem (Active PI)**

Este tipo de filtros muitas vezes não é utilizado nesta função mas sim como um compensador para cancelar polos indesejados em sistemas, contudo tem a possibilidade de servir nesta situação como filtro.

Para a entrada em tensão temos então o diagrama que é apresentado na Figura 23.

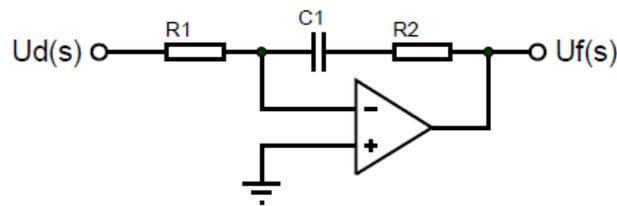


Figura 23 – Circuito de um filtro PI ativo (entrada em tensão).

Retirando a função transferência (equação (13)) obteve-se uma função idêntica à do filtro Leadlag passivo com entrada de corrente.

$$F(s) = \frac{U_f(s)}{U_d(s)} = -\frac{1 + s \tau_2}{\tau_1 s}, \text{ com } \tau_1 = R_1 C_1 \text{ e } \tau_2 = R_2 C_1 \quad (13)$$

Através desta expressão sabe-se que o diagrama de bode é igual ao da Figura 19.

Passando à entrada em corrente e analisando o circuito dado pela Figura 24 verifica-se que o circuito é aproximadamente o mesmo.

A função transferência é então dada pela equação (14), que mostra apenas uma diferença no ganho do filtro, sendo o diagrama de bode mostrado também pela Figura 19.

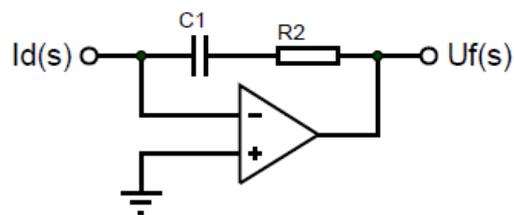


Figura 24 – Circuito de um filtro PI ativo (entrada em corrente).

$$F(s) = \frac{U_f(s)}{I_d(s)} = -\frac{1 + s \tau_2}{C_1 s}, \text{ com } \tau_2 = R_2 C_1 \quad (14)$$

- **Resumo dos filtros**

A Tabela imediatamente a seguir representa um resumo dos filtros existentes, as suas vantagens e desvantagens e a sua função transferência.

| Filtro | Entrada | Função transferência | Vantagens/Desvantagens |
|-------------------------|----------|--|--|
| Passive Lead-lag | Tensão | $\frac{U_f(s)}{U_d(s)} = \frac{1 + s \tau_2}{1 + s (\tau_1 + \tau_2)}$ | Montagem simples. |
| | Corrente | $\frac{U_f(s)}{I_d(s)} = \frac{1 + s \tau_2}{C s}$ | Montagem simples. Possui um polo em zero que aumento o tipo de sistema. |
| Active Lead-lag | Tensão | $\frac{U_f(s)}{U_d(s)} = -K_a \cdot \frac{1 + s \tau_2}{1 + s \tau_1}$ | Requer um Ampop. |
| | Corrente | $\frac{U_f(s)}{I_d(s)} = -K_a \cdot \frac{1 + s \tau_2}{1 + s \tau_1}$ | Requer um Ampop. |
| Active PI | Tensão | $\frac{U_f(s)}{U_d(s)} = -\frac{1 + s \tau_2}{\tau_1 s}$ | Possui um polo em zero que aumento o tipo de sistema. Requer um Ampop. |
| | Corrente | $\frac{U_f(s)}{I_d(s)} = -\frac{1 + s \tau_2}{C_1 s}$ | Possui um polo em zero que aumento o tipo de sistema. Requer um Ampop. |

Tabela 4 – Tabela com resumo dos filtros normalmente utilizados com o PLL.

2.4 – Osciladores controlados por tensão (VCO)

De seguida é feita uma introdução ao VCO utilizado nas implementações analógicas. Tal como no subcapítulo anterior estes componentes não serão utilizados na implementação discreta. A discretização deste componente é realizada no capítulo 4.2 – Discretização.

Um oscilador controlado por tensão é um circuito que permite a geração de uma forma de onda com frequência proporcional à tensão de entrada que lhe é aplicada.

Existem 2 tipos de osciladores controlados por tensão sendo eles o de relaxamento e o de ressonância.

- **Oscilador de relaxamento**

Este tipo de osciladores são muito utilizados nos PLL digitais tais como os PLL existentes nos circuito integrados 74HC/HCT4046 [6].

Funcionam através de comparadores histeréticos e possuem um circuito através de portas OR ligadas a semicondutores para a comutação do sentido de carga do condensador.

A carga do condensador pode ser feita através de uma conversão tensão/corrente que carregará mais rápido o condensador quanto maior for a corrente enviada.

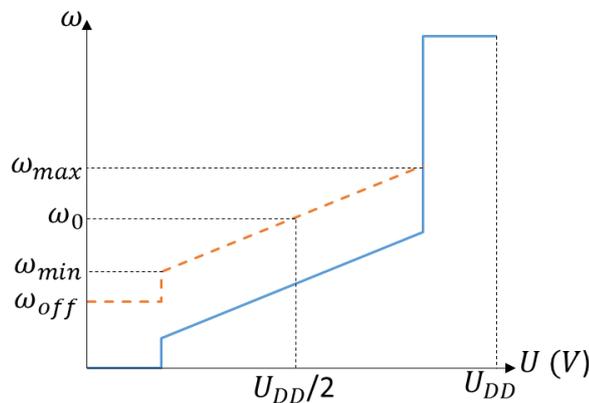


Figura 25 – Gráfico da evolução da frequência de saída do VCO em função da tensão de entrada.

Deste gráfico é de salientar que a frequência designada por ω_{off} é a frequência colocada na saída do VCO no caso de não existir tensão aplicada à entrada do mesmo e as frequências ω_0 , ω_{max} e ω_{min} a frequência angular média, máxima e mínima do VCO respetivamente.

Pode ainda definir-se a frequência angular de saída do VCO (ω_2) como função da tensão de saída do filtro (u_f), da frequência média do VCO (ω_0), da tensão de alimentação (V_{cc}) e do ganho do oscilador (K_0), sendo esta relação dada pela equação (15).

$$\omega_2 = \omega_0 + K_0 \cdot \left(u_f - \frac{V_{cc}}{2} \right) \quad (15)$$

O circuito que se segue representa o esquema elétrico de um VCO de relaxamento.

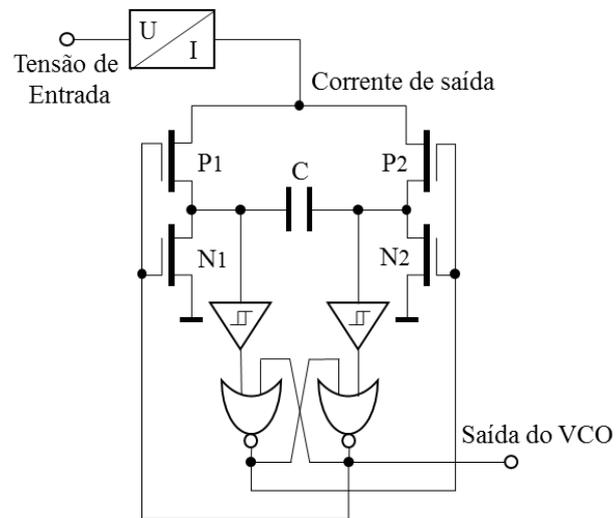


Figura 26 – Circuito de um VCO de relaxamento utilizado no circuito integrado 74HC/HCT4046.

- **Oscilador de ressonância**

Este tipo de oscilador é utilizado maioritariamente para elevadas frequências, mais concretamente, acima de 50 MHz.

Estes osciladores normalmente possuem um diodo varicap que possui uma capacitância variável com o valor de tensão inversa à qual é submetido.

Através disto e de malhas de bobinas e condensadores é possível gerar um oscilador controlado.

2.5 – Divisor de frequência

Os divisores de frequência são utilizados nestes circuitos para se conseguir frequências mais baixas, pois os VCO normalmente funcionam a frequências altas. Tal como o nome indica são dispositivos que dividem a frequência ou por outro lado multiplicam o período.

Um exemplo simples de um divisor de frequência é um contador digital.

A função de transferência associada a este tipo de divisor é:

$$FT = \frac{1}{N} \quad (16)$$

Onde N corresponde ao fator de divisão do divisor de frequências.

Este componente no caso de sincronização de ondas pode ser omissa na implementação discreta. Tal deve-se à introdução do ganho do divisor de frequência na fórmula de cálculo do ganho do VCO, poupando assim algum processamento que significará um aumento da frequência de aquisição.

Capítulo III: Dimensionamento e Modelização do PLL

Resumo:

Este capítulo inicia-se com a descrição dos requisitos que o PLL deve cumprir para a sincronização com a rede elétrica.

De seguida é feita uma seleção de qual o detetor de fase e filtro que deve ser utilizado tendo em conta os requisitos descritos e por fim é realizada uma análise das frequências limite do sistema.

3.1 – Introdução

Após ser descrito quais os blocos constituintes do PLL irá dimensionar-se um PLL da forma mais vantajosa para a situação abaixo descrita.

Pretende-se que seja um PLL para sincronização com a rede elétrica Europeia, ou seja:

- A tensão de entrada tem um valor eficaz de 230 V, sendo a sua forma sinusoidal com uma frequência de 50 Hz.
- Segundo a norma EN 50160 o intervalo de frequência para a qual a rede elétrica tem de estar 100% do tempo é [47,52] Hz, no entanto irá, neste trabalho, alargar-se o intervalo para frequências entre 40 e 60 Hz.
- Devido à existência de perturbações na rede elétrica é previsível a existência de um conteúdo harmónico na tensão cujos limites estão descritos na norma EN 50160 e representados na tabela abaixo.

| Harmónicas Impares | | | | Harmónicas pares | |
|--------------------|---------------------------|----------------|---------------------------|------------------|---------------------------|
| Não múltiplas de 3 | | Múltiplas de 3 | | | |
| Ordem n | Tensão Relativa (%) | Ordem n | Tensão Relativa (%) | Ordem n | Tensão Relativa (%) |
| 5 | 6,0 | 3 | 5,0 | 2 | 2,0 |
| 7 | 5,0 | 9 | 1,5 | 4 | 1,0 |
| 11 | 3,5 | 15 | 0,5 | 6 a 24 | 0,5 |
| 13 | 3,0 | 21 | 0,5 | - | - |
| 17 | 2,0 | - | - | - | - |
| 19 | 1,5 | - | - | - | - |
| 23 | 1,5 | - | - | - | - |
| 25 | - | - | - | - | - |

Tabela 5 – Tabela com valores harmónicos de tensão permitidos pela normalização.

- A normalização limita também a distorção harmónica total a 8%.

Além do descrito acima, o PLL vai ser dimensionado para que a sua sincronização demore aproximadamente 10 ciclos da rede, de forma a o tornar imune a algumas perturbações na rede elétrica tais como cavas de tensão ou variações impulsãoais de frequência ou tensão.

3.2 – Escolha do detetor de fase

Tendo em conta a utilização do PLL, conexão com a rede elétrica, fará mais sentido utilizar um PLL com um detetor de fase que permita entradas analógicas. Para responder a esta característica foi pensado utilizar-se o detetor de fase do tipo multiplicador pois o sinal de entrada é analógico.

Apesar de não cumprir este requisito, o detetor de fase e frequência (*Phase Frequency Detector – PFD*) seria útil no sentido em que as ondas de entrada e saída estão perfeitamente sincronizadas enquanto neste PLL estas duas ondas se encontram desfasadas de 90° , sendo que para obter a onda na sua forma correta teríamos de gerar duas ondas, uma para a retroação e outra para a saída do PLL.

3.3 – Escolha do filtro

O tipo de filtro escolhido é o PI ativo visto que coloca mais um polo na origem permitindo assim aumentar o tipo do sistema. Este aumento permite ao sistema sincronizar-se com variações no ângulo do tipo escalão e rampa devido à existência de dois polos na origem (um do PI e um do VCO). Note-se que uma variação no ângulo do tipo rampa, corresponde a uma variação do tipo escalão na frequência.

3.4 – Estudo do sistema através da análise de funções transferência

De seguida irá proceder-se ao estudo do sistema de forma a obter a melhor resposta para os requisitos impostos.

Para tal, irá proceder-se à criação do diagrama de blocos, com as respetivas funções transferência, correspondentes aos componentes que foram selecionados com base na aplicação do mesmo. Sendo as funções transferência utilizadas descritas no capítulo anterior.

• Diagrama de blocos do sistema

O diagrama de blocos resultante para a análise do comportamento do sistema é dado pela Figura 27.

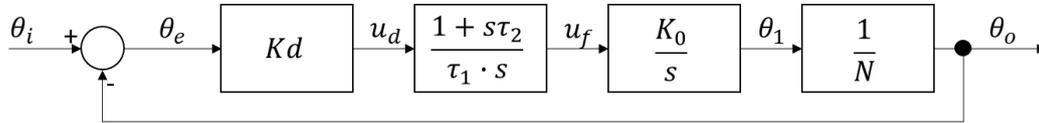


Figura 27 – Diagrama de blocos com funções transferência de um PLL.

• Funções transferência do sistema

A partir do diagrama de blocos e multiplicando as funções transferência dos componentes do PLL, obtém-se a função transferência de cadeia aberta, sendo a mesma dada pela equação (17).

$$G(s) = K_d \frac{1 + s \tau_2}{\tau_1 s} \frac{K_0}{s} \frac{1}{N} = \frac{\frac{K_d K_0 \tau_2}{N \tau_1} s + \frac{K_d K_0}{N \tau_1}}{s^2} \quad (17)$$

Também através do diagrama de blocos e considerando a retroação é possível obter a função transferência em cadeia fechada a partir da equação (17), obtendo-se a equação (18).

$$\frac{\theta_o(s)}{\theta_i(s)} = \frac{G(s)}{1 + G(s)} = \frac{\frac{K_d K_0 \tau_2}{N \tau_1} s + \frac{K_d K_0}{N \tau_1}}{s^2 + \frac{K_d K_0 \tau_2}{N \tau_1} s + \frac{K_d K_0}{N \tau_1}} \quad (18)$$

Fazendo: $\frac{K_d K_0 \tau_2}{N \tau_1} = 2 \zeta \omega_n$ e $\frac{K_d K_0}{N \tau_1} = \omega_n^2$ obtém-se a equação (19), onde o denominador possui a equação característica de 2.^a ordem do controlo de sistemas.

$$\frac{2 \zeta \omega_n s + \omega_n^2}{s^2 + 2 \zeta \omega_n s + (\omega_n)^2} \quad (19)$$

Nesta equação o coeficiente de amortecimento é representado pela variável ζ e a frequência natural do sistema por ω_n .

○ Análise do sistema para diferentes amortecimentos

Para sistemas de 2.^a ordem típicos sem zeros na equação da cadeia fechada o sistema estabiliza mais rápido para $\zeta = \sqrt{2}/2$, no entanto, como este sistema possui um zero, testou-se

várias respostas em Matlab para verificar se esta condição se mantém, mostrando o resultado na Figura 28.

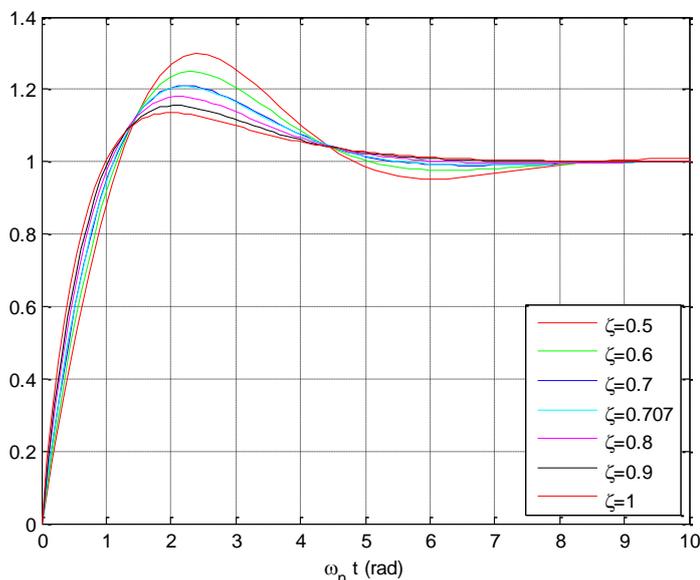


Figura 28 – Gráfico da evolução do sistema ao longo do tempo e frequência natural para diferentes valores de ζ .

O critério de escolha para este fator foi a estabilização mais rápida num intervalo de 1% em relação ao escalão aplicado, tendo-se concluído que a resposta mais rápida continua a ser para $\zeta = \sqrt{2}/2$, sendo esse o valor a utilizar.

Apesar de esta ser a melhor resposta, a sobrelevação é de cerca de 20%, o que é demasiado elevado para a sincronização de conversores à rede elétrica por geração de oscilações no trânsito de potência.

○ **Obtenção do parâmetro ω_n**

Para a obtenção deste parâmetro, efetuou-se o gráfico representado na Figura 28 não em função do tempo como é normal mas sim em função da frequência natural (ω_n) multiplicada pelo tempo, ou seja, $\omega_n t$.

A partir deste gráfico foi ainda possível, para o intervalo de erro 1% e um coeficiente de amortecimento ζ de 0,707, obter-se o valor de $\omega_n t$, sendo o mesmo 5,210 rad. Este valor foi obtido através de um ciclo de repetição elaborado no Matlab que permitiu a fácil obtenção deste parâmetro.

Igualando a variável t_{est} ao tempo de estabilização pretendido, 10 ciclos da rede, ou seja, 200 ms, pode obter-se o valor da frequência natural do sistema (ω_n), tal como mostram as equações (20), (21) e (22).

$$\omega_n t = 5,210 \text{ rad} \quad (20)$$

$$t_{est} = 0,2 \text{ s} \quad (21)$$

$$\omega_n = \frac{5,210}{0,2} = 26,052 \text{ rad/s} \quad (22)$$

Com os valores de ζ (0,707) e de ω_n (26,052 rad/s) calculados e recorrendo à instrução *step* do Matlab é possível obter a resposta do sistema a um escalão no ângulo de entrada, sendo o resultado mostrado na figura seguinte.

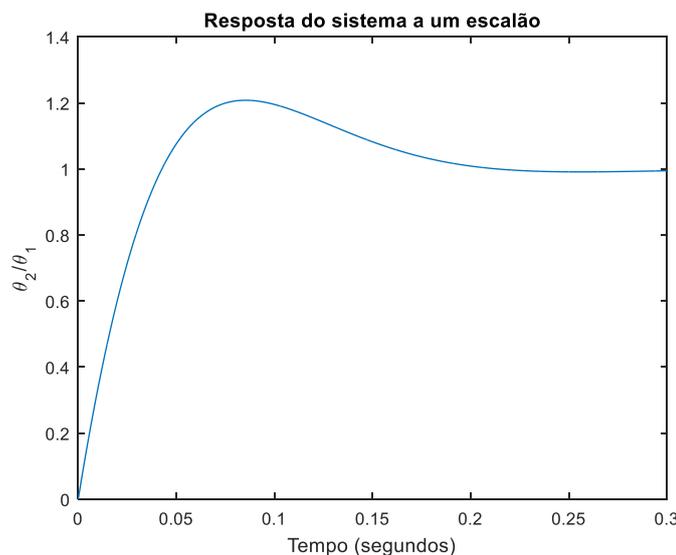


Figura 29 – Resposta temporal a uma variação do tipo escalão no ângulo de entrada.

Tendo sido pensado numa implementação digital do PLL, no Capítulo IV: *Software PLL* irá realizar-se o dimensionamento do SPLL baseado nos parâmetros ω_n e ζ , já calculados.

Seguidamente apresentam-se algumas frequências importantes ao funcionamento do PLL [5].

- **Cálculo dos parâmetros do PLL**

- **Frequência de Corte**

A frequência de corte representada (ω_{3dB}) resulta de um sistema de 2.^a Ordem com adição de um zero na função de transferência [5].

$$\omega_{3dB} = \omega_n \left[1 + 2\zeta^2 + \sqrt{(1 + 2\zeta^2)^2 + 1} \right]^{\frac{1}{2}} = 53,62 \text{ rad/s (8,53 Hz)} \quad (23)$$

A frequência calculada corresponde à frequência de corte do sistema de cadeia fechada.

Realizando a resposta do sistema de cadeia fechada, no domínio da frequência, para vários valores de ζ obteve-se a figura seguinte.

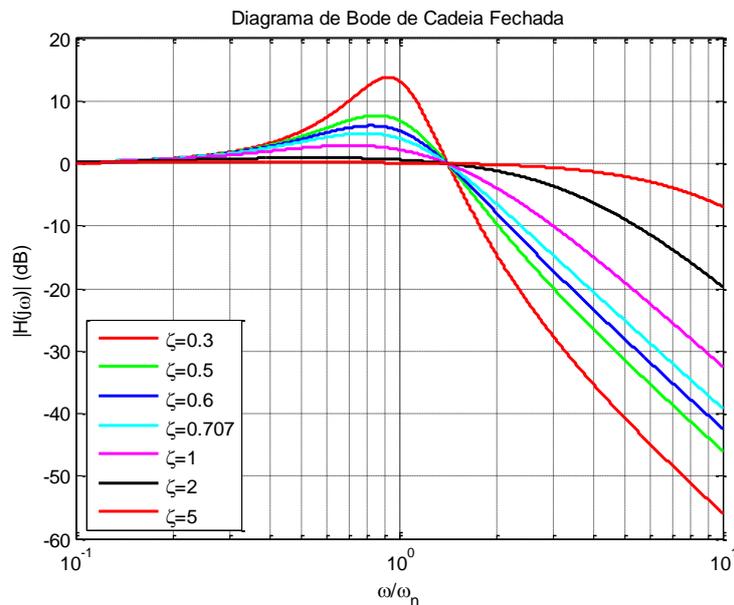


Figura 30 – Resposta em frequência da função de transferência em cadeia fechada do sistema simplificado.

Através da figura anterior e do uso do Matlab pode verificar-se que para $\zeta = 0,707$, pode obter-se a relação dada pela equação (24).

$$\omega_{3dB} = 2,06 \omega_n \quad (24)$$

- **Variação máxima da frequência**

Esta variação máxima de frequência ($\Delta\omega_{max}$) corresponde à maior variação repentina do ângulo θ que o PLL consegue acompanhar, ou seja, sincronizar-se.

$$\Delta\omega_{max} = \frac{\omega_n^2}{2} = 344,40 \text{ rad/s (54,81 Hz)} \quad (25)$$

○ **Limites de estabilidade estática e dinâmica**

A figura abaixo mostra as margens de funcionamento e estabilidade do PLL.

A nível dinâmico, a margem $\Delta\omega_L$ simboliza o intervalo de frequências para o qual o PLL está constantemente em sincronismo e a margem $\Delta\omega_{po}$ define um intervalo onde o PLL normalmente consegue sincronizar-se, no entanto deverá demorar um pouco.

Dentro do nível estático existe a margem $\Delta\omega_p$ que representa um intervalo de frequências onde o PLL pode sincronizar, mas irá demorar bastante mais tempo.

Por fim e ainda dentro do limite de estabilidade estática existe a margem $\Delta\omega_H$ que é o intervalo máximo onde o PLL pode funcionar.

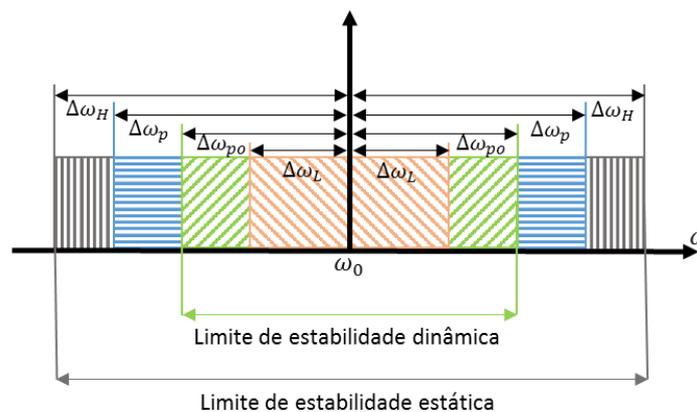


Figura 31 – Gráfico da estabilidade estática e dinâmica de um LPLL de 2ª Ordem.

○ **Margem de captura**

$$\Delta\omega_L = 2 \zeta \omega_n = 37,23 \text{ rad/s (5,93 Hz)} \quad (26)$$

○ **Margem de pull-out**

$$\Delta\omega_{po} = 1,8 \omega_n (\zeta + 1) = 80,75 \text{ rad/s (12,85 Hz)} \quad (27)$$

○ **Margem de manutenção**

$$\Delta\omega_H = \frac{K_0 K_d F(0)}{N} = \infty \quad (28)$$

○ **Margem de pull-in**

$$\Delta\omega_p = \infty \quad (29)$$

Capitulo IV: *Software* PLL

Resumo:

Neste capítulo será explicado o que é o SPLL, quais os tipos existentes e como é feito o projeto,

De seguida são mostradas os tipos de implementações realizadas bem como o seu dimensionamento.

Além disto, para o desenvolvimento do algoritmo procedeu-se à discretização de algumas das partes do SPLL, explicando também como deve ser feita esta discretização.

4.1 – Software PLL

O SPLL permite obter o maior número de graus de liberdade existentes no dimensionamento de um PLL, isto deve-se há possibilidade de variar todos os parâmetros.

Esta variação é bastante mais fácil devido aos parâmetros serem valores programados, ou seja, com a variação de alguns valores no código é possível variar o filtro ou até o ganho do DCO.

Segundo [5], é recomendado seguir-se o seguinte processo para implementar um SPLL.

1. Definição do algoritmo a utilizar;
2. Definição da linguagem de programação;
3. Estimar a largura de banda (em tempo real);
4. Testar a montagem.

• SPLL multiplicador

Neste modo de funcionamento, a tensão de entrada u_1 é um sinal analógico, por exemplo, uma onda sinusoidal. Para adquirir este sinal é necessário um conversor de analógico para digital (*Analog to Digital Converter - ADC*) e o seu tempo de amostragem é normalmente o tempo mínimo de amostragem do SPLL. Neste trabalho, como o tempo processamento dos dados era superior ao tempo de aquisição do conversor analógico para digital, o tempo de amostragem utilizado foi superior ao tempo de aquisição do ADC. Sabendo que a discretização necessita de um tempo de amostragem constante e que o tempo de cálculo do microcontrolador pode ser variável, para manter a frequência de amostragem constante utilizou-se um temporizador.

É de notar que todos os sinais estão sincronizados com o tempo de amostragem e que não é necessário utilizar um divisor de frequência.

O algoritmo utilizado para descrever esta implementação digital é o da figura abaixo.

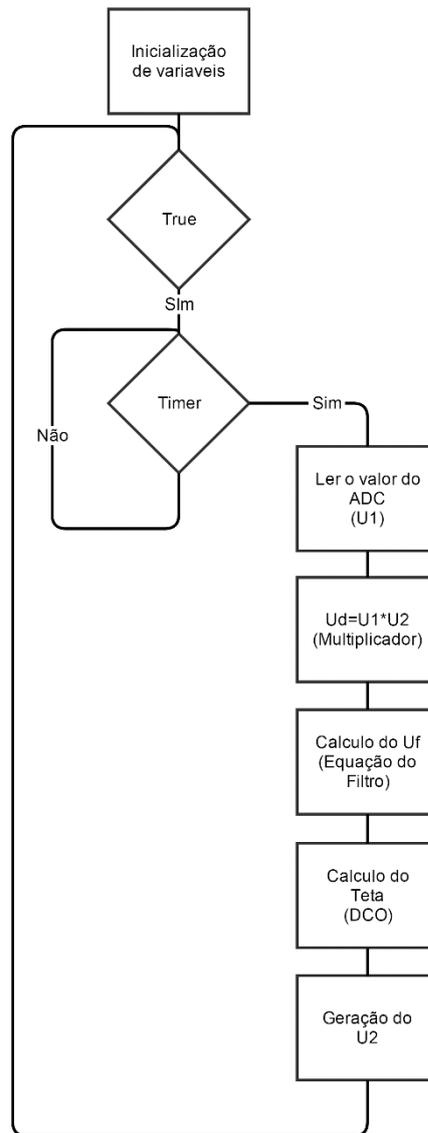


Figura 32 – Fluxograma representativo do algoritmo do PLL implementado com multiplicador.

Um dos inconvenientes da utilização de um detetor de fase do tipo multiplicador é a desfasagem do sinal de retroação em relação ao sinal de entrada (quadratura), desta forma quando utilizado, são gerados dois sinais, um para a realimentação e outro que será uma imagem da tensão de entrada depois de filtrada pelo SPLL.

Tal como nos PLL os SPLL podem ser de vários tipos. O exemplo seguinte dá uma pequena introdução a um SPLL detetor de fase e frequência.

- **SPLL detetor de fase e frequência**

Tal como no PLL implementado totalmente através de *hardware*, existem também outros tipos de SPLL que podem ser implementados.

No caso da implementação de um PFD os sinais de entrada e retroação possuem a inconveniência de serem digitais.

Um dos algoritmos utilizados na implementação deste detetor de fase vem representado em [5] e resulta de um conjunto de operações matemáticas e por vezes de operadores de decisão.

Toda a filosofia da implementação é idêntica à anterior com exceção do detetor de fase que tem 3 sinais e com isto vai alterar e complicar o algoritmo.

O algoritmo geral deste tipo de PLL é representado na Figura 33.

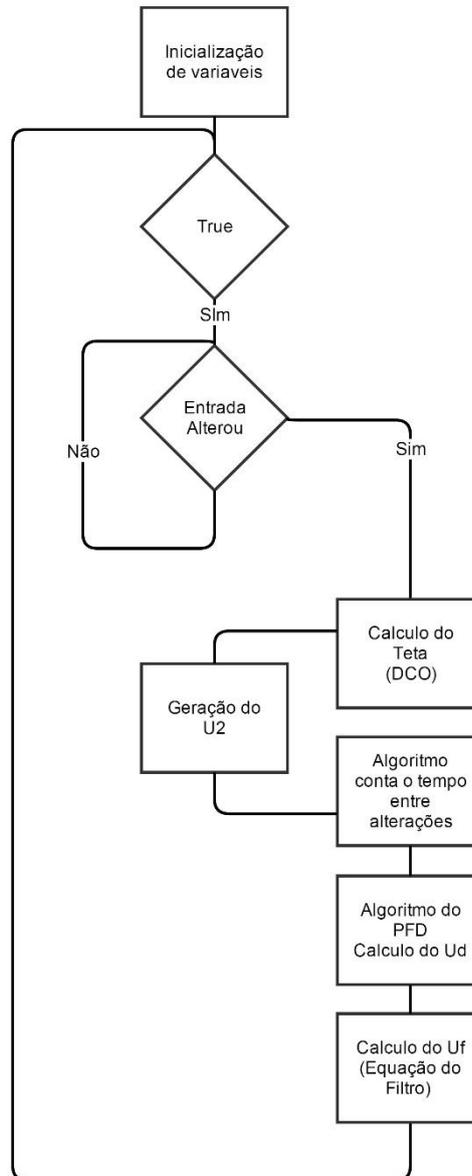


Figura 33 – Fluxograma representativo do algoritmo do PLL implementado com PFD.

Além do diagrama global representativo do algoritmo, existem ainda dentro de alguns blocos algoritmos para o cálculo de determinados parâmetros sendo o seu processamento por ciclo mais lento que o algoritmo do SPLL multiplicador.

Antes da implementação do software PLL é necessário obter-se as equações respeitantes à discretização dos blocos constituintes do mesmo, desta forma, de seguida é realizada esta mesma discretização.

4.2 – Discretização

Existem várias formas de realizar a discretização de um modelo no domínio de Laplace.

Os mais utilizados são obtidos através da substituição da variável s por uma função de z . Este método de conversão é chamado de integração numérica e pode ser feito de quatro formas.

- Integração retangular progressiva (*forward Euler method*)
- Integração retangular regressiva (*backward Euler method*)
- Integração trapezoidal ou bilinear (*trapezoidal Euler method/bilinear transformation*)
- Transformação bilinear com deformação da escala de frequências

Além destes métodos existem ainda outras formas de discretização evidenciadas em [7] mas não foram estudadas por não serem muito utilizadas.

O método mais utilizado e com melhores resultados segundo [7] é a integração trapezoidal e será este o método de discretização utilizado por preservar as características de amplitude e fase.

• Integração Trapezoidal

A diferença, nos resultados, desta para as outras integrações pode ser ilustrada na figura seguinte.

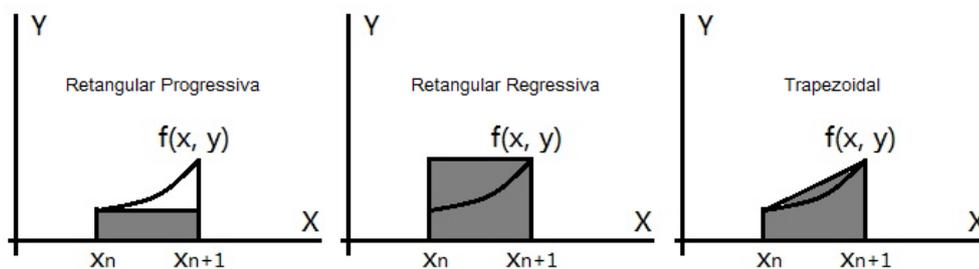


Figura 34 – Diagrama dos três tipos de integrações de Euler.

Para discretizar uma função de transferência tal como já foi dito acima vai substituir-se o s por uma função de z , neste caso a transformação é representada pela expressão abaixo, onde T_s representa o tempo de amostragem.

$$s \rightarrow \frac{2}{T_s} \frac{z - 1}{z + 1} \quad (30)$$

De seguida irá desenvolver-se as equações matemáticas do filtro e do VCO de forma a ser possível efetuar estes dois componentes de forma digital.

• **Discretização do filtro PI de 1.^a ordem**

Partindo da equação (13) e realizando a transformação $s \rightarrow \frac{2z-1}{T_s z+1}$ obtém-se a equação (31).

Depois de simplificar obtém-se a equação (32)

$$\frac{u_f(z)}{u_d(z)} = \frac{\tau_2 \left(\frac{2z-1}{T_s z+1} \right) + 1}{\tau_1 \left(\frac{2z-1}{T_s z+1} \right)} \tag{31}$$

$$\frac{u_f(z)}{u_d(z)} = \frac{\frac{(2\tau_2 + T_s)}{2\tau_1} z + \frac{(T_s - 2\tau_2)}{2\tau_1}}{z - 1} \tag{32}$$

Igualando à expressão mais utilizada a nível discreto fica-se com a equação (33):

$$\frac{u_f(z)}{u_d(z)} = \frac{b_0 + b_1 z^{-1}}{1 + a_1 z^{-1}} \tag{33}$$

Com:

$$a_1 = -1$$

$$b_0 = \frac{(2\tau_2 + T_s)}{2\tau_1}$$

$$b_1 = \frac{(T_s - 2\tau_2)}{2\tau_1}$$

Através da figura seguinte é possível validar a discretização feita para o filtro de 1^a Ordem.

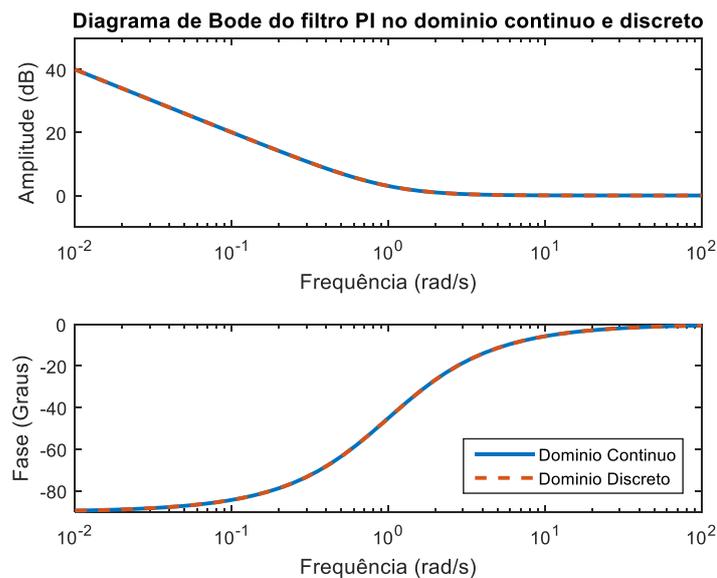


Figura 35 – Diagrama de Bode do filtro PI no domínio contínuo e discreto.

Como é possível observar as curvas são coincidentes o que confirma a discretização.

- **Discretização do DCO**

A discretização do DCO não foi feita recorrendo à substituição do s por uma função de z , mas sim recorrendo a cálculos realizados pelo VCO adaptados ao modo discreto.

$$\omega_2(t) = \omega_0 + K_0 u_f(t) \quad (34)$$

Logo, como o ângulo (θ_2) é a integração da frequência angular (ω_2), obtém-se:

$$\theta_2(t) = \int \omega_2(t) dt = \int \omega_0 dt + K_0 \int u_f(t) dt \quad (35)$$

Sabendo que n é um número inteiro que representa uma aquisição e supondo que entre a próxima aquisição ($n+1$) e a última aquisição realizada (n) a tensão na saída do filtro se mantém aproximadamente constante é possível realizando a integração mencionada na equação (35), verificar-se a seguinte equação.

$$\Delta\theta_2 = \theta_2(n+1) - \theta_2(n) = [\omega_0 + K_0 u_f(n)] T_s \quad (36)$$

Colocando o $\theta_2(n+1)$ em evidência, obtém-se a equação (37).

$$\theta_2(n+1) = \theta_2(n) + [\omega_0 + K_0 u_f(n)] T_s \quad (37)$$

Sendo neste momento possível implementar um PLL, em *software*, através das equações obtidas anteriormente, de seguida irá proceder-se à implementação dos SPLL descritos anteriormente.

4.3 – SPLL com tensão de realimentação quadrada

O diagrama de blocos correspondente à implementação é o mostrado na figura seguinte.

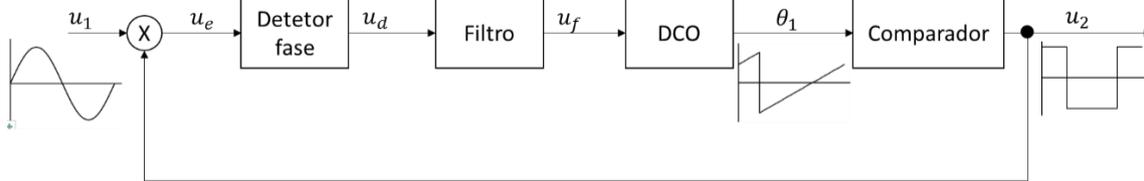


Figura 36 – Diagrama do PLL com tensão de retroação quadrada.

O intuito deste estudo é o seu código ser o mais rápido a ser executado. Desta forma é possível obter uma maior frequência de aquisição, no entanto pode ter alguns problemas mencionados em literatura e que vão ser estudados adiante.

- **Dimensionamento do Filtro PI**

De seguida será realizado o dimensionamento do filtro para a situação estudada, sendo necessários alguns parâmetros do sistema como o ganho do VCO e do detetor de fase calculados abaixo.

- **Obtenção do parâmetro K_d**

Seguidamente foi efetuado o cálculo do ganho do detetor de fase escolhido, para a situação onde a tensão de retroação é quadrada, equação (38), considerando que tanto a tensão de entrada como a tensão de saída estão normalizadas como sinais entre -1 V e 1 V.

$$K_d = \frac{2 U_{10} U_{20}}{\pi} = \frac{2 \times 1 \times 1}{\pi} = 0,637 \quad (38)$$

- **Obtenção do parâmetro K_0**

Como existe uma maior liberdade considerou-se um ganho do DCO (K_0) igual a 100. Não é utilizado um divisor de frequências, para que o microcontrolador possa realizar menos instruções diminuindo o tempo de execução de um ciclo.

$$K_0 = 100 \text{ rad/V} \quad (39)$$

$$N = 1 \quad (40)$$

○ **Obtenção do parâmetro τ_1 e τ_2**

Tendo por base os parâmetros calculados acima, igualando as equações (18) e (19) e colocando o parâmetro τ_1 e τ_2 em evidência, é possível obter estes dois parâmetros tal como mostram as equações (41) e (42). Desprezou-se o parâmetro N por ser unitário.

$$\tau_1 = \frac{K_p K_0}{\omega_n^2} = \frac{0,637 \times 100}{26,052^2} = 0,094 \text{ s} \quad (41)$$

$$\tau_2 = \frac{2 \zeta \omega_n \tau_1}{K_p \cdot K_v} = \frac{2 \times \frac{\sqrt{2}}{2} \times 26,052 \times 0,094}{0,637 \times 100} = 0,0543 \text{ s} \quad (42)$$

4.4 – SPLL com tensão de realimentação sinusoidal

Esta implementação é bastante idêntica à anterior mudando apenas o formato da onda de retroação, ficando o diagrama de blocos como o representado seguidamente.

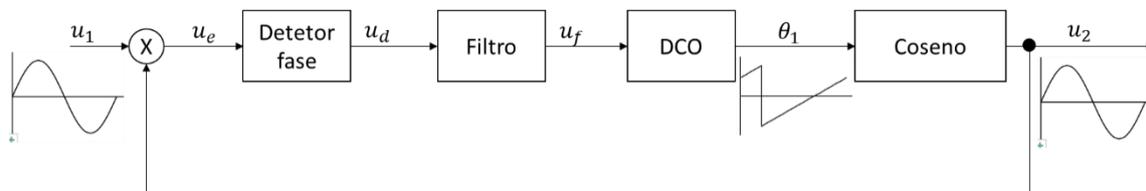


Figura 37 – Diagrama do PLL com tensão de retroação sinusoidal.

Ao nível das harmónicas nesta simulação apenas aparecem as de 2.^a Ordem na saída do detetor de fase. Ao nível de implementação, a geração de um seno vai aumentar o tempo de processamento e a quantidade de memória utilizada pelo microcontrolador.

Além disto para obter o sinal U_2 e não só o ângulo têm de ser gerados dois sinais desfasados, um para a realimentação e outro para a saída.

• **Dimensionamento do Filtro PI**

Realizando o dimensionamento tal como acima e modificando o valor do K_d para este tipo de implementação obtêm-se os parâmetros seguintes.

$$K_d = \frac{U_{10} U_{20}}{2} = \frac{1 \times 1}{2} = 0,5 \quad (43)$$

$$\tau_1 = \frac{K_p K_0}{\omega_n^2} = \frac{0,5 \times 100}{26,052^2} = 0,074 \text{ s} \quad (44)$$

$$\tau_2 = \frac{2 \zeta \omega_n \tau_1}{K_p \cdot K_v} = \frac{2 \times \frac{\sqrt{2}}{2} \times 26,052 \times 0,074}{0,5 \times 100} = 0,054 \text{ s} \quad (45)$$

Um dos problemas mencionados em literatura, evidenciado em [4], do detetor de fase do tipo multiplicador é a geração de uma harmónica de 2.^a ordem na saída do detetor de fase que provoca alguns erros no ângulo θ gerado pelo PLL tal como mostra a figura seguinte.

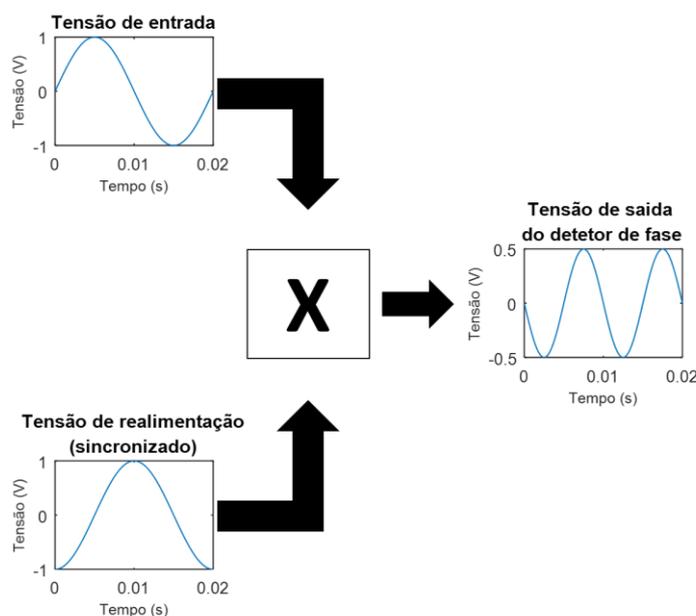


Figura 38 – Sinais de entrada e saída do detetor de fase multiplicador.

Desenvolvendo a multiplicação dos sinais de entrada do detetor de fase é possível verificar que existe uma explicação matemática para a existência destas harmónicas (Anexo 1). Como podemos ver nas expressões da tensão de saída do detetor de fase mostradas no Anexo 1 estas harmónicas são pares, sendo com a retroação sinusoidal, apenas de ordem 2.

Apesar de habitualmente no PLL o filtro servir para filtrar as altas frequências, a segunda 2.^a harmónica gerada pela multiplicação dos sinais não é suficientemente atenuada.

Para confirmar visualmente o aparecimento destas harmónicas, recorrendo ao Matlab, foi realizada a simulação do sistema com retroação sinusoidal e analisou-se o ângulo de saída, representado na Figura 39.

Erro no ângulo com retroação sinusoidal para tensão de entrada com 50 Hz

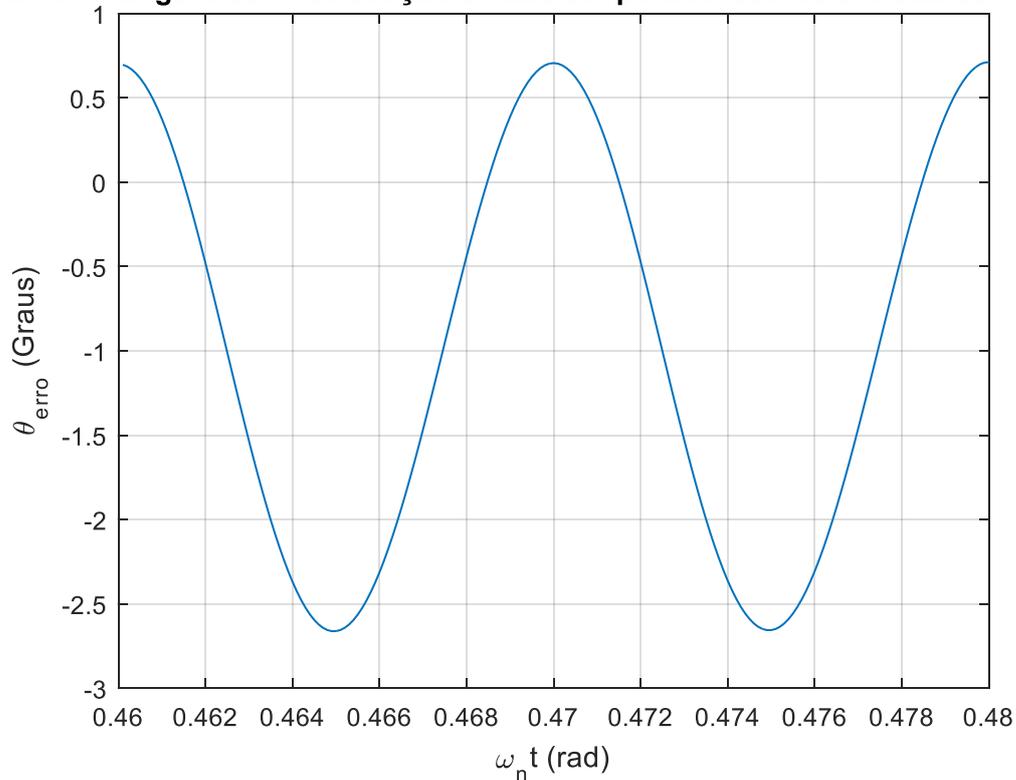


Figura 39 – Resultado da simulação para 50 Hz para o valor do erro no ângulo do PLL com retroação sinusoidal.

Como é possível verificar na figura anterior existe uma variação no erro do ângulo com o dobro da frequência de entrada, ou seja, 100 Hz. Desta forma irá proceder-se à atenuação desta componente através da implementação de um filtro rejeita-banda e de um gerador de sinal ortogonal [4].

Existem várias soluções para corrigir este problema, de seguida mostra-se o diagrama de blocos global do sistema para duas das soluções possíveis. É de notar que as soluções passam por adicionar um filtro, Figura 40, ou modificar o detetor de fase para um gerador de sinal ortogonal, Figura 41.

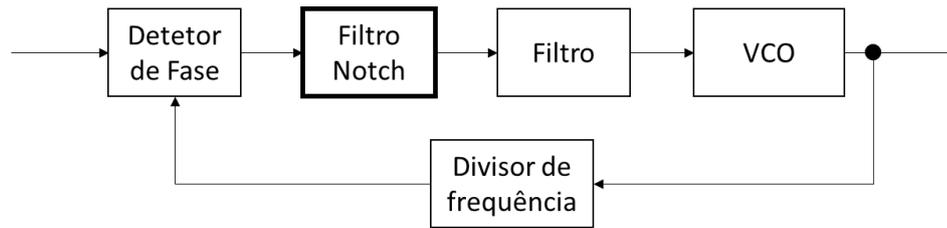


Figura 40 – Diagrama simplificado do PLL com filtro rejeita-banda.

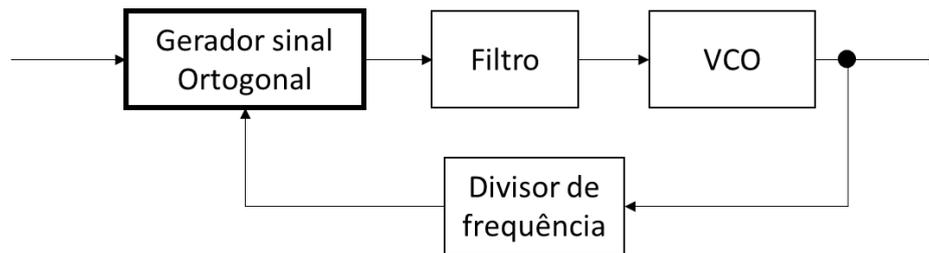


Figura 41 – Diagrama simplificado do PLL com gerador de sinal ortogonal.

Em termos ideais, qualquer uma destas soluções deve ter a capacidade de se adaptar às variações de frequência da rede a que está ligada, caso contrário aparece um pequeno erro no ângulo de saída quando a frequência for diferente de 50 Hz.

4.5 – SPLL com tensão de realimentação sinusoidal e filtro rejeita-banda

Para este SPLL o dimensionamento é igual ao dimensionamento anterior. Como tal vai dar-se importância ao filtro rejeita-banda que ainda não foi devidamente explorado.

- **Filtro rejeita-banda**

O filtro rejeita-banda é utilizado na saída do detetor de fase, tal como ilustra a figura seguinte, este atenua os sinais com o dobro da frequência da rede.

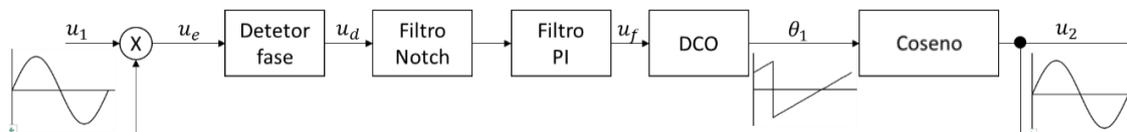


Figura 42 – Diagrama do PLL com filtro rejeita-banda.

A função transferência deste filtro é dada pela equação seguinte.

$$H_{f_{rejeitabanda}}(s) = \frac{G_0 (s^2 + \omega_c^2)}{s^2 + 2 \zeta \omega_c s + \omega_c^2} \quad (46)$$

Esta função transferência pode ser obtida de duas formas, através da substituição da variável s do domínio de *Laplace* por uma expressão que é função desta mesma variável na função transferência de um filtro passa baixo de 1.^a Ordem com frequência igual a 1 rad/s ou subtraindo a função de transferência de um filtro passa banda à unidade [8].

De seguida mostra-se a função transferência de um filtro passa baixo de 1.^aOrdem com frequência de 1 rad/s e uma tabela com as substituições a realizar para obtenção de outros filtros.

$$H_{f_{pbaixo}}(s) = \frac{1}{s + 1} \quad (47)$$

| | |
|-----------------------------|--|
| Passa-baixo → Passa-baixo | $s \rightarrow \frac{s}{\omega_c}$ |
| Passa-baixo → Passa-alto | $s \rightarrow \frac{\omega_c}{s}$ |
| Passa-baixo → Passa-banda | $s \rightarrow \frac{s^2 + \omega_s \omega_i}{s(\omega_s - \omega_i)}$ |
| Passa-baixo → Rejeita-banda | $s \rightarrow \frac{s(\omega_s - \omega_i)}{s^2 + \omega_s \omega_i}$ |

A função de transferência genérica de um filtro passa banda é dada pela equação (48), invertendo esta expressão, isto é subtraindo esta função transferência a 1 unidade, obtém-se também a função de transferência do filtro rejeita-banda, tal como mostra a equação (49).

$$H_{f_{passabanda}}(s) = \frac{G_0 2 \zeta \omega_c s}{s^2 + 2 \zeta \omega_c s + \omega_c^2} \quad (48)$$

$$H_{f_{rejeitabanda}}(s) = 1 - H_{f_{passabanda}}(s) \quad (49)$$

De seguida é mostrado um exemplo de diagrama de Bode genérico de um filtro rejeita banda.

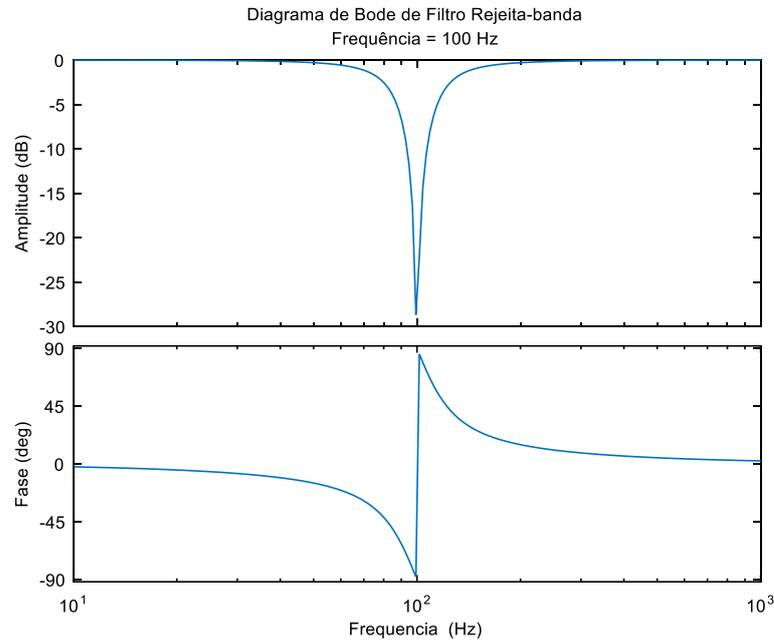


Figura 43 – Diagrama de bode de filtro rejeita-banda.

• Discretização do filtro rejeita-banda

Partindo da equação (46) e efetuando o processo de discretização por integração trapezoidal obtém-se a equação (50)

$$\frac{u_f(z)}{u_d(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \quad (50)$$

Com:

$$a_1 = \frac{2(T_s^2 \omega_c^2 - 4)}{T_s^2 \omega_c^2 + \left(\frac{1}{Q}\right) * 2 * T_s * \omega_c + 4}$$

$$a_2 = \frac{T_s^2 \omega_c^2 - \left(\frac{1}{Q}\right) * 2 * T_s * \omega_c + 4}{T_s^2 \omega_c^2 + \left(\frac{1}{Q}\right) * 2 * T_s * \omega_c + 4}$$

$$b_0 = \frac{G(T_s^2 \omega_c^2 + 4)}{T_s^2 \omega_c^2 + \left(\frac{1}{Q}\right) * 2 * T_s * \omega_c + 4}$$

$$b_1 = \frac{2 G(T_s^2 \omega_c^2 - 4)}{T_s^2 \omega_c^2 + \left(\frac{1}{Q}\right) * 2 * T_s * \omega_c + 4}$$

$$b_2 = \frac{G(T_s^2 \omega_c^2 + 4)}{T_s^2 \omega_c^2 + \left(\frac{1}{Q}\right) * 2 * T_s * \omega_c + 4}$$

Para validar o modelo discretizado do filtro rejeita-banda foi realizado o diagrama de Bode deste filtro no domínio contínuo e discreto como mostra a figura seguinte.

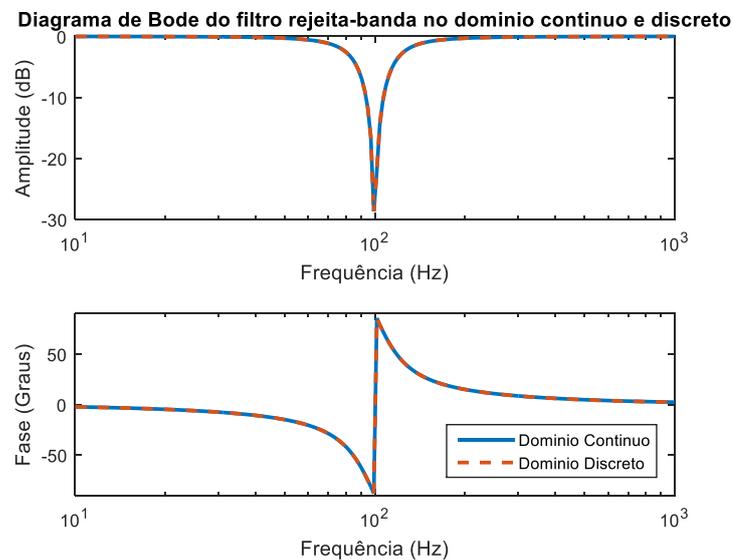


Figura 44 – Diagrama de Bode do filtro rejeita-banda no domínio contínuo e discreto.

• Dimensionamento do filtro rejeita-banda

O dimensionamento do filtro rejeita-banda foi realizado sendo a frequência de rejeição máxima em torno dos 100 Hz e com o objetivo de atenuar frequências entre 94 Hz (dobro da frequência mínima imposta pela normalização) e 104 Hz (dobro da frequência máxima imposta pela normalização).

Seguidamente foram experimentados vários fatores de qualidade e respetivos parâmetros ζ através da relação seguidamente representada.

$$Q = \frac{1}{2\zeta} \quad (51)$$

Além destas relações o fator de qualidade está ainda ligado com a largura de banda do filtro pela equação (52).

$$Q = \frac{f_0}{LB} \quad (52)$$

O gráfico seguinte mostra os diferentes diagramas de Bode para diferentes fatores de qualidade e conseqüentemente larguras de banda.

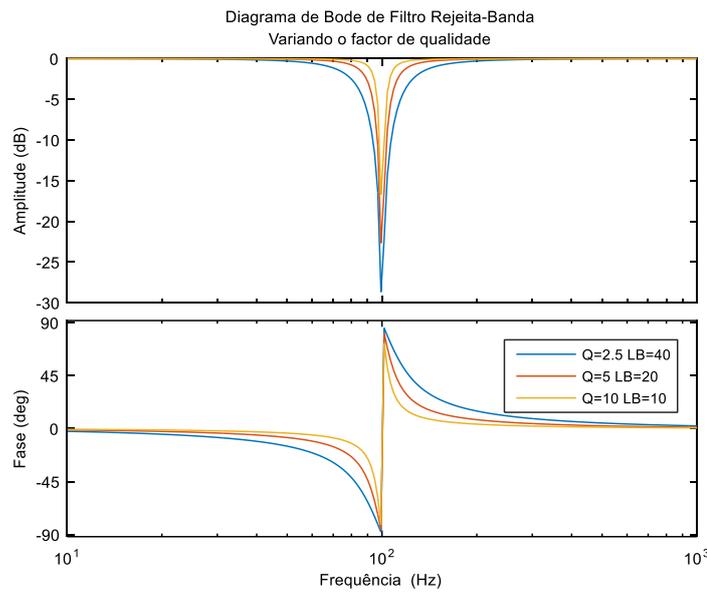


Figura 45 – Diagrama de Bode do filtro rejeita-banda variando o fator de qualidade.

Como este tipo de filtros deformam facilmente a sua característica em frequência quando implementados digitalmente, a largura de banda selecionada para a implementação foi 40 Hz, pois a característica não é tão abrupta como as restantes características testadas.

4.6 – SPLL com tensão de realimentação sinusoidal e gerador de sinal ortogonal

- Gerador de sinal ortogonal

Além do filtro é possível linearizar a saída do detetor utilizando um gerador de componentes ortogonais e uma transformação de *Park* que torna o sinal de entrada no filtro linear.

O diagrama funcional do sistema é representado pela figura imediatamente abaixo.

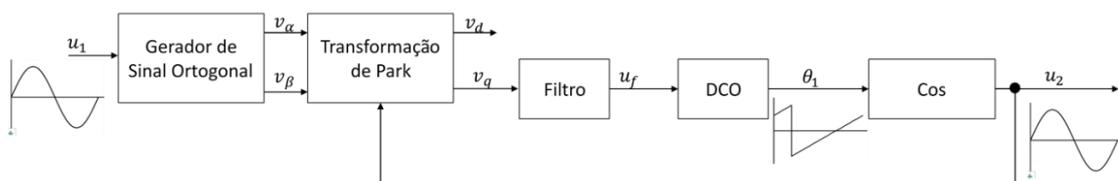


Figura 46 – Diagrama do PLL com gerador de sinal ortogonal.

A geração deste sinal ortogonal pode ser feita de várias formas, neste caso foi utilizado um integrador de 2.^a ordem para gerar as duas componentes, α e β .

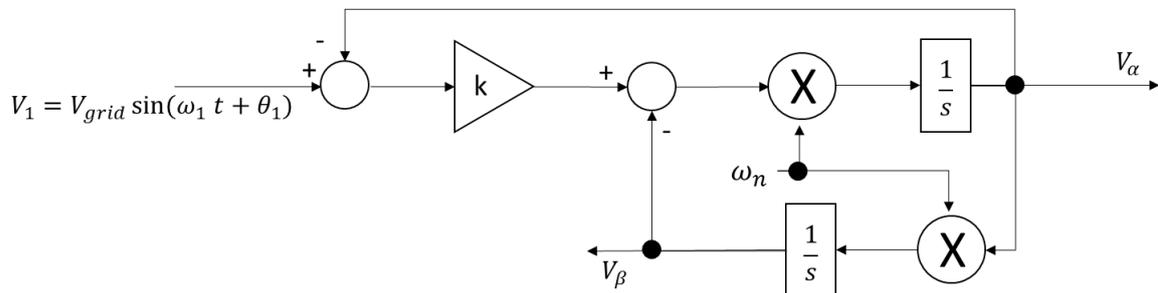


Figura 47 – Diagrama do gerador de sinal ortogonal.

Nas equações abaixo são mostradas as funções de transferência do sistema para a saída α e β .

$$H_{\alpha}(s) = \frac{2 \zeta \omega_n s}{s^2 + 2 \zeta \omega_n s + \omega_n^2} \quad (53)$$

$$H_{\beta}(s) = \frac{2 \zeta \omega_n^2}{s^2 + 2 \zeta \omega_n s + \omega_n^2} \quad (54)$$

A introdução do gerador de sinal ortogonal deverá minimizar os efeitos da 2.^a harmónica, no entanto o tempo de processamento é bastante incrementado. Isto deve-se ao facto dos filtros descritos serem de 2.^a ordem sendo necessária uma melhor resolução para obter um bom comportamento no filtro.

Ao nível de simulação desprezando o funcionamento do PIC esta solução irá provavelmente melhorar os resultados, no entanto quando for implementado é possível que os resultados não sejam tão evidentes devido à drástica redução de resolução ou frequência de aquisição.

- **Discretização do gerador de sinal ortogonal**

Foram utilizadas as discretizações efetuadas na referência [4], que são mostradas nas equações (55) e (56).

$$H_d = \frac{b_0 + b_2 z^{-2}}{1 - a_1 z^{-1} - a_2 z^{-2}} \quad (55)$$

Com:

$$a_1 = \frac{2(4 - (\omega_n T_s)^2)}{2\zeta\omega_n T_s + (\omega_n T_s)^2 + 4}$$

$$a_2 = \frac{2\zeta\omega_n T_s - (\omega_n T_s)^2 - 4}{2\zeta\omega_n T_s + (\omega_n T_s)^2 + 4}$$

$$b_0 = \frac{2\zeta\omega_n T_s}{2\zeta\omega_n T_s + (\omega_n T_s)^2 + 4}$$

$$b_2 = -\frac{2\zeta\omega_n T_s}{2\zeta\omega_n T_s + (\omega_n T_s)^2 + 4}$$

$$H_q = \frac{qb_0 + qb_1 z^{-1} + qb_2 z^{-2}}{1 - a_1 z^{-1} - a_2 z^{-2}} \quad (56)$$

Com:

$$a_1 = \frac{2(4 - (\omega_n T_s)^2)}{2\zeta\omega_n T_s + (\omega_n T_s)^2 + 4}$$

$$a_2 = \frac{2\zeta\omega_n T_s - (\omega_n T_s)^2 - 4}{2\zeta\omega_n T_s + (\omega_n T_s)^2 + 4}$$

$$qb_0 = \frac{\zeta(\omega_n T_s)^2}{2\zeta\omega_n T_s + (\omega_n T_s)^2 + 4}$$

$$qb_1 = 2\frac{\zeta(\omega_n T_s)^2}{2\zeta\omega_n T_s + (\omega_n T_s)^2 + 4}$$

$$qb_2 = \frac{\zeta(\omega_n T_s)^2}{2\zeta\omega_n T_s + (\omega_n T_s)^2 + 4}$$

- **Dimensionamento do gerador de sinal ortogonal**

Os parâmetros deste gerador foram calculados a partir do critério Integral do erro absoluto multiplicado pelo tempo (*Integral Time Absolute Error – ITAE*) de 2ª ordem mostrado em [9], onde tal como acima foi selecionado o parâmetro ζ e o parâmetro ω_n

$$\zeta = \frac{\sqrt{2}}{2} = 0,707 \text{ rad/s} \quad (57)$$

$$\omega_n = 2 \pi 50 = 314,15 \text{ rad/s} \quad (58)$$

Capítulo V: Implementação e Resultados

Resumo:

No presente capítulo, é representado o diagrama da montagem a implementar na prática através do PIC32MX230F256B e os resultados obtidos através de Matlab para as diferentes simulações.

Os resultados mostrados são para o modelo digital com e sem conteúdo harmónico no sinal de entrada, desprezando as limitações de resolução. De seguida são mostrados os resultados do Matlab considerando a capacidade de processamento do microcontrolador e por fim os resultados provenientes da implementação prática.

5.1 – Esquema de ligações da implementação prática

Para a implementação do SPLL pensou-se utilizar um microcontrolador de 16 bits de processamento e com capacidade de executar 50 milhões de instruções por segundo. Tal facto deveu-se a ser necessária a implementação de alguns filtros, sendo um microcontrolador de 8 bits provavelmente lento ao nível de processamento e para esta aplicação.

A primeira implementação num PIC de 16 bits foi realizada utilizando valores inteiros de 16 bits e inteiros longos de 32 bits e 64 bits, no entanto as operações realizadas com variáveis acima de 32 bits eram bastante lentas e não se conseguiu uma frequência de aquisição que permitisse o funcionamento do microcontrolador nas implementações com filtro rejeita-banda e gerador de sinal ortogonal.

Depois disto, tentou-se a implementação apenas com valores inteiros de 16 e 32 bits, tendo como consequência a perda de resolução nas variáveis. Desta forma surge um erro considerável proveniente das aproximações realizadas. Além deste erro, existia também o erro devido à frequência de aquisição ser baixa (1 kHz), que corresponde a um erro no ângulo de aproximadamente 17 graus.

Observando os resultados obtidos, verificou-se que a implementação num microcontrolador de 16 bits não era razoável devido às capacidades limitadas do microcontrolador evoluindo-se para um microcontrolador de 32 bits com capacidade de executar 83 milhões de instruções por segundo e com uma grande facilidade em executar cálculos com vírgula flutuante. O circuito implementado com o microcontrolador de 32 bits é representado na Figura 48.

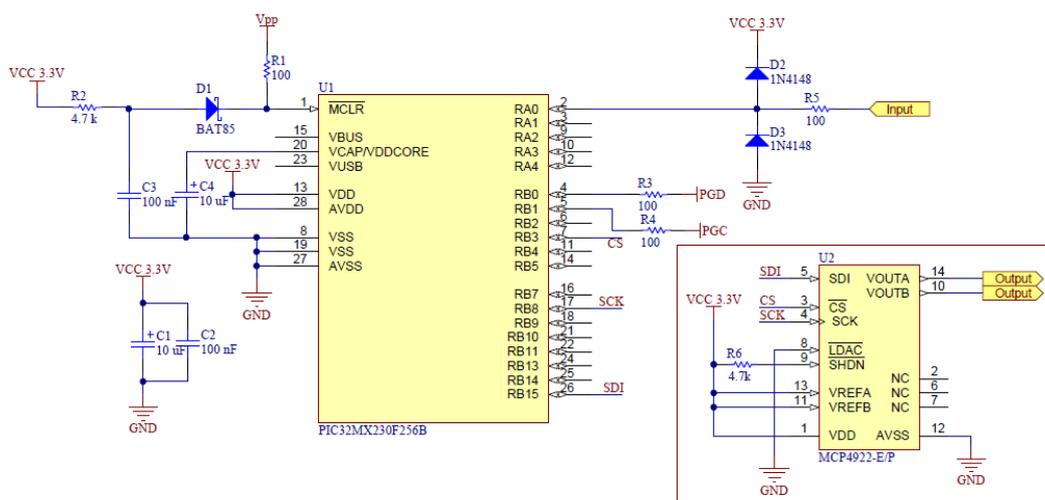


Figura 48 – Esquema de ligações do PIC32MX230F256B, utilizado para implementação.

Na figura anterior é possível observar que existem duas saídas no conversor digital para analógico (DAC) sendo a saída 'A' a forma de onda sinusoidal de saída do PLL desfasada 90° para uma melhor comparação dos resultados e a saída 'B' o ângulo de saída do PLL, cuja onda é um dente de serra.

Ao nível de resultados obtidos com este microcontrolador verificou-se que a frequência mais baixa de amostragem possível foi 4 kHz que corresponde a um erro no ângulo de 4,5 graus devido ao efeito da discretização. Através das operações com vírgula flutuante não se verificaram erros devido ao processo de cálculo.

5.2 – Resposta do sistema digital para uma tensão de entrada sem conteúdo harmónico (sinusoidal)

Os resultados das simulações que se seguem foram obtidos para frequências de entrada de 50 Hz e de acordo com os limites de frequência permitidos pela normalização, ou seja, 47 Hz e 52 Hz. Estes resultados foram obtidos através do Matlab e representam a tensão de entrada e tensões de saída para uma onda quadrada na retroação, uma onda sinusoidal na retroação sem filtragem da componente harmónica de 2.^a Ordem, com filtragem harmónica através de um filtro rejeita-banda e com um gerador de sinal ortogonal.

Cada figura é composta por 2 gráficos, o primeiro que é uma aproximação ao domínio contínuo e o segundo que foi realizado com as frequências de aquisição da implementação prática. A aproximação ao domínio contínuo foi feita através do aumento da frequência de aquisição para um valor onde não fosse visível o efeito da discretização da onda de saída, sendo usado o valor de 100 kHz.

Os resultados discretos foram obtidos depois de feita a implementação e da medição da frequência máxima de funcionamento de cada PLL implementado. Para retroação quadrada, os resultados foram obtidos com a frequência de aquisição a 6 kHz sendo a frequência de aquisição para retroação sinusoidal sem filtro rejeita-banda de 5 kHz. Para a implementação do filtro rejeita banda e do gerador de sinal ortogonal foi necessário baixar ainda mais a frequência de aquisição tendo-se baixado esta frequência para 4 kHz no caso do filtro rejeita-banda e 3 kHz para o gerador de sinal ortogonal. As frequências não são constantes devido aos resultados serem adquiridos sempre para o limite de capacidade de cálculo do microcontrolador, daí aos resultados com algoritmos mais complexos possuírem frequências de aquisição mais baixas devido ao aumento do tempo de processamento de um ciclo do SPLL.

• **Resultados em Tensão (regime estacionário)**

De forma a reduzir a legenda utilizada nos gráficos foram utilizados alguns nomes, que apesar de se tentar que fossem intuitivos, são descritos na tabela seguinte de forma a clarificar a informação.

| Referência no gráfico | Descrição |
|--------------------------------|--|
| u_1 Tensão Entrada | Tensão de Entrada dos PLLs |
| u_2 Quadrada | Tensão de retroação e de saída do PLL para a implementação com retroação quadrada. |
| u_2 Sinusoidal Simples | Tensão de saída do PLL para a implementação com retroação sinusoidal sem filtragem da 2. ^a harmónica (desfasada de 90° em relação ao sinal de retroação). |
| u_2 Sinusoidal Rejeita-Banda | Tensão de saída do PLL para a implementação com retroação sinusoidal com filtragem da 2. ^a harmónica através da colocação de um filtro rejeita-banda (desfasada de 90° em relação ao sinal de retroação). |
| u_2 Sinusoidal Ortogonal | Tensão de saída do PLL para a implementação com retroação sinusoidal com um gerador de sinal ortogonal. (desfasada de 90° em relação ao sinal de retroação). |

Tabela 6 – Tabela com descrição da legenda dos resultados obtidos (Tensão).

A Figura 49 representa os resultados descritos para uma frequência de entrada de 47 Hz.

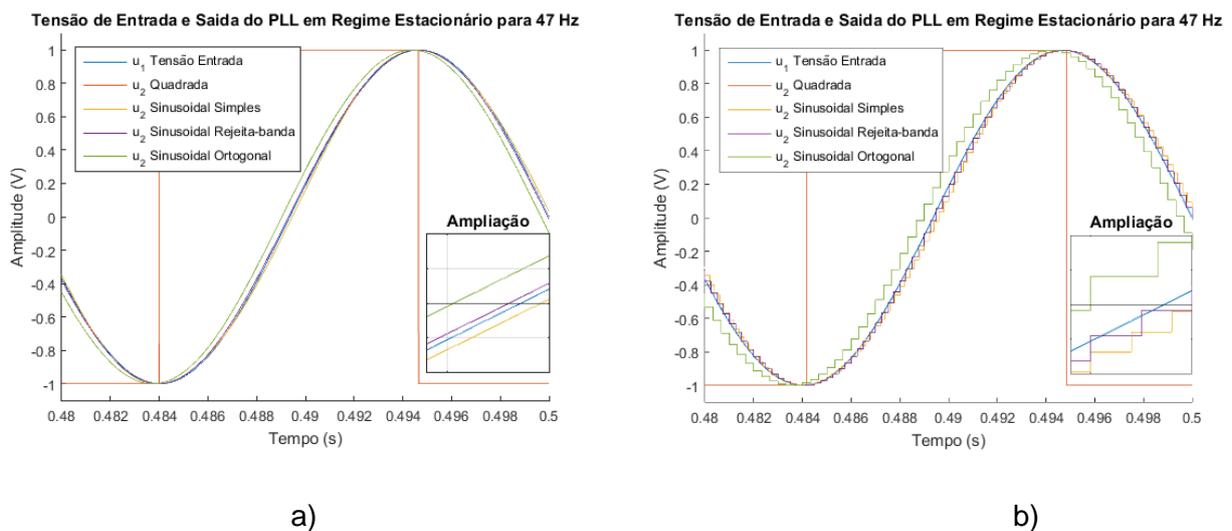


Figura 49 – Tensões de entrada e saídas do SPLL para uma frequência de 47 Hz para as várias implementações a) Simulação em regime contínuo b) Simulação em regime discreto.

A figura seguinte, Figura 50, representa os resultados descritos para uma frequência de entrada de 50 Hz.

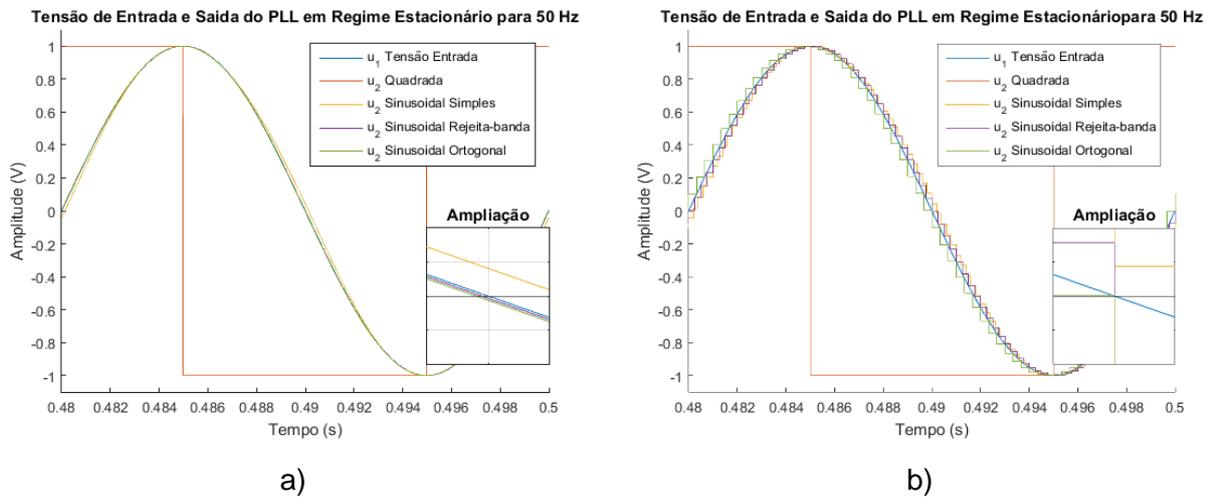


Figura 50 – Tensões de entrada e saídas do SPLL para uma frequência de 50 Hz para as várias implementações a) Simulação em regime contínuo b) Simulação em regime discreto.

Para uma frequência de entrada de 52 Hz os resultados são dados pela Figura 51.

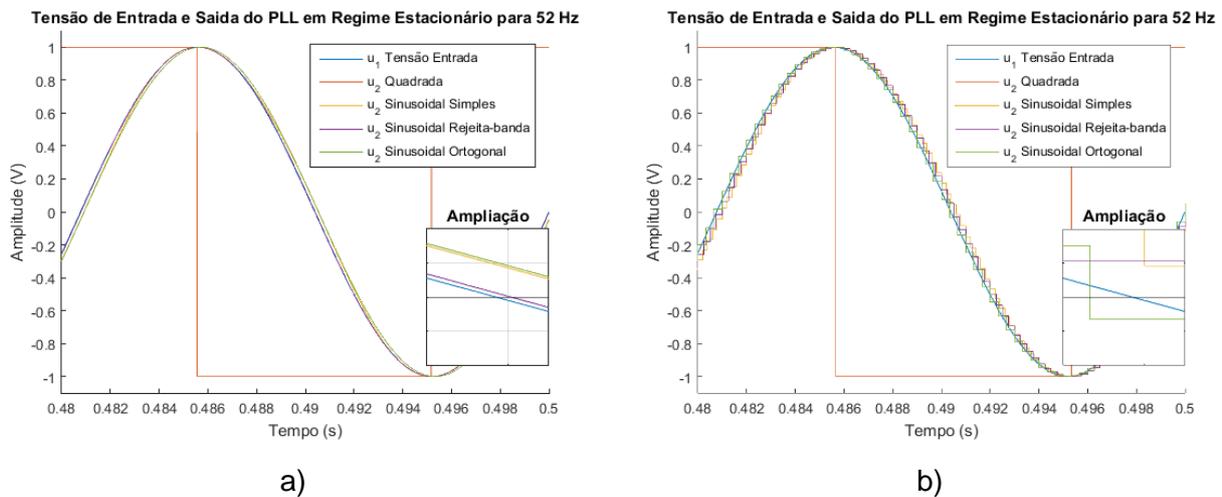


Figura 51 – Tensões de entrada e saídas do SPLL para uma frequência de 52 Hz para as várias implementações a) Simulação em regime contínuo b) Simulação em regime discreto.

Em primeira análise aos resultados obtidos pode verificar-se que, ignorando os degraus causados pela redução da frequência de aquisição, as curvas quer com alta frequência de aquisição quer com baixa são bastante semelhantes, notando-se apenas em alguns resultados uma ligeira melhoria na simulação em regime aproximadamente contínuo.

Estas simulações não são as mais indicadas para a análise do erro no ângulo, no entanto serão bastante úteis ao nível da comparação com os resultados práticos.

- **Resultados da diferença entre ângulo de entrada e saída (regime transitório)**

Nas 3 figuras que se seguem é mostrado o comportamento do ângulo em regime dinâmico, ou seja quando o PLL está dessincronizado. Os ângulos de saída das implementações foram obtidos através do cálculo efetuado pelo PLL e os ângulos de entrada calculados a partir do tempo de simulação multiplicado pela frequência angular da rede elétrica. A partir destes dois ângulos e da sua diferença obtiveram-se os resultados apresentados de seguida.

Para os gráficos obtidos, a descrição da legenda é idêntica à dos resultados anteriores, no entanto irá ser mostrada novamente na tabela seguinte, Tabela 7, a descrição das referências utilizadas com algumas correções.

| Referência no gráfico | Descrição |
|---------------------------------|---|
| Quadrada | Erro entre o ângulo de entrada e saída do PLL para a implementação com retroação quadrada. |
| Sinusoidal Simples | Erro entre o ângulo de entrada e saída do PLL para a implementação com retroação sinusoidal sem filtragem da 2. ^a harmónica (desfasada de 90° em relação ao sinal de retroação). |
| Sinusoidal Rejeita-Banda | Erro entre o ângulo de entrada e saída do PLL para a implementação com retroação sinusoidal com filtragem da 2. ^a harmónica através da colocação de um filtro rejeita-banda (desfasada de 90° em relação ao sinal de retroação). |
| Sinusoidal Ortogonal | Erro entre o ângulo de entrada e saída do PLL para a implementação com retroação sinusoidal com um gerador de sinal ortogonal. (desfasada de 90° em relação ao sinal de retroação). |

Tabela 7 – Tabela com descrição da legenda dos resultados obtidos (erro no ângulo).

Os resultados descritos para uma frequência de entrada de 47 Hz são mostrados na Figura 52, para 50 Hz são mostrados na Figura 53 e para 52 Hz na Figura 54.

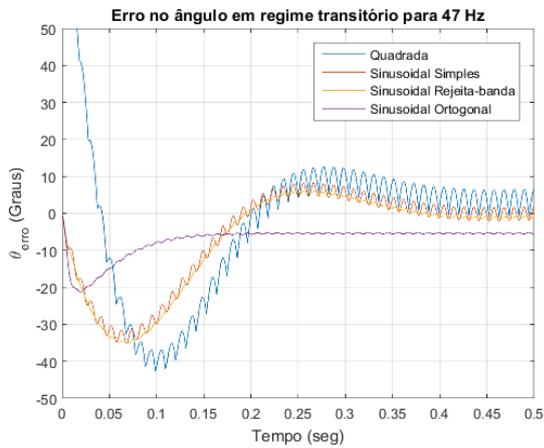


Figura 52 – Diferença entre o ângulo de entrada e saída do SPLL para uma frequência de 47 Hz para as diferentes implementações.

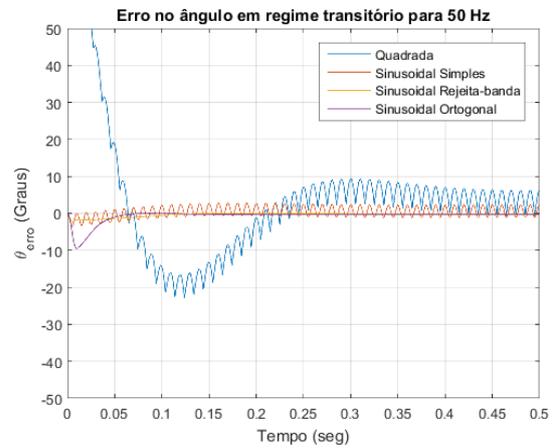


Figura 53 – Diferença entre o ângulo de entrada e saída do SPLL para uma frequência de 50 Hz para as diferentes implementações.

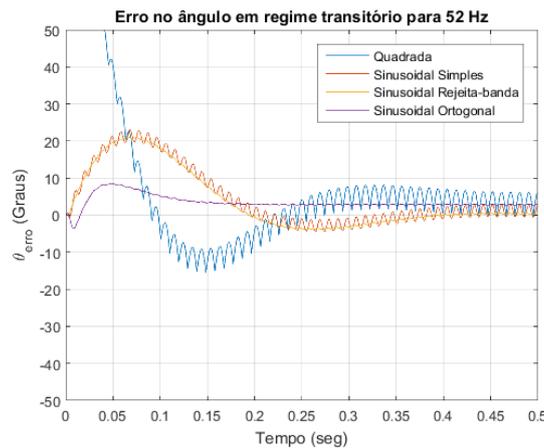


Figura 54 – Diferença entre o ângulo de entrada e saída do SPLL para uma frequência de 52 Hz para as diferentes implementações.

Através da análise do regime transitório pode-se observar que tal como era esperado existe algum tempo de estabilização do PLL. Com o aparecimento de cavas de tensão este regime é bastante importante pois possui uma dinâmica que permite a compensação das mesmas.

- **Resultados da diferença entre ângulo de entrada e saída (regime estacionário)**

Os resultados seguintes mostram também a diferença entre o ângulo de entrada e saída, no entanto, os que se seguem mostram uma ampliação do PLL quando este está em modo sincronizado.

Será a partir destes resultados que possivelmente será possível fazer uma melhor análise, como verificar qual o PLL com menor erro em regime estacionário.

Tal como anteriormente, os resultados foram obtidos para as frequências de 47 Hz (Figura 55), 50 Hz (Figura 56) e 52 Hz (Figura 57), sendo a descrição da legenda igual à já feita anteriormente na Tabela 7.

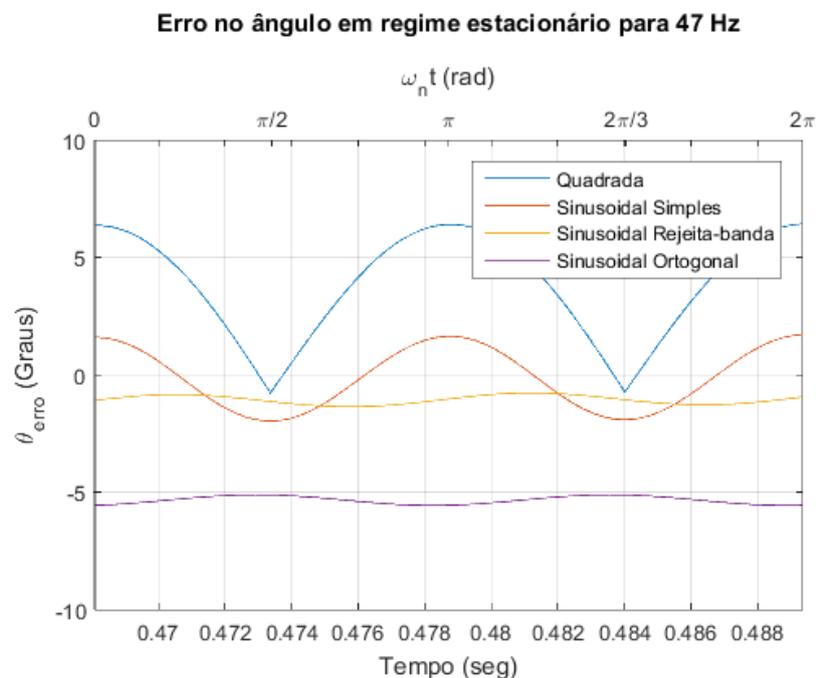


Figura 55 – Diferença entre o ângulo de entrada e saída do SPLL para uma frequência de 47 Hz para as diferentes implementações (em regime estacionário).

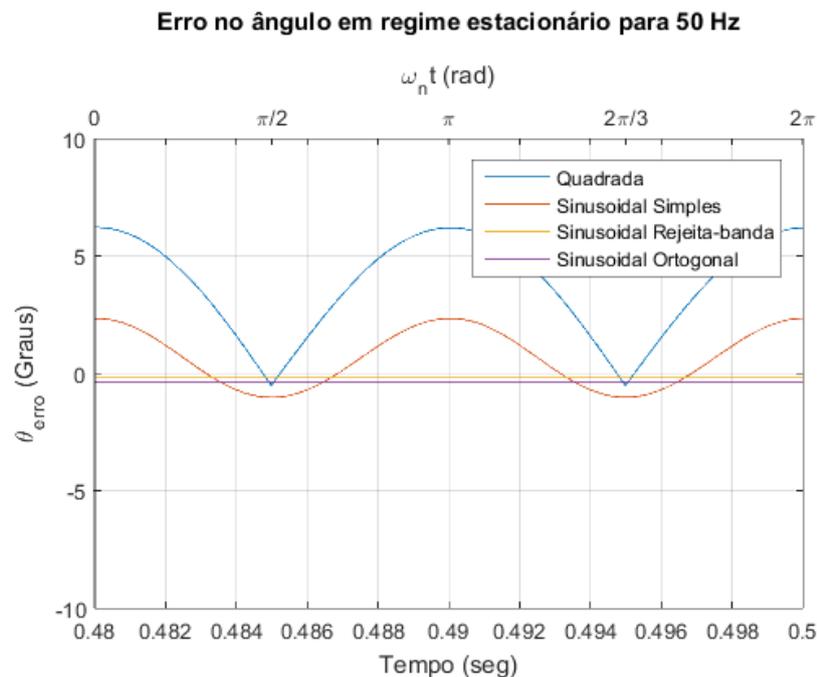


Figura 56 – Diferença entre o ângulo de entrada e saída do SPLL para uma frequência de 50 Hz para as diferentes implementações (em regime estacionário).

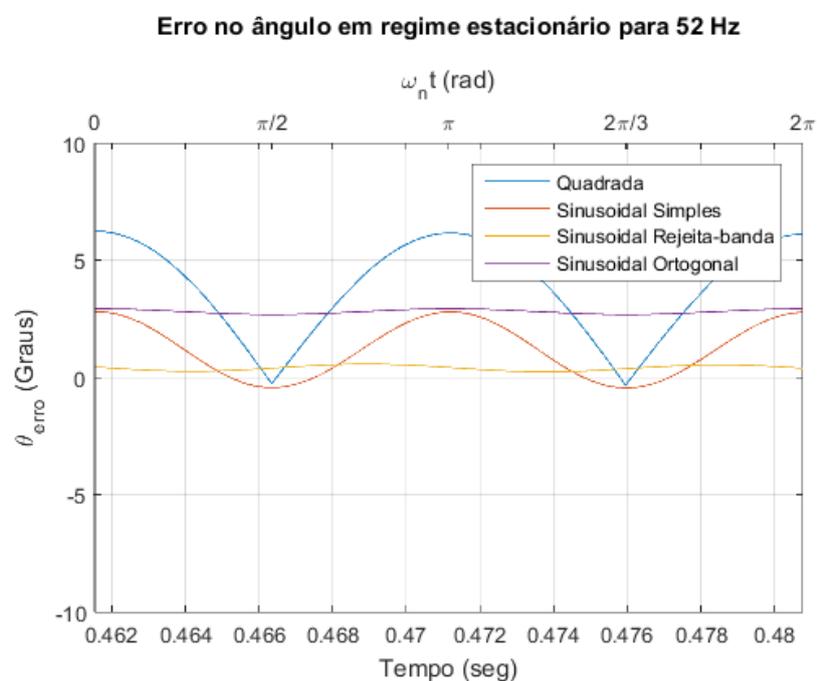


Figura 57 – Diferença entre o ângulo de entrada e saída do SPLL para uma frequência de 52 Hz para as diferentes implementações (em regime estacionário).

Analisando os resultados obtidos neste ponto verifica-se que a introdução do gerador de sinais e do filtro rejeita-banda permite uma grande capacidade de rejeição da 2ª harmónica gerada pela multiplicação dos sinais.

Apesar de existir sempre um erro constante quer nos resultados com gerador de sinal ortogonal quer nos resultados com filtro rejeita-banda, este erro é facilmente compensado com a adição ou subtração de uma constante. Nos resultados em que existe variação do erro ao longo do tempo, esta compensação é dificultada.

5.3 – Resposta do sistema digital para uma tensão de entrada com conteúdo harmónico

Neste subcapítulo serão mostrados os resultados obtidos através de simulação para uma tensão de entrada com componentes harmónicas segundo a normalização, até 8%. Todos os resultados serão idênticos aos do ponto 5.2 – Resposta do sistema digital para uma tensão de entrada sem conteúdo harmónico (sinusoidal), sendo a única alteração a componente harmónica de entrada.

Para esta situação apenas serão mostrados os resultados no domínio contínuo.

O cálculo das harmónicas necessárias para perfazer um valor total de 8% é mostrado na equação seguinte, onde U_n simboliza o valor eficaz da tensão da rede para a harmónica n e x_n um coeficiente correspondente à harmónica n .

$$\text{THD} = \frac{\sqrt{U_2^2 + U_3^2 + \dots}}{U_1} \quad (59)$$

Para a situação em questão ira utilizar-se apenas as harmónicas impares. Colocando o distorção harmónica total (THD) igual a 0,08, ou seja 8%, e substituindo o valor das tensão fica-se com:

$$(0,08 V_1)^2 = (x_3 V_1)^2 + (x_5 V_1)^2 + (x_7 V_1)^2 + \dots \quad (60)$$

Fazendo o cálculo dos diferentes quadrados do produto chegou-se à conclusão que a 5.^a harmónica não era suficiente e que a 7.^a dava um resultado superior a 8%. Desta forma procedeu-se ao cálculo da percentagem que a 7.^a harmónica deveria ter para que o resultado mais desfavorável fosse de 8% (Com conteúdo de 5% na 3.^a harmónica e 6% n4 5.^a harmónica).

$$x_7 = \sqrt{\frac{(0,08 U_1)^2 - (x_3 U_1)^2 - (x_5 U_1)^2}{U_1^2}} = 0,016 \approx 0,02 \quad (61)$$

Para o cálculo da senoide utilizou-se a equação (62).

$$U_1 = \sin(2\pi ft + \theta_1) + 0,05 \sin(2\pi 3ft) + 0,06 \sin(2\pi 5ft) + 0,02 \sin(2\pi 7ft) \quad (62)$$

• **Resultados em Tensão (regime estacionário)**

Tal como no subcapítulo anterior, de seguida serão mostrados os resultados obtidos da tensão de entrada e saída para as diferentes implementações realizadas.

A Figura 49 representa os resultados descritos para uma frequência de entrada de 47 Hz.

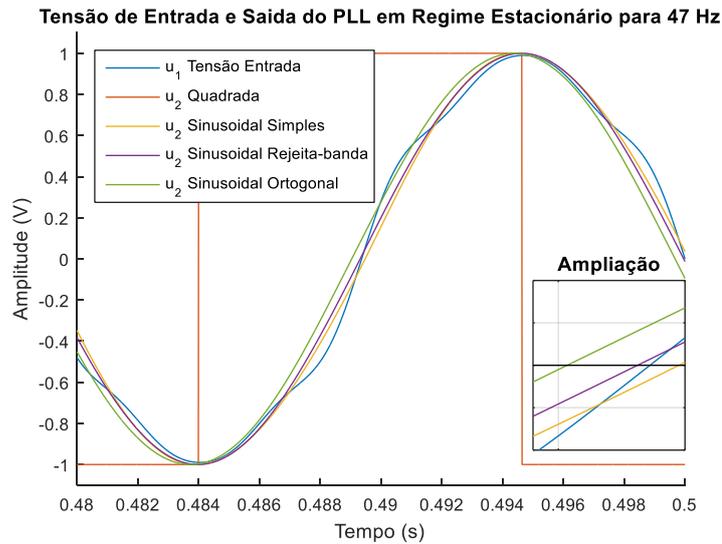


Figura 58 – Tensões de entrada e saídas do SPLL para uma frequência de 47 Hz para as várias implementações em regime contínuo.

A figura seguinte, Figura 50, representa os resultados descritos para uma frequência de entrada de 50 Hz.

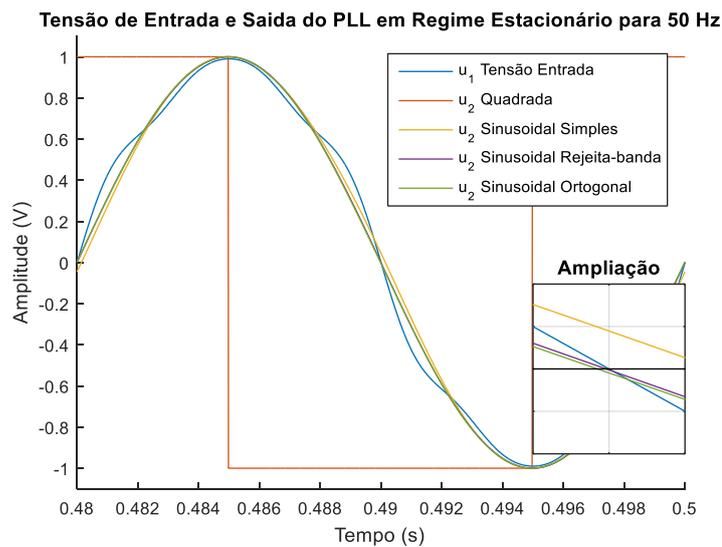


Figura 59 – Tensões de entrada e saídas do SPLL para uma frequência de 50 Hz para as várias implementações em regime contínuo.

Para uma frequência de entrada de 52 Hz os resultados são dados pela Figura 51.

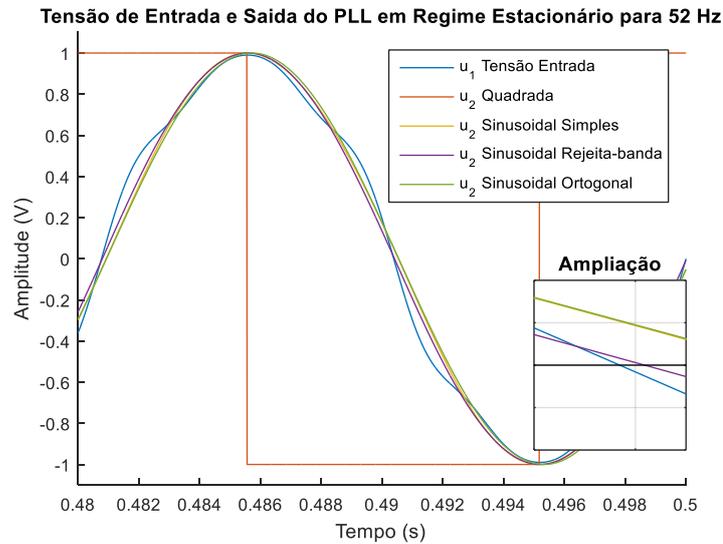


Figura 60 – Tensões de entrada e saídas do PLL para uma frequência de 52 Hz para as várias implementações em regime contínuo.

Através dos resultados obtidos com deformação do ângulo de entrada é possível verificar que apesar da distorção na tensão de entrada, a tensão de saída continua a ser aproximadamente sinusoidal, o que permite utilizar estes circuitos em aplicações como a compensação da potência deformante.

- **Resultados da diferença entre ângulo de entrada e saída (regime estacionário)**

Para os valores de erro do ângulo, a comparação foi feita apenas com a componente principal, ou seja 50 Hz, de forma a verificar se a deformação na onda de entrada influenciava o sinal de saída. Para isto irá verificar-se se os resultados são idênticos aos mostrados sem deformação do sinal de entrada.

As figuras seguintes mostram os resultados para 47, 50 e 52 Hz respetivamente.

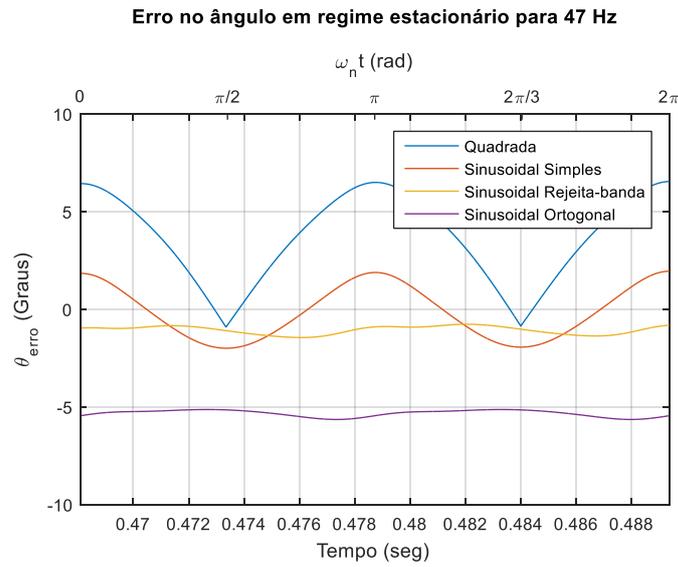


Figura 61 – Diferença entre o ângulo de entrada e saída do SPLL para uma frequência de 47 Hz para as diferentes implementações (em regime estacionário).

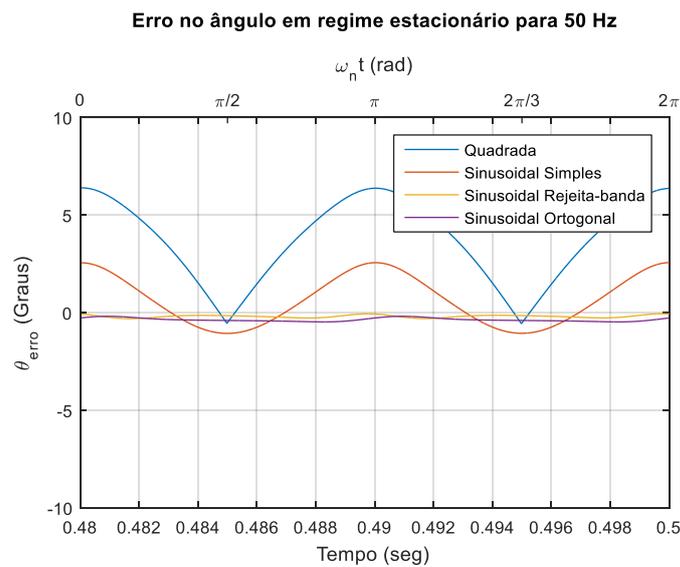


Figura 62 – Diferença entre o ângulo de entrada e saída do SPLL para uma frequência de 50 Hz para as diferentes implementações (em regime estacionário).

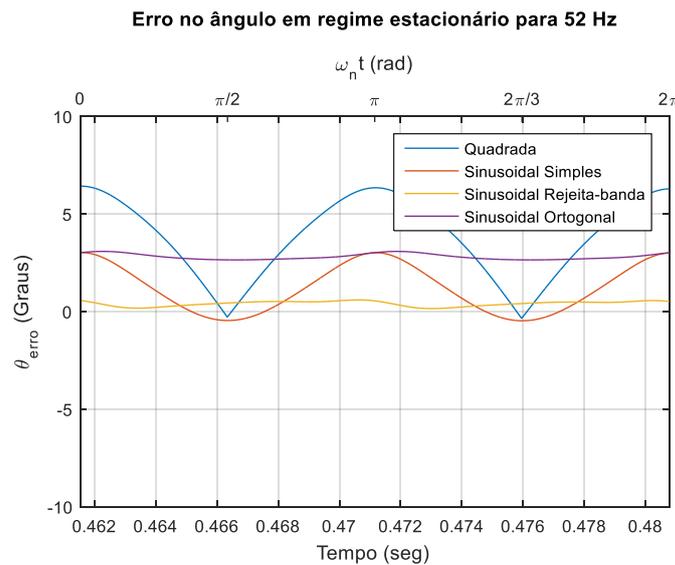


Figura 63 – Diferença entre o ângulo de entrada e saída do SPLL para uma frequência de 52 Hz para as diferentes implementações (em regime estacionário).

Com os resultados obtidos pela medição do erro entre os ângulos de entrada e saída é possível verificar que aparecem também algumas componentes harmónicas de maior ordem que não são completamente filtradas. No entanto, realizando uma aproximação da imagem verifica-se que esta variação é de cerca de 1 grau logo não é muito significativa.

É possível verificar também que os resultados são muito idênticos aos resultados sem perturbações harmónicas na entrada, o que comprova a robustez dos PLLs.

5.4 – Resultados obtidos através de implementação

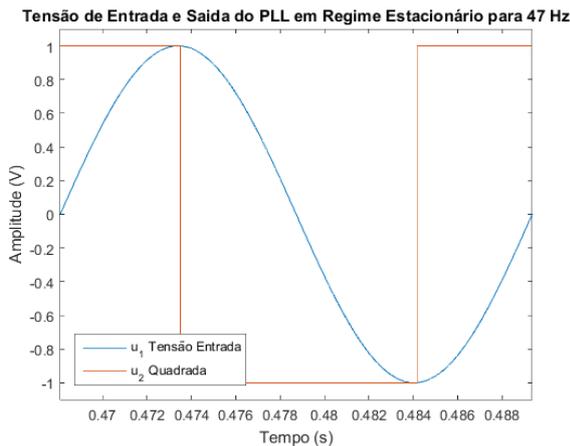
Os resultados que se seguem representam as simulações discretas e os resultados obtidos experimentalmente para uma tensão de entrada de 50 Hz sinusoidal de 0 a 3.3V.

Tal como já foi dito anteriormente, os resultados para retroação quadrada foram obtidos com a frequência de aquisição a 6 kHz, para retroação sinusoidal sem filtro rejeita-banda a frequência de aquisição é de 5 kHz, para a implementação do filtro rejeita banda a frequência de aquisição é de 4 kHz e para o gerador de sinal ortogonal a frequência de aquisição é de 3 kHz.

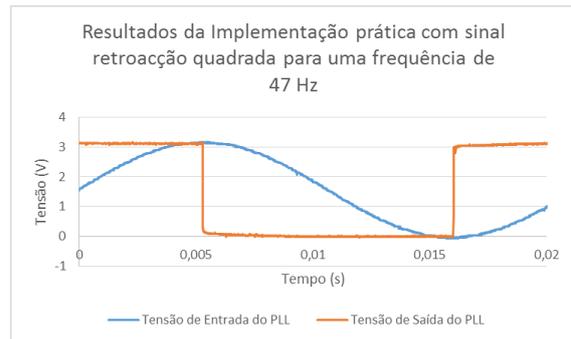
Para todas as figuras, a azul está representada a tensão de entrada do PLL e a laranja a tensão de saída. A descrição feita na Tabela 6 é aplicada também às legendas dos resultados seguintes.

• **Resultados para frequência de entrada de 47 Hz**

Os resultados da implementação prática, para uma frequência de entrada de 47 Hz, são os observados imediatamente abaixo para retroação com onda quadrada (Figura 64), sinusoidal sem filtragem da componente de 2ª harmónica (Figura 65), sinusoidal com filtro rejeita-banda (Figura 66) e sinusoidal com gerador de sinais ortogonais (Figura 67).

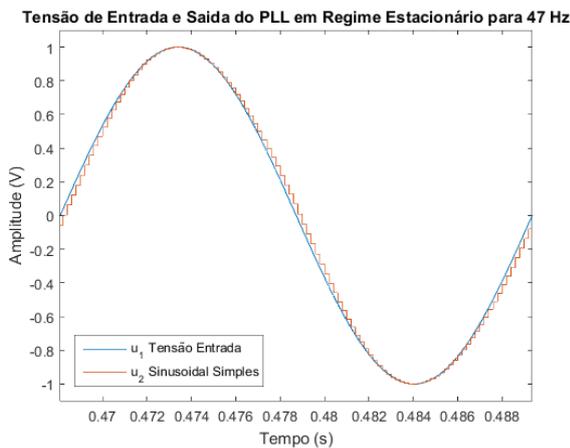


a)

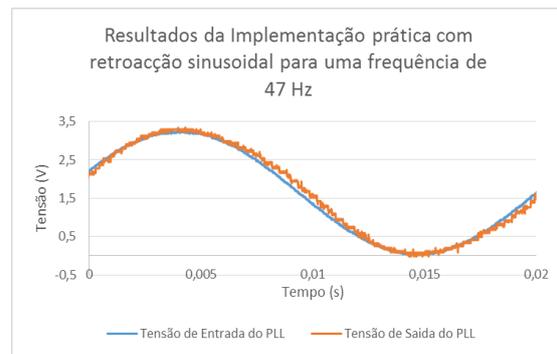


b)

Figura 64 – Tensões de entrada e saídas do com sinal de retroação quadrado para um sinal de entrada com 47 Hz a) Simulação b) Experimental.



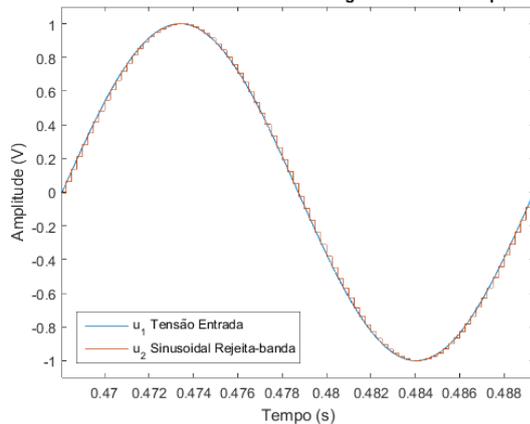
a)



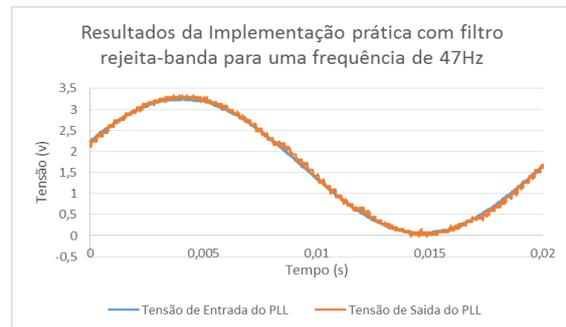
b)

Figura 65 – Tensões de entrada e saídas do SPLL com sinal de retroação sinusoidal para um sinal de entrada com 47 Hz a) Simulação b) Experimental.

Tensão de Entrada e Saída do PLL em Regime Estacionário para 47 Hz



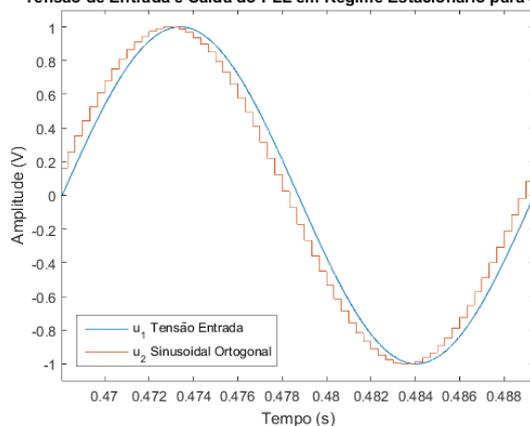
a)



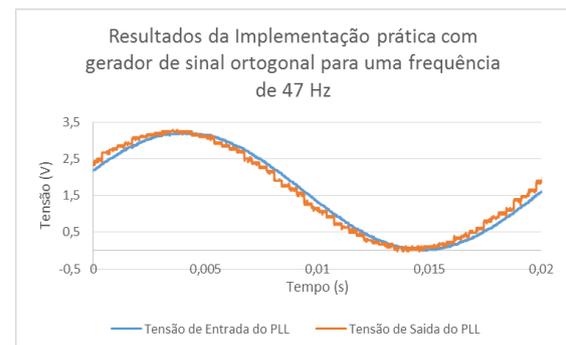
b)

Figura 66 – Tensões de entrada e saídas do SPLL com sinal de retroação sinusoidal e filtro rejeita-banda para um sinal de entrada com 47 Hz a) Simulação b) Experimental.

Tensão de Entrada e Saída do PLL em Regime Estacionário para 47 Hz



a)



b)

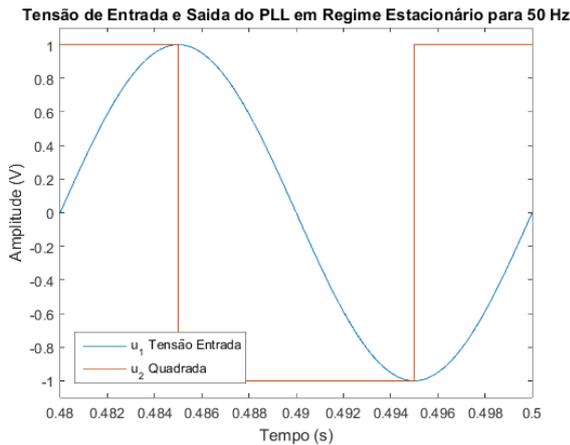
Figura 67 – Tensões de entrada e saídas do SPLL com gerador de sinal ortogonal para um sinal de entrada com 47 Hz a) Simulação b) Experimental.

A partir dos resultados obtidos para 47 Hz de frequência de entrada, pode verificar-se que os resultados das simulações e da implementação são bastante próximos, validando assim o modelo de simulação utilizado.

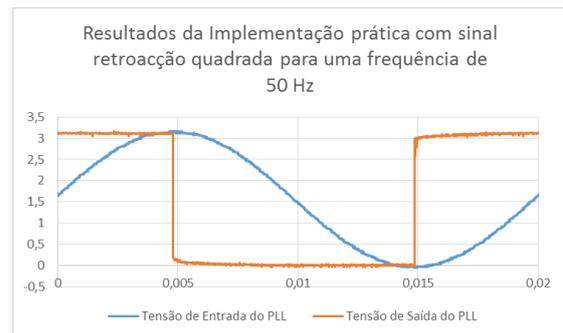
Comparando as respostas às diferentes configurações do PLL verifica-se que nesta situação a melhor resposta é obtida com o filtro rejeita-banda tal como era possível observar já nas simulações.

Resultados para frequência de entrada de 50 Hz

Tal como acima, de seguida são mostrados os resultados da implementação prática, para uma frequência de entrada de 50 Hz para retroação com onda quadrada (Figura 68), sinusoidal sem filtragem da componente de 2ª harmónica (Figura 69), sinusoidal com filtro rejeita-banda (Figura 70) e sinusoidal com gerador de sinais ortogonais (Figura 71).

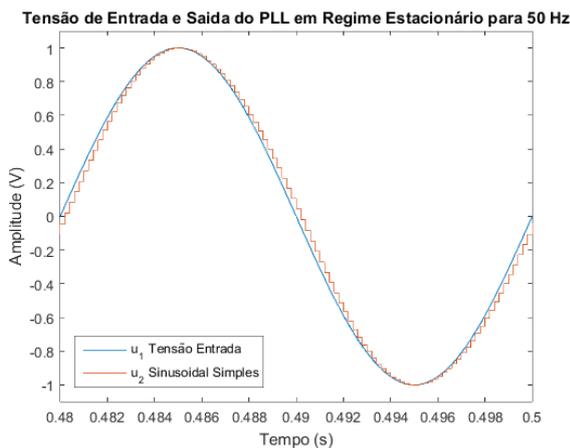


a)

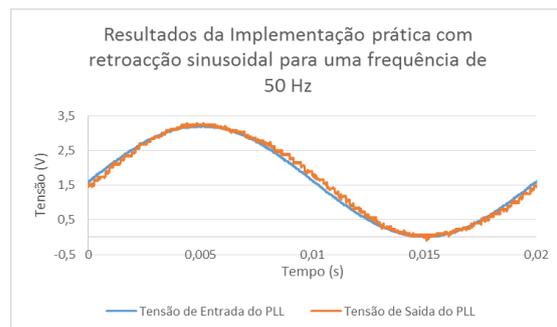


b)

Figura 68 – Tensões de entrada e saídas do com sinal de retroação quadrado para um sinal de entrada com 50 Hz a) Simulação b) Experimental.

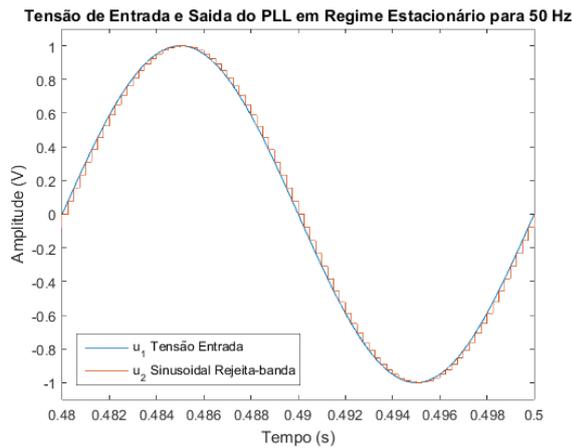


a)

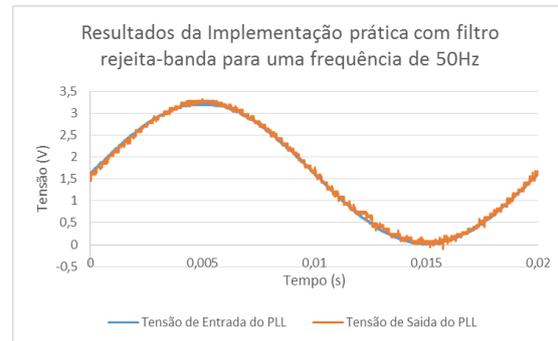


b)

Figura 69 – Tensões de entrada e saídas do SPLL com sinal de retroação sinusoidal para um sinal de entrada com 50 Hz a) Simulação b) Experimental.

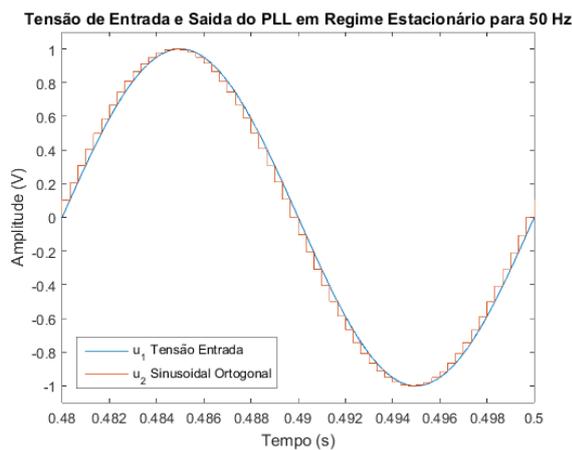


a)

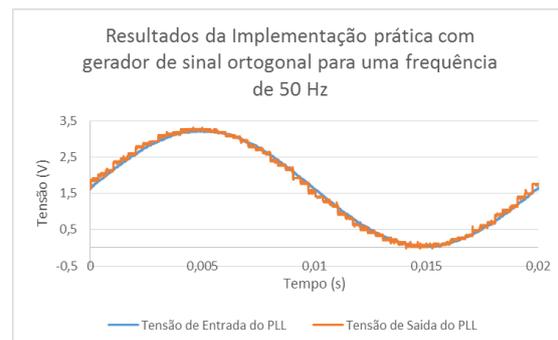


b)

Figura 70 – Tensões de entrada e saídas do SPLL com sinal de retroação sinusoidal e filtro rejeita-banda para um sinal de entrada com 50 Hz a) Simulação b) Experimental.



a)



b)

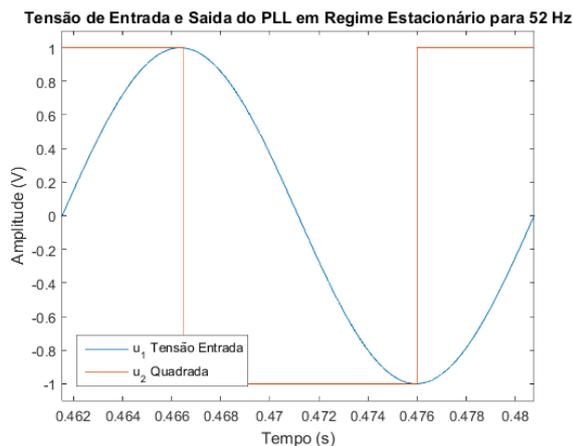
Figura 71 – Tensões de entrada e saídas do SPLL com gerador de sinal ortogonal para um sinal de entrada com 50 Hz a) Simulação b) Experimental.

Fazendo uma rápida comparação dos resultados anteriores, verifica-se que para uma frequência de entrada de 50 Hz a melhoria significativa que era visível entre a implementação com filtro rejeita-banda e com o gerador de sinal ortogonal foi bastante reduzida.

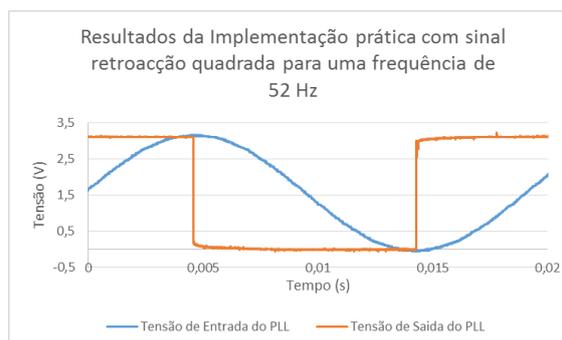
Para esta situação não é possível verificar qual a melhor opção entre a implementação com filtro rejeita-banda e com o gerador de sinal ortogonal, no entanto, observando os resultados acima é possível verificar que o erro do ângulo com o filtro rejeita-banda é inferior ao da implementação com gerador de sinal ortogonal.

Resultados para frequência de entrada de 52 Hz

De seguida são mostrados os resultados da implementação prática, para uma frequência de entrada de 52 Hz para retroação com onda quadrada, sinusoidal sem filtragem da componente de 2ª harmónica, sinusoidal com filtro rejeita-banda e sinusoidal com gerador de sinais ortogonais respetivamente.

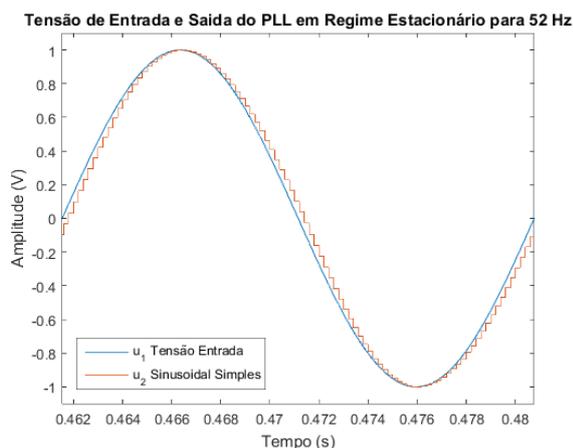


a)

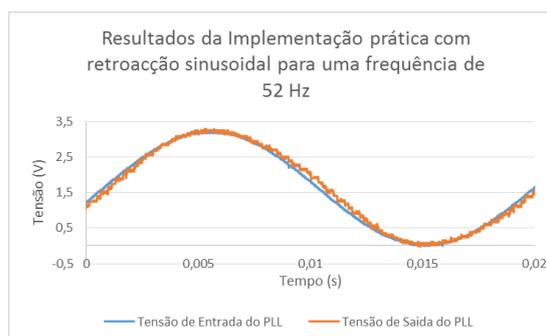


b)

Figura 72 – Tensões de entrada e saídas do com sinal de retroação quadrado para um sinal de entrada com 52 Hz a) Simulação b) Experimental.



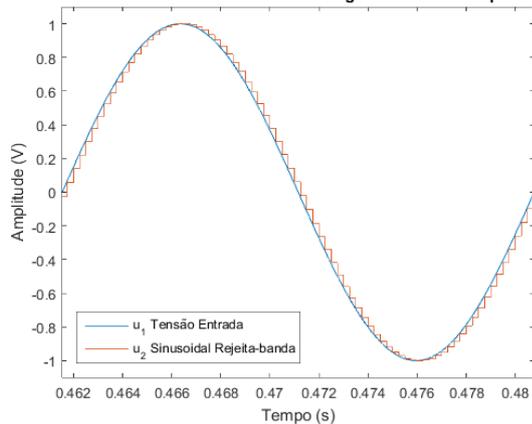
a)



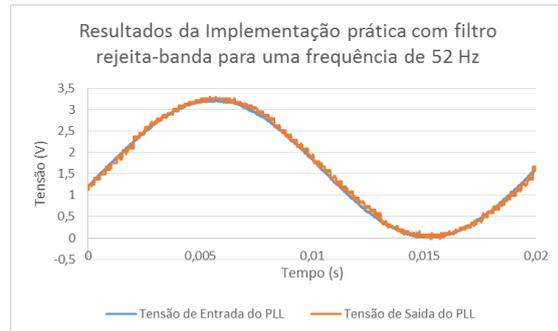
b)

Figura 73 – Tensões de entrada e saídas do SPLL com sinal de retroação sinusoidal para um sinal de entrada com 52 Hz a) Simulação b) Experimental.

Tensão de Entrada e Saída do PLL em Regime Estacionário para 52 Hz



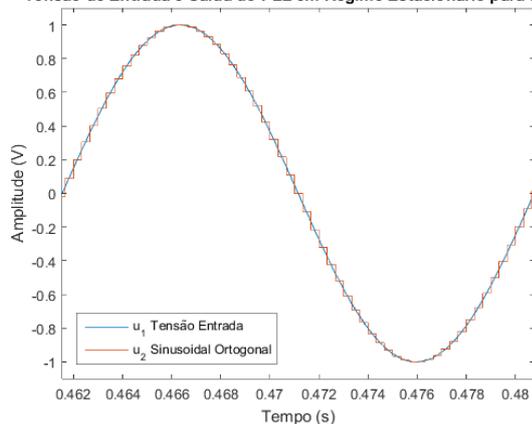
a)



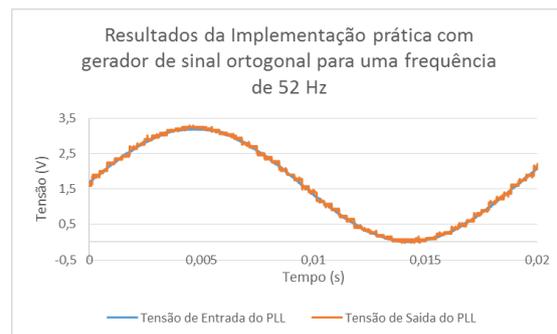
b)

Figura 74 – Tensões de entrada e saídas do SPLL com sinal de retroação sinusoidal e filtro rejeita-banda para um sinal de entrada com 52 Hz a) Simulação b) Experimental.

Tensão de Entrada e Saída do PLL em Regime Estacionário para 52 Hz



a)



b)

Figura 75 – Tensões de entrada e saídas do SPLL com gerador de sinal ortogonal para um sinal de entrada com 52 Hz a) Simulação b) Experimental.

Através da análise dos resultados obtidos para 52 Hz de frequência de entrada, pode verificar-se que nesta situação a melhor resposta é obtida com o gerador de sinal ortogonal. Este facto demonstra que a simulação no domínio contínuo (com maior frequência de aquisição), onde o menor erro era obtido com o filtro rejeita-banda, é ligeiramente diferente da implementação a baixa frequência, onde o menor erro é obtido com o gerador de sinal ortogonal.

Capítulo VI: Conclusões e propostas de trabalho futuras

Resumo:

Neste capítulo são apresentadas todas as conclusões referentes quer aos resultados obtidos em simulação quer através de implementação prática. Depois disto é deixado em aberto um possível estudo futuro para sistemas de compensação de fator de potência, compensação de harmónicas ou simplesmente sobre o funcionamento de conversores com bidirecionalidade de potência.

6.1 – Conclusões

Com as simulações realizadas quer em domínio contínuo quer em domínio discreto foi possível concluir que com a introdução de um filtro rejeita-banda ou um gerador de sinal ortogonal o conteúdo harmónico presente no erro entre o ângulo de entrada e de saída é reduzido substancialmente.

Apesar dos resultados para 47 Hz de frequência de entrada mostrarem maiores erros médios no ângulo com o filtro rejeita-banda e com o gerador de sinal ortogonal do que para a implementação com retroação sinusoidal sem compensação das harmónicas geradas na multiplicação, tal facto pode verificar-se devido à simulação não estar completamente em regime permanente, mas sim ainda em estabilização. Esta estabilização mais demorada pode ser provocada pela existência do filtro adicional ou pelo comportamento do gerador de sinal ortogonal não ser exatamente igual ao comportamento do detetor de fase multiplicador. Contudo tal como já foi mencionado é consideravelmente mais fácil reduzir um erro constante do que um erro variável, logo estas soluções continuam a ser bastante mais razoáveis.

Analisando os resultados obtidos na implementação prática é possível verificar que os resultados são extremamente próximos aos simulados. Apesar disto, verifica-se que para frequências aquisição diferentes, os resultados obtidos podem não ser exatamente iguais.

Verifica-se que em qualquer uma das implementações o erro no ângulo existe mas é satisfatório, no entanto não foi possível com o equipamento existente medir a diferença entre os ângulos de entrada e saída de forma a obter uma medição concreta. A forma de verificação destes erros foi feita através das formas de onda sinusoidais de entrada e saída.

Através da simulação feita com perturbações harmónicas pode verificar-se que este tipo de sistema é bastante imune ao ruído de entrada algo que pode não acontecer com um detetor de passagens por zero.

De um modo geral verificou-se que os problemas mais visíveis acabaram por ser a geração de harmónicas do detetor de fase, que facilmente podem ser compensadas quer com o gerador de sinal ortogonal quer com a adição do filtro rejeita-banda, sendo este último uma melhor opção para frequências de aquisição mais elevadas.

Conclui-se neste trabalho que a implementação de PLLs em *software* encontra-se validada pela igualdade entre os resultados de simulação e os resultados experimentais e que este tipo de sistemas é particularmente robusto para a sincronização de conversores com a rede elétrica.

6.2 – Propostas de trabalho futuro

Com esta dissertação pensa-se deixar em aberto o estudo de um conversor bidirecional em potência e que disponha da possibilidade de compensação de fator de potência ou harmónicas utilizando o PLL como detetor de fase.

Existe também a possibilidade de implementação deste circuito nouro tipo de plataformas mais potentes ao nível de processamento tal como numa FPGA, contudo o processador utilizado revelou um desempenho satisfatório apenas com 80 MIPS.

Além disto, é sempre possível o estudo de um PLL para outro tipo de aplicações que não a sincronização com a rede eléctrica, nomeadamente em aplicações de instrumentação (conversores tensão frequência) e controlo (regulação de velocidade).

Referências bibliográficas

- [1] R. M. d. S. Filho and P. F. S. P. C. Cortizo, "A Comparative Study Of Three-Phase And Single-Phase PII Algorithms For Grid-Connected Systems".
- [2] F. B. Yongheng Yang, "Sincronization in Single-Phase Grid-Connection".
- [3] F. Rodríguez, E. Bueno, M. Aredes, L. Rolim, F. Neves and M. Cavalcanti, "Discrete-time implementation of second order generalized integrators for grid converters," 2008.
- [4] M. Bhardwaj, "Software Phased-Locked Loop Design Using C2000™ Microcontrollers for Single Phase Grid Connected Inverter," *SPRABT3*, July 2013.
- [5] R. E. Best, *Phase-Locked Loops - Design, Simulation and Applications*, United States: McGraw Hill Companies, 2003.
- [6] Philips, "Phase-Locked-Loop with VCO" 74HC/HCT4046 datasheet, November 1997.
- [7] H. Anfíloquio and V. Soares, *Análise e Implementação de Filtros e Controladores num Sistema de Processamento Digital de Sinal*, 2014.
- [8] V. Soares, "Aquisição e Processamento de Sinais," Março 2012.
- [9] A. O. Shuaib and M. M. Ahmed, "Robust PID Control System Design Using ITAE Performance Index (DC Motor Model)," *International Journal of Innovative Research in Science, Engineering and Technology*, vol. III, no. 8, p. 3, 2014.
- [10] Texas Instruments, "High Performance Phase-Locked Loop" TLC2932 datasheet, September 1994 [Revised May 1997].
- [11] D. R. Stephens, *Phase-Locked Loops for Wireless Communications*, Springer US, 2002.

Anexos

Anexo 1 – Verificação matemática do aparecimento da 2ª harmónica na saída do detetor de fase multiplicador.

Anexo 1.1 – Cálculo da tensão u_d e do ganho K_d com tensão de realimentação quadrada

Considerou-se a tensão de entrada $u_1(t)$ uma tensão sinusoidal representada pela equação (63) e $u_2(t)$ uma onda retangular representada pela equação (64).

$$u_1(t) = U_{10} \sin(\omega_1 t + \theta_1) \quad (63)$$

$$u_2(t) = U_{20} \text{rect}(\omega_2 t + \theta_2) \quad (64)$$

Decompondo a equação (64) em séries de Fourier, obtém-se a equação (65).

$$u_2(t) = U_{20} \frac{4}{\pi} \left(\cos(\omega_2 t + \theta_2) + \frac{\cos(3 \omega_2 t + \theta_2)}{3} + \frac{\cos(5 \omega_2 t + \theta_2)}{5} + \dots \right) \quad (65)$$

A equação da tensão de saída do detetor de fase é dada pela equação (66).

$$u_d(t) = u_1(t) u_2(t) \quad (66)$$

Substituindo na equação (66) as tensões $u_1(t)$ e $u_2(t)$ representadas nas equações (63) e (65) respetivamente, obtém-se a equação (67).

$$u_d(t) = U_{10} \sin(\omega_1 t + \theta_1) U_{20} \frac{4}{\pi} \left(\cos(\omega_2 t + \theta_2) + \frac{\cos(3 \omega_2 t + \theta_2)}{3} + \dots \right) \quad (67)$$

Utilizando a relação imediatamente abaixo descrita, obtém-se a equação (69).

$$\sin(a) \cos(b) = \frac{1}{2} (\sin(a - b) + \sin(a + b)) \quad (68)$$

$$u_d(t) = \frac{4 U_{10} U_{20}}{\pi} \left(\sin(\omega_1 t + \theta_1) \cos(\omega_2 t + \theta_2) + \sin(\omega_1 t + \theta_1) \frac{\cos(3 \omega_2 t + \theta_2)}{3} \right) \quad (69)$$

Desenvolvendo, obtém-se:

$$\begin{aligned}
 u_d(t) &= \frac{2 U_{10} U_{20}}{\pi} \left(\sin(\omega_1 t - \omega_1 t + \theta_1 - \theta_2) + \sin(\omega_1 t + \omega_1 t + \theta_1 + \theta_2) \right. \\
 &\quad \left. + \frac{(\sin(\omega_1 t - 3 \omega_1 t + \theta_1 - \theta_2) + \sin(\omega_1 t + 3 \omega_1 t + \theta_1 + \theta_2))}{3} + \dots \right) \quad (70)
 \end{aligned}$$

Considerando que o PLL está sincronizado, verificam-se as duas igualdades abaixo, equações (71) e (72), que quando substituídas na equação (70), dão origem à equação (73), podendo ainda retirar-se o ganho do detetor de fase representado na equação (38)

$$\omega_1 = \omega_2 \quad (71)$$

$$\theta_1 = \theta_2 = 0 \quad (72)$$

$$u_d(t) = \frac{2 U_{10} U_{20}}{\pi} \left(\frac{2 \sin(2 \omega_1 t)}{3} + \frac{\sin(4 \omega_1 t)}{3} + \dots \right) \quad (73)$$

Anexo 1.2 Cálculo da tensão u_d e do ganho K_d com tensão de realimentação sinusoidal

Realizando exatamente o mesmo processo mas para uma tensão de retroação sinusoidal, equação (74), obtém-se a equação (75).

$$u_2(t) = U_{20} \cos(\omega_2 t + \theta_2) \quad (74)$$

$$u_d(t) = \frac{U_{10} U_{20}}{2} (\sin(\omega_1 t + \theta_1 - (\omega_2 t + \theta_2)) + \sin(\omega_1 t + \theta_1 + \omega_2 t + \theta_2)) \quad (75)$$

Considerando que o PLL está sincronizado tal como realizado acima obtém-se a equação (76), sendo o ganho dado pela equação (43).

$$u_d(t) = \frac{U_{10} U_{20}}{2} (\sin(2 \omega_1 t)) \quad (76)$$

Anexo 2 – Código para cálculo e obtenção de figuras do sistema analógico

```

close all
clear all
clc

%%Estudo do PLL de 2ªOrdem para a resposta mais rápida a entrar na banda de 5% do valor
final
% Obtenção do parametro zeta

aux=zeros(2,1); %geração de um arrai auxiliar
wn=2*pi*50/10; %frequência definida para 10 ciclos da rede
matrz=[0.5 0.6 0.7 sqrt(2)/2 0.8 0.9 1]; %matriz com valores de zeta
cores=['r','g','b','c','m','k','r','g','b','c','m','k','r','g','b','c','m','k'];
%array com cores para plot
aux1=1; %variavel auxiliar para verificação de entrada/saida
no intervalo de valores
for i=1:size(matrz,2) %ciclo para ler todos os valores de zeta
    zeta=matrz(i);
    FT1=tf([2*zeta*wn wn^2],[1 2*zeta*wn wn^2]); %geração da FTFC do sistema de segunda
    ordem com um zero em malha fechada
    [y,t]=step(FT1,50); %obtenção da resposta a um escalão (50 segundos de
    cálculo)

    plot(t*wn,y,cores(i)) %grafico do step em função de wnt
    grid on
    hold on

    for j=1:size(y,1) %ciclo para guardar o tempo de estabilização abaixo
    de x%
        if(y(j)>=0.99&&y(j)<=1.01) %verificação da estabilização em 1%
            if (aux1==1) %se se mantiver intervalo não grava
                aux(i)=t(j);
                aux1=0;
            end;
        else %se sair do intervalo permite a escrita novamente
            aux1=1;
        end;
    end;
end;
Legend('\zeta=0.5','\zeta=0.6','\zeta=0.7','\zeta=0.707','\zeta=0.8','\zeta=0.9','\zeta=
1',4) %Legenda
xlim([0 10]); %limite da escala no eixo x (wnt)
xlabel('\omega_n t (rad)') %Etiqueta do Eixo X
[C,I]=min(aux); %Obtem o valor de tempo minimo e a entrada do array
para o tempo minimo

zeta=matrz(I) %Obtem o melhor zeta para a função transferencia e
intervalo estudado

%%Obtenção do Parametro Wn
wnt=C*wn; %Obtem o Wn*t
    
```

```

t=0.2; %Fornece-se o tempo de estabilização pretendido para
1% (10 ciclos da rede)

Wn=wnt/t; %Obtem-se o Wn

FTCA=tf([2*zeta*Wn Wn^2],[1 0 0]) %Obtenção da função transferencia cadeia
aberta
FTCF=tf([2*zeta*Wn Wn^2],[1 2*zeta*Wn Wn^2]); %Obtenção da função transferencia cadeia
fechada

%%Cálculo dos Parametros do PLL
fsaida=50; %igual à frequência de entrada
Vcc=5; %Tensão de alimentação do circuito
f0=102400; %Frequência central do VCO 2048*50
f1=40960; %Variação da frequência central 20*2048 (30 a 50 e 50 a
70 Hz)

U1=5; %Tensão de entrada do PLL entre -5 e 5V
U2=5; %Tensão de realimentação do PLL entre -5 e 5V

Kp=2*U1*U2/(pi); %Determinação do ganho do detetor de fase (para
Multiplicador)

Kv=2*f1*2*pi/((Vcc-0.9)-0.9); %Determinação do ganho do VCO
N=f0/fsaida; %Determinação do divisor de frequência
tao1=Kp*Kv/(N*Wn^2); %Cálculo do tao1
tao2=2*zeta*Wn*N*tao1/(Kp*Kv); %Cálculo do tao2

%Função transterencia do sistema
Sistema=tf([Kp*Kv*tao2/(N*tao1) Kp*Kv/(N*tao1)],[1 Kp*Kv*tao2/(N*tao1) Kp*Kv/(N*tao1)])

%Obtenção das respostas com os valores dos componentes utilizados na
%implementação prática

C1=47e-6; %Considerando C1=47uF
R1=tao1/C1 %Cálculo do R1
R2=tao2/C1 %Cálculo do R2

%Mapa de Polos e zeros cadeia aberta
figure,
[p,z]=pzmap(FTCA);
plot(p,0,'o',z,0,'x')
hold on
plot(get(gca,'xlim'),[0 0], '--')
title('Mapa de Polos e Zeros Cadeia Aberta');
xlabel('Re'); ylabel('Im');

%Mapa de Polos e zeros cadeia aberta
figure, pzmap(FTCF);
title('Mapa de Polos e Zeros Cadeia Fechada');
xlabel('Re'); ylabel('Im');

%Cálculo das bandas de frequência
w3db=Wn*(1+2*zeta^2+sqrt((1+2*zeta^2)^2+1))^(1/2)
D_wmax=(Wn^2)/2
D_wl=2*zeta*Wn
D_wpo=1.8*Wn*(zeta+1)

```

```
f3db=w3db/(2*pi);  
D_fmax=D_wmax/(2*pi);  
D_f1=D_w1/(2*pi);  
D_fpo=D_wpo/(2*pi);  
  
Tp=4*tao1*2*f1/N*N/kv*vcc  
Tl=2*pi/wn;
```

Anexo 3 – Código utilizado para simulação do processamento do PIC

Anexo 3.1 – Código para a realização das respostas do PLL no domínio contínuo

```

close all
clear all
clc

%%Definição do Parametro wn e zeta
wn=26.051850514883434; %Definição do wn
zeta=sqrt(2)/2; %Definição do zeta

%%Definição dos Parametros do PLL
U1=1; %Tensão de entrada do PLL
U2=1; %Tensão de realimentação do PLL
Kp=U1*U2/2; %Determinação do ganho do detetor de fase
Kv=100; %Atribuição do ganho do VCO

tao1=Kp*Kv/(wn^2); %Calculo do tao1
tao2=2*zeta*wn*tao1/(Kp*Kv); %Calculo do tao1

fs=100e3; %Definição da frequência de aquisição do ADC
Ts=1/fs; %Calculo do tempo de aquisição do ADC
w0=2*pi*50; %Definição da frequência da rede electrica para o calculo dos parametros
do gerador ortogonal

%%Calculo dos parametros do filtro discreto
a1=-1;
b0=Ts/(2*tao1)*(1+1/(tan(Ts/(2*tao2))))
b1=Ts/(2*tao1)*(1-1/(tan(Ts/(2*tao2))))
%Geração de um array com os parametros
Filtro=[1 a1;b0 b1];

%%Calculo dos parametros do filtro rejeita banda
f0=50;
LB1=40;
Q=f0/LB1;
b0=(Ts^2*(2*w0)^2+4)/(Ts^2*(2*w0)^2+(1/Q)*2*Ts*(2*w0)+4);
b1=2*(Ts^2*(2*w0)^2-4)/(Ts^2*(2*w0)^2+(1/Q)*2*Ts*(2*w0)+4);
b2=(Ts^2*(2*w0)^2+4)/(Ts^2*(2*w0)^2+(1/Q)*2*Ts*(2*w0)+4);
a1=2*(Ts^2*(2*w0)^2-4)/(Ts^2*(2*w0)^2+(1/Q)*2*Ts*(2*w0)+4);
a2=(Ts^2*(2*w0)^2-(1/Q)*2*Ts*(2*w0)+4)/(Ts^2*(2*w0)^2+(1/Q)*2*Ts*(2*w0)+4);
B_notch=[b0 b1 b2];
A_notch=[1 a1 a2];

Notch=[A_notch;B_notch];

%%Calculo dos parametros do gerador do sinal ortogonal por aproximação trapezoidal
osg_k=2*zeta;
osg_x=2*osg_k*w0*Ts;
osg_y=(w0*w0*Ts*Ts);
osg_b0=osg_x/(osg_x+osg_y+4);

```

```

osg_b2=-1*osg_b0;
osg_a1=(2*(4-osg_y))/(osg_x+osg_y+4);
osg_a2=(osg_x-osg_y-4)/(osg_x+osg_y+4);
osg_qb0=(osg_k*osg_y)/(osg_x+osg_y+4);
osg_qb1=2*osg_qb0;
osg_qb2=osg_qb0;
%Geração de um array com os parametros
ORTO=[osg_b0 osg_b2 0;osg_a1 osg_a2 0;osg_qb0 osg_qb1 osg_qb2];

%%Simulação para frequência de 47Hz
f1=47;
[u1_1_47,u2_1_47,phi2_1_47]=PI(f1,0,Ts,Kp,Filtro,Kv,w0,'square',0);
%Simulação com retroação quadrada
[u1_2_47,u2_2_47,phi2_2_47]=PI(f1,0,Ts,Kp,Filtro,Kv,w0,'sinu',0,0,0);
%Simulação com retroação sinusoidal
[u1_3_47,u2_3_47,phi2_3_47]=PI(f1,0,Ts,Kp,Filtro,Kv,w0,'sinu_notch',0,Notch,0);
%Simulação com filtro rejeita-banda
[u1_4_47,u2_4_47,phi2_4_47]=PI(f1,0,Ts,Kp,Filtro,Kv,w0,'sinu_orto',0,0,ORTO);
%Simulação com gerador de sinal ortogonal

%%Simulação para frequência de 50Hz
f2=50;
[u1_1_50,u2_1_50,phi2_1_50]=PI(f2,0,Ts,Kp,Filtro,Kv,w0,'square',0);
[u1_2_50,u2_2_50,phi2_2_50]=PI(f2,0,Ts,Kp,Filtro,Kv,w0,'sinu',0,0,0);
[u1_3_50,u2_3_50,phi2_3_50]=PI(f2,0,Ts,Kp,Filtro,Kv,w0,'sinu_notch',0,Notch,0);
[u1_4_50,u2_4_50,phi2_4_50]=PI(f2,0,Ts,Kp,Filtro,Kv,w0,'sinu_orto',0,0,ORTO);

%%Simulação para frequência de 52Hz
f3=52;
[u1_1_52,u2_1_52,phi2_1_52]=PI(f3,0,Ts,Kp,Filtro,Kv,w0,'square',0);
[u1_2_52,u2_2_52,phi2_2_52]=PI(f3,0,Ts,Kp,Filtro,Kv,w0,'sinu',0,0,0);
[u1_3_52,u2_3_52,phi2_3_52]=PI(f3,0,Ts,Kp,Filtro,Kv,w0,'sinu_notch',0,Notch,0);
[u1_4_52,u2_4_52,phi2_4_52]=PI(f3,0,Ts,Kp,Filtro,Kv,w0,'sinu_orto',0,0,ORTO);

t=0:Ts:0.5;    %Geração do array de tempo para visualização dos resultados

%%Gráficos das tensões para as diferentes simulações
figure,    %f = 47 Hz
handaxes1 = axes('Position', [0.12 0.12 0.8 0.8]);
plot(t,u1_1_47,t,u2_1_47,t,u2_2_47,t,u2_3_47,t,u2_4_47)
axis([max(t)-0.02 max(t) -1.1 1.1]);
xlabel('Tempo (s)');
ylabel('Amplitude (V)');
title('Tensão de Entrada e Saida do PLL em Regime Estacionário para 47 Hz');
legend('u_1 Tensão Entrada','u_2 Quadrada','u_2 Sinusoidal Simples','u_2 Sinusoidal
Rejeita-banda','u_2 Sinusoidal Ortogonal','Location','Northwest')
set(handaxes1, 'Box', 'off')

handaxes2 = axes('Position', [0.72 0.18 0.2 0.3]);
plot(t,u1_1_47,t,u2_1_47,t,u2_2_47,t,u2_3_47,t,u2_4_47)
line([0 0.5],[0 0],'Color',[0 0 0])
title('Ampliação')
axis([max(t)-0.02+0.0089 max(t)-0.0105 -0.2 0.2]);
    
```

```

set(handaxes2, 'Box', 'on')
set(gca, 'YTickLabel', []);
set(gca, 'XTickLabel', []);
grid on

figure,      %f = 50 Hz
handaxes1 = axes('Position', [0.12 0.12 0.8 0.8]);
plot(t,u1_1_50,t,u2_1_50,t,u2_2_50,t,u2_3_50,t,u2_4_50)
axis([max(t)-0.02 max(t) -1.1 1.1]);
xlabel('Tempo (s)');
ylabel('Amplitude (V)');
title('Tensão de Entrada e Saida do PLL em Regime Estacionário para 50 Hz');
legend('u_1 Tensão Entrada','u_2 Quadrada','u_2 Sinusoidal Simples','u_2 Sinusoidal
Rejeita-banda','u_2 Sinusoidal Ortogonal','Location','Northeast')
set(handaxes1, 'Box', 'off')

handaxes2 = axes('Position', [0.72 0.18 0.2 0.3]);
plot(t,u1_1_50,t,u2_1_50,t,u2_2_50,t,u2_3_50,t,u2_4_50)
line([0 0.5],[0 0], 'Color',[0 0 0])
title('Ampliação')
axis([max(t)-0.02+0.0099 max(t)-0.0099 -0.1 0.1]);

set(handaxes2, 'Box', 'on')
set(gca, 'YTickLabel', []);
set(gca, 'XTickLabel', []);
grid on

figure,      %f = 52 Hz
handaxes1 = axes('Position', [0.12 0.12 0.8 0.8]);
plot(t,u1_1_52,t,u2_1_52,t,u2_2_52,t,u2_3_52,t,u2_4_52)
axis([max(t)-0.02 max(t) -1.1 1.1]);
xlabel('Tempo (s)');
ylabel('Amplitude (V)');
title('Tensão de Entrada e Saida do PLL em Regime Estacionário para 52 Hz');
legend('u_1 Tensão Entrada','u_2 Quadrada','u_2 Sinusoidal Simples','u_2 Sinusoidal
Rejeita-banda','u_2 Sinusoidal Ortogonal','Location','Northeast')
set(handaxes1, 'Box', 'off')

handaxes2 = axes('Position', [0.72 0.18 0.2 0.3]);
plot(t,u1_1_52,t,u2_1_52,t,u2_2_52,t,u2_3_52,t,u2_4_52)
line([0 0.5],[0 0], 'Color',[0 0 0])
title('Ampliação')
axis([max(t)-0.02+0.0103 max(t)-0.00955 -0.1 0.1]);

set(handaxes2, 'Box', 'on')
set(gca, 'YTickLabel', []);
set(gca, 'XTickLabel', []);
grid on

%%Obtenção dos ângulos
%Transferencia dos nomes das variaveis phi2 para teta para 47Hz
tetaref_47=(2*pi*47*t);
teta1_47=(phi2_1_47-pi/2);
teta2_47=(phi2_2_47);
teta3_47=(phi2_3_47);

```

```
teta4_47=(phi2_4_47);
%Transferencia dos nomes das variaveis phi2 para teta para 50Hz
tetaref_50=(2*pi*50*t);
teta1_50=(phi2_1_50-pi/2);
teta2_50=(phi2_2_50);
teta3_50=(phi2_3_50);
teta4_50=(phi2_4_50);
%Transferencia dos nomes das variaveis phi2 para teta para 52Hz
tetaref_52=(2*pi*52*t);
teta1_52=(phi2_1_52-pi/2);
teta2_52=(phi2_2_52);
teta3_52=(phi2_3_52);
teta4_52=(phi2_4_52);

%Geração das figuras do erro em regime transitório
figure,
plot(t,(tetaref_47-teta1_47)*180/pi,t,(tetaref_47-teta2_47)*180/pi,t,(tetaref_47-
teta3_47)*180/pi,t,(tetaref_47-teta4_47)*180/pi)
legend('Quadrada','Sinusoidal Simples','Sinusoidal Rejeita-banda','Sinusoidal
Ortogonal')
axis([0 0.5 -50 50]);
xlabel('Tempo (seg)');
ylabel('\theta_e_r_r_o (Graus)');
title('Erro no ângulo em regime transitório para 47 Hz')
grid on

figure,
plot(t,(tetaref_50-teta1_50)*180/pi,t,(tetaref_50-teta2_50)*180/pi,t,(tetaref_50-
teta3_50)*180/pi,t,(tetaref_50-teta4_50)*180/pi)
legend('Quadrada','Sinusoidal Simples','Sinusoidal Rejeita-banda','Sinusoidal
Ortogonal')
axis([0 0.5 -50 50]);
xlabel('Tempo (seg)');
ylabel('\theta_e_r_r_o (Graus)');
title('Erro no ângulo em regime transitório para 50 Hz')
grid on

figure,
plot(t,(tetaref_52-teta1_52)*180/pi,t,(tetaref_52-teta2_52)*180/pi,t,(tetaref_52-
teta3_52)*180/pi,t,(tetaref_52-teta4_52)*180/pi)
legend('Quadrada','Sinusoidal Simples','Sinusoidal Rejeita-banda','Sinusoidal
Ortogonal')
axis([0 0.5 -50 50]);
xlabel('Tempo (seg)');
ylabel('\theta_e_r_r_o (Graus)');
title('Erro no ângulo em regime transitório para 52 Hz')
grid on

%Geração das figuras do erro em regime estacionário ao longo de 1 periodo
figure,
plot(t,(tetaref_47-teta1_47)*180/pi,t,(tetaref_47-teta2_47)*180/pi,t,(tetaref_47-
teta3_47)*180/pi,t,(tetaref_47-teta4_47)*180/pi)
legend('Quadrada','Sinusoidal Simples','Sinusoidal Rejeita-banda','Sinusoidal
Ortogonal')
axis([22/47 23/47 -10 10]);
xlabel('Tempo (seg)');
```

```

ylabel('\theta_e_r_r_o (Graus)');
title({'Erro no ângulo em regime estacionário para 47 Hz',' ',' ',' '})
grid on
ax1 = gca; % current axes
ax1_pos = ax1.Position; % position of first axes
ax2 = axes('Position',ax1_pos,...
    'XAxisLocation','top',...
    'Color','none');

line([0,0],[0,0],'Parent',ax2,'Color','k')
axis([0 2*pi -40 40]);
set(gca,'XTick',[0,pi/2,pi,3*pi/2,2*pi]);
set(gca,'XTickLabel',{'0','\pi/2','\pi','2\pi/3','2\pi'});
set(gca,'YTick',[]);
xlabel('\omega_nt (rad)');

figure,
plot(t,(tetaref_50-teta1_50)*180/pi,t,(tetaref_50-teta2_50)*180/pi,t,(tetaref_50-
teta3_50)*180/pi,t,(tetaref_50-teta4_50)*180/pi)
legend('Quadrada','Sinusoidal Simples','Sinusoidal Rejeita-banda','Sinusoidal
Ortogonal')
axis([0.48 0.5 -10 10]);
xlabel('Tempo (seg)');
ylabel('\theta_e_r_r_o (Graus)');
title({'Erro no ângulo em regime estacionário para 50 Hz',' ',' ',' '})
grid on
ax1 = gca; % current axes
ax1_pos = ax1.Position; % position of first axes
ax2 = axes('Position',ax1_pos,...
    'XAxisLocation','top',...
    'Color','none');

line([0,0],[0,0],'Parent',ax2,'Color','k')
axis([0 2*pi -40 40]);
set(gca,'XTick',[0,pi/2,pi,3*pi/2,2*pi]);
set(gca,'XTickLabel',{'0','\pi/2','\pi','2\pi/3','2\pi'});
set(gca,'YTick',[]);
xlabel('\omega_nt (rad)');

figure,
plot(t,(tetaref_52-teta1_52)*180/pi,t,(tetaref_52-teta2_52)*180/pi,t,(tetaref_52-
teta3_52)*180/pi,t,(tetaref_52-teta4_52)*180/pi)
legend('Quadrada','Sinusoidal Simples','Sinusoidal Rejeita-banda','Sinusoidal
Ortogonal')
axis([24/52 25/52 -10 10]);
xlabel('Tempo (seg)');
ylabel('\theta_e_r_r_o (Graus)');
title({'Erro no ângulo em regime estacionário para 52 Hz',' ',' ',' '})
grid on
ax1 = gca; % current axes
ax1_pos = ax1.Position; % position of first axes
ax2 = axes('Position',ax1_pos,...
    'XAxisLocation','top',...
    'Color','none');

line([0,0],[0,0],'Parent',ax2,'Color','k')

```

```
axis([0 2*pi -40 40]);  
set(gca, 'XTick', [0, pi/2, pi, 3*pi/2, 2*pi]);  
set(gca, 'XTickLabel', {'0', '\pi/2', '\pi', '2\pi/3', '2\pi'});  
set(gca, 'YTick', []);  
xlabel('\omega_{nt} (rad)');
```

Anexo 3.2 – Código para a realização das respostas do PLL no domínio discreto

```
close all  
clear all  
clc  
  
feature('SetPrecision', 24); %Define as variaveis do tipo float  
  
%%Definição do Parametro Wn e zeta  
wn=26.051850514883434; %Definição do wn  
zeta=sqrt(2)/2; %Definição do zeta  
  
%%Definição dos Parametros do PLL  
U1=1; %Tensão de entrada do PLL  
U2=1; %Tensão de realimentação do PLL  
  
Kp_square=2*U1*U2/pi; %Determinação do ganho do detetor de fase para retroacção quadrada  
Kp=U1*U2/2; %Determinação do ganho do detetor de fase para retroacção sinusoidal  
Kv=100; %Atribuição do ganho do VCO  
  
tao1=Kp*Kv/(wn^2); %Calculo do tao1  
tao2=2*zeta*wn*tao1/(Kp*Kv); %Calculo do tao1  
  
w0=2*pi*50; %Definição da frequência da rede electrica para o calculo dos parametros  
do gerador ortogonal  
  
%%Calculo dos parametros do filtro discreto com retroacção sinusoidal  
Ts_square=1/(6e3);  
a1=-1;  
b0=Ts_square/(2*tao1)*(1+1/(tan(Ts_square/(2*tao2))))  
b1=Ts_square/(2*tao1)*(1-1/(tan(Ts_square/(2*tao2))))  
  
%Geração de um array com os parametros  
Filtro_square=[1 a1;b0 b1];  
  
%%Calculo dos parametros do filtro discreto com retroacção sinusoidal  
Ts_sinu=1/(5e3);  
a1=-1;  
b0=Ts_sinu/(2*tao1)*(1+1/(tan(Ts_sinu/(2*tao2))))  
b1=Ts_sinu/(2*tao1)*(1-1/(tan(Ts_sinu/(2*tao2))))  
  
%Geração de um array com os parametros  
Filtro_sinu=[1 a1;b0 b1];  
  
%%Calculo dos parametros do filtro discreto com retroacção sinusoidal e
```

```

%%rejeitabanda
Ts_notch=1/(4e3);
a1=-1;
b0=Ts_notch/(2*tao1)*(1+1/(tan(Ts_notch/(2*tao2))))
b1=Ts_notch/(2*tao1)*(1-1/(tan(Ts_notch/(2*tao2))))

%Geração de um array com os parametros
Filtro_notch=[1 a1;b0 b1];

%%Calculo dos parametros do filtro discreto para o gerador de sinal
%%ortogonal
Ts_orto=1/(3e3);
a1=-1;
b0=Ts_orto/(2*tao1)*(1+1/(tan(Ts_orto/(2*tao2))))
b1=Ts_orto/(2*tao1)*(1-1/(tan(Ts_orto/(2*tao2))))

%Geração de um array com os parametros
Filtro_orto=[1 a1;b0 b1];

%%Calculo dos parametros do filtro rejeita banda
f0=50;
LB1=40;
Q=f0/LB1;
b0=(Ts_notch^2*(2*w0)^2+4)/(Ts_notch^2*(2*w0)^2+(1/Q)*2*Ts_notch*(2*w0)+4);
b1=2*(Ts_notch^2*(2*w0)^2-4)/(Ts_notch^2*(2*w0)^2+(1/Q)*2*Ts_notch*(2*w0)+4);
b2=(Ts_notch^2*(2*w0)^2+4)/(Ts_notch^2*(2*w0)^2+(1/Q)*2*Ts_notch*(2*w0)+4);
a1=2*(Ts_notch^2*(2*w0)^2-4)/(Ts_notch^2*(2*w0)^2+(1/Q)*2*Ts_notch*(2*w0)+4);
a2=(Ts_notch^2*(2*w0)^2-
(1/Q)*2*Ts_notch*(2*w0)+4)/(Ts_notch^2*(2*w0)^2+(1/Q)*2*Ts_notch*(2*w0)+4);
B_notch=[b0 b1 b2];
A_notch=[1 a1 a2];

Notch=[A_notch;B_notch];

%%Calculo dos parametros do gerador do sinal ortogonal por aproximação trapezoidal
osg_k=2*zeta
osg_x=2*osg_k*w0*Ts_orto
osg_y=(w0*w0*Ts_orto*Ts_orto)
osg_b0=osg_x/(osg_x+osg_y+4)
osg_b2=-1*osg_b0
osg_a1=(2*(4-osg_y))/(osg_x+osg_y+4)
osg_a2=(osg_x-osg_y-4)/(osg_x+osg_y+4)
osg_qb0=(osg_k*osg_y)/(osg_x+osg_y+4)
osg_qb1=2*osg_qb0
osg_qb2=osg_qb0
%Geração de um array com os parametros
ORTO=[osg_b0 osg_b2 0;osg_a1 osg_a2 0;osg_qb0 osg_qb1 osg_qb2];

%%Simulação para frequência de 47Hz
f1=47;
[u1_1_47,u2_1_47,phi2_1_47]=PI(f1,0,Ts_square,Kp_square,Filtro_square,Kv,w0,'square',0);
%Simulação com retroação quadrada
[u1_2_47,u2_2_47,phi2_2_47]=PI(f1,0,Ts_sinu,Kp,Filtro_sinu,Kv,w0,'sinu',0,0,0);
%Simulação com retroação sinusoidal
[u1_3_47,u2_3_47,phi2_3_47]=PI(f1,0,Ts_notch,Kp,Filtro_notch,Kv,w0,'sinu_notch',0,Notch,
0); %Simulação com filtro rejeita-banda

```

```

[u1_4_47,u2_4_47,phi2_4_47]=PI(f1,0,Ts_orto,Kp,Filtro_orto,Kv,w0,'sinu_orto',0,0,ORTO);
%%Simulação com gerador de sinal ortogonal

%%Simulação para frequencia de 50Hz
f2=50;
[u1_1_50,u2_1_50,phi2_1_50]=PI(f2,0,Ts_square,Kp_square,Filtro_square,Kv,w0,'square',0);
[u1_2_50,u2_2_50,phi2_2_50]=PI(f2,0,Ts_sinu,Kp,Filtro_sinu,Kv,w0,'sinu',0,0,0);
[u1_3_50,u2_3_50,phi2_3_50]=PI(f2,0,Ts_notch,Kp,Filtro_notch,Kv,w0,'sinu_notch',0,Notch,0);
[u1_4_50,u2_4_50,phi2_4_50]=PI(f2,0,Ts_orto,Kp,Filtro_orto,Kv,w0,'sinu_orto',0,0,ORTO);

%%Simulação para frequencia de 52Hz
f3=52;
[u1_1_52,u2_1_52,phi2_1_52]=PI(f3,0,Ts_square,Kp_square,Filtro_square,Kv,w0,'square',0);
[u1_2_52,u2_2_52,phi2_2_52]=PI(f3,0,Ts_sinu,Kp,Filtro_sinu,Kv,w0,'sinu',0,0,0);
[u1_3_52,u2_3_52,phi2_3_52]=PI(f3,0,Ts_notch,Kp,Filtro_notch,Kv,w0,'sinu_notch',0,Notch,0);
[u1_4_52,u2_4_52,phi2_4_52]=PI(f3,0,Ts_orto,Kp,Filtro_orto,Kv,w0,'sinu_orto',0,0,ORTO);

t_square=0:Ts_square:0.5;    %Geração do array de tempo para vizualização dos resultados
t_sinu=0:Ts_sinu:0.5;    %Geração do array de tempo para vizualização dos resultados
t_notch=0:Ts_notch:0.5;
t_orto=0:Ts_orto:0.5;

%%Gráficos das tensões para as diferentes simulações
figure,    %f = 47 Hz
handaxes1 = axes('Position', [0.12 0.12 0.8 0.8]);
plot(t_sinu,u1_2_47)
hold on
stairs(t_square,u2_1_47)
hold on
stairs(t_sinu,u2_2_47)
hold on
stairs(t_notch,u2_3_47)
hold on
stairs(t_orto,u2_4_47)
axis([max(t_sinu)-0.02 max(t_sinu) -1.1 1.1]);
xlabel('Tempo (s)');
ylabel('Amplitude (V)');
title('Tensão de Entrada e Saida do PLL em Regime Estacionário para 47 Hz');
legend('u_1 Tensão Entrada','u_2 Quadrada','u_2 Sinusoidal Simples','u_2 Sinusoidal Rejeita-banda','u_2 Sinusoidal Ortogonal','Location','Northwest')
set(handaxes1, 'Box', 'off')

handaxes2 = axes('Position', [0.72 0.18 0.2 0.3]);
plot(t_sinu,u1_2_47)
hold on
stairs(t_square,u2_1_47)
hold on
stairs(t_sinu,u2_2_47)
hold on
stairs(t_notch,u2_3_47)
hold on
stairs(t_orto,u2_4_47)
line([0 0.5],[0 0], 'Color',[0 0 0])
    
```

```

title('Ampliação')
axis([max(t_sinu)-0.02+0.0089 max(t_sinu)-0.0105 -0.2 0.2]);

set(handaxes2, 'Box', 'on')
set(gca, 'YTickLabel', []);
set(gca, 'XTickLabel', []);

figure, %f = 50 Hz
handaxes1 = axes('Position', [0.12 0.12 0.8 0.8]);
plot(t_sinu,u1_2_50)
hold on
stairs(t_square,u2_1_50)
hold on
stairs(t_sinu,u2_2_50)
hold on
stairs(t_notch,u2_3_50)
hold on
stairs(t_orto,u2_4_50)
axis([max(t_sinu)-0.02 max(t_sinu) -1.1 1.1]);
xlabel('Tempo (s)');
ylabel('Amplitude (V)');
title('Tensão de Entrada e Saida do PLL em Regime Estacionário para 50 Hz');
legend('u_1 Tensão Entrada', 'u_2 Quadrada', 'u_2 Sinusoidal Simples', 'u_2 Sinusoidal Rejeita-banda', 'u_2 Sinusoidal Ortogonal', 'Location', 'Northeast')
set(handaxes1, 'Box', 'off')

handaxes2 = axes('Position', [0.72 0.18 0.2 0.3]);
plot(t_sinu,u1_2_50)
hold on
stairs(t_square,u2_1_50)
hold on
stairs(t_sinu,u2_2_50)
hold on
stairs(t_notch,u2_3_50)
hold on
stairs(t_orto,u2_4_50)
line([0 0.5],[0 0], 'Color',[0 0 0])
title('Ampliação')
axis([max(t_sinu)-0.02+0.0099 max(t_sinu)-0.0099 -0.1 0.1]);

set(handaxes2, 'Box', 'on')
set(gca, 'YTickLabel', []);
set(gca, 'XTickLabel', []);

figure, %f = 52 Hz
handaxes1 = axes('Position', [0.12 0.12 0.8 0.8]);
plot(t_sinu,u1_2_52)
hold on
stairs(t_square,u2_1_52)
hold on
stairs(t_sinu,u2_2_52)
hold on
stairs(t_notch,u2_3_52)
hold on
stairs(t_orto,u2_4_52)
axis([max(t_sinu)-0.02 max(t_sinu) -1.1 1.1]);
xlabel('Tempo (s)');

```

```

ylabel('Amplitude (V)');
title('Tensão de Entrada e Saida do PLL em Regime Estacionário para 52 Hz');
legend('u_1 Tensão Entrada','u_2 Quadrada','u_2 Sinusoidal Simples','u_2 Sinusoidal
Rejeita-banda','u_2 Sinusoidal Ortogonal','Location','Northeast')
%legend('u_1','u_2 Sinusoidal','u_2 Rejeita-banda','u_2
Ortogonal','Location','Northeast')
set(handaxes1, 'Box', 'off')

handaxes2 = axes('Position', [0.72 0.18 0.2 0.3]);
plot(t_sinu,u1_2_52)
hold on
stairs(t_square,u2_1_52)
hold on
stairs(t_sinu,u2_2_52)
hold on
stairs(t_notch,u2_3_52)
hold on
stairs(t_orto,u2_4_52)
line([0 0.5],[0 0], 'Color',[0 0 0])
title('Ampliação')
axis([max(t_sinu)-0.02+0.0103 max(t_sinu)-0.00955 -0.1 0.1]);

set(handaxes2, 'Box', 'on')
set(gca, 'YTickLabel', []);
set(gca, 'XTickLabel', []);
    
```

Anexo 3.3 – Função auxiliar com implementação do algoritmo do PLL

```

function [voltage1,voltage2out,phi2funcout]=
PI(f,phi,Tsample,Kdet,Filt,Kvco,w0func,teste,ruido,rejeita-banda,orto)

tempo=0:Tsample:0.5; %Criação do array de tempo que simula o tempo de
aquisição do microcontrolador
if ruido==0
    voltage1=sin(2*pi*f*tempo+phi); %Criação do array de tensão de entrada,
simulando a aquisição do ADC, sem harmónicas
elseif ruido==1
    voltage1=sin(2*pi*f*tempo+phi)+... %Criação do array de tensão de entrada,
simulando a aquisição do ADC, com harmonicas segundo a norma
    0.05*sin(3*2*pi*f*freq*t)+...
    0.06*sin(5*2*pi*f*freq*t)+...
    0.05*sin(7*2*pi*f*freq*t);
end;

%Inicialização de variáveis utilizadas na simulação
voltage2=0;
phi2func=0;
ud=[0,0,0];
uf=[0,0];
b=0;
uaux=[0,0,0];
osg_u=[0,0,0,0,0,0];
    
```

```

osg_qu=[0,0,0,0,0,0];
Mysin=[0,0,0,0,0];
Mycos=[1,1,1,1,1];

%Ciclo que simula o comportamento do microcontrolador (não tem em conta a capacidade de
bits processados pelo microcontrolador)
for i=1:length(tempo)

    if strcmp(teste,'sinu_orto')          %Verifica que simulação é pretendida
        %Gerador do sinal ortogonal
        try
            osg_u(1)=(orto(1,1)*(voltage1(i)-voltage1(i-
2)))+orto(2,1)*osg_u(2)+orto(2,2)*osg_u(3);
            osg_u(3)=osg_u(2);
            osg_u(2)=osg_u(1);
            osg_qu(1)=(orto(3,1)*voltage1(i)+orto(3,2)*voltage1(i-1)+orto(3,3)*voltage1(i-
2))+orto(2,1)*osg_qu(2)+orto(2,2)*osg_qu(3);
            osg_qu(3)=osg_qu(2);
            osg_qu(2)=osg_qu(1);
            %Transformada de Park de alfa-beta para d-q
            Uaux(1)=(Mycos(2)*osg_u(1)+Mysin(2)*osg_qu(1));
            end;
        else
            %Obtenção da tensão de saída do detetor de fase
            Ud(1)=kdet*voltage1(i)*voltage2;

            if strcmp(teste,'sinu_rejeita-banda')          %Verifica que simulação é pretendida
                %Cálculo realizado com filtro rejeita-banda
                Uaux(1)=-rejeita-banda(1,2)*Uaux(2)-rejeita-banda(1,3)*Uaux(3)+rejeita-
banda(2,1)*Ud(1)+rejeita-banda(2,2)*Ud(2)+rejeita-banda(2,3)*Ud(3);
            else
                %Cálculo realizado sem filtro rejeita-banda
                Uaux(1)=Ud(1);
            end;
        end;
        Uf(1)=-Filt(1,2)*Uf(2)+Filt(2,1)*Uaux(1)+Filt(2,2)*Uaux(2);          %cálculo da tensão
de saída do filtro PI
        phi2funcnm1=phi2func+(w0func+Kvco*Uf(1))*Tsample;          %obtenção do ângulo (DCO)
        if(i>1)
            phi2funcout(i)=phi2funcout(i-1)+(w0func+Kvco*Uf(1))*Tsample;
        else
            phi2funcout(i)=phi2funcnm1;
        end;
        while(phi2funcnm1>(pi))          %Ciclo para controlo do crescimento do ângulo
            phi2funcnm1=phi2funcnm1-2*pi;
        end;
        if strcmp(teste,'sinu') || strcmp(teste,'sinu_rejeita-banda') ||
strcmp(teste,'sinu_orto')          %Verifica que simulação é pretendida
            %Geração da tensão de retroacção sinusoidal
            voltage2=cos(phi2funcnm1);
            voltage2out(i)=sin(phi2funcnm1);
        end;

        if strcmp(teste,'square')
            %Geração da tensão de retroacção quadrada
            if(phi2funcnm1>0)
                voltage2 = 1;
            end;
        end;
    end;
end;

```

```
else
    voltage2=-1;
end;
voltage2out(i)=voltage2;
end;

%Colocação de todas as variaveis para n-1 para repetição do cálculo
Ud(3)=Ud(2);
Ud(2)=Ud(1);
Uf(2)=Uf(1);
Uaux(3)=Uaux(2);
Uaux(2)=Uaux(1);
phi2func=phi2funcnm1;
Mysin(1)=voltage2out(i);
Mycos(1)=voltage2;
Mysin(2)=Mysin(1);
Mycos(2)=Mycos(1);

end;
```

Anexo 4 – Código exemplo utilizado na implementação com microcontrolador

```
#include <xc.h>
#include <math.h>
#include <sys/attribs.h>

#pragma config FPLLIDIV = DIV_2          // PLL Input Divider
#pragma config FPLLODIV = DIV_1         // PLL Output Divider
#pragma config FPBDIV = DIV_1           // Peripheral Clock divisor
#pragma config FWDTEN = OFF             // Watchdog Timer
#pragma config POSCMOD = OFF            // Primary Oscillator
#pragma config FNOSC = PRIPLL           // Oscillator Selection
#pragma config FPLLMUL = MUL_20        // PLL Multiplier
// DEVCFG0
#pragma config JTAGEN = OFF             // JTAG Enable (JTAG Disabled)

void Init(); //Função de configuração inicial

int main() {

    Init();

    //Arrays com funções Seno e Coseno para a retroacção e para a saída do PLL
    float Coseno[81] = {-1, -0.99692, -0.98769, -0.97237, -0.95106, -0.92388, -0.89101,
-0.85264, -0.80902, -0.76041, -0.70711, -0.64945, -0.58779, -0.5225, -0.45399, -0.38268,
-0.30902, -0.23345, -0.15643, -0.078459, 6.1232e-17, 0.078459, 0.15643, 0.23345,
0.30902, 0.38268, 0.45399, 0.5225, 0.58779, 0.64945, 0.70711, 0.76041, 0.80902, 0.85264,
0.89101, 0.92388, 0.95106, 0.97237, 0.98769, 0.99692, 1, 0.99692, 0.98769, 0.97237,
0.95106, 0.92388, 0.89101, 0.85264, 0.80902, 0.76041, 0.70711, 0.64945, 0.58779, 0.5225,
0.45399, 0.38268, 0.30902, 0.23345, 0.15643, 0.078459, 6.1232e-17, -0.078459, -0.15643,
-0.23345, -0.30902, -0.38268, -0.45399, -0.5225, -0.58779, -0.64945, -0.70711, -0.76041,
-0.80902, -0.85264, -0.89101, -0.92388, -0.95106, -0.97237, -0.98769, -0.99692, -1};

    float Seno[81] = {1024, 944, 864, 785, 708, 632, 559, 489, 422, 359, 300, 245, 196,
151, 112, 78, 50, 28, 13, 3, 0, 3, 13, 28, 50, 78, 112, 151, 196, 245, 300, 359, 422,
489, 559, 632, 708, 785, 864, 944, 1024, 1104, 1184, 1263, 1340, 1416, 1489, 1559, 1626,
1689, 1748, 1803, 1852, 1897, 1936, 1970, 1998, 2020, 2035, 2045, 2048, 2045, 2035,
2020, 1998, 1970, 1936, 1897, 1852, 1803, 1748, 1689, 1626, 1559, 1489, 1416, 1340,
1263, 1184, 1104, 1024};

    float wn=26.051850515; //Definição do wn
    float zeta=0.70710678; //Definição do zeta

    //Definição dos Parametros do PLL
    float U1 = 1.0; //Tensão de entrada do PLL
    float U2 = 1.0; //Tensão de realimentação do PLL

    float Kp=U1*U2/2; //Determinação do ganho do detetor de fase
    float kv = 100.0; //Atribuição do ganho do vco
```

```
float tao1=Kp*Kv/(Wn*Wn);           //Calculo do tao1
float tao2=2*zeta*Wn*tao1/(Kp*Kv);  //Calculo do tao1

float fs = 6e3; //Definição da frequência de aquisição do ADC
float Ts = 1.0 / fs; //Calculo do tempo de aquisição do ADC
float w0 = 2.0 * 3.14159 * 50; //Definição da frequência da rede electrica para o
calculo dos parametros do gerador ortogonal

//Calculo dos parametros do filtro discreto
float b0=Ts/(2*tao1)*(1+1/(tan(Ts/(2*tao2))));
float b1=Ts/(2*tao1)*(1-1/(tan(Ts/(2*tao2))));

//Inicialização das variáveis a utilizar nos calculos
float Ud = 0;
float Uf = 0;
float Udn_1 = 0;
float Ufn_1 = 0;
float phi2 = 0;
float phi2nm1 = 0;
int teta=0;
float Uout = 0;
unsigned int auxsend=0;

while (1) //Repetição infinita
{
    while (TMR2 < 1333); //Espera que o TMR2 tenha o valor 1333 (166,6 us; 6 kHz)
    TMR2 = 0;           //Limpa o Timer

    while (!(AD1CON1 & 0x0001)); // Verifica se a conversão do ADC já foi feita

    U1 = ((float) (ADC1BUF0) - 512.0) / 512.0; //Obtenção da entrada analógica
    Ud = U1*U2;                               //Calculo da tensão de erro
    Uf = Ufn_1 + b0 * Ud + b1*Udn_1;          //Calculo da tensão de saída do
filtro PI
    phi2nm1 = phi2 + (w0 + Kv * Uf) * Ts;     //Calculo do ângulo teta de saída do
PLL

    //Ciclo para verificação do ângulo entre -pi e pi
    while (phi2nm1 > 3.14159)
        phi2nm1 = phi2nm1 - 6.28318;

    teta=phi2nm1*12.732406+40; //Calculo de um valor auxiliar para aceder ao array
de seno e coseno
}
```

```
//verificação para protecção de erros por indices do array desconhecidos
if (teta>80)
    teta=80;
    if (teta<0)
        teta=0;

Uout =Seno[teta]; //Obtenção do valor da Tensão de Saida

auxsend= ((int)(Uout) & 0x0FFF) + 0x3000; // Variavel auxiliar para envio de
Informação para MCP4922
SPI1BUF=auxsend; //Envio dos dados da tensão de saida

auxsend= ((int) (phi2nm1 * 256.0 + 1024.0) & 0x0FFF) + 0xB000; // variavel
auxiliar para envio de Informação para MCP4922
SPI1BUF=auxsend;//Envio dos dados do ângulo de saida

U2 =Coseno[teta]; //Obtenção da tensão de retroacção

//Passagem dos parametros n para n-1
Udn_1 = Ud;
Ufn_1 = Uf;
phi2 = phi2nm1;

//Inicio de uma nova conversão
AD1CON1SET = 0x0002; // Inicio da amostragem
asm("nop");
AD1CON1CLR = 0x0002; // Inicio da conversão

}
return 0;
}

void Init() {

    int rData;

    //Configuração do ADC
    ANSELA = 0xFFFE; //Define todas as portas como digitais excepto o pino AN0
    AD1CON1 = 0x0000; // Reinicia todos os bits de configuração do ADC
    AD1CHS = 0x00000000; // Conecta RA0/AN0 ao CH0

    AD1CSSL = 0;
    AD1CON3 = 0x0002; // Aquisição manual
```

```
AD1CON2 = 0;      //Limpa o registo ADCON2
AD1CON1SET = 0x8000; // Liga o módulo do ADC

//Configuração do SPI
SPI1CON = 0; // Limpa o Registo SPI1.
rData = SPI1BUF; // Apaga o buffer de recepção
SPI1BRG = 0x0; // Define a frequencia de clock
SPI1STATCLR = 0x40; // limpa o bit de overflow
SPI1CON = 0x8320; // Activa o SPI em Master mode
SPI1CONbits.MODE16 = 1; //Define o envio de 16 bits
SPI1CONbits.MSSEN=1; //Gera o sinal de select

//Definição dos pinos para saída de periféricos
SDI1R = 0x0001; //Define o Pino RB15 como SDI
RPB8R = 0x0003; //Define o pino RB8 como SCK
RPB3R = 0x0003; //Define o Pino RB3 como CS

//Configuração do Timer
T2CON = 0x0000; // Pára o timer e limpa os registos,
TMR2 = 0x0000; // Limpa o valor do Timer
T2CONSET = 0x8000; // Inicia o Timer

AD1CON1SET = 0x0002; // Inicia a Aquisição
asm("nop");
AD1CON1CLR = 0x0002; // Inicia a Conversão
}
```