

**Programa de Pós-Graduação em Engenharia Elétrica
Interunidades**



Elvio Alexandrini Maciel

Simulação de redes de sincronismo usando computação paralela

Sorocaba

2022

Elvio Alexandrini Maciel

Simulação de redes de sincronismo usando computação paralela

Dissertação apresentada como parte dos requisitos para obtenção do título de Mestre em Engenharia Elétrica, junto ao Programa de Pós-Graduação em Engenharia Elétrica, Interunidades, entre o Instituto de Ciência e Tecnologia de Sorocaba e o Campus de São João da Boa Vista da Universidade Estadual Paulista “Júlio de Mesquita Filho”.

Orientador: Prof. Dr. Átila Madureira Bueno

Coorientadora: Profa. Dra. Maria Glória Caño de Andrade

Sorocaba

2022

M152s Maciel, Elvio Alexandrini
Simulação de redes de sincronismo usando computação paralela /
Elvio Alexandrini Maciel. -- São João da Boa Vista, 2022
107 p. : il., tabs.

Dissertação (mestrado) - Universidade Estadual Paulista (Unesp),
Faculdade de Engenharia, São João da Boa Vista
Orientadora: Átila Madureira Bueno
Coorientadora: Maria Glória Caño de Andrade

1. Processamento paralelo (Computadores). 2. Sincronização. 3.
Osciladores não-lineares. 4. Osciladores elétricos. 5. Sistemas de
controle por realimentação. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Biblioteca da Faculdade de
Engenharia, São João da Boa Vista. Dados fornecidos pelo autor(a).

Essa ficha não pode ser modificada.

Elvio Alexandrini Maciel

Simulação de redes de sincronismo usando computação paralela

Dissertação apresentada como parte dos requisitos para obtenção do título de Mestre em Engenharia Elétrica, junto ao Programa de Pós-Graduação em Engenharia Elétrica, Interunidades, entre o Instituto de Ciência e Tecnologia de Sorocaba e o Campus de São João da Boa Vista da Universidade Estadual Paulista “Júlio de Mesquita Filho”.

Comissão examinadora

Prof. Dr. Átila Madureira Bueno
Escola Politécnica da Universidade de São Paulo - POLI-USP
Departamento de Engenharia de Telecomunicações e Controle
Orientador

Prof. Dr. José Roberto Castilho Piqueira
Escola Politécnica da Universidade de São Paulo - POLI-USP
Departamento de Engenharia de Telecomunicações e Controle

Prof. Dr. Wesley Angelino de Souza
Universidade Tecnológica Federal do Paraná - UTFPR
Departamento Acadêmico de Elétrica - Campus Cornélio Procopio

Sorocaba

2022

CERTIFICADO DE APROVAÇÃO

TÍTULO DA DISSERTAÇÃO: Simulação de redes de sincronismo usando computação paralela

AUTOR: ELVIO ALEXANDRINI MACIEL

ORIENTADOR: ATILA MADUREIRA BUENO

COORDINADORA: MARIA GLÓRIA CAÑO DE ANDRADE

Aprovado como parte das exigências para obtenção do Título de Mestre em Engenharia Elétrica, área:
Sistemas Eletrônicos pela Comissão Examinadora:

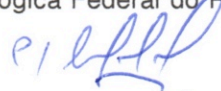
Prof. Dr. ATILA MADUREIRA BUENO (Participação Virtual)
Politécnico / Universidade de São Paulo USP



Prof. Dr. JOSÉ ROBERTO CASTILHO PIQUEIRA (Participação Virtual)
Departamento de Engenharia de Telecomunicações e Controle / USP / São Paulo



Prof. Dr. WESLEY ANGELINO DE SOUZA (Participação Virtual)
Departamento Acadêmico de Elétrica (DAELE) / Universidade Tecnológica Federal do Paraná - UTFPR /
Campus de Cornélio Procopio



Sorocaba, 19 de dezembro de 2022

*Em especial à minha namorada Eliani Farias e a minha mãe Maria Angélica, que me acompanharam em toda a trajetória, com muita ajuda e paciência para vencermos juntos as complexidades e tribulações que enfrentamos no curso...
dedico.*

“Nothing in life is to be feared, it is only to be understood. Now is the time to understand more, so that we may fear less.”

Marie Curie

The Discovery of Radium: Research on Radioactive Substances.

Agradecimentos

Agradeço aos meus pais, José Neil (*in memoriam*) e Maria Angélica, que me ensinaram desde cedo o caminho que eu deveria trilhar e me guiaram dando todo o apoio, sustento, direção e amor.

Ao meu Orientador, Professor Doutor Átila Madureira Bueno, pela paciência e oportunidade, por confiar e me aceitar como seu orientado, por toda atenção para que este trabalho se concretizasse.

À Universidade Estadual Paulista “Júlio de Mesquita Filho” (UNESP), em especial ao Instituto de Ciência e Tecnologia de Sorocaba (ICTS) que proporcionaram os meios necessários para a realização deste trabalho.

Aos professores do Programa de Pós-Graduação em Engenharia Elétrica (PGEE), pela dedicação em suas orientações prestadas na elaboração deste trabalho, que em todo tempo me apoiaram e motivaram para a minha formação.

Ao amigo Matheus Silva Batista, de São João da Boa Vista - SP, pelo estudo, apoio e confiança.

À professora Cristiane Mileo Batistela Gouvêa, da Universidade Federal do ABC (UFABC), pela atenção e disposição em ajudar com a publicação dos artigos científicos exigidos pelo Programa.

Àquelas pessoas que aqui não foram citadas, mas que de alguma forma, colaboraram para o sucesso da conclusão do Mestrado, meus sinceros agradecimentos.

Encerro fazendo um agradecimento especial à banca examinadora, pela revisão e sugestões de melhorias para este trabalho.

Resumo

Sistemas dinâmicos são usados para modelar e fazer previsões de sistemas físicos, biológicos e até financeiros, porém uma grande variedade de sistemas apresentam comportamentos irregulares e imprevisíveis, onde uma pequena perturbação nas condições iniciais pode resultar em uma grande diferença da resposta desejada. Dentro deste conceito, encaixam-se as malhas de sincronismo que integram inúmeras aplicações da Engenharia, principalmente as redes de comunicação que, estando em sincronismo, caracterizam-se pelo fato de diversos osciladores operarem na mesma frequência e fase. Um dos circuitos osciladores mais conhecidos e empregados nesse tipo de aplicação é o PLL (*Phase-Locked Loop*) que trata de um sistema de controle de malha fechada que a partir de um sinal de referência, gera um sinal de saída sincronizado. Por manter em regime permanente a frequência de sinal com o de entrada, tal sistema pode rastrear o sinal amostrado. Projetar corretamente redes de distribuição de sinal de tempo implicam no bom desempenho no desempenho, na confiabilidade da comunicação e na integridade dos dados transmitidos. Nesse contexto, pretende-se oferecer uma nova abordagem ao assunto, através da computação paralela, que é uma forma de se realizar vários cálculos ao mesmo tempo, nos diversos núcleos de processamento de um processador *multicore*, operando sob o princípio de que grandes problemas podem ser divididos em problemas menores, que então são resolvidos concorrentemente. Ao empregar paralelismo computacional, pretende-se reduzir o tempo necessário para solucionar um problema e consequentemente resolver problemas mais complexos e de maiores dimensões.

Palavras-chave: Sincronismo, PLL, Computação paralela, sistemas dinâmicos não-lineares.

Abstract

Dynamical systems are used to model and make predictions of physical, biological and even financial systems, but a wide variety of systems have irregular and unpredictable behavior, where a small disturbance in the initial conditions can result in a large difference in the intended response. Within this concept, the Synchronization networks that integrate countless Engineering applications, mainly the communication networks, being in synchronism, are characterized by the fact that several oscillators operate the same frequency and phase. The known oscillator circuits used in this application type is the PLL (Phase-Locked Loop), which deals with a closed-loop control system that generates a synchronized output signal from a reference signal. By keeping the signal frequency with input steady state, such a system can track the sampled signal. Properly designing time signal distribution networks generate good performance, communication reliability and transmitted data integrity. In this sense, we intend to offer a new approach to the subject, through parallel computing, which is a way of performing several calculations at the same time, in the different processing cores of a multicore processor, operating under the principle that big problems can be divided into smaller problems, which are then solved concurrently. By employing computational parallelism, it is intended to reduce the total time required to solve a problem, allowing to research solutions to more complex problems.

Keywords: Synchronism, PLL, Parallel Computing, Nonlinear Dynamical Systems.

Lista de Figuras

2.1	Diagrama de blocos do PLL.	22
2.2	Diagrama de blocos do PLL analógico clássico.	23
2.3	Diagrama de blocos do PLL linearizado	27
2.4	Diagrama de blocos do DPLL	28
2.5	Ilustração do processamento paralelo	32
2.6	Topologia hipercubo de processadores	34
2.7	Relação entre as várias camadas na arquitetura bcMPI	36
2.8	Toolbox de computação paralela e o Servidor de Computação Distribuída	38
2.9	Loop de iteração com parfor.	39
3.1	Estratégias de distribuição de sinais de tempo.	42
3.2	Hierarquias das redes de sincronismo	42
3.3	Arquiteturas MS Básicas	43
3.4	Arquiteturas Estrela	44
3.5	Arquiteturas Anel	44
3.6	Blocodiagrama do PLL escravo	45
3.7	Diagrama de blocos dos PLL de uma rede TWMS.	49
4.1	Rede OWMS simulada	55
4.2	Fluxograma de operação.	57
4.3	Saída do 1º PLL.	58
4.4	Saída do 2º PLL.	58

4.5	Saída do 3º PLL.	59
4.6	Saída do 4º PLL.	59
4.7	Lissajous de Fase.	60
4.8	Saída do 1º PLL.	61
4.9	Saída do 2º PLL.	61
4.10	Saída do 3º PLL.	62
4.11	Saída do 4º PLL.	62
4.12	Erro de fase para a excitação em rampa.	63
4.13	Lissajouse para a excitação em rampa.	63
4.14	Resposta em regime do PLL Mestre ao degrau com Filtro LLP.	66
4.15	Resposta em regime do 2º PLL ao degrau com Filtro LLP.	66
4.16	Resposta em regime do 3º PLL ao degrau com Filtro LLP.	67
4.17	Resposta em regime do 4º PLL ao degrau com Filtro LLP.	67
4.18	Lissajous da Rede OWMS com Filtro LLP.	68
4.19	Resposta em regime dos PLLs a rampa com Filtro LLP.	68
4.20	Sinais de controle dos PLLs.	69
4.21	Resposta em regime do PLL Mestre ao degrau com Filtro LLA.	70
4.22	Resposta em regime do 2º PLL ao degrau com Filtro LLA.	70
4.23	Resposta em regime do 3º PLL ao degrau com Filtro LLA.	70
4.24	Resposta em regime do 4º PLL ao degrau com Filtro LLA.	71
4.25	Lissajous da Rede OWMS com Filtro LLA.	71
4.26	Resposta em regime dos PLLs a rampa com Filtro LLA.	72
4.27	Sinais de controle dos PLLs $vc(t)$	72
4.28	Resposta em regime do PLL Mestre ao degrau com Filtro PI Ativo.	73
4.29	Resposta em regime do 2º PLL ao degrau com Filtro PI Ativo.	74
4.30	Resposta em regime do 3º PLL ao degrau com Filtro PI Ativo.	74
4.31	Resposta em regime do 4º PLL ao degrau com Filtro PI Ativo.	75
4.32	Lissajous para rede OWMS com Filtro PI Ativo.	75

4.33	Resposta em regime dos PLLs a rampa com Filtro PI Ativo.	76
4.34	Lissajous para rede OWMS com Filtro PI Ativo e entrada em rampa. .	77
4.35	Resposta em regime do 1º, 2º e 3º PLLs ao degrau com Filtro F_4	77
4.36	Resposta em regime do 4º PLL ao degrau com Filtro F_4	78
4.37	Lissajous da rede OWMS com Filtro F_4 e entrada em degrau.	78
4.38	Resposta em regime dos PLLs a rampa com Filtro F_4	79
4.39	Sinais de Controle dos 4 PLLs da rede a entrada em rampa.	79
4.40	Resposta em regime do PLL Mestre ao degrau com Filtro F_5	80
4.41	Resposta em regime do 2º PLL ao degrau com Filtro F_5	80
4.42	Resposta em regime do 3º PLL ao degrau com Filtro F_5	81
4.43	Resposta em regime do 4º PLL ao degrau com Filtro F_5	81
4.44	Lissajous da rede OWMS com filtro F_5 com excitação em degrau. . . .	82
4.45	Resposta em regime dos PLLs a rampa com Filtro F_5	82
4.46	Sinais de controle dos PLLs a rampa com Filtro F_5	83
4.47	Arquitura Estrela de via dupla	84
4.48	Resposta em regime dos PLLs ao degrau com Filtro LLP	84
4.49	Sinais de Controle dos PLLs ao degrau com Filtro LLP	85
4.50	Resposta em regime dos PLLs a rampa com Filtro LLP	85
4.51	Sinais de Controle dos PLLs a rampa com Filtro LLP	86
4.52	Resposta em regime dos PLLs ao degrau com Filtro LLA	86
4.53	Sinais de Controle dos PLLs ao degrau com Filtro LLA	87
4.54	Resposta em regime dos PLLs a rampa com Filtro LLA	87
4.55	Sinais de Controle dos PLLs a rampa com Filtro LLA	88
4.56	Lissajous dos PLLs ao degrau com Filtro PI Ativo	89
4.57	Sinal de Controle dos PLLs ao degrau com Filtro PI Ativo	89
4.58	Resposta em regime dos PLLs a rampa com Filtro PI Ativo	90
4.59	Lissajous para entrada em Rampa dos PLLs com Filtro PI Ativo	91
4.60	Resposta em regime dos PLLs ao degrau com Filtro F_4	91

4.61	Lissajous dos PLLs ao degrau com Filtro F_4	92
4.62	Resposta em regime dos PLLs a rampa com Filtro F_4	93
4.63	Sinal de Controle dos PLLs a rampa com Filtro F_4	93
4.64	Resposta em regime do PLL Mestre ao degrau com Filtro F_5	94
4.65	Resposta em regime do 2º e 3º PLLs ao degrau com Filtro F_5	94
4.66	Resposta em regime do 4º PLL ao degrau com Filtro F_5	95
4.67	Erro de fase dos PLLs	95
4.68	Lissajous de resposta da rede Estrela TWMS ao degrau com Filtro F_5 .	96
4.69	Resposta em regime dos PLLs Mestre a rampa com Filtro F_5	96
4.70	Sinais de controle dos PLLs a rampa com Filtro F_5	97
A.1	Movimento de oscilação harmônica	107

Lista de Tabelas

2.1	Seleção de funções MPI fornecidas pelo MatlabMPI	35
4.1	Valores de erro de fase dos PLLs - Rede OWMS com um Mestre e três PLLs Escravos	60
4.2	Comparação de tempos de simulação para a excitação em Degrau . . .	64
4.3	Comparação de tempos de simulação para a excitação em Rampa . . .	64
4.4	Comparação de tempos de simulação para o Processador Intel Core i3- 5005S Dual-Core 2,00GHz	98
4.5	Comparação de tempos de simulação para o Processador Intel Core i5- 4440S Quad-Core 2,80GHz	98

Sumário

1	Introdução	17
1.1	Motivação	19
1.2	Roteiro do trabalho	21
2	Revisão Bibliográfica	22
2.1	PLL: Phase-Locked Loop	22
2.1.1	Diagrama de Blocos do PLL	22
2.1.2	Modelagem Matemática do PLL	23
2.1.3	Modos de operação do PLL	27
2.2	PLL Digital	28
2.3	Programação Paralela	30
2.3.1	Histórico	30
2.3.2	Programação Paralela: Definição	31
2.3.3	Motivação do Paralelismo	31
2.3.4	Desafios do Paralelismo	33
2.3.5	Programação Paralela no Matlab™	33
2.3.6	Lei de Amdahl	38
2.3.7	Resumo da Revisão Bibliográfica	40
3	Sincronismo de Redes	41
3.1	Estratégias de distribuição de sinais de tempo	41
3.1.1	Classificação das redes	43

3.2	Modelo matemático da rede OWMS	43
3.3	Modelo matemático da rede TWMS	48
3.4	Estabilidade e sincronização de redes	53
3.5	Resumo do Capítulo	54
4	Simulação de Redes de Sincronismo com computação paralela	55
4.1	Algoritmo proposto para a Simulação Piloto	55
4.2	Resultados Numéricos da Simulação Piloto	58
4.2.1	Excitação em Degrau de Fase	58
4.2.2	Excitação em Rampa de Fase	61
4.2.3	Comparação de Tempos da Simulação Piloto	64
4.3	Filtros Controladores Propostos	65
4.4	Simulação de rede OWMS	66
4.4.1	Simulação de rede OWMS com Filtro LLP:	66
4.4.2	Simulação de rede OWMS com Filtro LLA:	69
4.4.3	Simulação de rede OWMS com Filtro PI Ativo:	73
4.4.4	Simulação de rede OWMS com Filtro de 2 ^a Ordem Tipo 0	77
4.4.5	Simulação de rede OWMS com Filtro de 2 ^a Ordem Tipo 1	80
4.5	Simulação de rede Estrela TWMS	84
4.5.1	Simulação de rede Estrela TWMS com Filtro LLP	84
4.5.2	Simulação de rede Estrela TWMS com Filtro LLA	86
4.5.3	Simulação de rede Estrela TWMS com Filtro PI Ativo	88
4.5.4	Simulação de rede TWMS com Filtro de 2 ^a Ordem Tipo 0	91
4.5.5	Simulação de rede TWMS com Filtro de 2 ^a Ordem Tipo 1	94
4.6	Comparação de Tempos da Simulação com os Filtros	98
4.7	Resumo do Capítulo	99
5	Conclusões	100

A	Sistemas Dinâmicos	105
A.1	Método de Integração Numérica	105
A.2	Curva de Lissajous	107

Capítulo 1

Introdução

A malha de sincronismo, ou *Phase-Locked Loop*, consiste no sistema de controle de malha fechada usado para ajustar automaticamente a fase de um sinal geral localmente à fase de um sinal de entrada. Sendo a rede de distribuição de sinais de tempo, composta por determinado número de osciladores conectados, conforme estratégia, com a finalidade de executar a coordenação temporal entre os mesmos, denominada sincronismo. [1, 2].

Presente em cada nó das redes de distribuição de sinais de tempo, o PLL (Phase-Locked Loops) é fundamental no desempenho de toda a rede, dado que os erros em cada nó podem acumular e serem transmitidos aos demais [1, 2, 3]. Além disso, fenômenos relacionados ao meio de transmissão e ao processo de modulação podem gerar variações instantâneas na fase do sinal de entrada, produzindo oscilações no sinal de saída de cada nó [1, 9].

O estudo de PLLs analógicos vem sendo desenvolvido em diferentes perspectivas, desde os trabalhos de Lindsey em [4, 5] sobre a teoria de distribuição de sinais de tempo em uma rede com relógios geograficamente dispersos; em sequência Piqueira em [2], com enfoque nas respostas das diferentes arquiteturas para entradas de processos de fase diversos; os métodos para o cálculo das taxas de falha de sincronização de Popken em [6] passando pelo estudo do *jitter*¹ proposta por Piqueira e Monteiro em [3], pela determinação do sincronismo em redes mestre-escravo com atraso realizada por Bueno em [9]; até os estudos mais recentes de Kuznetsov em [8], que não considera somente o erro de fase, mas também o espaço de fase do sinal ao incluir a variação de frequência entre os nós da rede e as perturbações produzidas pelos múltiplos detectores de fase,

¹Termo de frequência dupla que surge em função da dinâmica do Detector de Fase do PLL analógico.

assumindo a não-linearidade do PLL.

O PLL é um elemento fundamental para o desempenho das redes de distribuição de sinais de tempo, pois é o elemento de controle usado para ajustar automaticamente o sinal local àquele gerado a grande distância (principalmente em aplicações de GPS - “Global Positioning System”, em português, Sistema de Posicionamento Global) e sujeito a inúmeras fontes de interferência durante a transmissão. [1, 2, 9].

As redes de sincronismo e, conseqüentemente os PLLs, podem ser implementados tanto em software como em hardware [12, 13]. O primeiro PLL foi implementado em hardware por H. De Bellescize em 1932. O primeiro circuito integrado (CI) de um PLL puramente analógico surgiu em 1965. Os primeiros PLLs híbridos que utilizavam um detector de fase digital e um VCO analógico surgiram por volta de 1970. Os PLLs completamente digitais são implementados via software e não possuem nenhum componente elétrico, como resistores, capacitores e indutores [12, 13].

Atualmente, as redes de PLLs ajudam a compor a eletrônica do GPS. Ao produzir as fases dos componentes I (*in-phase*) e Q (*quadrature*) com 90° de defasagem, a amplitude do sinal resultante pode ser calculada a partir da soma vetorial dessas componentes e o ângulo de fase em relação ao eixo I pode ser determinado a partir do arcotangente de Q/I. Qualquer desalinhamento na fase do oscilador local em relação à fase do sinal amostrado, produzirá um ângulo diferente, de modo que possa ser detectado e corrigido pela malha de controle [31]. Este cálculo auxilia no processo de triangulação, onde pelo menos três satélites são utilizados para determinar uma posição no globo terrestre [31].

Com o intuito de prover uma colaboração às malhas de sincronismo, este trabalho apresenta a abordagem da computação paralela como auxiliar na concepção e simulação de redes de PLLs, visto que o sincronismo ocorre em todos os nós da rede ao mesmo tempo e o loop de rastreamento é sempre o elo fraco em um receptor GPS autônomo, que influencia no seu desempenho [31]. A computação paralela é o método computacional que executa vários cálculos (processos) simultaneamente, fazendo com que os núcleos do processador *multicore* trabalhem em cooperação e sincronia. O princípio da computação paralela é poder dividir grandes problemas em frações menores, e assim, esses pedaços serem processados simultaneamente.

Um programa é considerado programação paralela quando é visto como um conjunto de partes que podem ser resolvidos concorrentemente, sendo cada uma igualmente constituída por uma série de instruções sequenciais, mas em seu conjunto podem ser executadas simultaneamente em vários núcleos do processador [14, 15, 16].

1.1 Motivação

Distribuição de sinais de tempo, principalmente ao longo de uma rede, consiste no controle de fase e frequência de sinais de relógio difundidos em uma área geográfica. O objetivo principal é sincronizar escalas de fase e frequência de múltiplos osciladores da rede, usando a comunicação entre cada ponto de interconexão [2, 3, 4]. Essa questão é pertinente a inúmeras aplicações de Engenharia, como sistemas de comunicação e transmissão de dados, navegação por GPS [31], rastreamento de sinais, sistemas de monitoração e controle de processos industriais. As questões listadas são suficientes para justificar a relevância da aplicação exigindo tratamento matemático preciso [7]; além disso, as equações relativas aos processos, podem servir na modelagem de problemas de sincronismo entre osciladores biológicos, que constituem requisito fundamental para a vida [23].

Dentre as redes utilizadas em engenharia, as redes Mestre-Escravo (do inglês *Master-Slave: MS*) se destacam no meio comercial, por apresentarem baixo custo e facilidade de implementação. Nas redes MS, o nó Mestre costuma ser um oscilador extremamente preciso e independente, como por exemplo, um relógio atômico de Césio ou de Rubídio. As redes Mestre-Escravo estudadas neste trabalho serão OWMS (“*One-Way Master-Slave*”, em português, Rede Mestre-Escravo de Via Única) e TWMS (“*Two-Way Master-Slave*”, em português, Rede Mestre-Escravo de Via Dupla), populares no meio técnico e comercial devido as suas simplicidades topológicas. A designação OWMS se deve ao fato de que a transmissão de sinais de tempo nessas redes ocorre em um único sentido, isto é, sempre a partir do Mestre até o último Escravo. E a designação TWMS se deve ao fato de que a transmissão ocorrer em duplo sentido entre os Escravos e o Mestre. Ainda sobre redes de via dupla, é importante ressaltar a bibliografia [32], que aplica a Teoria de Sistemas Dinâmicos para demonstrar a instabilidade em redes com mais de 4 nós e que norteia o estudo da topologia Estrela simulada neste trabalho.

Para tratar o problema das redes de sincronismo, este trabalho propõe o uso da Computação Paralela, dividindo a questão principal em segmentos que podem ser resolvidos concorrentemente e executados simultaneamente em vários núcleos de um processador.

O hardware tem evoluído rapidamente e as plataformas de processamento paralelo tornam-se cada vez mais acessíveis. A busca por alto desempenho em aplicações de propósito geral e científicas tem-se voltado para a exploração do processamento pa-

ralelo fornecido por plataformas que congregam processadores *multicores*; potencializa a capacidade de resolução de questões de Engenharia, principalmente de sincronismo e proporcionam ganhos de desempenho para as aplicações [11].

Programar novas aplicações com paralelismo ou traduzir aplicações de código serial para plataformas *multicores* não é uma tarefa trivial se comparada ao modo sequencial. Existem pelo menos duas formas de se obter aplicações paralelas, uma delas é a concepção de aplicações naturalmente paralelas, sendo escritas desde o início em uma linguagem de programação com suporte a paralelismo. A outra maneira, empregada nesta dissertação, é a adaptação de aplicações existentes, reescrevendo e readequando seus códigos de forma a empregar todo o poder computacional disponível em cada núcleo de processamento e congregar os resultados parciais em uma resposta mais efetiva, complexa e mais realista [14, 15, 20].

Diante desse contexto, este trabalho tem como objetivo desenvolver simulações de redes de sincronismos de via única OWMS e via dupla TWMS com o auxílio da computação paralela através do software Matlab e suas ferramentas de paralelismo como forma tentar viabilizar toda a complexidade envolvida no estudo, visto que na prática os nós da rede são sincronizados simultaneamente. Em resumo, as contribuições propostas do estudo são:

- Apresentar a possibilidade do paralelismo computacional para a solução de redes de sincronismo;
- Simular e verificar se as respostas e gráficos gerados correspondem ao esperado, comparando-se a programação serial com a paralela, e assim, comprovando e ratificando a aplicação das redes de PLLs;
- Reafirmar a importância dos malhas de controle de oscilação e sincronismo, oferecendo uma nova vertente de estudo e possíveis novas aplicações;
- Discutir a importância do paralelismo computacional, apresentando suas vantagens e dificuldades de implementação, tanto pela adaptação da programação, quanto pela gerência do hardware para a execução do processamento;
- Permitir a continuidade da pesquisa do paralelismo no Matlab através de outras ferramentas que foram apresentadas aqui neste trabalho, visando aperfeiçoar o tempo de resposta e a possibilidade de propor redes mais complexas e, consequentemente, mais realistas.

1.2 Roteiro do trabalho

O Capítulo 1 destaca a motivação para o estudo das redes de sincronismo e apresenta a computação paralela como ferramenta a ser estudada.

O Capítulo 2 apresenta a modelagem matemática do PLL analógico clássico, comenta sobre os DPLLs e conceitua a programação paralela, realizando-se uma revisão bibliográfica.

O Capítulo 3 demonstra as estratégias de distribuição de sinais, trata sobre os modelos matemáticos das redes OWMS e TWMS, e discute sobre estabilidade e sincronização de redes.

O Capítulo 4 apresenta uma simulação piloto com filtro de dinâmica simples de 1ª Ordem, que valida o emprego do paralelismo computacional para aplicações em rede de sincronismo. Na sequência são apresentados outros Filtros e outras simulação são realizadas. Também são discutidos os sincronismos apresentados pelos gráficos de resposta além de demonstrar a comparação de tempos entre a programação serial e a paralela

O Capítulo 5 faz a reflexão do estudo realizado, avaliando o uso do paralelismo computacional para redes de sincronismo e aponta possíveis trabalhos futuros.

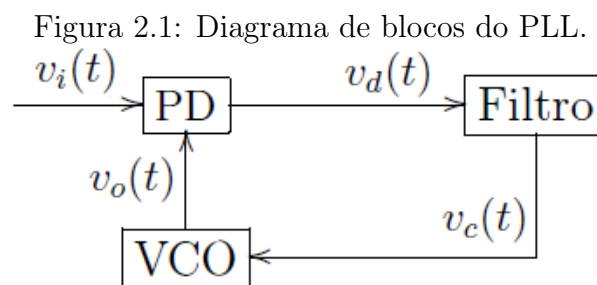
Capítulo 2

Revisão Bibliográfica

2.1 PLL: Phase-Locked Loop

2.1.1 Diagrama de Blocos do PLL

O PLL (Phase-Locked Loop) é um sistema de controle em malha fechada que gera, a partir de um sinal de referência, um sinal de saída com frequência e fase instantâneas em sincronismo com a entrada. A Figura 2.1 demonstra os principais blocos de um PLL.



Fonte: Adaptado de [1]

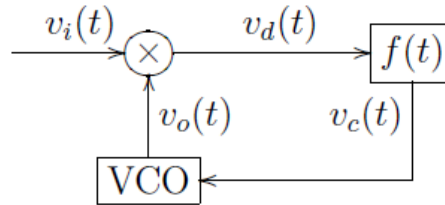
- $v_i(t)$: sinal de entrada ou referência;
- $v_o(t)$: sinal de saída;
- PD (Phase Detector): detector e comparador de fase dos sinais de entrada e saída;
- $v_d(t)$: sinal de erro (ou desvio) gerado pelo PD que excita o Filtro;

- Filtro¹: recebe o sinal de erro do PD, processa e gera o sinal de controle;
- $v_c(t)$: sinal de controle da saída do Filtro que excita o VCO;
- VCO (Voltage Controlled Oscillator): oscilador local controlado por tensão, recebe $v_c(t)$ e faz a compensação da saída $v_o(t)$.

2.1.2 Modelagem Matemática do PLL

O PLL é um sistema de controle de malha fechada que sincroniza o sinal de saída de um oscilador a um sinal de referência, tanto em frequência como em fase. Estando em sincronia, a diferença (ou erro) de fase entre o sinal de saída do oscilador local e o sinal de referência deve ser nula ou manter-se constante (erro de *offset*). Ocorrendo perturbações no sinal, o sistema tenta se autocorrigir [1, 2]. A Figura 2.2 demonstra como ocorre a interação entre os sinais envolvidos no loop de controle.

Figura 2.2: Diagrama de blocos do PLL analógico clássico.



Fonte: Adaptado de [1]

Para o PLL analógico, o detector de fase é um circuito multiplicador de sinais. O sistema de controle em malha fechada do PLL tem o objetivo de sincronizar a saída do VCO $v_o(t)$ ao sinal de referência $v_i(t)$. O sinal de controle $v_c(t)$ atua sobre a frequência do VCO, tentando de minimizar (ou até mesmo eliminar) o erro de fase [1, 2].

Os sinais de entrada e saída do PLL podem ser expressos de acordo com as Equações (2.1) e (2.2).

$$v_i(t) = v_i \sin(\omega_M(t) + \theta_i(t)) \quad (2.1)$$

$$v_o(t) = v_o \cos(\omega_M(t) + \theta_o(t)) \quad (2.2)$$

Para realizar a correta modelagem matemática do PLL, precisa-se estabelecer alguns critérios, que estão na Observações A, B e C.

¹Neste trabalho foi utilizada a nomenclatura Filtro no mesmo sentido tratado em Telecomunicações, pois aqui opera como um Controlador (ou Compensador) de sinal.

- **Observação A:** Erros de fase e frequência: O erro de fase $\vartheta(t)$ é a diferença entre a fase do sinal de entrada $\theta_i(t)$ e a fase do sinal de saída $\theta_o(t)$ do VCO local; o erro de frequência angular é a taxa de variação do erro de fase em relação ao tempo, conforme as Equações (2.3) e (2.4).

$$\vartheta(t) = \theta_i(t) - \theta_o(t) \quad (2.3)$$

$$\dot{\vartheta}(t) = \dot{\theta}_i(t) - \dot{\theta}_o(t) \quad (2.4)$$

- **Observação B:** Função de Transferência $F(s)$ do filtro: pode ser expressa como uma razão de polinômios no domínio s , conforme as Equações (2.5) a (2.7).

$$F(s) = \frac{N(s)}{D(s)} \quad (2.5)$$

onde:

$$N(s) = \sum_{m=0}^M \alpha_m s^m \quad (2.6)$$

$$D(s) = \sum_{p=0}^P \beta_p s^p \quad (2.7)$$

- **Observação C:** Operadores Q e L , conforme as Equações (2.8) e (2.9).

$$Q(x) = \mathcal{L}^{-1}[N(s)X(s)] = \sum_{m=0}^M \alpha_m \frac{d^m}{dt^m} x(t) \quad (2.8)$$

$$L(x) = \mathcal{L}^{-1}[sD(s)X(s)] = \sum_{p=0}^P \beta_p \frac{d^{p+1}}{dt^{p+1}} x(t) \quad (2.9)$$

sendo M e P as ordens dos polinômios $N(s)$ e $D(s)$, com $P > M$. Aplicando-se diretamente da propriedade linear da transformada de Laplace, conforme as Equações (2.10) e (2.11).

$$Q(ax + by) = aQ(x) + bQ(y) \quad (2.10)$$

$$L(ax + by) = aL(x) + bL(y) \quad (2.11)$$

Feitas as devidas observações, segue a modelagem matemática com o Ganho do PLL, conforme a Equação (2.12).

$$G = \frac{1}{2} k_m k_o v_i v_o \quad (2.12)$$

onde:

- k_m : Ganho do detector de fase (V/rad);

- k_o : Ganho do VCO (rad/V);
- v_i e v_o : amplitudes dos sinais entrada e saída (V).

A saída do detector de fase é proporcional ao produto do sinal de entrada pelo sinal de saída do VCO, conforme a Equação (2.13).

$$v_d(t) = k_m v_i(t) v_o(t) \quad (2.13)$$

Substituindo as Equações (2.1) e (2.2) em (2.13).

$$v_d(t) = k_m k_o v_i v_o \text{sen}(\omega_M(t) + \theta_i(t)) \cos(\omega_M(t) + \theta_i(t)) \quad (2.14)$$

Aplicando-se a identidade trigonométrica na Equação (2.15).

$$\text{sen}(A) \cos(B) = \frac{1}{2} (\text{sen}(A - B) + \text{sen}(A + B)) \quad (2.15)$$

E substituindo a Equação (2.12) na (2.13), obtém-se a Equação (2.16).

$$v_d(t) = \frac{G}{k_o} v_\delta(t) \quad (2.16)$$

com:

$$v_\delta(t) = \text{sen}(\vartheta(t)) + \text{sen}(\omega_M(t) + \theta_i(t) + \theta_o(t)) \quad (2.17)$$

A saída $\theta_o(t)$ do VCO é controlada através da Equação Diferencial (2.18).

$$\frac{d}{dt} \theta_o(t) = k_o v_c(t) \quad (2.18)$$

sendo $v_c(t)$ o sinal de controle, a saída do Filtro, dado pela convolução (2.19).

$$v_c(t) = f(t) * v_d(t) \quad (2.19)$$

Substituindo a Equação (2.18) na (2.19) tem-se a Equação (2.20).

$$\frac{d}{dt} \theta_o(t) = k_o f(t) * v_d(t) \quad (2.20)$$

Aplicando-se o Teorema da Convolução, conforme Equação (2.21).

$$k_o N(s) V_d(s) = s D(s) \Theta(s) \quad (2.21)$$

Aplicando a Transformada Inversa de Laplace $\mathcal{L}^{-1}$ e substituindo os operadores Q e L da **Observação C** na Equação (2.22).

$$k_o Q[v_d(t)] = L[\theta_o(t)] \quad (2.22)$$

Substituindo as Equações (2.16) e (2.3) em (2.22), obtém-se (2.23).

$$k_o Q\left[\frac{G}{k_o} v_\delta(t)\right] = L[\theta_i(t) - \vartheta(t)] \quad (2.23)$$

Aplicando-se a **Observação A** na Equação (2.24).

$$L(\vartheta(t)) + GQ[v_\delta(t)] = L[\theta_i(t)] \quad (2.24)$$

Substituindo a Equação (2.17) em (2.24) e aplicando-se novamente a **Observação A**, tem-se a Equação (2.25).

$$L[\vartheta(t)] + GQ[\text{sen}(\vartheta(t))] = L[\theta_i(t)] - GQ[\text{sen}(2\omega_M(t) + \theta_i(t) + \theta_o(t))] \quad (2.25)$$

Isolando $\theta_o(t)$ da Equação (2.3) e substituindo em (2.25), produz (2.26).

$$L[\vartheta(t)] + GQ[\text{sen}(\vartheta(t))] = L[\theta_i(t)] - GQ[\text{sen}(2(\omega_M(t) + \theta_i(t)) + \vartheta(t))] \quad (2.26)$$

A Equação (2.26) relaciona o erro de fase $\vartheta(t)$ do PLL a uma fase de entrada θ_i . Os operadores Q e L dependem dos coeficientes da função de transferência do Filtro, que por sua vez, precisa ter comportamento linear e invariante no tempo. Conforme a Equação (2.26), o PD gera uma saída com duas parcelas, uma de baixa frequência (*wander*) e outra com o dobro da frequência de livre curso do VCO (*jitter*) [1, 9]. Considerando-se que o Filtro, devidamente projetado, consegue rejeitar as oscilações indesejadas, principalmente o *jitter* e aproximando $\text{sen}(\vartheta) \approx \vartheta$ (erro de fase pequeno) pode-se reescrever, conforme a Equação (2.27).

$$L[\vartheta(t)] + GQ[\vartheta(t)] = L[\theta_i(t)] \quad (2.27)$$

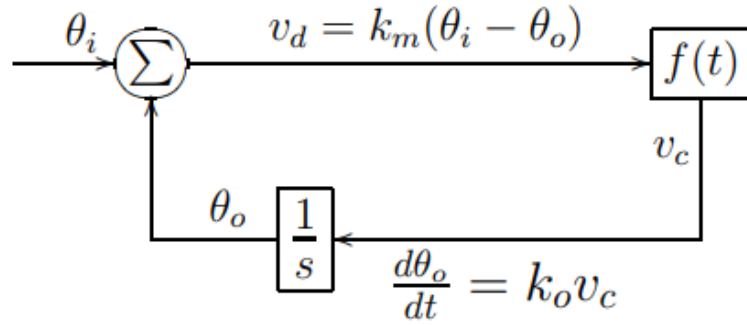
Após toda a modelagem matemática, consegue-se linearizar o comportamento do PLL, analógico clássico, conforme Figura 2.3, e a função de transferência relacionando diretamente a fase de saída pela fase de entrada, conforme a Equação (2.28).

$$\frac{\Theta_o(s)}{\Theta_i(s)} = \frac{k_m k_o F(s)}{s + k_m k_o F(s)} \quad (2.28)$$

onde $F(s)$ é a dinâmica do Filtro.

A modelagem matemática do PLL será importante para a construção da simulação das redes de sincronismo que será abordado no Capítulo 4.

Figura 2.3: Diagrama de blocos do PLL linearizado



Adaptado de [1]

2.1.3 Modos de operação do PLL

Nos estudos de Engenharia, existem dois modos dinâmicos de operação do PLL, conforme Piqueira em [2]:

- Aquisição: modo relativo ao alcance de um estado síncrono a partir de uma condição inicial.
- Rastreamento: modo relativo aos ajustes automáticos realizados pelo PLL para manter o estado síncrono mediante certas perturbações.

Conforme descrito na Equação (2.27), como um sistema dinâmico, deve-se ressaltar que o modo de rastreamento é um caso particular do modo de aquisição, reduzindo-se as condições iniciais possíveis àquelas próximas de um ponto de equilíbrio, assintoticamente estável, no espaço dos estados [1, 2]. A faixa de valores de frequência do sinal externo de referência, sincronizável com o PLL, recebe o nome de captura no modo de aquisição e de faixa de retenção no modo de rastreamento.

Para o estudo do comportamento dinâmico do PLL, normalmente consideram-se duas funções de excitação para a fase inicial θ_i , descritas analiticamente em [2], correspondendo a variações de fase com velocidade constante em (2.29) e com aceleração constante em (2.30).

$$\theta_i = \Omega t + \phi \quad (2.29)$$

$$\theta_i = \frac{Rt^2}{2} + \Omega t + \phi \quad (2.30)$$

Para simplificar as Equações (2.29) e (2.30), assume-se que R , Ω e ϕ são constantes reais não negativas.

Definições da terminologia para operação do PLL:

- **Definição 1** - Estado síncrono: O PLL em estado síncrono está em um ponto de equilíbrio de sua equação diferencial, conforme Equação (2.27) com erro de frequência nulo.

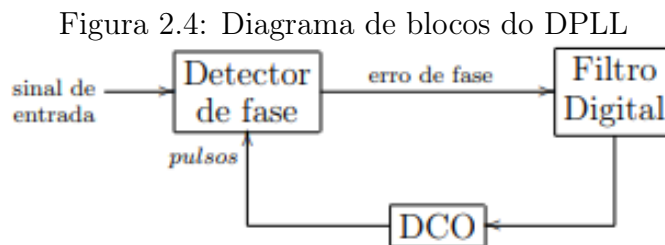
- **Definição 2** - Faixa de retenção: para o conjunto de valores de R e Ω , a malha estando em estado síncrono, esses valores mudem infinitesimalmente, a malha tenderá a voltar ao estado síncrono por ser assintoticamente estável.

- **Definição 3** - Faixa de captura: para o conjunto de valores de R e Ω , para a Equação (2.27) venha a convergir para um ponto de equilíbrio, assintoticamente estável, independentemente da condição inicial [22].

Essas definições serão essenciais para a compreensão e interpretação dos gráficos produzidos no Capítulo 4.

2.2 PLL Digital

Na literatura técnica, normalmente, os autores reservam o termo DPLL (Digital PLL), para aqueles cujos detectores de fase são digitais porém os demais blocos são analógicos e o termo ADPLL (All digital PLL) é usado para aqueles que são totalmente digitais e implementados em hardware [10]. Para normalizar a linguagem, neste trabalho será utilizado o termo DPLL para se referir as duas formas, pois objetiva-se apenas fazer uma revisão bibliográfica sobre o assunto, afim de se ter um panorama geral sobre PLLs. Os DPLLs com componentes analógicos, são constituídos de três blocos principais conforme Figura 2.4:



Fonte: Adaptado de [5].

O Detector de Fase (PD) calcula a diferença de fase e frequência entre o sinal de entrada e o sinal de pulsos gerado pelo DPLL, o filtro digital $F(z)$ determina a dinâmica geral da malha, e o DCO (*Digitally Controlled Oscillator*: Oscilador Controlado Digitalmente) gera pulsos de saída que sincroniza em frequência e fase com o sinal de

entrada.

No trabalho [23], Lindsey e Chie publicaram uma revisão sobre vários tipos de DPLLs voltados para o rastreamento de um sinal senoidal. Como os sistemas discretos necessitam de uma taxa de amostragem para a aquisição de dados, inicialmente, a classificação dos DPLLs pode ser estabelecida pelo período de amostragem uniforme e não-uniforme. Na primeira classificação, o intervalo de tempo entre amostras seguidas é constante. Na segunda, o intervalo é uma função que depende da diferença de fase e frequência entre trem de pulsos de entrada e o de saída. Os DPLLs também podem ser classificados segundo o detector de fase empregado:

- DPLL Flip-Flop: O sinal senoidal de entrada é convertido em um sinal quadrado por meio do comparador (*trigger*) e a detecção de fase é realizada por um *flip-flop RS* e um circuito contador síncrono.
- DPLL Nyquist-Rate: o sinal senoidal de entrada é amostrado uniformemente com uma frequência f_s (frequência de Nyquist) e convertido em um sinal digital de N -bits por meio do ADC (Conversor Analógico Digital) e em seguida, é multiplicado digitalmente com a saída do DCO, para gerar o sinal de erro de fase.
- DPLL avanço-atraso: gera um sinal de saída binário (0 ou 1) no detector de fase que indica se a saída do DCO é avançada ou atrasada em relação ao sinal de entrada que precisa ser convertido para sinal quadrado, por meio de um comparador feito por uma porta lógica *XOR* (*exclusive or*: ou exclusivo). O circuito integrado CD4046 da Texas InstrumentsTM é o mais conhecido DPLL desta aplicação.
- DPLL detecção de cruzamento por zero (ZC-DPLL): a função de amostragem e a detecção de fase são implementadas conjuntamente, utilizando um conversor analógico digital controlado pela saída do DCO, sendo a malha projetada para seguir um sinal quadrado. Basicamente o ZC-DPLL amostra o sinal de entrada nos pontos em que cruza o eixo temporal (ZC: *zero-crossing*), acumulando as amostras e gerando uma variação na fase do DCO, por meio do filtro de malha no sentido de levar o valor acumulado perto de zero.
- DTL: No trabalho [26], foi proposto um novo tipo de DPLL amostrado não uniformemente, chamado *Digital Tanlock Loop* (DTL), que utiliza um detector de fase linear baseado na função arco tangente, caracterizando o DTL por meio de uma equação diferencial linear, ou seja, sem a necessidade de aproximação

comum na análise de PLLs. A malha também utiliza a Transformada de Hilbert² que é a responsável por produzir o deslocamento de fase de 90° para a quadratura dos sinais, necessária na função arco tangente.

2.3 Programação Paralela

2.3.1 Histórico

De 1965 a 2015, conforme a Lei de Moore³, o desempenho dos microprocessadores dobrou, em média, a cada 18 meses. Esse aumento significou que usuários e desenvolvedores de software muitas vezes simplesmente esperassem pela próxima geração de microprocessadores para obter um desempenho maior de um programa. Desde 2015, no entanto, a melhoria do desempenho de um único processador diminuiu para cerca de 20% ao ano [11].

Grande parte do aumento no desempenho de um único processador foi impulsionado pela miniaturização de transistores em circuitos integrados. À medida que o tamanho dos transistores diminuiu, sua velocidade pode ser aumentada impactando na velocidade geral do circuito integrado (CI). Porém, a capacitância parasita do CI aumenta o consumo de energia e também o calor produzido; se um CI fica muito quente, começa a dar respostas espúrias no processamento [11, 12].

A partir de 2005, a maioria dos fabricantes de microprocessadores perceberam que o caminho para o rápido aumento do desempenho estava na direção do paralelismo, ou seja, em vez de tentar continuar a desenvolver cada vez mais velocidade para os processadores, começaram a colocar vários núcleos de processadores em um único circuito integrado. Para continuar melhorando o desempenho, a resposta dada foi o paralelismo. Em vez de construir um processador cada vez mais rápido e mais complexo, a indústria decidiu colocar vários núcleos, relativamente mais simples e completos, em um único chip. Esses circuitos integrados são chamados de processadores *multicore*

²A Transformada de Hilbert (TH) é uma operação linear que gera como saída uma função no mesmo domínio e ortogonal à função de entrada.

³Em abril 1965, o então presidente da Intel, Gordon Earle Moore, profetizou que a quantidade de transistores que poderiam ser colocados em uma mesma área dobraria a cada 18 meses mantendo-se o mesmo custo de fabricação. Embora na época a afirmação de Moore tenha sido levada como apenas um comentário ou brincadeira, as empresas de semicondutores baseiam-se nela para definir as suas metas de produção, investindo em pesquisa e desenvolvimento para manter o ritmo acelerado de inovação.

(multinúcleos) [10, 11].

Esta mudança tem uma consequência muito importante no desenvolvimento de software, pois simplesmente adicionar mais processadores não melhorará o desempenho dos programas que foram escritos para rodar em um único núcleo do processador pois desconhecem a existência de outros núcleos e são geridos pelo Sistema Operacional do computador. [11].

Quando os núcleos podem trabalhar independentemente, escrever um programa paralelo é mais complexo do que um programa serial, pois precisam de coordenação temporal para realizar o trabalho e tornar o processamento adequado [10, 12].

Atualmente, a arquitetura RAW desenvolvida no Instituto de Massachusetts de Tecnologia (MIT), tem a proposta de viabilizar um processador de propósito geral que tenha bom desempenho tanto em aplicações de paralelismo, quanto em programas sequenciais. O grande avanço da tecnologia de circuitos integrados tem expandido de maneira considerável o número de aplicações implementadas em *VLSI* (*Very Large Scale of Integrations*). Tais aplicações incluem tanto programas sequenciais executados sobre processadores de propósito geral (que empregam técnicas para exploração de paralelismo no nível de instrução), quanto aplicações que combinam vários núcleos para trabalharem em conjunto em uma determinada tarefa, principalmente em hardware dedicado. Exemplos de tais circuitos são aceleradores gráficos e controladores de redes [14, 15, 16].

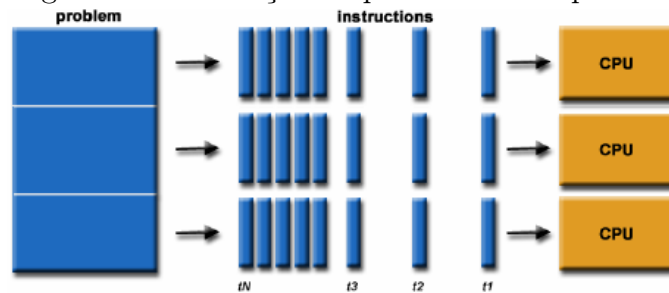
2.3.2 Programação Paralela: Definição

Um programa é considerado “serial” quando este é visto como uma sequência de instruções que devem ser executadas em um único núcleo de processamento. Um programa é considerado “paralelo” quando este é visto como um conjunto de partes que podem ser resolvidas concorrentemente e cada parte é igualmente constituída por uma série de instruções sequenciais conforme Figura 2.5, mas que no seu conjunto, podem ser executadas simultaneamente em vários núcleos de processamento.

2.3.3 Motivação do Paralelismo

Conforme as referências [10, 11, 12], os principais motivos para a utilização da programação paralela são:

Figura 2.5: Ilustração do processamento paralelo



Fonte: Adaptado de [10]

- Reduzir o tempo necessário para solucionar um problema;
- Solucionar problemas mais complexos e de maiores dimensões;
- Ultrapassar limitações de memória e processamento de um único computador e acionar as demais máquinas conectadas na mesma rede local,
- Transpor os limites físicos de velocidade e de miniaturização que restringem a construção de computadores sequenciais mais rápidos.

As motivações listadas dependem de inúmeros fatores (construção da lógica de programação e da arquitetura de hardware) e nem todas ocorrem da forma esperada pelo desenvolvedor, inclusive isso será objeto de discussão no Capítulo 5.

Tradicionalmente, a programação paralela foi motivada pela resolução e simulação de problemas fundamentais da Engenharia de grande relevância científica e econômica, denominados como *Grand Challenge Problems* (GCPs) [10]. Tipicamente, os GCPs simulam fenômenos que não podem ser medidos por experimentação, como por exemplo:

- Climático: alteração da temperatura na superfície do oceano;
- Físico: órbita de corpos celestes;
- Químico: reação nuclear;
- Biológico: código do genoma humano;
- Geológico: atividade sísmica;
- Mecânico: aerodinâmica de veículos espaciais;
- Eletrônico: verificação da integridade funcional de placas de circuitos eletroeletrônicos.

Atualmente, as aplicações que exigem o desenvolvimento de computadores cada vez mais rápidos estão por toda a parte. Estas aplicações requerem um grande poder computacional e o processamento de grandes volumes de dados [10]. Exemplos:

- Base de dados e mineração de dados;
- Serviços de busca na Internet;
- Serviços de *streaming* e telecomunicações;
- Computação gráfica e realidade virtual;
- Diagnóstico médico assistido por computador;
- Gestão de megacorporações.

2.3.4 Desafios do Paralelismo

As arquiteturas de computação paralelas são realidade, porém a programação paralela ainda é uma tarefa complexa e, por isso, motiva este trabalho. O paralelismo depende da disponibilidade de ferramentas e ambientes de programação adequados para memória compartilhada e/ou distribuída e encontram-se vários problemas que não existem na programação sequencial [10, 11, 12]:

- Concorrência: identificar as partes do programa que podem ser executadas em paralelo simultaneamente;
- Comunicação e Sincronização: esboçar o fluxo de dados processados, de forma que a programação (toda ou fragmentos dela) possa ser executada simultaneamente pelos diversos núcleos do processador, evitando situações de *deadlock* (uma operação depende de outra e o sequenciamento fica em um impasse) e *race condition* (resultado espúrio por falta de sincronismo);
- Balanceamento de Carga e Escalonamento: distribuir de forma equilibrada e eficiente as diferentes partes da programação pelos núcleos do processador, afim de que todos estejam majoritariamente ocupados durante toda a execução.

2.3.5 Programação Paralela no Matlab™

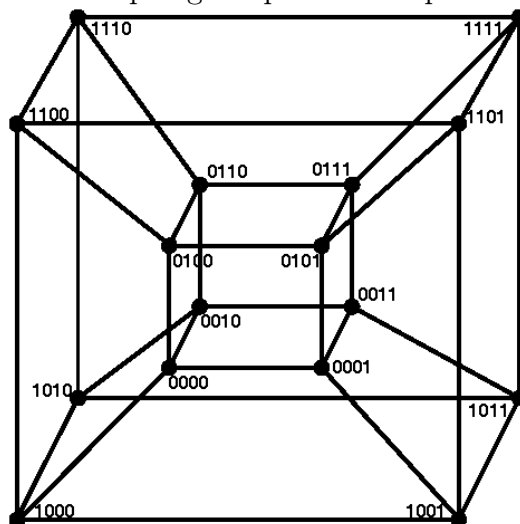
O Matlab é um software para a computação técnica e científica, sendo sua popularidade à facilidade de utilização desse programa sendo que o nome Matlab vem da junção das palavras *Matrix Laboratory*, isto se deve à base operacional, que são as

matrizes [13].

Esse software é utilizado para cálculos matemáticos, modelagens e simulações de sistemas lineares e não-lineares, discretos e contínuos no tempo, análises numéricas e processamentos, visualização de gráficos, desenvolvimentos de algoritmos [13, 14].

O paralelismo no Matlab é uma área de interesse de pesquisas para cientistas há vários anos e já foram realizadas várias abordagens e diferentes tentativas foram apresentadas. A primeira tentativa foi realizada em 1995, por Cleve Moler, autor da versão original do Matlab e cofundador da empresa MathWorks [21]. O autor descreveu os desafios no desenvolvimento de uma linguagem de programação paralela no Matlab e explicou o mais importante atributo da computação paralela: o uso compartilhado ou distribuído de memórias. Nessa direção, Moler discutiu a primeira tentativa de fazer paralelismo no Matlab em um iPSC (Intel Personal SuperComputer), que consistia em um supercomputador com vários processadores organizados em uma topologia chamada de hipercubo (Figura 2.6), com 32 a 128 nós dispostos e contendo um processador 80286 e 512K de RAM em cada nó, podendo executar um programa diferente em cada um.

Figura 2.6: Topologia hipercubo de processadores



Fonte: Adaptado de [21]

Depois de Cleve Moler, houveram muitas tentativas de paralelizar a programação no Matlab, mas todos enfrentaram problemas [21], onde a maioria deles consiste em:

- 1) o tempo de execução pode aumentar para pequenas aplicações;
- 2) falta de recursos e ferramentas computacionais;
- 3) aprimoramento de processadores com apenas um núcleo ao invés de processa-

dores multinúcleos até 2005 [11].

Atualmente existem dois métodos para desenvolver o paralelismo no Matlab [21]:

- 1) escrever código explícito para executar comunicação entre processadores na plataforma HPC (high performance computing), como MatlabMPI e bcMPI;
- 2) escrever código implícito, como pMatlab, StarP e a toolbox de Computação Paralela e Distribuída (que será objeto de estudo deste trabalho).

Fazendo uso da computação paralela, o Matlab pode ajudar a resolver grandes problemas de computação de diferentes maneiras. O software fornece um ambiente de programação interativo onde é possível vetorizar tarefas e usar o suporte de computação paralela através de matrizes distribuídas, permitindo realizar vários cálculos simultaneamente. Grandes problemas podem frequentemente ser divididos em menores, que são então resolvidos ao mesmo tempo (em sincronia) [15].

As principais razões para considerar a computação paralela no Matlab são: otimização do tempo de simulação ao distribuir tarefas com execução simultânea, resolução de problemas de *big data* e simulação de propagação de sinais em telecomunicações e eletrônica [15, 16, 19].

MatlabMPI

O MatlabMPI é uma implementação de script em Matlab baseada nas funções básicas de MPI (*Message Passing Interface*, interface de passagem de mensagens). Projetado por Jeremy Kepner do Laboratório Lincoln no Instituto de Massachusetts de Tecnologia (MIT)[22]. As funções necessárias para o MatlabMPI são as seguintes, conforme Tabela 2.1 adaptada de [19].

Tabela 2.1: Seleção de funções MPI fornecidas pelo MatlabMPI

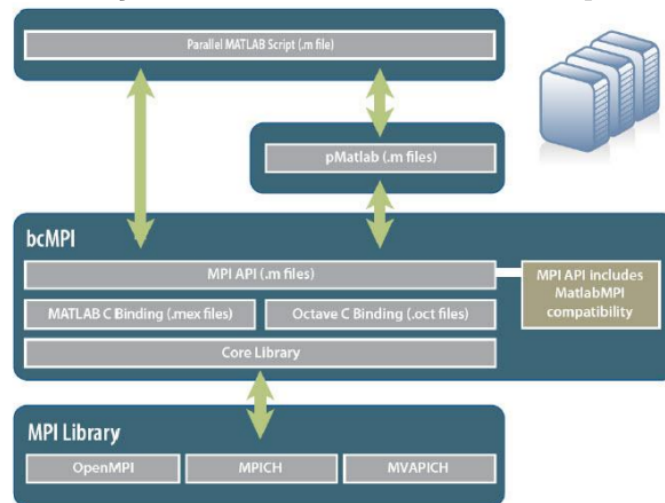
Função	Descrição
MPI_Init	Inicia o MPI
MPI_Comm_size	Número de processadores na comunicação
MPI_Comm_rank	Classificação de cada processador na comunicação
MPI_Send	Envia uma mensagem para o processador
MPI_Recv	Recebe uma mensagem do processador
MPI_Finalize	Finaliza o MPI

Este método tem muitas vantagens como a sua utilização apenas “Linguagem Matlab” para implementação (arquivo .m) e pode ser executado em qualquer máquina que tenha o programa. A segunda vantagem do MatlabMPI é a aplicação da comunicação MPI ao padrão entrada e saída de dados do Matlab. Além disso, essa abordagem possui uma biblioteca pequena (cerca de 300 linhas) e portátil. A desvantagem é a sua latência para pequenos códigos sendo mais alta do que um programa serial. Para ser executado em um rede de computadores, cada estação precisa ter uma licença do Matlab para compartilhar processamento e memória.

bcMPI

bcMPI é um biblioteca de código livre, uma alternativa para MatlabMPI, desenvolvido pela Blue Collar Computing™ em 2006, para fornecer um mecanismo de comunicação escalável e um eficiente para desenvolvimento de paralelismo no Matlab e softwares semelhantes como o GNU Octave da plataforma Linux [22].

Figura 2.7: Relação entre as várias camadas na arquitetura bcMPI



Fonte: Adaptado de [15]

A Figura 2.7 ilustra a relação entre as camadas BcMPI. A biblioteca bcMPI fornece funções para comunicação síncrona e assíncrona entre os processos do Matlab. Ele suporta funções básicas de MPI, bem como operações coletivas como MPI_Reduce, MPI_Gather e MPI_Barrier. O bcMPI também possui uma implementação eficiente da função MPI_Broadcast usando a biblioteca MPI subjacente.

pMatlab

O pMatlab foi desenvolvido pelo Laboratório Lincoln no MIT (Instituto Tecnológico de Massachusetts) em 2004. É definido como uma toolbox de programação paralela, para ser usada em uma rede de computadores, baseada em PGAS (*partitioned global address space*, espaço de endereçamento global particionado) [22].

O pMatlab implementou uma abordagem de programação implícita que deu várias vantagens como:

- 1) Suporte para matrizes de dados globais para desempenho otimizado, permitindo a integração e passagem direta de informações;
- 2) Não necessita de biblioteca externa,
- 3) fornece suporte para estruturação de matrizes distribuídas de até 4 dimensões com qualquer combinação de distribuição cíclica em bloco.

PCT e MDCS

O *Parallel Computing Toolbox* (PCT) junto com o *MATLAB Distributed Computing Server* (MDCS) são ferramentas oferecidas pela MathWorks. Enquanto o próprio software Matlab principal suporta multithreading⁴, o PCT fornece funcionalidade para executar código Matlab em processadores multicore e clusters⁵. O PCT fornece funções para execução paralela de *loops for*, criação/manipulação de arrays distribuídos, bem como funções de passagem de mensagens para implementação de algoritmos paralelos refinados. A Figura 2.8 ilustra as duas *toolboxes*. O PCT será alvo de estudo deste trabalho.

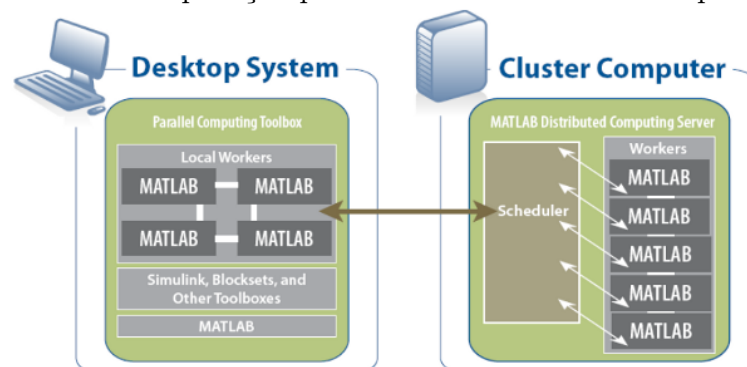
O MDCS oferece a capacidade de dimensionar algoritmos paralelos maiores para atuarem em rede de computadores, o que neste contexto se torna um cluster de comunicação, fazendo com que os diversos núcleos de todos os processadores das máquinas trabalhem na mesma solução.

São denominados **workers** no Matlab, os núcleos do processador *multicore* e são encarados como mecanismos computacionais que executam tarefas menores (*threads*)

⁴Thread é um conjunto de tarefas existentes no programa, executadas ao mesmo tempo pelo processador. Multithreading é a capacidade dos núcleos de processamento de executarem várias threads simultaneamente sem que uma interfira na outra

⁵Cluster, no contexto deste trabalho, se refere a uma arquitetura de rede capaz combinar vários computadores para trabalharem em conjunto em uma determinada tarefa

Figura 2.8: Toolbox de computação paralela e o Servidor de Computação Distribuída



Fonte: Adaptado de [14]

automaticamente em segundo plano quando acionados pelos comandos de paralelismo [14].

Neste trabalho foi empregado o laço **parfor** (comando integrante do PCT) que executa a sequência de instruções do laço de repetição em paralelo e aqui empregado analogamente a um contador síncrono, conforme ilustra a Figura 2.10. Cada execução do laço é uma iteração em que os workers realizam em qualquer ordem e independentemente umas das outras. Como cada passagem é independente, não há garantia de que as iterações sejam sincronizadas de alguma forma [14]. Se o número de workers for igual ao número de iterações do loop, cada um executará uma iteração. Se houver mais iterações do que núcleos, alguns irão executar mais de uma, recebendo várias threads de uma vez para reduzir o tempo de processamento. Ao usar comandos de paralelismo, deve-se considerar a sobrecarga paralela, que inclui a parametrização dos núcleos do processador, o tempo necessário para comunicação, coordenação e transferência de dados (envio e recebimento de informações entre os núcleos) [14, 19].

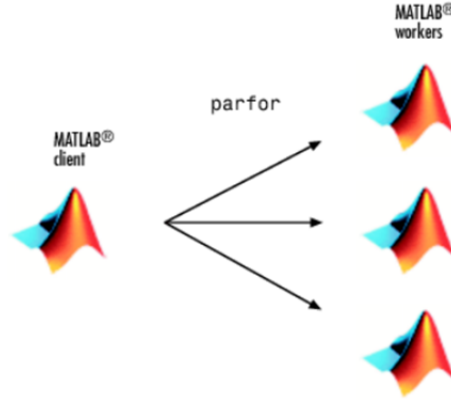
2.3.6 Lei de Amdahl

A Lei de Amdahl é usada para encontrar a máxima melhora esperada para uma aplicação quando apenas uma parte do código é paralelizada [26]. A lei possui o nome do arquiteto computacional Gene Amdahl e foi apresentada na Conferência Conjunta de Informática em 1967 [19].

A Lei de Amdahl é um modelo de *speedup*⁶ esperado sobre a relação entre a im-

⁶*Speedup* é uma medida do grau de desempenho do processamento, é a razão entre o tempo de execução sequencial e o tempo de execução em paralelo.

Figura 2.9: Loop de iteração com parfor.



Fonte: Adaptado de [15]

plementação paralelizada de um algoritmo e a sua implementação sequencial, supondo que o tamanho do problema continua a ser o mesmo quando paralelizado. Essa lei é empregada na computação paralela para prever o máximo de *speedup* teórico usando múltiplos núcleos do processador para rodar uma mesma aplicação [26], conforme Equação (2.31).

$$S = \frac{1}{(1 - F) + \frac{F}{N}} \quad (2.31)$$

onde:

S: *speedup* teórico, valores entre 1 e 2;

F: fração do código que pode ser paralelizável, valores entre 0 e 1;

N: número de núcleos do processador executando o paralelismo.

Por exemplo: em uma aplicação, somente 30% do código é paralelizável e está rodando em processador *Quad-Core* (4 núcleos):

$$S = \frac{1}{(1 - 0,3) + \frac{0,3}{4}} \quad (2.32)$$

$$S = 1,29$$

De acordo com o resultado $S = 1,29$, a aplicação terá speedup de 29%, ou seja, a velocidade de execução tenderá a aumentar conforme esse percentual. No caso hipotético de infinitos núcleos de processamento ($N \rightarrow \infty$) tem-se:

$$S = \frac{1}{(1 - 0,3) + \frac{0,3}{\infty}} \quad (2.33)$$

$$S = 1,43$$

Com 4 núcleos, segundo a Lei de Amdahl, tem-se um speedup de 29%; com infinitos núcleos, 43%. Na literatura acadêmica e científica há uma série de estudos

demonstrando que aumentar o número de núcleos de processamento (e respectivamente as tarefas executadas em paralelo) não resulta em proporção direta de aumento de velocidade [26].

Declarações semelhantes são aplicadas na teoria da organização da produtividade humana: se n vezes mais pessoas forem contratadas para determinada operação, esta não será concluída n vezes mais rápido [26].

Dividir uma tarefa em subtarefas que são executadas em paralelo é uma forma básica de reduzir o tempo de execução. No entanto, como observado, existe um limite prático para isso.

A Equação (2.31) fornece um indicativo do desempenho computacional em sistemas paralelos e multinúcleos, porém desconsidera que, quanto mais núcleos, mais *overhead*⁷ existe no gerenciamento destes (pelo Matlab ou qualquer outro ambiente de desenvolvimento), ou seja, sempre terá uma sobrecarga de processamento ao coordenar múltiplos núcleos. As subtarefas de uma aplicação não tem exatamente o mesmo tamanho; algumas terminam e precisam esperar por outras e, quanto mais tarefas em paralelo maior é o impacto na memória de processamento.

Conforme apresentado, existe um limiar em relação ao uso do paralelismo com o objetivo de acelerar a resposta de processamento, mas é importante destacar que não existe restrição em relação complexidade do problema abordado; ou seja, ainda que não haja um speedup considerável, o uso de núcleos trabalhando concorrentemente deve ser considerado. É importante citar também o *trade-off* de espaço-tempo que estipula o quão rápido a quantidade de uso daquele recurso (tempo ou espaço, no contexto de computação) cresce em proporção ao crescimento do tamanho da entrada, para, dessa forma, comparar as soluções de um dado problema pelo potencial de complexidade, e não apenas casos isolados do uso destas soluções [26].

2.3.7 Resumo da Revisão Bibliográfica

Neste Capítulo 2 foi apresentada a modelagem matemática do PLL analógico clássico e contextualização da programação paralela, pois ambos os conceitos serão empregados nos Capítulos 4 e 5 deste trabalho. O modelo matemático servirá de base para a construção do algoritmo e o paralelismo irá ditar a forma de escrita do código, visando o uso do recurso computacional.

⁷Overhead, na computação, é considerado qualquer processamento ou armazenamento em excesso para executar uma determinada tarefa.

Capítulo 3

Sincronismo de Redes

3.1 Estratégias de distribuição de sinais de tempo

O PLL é o elemento fundamental para o desempenho das redes de distribuição e propagação de sinais de clock (tempo), pois trata-se do servomecanismo em malha fechada que ajusta automaticamente a fase de um sinal gerado localmente à fase de um sinal de entrada de referência, que normalmente é gerado a uma grande distância e sujeito a inúmeras interferências durante a transmissão [2, 8]. Cada nó da rede de sincronismo é composto de um PLL.

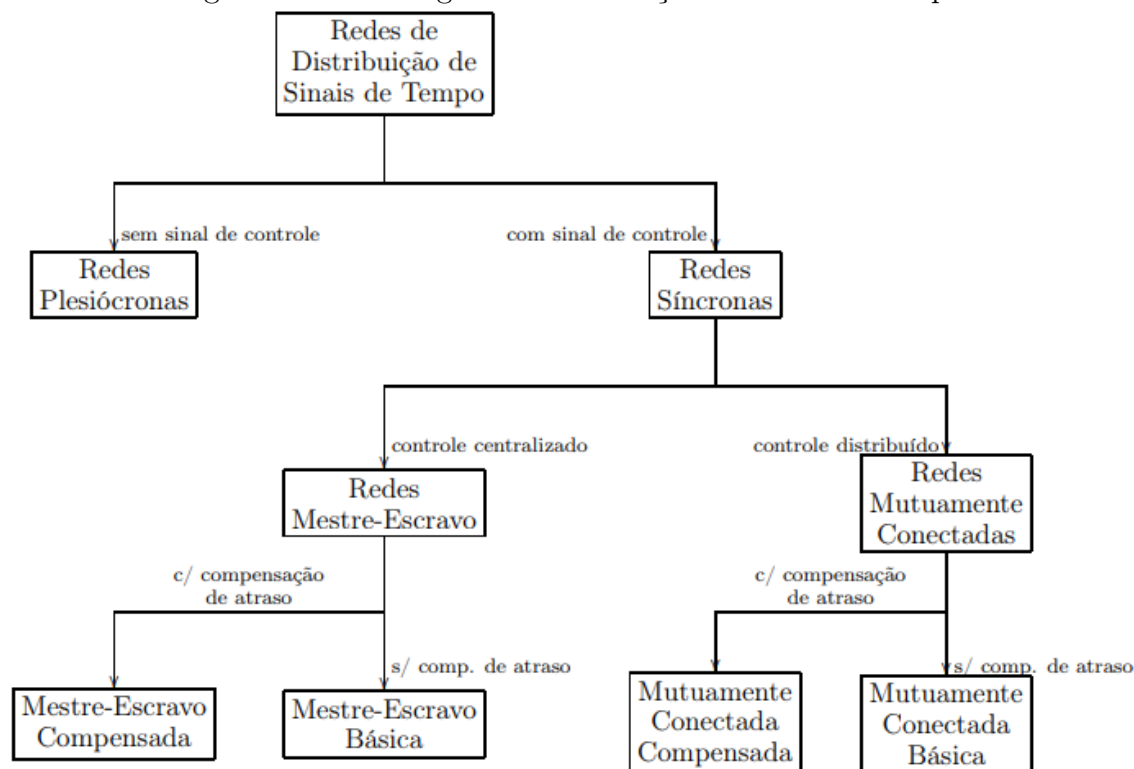
Para a questão da distribuição e propagação de sinais de tempo, foram propostos vários métodos para a sincronização de relógios (PLLs) dispersos sobre uma área geográfica [1, 2]. Esses métodos podem ser classificados de acordo com a técnica de controle. A existência ou não de um sinal de controle, define as duas classes mais comuns de redes de distribuição de sinais de tempo: as redes plesiócronas e as redes síncronas [1, 2, 7, 9].

A plesiócrona é uma estratégia em que o sinal de tempo é gerado individualmente em cada nó de forma independente, sendo o modelo mais simples, conforme Figura 3.2a.

Para as redes síncronas, todos os relógios têm suas fases e frequências enredadas e fixadas a uma fase e frequência comuns de referência. O sincronismo pode ser alcançado de diversas formas, podendo ser classificado como centralizado e descentralizado. A Figura 3.1 demonstra de forma simplificada as principais estratégias de distribuição de sinais de tempo.

As redes síncronas centralizadas utilizam estratégias Mestre-Escravo. O princípio

Figura 3.1: Estratégias de distribuição de sinais de tempo.

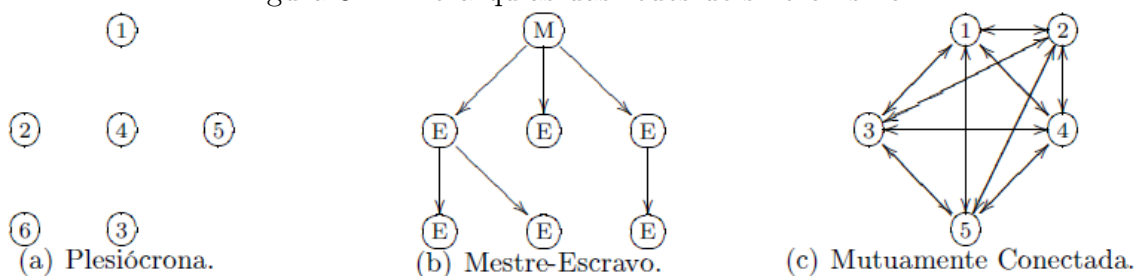


Fonte: Adaptado de [1]

é baseado na distribuição da referência de tempo por relógio Mestre a todos os demais relógios da rede (Escravos), direta ou indiretamente, conforme Figura 3.2b.

As redes síncronas descentralizadas são baseadas no princípio da sincronização mútua, não possuindo um Mestre, mas todos os nós contribuem para a determinação das escalas de tempo e frequência da rede [1, 2], com modelagens, estabilidades e comportamentos dinâmicos bem mais capciosos e complexos em comparação à estratégia centralizada, conforme Figura 3.2c.

Figura 3.2: Hierarquias das redes de sincronismo

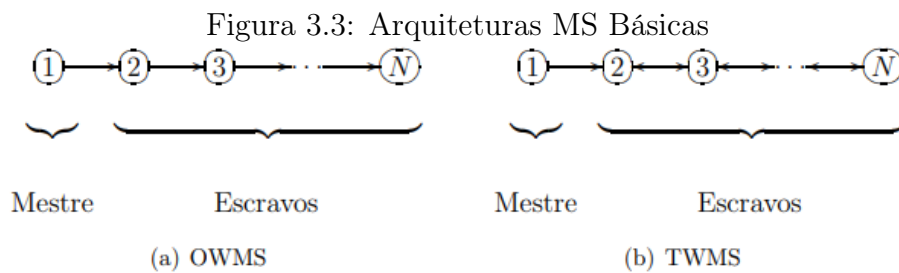


Fonte: Adaptado de [1]

3.1.1 Classificação das redes

Para modelar as redes mestre-escravo, costuma-se classificá-las, quanto ao sentido de distribuição das bases de tempo, em redes de via única (OWMS: One-Way Master-Slave) e redes de via dupla (TWMS: Two-Way Master-Slave). A Figura 3.3 mostra arquiteturas básicas das redes de via única e via dupla. Nas redes OWMS, o nó mestre tem sua base de tempo própria e independente dos demais. Os nós escravos tem base de tempo dependente, exclusivamente, de um único nó, que pode ser o mestre ou outro escravo.

As redes OWMS classificam-se, quanto a topologia, em redes em cadeia e redes em estrela. Nas redes TWMS, o nó mestre tem sua base de tempo própria, mas ajustável aos outros nós. Os nós escravos podem ter sua base de tempo dependente de mais um nó. As redes TWMS classificam-se, quanto a topologia, em redes em cadeia, em estrela ou em enlace. Ambas as topologias serão objetos de estudo neste trabalho. A Figura 3.3 mostra a arquitetura básica das MS.



Fonte: Adaptado de [1]

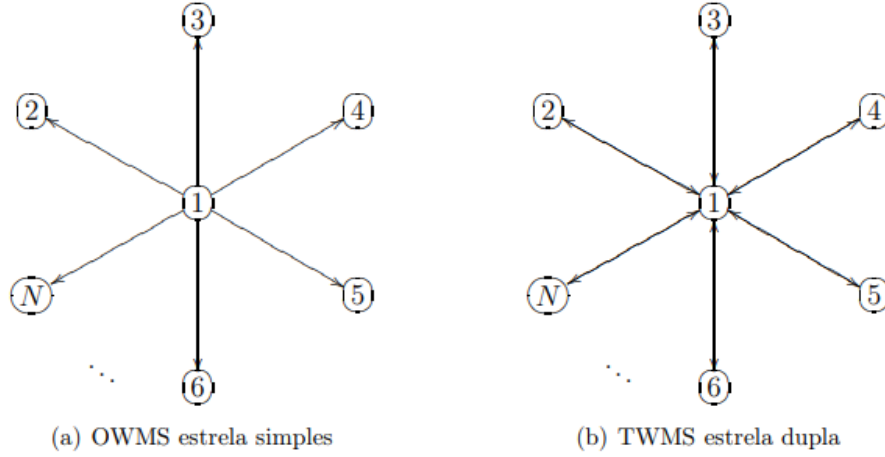
A Figura 3.4 demonstra as arquiteturas do tipo Estrela OWMS e TWMS. Esta última será objeto de simulações no Capítulo 4 deste trabalho.

A Figura 3.5 demonstra as arquiteturas do tipo Anel, onde o sinal de sincronismo retorna ao Mestre após a propagação dentre os Escravos. No Anel simples, o sinal é fornecido pelo Mestre e propagada em sequência para cada Escravo. No Anel duplo, o nó atual recebe influência do nó anterior e do nó posterior para atingir o sincronismo.

3.2 Modelo matemático da rede OWMS

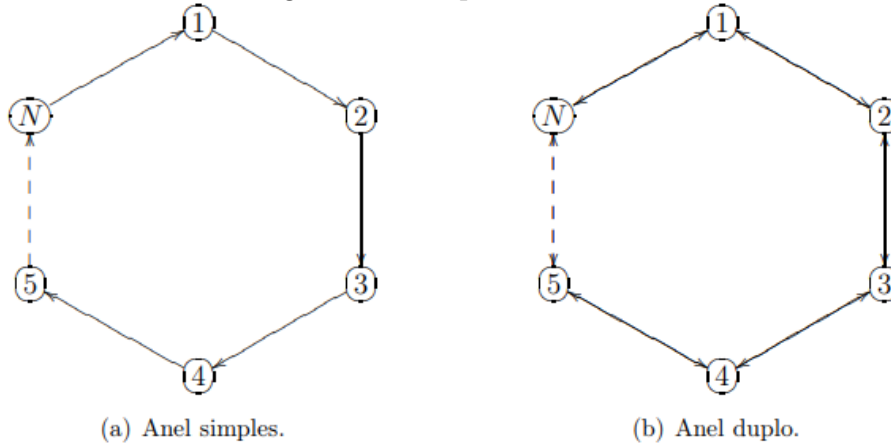
Em uma rede estruturada com forte hierarquização, do tipo Mestre-Escravo (MS),

Figura 3.4: Arquiteturas Estrela



Fonte: Adaptado de [1]

Figura 3.5: Arquiteturas Anel



Fonte: Adaptado de [1]

a posição de um nó em relação aos demais é o indicativo de sua influência na rede. Na topologia OWMS, o primeiro nó exclusivamente coordena a base de tempo para toda a rede, definindo-se assim:

- **Definição 3.1 (Nó-mestre):** Seja $j = 1, 2, \dots, N$ a designação da posição de um nó em uma rede MS. O primeiro nó da rede, designado por $j = 1$, é definido como Mestre. Todos os demais nós $1 < j \leq N$ são chamados de Escravos. Considera-se que o Relógio Mestre gera a base de tempo de forma estável e precisa. Dessa forma, o Mestre pode ser representado por seu sinal de saída na Equação (3.1):

$$v_o^{(1)} = v_o^{(1)} \cos(\omega_M(t) + \theta_o^{(1)}(t)) \quad (3.1)$$

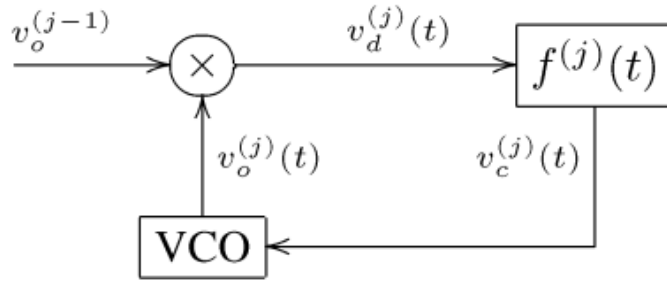
Para qualquer nó-escravo em uma rede OWMS, pode-se definir os sinais de entrada e saída de forma semelhante às equações (2.1) e (2.2). Neste trabalho, não será considerado o atraso de transmissão e propagação τ entre nós consecutivos.

- **Definição 3.2 (Sinais de entrada e saída para os nós-escravos):** Em uma rede síncrona OWMS (figura 3.3(a)), define-se os sinais de entrada e saída, conforme as Equações (3.2) e (3.3).

$$v_i^{(j)} = v_o^{(j-1)}(t) = v_o^{(j-1)} \text{sen}(\omega_M(t) + \theta_o^{(j-1)}(t) + (j-1)\frac{\pi}{2}) \quad (3.2)$$

$$v_o^{(j)} = v_o^{(j)} \text{cos}(\omega_M(t) + \theta_o^{(j)}(t) + (j-1)\frac{\pi}{2}) \quad (3.3)$$

Figura 3.6: Blocodiagrama do PLL escravo



Fonte: Adaptado de [9].

A inclusão dos termos múltiplos de $\frac{\pi}{2}$ nas Equações (3.2) e (3.3) deve-se ao fato de que cada nó gera um erro de fase de $\frac{\pi}{2}$ radiano (ou 90° de defasagem), pois o PLL analógico apresenta erro de fase estático, observando-se que a entrada é uma função seno e a saída é uma função cosseno, $\text{sen}\theta = \text{cos}(\theta - \frac{\pi}{2})$.

Assim a cada sequência de 4 PLLs (1 Mestre seguido de 3 Escravos) tem-se o atraso de 1 ciclo ($\frac{4\pi}{2} = 2\pi$), ou seja $\theta_i = \theta_o$, de forma que os nó $j = 5$ e seus múltiplos terão erro estático de fase nulo em relação ao mestre.

Considera-se que todos os nós operam com a mesma frequência de livre curso gerada pelo VCO local e, portanto, todo erro de fase se deve ao desalinhamento entre as fases e é dado em função da própria dinâmica da rede de sincronismo.

- **Definição 3.3 (Desvios de fase e frequência de um nó-escravo):** O erro de fase $\vartheta^{(j)}(t)$ de um nó-escravo é a diferença entre a fase do sinal de entrada $v_i^{(j)}$ e a fase do sinal de saída $v_o^{(j)}$ e o erro de frequência é a diferencial do erro de fase no tempo, conforme as Equações (3.4) e (3.5).

$$\vartheta^{(j)}(t) = \theta_o^{(j-1)}(t) - \theta_o^{(j)}(t) \quad (3.4)$$

$$\dot{v}^{(j)}(t) = \dot{\theta}_i^{(j-1)}(t) - \dot{\theta}_o^{(j)}(t) \quad (3.5)$$

Em uma rede OWMS, cada nó pode ser construído utilizando filtros e/ou ganhos diferentes, podendo-se definir:

- **Definição 3.4 (Função de transferência do filtro de um nó-escravo):** muito semelhante a **Observação B** da subseção 2.1.2, bastando fazer a inclusão do índice j para indicar o nó atual, visto na Equação (3.6):

$$F^{(j)}(s) = \frac{N^{(j)}(s)}{D^{(j)}(s)} \quad (3.6)$$

onde:

$$N^{(j)}(s) = \sum_{m=0}^{M^{(j)}} \alpha_m^{(j)} s^m \quad (3.7)$$

$$D^{(j)}(s) = \sum_{p=0}^{p^{(j)}} \beta_p^{(j)} s^p \quad (3.8)$$

- **Definição 3.5 (Operadores Q e L em redes síncronas):**

$$Q^{(j)}(x) = \mathcal{L}^{-1}[N^{(j)}(s)X(s)] = \sum_{m=0}^{M^{(j)}} \alpha_m^{(j)} \frac{d^m}{dt^m} x(t) \quad (3.9)$$

$$L^{(j)}(x) = \mathcal{L}^{-1}[sD^{(j)}(s)X(s)] = \sum_{p=0}^{P^{(j)}} \beta_p^{(j)} \frac{d^{p+1}}{dt^{p+1}} x(t) \quad (3.10)$$

Aplicando-se diretamente da propriedade linear da transformada de Laplace nas Equações (3.9) e (3.10).

$$Q^{(j)}(ax + by) = aQ^{(j)}(x) + bQ^{(j)}(y) \quad (3.11)$$

$$L^{(j)}(ax + by) = aL^{(j)}(x) + bL^{(j)}(y) \quad (3.12)$$

com a e $b \in \mathcal{R}$.

- **Definição 3.6 (Ganho de um nó na rede OWMS).**

$$G^{(j)} = \frac{1}{2} k_m^{(j)} k_o^{(j)} v_o^{(j-1)} v_o^{(j)} \quad (3.13)$$

- **Definição 3.7 (Saída do detector de fase do nó).**

$$v_d^{(j)}(t) = k_m^{(j)} v_o^{(j-1)}(t) v_o^{(j)}(t) \quad (3.14)$$

Dada as sete definições acima, pode-se prosseguir similarmente a modelagem da subseção 2.1.2. Substituindo as equações (3.2) e (3.3) em (3.14), juntamente com a **Definição 3.6**, obtém-se a Equação (3.15).

$$v_d^{(j)}(t) = \frac{G^{(j)}}{k_o} \begin{aligned} &\text{sen}(\omega_M(t) + \theta_o^{(j-1)}(t) + (j-1)\frac{\pi}{2}) \\ &\cos(\omega_M(t) + \theta_o^{(j)}(t) + (j-1)\frac{\pi}{2}) \end{aligned} \quad (3.15)$$

Aplicando a identidade trigonométrica da equação (2.15) e a **Definição 3.3** tem-se:

$$v_d^{(j)}(t) = \frac{G^{(j)}}{k_o} v_\delta^{(j)}(t) \quad (3.16)$$

com

$$\begin{aligned} v_\delta^{(j)}(t) &= \text{sen}(\vartheta^{(j)}(t) - \omega_M(t)) \\ &+ \text{sen}(2\omega_M(t) + \theta_o^{(j-1)}(t) + \theta_o^{(j)}(t) - \omega_M(t) + (j-1)\pi) \end{aligned} \quad (3.17)$$

A correção da fase de saída gerada pelo VCO local de cada nó é expressa pela Equação Diferencial (3.18).

$$\frac{d}{dt} \theta_o^{(j)}(t) = k_o^{(j)} v_c^{(j)}(t) \quad (3.18)$$

Sendo o sinal de saída do Filtro dado pela convolução (3.19).

$$v_c^{(j)}(t) = f^{(j)}(t) * v_d^{(j)}(t) \quad (3.19)$$

Substituindo a equação (3.19) na (3.18) tem-se a Equação (3.20).

$$\frac{d}{dt} \theta_o^{(j)}(t) = k_o^{(j)} f^{(j)}(t) * v_d^{(j)}(t) \quad (3.20)$$

Aplicando-se o Teorema da Convolução e a **Definição 3.4** em (3.21).

$$k_o^{(j)} N^{(j)}(s) V_d^{(j)}(s) = s D^{(j)}(s) \Theta_o^{(j)}(s) \quad (3.21)$$

Aplicando a Transformada Inversa de Laplace $\mathcal{L}^{-1}$ e substituindo os operadores $Q^{(j)}$ e $L^{(j)}$ da **Definição 3.5** em (3.22):

$$k_o^{(j)} Q^{(j)}[v_d^{(j)}(t)] = L^{(j)}[\theta_o^{(j)}(t)] \quad (3.22)$$

Substituindo as Equações (3.16) e (3.4) em (3.22), obtém-se (3.23):

$$k_o^{(j)} Q^{(j)}\left[\frac{G^{(j)}}{k_o^{(j)}} v_\delta^{(j)}(t)\right] = L^{(j)}[\theta_o^{(j-1)}(t) - \vartheta^{(j)}(t)] \quad (3.23)$$

Aplicando-se a **Definição 3.5** em (3.23) tem-se (3.24):

$$L^{(j)}(\vartheta^{(j)}(t)) + G^{(j)} Q^{(j)}[v_\delta^{(j)}(t)] = L^{(j)}[\theta_o^{(j-1)}(t)] \quad (3.24)$$

Substituindo-se a Equação (3.17) e reaplicando a **Definição 3.5** em (3.24), obtém-se (3.25):

$$\begin{aligned} L^{(j)}[\vartheta^{(j)}(t)] + G^{(j)}Q^{(j)}[\text{sen}(\vartheta^{(j)}(t) - \omega_M(t))] &= L^{(j)}[\theta_o^{(j-1)}(t)] \\ -G^{(j)}Q^{(j)}[\text{sen}(2\omega_M t + \theta_o^{(j-1)}(t) + \theta_o^{(j)}(t) - \omega_M(t) + (j-1)\pi)] & \end{aligned} \quad (3.25)$$

Aplicando a identidade trigonométrica da equação (2.15) na segunda parcela do membro direito da Equação (3.25) e substituindo, tem-se (3.26):

$$\begin{aligned} L^{(j)}[\vartheta^{(j)}(t)] + G^{(j)}Q^{(j)}[\text{sen}(\vartheta^{(j)}(t) - \omega_M(t))] &= L^{(j)}[\theta_o^{(j-1)}(t)] \\ +(-1)^{(j)}G^{(j)}Q^{(j)}[\text{sen}(2\omega_M t + \theta_o^{(j-1)}(t) + \theta_o^{(j)}(t) - \omega_M(t))] & \end{aligned} \quad (3.26)$$

Isolando $\theta_o^{(j)}(t)$ na equação (3.4) e substituindo em (3.26), tem-se um modelo matemático para os nós-escravos de uma rede OWMS em (3.27):

$$\begin{aligned} L^{(j)}[\vartheta^{(j)}(t)] + G^{(j)}Q^{(j)}[\text{sen}(\vartheta^{(j)}(t))] &= L^{(j)}[\theta_o^{(j-1)}(t)] \\ +(-1)^{(j)}G^{(j)}Q^{(j)}[\text{sen}(2(\omega_M t + \theta_o^{(j-1)}(t)) - \vartheta^{(j)}(t))] & \end{aligned} \quad (3.27)$$

A Equação (3.27) relaciona o erro de fase de cada nó-escravo à fase do nó anterior, considerando o erro estático de $\frac{\pi}{2}$ rad. As mesmas simplificações realizadas na Equação (2.26) (eliminação do termo de frequência dupla e seno linearizado) também são aplicadas na (3.28):

$$L^{(j)}[\vartheta^{(j)}(t)] + G^{(j)}Q^{(j)}[\vartheta^{(j)}(t)] = L^{(j)}[\theta_o^{(j-1)}(t)] \quad (3.28)$$

3.3 Modelo matemático da rede TWMS

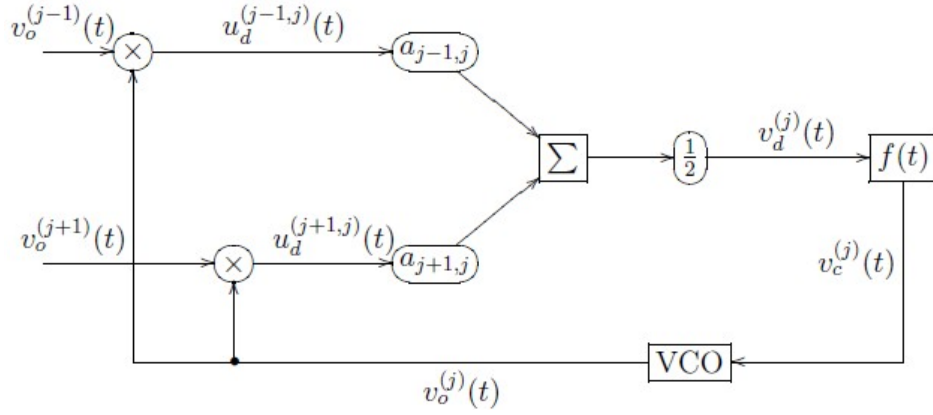
A Figura 3.7 apresenta o diagrama de blocos de uma rede TWMS, onde cada nó compara as fases dos nós anterior e posterior com a sua própria fase, tendo a necessidade de um comparador de fase a mais e realizar a somatória dos sinais.

Para o modelo matemática da rede TWMS, considera-se que o relógio mestre gera uma base de tempo de forma estável e precisa, tendo sua saída representada conforme a Equação (3.29).

$$v_o^{(1)}(t) = v_o^{(1)} \cos(\omega_M t + \theta_o^{(1)}(t)) \quad (3.29)$$

onde, a representação $^{(1)}$ indica o sinal do mestre.

Figura 3.7: Diagrama de blocos dos PLL de uma rede TWMS.



Fonte: Adaptado de [1].

Na rede TMWS, o atraso produzido pelos nós está associado à média ponderada dos sinais de entrada e, para esta modelagem, irá se considerar que as diferenças de fases produzidas por cada escravo j fazem parte da estimativa da fase $\theta_o^{(j)}$ [1].

- **Definição 3.8** (Sinais de entrada e saída para nós-escravos): Em sincronismo, os sinais de entrada e saída são definidos pelas Equações (3.30) e (3.31).

$$v_o^{(l)}(t) = v_i^{(l)} \sin(\omega_M(t) + \theta_o^{(l)}(t)) \quad (3.30)$$

$$v_o^{(l)}(t) = v_o^{(l)} \cos(\omega_M t + \theta_o^{(l)}(t)) \quad (3.31)$$

com $l = j - 1, j + 1$.

É importante observar que todos os nós possuem a mesma frequência de livre curso do VCO, ω_M , porém, por possuir duas influências, é necessário avaliar os erros de fase e frequência.

- **Definição 3.9** (Erros de fase e frequência do nó-escravo): Para o erro de fase $\vartheta^{(l,j)}(t)$ do nó-escravo, deve-se considerar a diferença entre a fase do sinal de entrada $v_o^{(l)}(t)$ e a fase do sinal de saída $v_o^{(j)}(t)$, com $l = j - 1, j + 1$. O erro de frequência é composta pela taxa de variação do erro de fase em relação ao tempo, conforme apresentado nas Equações (3.32) e (3.33), respectivamente.

$$\vartheta^{(l,j)}(t) = \theta_o^{(l)}(t) - \theta_o^{(j)}(t) \quad (3.32)$$

$$\dot{v}^{(l,j)}(t) = \dot{\theta}_o^{(l)}(t) - \dot{\theta}_o^{(j)}(t) \quad (3.33)$$

Para o modelo TWMS, as **Observação B** e a **Observação C**, apresentadas anteriormente no Capítulo 2 representam os operadores $Q^{(j)}$ e $L^{(j)}$.

- **Definição 3.10** (Ganho associado a cada entrada): O ganho associado a um nó de uma rede TWMS pode ser definido conforme a Equação (3.34).

$$G^{(l,j)} = \frac{1}{2} a_{l,j} k_m^{(j)} k_o^{(j)} v_o^{(j)} v_o^{(l)} \quad (3.34)$$

onde, $l = j - 1, j + 1$. Os ganhos do detector de fase e do VCO são expressos por $k_m^{(j)}(V/rad)$ e $k_o^{(j)}(V/rad)$ respectivamente. Os termos $v_o^{(j)}$ e $v_o^{(l)}$ são as amplitudes dos sinais de entrada e saída do nó j , respectivamente. O coeficiente $a_{l,j}$ é um ponderador para a saída de cada multiplicador.

Como representado na figura 3.7, o detector de fase é constituído por dois multiplicadores e por um somador. que realiza uma ponderação sobre as diferenças de fase. Dessa forma, o termo $v_d^{(j)}(t)$ é positivo, se a média ponderada dos erros de fase for positiva e negativo caso contrário. Universalizando, tem-se a Equação (3.35).

$$v_d^{(j)}(t) = \frac{1}{2} \sum_{l=j-1, l \neq j}^{j+1} a_{l,j} u_d^{(l,j)}(t) \quad (3.35)$$

com

$$\frac{1}{2} \sum_{l=j-1, l \neq j}^{j+1} a_{l,j} = 1 \quad (3.36)$$

e

$$u_d^{(l,j)}(t) = k_m^{(l)} v_o^{(l)}(t) v_o^{(j)}(t) \quad (3.37)$$

Substituindo as Equações 3.30 e 3.31, apresentadas na **Definição 3.11**, na Equação (3.37), obtém-se (3.38):

$$u_d^{(l,j)}(t) = \frac{G^{(l,j)}}{a_{l,j} k_o^{(j)}} \sin(\omega_m(t) + \theta_o^{(l)}(t)) \cos(\omega_M t + \theta_o^{(j)}(t)) \quad (3.38)$$

Aplicando a identidade trigonométrica apresentada da Equação (2.15) e a **Definição 3.12** na equação (3.38), tem-se:

$$u_d^{(l,j)}(t) = \frac{G^{(l,j)}}{a_{l,j}k_o^{(j)}}(t) \quad (3.39)$$

com,

$$u_\delta^{(l,j)}(t) = \text{sen}(\vartheta^{(l,j)}(t)) + \text{sen}(2\omega_M t + \theta_o^{(l)}(t) + \theta_o^{(l)}(t)) \quad (3.40)$$

A taxa de variação da fase de saída $\theta_o^{(j)}(t)$ do nó escravo j é determinada através da lei de controle da Equação (3.41).

$$\frac{d}{dt}\theta_o^{(j)}(t) = k_o^{(j)}v_c^{(j)}(t) \quad (3.41)$$

E o sinal de controle $v_c^{(j)}$ pode ser expresso pela Convolução (3.42):

$$v_c^{(j)}(t) = f^{(j)}(t) * v_d^{(j)}(t) \quad (3.42)$$

Substituindo a Equação (3.42) em (3.41), tem-se (3.43):

$$\frac{d}{dt}\theta_o^{(j)}(t) = k_o^{(j)}f^{(j)}(t) * v_d^{(j)}(t) \quad (3.43)$$

Aplicando o teorema da convolução e a **Observação B** na expressão 3.43, obtém-se (3.44):

$$k_o^{(j)}N^{(j)}(s)V_d^{(j)}(s) = sD^{(j)}(s)\Theta_o^{(j)}(s) \quad (3.44)$$

Para Equação 3.44 e aplicando-se a transformada inversa de Laplace na expressão em conjunto com os operadores $Q^{(j)}$ e $L^{(j)}$ da **Observação C**, obtém-se:

$$k_o^{(j)}Q^{(j)}[v_d^{(j)}(t)] = L^{(j)}[\theta_o^{(j)}] \quad (3.45)$$

A partir do erro de fase, apresentado na Equação (3.32), tem-se:

$$\vartheta^{(j-1,j)}(t) = \theta_o^{(j-1)}(t) - \theta_o^{(j)}(t) \quad (3.46)$$

e,

$$\vartheta^{(j+1,j)}(t) = \theta_o^{(j+1)}(t) - \theta_o^{(j)}(t) \quad (3.47)$$

Somando-se as Equações (3.46) e (3.47) e isolando saída $\theta_o^{(j)}(t)$, tem-se em (3.48):

$$\theta_o^{(j)}(t) = \frac{1}{2} \sum_{l=j-1, l \neq j}^{j+1} \theta_o^{(l)}(t) - \frac{1}{2} \sum_{l=j-1, l \neq j}^{j+1} \vartheta^{(j,l)}(t) \quad (3.48)$$

Substituindo-se a Equação (3.48) em (3.45) e considerando-se a **Observação C**, obtém-se:

$$L^{(j)} \left[\frac{1}{2} \sum_{l=j-1, l \neq j}^{j+1} \vartheta^{(l,j)}(t) \right] + k_o^{(j)} Q^{(j)} [v_d^{(j)}(t)] = L^{(j)} \left[\frac{1}{2} \sum_{l=j-1, l \neq j}^{j+1} \theta_o^{(j)}(t) \right] \quad (3.49)$$

Substituindo-se as Equações (3.35) e (3.39) em (3.49), tem-se em (3.50):

$$L^{(j)} \left[\sum_{l=j-1, l \neq j}^{j+1} \vartheta^{(l,j)}(t) \right] + Q^{(j)} \left[\sum_{l=j-1, l \neq j}^{j+1} G^{(l,j)} u_\delta^{(l,j)}(t) \right] = L^{(j)} \left[\sum_{l=j-1, l \neq j}^{j+1} \theta_o^{(j)}(t) \right] \quad (3.50)$$

E substituindo-se em (3.50) a Equação (3.40), tem-se:

$$\begin{aligned} & L^{(j)} \left[\sum_{l=j-1, l \neq j}^{j+1} \vartheta^{(l,j)}(t) \right] + Q^{(j)} \left[\sum_{l=j-1, l \neq j}^{j+1} G^{(l,j)} \text{sen}(\vartheta^{(l,j)}(t)) \right] = \\ & - Q^{(j)} \left[\sum_{l=j-1, l \neq j}^{j+1} G^{(l,j)} \text{sen}(2\omega_M t + \theta_o^{(l)}(t) + \theta_o^{(j)}(t)) \right] \\ & + L^{(j)} \left[\sum_{l=j-1, l \neq j}^{j+1} \theta_o^{(l)}(t) \right] \end{aligned} \quad (3.51)$$

Isolando-se a fase de saída $\theta_o^{(j)}$ na Equação (3.32) e substituindo-se em (3.51), em (3.52) tem-se:

$$\begin{aligned} & L^{(j)} \left[\sum_{l=j-1, l \neq j}^{j+1} \vartheta^{(l,j)}(t) \right] + Q^{(j)} \left[\sum_{l=j-1, l \neq j}^{j+1} G^{(l,j)} \text{sen}(\vartheta^{(l,j)}(t)) \right] = \\ & - Q^{(j)} \left[\sum_{l=j-1, l \neq j}^{j+1} G^{(l,j)} \text{sen}(2(\omega_M t + \theta_o^{(l)}(t)) - \vartheta_o^{(l,j)}(t)) \right] \\ & + L^{(j)} \left[\sum_{l=j-1, l \neq j}^{j+1} \theta_o^{(l)}(t) \right] \end{aligned} \quad (3.52)$$

Simplificando as expressões, conforme proposto por Bueno em [1], obtém-se o conjunto de (3.53) a (3.55):

$$\begin{aligned}
& L^{(j)} \left[\sum_{l=j-1, l \neq j}^{j+1} \vartheta^{(l,j)}(t) \right] + Q^{(j)} \left[\sum_{l=j-1, l \neq j}^{j+1} G^{(l,j)} \text{sen}(\vartheta^{(l,j)}(t)) \right] = \\
& - Q^{(j)} \left[\sum_{l=j-1, l \neq j}^{j+1} G^{(l,j)} \text{sen}(2(\omega_M t + \theta_o^{(l)}(t)) - \vartheta_o^{(l,j)}(t)) \right] \\
& + L^{(j)} \left[\sum_{l=j-1, l \neq j}^{j+1} \theta_o^{(l)}(t) \right]
\end{aligned} \tag{3.53}$$

$$L^{(j)} \left[\sum_{l=j-1, l \neq j}^{j+1} \vartheta^{(l,j)}(t) \right] + Q^{(j)} \left[\sum_{l=j-1, l \neq j}^{j+1} G^{(l,j)} \text{sen}(\vartheta^{(l,j)}(t)) \right] = L^{(j)} \left[\sum_{l=j-1, l \neq j}^{j+1} \theta_o^{(l)}(t) \right] \tag{3.54}$$

$$L^{(j)} \left[\sum_{l=j-1, l \neq j}^{j+1} \vartheta^{(l,j)}(t) \right] + Q^{(j)} \left[\sum_{l=j-1, l \neq j}^{j+1} G^{(l,j)}(\vartheta^{(l,j)}(t)) \right] = L^{(j)} \left[\sum_{l=j-1, l \neq j}^{j+1} \theta_o^{(l)}(t) \right] \tag{3.55}$$

Na expressão (3.53) é desconsiderado o atraso de transmissão; a expressão (3.54), é desconsiderado o termo de de frequência dupla e, por fim, em (3.55) são linearizados os termos que contém a função $\text{sen}(\vartheta^{(l,j)}(t))$. Dessa forma, esse encadeamento de equações e expressões matemáticas perfazem a representação de cada nó-escravo de uma rede TWMS com $j = 2, 3, \dots, N$.

3.4 Estabilidade e sincronização de redes

- **Definição 3.11 (Estabilidade no sentido de Lyapunov):** Seja x_0 um ponto de equilíbrio da equação de estados autônoma $\dot{x} = X(x)$. Então x_0 é estável no sentido de Lyapunov [27], se: dado $\epsilon > 0$ existe $\delta = \delta(\epsilon) > 0$ tal que, para qualquer outra solução $y(t)$ satisfazendo $|x_0 - y(t)| < \delta$, tem-se $|x_0 - y(t)| < \epsilon$ para $t > t_0$, $t_0 \in \mathcal{R}^+$.

- **Definição 3.12 (Estabilidade assintótica):** Seja x_0 um ponto de equilíbrio da equação de estados autônoma $\dot{x} = X(x)$. Então, x_0 é dito assintoticamente estável se é estável no sentido de Lyapunov [27] e se existe uma constante $b > 0$ tal que se $|x_0 - y(t_0)| < b$ implica no limite de $t \rightarrow \infty$ para $|x_0 - y(t)| = 0$.

- **Definição 3.13 (Estado Síncrono):** O estado síncrono $\vartheta_s^{(j)} \in R^{P+1}$ é um ponto de equilíbrio assintoticamente estável da equação (3.28).

- **Definição 3.14 (Sincronização):** Seja $\vartheta_s^{(j)}$ um estado síncrono e $\vartheta^{(j)}(t)$ a solução da equação (3.28). Se $\vartheta^{(j)}(t) \subset V_1(\vartheta_s^{(j)}, \epsilon_1)$ e $(d/dt)\theta_o^{(j)}(t) \subset V_2(c, \epsilon_2)$ assegura que para todo $t > t_s$, $t_s < \infty$, com $\epsilon_1, \epsilon_2 > 0$ e $|c| < \infty$, $c \in \mathcal{R}$, então o nó j estará sincronizado. Se o erro de fase oscila em torno de um valor constante de forma limitada, o nó escravo é considerado sincronizado, o que significa que, para fins práticos, o nó escravo atingiu o estado síncrono.

Uma forma de se observar o comportamento das redes é através de simulações. Como visto nas definições acima, apesar da impossibilidade de generalização, pode-se tentar inferir, através de um conjunto de simulações, se uma determinada rede consegue alcançar o estado síncrono, observando-se como as trajetórias se comportam dentro de uma região limitada e fixa.

3.5 Resumo do Capítulo

Neste capítulo foram apresentadas as estratégias de distribuição de sinais que serão responsáveis pela organização da rede e as modelagens das redes de via única e via dupla que serão critérios para a recepção do sinal por cada nó envolvido nas simulações do Capítulo 4. Também foram apresentados os conceitos de estabilidade e sincronização que serão discutidos sobre os resultados gráficos produzidos pelas simulações a seguir.

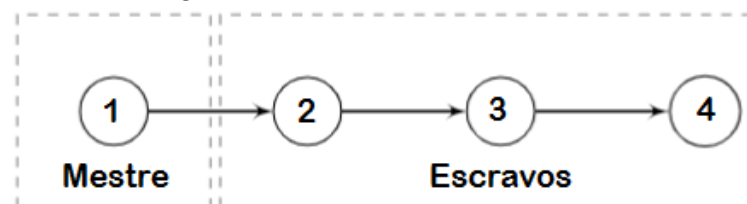
Capítulo 4

Simulação de Redes de Sincronismo com computação paralela

4.1 Algoritmo proposto para a Simulação Piloto

Para o desenvolvimento da simulação foram seguidas algumas etapas necessárias para a execução da rede OWMS. Primeiramente, foi considerada uma rede com 4 PLLs, sendo 1 Mestre e 3 Escravos.

Figura 4.1: Rede OWMS simulada



Fonte: autoria própria.

Primeiramente, é realizada a comparação de fase pelo PD do sinal de entrada, oriundo do anterior, com sinal de saída do VCO local do próprio PLL. Com isso, é produzido um sinal de erro (ou desvio) que é processado pela dinâmica do Filtro local e produz um sinal de controle para o oscilador fazer a devida compensação. O algoritmo segue algumas etapas:

1. Parâmetros de simulação: número de PLLs, tempo inicial e final, vetor de tempo e número de pontos de integração numérica (Apêndice A1);

2. Parâmetros de PLL: amplitudes das senoides, ganhos do detector de fase e do filtro , frequência angular de livre curso VCO e sinal de excitação;
3. Projetar a rede de sincronismo (OWMS ou TWMS): fazer um seletor para identificar a condição do PLL1 (Mestre) e dos demais PLLs (Escravos), que identificam os blocos que devem receber sinal do PLL anterior;
4. Executar a Rede em processamento paralelo (*loop parfor*);
5. Receber os valores calculados da simulação em variáveis auxiliares para tratamento de processamento paralelo, compilar os resultados e construir os gráficos.

O pseudocódigo da solução proposta é:

Entrada: $t_0, y_0, h, n, q, kVi, kVo, K_m, K_o, \omega_m, \theta_i$

Saída: $v_{c(j+1)k}, \theta_{o(j+1)k}$

Escreva t_0 e y_0

para $j = 1, j \leq n, j + 1$ **faça**

Escreva $v_d(j)$ recebendo $v_i(j)$ e $v_o(j)$

para $k = 1, k \leq q, k + 1$ **faça**

$$L_1 = v_{djk} - v_{cjk}$$

$$L_2 = v_{djk} - v_{cjk} + 0.5 * L_1$$

$$L_3 = v_{djk} - v_{cjk} + 0.5 * L_2$$

$$L_4 = v_{djk} - v_{cjk} + L_3$$

$$v_{c(j+1)k} = v_{cjk} + \frac{h}{6} * (L_1 + 2 * L_2 + 2 * L_3 + L_4)$$

$$L_1 = k_o * v_{cjk}$$

$$L_2 = k_o * v_{cjk} + 0.5 * L_1$$

$$L_3 = k_o * v_{cjk} + 0.5 * L_2$$

$$L_4 = k_o * v_{cjk} + L_3$$

$$\theta_{o(j+1)k} = \theta_{ojk} + \frac{h}{6} * (L_1 + 2 * L_2 + 2 * L_3 + L_4)$$

fim para

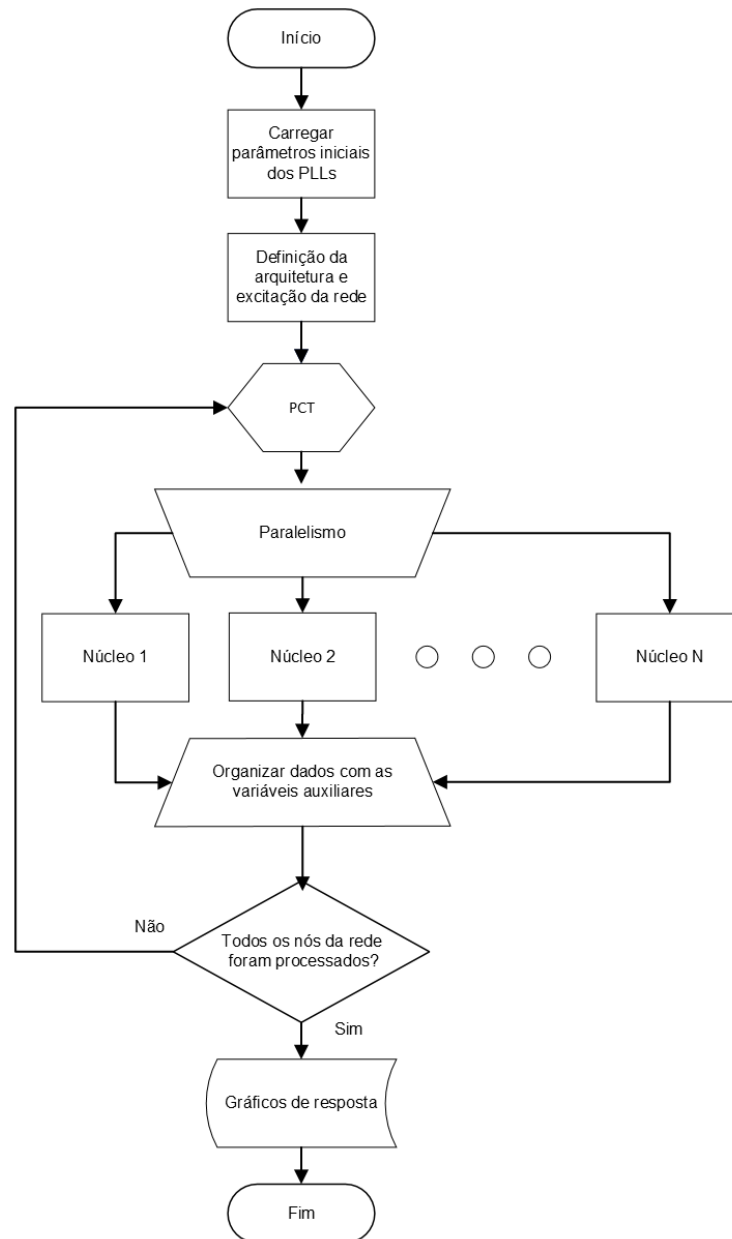
$v_0(j + 1)$ recebe $\theta_{o(j+1)k}$ e $v_{o(j+1)k}$

fim para

Para o paralelismo computacional, foi empregado o TPC do Matlab (ver subseção 2.3.5) e a estrutura de repetição *parfor*. Dentro do laço de integração, a primeira etapa é ajustar a rede, identificando o Mestre e os Escravos por meio dos cálculos dos parâmetros de entrada. Após isso, são calculas as respostas em paralelo conforme o controle do PCT do Matlab e a instrução *parfor* direciona a um núcleo do processa-

dor, as expressões para encontrar a saída de cada um dos PLLs da rede, fazendo essa tarefa de forma automática, produzindo o paralelismo. o PCT gerencia a paralelização, calculando os valores de $v_{c(i)}$ $\theta_o(i)$ para cada um dos PLLs da rede, sem que o programador interfira no arranjo e controle dos dos núcleos [19]. A simulação, por possuir uma estrutura iterativa que não é dependente de resultados encontrados por outras iterações, torna-se viável a aplicação do paralelismo no cálculo dos valores da saída para se calcular a integral numérica. Cada nó da rede é processado de forma independente, conforme é demonstrado no Fluxograma da Figura 4.2.

Figura 4.2: Fluxograma de operação.



Fonte: autoria própria.

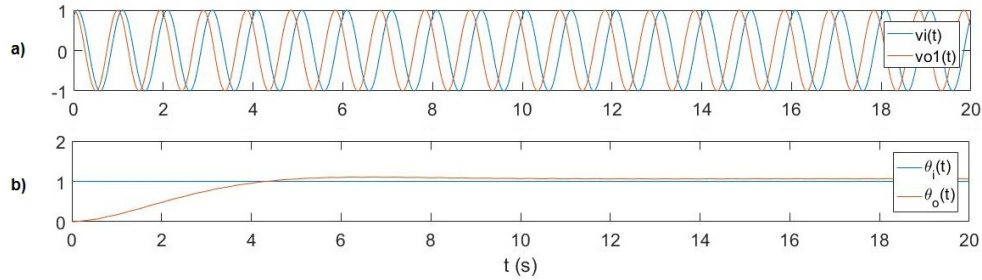
4.2 Resultados Numéricos da Simulação Piloto

4.2.1 Excitação em Degrau de Fase

A simulação piloto foi realizada em uma rede em cadeia OWMS com 1 PLL Mestre e 3 PLLs escravos, Filtro de 1ª Ordem $F(s) = \frac{1}{s+1}$ e entrada de excitação em degrau na fase $\theta_i(t) = 1$ radiano durante 20 segundos. Cada Figura (de 4.3 a 4.6) está dividida em duas partes:

- (a) sinal de saída do PLL Escravo;
- (b) defasagem em relação ao Mestre.

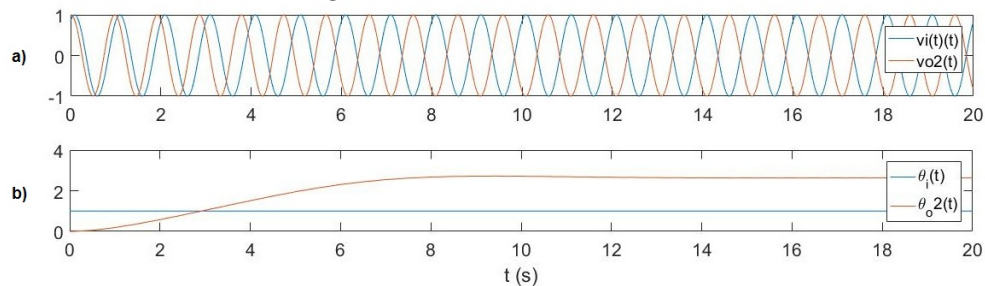
Figura 4.3: Saída do 1º PLL.



Fonte: autoria própria.

É possível observar na Figura 4.3a, a saída do 1º PLL está com o sinal defasado em $\frac{\pi}{2}$ rad em relação ao sinal de entrada produzido pelo próprio VCO, o que comprova as equações (3.2) e (3.3). A Figura 4.3b ilustra o degrau de excitação de 1 radiano.

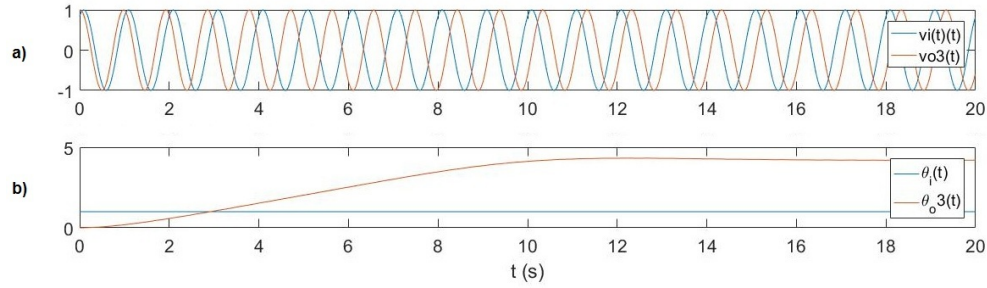
Figura 4.4: Saída do 2º PLL.



Fonte: autoria própria.

Na Figura 4.4a, a saída do 2º PLL está com o sinal defasado em π rad (180°) em relação ao sinal de entrada vindo do PLL anterior, ou seja, subtraiu-se $\frac{\pi}{2}$ rad, ocorrendo a inversão de sinal em relação ao sinal de entrada gerado no Mestre.

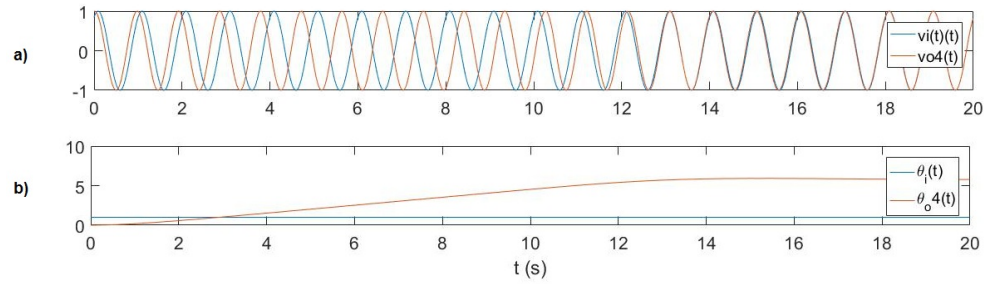
Figura 4.5: Saída do 3º PLL.



Fonte: autoria própria.

Na Figura 4.5a, a saída do 3º PLL está com o sinal defasado em $\frac{3\pi}{2}$ rad (270°) em relação ao sinal de entrada, subtraindo-se da saída do anterior $\frac{\pi}{2}$ rad.

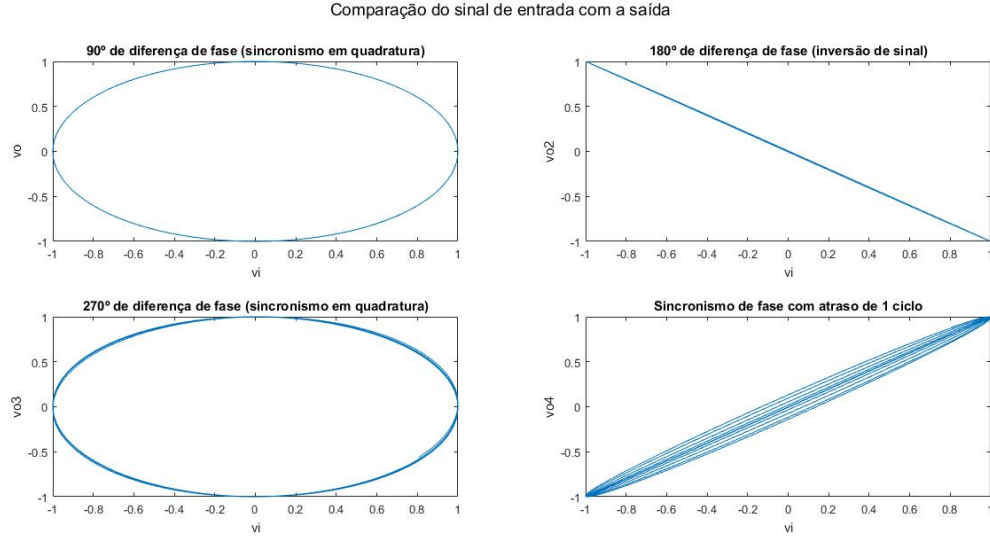
Figura 4.6: Saída do 4º PLL.



Fonte: autoria própria.

Na Figura 4.6a, a saída do 4º PLL está com o sinal defasado em 2π rad (360°), 1 ciclo completo, em relação ao sinal de entrada, comprovando a **Definição 3.2**. Percebe-se que a simulação piloto, utilizando paralelismo computacional, não afetou a resposta esperada para a rede de PLLs, possibilitando o ensaio de outros tipos de arquiteturas e Filtros Controladores conforme será apresentado na sequência deste trabalho.

Figura 4.7: Lissajous de Fase.



Fonte: autoria própria.

Considerando-se que o sinal propagado e o VCO local de cada nó são oscilações harmônicas simples, com mesma amplitude e frequência, diferenciando-se somente em fase, pode-se correlacionar as trajetórias através de uma Curva de Lissajous, conforme bibliografia [29] e Apêndice A.2. Essa curva será fechada se o movimento repete-se em intervalos regulares de tempo, com a sincronia dos osciladores. Isso é possível pois o oscilador mestre e cada VCO tem a mesma frequência angular e são comensuráveis, com razão igual a 1 (Apêndice A.2). Se a relação entre as frequências angulares não fosse uma fração racional, a curva seria aberta, ou seja, não haveria o sincronismo desejado [28, 29]. A Figura 4.7 comprova a estabilidade e a sincronização da Rede OWMS conforme as **Definições de 3.11 a 3.14**.

Tabela 4.1: Valores de erro de fase dos PLLs - Rede OWMS com um Mestre e três PLLs Escravos

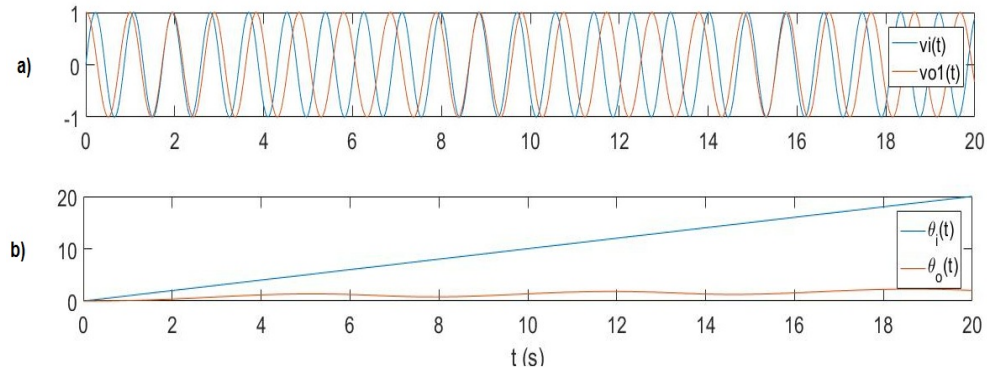
Nó PLL	$\vartheta(rad)$	$\vartheta(^{\circ})$
PLL 1	-1,5768	-90° 20'
PLL Escravo 2	-3,1353	-179° 38'
PLL Escravo 3	-4,7142	-270° 6'
PLL Escravo 4	-6,2852	-360° 6'

4.2.2 Excitação em Rampa de Fase

A simulação foi realizada em uma rede em cadeia OWMS com 1 PLL Mestre e 3 PLLs escravos, Filtro de 1ª Ordem $F(s) = \frac{1}{s+1}$ e entrada de excitação em rampa unitária na fase $\theta_i(t) = 1 \text{ rad/s}$ durante 20 segundos. Cada Figura (de 4.8 a 4.11) está dividida em duas partes:

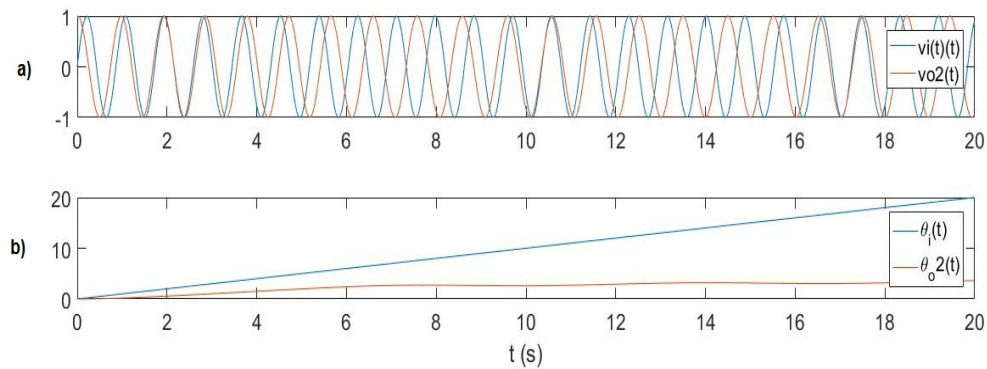
- (a) sinal de saída do PLL Escravo em relação ao Mestre;
- (b) defasagem angular em relação ao Mestre (erro de fase).

Figura 4.8: Saída do 1º PLL.



Fonte: autoria própria.

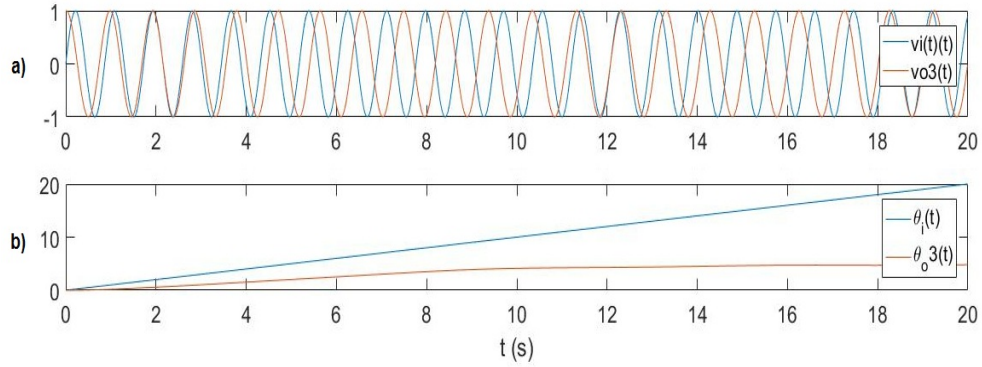
Figura 4.9: Saída do 2º PLL.



Fonte: autoria própria.

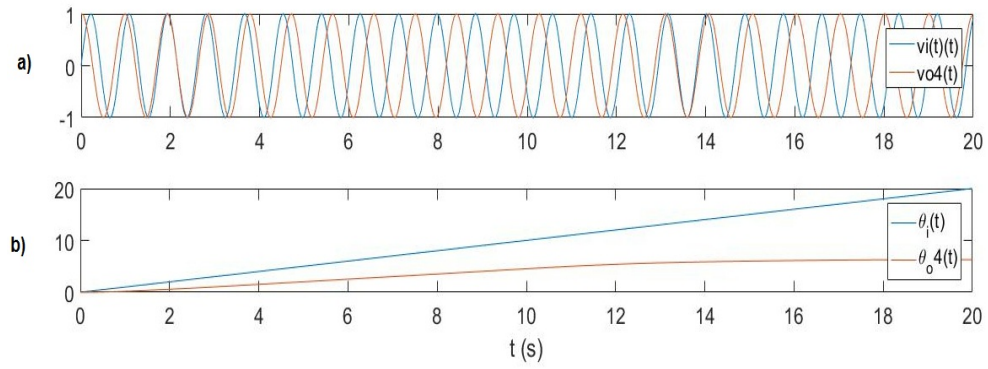
Nenhuma das Figuras dos sinais de saída dos PLLs (4.8 a 4.11) indicam sincronismo, seja em fase ou em quadratura. Para a excitação em rampa, não se consegue sincronizar a rede OWMS com o oscilador Mestre recebendo uma excitação crescente. Para o Filtro $F(s) = \frac{1}{s+1}$ a saída dos PLLs não conseguiu acompanhar a entrada, o que era esperado e ajuda a validar a Simulação Piloto como programação paralela. Na

Figura 4.10: Saída do 3º PLL.



Fonte: autoria própria.

Figura 4.11: Saída do 4º PLL.

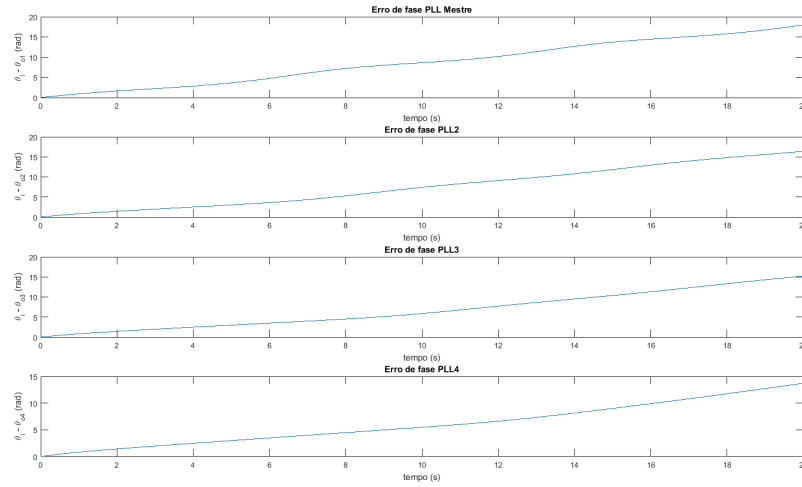


Fonte: autoria própria.

subseção 4.3, serão propostos outros Filtros para que se consiga atingir a sincronia ou estado síncrono para a rede OWMS com excitação em rampa unitária.

Analisando-se o erro de fase $\vartheta(t)$ na Figura 4.12, percebe-se que a excitação em rampa é propagada as fases de saída dos PLLs escravos, não provendo sincronismo.

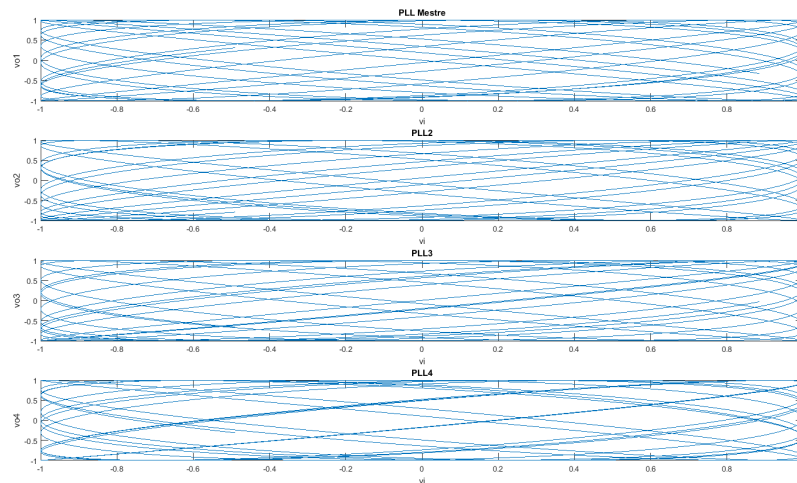
Figura 4.12: Erro de fase para a excitação em rampa.



Fonte: autoria própria.

Ao plotar a Curva de Lissajous das fases de saída na Figura 4.13, comparando-se com a entrada, confirma-se que a rede não entrou em sincronismo e não atendeu as **Definições de 3.11 a 3.14**. Para ajustar a resposta, na sequência será proposto o uso de dinâmicas mais refinadas para o Filtro.

Figura 4.13: Lissajouse para a excitação em rampa.



Fonte: autoria própria.

4.2.3 Comparação de Tempos da Simulação Piloto

Tabela 4.2: Comparação de tempos de simulação para a excitação em Degrau

Processador	Serial	Paralelo
Intel Core i5-4440S Quad-Core 2,80GHz	0,0391s	147,1245s
Intel Core i3-5005S Dual-Core 2,00GHz	0,0641s	200,3698s

Conforme observado na Tabela 4.2, a programação serial é mais rápida em relação a paralela. Observa-se também que o aumento de núcleos de processamento produziu uma resposta mais rápida, comparando-se o *Intel i3 Dual Core* com o *Intel i5 Quad Core*, conforme previsto na Lei de Amdahl.

Tabela 4.3: Comparação de tempos de simulação para a excitação em Rampa

Processador	Serial	Paralelo
Intel Core i5-4440S Quad-Core 2,80GHz	0,0403s	168,0015s
Intel Core i3-5005S Dual-Core 2,00GHz	0,0718s	195,6086s

Semelhante a Tabela 4.2, a Tabela 4.3 demonstra que programação serial foi mais eficiente em tempo do que a paralela. Afim de se observar o *trade-off* espaço-tempo computacional, na sequência deste trabalho, serão propostos outras dinâmicas de Filtros e uma nova comparação será produzida, pois o uso do paralelismo poderá compensar a complexidade da simulação desenvolvida que terá produtos escalares e vetoriais exigidos pela integração numérica.

4.3 Filtros Controladores Propostos

Conforme observado nas Subseções 4.2.1 e 4.2.2 onde foram realizadas as simulações piloto, a programação paralela não influenciou negativamente ou produziu algum resultado espúrio nas respostas esperadas. Desta forma, para haver um aprofundamento no estudo de redes de sincronismo, são propostos outros tipos de Filtros Controladores para a malha do PLL. Os Filtros empregados neste trabalho foram sugeridos por Bueno em [9].

Filtros de 1a ordem:

Lead-Lag Passivo (LLP):

$$F_1(s) = \frac{sT_2 + 1}{s(T_2 + T_1) + 1} \quad (4.1)$$

Lead-Lag Ativo (LLA):

$$F_2(s) = \frac{sT_2 + 1}{sT_1 + 1} \quad (4.2)$$

Proporcional-Integral Ativo (PI Ativo):

$$F_3(s) = \frac{sT_2 + 1}{sT_1} \quad (4.3)$$

para $T_1 = 1000(2\pi)$ e $T_2 = 10(2\pi)$ em segundos.

Filtros de 2a ordem:

Tipo 0:

$$F_4(s) = \frac{10^{-4}(s + 0,3393)}{s^2 + 0,2262s + 0,01579} \quad (4.4)$$

Tipo 1:

$$F_5(s) = \frac{79,58s + 1}{s(7,958s + 1)} \quad (4.5)$$

A aplicação dos Filtros será para a mesma rede OWMS da Simulação Piloto (1 Mestre com 3 Escravos) e também será para a rede Estrela TWMS, também com 1 Mestre que recebe a influência de 3 Escravos. Para expor as simulações serão observados e discutidos de forma objetiva, os gráficos de resposta e o sinal de controle $vc(t)$ dos PLLs, o erro de fase e o Lissajous. Não necessariamente serão apresentados todos os gráficos para cada simulação realizada, e sim aqueles passíveis de análise para estabilidade e sincronização da rede conforme as **Definições de 3.11 a 3.14**.

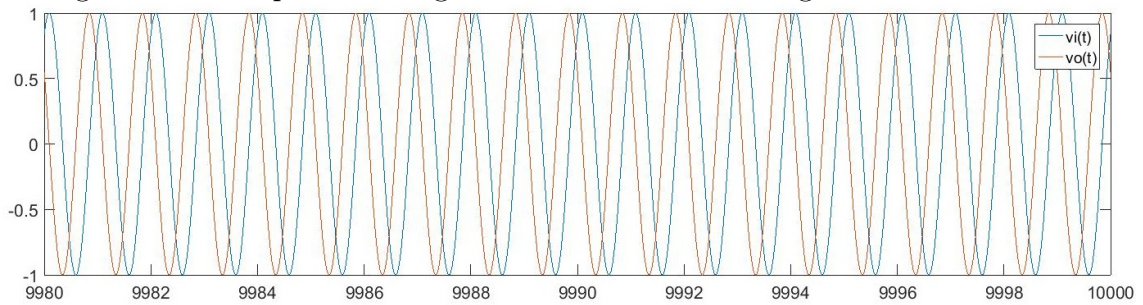
4.4 Simulação de rede OWMS

A rede simulada com os Filtros será a mesma da Simulação Piloto da subseção 4.1 descrita na Figura 4.1.

4.4.1 Simulação de rede OWMS com Filtro LLP:

Filtro F_1 com Excitação em Degrau de Fase:

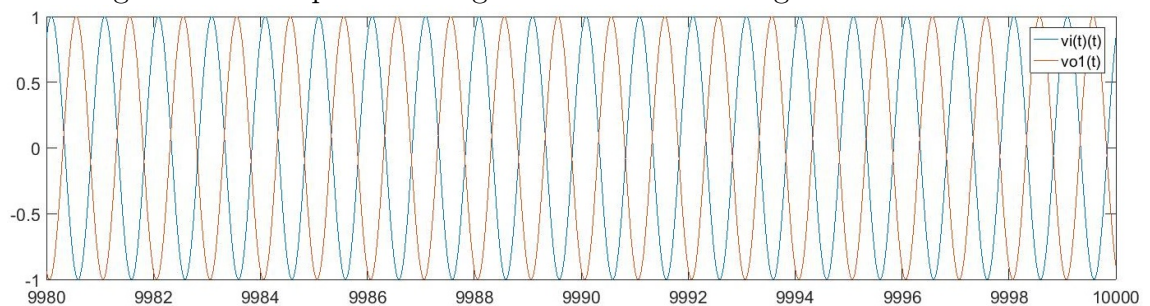
Figura 4.14: Resposta em regime do PLL Mestre ao degraú com Filtro LLP.



Fonte: autoria própria.

É possível observar na Figura 4.14 a saída do 1º PLL (Mestre) está com o sinal defasado em $\frac{\pi}{2}$ rad em relação ao sinal de entrada produzido pelo próprio VCO.

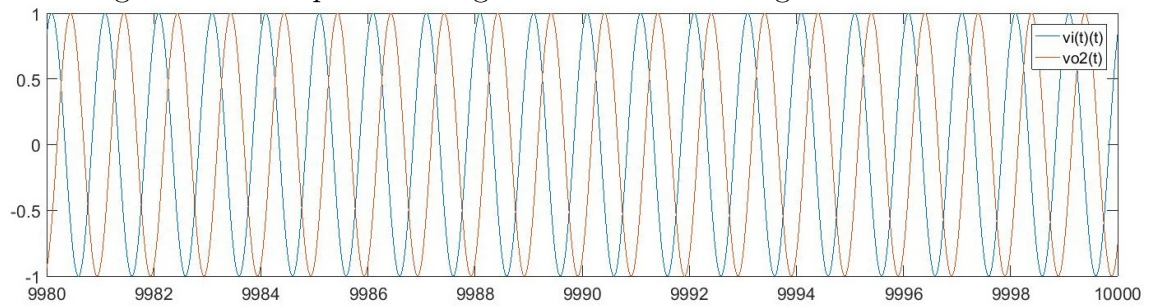
Figura 4.15: Resposta em regime do 2º PLL ao degraú com Filtro LLP.



Fonte: autoria própria.

Na Figura 4.15 a saída do 2º PLL está com o sinal defasado em π rad (180°) em relação ao sinal de entrada, ou seja, subtraiu-se $\frac{\pi}{2}$ rad, ocorrendo a inversão de sinal em relação ao sinal gerado pelo Mestre.

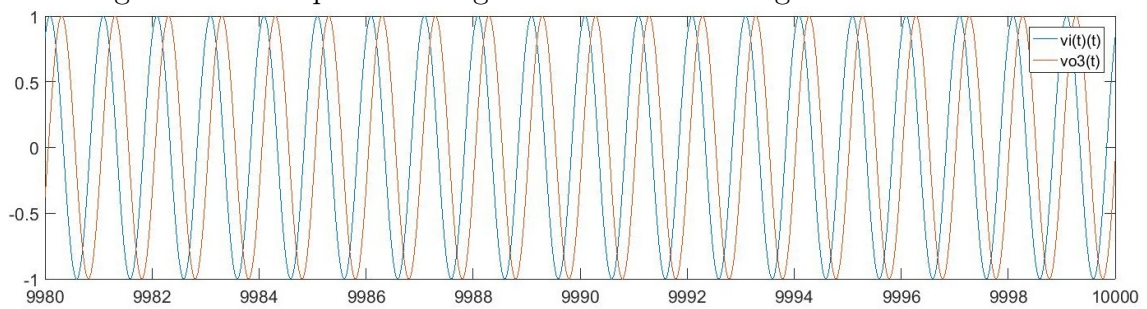
Figura 4.16: Resposta em regime do 3º PLL ao degrau com Filtro LLP.



Fonte: autoria própria.

Na Figura 4.16 a saída do 3º PLL não está exatamente com o sinal defasado em $\frac{3\pi}{2}$ rad (270°) em relação ao sinal de entrada, demonstrando a influência do Filtro LLP.

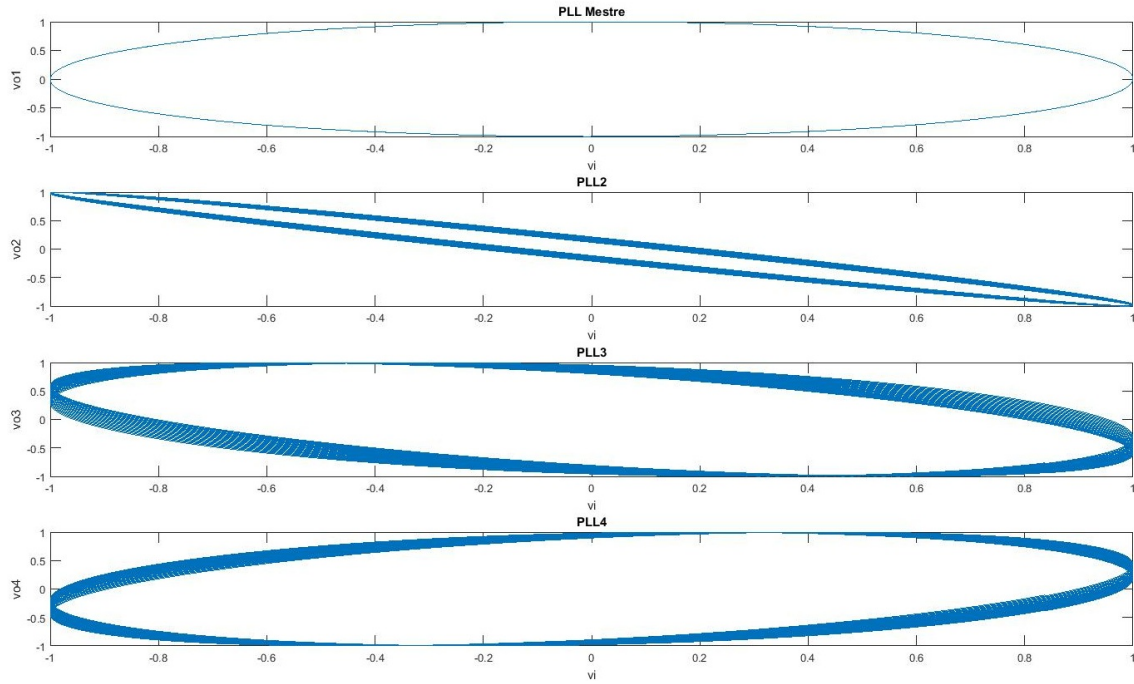
Figura 4.17: Resposta em regime do 4º PLL ao degrau com Filtro LLP.



Fonte: autoria própria.

Na Figura 4.17 a saída do 4º PLL não está com 1 ciclo de defasagem, dessa forma será preciso o gráfico de Lissajous para determinação do estado síncrono.

Figura 4.18: Lissajous da Rede OWMS com Filtro LLP.

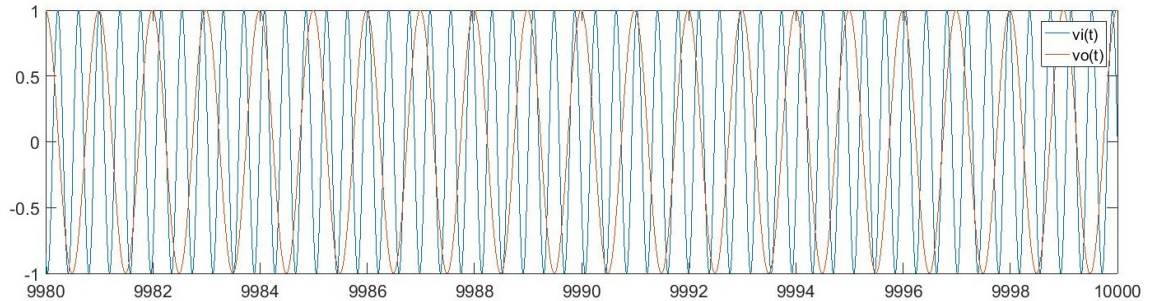


Fonte: autoria própria.

A Figura 4.18 demonstra o Lissajous de Fase dos nós da rede. Essa curva será fechada se o movimento repete-se em intervalos regulares de tempo, com a sincronia dos osciladores. o que comprova a estabilidade e a sincronização da Rede com filtro F_1 proposto e a entrada de fase em degrau unitário.

Filtro F_1 com Excitação em Rampa de Fase:

Figura 4.19: Resposta em regime dos PLLs a rampa com Filtro LLP.

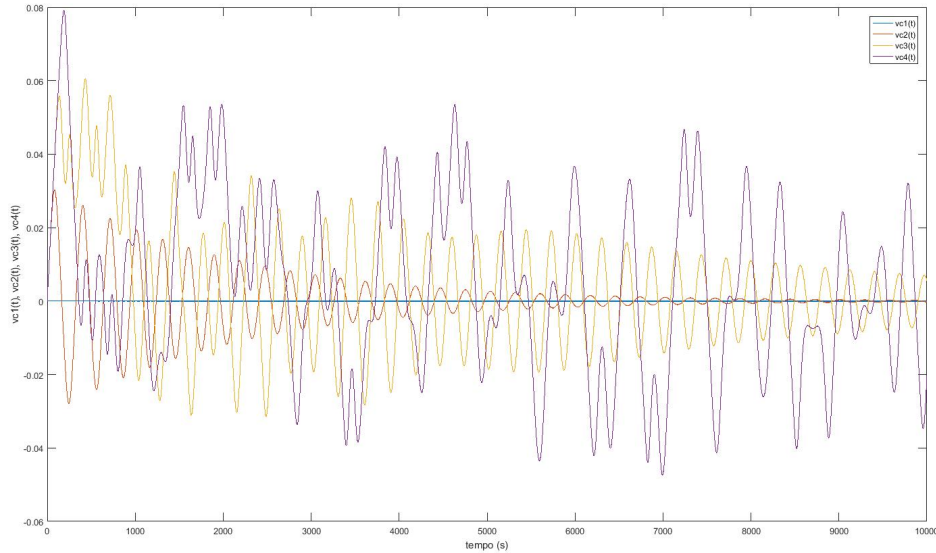


Fonte: autoria própria.

Todos os PLLs apresentaram o mesmo sinal de saída e a Figura 4.19 faz a representação. É possível observar que o sinal de saída tenta acompanhar a entrada,

desta forma para se observar o estado síncrono, será analisado o gráfico dos sinais de controle.

Figura 4.20: Sinais de controle dos PLLs.



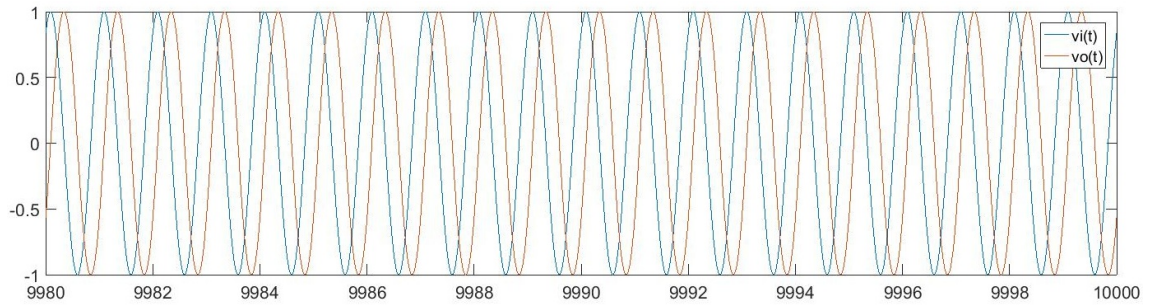
Fonte: autoria própria.

Os sinais de controle dos 4 PLLs são apresentados na Figura 4.20 e verifica-se que todos tem média zero e a diminuição das amplitudes de oscilação ao longo do tempo, indicando que existe sincronia. É importante observar que na Simulação Piloto, a excitação em rampa não atingiu o estado síncrono e a alteração da dinâmica do Filtro para 1^a ordem com Lead-Lag Passivo levou a um estado síncrono.

4.4.2 Simulação de rede OWMS com Filtro LLA:

Filtro F_2 com Excitação em Degrau de Fase:

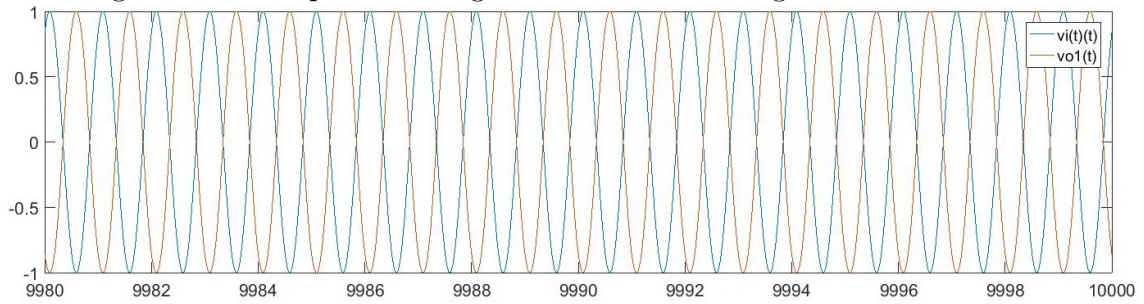
Figura 4.21: Resposta em regime do PLL Mestre ao degrau com Filtro LLA.



Fonte: autoria própria.

É possível observar na Figura 4.21 a saída do 1º PLL está com o sinal defasado em $\frac{\pi}{2}$ rad em relação ao sinal de entrada produzido pelo próprio VCO.

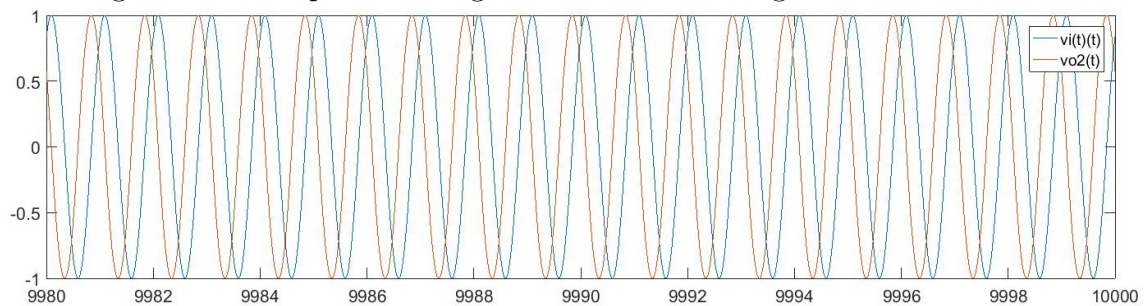
Figura 4.22: Resposta em regime do 2º PLL ao degrau com Filtro LLA.



Fonte: autoria própria.

Na Figura 4.22 a saída do 2º PLL está com o sinal defasado em π rad (180°) em relação ao sinal de entrada, ou seja, subtraiu-se $\frac{\pi}{2}$ rad, ocorrendo a inversão de sinal em relação ao sinal gerado pelo Mestre.

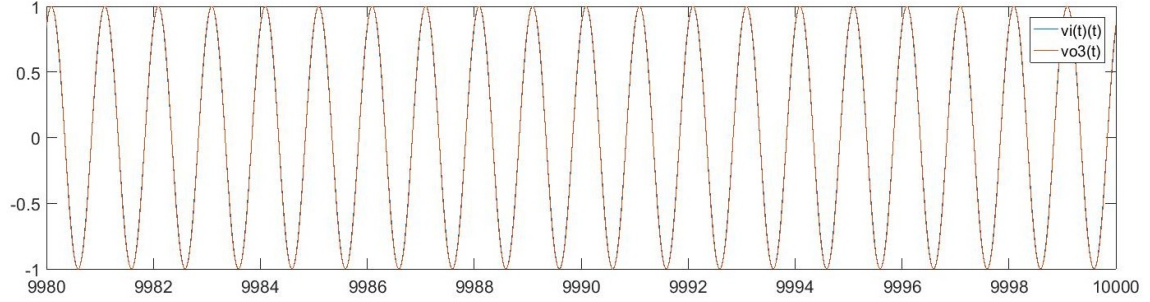
Figura 4.23: Resposta em regime do 3º PLL ao degrau com Filtro LLA.



Fonte: autoria própria.

Na Figura 4.23 a saída do 3º PLL está com o sinal defasado em $\frac{3\pi}{2}$ rad (270°) em relação ao sinal de entrada.

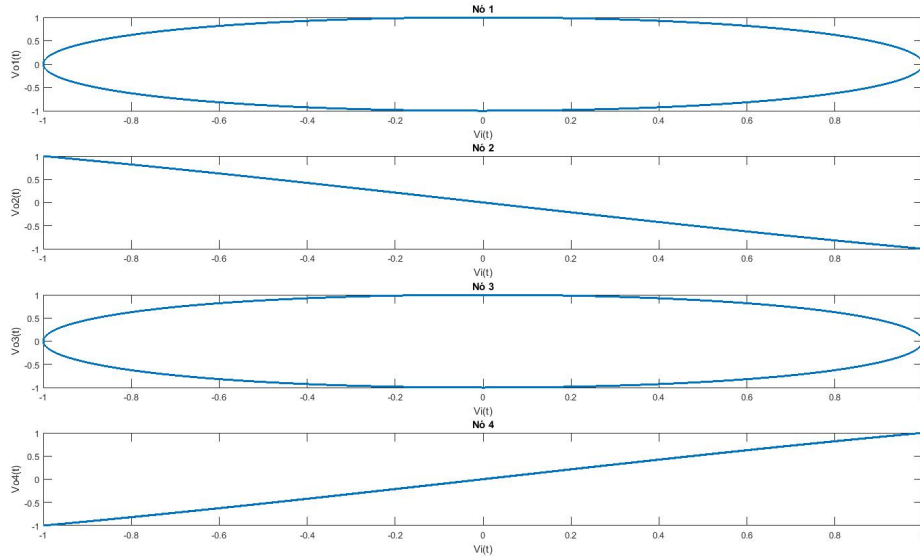
Figura 4.24: Resposta em regime do 4º PLL ao degrau com Filtro LLA.



Fonte: autoria própria.

Na Figura 4.24 a saída do 4º PLL está com 1 ciclo de defasagem, confirmando a teoria apresentada no Capítulo 3 dentro da **Definição 3.2**.

Figura 4.25: Lissajous da Rede OWMS com Filtro LLA.

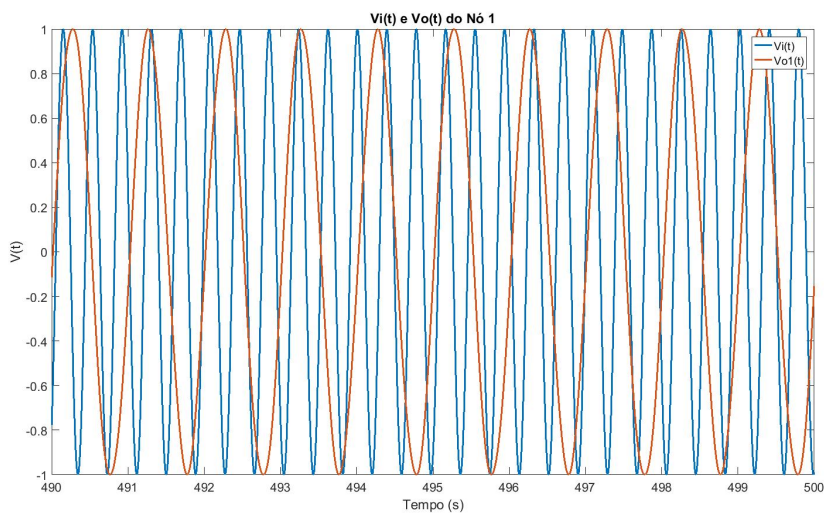


Fonte: autoria própria.

A Figura 4.25 demonstra o Lissajous de Fase dos nós da rede. Essa curva será fechada se o movimento repete-se em intervalos regulares de tempo, com a sincronia dos osciladores. o que comprova a estabilidade e a sincronização da Rede com filtro F_2 proposto e a entrada de fase em degrau unitário.

Filtro F_2 com Excitação em Rampa de Fase:

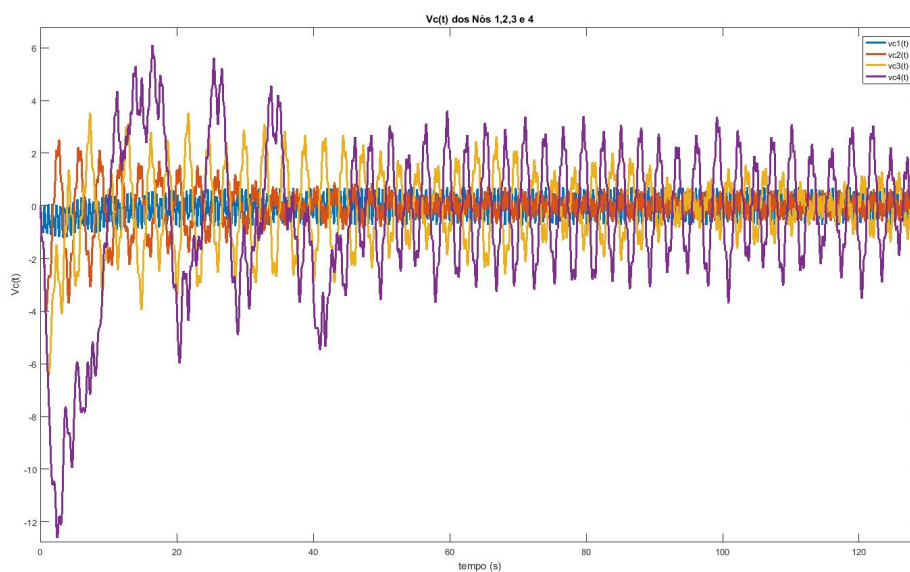
Figura 4.26: Resposta em regime dos PLLs a rampa com Filtro LLA.



Fonte: autoria própria.

Todos os PLLs apresentaram o mesmo sinal de saída conforme a Figura 4.26. É possível observar que o sinal de saída tenta acompanhar a entrada, desta forma para se observar o estado síncrono, será analisado o gráfico dos sinais de controle.

Figura 4.27: Sinais de controle dos PLLs $vc(t)$.



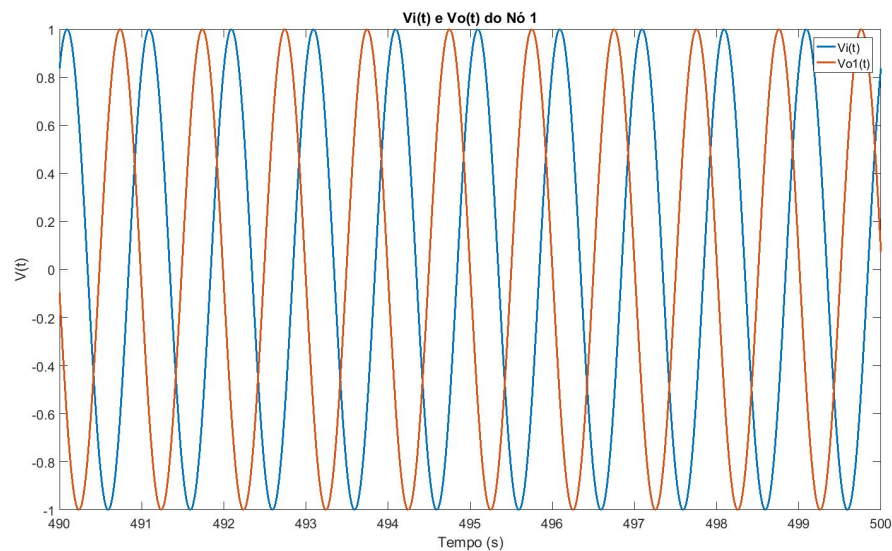
Fonte: autoria própria.

Os sinais de controle dos 4 PLLs são apresentados na Figura 4.20 e verifica-se que todos tem média zero e a diminuição das amplitudes de oscilação ao longo do tempo, indicando que existe sincronia.

4.4.3 Simulação de rede OWMS com Filtro PI Ativo:

Filtro F_3 com Excitação em Degrau de Fase:

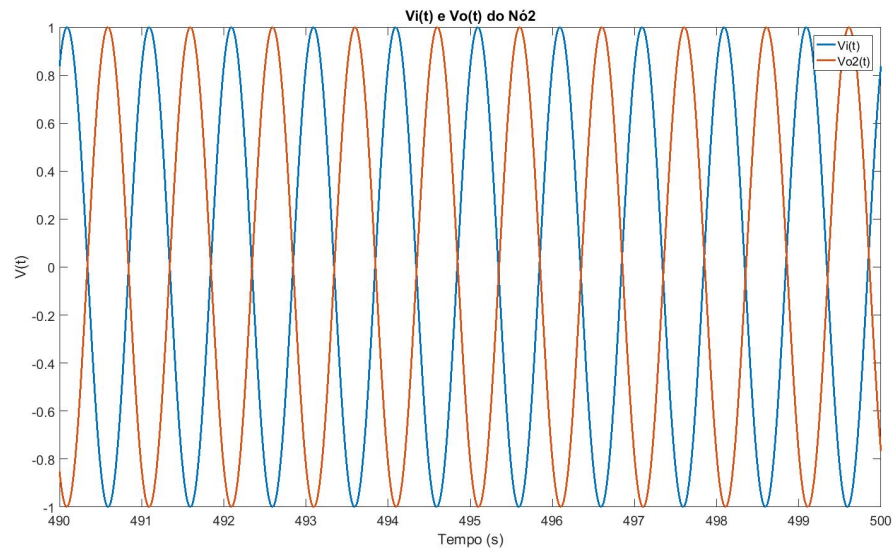
Figura 4.28: Resposta em regime do PLL Mestre ao degrau com Filtro PI Ativo.



Fonte: autoria própria.

É possível observar na Figura 4.28 a saída do 1º PLL está com o sinal defasado em relação ao sinal de entrada produzido pelo próprio VCO, demonstrando o uso do Filtro PI Ativo.

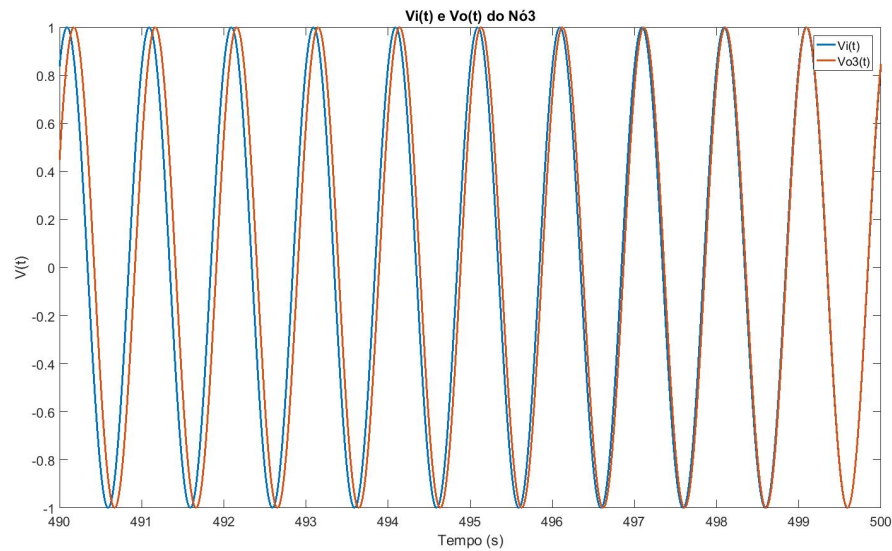
Figura 4.29: Resposta em regime do 2º PLL ao degrau com Filtro PI Ativo.



Fonte: autoria própria.

Na Figura 4.29 a saída do 2º PLL também apresenta defasagem próxima a 180º.

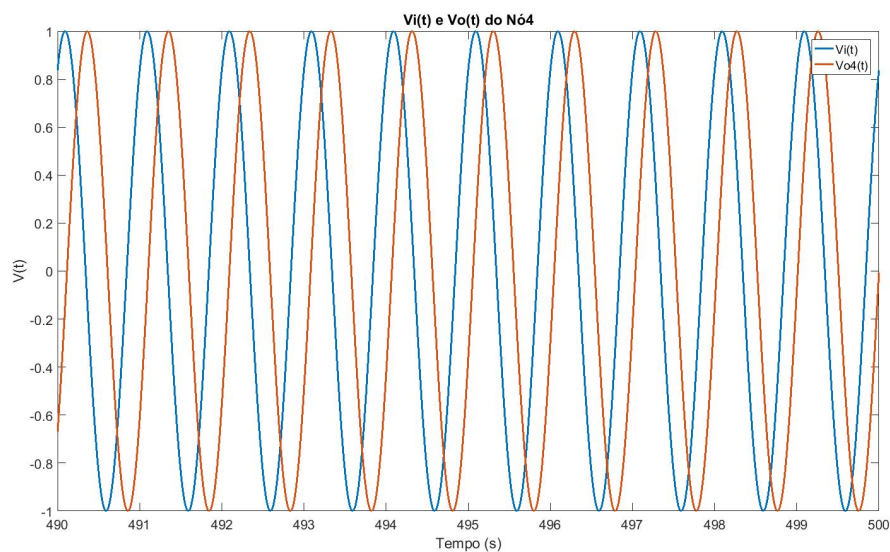
Figura 4.30: Resposta em regime do 3º PLL ao degrau com Filtro PI Ativo.



Fonte: autoria própria.

Na Figura 4.30 a saída do 3º PLL tende a reduzir a defasagem até entrar em fase novamente com o sinal de entrada.

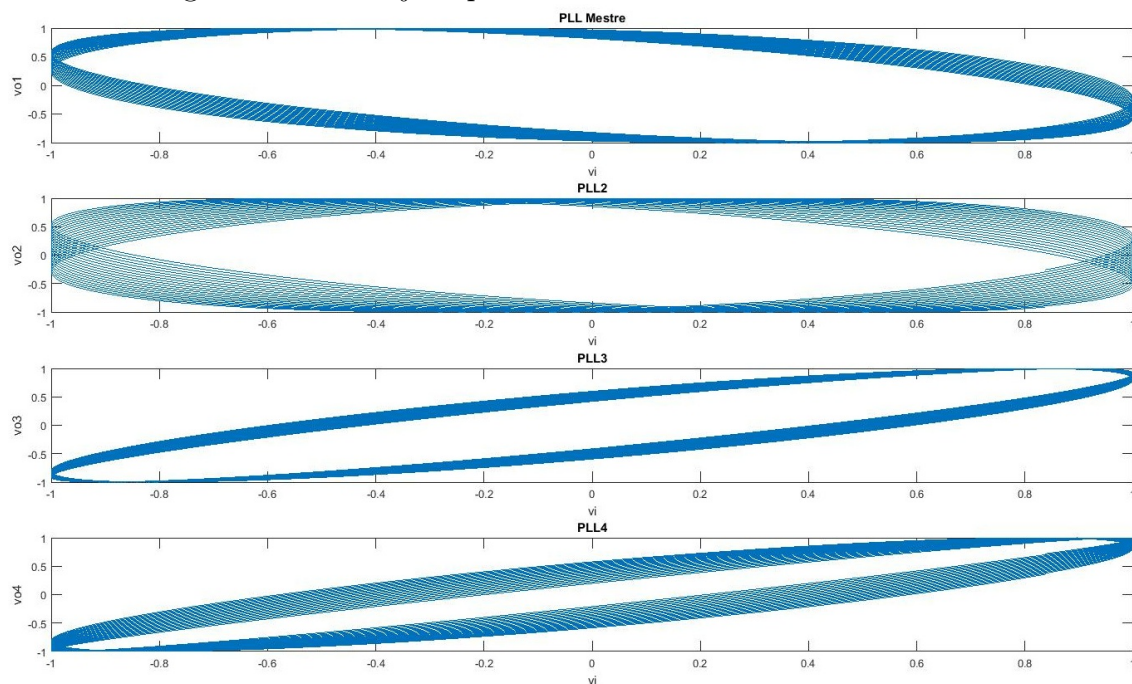
Figura 4.31: Resposta em regime do 4º PLL ao degrau com Filtro PI Ativo.



Fonte: autoria própria.

Na Figura 4.31 a saída do 4º PLL apresenta uma saída semelhante ao 1º PLL. Como a cadeia não teve um comportamento esperado, é necessário analisar o gráfico de Lissajous para indicar se houve a busca por sincronismo

Figura 4.32: Lissajous para rede OWMS com Filtro PI Ativo.

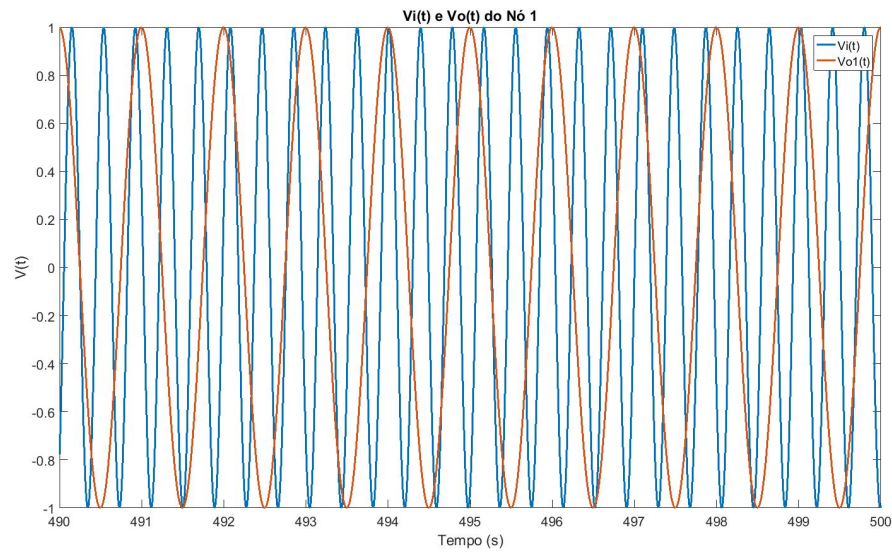


Fonte: autoria própria.

A Figura 4.32 demonstra o Lissajous de Fase de cada PLL da rede. Essa curva será fechada se o movimento repete-se em intervalos regulares de tempo, com a sincronia dos osciladores. É possível observar em cada Lissajous que existem várias oscilações mas todas em intervalos fechados e dentro de uma região limitada. Pode-se afirmar que esta rede entrou em sincronia.

Filtro F_3 com Excitação em Rampa de Fase:

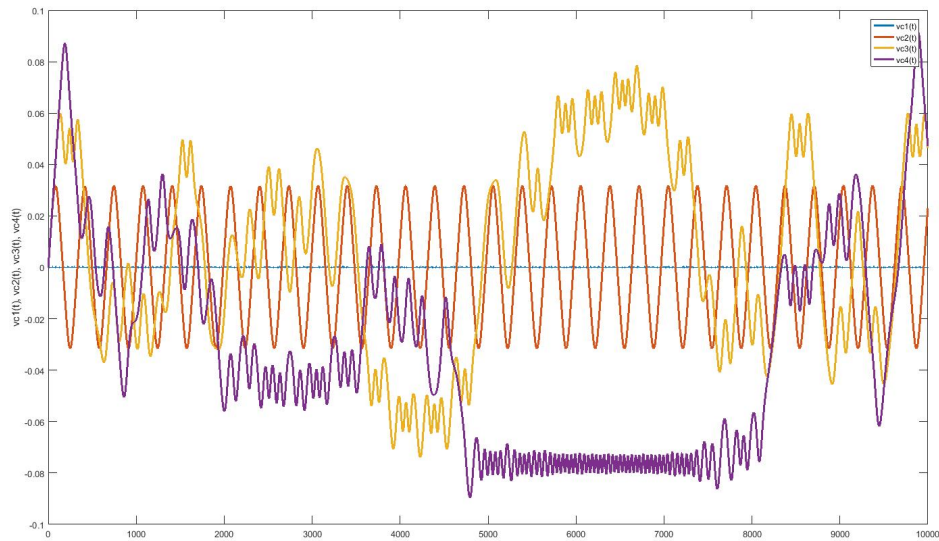
Figura 4.33: Resposta em regime dos PLLs a rampa com Filtro PI Ativo.



Fonte: autoria própria.

Todos os PLLs apresentaram o mesmo sinal de saída conforme a Figura 4.33. É possível observar que o sinal de saída tenta acompanhar a entrada. Em seguida será analisado o gráfico dos sinais de controle.

Figura 4.34: Lissajous para rede OWMS com Filtro PI Ativo e entrada em rampa.

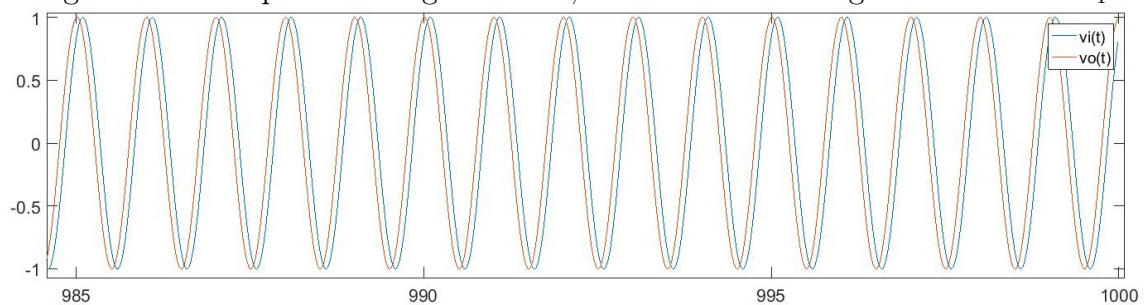


Fonte: autoria própria.

Os sinais de controle dos 4 PLLs são apresentados na Figura 4.34 e verifica-se que eles não tem média zero e não ocorre diminuição das amplitudes de oscilação ao longo do tempo, indicando que não existe sincronia e/ou um estado síncrono para o Filtro PI Ativo com excitação em rampa unitária de fase.

4.4.4 Simulação de rede OWMS com Filtro de 2^a Ordem Tipo 0

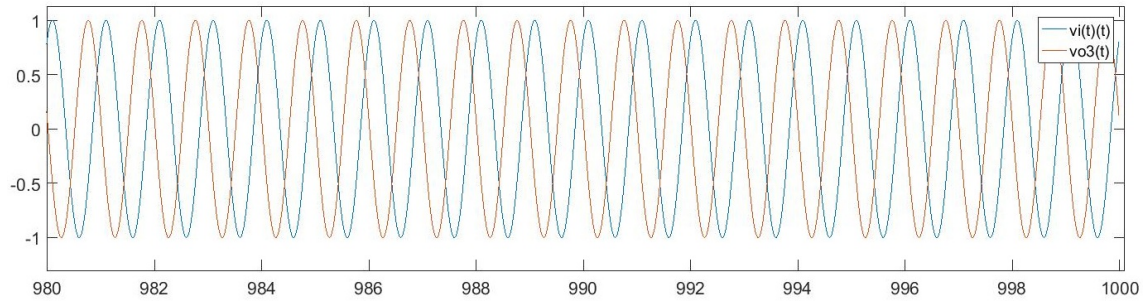
Filtro F_4 com Excitação em Degrau de Fase:

Figura 4.35: Resposta em regime do 1^o, 2^o e 3^o PLLs ao degra com Filtro F_4 .

Fonte: autoria própria.

Para esta simulação os PLLs 1,2 e 3 forneceram o mesmo sinal de saída, conforme a Figura 4.35. Somente o 4º PLL se diferenciou.

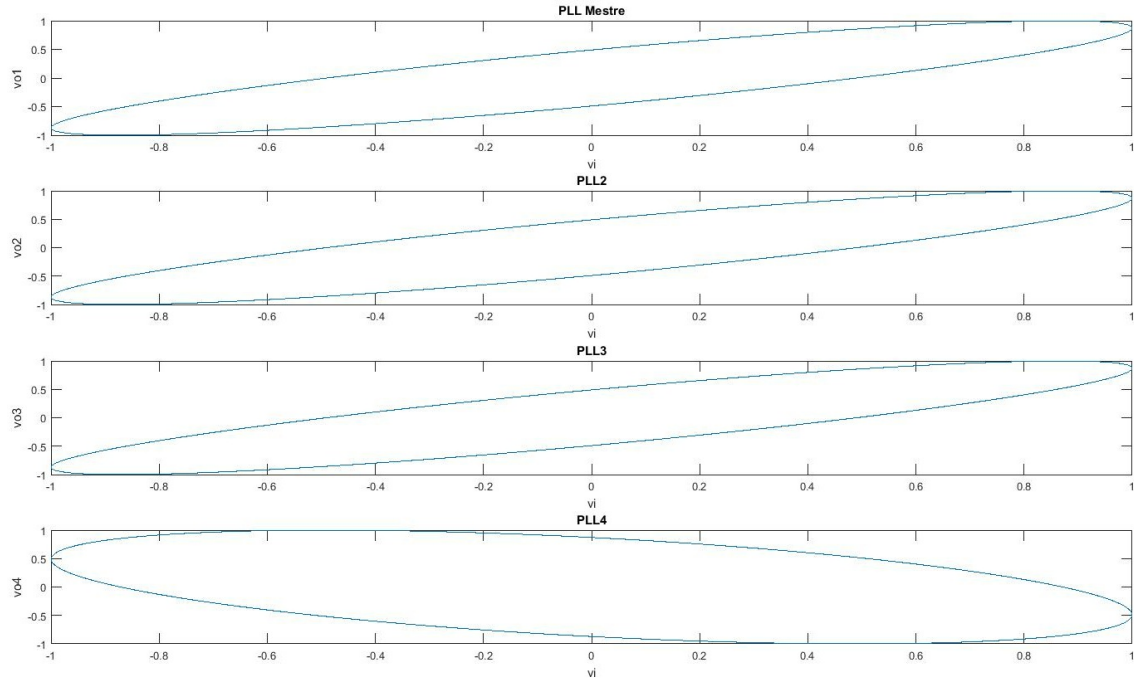
Figura 4.36: Resposta em regime do 4º PLL ao degrau com Filtro F_4 .



Fonte: autoria própria.

Conforme a Figura 4.35, o PLL4 forneceu um sinal de saída com defasagem diferente apresentada pelos anteriores. Será necessário a avaliação do gráfico de Lissajous de Fase.

Figura 4.37: Lissajous da rede OWMS com Filtro F_4 e entrada em degrau.



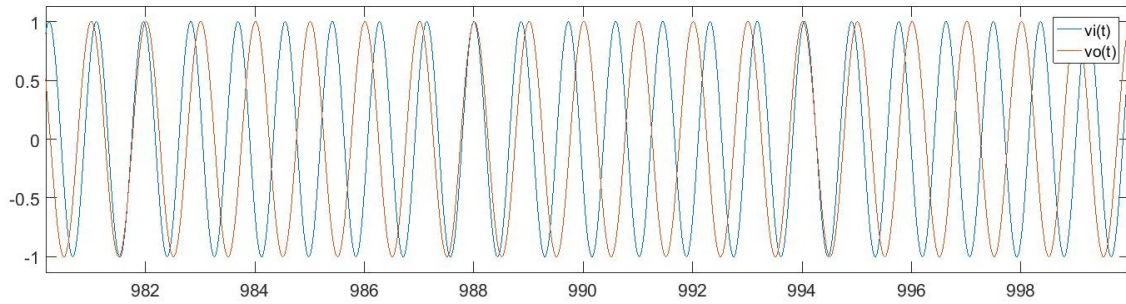
Fonte: autoria própria.

A Figura 4.37 demonstra o Lissajous da rede. Essa curva será fechada se o movimento repete-se em intervalos regulares de tempo, com a sincronia dos osciladores. É possível observar que o Lissajous do 1º, 2º e 3º PLLs são iguais e somente o 4º PLL é

diferente, mas as todas as oscilações são em intervalos fechados e dentro de uma região limitada. Pode-se afirmar que esta rede entrou em sincronia.

Filtro F_4 com Excitação em Rampa de Fase:

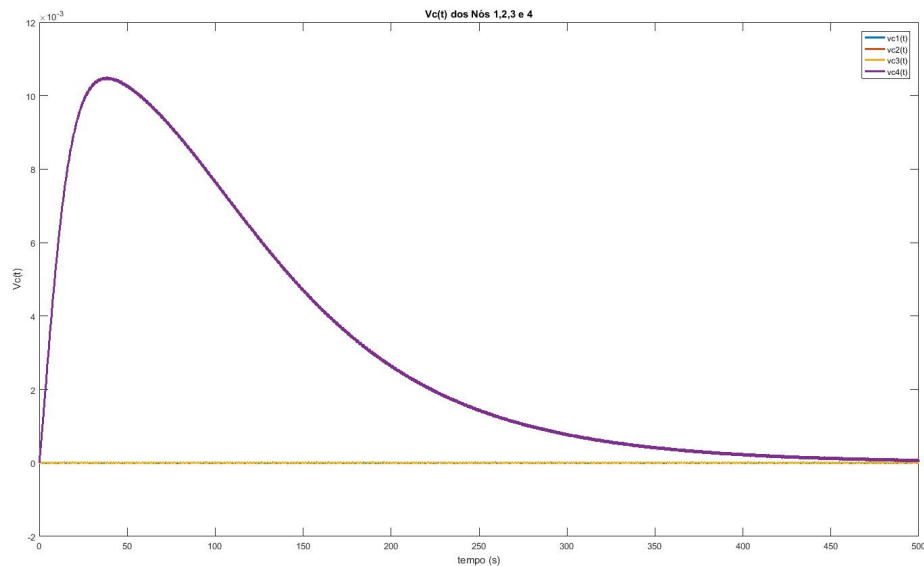
Figura 4.38: Resposta em regime dos PLLs a rampa com Filtro F_4 .



Fonte: autoria própria.

Para esta simulação todos PLLs da cadeia forneceram o mesmo sinal de saída, conforme a Figura 4.38. Será necessário analisar o gráfico do sinal de controle $vc(t)$.

Figura 4.39: Sinais de Controle dos 4 PLLs da rede a entrada em rampa.



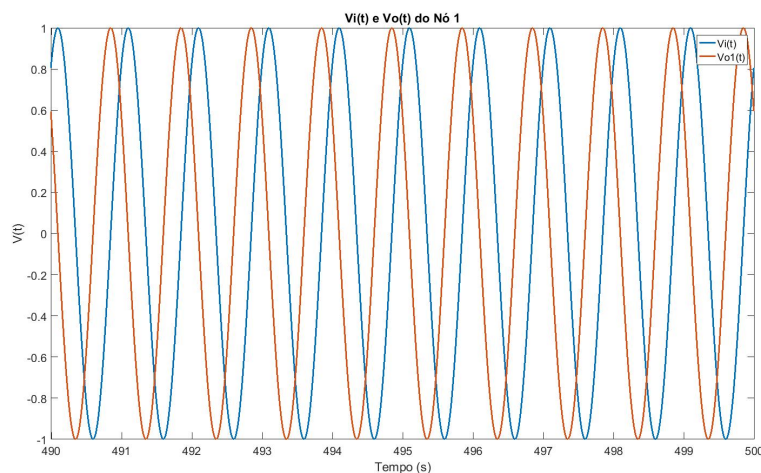
Fonte: autoria própria.

A Figura 4.39, demonstra a sinal de controle de cada PLL da rede. Observa-se que as amplitudes de oscilação diminuem ao longo do tempo e que o valor médio de cada sinal em regime permanente é zero, indicando sincronismo na rede.

4.4.5 Simulação de rede OWMS com Filtro de 2^a Ordem Tipo 1

Filtro F_5 com Excitação em Degrau de Fase:

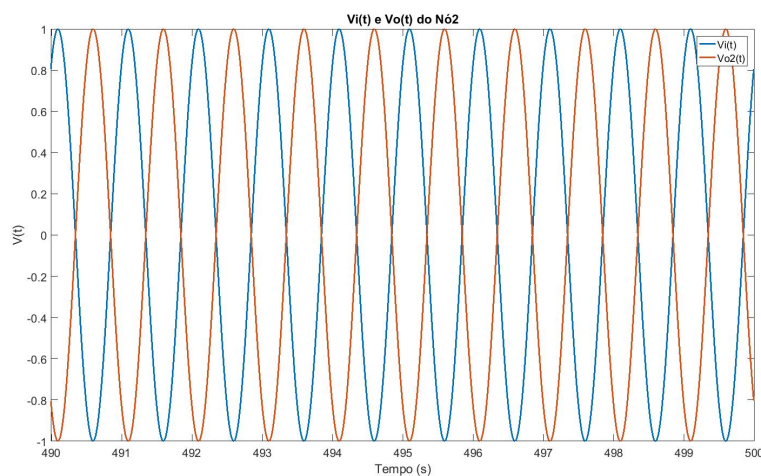
Figura 4.40: Resposta em regime do PLL Mestre ao degraú com Filtro F_5 .



Fonte: autoria própria.

É possível observar na Figura 4.40 que a saída do 1^o PLL está com o sinal defasado em $\frac{\pi}{2}$ rad em relação ao sinal de entrada produzido pelo próprio VCO, de forma semelhante a Simulação Piloto.

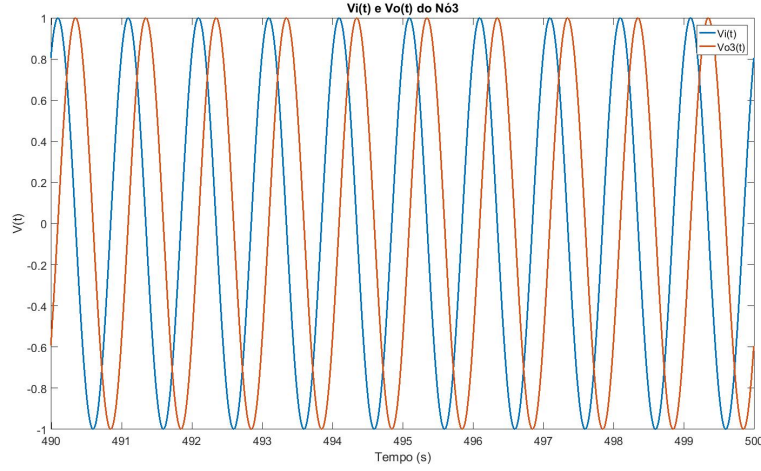
Figura 4.41: Resposta em regime do 2^o PLL ao degraú com Filtro F_5 .



Fonte: autoria própria.

Na Figura 4.41 a saída do 2º PLL está com o sinal defasado em π rad (180°), ocorrendo a inversão de sinal em relação ao sinal de entrada gerado no Mestre.

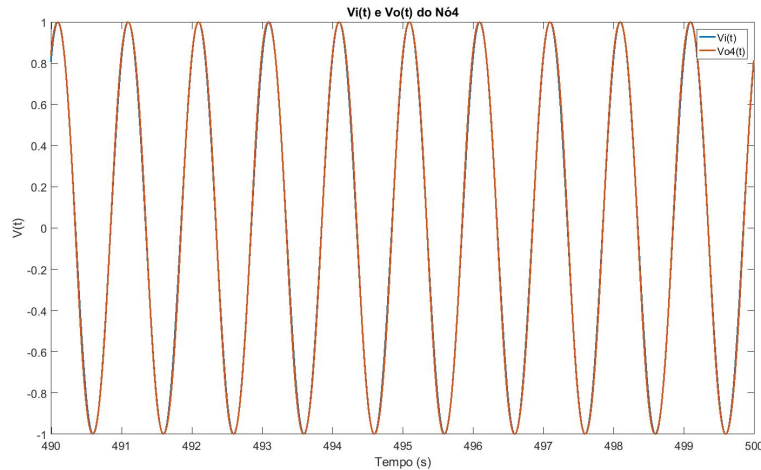
Figura 4.42: Resposta em regime do 3º PLL ao degrau com Filtro F_5 .



Fonte: autoria própria.

Na Figura 4.42 a saída do 3º PLL está com o sinal defasado em $\frac{3\pi}{2}$ rad (270°) em relação ao sinal de entrada, subtraindo-se da saída do anterior $\frac{\pi}{2}$ rad.

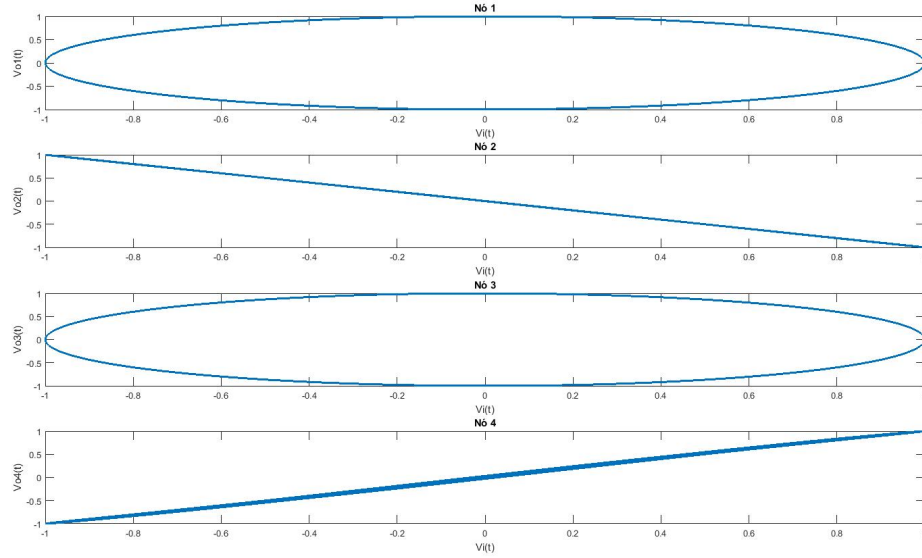
Figura 4.43: Resposta em regime do 4º PLL ao degrau com Filtro F_5 .



Fonte: autoria própria.

Na Figura 4.43 a saída do 4º PLL está com o sinal defasado em 2π rad (360°), 1 ciclo completo, em relação ao sinal de entrada, conforme a **Definição 3.2**.

Figura 4.44: Lissajous da rede OWMS com filtro F_5 com excitação em degrau.

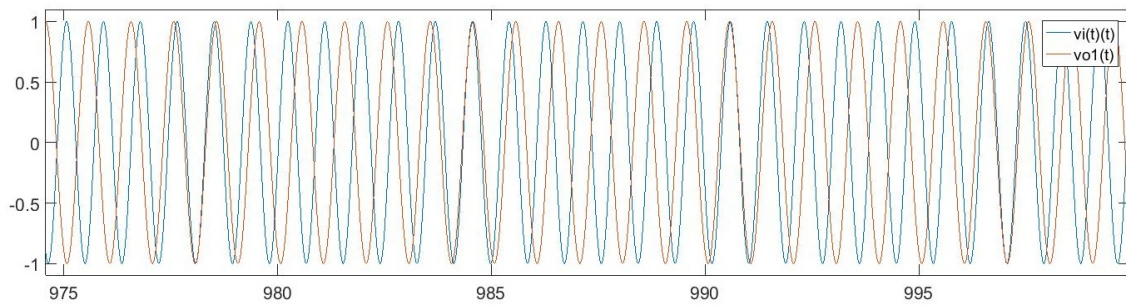


Fonte: autoria própria.

O Lissajous apresentado na Figura 4.44 é semelhante àquele da Simulação Piloto, indicando sincronismo em fase (1º e 3º PLLs) e em quadratura (2º e 4º PLLs).

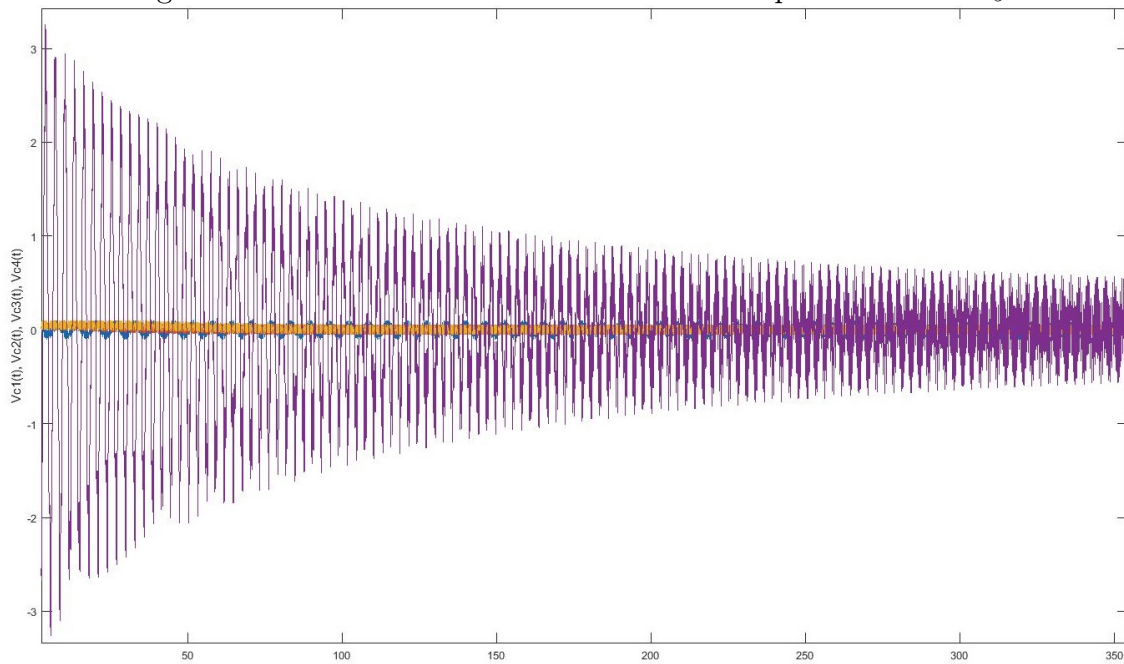
Filtro F_5 com Excitação em Rampa de Fase:

Figura 4.45: Resposta em regime dos PLLs a rampa com Filtro F_5 .



Fonte: autoria própria.

A Figura 4.45 demonstra que o sinal de saída de todos os PLLs são mesmo para a excitação em rampa. Desta forma, para se verificar se há sincronismo, é necessário analisar os sinais de controle de cada nó.

Figura 4.46: Sinais de controle dos PLLs a rampa com Filtro F_5 .

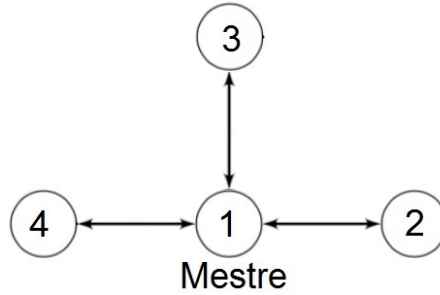
Fonte: autoria própria.

A Figura 4.46 representa o sinal de controle de cada PLL da rede. Observa-se que as amplitudes de oscilação diminuem ao longo do tempo e que o valor médio de cada sinal é zero, indicando sincronismo na rede.

4.5 Simulação de rede Estrela TWMS

A topologia de rede empregada para esta simulação é ilustrada pela Figura 4.47. O nó Mestre fornece e recebe sinal de 3 nós Escravos.

Figura 4.47: Arquitetura Estrela de via dupla

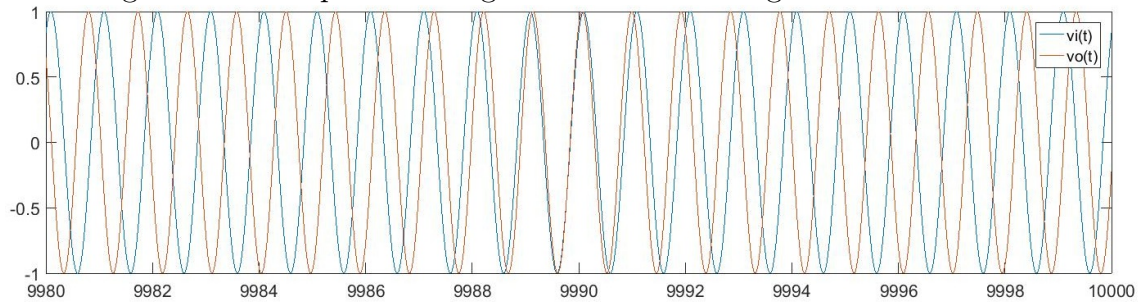


Fonte: autoria própria.

4.5.1 Simulação de rede Estrela TWMS com Filtro LLP

Filtro F_1 com Excitação em Degrau de Fase:

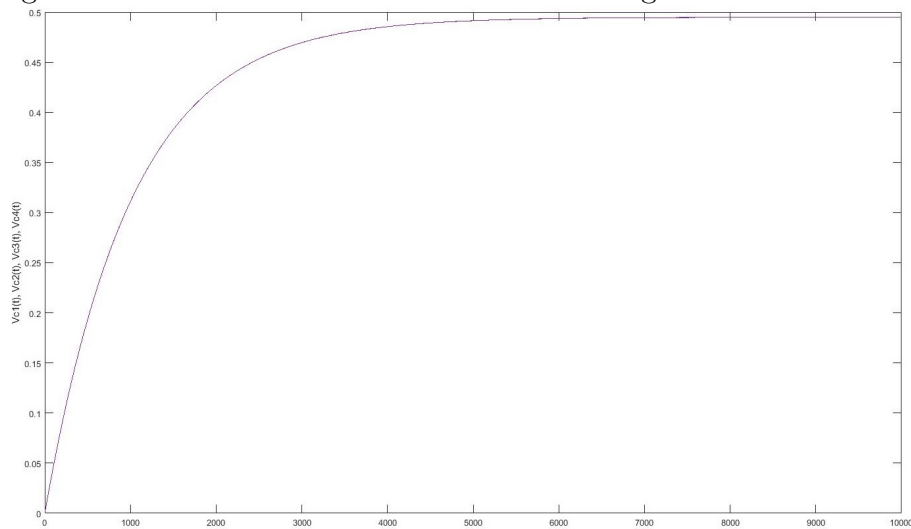
Figura 4.48: Resposta em regime dos PLLs ao degrau com Filtro LLP



Fonte: autoria própria.

A Figura 4.48 demonstra que o sinal de saída de todos os PLLs são mesmo, tanto para o Mestre quanto para os 3 escravos para a excitação em degrau. Desta forma, para se verificar se há sincronismo, é necessário analisar os sinais de controle de cada nó. Essa situação de sinais de saída iguais em regime permanente será recorrente na análise da topologia Estrela TWMS.

Figura 4.49: Sinais de Controle dos PLLs ao degrau com Filtro LLP

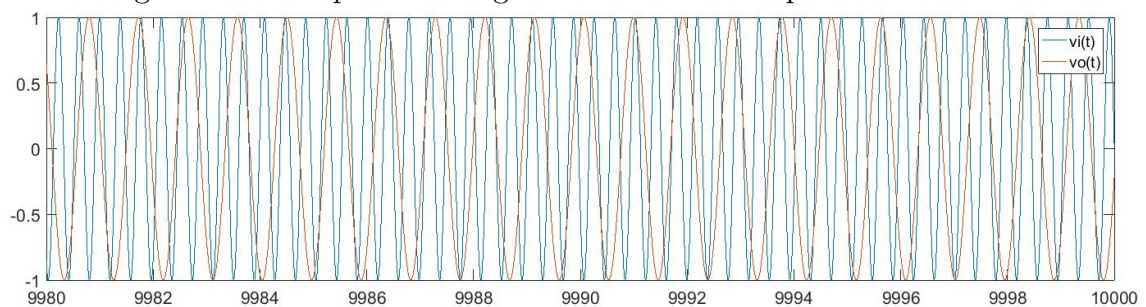


Fonte: autoria própria.

A Figura 4.49 mostra os sinais de controle da rede. É notório que sinal tenta acompanhar a excitação em degrau e se estabiliza próximo 0,5.

Filtro F_1 com Excitação em Rampa de Fase:

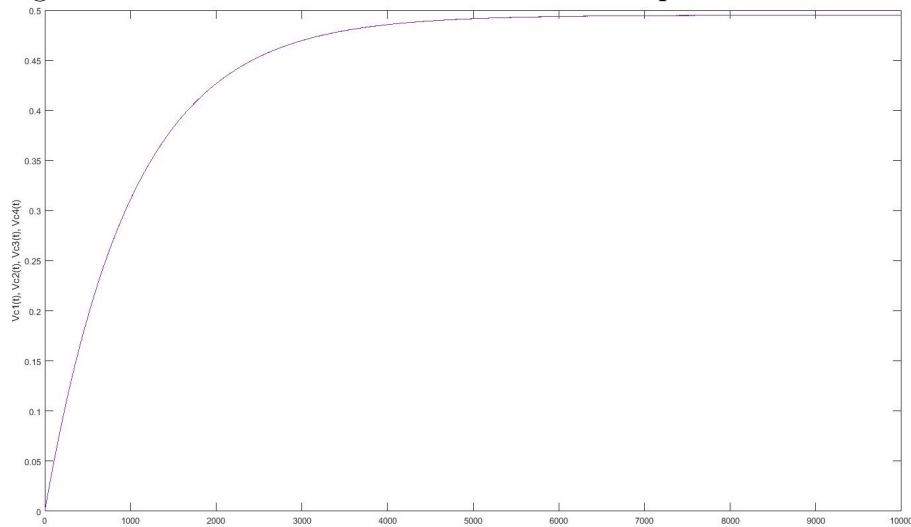
Figura 4.50: Resposta em regime dos PLLs a rampa com Filtro LLP



Fonte: autoria própria.

A Figura 4.50 demonstra que o sinal de saída de todos os PLLs são mesmo, tanto para o Mestre quanto para os 3 escravos para a excitação em rampa.

Figura 4.51: Sinais de Controle dos PLLs a rampa com Filtro LLP



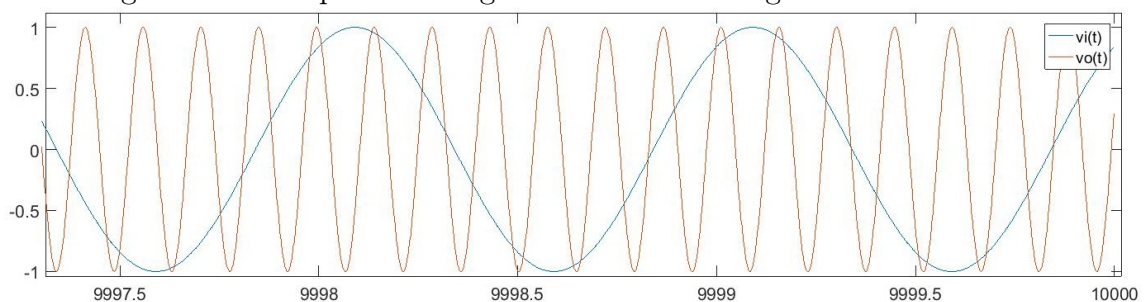
Fonte: autoria própria.

A Figura 4.51 mostra os sinais de controle da rede. Apesar das excitações diferentes, os sinais de controle do TWMS Estrela com o Filtro F_1 responderam exatamente da mesma forma.

4.5.2 Simulação de rede Estrela TWMS com Filtro LLA

Filtro F_2 com Excitação em Degrau de Fase:

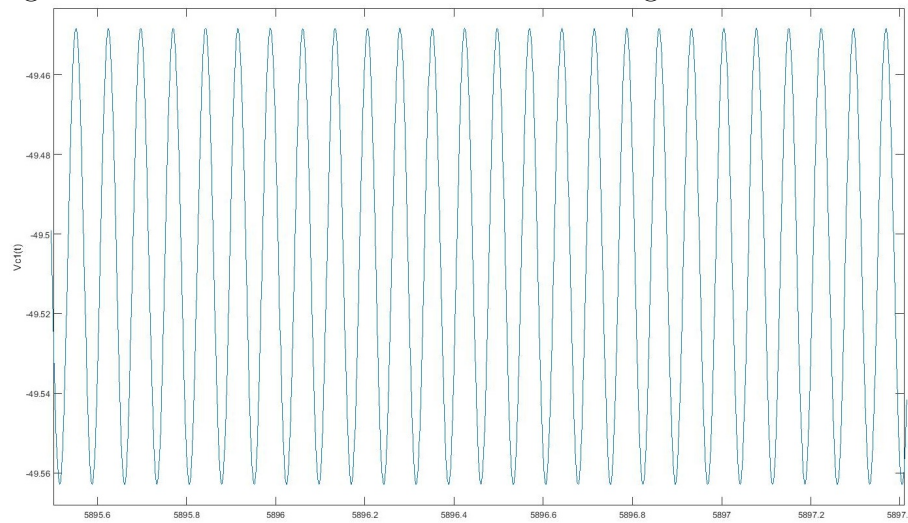
Figura 4.52: Resposta em regime dos PLLs ao degraú com Filtro LLA



Fonte: autoria própria.

A Figura 4.52 mostra os sinais de saída da rede. É possível observar que o sinais de $vo(t)$ tem uma frequência maior que a entrada $vi(t)$.

Figura 4.53: Sinais de Controle dos PLLs ao degrau com Filtro LLA

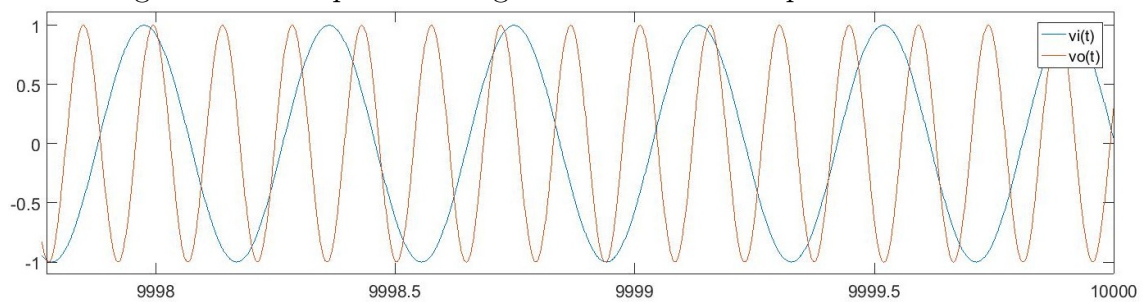


Fonte: autoria própria.

A Figura 4.53 indica que os sinais de controle dos PLLs oscilam de forma sustentada, não reduzem a amplitude e não tem média zero, indicando que não encontrou sincronismo com Filtro F_2 com excitação em degrau.

Filtro F_2 com Excitação em Rampa de Fase:

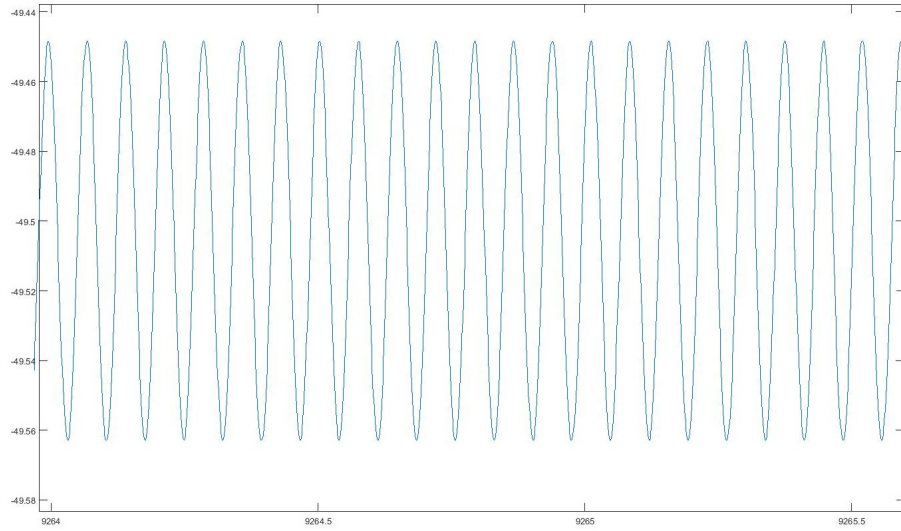
Figura 4.54: Resposta em regime dos PLLs a rampa com Filtro LLA



Fonte: autoria própria.

A Figura 4.52 mostra os sinais de saída da rede. O mesmo sinal de saída se repete para o Mestre e as pontas da Estrela TWMS.

Figura 4.55: Sinais de Controle dos PLLs a rampa com Filtro LLA



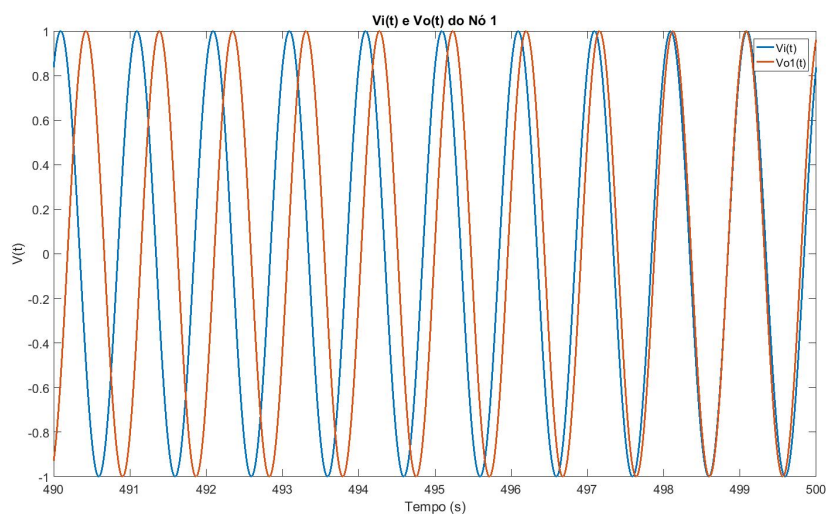
Fonte: autoria própria.

A Figura 4.55 indica que os sinais de controle dos PLLs oscilam de forma sustentada, não reduzem a amplitude e não tem média zero, indicando que não encontrou sincronismo com Filtro F_2 com excitação em rampa, situação muito semelhante ao Filtro F_1 . Respostas semelhantes mesmo com diferentes excitações são comuns na topologia Estrela TWMS, onde o Mestre recebe a influência dos 3 Escravos e estes só rebecem a influência do nó central.

4.5.3 Simulação de rede Estrela TWMS com Filtro PI Ativo

Filtro F_3 com Excitação em Degrau de Fase:

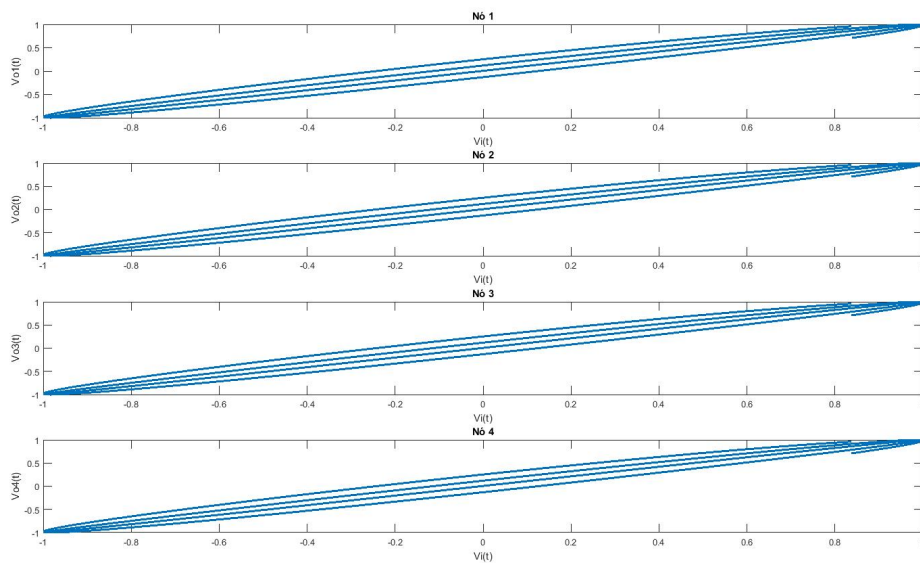
Figura 4.56: Lissajous dos PLLs ao degrau com Filtro PI Ativo



Fonte: autoria própria.

O Sinal de Saída dos PLLs estão indicados na Figura 4.56 e são semelhantes, tanto do Mestre quanto dos Escravos. É possível perceber que existe um tentativa de sincronização próximo ao final da simulação; dessa forma, é preciso analisar o gráfico de Lissajous para confirmar (ou não).

Figura 4.57: Sinal de Controle dos PLLs ao degrau com Filtro PI Ativo

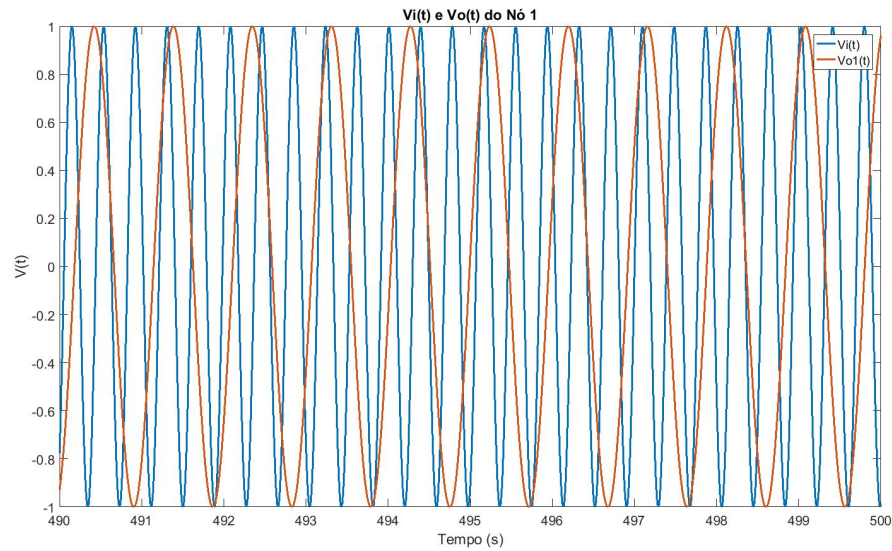


Fonte: autoria própria.

O Lissajous da rede Estrela TWMS na Figura 4.57 indica que houve uma tentativa de sincronização que acabou não ocorrendo pois não atende as **Definição 3.11 e 3.12**, visto que existe uma quebra na Figura (próximo a 0,8 nas abscissas dos gráficos), de forma simplificada, a elipse de Lissajous não fechou.

Filtro F_3 com Excitação em Rampa de Fase:

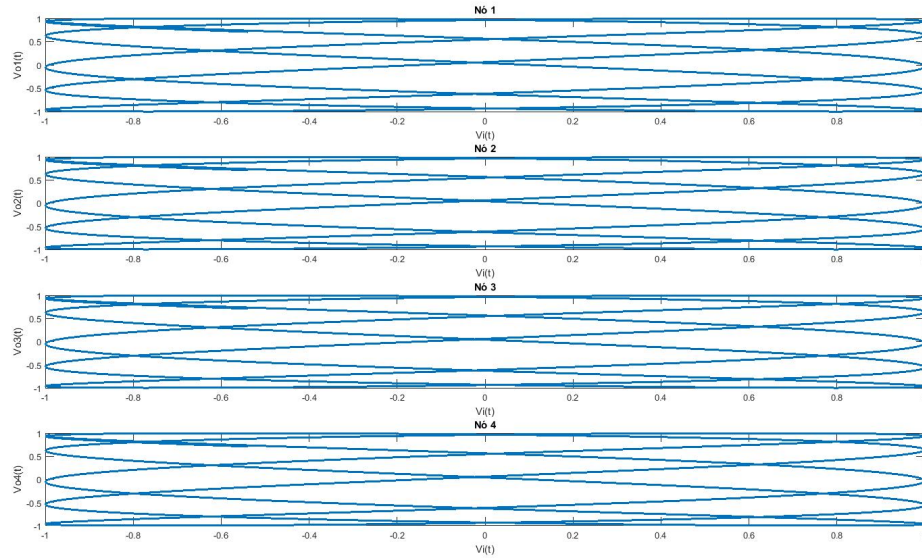
Figura 4.58: Resposta em regime dos PLLs a rampa com Filtro PI Ativo



Fonte: autoria própria.

O Sinal de Saída dos PLLs estão indicados na Figura 4.58 e são semelhantes, tanto do Mestre quanto dos Escravos e oscilam em frequências distintas, com a saída tentando buscar seguir a entrada. Os sinais de controle dos PLLs para a excitação em rampa são os mesmos da Figura 4.58. Dessa forma será exibido o gráfico de Lissajous que confirma a não sincronização da rede para esta excitação aplicada.

Figura 4.59: Lissajous para entrada em Rampa dos PLLs com Filtro PI Ativo

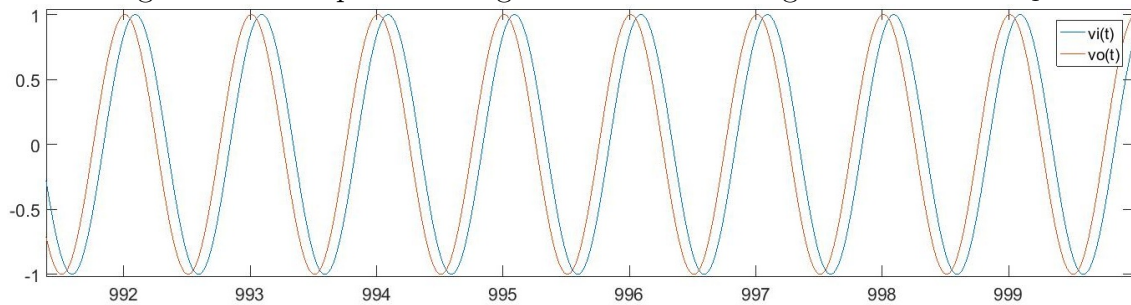


Fonte: autoria própria.

O Lissajous da Figura 4.59 indica uma oscilação ilimitada para todos os nós da rede, ou seja, não ocorre a sincronização desejada.

4.5.4 Simulação de rede TWMS com Filtro de 2^a Ordem Tipo 0

Filtro F_4 com Excitação em Degrau de Fase:

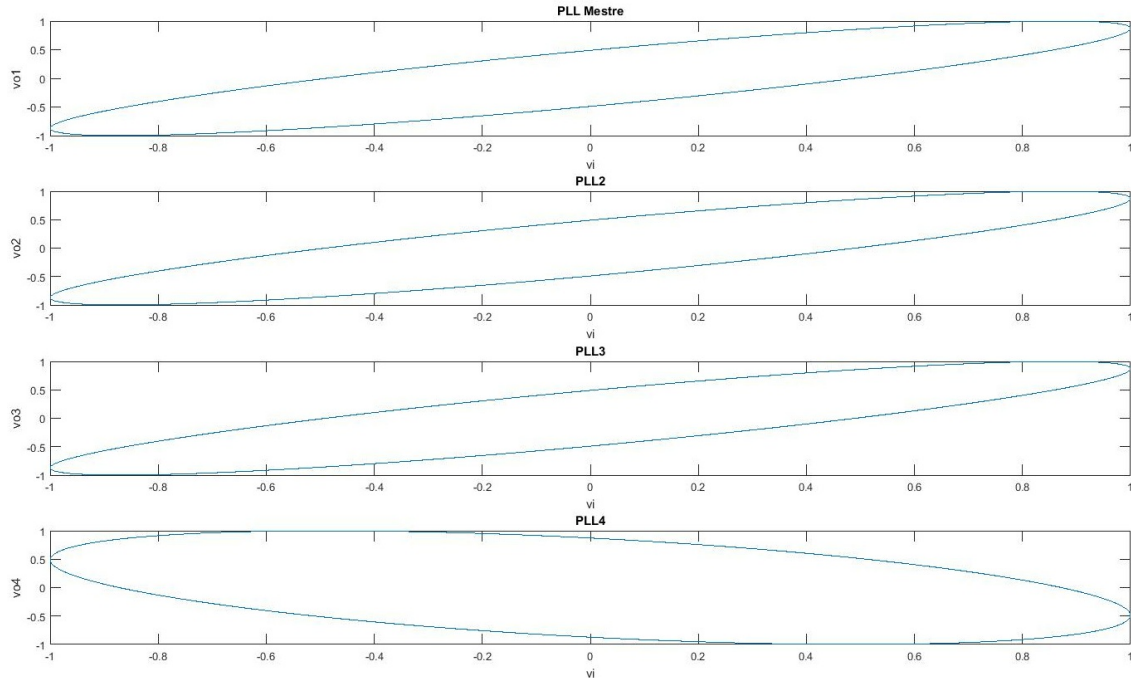
Figura 4.60: Resposta em regime dos PLLs ao degrau com Filtro F_4 

Fonte: autoria própria.

Os sinais de saída dos PLLs estão indicados na Figura 4.60 e são todos semelhantes, tanto do Mestre quanto dos Escravos, indicando um erro de fase constante em

regime permanente.

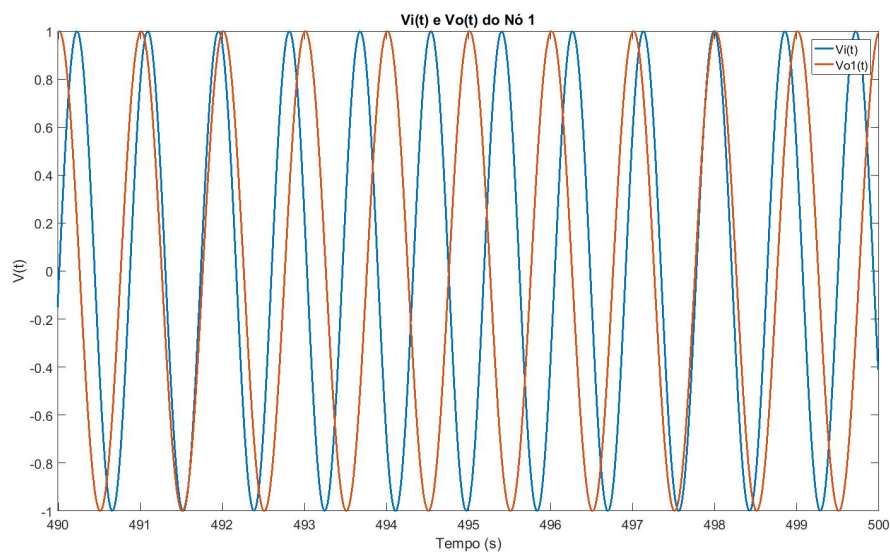
Figura 4.61: Lissajous dos PLLs ao degrau com Filtro F_4



Fonte: autoria própria.

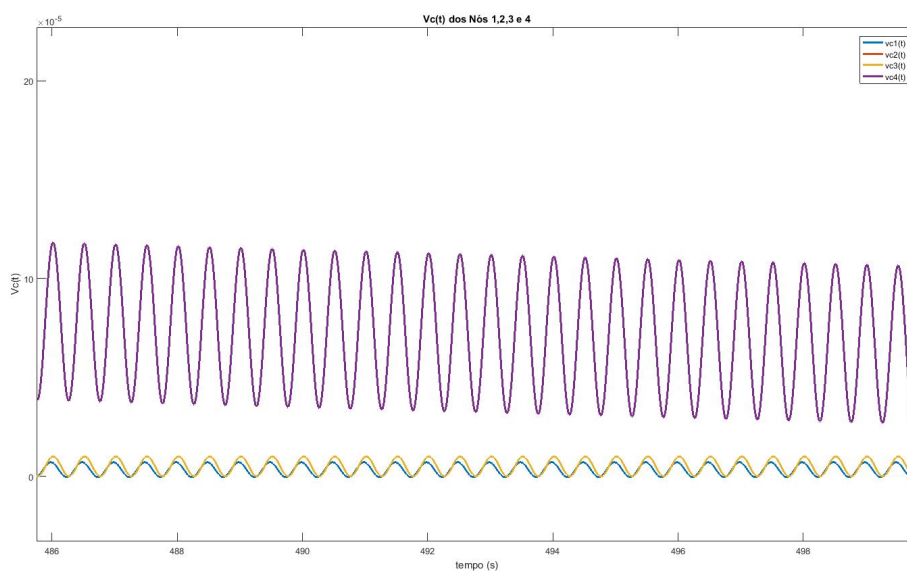
Conforme a Figura 4.61, o Lissajous da rede Estrela TWMS com o Filtro F_4 atingiu o sincronismo. É importante observar que a resposta do Mestre e dos Esravos 2 e 3 são semelhantes e somente o Escravo 4 se diferencia.

Filtro F_4 com Excitação em Rampa de Fase:

Figura 4.62: Resposta em regime dos PLLs a rampa com Filtro F_4 

Fonte: autoria própria.

Os sinais de saída dos PLLs estão indicados na Figura 4.62 e são todos semelhantes, tanto do Mestre quanto dos Escravos, indicando uma possível oscilação periódica.

Figura 4.63: Sinal de Controle dos PLLs a rampa com Filtro F_4 

Fonte: autoria própria.

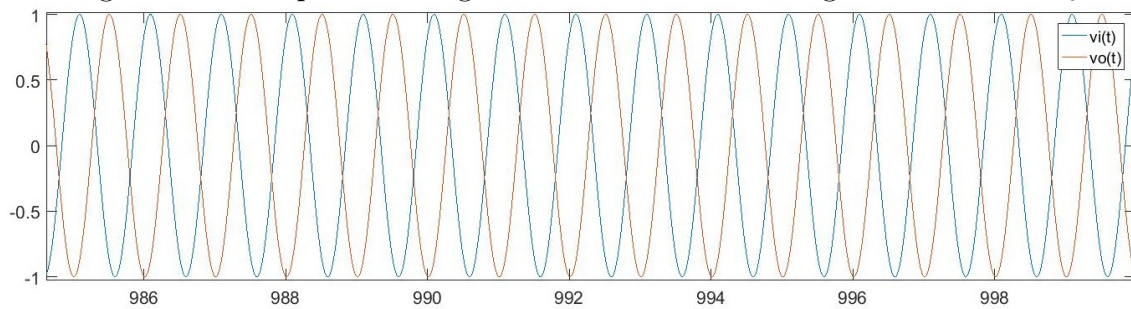
Conforme a Figura 4.63, os sinais de controle oscilam e decaem, tendendo a média zero. É importante observar que a amplitude de oscilação é da ordem de 10^{-5} , indicando

possivelmente que um ajuste no Filtro Controlador possa levar à sincronização da rede em um tempo mais conveniente.

4.5.5 Simulação de rede TWMS com Filtro de 2^a Ordem Tipo 1

Filtro F_5 com Excitação em Degrau de Fase:

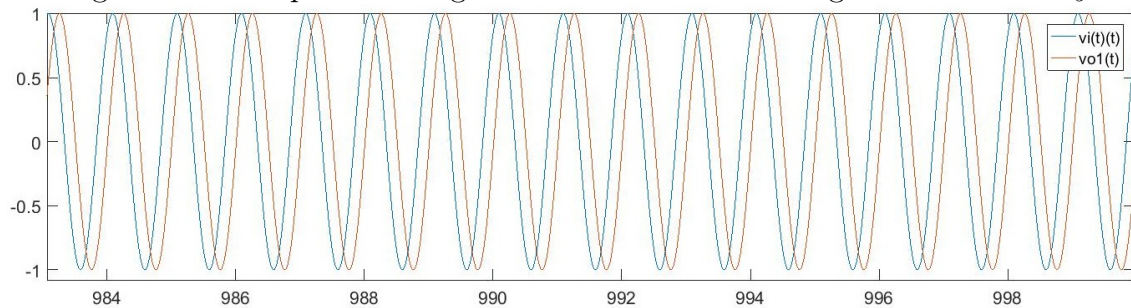
Figura 4.64: Resposta em regime do PLL Mestre ao degraú com Filtro F_5



Fonte: autoria própria.

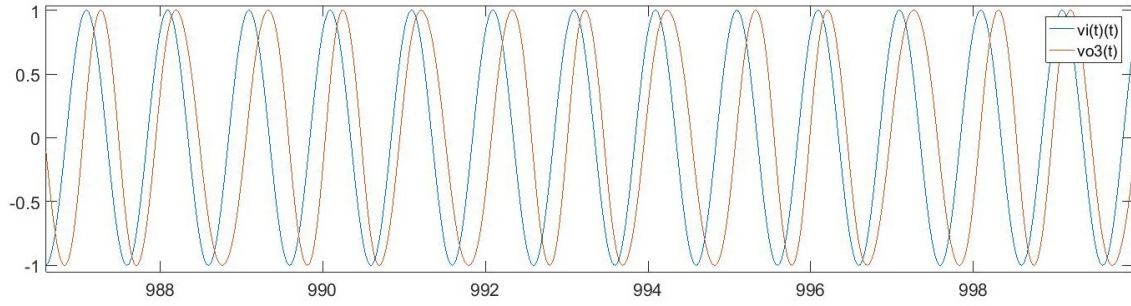
O sinal de saída do PLL Mestre está indicado na Figura 4.64 e observa-se uma defasagem próxima a 180° (inversão de fase).

Figura 4.65: Resposta em regime do 2^o e 3^o PLLs ao degraú com Filtro F_5



Fonte: autoria própria.

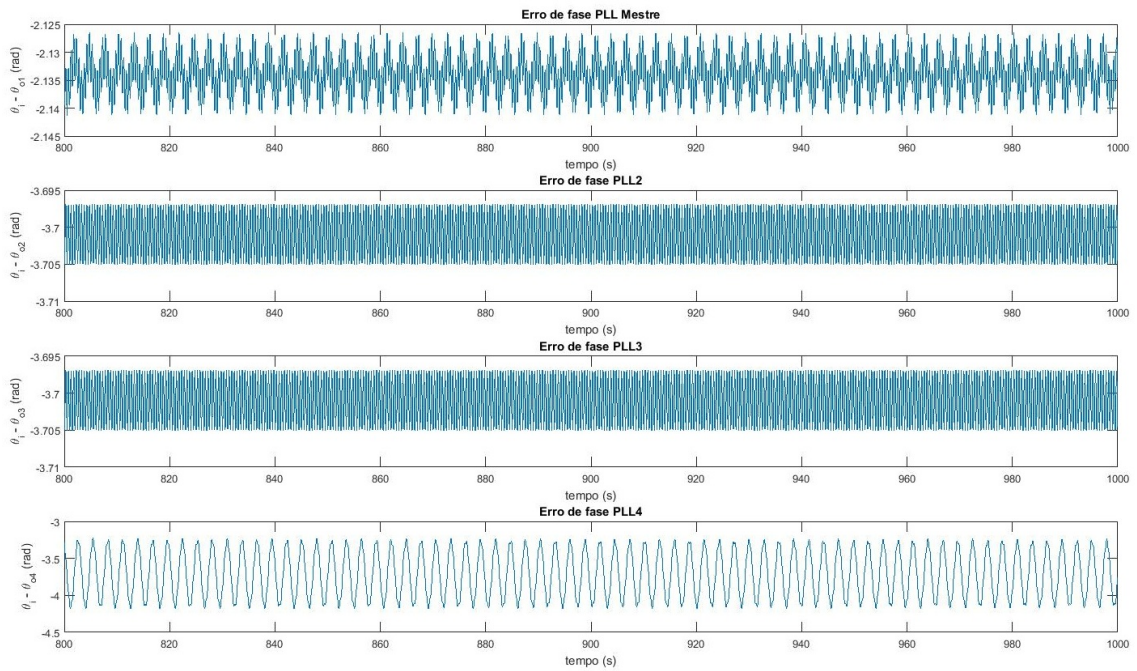
Os sinais de saída dos PLLs Escravos 2 e 3 estão indicados na Figura 4.65 e observa-se uma pequena defasagem constante entre saída e entrada.

Figura 4.66: Resposta em regime do 4º PLL ao degrau com Filtro F_5 

Fonte: autoria própria.

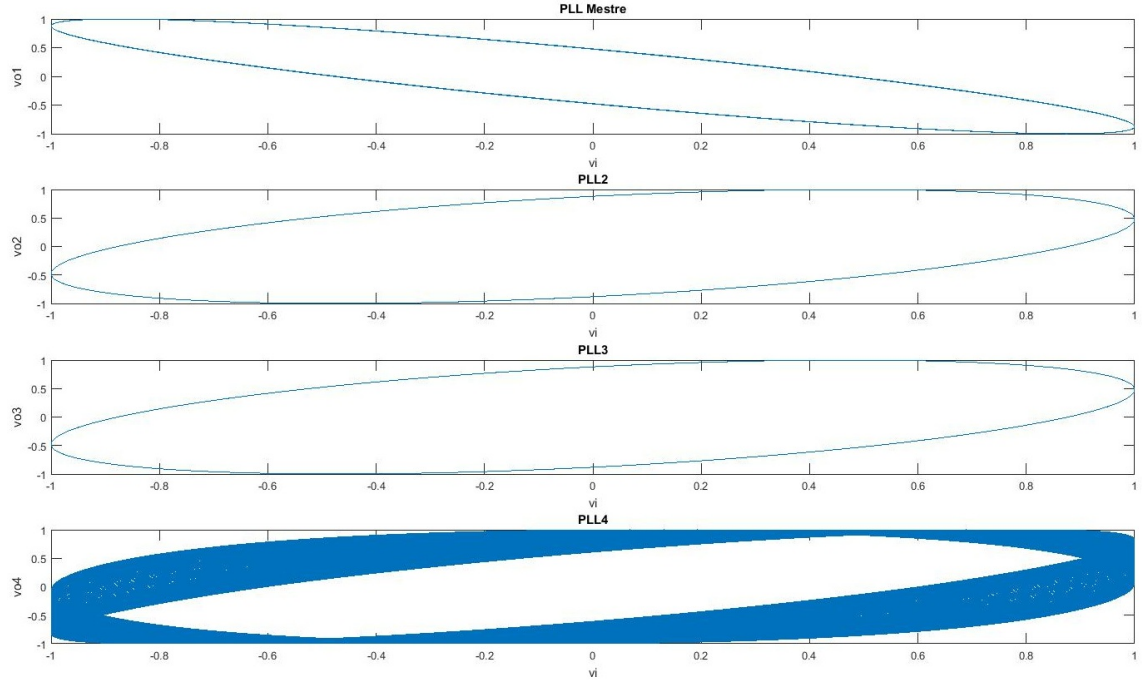
O Sinal de saída do 4º PLL na Figura 4.66 não tem sinal de saída constante 4.66 porém demonstra ser periódico em regime permanente.

Figura 4.67: Erro de fase dos PLLs



Fonte: autoria própria.

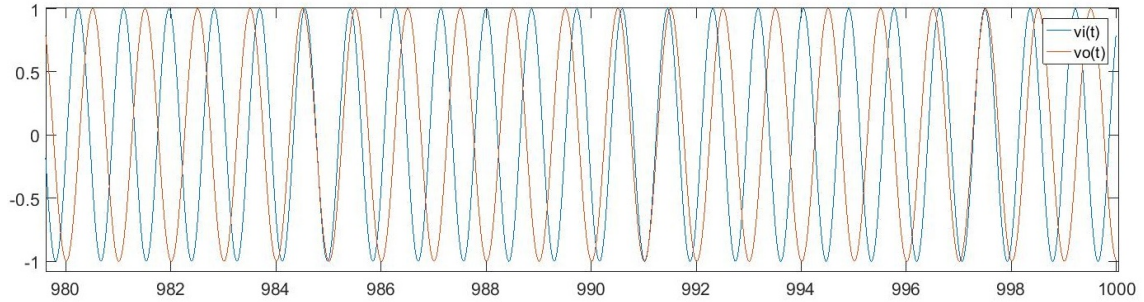
Na Figura 4.67 é perceptível que o erro de fase do Mestre carrega duas componentes: uma em baixa frequência e outra em alta. A componente em alta frequência é repassada para os Escravos 2 e 3, enquanto que a baixa vai para o PLL 4.

Figura 4.68: Lissajous de resposta da rede Estrela TWMS ao degrau com Filtro F_5 

Fonte: autoria própria.

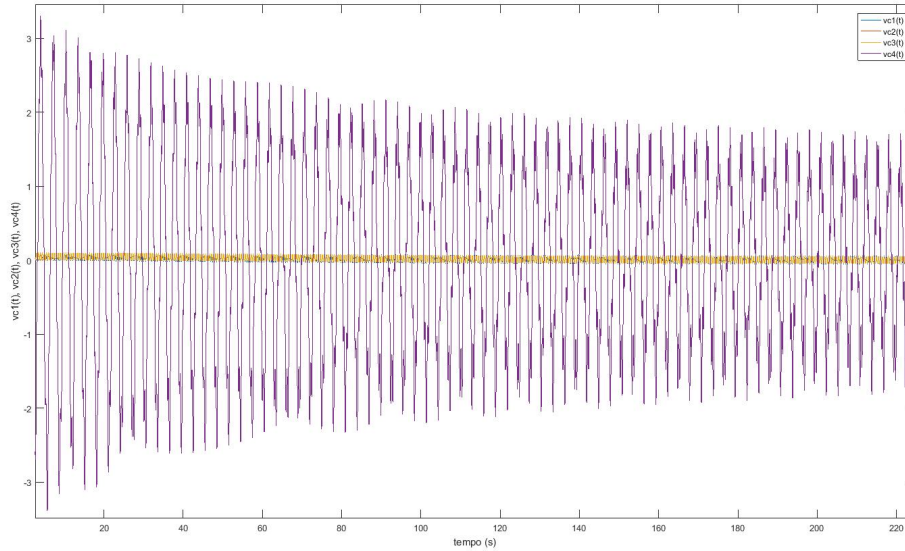
O Lissajous mostrado na Figura 4.68 reafirma a observação feita na Figura 4.68. O PLL 4 encontra o sincronismo, porém oscila mais do que os PLLs 2 e 3, indicando ter uma frequência menor.

Filtro F_5 com Excitação em Rampa de Fase:

Figura 4.69: Resposta em regime dos PLLs Mestre a rampa com Filtro F_5 

Fonte: autoria própria.

Os sinais de saída dos PLLs estão indicados na Figura 4.69 e são todos semelhantes, tanto do Mestre quanto dos Escravos, indicando uma oscilação periódica.

Figura 4.70: Sinais de controle dos PLLs a rampa com Filtro F_5 

Fonte: autoria própria.

A Figura 4.70 representa o sinal de controle de cada PLL da rede. Observa-se que as amplitudes de oscilação diminuem ao longo do tempo e que o valor médio de cada sinal é zero, indicando sincronismo na rede.

Na bibliografia [33], Piqueira e Godoi discutem aplicações semelhantes às apresentadas nesta dissertação, usando o Matlab SimulinkTM com Filtros de 2^a Ordem e atraso de propagação entre os nós. Foram consideradas Cadeias OWMS e TWMS, além de arranjos com nós Mutuamente Conectados (MC). Em particular, é importante ressaltar que o artigo concluiu que a arquitetura Estrela TWMS, apesar de apresentar sincronização total entre os nós escravos, mantém todos dessincronizados em relação ao Mestre, demonstrando que o uso dessa arquitetura está inadequado. E sobre a cadeia OWMS, o mesmo artigo concluiu que existe sincronização rápida e precisa mediante uma perturbação em degrau de fase no sinal do Mestre; diante da perturbação em rampa de fase, podem ser empregados atrasos de compensação para a eliminação de erro de fase residual. As conclusões obtidas por Piqueira e Godoi em [33], são próximas às conclusões obtidas nesta dissertação, ainda que os métodos de simulação sejam diferentes.

Outro ponto a ser destacado e observado nas Simulações desta dissertação, que o emprego de Filtros de 2^a Ordem ou superiores estão sujeitos a comportamentos não-lineares como bifurcações, ciclos-limite (oscilações autossustentadas) e caos, conforme

argumentados em [1, 2, 30, 32].

4.6 Comparação de Tempos da Simulação com os Filtros

Tabela 4.4: Comparação de tempos de simulação para o Processador Intel Core i3-5005S Dual-Core 2,00GHz

Filtro e Excitação	Serial	Paralelo
F_1 - Degrau	10,1978s	18.047,1245s
F_1 - Rampa	11,0913s	18.146,2451s
F_2 - Degrau	10,3914s	18.470,2507s
F_2 - Rampa	10,0258s	18.471,1155s
F_3 - Degrau	11,1413s	18.557,2045s
F_3 - Rampa	11,0772s	18.559,2488s
F_4 - Degrau	135,4564s	20.047,1165s
F_4 - Rampa	135,0012s	20.048,2045s
F_5 - Degrau	137,0319s	20.053,3564s
F_5 - Rampa	137,2154s	20.055,3337s

Tabela 4.5: Comparação de tempos de simulação para o Processador Intel Core i5-4440S Quad-Core 2,80GHz

Filtro e Excitação	Serial	Paralelo
F_1 - Degrau	2,3305s	6.281,8197s
F_1 - Rampa	2,4305s	6.601,1478s
F_2 - Degrau	2,3407s	6.598,1271s
F_2 - Rampa	2,4388s	6.330,2450s
F_3 - Degrau	2,3211s	6.357,4577s
F_3 - Rampa	2,4405s	6.299,7485s
F_4 - Degrau	155,3242s	7.275,5326s
F_4 - Rampa	156,0008s	7.290,1245s
F_5 - Degrau	157,1598s	7.292,0013s
F_5 - Rampa	158,6689s	7.295,5577s

Os tempos tabelados são para a simulação de redes OWMS, visto que os valores para a TWMS são praticamente os mesmos.

Analisando-se as Tabelas 4.4 e 4.5, é possível verificar que o tempo da simulação utilizando paralelismo é maior que o serial. É perceptível dentro da comparação de tempos do Paralelo, que o processador com 4 núcleos ofereceu um tempo menor de simulação em relação ao de 2 núcleos, confirmando a Lei de Amdahl.

Para a simulação dos Filtros de 2^a Ordem (F_4 e F_5) é perceptível, pelo tempo na programação serial, que houve um esforço computacional maior em relação aos Filtros de 1^a Ordem. Isso se deve ao aumento das matrizes (produto vetorial) e consequentemente o volume de dados processados. Em valores absolutos, o serial demonstrou ser mais rápido, porém em valores relativos, comparados na mesma coluna, o serial teve um aumento de 67 vezes o tempo de processamento, enquanto que o paralelo teve 16%, utilizando a simulação no processador com 4 núcleos. É importante observar na programação paralela que o aumento relativo não foi tão considerável, indicando um possível *trade-off* de espaço-tempo computacional para aplicações mais exigentes em *hardware*.

4.7 Resumo do Capítulo

Neste Capítulo 4 foi apresentada a proposta de simulação de redes de sincronismo com o auxílio da *toolbox* PCT do Matlab que provém a programação paralela. Inicialmente, foi realizada uma simulação piloto com dinâmica simples para garantir que o uso do paralelismo não iria influenciar ou distorcer a resposta da rede. Confirmando a possibilidade do paralelismo, foram propostos outros tipos de Filtros Controladores com dinâmicas diferentes para o estudo e análise das redes OWMS e Estrela TWMS. Por fim, são demonstrados os tempos de simulação serial e paralelo.

Capítulo 5

Conclusões

Redes de sincronismo são fundamentais em diversas aplicações da Engenharia e este trabalho colabora com uma aplicação de computação paralela para avançar com as bibliografias já existentes sobre o assunto.

A principal contribuição desta dissertação é a proposta do uso do paralelismo computacional no software Matlab para a solução de problemas de Engenharia. As redes de sincronismo foram empregadas, pois oferecem de forma inerente esta possibilidade, o processamento de cada nó da rede em um núcleo distinto do processador. Como a modelagem do PLL analógico clássico foi revisto em diversas bibliografias, seu comportamento é totalmente conhecido, de forma que proporcionou a comparação dos resultados esperados com aqueles produzidos concorrentemente pelo paralelismo.

A abordagem da computação paralela no Matlab com a ferramenta PCT exigiu a reescrita do código e a vetorização das informações processadas para organizar os resultados provenientes de cada núcleo de processamento.

O estudo exigiu o aprofundamento em conceitos de servomecanismo e sistemas de controle, arquitetura de processadores, lógica de programação, investigação de ferramentas para o paralelismo computacional, bem como a análise das respostas das funções para interpretação correta dos resultados.

O uso do poder computacional, proporciona a simulação de arquiteturas mais complexas de redes de sincronismo, o que aproxima a teoria da prática e permite o desenvolvimento de modelos mais aprimorados e realísticos de aplicações.

Durante as simulações e comparações entre as respostas da programação serial com a paralela, verificou-se que a organização inicial do paralelismo no Matlab ocupa um tempo considerável da simulação. Por se tratar de um software comercial de código

fechado, não é possível fazer intervenções no tratamento que o programa realiza no hardware.

Dessa forma, o paralelismo, como busca de resposta mais rápida de processamento em relação a programação sequencial não correspondeu, porém pelos resultados obtidos, evidenciou-se que aumentando o número de núcleos, o tempo de resposta diminui.

Neste estudo, percebeu-se que existem muitas possibilidades a serem exploradas com o paralelismo computacional dentro do próprio Matlab, desde a adaptação da linguagem de programação, passando pelas diretivas de *hardware* para organizar o paralelismo, até o uso de processadores auxiliares como os GPUs (processadores gráficos das placas de vídeo) para complementar as ações distribuídas.

Conforme observado nas simulações, o uso do paralelismo, empregado nas condições do *hardware* apresentado, foi essencial para compor a complexidade da cadeia de PLLs, principalmente na etapa de integração numérica do sinal de saída do Filtro.

Como possíveis melhorias para este trabalho, indica-se a simulação utilizando processadores com mais núcleos e com maior frequência de operação, testar outras configurações da arquitetura de via dupla (TWMS) e considerar ganhos diferentes para a malha de cada nó.

Para trabalhos futuros, o Autor sugere que sejam exploradas outras arquiteturas de redes de sincronismo considerando o espaço de fase do sinal, respeitando a não-linearidade do PLL ao incluir a variação de frequência entre os nós da rede e as perturbações produzidas pelos múltiplos detectores de fase; o estudo de ciclos-limite e bifurcações; a aplicação de redes de sincronismo com o DTL (ver subseção 2.2) e o aumento da fração do código a ser paralelizável. O Autor também indica a possibilidade de uso de outros ambientes de desenvolvimento da programação paralela, principalmente os de código aberto, onde seja possível manipular diretivas de gerenciamento do *hardware*.

Referências Bibliográficas

- [1] BUENO, Âtila Madureira. Estudo do jitter de fase em redes de distribuição de sinais de tempo. 2009. Tese de Doutorado. Universidade de São Paulo.
- [2] PIQUEIRA, José Roberto Castilho. Uma contribuição ao estudo das redes com malhas de sincronismo de fase. 1997. Tese de Livre Docência. Universidade de São Paulo.
- [3] PIQUEIRA, José RC; CALIGARES, Andrea Z.; MONTEIRO, Luiz HA. Double-frequency jitter figures in master–slave PLL networks. *AEU-International Journal of Electronics and Communications*, v. 61, n. 10, p. 678-683, 2007.
- [4] LINDSEY, William C.; CHIE, Chak Ming. A survey of digital phase-locked loops. *Proceedings of the IEEE*, v. 69, n. 4, p. 410-431, 1981.
- [5] LINDSEY, W. C. et al. Network synchronization. In: *Proceedings of the IEEE*. [S.l.: s.n.], 1985. v. 73, n. 10, p. 1445–1467.
- [6] POPKEN, LUITJENS et al. Synchronization failure rates in master-slave synchronized phase-locked loops. *IEEE transactions on aerospace and electronic systems*, v. 29, n. 2, p. 328-335, 1993.
- [7] MEYR, H.; ASCHEID, G. Synchronization in Digital Communications Phase-Frequency-Locked Loops, and Amplitude Control. [S.l.]: John Wiley & Sons, 1990. 510 p.
- [8] KUZNETSOV, N. V. et al. Nonlinear analysis of classical phase-locked loops in signal’s phase space. *IFAC Proceedings Volumes*, v. 47, n. 3, p. 8253-8258, 2014.
- [9] BUENO, Atila Madureira; FERREIRA, André Alves; PIQUEIRA, José Roberto Castilho. Modeling and filtering double-frequency jitter in one-way master–slave chain networks. *IEEE Transactions on Circuits and Systems I: Regular Papers*, v. 57, n. 12, p. 3104-3111, 2010.

- [10] BEST, R. E. Phase-Locked Loops - Design, Simulation, and Applications. 4. ed. [S.l.]: McGraw-Hill, 1999. 408 p.
- [11] GARDNER, F. M. Phaselock Techniques. 3. ed. [S.l.]: John Wiley & Sons, Inc., 2005.
- [12] BREGNI, S. Synchronization of Digital Telecommunications Networks. [S.l.]: John Wiley & Sons, 2002.
- [13] BREGNI, S. A historical perspective on telecommunications network synchronization. IEEE Communications Magazine, v. 36, n. 6, p. 158–166, jun. 1998.
- [14] PACHECO, Peter. An introduction to parallel programming. Elsevier, 2011.
- [15] HERLIHY, Maurice; SHAVIT, Nir. The Art of Multiprocessor Programming [SMP]. 2008.
- [16] ANDREWS, Gregory R. Foundations of multithreaded, parallel, and distributed programming. Addison-Wesley, 2020.
- [17] LYNCH, Stephen. Dynamical systems with applications using MATLAB. Boston: Birkhäuser, 2004.
- [18] GILAT, Amos. MATLAB com aplicações em engenharia. Bookman Editora, 2009.
- [19] ALYASSERI, Zaid Abdi Alkareem. Survey of Parallel Computing with MATLAB. arXiv preprint arXiv:1407.6878, 2014.
- [20] HEMANTH, D. J.; ELHOSNEY, M.; NGUYEN, T. N. (Ed.). Advances in parallel computing technologies and applications. IOS Press, 2021.
- [21] GREEN, D. Lock-in, tracking, and acquisition of AGC-aided phase-locked loops. IEEE transactions on circuits and systems, v. 32, n. 6, p. 559-568, 1985.
- [22] VAN TREES, Harry L. Detection, estimation, and modulation theory, part I: detection, estimation, and linear modulation theory. John Wiley & Sons, 2004.
- [23] MURRAY, James Dickson. Mathematical biology: I. An introduction. 2002.
- [24] MOLER, C. Why isn't There a Parallel MATLAB. The MathWorks Newsletter, Spring 1995

- [25] KRISHNAMURTHY, Ashok et al. Survey of parallel MATLAB techniques and applications to signal and image processing. In: 2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07. IEEE, 2007. p. IV-1181-IV-1184.
- [26] LEE, Jae; UN, Chong. Performance analysis of digital tanlock loop. IEEE Transactions on Communications, v. 30, n. 10, p. 2398-2411, 1982.
- [27] POPOV, George; MASTORAKIS, Nikos; MLADENOV, Valeri. Calculation of the acceleration of parallel programs as a function of the number of threads. Proceedings of the ICCOMP, v. 10, p. 411-414, 2010.
- [28] WIGGINS, Stephen; WIGGINS, Stephen; GOLUBITSKY, Martin. Introduction to applied nonlinear dynamical systems and chaos. New York: Springer, 2003.
- [29] THORTON, Stephen T.; MARION, Jerry B. Dinâmica clássica de partículas e sistemas. Cengage Learning, 2011.
- [30] PIKOVSKY, Arkady; ROSENBLUM, Michael; KURTHS, Jürgen. Synchronization: a universal concept in nonlinear science. 2002.
- [31] KAPLAN, Elliott D.; HEGARTY, Christopher (Ed.). Understanding GPS/GNSS: Principles and applications. Artech house, 2017.
- [32] PIQUEIRA, José Roberto Castilho; CASTILLO-VARGAS, S. A.; MONTEIRO, Luiz Henrique Alves. Two-way master-slave double-chain networks: limitations imposed by linear master drift for second order PLLs as slave nodes. IEEE Communications Letters, v. 9, n. 9, p. 829-831, 2005.
- [33] PIQUEIRA, José Roberto C.; DE GODOI, Antonio Carlos B. Clock signal distribution with second order nodes: Design hints. ISA transactions, v. 115, p. 124-142, 2021.

Apêndice A

Sistemas Dinâmicos

A.1 Método de Integração Numérica

Grande parte dos problemas encontrados na Engenharia, são formulados em termos de equações diferenciais. Problemas envolvendo estudos de distribuição de sinais de relógio, dinâmica de transmissão de sinais, circuitos elétricos, dinâmica de estruturas e vibrações, reações químicas e muito mais. Um método numérico para resolver uma EDO (Equação Diferencial Ordinária), a partir de uma condição inicial dada, é um processo de solução aproximada em pontos particulares utilizando apenas as quatro operações matemáticas básicas e alguns cálculos funcionais.

$$\dot{x}(t) = f(x, t), \quad x(t_0) = x_0 \quad (\text{A.1})$$

Cada método numérico calcula soluções aproximadas para $t_0, t_1, t_2, \dots$, sendo que a diferença de tempo entre dois instantes sucessivos é T , ou seja, $T = t_{k+1} - t_k$, e $t_k = kT$ para $k = 0, 1, 2, \dots$

Designa-se a solução $x(t_k) = x_k$ e conhecendo-se x_k pode-se utilizar a equação (3.30) para se calcular $\dot{x}_k$:

$$\dot{x}_k = f(x_k, t_k) \quad (\text{A.2})$$

O método mais simples de integração numérica ponto-a-ponto é o método de Euler, que calcula aproximadamente o valor de x_k em um único passo. A desvantagem é a de ser o método que gera o maior erro entre a solução exata analítica e a aproximada por pontos.

O método de Runge-Kutta¹ de 4a ordem é um método de múltiplos estágios,

¹Por volta do ano 1900, dois matemáticos alemães, Carl David Tolmé Runge e Martin Wilhelm

podendo-se aumentar ou diminuir o número de estágios dependendo da necessidade. é empregado neste trabalho para fazer a integração da saída do sinal do filtro e o cálculo realizado com paralelismo computacional. O método é descrito por:

$$x_{k+1} = x_k + \frac{1}{6}(L_1 + 2L_2 + 3L_3 + L_4) \quad (\text{A.3})$$

com

$$\begin{aligned} L_1 &= Tf(x_k, kT) \\ L_2 &= Tf\left(x_k + \frac{1}{2}L_1, kT + \frac{1}{2}T\right) \\ L_3 &= Tf\left(x_k + \frac{1}{2}L_2, kT + \frac{1}{2}T\right) \\ L_4 &= Tf(x_k + L_3, kT + T) \end{aligned} \quad (\text{A.4})$$

Kutta, aprimoraram os métodos iterativos para resoluções de EDOs de uma forma extremamente precisa, pois perceberam que quanto mais inclinações (derivadas) fossem calculadas, melhor seria a precisão, sendo cada inclinação associada a um determinado coeficiente de compensação.

A.2 Curva de Lissajous

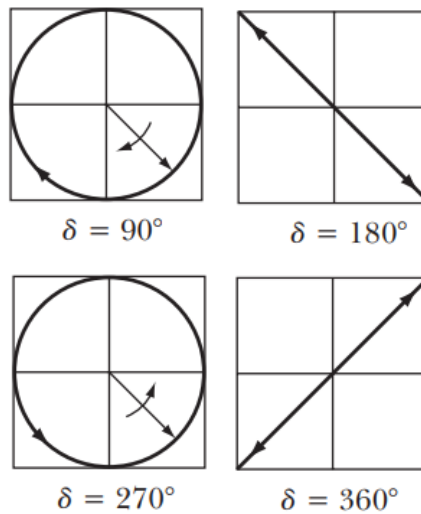
A curva de Lissajous é o gráfico produzido por um sistema de equações paramétricas do tipo:

$$\begin{aligned}x(t) &= A.\text{sen}(\omega_a t + \delta) \\ y(t) &= B.\text{sen}(\omega_b t)\end{aligned}\tag{A.5}$$

que descrevem um complexo movimento harmônico. Essa família de curvas foi estudada por Nathaniel Bowditch em 1815, e mais tarde por Jules Antoine Lissajous, em 1857 [28].

A aparência do gráfico é sensível à razão das frequências angulares ω_a/ω_b que, sendo um número racional, se torna estático. Quando esta razão é 1, o gráfico produzido é uma elipse, podendo formar círculos quando as amplitudes $A = B$ e $\delta = 90^\circ$ ou 270° , e também retas quando $\delta = 180^\circ$ ou 360° .

Figura A.1: Movimento de oscilação harmônica



Adaptado de [28]