

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



Matheus Silva Batista

Simulação de rede de sincronização mestre escravo de duas vias
(TWMS) com computação paralela

Sorocaba

2022

Matheus Silva Batista

Simulação de rede de sincronização mestre escravo de duas vias
(TWMS) com 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: Prof.Dra. Maria Gloria Caño de Andrade

Sorocaba

2022

B333s Batista, Matheus Silva
Simulação de rede de sincronização mestre escravo de duas vias (TWMS) com computação paralela / Matheus Silva Batista. -- Sorocaba, 2022
133 p.

Dissertação (mestrado) - Universidade Estadual Paulista (Unesp), Instituto de Ciência e Tecnologia, Sorocaba
Orientador: Átila Madureira Bueno
Coorientadora: Maria Gloria Caño de Andrade

1. Teoria dos sistemas. 2. Sistemas não lineares. 3. Circuito de retenção de fase. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Biblioteca do Instituto de Ciência e Tecnologia, Sorocaba. Dados fornecidos pelo autor(a).

Essa ficha não pode ser modificada.

Matheus Silva Batista

Simulação de rede de sincronização mestre escravo de duas vias
(TWMS) com 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

Universidade de São Paulo - Escola Politécnica - Departamento de Engenharia de Telecomunicações e Controle

Orientador

Prof. Dra Maria Gloria Caño de Andrade

Universidade Estadual Paulista “Júlio de Mesquita Filho” - Instituto de Ciência e Tecnologia - Campus Sorocaba

Coorientadora

Prof. Dr Paulo José Amaral Serni

Universidade Estadual Paulista “Júlio de Mesquita Filho” - Faculdade de Engenharia e Tecnologia de Bauru - Campus Bauru

Prof. Dr Diego Paolo Ferruzzo Correa

Universidade Federal do ABC - Centro de Engenharia, Modelagem e Ciências Sociais Aplicadas - Campus São Bernardo do Campo

Sorocaba

2022

CERTIFICADO DE APROVAÇÃO

TÍTULO DA DISSERTAÇÃO: Simulação de rede de sincronização TWMS com computação paralela

AUTOR: MATHEUS SILVA BATISTA

ORIENTADOR: ATILA MADUREIRA BUENO

COORIENTADORA: MARIA GLÓRIA CAÑO DE ANDRADE

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

Prof. Dr. ATILA MADUREIRA BUENO (Participação Virtual)
Politecnico / Universidade de Sao Paulo USP



Prof. Dr. DIEGO PAOLO FERRUZZO CORREA (Participação Virtual)
Centro de Engenharia, Modelagem e Ciências Sociais Aplicadas / Universidade Federal do ABC / UFABC



Prof. Dr. PAULO JOSÉ AMARAL SERNI (Participação Virtual)
Departamento de Engenharia Elétrica - docente voluntario / Faculdade de Engenharia de Bauru - UNESP



Sorocaba, 16 de dezembro de 2022

*Em especial à minha namorada Giovanna, aos meus pais Jesus e Andreia e ao meu irmão Rafael, que me acompanharam nessa jornada e deram sua ajuda, amor e muita paciência para superar os obstáculos e momentos adversos que tivemos que enfrentar durante o curso...
dedico.*

“All we have to decide is what to do with the time that is given us.”

Gandalf The Gray
The Lord of the Rings

Agradecimentos

Aos meus pais Jesus Lopes Batista e Andreia Aparecida Silva Batista, que me apoiaram e me mostraram desde pequeno o caminho de dedicação, interesse e amor aos estudos e me possibilitaram estar alcançando mais esse título. Ao meu irmão, Rafael da Silva Batista, pela amizade e companheirismo.

À minha namorada Giovanna de Souza Ragassi, por ter me incentivado em momentos de dúvidas durante a trilha e pela companhia em momentos de felicidade.

Ao meu orientador, Professor Doutor Átila Madureira Bueno, pela oportunidade, confiança e paciência no desenvolvimento do projeto e pela distinção na entrega desse projeto.

Ao meu amigo e companheiro de jornada, Elvio Alexandrini Maciel, de Sorocaba, pelos momentos de estudos, perrengues e superação.

À professora Cristiane Mileo Batistela Gouvêa (UFABC), por sua atenção e disposição na colaboração e apresentação desse trabalho no ENOC 2022, publicação internacional a qual sou honrado de ter alcançado.

À Unesp, em especial ao Campus de São João da Boa Vista, pelas clarezas tanto às dúvidas processuais quanto na disponibilidade de meios para o estudo.

Àquelas pessoas que não foram citadas, mas que de alguma forma colaboraram com o meu sucesso no decorrer das disciplinas e na conclusão do Mestrado, agradeço.

Resumo

A aplicação de redes de distribuição de sinal de relógio mestre escravo de duas vias (do inglês, two-way master slave - TWMS) atua no espalhamento e troca de informações entre os componentes de computação, comunicação e sistemas dinâmicos complexos. A crescente necessidade de comunicação e trocas de informações trouxeram a necessidade de desenvolvimento de uma rede mundial interconectada para suprir essa demanda. A rede de distribuição de sinal de relógio é uma parte importante dessas redes e tem com o Phase locked-loop (PLL), a dinâmica necessária para a sincronização de um oscilador local com um sinal de referencia em fase e frequência. Usualmente, as redes PLL são construídas em topologias mutuamente conectadas ou mestre-escravo, sendo a utilização dessa configuração devido à sua simplicidade e baixo custo. A simulação numérica de redes de distribuição de sinal de relógio é importante devido à capacidade de se testar possibilidades e características dos componentes do PLL. Neste trabalho, uma rede TWMS é construída usando lógica de algoritmos paralelos, com o objetivo de desenvolver uma estratégia de simulação mais eficiente, e estudar os efeitos da interação de PLLs rodando em um sistema de computação paralela.

Palavras-chave: teoria dos sistemas, sistemas não lineares, circuitos de retenção de fase.

Abstract

The application of two-way master slave clock signal distribution networks (TWMS) operates in the spreading and exchange of information between components of computing, communication and complex dynamic systems. The growing need for communication and information exchange brought the need to develop an interconnected world network to meet this demand. The clock signal distribution network is an important part of these networks and has, with the Phase locked-loop (PLL), the necessary dynamics for the synchronization of a local oscillator with a reference signal in phase and frequency. Usually, PLL networks are built in mutually connected or master-slave topologies, and the use of this configuration is due to its simplicity and low cost. Numerical simulation of clock signal distribution networks is important due to the ability to test possibilities and characteristics of PLL components. In this work, a TWMS network is built using parallel algorithm logic, with the objective of developing a more efficient simulation strategy, and studying the interaction effects of PLLs running in a parallel computing system.

Keywords: systems theory, nonlinear systems, phase retention circuits.

Lista de Figuras

2.1	Diagrama de blocos de PLL	22
2.2	Estrutura convencional de ADPLL em diagrama de blocos	24
2.3	Diagrama de blocos do PLL analógico clássico	25
2.4	Diagrama de fasor descrevendo a operação do PLL (números complexos estão sublinhados)	25
2.5	Diagrama de blocos do PLL linearizado	30
2.6	Estratégia de distribuição de sinais de tempo	32
2.7	Rede OWMS de cadeia simples	33
2.8	Rede OWMS estrela simples	33
2.9	Rede TWMS cadeia dupla	34
2.10	Rede TWMS estrela dupla	34
2.11	Rede TWMS enlace simples	35
2.12	Rede TWMS enlace duplo	35
2.13	Diagrama de blocos dos PLL de uma rede TWMS	36
2.14	<i>Multitasking</i>	42
2.15	<i>Microtasking</i>	43
2.16	<i>Macrotasking</i>	43
3.1	Topologia hipercubo de processadores	48
3.2	Topologia das <i>toolboxes</i> PCT e MDSC	49
3.3	Fluxograma do algoritmo proposto	52
3.4	Fluxograma da paralelização aplicada	54

4.1	Rede TWMS com 3 nós escravos	56
4.2	Resposta em regime do Nó Escravo 1 - Degrau unitário com o Filtro 1 .	57
4.3	Resposta em regime do Nó Escravo 1 - Degrau unitário com o Filtro 2 .	58
4.4	Resposta em regime do Nó Escravo 1 - Degrau unitário com o Filtro 3 .	58
4.5	Resposta em regime do Nó Escravo 1 - Degrau unitário com o Filtro 4 .	59
4.6	Resposta em regime do Nó Escravo 1 - Degrau unitário com o Filtro 5 .	59
4.7	Resposta em regime do Nó Escravo 1 - Degrau unitário com o Filtro 6 .	60
4.8	Resposta em regime do Nó Escravo 2 - Degrau unitário com o Filtro 1 .	61
4.9	Resposta em regime do Nó Escravo 2 - Degrau unitário com o Filtro 2 .	61
4.10	Resposta em regime do Nó Escravo 2 - Degrau unitário com o Filtro 3 .	62
4.11	Resposta em regime do Nó Escravo 2 - Degrau unitário com o Filtro 4 .	62
4.12	Resposta em regime do Nó Escravo 2 - Degrau unitário com o Filtro 5 .	63
4.13	Resposta em regime do Nó Escravo 2 - Degrau unitário com o Filtro 6 .	63
4.14	Resposta em regime do Nó Escravo 3 - Degrau unitário com o Filtro 1 .	64
4.15	Resposta em regime do Nó Escravo 3 - Degrau unitário com o Filtro 2 .	65
4.16	Resposta em regime do Nó Escravo 3 - Degrau unitário com o Filtro 3 .	65
4.17	Resposta em regime do Nó Escravo 3 - Degrau unitário com o Filtro 4 .	66
4.18	Resposta em regime do Nó Escravo 3 - Degrau unitário com o Filtro 5 .	66
4.19	Resposta em regime do Nó Escravo 3 - Degrau unitário com o Filtro 6 .	67
4.20	Erro de fase da rede TWMS - Degrau unitário com o Filtro 1	68
4.21	Erro de fase da rede TWMS - Degrau unitário com o Filtro 2	68
4.22	Erro de fase da rede TWMS - Degrau unitário com o Filtro 3	69
4.23	Erro de fase da rede TWMS - Degrau unitário com o Filtro 4	69
4.24	Erro de fase da rede TWMS - Degrau unitário com o Filtro 5	70
4.25	Erro de fase da rede TWMS - Degrau unitário com o Filtro 6	70
4.26	$V_c(t)$ para cada nó da rede TWMS - Degrau unitário com o Filtro 1 . .	71
4.27	$V_c(t)$ para cada nó da rede TWMS - Degrau unitário com o Filtro 2 . .	72
4.28	$V_c(t)$ para cada nó da rede TWMS - Degrau unitário com o Filtro 3 . .	72

4.29	$V_c(t)$ para cada nó da rede TWMS - Degrau unitário com o Filtro 4 . .	73
4.30	$V_c(t)$ para cada nó da rede TWMS - Degrau unitário com o Filtro 5 . .	73
4.31	$V_c(t)$ para cada nó da rede TWMS - Degrau unitário com o Filtro 6 . .	74
4.32	Lissajous para cada nó da rede TWMS - Degrau unitário com o Filtro 1	75
4.33	Lissajous para cada nó da rede TWMS - Degrau unitário com o Filtro 2	75
4.34	Lissajous para cada nó da rede TWMS - Degrau unitário com o Filtro 3	76
4.35	Lissajous para cada nó da rede TWMS - Degrau unitário com o Filtro 4	76
4.36	Lissajous para cada nó da rede TWMS - Degrau unitário com o Filtro 5	77
4.37	Lissajous para cada nó da rede TWMS - Degrau unitário com o Filtro 6	77
4.38	Resposta em regime do Nó Escravo 1 - Rampa unitária com o Filtro 1 .	80
4.39	Resposta em regime do Nó Escravo 1 - Rampa unitária com o Filtro 2 .	80
4.40	Resposta em regime do Nó Escravo 1 - Rampa unitária com o Filtro 3 .	81
4.41	Resposta em regime do Nó Escravo 1 - Rampa unitária com o Filtro 4 .	81
4.42	Resposta em regime do Nó Escravo 1 - Rampa unitária com o Filtro 5 .	82
4.43	Resposta em regime do Nó Escravo 1 - Rampa unitária com o Filtro 6 .	82
4.44	Resposta em regime do Nó Escravo 2 - Rampa unitária com o Filtro 1 .	83
4.45	Resposta em regime do Nó Escravo 2 - Rampa unitária com o Filtro 2 .	83
4.46	Resposta em regime do Nó Escravo 2 - Rampa unitária com o Filtro 3 .	84
4.47	Resposta em regime do Nó Escravo 2 - Rampa unitária com o Filtro 4 .	84
4.48	Resposta em regime do Nó Escravo 2 - Rampa unitária com o Filtro 5 .	85
4.49	Resposta em regime do Nó Escravo 2 - Rampa unitária com o Filtro 6 .	85
4.50	Resposta em regime do Nó Escravo 3 - Rampa unitária com o Filtro 1 .	86
4.51	Resposta em regime do Nó Escravo 3 - Rampa unitária com o Filtro 2 .	87
4.52	Resposta em regime do Nó Escravo 3 - Rampa unitária com o Filtro 3 .	87
4.53	Resposta em regime do Nó Escravo 3 - Rampa unitária com o Filtro 4 .	88
4.54	Resposta em regime do Nó Escravo 3 - Rampa unitária com o Filtro 5 .	88
4.55	Resposta em regime do Nó Escravo 3 - Rampa unitária com o Filtro 6 .	89
4.56	Erro de fase da rede TWMS - Rampa unitária com o Filtro 1	90

4.57	Erro de fase da rede TWMS - Rampa unitária com o Filtro 2	90
4.58	Erro de fase da rede TWMS - Rampa unitária com o Filtro 3	91
4.59	Erro de fase da rede TWMS - Rampa unitária com o Filtro 4	91
4.60	Erro de fase da rede TWMS - Rampa unitária com o Filtro 5	92
4.61	Erro de fase da rede TWMS - Rampa unitária com o Filtro 6	92
4.62	$V_c(t)$ para cada nó da rede TWMS - Rampa unitária com o Filtro 1 . .	93
4.63	$V_c(t)$ para cada nó da rede TWMS - Rampa unitária com o Filtro 2 . .	94
4.64	$V_c(t)$ para cada nó da rede TWMS - Rampa unitária com o Filtro 3 . .	94
4.65	$V_c(t)$ para cada nó da rede TWMS - Rampa unitária com o Filtro 4 . .	95
4.66	$V_c(t)$ para cada nó da rede TWMS - Rampa unitária com o Filtro 5 . .	95
4.67	$V_c(t)$ para cada nó da rede TWMS - Rampa unitária com o Filtro 6 . .	96
4.68	Lissajous para cada nó da rede TWMS - Rampa unitária com o Filtro 1	97
4.69	Lissajous para cada nó da rede TWMS - Rampa unitária com o Filtro 2	97
4.70	Lissajous para cada nó da rede TWMS - Rampa unitária com o Filtro 3	98
4.71	Lissajous para cada nó da rede TWMS - Rampa unitária com o Filtro 4	98
4.72	Lissajous para cada nó da rede TWMS - Rampa unitária com o Filtro 5	99
4.73	Lissajous para cada nó da rede TWMS - Rampa unitária com o Filtro 6	99
4.74	Resposta em regime do Nó Escravo 1 - Dente de Serra com o Filtro 1 .	101
4.75	Resposta em regime do Nó Escravo 1 - Dente de Serra com o Filtro 2 .	102
4.76	Resposta em regime do Nó Escravo 1 - Dente de Serra com o Filtro 3 .	102
4.77	Resposta em regime do Nó Escravo 1 - Dente de Serra com o Filtro 4 .	103
4.78	Resposta em regime do Nó Escravo 1 - Dente de Serra com o Filtro 5 .	103
4.79	Resposta em regime do Nó Escravo 1 - Dente de Serra com o Filtro 6 .	104
4.80	Resposta em regime do Nó Escravo 2 - Dente de Serra com o Filtro 1 .	105
4.81	Resposta em regime do Nó Escravo 2 - Dente de Serra com o Filtro 2 .	105
4.82	Resposta em regime do Nó Escravo 2 - Dente de Serra com o Filtro 3 .	106
4.83	Resposta em regime do Nó Escravo 2 - Dente de Serra com o Filtro 4 .	106
4.84	Resposta em regime do Nó Escravo 2 - Dente de Serra com o Filtro 5 .	107

4.85	Resposta em regime do Nó Escravo 2 - Dente de Serra com o Filtro 6 .	107
4.86	Resposta em regime do Nó Escravo 3 - Dente de Serra com o Filtro 1 .	108
4.87	Resposta em regime do Nó Escravo 3 - Dente de Serra com o Filtro 2 .	108
4.88	Resposta em regime do Nó Escravo 3 - Dente de Serra com o Filtro 3 .	109
4.89	Resposta em regime do Nó Escravo 3 - Dente de Serra com o Filtro 4 .	109
4.90	Resposta em regime do Nó Escravo 3 - Dente de Serra com o Filtro 5 .	110
4.91	Resposta em regime do Nó Escravo 3 - Dente de Serra com o Filtro 6 .	110
4.92	Erro de fase da rede TWMS - Dente de Serra com o Filtro 1	111
4.93	Erro de fase da rede TWMS - Dente de Serra com o Filtro 2	112
4.94	Erro de fase da rede TWMS - Dente de Serra com o Filtro 3	112
4.95	Erro de fase da rede TWMS - Dente de Serra com o Filtro 4	113
4.96	Erro de fase da rede TWMS - Dente de Serra com o Filtro 5	113
4.97	Erro de fase da rede TWMS - Dente de Serra com o Filtro 6	114
4.98	$V_c(t)$ para cada nó da rede TWMS - Dente de Serra com o Filtro 1 . . .	115
4.99	$V_c(t)$ para cada nó da rede TWMS - Dente de Serra com o Filtro 2 . . .	115
4.100	$V_c(t)$ para cada nó da rede TWMS - Dente de Serra com o Filtro 3 . . .	116
4.101	$V_c(t)$ para cada nó da rede TWMS - Dente de Serra com o Filtro 4 . . .	116
4.102	$V_c(t)$ para cada nó da rede TWMS - Dente de Serra com o Filtro 5 . . .	117
4.103	$V_c(t)$ para cada nó da rede TWMS - Dente de Serra com o Filtro 6 . . .	117
4.104	Lissajous para cada nó da rede TWMS - Dente de Serra com o Filtro 1	118
4.105	Lissajous para cada nó da rede TWMS - Dente de Serra com o Filtro 2	119
4.106	Lissajous para cada nó da rede TWMS - Dente de Serra com o Filtro 3	119
4.107	Lissajous para cada nó da rede TWMS - Dente de Serra com o Filtro 4	120
4.108	Lissajous para cada nó da rede TWMS - Dente de Serra com o Filtro 5	120
4.109	Lissajous para cada nó da rede TWMS - Dente de Serra com o Filtro 6	121

Lista de Tabelas

3.1	Filtros passa-baixas	55
4.1	Tempos de simulação para a resposta ao degrau de fase - Intel [®] Core [™] i3-4005U CPU @ 1.70GHz (dois núcleos)	78
4.2	Tempos de simulação para a resposta ao degrau de fase - Intel [®] Core [™] i5-4440S CPU @ 2.8GHz (quatro núcleos)	79
4.3	Tempos de simulação para a resposta à rampa de fase - Intel [®] Core [™] i3-4005U CPU @ 1.70GHz (dois núcleos)	100
4.4	Tempos de simulação para a resposta à rampa de fase - Intel [®] Core [™] i5-4440S CPU @ 2.8GHz (quatro núcleos)	100
4.5	Tempos de simulação para a resposta ao dente de serra de fase - Intel [®] Core [™] i3-4005U CPU @ 1.70GHz (dois núcleos)	122
4.6	Tempos de simulação para a resposta ao dente de serra de fase - Intel [®] Core [™] i5-4440S CPU @ 2.8GHz (quatro núcleos)	122

Sumário

1	Introdução	17
1.1	Motivação	19
1.2	Objetivos	20
1.3	Roteiro do trabalho	20
2	Revisão Bibliográfica	22
2.1	<i>Phase Locked Loop - PLL</i>	22
2.1.1	Modelo matemático do PLL analógico	24
2.1.2	Modos de Operação do PLL	30
2.1.3	Estabilidade e Sincronização	31
2.2	Redes de distribuição de sinal de relógio	32
2.2.1	OWMS - Cadeia Simples	33
2.2.2	OWMS - Estrela simples	33
2.2.3	TWMS - Cadeia dupla	34
2.2.4	TWMS - Estrela dupla	34
2.2.5	TWMS - Enlace simples	34
2.2.6	TWMS - Enlace duplo	35
2.2.7	Modelagem da rede TWMS	36
2.3	Algoritmos paralelos	41
2.3.1	Macrotarefas e Microtarefas	42
2.3.2	Desempenho e métricas	43

2.3.3	Desafios e limitações da paralelização	45
3	Ambiente de programação paralela e o algoritmo proposto	47
3.1	Apresentação do MATLAB®	47
3.1.1	Ambiente PCT (<i>Parallel Computing Toolbox</i>) e MDCS (<i>Distributed Computing Server</i>)	49
3.1.2	Comandos para paralelização	50
3.2	Algoritmo proposto	51
3.3	Condições de avaliação do algoritmo proposto	54
4	Resultados e discussões	56
4.1	Resposta ao degrau unitário	57
4.2	Resposta à rampa unitária	79
4.3	Resposta ao dente de serra	101
5	Conclusões	123
6	Sugestões de Trabalhos Futuros	125
A	Métodos de integração numérica	131
A.1	Método de Euler	131
A.2	Método de Runge-Kutta de quarta ordem	132

Capítulo 1

Introdução

Ao longo dos anos, os estudos de sistemas dinâmicos produziram diversas contribuições à ciência e à tecnologia, dentre elas a análise qualitativa apresentada pelo cientista francês Henri Poincaré (1854-1912), que trouxe técnicas geométricas para a análise de equações diferenciais não-lineares. Nessa abordagem, as conclusões qualitativas sobre a estabilidade dos pontos de equilíbrio, são obtidas a partir de estudos das aproximações assintóticas. Contudo, essas análises das informações quantitativas, sobre o comportamento transiente do sistema, não são observadas pois procura-se trabalhar com a resposta em estado estacionário [1].

Os estudos na área de sistemas dinâmicos surgem da necessidade de compreensão desses fenômenos e das regras internas que orquestram. Ao observá-los, nota-se a inter-relação entre seus componentes apesar de possuírem comportamentos distintos quando analisados isoladamente.

Dentre os sistemas complexos, o estudo de redes de transmissão de sinais de tempo compõem uma necessidade atual e crescente, devido às demandas da sociedade, por exemplo, nas áreas de telecomunicação, sistemas de posicionamento global e controle de processos [1, 2].

Formada pela conexão de um determinado número de relógios osciladores, a rede de distribuição de sinal de tempo possui a finalidade de organizar a coordenação temporal entre eles, denominado sincronismo [1].

O sincronismo consiste na associação de dois ou mais osciladores, na qual, o estado em que se encontram não possuem diferenças entre suas frequências, desse modo as suas diferenças de fases são constantes. Os osciladores que compõem a rede, podem estar conectados tanto por ligações físicas, quanto constituídas por conexões imple-

mentadas via software [1].

Além disso, as redes de distribuição citadas buscam atender principalmente os sistemas comerciais de transmissão de comunicação digital, que com o avanço das tecnologias, necessitam da confiabilidade em sua base de tempo para validação da comunicação [2, 3].

O uso de redes de PLLs está relacionado com a composição dos sinais do Sistema de Posicionamento Global, (GPS, do inglês Global Positioning System) [4].

Seu comportamento reproduz as fases dos componentes I (em fase, do inglês *in-phase*) e Q (quadratura, do inglês *quadrature*) com 90° de defasagem. O sinal resultante possui a amplitude calculada por meio da soma vetorial desses componentes e o ângulo de fase relacionado ao eixo I pode ser determinado a partir da relação trigonométrica entre Q/I [4].

Devido à utilização de PLLs, como efeito, qualquer desalinhamento na fase do oscilador local em relação à fase do sinal amostrado, portanto terá como efeito um ângulo diferente de saída, de modo que tal o mesmo é detectado e corrigido pela malha de controle. De modo usual, o uso dessa rede de PLLs auxilia no processo de triangulação, onde a participação de pelo menos três satélites determinam uma posição no globo terrestre [4].

Dentre as soluções comerciais, é comum a utilização de redes Mestre-Escravo (MS, do inglês Master-Slave) devido a sua facilidade de aplicação, confiabilidade e baixo custo. Nesse tipo de rede, o Nó-Mestre é um relógio extremamente preciso, normalmente do tipo atômico [2].

O sinal gerado pelo mestre é enviado ao primeiro Nó- Escravo, cuja sua fase e frequência extraídas para o sincronismo do mesmo. Tal processo é repetido em todos os nós da rede, de modo que o sinal gerado Nó-Mestre seja propagado por todas as conexões [2].

Para fins que exemplificação, nesse tipo de configuração de rede, a forma como é feita a transmissão e comunicação entre os nós da rede, é denominada como [2, 5]:

- Rede mestre-escravo de via única (OWMS, do inglês *One-Way Master Slave*): quando um nó tem informações de fase e frequência apenas do nó imediatamente anterior e;
- Rede mestre-escravo de via dupla (TWMS, do inglês *Two-Way Master Slave*): quando o nó tem informações tanto do nó imediatamente anterior quanto poste-

rior.

A utilização de circuitos PLL, pode ser aplicada tanto na versão em hardware quanto em software, no entanto para a recuperação da fase e frequência do relógio mestre se faz necessária tipo específico de configuração, pois a sua malha de sincronismo é um sistema de controle em malha fechada, que se ajusta automaticamente a fase e frequência do sinal gerado internamente e localmente à fase e frequência do sinal de entrada [6].

O PLL está presente em todos os nós das redes de distribuição de sinal de tempo, pois seu funcionamento auxilia a rede na correta e eficiente transmissão do sinal de tempo, corrigindo eventuais perturbações. Adicionalmente, os meios de transmissão e o processo de modulação faz com que aconteçam variações instantâneas na fase do sinal de entrada, produzindo assim oscilações no sinal de saída de cada nó. Dessa forma, o efeito faz com que as transições de zero do sinal do relógio tenham sua posição desviada da posição ideal [2, 7, 8].

1.1 Motivação

A análise dos efeitos de sincronismo das diferentes topologias de redes mestre-escravos bidirecionais terá investigação inicial no estudo do funcionamento e parametrização da rede em questão.

Dentre as redes propostas por [9], a análise dos estados de sincronismos é primordial para o entendimento do funcionamento das redes TWMS. Logo, destacam-se a compreensão das topologias de redes TWMS, cujo algoritmo proposto por esse trabalho terá foco.

O estudo das redes mutualmente conectadas, de acordo com as proposições de [2, 9] consideram a observação e comparação entre o sincronismo do nó local em relação ao mestre.

O desenvolvimento do projeto de aplicação e avaliação do sincronismo dos nós será via software MATLAB[®] por meio de seu recurso *Parallel Computing Toolbox*, que é um ambiente de programação que permite a resolução de problemas utilizando os múltiplos processadores, processadores gráficos e *clusters* do computador em que se está realizando a simulação [10].

1.2 Objetivos

Esse trabalho busca trazer contribuições para o estudo de redes de distribuição de sinal de relógio sob a abordagem de paralelismo. Podemos listar os objetivos de desenvolvimento da pesquisa como:

- Estudar as opções de aplicações de paralelismo via MATLAB[®];
- Desenvolver a aplicação de um algoritmo que resolva o sincronismo de PLLs em redes de distribuição de sinal de relógio;
- Aplicar o paralelismo no algoritmo e comparar as respostas com a lógica sequencial e;
- Apresentar direcionamento a assuntos que podem ser estudados com o desenvolvimento da pesquisa.

1.3 Roteiro do trabalho

No Capítulo 1, há a introdução ao assunto, partindo de um ponto histórico, passando pelos diferentes tipos de PLLs, redes de distribuição e chegando à contextualização do problema.

No Capítulo 2, são apresentados os fundamentos e revisão sobre o que é PLL, com base no levantamento de seu modelo matemático, modos de operação e definição de estabilidade e sincronização. Em seguida, serão apresentadas as redes de distribuição, através do modelo matemático da rede TWMS e seu comportamento teórico. Por fim, são introduzidos algoritmos paralelos, listadas as formas de paralelização e os desafios quanto ao seu desenvolvimento.

No Capítulo 3 temos a apresentação do ambiente de programação paralela no MATLAB[®], quais são os métodos possíveis e funções para desenvolvimento e a solução proposta, destacando as particularidades para operação em paralelo.

No Capítulos 4 divulga-se resultados alcançados com o algoritmo proposto do Capítulo 3, para perturbações na fase do sinal do oscilador, sendo elas degrau unitário, rampa e onda do tipo dente de serra respectivamente.

No Capítulo 5 conclui-se quanto ao desenvolvimento do trabalho, a síntese do conhecimento aplicado/construído e comparação entre os objetivos projetados e resultados alcançados.

Por fim, no capítulo 6 são listadas as principais contribuições construídas pela pesquisa e são apresentadas sugestões para trabalhos futuros relacionados ao tema.

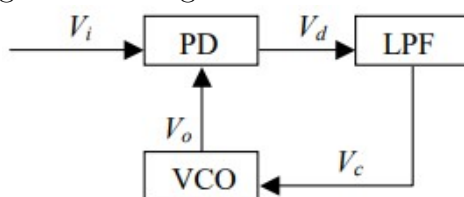
Capítulo 2

Revisão Bibliográfica

2.1 *Phase Locked Loop - PLL*

Na área de estudo de sincronismo das redes de distribuição de sinal de relógio, o estudo de malhas de sincronismo, *phase-locked loop* (PLL), é um sistema de controle em malha fechada usado para ajustar, ou seja, sincronizar automaticamente a fase de um sinal gerado localmente à fase de um sinal periódico de entrada. O PLL é constituído por um circuito eletrônico formado por: detector de fase - *phase detector* (PD), de um filtro passa-baixas - *low pass filter* (LPF) e, de um oscilador controlado por tensão - *voltage-controlled oscillator* (VCO) [2]. Na Figura 2.1 podemos ver a representação por diagrama de blocos de um PLL.

Figura 2.1: Diagrama de blocos de PLL



Fonte: Adaptado de [11].

O PD, é um multiplicador de sinal, que compara a fase $\phi_o(t)$ do sinal gerado pelo VCO, com a fase do sinal de entrada $\phi_i(t)$. Como exemplo, considerando a entrada V_i seja um sinal da forma:

$$V_i = \text{sen}(\phi_i(t) + \omega_o t), \quad (2.1)$$

e a saída do VCO, V_o , seja da forma:

$$V_o = \cos(\phi_o(t) + \omega_o t), \quad (2.2)$$

a saída V_d do PD se torna proporcional a:

$$V_d = \text{sen}(\phi_i(t) - \phi_o(t)) + \text{sen}(2\omega_o t + \phi_i(t) + \phi_o(t)). \quad (2.3)$$

Por se escolher tais características para V_i e V_o , têm-se que ambos possuem a frequência de livre curso em comum (o termo ω_o). Para a sincronização, as diferenças de fase são consideradas nas fases $\phi_o(t)$ e $\phi_i(t)$ [11].

O sinal V_d é aplicado ao filtro passa baixas, *low-pass filter - LPF*, para que a componente $\text{sen}(2\omega_o t + \phi_i(t) + \phi_o(t))$, de alta frequência, seja eliminado.

A saída V_c do LPF alimenta VCO e, devido a isso, modifica a fase do seu sinal de acordo com a equação:

$$\frac{d\phi_o(t)}{dt} = k_o V_c, \quad (2.4)$$

onde k_o é uma constante de proporcionalidade [11].

Devido a essa característica de sua formação, o PLL pode conter filtros de lineares de 1^a ou 2^a ordem, fazendo com que o comportamento dinâmico dessa malha seja descrito por uma equação diferencial não-linear de 2^a ou 3^a ordem. Generalizando, um filtro de ordem n resulta em um PLL de ordem $n+1$ [12].

O estudo de PLLs possui diferentes formas de aplicações, sejam elas analógicos, híbridos ou implementados via software [2]. Todos eles necessitam dos mesmos três elementos básicos: PD, LPF e VCO.

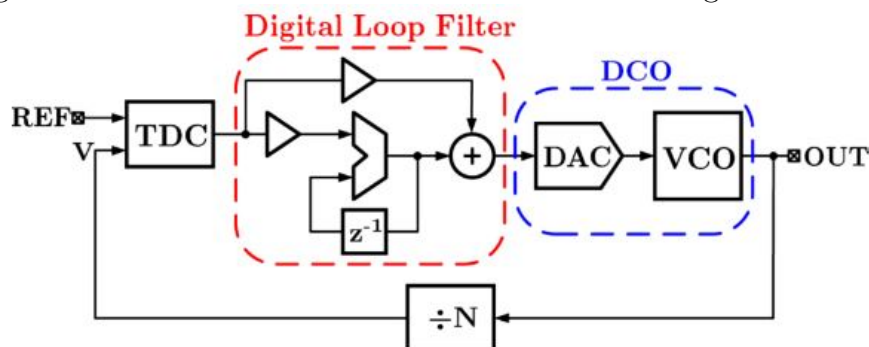
Os PLLs analógicos, como o nome próprio diz, são feitos por componentes eletrônicos analógicos.

Em PLLs híbridos - *Digital PLL*, representado também pela Figura 2.1, onde o detector de fase é substituído por um detector de fase e frequência implementado em um circuito digital capaz de determinar os erros de fase e frequência do sinal de entrada e local, com sua saída conectada a uma bomba de corrente, mantendo a carga para o LPF [2].

Nos PLLs digitais, *All-digital PLL* (ADPLL), conforme apresentado na Figura 2.2, possui algumas adaptações na sua estrutura básica. Nos ADPLL, temos os se-

guintes componentes que se decompõem do PLL apresentado na Figura 2.2, o conversor digital-analógico, *digital-analogic converter* - DAC, cuja a função é fazer o VCO com que as limitações de estabilidade de ajustes sejam superadas, pois quanto melhor o bloco de sintonia, melhor é a resolução de tempo trazida [13].

Figura 2.2: Estrutura convencional de ADPLL em diagrama de blocos



Fonte: Adaptado de [13].

Os fenômenos que surgem devido ao funcionamento do ADPLL, como o meio de transmissão de sinal e modulação provocam variações instantâneas de fase do sinal de entrada causando assim oscilações não desejadas na saída de cada nó. Tal efeito recebe o nome de *jitter* de fase, o que é indesejado e compromete o funcionamento do sistema [2, 5].

Há a possibilidade de implementação da malha PLL por meio de aplicações de software (SPLL), que são algoritmos implementados em dispositivos processadores de sinais, *Digital Signal Process* - DSP, cujo objetivo é instituir o sincronismo dos softwares. Um exemplo disso está nos desktops, em especial os relógios dispostos no canto inferior direito, seu funcionamento está sincronizado com um relógio mestre, um serviço disponibilizado de forma online [2].

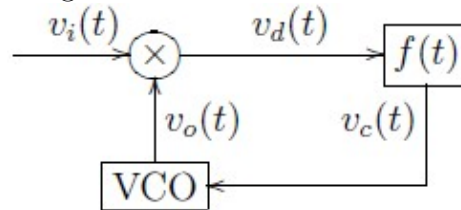
2.1.1 Modelo matemático do PLL analógico

O PLL é um circuito que sincroniza o sinal de saída de um oscilador a um sinal de referência. O estado síncrono é considerado quando a diferença de fase do sinal de saída do oscilador local é nula ou se mantém constante em relação ao de entrada. Por ser um circuito de malha fechada, a sua operação atua reduzindo o erro de fase para quaisquer perturbações realizadas na entrada [2, 14].

No caso do PLL analógico, o detector de fase é um circuito multiplicador, con-

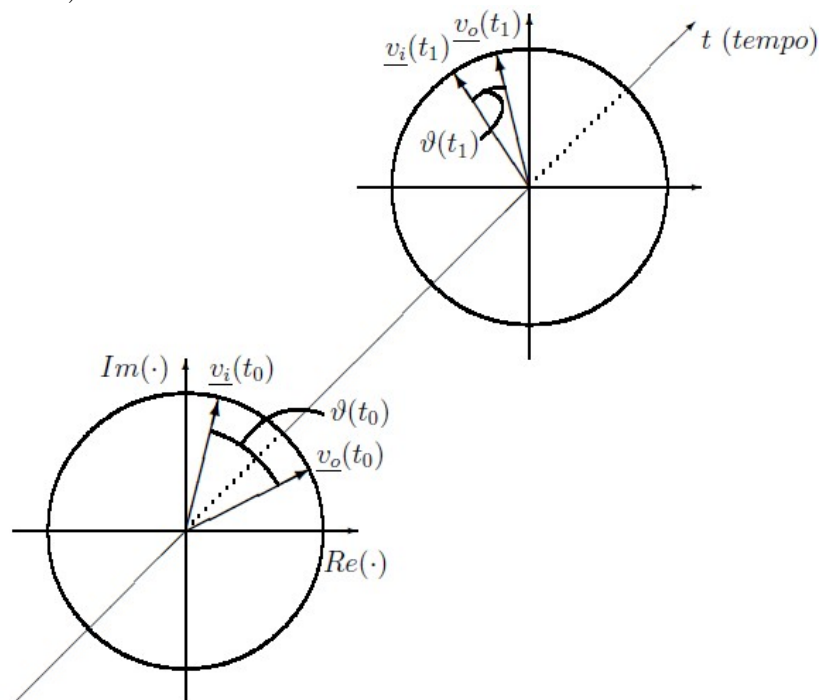
forme apresentado na Figura 2.3. O funcionamento desse circuito pode ser observado por meio do diagrama de fasores, apresentados na 2.4. No instante t_0 o erro de fase é $\vartheta(t_0)$, sendo que, para essa situação, o sinal de referência v_i lidera. A partir do valor $\vartheta(t_0)$, o sistema atua controlando VCO, aumentando sua frequência, objetivando a diminuição do erro de fase. Em um instante t_1 , pode-se notar que o erro de fase $\vartheta(t_1)$ diminuiu. Idealmente, em um instante $t_1 < t_s < \infty$, o erro de fase é nulo, ou seja, a frequência de VCO passa a ser a mesma que a do sinal v_i [2, 14].

Figura 2.3: Diagrama de blocos do PLL analógico clássico



Fonte: Adaptado de [2].

Figura 2.4: Diagrama de fasor descrevendo a operação do PLL (números complexos estão sublinhados)



Fonte: Adaptado de [2].

O modelo matemático desse tipo de PLL é amplamente conhecido e difundido.

O modelo que será apresentado é baseado no desenvolvimento proposto por [2] e [14]. Dada a Figura 2.3, os sinais de entrada e saída do PLL são expressos por:

$$v_i(t) = v_i \text{sen}(\omega_M t + \theta_i(t)), \quad (2.5)$$

$$v_o(t) = v_o \text{cos}(\omega_M t + \theta_o(t)), \quad (2.6)$$

sendo que $\theta_o(t)$ é a estimativa da malha para a fase $\theta_i(t)$, assim como $\dot{\theta}_o(t)$ é a estimativa para a frequência $\dot{\theta}_i(t)$. A frequência de livre curso ω_M é a frequência de operação do VCO quando $v_c(t) = 0$.

De modo a levantar o modelo matemático do PLL, necessita-se estabelecer alguns parâmetros de contorno, conforme apresentado nas definições 1, 2 e 3, listadas a seguir [2]

Definição 1: (Erros de fase e frequência de um PLL): O erro de fase $\vartheta(t)$ é a diferença entre a fase do sinal de entrada $\vartheta_i(t)$ e a fase do sinal de saída $\vartheta_o(t)$ do VCO local; o erro de frequência angular está associado a taxa de variação do erro de fase em relação ao tempo [2]:

$$\vartheta(t) = \theta_i(t) - \theta_o(t) \quad (2.7)$$

$$\dot{\vartheta}(t) = \dot{\theta}_i(t) - \dot{\theta}_o(t) \quad (2.8)$$

Nesse ponto, é importante ressaltar que o PLL analógico apresenta erro de fase estático. Levando em consideração as equações 2.5 e 2.6 e que o erro de fase é nulo, temos que $\theta_i(t) = \theta_o(t)$, concluímos que $\vartheta = 0$, tendo o PLL um comportamento com diferença de fase de $\frac{\pi}{2} \text{rad}$ entre o sinal de entrada e saída [2].

Definição 2: (Função de Transferência $F(s)$ do filtro): pode ser expressa como uma razão de polinômios no domínio de Laplace de acordo com a expressão 2.9 [2]:

$$F(s) = \frac{N(s)}{D(s)} \quad (2.9)$$

com os termos $N(s)$ e $D(s)$ definidos por:

$$N(s) = \sum_{m=0}^M \alpha_m s^m \quad (2.10)$$

$$D(s) = \sum_{p=0}^P \beta_p s^p \quad (2.11)$$

onde, $M \leq P$.

O modelo do PLL necessita da definição dos operadores Q e L e o ganho G .

Observação 1: (Definição dos operadores Q e L): Partindo das equações 2.10 e 2.11, definimos que [2]:

$$Q(x) = \mathcal{L}^{-1}[N(s)X(s)] = \sum_{m=0}^M \alpha_m \frac{d^m}{dt^m} x(t) \quad (2.12)$$

e,

$$L(x) = \mathcal{L}^{-1}[sD(s)X(s)] = \sum_{p=0}^P \beta_p \frac{d^{p+1}}{dt^{p+1}} x(t) \quad (2.13)$$

sendo M e P as ordens dos polinômios $N(s)$ e $D(s)$, com $M \leq P$. Partindo de 2.12 e 2.13 e aplicando diretamente da propriedade linear da transformada de Laplace, chegamos a:

$$Q(ax + by) = aQ(x) + bQ(y) \quad (2.14)$$

e,

$$L(ax + by) = aL(x) + bL(y) \quad (2.15)$$

onde a e b são valores reais.

Definição 3: (Ganho do PLL): O ganho do PLL pode ser definido com a expressão 2.16, apresentada a seguir [2]:

$$G = \frac{1}{2} k_m k_o v_i v_o \quad (2.16)$$

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:

$$v_d(t) = k_m v_i(t) v_o(t) \quad (2.17)$$

Substituindo as equações 2.5, 2.6 em 2.17, temos:

$$v_d(t) = k_m k_o v_i v_o \text{sen}(\omega_M t + \theta_i(t)) \text{cos}(\omega_M t + \theta_i(t)) \quad (2.18)$$

Aplicando-se a identidade trigonométrica, apresentada por [2]:

$$\text{sen}(A)\text{cos}(B) = \frac{1}{2}(\text{sen}(A - B) + \text{sen}(A + B)) \quad (2.19)$$

E substituindo a equação 2.16 em 2.17, obtemos:

$$v_d(t) = \frac{G}{k_o} v_\delta(t) \quad (2.20)$$

com:

$$v_\delta(t) = \text{sen}(\vartheta(t)) + \text{sen}(\omega_M t + \theta_i(t) + \theta_o(t)) \quad (2.21)$$

A saída $\theta_o(t)$ do VCO é por meio da equação:

$$\frac{d}{dt}\theta_o(t) = k_o v_c(t) \quad (2.22)$$

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

$$v_c(t) = f(t) * v_d(t) \quad (2.23)$$

Substituindo a equação 2.23 em 2.22, temos:

$$\frac{d}{dt}\theta_o(t) = k_o f(t) * v_d(t) \quad (2.24)$$

Aplicando-se o Teorema da Convolução [17] e a **Definição 2** na equação 2.24, temos:

$$k_o N(s) V_d(s) = s D(s) \Theta(s) \quad (2.25)$$

Aplicando a Transformada Inversa de Laplace em ambos os lados da expressão 2.25 e substituindo os operadores Q e L da **Definição 3**:

$$k_o Q[v_d(t)] = L[\theta_o(t)] \quad (2.26)$$

Substituindo as equações 2.7 e 2.17 :

$$k_o Q\left[\frac{G}{k_o} v_\delta(t)\right] = L[\theta_i(t) - \vartheta(t)] \quad (2.27)$$

Aplicando-se a **Observação 1**, resulta em:

$$L(\vartheta(t)) + GQ[v_\delta(t)] = L[\theta_i(t)] \quad (2.28)$$

Substituindo a equação 2.21 e aplicando novamente a **Observação 1**:

$$L[\vartheta(t)] + GQ[\text{sen}(\vartheta(t))] = L[\theta_i(t)] - GQ[\text{sen}(2\omega_M t + \theta_i(t) + \theta_o(t))] \quad (2.29)$$

por fim, ao se isolar $\theta_o(t)$ da equação 2.27 e substituindo em 2.29:

$$L[\vartheta(t)] + GQ[\text{sen}(\vartheta(t))] = L[\theta_i(t)] - GQ[\text{sen}(2(\omega_M(t) + \theta_i(t)) + \vartheta(t))] \quad (2.30)$$

A equação 2.30 relaciona o erro de fase $\vartheta(t)$ do PLL a uma fase de entrada θ_i . Os operadores Q e L são dependentes dos coeficientes da função transferência do Filtro, que necessita ser linear e invariante no tempo, conforme a **Definição 2**.

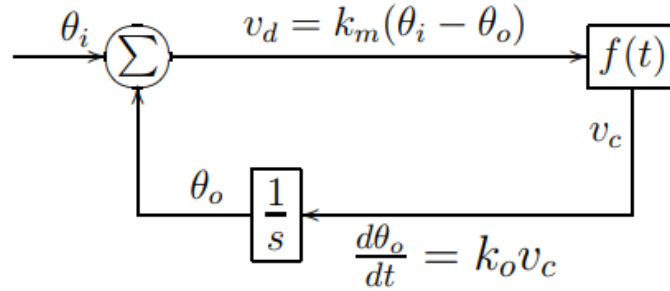
Conforme a equação 2.17, o PD gera uma saída com duas parcelas, uma de baixa frequência e outra com o dobro da frequência de livre curso do VCO (*jitter*) [2, 15]. Projetando o filtro para rejeitar as oscilações indesejadas, principalmente o *jitter* e aproximando $\text{sen}(\vartheta) \approx \vartheta$ (erro de fase pequeno), podemos reescrever 2.30, como:

$$L[\vartheta(t)] + GQ[\vartheta(t)] = L[\theta_i(t)] \quad (2.31)$$

O diagrama de blocos do PLL analógico, linearizado, é apresentado na Figura 2.5, cuja a função transferência é apresentada em 2.32, onde $F(s)$ está associado à dinâmica do filtro.

$$\frac{\Theta_o(s)}{\Theta_i(s)} = \frac{k_m k_o F(s)}{s + k_m k_o F(s)} \quad (2.32)$$

Figura 2.5: Diagrama de blocos do PLL linearizado



Fonte: Adaptado de [2].

2.1.2 Modos de Operação do PLL

Existem dois modos de operação para o PLL, o modo de aquisição e o modo de rastreamento. Resumidamente, enquanto o primeiro está operando dinamicamente, buscando atingir o estado síncrono a partir de uma condição inicial, o segundo modo opera na região síncrona, realizando pequenos ajustes buscando manter esse estado, mesmo com perturbações [2].

Para facilitar o entendimento dos modos de operação do PLL, devemos partir das definições dos conceitos de sincronismo, faixa de captura e faixa de retenção, conforme apresentado a seguir, seguindo o exemplo de [2].

Definição 4 (Função de entrada ou excitação). Considerando a fase de entrada $\theta_i(t)$ do PLL, conforme apresentado na 2.1 uma função no domínio real, $\theta_i: \mathbb{R} \rightarrow \mathbb{R}$ [2].

$$\theta_i(t) = \frac{R}{2}t^2 + \Omega t + \phi \quad (2.33)$$

A definição dos coeficientes R , Ω e ϕ com valores reais, podemos obter isoladamente qualquer uma das entrada desejadas [2]. Essa representação possibilita ao sistema a utilização de três sinais, por exemplo: ao se manter constante os termos R e Ω , temos um degrau de fase ao variar ϕ . Variando Ω e mantendo constante os termos R e ϕ , temos uma rampa de fase. Por fim, ao variarmos o termo R e mantivermos constante os termos Ω e ϕ , temos uma parábola de fase.

Definição 5 (Sincronismo de um PLL analógico). O estado síncrono de um PLL é atingido quando está operando no ponto de equilíbrio, com erro de fase ϑ nulo, na expressão 2.34 [2].

$$L[\vartheta(t)] + GQ[\vartheta(t)] = L[\theta_i(t)] \quad (2.34)$$

Definição 6 (Faixa de retenção de um PLL). A faixa de retenção de um PLL é formada pelo conjunto de valores dos coeficientes R , Ω e ϕ da função de entrada, de modo que a resposta da equação 2.34 tenha um ponto de equilíbrio assintoticamente estável correspondente a um estado síncrono [2].

Definição 7 (Modo de aquisição de um PLL). Um PLL opera em modo de aquisição se opera dentro de sua faixa de captura [2].

Definição 8 (Modo de rastreamento de um PLL). Um PLL opera em modo de rastreamento se opera dentro de sua faixa de retenção [2].

2.1.3 Estabilidade e Sincronização

Definição 9 (Estabilidade no sentido de Lyapunov): Seja x_0 um ponto de equilíbrio da equação de estados autônoma $\dot{x} = X(x)$. Então, no sentido de Lyapunov, x_0 é estável [16], se: dado $\epsilon > 0$ existe $\delta = \delta(\epsilon) > 0$ tal como, para qualquer outra solução $y(t)$ satisfazendo $|x_0 - y(t)| < \delta$, tem-se $|x_0 - y(t)| < \epsilon$ para $t > t_0$, $t_0 \in \mathcal{R}^+$ [2, 16].

Definição 10 (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 [16] e se existe uma constante $b > 0$ tal que se $|x_0 - y(t_0)| < b$ implica no limite de $t \rightarrow \infty$ para $|x_0 - y(t)| = 0$ [2].

Definição 11 (Estado Síncrono): O estado síncrono $\vartheta_s^{(j)} \in R^{P+1}$ é um ponto de equilíbrio assintoticamente estável da equação 2.34 [2, 15].

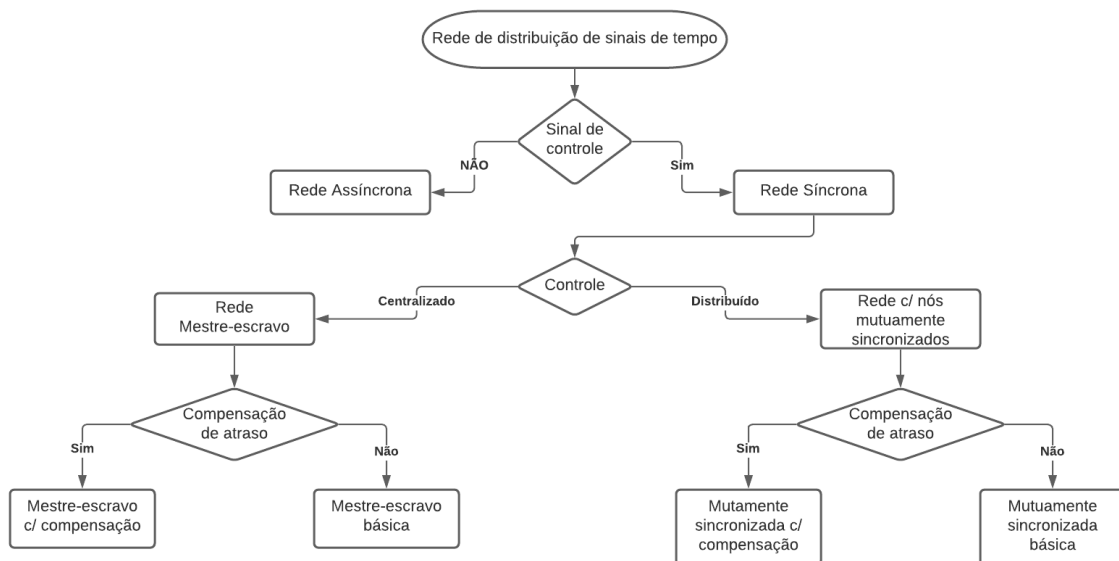
Definição 12 (Sincronização): Seja $\vartheta_s^{(j)}$ um estado síncrono e $\vartheta^{(j)}(t)$ a solução da equação (3.28). Se $\vartheta^{(j)}(t) \subset V_1(\vartheta_s^{(j)}, \epsilon_1)$ e $(d/dt)\vartheta^{(j)}(t) \subset V_2(c, \epsilon_2)$ assegura que para todo $t > t_s$, $t_s < \infty$, com $\epsilon_1, \epsilon_2 > 0$ e $|c| < \infty$, $c \in \mathcal{R}$, então o nó j está síncrono. Se, dentro de um valor constante e limitado, o erro de fase oscilar, é considerado que esse nó escravo está sincronizado, e, para fins de análise, esse nó atingiu o estado síncrono [2, 15].

2.2 Redes de distribuição de sinal de relógio

A distribuição de sinal de relógio em redes síncronas necessita que as frequências e fases sejam comuns a todos os elementos da rede. Dentre as formas de se buscar o sincronismo nesse tipo de aplicação, pode-se classificar em: centralizadas e descentralizadas [2].

Em redes com estratégia de sincronismo centralizadas, caso possuam prioridades de relógios, são denominadas redes Mestre-escravo, do inglês *Master-slave* - (MS). Nas redes em que todos os relógios têm importância na determinação do sincronismo da malha, são chamadas de redes com nós mutuamente conectados [9]. Esses dois tipos de rede podem possuir ou não técnicas para a compensação de atraso, conforme apresentado na Figura 2.6.

Figura 2.6: Estratégia de distribuição de sinais de tempo



Fonte: Adaptado de [9].

Para a modelagem de redes MS em redes de distribuição de bases de tempo, a observância se a rede é via única, do inglês *one way master-slave* - OWMS, e via dupla, do inglês *two way master-slave* - TWMS [2, 9].

Em redes OWMS, o nó mestre possui a sua base de tempo e ela é independente das demais enquanto os nós escravos têm sua base de tempo dependente de um único nó, podendo ser ele do mestre ou de outro escravo. Nesses tipos de rede, podem ser

classificadas topologicamente como redes cadeia e redes estrela [9].

Nas redes TWMS, apesar do nó mestre possuir sua base de tempo própria, ela também é ajustável aos outros nós. Nos nós escravos, a sua base de tempo pode ser dependente de mais de um nó. Topologicamente classificas em redes em cadeia, em estrela ou em enlace [9].

2.2.1 OWMS - Cadeia Simples

A Figura 2.7 apresenta o exemplo desse tipo de rede, onde o nó 1 pertence a posição do mestre, fornecendo base de tempo para o nó 2, e assim sucessivamente.

Figura 2.7: Rede OWMS de cadeia simples

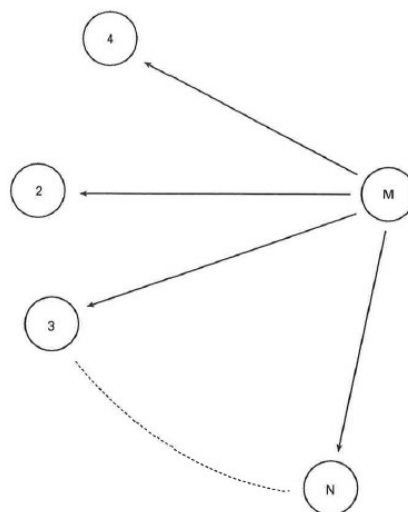


Fonte: Adaptado de [9].

2.2.2 OWMS - Estrela simples

Conforme a estrutura apresentada na Figura 2.8, o nó mestre fornece a base de tempo para todos os nós dessa rede.

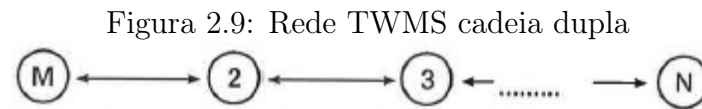
Figura 2.8: Rede OWMS estrela simples



Fonte: Adaptado de [9].

2.2.3 TWMS - Cadeia dupla

Conforme a estrutura apresentada na Figura 2.9, o mestre, representado pelo nó 1, estabelece sua base de tempo ponderando sua própria fase e a fase do nó 2. Ao estabelecer sua base de tempo, cada nó escravo levará em conta o nó vizinho, que lhe serve como mestre e o que lhe serve como escravo.

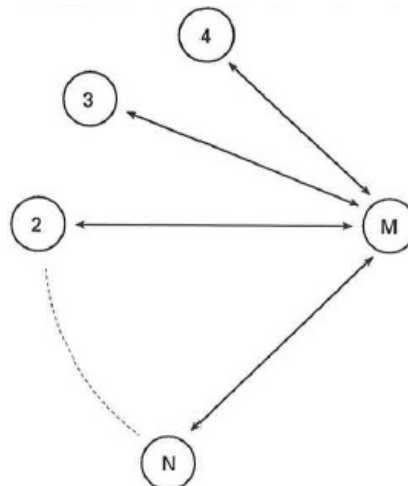


Fonte: Adaptado de [9].

2.2.4 TWMS - Estrela dupla

Conforme apresentado na Figura 2.10, o nó mestre fornece a base de tempo para todos os escravos dessa rede. A base do nó mestre será resultado da ponderação de sua própria fase e a fase de todos os outros nós.

Figura 2.10: Rede TWMS estrela dupla

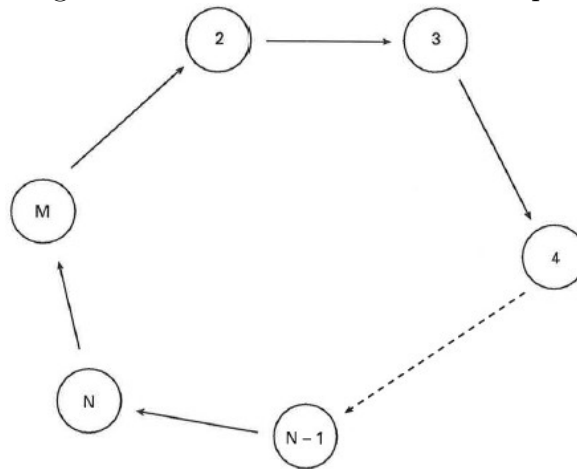


Fonte: Adaptado de [9].

2.2.5 TWMS - Enlace simples

A Figura 2.11 apresenta o esquemático dessa rede, onde o nó n-1 funcionando como mestre para o nó i. A base do nó mestre será produto da ponderação entre sua própria fase e da fase do n-ésimo nó.

Figura 2.11: Rede TWMS enlace simples

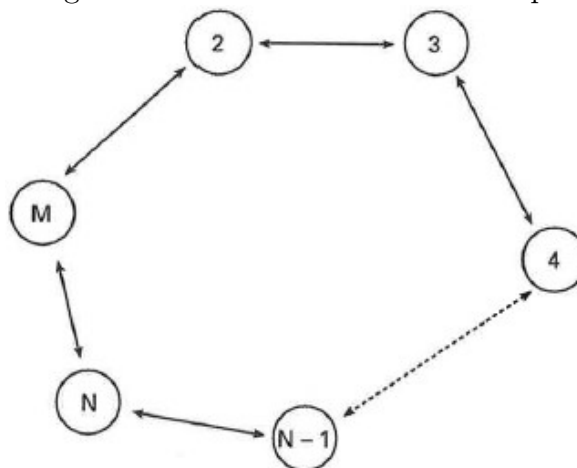


Fonte: Adaptado de [9].

2.2.6 TWMS - Enlace duplo

Por fim, a Figura 2.12 apresenta o esquemático dessa rede, onde o nó $n-1$ funcionando como mestre para o nó i . Todavia, ao estabelecer a sua base de tempo, cada nó escravo considera os nós mestres e escravos que lhe servem.

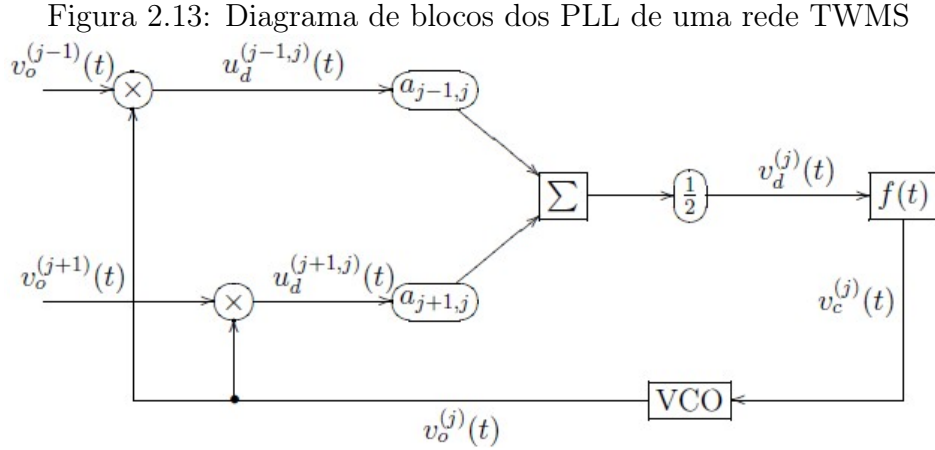
Figura 2.12: Rede TWMS enlace duplo



Fonte: Adaptado de [9].

2.2.7 Modelagem da rede TWMS

A Figura 2.13 apresenta o diagrama de blocos de uma rede TWMS, onde cada nó compara as fases dos nós anterior e posterior com sua própria fase, fazendo com que haja a necessidade de um comparador de fase a mais .



Fonte: Adaptado de [2].

Para levantar o modelo dessa rede, considera-se que o relógio mestre gera uma base de tempo de forma estável e precisa, tendo sua saída representada conforme a equação 2.35.

$$v_o^{(1)}(t) = v_o^{(1)} \cos(\omega_M t + \theta_o^{(1)}(t)) \quad (2.35)$$

onde, a representação ⁽¹⁾ indica que esse é o sinal do nó mestre.

Na rede TMWS o atraso gerado pelos seus nós está associado à média ponderada dos sinais de entrada e nesse caso, considera-se que as diferenças de fases produzidas por cada nó-escravo j fazem parte da estimativa da fase $\theta_o^{(j)}$ [2].

Definição 11 (Sinais de entrada e saída para nós-escravos): Em sincronismo, os sinais de entrada e saída de uma rede TWMS são definidos pelas equações 2.36 e 2.37.

$$v_o^{(l)}(t) = v_i^{(l)} \sin(\omega_M t + \theta_o^{(l)}(t)) \quad (2.36)$$

$$v_o^{(l)}(t) = v_o^{(l)} \cos(\omega_M t + \theta_o^{(l)}(t)) \quad (2.37)$$

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

Considera-se que todos os nós possuem a mesma frequência de livre curso, ω_M , todavia, por possuir duas, é necessário avaliar os erros de fase e frequência por possuir duas entradas.

Definição 12 (Erros de fase e frequência de um nó-escravo): Para o erro de fase $\vartheta^{(l,j)}(t)$ de um nó-escravo, devemos 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 $l = j - 1, j + 1$. Nesse caso, o erro de frequência é composta pela derivada do erro de fase com relação ao tempo, conforme apresentado nas expressões 2.38 e 2.39, respectivamente.

$$\vartheta^{(l,j)}(t) = \theta_o^{(l)}(t) - \theta_o^{(j)}(t) \quad (2.38)$$

$$\dot{\vartheta}^{(l,j)}(t) = \dot{\theta}_o^{(l)}(t) - \dot{\theta}_o^{(j)}(t) \quad (2.39)$$

Para o modelo de rede TWMS, a **Definição 2** e a **Observação 1**, apresentadas anteriormente representam os operadores $Q^{(j)}$ e $L^{(j)}$.

Definição 13 (Ganho associado a cada entrada): O ganho associado a um nó de uma rede TWMS pode ser definido conforme a equação 2.40.

$$G^{(l,j)} = \frac{1}{2} a_{l,j} k_m^{(j)} k_o^{(j)} v_o^{(j)} v_o^{(l)} \quad (2.40)$$

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 apresentado na Figura 2.13, o detector de fase é constituído por um multiplicador e pelo somatório. Isso ocasiona a comparação dos sinais de entrada com a fase do VCO local, e o somador 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. Generalizando, temos a expressão 2.41.

$$v_d^{(j)}(t) = \frac{1}{2} \sum_{l=j-1, l \neq j}^{j+1} a_{l,j} u_d^{(l,j)}(t) \quad (2.41)$$

com

$$\frac{1}{2} \sum_{l=j-1, l \neq j}^{j+1} a_{l,j} = 1 \quad (2.42)$$

e

$$u_d^{(l,j)}(t) = k_m^{(l)} v_o^{(l)}(t) v_o^{(j)}(t) \quad (2.43)$$

Substituindo as equações 2.36 e 2.37, apresentadas na **Definição 11** na equação 2.43, encontramos:

$$u_d^{(l,j)}(t) = \frac{G^{(l,j)}}{a_{l,j} k_o^{(j)}} \text{sen}(\omega_m t + \theta_o^{(l)}(t)) \text{cos}(\omega_M t + \theta_o^{(j)}(t)) \quad (2.44)$$

Aplicando a identidade trigonométrica apresentada por [2] e a **Definição 12** na equação 2.44, encontramos:

$$u_d^{(l,j)}(t) = \frac{G^{(l,j)}}{a_{l,j} k_o^{(j)}}(t) \quad (2.45)$$

com,

$$u_\delta^{(l,j)}(t) = \text{sen}(\vartheta^{(l,j)}(t)) + \text{sen}(2\omega_M t + \theta_o^{(l)}(t) + \theta_o^{(j)}(t)) \quad (2.46)$$

A fase de saída $\theta_o^{(j)}(t)$ do nó escravo j é determinada através da lei de controle é apresentada a seguir, pela equação 2.47.

$$\frac{d}{dt} \theta_o^{(j)}(t) = k_o^{(j)} v_c^{(j)}(t) \quad (2.47)$$

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

$$v_c^{(j)}(t) = f^{(j)}(t) * v_d^{(j)}(t) \quad (2.48)$$

Substituindo a expressão 2.48 em 2.47, temos:

$$\frac{d}{dt} \theta_o^{(j)}(t) = k_o^{(j)} f^{(j)}(t) * v_d^{(j)}(t) \quad (2.49)$$

Aplicando o teorema da convolução, a **Definição 2** na expressão 2.49, obtemos:

$$k_o^{(j)} N^{(j)}(s) V_d^{(j)}(s) = s D^{(j)}(s) \Theta_o^{(j)}(s) \quad (2.50)$$

Partindo da equação 2.50 e aplicando a transformada inversa de Laplace em ambos os lados da expressão em conjunto com os operadores $Q^{(j)}$ e $L^{(j)}$ da **Observação 1**, encontramos:

$$k_o^{(j)} Q^{(j)} [v_d^{(j)}(t)] = L^{(j)} [\theta_o^{(j)}] \quad (2.51)$$

Partindo da equação do erro de fase, apresentado na equação 2.38, temos:

$$\vartheta^{(j-1,j)}(t) = \theta_o^{(j-1)}(t) - \theta_o^{(j)}(t) \quad (2.52)$$

e,

$$\vartheta^{(j+1,j)}(t) = \theta_o^{(j+1)}(t) - \theta_o^{(j)}(t) \quad (2.53)$$

Ao somarmos as equações 2.52 e 2.53 e isolando $\theta_o^{(j)}(t)$, chegamos a:

$$\theta_o^{(j)}(t) = \frac{1}{2} \sum_{l=j-1, l \neq j}^{j+1} \theta_o^{(l)}(t) - \frac{1}{2} \sum_{l=j-1, l \neq j}^{j+1} \vartheta^{(j,l)}(t) \quad (2.54)$$

Substituindo a equação 2.54 na equação 2.51 e considerando a **Observação 1**, obtemos:

$$L^{(j)} \left[\frac{1}{2} \sum_{l=j-1, l \neq j}^{j+1} \vartheta^{(l,j)}(t) \right] + k_o^{(j)} Q^{(j)} [v_d^{(j)}(t)] = L^{(j)} \left[\frac{1}{2} \sum_{l=j-1, l \neq j}^{j+1} \theta_o^{(j)}(t) \right] \quad (2.55)$$

Substituindo as equações 2.41 e 2.45 na equação 2.55, podemos escrever:

$$L^{(j)} \left[\sum_{l=j-1, l \neq j}^{j+1} \vartheta^{(l,j)}(t) \right] + Q^{(j)} \left[\sum_{l=j-1, l \neq j}^{j+1} G^{(l,j)} u_\delta^{(l,j)}(t) \right] = L^{(j)} \left[\sum_{l=j-1, l \neq j}^{j+1} \theta_o^{(j)}(t) \right] \quad (2.56)$$

e substituindo em 2.56 a equação 2.46, temos:

$$\begin{aligned}
& L^{(j)} \left[\sum_{l=j-1, l \neq j}^{j+1} \vartheta^{(l,j)}(t) \right] + Q^{(j)} \left[\sum_{l=j-1, l \neq j}^{j+1} G^{(l,j)} \text{sen}(\vartheta^{(l,j)}(t)) \right] = \\
& - Q^{(j)} \left[\sum_{l=j-1, l \neq j}^{j+1} G^{(l,j)} \text{sen}(2\omega_M t + \theta_o^{(l)}(t) + \theta_o^{(j)}(t)) \right] \\
& + L^{(j)} \left[\sum_{l=j-1, l \neq j}^{j+1} \theta_o^{(l)}(t) \right]
\end{aligned} \tag{2.57}$$

Por fim, isolando o termo $\theta_o^{(j)}$ na equação 2.38 e substituindo em 2.57, encontramos:

$$\begin{aligned}
& L^{(j)} \left[\sum_{l=j-1, l \neq j}^{j+1} \vartheta^{(l,j)}(t) \right] + Q^{(j)} \left[\sum_{l=j-1, l \neq j}^{j+1} G^{(l,j)} \text{sen}(\vartheta^{(l,j)}(t)) \right] = \\
& - Q^{(j)} \left[\sum_{l=j-1, l \neq j}^{j+1} G^{(l,j)} \text{sen}(2(\omega_M t + \theta_o^{(l)}(t)) - \vartheta_o^{(l,j)}(t)) \right] \\
& + L^{(j)} \left[\sum_{l=j-1, l \neq j}^{j+1} \theta_o^{(l)}(t) \right]
\end{aligned} \tag{2.58}$$

Aplicando simplificações, conforme proposto por [2], obtemos as expressões:

$$\begin{aligned}
& L^{(j)} \left[\sum_{l=j-1, l \neq j}^{j+1} \vartheta^{(l,j)}(t) \right] + Q^{(j)} \left[\sum_{l=j-1, l \neq j}^{j+1} G^{(l,j)} \text{sen}(\vartheta^{(l,j)}(t)) \right] = \\
& - Q^{(j)} \left[\sum_{l=j-1, l \neq j}^{j+1} G^{(l,j)} \text{sen}(2(\omega_M t + \theta_o^{(l)}(t)) - \vartheta_o^{(l,j)}(t)) \right] \\
& + L^{(j)} \left[\sum_{l=j-1, l \neq j}^{j+1} \theta_o^{(l)}(t) \right]
\end{aligned} \tag{2.59}$$

$$L^{(j)} \left[\sum_{l=j-1, l \neq j}^{j+1} \vartheta^{(l,j)}(t) \right] + Q^{(j)} \left[\sum_{l=j-1, l \neq j}^{j+1} G^{(l,j)} \text{sen}(\vartheta^{(l,j)}(t)) \right] = L^{(j)} \left[\sum_{l=j-1, l \neq j}^{j+1} \theta_o^{(l)}(t) \right] \tag{2.60}$$

$$L^{(j)} \left[\sum_{l=j-1, l \neq j}^{j+1} \vartheta^{(l,j)}(t) \right] + Q^{(j)} \left[\sum_{l=j-1, l \neq j}^{j+1} G^{(l,j)} (\vartheta^{(l,j)}(t)) \right] = L^{(j)} \left[\sum_{l=j-1, l \neq j}^{j+1} \theta_o^{(l)}(t) \right] \tag{2.61}$$

Na equação 2.59 é desconsiderado o atraso de transmissão. Na equação 2.60, é desconsiderado o termo de de frequência dupla e, por fim, na equação 2.61 é linearizado os termos que contém a função $sen(\vartheta^{(l,j)}(t))$, sendo essa a representação de cada nó-escravo de uma rede TWMS ($j = 2, 3, \dots, N$), considerando os atrasos de transmissão de cada nó e o termo de frequência dupla [2].

2.3 Algoritmos paralelos

A evolução da computação e o desenvolvimento de algoritmos na lógica sequencial trouxeram diversos benefícios e capacidades para a sociedade. Todavia, estudos recentes, demandas e acesso a unidades processadoras, a utilização de dispositivos que operam paralelamente, ou seja, capazes de realizar operações simultâneas em diferentes núcleos, estão sendo aplicados para diversos problemas [18, 19].

Para o desenvolvimento de algoritmos que seguem a lógica paralela de processamento, são necessários esforços para que a solução atenda os seguintes passos [18], sendo exemplificados por:

1. Descrever a classificação do problema a ser solucionado;
2. Desenvolver o programa utilizando linguagens/aplicações que suportem a lógica paralela;
3. Compilar e executar o algoritmo construído em uma plataforma que tenha suporte para a computação paralela e resolver o problema.

A performance do algoritmo está diretamente relacionada ao seu desenvolvimento em si e também, com as características de hardware do computador que se está utilizando para execução dentre outros requisitos de desempenho da operação [18].

Dentre as dificuldades para aplicações nesse modelo, têm-se o surgimento de novas classes de erros de software, falhas de comunicação e sincronização entre as unidades processadoras de sub-tarefas [20].

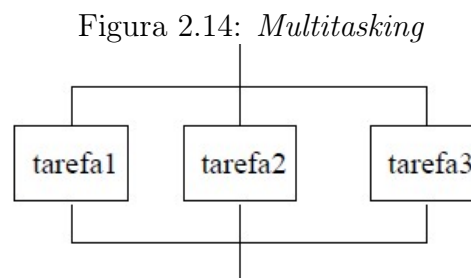
O desenvolvimento de softwares paralelos é mais difícil do que a mesma aplicação para a lógica sequencial, devido a necessidade de atendimento de fatores como a sincronização de tarefas e análise da dependência de dados. Quando utilizado um sistema paralelizado, tais dificuldades são diminuídas e, adicionalmente, permitem o reaproveitamento de programas sequenciais já desenvolvidos [21].

2.3.1 Macrotarefas e Microtarefas

A paralelização de um programa pode ser realizada ao se particionar o programa original em múltiplos subprogramas, caracterizando essa paralelização como *multitasking* ou *macrotasking*. Quando essa paralelização alcança um nível maior, segmentando em diversas rotinas o programa original, temos a abordagem técnica denominada *microtasking* pois a execução do trabalho está em microtarefas. Algumas linguagens de programação possuem tais aplicações ao se desenvolver o programa nos laços e blocos paralelos oferecidos [21].

Multitasking

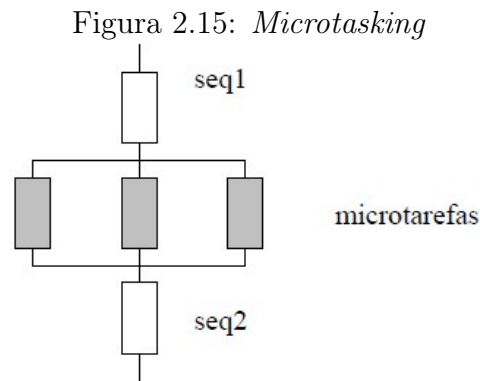
Conforme ilustrado pela Figura 2.14, a paralelização pode ser realizada particionando o programa em múltiplos subprogramas, sendo referenciada como *multitasking* [21, 22, 23]



Fonte: Adaptado de [21].

Microtasking

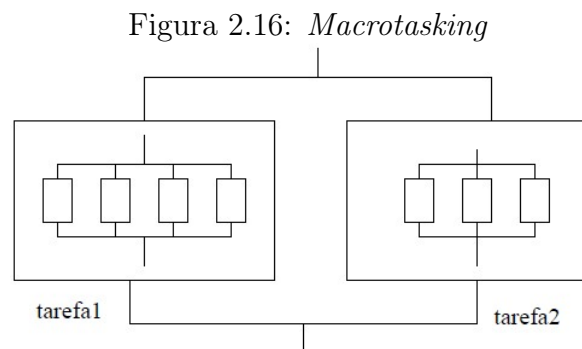
Quando o programa é segmentado em níveis mais finos, conforme o modelo da técnica *microtasking*, o trabalho é executado em microtarefas. Essas tarefas são porções de códigos sequenciais, contidas em um laço, cujas as iterações são executadas paralelamente, ou em bloco de comandos, que é executado paralelamente a outros blocos [21, 24]. A Figura 2.15 exemplifica esse modelo.



Fonte: Adaptado de [21].

Macrotasking

Juntando os recursos apresentados por *multitasking* e *microtasking*, temos cada uma das tarefas particionadas em múltiplas micro tarefas. Esse modelo, conforme a Figura 2.16, é conhecido como *macrotasking* e as tarefas como macrotarefas [21].



Fonte: Adaptado de [21].

2.3.2 Desempenho e métricas

A performance de um software tipicamente sequencial pode ser caracterizada nos termos de taxa de execução de instruções em um processador e o tempo total gasto pelo software em taxa unitária. Para avaliar um software construído com tarefas em paralelo, temos a abordagem em dois indicadores principais: *speedup* e eficiência [25].

Speedup

A definição da métrica *Speedup* é, para cada processador da ordem n , como a taxa do tempo decorrido ao se executar um programa em um único processador em relação ao tempo de execução quando se utiliza n processadores, da forma [21, 25, 26]:

$$S(n) = \frac{T_1}{T_n}. \quad (2.62)$$

Eficiência

A eficiência é definida como a média entre a razão do *Speedup* obtido com a execução de n processadores e o número de processadores n utilizados. Essa medida explora o quanto do paralelismo foi explorado pelo algoritmo. Quanto maior for a fração sequencial, menor será a eficiência, conforme [21, 25]:

$$E = \frac{S}{n}. \quad (2.63)$$

Lei de Amdahl's

A lei de Amdahl foi postulada pelo arquiteto operacional Gene Amdahl e serve para encontrar a melhoria máxima para um software quando uma parte do código é paralelizada [27].

Essa lei é baseada no *speedup*, apresentado pela relação entre a parte paralelizada do algoritmo sob sua implementação sequencial, considerando que o tamanho do problema se mantém mesmo após a paralelização local. Em computação paralela, a sua aplicação é utilizada para prever o máximo teórico de *speedup* quando se utiliza múltiplos núcleos de processamento para executar uma aplicação [21, 27, 28]. A equação 2.64 apresenta a formulação matemática da lei de Amdahl.

$$S = \frac{1}{(1 - F) + \frac{F}{N}} \quad (2.64)$$

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.

Partindo da equação 2.64, deduzimos que existem possíveis limitações para o ganho de desempenho para os trabalhos paralelizados. Considerando o efeito quando N tende ao infinito e F tende a zero, temos nessa situação que a capacidade de processamento infinitamente paralela está disponível, não tendo efeitos de sobrecarga de paralelização. Tal análise apresenta o limite superior teórico para o aumento de desempenho disponível quando for aplicada a paralelização. Ao passo que N se torna infinitamente grande, a expressão $(1 - S)/N$ torna-se tão pequeno quanto, de modo que a equação 2.64 se reduz a $1/S$ [29].

Quando analisado utilizando os mesmos valores propostos por [27], com a porção inerente serial igual a 40 por cento da carga total, temos que o máximo *speedup*, independente do número de núcleos do processador aplicado para a solução do problema seria $(1/.4)$, ou 2,5x. Conforme concluído por Amdahl, o esforço realizado para se alcançar altas taxas de processamento não é aproveitado a não ser que se a taxa de processamento sequencial também seja aumentada.

2.3.3 Desafios e limitações da paralelização

No primeiro momento, ao reconhecer que a sobrecarga de um determinado número de *threads*¹ é uma quantidade indefinida, e que seu valor é um reflexo da eficiência com que a taxa de paralelização que foi aplicada a um determinado pedaço de código. Uma vez que a paralelização foi introduzida, o processo de ajuste deve identificar os gargalos que representam a sobrecarga das *threads*. Essas sobrecargas se enquadram nas seguintes categorias [29]:

- **Criação/Destruição de *threads*:** dentre as técnicas comuns de mitigação incluem o agrupamento de encadeamentos (*threads* reutilizáveis), de modo a diminuir a necessidade de criar e remover *threads* de trabalho, bem como encadeamento de laços externos para diminuir o número de carga pesadas para as *threads* individuais;
- **Sincronização e gerenciamento de bloqueios:** As *threads* do trabalho bloqueiam seções críticas do código para protegerem os dados que estão sendo utilizados de serem substituídos por outras *threads*, de modo a evitar resultados inesperados. Esse bloqueio força as *threads* a aguardarem as demais, e, conse-

¹Thread é um conjunto de tarefas existentes no programa, executadas ao mesmo tempo pelo processador.

quentemente, aumenta os requisitos de processamento sequencial, e por isso tais bloqueios devem ser utilizados de forma criteriosa;

- **Desequilíbrio de carga:** Caso extremo em que um problema gera dois *threads*. Enquanto a primeira executa a computação pesada, a outra escreve somente o resultado do cálculo na tela. A segunda *thread* permanece ociosa a maior parte do tempo, diminuindo a eficiência geral da solução.

Capítulo 3

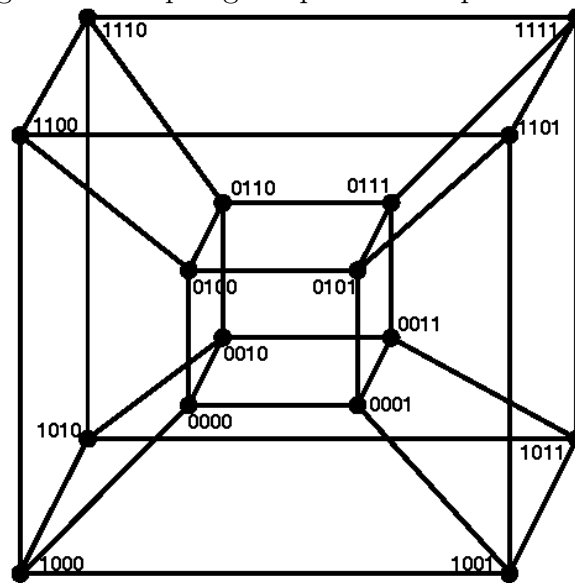
Ambiente de programação paralela e o algoritmo proposto

3.1 Apresentação do MATLAB[®]

O software MATLAB[®] é um ambiente utilizado em diferentes campos da computação técnica e científica. Esse software possui diversas rotinas especializadas para campos como biologia computacional, controle de sistemas, engenharia, finanças, *machine learning* e modelagem. Os pacotes de rotinas, ou como são chamadas, *toolboxes*, simplificam o desenvolvimento das interfaces pois são desenvolvidas tanto pela MathWorks[®], quanto por sua comunidade por meio da participação nos fóruns oficiais [30, 31].

A abordagem de Cleve Moler [32] em 1995 foi a primeira tentativa para o desenvolvimento de uma linguagem de programação paralela no MATLAB[®]. No documento técnico produzido pelo autor, um dos atributos que destacados é sobre o uso compartilhado ou distribuído de memórias. Moler buscou desenvolver o paralelismo em um iPSC (*Intel Personal SuperComputer*), que era um supercomputador de vários processadores, organizados em um arranjo denominado hipercubo, com 32 a 128 nós dispostos, contendo em cada nó um processador 80286 e 512Kb de memória RAM, tendo assim a possibilidade de se executar um programa diferente em cada nó. A figura 3.1 apresenta a topologia utilizada por Moler em seus estudos.

Figura 3.1: Topologia hipercubo de processadores



Fonte: Adaptado de [32].

Dentre os problemas enfrentados nos períodos após Moler, podemos destacar os seguintes [33]:

1. Em pequenas aplicações o tempo de execução é pouco atraente;
2. Falta de recursos e ferramentas computacionais;
3. Disponibilidade de processadores com apenas um núcleo até 2005.

Hoje, com o desenvolvimento de processadores multinúcleos e com a melhoria nas *toolboxes*, o MATLAB® possui duas abordagens para o desenvolvimento do paralelismo [33]:

- Desenvolver o código explícito de forma a executar a comunicação entre processadores na plataforma HPC (*High Performance Computing*), como o MATLAB®MPI e bcMPI,
- Desenvolver o código implícito, como pMatlab, StarP e a utilização da *toolbox* PCT (*Parallel Computing Toolbox*).

Ao se utilizar o ambiente de computação paralela do MATLAB®, há diversas facilidades que podem ser aproveitadas, como o ambiente de programação iterativo,

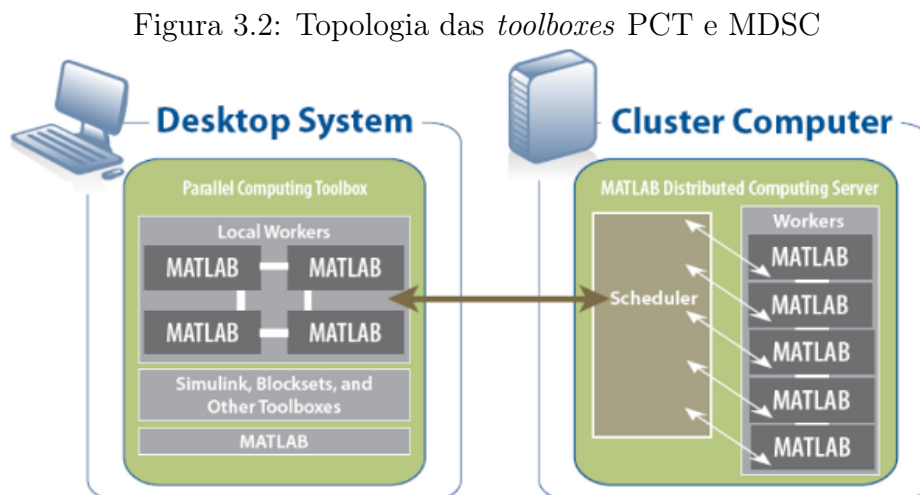
vetorização de tarefas e a utilização do suporte de computação paralela por meio das matrizes distribuídas. Dentre as vantagens desse ambiente, podemos destacar: a otimização do tempo de simulação ao se distribuir tarefas garantindo a execução simultânea, resolução de problemas de *big data* e simulação de propagação de sinais em eletrônica e telecomunicações [34, 35].

3.1.1 Ambiente PCT (*Parallel Computing Toolbox*) e MDCS (*Distributed Computing Server*)

A *toolbox* PCT apresenta funções para a execução paralela de laços de repetições, criação e manipulação de arranjos distribuídos e passagem de mensagens para o desenvolvimento de algoritmos paralelos mais refinados [34, 35, 36].

O MDCS oferece soluções com capacidade de dimensionar algoritmos paralelos para *cluster*¹ maiores. Ele é constituído de processos do MATLAB® *Worker*, que são executados em um *cluster* responsável pela execução do código paralelo e controle dos processos [34, 35, 36].

A figura 3.2 ilustra as arquiteturas do PCT e MDSC.



Fonte: Adaptado de [34].

O PCT permite que o trabalho com compartilhamento de memória e arquitetura de memória distribuída. O desenvolvimento da infraestrutura local é feita de um

¹Cluster no contexto deste trabalho, se refere a uma arquitetura de rede capaz combinar vários computadores para trabalharem em conjunto numa determinada tarefa.

computador com vários núcleos. De modo geral, o PCT que seja utilizado todos os núcleos da máquina, até o máximo de oito *workers* e caso seja necessário uma maior escalabilidade na solução paralela, a *toolbox* MDCS é recomendada para a aplicação [34, 37].

3.1.2 Comandos para paralelização

Dentro do ambiente de desenvolvimento PCT, os seguintes comandos podem ser utilizados para a paralelização [38, 36]:

- *parfor*, cujo a aplicação é para a execução de laço de repetição, do tipo *for* em paralelo;
- *spmd*, define uma instrução que é encaminhada a diferentes núcleos durante a paralelização;
- *task*, que cria vários programas independentes.

Para o desenvolvimento desse trabalho, utilizou-se o recurso *parfor* pois seu uso é bastante útil em situações que requerem várias iterações em um laço de repetição. Como exemplo de utilização, destaca-se o uso em simulações de Monte Carlo e no processamento de imagens. Para a utilização junto ao PCT, o código original, na estrutura sequencial é adaptado para a utilização do comando *parfor* [39].

Como o PCT levanta junto ao sistema operacional o número de núcleos que poderá ser utilizada, a cada iteração executada no *parfor* é independente pois são executadas em instâncias diferentes do MATLAB[®]. Dessa forma, o conjunto de instruções contidos dentro do *parfor* não pode depender de resultados de outras iterações. Essa limitação é devido a construção da função *parfor* dentro do PCT, que não compartilha a memória da iteração com o ambiente externo a ela. A aplicação desta solução é viável para diferentes computadores conectados via rede, pois o PCT é capaz de transformar cada núcleo desta rede em uma unidade de processamento de iterações do *parfor* [37, 36, 39].

Caso, durante a construção do código, o número de *workers*² for igual ao número de iterações do laço *parfor*. Caso haja mais iterações do que *workers*, alguns irão executar mais de uma iteração.

²Workers no contexto desse trabalho quer dizer instâncias do MATLAB[®] e está relacionado à quantidade de núcleos disponíveis

3.2 Algoritmo proposto

Para o desenvolvimento do trabalho foram seguidas as etapas de cálculo necessárias para a execução do PLL. Assim, no primeiro momento, foi realizada a comparação de fase do sinal de entrada e do sinal de saída do VCO do próprio PLL. Este sinal é chamado de VD e alimenta o filtro LPF do PLL, no qual é utilizado o método integrativo Runge-Kutta de quarta ordem, cuja implementação foi baseada em [40, 41] e resumido no Apêndice A. Por fim, o sinal de saída do LPF, denominado VC, é direcionado ao VCO, fechando a malha. O algoritmo proposto para executar as simulações tem três seções principais:

1. Parâmetros iniciais e de simulação – neste região são respondidas as seguintes questões: Qual é o sinal de entrada; Qual é o tamanho da rede; Qual é o tempo inicial t_0 , tempo final t_f e taxa de amostragem h da simulação; Quais são os parâmetros do filtro;
2. Simulação de rede TWMS – são realizados os cálculos referentes à rede TWMS construída, em que o cálculo da saída LPF é realizado em paralelo;
3. Construção e apresentação dos resultados obtidos – criação de um arquivo de saída com os resultados da simulação e gráficos do comportamento do sistema.

Para o desenvolvimento deste trabalho, a aplicação da *toolbox* TPC foi feita juntamente com a aplicação da função *parfor*. O trabalho, por possuir uma estrutura iterativa que não é dependente de resultados encontrados por outras iterações, se torna viável a aplicação do paralelismo no cálculo dos valores da saída V_c e θ_o quando se calcula a integral pelo método de Runge-Kutta. Devido a isso, a paralelização dos PLLs está na solução do sinal de controle para o VCO e para a rede TWMS, nessa proposição, é calculada simultaneamente.

O pseudocódigo da solução proposta é apresentada a seguir.

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**

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

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

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

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

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

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

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

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

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

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

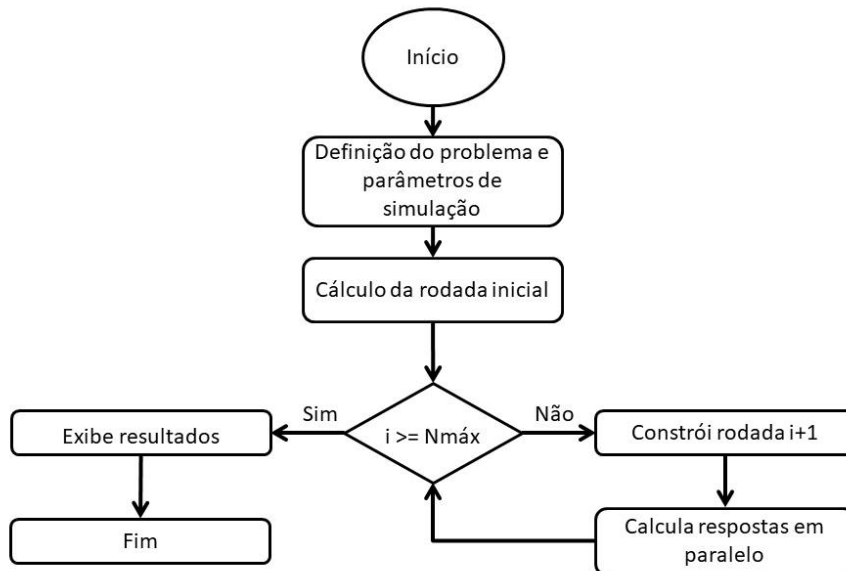
fim para

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

fim para

De maneira ilustrativa, podemos segmentar o pseudocódigo da solução proposta em blocos principais, conforme apresentado no fluxograma da 3.3.

Figura 3.3: Fluxograma do algoritmo proposto



Fonte: Autor.

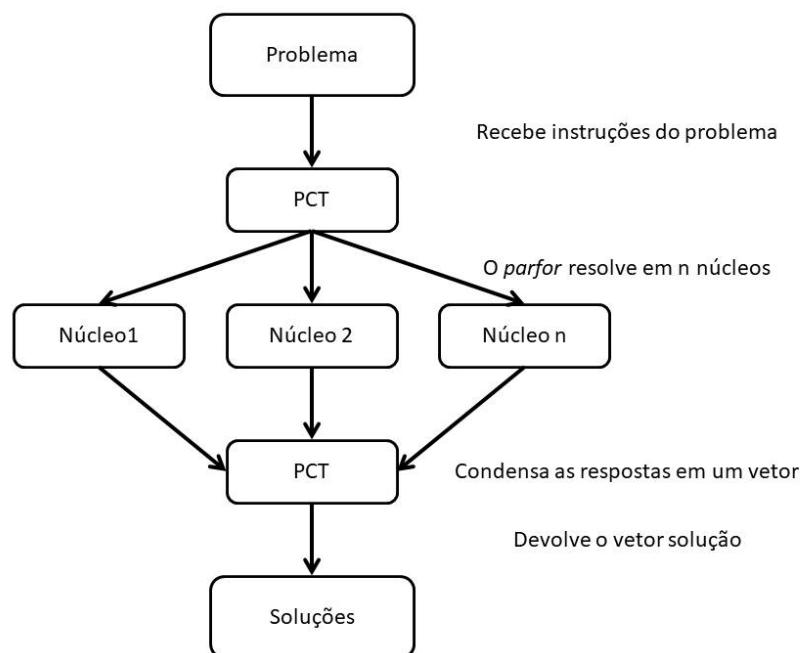
Na etapa **Definição do problema e parâmetros de simulação**, são criadas as constantes de simulação, como: o número de PLLs da rede e suas definições, o tipo de sinal e suas definições, o período de análise que será considerado, o passo de integração

numérica (no caso do algoritmo proposto foi baseado no Runge-Kutta de 4^a ordem, conforme o [40, 41], bem como as variáveis auxiliares para a execução do algoritmo.

Após feita essa definição, na etapa **Cálculo da rodada inicial**, são encontrados os valores de condição inicial da rede TWMS que será simulada, com os parâmetros $v_{c(0)}$ $\theta_o(0)$ para cada um dos PLLs da rede em questão, que servirão de ponto de partida para encontrar os demais valores. Com isso, verifica-se se a condição de estouro $i \geq Nmx$ foi atingida, caso a resposta seja **Não**, o algoritmo entra no laço de repetição para o cálculo das demais rodadas. Nessa aplicação o valor de Nmx está diretamente relacionada à quantidade de passos de integração, partindo de $t(0)$ a $t(f)$, considerando um intervalo h .

Dentro do laço de integração, a primeira etapa é dada por **Construir a rodada $i + 1$** , que é o ajuste da rede, por meio dos cálculos dos parâmetros de entrada de cada nó. Feito isso, o **Cálculo das respostas em paralelo** é executado, direcionando, de acordo com o controle do PCT do MATLAB[®] e a utilização da instrução *parfor* a um núcleo do processador, as expressões para encontrar a saída de cada um dos PLLs da rede, fazendo essa tarefa de forma automática e paralelizada. Conforme a 3.4, vemos que o PCT gerencia a paralelização, calculando os valores de $v_{c(i)}$ $\theta_o(i)$ para cada um dos PLLs da rede, de forma automática e paralela. A utilização do PCT nos permite não preocupar com o arranjo e controle da utilização dos núcleos de forma paralela, pois sua aplicação atua nesse sentido, facilitando a implementação [36].

Figura 3.4: Fluxograma da paralelização aplicada



Fonte: Autor.

Na condição de estouro do laço de repetição, o algoritmo entra no bloco **Exibe resultados** onde os valores encontrados para cada um dos PLLs, são apresentados de acordo com o interesse de análise. Nessa etapa, é de interesse a exibição do comportamento transitório e em regime do rede TWMS, para verificar se o sincronismo foi alcançado ou não e também para a comparação de resultados alcançados *versus* resultados esperados.

3.3 Condições de avaliação do algoritmo proposto

O algoritmo proposto foi submetido a três perturbações de fase do Nó Mestre distintas: quando submetido a um degrau, a uma rampa e a um sinal do tipo dente de serra. A partir desses três tipos de perturbações apresentadas, foi avaliada o comportamento dos Nós escravos da rede sob influência de diferentes filtros passa-baixas de primeira e segunda ordem, conforme [5, 15] e apresentados na tabela 3.1, contendo os filtros de primeira ordem e segunda ordem.

Tabela 3.1: Filtros passa-baixas

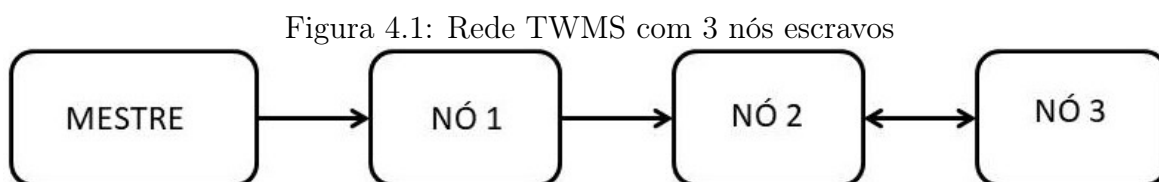
Denominação	Filtro	Função transferência
Filtro 1	PI Ativo	$F_1(s) = \frac{sT_2+1}{sT_1}$
Filtro 2	Lag-Lead Passivo	$F_2(s) = \frac{sT_2+1}{s(T_1+T_2)+1}$
Filtro 3	Lag-Lead Ativo	$F_3(s) = \frac{sT_2+1}{sT_1+1}$
Filtro 4	Filtro de primeira ordem	$F_4(s) = \frac{1}{s+1}$
Filtro 5	Filtro de segunda ordem 1	$F_5(s) = \frac{10^{-4}(s+0.3393)}{s^2+0.2262s+0.01579}$
Filtro 6	Filtro de segunda ordem 2	$F_6(s) = \frac{79.58s+1}{s(7.958s+1)}$

Para: $T_1 = 10000(2\pi)$ e $T_2 = 10(2\pi)$.

Capítulo 4

Resultados e discussões

Nessa dissertação, foi desenvolvida uma rede TWMS em que o sinal de tempo do mestre depende somente de seu oscilador. O sinal de entrada do Nó 1 é dado pelo sinal de saída do Mestre. O sinal de entrada do Nó 2 é composto pelo sinal de saída do Nó 1 e pelo sinal de saída do Nó 3. E, por fim, o Nó 3, tem seu sinal de entrada depende apenas do sinal de saída do Nó 2. Para realizar essas simulações, os modelos foram baseados nas aplicações das referências [42, 43, 44, 45]. A Figura 4.1 apresenta o esquema ilustrativo dessa rede.



Fonte: Autor.

Nessa rede, ilustrada pela Figura 4.1 foram aplicados, como dito anteriormente, três tipos de perturbações na fase do Nó Mestre. A primeira perturbação foi um degrau unitário, a segunda uma rampa unitária e em terceiro a onda periódica do tipo dente de serra. Tais resultados estão apresentados e discutidos nas próximas sessões desse capítulo. Dentre os pontos analisados estão:

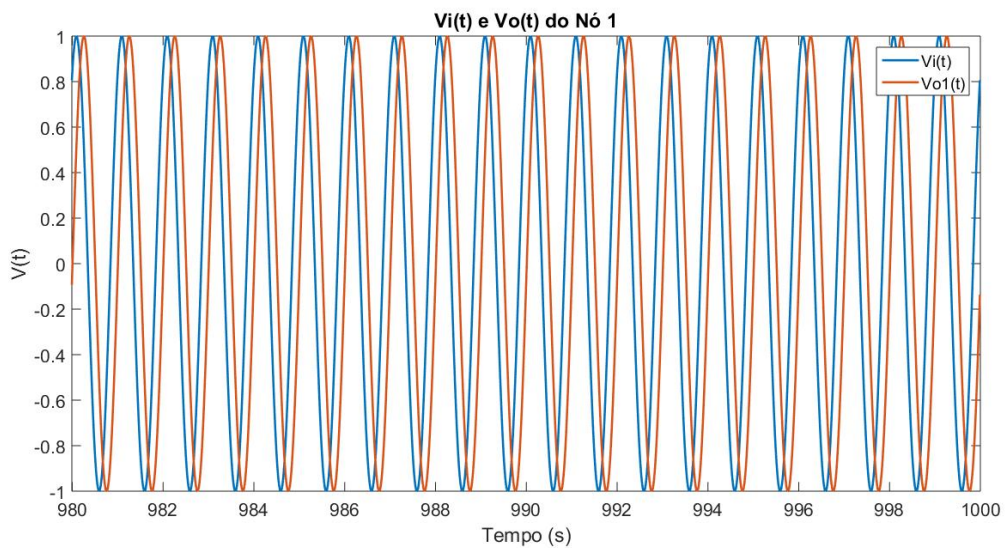
- Resposta dos nós escravos em regime;
- Resposta do erro de fase entre o nó atual e o anterior;

- Resposta da tensão de controle $V_c(t)$, se está estável e constante (representação de sincronismo);
- Representação da Figura de Lissajouls [44, 46] construída pelo sinal de entrada e saída do nó.

4.1 Resposta ao degrau unitário

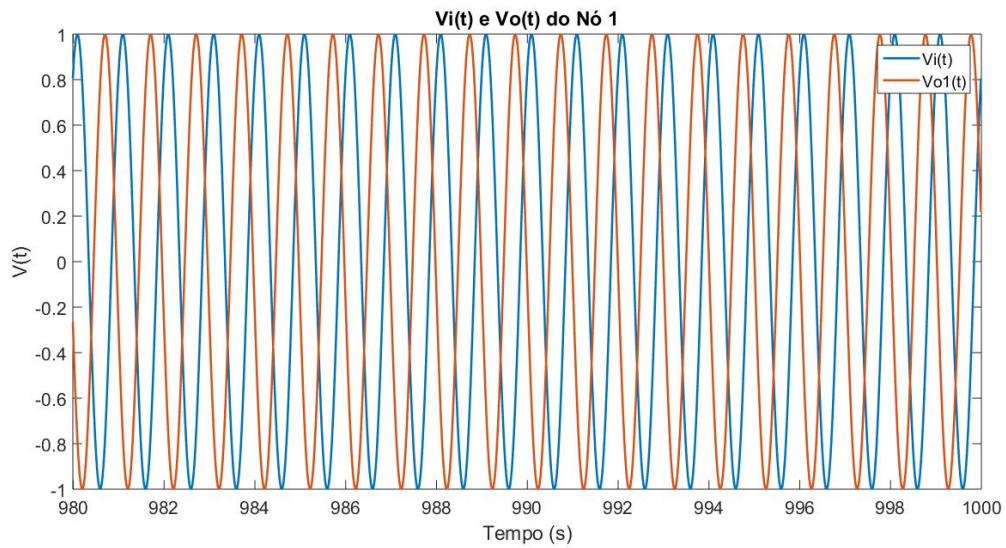
Para esse tipo de perturbação de entrada na fase do Nó Mestre, foram observados os seguintes comportamentos para no Nó Escravo 1, conforme apresentados nas Figuras 4.2, 4.3, 4.4, 4.5, 4.6 e 4.7, onde cada gráfico representa o comportamento do nó de acordo com a característica e função transferência do filtro, conforme apresentado em 3.1.

Figura 4.2: Resposta em regime do Nó Escravo 1 - Degrau unitário com o Filtro 1



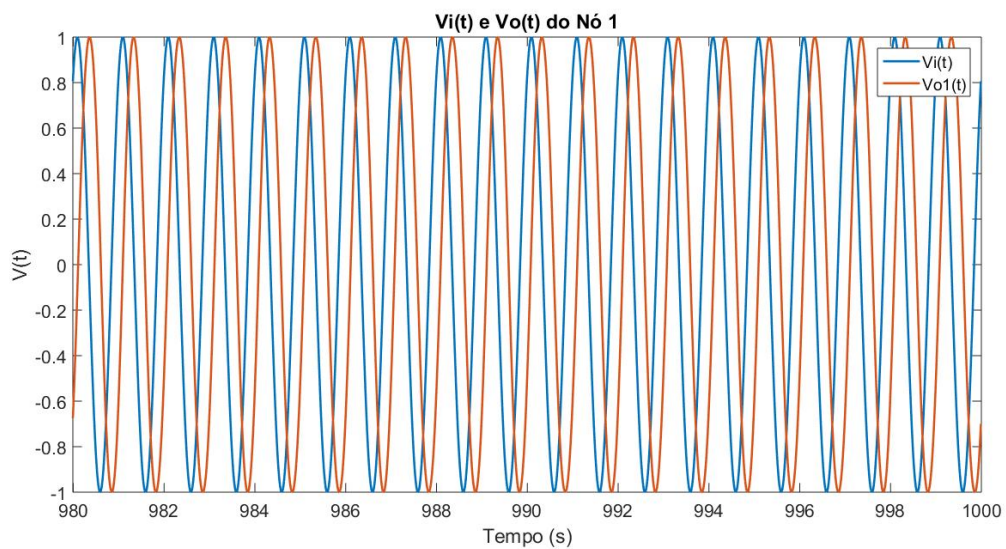
Fonte: Autor.

Figura 4.3: Resposta em regime do Nó Escravo 1 - Degrau unitário com o Filtro 2



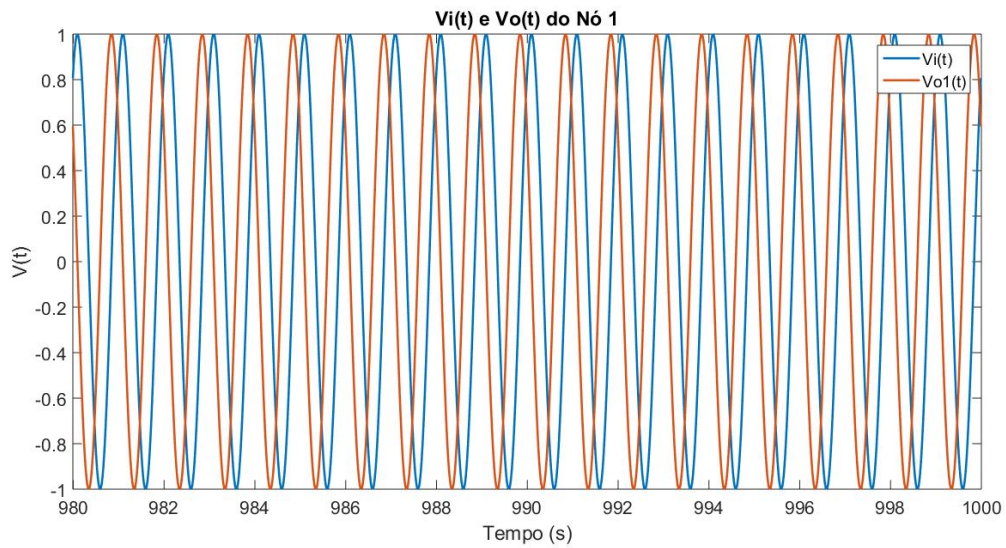
Fonte: Autor.

Figura 4.4: Resposta em regime do Nó Escravo 1 - Degrau unitário com o Filtro 3



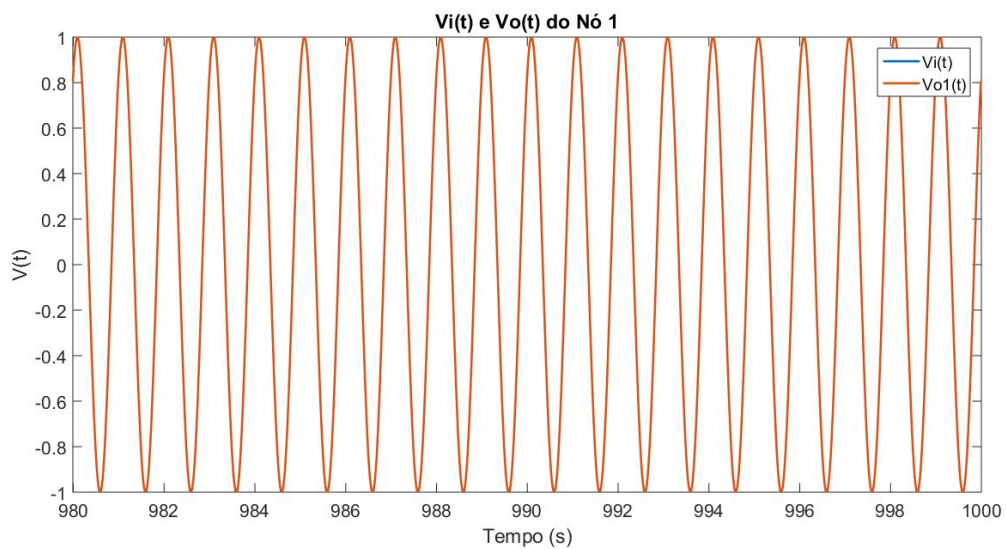
Fonte: Autor.

Figura 4.5: Resposta em regime do Nó Escravo 1 - Degrau unitário com o Filtro 4



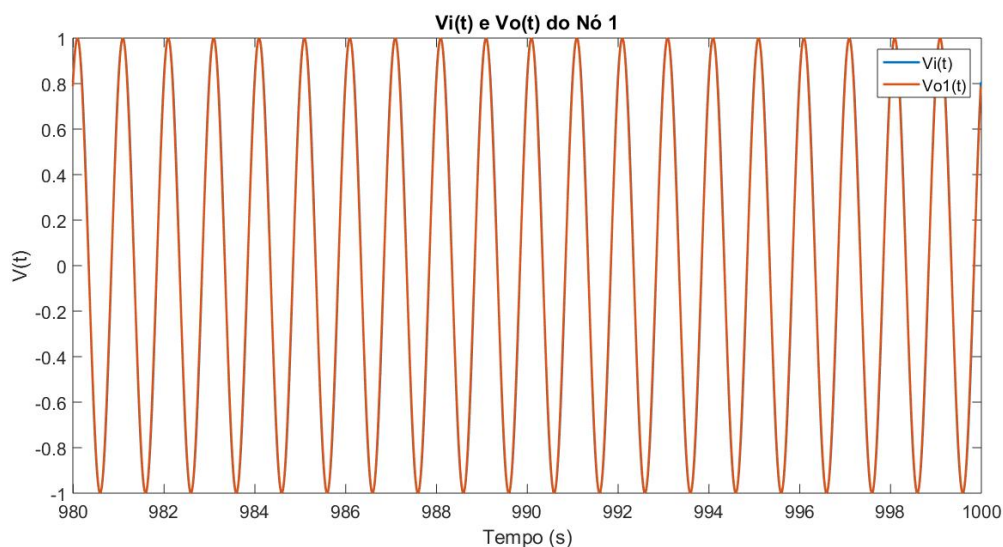
Fonte: Autor.

Figura 4.6: Resposta em regime do Nó Escravo 1 - Degrau unitário com o Filtro 5



Fonte: Autor.

Figura 4.7: Resposta em regime do Nó Escravo 1 - Degrau unitário com o Filtro 6

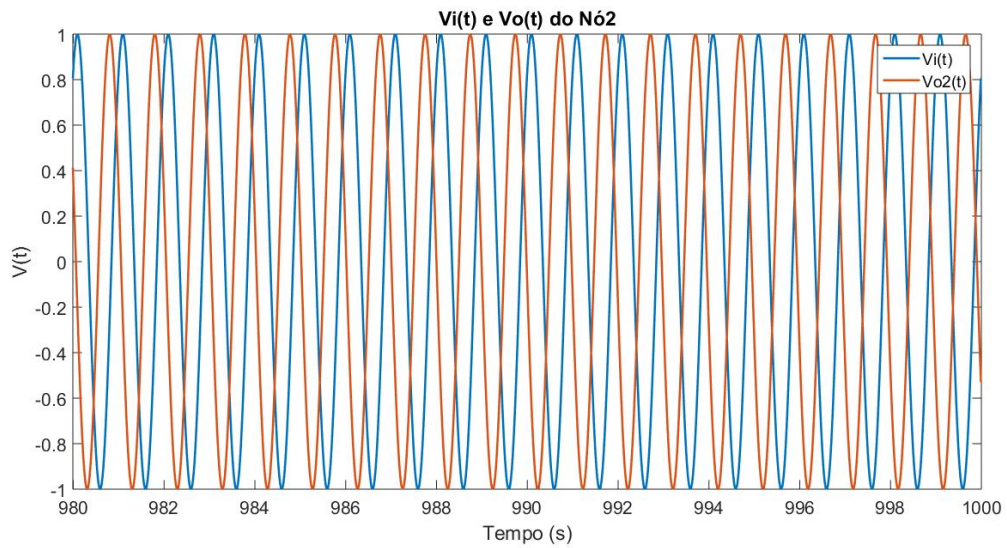


Fonte: Autor.

Ao analisarmos as respostas de saídas dos Nós Escravos 1 para cada um dos filtros desenvolvidos, notamos que os sinais apresentados nas Figuras 4.2, 4.3, 4.4 e 4.5 possuem o sincronismo determinado pelo atraso de fase do sinal de saída em relação à entrada em 90° , conforme o apresentado na teoria. Em um primeiro momento, ao se observar os resultados das 4.6 e 4.7, notamos que os sinais estão sincronizados em fase, ou seja, nesse nó houve a reprodução do sinal de entrada, gerado pelo mestre de forma integral, tendo a mesma frequência e fase, ao se observar em regime.

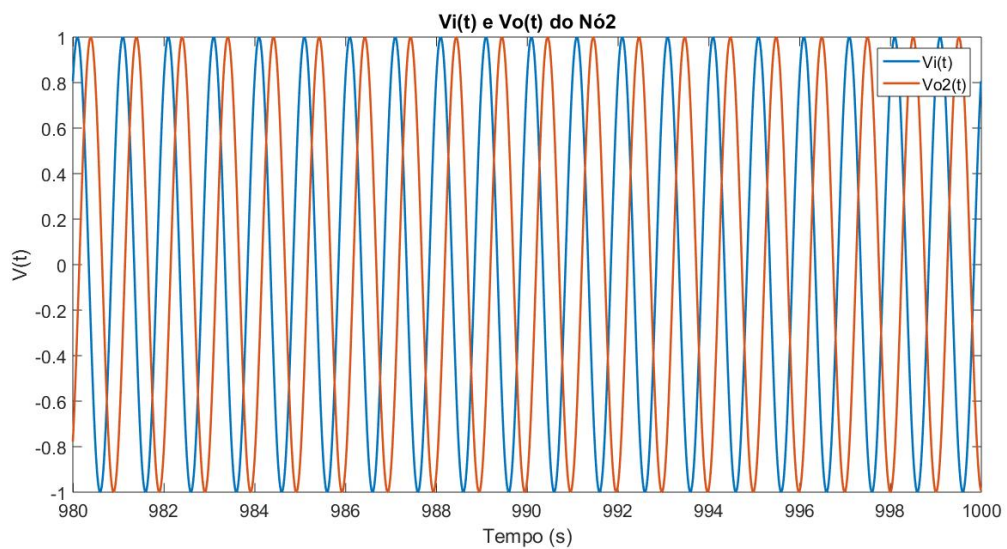
Para o Nó Escravo 2, as Figuras 4.8, 4.9, 4.10, 4.11, 4.12 e 4.13 apresentam o comportamento da saída do respectivo nó em relação ao sinal gerado pelo Nó Mestre, de acordo com cada filtro aplicado à rede.

Figura 4.8: Resposta em regime do Nó Escravo 2 - Degrau unitário com o Filtro 1



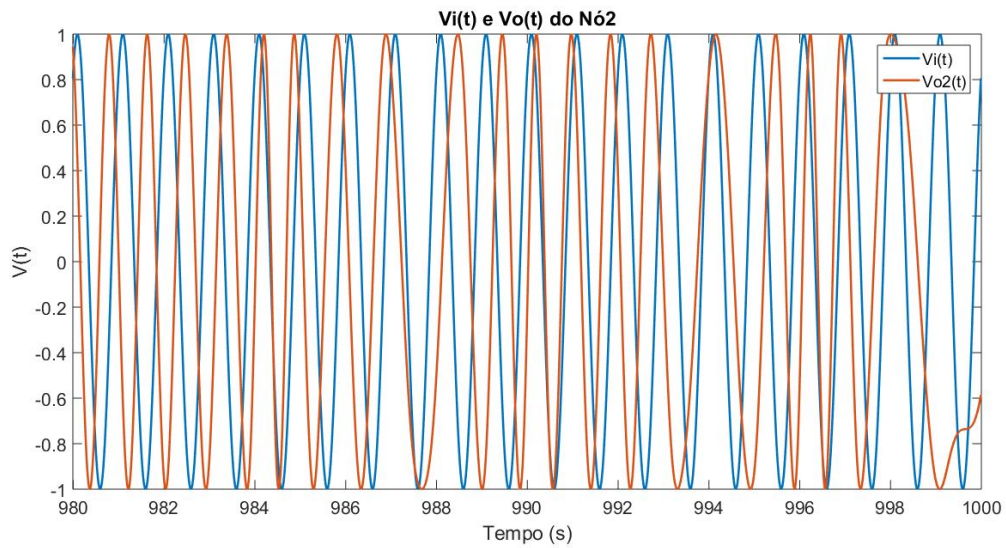
Fonte: Autor.

Figura 4.9: Resposta em regime do Nó Escravo 2 - Degrau unitário com o Filtro 2



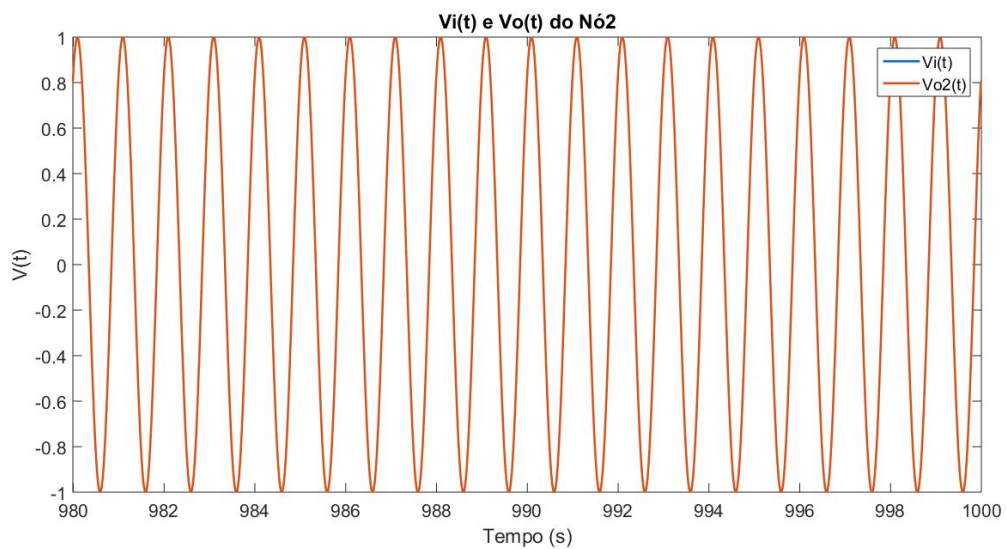
Fonte: Autor.

Figura 4.10: Resposta em regime do Nó Escravo 2 - Degrau unitário com o Filtro 3



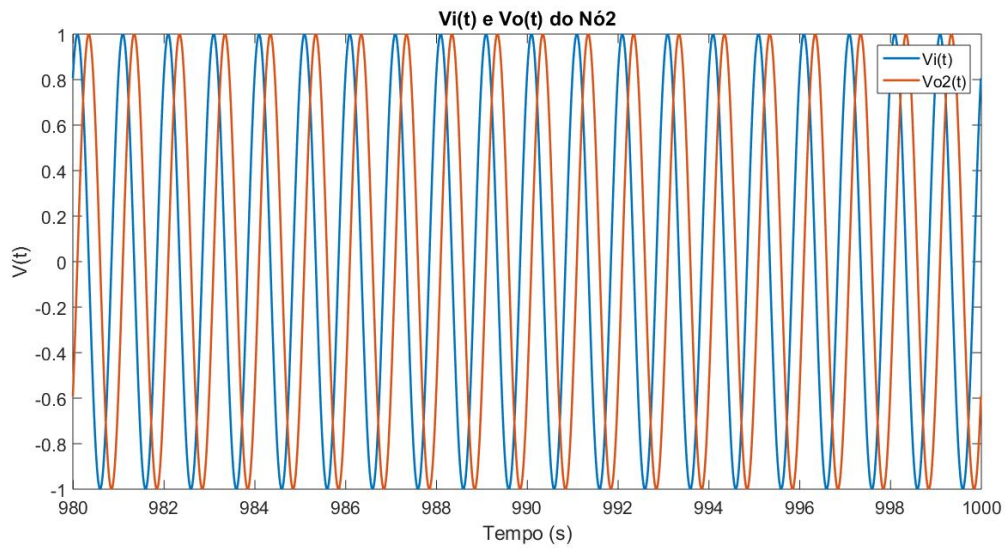
Fonte: Autor.

Figura 4.11: Resposta em regime do Nó Escravo 2 - Degrau unitário com o Filtro 4



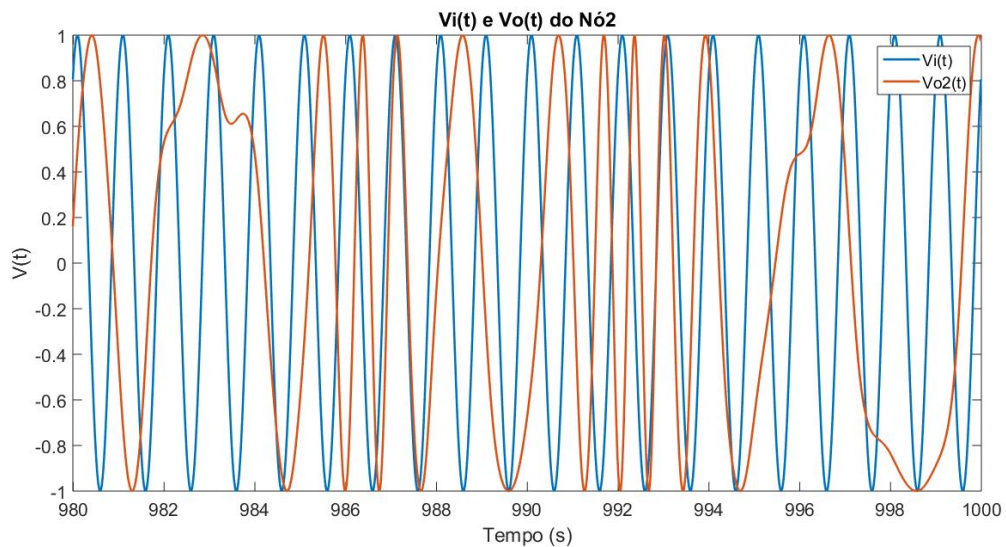
Fonte: Autor.

Figura 4.12: Resposta em regime do Nó Escravo 2 - Degrau unitário com o Filtro 5



Fonte: Autor.

Figura 4.13: Resposta em regime do Nó Escravo 2 - Degrau unitário com o Filtro 6



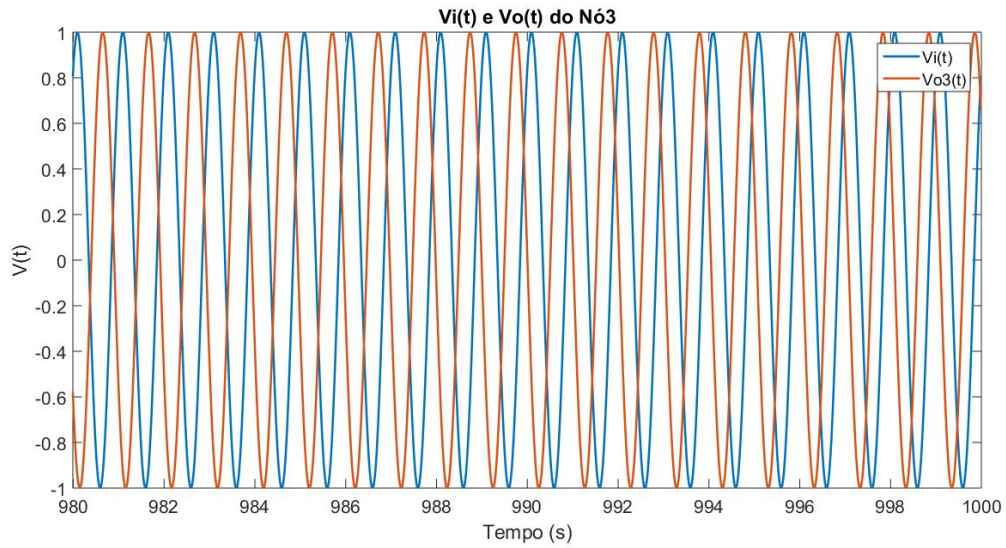
Fonte: Autor.

Ao analisarmos a resposta do Nó Escravo 2, notamos que para os Filtros 1 e 2, representados pela Figuras 4.8 e 4.9 um escorregamento constante da fase do sinal de saída. Com os Filtros 3 e 6 (Figuras 4.10 e 4.13 respectivamente) temos um comportamento inconclusivo para o período analisado, sendo necessária a confirmação do

comportamento de $V_c(t)$, que será discutido posteriormente. os Filtros 4 e 5 (Figuras 4.11 e 4.12 respectivamente), notamos o sincronismo com o escorregamento em 90° da fase do sinal de saída em relação à fase do Mestre.

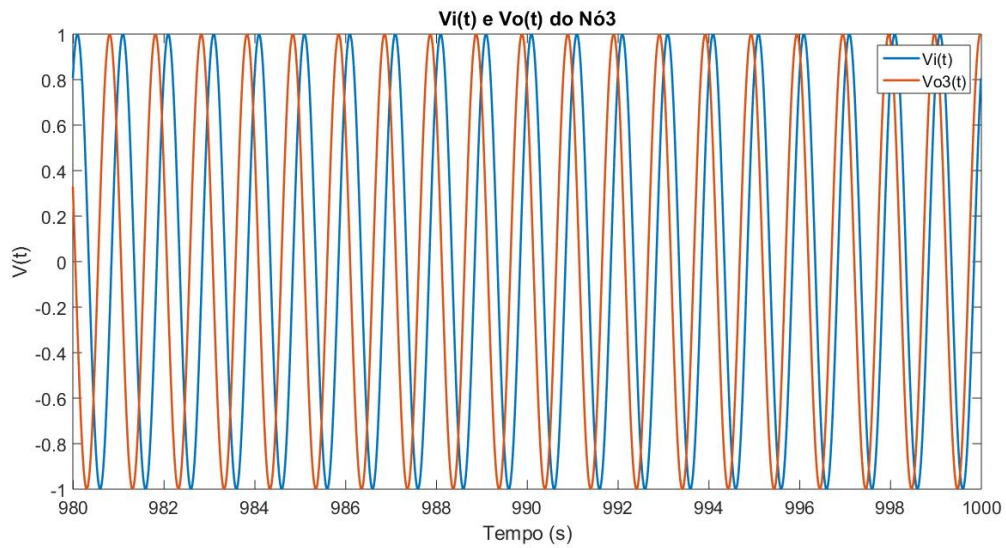
Para o Nó Escravo 3, as Figuras 4.14, 4.15, 4.16, 4.17, 4.18 e 4.19 apresentam o comportamento da saída do respectivo nó em relação ao sinal gerado pelo Nó Mestre, de acordo com cada filtro aplicado à rede.

Figura 4.14: Resposta em regime do Nó Escravo 3 - Degrau unitário com o Filtro 1



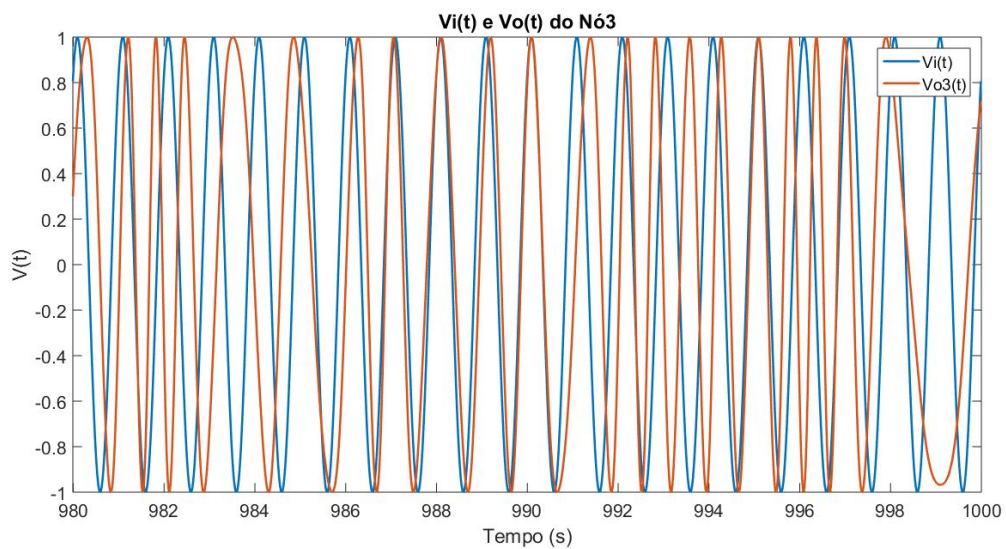
Fonte: Autor.

Figura 4.15: Resposta em regime do Nó Escravo 3 - Degrau unitário com o Filtro 2



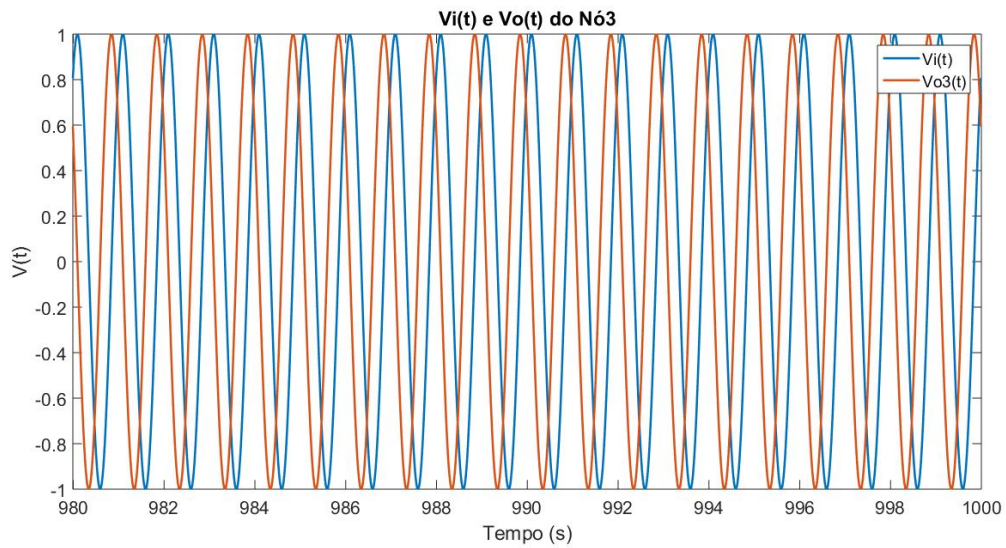
Fonte: Autor.

Figura 4.16: Resposta em regime do Nó Escravo 3 - Degrau unitário com o Filtro 3



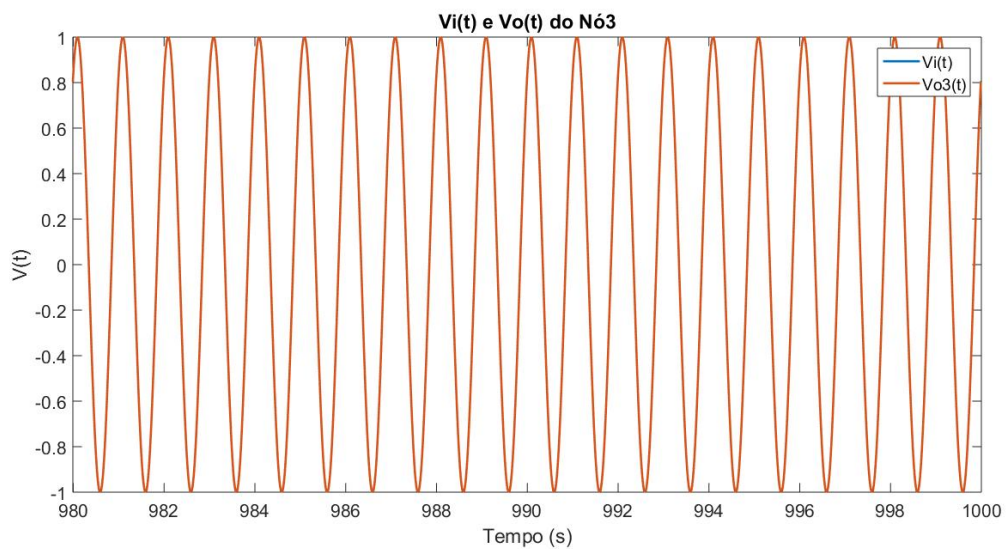
Fonte: Autor.

Figura 4.17: Resposta em regime do Nó Escravo 3 - Degrau unitário com o Filtro 4



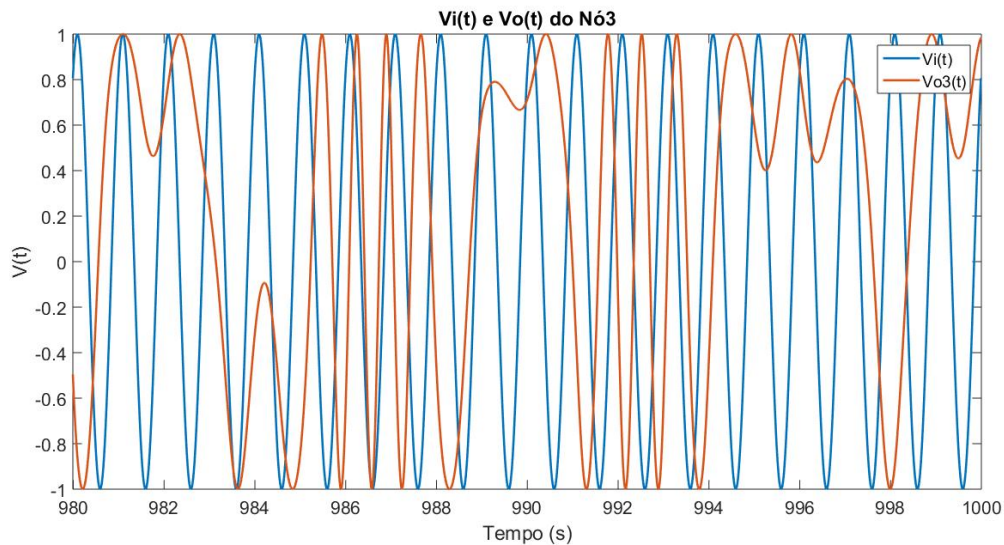
Fonte: Autor.

Figura 4.18: Resposta em regime do Nó Escravo 3 - Degrau unitário com o Filtro 5



Fonte: Autor.

Figura 4.19: Resposta em regime do Nó Escravo 3 - Degrau unitário com o Filtro 6

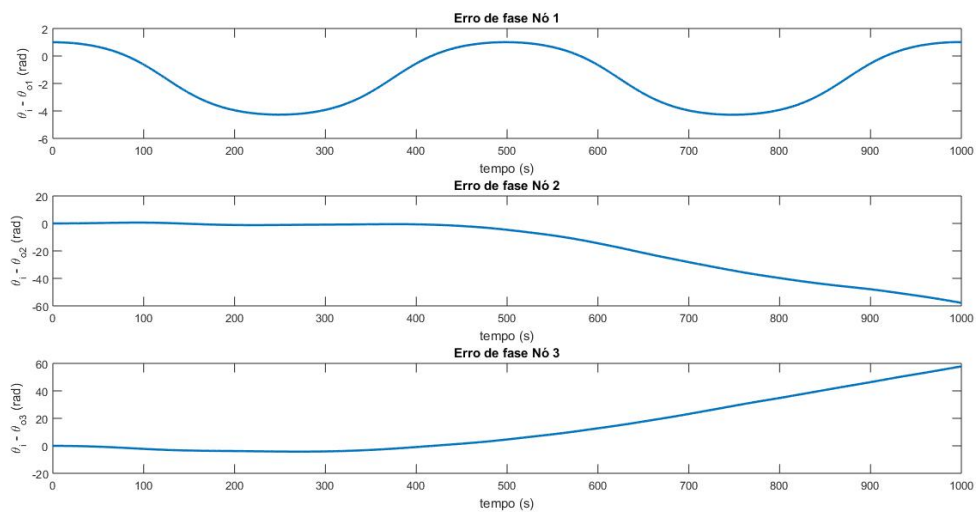


Fonte: Autor.

Ao analisarmos a resposta do Nó Escravo 3, notamos que para os Filtros 4 e 5, ilustrados pelas Figuras 4.17 e 4.18 alcançam o sincronismo, sendo o primeiro caso um sincronismo de frequência e com um deslocamento de 90° de fase e no segundo caso há um sincronismo de fase e frequência. Como o Nó Escravo 3 tem seu sinal de entrada dependente do sinal de saída do Escravo 2, a sua onda de saída é diretamente afetada por isso. Nos Filtros 1 e 2 (Figuras 4.14 e 4.15, respectivamente), temos o escorregamento de fase, conforme foi visto nos gráficos de saída do Nó 2 para esses dois filtros, apresentados anteriormente pelas Figuras 4.8 e 4.9 respectivamente. Por fim, conforme encontrado na saída do Nó Escravo 2, notamos que os Filtros 3 e 6 (Figuras 4.16 e 4.19, respectivamente), o período de análise é inconclusivo, necessitando a análise de $V_c(t)$ para a determinação do estado.

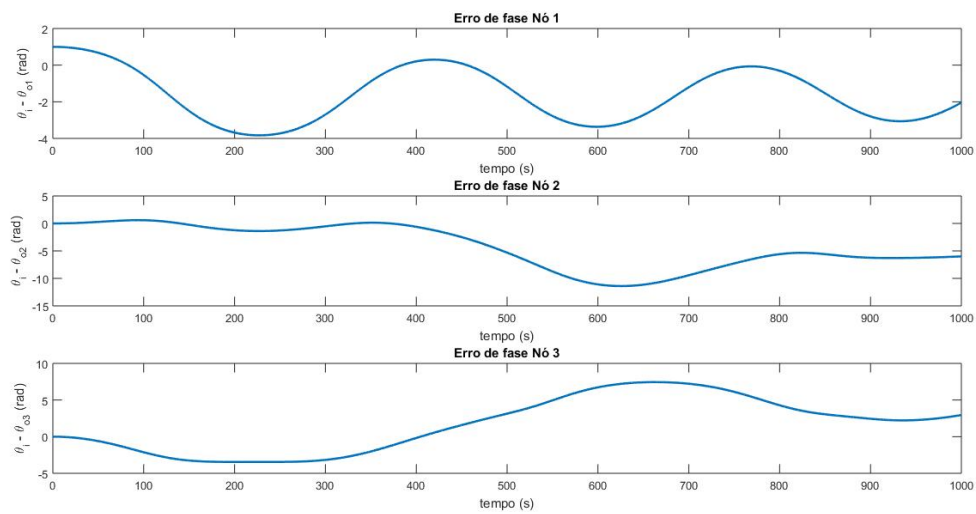
As Figuras 4.20, 4.21, 4.22, 4.23, 4.24 e 4.25 apresentam o erro de fase do sinal de saída dos nós escravos em relação à sua entrada. No caso dos Nós Escravos 1 e 3, o valor de $\theta_i(t)$ está relacionado à $\theta_o(t)$ do Nó Mestre e do Nó 2, respectivamente, enquanto o Nó 2 tem sua fase de acordo com a composição de $\theta_o(t)$ do Nó 1 e 3.

Figura 4.20: Erro de fase da rede TWMS - Degrau unitário com o Filtro 1



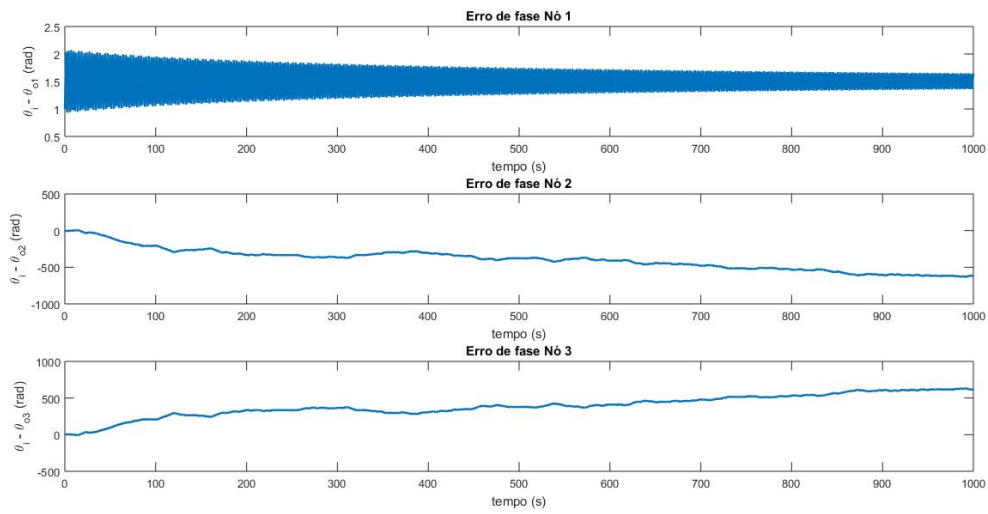
Fonte: Autor.

Figura 4.21: Erro de fase da rede TWMS - Degrau unitário com o Filtro 2



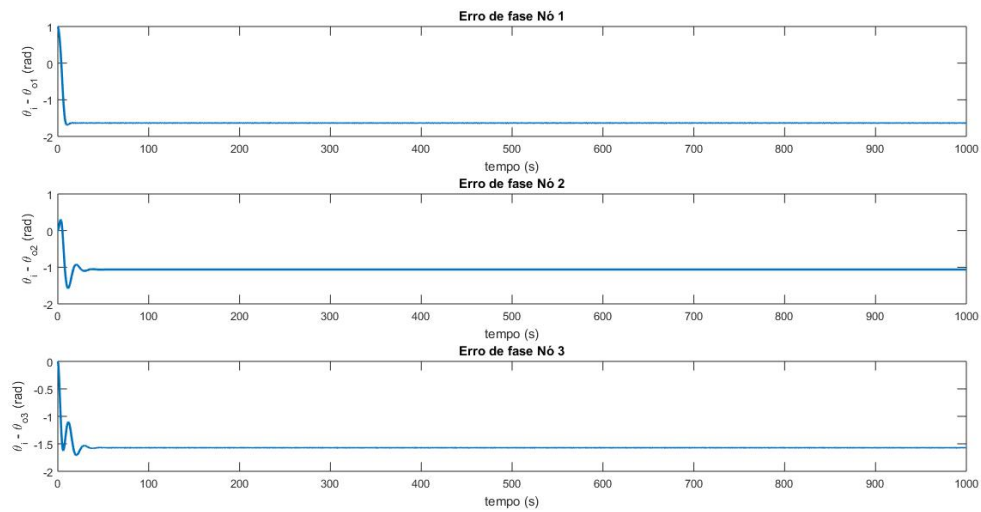
Fonte: Autor.

Figura 4.22: Erro de fase da rede TWMS - Degrau unitário com o Filtro 3



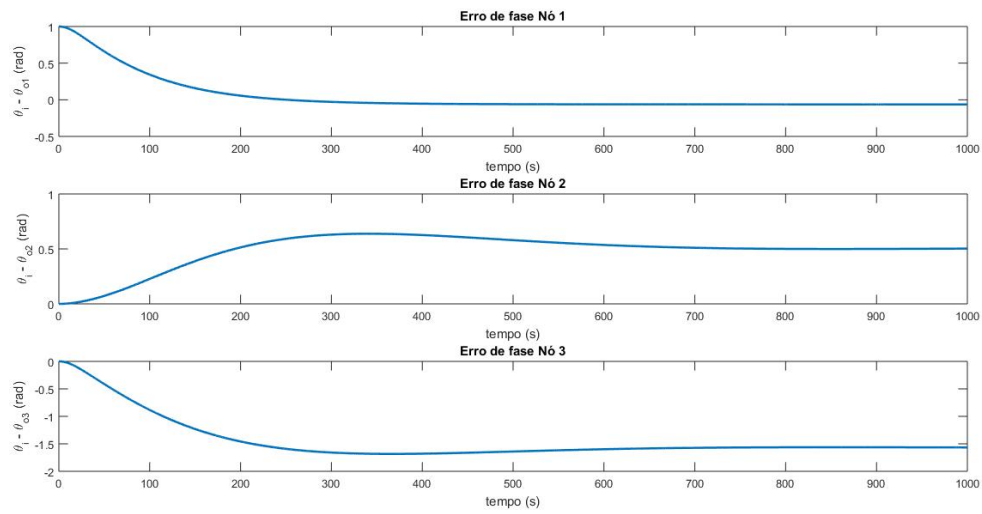
Fonte: Autor.

Figura 4.23: Erro de fase da rede TWMS - Degrau unitário com o Filtro 4



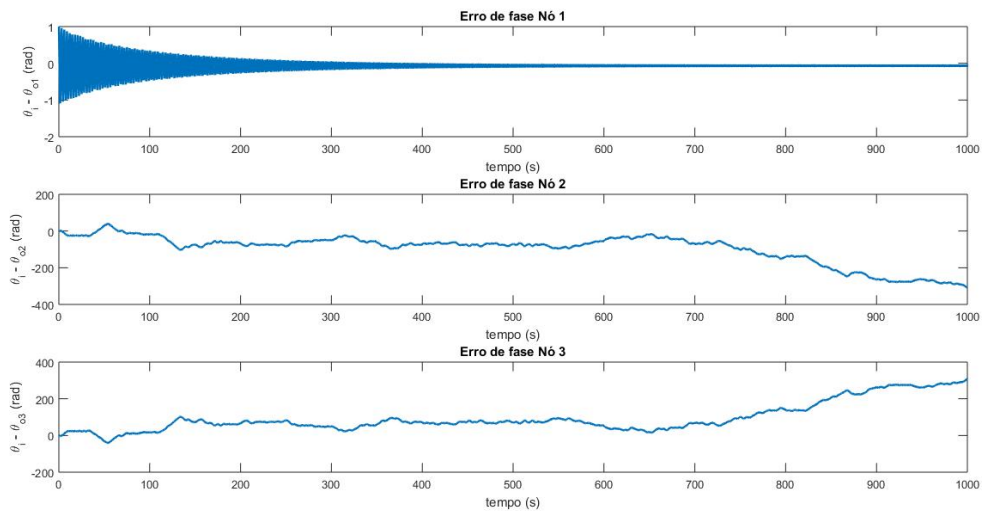
Fonte: Autor.

Figura 4.24: Erro de fase da rede TWMS - Degrau unitário com o Filtro 5



Fonte: Autor.

Figura 4.25: Erro de fase da rede TWMS - Degrau unitário com o Filtro 6



Fonte: Autor.

Ao observarmos o primeiro gráfico dos resultados apresentados Figuras 4.22, 4.23, 4.24 e 4.21, notamos que o erro de fase possuem a tendência de se estabilizarem em um valor constante, o que representa a sincronicidade do Nó 1. Nos resultados apresentados nas Figuras 4.20 e 4.21 notamos que, ao subtrairmos o degrau unitário aplicado ao

sistema, notamos que ambos erros de fase possuem média zero, comportamento que representa também o estado síncrono do Nó Escravo 1 em relação à sua entrada, que no caso é a saída do Nó Mestre.

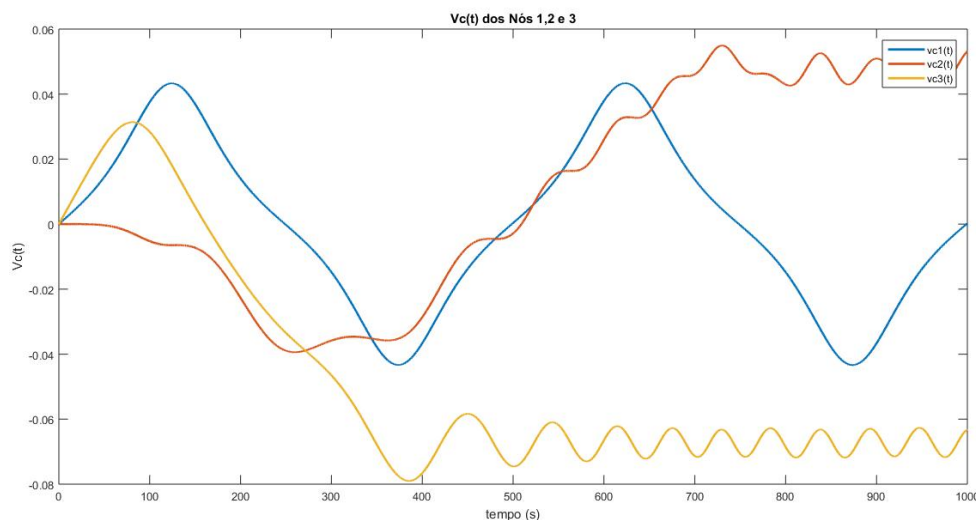
No segundo e terceiro gráficos dos resultados apresentados nas Figuras 4.20 e 4.21 observamos que os erros de fases dos Nós 2 e 3 ainda estão variando, sendo esses resultados operando na região considerada faixa de captura.

Nos gráficos apresentados nas Figuras 4.22 e 4.25, notamos que os erros de fase dos Nós Escravos 2 e 3 são crescentes, o que explica o deslocamento de fase observados nas Figuras 4.10 e 4.13, para o Nó Escravo 2 e as Figuras 4.16 e 4.19, para o Nó Escravo 3, respectivamente.

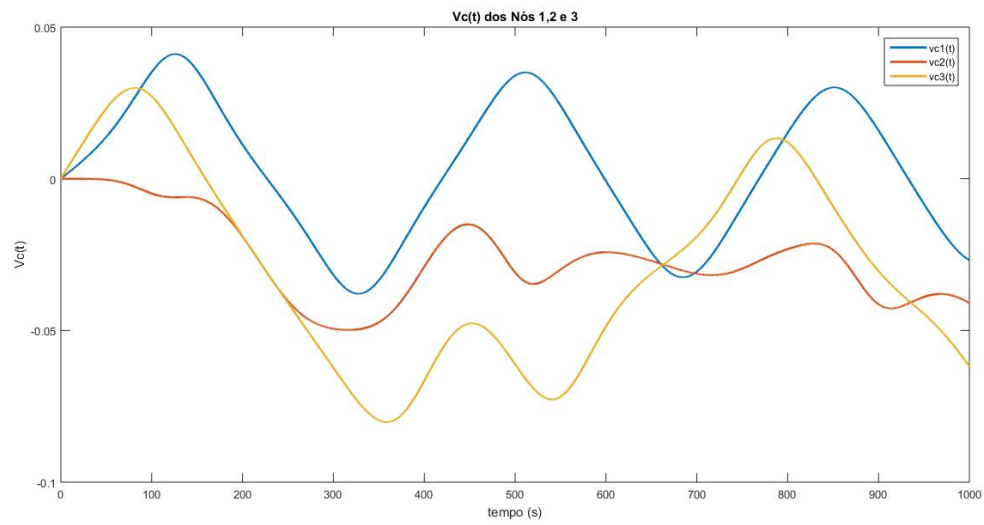
As Figuras 4.23 e 4.24 confirmam o estado síncrono dos Nós 2 e 3 para os Filtros 4 e 5, conforme discutido anteriormente, nos resultados das Figuras 4.11 e 4.12, para o Nó Escravo 2 e, das Figuras 4.17 e 4.18 para o Nó Escravo 3, respectivamente. Esse sincronismo se dá devido à estabilização em um valor constante para o erro de fase desses dois nós para a rede construída com os Filtros 4 e 5.

As Figuras 4.26, 4.27, 4.28, 4.29, 4.30 e 4.31 apresentam o sinal de controle $V_c(t)$ da rede TWMS de acordo com os filtros utilizados.

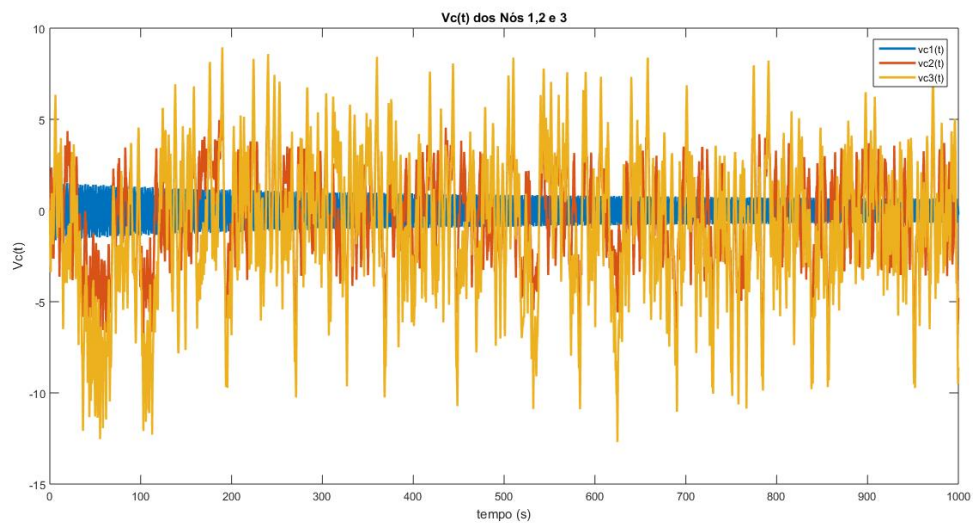
Figura 4.26: $V_c(t)$ para cada nó da rede TWMS - Degrau unitário com o Filtro 1



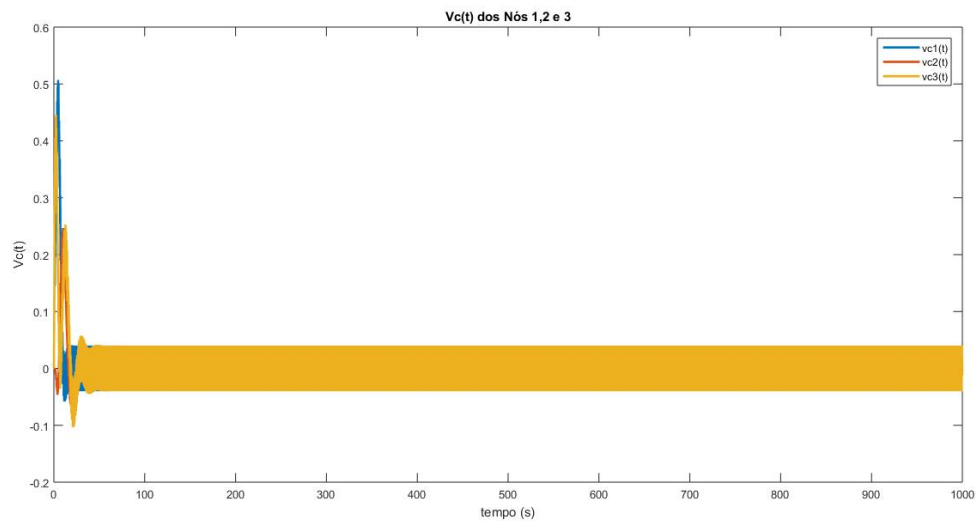
Fonte: Autor.

Figura 4.27: $V_c(t)$ para cada nó da rede TWMS - Degrau unitário com o Filtro 2

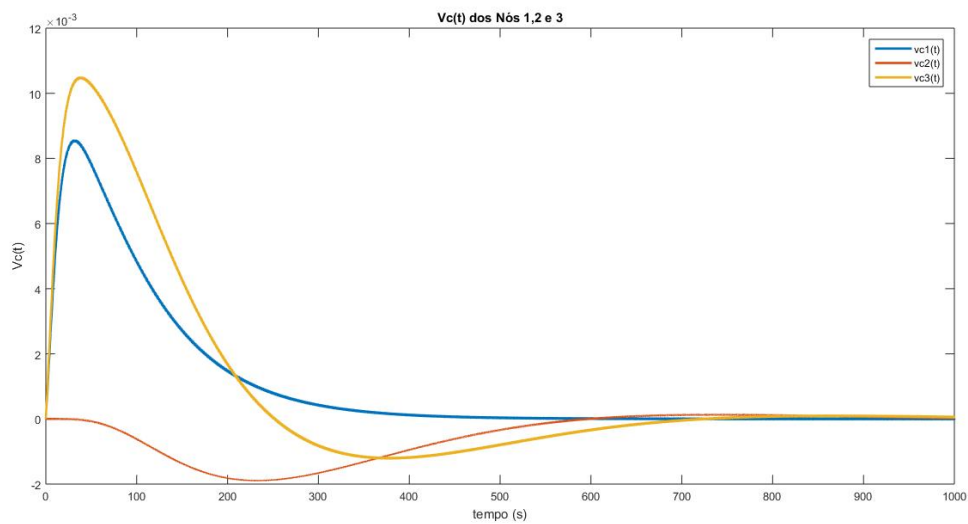
Fonte: Autor.

Figura 4.28: $V_c(t)$ para cada nó da rede TWMS - Degrau unitário com o Filtro 3

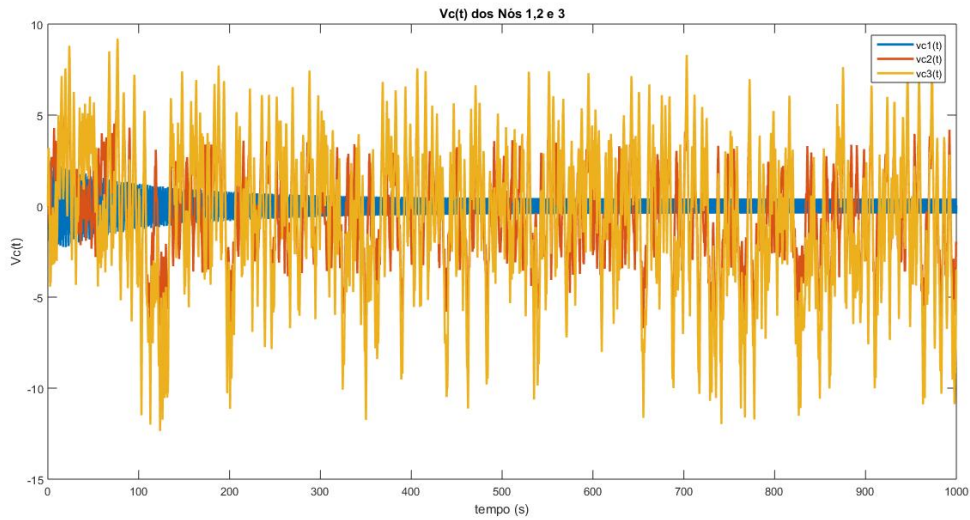
Fonte: Autor.

Figura 4.29: $V_c(t)$ para cada nó da rede TWMS - Degrau unitário com o Filtro 4

Fonte: Autor.

Figura 4.30: $V_c(t)$ para cada nó da rede TWMS - Degrau unitário com o Filtro 5

Fonte: Autor.

Figura 4.31: $V_c(t)$ para cada nó da rede TWMS - Degrau unitário com o Filtro 6

Fonte: Autor.

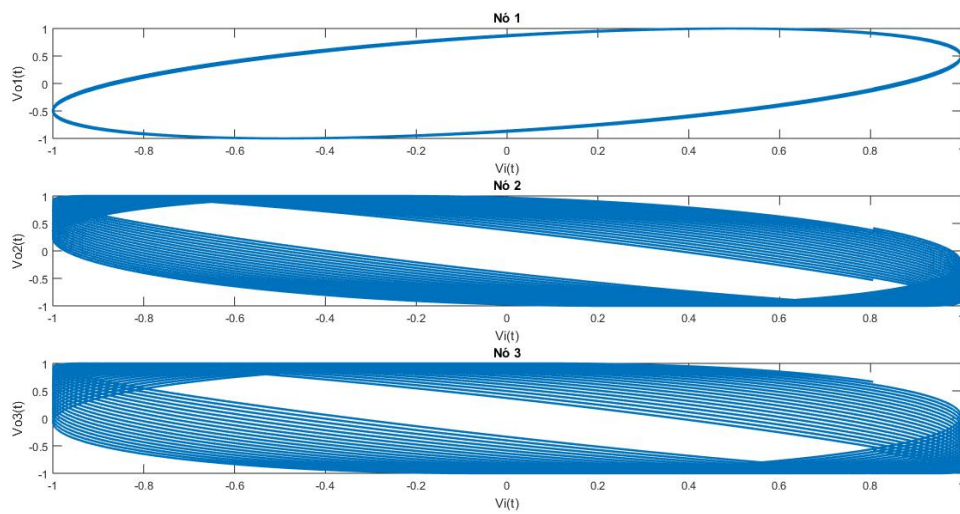
Ao observarmos as Figuras 4.26 e 4.27, os Filtros 1 e 2 possui em seu sinal de controle $V_c(t)$ do Nó 1 a média zero, representando o estado síncrono desse nó em relação à sua entrada, conforme observado anteriormente. Todavia, ao observarmos o sinal $V_c(t)$ nos demais nós, notamos que não possuem média zero, representando o estado não síncrono desses nós.

Com as Figuras 4.28 e 4.31 notamos que os Filtros 3 e 6 apresentam média zero somente no Nó Escravo 1, tendo os demais Nós Escravos com médias diferentes de zero, apresentando o estado não síncrono desses nós.

Nas Figuras 4.29 e 4.30 temos que os sinais $V_c(t)$ dos Nós Escravos 1, 2 e 3 alcançam o estado síncrono para os Filtros 4 e 5, pois a média zero para todos os nós.

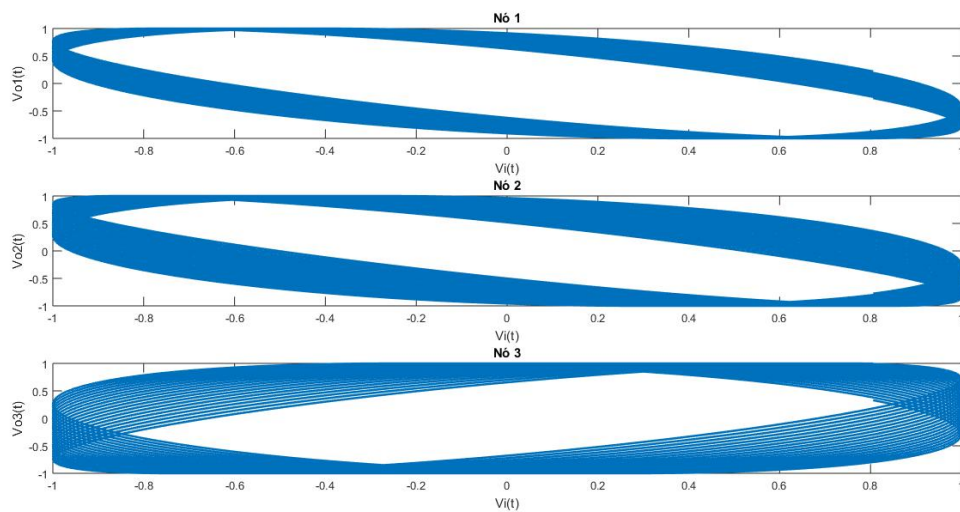
Por fim, as Figuras 4.32, 4.33, 4.34, 4.35, 4.36 e 4.37 apresentam as Figuras de Lissajous, conforme [44, 46].

Figura 4.32: Lissajous para cada nó da rede TWMS - Degrau unitário com o Filtro 1



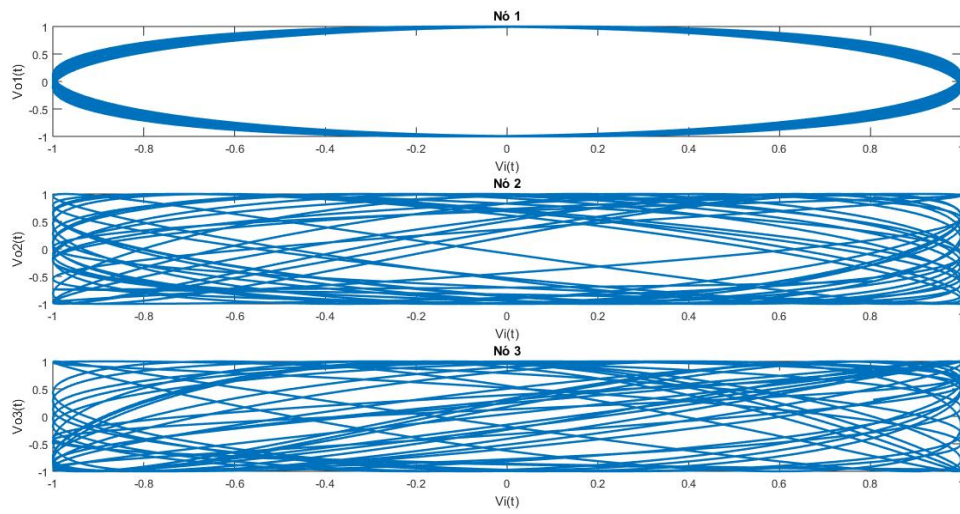
Fonte: Autor.

Figura 4.33: Lissajous para cada nó da rede TWMS - Degrau unitário com o Filtro 2



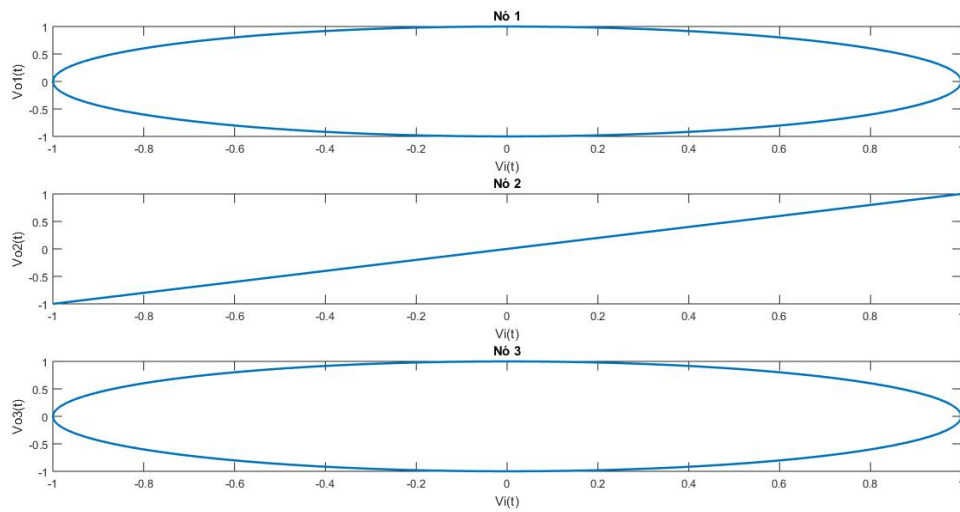
Fonte: Autor.

Figura 4.34: Lissajous para cada nó da rede TWMS - Degrau unitário com o Filtro 3



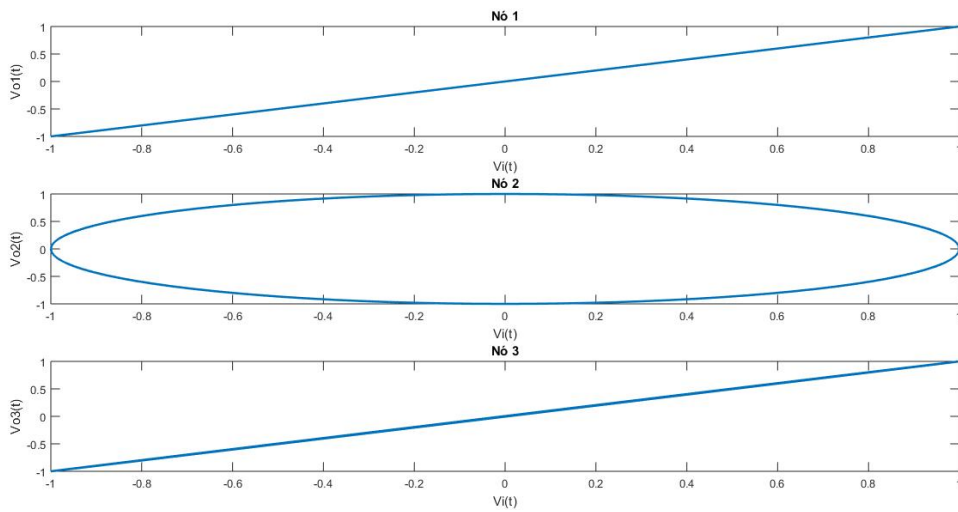
Fonte: Autor.

Figura 4.35: Lissajous para cada nó da rede TWMS - Degrau unitário com o Filtro 4



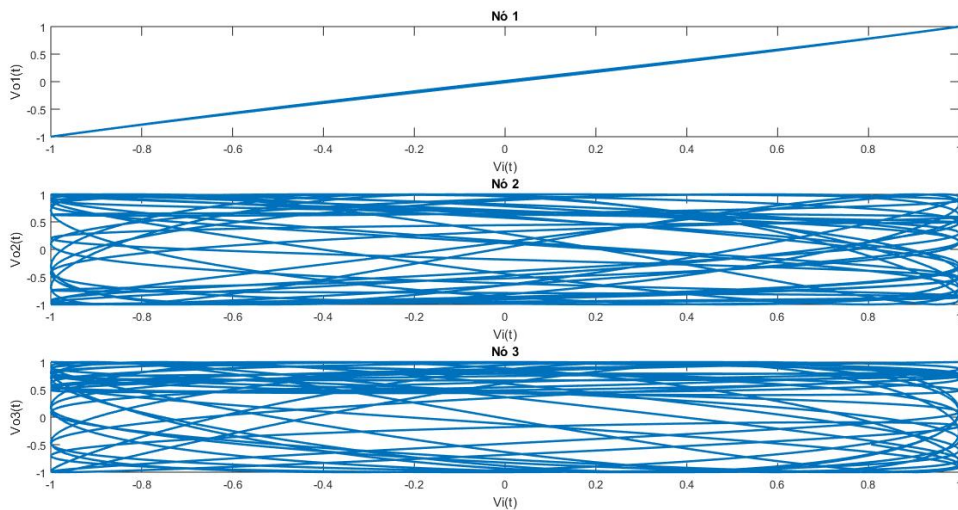
Fonte: Autor.

Figura 4.36: Lissajous para cada nó da rede TWMS - Degrau unitário com o Filtro 5



Fonte: Autor.

Figura 4.37: Lissajous para cada nó da rede TWMS - Degrau unitário com o Filtro 6



Fonte: Autor.

Ao observarmos as Figuras 4.32 e 4.33, podemos notar, por outra forma, que para os Filtros 1 e 2, o Nó Escravo 1 encontra o sincronismo, enquanto os Nós Escravos 2 e 3 possuem o escorregamento de fase, devido a não coincidência dos pontos na construção da imagem de Lissajous.

Com os Filtros 3 e 6, apresentados pelas Figuras 4.34 e 4.37 respectivamente, notamos que somente o Nó Escravo 1 encontra o estado síncrono, no primeiro caso sincronismo de quadratura enquanto no segundo é sincronismo de fase. Para os Nós Escravos 2 e 3, notamos que não há um padrão definido, caracterizando o estado como assíncrono desses respectivos nós.

Nas Figuras 4.35 e 4.36 confirmamos a sincronicidade de todos os Nós Escravos em Relação ao Mestre para os Filtros 4 e 5. Enquanto que para o Filtro 4, o sincronismo se dá na seguinte forma: sincronismo de quadratura para o primeiro e terceiro nó e sincronismo de fase para o segundo nó, para o Filtro 5, o sincronismo é de fase para o primeiro e terceiro nó e quadratura para o segundo nó.

A Tabela 4.1 apresenta os tempos de simulação para a rede TWMS com os diferentes filtros para a versão sequencial e em paralelo do algoritmo proposto em um processador de dois núcleos.

Tabela 4.1: Tempos de simulação para a resposta ao degrau de fase - Intel® Core™ i3-4005U CPU @ 1.70GHz (dois núcleos)

Filtro	Sequencial (s)	Paralelo (s)
Filtro 1	10,18	18906,30
Filtro 2	10,18	18916,19
Filtro 3	10,04	18865,72
Filtro 4	10,14	18942,84
Filtro 5	136,88	20056,77
Filtro 6	137,61	20055,65

A Tabela 4.2 apresenta os tempos de simulação para a rede TWMS com os diferentes filtros para a versão sequencial e em paralelo do algoritmo proposto em um processador de quatro núcleos.

Tabela 4.2: Tempos de simulação para a resposta ao degrau de fase - Intel[®] Core[™] i5-4440S CPU @ 2.8GHz (quatro núcleos)

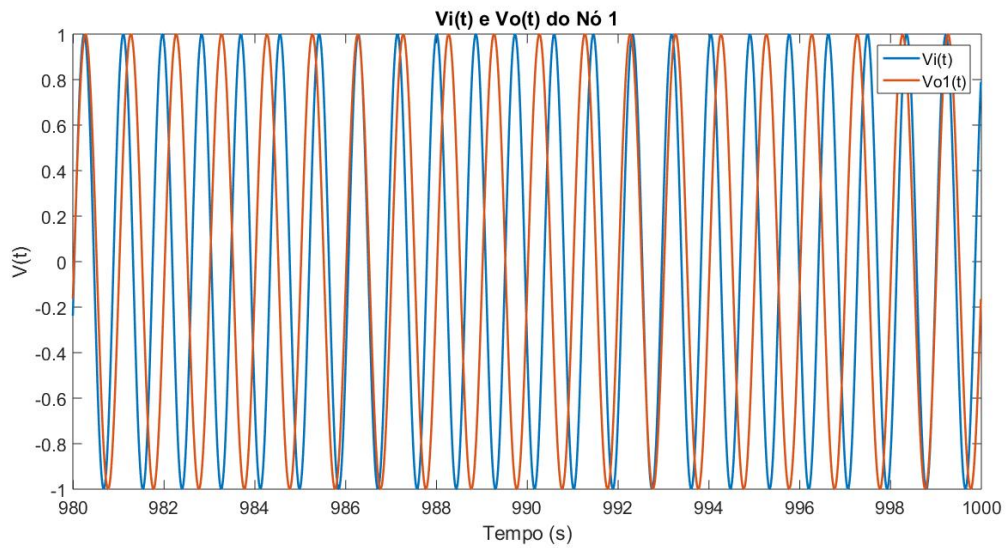
Filtro	Sequencial (s)	Paralelo (s)
Filtro 1	2,33	6600,81
Filtro 2	2,34	6229,20
Filtro 3	2,35	6597,44
Filtro 4	2,35	6575,10
Filtro 5	156,98	7290,53
Filtro 6	156,93	7275,32

Ao compararmos os tempos de simulação entre os dois processadores, notamos que houve um ganho significativo com a execução tanto algoritmo na lógica sequencial quanto na lógica em paralelo. Notamos que, com problemas mais complexos, como no caso dos cálculos para os filtros de 2^a ordem, representados pelos Filtros 5 e 6, o custo computacional para resolver esse problema, que exige mais recurso, sob a ótica do paralelismo, está na mesma ordem de grandeza para os problemas de primeira ordem, representado pelos Filtros 1, 2, 3 e 4, evidenciando que, com problemas mais complexos, o paralelismo se torna vantajoso.

4.2 Resposta à rampa unitária

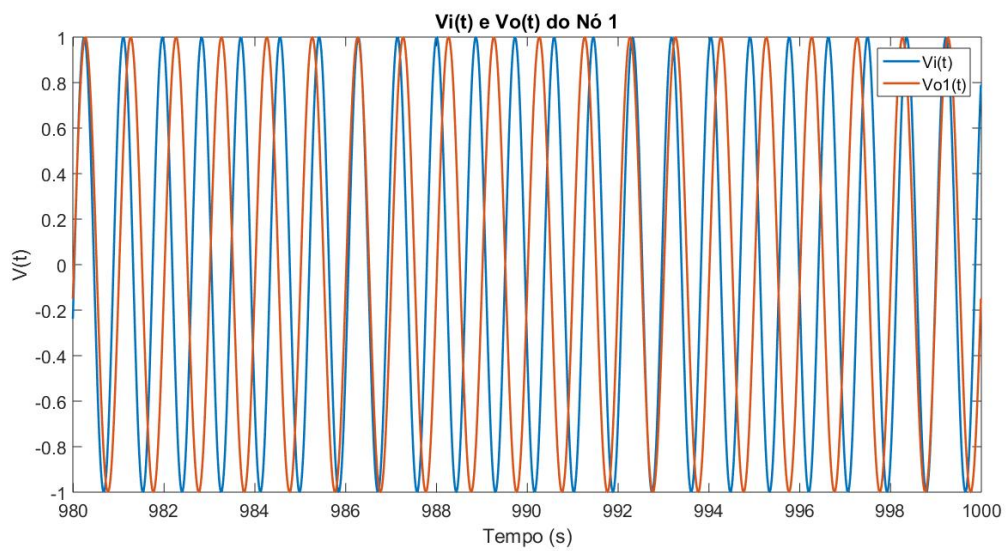
Para esse tipo de perturbação de entrada na fase do Nó Mestre, foram observados os seguintes comportamentos para no Nó Escravo 1, conforme apresentados nas Figuras 4.38, 4.39, 4.40, 4.41, 4.42 e 4.43, onde cada gráfico representa o comportamento do nó de acordo com a característica e função transferência do filtro, conforme apresentado em 3.1.

Figura 4.38: Resposta em regime do Nó Escravo 1 - Rampa unitária com o Filtro 1



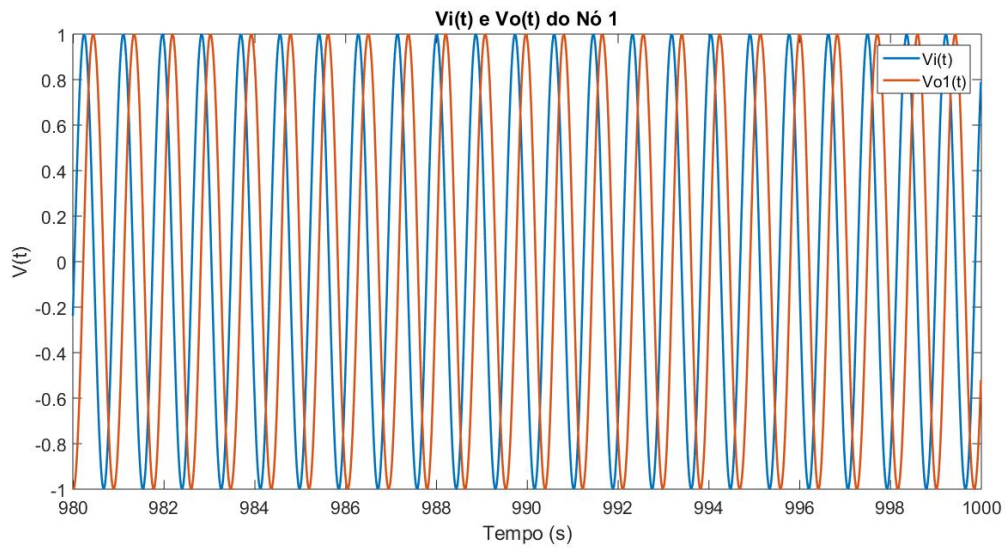
Fonte: Autor.

Figura 4.39: Resposta em regime do Nó Escravo 1 - Rampa unitária com o Filtro 2



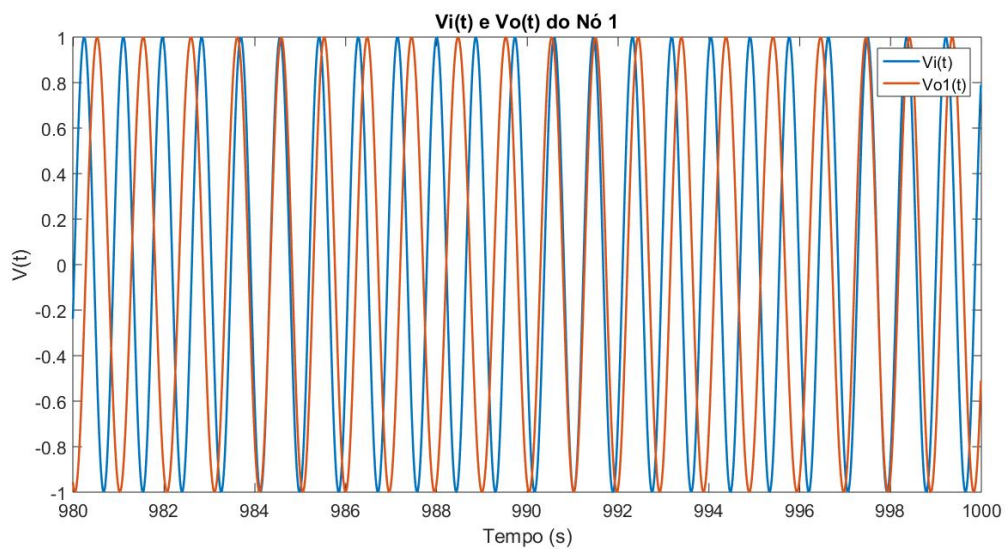
Fonte: Autor.

Figura 4.40: Resposta em regime do Nó Escravo 1 - Rampa unitária com o Filtro 3



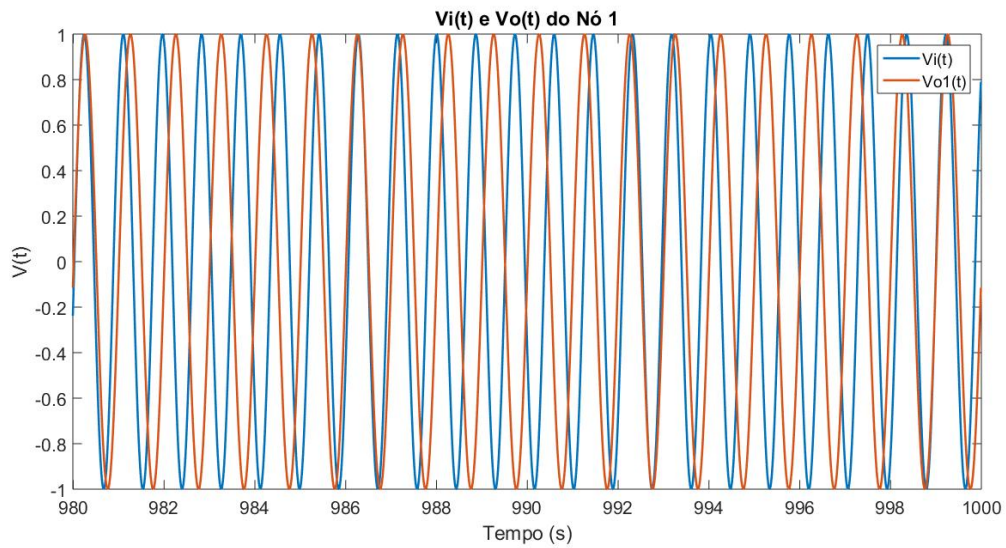
Fonte: Autor.

Figura 4.41: Resposta em regime do Nó Escravo 1 - Rampa unitária com o Filtro 4



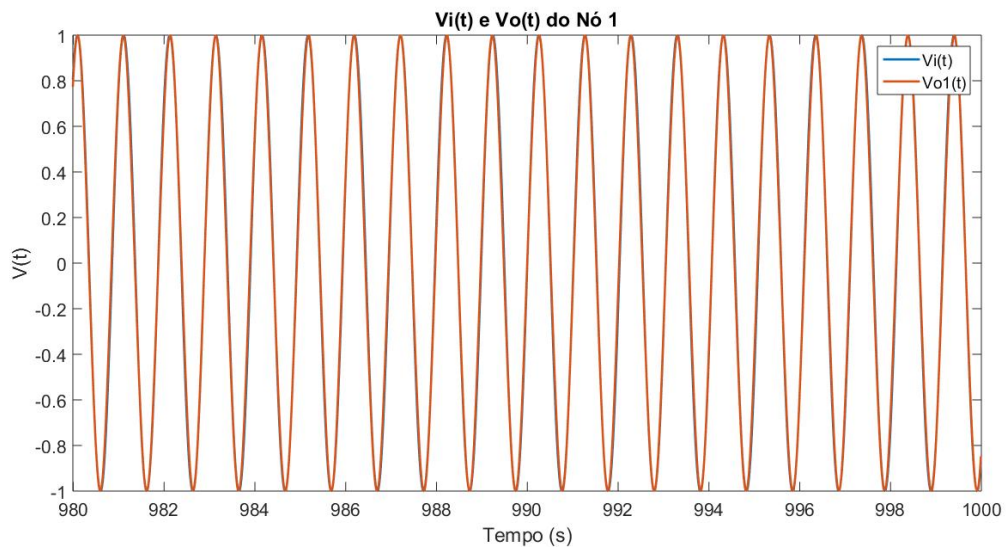
Fonte: Autor.

Figura 4.42: Resposta em regime do Nó Escravo 1 - Rampa unitária com o Filtro 5



Fonte: Autor.

Figura 4.43: Resposta em regime do Nó Escravo 1 - Rampa unitária com o Filtro 6



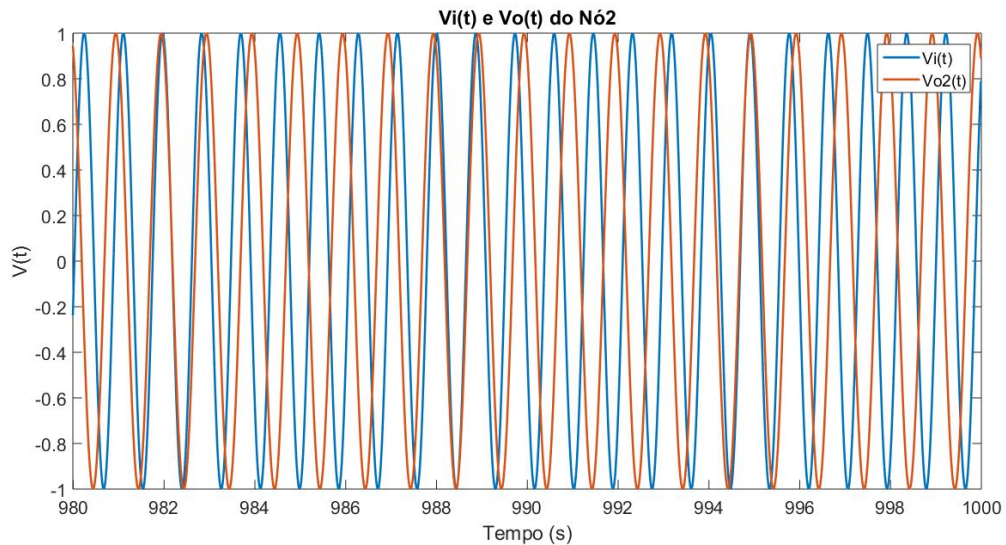
Fonte: Autor.

Ao observarmos as respostas dos filtros para a rampa, representados pelas Figuras 4.38, 4.39, 4.40, 4.41 e 4.42 notamos periodicidade entre o sinal de saída do Nó Mestre com o Nó Escravo 1 para cada filtro. Na Figura 4.43, há um particularidade no comportamento do funcionamento do filtro, em que há o sincronismo de fase entre

o nó escravo e o nó mestre, sendo tal efeito consequência da dinâmica do filtro.

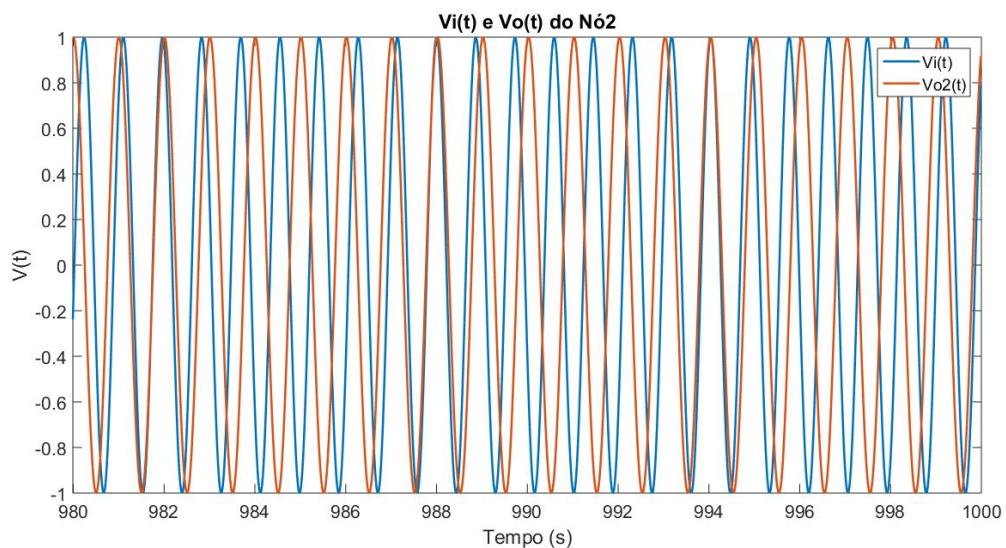
Para o Nó Escravo 2, as Figuras 4.44, 4.45, 4.46, 4.47, 4.48 e 4.49 apresentam o comportamento da saída do respectivo nó em relação ao sinal gerado pelo Nó Mestre, de acordo com cada filtro aplicado à rede.

Figura 4.44: Resposta em regime do Nó Escravo 2 - Rampa unitária com o Filtro 1



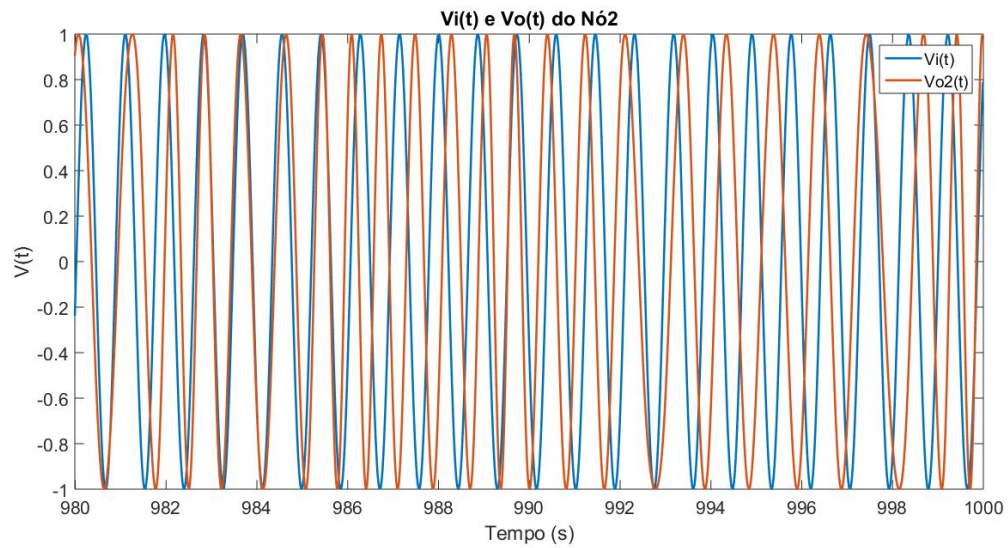
Fonte: Autor.

Figura 4.45: Resposta em regime do Nó Escravo 2 - Rampa unitária com o Filtro 2



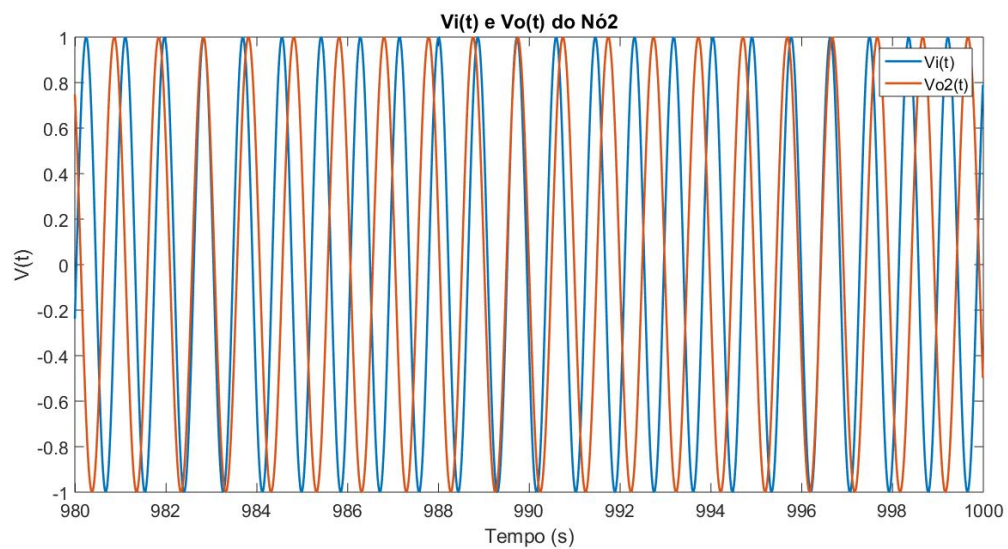
Fonte: Autor.

Figura 4.46: Resposta em regime do Nó Escravo 2 - Rampa unitária com o Filtro 3



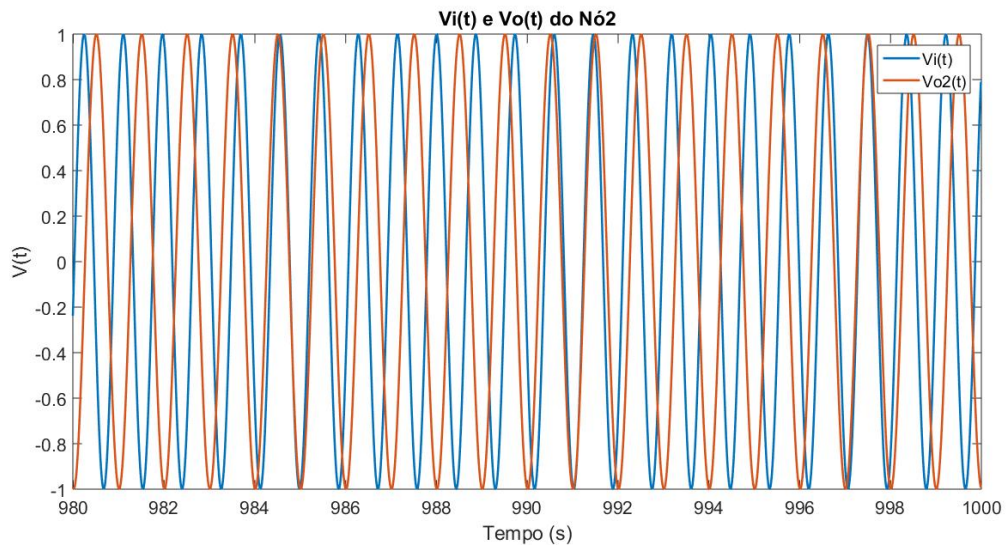
Fonte: Autor.

Figura 4.47: Resposta em regime do Nó Escravo 2 - Rampa unitária com o Filtro 4



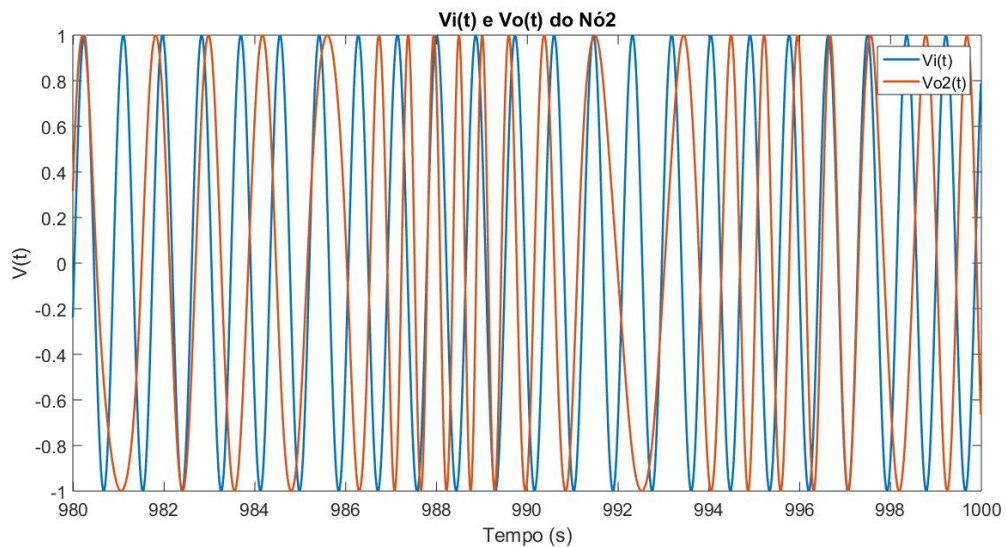
Fonte: Autor.

Figura 4.48: Resposta em regime do Nó Escravo 2 - Rampa unitária com o Filtro 5



Fonte: Autor.

Figura 4.49: Resposta em regime do Nó Escravo 2 - Rampa unitária com o Filtro 6



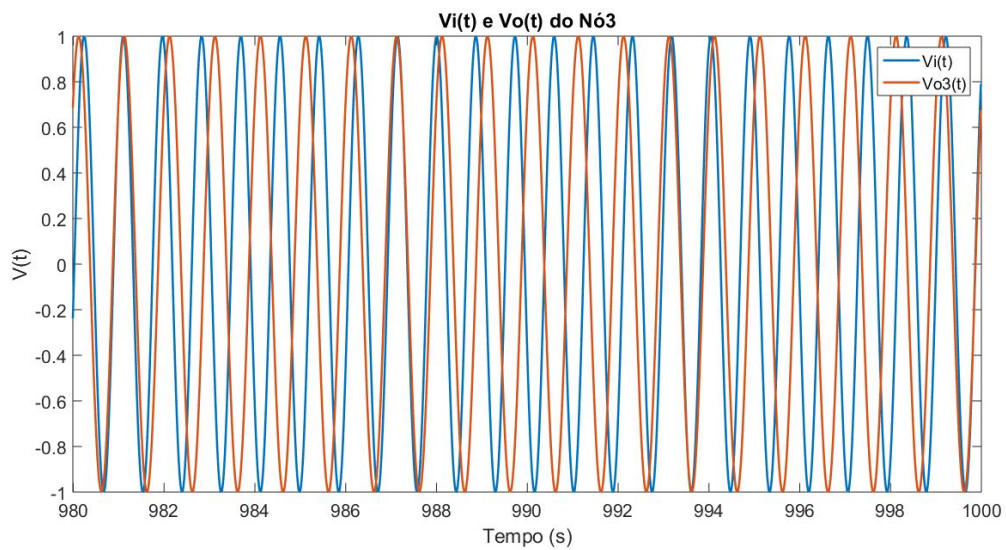
Fonte: Autor.

Ao observarmos as Figuras 4.44, 4.45, 4.46 e 4.49, notamos que não há um padrão o sinal de saída do Nó Escravo 2, tendo o sinal de $V_o(t)$ para esse nó um escorregamento de fase constante não padronizado. Todavia, ao observarmos os gráficos 4.47 e 4.48, gerados pela dinâmica dos Filtros 4 e 5, notamos reprodutibilidade e repetição, evi-

denciando o sincronismo. Nesse caso, será necessária a confirmação do estado síncrono por meio da observação do Erro de Fase e da tensão de controle $V_c(t)$.

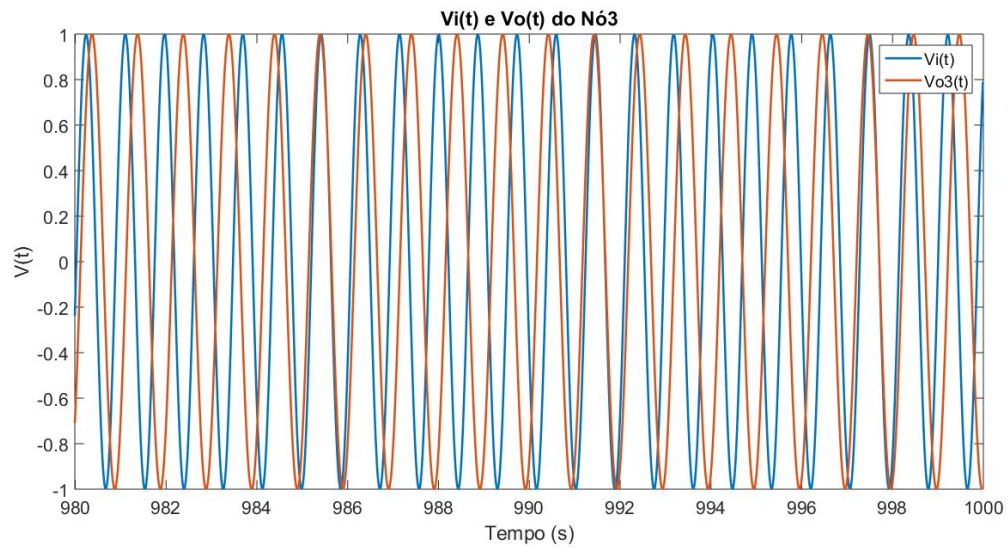
Para o Nó Escravo 3, as Figuras 4.50, 4.51, 4.52, 4.53, 4.54 e 4.55 apresentam o comportamento da saída do respectivo nó em relação ao sinal gerado pelo Nó Mestre, de acordo com cada filtro aplicado à rede.

Figura 4.50: Resposta em regime do Nó Escravo 3 - Rampa unitária com o Filtro 1



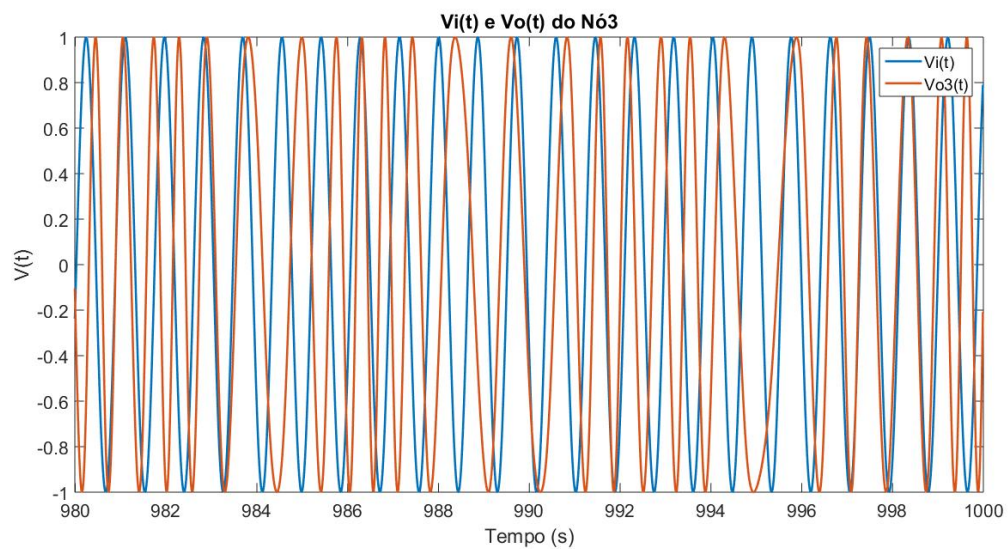
Fonte: Autor.

Figura 4.51: Resposta em regime do Nó Escravo 3 - Rampa unitária com o Filtro 2



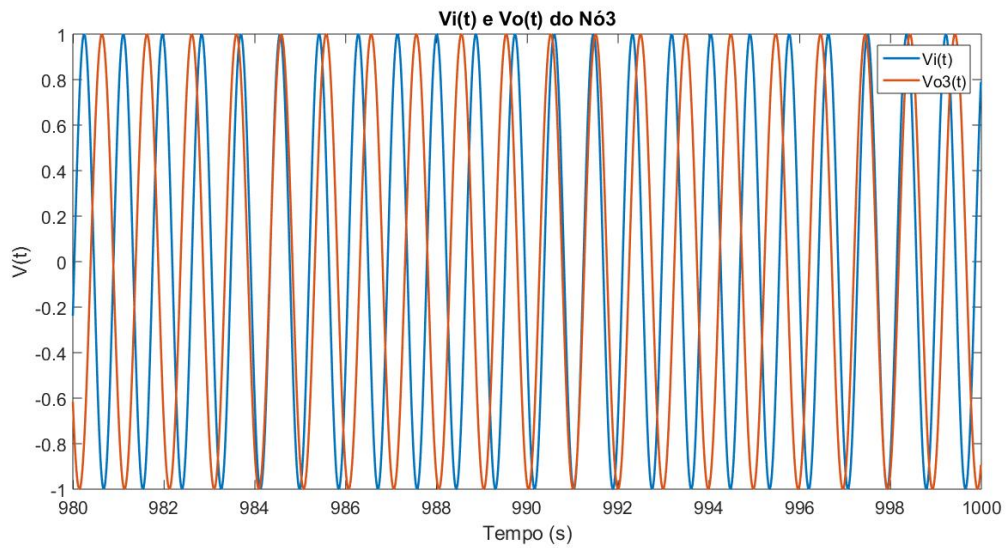
Fonte: Autor.

Figura 4.52: Resposta em regime do Nó Escravo 3 - Rampa unitária com o Filtro 3



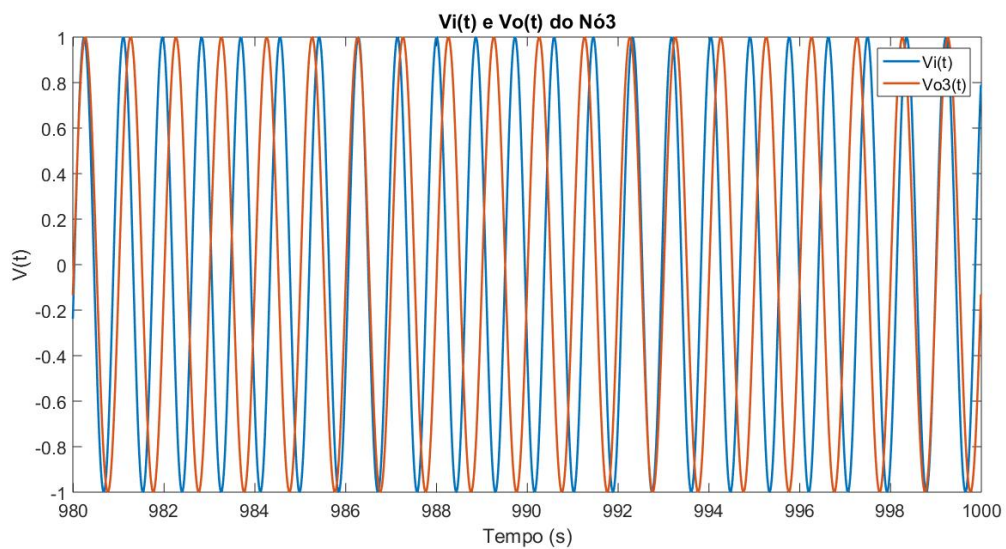
Fonte: Autor.

Figura 4.53: Resposta em regime do Nó Escravo 3 - Rampa unitária com o Filtro 4



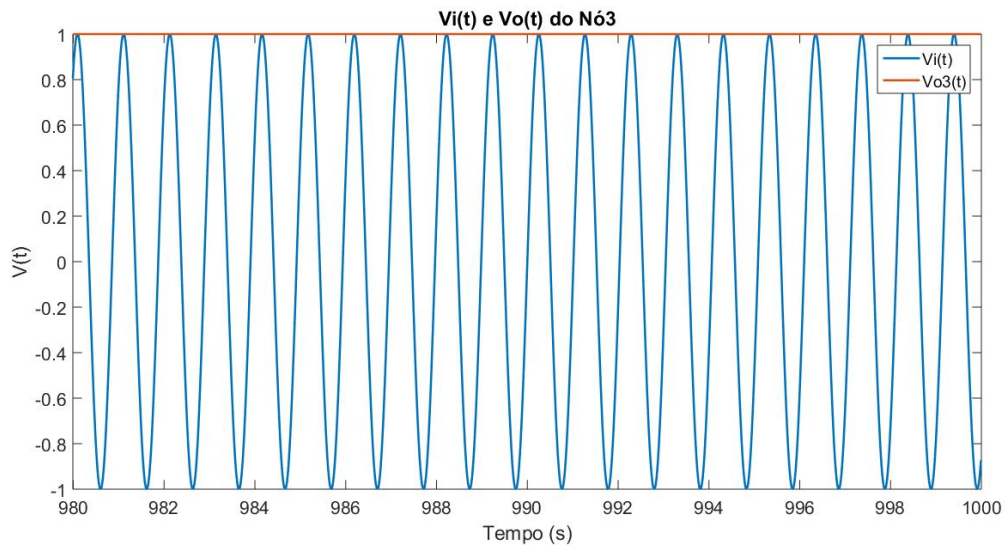
Fonte: Autor.

Figura 4.54: Resposta em regime do Nó Escravo 3 - Rampa unitária com o Filtro 5



Fonte: Autor.

Figura 4.55: Resposta em regime do Nó Escravo 3 - Rampa unitária com o Filtro 6

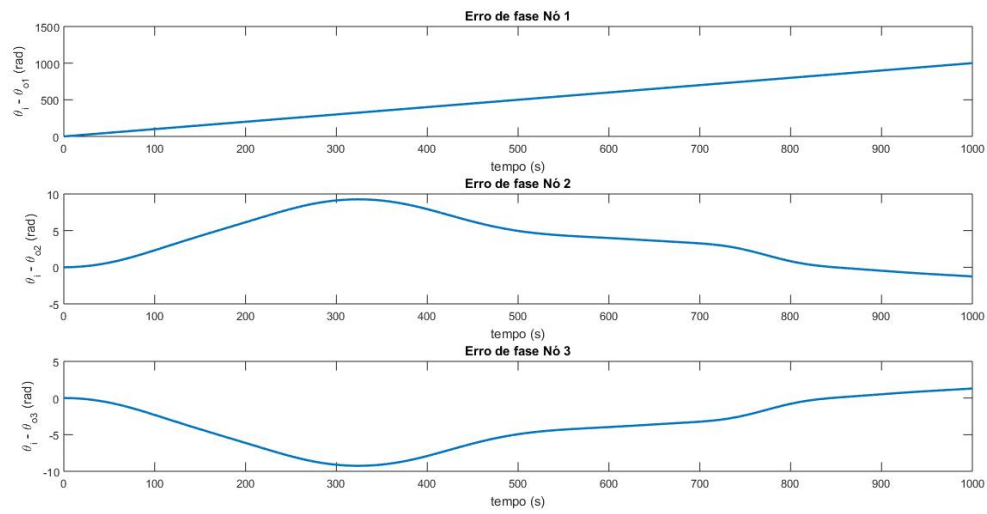


Fonte: Autor.

O sinal de entrada do Nó Escravo 3 depende diretamente do sinal de saída do Nó Escravo 2. Ao observarmos as Figuras 4.44, 4.45, 4.46 e 4.49, notamos que, como vimos esses gráficos para o Nó Escravo 2 anteriormente, não há um padrão o sinal de saída do Nó Escravo 3, tendo o sinal de $V_o(t)$ para esse nó um escorregamento de fase constante não padronizado. Observando os gráficos das Figuras 4.47 e 4.48, notamos que os Filtros 4 e 5 possuem periodicidade, indicando que esse nó possui sincronismo em relação ao sinal gerado pelo Nó Mestre. Tal evidencia será confirmada ou não ao observarmos os gráficos de Erro de fase e $V_c(t)$ posteriormente.

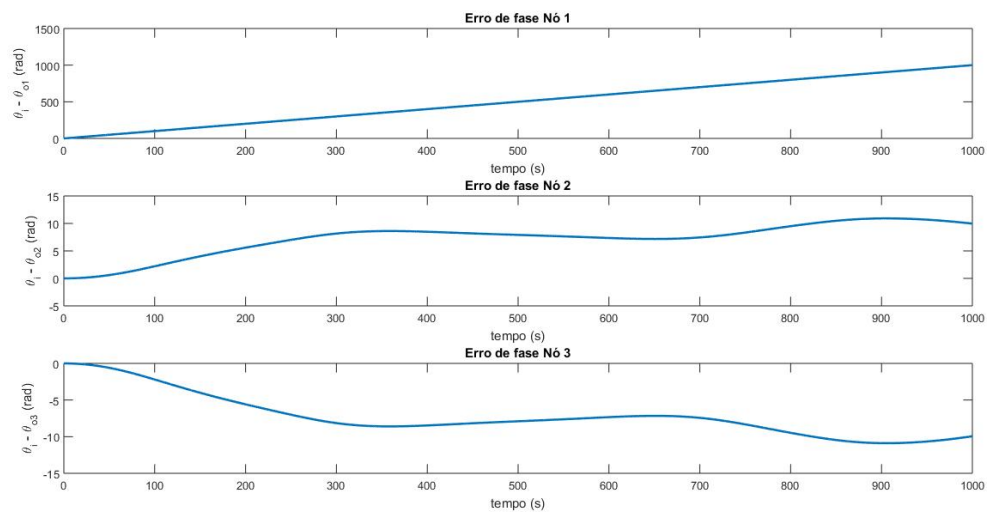
As Figuras 4.56, 4.57, 4.58, 4.59, 4.60 e 4.61 apresentam o erro de fase do sinal de saída dos nós escravos em relação à sua entrada. No caso dos Nós Escravos 1 e 3, o valor de $\theta_i(t)$ está relacionado à $\theta_o(t)$ do Nó Mestre e do Nó 2, respectivamente, enquanto o Nó 2 tem sua fase de acordo com a composição de $\theta_o(t)$ do Nó 1 e 3.

Figura 4.56: Erro de fase da rede TWMS - Rampa unitária com o Filtro 1



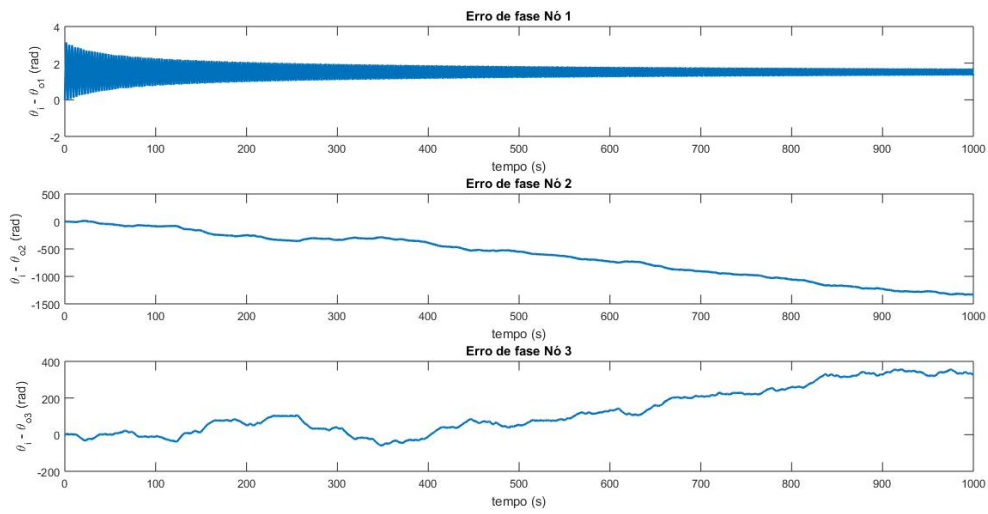
Fonte: Autor.

Figura 4.57: Erro de fase da rede TWMS - Rampa unitária com o Filtro 2



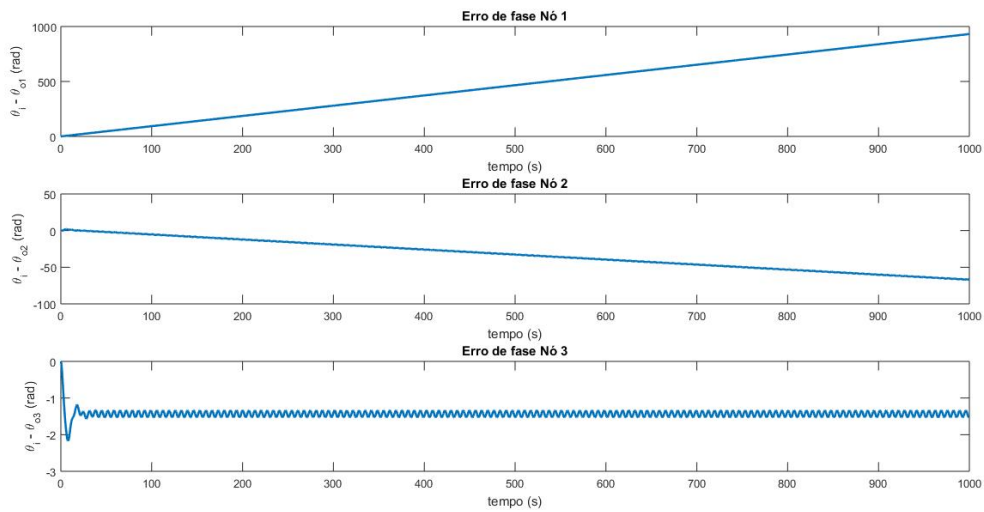
Fonte: Autor.

Figura 4.58: Erro de fase da rede TWMS - Rampa unitária com o Filtro 3



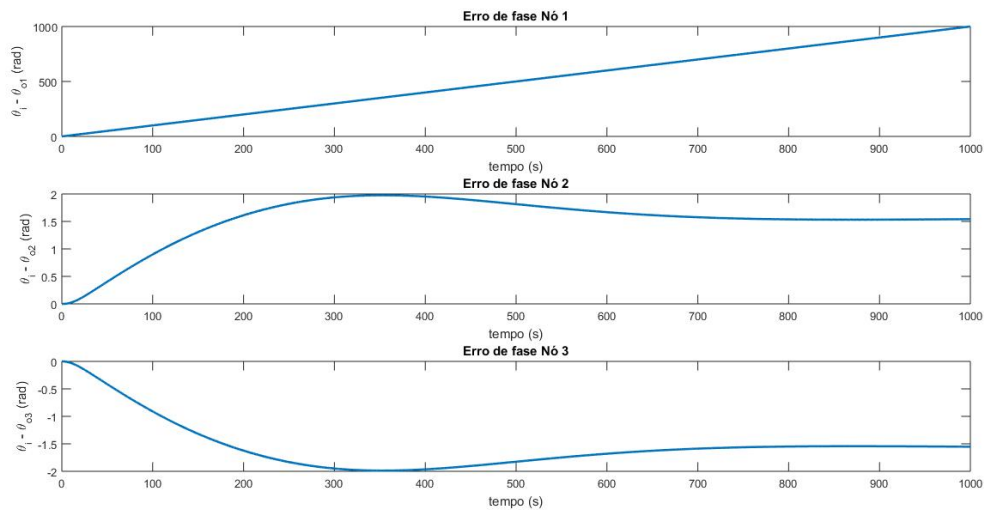
Fonte: Autor.

Figura 4.59: Erro de fase da rede TWMS - Rampa unitária com o Filtro 4



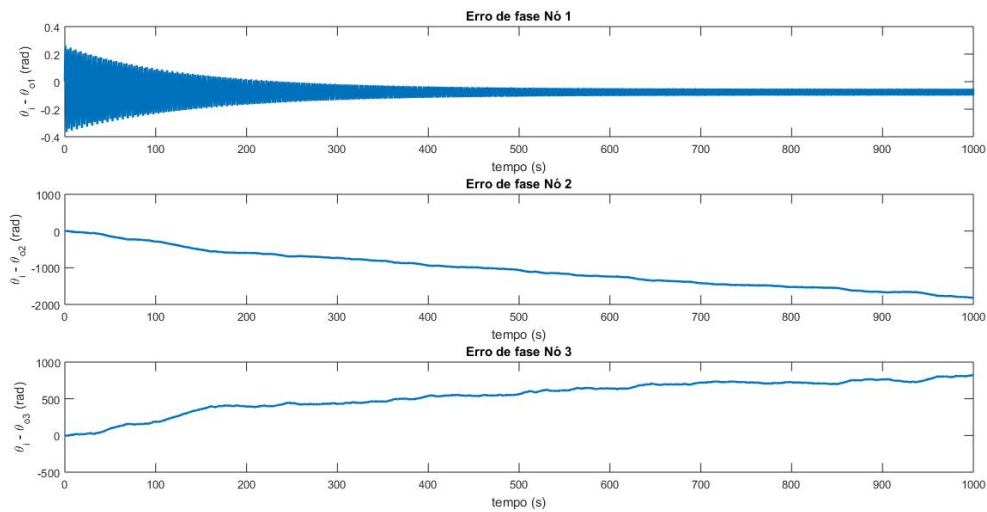
Fonte: Autor.

Figura 4.60: Erro de fase da rede TWMS - Rampa unitária com o Filtro 5



Fonte: Autor.

Figura 4.61: Erro de fase da rede TWMS - Rampa unitária com o Filtro 6



Fonte: Autor.

Ao observarmos os primeiros gráficos das Figuras 4.56, 4.57, 4.59 e 4.60, notamos que a fase do sinal do Nó Escravo 1 segue a a rampa unitária de fase aplicada à fase do Nó Mestre, apresentando sincronismo em sua saída. Nas Figuras 4.58 e 4.61, notamos que o erro de fase se estabiliza na região próxima de $\pi/2$, ou seja, há o sincronismo por

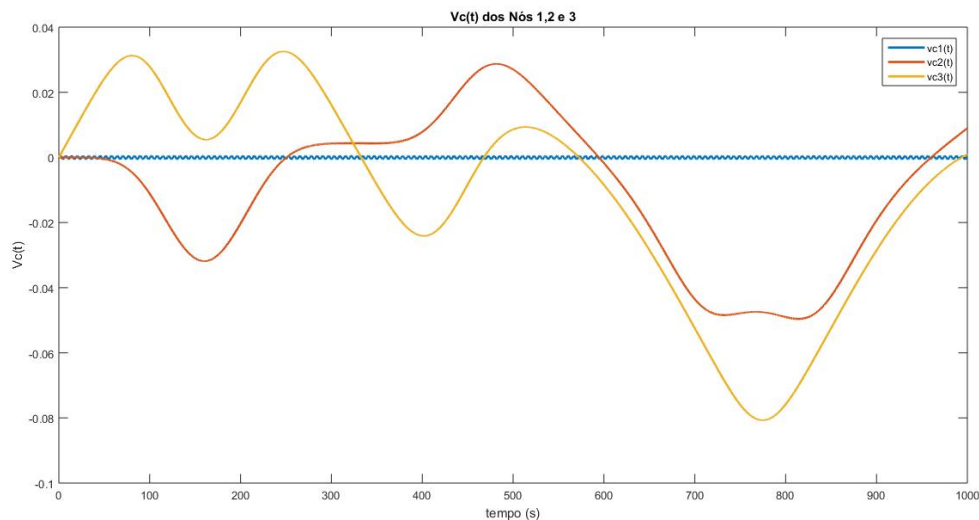
inversão de sinal.

Nos segundos e terceiros gráficos das Figuras 4.56, 4.57, 4.58 e 4.61 não notamos estabilização em valor fixo ou a réplica da rampa da rampa de entrada, sendo necessária a observação de $V_c(t)$ dos Nós Escravos 2 e 3 para confirmar ou não o sincronismo.

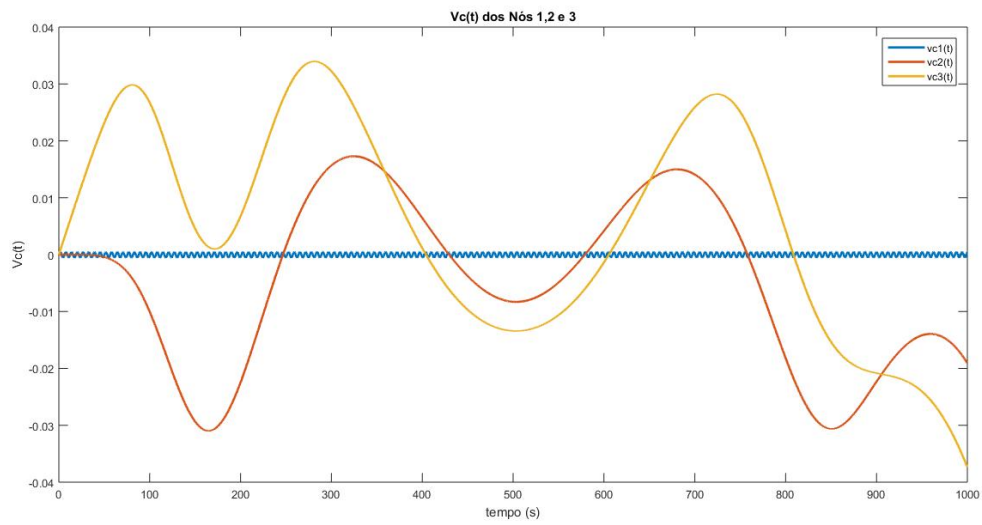
No segundo gráfico da Figura 4.59 vemos uma rampa decrescente do Erro de Fase do Nó Escravo 2 para o Filtro 4, indicando que tal nó está sincronizado com o deslocamento em 180° em relação à fase do Nó Escravo 1. Ao observarmos o terceiro gráfico e o segundo e terceiro gráfico das Figuras 4.59 e 4.60 respectivamente, notamos a estabilização do erro de fase, evidenciando o sincronismo entre o sinal de entrada do Nó 2 e o próprio nó e também entre a saída do Nó 2 com a saída do Nó 3.

As Figuras 4.62, 4.63, 4.64, 4.65, 4.66 e 4.67 apresentam o sinal de controle $V_c(t)$ da rede TWMS de acordo com os filtros utilizados.

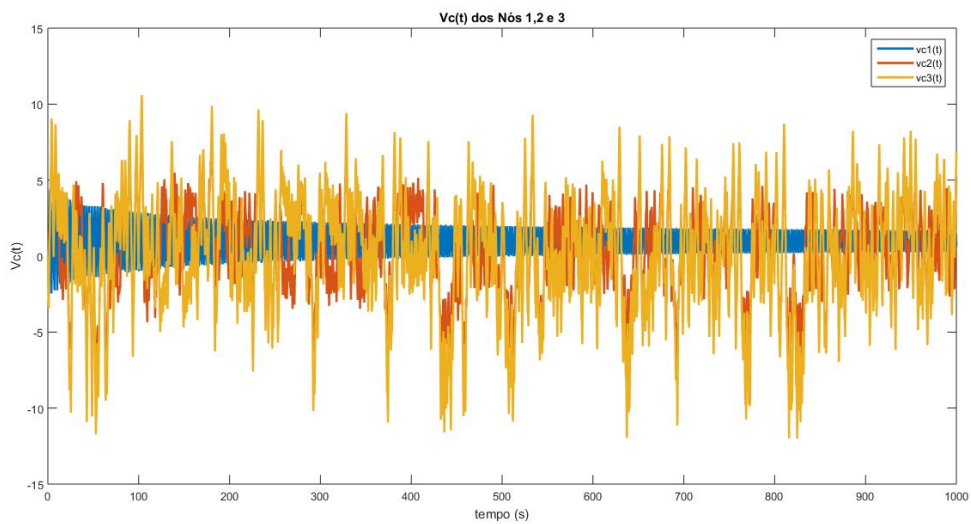
Figura 4.62: $V_c(t)$ para cada nó da rede TWMS - Rampa unitária com o Filtro 1



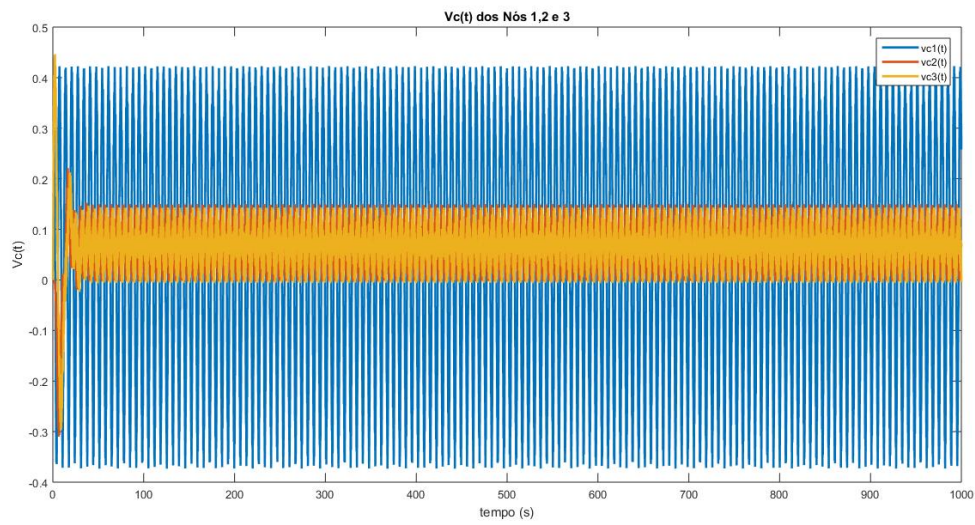
Fonte: Autor.

Figura 4.63: $V_c(t)$ para cada nó da rede TWMS - Rampa unitária com o Filtro 2

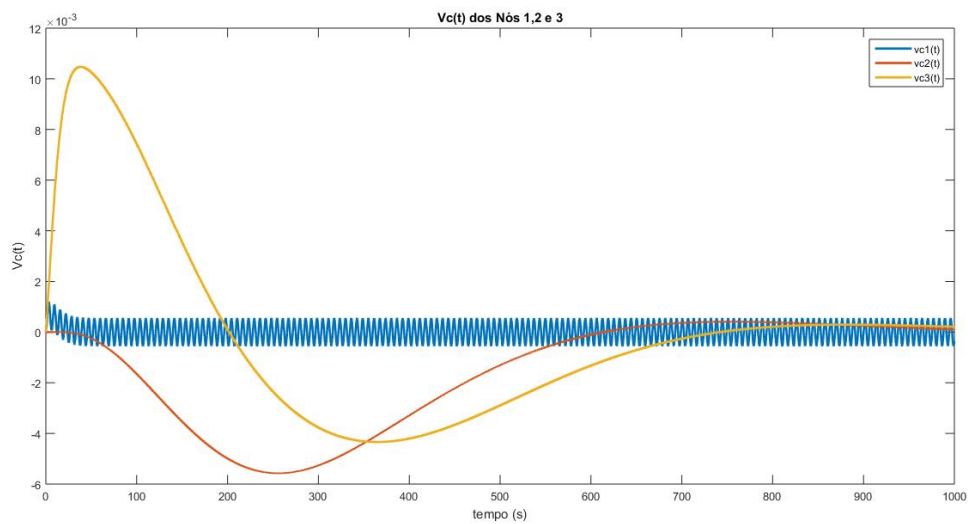
Fonte: Autor.

Figura 4.64: $V_c(t)$ para cada nó da rede TWMS - Rampa unitária com o Filtro 3

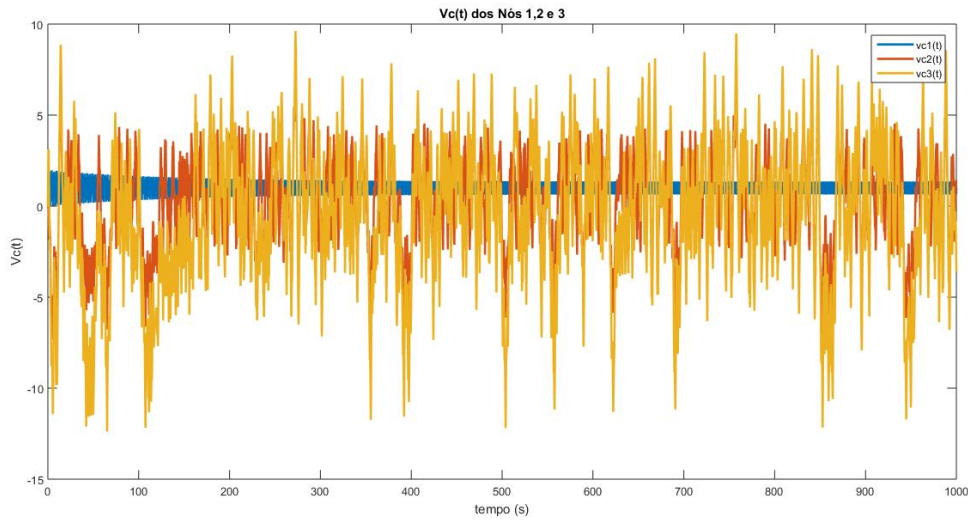
Fonte: Autor.

Figura 4.65: $V_c(t)$ para cada nó da rede TWMS - Rampa unitária com o Filtro 4

Fonte: Autor.

Figura 4.66: $V_c(t)$ para cada nó da rede TWMS - Rampa unitária com o Filtro 5

Fonte: Autor.

Figura 4.67: $V_c(t)$ para cada nó da rede TWMS - Rampa unitária com o Filtro 6

Fonte: Autor.

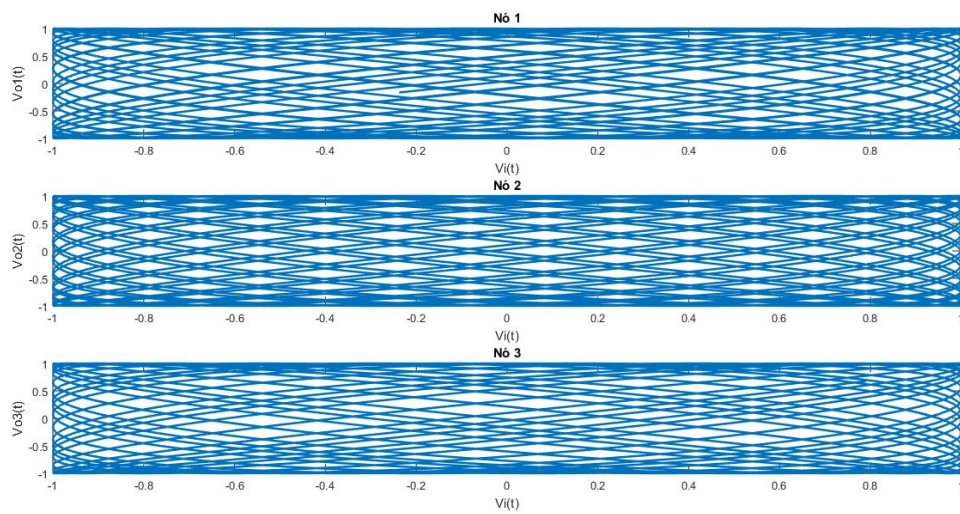
Ao observarmos os valores de $V_c(t)$ para o Nó Escravo 1 nas Figuras 4.62, 4.63, 4.64, 4.65, 4.66 e 4.67, notamos a oscilação constante com média zero, indicando o sincronismo desse nó em relação à sua entrada, independente do Filtro aplicado.

Para os valores de $V_c(t)$ para os Nós Escravos 2 e 3, nos gráficos das Figuras 4.62, 4.63, 4.64 e 4.67 não observamos uma observação constante com média zero, indicando que esses nós não alcançam o sincronismo com a utilização dos Filtros 1, 2, 3 e 6.

Analisando as curvas de $V_c(t)$ para os os Nós Escravos 2 e 3, notamos que possuem oscilação constante com média zero, indicando o sincronismo dessa rede com a aplicação dos Filtros 4 e 5.

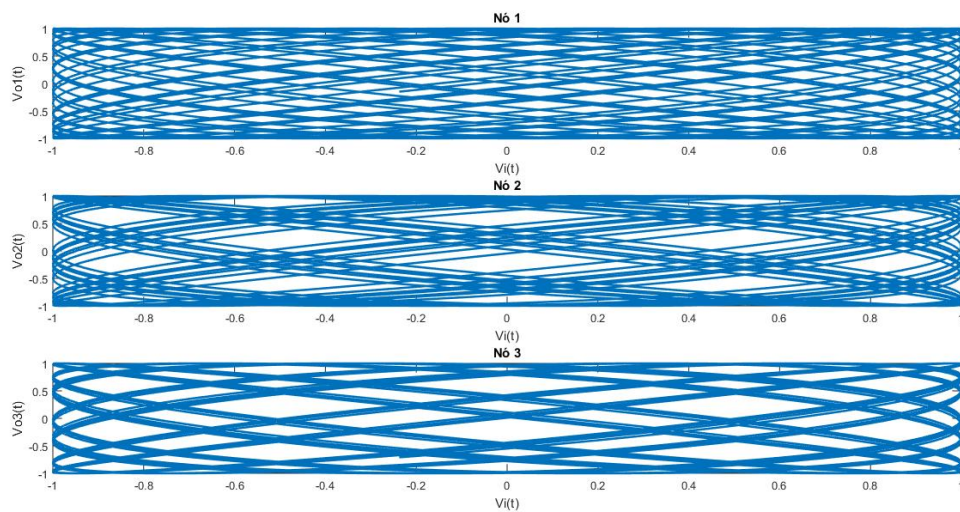
Por fim, as Figuras 4.68, 4.69, 4.70, 4.71, 4.72 e 4.73 apresentam as Figuras de Lissajous, conforme [44, 46].

Figura 4.68: Lissajous para cada nó da rede TWMS - Rampa unitária com o Filtro 1



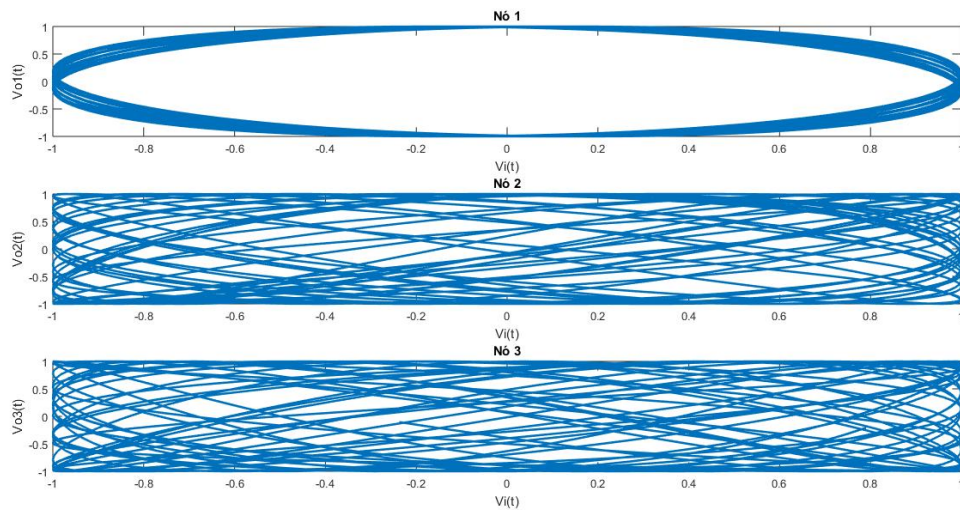
Fonte: Autor.

Figura 4.69: Lissajous para cada nó da rede TWMS - Rampa unitária com o Filtro 2



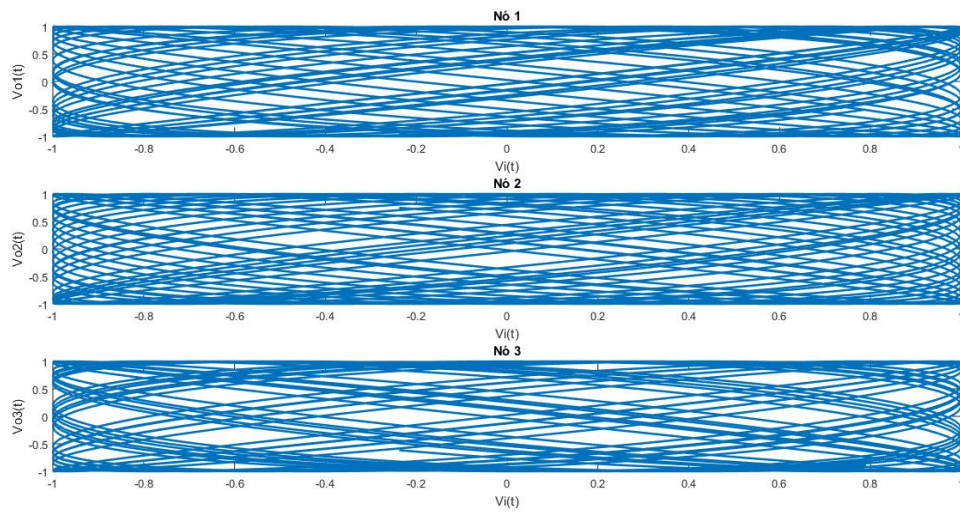
Fonte: Autor.

Figura 4.70: Lissajous para cada nó da rede TWMS - Rampa unitária com o Filtro 3



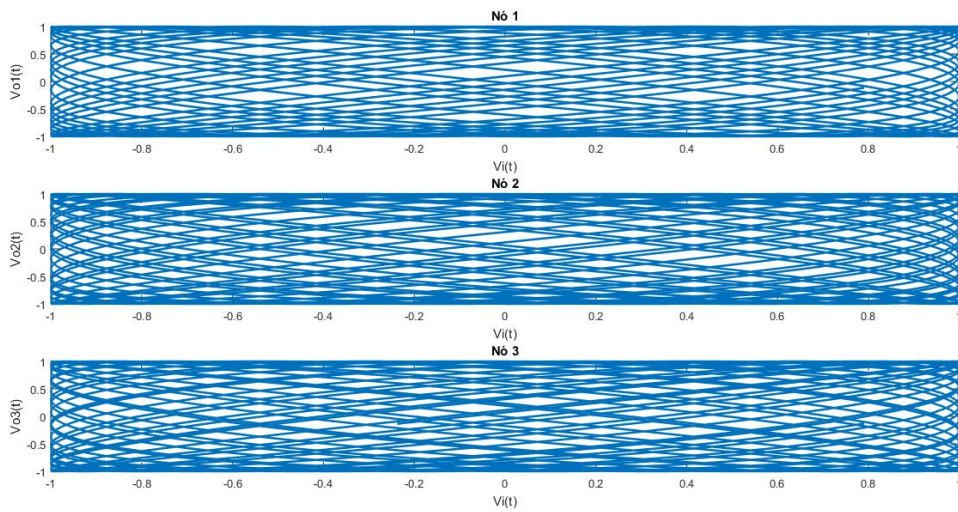
Fonte: Autor.

Figura 4.71: Lissajous para cada nó da rede TWMS - Rampa unitária com o Filtro 4



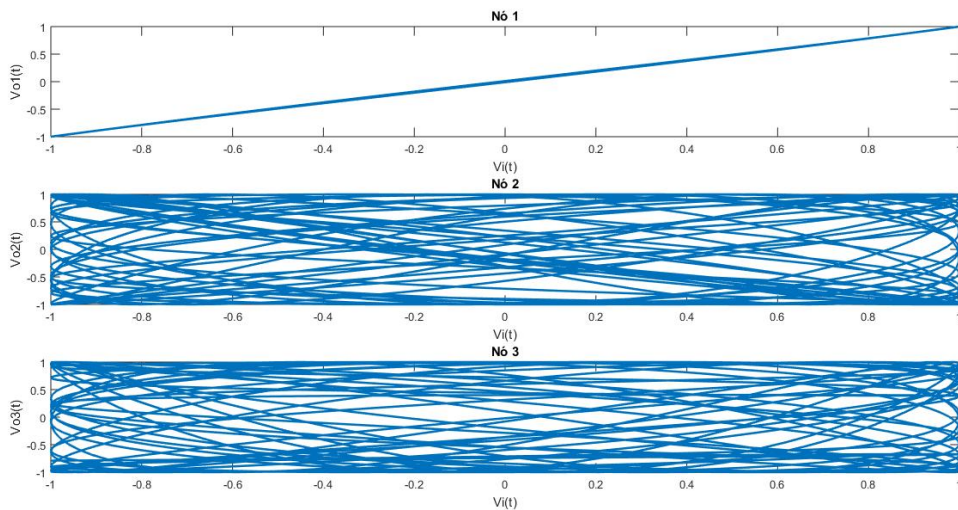
Fonte: Autor.

Figura 4.72: Lissajous para cada nó da rede TWMS - Rampa unitária com o Filtro 5



Fonte: Autor.

Figura 4.73: Lissajous para cada nó da rede TWMS - Rampa unitária com o Filtro 6



Fonte: Autor.

Ao se construir as Figuras de Lissajous, notamos que somente os primeiros gráficos das Figuras 4.70 e 4.73 apresentam formas conhecidas, no primeiro caso o sincronismo em quadratura do Nó Escravo 1 para o Filtro 3 e o sincronismo de fase do Nó Escravo 1 para o Filtro 6. Para os demais gráficos e também para as Figuras 4.68,

4.69, 4.71 e 4.72, a representação da Figura de Lissajous não é conclusiva quanto ao sincronismo e por isso, devemos considerar a análise feita anteriormente sob a ótica de $V_c(t)$ de cada Nó Escravo.

A Tabela 4.3 apresenta os tempos de simulação para a rede TWMS com os diferentes filtros para a versão sequencial e em paralelo do algoritmo proposto em um processador de dois núcleos.

Tabela 4.3: Tempos de simulação para a resposta à rampa de fase - Intel[®] Core[™] i3-4005U CPU @ 1.70GHz (dois núcleos)

Filtro	Sequencial (s)	Paralelo (s)
Filtro 1	10,23	18943,32
Filtro 2	10,26	19132,45
Filtro 3	10,83	18947,07
Filtro 4	10,30	19051,50
Filtro 5	137,66	20037,13
Filtro 6	137,65	19968,79

A Tabela 4.4 apresenta os tempos de simulação para a rede TWMS com os diferentes filtros para a versão sequencial e em paralelo do algoritmo proposto em um processador de quatro núcleos.

Tabela 4.4: Tempos de simulação para a resposta à rampa de fase - Intel[®] Core[™] i5-4440S CPU @ 2.8GHz (quatro núcleos)

Filtro	Sequencial (s)	Paralelo (s)
Filtro 1	2,43	6646,86
Filtro 2	2,44	6601,99
Filtro 3	2,43	6641,31
Filtro 4	2,33	6683,47
Filtro 5	156,92	7287,03
Filtro 6	156,67	7273,64

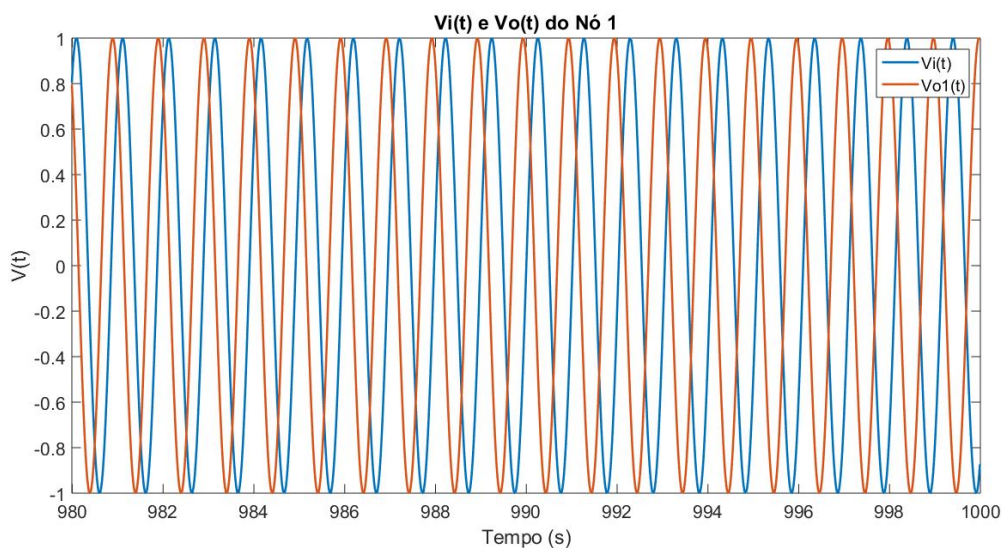
Ao observarmos os resultados de tempo de simulação, apresentados pelas Tabelas 4.3 e 4.4, não observamos variações de grandezas nos resultados tanto para lógica sequencial quanto para a paralela do que já foi apresentado nas Tabelas 4.1 e 4.2 para

o degrau de fase. Tais resultados confirmam que independente do tipo de sinal de entrada, o sistema possui um comportamento e tempo de processamento semelhante.

4.3 Resposta ao dente de serra

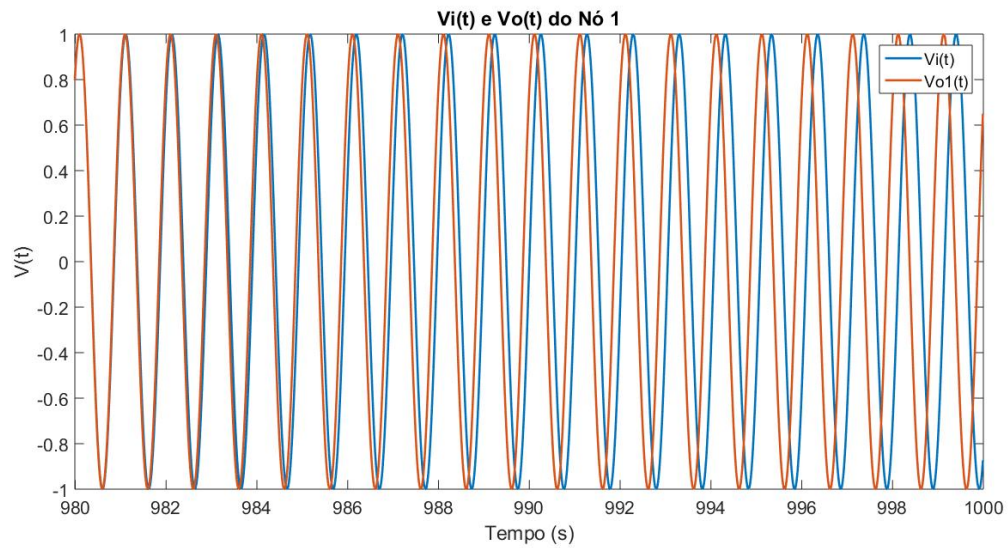
Para esse tipo de perturbação de entrada na fase do Nó Mestre, foram observados os seguintes comportamentos para no Nó Escravo 1, conforme apresentados nas Figuras 4.74, 4.75, 4.76, 4.77, 4.78 e 4.79, onde cada gráfico representa o comportamento do nó de acordo com a característica e função transferência do filtro, conforme apresentado em 3.1.

Figura 4.74: Resposta em regime do Nó Escravo 1 - Dente de Serra com o Filtro 1



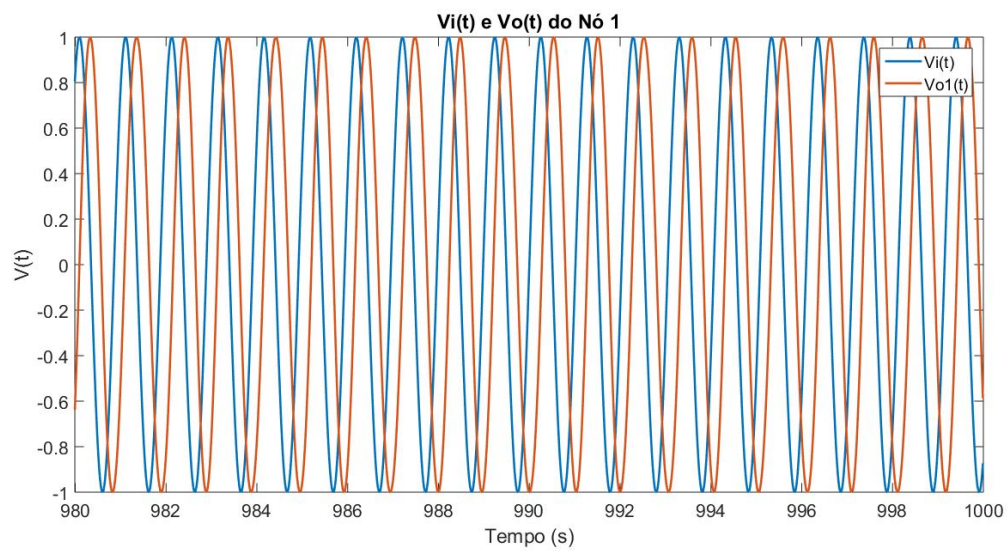
Fonte: Autor.

Figura 4.75: Resposta em regime do Nó Escravo 1 - Dente de Serra com o Filtro 2



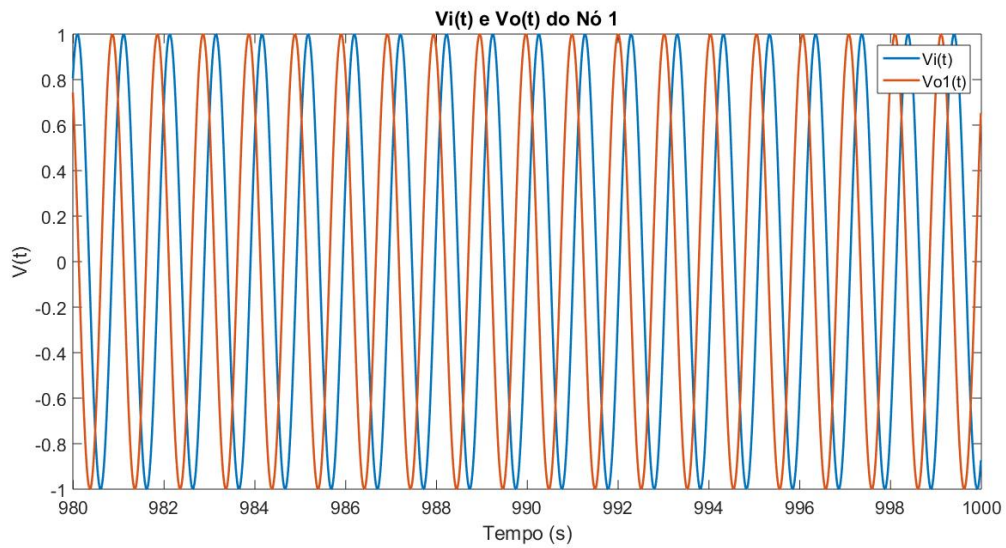
Fonte: Autor.

Figura 4.76: Resposta em regime do Nó Escravo 1 - Dente de Serra com o Filtro 3



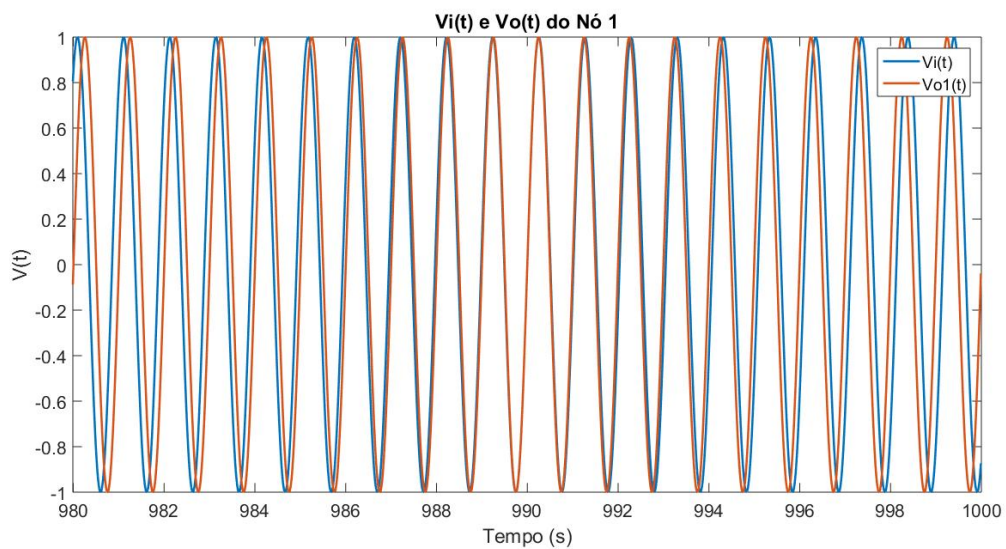
Fonte: Autor.

Figura 4.77: Resposta em regime do Nó Escravo 1 - Dente de Serra com o Filtro 4



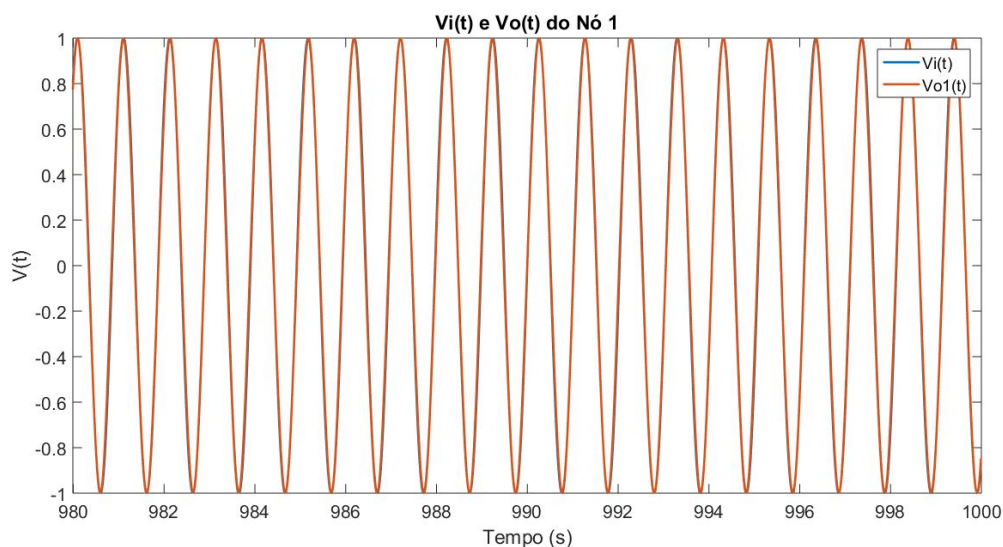
Fonte: Autor.

Figura 4.78: Resposta em regime do Nó Escravo 1 - Dente de Serra com o Filtro 5



Fonte: Autor.

Figura 4.79: Resposta em regime do Nó Escravo 1 - Dente de Serra com o Filtro 6

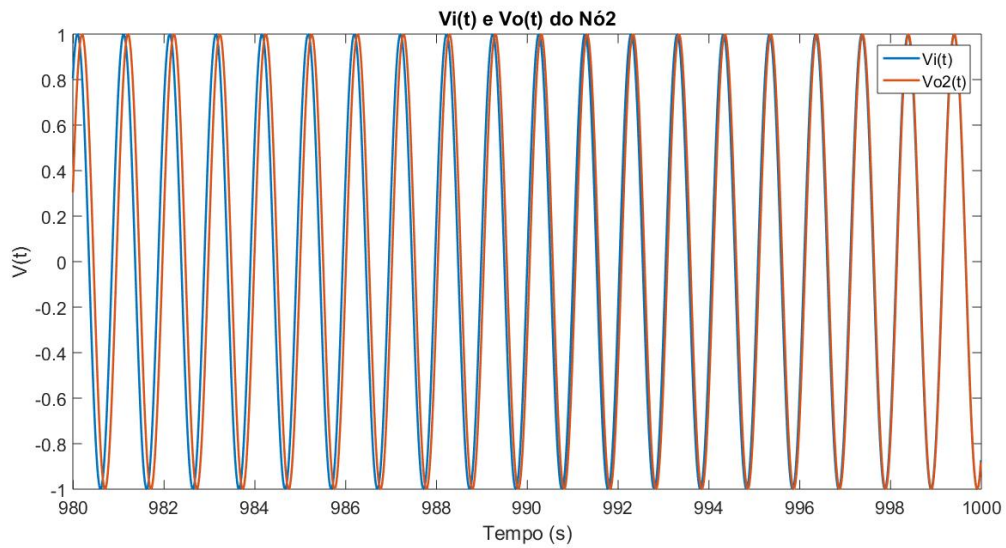


Fonte: Autor.

Ao observarmos as Figuras 4.74, 4.75, 4.77 e 4.78 um escorregamento na fase do sinal do Nó Escravo 1 em relação à saída do Nó Mestre e, para o período analisado, não é conclusivo, necessitando a observação do erro de fase e de $V_c(t)$ no período todo para sinalizar sincronismo ou não do nó. Para as Figuras 4.76 e 4.79, notamos que há o sincronismo entre o sinal de saída do nó em relação ao Nó Mestre. No caso do Filtro 6, podemos considerar que este sincronismo é de fase devido a sobreposição das curvas no gráfico.

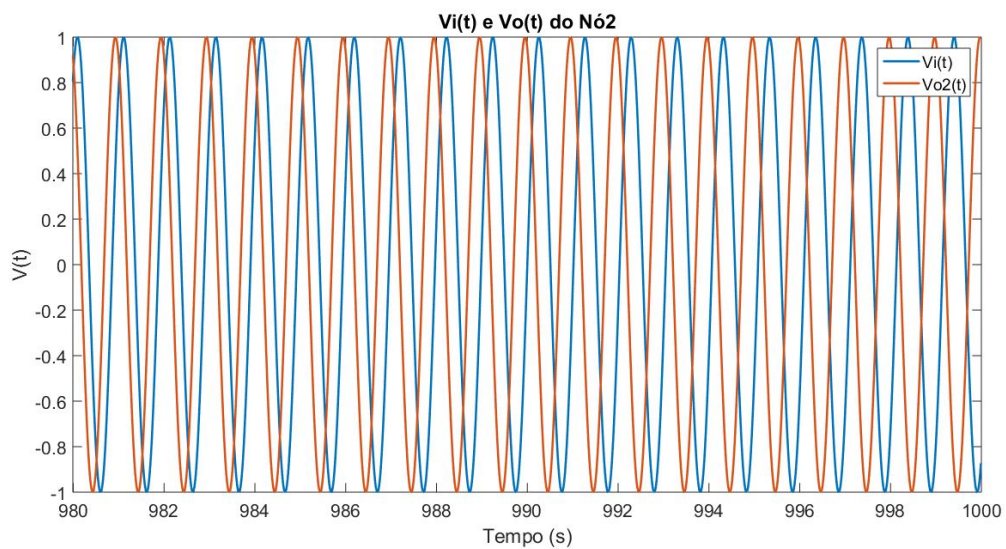
Para o Nó Escravo 2, as Figuras 4.80, 4.81, 4.82, 4.83, 4.84 e 4.85 apresentam o comportamento da saída do respectivo nó em relação ao sinal gerado pelo Nó Mestre, de acordo com cada filtro aplicado à rede.

Figura 4.80: Resposta em regime do Nó Escravo 2 - Dente de Serra com o Filtro 1



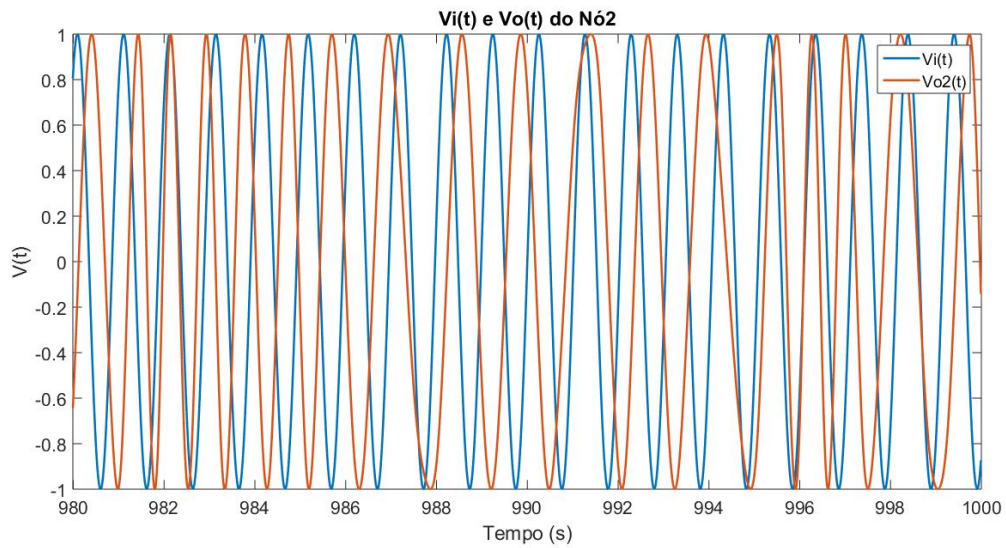
Fonte: Autor.

Figura 4.81: Resposta em regime do Nó Escravo 2 - Dente de Serra com o Filtro 2



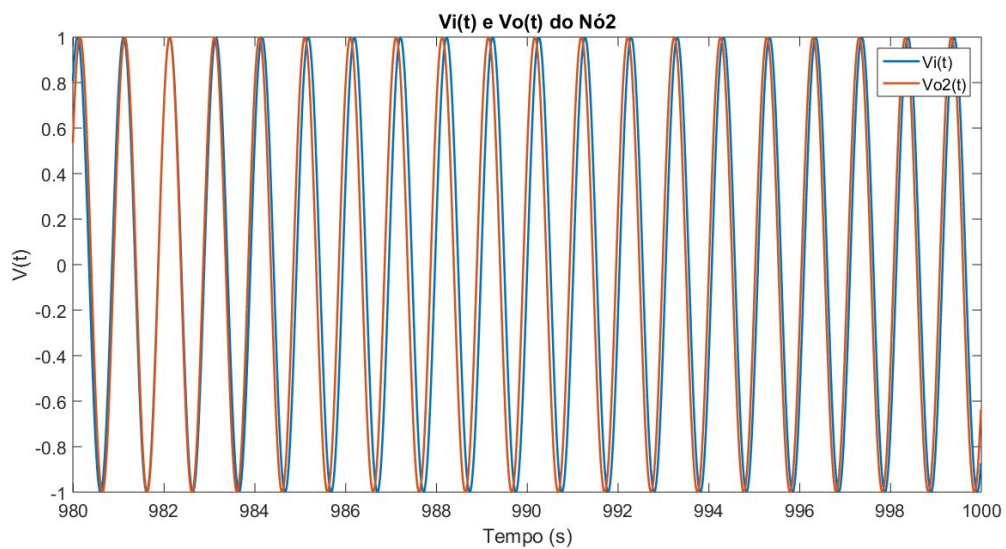
Fonte: Autor.

Figura 4.82: Resposta em regime do Nó Escravo 2 - Dente de Serra com o Filtro 3



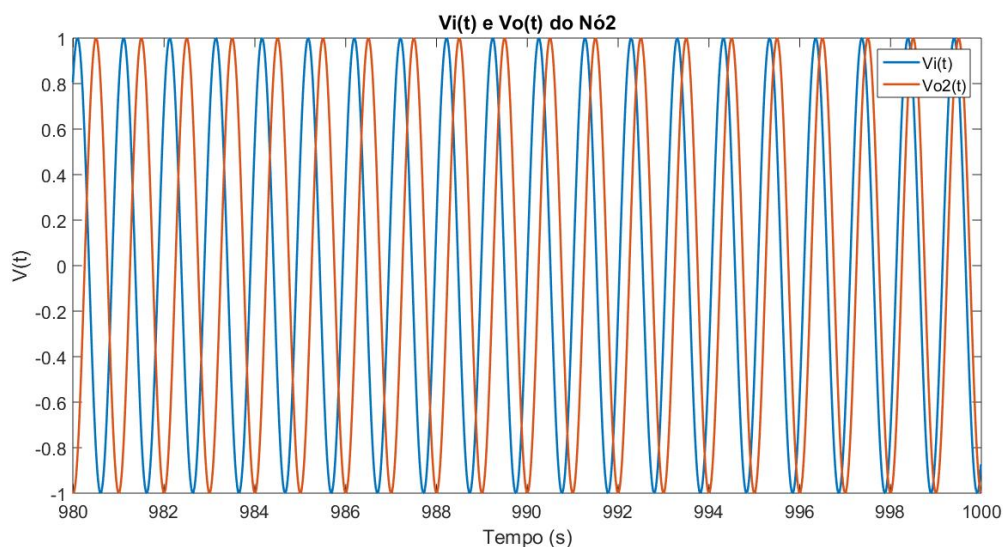
Fonte: Autor.

Figura 4.83: Resposta em regime do Nó Escravo 2 - Dente de Serra com o Filtro 4



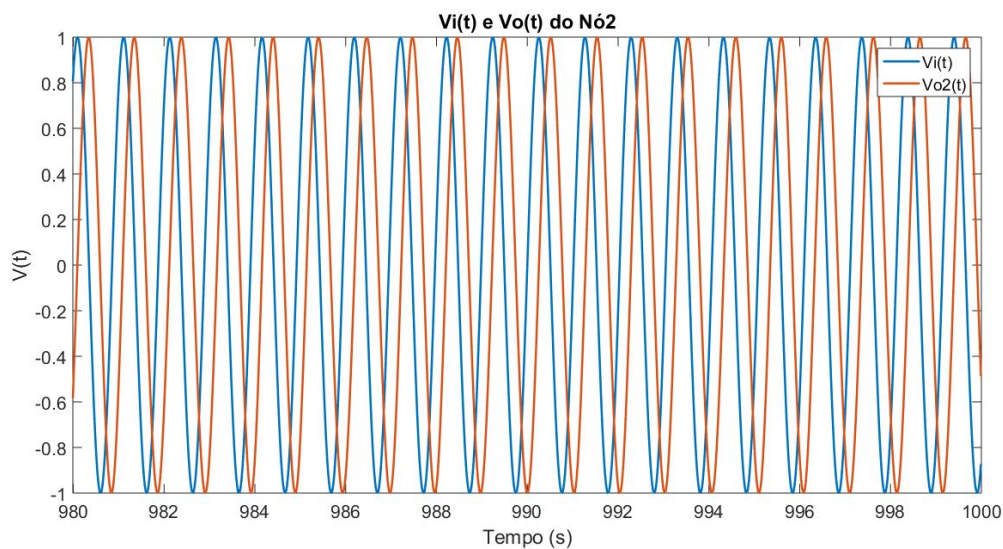
Fonte: Autor.

Figura 4.84: Resposta em regime do Nó Escravo 2 - Dente de Serra com o Filtro 5



Fonte: Autor.

Figura 4.85: Resposta em regime do Nó Escravo 2 - Dente de Serra com o Filtro 6

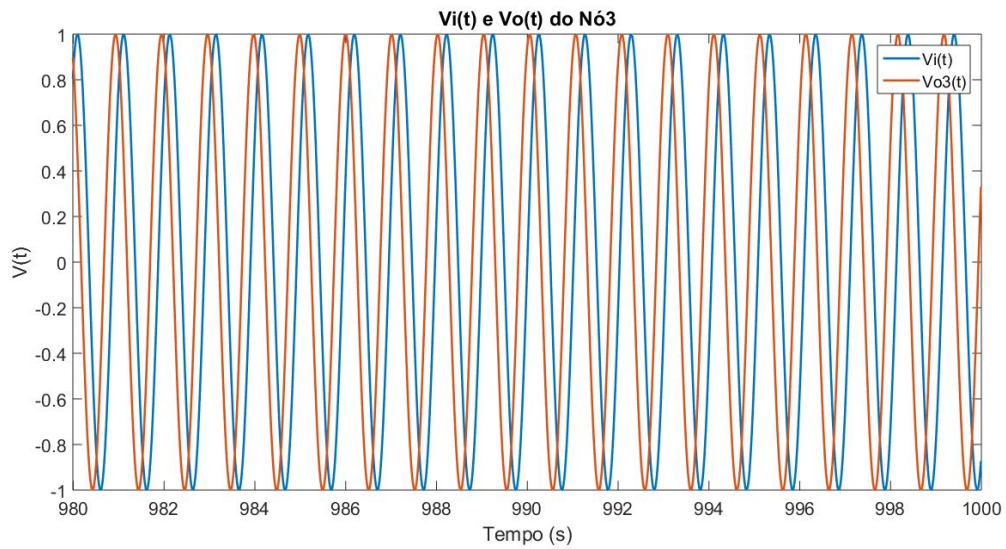


Fonte: Autor.

Para o período analisado, notamos que os resultados apresentados nas Figuras 4.80, 4.81, 4.82, 4.83, 4.84 e 4.85 apresentam escorregamento de fase, sendo necessária a verificação do erro de fase e da tensão de controle do Nó 2 para podermos se o estado apresentado está em sincronismo ou não em relação ao sinal de entrada.

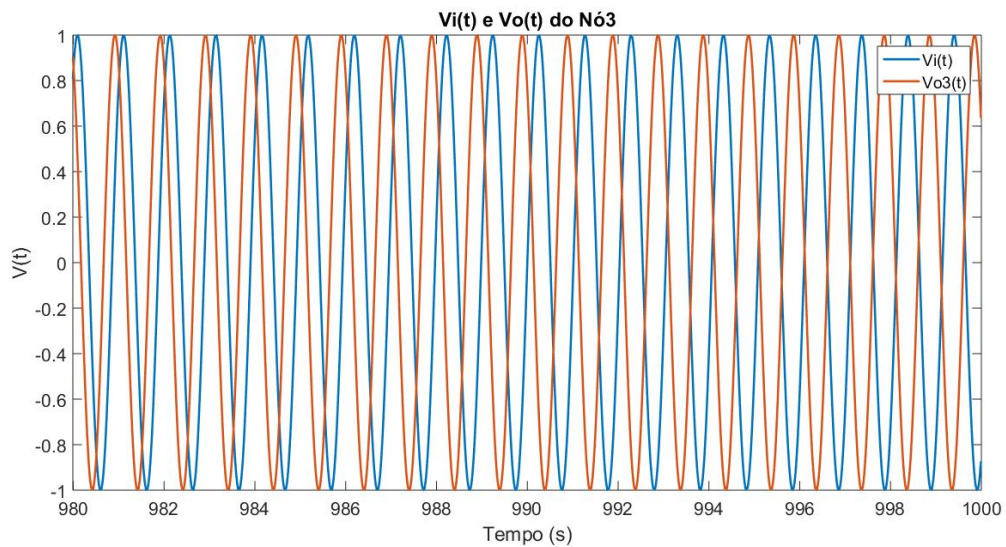
Para o Nó Escravo 3, as Figuras 4.86, 4.87, 4.88, 4.89, 4.90 e 4.91 apresentam o comportamento da saída do respectivo nó em relação ao sinal gerado pelo Nó Mestre, de acordo com cada filtro aplicado à rede.

Figura 4.86: Resposta em regime do Nó Escravo 3 - Dente de Serra com o Filtro 1



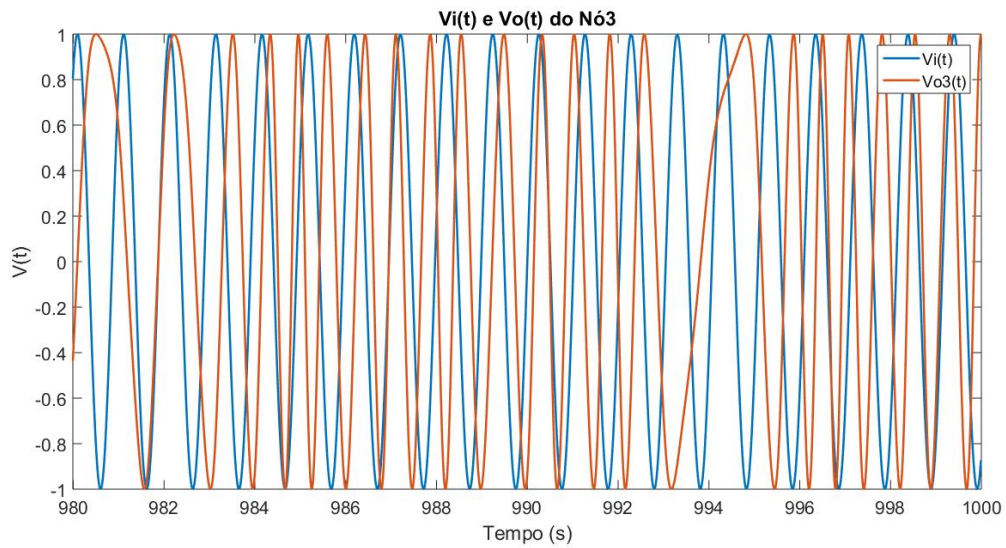
Fonte: Autor.

Figura 4.87: Resposta em regime do Nó Escravo 3 - Dente de Serra com o Filtro 2



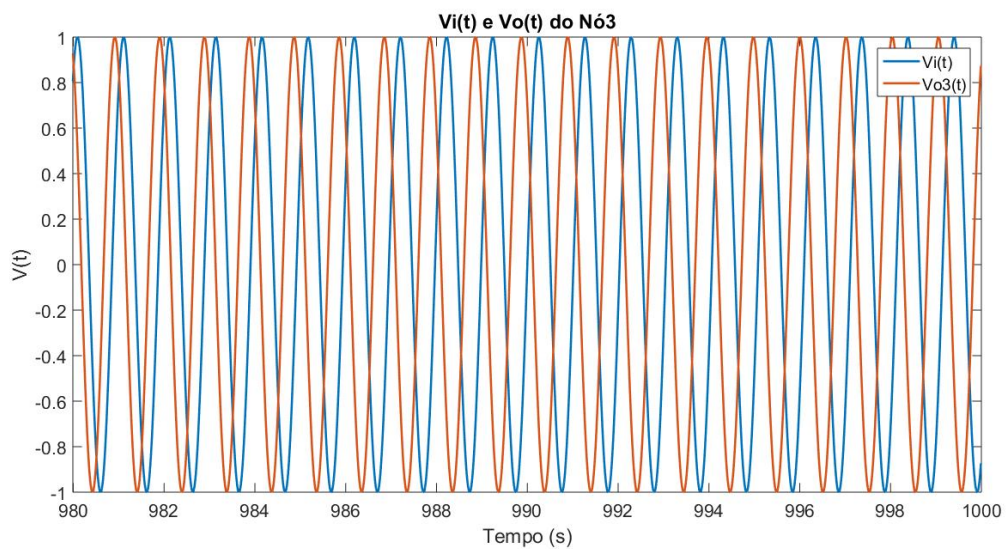
Fonte: Autor.

Figura 4.88: Resposta em regime do Nó Escravo 3 - Dente de Serra com o Filtro 3



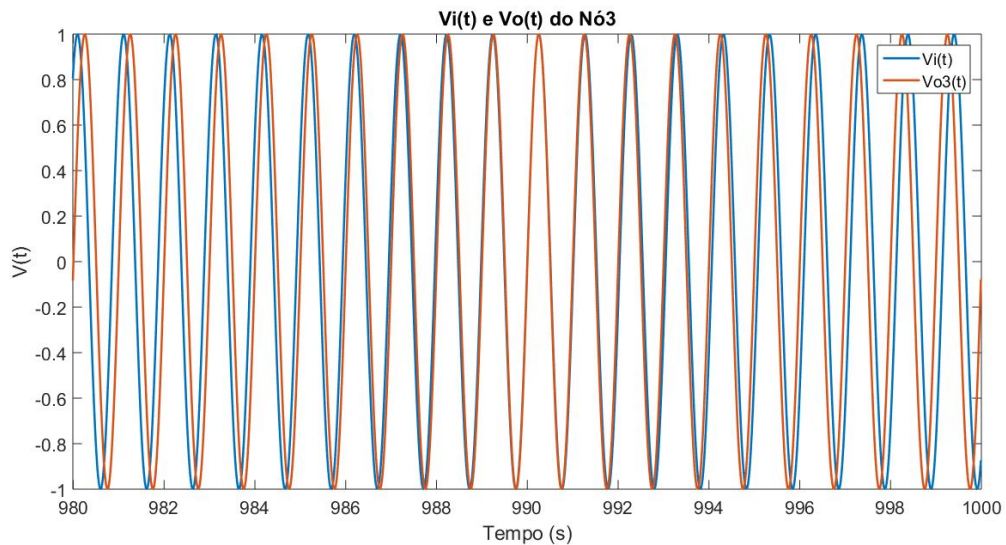
Fonte: Autor.

Figura 4.89: Resposta em regime do Nó Escravo 3 - Dente de Serra com o Filtro 4



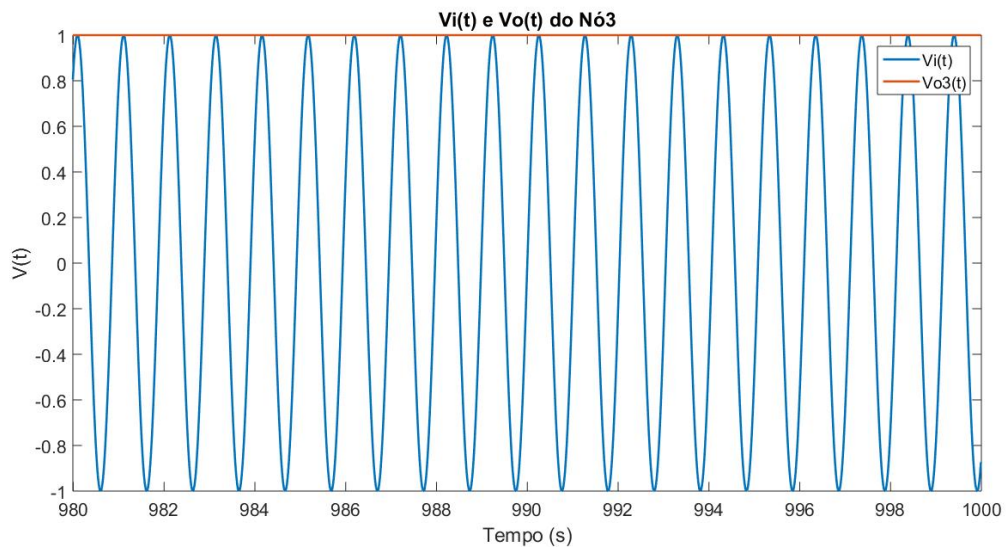
Fonte: Autor.

Figura 4.90: Resposta em regime do Nó Escravo 3 - Dente de Serra com o Filtro 5



Fonte: Autor.

Figura 4.91: Resposta em regime do Nó Escravo 3 - Dente de Serra com o Filtro 6



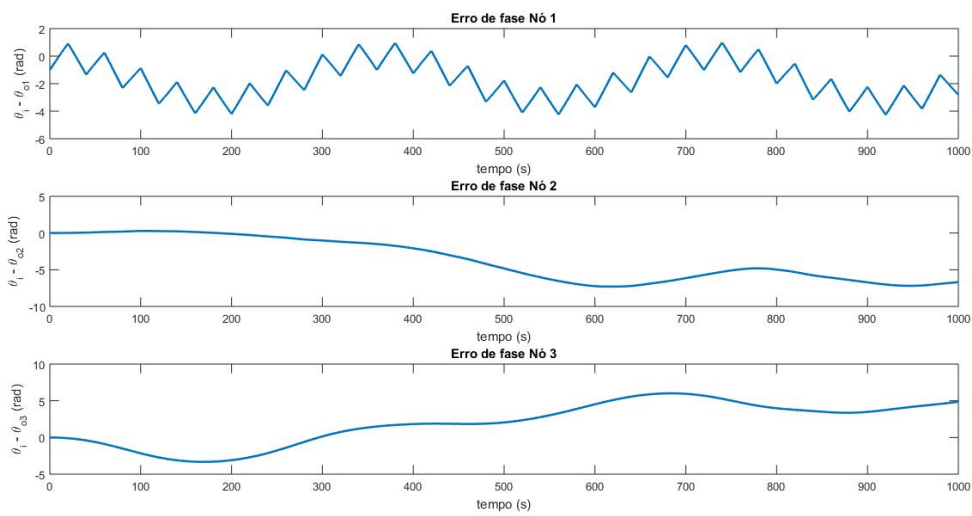
Fonte: Autor.

Ao observarmos os resultados das Figuras 4.86, 4.87, 4.88, 4.89 e 4.90, notamos que, conforme observado no Nó Escravo 2, a resposta do Nó Escravo 3 depende também da análise do erro de fase e de $V_c(t)$ para podermos considerar o estado desse nó. Na Figura 4.91 notamos que não há uma onda de saída desse nó no período analisado, o que

acende uma alerta para observarmos a tensão de controle do Filtro 6 posteriormente e entendermos o porquê desse comportamento.

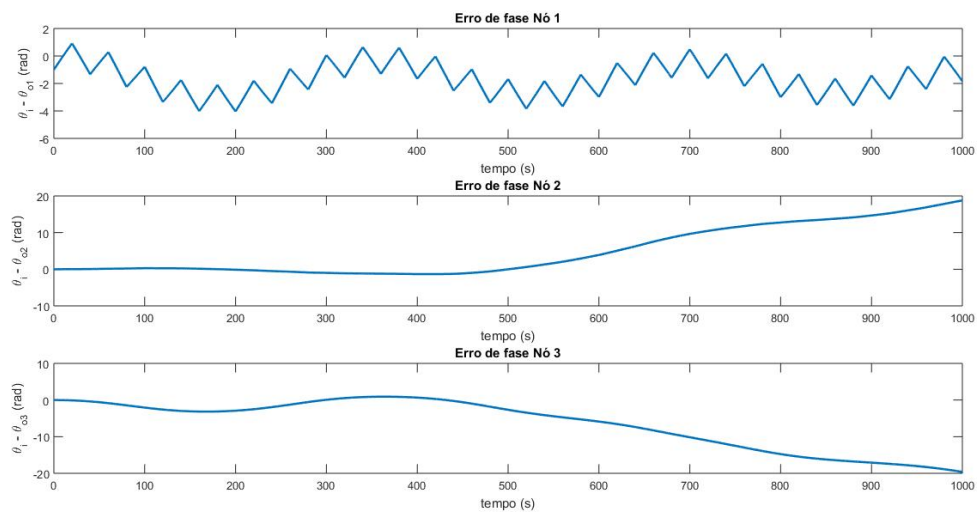
As Figuras 4.92, 4.93, 4.94, 4.95, 4.96 e 4.97 apresentam o erro de fase do sinal de saída dos nós escravos em relação à sua entrada. No caso dos Nós Escravos 1 e 3, o valor de $\theta_i(t)$ está relacionado à $\theta_o(t)$ do Nó Mestre e do Nó 2, respectivamente, enquanto o Nó 2 tem sua fase de acordo com a composição de $\theta_o(t)$ do Nó 1 e 3.

Figura 4.92: Erro de fase da rede TWMS - Dente de Serra com o Filtro 1



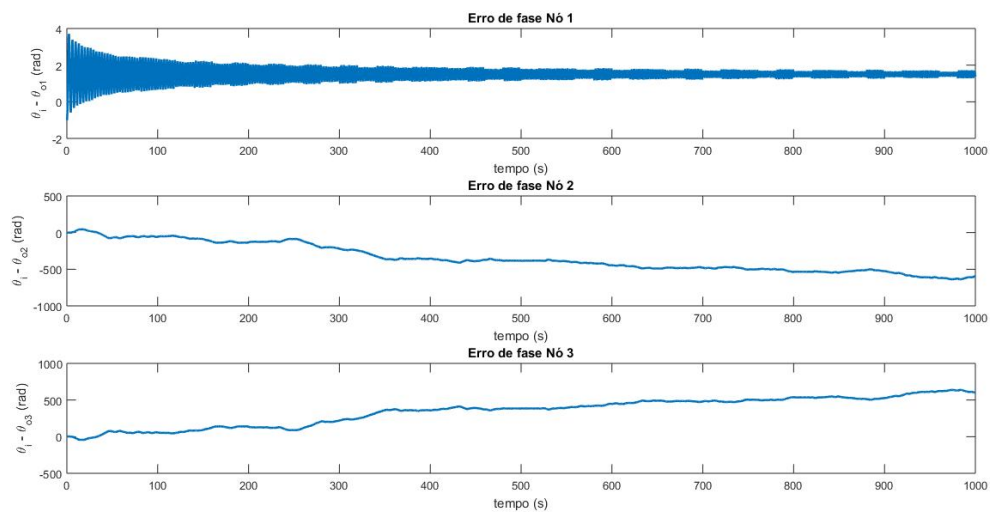
Fonte: Autor.

Figura 4.93: Erro de fase da rede TWMS - Dente de Serra com o Filtro 2



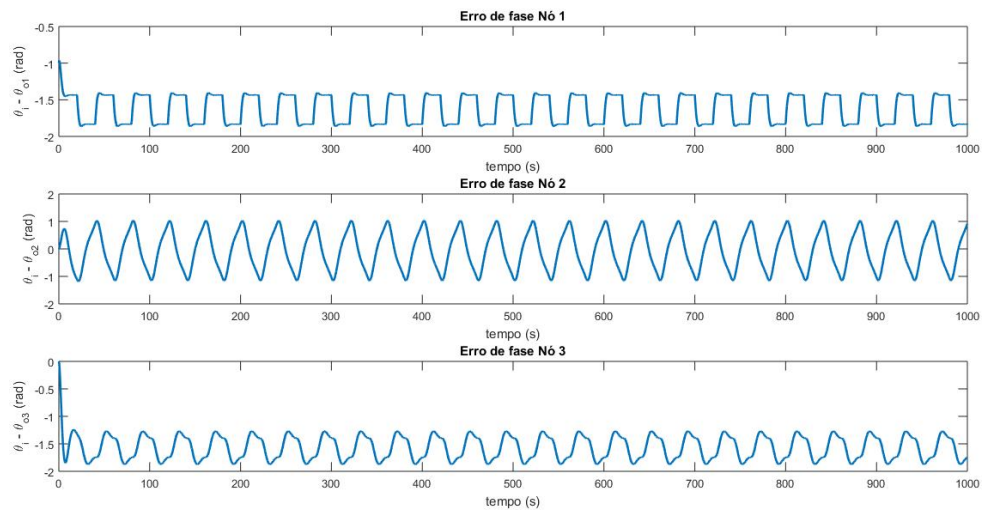
Fonte: Autor.

Figura 4.94: Erro de fase da rede TWMS - Dente de Serra com o Filtro 3



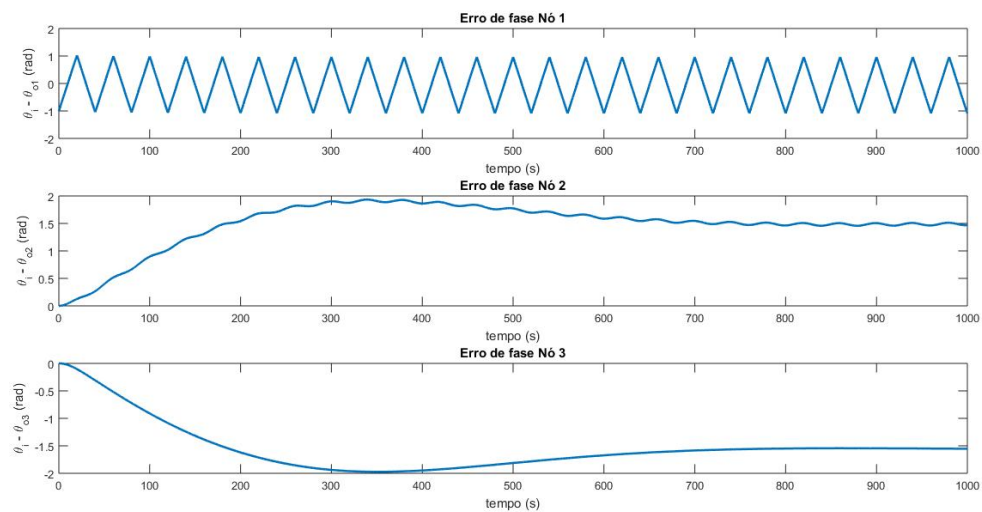
Fonte: Autor.

Figura 4.95: Erro de fase da rede TWMS - Dente de Serra com o Filtro 4



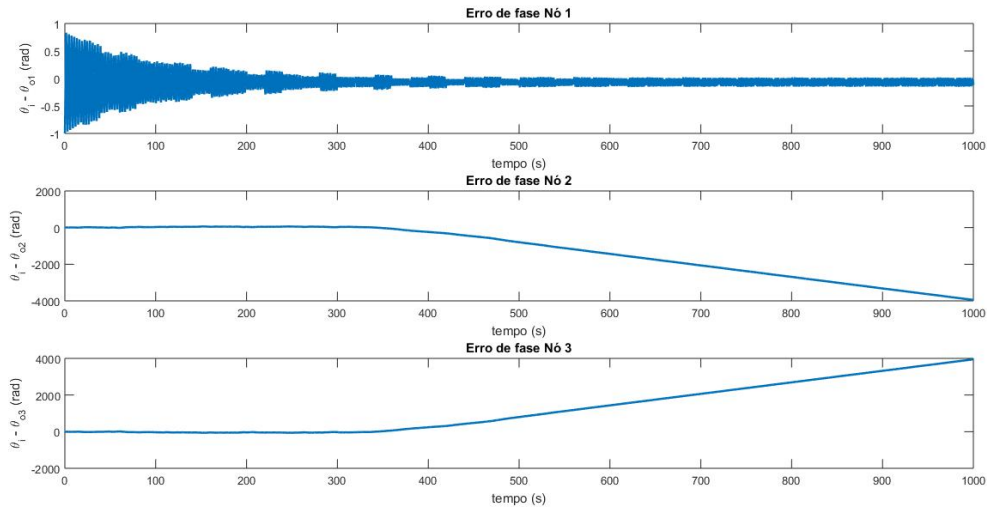
Fonte: Autor.

Figura 4.96: Erro de fase da rede TWMS - Dente de Serra com o Filtro 5



Fonte: Autor.

Figura 4.97: Erro de fase da rede TWMS - Dente de Serra com o Filtro 6



Fonte: Autor.

Ao observarmos o primeiro gráfico, referente ao Nó Escravo 1 das Figuras 4.92, 4.93, e 4.95 notamos a periodicidade do erro de fase que possuem média de $-\pi/2$ para os Filtros 1, 2 e 4, representando o sincronismo do nó com um atraso de 270° , ou -90° . Para as Figuras 4.96 e 4.97, notamos que há o sincronismo de fase para esse nó, devido a oscilação do erro de fase ser em torno de 0° . Para o Filtro 3, apresentado na Figura 4.94, notamos que o erro de fase está orbitando a região de $\pi/2$ ou 90° , indicando esse valor como a fase de sincronismo desse nó.

Nos segundos e terceiros gráficos das Figuras 4.92, 4.93, 4.94 e 4.97, os erros de fases apresentados para os Nós Escravos 2 e 3 não são conclusivos, pois há uma variação existente no período analisado, sendo necessária a verificação de $V_c(t)$ para a validação ou não de sincronismo do nó.

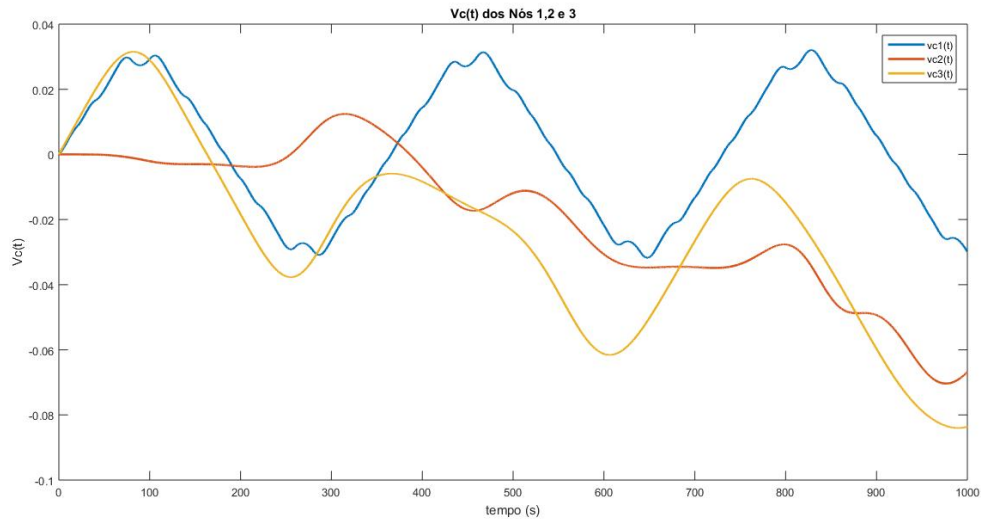
No segundo gráfico das Figuras 4.95 e 4.96, notamos que o Nó Escravo 2 no caso do Filtro 4 sincroniza em fase com o sinal de entrada, devido à média do erro de fase ser zero, enquanto que com o Filtro 5 mantém o erro de fase constante em $\pi/2$ ou 90° , indicando o sincronismo em quadratura.

No terceiro gráfico das Figuras 4.95 e 4.96 observamos que com ambos Filtros 4 e 5, o sincronismo do Nó Escravo 3 se dá em quadratura devido ao erro de fase possuir média igual a $-\pi/2$, indicando 270° ou -90° de defasagem.

As Figuras 4.98, 4.99, 4.100, 4.101, 4.102 e 4.103 apresentam o sinal de controle

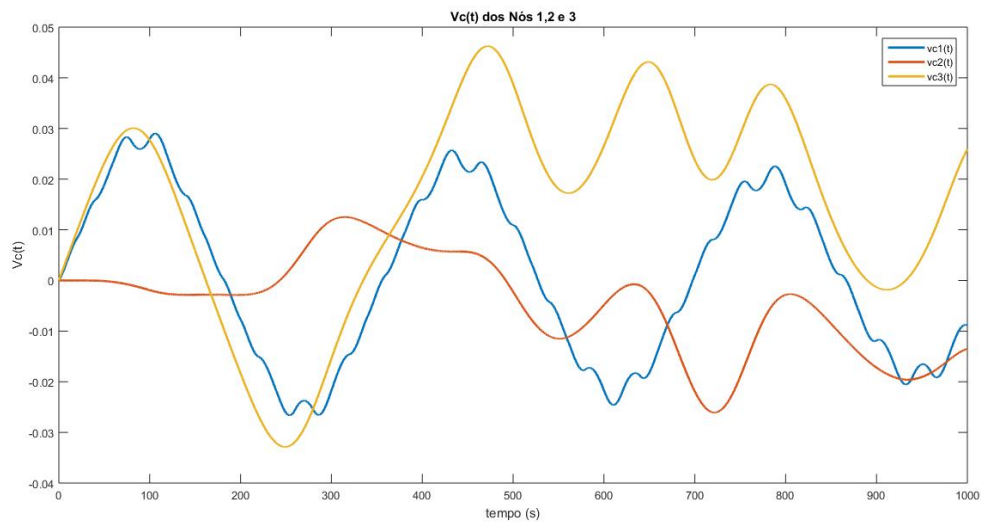
$V_c(t)$ da rede TWMS de acordo com os filtros utilizados.

Figura 4.98: $V_c(t)$ para cada nó da rede TWMS - Dente de Serra com o Filtro 1

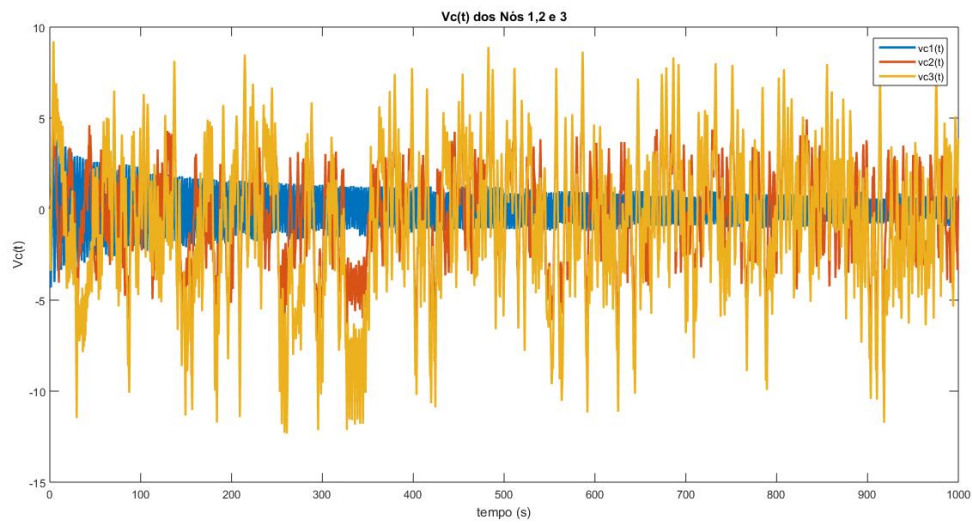


Fonte: Autor.

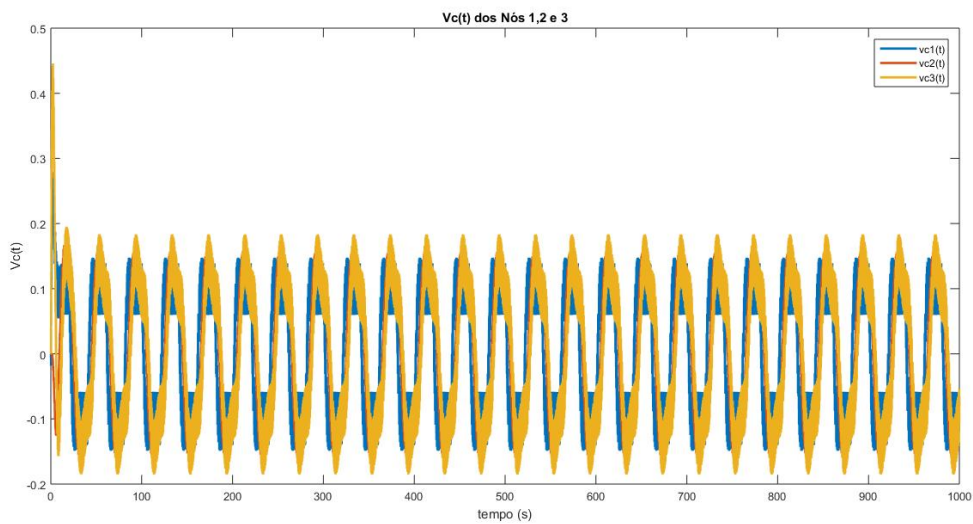
Figura 4.99: $V_c(t)$ para cada nó da rede TWMS - Dente de Serra com o Filtro 2



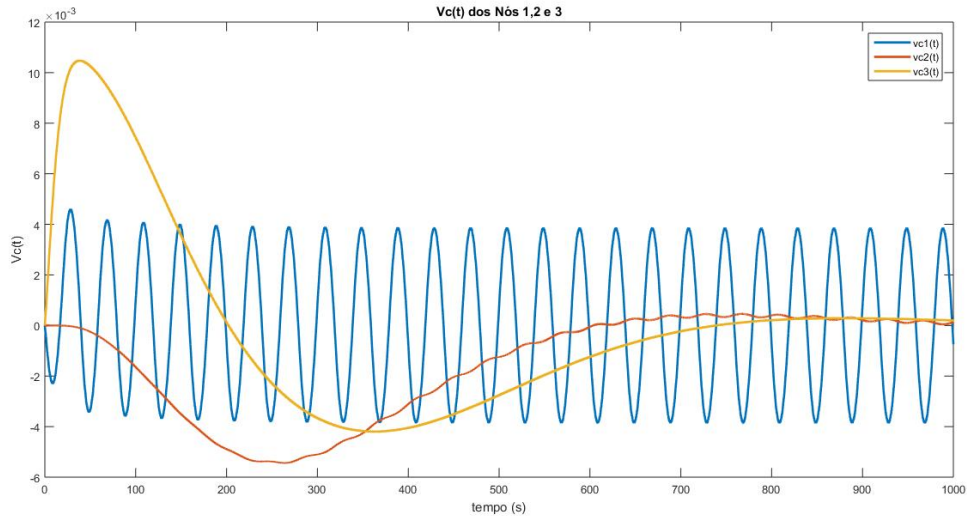
Fonte: Autor.

Figura 4.100: $V_c(t)$ para cada nó da rede TWMS - Dente de Serra com o Filtro 3

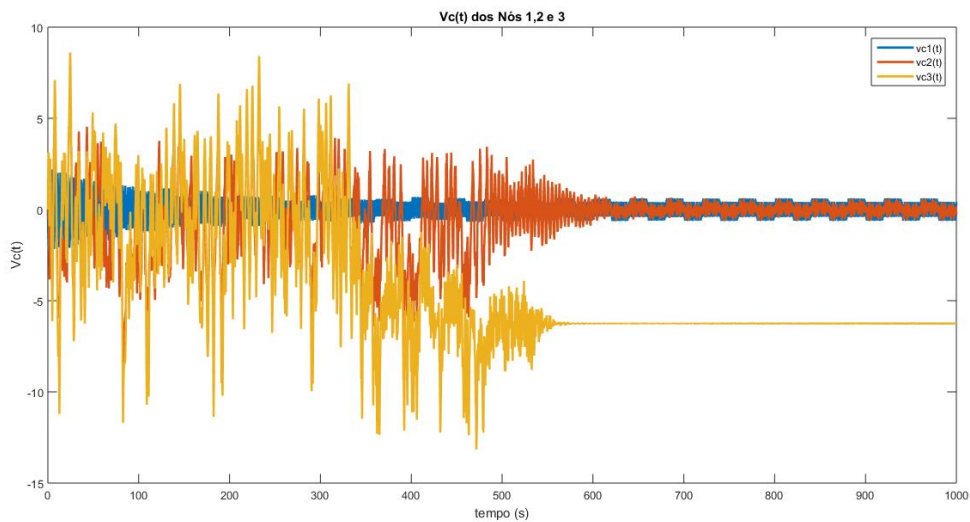
Fonte: Autor.

Figura 4.101: $V_c(t)$ para cada nó da rede TWMS - Dente de Serra com o Filtro 4

Fonte: Autor.

Figura 4.102: $V_c(t)$ para cada nó da rede TWMS - Dente de Serra com o Filtro 5

Fonte: Autor.

Figura 4.103: $V_c(t)$ para cada nó da rede TWMS - Dente de Serra com o Filtro 6

Fonte: Autor.

Ao observarmos as Figuras 4.98, 4.99, 4.100, 4.101, 4.102 e 4.103 sob a ótica de $V_c(t)$ do Nó Escravo 1, vemos que em todas as aplicações de filtros, há o sincronismo desse nó, devido à média zero do valor da tensão de controle.

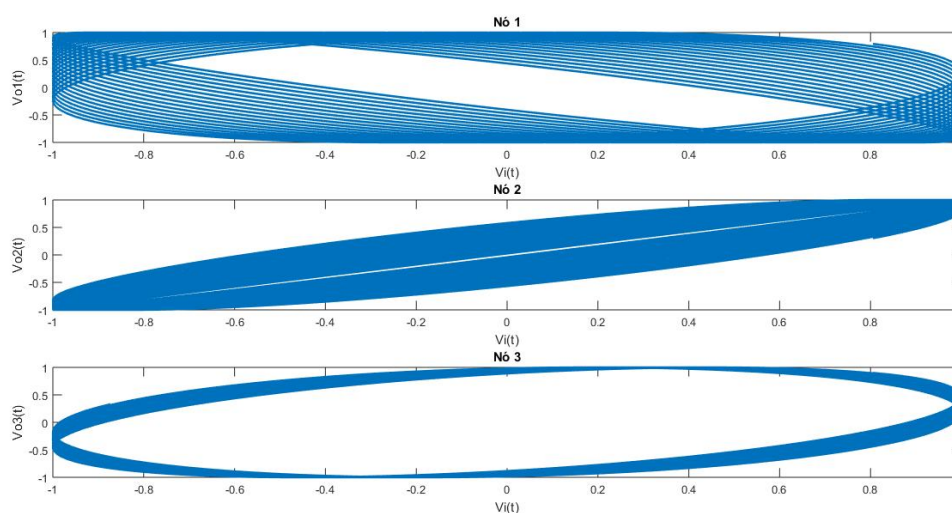
Sob a ótica de $V_c(t)$ para os Nós Escravos 2 e 3, notamos que para os Figuras 4.98,

4.99, 4.100 e 4.103, o valor da tensão de controle não se estabiliza ou possui média zero, indicando que esses nós não encontram a estabilidade para esses filtros. Cabe ressaltar que a tensão $V_c(t)$ para o nó Escravo 3 do Filtro 6 subitamente estabiliza. Tal efeito surgiu devido aos cálculos das expressões de integração implementadas no algoritmo que devem chegar a um erro e por isso, a aplicação do sinal de Dente de Serra ao Filtro 6 não alcançou o comportamento desejado.

Para os Nós Escravos 2 e 3 dos Filtros 4 e 5, notamos por meio das Figuras 4.101 e 4.102 que a rede TWMS está em sincronismo pois a média de $V_c(t)$ para ambas situações possui valor igual a 0.

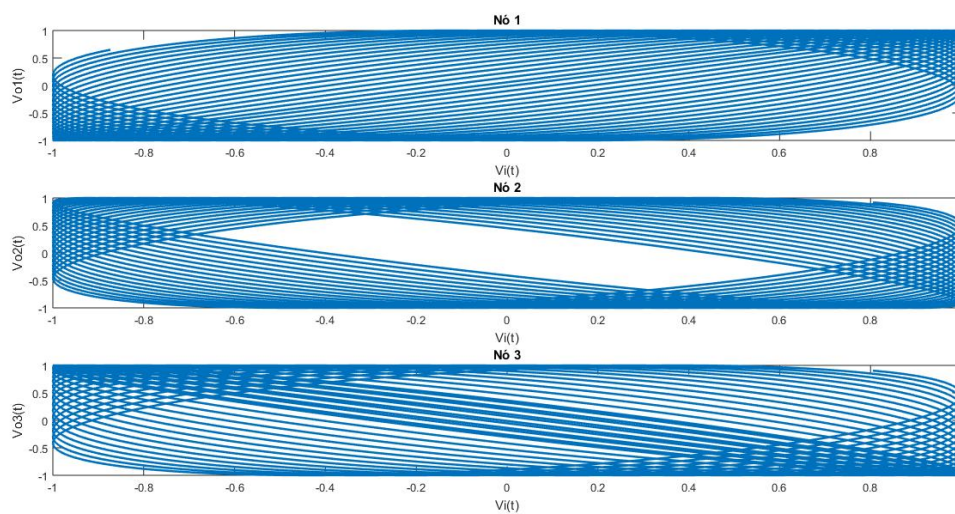
Por fim, as Figuras 4.104, 4.105, 4.106, 4.107, 4.108 e 4.109 apresentam as Figuras de Lissajous, conforme [44, 46].

Figura 4.104: Lissajous para cada nó da rede TWMS - Dente de Serra com o Filtro 1



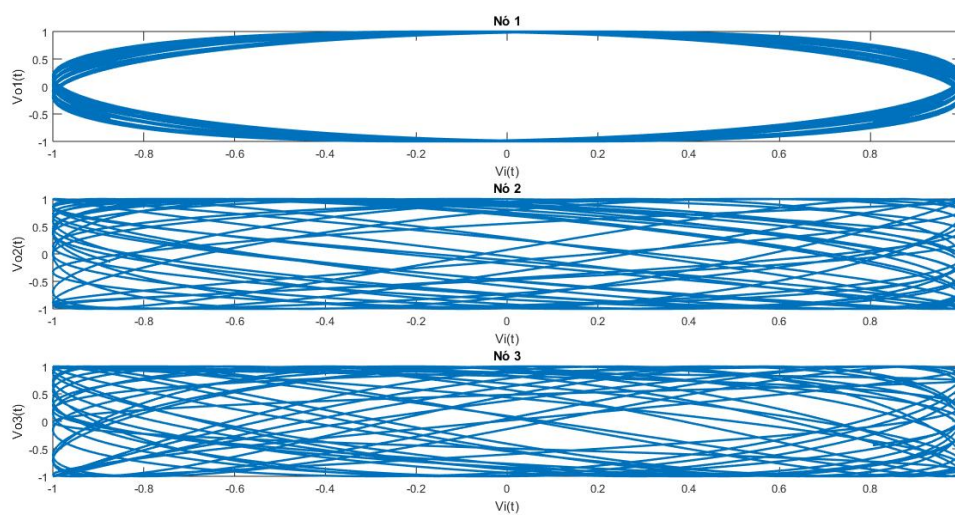
Fonte: Autor.

Figura 4.105: Lissajous para cada nó da rede TWMS - Dente de Serra com o Filtro 2



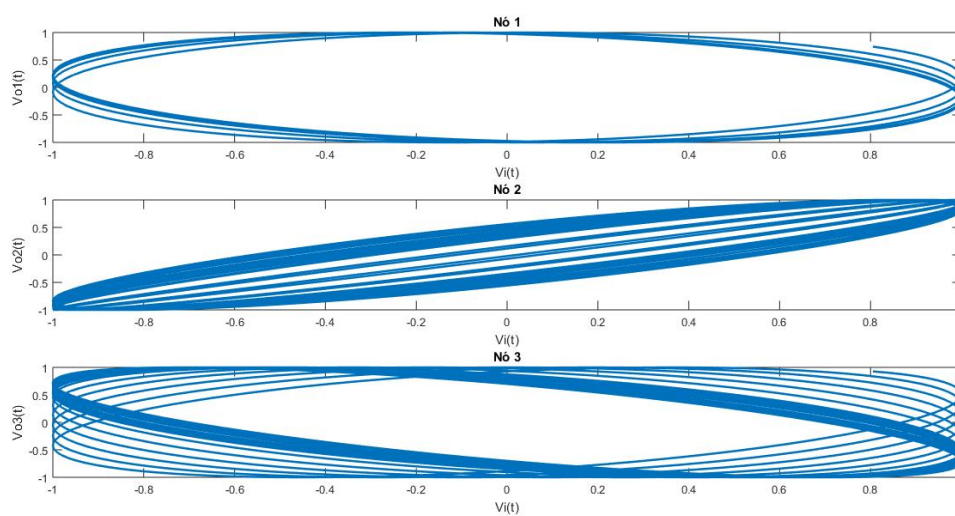
Fonte: Autor.

Figura 4.106: Lissajous para cada nó da rede TWMS - Dente de Serra com o Filtro 3



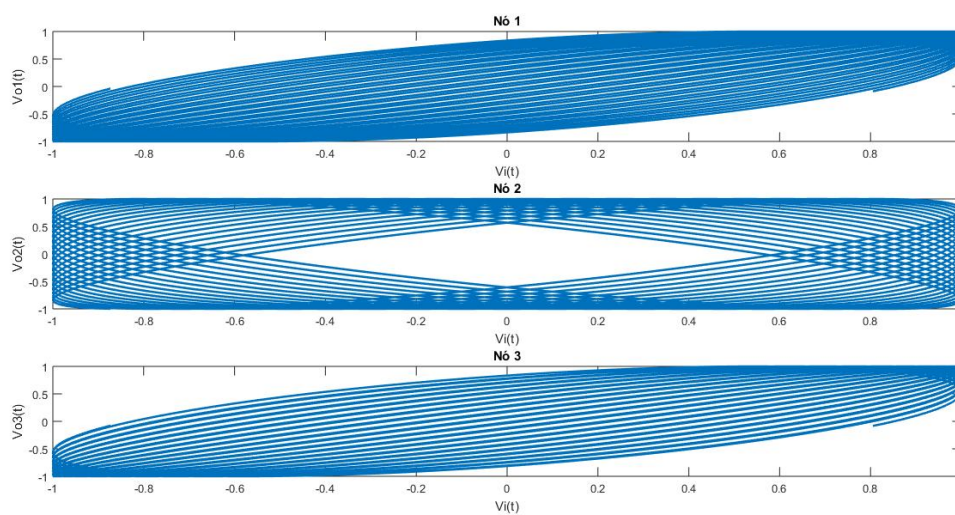
Fonte: Autor.

Figura 4.107: Lissajous para cada nó da rede TWMS - Dente de Serra com o Filtro 4



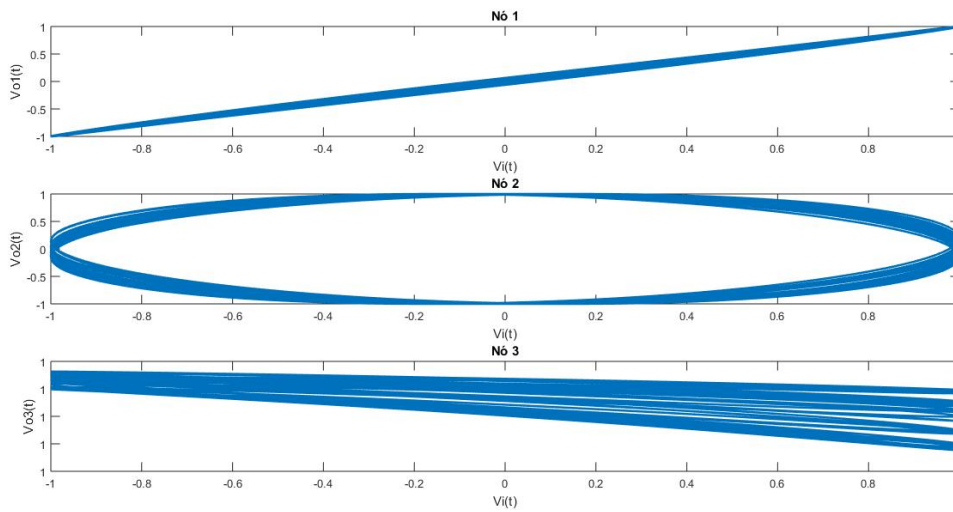
Fonte: Autor.

Figura 4.108: Lissajous para cada nó da rede TWMS - Dente de Serra com o Filtro 5



Fonte: Autor.

Figura 4.109: Lissajous para cada nó da rede TWMS - Dente de Serra com o Filtro 6



Fonte: Autor.

Ao observarmos as Figuras 4.104, 4.105, 4.106, 4.107, 4.108 e 4.109, notamos a reprodução da análise previamente apresentada. Percebemos que, para o Nó Escravo 1, todos os filtros chegam ao sincronismo. Para os Nós Escravos 2 e 3, validamos que para os Filtros 1, 2, 3 e 6 não alcançam o sincronismo devido ao escorregamento das fases das suas respectivas saídas. Vale ressaltar também que para o Filtro 6 para o Nó Escravo 3 uma representação que distorce o comportamento esperado para o nó na rede TWMS, que, conforme já apresentado, surge devido a alguma inconsistência no resultado da da tensão $V_c(t)$ durante o processo de integração. Por fim, para os Filtros 4 e 5, temos a confirmação do sincronismo da rede TWMS, conforme concluído na análise de $V_c(t)$ para os nós em questão.

A Tabela 4.5 apresenta os tempos de simulação para a rede TWMS com os diferentes filtros para a versão sequencial e em paralelo do algoritmo proposto em um processador de dois núcleos.

Tabela 4.5: Tempos de simulação para a resposta ao dente de serra de fase - Intel® Core™ i3-4005U CPU @ 1.70GHz (dois núcleos)

Filtro	Sequencial (s)	Paralelo (s)
Filtro 1	10,31	19016,14
Filtro 2	10,25	18937,00
Filtro 3	10,08	18872,76
Filtro 4	10,05	18836,55
Filtro 5	140,66	20435,44
Filtro 6	136,65	20076,40

A Tabela 4.6 apresenta os tempos de simulação para a rede TWMS com os diferentes filtros para a versão sequencial e em paralelo do algoritmo proposto em um processador de quatro núcleos.

Tabela 4.6: Tempos de simulação para a resposta ao dente de serra de fase - Intel® Core™ i5-4440S CPU @ 2.8GHz (quatro núcleos)

Filtro	Sequencial (s)	Paralelo (s)
Filtro 1	2,43	6731,58
Filtro 2	2,44	6678,02
Filtro 3	2,44	6717,93
Filtro 4	2,33	6748,55
Filtro 5	156,93	7273,89
Filtro 6	156,78	7295,02

Por fim, os tempos de simulação apresentados nas Tabelas 4.5 e 4.6 não fogem do comportamento esperado para o sistema, tendo a mesma ordem de grandeza dos resultados encontrados nas simulações de resposta dos filtros ao degrau unitário (vide Tabelas 4.1 e 4.2) e rampa unitária (vide Tabelas 4.3 e 4.4). Tais resultados, mais uma vez, confirmam que independente do tipo de sinal de entrada, o sistema possui um comportamento e tempo de processamento semelhante.

Capítulo 5

Conclusões

O estudo de redes de PLL, redes de distribuição de sinal de relógio e a aplicação de paralelismo são assuntos que foram conectados nesse trabalho, que busca agregar, junto às referências apresentadas e as demais existentes, uma contribuição à esse estudo e seu comportamento.

O desenvolvimento do paralelismo, por meio do estudo e da aplicação do recurso PCT disponibilizado pela MathWorks[®], trouxe uma nova abordagem para essa linha de pesquisa, devido a sua viabilidade.

Conforme apresentado na referência [43], notamos que o comportamento do erro de fase nos Nós 2 e 3 da rede, quando foram submetidos a rampa de entrada, ocorre somente os Filtros 4 e 5, responsáveis pela sincronização nesses nós. Tal efeito ocorreu devido esses dois filtros trazerem ao sistema pontos de equilíbrio estáveis.

Os resultados observados no comportamento da rede TWMS sob influência de diferentes perturbações da fase do Nó Mestre, demonstram que a aplicação do paralelismo possui a mesma resposta em relação à lógica procedural tradicional (sequencial). Tal fato foi observado durante a execução das simulações, tendo como característica para esse problema analisado, o custo computacional menor para a lógica sequencial.

A abordagem por meio da paralelização de tarefas, impôs a busca de soluções factíveis para o algoritmo construído para essa pesquisa. Destaca-se a aplicação da função *parfor* que foi escolhida para o desenvolvimento da solução. As particularidades desse desenvolvimento trouxe o levantamento das características necessárias para seu uso, bem como direcionamento para o seu funcionamento.

Nota-se que o *trade-off* da lógica sequencial para o paralelismo torna-se interessante com o aumento nível de complexidade do problema a ser analisado, conforme

observado na mudança de ordem da função transferência do filtro, implicando no aumento na ordem na casa de dezenas no tempo de execução.

O tempo de simulação encontrado para a execução do algoritmo, principalmente em paralelismo não possui um resultado praticável, devido a falta de acesso das diretivas e tratamentos que o PCT faz para acessar e trabalhar com o hardware do computador em paralelo, temos que trabalhar com o aprimoramento do algoritmo proposto.

Por fim, a abordagem defendida pela referência [47], que considera o erro de fase gerado pelo detector, como um espaço de fase é bastante interessante visto que esse desenvolvimento evitaria o aparecimento de oscilações ocultas, o que possibilitaria uma melhor operação do algoritmo, à vista disso, a aplicação poderia solucionar a estabilidade do sistema para os diferentes filtros e sinais avaliados.

Capítulo 6

Sugestões de Trabalhos Futuros

Baseando-se nas conclusões apresentadas, podemos resumir e direcionar as contribuições desse trabalho de acordo com:

- Introdução ao paralelismo de tarefas às redes de sincronismo de PLLs;
- Discussão e resultados para redes TWMS de forma gráfica compatível com a solução matemática proposta;
- Introdução ao PCT como ferramenta de paralelismo de tarefa à redes de distribuição de sinais;
- Análise e comparação de desempenhos de diferentes recursos computacionais aplicados à redes TWMS.

Como melhorias a este trabalho, a simulação e acompanhamento utilizando recursos computacionais melhores, com processadores com mais núcleos e a aplicação de redes de computadores. Recomenda-se também o estudo e aprimoramento quanto ao tipo de paralelismo aplicado, testando e validando outras alternativas. Bem como, o estudo do comportamento dinâmico da resposta dos filtros apresentados, devido a um comportamento aparentemente ruidoso em alguns casos, mas que devem ser investigados se tais resultados não são frutos da resposta do sistema dinâmico complexo da rede e/ou problemas com a adoção da integração numérica.

Por fim, como sugestões a trabalhos futuros destacam-se: a exploração às outras arquiteturas de redes de sincronismo; melhoria aos ganhos e modelo de filtro de modo a diminuir o *jitter* de frequência dupla; utilização de outros métodos de integração numérica; aumento das tarefas paralelizadas e; abordagem do detector de fase como um sistema de espaço de fases.

Referências Bibliográficas

- [1] Marmo, Carlos Nehemy. Bifurcações em PLLs de terceira ordem em redes OWMS. 2008. Tese de Doutorado. Universidade de São Paulo.
- [2] Bueno, Atila Madureira. Estudo do jitter de fase em redes de distribuição de sinais de tempo. 2009. Tese de Doutorado. Universidade de São Paulo.
- [3] Piqueira, J. R. C.; Monteiro, L. H. A. All-pole phase-locked loops: calculating lock-in range by using Evan’s root-locus. *International Journal of Control*, v. 79, n. 07, p. 822-829, 2006.
- [4] Kaplan, Elliott D.; Hegarty, Christopher (Ed.). *Understanding GPS/GNSS: Principles and applications*. Artech house, 2017.
- [5] 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.
- [6] Bregni, Stefano. *Synchronization of digital telecommunications networks*. John Wiley Sons, 2002.
- [7] Best, Roland E. *Phase-locked loops: design, simulation, and applications*. McGraw-Hill Education, 2007.
- [8] Gardner, Floyd M. *Phaselock techniques*. John Wiley Sons, 2005.
- [9] Piqueira, Jose Roberto Castilho. Uma contribuição ao estudo das redes com malhas de sincronismo de fase. 1997. Tese de Livre Docência. Universidade de São Paulo.
- [10] MathWorks. *Parallel Computing Toolbox*. Perform parallel computations on multi-core computers, GPUs, and clusters. [S.l.]. MathWorks, 2022. Disponível em: <https://www.mathworks.com/products/parallel-computing.html>. Acesso em: 27 ago. 2022.

- [11] Monteiro, Luiz Henrique Alves; Garcia, Paulo Alves; Piqueira, José Roberto Castilho. Sincronismo em redes mestre-escravo com atraso. Sba: Controle & Automação Sociedade Brasileira de Automatica, v. 14, n. 2, p. 121-126, 2003.
- [12] Piqueira, José Roberto Castilho. Aplicação da teoria qualitativa de equações diferenciais a problemas de sincronismo de fase. 1987. Tese de Doutorado. Universidade de São Paulo.
- [13] Chong, Florence; Reaz, Mamun Ibne; Kamaruzzaman, Mohamed Ibrahim; Badal, Torikul Islam; Farayez, Araf; Crespo, Maria Liz; Cicuttin, Andres. Digital Controlled Oscillator (DCO) for All Phase-Locked Loop (ADPLL) – A REVIEW. Jurnal Teknologi, v. 82, n. 1, 2020.
- [14] Ferruzo Correa, Diego Paolo. Malha síncrona digital Tanlock com estimação de frequência e ganho adaptativo para convergência rápida. Tese de Doutorado. Universidade de São Paulo. 2011.
- [15] 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.
- [16] Wiggins, Stephen; Golubitsky, Martin. Introduction to applied nonlinear dynamical systems and chaos. New York: Springer, 2003.
- [17] W. E. Boyce and R. C. DiPrima. Elementary Differential Equations and Boundary Value Problems, volume 1. John Wiley Sons, 6 edition, 1997.
- [18] Sheng, Yanxiu; Gui, Lin; Wei, Zhiqiang; Duan, Jibing; Liu, Yingying. Layered Models for General Parallel Computation Based on Heterogeneous System. In: 2012 13th International Conference on Parallel and Distributed Computing, Applications and Technologies. IEEE, 2012. p. 45-49.
- [19] Azevedo, Ana Isabel Rojão Lourenço. Introdução ao estudo da paralelização de algoritmos de planeamento operacional com métodos genéticos. 1997. Dissertação de Mestrado. Universidade do Porto.
- [20] Mattson, Timothy G.; Sanders, Beverly; Massingill, Berna. Patterns for parallel programming. Pearson Education, 2004.

- [21] Sato, Liria Matsumoto; Midorikawa, Edson Toshimi; Senger, Hermes. Introdução a programação paralela e distribuída. Anais do XV Jornada de Atualização em Informática, Recife, PE, p. 1-56, 1996.
- [22] Guzzi, M.D. Multitasking runtime systems for the CEDAR multiprocessor. Technical Report CSRD n.604, Center for Supercomputing Research and Development, University of Illinois at Urbana Champaign, 1986.
- [23] Gehani, N.H.; Roome, W.D. Implementing Concurrent C. Software Practice and Experience, v.22, n.3, p.267-85. March 1992..
- [24] Thakkar, S. The Balance multiprocessor system. IEEE Micro, v.8, n.1, p.57-69.Feb. 1988.
- [25] Eager, Derek L.; Zahorjan, John; Lazowska, Edward D. Speedup versus efficiency in parallel systems. IEEE transactions on computers, v. 38, n. 3, p. 408-423, 1989.
- [26] Sun, Xian-He; Ni, Lionel M. Another view on parallel speedup. In: Proceedings of the 1990 ACM/IEEE conference on Supercomputing. 1990. p. 324-333.
- [27] Amdahl, Gene M. Validity of the single processor approach to achieving large scale computing capabilities. In: Proceedings of the April 18-20, 1967, spring joint computer conference. 1967. p. 483-485.
- [28] Gustafson, John L. Reevaluating Amdahl's law. Communications of the ACM, v. 31, n. 5, p. 532-533, 1988.
- [29] Gillespie, Matt. Amdahl's Law, Gustafson's Trend, and the Performance Limits of Parallel Applications. Online: http://software.intel.com/sites/default/files/m/d/4/1/d/8/Gillespie-0053-AAD_Gustafson-Amdahl_v1__2_.rh.final.pdf, 2008.
- [30] Lynch, Stephen. Dynamical systems with applications using MATLAB. Boston: Birkhäuser, 2004.
- [31] Sharma, Gaurav; Martin, Jos. MATLAB®: a language for parallel computing. International Journal of Parallel Programming, v. 37, n. 1, p. 3-36, 2009.
- [32] Moler, C. Why isn't There a Parallel MATLAB. The MathWorks Newsletter, Spring 1995.
- [33] Pacheco, Peter. An introduction to parallel programming. Elsevier, 2011.

- [34] Alyasseri, Zaid Abdi Alkareem. Survey of Parallel Computing with MATLAB. arXiv preprint arXiv:1407.6878, 2014.
- [35] Hemanth, D. J.; Elhosney, M.; Nguyen, T. N. (Ed.). Advances in parallel computing technologies and applications. IOS Press, 2021.
- [36] Mathworks. *Parallel Computing Fundamentals web site*, <https://www.mathworks.com/help/parallel-computing/parallel-computing-fundamentals.html>
- [37] D. N. Varsamis, C. Talagkozis, P. A. Mastorocostas, E. Outsios and N. P. Karampetakis. *The performance of the MATLAB Parallel Computing Toolbox in specific problems*. In: Federated Conference on Computer Science and Information Systems (FedCSIS), Wroclaw, Poland. 2012. p. 587-593.
- [38] M.Szymczyk and P. Szymczyk. *Matlab and parallel computing*. Image Processing & Communications, v. 17, n. 4, p. 207, 2012.
- [39] P. Luszczek. *Parallel programming in MATLAB*. The International Journal of High Performance Computing Applications, v. 23, n. 3, p. 277-283, 2009.
- [40] Butcher, John Charles. The numerical analysis of ordinary differential equations: Runge-Kutta and general linear methods. Wiley-Interscience, 1987.
- [41] Rocha, Alex Fraga et al. Integração associativa em métodos Runge-Kutta. 2016.
- [42] A. M. Bueno, A. G. Rigon, A. A. Ferreira, and J. R. C. Piqueira. Design constraints for third-order pll nodes in master-slave clock distribution networks. *Communications in Nonlinear Science and Numerical Simulation*, 15(9):2565–2574, 2010.
- [43] J. R. C.Piqueira, S. A. Castillo-Vargas, L. H. A. Monteiro. *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.
- [44] J. R. C.Piqueira. *Master-Slave Topologies with Phase-Locked Loops*. Wireless Communications and Mobile Computing, v. 2020, 2020.
- [45] J. R. C.Piqueira and A. C. B. de Godoi. *Clock signal distribution with second order nodes: Design hints*. ISA transactions, v. 115, p. 124-142, 2021.
- [46] Marion, Jerry B. Classical dynamics of particles and systems. Academic Press, 2013.

- [47] 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.

Apêndice A

Métodos de integração numérica

Dentre os métodos de soluções de integração definidas, a aplicação de métodos numéricos é essencial, devido ao fato de que muitas expressões não possuem uma equação primitiva fácil de se definir.

O avanço da informática possibilitou a implementação desses métodos devido a fácil compreensão como algoritmo, tendo o destaque aos métodos: Método de Euler ou Método das Retas Tangentes, Método de Runge-Kutta de quarta ordem e os Métodos de Adams-Bashfort e Adam-Moulton [41]. Nesse trabalho, será apresentado os dois primeiros, com o foco de aplicação do método de Runge-Kutta de quarta ordem.

A.1 Método de Euler

Para entendermos esse método, a equação A.1 será o ponto de partida.

$$\frac{dy}{dt} = f(t, y), \quad (\text{A.1})$$

com condições iniciais de modo que $y(t_0) = y_0$. Considerando que as funções f e f_y sejam contínuas em um retângulo do plano ty , contendo o ponto (t_0, y_0) . Então, pelo teorema da unicidade, ou seja, única solução, existe nesse retângulo uma única solução $y = \phi(t)$ do problema em algum ponto em torno de t_0 . Necessita-se que A.1 seja linear, para que a solução exista [40, 41].

A A.1 no ponto t_0 tem a forma:

$$\frac{d\phi}{dt}(t_n) = f[(t_n, \phi(t_n))], \quad (\text{A.2})$$

Aproximando a derivada da A.2 utilizando o quociente das diferenças, obtemos:

$$\frac{\phi(t_{n+1}) - \phi(t_n)}{t_{n+1} - t_n} \cong f[(t_n, \phi(t_n))]. \quad (\text{A.3})$$

Ao substituírmos $\phi(t_{n+1})$ e $\phi(t_n)$ por seus valores y_{n+1} e y_n , respectivamente, e resolvendo para y_{n+1} , temos a fórmula de Euler, dada por:

$$y_{n+1} = y_n + f(t_n, y_n)(t_{n+1} - t_n) \quad n = 0, 1, 2, \dots, . \quad (\text{A.4})$$

Caso o passo $t_{n+1} - t_n$ seja uniforme, de valor h , então, para todo n e denotando $f(t_n, y_n)$ por f_n , podemos simplificar a equação A.4 em:

$$y_{n+1} = y_n + hf_n \quad n = 0, 1, 2, \dots, . \quad (\text{A.5})$$

Computacionalmente, a aplicação do método de Euler para a resolução de integrações numéricas consiste no cálculo repetidamente da equação A.4 ou A.5, carregando os valores obtidos para a rodada n para a rodada seguinte. Realizando isso, obtemos uma sequência de valores $y_0, y_1, \dots, y_n, y_{n+1}, \dots$ que se aproxima da solução $\phi(t)$ nos pontos $t_0, t_1, \dots, t_n, t_{n+1}, \dots$, [40, 41].

A.2 Método de Runge-Kutta de quarta ordem

O método de Runge-Kutta é uma extensão da aproximação do método proposto por Euler, cuja a aplicação promove uma melhor precisão nos resultados. Esse método é composto por uma média ponderada dos valores $f(t, y)$ em diferentes pontos no intervalo $t_n \leq t \leq t_{n+1}$, cuja a expressão é apresentada na A.6 [40, 41].

$$y_{n+1} = y_n + h \left(\frac{k_{n1} + 2k_{n2} + 2k_{n3} + k_{n4}}{6} \right), \quad (\text{A.6})$$

onde,

$$k_{n1} = f(t_n, y_n),$$

$$k_{n2} = f\left(t_n + \frac{1}{2}h, y_n + \frac{1}{2}hk_{n1}\right),$$

$$k_{n3} = f\left(t_n + \frac{1}{2}h, y_n + \frac{1}{2}hk_{n2}\right),$$

$$k_{n4} = f(t_n + h, y_n + hk_{n3}).$$

Analisando a equação A.6, temos que a soma $\frac{k_{n1} + 2k_{n2} + 2k_{n3} + k_{n4}}{6}$ pode ser considerada como um coeficiente angular médio. Ao se observar a expressão, nota-se que o

fator k_{n1} é o coeficiente angular do extremo esquerdo do intervalo, k_{n2} é o coeficiente angular no ponto médio quando utilizamos a fórmula de Euler para ir de t_n a $t_n + \frac{1}{2}h$, k_{n3} é a segunda aproximação do coeficiente angular no ponto médio e k_{n4} é o coeficiente angular no ponto $t_n + h$ utilizando a fórmula de Euler e o coeficiente angular k_{n3} para ir de t_n a $t_n + h$ [40, 41].

Generalizando, da mesma forma que feito com o método de Euler, o método de Runge-Kutta de quarta ordem para sistemas de equações diferenciais possui a seguinte forma:

$$x_{n+1} = x_n + h\left(\frac{k_{n1} + 2k_{n2} + 2k_{n3} + k_{n4}}{6}\right),$$

em que:

$$k_{n1} = f(t_n, x_n),$$

$$k_{n2} = f\left(t_n + \frac{1}{2}h, x_n + \frac{1}{2}hk_{n1}\right),$$

$$k_{n3} = f\left(t_n + \frac{1}{2}h, x_n + \frac{1}{2}hk_{n2}\right),$$

$$k_{n4} = f(t_n + h, x_n + hk_{n3}).$$