



Ministério da  
Ciência e Tecnologia



INPE-16689-TDI/1633

**ESTUDO DOS ALGORITMOS WELCH-LYNCH (FTM),  
FAULT-TOLERANT AVERAGE (FTA) E FILTRO DE  
KALMAN (FK) PARA SINCRONIZAÇÃO DE  
RELÓGIOS E SUAS INFLUÊNCIAS SOBRE UM  
SISTEMA DE CONTROLE**

Eloy Martins de Oliveira Júnior

Dissertação de Mestrado do Curso de Pós-Graduação em Engenharia e Tecnologia  
Espaciais/Mecânica Espacial e Controle, orientada pelo Dr. Marcelo Lopes de  
Oliveira e Souza, aprovada em 26 de fevereiro de 2010.

Registro do documento original:

<<http://urlib.net/sid.inpe.br/mtc-m19@80/2010/04.20.00.49>>

INPE  
São José dos Campos  
2010

## **PUBLICADO POR:**

Instituto Nacional de Pesquisas Espaciais - INPE

Gabinete do Diretor (GB)

Serviço de Informação e Documentação (SID)

Caixa Postal 515 - CEP 12.245-970

São José dos Campos - SP - Brasil

Tel.:(012) 3945-6911/6923

Fax: (012) 3945-6919

E-mail: [pubtc@sid.inpe.br](mailto:pubtc@sid.inpe.br)

## **CONSELHO DE EDITORAÇÃO:**

### **Presidente:**

Dr. Gerald Jean Francis Banon - Coordenação Observação da Terra (OBT)

### **Membros:**

Dr<sup>a</sup> Maria do Carmo de Andrade Nono - Conselho de Pós-Graduação

Dr. Haroldo Fraga de Campos Velho - Centro de Tecnologias Especiais (CTE)

Dr<sup>a</sup> Inez Staciarini Batista - Coordenação Ciências Espaciais e Atmosféricas (CEA)

Marciana Leite Ribeiro - Serviço de Informação e Documentação (SID)

Dr. Ralf Gielow - Centro de Previsão de Tempo e Estudos Climáticos (CPT)

Dr. Wilson Yamaguti - Coordenação Engenharia e Tecnologia Espacial (ETE)

## **BIBLIOTECA DIGITAL:**

Dr. Gerald Jean Francis Banon - Coordenação de Observação da Terra (OBT)

Marciana Leite Ribeiro - Serviço de Informação e Documentação (SID)

Jefferson Andrade Ancelmo - Serviço de Informação e Documentação (SID)

Simone A. Del-Ducca Barbedo - Serviço de Informação e Documentação (SID)

## **REVISÃO E NORMALIZAÇÃO DOCUMENTÁRIA:**

Marciana Leite Ribeiro - Serviço de Informação e Documentação (SID)

Marilúcia Santos Melo Cid - Serviço de Informação e Documentação (SID)

Yolanda Ribeiro da Silva Souza - Serviço de Informação e Documentação (SID)

## **EDITORAÇÃO ELETRÔNICA:**

Viveca Sant´Ana Lemos - Serviço de Informação e Documentação (SID)



Ministério da  
Ciência e Tecnologia



INPE-16689-TDI/1633

**ESTUDO DOS ALGORITMOS WELCH-LYNCH (FTM),  
FAULT-TOLERANT AVERAGE (FTA) E FILTRO DE  
KALMAN (FK) PARA SINCRONIZAÇÃO DE  
RELÓGIOS E SUAS INFLUÊNCIAS SOBRE UM  
SISTEMA DE CONTROLE**

Eloy Martins de Oliveira Júnior

Dissertação de Mestrado do Curso de Pós-Graduação em Engenharia e Tecnologia Espaciais/Mecânica Espacial e Controle, orientada pelo Dr. Marcelo Lopes de Oliveira e Souza, aprovada em 26 de fevereiro de 2010.

Registro do documento original:

<<http://urlib.net/sid.inpe.br/mtc-m19@80/2010/04.20.00.49>>

INPE  
São José dos Campos  
2010

Oliveira Júnior, Eloy Martins de.  
Ol3es      Estudo dos algoritmos Welch-Lynch (FTM), Fault-Tolerant Average (FTA) e Filtro de Kalman (FK) para sincronização de relógios e suas influências sobre um sistema de controle / Eloy Martins de Oliveira Júnior. – São José dos Campos : INPE, 2010. xxxviii + 417 p. ; (INPE-16689-TDI/1633)

Dissertação (Mestrado em Engenharia e Tecnologia Espaciais/Mecânica Espacial e Controle) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2010.

Orientador : Dr. Marcelo Lopes de Oliveira e Souza.

1. Sincronização de relógios. 2. Tempo real. 3. Algoritmo Welch-Lynch. 4. Sistema de controle por rede. 5. Filtro de Kalman. 6. Microtik. 7. Tempo global. 8. Sincronização interna. 9. FTA (Fault-Tolerant Average). I.Título.

CDU 681.527.7

---

Copyright © 2010 do MCT/INPE. Nenhuma parte desta publicação pode ser reproduzida, armazenada em um sistema de recuperação, ou transmitida sob qualquer forma ou por qualquer meio, eletrônico, mecânico, fotográfico, reprográfico, de microfilmagem ou outros, sem a permissão escrita do INPE, com exceção de qualquer material fornecido especificamente com o propósito de ser entrado e executado num sistema computacional, para o uso exclusivo do leitor da obra.

Copyright © 2010 by MCT/INPE. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, microfilming, or otherwise, without written permission from INPE, with the exception of any material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use of the reader of the work.

**Aprovado (a) pela Banca Examinadora  
em cumprimento ao requisito exigido para  
obtenção do Título de Mestre em**

**Engenharia e Tecnologia Espaciais/Mecânica  
Espacial e Controle**

**Dr. Mario Cesar Ricci**



---

Presidente / INPE / SJCampos - SP

**Dr. Marcelo Lopes de Oliveira e Souza**



---

Orientador(a) / INPE / SJCampos - SP

**Dr. Hélio Koiti Kuga**



---

Membro da Banca / INPE / SJCampos - SP

**Dr. Paulo Glácomo Milani**



---

Membro da Banca / INPE / SJCampos - SP

**Dr. Gilberto da Cunha Trivelato**



---

Convidado(a) / MECTRON / São José dos Campos - SP

**Aluno (a): Eloy Martins de Oliveira Junior**

**São José dos Campos, 26 de fevereiro de 2010**



*"O que é, por conseguinte, o tempo? Se ninguém me perguntar, eu sei; porém, se quiser explicar a quem me perguntar, já não sei".*

*(Santo Agostinho, Conf., XI, 14, 17)*

*"Tudo aquilo que pode ser imaginado, pode ser realizado".*

*(Nikola Tesla)*





*Dedico esta pesquisa à minha família, o porto seguro que é o meu abrigo em todos os momentos da minha vida.*



## AGRADECIMENTOS

À minha avó Rosa, por sua dedicação e carinho incondicionais. À minha irmã Erika, sempre ao meu lado, cuja graça, compreensão e confiança me forneceram o combustível para buscar sempre o melhor. Ao meu pai, Eloy (*in memoriam*), de quem herdei a curiosidade e a determinação necessária à pesquisa científica. À minha mãe, minha primeira mestra, cuja inteligência, cultura e criatividade me despertaram para a busca de complexas realidades e realização de sonhos considerados impossíveis. Agradeço ainda, de coração, à minha irmã Fabiana, ao meu tio e padrinho Luiz Fernando e sua esposa Gorete, pela consideração e carinho com que me brindam desde sempre:

***"Aquele que toma a realidade e faz dela um sonho é um artista.  
Também será artista aquele que do sonho faz a realidade."***

*(Malba Tahan).*

Ao professor Dr. Marcelo Lopes de Oliveira e Souza, orientador que dedicou conhecimento, tempo, esforço e, mais do que tudo, interesse a este estudo. Aos Engs. do *Department of Automatic Control da Lund University* pela disponibilização da ferramenta de simulação TrueTime® e pelo suporte prestado. Também agradeço o auxílio valioso da Dra. Nadjara dos Santos, do Dr. Rolf Valdivia, do engenheiro Rafael Navet e a todos os professores e colegas com quem tive o privilégio de conviver durante o Mestrado:

***Os homens de bem te ensinarão boas coisas.***

*(Platão in Xenofonte, Ditos e Feitos Memoráveis de Sócrates, Capítulo II)*

Aos meus amigos Sinval Ladeira e Simão Pedro Toledo (*in memoriam*):

***O valor das coisas não está no tempo que elas duram, mas na  
intensidade com que acontecem. Por isso existem momentos  
inesquecíveis, coisas inexplicáveis e pessoas incomparáveis.***

*(Fernando Sabino)*

A Deus:

***Para os crentes, Deus está no princípio das coisas. Para os  
cientistas, no final de toda reflexão.*** *(Max Planck).*



## RESUMO

O problema da sincronização de relógios pode aparecer em vários contextos diferentes, principalmente em aplicações complexas e altamente integradas, como um satélite artificial, integram a computação, comunicação e o controle.

No contexto de sistemas de tempo real, a composição requer que a aplicação seja especificada tanto no domínio lógico quanto no domínio temporal. Para garantir os requisitos temporais e um comportamento consistente e estável é necessário um estudo mais aprofundado de sincronização de relógios em um sistema distribuído de tempo real. Este trabalho tem por objetivo principal estudar algoritmos de sincronização de relógios e suas influências sobre um sistema de controle. Para isto, foram escolhidos: 1) o algoritmo de sincronização de relógios Welch-Lynch (FTM – Fault-Tolerant Mid-Point); 2) Filtro de Kalman; 3) FTA (Fault-Tolerant Average); 4) Duas arquiteturas de controle distribuído (Federada e Integrada).

Nas duas arquiteturas de controle utilizadas foram analisados e simulados os efeitos das imperfeições do relógio no sistema. Na arquitetura de controle federada a malha de controle não está em rede e na arquitetura de controle integrada a malha de controle está em rede. Para a simulação foi usada uma ferramenta de simulação chamada TrueTime, baseada em Matlab/Simulink e o Microsoft Excel. Variando os parâmetros de deriva e viés inicial do relógio, foram feitos os seguintes casos de simulação: 1) Simulação com o Microsoft Excel dos Algoritmos Welch-Lynch e FTA, para análise e comparação; 2) Sincronização de Relógios utilizando-se do Welch-Lynch em um sistema de controle com arquitetura federada com barramento de comunicação TDMA; 3) Sincronização de Relógios com Filtro de Kalman em um sistema de controle com arquitetura federada com barramento de comunicação TDMA; 4) Sincronização de Relógios utilizando-se o Welch-Lynch em um sistema de controle com arquitetura integrada com os barramentos de comunicação TDMA e Ethernet. Estes estudos de caso possibilitaram analisar diversos casos de sincronização de relógios, e as várias influências que os relógios podem causar no sistema de controle e na congruência dos dados. Concluiu-se que: 1) A arquitetura de controle Federada é muito mais robusta a erros de relógios que a arquitetura de controle Integrada que é mais suscetível a erros; 2) O algoritmo Welch-Lynch não garante a correção de erros com viés inicial; 3) O Filtro de Kalman corrigiu melhor erros de viés inicial; 4) O FTA possui um desempenho melhor quando o sistema possui mais de 7 relógios a se sincronizar; 5) Os três algoritmos, Welch-Lynch, FTA e Filtro de Kalman corrigiram muito bem o erro de deriva de relógio.

Os Algoritmos Welch-Lynch, o FTA e o Filtro de Kalman são três métodos muito bons, desde que respeitadas as condições e restrições impostas pelo sistema. Neste trabalho os seus resultados foram muito bons. Com os algoritmos foi possível simular, analisar e verificar de uma forma geral, suas suposições, suas vantagens e desvantagens e a precisão alcançada em cada caso de sincronização.



# **STUDY OF ALGORITHMS WELCH-LYNC(FTM), FAULT-TOLERANT AVERAGE (FTA) AND KALMAN FILTER FOR CLOCK SYNCHRONIZATION AND THEIR INFLUENCES ON A CONTROL SYSTEM**

## **ABSTRACT**

The clock synchronization problems can appear in several different contexts, mainly in complex applications and highly integration, as an artificial satellite, which integrate the computer, communication and control systems. In the real time systems context the composition requires that the application is specified in both logical domain and in the temporal domain. To ensure the timing requirements and a consistent and stable behavior is necessary to further study the synchronization of clocks in a distributed real time system. This work intends to study the clock synchronization algorithms and their influences on a control system. For do this, we chose: 1) The Welch-Lynch (FTM – Fault-Tolerant Mid-Point) clock synchronization algorithm; 2) Kalman Filter; 3) FTA (Fault-Tolerant Average); 4) Two Distributed Control System Architecture (Federated and Integrated). We used two different control architectures to analyze and simulate the effects of imperfections of clock in the system, a Federated Control Architecture (the control loop is not networked) and an Integrated Control Architecture (the control loop on the network / Networked Control System). For the simulation was used a TrueTime toolbox, based on Matlab/Simulink and the Microsoft Excel. By varying the parameters of drift and offset of the clock, were made the following cases of simulation: 1) Simulation with Microsoft Excel of a FTM Algorithm and a FTA algorithm for review and comparison; 2) Clock Synchronization using the Welch-Lynch Algorithm in a federated control system architecture with TDMA databus; 3) Clock Synchronization with Kalman Filter technique in a federated control system architecture with TDMA databus; 4) Clock Synchronization using the FTM Algorithm in a integrated control system architecture with a TDMA and Ethernet databus. These case studies made it possible to analyze several cases of clock synchronization, and the various influences that the clock errors may cause the clocks in the control system and in the data congruence. It was concluded that: 1) The Federated control architecture is much more robust to clock errors that integrated control architecture that is more susceptible to clock errors; 2) The FTM algorithm does not guarantee the convergence with initial errors; 3) The Kalman Filter was best corrected with the initial errors; 4) The FTA has a better performance when the system has more than 7 clocks to synchronize; 5) The three algorithms, FTM, FTA and Kalman filter corrected very well the error of clock drift. The FTM, FTA and Kalman Filter algorithms are three good methods to synchronize clocks, since it the constraints and conditions are respected. In this work the results of three algorithms are very good. With the algorithms were simulated, analyzed and verified, in general, theirs assumptions, advantages, disadvantages and the precision achieved of each case simulated.





## LISTA DE FIGURAS

	<b><u>Pág.</u></b>
Figura 2.1 – Especificações da resposta transitória a um degrau unitário. ....	10
Figura 2.2 – Componentes essenciais de um Sistema de Controle. ....	11
Figura 2.3 – Diagrama de blocos de um sistema de controle analógico-digital. ....	16
Figura 2.4 – Modelo de Comunicação <i>Time-Triggered</i> . ....	23
Figura 2.5 – Divisor de Freqüências. ....	32
Figura 2.6 – Divisor de Freqüências no TTP/C. ....	33
Figura 2.7 – Resolução de um Relógio Físico. ....	34
Figura 2.8 – Relógio Físico Ideal. ....	35
Figura 2.9 – Imperfeições do Relógio. ....	37
Figura 2.10 – Modelo Geométrico do Relógio Físico. ....	38
Figura 2.11 – Microticks. ....	41
Figura 2.12 – Intervalo da Taxa de Deriva. ....	43
Figura 2.13 – Máximo viés instantâneo entre dois relógios. ....	44
Figura 2.14 – Precisão em um conjunto de dois relógios. ....	45
Figura 2.15 – Constelação GPS. ....	47
Figura 2.16 – Macrotick. ....	48
Figura 2.17 – Tempo Global. ....	50
Figura 2.18 – Estampa de Tempo para Evento Simples. ....	51
Figura 2.19 – Quatro Eventos. ....	52
Figura 2.20 – Intervalo de Re-sincronização e Função de convergência. ....	55
Figura 2.21 – Simulação do Algoritmo Welch-Lynch. ....	57
Figura 2.22 – Sincronização Externa com GPS. ....	57
Figura 2.23 – Família de Soluções de Sincronização de Relógios. ....	58
Figura 2.24 – Diagrama em blocos de um PLL. ....	58
Figura 2.25 – Diagrama em blocos de fases de uma arquitetura distribuída. ..	60
Figura 2.26 – Arquitetura Distribuída. ....	61
Figura 2.27 – Arquitetura Centralizada. ....	62
Figura 2.28 – Arquitetura com Cálculo Central. ....	63

Figura 2.29 – Fluxograma do Algoritmo de Welch-Lynch.....	67
Figura 2.30 – Broadcast do Algoritmo de Welch-Lynch. ....	68
Figura 2.31 – Biblioteca do True Time 1.5. ....	80
Figura 2.32 – Estrutura do SIDERA. ....	81
Figura 3.1 – Modelo de Simulação dos Algoritmos FTM e FTA. ....	84
Figura 3.2 – Modelo de Variação de parâmetros da simulação dos Algoritmos FTM e FTA. ....	87
Figura 3.3 – Modelo de Simulação do Excel. ....	88
Figura 3.4 – Visão funcional de uma Federação HLA. ....	90
Figura 3.5 – Arquitetura Federada. ....	93
Figura 3.6 – Malha simples de controle do atuador eletro-hidráulico de um Federado.....	93
Figura 3.7 – Malha simples de controle do atuador eletro-hidráulico de um Federado pelos blocos do TrueTime.....	94
Figura 3.8 – Bloco TrueTime Kernel.....	94
Figura 3.9 – Modelo da Arquitetura Federada no TrueTime. ....	96
Figura 3.10 – Interface de Configuração da Rede de Comunicação: (a) Configuração de rede TDMA (b) Configuração de Rede Ethernet.....	97
Figura 3.11 – Modelo de Simulação.....	98
Figura 3.12 – Interface de configuração do bloco TrueTime Kernel.....	99
Figura 3.13 – Arquitetura Integrada.....	102
Figura 3.14 – Malha simples de controle por rede de comunicação do atuador eletro-hidráulico.....	103
Figura 3.15 – Modelo do Controlador pelos blocos do TrueTime.....	103
Figura 3.16 – Modelo do Atuador Eletro-Hidráulico no TrueTime. ....	104
Figura 3.17 – Modelo do Atuador no TrueTime.....	104
Figura 3.18 – Modelo do Sensor no TrueTime. ....	105
Figura 3.19 – Modelo da Arquitetura Integrada no TrueTime.....	107
Figura 4.1 – FTM/FTA – 1º Caso Com Sincronismo. ....	115
Figura 4.2 – FTM/ FTA – Correção do 1º Caso.....	115
Figura 4.3 – FTM/FTA – 2º Caso Com Sincronismo pelo Algoritmo FTM. ....	120

Figura 4.4 – FTM/FTA – 2º Caso Com Sincronismo pelo Algoritmo FTA.....	121
Figura 4.5 – FTM/FTA – Correção do 2º Caso pelo Algoritmo FTM.....	121
Figura 4.6 – FTM/FTA – Correção do 2º Caso pelo Algoritmo FTA. ....	122
Figura 4.7 – Linha do Tempo com deriva de 10% - AF.....	128
Figura 4.8 – Resposta ao Degrau com deriva de 10% - AF.....	128
Figura 4.9 – Lei de Controle com deriva de 10% - AF.....	129
Figura 4.10 – Correção Algoritmo FTM com deriva de 10% - AF.....	129
Figura 4.11 – Linha do Tempo Sincronizada com deriva de 10% - AF. ....	130
Figura 4.12 – Resposta ao Degrau com deriva de 10% Sincronizado - AF. ..	130
Figura 4.13 – Lei de Controle com deriva de 10% Sincronizado - AF.....	131
Figura 4.14 – Linha do Tempo com viés inicial de -0.2 segundos - AF. ....	134
Figura 4.15 – Resposta ao Degrau com viés inicial de -0.2 segundos - AF...	134
Figura 4.16 – Lei de Controle com viés inicial de -0.2 segundos - AF.....	135
Figura 4.17 – Correção Algoritmo FTM com viés inicial de -0.2 segundos - AF. .....	135
Figura 4.18 – Linha do Tempo com viés inicial de -0.2 segundos – Sincronizado - AF. ....	136
Figura 4.19 – Resposta ao Degrau com viés inicial de -0.2 seg. – Sincronizado - AF. ....	136
Figura 4.20 – Lei de Controle com viés inicial de -0.2 segundos – Sincronizado - AF. ....	137
Figura 4.21 – Linha do Tempo com viés inicial de 0.39 s e deriva 10% - AF.	139
Figura 4.22 – Resposta ao Degrau com viés inicial de 0.39 s e deriva 10% - AF. .....	140
Figura 4.23 – Lei de Controle com viés inicial de 0.39 segundos e deriva 10% - AF. ....	140
Figura 4.24 – Correção Algoritmo FTM: Viés inicial: 0.39 segundos e Deriva 10% - AF.....	141
Figura 4.25 – Linha do Tempo Sincronizada: Viés inicial: 0.39 seg. e Deriva: 10% - AF.....	141

Figura 4.26 – Resposta ao Degrau Sincronizado: Viés inicial:0.39 seg, Deriva: 10% AF. ....	142
Figura 4.27 – Lei de Controle Sincronizada: Viés inicial: 0.39 seg., Deriva: 10% - AF. ....	142
Figura 4.28 – Linha do Tempo Sincronizada com deriva de 10% - AF. ....	145
Figura 4.29 –Correção com Filtro de Kalman com deriva de 10% - AF. ....	146
Figura 4.30 – Resposta ao Degrau com deriva de 10% Sincronizado - AF. ..	146
Figura 4.31 – Lei de Controle com deriva de 10% Sincronizado - AF. ....	147
Figura 4.32 – Correção do Filtro de Kalman, Viés: -0.2 segundos - AF. ....	149
Figura 4.33 – Linha do Tempo Sincronizada - Viés inicial: -0.2 segundos - AF. ....	149
Figura 4.34 – Resposta ao Degrau com viés inicial de -0.2 (s) – Sincronizado - AF. ....	150
Figura 4.35 – Lei de Controle com viés inicial de -0.2 segundos – Sincronizado – AF.....	150
Figura 4.36 – Linha do Tempo Sincronizada - Viés inicial: 0.39 (s) e Deriva: 10% - AF. ....	153
Figura 4.37 – Correção com Filtro de Kalman, Viés: 0.39 (s) e Deriva: 10% - AF. ....	153
Figura 4.38 – Resposta ao Degrau com viés inicial de 0.39 (s) e Deriva: 10% - Sincronizado - AF.....	154
Figura 4.39 – Lei de Controle: Viés inicial de 0.39(seg) e Deriva:10% - Sincronizado - AF.....	154
Figura 4.40 – Resposta ao Degrau – Arquitetura Integrada – TDMA.....	157
Figura 4.41 – Lei de Controle – Arquitetura Integrada - TDMA. ....	158
Figura 4.42 – Resposta ao Degrau – Arquitetura Integrada - Ethernet. ....	158
Figura 4.43 – Lei de Controle – Arquitetura Integrada - Ethernet.....	159
Figura 4.44 – Linha do Tempo com deriva de 1% - AI. ....	163
Figura 4.45 – Resposta ao Degrau com deriva de 1% - AI. ....	163
Figura 4.46 – Lei de Controle com deriva de 1% - AI.....	164
Figura 4.47 – Linha do Tempo Sincronizada com deriva de 1% - AI.....	164

Figura 4.48 – Correção Algoritmo FTM com deriva de 1% - AI.....	165
Figura 4.49 – Resposta ao Degrau com deriva de 1% - Sincronizado - AI. ...	165
Figura 4.50 – Lei de Controle com deriva de 1% - Sincronizado - AI.....	166
Figura 4.51 – Linha do Tempo com viés inicial de 0.39 segundos – AI.....	168
Figura 4.52 – Resposta ao Degrau com viés inicial de 0.39 segundos - AI. ..	169
Figura 4.53 – Lei de Controle com viés inicial de 0.39 segundos - AI.....	169
Figura 4.54 – Linha do Tempo Sincronizada: Viés de 0.39 s - AI.....	170
Figura 4.55 – Correção Algoritmo FTM com viés inicial de 0.39 s - AI.....	170
Figura 4.56 – Resposta ao Degrau com Viés de 0.39 s – AI - Sincronizado..	171
Figura 4.57 – Lei de Controle com Viés de 0.39 s – AI - Sincronizado. ....	171
Figura 4.58 – Linha do Tempo com Viés inicial: 0.39 segundos, Deriva: 10% - AI.....	174
Figura 4.59 – Resposta ao Degrau: Viés inicial: 0.39 segundos, Deriva: 10% - AI.....	174
Figura 4.60 – Lei de Controle: Viés inicial: 0.39 segundos, Deriva: 10% - AI.	175
Figura 4.61 – Linha do Tempo Sincronizada: Viés inicial: 0.39(s), Deriva: 10% - AI.....	175
Figura 4.62 – Correção Algoritmo FTM: Viés inicial: 0.39 (s), Deriva: 10% - AI. .....	176
Figura 4.63 – Resposta ao Degrau Sincronizada: Viés inicial:0.39(s), Deriva:10% - AI. ....	176
Figura 4.64 – Lei de Controle Sincronizada: Viés inicial: 0.39(s), Deriva: 10% - AI.....	177
Figura 4.65 – (a) Correção: Rede Ethernet, Deriva 1% (b) Linha do Tempo - AI. .....	178
Figura 4.66 – (a) Resposta ao Degrau, Deriva 1% (b) Lei de Controle - AI. ..	179
Figura 4.67 – Correção: Rede Ethernet, Viés 0.39s -AI. ....	180
Figura 4.69 – Resposta ao Degrau, Ethernet - Viés 0.39s - AI.....	181
Figura 4.70 – Lei de Controle: Ethernet - Viés 0.39s - AI. ....	182
Figura 4.71 – Correção: Rede Ethernet, Viés 0.39s, Deriva: 10% -AI.....	183
Figura 4.73 – Resposta ao Degrau, Ethernet - Viés 0.39s, Deriva: 10% - AI.	184

Figura 4.74 – Lei de Controle: Ethernet - Viés 0.39s, Deriva: 10% - AI.....	184
Figura 4.75 – Agendador da Rede TDMA Sincronizado - AI.....	186
Figura 4.76 – Agendador da Rede Ethernet Sincronizado - AI.....	187
Figura 4.77 – Fase Invertida do Caso 3 - AI.....	187
Figura 4.78 – Aproximação da Figura 4.63 no instante da inversão de fase - AI. .....	188
Figura D.1 – (a) Linha do Tempo com deriva de -10%. (b) Aproximação da Linha do Tempo com deriva de -10% .....	232
Figura D.2 – Resposta ao Degrau com deriva de -10%.....	233
Figura D.3 – Lei de Controle com deriva de -10%.....	233
Figura D.4 – Correção Algoritmo FTM com deriva de -10%.....	235
Figura D.5 – (a) Linha do Tempo com deriva de -10%. Sincronizado (b) Aproximação. ....	235
Figura D.6 – Resposta ao Degrau com deriva de -10% Sincronizado. ....	236
Figura D.7 – Lei de Controle com deriva de -10% Sincronizado.....	237
Figura D.8 – (a) Linha do Tempo com deriva de 1%. (b) Aproximação.....	238
Figura D.9 – Resposta ao Degrau com deriva de 1%. ....	239
Figura D.10 – Lei de Controle com deriva de 1%.....	239
Figura D.11 – Correção Algoritmo FTM com deriva de 1%.....	240
Figura D.12 – (a) Linha do Tempo com deriva de 1%. Sincronizado (b) Aproximação. ....	241
Figura D.13 – Resposta ao Degrau com deriva de 1% Sincronizado.....	242
Figura D.14 – Lei de Controle com deriva de 1% Sincronizado. ....	243
Figura D.15 – (a) Linha do Tempo com deriva de 10%. (b) Aproximação.....	244
Figura D.16 – Resposta ao Degrau com deriva de 10%. ....	245
Figura D.17 – Lei de Controle com deriva de 10%.....	245
Figura D.18 – Correção Algoritmo FTM com deriva de 10%.....	246
Figura D.19 – (a) Linha do Tempo com deriva de 10%. Sincronizado (b) Aproximação. ....	247
Figura D.20 – Resposta ao Degrau com deriva de 10% Sincronizado.....	248
Figura D.21 – Lei de Controle com deriva de 10% Sincronizado. ....	249

Figura E.1 – Linha do Tempo com viés inicial de -0.2 segundos. ....	252
Figura E.2 – Resposta ao Degrau com viés inicial de -0.2 segundos. ....	252
Figura E.3 – Lei de Controle com viés inicial de -0.2 segundos. ....	253
Figura E.4 – Correção Algoritmo FTM com viés inicial de -0.2 segundos. ....	254
Figura E.5 – Linha do Tempo com viés inicial de -0.2 segundos - Sincronizado. .....	254
Figura E.6 – Resposta ao Degrau com viés inicial de -0.2 segundos - Sincronizado. ....	256
Figura E.7 – Lei de Controle com viés inicial de -0.2 segundos - Sincronizado. .....	256
Figura E.8 – Linha do Tempo com viés inicial de 0.39 segundos.....	257
Figura E.9 – Resposta ao Degrau com viés inicial de 0.39 segundos.....	258
Figura E.10 – Lei de Controle com viés inicial de 0.39 segundos. ....	258
Figura E.11 – Correção Algoritmo FTM com viés inicial de -0.2 segundos....	259
Figura E.12 – Linha do Tempo com viés inicial de 0.39 segundos - Sincronizado. ....	260
Figura E.13 – Resposta ao Degrau com viés inicial de 0.39 segundos - Sincronizado. ....	261
Figura E.14 – Lei de Controle com viés inicial de 0.39 segundos - Sincronizado. .....	261
Figura F.1 – Linha do Tempo com viés inicial de -0.2 segundos e deriva -10%. .....	264
Figura F.2 – Resposta ao Degrau com viés inicial de -0.2 segundos e deriva - 10%. ....	264
Figura F.3 – Lei de Controle com viés inicial de -0.2 segundos e deriva -10%. .....	265
Figura F.4 – Correção Algoritmo FTM: Viés inicial de -0.2 segundos e Deriva - 10%. ....	266
Figura F.5 – (a)Linha do Tempo Sincronizada: Viés inicial de -0.2 segundos e Deriva -10%. (b) Aproximação. ....	267

Figura F.6 – Resposta ao Degrau Sincronizado: Viés inicial: -0.2 segundos, Deriva: -10%.	268
Figura F.7 – Lei de Controle Sincronizada: Viés inicial: -0.2 segundos, Deriva: -10%.	269
Figura F.8 – Linha do Tempo com viés inicial de 0.39 segundos e deriva 10%.	270
Figura F.9 – Resposta ao Degrau com viés inicial de 0.39 segundos e deriva 10%.	270
Figura F.10 – Lei de Controle com viés inicial de 0.39 segundos e deriva 10%.	271
Figura F.11 – Correção Algoritmo FTM: Viés inicial de 0.39 segundos e Deriva 10%.	272
Figura F.12 – Linha do Tempo Sincronizada: Viés inicial de 0.39 segundos e Deriva: 10%.	273
Figura F.13 – Resposta ao Degrau Sincronizado: Viés inicial: 0.39 segundos, Deriva: 10%.	274
Figura F.14 – Lei de Controle Sincronizada: Viés inicial: 0.39 segundos, Deriva: 10%.	274
Figura G.1 – Correção com Filtro de Kalman com deriva de -10%.	278
Figura 4.2 – Aproximação - Correção Filtro de Kalman com deriva de -10%.	279
Figura G.3 – (a) Linha do Tempo com deriva de -10%. Sincronizado (b) Aproximação.	280
Figura G.4 – Resposta ao Degrau com deriva de -10% Sincronizado.	281
Figura G.5 – (a) Lei de Controle com deriva de -10% Sincronizado. (b) Aproximação.	282
Figura G.6 – (a) Correção com Filtro de Kalman com deriva de 1%. (b) Aproximação.	284
Figura G.7 – (a) Linha do Tempo com deriva de 1%. Sincronizado (b) Aproximação.	285
Figura G.8 – Resposta ao Degrau com deriva de 1% Sincronizado.	286



Figura G.9 – (a) Lei de Controle com deriva de 1% Sincronizado. (b) Aproximação. ....	287
Figura G.10 – (a) Correção com Filtro de Kalman com deriva de 10%. (b) Aproximação. ....	289
Figura G.11 – (a) Linha do Tempo com deriva de 10%. Sincronizado (b) Aproximação. ....	290
Figura G.12 – Resposta ao Degrau com deriva de 10% Sincronizado. ....	291
Figura G.13 – (a) Lei de Controle com deriva de 10% Sincronizado. (b) Aproximação. ....	292
Figura H.1 – (a) Correção do Filtro de Kalman, Viés: -0.2 segundos. ....	296
Figura H.2 – Aproximação: Correção do Filtro de Kalman, Viés: -0.2 segundos. ....	297
Figura H.3 – Linha do Tempo Sincronizada - Viés inicial: -0.2 segundos. ....	297
Figura H.4 – Aproximação Linha do Tempo Sincronizada - Viés inicial: -0.2 segundos. ....	298
Figura H.5 – Resposta ao Degrau com viés inicial de -0.2 segundos - Sincronizado. ....	299
Figura H.6 – Lei de Controle com viés inicial de -0.2 segundos - Sincronizado. ....	299
Figura H.7 – Aproximação Lei de Controle com viés inicial de -0.2 segundos - Sincronizado. ....	300
Figura H.8 – Correção do Filtro de Kalman, Viés: 0.39 segundos. ....	302
Figura H.9 – Aproximação: Correção do Filtro de Kalman, Viés: 0.39 segundos. ....	302
Figura H.10 – Linha do Tempo Sincronizada - Viés inicial: 0.39 segundos. ....	303
Figura H.11 – Aproximação: Linha do Tempo Sincronizada - Viés inicial: 0.39 segundos. ....	303
Figura H.12 – Resposta ao Degrau com viés inicial de 0.39 segundos - Sincronizado. ....	304
Figura H.13 – Lei de Controle com viés inicial de 0.39 segundos – Sincronizado. ....	305

Figura I.1 – Correção com Filtro de Kalman, Viés: -0.2 segundos e Deriva: -10%.....	308
Figura I.2 –Aproximação - Correção com Filtro de Kalman, Viés: -0.2 segundos e Deriva: -10%. .....	309
Figura I.3 – (a) Linha do Tempo Sincronizada - Viés inicial: -0.2 segundos e Taxa de Deriva: -10%. (b) Aproximação. ....	310
Figura I.4 – Resposta ao Degrau com viés inicial de -0.2 segundos - Sincronizado. ....	311
Figura I.5 – (a) Lei de Controle: Viés inicial de -0.2 segundos e Taxa de Deriva: -10% - Sincronizado. (b) Aproximação.....	312
Figura I.6 –Aproximação - Correção com Filtro de Kalman, Viés: 0.39 segundos e Deriva: 10%.....	314
Figura I.7 – (a) Linha do Tempo Sincronizada - Viés inicial: 0.39 segundos e Taxa de Deriva: 10%. (b) Aproximação.....	316
Figura I.8 – Resposta ao Degrau com viés inicial de 0.39 segundos e Taxa de Deriva: 10% - Sincronizado.....	317
Figura I.9 – Lei de Controle: Viés inicial de 0.39(seg) e Deriva:10% - Sincronizado. ....	317
Figura J.1 – (a) Linha do Tempo com deriva de -10%. (b) Aproximação. ....	321
Figura J.2 – Resposta ao Degrau com deriva de -10%.....	322
Figura J.3 – Lei de Controle com deriva de -10%. ....	322
Figura J.4 – (a) Linha do Tempo Sincronizada com deriva de -10%.(b) Aproximação. ....	323
Figura J.5 – Correção Algoritmo FTM com deriva de -10%.....	324
Figura J.6 – Resposta ao Degrau com deriva de -10% Sincronizado. ....	324
Figura J.7 – Lei de Controle com deriva de -10% - Sincronizado.....	325
Figura J.8 –Linha do Tempo com deriva de 1%. ....	326
Figura J.9 – Aproximação: Linha do Tempo com deriva de 1%. ....	326
Figura J.10 – Resposta ao Degrau com deriva de 1%. ....	327
Figura J.11 – Lei de Controle com deriva de 1%.....	328

Figura J.12 – (a) Linha do Tempo Sincronizada com deriva de 1%.(b) Aproximação. ....	329
Figura J.13 – Correção Algoritmo FTM com deriva de 1%.....	330
Figura J.14 – Resposta ao Degrau com deriva de 1% - Sincronizado. ....	331
Figura J.15 – Lei de Controle com deriva de 1% - Sincronizado.....	331
Figura J.16 – Linha do Tempo com deriva de 10%. ....	332
Figura J.17 – Resposta ao Degrau com deriva de 10%. ....	333
Figura J.18 – Lei de Controle com deriva de 10%.....	334
Figura J.19 – (a) Linha do Tempo Sincronizada com deriva de 10%.(b) Aproximação. ....	335
Figura J.20 – Correção Algoritmo FTM com deriva de 10%.....	336
Figura J.21 – Resposta ao Degrau com deriva de 10% Sincronizado. ....	337
Figura J.22 – Lei de Controle com deriva de 10% - Sincronizado.....	337
Figura K.1 – Linha do Tempo com viés inicial de -0.2 segundos. ....	339
Figura K.2 – Resposta ao Degrau com viés inicial de -0.2 segundos - AI.....	340
Figura K.3 – Lei de Controle com viés inicial de -0.2 segundos - AI. ....	341
Figura K.4 – (a) Linha do Tempo Sincronizada: Viés de -0.2 s - AI. (b) Aproximação. ....	342
Figura K.5 – Correção Algoritmo FTM com viés inicial de -0.2 segundos. ....	343
Figura K.6 – Resposta ao Degrau com Viés de -0.2 seg. – AI - Sincronizado. .....	344
Figura K.7 – Lei de Controle com Viés de -0.2 seg. – AI - Sincronizado.....	344
Figura K.8 – Linha do Tempo com viés inicial de 0.39 segundos – AI. ....	345
Figura K.9 – Resposta ao Degrau com viés inicial de 0.39 segundos - AI. ....	346
Figura K.10 – Lei de Controle com viés inicial de 0.39 segundos - AI.....	347
Figura K.11 – (a) Linha do Tempo Sincronizada: Viés de 0.39 s - AI. (b) Aproximação. ....	348
Figura K.12 – Correção Algoritmo FTM com viés inicial de 0.39 seg. - AI. ....	349
Figura K.13 – Resposta ao Degrau com Viés de 0.39 seg. – AI - Sincronizado. .....	350
Figura K.14 – Lei de Controle com Viés de 0.39 seg. – AI - Sincronizado. ....	350

Figura L.1 – Linha do Tempo com Viés inicial: -0.2 segundos, Deriva: -10% - AI. .....	354
Figura L.2 – Resposta ao Degrau: Viés inicial: -0.2 segundos, Deriva: -10% - AI. .....	355
Figura L.3 – Lei de Controle: Viés inicial: -0.2 segundos, Deriva: -10% - AI. .	355
Figura L.4 – Correção Algoritmo FTM: Viés inicial: -0.2 segundos, Deriva: -10% - AI.....	356
Figura L.5 – (a) Linha do Tempo Sincronizada: Viés inicial: -0.2(s), Deriva: - 10% - AI. (b) Aproximação. ....	357
Figura L.6 – Resposta ao Degrau Sincronizada: Viés inicial: -0.2(s), Deriva: - 10% - AI. ....	358
Figura L.7 – Lei de Controle Sincronizada: Viés inicial: -0.2(s), Deriva: -10% - AI.....	359
Figura L.8 – Linha do Tempo com Viés inicial: 0.39 segundos, Deriva: 10% - AI. .....	360
Figura L.9 – Resposta ao Degrau: Viés inicial: 0.39 segundos, Deriva: 10% - AI. .....	361
Figura L.10 – Lei de Controle: Viés inicial: 0.39 segundos, Deriva: 10% - AI.	361
Figura L.11 – Correção Algoritmo FTM: Viés inicial: 0.39 segundos, Deriva: 10% - AI. ....	362
Figura L.12 – (a) Linha do Tempo Sincronizada: Viés inicial: 0.39(s), Deriva: 10% - AI. ....	363
Figura L.13 – Resposta ao Degrau Sincronizada: Viés inicial:0.39(s), Deriva:10% - AI.....	364
Figura L.14 – Lei de Controle Sincronizada: Viés inicial: 0.39(s), Deriva: 10% - AI.....	364
Figura M.1 –Correção com Rede Ethernet, Deriva -10% - AI.....	367
Figura M.2 – Linha do Tempo corrigida - AI. ....	368
Figura M.3 –Resposta ao Degrau, Deriva -10% - AI. ....	368
Figura M.4 – Deriva -10% - Lei de Controle - AI.....	369

Figura M.5 – (a) Correção: Rede Ethernet, Deriva 1% (b) Linha do Tempo - AI. .....	370
Figura M.6 – (a) Resposta ao Degrau, Deriva 1% (b) Lei de Controle - AI. ...	371
Figura M.7 – (a) Correção: Rede Ethernet, Deriva 10% (b) Linha do Tempo - AI. .....	372
Figura M.8 – (a) Resposta ao Degrau, Deriva 10% (b) Lei de Controle - AI. .	373
Figura N.1 – Correção: Rede Ethernet, Viés -0.2s -AI. ....	375
Figura N.3 – Resposta ao Degrau, Ethernet - Viés -0.2s - AI.....	376
Figura N.4 – Lei de Controle: Ethernet - Viés -0.2s - AI. ....	377
Figura N.5 – Correção: Rede Ethernet, Viés 0.39s -AI.....	377
Figura N.7 – Resposta ao Degrau, Ethernet - Viés 0.39s - AI.....	378
Figura N.8 – Lei de Controle: Ethernet - Viés 0.39s - AI. ....	379
Figura O.1 – Correção: Rede Ethernet, Viés -0.2s, Deriva: -10% -AI.....	381
Figura O.3 – Resposta ao Degrau, Ethernet - Viés -0.2s, Deriva: -10% - AI..	382
Figura O.4 – Lei de Controle: Ethernet - Viés -0.2s, Deriva: -10% - AI. ....	383
Figura O.5 – Correção: Rede Ethernet, Viés 0.39s, Deriva: 10% -AI.....	383
Figura O.7 – Resposta ao Degrau, Ethernet - Viés 0.39s, Deriva: 10% - AI. .	384
Figura O.8 – Lei de Controle: Ethernet - Viés 0.39s, Deriva: 10% - AI.....	385
Figura P.1 – FTM/FTA – 3º Caso Com Sincronismo.....	389
Figura P.2 – FTM/FTA – Correção do 3º Caso. ....	390
Figura P.3 – FTM/FTA – 4º Caso Com Sincronismo.....	392
Figura P.4 – FTM x FTA – Correção do 4º Caso.....	393
Figura P.5 – FTM/FTA – 5º Caso Com Sincronismo - FTM. ....	395
Figura P.6 – FTM/ FTA – 5º Caso Com Sincronismo - FTA.....	396
Figura P.7 – FTM/FTA – Correção do 5º Caso pelo Algoritmo FTM. ....	398
Figura P.8 – FTM/FTA – Correção do 5º Caso pelo Algoritmo FTA. ....	398
Figura P.9 – FTM/FTA – 6º Caso Com Sincronismo pelo Algoritmo FTM.....	401
Figura P.10 – FTM/FTA – 6º Caso Com Sincronismo pelo Algoritmo FTA. ...	402
Figura P.11 – FTM/FTA – Correção do 6º Caso pelo Algoritmo FTM. ....	404
Figura P.12 – FTM/FTA – Correção do 6º Caso pelo Algoritmo FTA.....	404
Figura P.13 – FTM/FTA – 7º Caso Com Sincronismo pelo Algoritmo FTM....	407

Figura P.14 – FTM/FTA – 7º Caso Com Sincronismo pelo Algoritmo FTA. ...	408
Figura P.15 – FTM/FTA – Correção do 7º Caso pelo Algoritmo FTM. ....	410
Figura P.16 – FTM/FTA – Correção do 7º Caso pelo Algoritmo FTA.....	410

## LISTA DE TABELAS

	<b><u>Pág.</u></b>
Tabela 2.1 – Tabela de Vantagens e Desvantagens de Famílias de Sincronização.....	59
Tabela 2.2 – Tabela de valores da flutuação em mensagens de sincronização. ....	71
Tabela 2.3 – Tabela de custo de tolerância à falhas bizantinas em relação à precisão.....	74
Tabela 4.1 – Tabela de Casos do Experimento dos Algoritmos FTA e FTA. .	111
Tabela 4.2 – Tabela de Dados do Experimento FTM/FTA – 1º Caso. ....	114
Tabela 4.3 – Tabela de Casos do Experimento com Arquitetura Federada...	124
Tabela 4.4 – Tabela de Resultados da Arquitetura Federada. ....	125
Tabela 4.5 – Tabela de Casos do Experimento com Arquitetura Integrada. ...	155
Tabela 4.6 – Tabela de Resultados da Arquitetura Integrada. ....	160
Tabela C.1 – Tabela de Dados do Experimento FTM/FTA – 1º Caso.....	213
Tabela C.2 – Dados do Experimento FTM/FTA – 2º Caso.....	215
Tabela C.3 – Modelo do Experimento FTM x FTA – 3º Caso.....	217
Tabela C.4 – Dados do Experimento FTM/FTA – 4º Caso.....	219
Tabela C.5 – Dados do Experimento FTM/FTA – 5º Caso – 6 Relógios. ....	221
Tabela C.6 – Dados do Experimento FTM – 6º Caso – 7 Relógios.....	222
Tabela C.7 – Dados do Experimento FTA – 6º Caso – 7 Relógios. ....	225
Tabela C.8 – Dados do Experimento FTM – 7º Caso – 7 Relógios.....	227
Tabela C.9 – Modelo do Experimento FTA – 7º Caso – 7 Relógios.....	229
Tabela P.1 – Tabela de Casos do Experimento dos Algoritmos FTA e FTA. .	387





## LISTA DE SIGLAS E ABREVIATURAS

AFDX	<i>Avionics Full Duplex Switched Ethernet</i>
AD	<i>Conversor Analógico-Digital</i>
AF	<i>Arquitetura Federada</i>
AI	<i>Arquitetura Integrada</i>
AS	<i>Algoritmo de Sincronização</i>
BIPM	<i>Bureau Internacional de Pesos e Medidas</i>
BROADCAST	<i>Informação enviada a todas as máquinas interligadas na rede de comunicação</i>
COTS	<i>Commercial of The Shelf</i>
Cluster	<i>Conjunto de Computadores interligados em Rede</i>
Cluster Cycle	<i>Ciclo de mensagens do Conjunto de Computadores interligados em Rede</i>
CSMA/CD	<i>Carrier Sense Multiple Access / Collision Detection</i>
CPU	<i>Central Processing Unit</i>
DA	<i>Conversor Digital-Analógico</i>
FADEC	<i>Full-Authority Digital Electronic Controller or Full-Authority Digital Engine Controller</i>
FlexRay	<i>Protocolo de Comunicação</i>
FK	<i>Filtro de Kalman</i>
FTM	<i>Fault Tolerant Mid-Point</i>
FTA	<i>Fault Tolerant Average</i>
GPS	<i>Sistema de Posicionamento Global</i>
GMT	<i>Greenwich Mean Time</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IEEE 802.3-2002	<i>Padrão da Ethernet</i>
IEEE 1588	<i>Padrão de Sincronização de Relógios</i>
INPE	<i>Instituto Nacional de Pesquisas Espaciais</i>
IP	<i>Instante de Pico</i>
MHZ	<i>MegaHertz</i>

NTP	<i>Network Time Protocol</i>
NOP	<i>No Operation</i>
O	<i>Overshoot</i>
PLL	<i>Phase Loop Locker</i>
PITCH	<i>Eixo Y</i>
PD	<i>Proporcional-Derivativo</i>
PID	<i>Proporcional-Integral-Derivativo</i>
ROW	<i>Eixo X</i>
RF	<i>Relógio com Falha</i>
SPG	<i>Serviço de Pós-Graduação</i>
SISO	<i>Single-Input-Single-Output</i>
SID	<i>Serviço de Informação e Documentação</i>
SIDERA	<i>Simulation environment for Dependable Realtime Architectures</i>
TTP	<i>Transmission Control Protocol</i>
TTP/A	<i>Time-Triggered Protocol versão A</i>
TTP/C	<i>Time-Triggered Architecture versão C</i>
TTA	<i>Time-Triggered Architecture</i>
TD	<i>Taxa de Deriva</i>
TDMA	<i>Time Divison Multiple Access</i>
TDMA ROUND	<i>Ciclo do Time Divison Multiple Access</i>
TCP	<i>Transmission Control Protocol</i>
TAI	<i>Tempo Atómico Internacional</i>
TDI	<i>Teses e Dissertações Internas</i>
TTETHERNET	<i>Time-Triggered Ethernet</i>
UDP	<i>User Datagram Protocol</i>
UTC	<i>Tempo Universal Coordenado</i>
VI	<i>Viés Inicial</i>
VP	<i>Valor de Pico</i>
YAW	<i>Eixo Z</i>

## LISTA DE SÍMBOLOS

ADJ	Ajuste do Relógio
A	Vetor de Dados do Relógio
$a_1$	Valor da Pendente do Relógio 1
$a_2$	Valor da Pendente do Relógio 2
$\Delta a_k^{2,1}$	Diferença entre as Pendentes do Relógio 1 e do Relógio 2 no instante k
$\Delta_{i,j}$	Vetor de diferenças entre relógios
$b_1$	Viés inicial do Relógio 1
$b_2$	Viés inicial do Relógio 2
$C^2$	Função do Relógio 2
$C^1$	Função do Relógio 1
cfn	Função de Média/Mediana do Algoritmo de Correção
CM	Ciclo de Máquina
D	Deriva
e	Evento
f	Número de Falhas
$g^{\mu T}$	Granularidade do microtick
$g^{CM}$	Granularidade do Ciclo de Máquina
$g^r$	Granularidade do Relógio r
$g^i$	Granularidade do Relógio i
$g^1$	Granularidade do Relógio 1
$g^{tg}$	Granularidade do Tempo Global
$K_p$	Ganho Proporcional
$K_D$	Ganho Derivativo
$K_i$	Ganho Integral
k	Define o instante do tempo
$M_p$	Sobre sinal máximo
mT	Macrotick
$mT_k$	Macrotick no instante k
n	Número de relógios no conjunto

$off_k^{2,1}$	Viés instantâneo dos relógios 2 e 1
$PC(t)$	Tempo do relógio físico
$R_{int}$	Intervalo de Re-Sincronização
$T_p$	Instante do Pico
$T_r$	Tempo de Subida
$t_a$	Tempo de atraso
$t_d$	Tempo de Acomodação
$t_k^L$	Tempo no instante k do relógio L
$t_k^i$	Tempo no instante k do relógio i
$t_k^R$	Tempo no instante k do relógio R
$t_k^1$	Tempo no instante k do relógio 1
$t^1(e)$	Instante de Tempo do Evento e
$t$	Tempo de Referência
$T$	Viés
$\mu(n,f)$	Função de Compensação do Erro Bizantino
$VC(t)$	Relógio Virtual
$W$	Ruído Branco
$\mu T$	Microtick
$\mu T_k$	Microtic no Instante k
$\rho$	Taxa de Deriva
$\delta$	Máximo Atraso da Rede
$\Phi$	Função de Convergência
$\varepsilon$	Erro
$\Pi$	Precisão
$\Gamma$	Máximo viés instantâneo

## SUMÁRIO

	<u>Pág.</u>
<b>1 INTRODUÇÃO .....</b>	<b>1</b>
1.1. Objetivos .....	3
1.2. Motivação e Justificação .....	4
1.3. Organização do Trabalho .....	6
<b>2 CONCEITOS BÁSICOS E REVISÃO DA LITERATURA.....</b>	<b>9</b>
2.1 Sistemas de Controle .....	9
2.1.1 Controle Clássico e Controle Moderno .....	9
2.1.2 Controle Digital .....	15
2.1.3 Sistemas de Controle em Tempo Real .....	16
2.2 Arquitetura Federada vs. Integrada .....	18
2.2.1 Vantagens de uma Arquitetura Federada .....	19
2.2.2 Vantagens de uma Arquitetura Integrada .....	20
2.3 Protocolos de Comunicação .....	21
2.3.1 Time Triggered Protocol - TTP .....	21
2.3.2 Ethernet.....	25
2.4 Teoria dos Relógios .....	27
2.4.1 Relógios Físicos.....	31
2.4.2 Precisão e Exatidão .....	43
2.4.3 Padrões de Tempo e GPS.....	46
2.4.4 Relógio Lógico .....	47
2.4.5 Tempo Global .....	48
2.4.6 Sincronização Interna .....	54
2.4.7 Sincronização Externa.....	56
2.4.8 Família de Soluções.....	58
2.5 Arquiteturas de Sincronização .....	59
2.5.1 Arquitetura Distribuída .....	60
2.5.2 Arquitetura Centralizada.....	61
2.5.3 Arquitetura com Cálculo Central .....	63
2.6 Algoritmos de Sincronização .....	64
2.6.1 Algoritmo Welch-Lynch .....	65

2.6.2 Algoritmo FTA (Fault-Tolerant Average) .....	71
2.7 Filtro de Kalman.....	74
2.7.1 Modelo Matemático .....	76
2.8 Ferramentas de Simulação .....	79
2.8.1 TrueTime .....	79
2.8.2 SIDERA.....	81
<b>3 FORMULAÇÃO DO PROBLEMA E ABORDAGENS PARA SUA SOLUÇÃO .....</b>	<b>83</b>
3.1. Análise dos Algoritmos de Sincronização de Relógios FTM e FTA ...	83
3.1.1. Metodologia .....	84
3.1.2. Modelo Matemático .....	85
3.1.3. Variação dos Parâmetros .....	87
3.1.4. Modelo de Simulação.....	88
3.2. Sincronização de Relógios em Sistemas de Controle sobre uma Arquitetura Federada. ....	89
3.2.1. Metodologia .....	91
3.2.2. Modelo do Sistema.....	92
3.2.3. Variação da Rede de Comunicação.....	97
3.2.4. Modelo da Sincronização de Relógios .....	97
3.2.5. Avaliação dos efeitos da sincronização.....	100
3.3. Sincronização de Relógios em Sistemas de Controle sobre uma Arquitetura Integrada.....	100
3.3.1. Metodologia .....	101
3.3.2. Modelo do Sistema.....	102
3.3.3. Variação da Rede de Comunicação.....	108
3.3.4. Modelo da Sincronização de Relógios .....	108
3.3.5. Avaliação dos efeitos da sincronização.....	109
<b>4 RESULTADOS E SIMULAÇÕES.....</b>	<b>111</b>
4.1. Análise e Simulação dos Algoritmos de Sincronização de Relógios FTM ( <i>Fault-Tolerant Mid-Point</i> ) e FTA ( <i>Fault-Tolerant Average</i> ) .....	111
4.1.1. Projeto dos Experimentos .....	111
4.1.2. Caso 1 – 4 Relógios, um relógio com erro de deriva .....	112

4.1.3. Caso 2 – 7 Relógios - Erro de deriva .....	116
4.2. Sincronização de Relógios em Sistemas de Controle sobre uma Arquitetura Federada.....	123
4.2.1. Projeto dos Experimentos.....	123
4.2.2. Caso 1 – FTM - Relógio com Deriva .....	126
4.2.3. Caso 2 – FTM - Relógio com Viés Inicial.....	131
4.2.4. Caso 3 - FTM– Relógio com Deriva e Viés Inicial .....	137
4.2.5. Caso 4 – Filtro de Kalman – Relógio com Deriva .....	143
4.2.6. Caso 5 – Filtro de Kalman – Relógio com Viés Inicial.....	147
4.2.7. Caso 6 – Filtro de Kalman – Relógio com Deriva e Viés Inicial.....	151
4.3. Sincronização de Relógios em Sistemas de Controle sobre uma Arquitetura Integrada.....	155
4.3.1. Projeto dos Experimentos.....	155
4.3.2. Caso 1 – Relógio com Deriva - TDMA.....	161
4.3.3. Caso 2 – Relógio com Viés Inicial - TDMA.....	166
4.3.4. Caso 3 – Relógio com Deriva e Viés Inicial - TDMA .....	172
4.3.5. Caso 4 – Relógio com Deriva - Ethernet .....	177
4.3.6. Caso 5 – Relógio com Viés Inicial - Ethernet.....	180
4.3.7. Caso 6 – Relógio com Deriva e Viés Inicial - Ethernet.....	182
4.3.8. Caso 3 - TDMA versus Caso 6 - Ethernet.....	184
5 CONCLUSÕES E SUGESTÕES PARA TRABALHOS FUTUROS.....	189
REFERÊNCIAS BIBLIOGRÁFICAS.....	195
GLOSSÁRIO.....	201
APÊNDICE A - CÓDIGO DAS FUNÇÕES CHAMADAS PELA SIMULAÇÃO DA ARQUITETURA FEDERADA .....	203
APÊNDICE B - CÓDIGO DAS FUNÇÕES CHAMADAS PELA SIMULAÇÃO DA ARQUITETURA INTEGRADA.....	207
B.1 EXEMPLO DE FUNÇÃO DO SENSOR .....	207
B.2 EXEMPLO DE FUNÇÃO DO CONTROLADOR .....	207
APÊNDICE C – TABELAS DE DADOS DA SIMULAÇÃO COM EXCEL .....	213
APÊNDICE D – SIMULAÇÃO DO 1º CASO – ARQUITETURA FEDERADA – ALGORITMO FTM.....	231

<b>APÊNDICE E – SIMULAÇÃO DO 2º CASO – ARQUITETURA FEDERADA</b>	
– ALGORITMO FTM .....	251
<b>APÊNDICE F – SIMULAÇÃO DO 3º CASO – ARQUITETURA FEDERADA</b>	
– ALGORITMO FTM .....	263
<b>APÊNDICE G – SIMULAÇÃO DO 4º CASO – ARQUITETURA FEDERADA</b>	
– ALGORITMO FK.....	277
<b>APÊNDICE H – SIMULAÇÃO DO 5º CASO – ARQUITETURA FEDERADA</b>	
– ALGORITMO FK.....	295
<b>APÊNDICE I – SIMULAÇÃO DO 6º CASO – ARQUITETURA FEDERADA</b>	
– ALGORITMO FK.....	307
<b>APÊNDICE J – SIMULAÇÃO DO 1º CASO – ARQUITETURA INTEGRADA</b>	
– REDE TDMA.....	319
<b>APÊNDICE K – SIMULAÇÃO DO 2º CASO – ARQUITETURA INTEGRADA</b>	
– REDE TDMA.....	339
<b>APÊNDICE L – SIMULAÇÃO DO 3º CASO – ARQUITETURA INTEGRADA</b>	
– REDE TDMA.....	353
<b>APÊNDICE M – SIMULAÇÃO DO 4º CASO – ARQUITETURA INTEGRADA</b>	
– REDE ETHERNET .....	367
<b>APÊNDICE N – SIMULAÇÃO DO 5º CASO – ARQUITETURA INTEGRADA</b>	
– REDE ETHERNET .....	375
<b>APÊNDICE O – SIMULAÇÃO DO 6º CASO – ARQUITETURA INTEGRADA</b>	
– REDE ETHERNET .....	381
<b>APÊNDICE P – SIMULAÇÃO DO ALGORITMO FTM/FTA .....</b>	<b>387</b>
<b>APÊNDICE Q – MODELO DO FILTRO DE KALMAN.....</b>	<b>413</b>



## 1 INTRODUÇÃO

Sistemas de controle de tráfego aéreo, terrestre, ferroviário, usinas nucleares, controles industriais, estações espaciais, satélites artificiais são exemplos de aplicações complexas e altamente integradas onde o mau funcionamento do sistema pode provocar grandes perdas econômicas, causar danos ao meio ambiente, ou mesmo infringir perdas de vidas humanas. Conseqüentemente, existe uma grande preocupação no mundo da engenharia com a confiabilidade dos sistemas computacionais e de comunicações utilizados nos sistemas de controle destes tipos de aplicações complexas e altamente integradas.

Tais exemplos utilizam sistemas de controle que estão se tornando cada vez mais complexos e altamente integrados, integrando a comunicação e a computação com o controle de sistemas. Nestes casos, a tradicional arquitetura ponto a ponto (isto é, onde existe um cabo que conecta o computador central com cada sensor ou atuador) que vinha sendo implementada até então, não é mais satisfatória. Por isto, expandindo as funcionalidades dos sistemas, chega-se aos limites da arquitetura ponto a ponto; e o tradicional sistema centralizado ponto a ponto, não é mais adequado para satisfazer os novos requisitos tais como, redução do número de fios e seus inconvenientes.

A principal vantagem de uma arquitetura distribuída é que ela permite a composição de subsistemas, sendo que as características de um subsistema não devem ser alteradas quando parte de um sistema maior. No contexto de sistemas de tempo real, a composição requer que a interface entre os hospedeiros e a rede de comunicação seja completamente especificada, não só no domínio lógico (valores e transições), mas também no domínio temporal (instantes e intervalos).

Além disto, os requisitos temporais de um sistema de tempo real devem ser obedecidos estritamente e com alta precisão. Isto cria a necessidade de se trabalhar com relógios de alta precisão e sincronizados.

A maioria dos modernos sistemas de comunicação digital não transmite dados sem o uso de relógios. Com isso, a arquitetura distribuída de nosso interesse, é interligada por um barramento de comunicação e possui em cada nó, um relógio. A inclusão de um relógio em cada nó pode trazer muito benefícios, mas traz um problema para o sistema, principalmente porque os relógios sofrem com a dessincronização. A dessincronização, normalmente, esta relacionada às imperfeições de um relógio, que incluem a deriva (“*drift*”), o viés (“*offset*”) e a flutuação (“*jitter*”).

No setor aeronáutico, controles de tráfego aéreo estão cada vez mais eficientes e os aviões conseguem pousar e decolar com uma diferença de tempo cada vez menor, tornado estas aplicações dependentes de uma sincronização de relógios eficiente (alta precisão e rápida convergência).

No setor aeroespacial, existem muitos problemas, Einwoegerer (2009) publicou um trabalho que desenvolve uma solução para navegação de satélites artificiais com o uso de uma Unidade de Medida Inercial (UMI) integrada com medidas do sistema *GPS*, onde a UMI é uma tecnologia cuja precisão é dependente do tempo.

Para resolver e/ou minimizar os problemas de sincronização de relógios existem vários tipo de soluções, uma delas são os algoritmos de sincronização baseados em lógicas matemáticas que corrigem e/ou minimizam, os problemas causados pelos relógios. Lamport e Melliar-Smith (1985) publicaram um algoritmo de sincronização de relógios de sistemas distribuídos tolerante a imperfeições conhecido como Algoritmo de Lamport. Lundelius e Lynch (1984) publicaram outro algoritmo de sincronização de relógios de sistemas distribuídos tolerante a imperfeições conhecido como Welch-Lynch ou FTM (*Fault-Tolerant Mid-Point*). Os algoritmos de Lamport e

Welch-Lynch diferem no método de leitura dos relógios e no cálculo da função de convergência. Existem ainda outros algoritmos e protocolos de sincronização, tal como o padrão *IEEE 1588*, que é um protocolo de sincronização, originalmente desenvolvido pela empresa *Agilent Technologies*, para instrumentação e controle de tarefas distribuídas. Além do uso de algoritmos de sincronização de relógios, existem também outras possibilidades, como o uso do filtro de Kalman para estimar as imperfeições dos relógios e permitir compensá-las parcialmente.

As possibilidades do relógio prejudicar um sistema de controle distribuído de tempo real motivaram o desenvolvimento de uma investigação sobre os algoritmos de sincronização de relógios existentes, suas características e suas influências sobre um sistema de controle distribuído.

### **1.1. Objetivos**

Este trabalho tem por objetivo principal estudar os algoritmos de sincronização de relógios Welch-Lynch (FTM), FTA e Filtro de Kalman e suas influências sobre um sistema de controle distribuído. Para isto, foram escolhidos:

- a) o algoritmo de sincronização de relógios Welch-Lynch (*FTM – Fault-Tolerant Mid-Point*);
- b) Filtro de Kalman para estimar as imperfeições do relógio e compensá-las;
- c) o algoritmo de sincronização de relógios FTA (*Fault-Tolerant Average*);
- d) Duas arquiteturas de controle distribuído (Federada e Integrada).

Para alcançar o objetivo o trabalho adota os seguintes passos:

- a) Busca na literatura e discute os algoritmos de sincronização de relógios Welch-Lynch (FTM), FTA e Filtro de Kalman;
- b) Estuda os algoritmos de sincronização de relógios Welch-Lynch e FTA (*Fault-Tolerant Average*), destacando suas características, vantagens e desvantagens;
- c) Simula o algoritmo de sincronização de relógios Welch-Lynch, em um Sistema de Controle PID com arquitetura Federada, utilizando um protocolo de comunicação com acesso TDMA;
- d) Simula o algoritmo de sincronização Welch-Lynch, em um Sistema de Controle PID com arquitetura integrada, utilizando um protocolo de comunicação com acesso TDMA;
- e) Simula o algoritmo de sincronização Welch-Lynch, em um Sistema de Controle PID com arquitetura integrada, utilizando um protocolo de comunicação com acesso Ethernet;
- f) Utiliza o Filtro de Kalman para estimar as imperfeições dos relógios (deriva e viés), compensá-las parcialmente e sincronizar os relógios dos nós em um Sistema de Controle PID com arquitetura federada, utilizando um protocolo de comunicação com acesso TDMA;

## **1.2. Motivação e Justificação**

A maior tendência atual em sistemas complexos e altamente integrados é a de integrar a computação e a comunicação com o controle de sistemas nos diferentes níveis de operação industrial e/ou da maquinaria. A arquitetura tradicional para controle de sistemas, a qual tem sido implementada satisfatoriamente na indústria, é a ponto a ponto, isto é, existe um cabo que

conecta o computador central com cada sensor ou atuador. Entretanto, expandindo as funcionalidades dos sistemas, chega-se aos limites da arquitetura ponto a ponto. Assim, o tradicional sistema centralizado não é o mais adequado para atingir os novos requisitos tais como, redução do número de fios e seus inconvenientes (tempo, custo, qualidade na instalação, manutenção, modificação, retirada) modularidade, descentralização do controle, rapidez, facilidade de manutenção e baixo custo, entre outras, (note-se que muitas destas características também são procuradas por sistemas tolerantes à falhas). A introdução de uma arquitetura de rede com um barramento comum pode melhorar a eficiência, flexibilidade e confiabilidade destas aplicações integradas, e reduzir os tempos de instalação, configuração, manutenção e, sobretudo, custo (RAJI, 1994). A confiabilidade aumenta com a redução de número de conectores e cabos. Em ambientes automotivos 30% da falhas apresentadas são falhas elétricas atribuídas por conectores com problemas (Obermaisser; Peti; Tagliabo, 2007). Com a implementação distribuída, os sistemas que devem obedecer, principalmente, às restrições de causalidade e temporalidade, típicas de um sistema de tempo real, necessitando de sincronismo de relógios em suas operações, crescendo assim a demanda por eficientes algoritmos de sincronização de relógios.

No âmbito aeroespacial, de acordo com Einwoegerer (2009), com o uso de uma Unidade de Medida Inercial (UMI) baseada em tecnologia *Micro Electro-Mechanical System* (MEMS) quando integrada com medidas do sistema *GPS*, permite implementar aplicações de navegação de baixo custo como as unidades de navegação *UMI/GPS*.

Nestes contextos, é de suma importância que existam estudos a respeito da viabilidade, da eficiência e de como os algoritmos de sincronização de relógios influenciam o sistema de controle, principalmente no âmbito das aplicações aeroespaciais, tônica deste trabalho.

Os exemplos de aplicação escolhidos procuram realçar algumas das vantagens da utilização dos algoritmos de sincronização de relógios na sua adoção e de como esses algoritmos podem influenciar no sistema de controle.

Por isto nos propusemos a estudar, modelar, entender e divulgar a influência dos algoritmos de sincronização de relógios Welch-Lynch (FTM), FTA e Filtro de Kalman em um sistema de controle distribuído de tempo real.

### **1.3. Organização do Trabalho**

Esse trabalho se divide em cinco Capítulos, com a seguinte abordagem:

- O primeiro capítulo inclui a introdução, motivação e objetivos.
- O segundo capítulo inclui a revisão da literatura e os conceitos básicos que suportam o entendimento de Sistemas de Controle por Rede, os tipos de barramentos de comunicação de dados utilizados neste trabalho, a teoria sobre relógios e sincronização de relógios e suas principais características.
- O terceiro capítulo apresenta a metodologia aplicada. Inclui a formulação do problema e a modelagem propriamente dita. São discutidos os elementos que compõem o modelo bem como suas interligações.
- O quarto capítulo apresenta as simulações e os resultados. Inclui as simulações e seus desdobramentos, bem como a discussão dos resultados. Inclui também, o resultado da pesquisa realizada na área do sincronismo de relógios e a sua influência sobre sistemas de controle por rede, uma análise individual de vários casos de dessincronização de relógios em sistemas de controle integrados e federados.

- O quinto capítulo inclui a conclusão do trabalho e uma sugestão de trabalhos futuros, no intuito de dar continuidade ao mesmo.
- Os Apêndices contém as simulações de vários casos de estudo.





## 2 CONCEITOS BÁSICOS E REVISÃO DA LITERATURA

### 2.1 Sistemas de Controle

#### 2.1.1 Controle Clássico e Controle Moderno

Ogata (1997) abrange em seu texto tanto a Teoria de Controle Clássico quanto a Teoria de Controle Moderno. Em (Lustosa, 2009) é feita uma excelente revisão bibliográfica sobre o assunto, onde é dito que a resposta transitória de um sistema de controle geralmente apresenta oscilações amortecidas antes de alcançar o estado ou regime estacionário. Ao especificar as características de resposta transitória de um sistema de controle, geralmente para uma entrada a degrau, é comum especificar os seguintes indicadores:

- Tempo de atraso,  $t_d$ : é o tempo necessário para a resposta alcançar pela primeira vez a metade do valor final;
- Tempo de subida,  $t_r$ : é o tempo necessário para a resposta passar de 10% a 90%, 5% a 95% ou 0% a 100% do seu valor final. Para sistemas de segunda ordem sub-amortecidos, usa-se normalmente o tempo de subida de 0% a 100%. Para sistemas sobre-amortecidos, usa-se normalmente o tempo de subida de 10% a 90%;
- Instante do pico,  $t_p$ : é o tempo necessário para a resposta alcançar o primeiro pico de sobre-sinal;
- Sobre-sinal máximo,  $M_p$ : o sobre sinal máximo é o máximo valor de pico da curva de resposta medido a partir do seu valor final. Também se usa o máximo sobre-sinal percentual. O sobre-sinal máximo percentual indica indiretamente a estabilidade relativa do sistema;

- Tempo de acomodação,  $t_s$ : é o tempo necessário para a curva de resposta alcançar e permanecer dentro de uma faixa em torno do seu valor final, faixa essa de magnitude especificada por uma porcentagem absoluta do valor final (geralmente 5% ou 2%). O tempo de acomodação está relacionado com a constante de tempo do sistema de controle. A escolha de que porcentagem usar no critério de erro pode ser determinada a partir dos objetivos do projeto do sistema em questão.

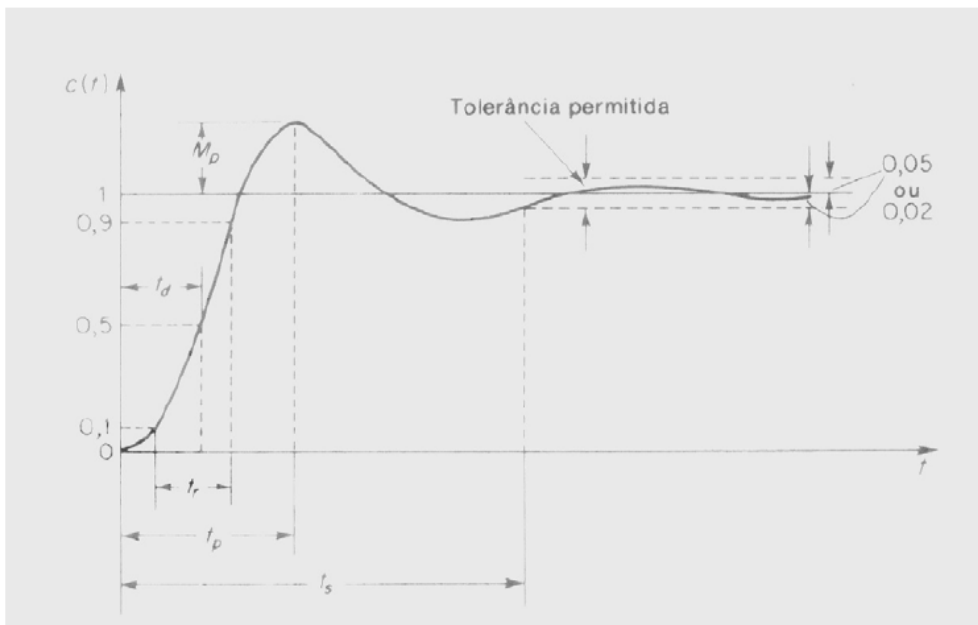


Figura 2.1 – Especificações da resposta transitória a um degrau unitário.

Fonte: Ogata (1997).

Goodwin, Graebe e Salgado (2000) também apresentam tanto a Teoria de Controle Clássico quanto a Teoria de Controle Moderno. Tal livro foi utilizado por Gobato (2006), como referência no cálculo dos ganhos dos controladores SISO PID, pela metodologia da alocação de pólos dominantes.

As estruturas de controle clássico mais comumente usada são a PID, rede *lead* e rede *lag*, ou redes de avanço e atraso de fase.

De acordo com Goodwin, Graebe e Salgado (2000), a simplicidade dessas estruturas são, ao mesmo tempo, seus pontos fracos. Enquanto se pode provar que a estrutura PID é robusta em diversas aplicações, ela não é recomendável em alguns casos particulares de plantas instáveis. No entanto, a versatilidade do controle PID (controlador de até segunda ordem) garante ainda sua continuidade e popularidade.

Considerando o sistema SISO mostrado na Figura 2.2, as expressões tradicionais para controladores PID estão mostradas nas eqs. 2.1, 2.2, 2.3 e 2.4:

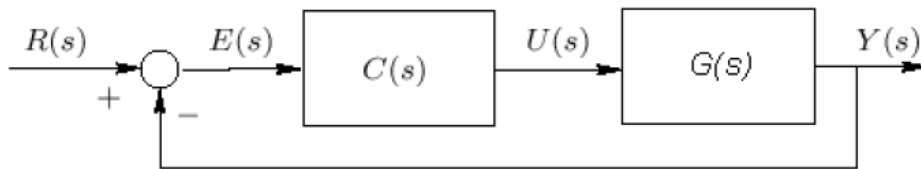


Figura 2.2 – Componentes essenciais de um Sistema de Controle..

Fonte: Gobato (2006).

$$C_p(s) = K_p \quad (2.1)$$

$$C_{PI}(s) = K_p + \frac{1}{K_I s} \quad (2.2)$$

$$C_{PD}(s) = K_p + sK_D \quad (2.3)$$

$$C_{PID}(s) = K_p + \frac{1}{K_I s} + sK_D \quad (2.4)$$

Nesse caso  $K_p$ ,  $K_D$  e  $K_I$  são conhecidos como **ganho proporcional**, **ganho derivativo** e **ganho integral** respectivamente. Fica evidente que os membros dessa estrutura incluem, em diferentes combinações, três ações de controle básicas: Proporcional (P), Integral (I) e Derivativo (D).

De acordo com Ogata (1997):

A **ação proporcional de controle** provê uma contribuição que depende do valor instantâneo do erro, definido pela diferença entre o valor real e o valor

desejável. Um controlador proporcional pode controlar qualquer planta estável, mas oferece desempenho limitado e erro estacionário não nulo. Essa última limitação decorre do fato de que sua resposta em frequência é limitada para todas as frequências.

A **ação integral de controle** (aproximada pela **rede lag**) provê uma contribuição que é proporcional ao erro acumulado, implicando um modo de controle com ação mais lenta. Esse modo, analisado separadamente, tem duas grandes desvantagens: a primeira é o fato de o pólo na origem ser altamente prejudicial à estabilidade da malha; a segunda é o fato de dar margem ao efeito de *wind-up* (entrada atinge o limite de saturação e continua sendo integrada na malha). Em contrapartida, a ação integral de controle tem a vantagem de zerar o erro em regime estacionário.

A **ação derivativa de controle** (aproximada pela **rede lead**) atua na taxa de mudança do erro, implicando uma ação de controle rápida que desaparece na presença de erros constantes. É, muitas vezes, chamada de modo preditivo por ser dependente da tendência de variação do erro. A maior limitação do modo derivativo, quando analisado isoladamente, é a geração de ações de controle de amplitude elevada em resposta a erros de frequência elevada, tais como ruído na medição.

Ainda de acordo com Ogata (1997), uma função de transferência de um sistema de segunda ordem em malha fechada pode ser representada da seguinte forma:

$$\frac{C(s)}{R(s)} = \frac{K}{Js^2 + Fs + K} = \frac{\frac{K}{J}}{\left[ s + \frac{F}{2J} + \sqrt{\left(\frac{F}{2J}\right)^2 - \frac{K}{J}} \right] \cdot \left[ s + \frac{F}{2J} - \sqrt{\left(\frac{F}{2J}\right)^2 - \frac{K}{J}} \right]} \quad (2.5)$$

Os pólos de malha fechada são complexos se  $F^2 - 4JK < 0$ , e são reais se  $F^2 - 4JK \geq 0$ . Na análise da resposta transitória, é conveniente escrever:

$$\frac{K}{J} = \omega_n^2 \quad (2.6)$$

$$\frac{F}{J} = 2 \cdot \zeta \cdot \omega_n = 2 \cdot \sigma \quad (2.7)$$

Em que  $\sigma$  é chamada de atenuação,  $\omega_n$  é a frequência natural não amortecida e  $\zeta$  é o coeficiente ou razão de amortecimento do sistema.

O comportamento dinâmico de sistemas de segunda ordem pode então ser descrito em termos de dois parâmetros:  $\zeta$  e  $\omega_n$ .

Em muitos casos práticos as características de desempenho desejadas de um sistema de controle são especificadas em termos de grandezas no domínio do tempo. Sistemas com armazenamento de energia não podem responder instantaneamente e terão respostas transitórias sempre que sujeitos a entradas ou perturbações.

Nem todas as especificações acima são aplicáveis para qualquer caso em estudo. Por exemplo, um sistema sobre-amortecido, os termos instante do pico e sobre-sinal máximo não se aplicam.

As características de resposta transitória são facetas importantes do desempenho de sistemas de controle. Uma outra característica importante se relaciona com o erro do sistema. Os erros de um sistema de controle podem ser atribuídos a muitos fatores. Variações na entrada de referência causarão erros durante períodos transitórios e podem também ocasionar erros estacionários.

Qualquer sistema de controle físico sofre inerentemente de erro estacionário em resposta a certos tipos de entradas. Um sistema pode não apresentar erro estacionário para entradas em degrau, mas o mesmo pode apresentar erro estacionário não nulo para uma entrada em rampa. A única maneira de se eliminar este erro é modificando a estrutura do sistema.

Portanto, os seguintes critérios de projeto do sistema de controle serão observados:

- $0.4 < \zeta < 0.8$ : que mantém o compromisso razoável entre velocidade de resposta e sobre-sinal máximo;
- $t_s = \frac{4}{\zeta \cdot \omega_n}$  : que garante uma faixa de tolerância de 2% para o erro em regime estacionário em aproximadamente quatro vezes a constante de tempo do sistema;
- As escolhas de  $\zeta$  e  $\omega_n$  baseado nos critérios acima vai resultar em um sobre-sinal máximo percentual de aproximadamente  $M_p = e^{-\left(\frac{\sigma}{\omega_d}\right)\pi} \cdot 100\%$ , onde  $\omega_d = \omega_n \cdot \sqrt{(1 - \zeta^2)}$  é a frequência natural amortecida do sistema.

Quando um sistema de segunda ordem tem um zero perto dos pólos de malha fechada (por exemplo, a planta de “roll”, pitch ou “yaw” controlada por um controlador PD) o comportamento da resposta transitória pode se tornar consideravelmente diferente daquele esperado pela escolha dos parâmetros descritos acima. Nesse caso o ajuste dos parâmetros de projeto deve ser feito para compatibilizar a resposta transitória aos requisitos da especificação.

A introdução da ação integral ao controlador PD, resultando no controlador PID, aumenta para terceira ordem a função de transferência em malha fechada das malhas de “roll”, “pitch” e “yaw”. A resposta temporal de um sistema estável de terceira ordem é a soma de certo número de curvas senoidais e exponenciais amortecidas.

### 2.1.2 Controle Digital

A tendência atual para o uso de controle digital ao invés de se usar o controle analógico para sistemas dinâmicos se dá principalmente devido à disponibilidade de computadores digitais a baixo custo e à grande vantagem da flexibilidade de se programar o controlador em um computador digital (Ogata, 1997). Neste, se trabalha com sinais digitais (discretamente definidos no tempo, quantizadamente definidos em amplitude, e não instantaneamente/atrasadamente produzidos no tempo) ao invés de sinais analógicos (continuamente definidos no tempo, continuamente definidos em amplitude, e instantaneamente produzidos no tempo) (Souza, 1990).

Segundo Moreira (2006), sistemas de controle discretos diferem de sistemas de controle analógicos devido ao fato de seus sinais se encontrarem na forma discreta no tempo. Um sinal discreto no tempo é um sinal definido somente em instantes discretos de tempo. Se um sinal discreto no tempo originar-se por amostragem de um sinal analógico em instantes discretos de tempo, então o sinal é chamado de sinal amostrado. Na prática, muitos sinais digitais são obtidos através de conversores analógico-digitais, que além de amostrar no tempo o sinal, transforma-o em um sinal quantizado na amplitude.

A Figura 2.3 mostra o diagrama de blocos de um sistema de controle digital. Os elementos básicos desse diagrama estão nos blocos. A saída da planta é analógica, assim como sua comparação com a referência. Conversores analógicos-digitais (AD) transformam os sinais das saídas analógicas dos sensores em entradas digitais, para o controlador digital; e conversores digitais-analógicos (DA) transformam as saídas digitais do controlador em entradas analógicas para os atuadores. Tais conversores são gerenciados pelo relógio do computador (*clock*).

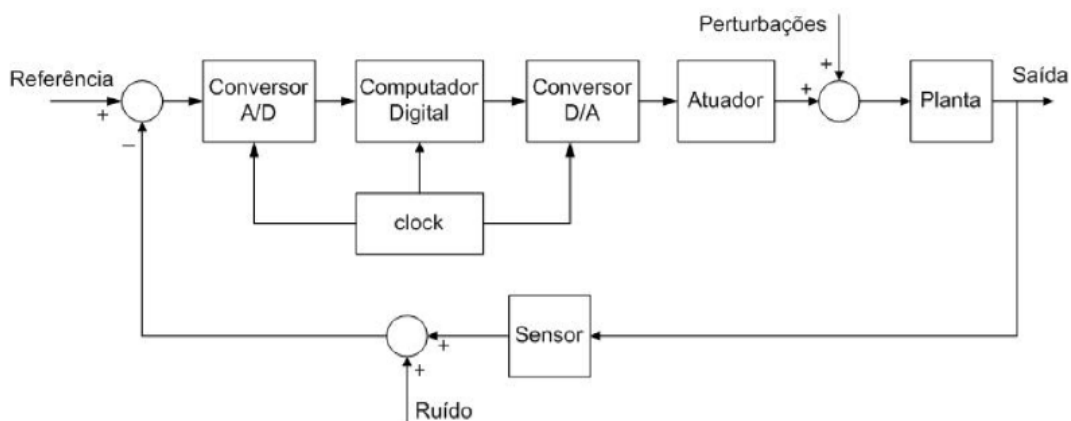


Figura 2.3 – Diagrama de blocos de um sistema de controle analógico-digital.  
 Fonte adaptada: Ogata (1997).

O que diferencia as técnicas de discretização é a forma como se dá a aproximação da integral contínua pela somatória discreta. As regras de Euler: retangular para frente e retangular para trás aproximam a área representada pela integral por retângulos com largura do período de discretização ( $T$ ) e amplitude do valor da função no instante anterior e posterior de discretização, respectivamente. Porém, a literatura de controle consultada (FRANKLIN e POWELL, 1981), mostra que a regra de Euler para frente pode fazer com que uma função de transferência contínua estável no plano  $s$  seja mapeada na região instável do plano  $z$ . A regra de Euler para trás mapeia o semi-plano estável em um círculo de raio  $1/2$  na região estável do plano  $z$ .

O presente trabalho utilizou a regra de Dormand-Prince para a discretização dos controladores, como sugerido pelo fabricante da ferramenta True Time®.

### 2.1.3 Sistemas de Controle em Tempo Real

Segundo Stankovic (1988), na computação de tempo real a correção de um processo depende não somente do resultado lógico ou numérico da computação, mas também do tempo no qual os resultados são produzidos. Isso quer dizer que o resultado de tempo é muito importante em um sistema



de controle em tempo real. Os problemas de variação do tempo em sistemas de controle distribuídos, principalmente em sistemas de tempo real, é um problema inerente ao sistema que pode prejudicar o resultado temporal do sistema.

Para o controle do tempo, normalmente um sistema se utiliza de relógios, tanto físicos quanto lógicos. O problema da dessincronização inerente aos relógios pode prejudicar tanto a comunicação, quanto a computação que indiretamente pode afetar o sistema de controle.

De acordo com Törgren (1998) para eliminar a variação de tempo é possível utilizar-se do método de sincronização ou através do uso do buffer em um circuito. Em um sistema que opera com algoritmos de sincronização, os relógios são sincronizados dentro de uma determinada precisão.

Segundo Moreira (2006), computação em tempo real não é equivalente à computação rápida. O objetivo da computação rápida é minimizar o tempo de resposta médio de um dado conjunto de tarefas. O objetivo de computação de tempo real é atender aos requisitos individuais de tempo de cada tarefa. A propriedade mais importante de sistemas de tempo real deve ser a previsibilidade: sua funcionalidade e seu comportamento temporal devem ser tão previsíveis em tempo de projeto quanto forem necessários para satisfazer as especificações do sistema em tempo de execução. Computação rápida por si só não garante previsibilidade. Segundo Moreira (2006), sistemas de tempo real podem ser compostos por três estágios: aquisição de dados através de sensores, processamento de dados, no controlador e saída para atuadores. Uma arquitetura de um sistema de tempo real deve ser projetada para suportar esses componentes com a fidelidade adequada, de maneira a se obter um controle satisfatório se comparado a um controle contínuo dentro de um envelope de operação previamente estabelecido.

Em uma configuração de tempo não real é suficiente verificar se a lógica de computação está correta; em configurações de tempo real também é necessário verificar se os tempos daquela estão corretos. Isto inclui assegurar a agendabilidade (*schedulability*), tanto de mensagens síncronas como de mensagens assíncronas (esporádicas ou aperiódicas), assim como assegurar que os requisitos temporais de mensagens assíncronas sejam atendidos.

Segundo Moreira (2006), o conceito de agendamento em sistemas de tempo real dita a habilidade em prover resposta imediata para eventos externos específicos, particularmente a necessidade de agendamento de tarefas para serem executadas dentro de um limite específico de tempo após a ocorrência de um evento.

O componente responsável por prover essa capacidade é chamado de agendador. O único propósito do agendador é alocar o recurso de tempo de processamento para as várias tarefas competindo por prioridade de execução.

## **2.2 Arquitetura Federada vs. Integrada**

De acordo com Obermaisser, Peti e Tagliabo (2007), pode-se distinguir duas classes extremas de paradigmas de arquiteturas para aplicações distribuídas, denominadas Arquiteturas Federadas (*AF*) e Arquiteturas Integradas (*A*). Sistemas de tempo real freqüentemente se posicionam entre estes dois extremos, inclinando-se mais para um lado do que para outro. Em um sistema totalmente federado, para cada subsistema/aplicação (e.g., sistema multimídia de um carro) existe um computador dedicado ao subsistema. Já um sistema integrado é caracterizado pela integração de múltiplas aplicações/subsistemas em um único computador.

### 2.2.1 Vantagens de uma Arquitetura Federada

Embora a abordagem de Sistemas Federados tenha deficiências significativas em comparação a abordagem de Sistemas Integrados, a abordagem de sistemas federados é superior em relação ao controle da complexidade, desenvolvimento independente de subsistemas, proteção de propriedade intelectual e contenção de erros externos (Obermaisser; Peti; Tagliabo, 2007).

Segundo Obermaisser, Peti e Tagliabo (2007), as principais vantagens de uma Arquitetura Federada são:

- **Contenção de Falhas (*Fault Containment*)** – Sistemas Federados oferecem vantagem em relação a contenção de falhas. Quando um hardware é afetado por uma falha, os impactos da falha são sentidos somente em um único subsistema. Ao contrário, uma falha na arquitetura integrada pode ter impacto em vários de seus subsistemas.
- **Contenção de Erros (*Error Containment*)** – A contenção de erros evita que erros que ocorrem num subsistema sejam propagados a outros subsistemas. Com isso é possível manter os outros subsistemas funcionais, mesmo com o erro em um deles.
- **Desenvolvimento Independente (*Independent Development*)** – Sistemas Federados suportam um desenvolvimento independente de seus subsistemas. Em Sistemas Federados, um grau limitado de interações entre seus subsistemas ocorre via gateways.
- **Controle de Complexidade (*Complexity Control*)** - A implementação de cada subsistema possui um computador distribuído dedicado, cuja interação com outros subsistemas é controlada através de “gateways” e ajuda na gestão da complexidade. A ocultação do funcionamento interno de cada subsistema permite que um

projetista entenda o comportamento particular de qualquer subsistema sem analisar e compreender plenamente o resto do sistema.

### **2.2.2 Vantagens de uma Arquitetura Integrada**

De acordo com Obermaisser, Peti e Tagliabo (2007), as principais vantagens de uma Arquitetura Integrada são:

- Redução de Custo de Hardware (*Hardware Cost Reduction*) – Em contraste com sistemas federados, que requerem um sistema de computação distribuída para cada subsistema, sistemas integrados facilitam a multiplexação de recursos de hardware.
- Dependência de Recursos dado a redução de cabos e conectores (*Dependability Improvements*) – Um número reduzido de computadores no sistema, reduz também o número de problemas. A confiabilidade aumenta com a redução de número de conectores e cabos. Em ambientes automotivos 30% das falhas apresentadas são falhas elétricas atribuídas por conectores com problemas (Obermaisser; Peti; Tagliabo, 2007).
- Tolerância a Falhas (*Fault-Tolerance*) – Em ambas as arquiteturas, federadas e integradas, hardware adicional é necessário para tolerar falhas de hardware. Contudo, em sistemas federados estes recursos devem ser disponíveis em cada computador do sistema. Com isso são necessários inúmeros computadores dedicados a vários subsistemas para tolerar falhas. Em sistemas integrados, os recursos estão disponíveis universalmente ao longo das diferentes aplicações.

## **2.3 Protocolos de Comunicação**

O termo protocolo se refere a um conjunto de regras que regem as comunicações. Seguindo esse raciocínio, dois dispositivos, interligados por uma rede de comunicação, precisam entender o mesmo protocolo para poder se comunicar.

Existe uma ampla variedade de protocolos de comunicação empregados em sistemas de controle. Esperar que algum dia apareça um protocolo que sirva como padrão para uma ampla gama de aplicações é inútil. A diversidade de aplicações faz que um protocolo seja mais adequado para uma aplicação que para outra. Portanto é comum ver na bibliografia estudos de comparação entre protocolos como em Gwaltney e Briscoe (2006).

O estudo comparativo feito por Gwaltney e Briscoe (2006), chamou muito a atenção por três motivos: primeiro pela quantidade de protocolos que são estudados (11 no total); segundo, porque o projeto onde se vai utilizar o protocolo estudado pelo artigo pertence à área aeroespacial; terceiro, porque os autores do estudo são da NASA.

Neste trabalho foram escolhidos dois protocolos, o TTP (*Time Triggered Protocol*) e o Ethernet, para uma descrição mais detalhada. O motivo a escolher estes protocolos foi o seu potencial uso na área aeroespacial.

### **2.3.1 Time Triggered Protocol - TTP**

Em Lustosa (2009) é feita uma excelente revisão bibliográfica sobre arquitetura TTA, onde é dito que a Arquitetura Disparada por Tempo (*Time-Triggered Architecture - TTA*) engloba um modelo completo de projeto e funcionamento de sistemas distribuídos com características de tempo-real. A proposta mais atual, apresentada em Kopetz e Bauer (2003), é resultado de um extenso trabalho desenvolvido no contexto do projeto MARS (*Maintainable Real-time System*) que é um sistema distribuído tolerante a

falhas para controle de processos de tempo real, iniciado em 1979 na Escola Técnica de Berlin, sob a liderança do Prof. Dr. Hermann Kopetz. Em 1995 um trabalho de cooperação com a Daimler Chrysler resultou em um estudo de caso onde a TTA foi empregada no protótipo de um veículo equipado com sistema *brake by wire* embarcado. Contudo, sua utilização tem sido também de grande interesse em outras áreas. Ainda de acordo com Lustosa (2009), em 1999, a Alcatel deu início a um trabalho interno de pesquisa visando a implantação de sistemas de controle para trens. Além disso, em 2000, a arquitetura TTA foi adotada pela Honeywell para desenvolvimento de sistemas de propulsão, abrindo caminho para o uso de componentes COTS (*Commercial Of-The-Shelf*) em projetos de sistemas altamente críticos.

Desde sua concepção inicial até os dias de hoje, o modelo de comunicação da TTA tem sido adotado como princípio básico no projeto de diversos protocolos destinados ao projeto de Sistemas de Controle por Rede. Segundo Lustosa e Souza (2008c), o princípio fundamental do modelo de comunicação *time-triggered* estabelece que as ações mais relevantes do sistema sejam disparadas em instantes de tempo previamente definidos à medida que o tempo avança, conforme ilustra a Figura 2.4. Em se tratando de Sistemas de Controle por Rede, onde as unidades de processamento estão fisicamente separadas no espaço, fica clara a necessidade de uma base de tempo global sincronizada. Em razão disso, cada controlador de rede da TTA possui um mecanismo de sincronização de relógios (*clocks*) implantado com o uso de hardware dedicado. De acordo com a versão do protocolo utilizado, o mecanismo de sincronização pode apresentar as seguintes arquiteturas: 1) centralizada, como na versão TTP/A, versão mais simples e de baixo custo; 2) essencialmente distribuída, como na versão TTP/B; e 3) tolerante às falhas como na versão TTP/C, a qual será apresentada em maiores detalhes nesta seção.

O tempo de acesso ao barramento é particionado em intervalos (*slots*) dedicados segundo um mecanismo TDMA. Todos os *slots* têm o mesmo tamanho. O que precisa ser definido na fase de projeto é a alocação dos slots aos nós, em função do tempo de transmissão do maior quadro e da velocidade do controlador de rede na execução das funções primitivas da plataforma, de acordo com Lustosa e Souza (2008c).

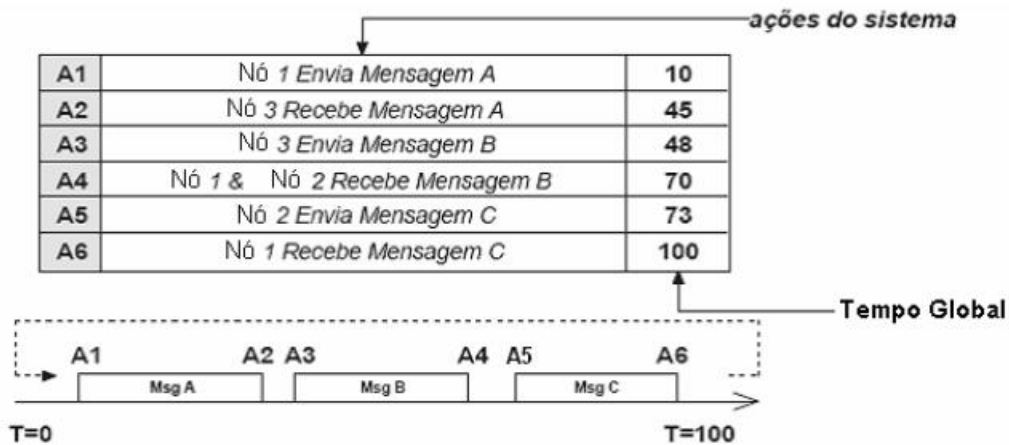


Figura 2.4 – Modelo de Comunicação *Time-Triggered*.

Fonte: Lustosa (2009).

A seqüência na qual cada nó possui um *slot* de tempo exclusivo para transmitir é chamada de rodada TDMA (*TDMA round*). Após o término de uma rodada, uma outra rodada com uma seqüência diferente de mensagens é iniciada. O conjunto de todas as rodadas previstas recebe o nome de ciclo TDMA (*TDMA cycle*), ou ciclo do aglomerado de computadores (*cluster cycle*), que consiste numa linha base de tempo finita dentro da qual as ações do sistema são agendadas. Quando o ciclo de sistema chega ao seu fim, a sincronização dos relógios ocorre.

As especificações para o TTP/C foram primeiramente publicadas em 1993. O “C” no TTP/C estabelece a classe C automotiva que está referida a requisitos de comunicação de difícil execução em tempo real. De fato, a indústria automobilística considerou o protocolo TTP/C bastante adequado para aplicações de drive-by-wire. A TTTECH, uma companhia com sede na

Áustria, tem comercializado o TTP/C e seus componentes de controlador da comunicação estão disponíveis à venda.

Estes dispositivos implementam o hardware do protocolo e são disponíveis abertamente a qualquer projetista de sistemas. O TTP/C tem sido empregado numa ampla variedade de transportes tripulados, incluindo o sistema de controle de pressão da cabine do Airbus A380, controladores digitais de motores com completa autoridade (FADECs), em sinalização de trens, etc. Tem sido também usado no sistema de drive-by wire de automóveis conceituais, além de estar presente em partes do ônibus espacial da NASA.

O TTP/C é projetado para prover um alto nível de confiabilidade e disponibilidade a um custo adequado para a produção em massa.

O TTP/C provê importantes serviços tais como autonomia de transporte de mensagens baseado no agendamento com conhecimento do retardo e delimitação da variabilidade (jitter) sobre os canais de comunicação redundantes duais. O TTP/C e, conseqüentemente, a TTA, suportam a implementação de nós redundantes ou execução de funções redundantes em múltiplos nós. Implementações atuais do chip controlador de comunicação inclui tolerância a falhas do relógio global para estabelecer a base de tempo, serviços de informação para informar a todos os nós sobre o estado de “saúde” dos outros nós, e fixação do status da mensagem por ambos, o transmissor e o receptor. O protocolo não tem um “mestre” e permite continuar a comunicação com os nós restantes quando um nó falha. Guardas de barramento são incluídos no hardware do controle de comunicação, mas é parte do mesmo dispositivo e compartilham o mesmo relógio. O TTP/C foi projetado para ser independente da camada física. Chips de controladores atuais suportam comunicações de 5 MB/s sobre RS-485 e 25 MB/s sobre Ethernet. As hipóteses de falha do TTP/C garantem que o sistema de comunicação pode tolerar qualquer falha simples em



qualquer componente da arquitetura e pode tolerar, também, múltiplas falhas dependendo da aplicação. O TTP/C utiliza o algoritmo Welch-Lynch para sincronização de relógios. Maiores informações sobre TTP/C podem se encontrar em TTTECH (2003).

### **2.3.2 Ethernet**

O protocolo de comunicação Ethernet é amplamente utilizado em arquiteturas de comunicação para rede de computadores comerciais, governamentais e instituições de educação. Está sendo utilizado, com adaptações, também em aplicações militares e aeroespaciais, e atualmente é usado na Estação Espacial Internacional. O padrão IEEE 802.3-2002 define a Ethernet e em sua versão atual define o padrão incluindo as especificações para o Gigabit Ethernet.

Como sugere a designação, a Ethernet prove transmissão de dados com taxa de 10 MB/s, 100 MB/s e agora 1 Gb/s sobre cabo de par trançado. A Ethernet opera em modo half-duplex (todos os nós compartilham o mesmo cabo) ou em modo full-duplex (os nós podem se comunicar sobre um cabo dedicado com outro nó). Em operação *half-duplex*, o CSMA/CD governa o compartilhamento dos canais. Em operação *full-duplex* o *Flow-Control* governa o compartilhamento de canais. O nó transmite dados somente quando o canal esta vazio. Se dois nós iniciam a transmissão ao mesmo tempo, a colisão é detectada e a transmissão é cancelada. Cada nó espera um tempo aleatório para a linha de transmissão ficar disponível, e inicia a transmissão novamente. Se houver colisão a operação é repetida. Claramente, isto pode resultar em uma ineficiente comunicação, especialmente quando o tráfego de dados é intenso, pois a Ethernet é uma rede de “melhor-esforço” (*Best-effort*). Isto é, os pacotes são tratados sem distinção alguma, sem prioridade, sem garantia de tempo de entrega e até mesmo sem garantia de entrega ao destino, independente da sua aplicação, porém, sendo feito o máximo de esforço para a entrega destes. Assim todos os pacotes são aceitos e não havendo congestionamento e colisões eles são

encaminhados ao destino, caso contrário podem ser descartados. O modo full-duplex é possível quando os nós são conectados a um switch que permite uma conexão dedicada entre a porta do switch e a do nó. O switch fica assim, responsável pelo roteamento das mensagens para o destinatário sem colisão.

O protocolo utilizado para enviar mensagens também afeta a confiabilidade da transmissão, o “overhead” no pacote da mensagem e o tempo necessário para completar a transmissão. Dois protocolos populares são: 1) *User Datagram Protocol* (UDP) e 2) *Transmission Control Protocol* (TCP). O UDP não é confiável, pois não garante que os dados cheguem ao seu destino. O TCP adiciona um significativo “overhead” para o processo de transmissão em comparação ao UDP, mas provê uma conexão segura que requer para o remetente e o receptor abrirem uma conexão antes de enviar dados, assegurando que as mensagens sejam recebidas corretamente, na seqüência de pacotes para transmissão, e provê controle de fluxo. O protocolo Ethernet permite ainda criar, até um certo nível, sistemas síncronos (sincronizados). Com isso é necessário o uso de algum mecanismo de sincronização, comumente é utilizado os protocolos IEEE 1588 ou o NTP.

Ao longo dos anos muito foi discutido de como adaptar a Ethernet para novas aplicações, tais como tarefas de tempo real em ambientes industriais. Mais de 20 versões diferentes do protocolo ethernet estão disputando o reconhecimento na indústria de automação. Existe também um esforço para adaptar a Ethernet para atender requisitos especiais em outras áreas diferentes da indústria de automação, como exemplo pode-se citar o AFDX na indústria aeroespacial. O AFDX é um protocolo baseado na Ethernet que foi criado para prover redes determinísticas para aviões.

Os sistemas de Ethernet de hoje têm limites quando se combinam redes de Ethernet clássicas com dispositivos e serviços. A escalabilidade destes

sistemas também é limitada e a solução de rede é desenvolvida para uma área de aplicação específica.

Em 2009 a empresa TTECH lançou o protocolo de comunicação TTEthernet que combina o determinismo, tolerância a falhas e propriedades de tempo real da tecnologia *time-triggered* com a flexibilidade, dinâmica e o legado de “melhor-esforço” do protocolo Ethernet, servindo de comunicação para diversos tipos de aplicações diferentes, (TTECH Tutorial, 2009).

## 2.4 Teoria dos Relógios

A noção de tempo é fundamental para a nossa existência e muito familiar em nossas vidas. Nós podemos refletir sobre eventos passados, presentes e possíveis eventos futuros, e assim argumentamos sobre eventos no domínio do tempo. Em muitos modelos de fenômeno natural, o tempo é a variável independente que determina a seqüência de estados do sistema (Kopetz, 1997).

O tempo possui muitas propriedades, como a monotonicidade que significa que o tempo sempre avança. Essa propriedade de simples entendimento parece fácil de ser mantida, mas de fato não é. Em um sistema distribuído o entendimento pode se tornar muito difícil e complicado, pois o tempo pode ser facilmente ajustado, intencionalmente ou não, para representar um tempo no passado ou no futuro, desrespeitando a propriedade monotônica. Além disso um sistema distribuído pode apresentar várias noções de tempo diferentes. Abaixo esta listado os conceitos utilizados no decorrer deste trabalho:

- **Relógio Físico (*Physical Clock*):** é um dispositivo para medir a progressão do tempo (KOPETZ, 1997);
- **Relógio Local (*Local Clock*):** relógio que fornece as estampas de tempo para eventos locais;

- **Relógio de Referência (*Reference Clock*):** relógio utilizado como referência a medidas, proveniente de padrões de tempo universais, tais com UTC e GPS;
- **Relógio Ideal (*Ideal Clock*):** Relógio universal sem a presença de nenhum erro ou falha;
- **Relógio Lógico (*Logical Clock*):** relógios virtuais que são gerenciados por software (LAMPOR, 1985);
- **Tempo Físico (*Physical Time*):** refere-se ao tempo no sistema físico, i.e. o sistema modelado pela simulação (REIS, 2008).;
- **Tempo de Relógio de Parede (*WallClock Time*):** é o tempo durante o qual o simulador é executado (REIS, 2008). Neste trabalho é também denominado **Tempo de Simulação**;
- **Tempo Simulado (*Simulated Time*):** é o tempo como representado em uma simulação (DMSO, 1998);
- **Tempo Lógico (*Logical Time*):** é a estampa de tempo de um relógio lógico, também conhecido como **Tempo Virtual (*Virtual Time*)**;
- **Tempo de Resposta (*Response Time*):** é o tempo entre o início e a finalização de uma tarefa (REIS, 2008);
- **Tempo de Processamento (*CPU Time*):** é o tempo que a CPU gasta realizando computações de uma determinada tarefa (REIS, 2008);
- **Tempo Decorrido (*Elapsed Time*):** a quantidade de Tempo no Relógio de Parede contada a partir do instante em que uma determinada ação é solicitada (REIS, 2008).

Com o advento da ciência moderna, mecanismos para medir a progressão do tempo utilizando relógios físicos têm sido inventados. Um relógio físico é um dispositivo para medir a progressão do tempo. Com a introdução de relógios em sistemas, houve a necessidade de criarem-se métodos para medição, datação e sincronização do tempo.

Em Allan (1986) é exposto o conceito dos relógios físicos e osciladores, caracterizando-os matematicamente para análise, medição e predição do tempo. Em Allan et. al. (1997) a perspectiva histórica da evolução da ciência do tempo é apresentada. Apresenta também, o padrão mundial de tempo UTC, o GPS, um pouco da visão da teoria da relatividade de Einstein em relação ao tempo e futuras técnicas para medição do tempo.

Diferentes aplicações podem ser sensíveis a problemas relativos à sincronização do tempo. Em uma aplicação distribuída de tempo real, o sistema de computação distribuída executa suas diferentes funções simultaneamente. Estas funções são, normalmente, executadas em diferentes nós. Para garantir um comportamento consistente e estável, do sistema distribuído, é necessário que todos os nós do processo obedeçam a mesma ordem temporal, isto é, deve ser obedecido o conceito de “Acontecer Antes (*Happening Before*)” proposto por Lamport (1978). Para que a ordem temporal e o conceito “Acontecer Antes” sejam obedecidos é necessário o uso de relógios. Lamport (1978) define que relógios lógicos são relógios virtuais que são gerenciados por software, descreve um método simples para sincronizar os relógios lógicos de cada nó dentro de um limite de precisão e define condições de operação dos relógios para que a ordem temporal seja obedecida. Com o avanço do conceito de relógio lógico, em meados da década de 1980, começaram a surgir algoritmos para a sincronização dos relógios lógicos. Os relógios lógicos apesar de não serem tão precisos quanto uma solução por hardware possuem a vantagem da flexibilidade e de menor custo (Ramanathan; 1990).

Em Lamport (1985) é proposto um algoritmo para solucionar o problema de sincronização de relógios lógicos em um sistema distribuído com tolerância a falhas bizantinas, que ficou conhecido como Algoritmo de Lamport.

Em Welch e Lynch (1984) é proposto outro algoritmo de sincronização de relógios lógicos em um sistema distribuído com tolerância a falhas bizantinas, que ficou conhecido como Algoritmo de Welch-Lynch ou FTM (*Fault Tolerant Mid-Point*). Os algoritmos de Lamport e Welch-Lynch apesar de serem muito parecidos, diferem-se na leitura dos relógios e na função de convergência, de acordo com Dutertre (1998). O Algoritmo de Welch-Lynch possui uma função de convergência baseada na mediana do vetor de valores de tempo. Kopetz e Ochsenreiter (1987) propõem uma modificação no algoritmo de Welch-Lynch, ao invés de somente trabalhar com a mediana, é proposto um algoritmo que tem a função de convergência por média dos valores de tempo. Ainda em Kopetz e Ochsenreiter (1987) é introduzido o fator de custo da tolerância a erros bizantinos em relação à precisão final alcançada pelo algoritmo.

Em Schneider (1986) é feita uma análise sobre o desempenho de algoritmos de sincronização tolerante a falhas bizantinas, caracterizando suas funções de convergência através dos parâmetros de precisão (*precision*) e exatidão (*accuracy*). Em Miner (1993) é apresentada uma verificação formal sobre o funcionamento e a convergência de algoritmos tolerantes a falhas bizantinas em sistemas.

Para manter os relógios dos nós do sistema em sincronia, existem algoritmos de sincronização que são baseados em lógicas matemáticas que, basicamente, definem as operações que o sistema deve realizar para manter seus relógios sincronizados ou com o tempo mais próximo possível dos limites pré-especificados. Existem também outras soluções além de uma solução por software. Existem soluções por hardware e soluções híbridas, que envolvem hardware e software.

Nas próximas seções, serão apresentados os principais conceitos para um bom entendimento sobre o funcionamento de relógios e sincronização de relógios em um sistema distribuído.

#### **2.4.1 Relógios Físicos**

Os relógios físicos são caracterizados pelo material utilizado para gerar a oscilação. Existem muitos tipos de relógios físicos, podemos citar entres os principais, os relógios atômicos e os relógios de quartzo. O relógio atômico é um medidor de tempo que funciona baseado em uma propriedade do átomo que quando estimulado por ondas eletromagnéticas faz com que a sua energia oscile de forma regular. Aproximadamente a cada 9.192.631.770 oscilações de energia do átomo de césio-133 o relógio entende que se passou um segundo. O relógio de quartzo baseia-se na vibração do quartzo quando estimulado por uma energia elétrica alternada. Maiores informações sobre relógios atômicos e de quartzo podemos encontrar em Mallette (2007). Um relógio físico é um dispositivo para medir o tempo. Kopetz (1997) define o relógio físico como um contador e um mecanismo físico de oscilação que, periodicamente, gera eventos que vão adicionar contagens ao contador, onde esse evento periódico é denominado microtick ( $\mu T$ ) do relógio e a duração entre dois microticks consecutivos são denominadas de resolução (*granularity*) do relógio. A resolução de um relógio é calculada pelo inverso da frequência de oscilação do oscilador do relógio. A resolução de qualquer relógio digital conduz a um erro de quantização na medida do tempo.

Mas existe ainda, uma medida de mais baixo nível que é denominada ciclo da máquina (CM). O ciclo da máquina que realmente é o inverso da frequência de oscilação do oscilador local.

De acordo com Ganssle (2005) uma instrução básica de linguagem de máquina tal como o NOP, que é a instrução em *assembly* que indica que o computador não deve fazer nada, demora em torno de 4 ciclos de máquina para executar a operação. Supondo um sistema que defini que a cada 4

ciclos de máquinas é gerado um microtick, então pode-se dizer que a operação NOP demora 4 CM e 1 microtick para ser realizada.

Para geração dos microticks é necessário o uso de um divisor de frequências que faz a contagem de microticks em uma frequência menor que a frequência do oscilador do relógio local. Portanto é comum o microtick não ter o mesmo período que o ciclo de máquina, ou seja, não possui a mesma frequência que o oscilador do relógio físico. A figura 2.5 exemplifica o conceito.

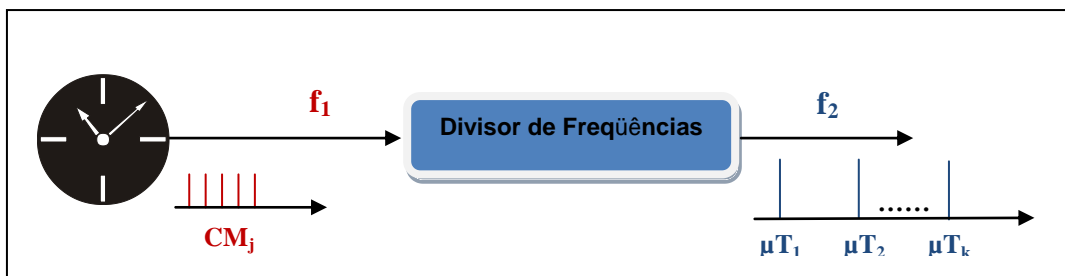


Figura 2.5 – Divisor de Frequências.

Trabalhar com os ciclos de máquina pode ser uma tarefa muito complicada, e o uso do método do divisor de frequências visa facilitar, já que ao desenvolver as aplicações pode-se trabalhar somente com os microticks, não se importando com o ciclo de máquina. Com isso é possível utilizar métodos de sincronização em computadores com relógios de diferentes frequências de oscilação local. Não existem muitas referências sobre o ciclo de máquina nas bibliografias de controle, um dos autores que abordam o ciclo de máquina é Ganssle (2005).

Kopetz et. al. (1997) mostra em seu trabalho, as estratégias de sincronização de relógios no protocolo de comunicação TTP/C. A Figura 2.6, mostra no protocolo de comunicação TTP/C a presença do divisor de frequências.



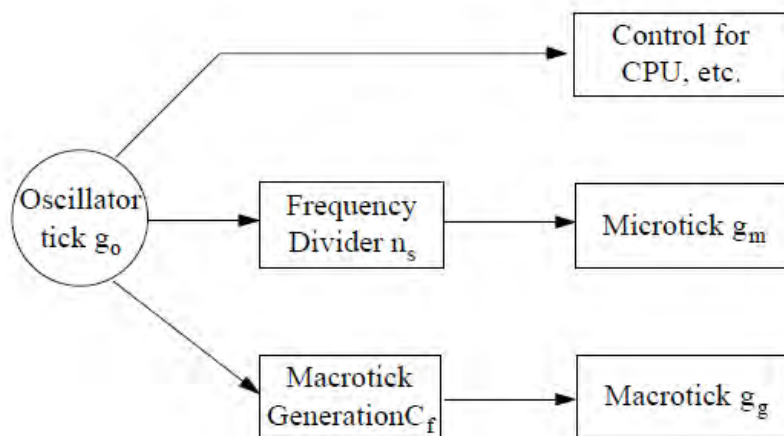


Figura 2.6 – Divisor de Frequências no TTP/C.

Fonte: Kopetz et. al. (1997)

Segundo Kopetz et. al. (1997) é importante separar os conceitos dos vários níveis de tempo dentro do gerador de tempo do nó local durante o projeto do sistema. Na figura 2.6 é possível ver a presença de um oscilador local, que gera os ticks do relógio. O controle da CPU é feito através destes ticks, isto é, o ciclo de máquina possui a mesma freqüência do oscilador local. Na Figura 2.6 ainda, é possível ver ligado ao oscilador local um divisor de freqüências que gera os microticks. Ligado ainda ao oscilador local é possível ver um Gerador de Macroticks. De acordo com Kopetz et. al. (1997) o gerador de macroticks é controlado através de um divisor de freqüências também, mas tudo isso já a nível de software. Não encontramos na literatura os métodos e modos que é implementado o divisor de freqüências para geração do microtick e do macrotick. Ainda segundo Kopetz et. al. (1997) o oscilador local de um nó deve servir aos seguintes propósitos:

- **Oscilador Local:** Gerar sinais de tempo para todas as unidades computacionais do nó;
- **MicroTick:** Prover o tempo de referência para as medidas de tempo e gerar a codificação de bit e a amostragem do bit do fluxo de entrada do canal de comunicação ;

- **Macrotick:** Gerar os sinais de tempo para o Tempo Global e o Tempo externo de referência;

Com isto neste presente trabalho, define-se o relógio físico como um contador e um mecanismo físico de oscilação que, periodicamente, gera eventos que vão adicionar contagens ao contador, onde esse evento periódico é definido como ciclo de máquina (CM) do relógio e o microtick ( $\mu T$ ) é definido como um evento periódico gerado pelo divisor de freqüências, com freqüência diferente do relógio físico local.

A duração entre dois microticks consecutivos é denominada de resolução (*granularity*) do microtick e a duração entre dois ciclos de máquina consecutivos é denominada a resolução do relógio físico. A resolução de um ciclo de máquina é calculada pelo inverso da freqüência de oscilação do oscilador do relógio físico e a resolução de um microtick é determinada como o inverso da freqüência do divisor de freqüências. É importante lembrar, que a resolução de qualquer relógio digital conduz a um erro de quantização na medida do tempo.

A figura 2.7 apresenta a resolução do relógio  $i$ , os ciclos de máquina e os instantes dos microticks dentro de uma linha de tempo. O valor de subscrito indica o instante do relógio ( $\mu T_k$ ), e o valor sobrescrito indica qual relógio estamos referenciando. No caso temos  $g^{\mu T}$  que significa a resolução do microtick do relógio e  $g^{CM}$  significa a resolução do ciclo de máquina.

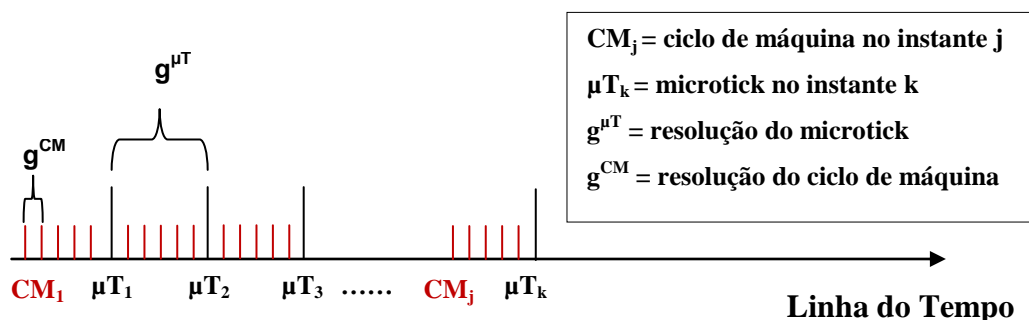


Figura 2.7 – Resolução de um Relógio Físico.

Neste presente trabalho não será abordado o ciclo de máquina de um computador, e, portanto, para efeitos práticos somente será considerado o microtick em nosso sistema.

Para se medir o relógio local é necessário o uso de um relógio comum a todos, normalmente chamado de relógio de referência. Assume-se, normalmente, que o relógio de referência coincide com algum padrão de tempo internacional. Para se medir o tempo de outros relógios, o relógio de referência precisa ter uma resolução menor que o relógio local. Conforme equação 2.8.

$$g^r < g^l \quad (2.8)$$

O relógio ideal é aquele onde a deriva (*drift*) do relógio local em relação ao relógio de referência é igual a um, ou seja, não apresenta imperfeições e os valores de tempo dos microticks coincidem, como mostra a Figura 2.8.

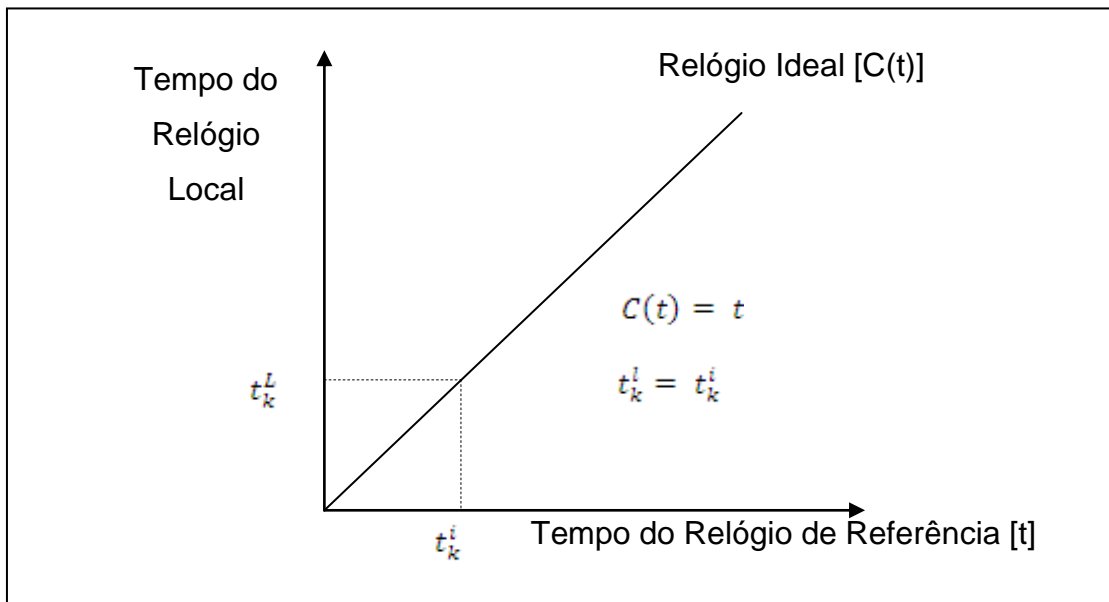


Figura 2.8 – Relógio Físico Ideal.

Os relógios físicos possuem imperfeições. Essas imperfeições podem ser causadas por vários motivos, principalmente na variação das condições ambientais, tais como variação de temperatura, variação de tensão ou envelhecimento do cristal, no caso de um relógio de quartzo.

A figura 2.9 apresenta as principais imperfeições possíveis de operação de um relógio. Estas principais imperfeições do relógio são conhecidas como a deriva (*drift*), o viés inicial/instantâneo (*offset*), a flutuação (*jitter*) e erro de estado (*state error*):

- A deriva é quando um relógio local tem uma frequência de oscilação maior ou menor que um relógio de referência, isto é, a deriva é a variação (ou diferença) entre dois relógios;
- Existem dois tipos de viés, o viés inicial que é quando existe uma diferença inicial entre o tempo dos relógios local e de referência e o viés instantâneo que é a diferença entre o tempo dos relógios local e de referência a cada instante;
- A flutuação é a incerteza na medida do relógio;
- O erro de estado é o erro quando um relógio pára, isto é, o relógio fixa em um valor.

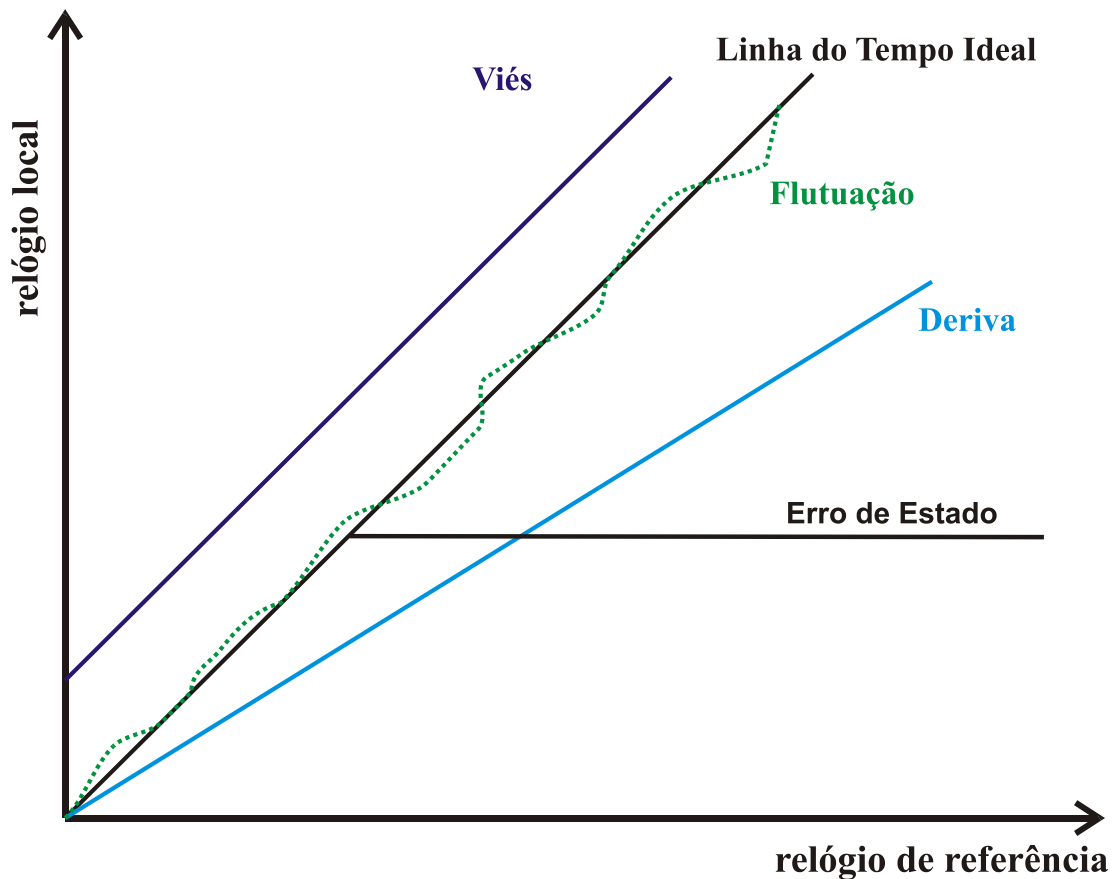


Figura 2.9 – Imperfeições do Relógio.

Quando um relógio local possui uma freqüência de oscilação mais rápida ou mais lenta do que outro relógio, é dito que o relógio local possui uma taxa de variação em relação ao outro relógio; conseqüentemente os microticks dos relógios se diferem. Para se fazer a medida do tempo de dois relógios diferentes, precisamos de um terceiro relógio, chamado de relógio de referência.

Existem muitos modelos para se representar um relógio físico. Varnum (1983) propôs um modelo estocástico simples de um relógio físico, onde será abordado melhor na seção 2.7.

Outro modelo muito utilizado é o modelo geométrico, representando as linhas de tempo de relógios através de retas. Neste modelo os efeitos da flutuação serão descartados. Este efeito será tratado no modelo não

determinístico que será abordado mais à frente. Somente os efeitos de deriva e viés são considerados, conforme a Figura 2.10.

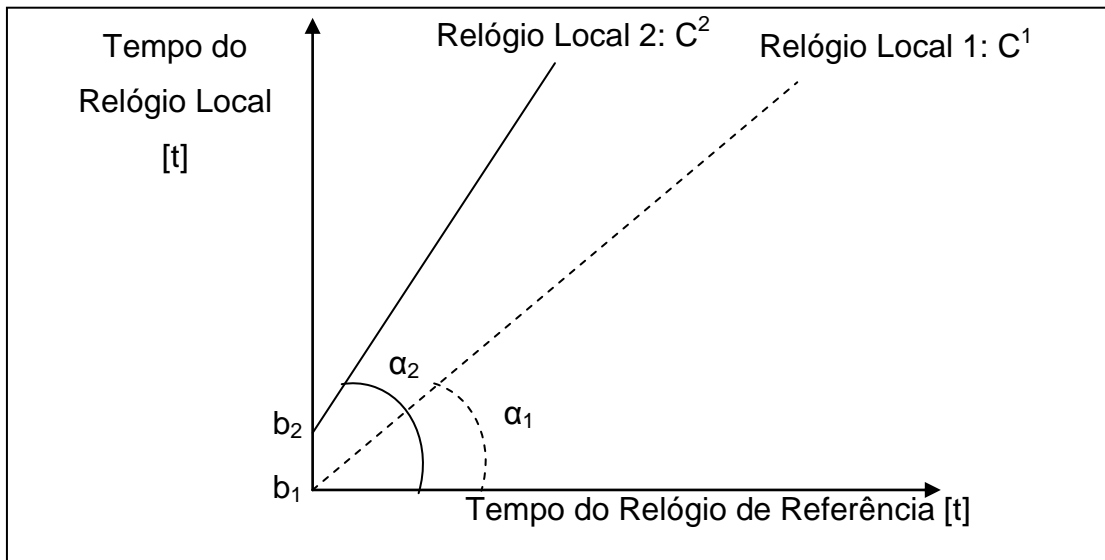


Figura 2.10 – Modelo Geométrico do Relógio Físico.

Na Figura 2.10, duas retas são representadas, uma representando a reta do relógio local 2,  $C^2$ , e a outra o relógio local 1,  $C^1$ . O relógio 2 possui um viés inicial e deriva em relação ao relógio 1. O eixo horizontal representa o tempo do relógio de referência utilizado para medir o tempo de ambos relógios. O eixo vertical representa o tempo dos relógios locais.

O modelo da Figura 2.10 é equacionado nas Equações (2.9), (2.10) e (2.11):

$$C^1(t) = a_1 t + b_1 \quad (2.9)$$

$$C^2(t) = a_2 t + b_2 \quad (2.10)$$

A equação 2.9 representa a equação da reta do relógio  $C^1$  e a equação 2.10 representa a equação de reta do relógio  $C^2$ . O coeficiente ' $b$ ' é o viés inicial e o coeficiente ' $a$ ' representa o coeficiente angular de reta. Ao fazer a diferença das equações 2.9 e 2.10, chega-se a equação 2.11.

$$\begin{aligned}\Delta C(t) &= C^2(t) - C^1(t) = \Delta at + \Delta b \\ \Delta a &= a_2 - a_1 \\ \Delta b &= b_2 - b_1\end{aligned}\tag{2.11}$$

Derivando 2.11 em relação ao tempo, tem-se:

$$\frac{d\Delta C(t)}{dt} = \frac{d(\Delta at + \Delta b)}{dt}\tag{2.12}$$

Na equação 2.12 o viés inicial,  $\Delta b$ , é uma constante, portanto sua derivada é zero. Existem, portanto dois casos possíveis: 1) O 1º caso para  $\Delta a$  constante; 2) o 2º caso para  $\Delta a$  não constante.

Para o **1º caso**, onde  $\Delta a$  é constante tem-se:

$$\dot{\Delta C}(t) = \Delta a\tag{2.13}$$

A equação 2.13 mostra a taxa de variação do coeficiente angular das retas 1 e 2, e podemos concluir que  $\Delta a$  é a deriva do relógio 2 em relação ao relógio 1.

O **2º caso**, um caso mais complexo, significa que além do relógio local 2 possuir uma variação do coeficiente angular em relação ao relógio local 1, o coeficiente angular esta se variando com o tempo e portanto a frequência de oscilação do relógio não é mais constante. Seria interessante adicionar ao modelo geométrico um terceiro termo quadrático para se trabalhar corretamente neste caso. Em Allan (1983) é caracterizado um modelo de relógio que considera este efeito. Neste modelo de Allan (1983), a variação de frequência é considerada em torno de  $0,5\Delta a$ . De acordo com Kopetz (1997) uma deriva de um relógio de quartzo está entre  $10^{-2}$  a  $10^{-7}$  [seg/seg]. Com isso é possível analisar que a variação de frequência, ou a variação da

pendente, é relativamente muito pequena e assim neste presente trabalho, será desconsiderado este caso.

Normalmente relógios de computadores são digitais, gerando eventos periódicos chamados de microticks. Discretizando as equações 2.11 e 2.13 têm-se o modelo de relógio discretizado, onde o subscrito  $k$  indica o instante de amostragem do sinal e o sobrescrito indica o relógio local, ou seja:

$$\Delta C_k(t) = C_k^2(t) - C_k^1(t) = \Delta a_k t_k^R + \Delta b_k \quad (2.14)$$

$$\dot{\Delta C}_k(t) = \Delta a_k \quad (2.15)$$

Com o modelo geométrico discreto do relógio, será definido o modelo de medidas.

A figura 2.11 apresenta três relógios digitais, o relógio de referência, o relógio 1, em vermelho, e o relógio 2, em azul. Com esta figura 2.11 é possível entender como funciona as medidas de um relógio digital. O relógio de referência normalmente é um relógio considerado bom e, portanto, considera-se que ele não possui imperfeições em sua operação. A resolução do relógio de referência contém sua resolução especificada e com isso é possível medir os relógios 1 e 2 em relação ao relógio de referência. Neste presente trabalho, os efeitos de atraso de medidas são ignorados, considerando-as operações instantâneas.



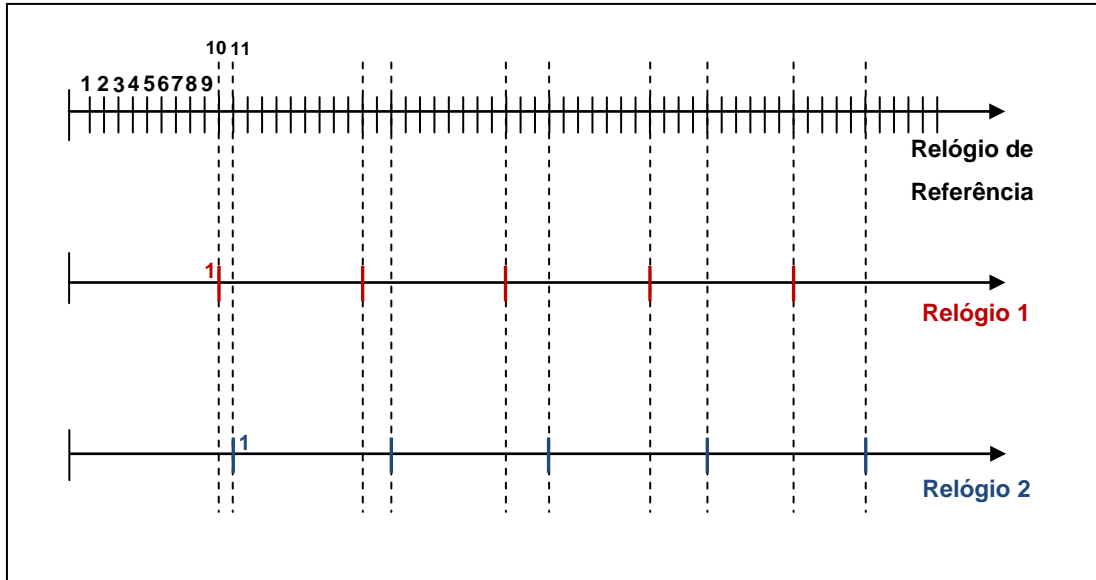


Figura 2.11 – Microticks.

Na figura 2.11, o relógio de referência possui uma resolução conhecida. Com isso somente a contagem de microticks dos relógios 1 e 2 em relação aos microticks do relógio de referência é necessária. A equação de relação do tempo com o microtick do relógio de referência é:

$$t_k^R = \mu T_k^R = k g^r \quad (2.16)$$

Com o tempo de referência definido, vamos definir os tempos do relógio 1 e relógio 2. Quando o relógio 1 gera um microtick o instante do relógio de referência é 10. Quando o relógio 2 gera o seu microtick o instante do relógio de referência é 11. E, portanto a equação que relaciona o tempo com o microtick dos relógios locais são:

$$t_k^1 = \mu T_k^1 = k g^1 = k(\mu T_{k+1}^1 - \mu T_k^1) \quad (2.17)$$

$$t_k^2 = \mu T_k^2 = k g^2 = k(\mu T_{k+1}^2 - \mu T_k^2) \quad (2.18)$$

Onde  $g^1$  e  $g^2$  são as resoluções do relógio 1 e 2, respectivamente.

Com isso é possível definir o modelo de medidas. Para o viés instantâneo defini-se a diferença entre os microticks, como em 2.19.

$$\Delta b_k = b_k^2 - b_k^1 = k(\mu T_k^2 - \mu T_k^1) \quad (2.19)$$

Em 2.20, 2.21 e 2.22, define-se a deriva entre os relógios. Em 2.20 tem-se a deriva do relógio 1 em relação ao relógio de referência.

$$a_k^1 = \frac{k(\mu T_{k+1}^1 - \mu T_k^1)}{kg^r} \quad (2.20)$$

Em 2.21 tem-se a deriva do relógio 2 em relação ao relógio de referência.

$$a_k^2 = \frac{k(\mu T_{k+1}^2 - \mu T_k^2)}{kg^r} \quad (2.21)$$

Em 2.22 tem-se a deriva do relógio 2 em relação ao relógio 1.

$$\Delta a_k^{2,1} = \frac{\mu T_{k+1}^2 - \mu T_k^2}{\mu T_{k+1}^1 - \mu T_k^1} \quad (2.22)$$

Quando um relógio não apresenta variação de deriva em relação a outro, a sua deriva é igual a um. Este fenômeno caracteriza o relógio ideal, descrito anteriormente e visto na Figura 2.9.

Um bom relógio apresenta uma deriva muito próxima de 1, e assim o conceito de taxa de deriva é importante, representada pelo símbolo  $\rho$ . A taxa de deriva é uma normalização da deriva. A taxa de deriva é descrita na equação 2.23, onde o subscrito representa o instante da deriva e o sobrescrito representa o relógio local.

$$\rho_k^L = \left| \frac{\mu T_{k+1}^1 - \mu T_k^1}{g^r} - 1 \right| \quad (2.23)$$

Na figura 2.12 está definido o intervalo da taxa de deriva que um relógio pode apresentar.

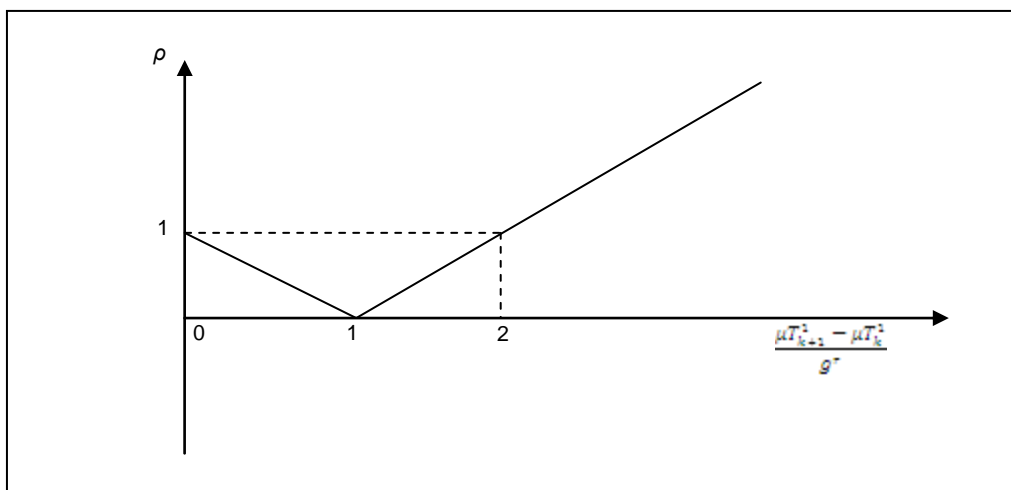


Figura 2.12 – Intervalo da Taxa de Deriva.

Com a normalização, um relógio ideal possui taxa de deriva igual a zero. De acordo com Kopetz (1997), relógios apresentam taxa de deriva na faixa de  $10^{-2}$  a  $10^{-7}$  [seg/seg], ou melhor, dependendo da qualidade do oscilador local.

A deriva é um problema intrínseco do relógio. Corrigir a deriva ou minimizar seu efeito, só é possível através do uso de técnicas, tais como algoritmos de sincronização.

### 2.4.2 Precisão e Exatidão

A sincronização de relógios visa reduzir ou anular o efeito da deriva que o relógio proporciona. Para isso, o conceito de precisão e exatidão é muito importante para o entendimento do processo de sincronização.

O viés instantâneo entre dois relógios é a diferença de microtick entre dois relógios de um conjunto de relógios. A equação 2.24 define o viés instantâneo existente entre dois relógios com mesma resolução, onde o subscrito indica o instante e o sobrescrito indica o relógio.

$$off_k^{2,1} = \mu T_k^2 - \mu T_k^1 \quad (2.24)$$

De acordo com Kopetz (1997) a precisão é o máximo viés instantâneo entre dois relógios dado um conjunto de relógios, durante um período de interesse. Na equação 2.25 é possível ver o equacionamento da precisão.

$$\Gamma_k = \max(\text{off}f_k^{j,l}) \quad (2.25)$$

Na figura 2.13 é possível indicar o viés instantâneo, no modelo geométrico de retas, de um conjunto de dois relógios.  $R_{int}$  é o parâmetro que define o período de re-sincronização e  $\Gamma$  é o parâmetro que define o máximo viés instantâneo que o sistema pode apresentar.

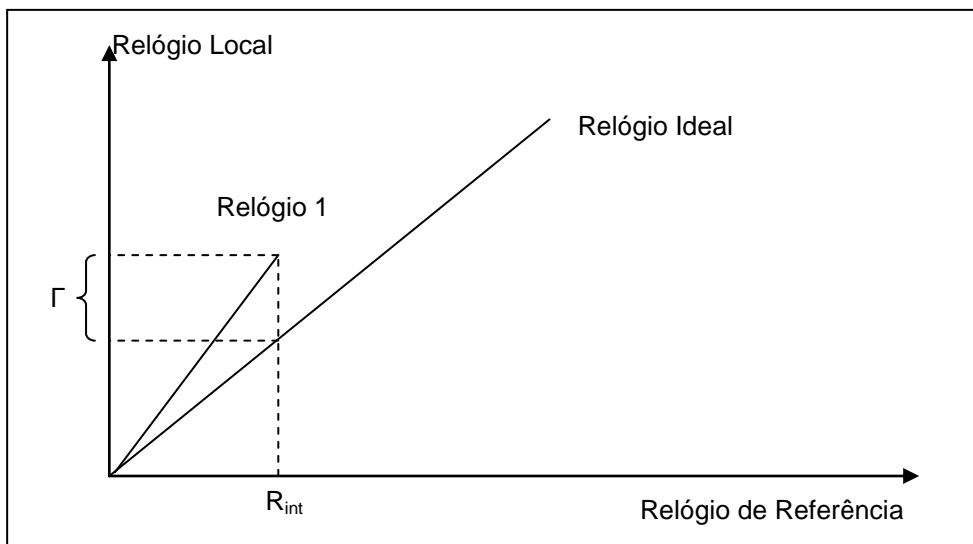


Figura 2.13 – Máximo viés instantâneo entre dois relógios.

A deriva do relógio é uma imperfeição intrínseca e indica que é preciso corrigir os relógios periodicamente. Assim, é necessário definir um período de re-sincronização. Este conceito será melhor abordado na seção sobre Sincronização Interna.

A figura 2.14 mostra graficamente o conceito de precisão entre um conjunto de dois relógios.

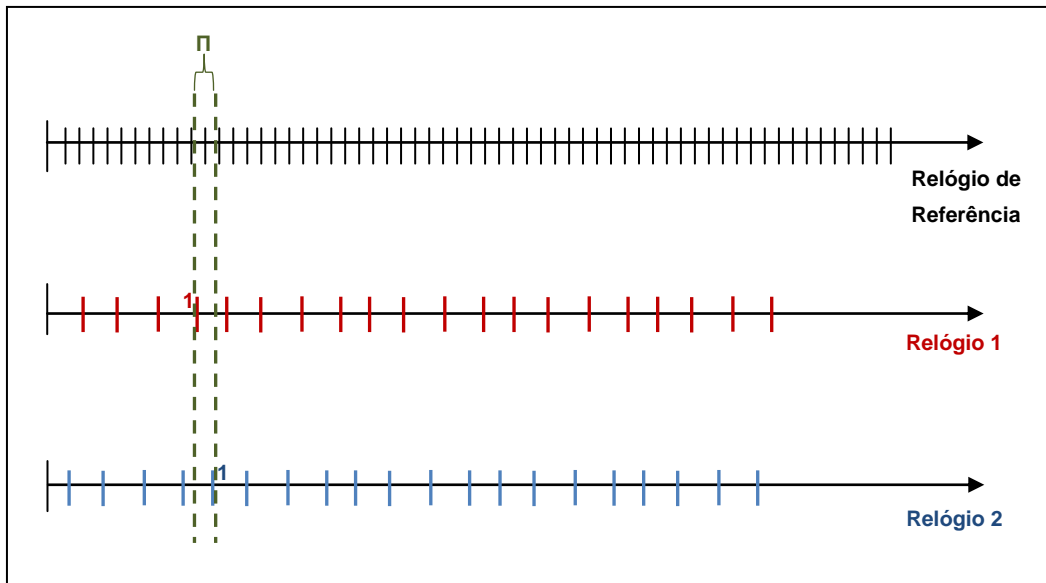


Figura 2.14 – Precisão em um conjunto de dois relógios.

Na figura 2.14, a precisão é a diferença dos microticks do relógio 1, em vermelho, e do relógio 2, em azul.

Apesar de ser um conceito fácil, definir a precisão, principalmente em tempo de projeto, pode-se tornar uma tarefa muito complicada. Dado um conjunto de relógios, a equação 2.26, através de uma combinação simples, define quantas análises são necessárias para se definir a precisão.

Dado um conjunto de relógios,  $R_n = [R_1, R_2, R_3, R_4, \dots, R_n]$ , tem-se, onde  $n$  é o número de relógios do conjunto e  $p$  indica elementos tomados  $p$  a  $p$ :

$$C(n, p) = \frac{n!}{(n - p)! p!} \quad (2.26)$$

A equação 2.26 indica que quanto maior o número de relógios do conjunto, mais complicada será a tarefa de definir a precisão.

De acordo com Kopetz (1997) a exatidão é o máximo viés instantâneo entre o relógio local e o relógio de referência externo, durante um período de

interesse. Este conceito é muito utilizado quando o sistema possui um relógio externo como referência, tal como o GPS ou um padrão de tempo internacional.

### **2.4.3 Padrões de Tempo e GPS**

É possível medir a diferença de tempo entre dois eventos, e com isso é possível ter a noção da progressão do tempo. Padrões de tempo servem para uniformizar os métodos utilizados para medir o tempo e fazer com que as medidas de tempo tenham a maior exatidão possível. Padrões de tempo são muito utilizados, principalmente, quando o sistema necessita de sincronização externa, que será abordada mais a frente.

Muitos padrões de tempo já foram inventados. Os mais conhecidos são o UTC (Tempo Universal coordenado) e o TAI (Tempo Atômico Internacional). O UTC é um padrão de tempo mantido pela *Bureau International de L'Heure*. O padrão de tempo é derivado das observações astronômicas da rotação terrestre em relação ao Sol. Em 1972, o UTC substituiu o GMT (Tempo Médio de Greenwich). Por causa da rotação da Terra ser irregular, o padrão GMT se modificava com o tempo. Com isso, definiu-se que o padrão UTC deve seguir, com maior exatidão possível, conforme o padrão de tempo TAI. O UTC não é um padrão de tempo contínuo, é necessária em alguns momentos a inserção de um segundo extra para se manter a sincronia do UTC com o fenômeno astronômico.

O TAI é um padrão de tempo que define a duração do segundo. A cada 9.192.631.770 oscilações de energia do átomo de césio-133 o relógio entende que se passou um segundo. Foi calculada pelo BIPM, Escritório Internacional de Pesos e Medidas, na França. A intenção do TAI é definir a duração do segundo em conformidade com os efeitos astronômicos da rotação terrestre.

O GPS (Sistema de Posicionamento Global) tem se tornado o principal meio para fornecer medidas de tempo no mundo. O GPS é formado por um conjunto de 24 satélites em órbita da Terra, cada um com um relógio atômico sincronizado. Com isso é possível em qualquer ponto da Terra, em qualquer instante de tempo, ter medidas de posição e tempo com uma boa exatidão.



Figura 2.15 – Constelação GPS.

Um excelente artigo que trata mais detalhadamente sobre esses assuntos está em Allan et. al. (1997).

#### **2.4.4 Relógio Lógico**

O relógio lógico, conhecido também como relógio virtual, é um relógio que opera sobre o relógio físico. Um relógio lógico é normalmente gerenciado e gerado em nível de software. A principal importância deste relógio lógico está na ordenação temporal de eventos, tais como o agendamento de computadores, agendamento de comunicação entre outros fenômenos que necessitam de gerenciamento lógico do tempo.

Um relógio lógico, similarmente ao relógio físico, gera periodicamente eventos. Esses eventos são denominados de macrotick do relógio. O

macrotick é um conjunto de microticks e assim, possui todas as propriedades de um microtick.

A cada  $i$  microticks de um relógio físico define-se um macrotick. Onde  $i$  é o número de vezes que o microtick ocorreu antes de um macrotick. Tem-se a equação 2.27, onde  $k$  indica o instante,  $i$  o número de microticks que ocorrem antes de um macrotick e  $j$  indica o relógio físico.

$$mT_k^j = i(\mu T_k^j) \quad (2.27)$$

A figura 2.16 ilustra melhor o conceito de macrotick. Onde  $\mu T_k$  é o microtick no instante  $k$  e o  $mT_k$  é o macrotick no instante  $k$ .

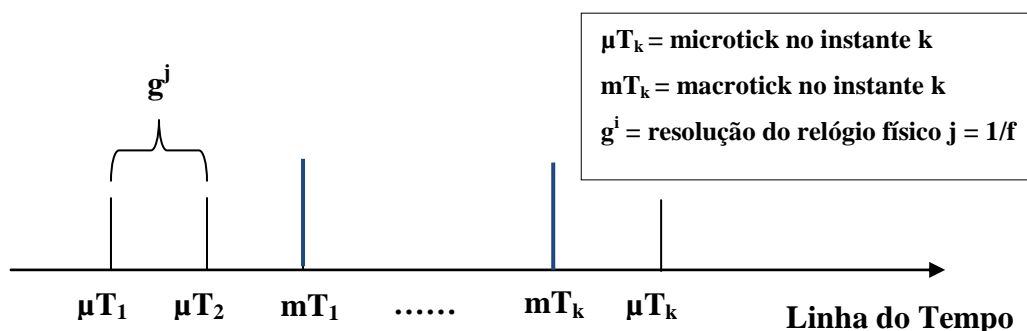


Figura 2.16 – Macrotick.

O macrotick é o evento gerado para implementar o tempo global. O conceito de macrotick é importante para definir o conceito de tempo global. Tempo global é muito importante em aplicações dos algoritmos de sincronização de relógios.

### 2.4.5 Tempo Global

Em um sistema distribuído é necessário estabelecer uma base de tempo comum entre todos os nós do sistema para se estabelecer a sincronização de relógios. Essa base de tempo comum entre todos os nós é denominada de tempo global.



Tempo global é uma base de tempo virtual, que mantém os *ticks* (*macroticks*) dos relógios lógicos do sistema dentro de uma determinada precisão  $\Pi$ . Algoritmos de sincronização de relógios utilizam o tempo global como base de tempo comum entre os nós para estabelecer a sincronização de relógios.

Supondo em um sistema distribuído, onde todos os macroticks de um conjunto de relógios estejam sincronizados em uma determinada precisão  $\Pi$ , o sistema é dito sincronizado se a condição imposta pela equação 2.28 é respeitada.

$$|\max(off_k^{j,l})| < \Pi \quad (2.28)$$

Onde,  $j$  e  $l$  representam os relógios lógicos e  $k$  o instante de análise. De acordo com (KOPETZ, 1997) se a condição imposta pela equação 2.28 é respeitada, então é possível construir a noção de tempo global em um sistema distribuído.

O tempo global é um relógio lógico, isto é, ele de fato não existe fisicamente, mas é possível construir sua noção através de relógios lógicos locais. A idéia do tempo global é fazer com que todos os nós do sistema distribuído gerem seus eventos de tempo, conhecidos como macroticks, em períodos iguais ou o mais próximo possível dentro de uma determinada precisão. A Figura 2.17 mostra a noção de tempo global, onde os macroticks estão interligados por uma linha pontilhada.

Com o uso do conceito de tempo global é possível estabelecer uma ordem temporal de eventos entre todos os nós do sistema distribuído. Protocolos de comunicação que usam o TDMA como acesso ao meio normalmente fazem uso do conceito de tempo global para ordenar as transmissões dos nós do sistema distribuído.

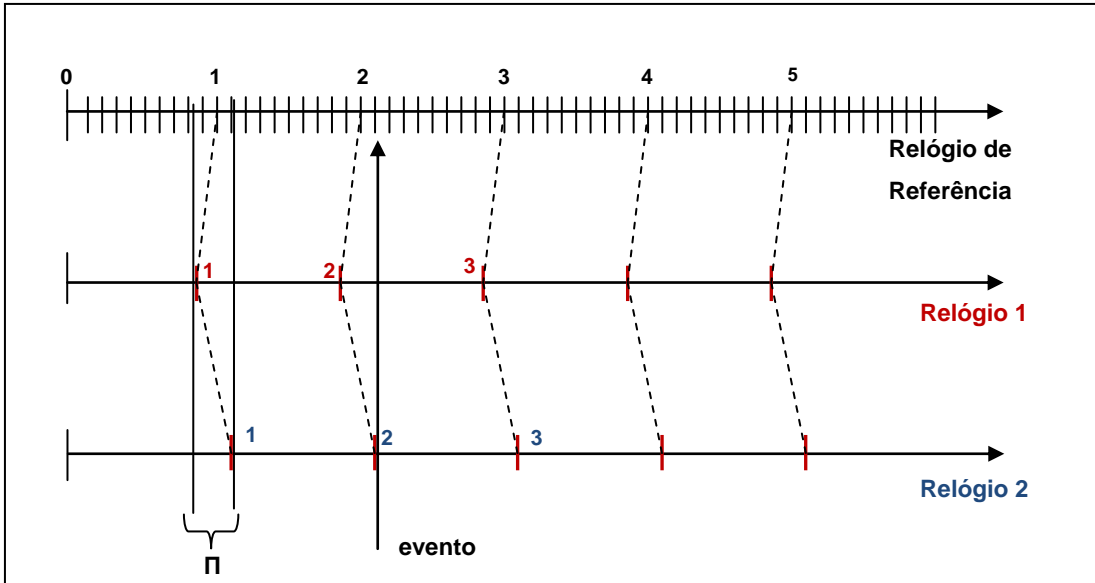


Figura 2.17 – Tempo Global.

Na figura 2.17, a estampa de tempo de microticks do evento nos relógios 1 e 2 são diferentes, quer dizer que o relógio 1 e relógio 2 identificam a ocorrência do evento em microticks diferentes. Mas com o conceito de tempo global, vemos que a identificação da ocorrência do evento pelo macrotick é igual, apesar dos microticks serem diferentes, isto é, ambos identificam o evento na estampa de tempo 2. O tempo global é um relógio lógico e possui sua própria granularidade ( $g^{tg}$ ), também conhecido como macro-granularidade.

De acordo com Kopetz (1997) para que o tempo global seja considerado razoável a seguinte condição deve ser obedecida:

a) Condição Razoável:

$$g^{tg} > \Pi \quad (2.29)$$

Esta condição razoável indica que o erro de sincronização é limitado a menos de uma macro-granularidade, i.e., o erro é limitado à duração entre dois macroticks. Sendo a condição da equação 2.29 sendo satisfeita, então é possível observar, na Figura 2.18, que em dois relógios diferentes do

conjunto, em um evento simples a estampa de tempo do evento pode diferir em no máximo um macrotick.

b) Condição para Evento Simples:

$$|t^1(e) - t^2(e)| \leq 1 \quad (2.30)$$

Onde  $t^1(e)$  é a estampa de tempo do evento (e) para o relógio 1 e  $t^2(e)$  é a estampa de tempo do mesmo evento (e) para o relógio 2.

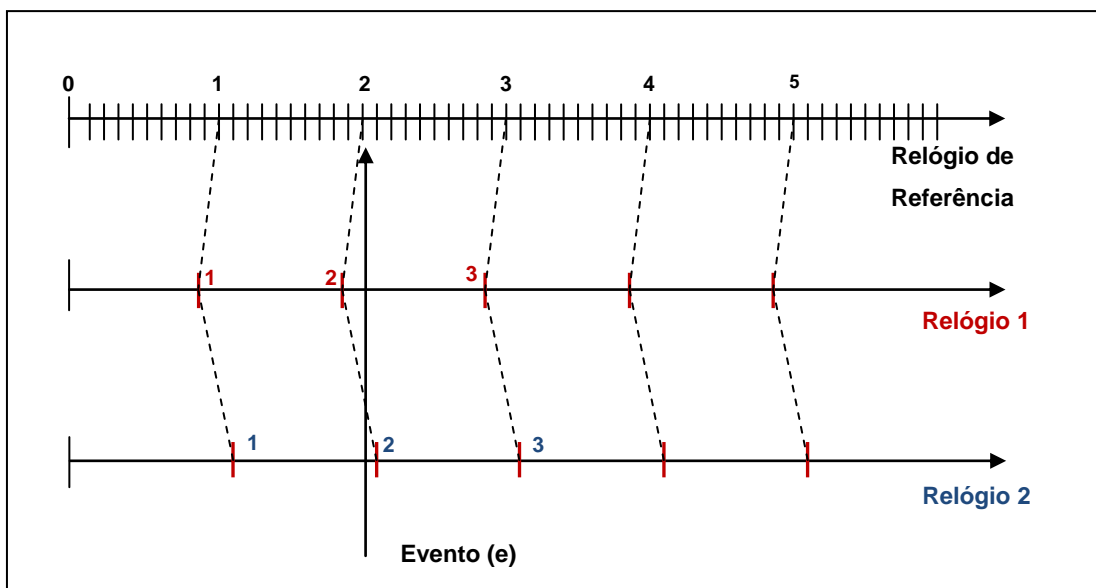


Figura 2.18 – Estampa de Tempo para Evento Simples.

Fonte adaptada de Kopetz (1997)

De acordo com Kopetz (1997), a diferença de um macrotick de um evento simples é o melhor que um sistema pode conseguir. Isto ocorre, por causa da impossibilidade de sincronizar relógios perfeitamente.

Um dos fatores principais do uso do tempo global é a reconstrução temporal da ordem dos eventos do sistema distribuído. Kopetz (1997) afirma que nem sempre será possível a reconstrução temporal da ordem dos eventos quando duas estampas de tempo de eventos diferentes diferem somente de um macrotick.

A Figura 2.19 mostra um conjunto de quatro eventos, eventos *A*, *B*, *C* e *D*, que ocorreram no sistema distribuído.

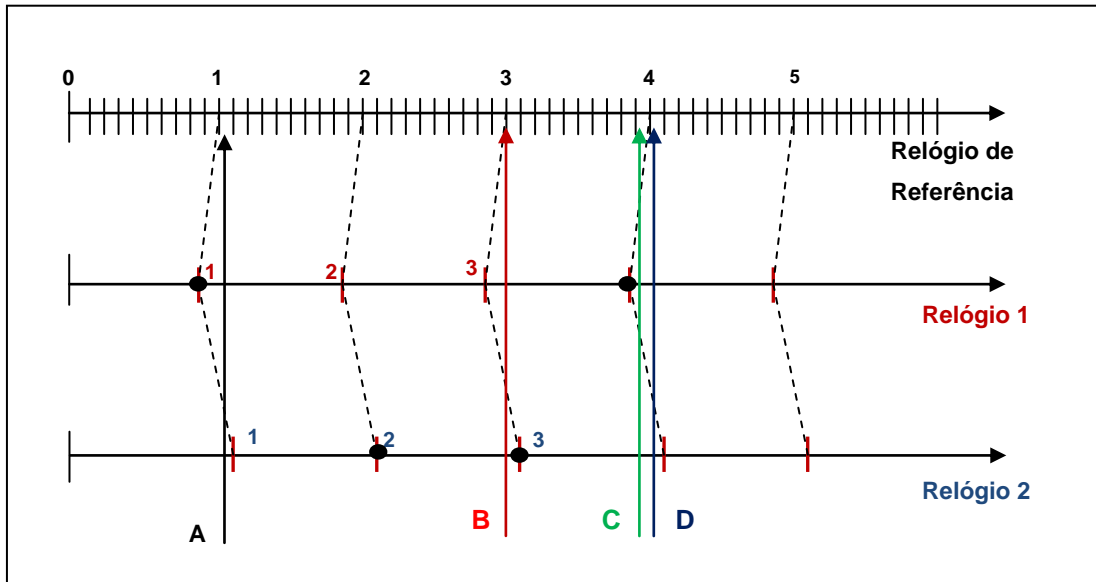


Figura 2.19 – Quatro Eventos.

Fonte adaptada de (Kopetz, 1997)

Depurando a figura 2.19 tem-se que:

- A estampa de tempo global do *evento A* no relógio 1:  $t^1(A) = 1$ ;
- A estampa de tempo global do *evento B* no relógio 2:  $t^2(B) = 2$ ;
- A estampa de tempo global do *evento C* no relógio 1:  $t^1(C) = 4$ ;
- A estampa de tempo global do *evento D* no relógio 1:  $t^2(d) = 3$ ;

Os eventos ocorrem na seqüência, *A*, *B*, *C* e *D*.

É possível analisar que a diferença de microticks entre os eventos *A* e *B* é de 20 microticks, mas ao analisar pelo tempo global a diferença é de 1 macrotick. Neste caso é possível a reconstrução temporal dos eventos, pois o *evento A* ocorreu no macrotick 1 e o *evento B* ocorreu no macrotick 2.

Analisando os eventos C e D, a diferença entre os eventos é de apenas 1 microtick. Analisando pelo tempo global a diferença entre os eventos é de 1 macrotick, igual ao caso anterior. Mas apesar de a diferença ser um macrotick, a reconstrução temporal dos eventos não é possível, já que pela estampa de tempo global o evento C ocorreu no macrotick 4 e o evento D ocorreu no macrotick 3, isto quer dizer que para o sistema distribuído o evento D ocorreu primeiro que o evento C, o que não é verdade. Isto ocorre por causa da acumulação de erros de sincronização e erros de digitalização.

Este caso leva a duas condições para reconstrução temporal:

c) Condição necessária:

$$|t^1(A) - t^2(B)| = 1 \quad (2.31)$$

d) Condição suficiente:

$$|t^1(A) - t^2(B)| > 1 \quad (2.32)$$

Isto quer dizer que para existir a reconstrução temporal de eventos com o uso do tempo global, a condição necessária, equação 2.31, é que a diferença da estampa de tempo global dos eventos seja igual a 1 macrotick, mas esta condição não é suficiente, portanto para ser suficiente é necessário que a diferença entre a estampa de tempo global de dois eventos seja maior que 1 macrotick, conforme equação 2.32.

Outro fator importante no conceito de tempo global é a precisão. A figura 2.17 mostra a precisão do tempo global, dada por  $\Pi$ . Em eventos que ocorrem durante esta precisão ou que possuem o mesmo valor de estampa de tempo global (macrotick), não são possíveis de serem ordenados temporalmente e o sistema considera que ambos os eventos ocorrem no mesmo instante.

#### 2.4.6 Sincronização Interna

Em sistemas distribuídos de controle em tempo real que utilizam o TDMA para comunicação, ou algum protocolo que necessite de sincronismo, é crucial se ter uma boa sincronização interna de relógios. A proposta da sincronização interna de relógios é estabelecer uma base de tempo global com tolerância a erros bizantinos.

Cada nó do sistema distribuído possui seu relógio local que gera seus microticks de acordo com a frequência do oscilador local. O macrotick de cada nó é formado pelo conjunto de microticks. Os macroticks são os ticks do relógio global.

Como os relógios físicos de cada nó possuem taxas de derivas diferentes é necessário o uso de algoritmos de sincronização que ajustem os relógios do nó periodicamente para que certa precisão seja obedecida. Esse período é denominado de período de re-sincronização.

Os macroticks de cada nó devem ser periodicamente re-sincronizados para estabelecer o tempo global dentro de uma determinada precisão  $\Pi$ . O período de re-sincronização é normalmente chamado de intervalo de re-sincronização. Na figura 2.20 é demonstrado o período de re-sincronização, e o local aonde é aplicada a função de convergência que atualiza os dados de relógio para manter os valores de tempo dentro de uma determinada precisão.

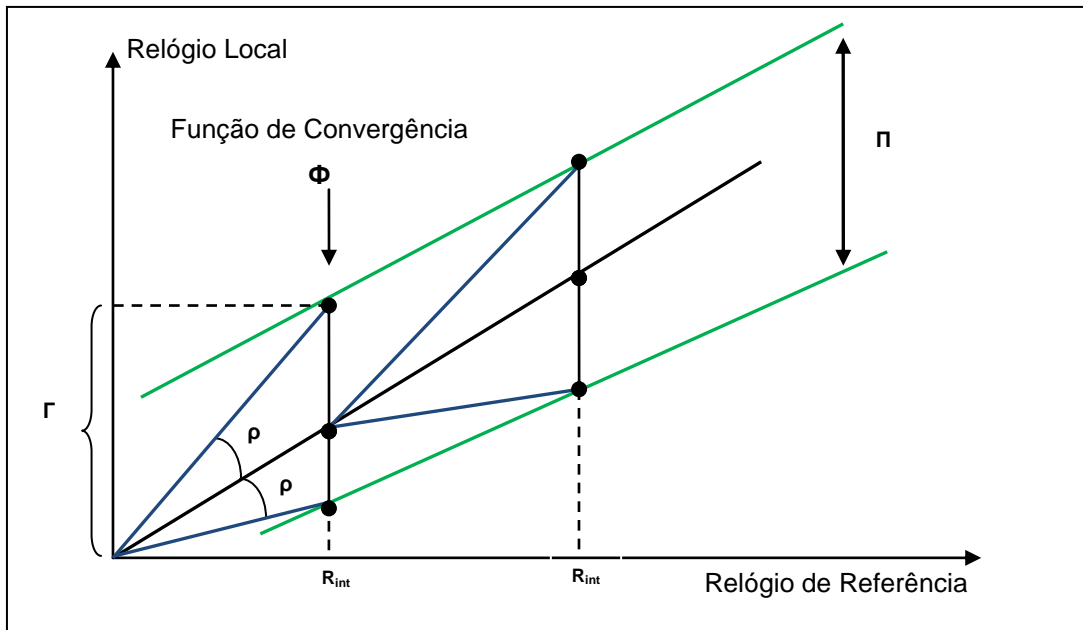


Figura 2.20 – Intervalo de Re-sincronização e Função de convergência.

Onde  $\Gamma$  indica máximo viés durante o intervalo de re-sincronização,  $\rho$  é a taxa de deriva e  $\Phi$  a função de convergência. O máximo viés depende do período de re-sincronização. A função de convergência  $\Phi$  denota o valor do viés de tempo imediatamente após a re-sincronização.

Da figura 2.18 tem-se o máximo viés, dado pela equação 2.33:

$$\Gamma = 2\rho R_{int} \quad (2.33)$$

Da equação 2.34, da Figura 2.20 e supondo um algoritmo tolerante a erros bizantinos, têm-se as seguintes condições de sincronização de relógios:

a) Condição de Sincronização:

$$\Phi + \Gamma \leq \Pi \quad (2.34)$$

b) Teorema de Tolerância a Erros Bizantinos, onde  $f$  é número de erros possíveis e  $n$  o número de relógios do conjunto:

$$n \geq 3f + 1 \quad (2.35)$$

Os algoritmos de sincronização devem garantir que o viés instantâneo dos relógios do conjunto não se divirja mais do que o intervalo de precisão, além de serem tolerantes a erros bizantinos. Em Lamport (1985) é feita a demonstração do teorema de que um algoritmo de sincronização tolerante a erros bizantinos deve obedecer à condição da equação 2.35.

O máximo viés é calculado pela equação 2.33 e a função de convergência é dependente do algoritmo de sincronização utilizado.

### 2.4.7 Sincronização Externa

A proposta da sincronização externa de relógios é de sincronizar o tempo global com algum padrão de tempo externo.

O tempo global, por causa da influência de alguns parâmetros como a taxa de deriva dos relógios, a latência da rede, entre outros pode derivar em relação a um padrão de tempo de referência. Como mostra a figura 2.21.

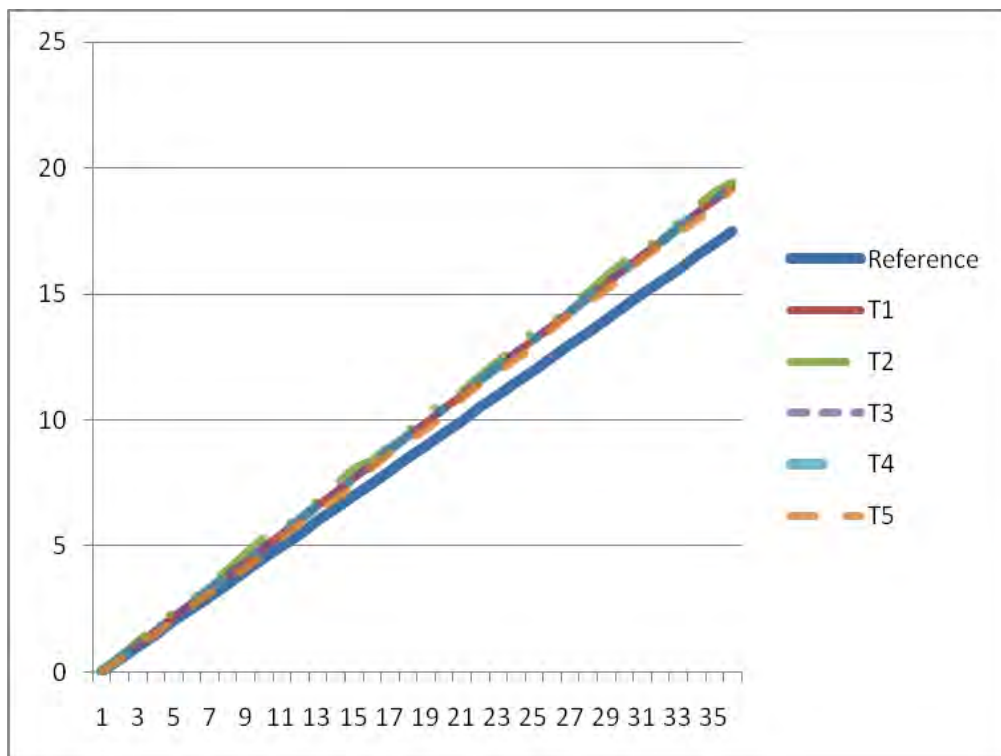




Figura 2.21 – Simulação do Algoritmo Welch-Lynch.

A Figura 2.21 é o resultado de uma simulação do algoritmo Welch-Lynch, que será abordado mais a frente. A simulação possui cinco relógios e um relógio de referência, considerado ideal. Através desta simulação, foi possível mostrar que os relógios do conjunto estão sincronizados entre si, mas podem derivar em relação a um relógio de referência ou no caso, de um relógio ideal.

Existem vários meios para se ter uma referência de tempo, tal como o protocolo NTP, o GPS, entre outros. O GPS, atualmente, é um dos meios mais utilizados, principalmente em sistemas de controle, para sincronização de relógios. O GPS é muito utilizado no mundo para navegação e sistemas que necessitam de boa precisão. Periodicamente, é realizado um broadcast na rede enviando os dados de tempo a todos os nós onde, no caso, o GPS é responsável pelo fornecimento de dados do tempo. Um algoritmo de sincronização de relógios então é utilizado para o cálculo da correção. A figura 2.22, ilustra a arquitetura de um sistema que utiliza o GPS como sua referência para sincronização externa.

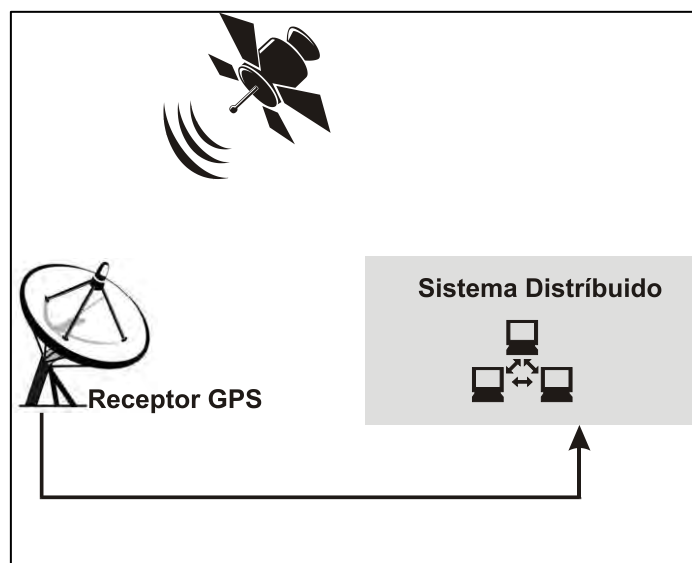


Figura 2.22 – Sincronização Externa com GPS.

### 2.4.8 Família de Soluções

Em Ramanathan (1990) é feita uma análise sobre alguns métodos de sincronização. Ainda, em Ramanathan (1990), o artigo divide as soluções em famílias, como software, hardware e híbrida.

Na Figura 2.23 é possível sintetizar as famílias de soluções possíveis para desenvolver uma aplicação síncrona.

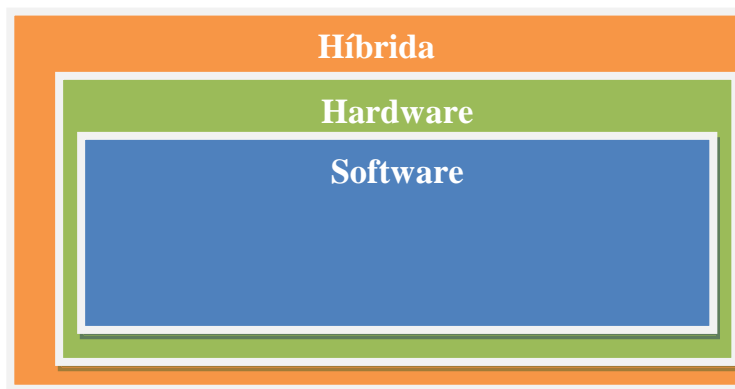


Figura 2.23 – Família de Soluções de Sincronização de Relógios.

Soluções híbridas são soluções conjuntas entre hardware e software. Normalmente utilizada quando é necessária uma precisão muito grande na sincronização. Maiores informações sobre sincronização por métodos híbridos podem ser encontrada em Ramanathan (1990).

Soluções por hardware são soluções em baixo nível, que aplicam a correção do relógio diretamente no circuito eletrônico, como por exemplo, o PLL (*Phased Locked Loop*), a figura 2.24 apresenta um diagrama em blocos de um sistema PLL.

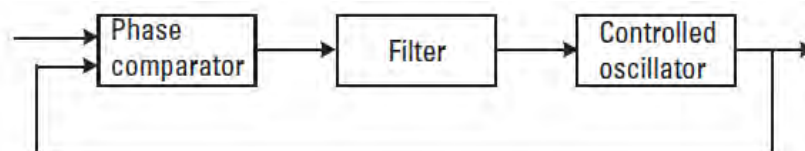


Figura 2.24 – Diagrama em blocos de um PLL.

Fonte: Kihara. (2001).

Maiores informações sobre sincronização por hardware pode ser encontrada em Kihara (2001) e Ramanathan (1990).

O foco desse trabalho é em sincronização de relógios por software. Sincronização por software utiliza-se de algoritmos de sincronização que serão abordados mais à frente na seção 2.6.

As três famílias de soluções possuem suas vantagens e desvantagens. Ramanathan (1990) descreve várias vantagens e desvantagens. A Tabela 2.1 apresenta as principais características qualitativas das famílias de soluções.

Tabela 2.1 – Tabela de Vantagens e Desvantagens de Famílias de Sincronização.

<b>Família</b>	<b>Custo</b>	<b>Precisão</b>	<b>Vantagens</b>	<b>Desvantagem</b>	<b>Exemplo</b>
<b>Híbrida</b>	Muito Alto	Muito Alta	Precisão Muito Alta	Baixa Flexibilidade;	Integração Hardware-Software
<b>Hardware</b>	Alto	Alta	Precisão Alta	Baixa Flexibilidade	PLL – <i>Phase Locked Loop</i>
<b>Software</b>	Médio	Média	Alta Flexibilidade	Depende de um relógio físico	Algoritmo Welch-Lynch / Algoritmo de Lamport

## 2.5 Arquiteturas de Sincronização

Existem muitos algoritmos de sincronização de relógios divididos em famílias, como visto anteriormente. Estes algoritmos de sincronização seguem padrões de arquitetura, tais como distribuídos, centralizado e com cálculo central. A maioria dos algoritmos de sincronização de relógios segue um dos padrões apresentados na seqüência.

### 2.5.1 Arquitetura Distribuída

A arquitetura distribuída de acordo com Kopetz (1997) é um algoritmo que tolera erros bizantinos, desde que a condição da equação 2.35 seja obedecida.

Kopetz (1997) mostra que a re-sincronização de uma arquitetura distribuída opera em três fases diferentes. A figura 2.25 mostra um diagrama em blocos da re-sincronização da arquitetura distribuída.

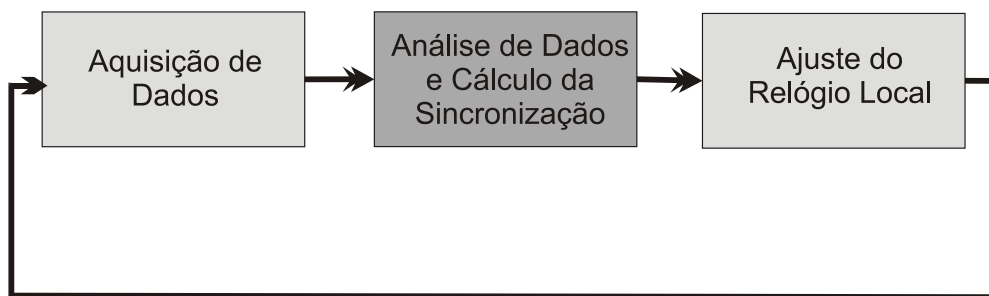


Figura 2.25 – Diagrama em blocos de fases de uma arquitetura distribuída.

No primeiro bloco da figura 2.25, o nó recebe os dados de tempo dos relógios presentes na rede de comunicação.

O segundo bloco é responsável pela análise, para evitar erros bizantinos, e pelo cálculo da função de convergência.

O terceiro bloco é responsável pelo ajuste do relógio lógico do nó local e com isso a seqüência se reinicia.

Todos os nós presentes na rede de comunicação utilizam-se destas três fases para manter sua sincronização de relógios.

A Figura 2.26 mostra uma arquitetura de sincronização distribuída, aonde é possível ver que não existe a presença de um relógio mestre.

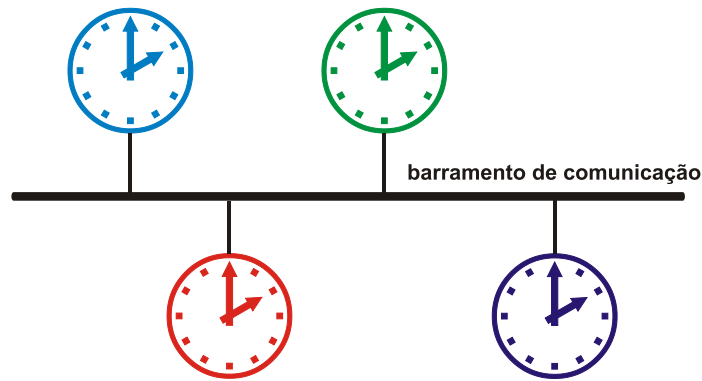


Figura 2.26 – Arquitetura Distribuída.

A figura 2.26 mostra quatro relógios interligados por um barramento de rede. Cada relógio é um nó da rede. A arquitetura distribuída não utiliza um relógio mestre como referência, esta arquitetura faz uso do tempo global para se sincronizar.

Protocolos que utilizam o TDMA como forma de comunicação, tais como TTP e FlexRay, aplicam essa arquitetura de sincronização em suas operações.

Mais a frente neste trabalho, será apresentado o algoritmo Welch-Lynch e uma simulação abordando este tipo de arquitetura.

### **2.5.2 Arquitetura Centralizada**

A arquitetura centralizada utiliza-se de um relógio mestre que periodicamente envia os dados de tempo aos relógios escravos. Os relógios escravos recebem o valor de tempo do relógio e comparam com o tempo de chegada mais o atraso de rede. Com isso é possível medir a diferença entre o relógio mestre e o relógio escravo e assim corrigir os valores do relógio escravo periodicamente.

A Figura 2.27 mostra a arquitetura de sincronização centralizada, aonde é possível ver a presença de um relógio mestre.

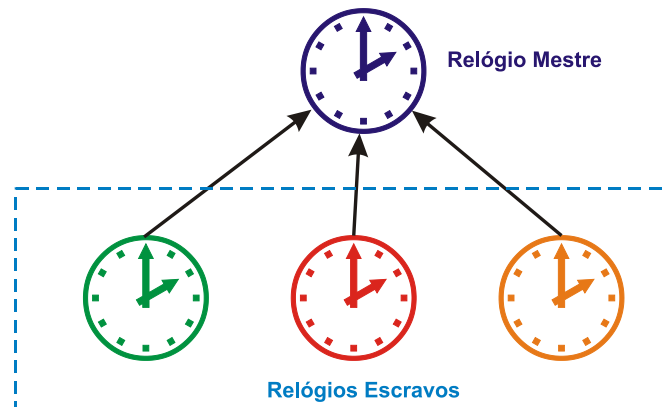


Figura 2.27 – Arquitetura Centralizada.

O relógio mestre normalmente é um relógio de excelente qualidade, tal como um relógio atômico, ou um padrão de tempo internacional. Esta arquitetura é muito utilizada em sistemas que se utilizam do GPS, como relógio mestre, para sincronização de relógios.

Esta arquitetura centralizada além de não ser tolerante a erros bizantinos tem a desvantagem de possuir um ponto comum de falha, isto é, se o relógio mestre falhar, todos os relógios escravos perdem a referência comprometendo a sincronização do sistema. Essa desvantagem é minimizada com o uso de relógios de excelente qualidade, com elevado preço no mercado, ou o uso de um GPS como referência, o que pode elevar o custo de um sistema.

Para eliminar o ponto comum de falha é possível utilizar a estratégia multi-mestre. Se o relógio mestre principal falhar, um dos relógios da rede, previamente selecionado, assume o papel de mestre e continua a re-sincronização do sistema.

Alguns algoritmos de sincronização, tal como IEEE 1588 e o NTP utilizam-se desta arquitetura para sincronização. Normalmente é aplicada à rede de comunicações que se utilizam do protocolo Ethernet. Este tipo de arquitetura é muito usado também em sincronização externa. Mais à frente neste

trabalho, será apresentado um modelo com filtro de Kalman e uma simulação abordando este tipo de arquitetura.

### 2.5.3 Arquitetura com Cálculo Central

A arquitetura com cálculo central é uma variante da arquitetura centralizada. Diferentemente da centralizada, nesta arquitetura é possível implementar um algoritmo tolerante a erros bizantinos.

A arquitetura com cálculo central possui, conforme visto na Figura 2.28, um ponto central aonde o cálculo da sincronização é realizado e depois é feito o envio a todos os nós a sua correção.

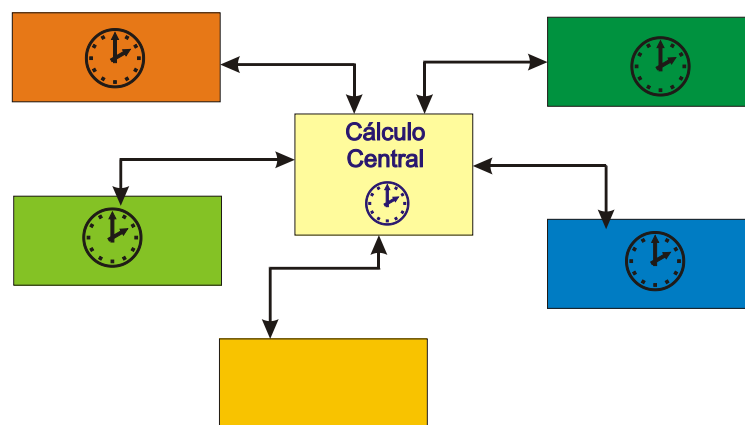


Figura 2.28 – Arquitetura com Cálculo Central.

Os outros blocos da figura 2.28 são os nós da rede. Os nós com relógio são os relógios em que o ponto central de cálculo utiliza como referência para o cálculo da função de convergência, além de receber o *broadcast* com a correção do relógio. O nó, em amarelo, que não possui relógios somente recebe os dados de correção do relógio, não fornecendo dados para o cálculo da função de convergência.

Este tipo de arquitetura pode ser tolerante a erros bizantinos, mas assim como a centralizada, possui um ponto comum de falha, isto é, se o nó aonde o cálculo é realizado falhar, toda a sincronização fica comprometida. Para

eliminar este ponto comum de falha é possível utilizar uma estratégia parecida com o multi-mestre, isto é, se o ponto comum do cálculo da sincronização falhar, outro nó da rede assume a tarefa de calcular a re-sincronização.

Esta é uma arquitetura mais nova, não contendo muitas referências sobre o assunto. Com esta arquitetura de cálculo central é possível utilizar tanto em sincronização interna quanto externa. Um dos protocolos de comunicação que está utilizando este tipo de arquitetura de sincronização é o protocolo TTEthernet.

## **2.6 Algoritmos de Sincronização**

Em uma aplicação de tempo real típica, o sistema de computação distribuída executa suas diferentes funções simultaneamente. Estas funções são, normalmente, executadas em diferentes nós. Para garantir um comportamento consistente e estável do sistema distribuído, é necessário que todos os nós do processo tenham a mesma ordem temporal, isto é, que seus relógios estejam dentro dos limites da sincronização.

Para manter os relógios dos nós em sincronia, existem algoritmos de sincronização que são baseados em lógicas matemáticas que, basicamente, definem as operações que o sistema deve realizar para manter seus relógios sincronizados ou com o tempo dentro de uma precisão pré-especificada.

Existe uma gama de algoritmos de sincronização, este trabalho irá abordar o Algoritmo de *Welch-Lynch*, conhecido como FTM, o algoritmo de *Welch-Lynch* modificado por *Kopetz-Ochsenreiter*, conhecido como FTA e sincronização de relógios utilizando Filtro de Kalman.



### 2.6.1 Algoritmo Welch-Lynch

O algoritmo de Welch-Lynch, também conhecido como FTM (*Fault-Tolerant Mid-Point*), é um algoritmo com arquitetura distribuída e que apresenta tolerância a falhas bizantinas para a sincronização de relógios de sistemas distribuídos (Lundelius; Lynch, 1984).

Muitas informações sobre o algoritmo de Welch-Lynch pode ser encontrado na literatura, como em Dutertre (1998), onde é feito um relatório com as provas de que o algoritmo mantém os relógios dentro de uma dada precisão. Lundelius e Lynch (1984) é o artigo original do algoritmo, com muitas informações relevantes sobre o mesmo. Neste trabalho, o algoritmo será explicado abaixo, da melhor forma possível.

O algoritmo de Welch-Lynch faz uma série de suposições, tais como:

- a) Todos os nós estão conectados por uma rede de comunicação confiável;
- b) Relógios da rede estão inicialmente sincronizados entre si, ou seja, não possuem viés inicial;
- c) A comunicação por rede é confiável e os atrasos de transmissão da rede são limitados por duas constantes:

$$0 \leq \varepsilon < \delta \quad (2.36)$$

- d) O nó não possui controle sobre o relógio físico, somente sobre o relógio lógico; Assim, o tempo local para o nó é dado por um relógio virtual (*VC*) obtido adicionando a correção ao relógio físico (*PC*). A correção é periodicamente computada e armazenada na variável local *ADJ*. O relógio virtual do nó então é definido por:

$$VC(t) = PC(t) + Adj \quad (2.37)$$

- e) Para ser tolerante a falhas bizantinas, a condição da equação 2.32 deve ser obedecida;
- f) Cada nó tem seu relógio físico, o qual pode estar com uma deriva menor (ou maior) em relação ao tempo real a uma taxa limitada por uma pequena constante  $\rho$  tal que  $0 < \rho \ll 1$ .
- g) Se o relógio C não falhar durante o intervalo do tempo real  $[t_1, t_2]$ , então a condição abaixo deve ser obedecida:

$$(1 - \rho)(t_2 - t_1) \leq C(t_2) - C(t_1) \leq (1 + \rho)(t_2 - t_1) \quad (2.38)$$

Onde  $C(t_1)$  e  $C(t_2)$  são os valores do relógio C no tempo  $t_1$  e no tempo  $t_2$  respectivamente. O tempo decorrido do relógio  $C(t_2) - C(t_1)$  esta dentro de  $\rho(t_2 - t_1)$  do atraso de tempo real  $t_2 - t_1$ . Durante o mesmo intervalo, os relógios físicos de dois nós podem derivar separadamente no máximo  $2\rho(t_2 - t_1)$ . Até mesmo para valores pequenos de  $\rho$ , o erro pode ser significativo para grandes valores de  $t_2$ .

Para assegurar que todos os nós tenham uma consistente visão do tempo, é necessária a re-sincronização de relógios regularmente (periodicamente). Para isso o algoritmo segue uma seqüência lógica. Cada nó aplica essa seqüência com o objetivo de chegar a um termo de correção. Com esse termo de correção, os desvios dos relógios causados pela deriva, são corrigidos para que todos os relógios do sistema estejam dentro de uma determinada precisão.

A figura 2.29 mostra o fluxograma do algoritmo de Welch-Lynch:

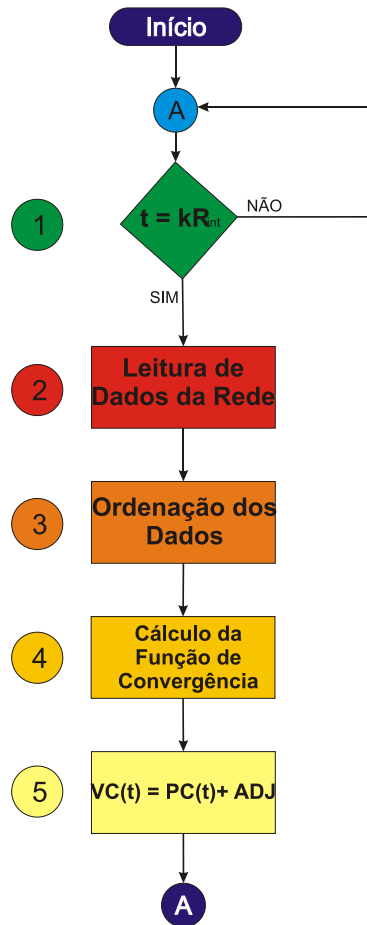


Figura 2.29 – Fluxograma do Algoritmo de Welch-Lynch.

Na figura 2.29, onde se tem o número 1 indicando, tem-se um *loop* de condição. Esta condição significa que, se o tempo do relógio local do nó é igual ao tempo de re-sincronização, então a sincronização tem o seu início.  $R_{int}$  é o período de re-sincronização pré-determinado e  $k$  é o instante.

No número 2, onde se tem a leitura de dados da rede, significa que os relógios locais trocam informações de seus relógios entre si, isto é, todos os relógios enviam um “broadcast” com a estampa de tempo de seu relógio. A figura 2.30, ajuda a entender como o “broadcast” é realizado. Neste exemplo utiliza-se um sistema com 4 nós.

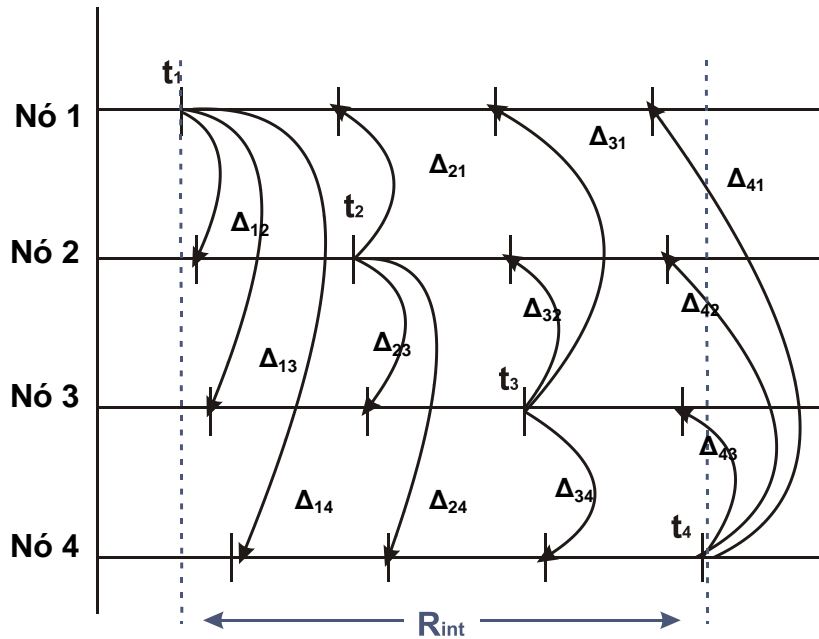


Figura 2.30 – Broadcast do Algoritmo de Welch-Lynch.

Na figura 2.30 o nó 1 envia sua estampa de tempo,  $t_1$ , ao nó 2, 3 e 4. O nó 2, envia sua estampa de tempo,  $t_2$ , ao nó 1, 3 e 4. O nó 3 envia sua estampa de tempo,  $t_3$ , ao nó 1, 2 e 3. O nó 4 envia sua estampa de tempo,  $t_4$ , ao nó 1, 2 e 3. Quando os valores da estampa de tempo chegam ao nó, é feita a diferença de tempo entre o valor recebido e o valor da estampa de tempo do nó que recebeu. Com isso tem-se uma matriz de diferenças de tempo dada na equação 2.39.

$$D = \begin{bmatrix} \Delta_{11} & \Delta_{12} & \Delta_{13} & \Delta_{14} \\ \Delta_{21} & \Delta_{22} & \Delta_{23} & \Delta_{24} \\ \Delta_{31} & \Delta_{32} & \Delta_{33} & \Delta_{34} \\ \Delta_{41} & \Delta_{42} & \Delta_{43} & \Delta_{44} \end{bmatrix} \quad (2.39)$$

Na matriz de diferenças de tempo, da equação 2.39, o  $\Delta$  indica a diferença das estampas de tempo dos relógios do nó. Onde o índice que indica a linha indica o nó do relógio que enviou sua estampa de tempo, e o índice que indica a coluna indica o nó do relógio que recebe a estampa de tempo. Quando os índices de linha e coluna são iguais, indica que o relógio enviou e recebeu seu próprio valor, e com isso a diferença de tempo é nula. Então, é possível reescrever a matriz da equação 2.39:

$$D = \begin{bmatrix} 0 & \Delta_{12} & \Delta_{13} & \Delta_{14} \\ \Delta_{21} & 0 & \Delta_{23} & \Delta_{24} \\ \Delta_{31} & \Delta_{32} & 0 & \Delta_{34} \\ \Delta_{41} & \Delta_{42} & \Delta_{43} & 0 \end{bmatrix} \quad (2.40)$$

A equação 2.40 mostra a matriz de desvios de tempo dos nós do sistema onde cada nó consegue enxergar somente a sua linha.

Após a formação da matriz de desvios de tempo, o próximo passo do algoritmo é ordenar os dados.

Na figura 2.29, o número 2 indica a ordenação de dados. Cada nó enxerga somente a sua linha da matriz 2.40, formando um vetor de valores  $A$ . Esse vetor é ordenado em ordem crescente, seguindo a regra da equação 2.41:

$$A[1] \leq A[2] \leq \dots \leq A[n] \quad (2.41)$$

Onde  $n$  indica o número de relógios ou nós do sistema, e o vetor  $A$  é formado pelos valores da linha do nó na equação 2.40.

Após ordenar os dados de forma crescente, o algoritmo de Welch-Lynch faz o cálculo da função de convergência, representado pelo número 3 na figura 2.29.

Com os dados do vetor  $A$  ordenados de forma crescente, agora é necessário encontrar o ponto-médio do vetor. Descarta-se o maior e o menor valor de  $A$ . Esse descarte faz com que a função seja tolerante a falhas bizantinas. Depois faz-se a média aritmética do maior e menor valor dos elementos restantes no vetor. A equação 2.42, mostra a função de ponto-médio tolerante a falhas bizantinas, onde assumimos que  $f$  é o número de falhas que o algoritmo pode tolerar e  $n$  é o número de relógios ou nós no sistema:

$$cfn(A) = \frac{A[f + 1] + A[n - f]}{2} \quad (2.42)$$

A equação 2.42 encontra o ponto médio do vetor, com isso agora é possível encontrar a função de ajuste, dado pela equação 2.43:

$$Adj = kR_{int} + \delta - cfn(A) \quad (2.43)$$

Onde  $k$  é o intervalo,  $R_{int}$  é o período de re-sincronização,  $\delta$  é o termo de compensação do atraso de rede e  $cfn(A)$  é a função de ponto-médio tolerante a falhas bizantinas.

A função de ajuste foi calculada em 2.43. Representado o número 5 da figura 2.29, o relógio virtual agora corrige o seu valor através da equação 2.44.

$$VC(t) = PC(t) + Adj \quad (2.44)$$

Onde,  $VC(t)$  indica o relógio virtual,  $PC(t)$  é o relógio físico e  $Adj$  é o ajuste calculado pelo algoritmo.

Entre duas sucessivas re-sincronizações, os relógios virtuais podem derivar entre si, mas com o ajuste em todos os nós do sistema faz com que todos os relógios fiquem com um desvio limitado. Esse desvio limitado é chamado de precisão.

De acordo com Kopetz (1997), a flutuação do tempo, a flutuação no atraso de rede, a flutuação e o atraso causado pelo agendador (*scheduler*) do computador, o sistema operacional, interrupções, entre outros, afeta e deteriora a precisão da sincronização de relógios.

A Tabela 2.2 apresenta valores de flutuação nas mensagens de sincronização que o sistema pode apresentar em diferentes níveis de análise.

Tabela 2.2 – Tabela de valores da flutuação em mensagens de sincronização.

<b>Mensagem de Sincronização</b>	<b>Valores aproximados da Flutuação</b>
<b>Em nível de software</b>	500 $\mu$ s a 5 ms
<b>No núcleo do sistema operacional</b>	10 $\mu$ s a 100 $\mu$ s
<b>No hardware de controle de comunicação</b>	Menor que 10 $\mu$ s

Fonte: Traduzida de Kopetz (1997).

De acordo com Scheinder (1988) a precisão do algoritmo de Welch-Lynch é dada pela equação 2.45:

$$\Pi(\Gamma, \varepsilon) = \frac{\Gamma}{2} + \varepsilon \quad (2.45)$$

Onde  $\Gamma$  é o máximo offset possível entre dois relógios e  $\varepsilon$  é a flutuação, que pode ser causada por diversos fatores.

Para que o sistema distribuído alcance uma ótima precisão em seu tempo global é importante que a flutuação nas mensagens de sincronização seja pequena.

### **2.6.2 Algoritmo FTA (Fault-Tolerant Average)**

O algoritmo FTA (*Fault-Tolerant Average*) é o algoritmo de Welch-Lynch modificado por *Kopetz e Ochsenreiter*. O algoritmo FTA possui uma arquitetura distribuída e apresenta tolerância a falhas bizantinas para a sincronização de relógios de sistemas distribuídos. Mais informações sobre o algoritmo pode ser encontrado na literatura, como em *Kopetz e Ochsenreiter (1987)*.

O Algoritmo FTA é praticamente igual ao algoritmo de Welch-Lynch, modificando somente a forma de calcular a função de convergência. Portanto o algoritmo segue toda a seqüência e condições apresentadas anteriormente para o algoritmo Welch-Lynch.

O algoritmo ordena os dados de forma crescente, assim como o algoritmo de Welch-Lynch, e faz o cálculo da função de convergência. Os dados da matriz 2.40 são guardados em um vetor  $A$ . Com os dados do vetor  $A$  ordenados de forma crescente, agora é necessário encontrar o valor-médio do vetor. Com isso descarta-se o maior e o menor valor de  $A$ , esse descarte faz com que a função seja tolerante a falhas bizantinas. Agora, diferentemente do algoritmo de Welch-Lynch, o algoritmo FTA faz a média aritmética dos elementos restantes no vetor. A equação 2.46, mostra a função de valor-médio tolerante a falhas bizantinas, onde assumimos que  $f$  é o número de falhas que o algoritmo pode tolerar e  $n$  é o número de relógios ou nós no sistema:

$$cf_n(A) = \frac{1}{n - 2f} \sum_{i=f+1}^{n-f} \Delta_{i,j} \quad (2.46)$$

A equação 2.46 encontra o valor médio do vetor e com isso, agora é possível encontrar a função de ajuste, dado pela equação 2.47:

$$Adj = kR_{int} + \delta - cf_n(A) \quad (2.47)$$

Onde  $k$  é o intervalo,  $R_{int}$  é o período de re-sincronização,  $\delta$  é o termo de compensação do atraso de rede e  $cf_n(A)$  é a função de valor-médio tolerante a falhas bizantinas.

Todos os outros passos deste algoritmo FTA são iguais aos do algoritmo de Welch-Lynch.



A exatidão e precisão de ambos os algoritmos FTA e Welch-Lynch dependem de:

- a) A taxa de deriva dos relógios físicos locais e a duração do intervalo de re-sincronização;
- b) Erro ou flutuação na leitura de um relógio em relação a outro;
- c) Falhas no sistema causadas por perda de mensagens na rede ou falhas maliciosas (conhecidos como erros bizantinos);

A precisão do algoritmo FTA de acordo com Kopetz e Ochsenreiter (1987) e Schneider (1986) obedece a equação 2.48:

$$\Pi(\Gamma, n, f, \varepsilon) = \frac{f\Gamma}{n - 2f} + \varepsilon \quad (2.48)$$

Onde  $\Gamma$  é o máximo offset possível entre dois relógios,  $\varepsilon$  é a flutuação, que pode ser causada por vários fatores e  $f$  é o número de falhas bizantinas que o sistema distribuído pode suportar.

De acordo com Kopetz e Ochsenreiter (1987) para o sistema ser tolerante a falhas bizantinas existe um custo em relação à precisão da sincronização. Esse custo é dado pela equação 2.46:

$$u(n, f) = \frac{n - 2f}{n - 3f} \quad (2.49)$$

Com a equação 2.49 é possível montar uma tabela de custo da tolerância a falhas bizantinas em relação à precisão da sincronização.

Tabela 2.3 – Tabela de custo de tolerância à falhas bizantinas em relação à precisão.

Falhas f	Número de nós n									
	4	5	6	7	8	9	10	15	20	30
1	2	1,5	1,33	1,25	1,2	1,16	1,14	1,08	1,06	1,03
2				3	2	1,66	1,5	1,22	1,14	1,08
3							4	1,5	1,27	1,14
4								2,33	1,5	1,22

Fonte: Traduzida de Kopetz (1997).

É possível reescrever a equação 2.48 incluindo o custo da tolerância a falhas bizantinas da equação 2.49, tem-se então a equação 2.50 dada por Kopetz (1997).

$$\Pi(\Gamma, n, f, \varepsilon) = \frac{n - 2f}{n - 3f} \left( \frac{f\Gamma}{n - 2f} + \varepsilon \right) = \left( \frac{f\Gamma}{n - 2f} + \varepsilon \right) u(n, f) \quad (2.50)$$

É possível também aplicar ao algoritmo de Welch-Lynch o conceito de custo da falha bizantina.

Existem muitas outras funções de convergência propostas para sincronização interna de relógios. Em Schneider (1988) existem análises de várias funções de convergência diferentes.

## 2.7 Filtro de Kalman

Segundo Kuga (2005), o Filtro de Kalman é um conjunto de equações matemáticas que provêem uma solução linear recursiva ótima para problemas de estimação linear. Além disso, O filtro de Kalman é um estimador que pode incorporar ruído dinâmico no modelo da dinâmica e no modelo de medidas. O Filtro possui quatro casos: 1) Contínuo-Contínuo, onde a dinâmica do sistema e as observações são contínuas; 2) Contínuo-

Discreto, onde a dinâmica do sistema é contínua e as observações discretas; 3) Discreto-Contínuo, onde a dinâmica do sistema é discreta e as observações contínuas; 4) Discreto-Discreto, onde a dinâmica do sistema é discreta e as observações discretas. Neste trabalho, será utilizado o modelo Discreto-Discreto, pois estaremos trabalhando com relógios digitais que fornecem dados digitais e possuem sua dinâmica de forma discreta.

O filtro de Kalman é dividido em duas etapas: 1) Propagação ou predição; 2) Atualização ou correção. A fase de propagação propaga o estado e a covariância do instante. A fase de atualização corrige o estado e a covariância para o instante, devido à medida. O método tem, portanto, natureza recursiva e não necessita armazenar as medidas previamente em grandes matrizes.

As equações formais são as seguintes:

Fase de Propagação:

$$\bar{x}_k = \Phi_{k,k-1} \hat{x}_{k-1} \quad (2.51)$$

$$\bar{P}_k = \Phi_{k,k-1} \hat{P}_{k-1} \Phi_{k,k-1}^t + \Gamma_k Q_k \Gamma_k^t \quad (2.52)$$

Onde  $\bar{x}_k$  e  $\bar{P}_k$  representam o estado e a covariância propagada no instante  $k$ .

Fase de Atualização:

$$K_k = \bar{P}_k H_k^t (H_k \bar{P}_k H_k^t + R_k)^{-1} \quad (2.53)$$

$$\hat{P}_k = (I - K_k H_k) \bar{P}_k \quad (2.54)$$

$$\hat{x}_k = \bar{x}_k + K_k (y_k - H_k \bar{x}_k) \quad (2.55)$$

Onde  $K_k$  é o ganho de Kalman, e  $\hat{x}_k$  e  $\hat{P}_k$  são o estado e a covariância atualizada no instante  $k$ .

Com isso é possível aplicar o filtro de Kalman para sincronização de relógios. O filtro de Kalman se mostra muito eficiente, principalmente, para

eliminar problemas de viés inicial de relógios, como podemos ver em Marques Filho et. al. (2003). Maiores informações da aplicação do filtro de Kalman em sincronização de relógios podem ser encontradas também em Varnum (1983).

### 2.7.1 Modelo Matemático

Existem muitos modelos para se representar um relógio. O modelo mais simples de um relógio, baseado em Varnum (1983), é definido por duas variáveis de estado, o viés instantâneo e a deriva, como na equação 2.56.

$$\begin{bmatrix} \dot{T} \\ \dot{D} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} T \\ D \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \quad (2.56)$$

Onde T é o viés instantâneo, D a deriva e  $w_1$  e  $w_2$  são os vetores de ruídos gaussianos brancos que representam a parte estatística, estocástica, da equação, que tem por objetivo modelar as flutuações, erros no modelo, e outros erros de natureza aleatória ou impossíveis de serem modelados.

Neste modelo de sincronização será usada uma arquitetura centralizada, interligados por uma rede de comunicação confiável. Este algoritmo, que utiliza o Filtro de Kalman, não é tolerante a falhas. E feita uma série de suposições ao algoritmo, tais como:

- a) Todos os nós estão conectados por uma rede de comunicação confiável;
- b) A comunicação por rede é confiável e os atrasos de transmissão da rede são limitados por duas constantes:

$$0 \leq \varepsilon < \delta \quad (2.57)$$

- c) Cada nó tem seu relógio físico a qual pode estar com uma deriva menor (ou maior) em relação ao tempo real a uma taxa limitada por uma pequena constante  $\rho$  tal que  $0 < \rho \ll 1$ .
- d) Se o relógio C não falhar durante o intervalo do tempo real  $[t_1, t_2]$ , então deve obedecer a condição:

$$(1 - \rho)(t_2 - t_1) \leq C(t_2) - C(t_1) \leq (1 + \rho)(t_2 - t_1) \quad (2.58)$$

Onde  $C(t_1)$  e  $C(t_2)$  são os valores do relógio C no tempo  $t_1$  e no tempo  $t_2$  respectivamente. O tempo decorrido do relógio  $C(t_2) - C(t_1)$  esta dentro de  $\rho(t_2 - t_1)$  do atraso de tempo real  $t_2 - t_1$ . Durante o mesmo intervalo, os relógios físicos de dois nós podem derivar separadamente no máximo  $2\rho(t_2 - t_1)$ . Até mesmo para valores pequenos de  $\rho$ , o erro pode ser significativo para grandes valores de  $(t_2 - t_1)$ .

Em Kopetz (1997), definiu-se como é medido o viés instantâneo e a deriva, respectivamente nas equações 2.59 e 2.60:

$$T_k = t_k^2 - t_k^1 \quad (2.59)$$

$$D_k = \frac{t_k^2 - t_{k-1}^2}{t_k^1 - t_{k-1}^1} - 1 \quad (2.60)$$

Onde,  $t_k^2$  é o valor do tempo no relógio local, definido como 2, no instante k e o  $t_k^1$  é o valor do tempo no relógio de referência, definido como 1, no instante k.

O modelo de medidas do filtro de Kalman, neste trabalho é um modelo discreto. O modelo de medidas proposto, também baseado em Varnum (1983), é dado por:

$$y_k = H_k x_k + v_k \quad (2.61)$$

Com:

$$v = N(0, R_k(t)) \quad (2.62)$$

Onde  $y_k$  é o vetor de  $m$ -medidas,  $H_k$  é a matriz de observação que relaciona as medidas com os estados e  $x_k$  é o vetor de  $n$ -estados do sistema,  $v_k$  é a matriz de ruídos gaussianos  $m \times m$ , todos os termos estão no instante  $t_k$ .

Com isso o modelo proposto é dado por:

$$x_k = \begin{bmatrix} T_k \\ D_k \end{bmatrix} \quad (2.63)$$

$$H_k = [1 \quad \Delta t] \quad (2.64)$$

Substituindo as equações 2.63 e 2.64 em 2.61 tem-se:

$$y_k = [1 \quad \Delta t] \begin{bmatrix} T_k \\ D_k \end{bmatrix} + v_k \quad (2.65)$$

Onde  $\Delta t$  é o tempo decorrido.

Fazendo as equações (2.59) e (2.60) em (2.65), tem-se o modelo de medidas:

$$y_k = [1 \quad \Delta t] \begin{bmatrix} t_k^2 - t_k^1 \\ \frac{t_k^2 - t_{k-1}^2}{t_k^1 - t_{k-1}^1} - 1 \end{bmatrix} + v_k \quad (2.66)$$

Com este modelo de medidas e o modelo dos relógios é possível utilizar o filtro de Kalman para sincronização de relógios. Este modelo não é tolerante a falhas bizantinas, mas o modelo mostrou-se simples e com bons resultados.

## **2.8 Ferramentas de Simulação**

Não existem muitas ferramentas específicas disponíveis para simulação de algoritmos de sincronização de relógios. Mas podemos citar a ferramenta de simulação SIDERA e o TrueTime. Outros simuladores para algoritmos de sincronização podem ser encontrados na literatura, como o NTP Simulator e o SynUTC.

Neste trabalho foi utilizado a ferramenta TrueTime, pois se mostrou uma ferramenta com a possibilidade de analisar, programar e simular algoritmos de sincronização em conjunto com o sistema de controle. Foi apresentado também a ferramenta SIDERA, como uma ferramenta de simulação para uso futuro de sincronização de relógios.

### **2.8.1 TrueTime**

O TrueTime é um simulador baseado no Matlab/Simulink para sistemas de controle em tempo real. Com o TrueTime é possível simular o comportamento temporal de kernels de tempo real multi-tarefa contendo tarefas de controle e estudar os efeitos da CPU e agendamento da rede sob a performance do controle.

O kernel de simulação de tempo real é “event-driven” e pode manipular interrupções externas assim como detalhes de alto grau no caso de switches. Políticas arbitrárias de agendamento podem ser definidas, e tarefas de controle podem ser implementadas usando funções na linguagem C, arquivos M do Matlab ou diagramas em blocos do Simulink.

A Figura 2.31 apresenta as bibliotecas do TrueTime na forma de blocos do Simulink.

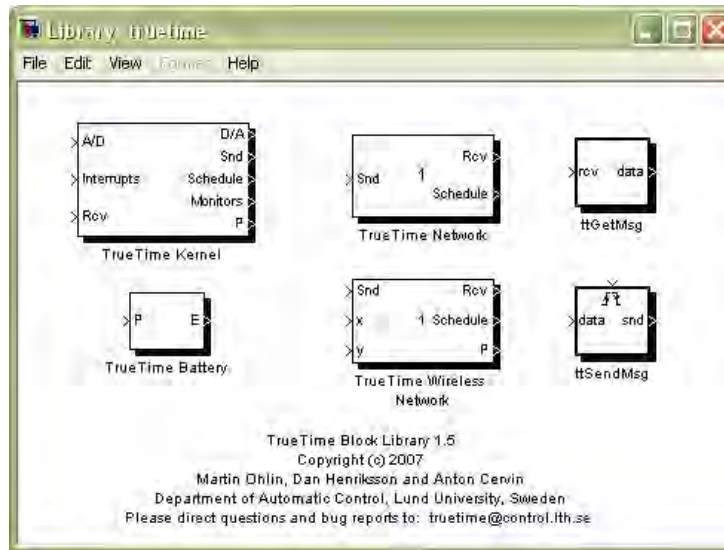


Figura 2.31 – Biblioteca do True Time 1.5.

Seis modelos simples de rede podem ser simulados: CSMA/CD (e.g. Ethernet), CSMA/AMP (e.g. CAN), Round Robin (e.g. Token Bus); FDMA, TDMA (e.g. TTP), e Switched Ethernet (e.g. AFDX). Em sua *versão 2.0 Beta 4*, já são suportados mais dois protocolos de comunicação para simulação: FlexRay e PROFINET. O retardo de propagação é ignorado, porque este é muito pequeno em uma rede de área local. Só a simulação ao nível de pacote é suportada. Supõe-se que protocolos de níveis maiores dividem longas mensagens em pacotes menores.

O TrueTime não possui os algoritmos de sincronização implementados e prontos a serem simulados, mas o TrueTime possui funções para que se possa trabalhar com os relógios virtuais de cada processador virtual. Cada processador virtual utiliza o relógio do simulink como seu relógio de referência. Neste trabalho foram implementados no TrueTime 1.5, através de programação, simulações com o algoritmo Welch-Lynch (FTM), e o Filtro de Kalman, para a sincronização dos relógios virtuais dos sistemas de controle de tempo real implementados nos processadores virtuais do TrueTime.



## 2.8.2 SIDERA

O SIDERA (Simulation environment for DEpendable Real-time Architectures), é um simulador baseado na arquitetura TTA e realiza simulações de larga escala para sistema de tempo real. O SIDERA prove simulação de vários serviços do protocolo de tempo real, tais como: 1) Startup; 2) A comunicação; 3) A sincronização de relógios; 4) O protocolo de erro de detecção; 5) Manipulação dos dados.

O modulo de simulação de falhas testa a estabilidade do sistema em presença de nós com falhas. O SIDERA ainda suporta simulações do protocolo de comunicação FlexRay.

A figura 2.32 mostra a estrutura do simulador SIDERA:

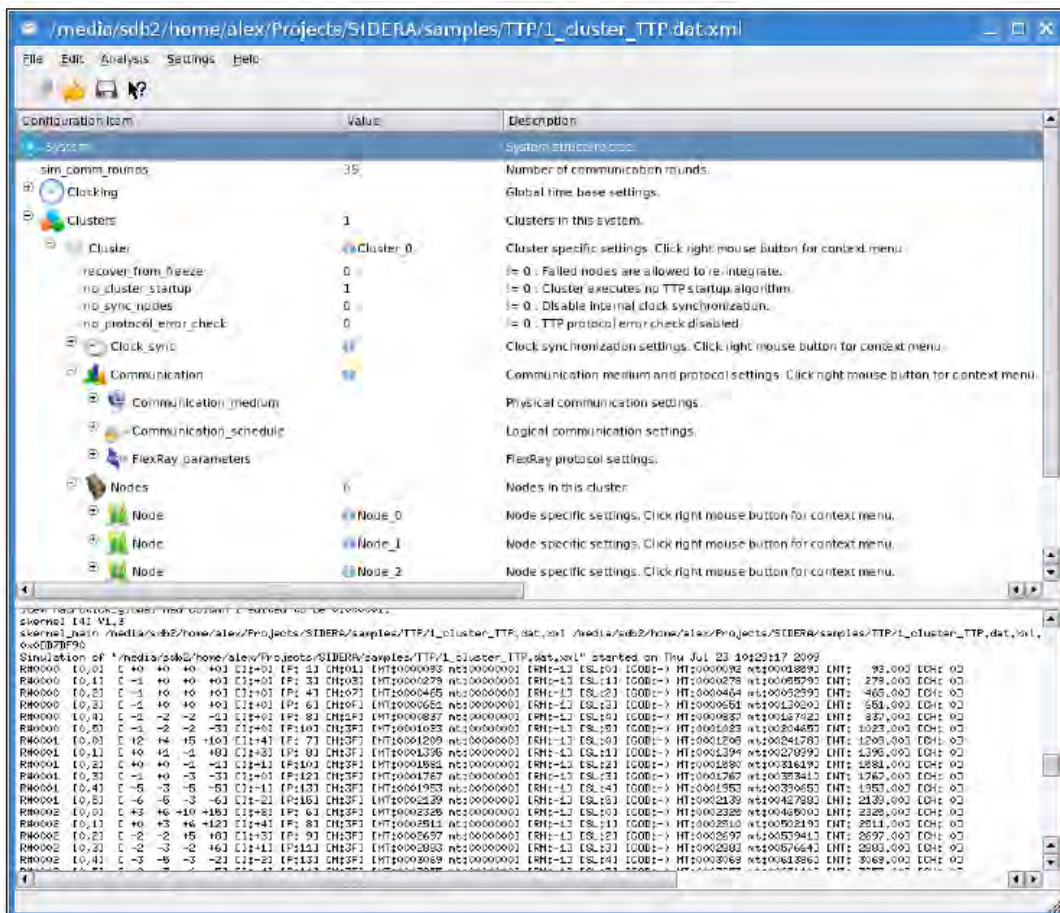


Figura 2.32 – Estrutura do SIDERA.

O SIDERA é uma ferramenta de simulação independente e funciona em sistemas operacionais Linux, criada por Hanzlik (2004). Maiores informações sobre a ferramenta pode ser encontrada em Hanzlik (2004).

### **3 FORMULAÇÃO DO PROBLEMA E ABORDAGENS PARA SUA SOLUÇÃO**

O problema de sincronização de relógios pode aparecer em muitos contextos diferentes, mas a causa da dessincronização é comum e normalmente relacionada às imperfeições de um relógio, vide Figura 2.9, que incluem a deriva (“*drift*”), o viés inicial (“*offset*”), a flutuação (“*jitter*”) e também o erro de estado (*state error*). Além destas conhecidas imperfeições de relógio que prejudicam a ação do sistema, a latência das redes de comunicação também prejudica o sistema e em contrapartida o processo de sincronização.

O problema que se quer resolver consiste em: 1) descrever as principais características e diferenças dos algoritmos de sincronização a serem considerados, no caso o FTM (*Fault-Tolerant Mid-Point*) e o FTA (*Fault-Tolerant Average*); 2) Simular e analisar a sincronização de relógios com o uso dos Algoritmos FTM e Filtro de Kalman e as suas influências sobre um Sistema de Controle com uma Arquitetura Federada (AF); 3) Simular e analisar a sincronização de relógios com o uso do algoritmo FTM e a sua influências sobre um Sistema de Controle por Redes, no caso uma Arquitetura Integrada, utilizando dois protocolos de comunicação um TDMA e outro Ethernet.

#### **3.1. Análise dos Algoritmos de Sincronização de Relógios FTM e FTA**

Existem muitos algoritmos de sincronização de relógios na literatura, foram escolhidos dois algoritmos com arquitetura de sincronização de relógios distribuída para análise, simulação e comparação. Os algoritmos escolhidos são o algoritmo de sincronização de relógios FTM (*Fault-Tolerant Mid-Point*), conhecido também como Welch-Lynch, e o FTA (*Fault-Tolerant Average*).

O Objetivo desta simulação é mostrar em vários casos simulados, as diferenças e semelhanças que ambos os algoritmos, FTM e FTA, possuem.

Além disso, com esta simulação é possível adquirir conhecimento da forma como o algoritmo opera. Com esta simulação é possível variar alguns parâmetros. E com isso vários gráficos para análise foram gerados, tendo assim de forma gráfica método para análise da função de convergência.

### 3.1.1. Metodologia

A metodologia desta simulação consistiu em analisar os dois algoritmos de sincronização de relógios, o FTM e o FTA, variando os parâmetros de deriva de relógio, latência da rede e erro de estado, sempre obedecendo a suas condições e o ao Teorema de erro bizantino. O sistema de comunicação utilizou uma simulação de um sistema TDMA. O processo de sincronização foi desenvolvido com a ajuda dos seguintes recursos de hardware:

- Programa de Simulação: Microsoft Excel versão 2007;
- Sistema Operacional: Windows XP SP2;
- Computador: Intel Celeron, 1.86 Ghz, 1 Gb de Memória RAM,

A Figura 3.1, mostra o modelo do experimento montado:

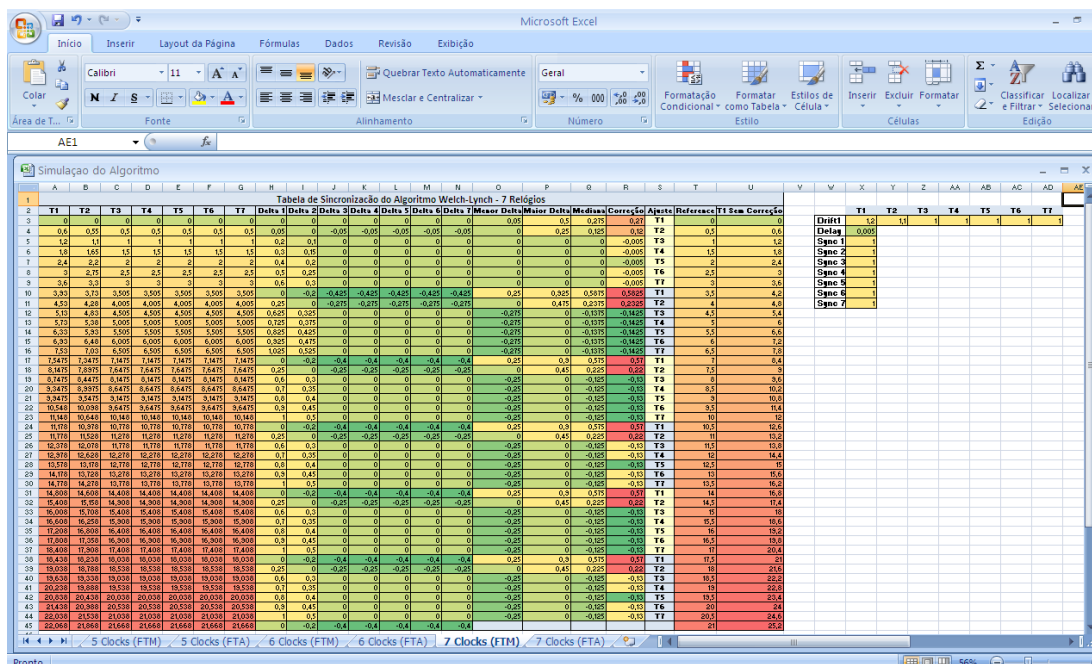


Figura 3.1 – Modelo de Simulação dos Algoritmos FTM e FTA.

Na Figura 3.1, as colunas indicadas por  $T1, T2, T3, T4, T5, T6$  E  $T7$ , são as colunas que simulam os relógios, chamados de Tempos Virtuais Simulados. Nas colunas seqüenciais, existem as colunas de *delta 1 até delta 7* que monta a matriz de diferenças de tempo da equação 2.40. Com estes valores, é aplicado a função de convergência descrito nas equações 2.42. Em seqüência a correção é aplicada novamente nas colunas dos relógios de  $T1$  até  $T7$  com as respectivas correções que são gravadas na coluna *correção*. A coluna *Reference* é o tempo virtual de referência utilizado pelos simulação. A variação dos parâmetros de relógio é explicado logo na seqüência.

### 3.1.2. Modelo Matemático

O modelo de propagação de relógios utilizado foi o modelo geométrico discreto dado por:

$$C_k(t_k) = a_k t_k^R - b_k \quad (3.1)$$

Em que os parâmetros utilizados são:

$$\begin{aligned} b_k &= b_{k-1} = 0 \\ a_k &= a_{k-1} = \text{Constante} \\ t_k^R &= k T_{amostragem} \\ T_{amostragem} &= 0.5 \end{aligned}$$

O modelo de medidas foi dado por:

$$\Delta C_k(\Delta t_k) = \Delta a_k \Delta t_k^R - \Delta b_k \quad (3.2)$$

Com o modelo de medidas, a cada ciclo é feita uma medida formando a seguinte matriz (dada pela equação 2.40):

$$D = \begin{bmatrix} 0 & \Delta_{12} & \dots & \Delta_{1n} \\ \Delta_{21} & 0 & \dots & \Delta_{2n} \\ \Delta_{31} & \Delta_{32} & \ddots & \vdots \\ \Delta_{41} & \Delta_{42} & \dots & 0 \end{bmatrix} \quad (3.4)$$

Onde  $n$  é o número de relógios da simulação. Cada linha desta matriz forma um vetor de medidas  $A$ , que é ordenado de forma crescente seguindo a seguinte regra:

$$A[1] \leq A[2] \leq A[3] \leq \dots \leq A[n] \quad (3.5)$$

Após a ordenação dos dados medidos, o algoritmo de sincronização aplica sua função de convergência dados por:

• **Algoritmo FTM:**

$$cf_n(A) = \frac{A[n - f] + A[f + 1]}{2} \quad (3.6)$$

• **Algoritmo FTA:**

$$cf_n(A) = \frac{1}{n - 2f} \sum_{i=f+1}^{n-f} \Delta_{i,j} \quad (3.7)$$

Onde  $f$  é o número de falhas maliciosas e  $n$  o número de relógios.

Após o cálculo da função de convergência, a cada final de ciclo, todos os relógios da simulação são corrigidos, de acordo com as equações:

$$correção = cf_n(A) - \delta \quad (3.8)$$

$$C_k^i(t_k) = C_k^i(t_k) - correção \quad (3.9)$$

$$i = [1, 2, 3, \dots, n]$$

Onde  $i$  é o número do relógio que esta sendo corrigido e  $\delta$  é o valor de atraso da rede.

### 3.1.3. Variação dos Parâmetros

Com o auxílio do Microsoft Excel, foi criada uma tabela correspondente para poder variar os parâmetros da taxa de deriva dos relógios, conforme a Figura 3.2, de maneira a possibilitar modificar a taxa de deriva de qualquer um dos relógios do conjunto.

V	W	X	Y	Z	AA	AB	AC	AD
		T1	T2	T3	T4	T5	T6	T7
Drift1	1,2	1,1	1	1	1	1	1	1
Delay	0,005							
Sync 1	1							
Sync 2	1							
Sync 3	1							
Sync 4	1							
Sync 5	1							
Sync 6	1							
Sync 7	1							

Figura 3.2 – Modelo de Variação de parâmetros da simulação dos Algoritmos FTM e FTA.

Para a operação de um relógio perfeito, devemos colocar a taxa de deriva igual a 1 (um) no campo descrito *Drift1* da Figura 3.2, conforme equação 3.10.

$$\Delta a = 1 + \rho \quad (3.10)$$

Onde  $\rho$  é a porcentagem que queremos de deriva no relógio correspondente.

É possível variar este parâmetro de Taxa de deriva para todos os 7 relógios, indicados na figura 3.2 por: T1, T2, T3, T4, T5, T6 E T7. Ainda na Figura 3.2, é possível ver um campo correspondente denominado “*delay*”. Neste campo é colocado o valor de atraso de rede ( $\delta$ ), utilizado pela equação 3.8.

Para inserir um erro de estado é necessário colocar no campo *Drift1* o valor de 0 (zero). Com isto, o valor da taxa de deriva será nula. Com o valor zero, o relógio é simulado com um erro de estado, isto é, o relógio está parado.

### 3.1.4. Modelo de Simulação

Nesta simulação feita com o Microsoft Excel, foram simulados 8 relógios. Foi simulado o Tempo Virtual de Referência e os Tempos Virtuais Simulados de cada relógio. A Figura 3.3 ajuda a entender os diversos tipos de relógios que estão sendo utilizados para chegar ao resultado desejado.

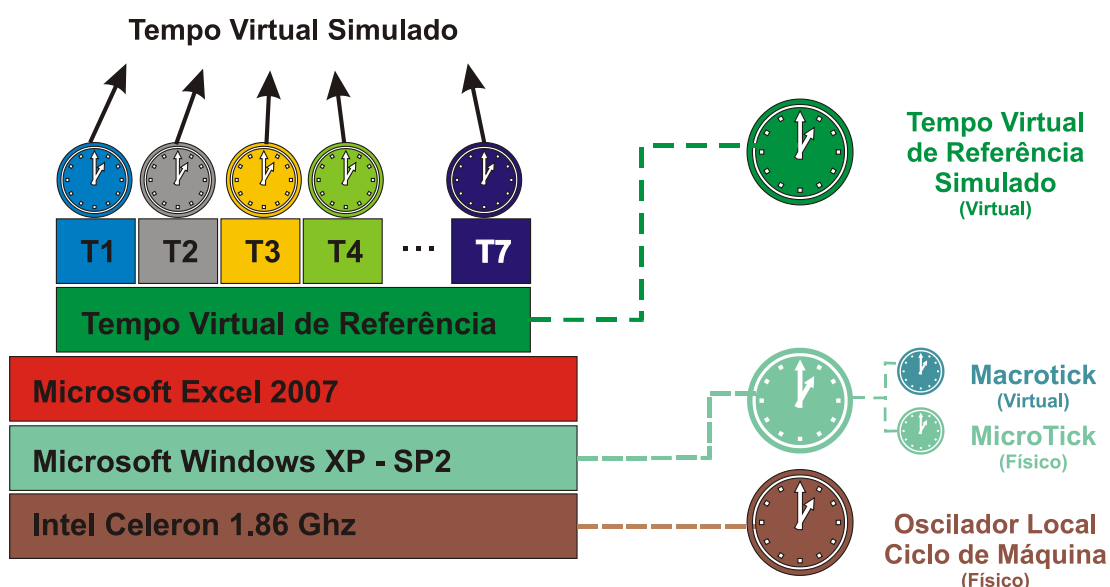


Figura 3.3 – Modelo de Simulação do Excel.

Foi possível variar os parâmetros de deriva (*Clock Drift*) dos relógios locais (Tempo Virtual Simulado) através da interface de configuração criada, como visto na Figura 3.2. Os outros relógios da simulação, tais como o MacroTick, MicroTick e Ciclo de Máquina não foram modificados e tratados nesta simulação. Os resultados são mostrados na seção 4.



### **3.2. Sincronização de Relógios em Sistemas de Controle sobre uma Arquitetura Federada.**

Em sistemas de controle em tempo real a sua funcionalidade e seu comportamento temporal devem ser tão previsíveis quanto forem necessários para satisfazer as especificações do sistema. Problemas de variação do tempo em sistemas de controle distribuídos são problemas inerentes ao sistema que podem prejudicar tanto o resultado temporal quanto o lógico. Para garantir o comportamento temporal, é necessário o uso de mecanismos para o controle do tempo. Normalmente um sistema se utiliza de relógios, tanto físicos quanto lógicos para o controle temporal. Mas as imperfeições inerente aos relógios podem prejudicar tanto a comunicação, quanto a computação que direta ou indiretamente pode afetar o sistema de controle por rede através do dessincronismo de relógios.

Um dos mecanismos para controlar o tempo em um sistema distribuído, i.e., sincronizar os vários nós do sistema distribuído, é a sincronização de relógios. Existem vários métodos e arquiteturas para sincronização de relógios, como descrito anteriormente.

Para mostrar a influência da variação do tempo em sistemas de controle e a eficiência dos algoritmos de sincronização de relógios, nesta seção foi escolhida uma arquitetura de controle com 4 subsistemas de controle independentes.

O sistema escolhido é inspirado na arquitetura HLA (*High Level Architecture*). De acordo com (REIS; SOUZA, 2008) HLA é uma arquitetura de propósito geral para reutilização e interoperabilidade de simulações. Esta arquitetura foi desenvolvida por encomenda do DMSO (*Defense Modeling and Simulation Office*) com o objetivo de possibilitar o reuso e a interoperabilidade entre os diversos tipos de simulação desenvolvidos e mantidos pelo DoD (*Department of Defense*). A HLA permite o desenvolvimento de simulações baseadas em componentes. Antes de ser

uma arquitetura de simulação distribuída, a HLA é uma arquitetura de componentes. Esses componentes são denominados Federados. Os Federados podem ser combinados utilizando a arquitetura HLA, formando outra simulação. Essa simulação combinada, no contexto HLA, é denominada Federação.

Toda troca de dados entre os Federados é realizada através de uma infraestrutura de execução, ou simplesmente RTI (*Run-Time Infrastructure*). Nada é definido pela arquitetura a respeito da construção ou do funcionamento interno dessa infra-estrutura. Isso torna a arquitetura independente da tecnologia e também torna possível o desenvolvimento de RTIs baseadas em várias tecnologias existentes e futuras. A Figura 3.4 mostra uma visão funcional de uma Federação onde diversos tipos de Federados acessam a RTI (representada na figura por Runtime Infrastructure) através de uma interface comum. (REIS, 2008).

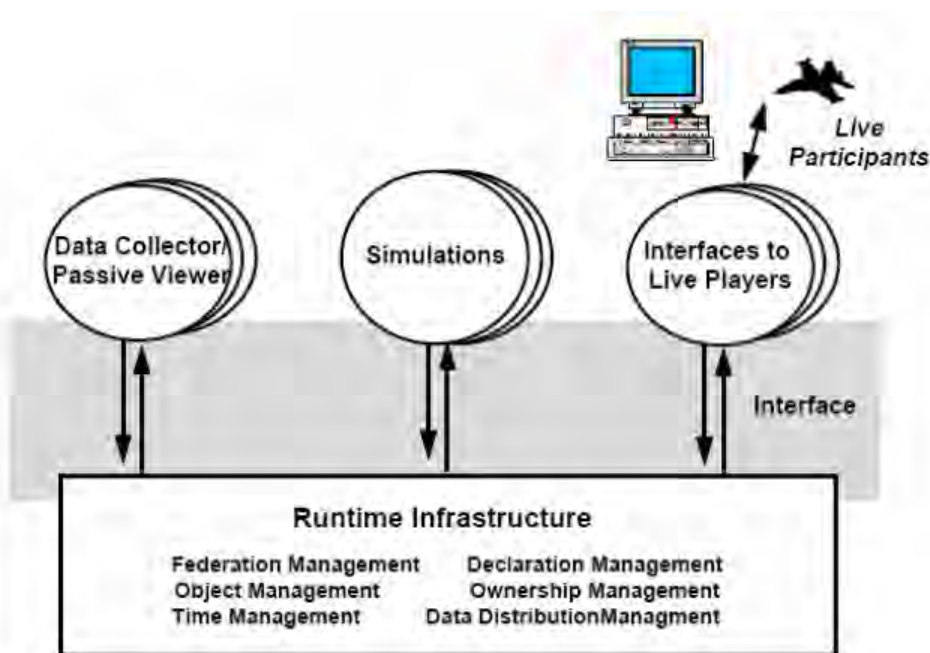


Figura 3.4 – Visão funcional de uma Federação HLA.

Fonte: Dahmann, *et al.* (1998).

A intenção deste trabalho não é reproduzir toda uma simulação de arquitetura HLA e sim reproduzir uma pequena parcela das exigências que o gerenciamento de tempo necessita que é a sincronização de relógios, utilizando de algoritmos de sincronização de relógios e mostrar as influências sobre sistemas de controle com arquitetura federada.

### 3.2.1. Metodologia

A metodologia consiste em analisar um sistema com 4 subsistemas de controle, onde cada subsistema (nó) será denominado, neste trabalho, de federado. Os nós (Federados) estão conectados a um barramento de comunicação. Todo o processo será controlado pelos blocos do TrueTime, que executa no ambiente de simulação MatLab/Simulink.

O atuador de cada Federado é descrito como contínuo no tempo, seguindo a seguinte função de transferência:

$$G(s) = \frac{1000}{s(s + 1)} \quad (3.11)$$

O controlador PID foi implantado de acordo com as seguintes equações:

$$P(k) = K[\beta r(k) - y(k)] \quad (3.12)$$

$$I(k + 1) = I(k) + \frac{Kh}{T_i} [r(k) - y(k)] \quad (3.13)$$

$$D(k) = a_d D(k - 1) + b_d [y(k - 1) - y(k)] \quad (3.14)$$

$$u(k) = P(k) + I(k) + D(k) \quad (3.15)$$

Em que:

$$a_d = \frac{T_d}{N_h + T_d} \quad (3.16)$$

$$b_d = \frac{NKT_d}{N_h + T_d} \quad (3.17)$$

Obedecendo os seguintes parâmetros:

$$w_c = 20 \text{ rad/s}$$

$$\zeta = 0.7$$

Cada nó (federado), possui um relógio lógico, dado pelo computador simulado do TrueTime e utilizam a rede de comunicação para trocar os dados de seus relógios.

Cada federado, além de implementar uma tarefa de controle periódica, também implementa uma tarefa de sincronização de relógios periódica.

O objetivo é sincronizar os relógios lógicos dos Federados, utilizando-se de algoritmos de sincronização de relógios. Foram escolhidos dois algoritmos de sincronização de relógios diferentes, com arquiteturas de sincronização diferentes para fazer esta simulação. Foi escolhido o algoritmo de sincronização de relógios FTM (*Welch-Lynch*), com arquitetura distribuída, e sincronização de relógio com o Filtro de Kalman, com arquitetura centralizada.

### **3.2.2. Modelo do Sistema**

O sistema distribuído, ou Federação, constitui-se de 4 Federados interligados a um barramento de comunicação (RTI). A Figura 3.5 mostra a arquitetura Federada utilizada, com 4 federados interligados por uma RTI. De acordo com a Figura 3.4 uma RTI possui vários serviços para gerenciamento, no nosso caso a RTI é representada por um barramento de comunicação e cada federado implementa uma malha de controle. A Figura 3.9 mostra o nosso modelo de Arquitetura Federada implementada no TrueTime, onde cada bloco laranja implementa um Federado e o bloco em cinza implementa a RTI, ou no nosso caso o barramento de comunicação. Em todos os nossos casos com Arquitetura Federada é utilizado este modelo.

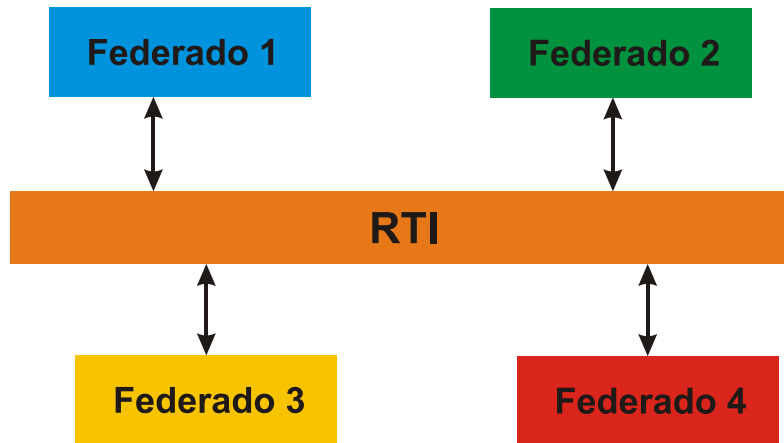


Figura 3.5 – Arquitetura Federada.

Cada Federado da Figura 3.5 implementa a seguinte malha de controle da Figura 3.6, com um simples controlador PID de um atuador eletro-hidráulico:

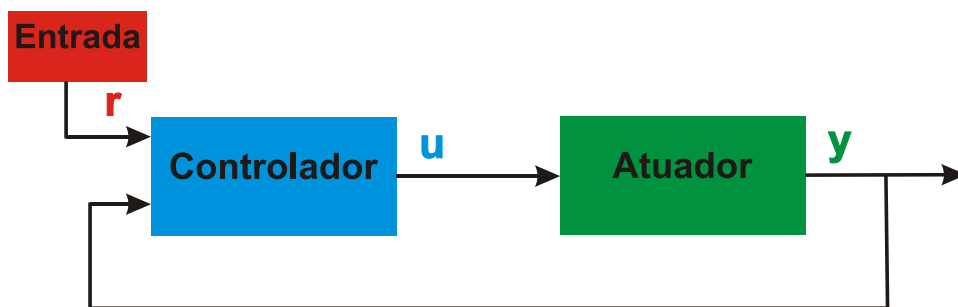


Figura 3.6 – Malha simples de controle do atuador eletro-hidráulico de um Federado.

A Figura 3.7 mostra o modelo de um Federado implementado com os blocos do TrueTime. Cada Federado implementa o controle em um computador de tempo real simulado, dado pelo bloco do TrueTime Kernel da Figura 3.8. Neste bloco são implementadas 3 tarefas periódicas que o computador tem a realizar. A tarefa do controle PID, a tarefa de envio de dados do tempo na rede e a tarefa de sincronismo de relógio. Existe também implementada uma interrupção causada quando chega um dado de rede ao nó (Federado).

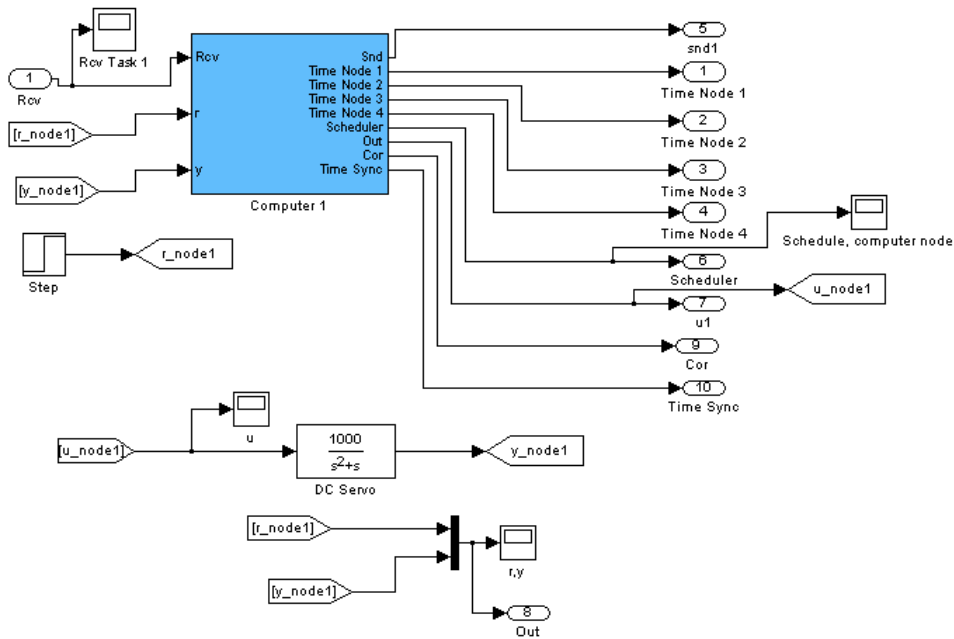


Figura 3.7 – Malha simples de controle do atuador eletro-hidráulico de um Federado pelos blocos do TrueTime.

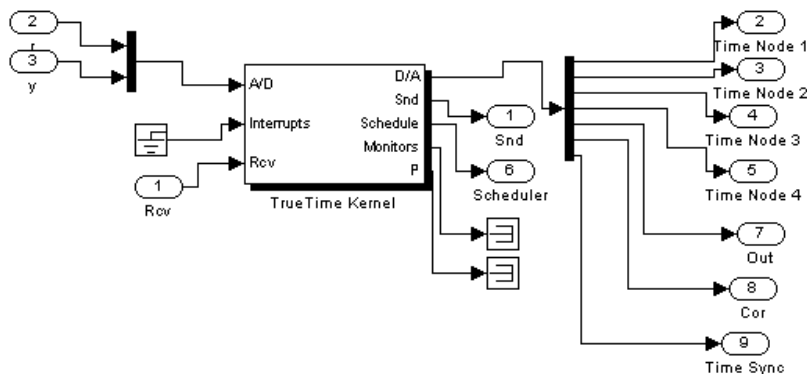
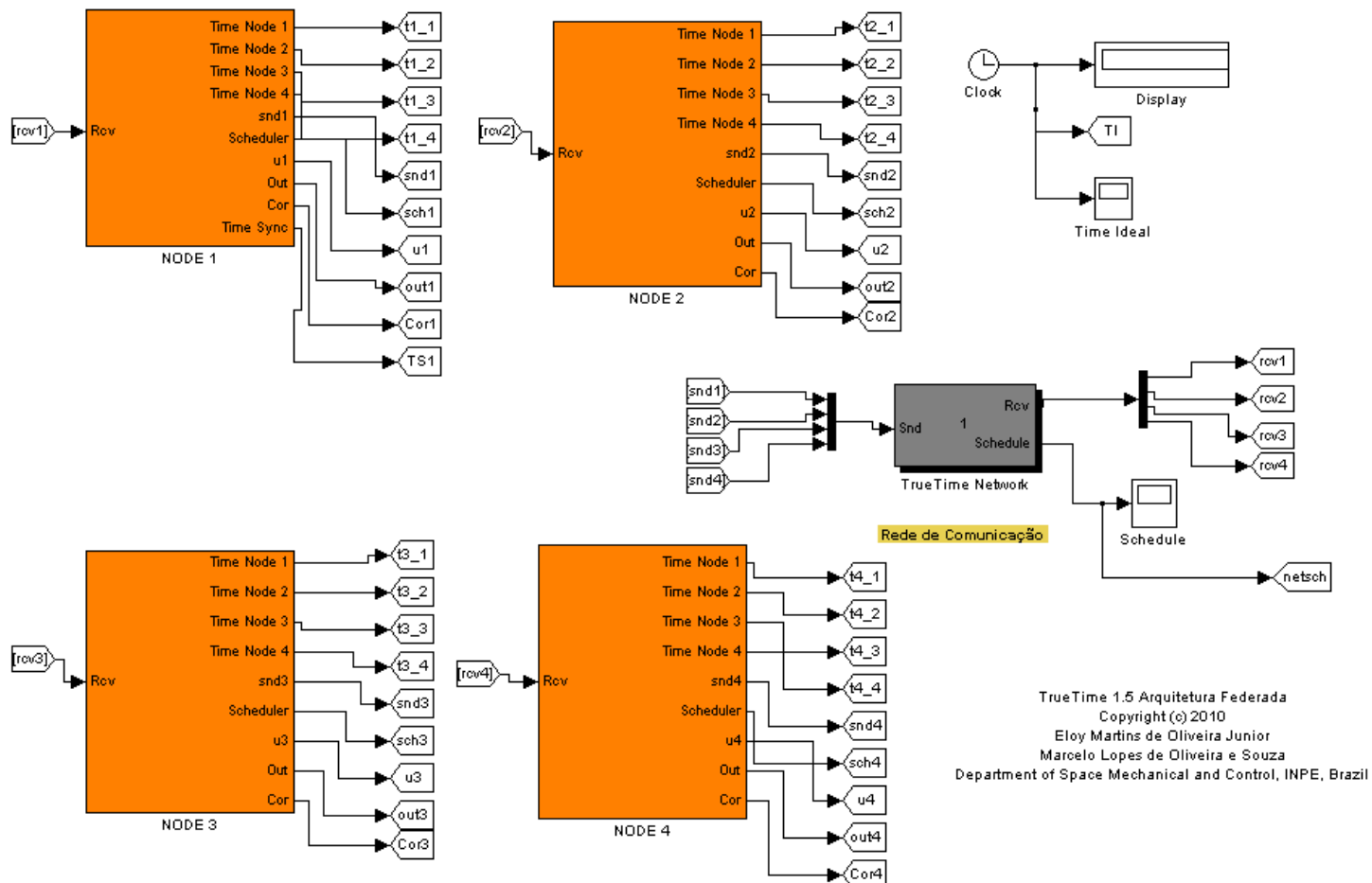


Figura 3.8 – Bloco TrueTime Kernel.

Cada nó (Federado), consiste em um modelo apresentado na Figura 3.7. Cada nó inclui dentro do Bloco Azul, um bloco do *TrueTime Kernel* da Figura 3.8. Cada bloco do *TrueTime Kernel* deste modelo chama uma função ***task\_init.m***, ***task\_init2.m***, ***task\_init3.m***, ***task\_init4.m***. O Apêndice A descreve um exemplo destas funções. Cada arquivo cria: 1) uma tarefa para o controlador, que calcula o sinal de controle periodicamente e o envia ao atuador; 2) uma tarefa de sincronismo, que faz a sincronização de relógios; 3) uma tarefa para envio dos dados de tempo pelo barramento de

comunicação via *broadcast*; 4) configura a interrupção causada pelos dados provenientes da rede de comunicação; 5) inicializa o núcleo ou (kernel) do TrueTime®.

A tarefa de envio de dados de tempo pelo barramento de comunicação e de sincronização são tarefas condicionadas. Neste caso, o nó 1 possui as duas tarefas periódicas. O nó envia o broadcast, quando o dado do nó 1 chega ao nó 2, o nó faz o seu broadcast, quando o dado do nó 2 chega ao nó 3, o nó 3 faz seu broadcast e quando o nó 4 recebe o dado de 3, ele faz o broadcast. Quando todos os nós recebem os dados de 4, ai sim a sincronização é autorizada. Foi feito esse loop lógico para que a simulação garanta que todos os nós do sistema tenham recebido os dados e assim minimizando o erro do algoritmo.



TrueTime 1.5 Arquitetura Federada  
 Copyright (c) 2010  
 Eloy Martins de Oliveira Junior  
 Marcelo Lopes de Oliveira e Souza  
 Department of Space Mechanical and Control, INPE, Brazil

Figura 3.9 – Modelo da Arquitetura Federada no TrueTime.



### 3.2.3. Variação da Rede de Comunicação

No TrueTime é possível trabalhar com diferentes tipos de protocolos de comunicação. Neste trabalho foram escolhidos dois protocolos de comunicação para análise de resultados. Os protocolos escolhidos são o Ethernet e um protocolo com acesso ao meio TDMA.

A Figura 3.10 mostra interface de configuração da rede de comunicação no TrueTime.

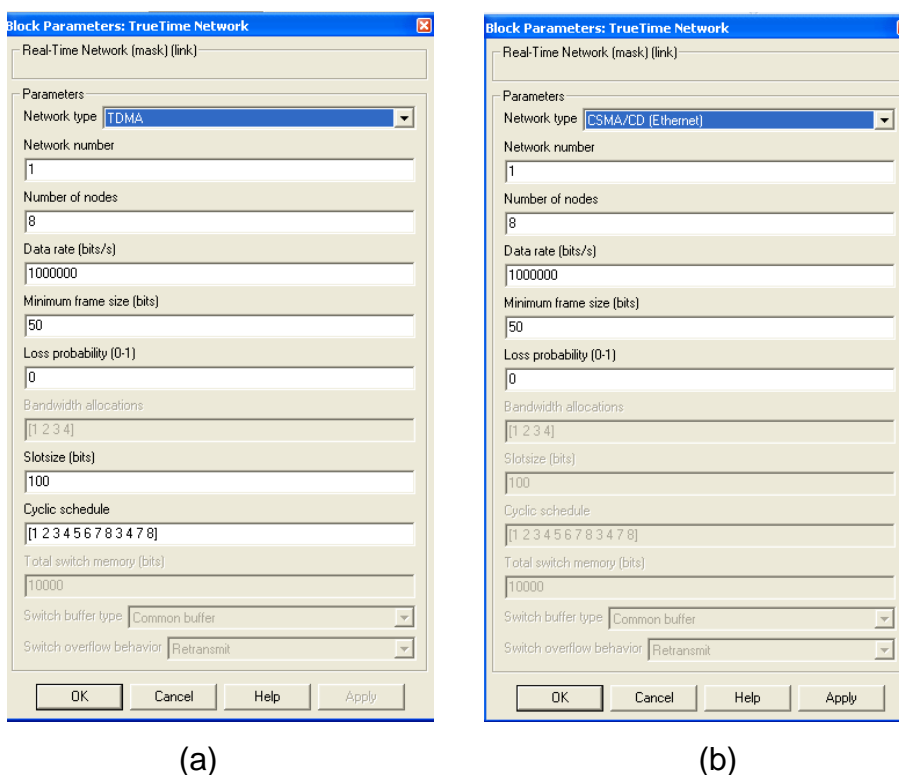


Figura 3.10 – Interface de Configuração da Rede de Comunicação: (a) Configuração de rede TDMA (b) Configuração de Rede Ethernet

### 3.2.4. Modelo da Sincronização de Relógios

No TrueTime é possível trabalhar com sincronização de relógios, pois além de implementar um computador de tempo real virtual pelo bloco TrueTime Kernel da figura 3.8, é implementado também em cada bloco um relógio virtual independente, o qual neste trabalho foi denominado Tempo Virtual do

*TrueTime Kernel*. O relógio do TrueTime 1.5, denominado Tempo Virtual, é o relógio que o bloco do *TrueTime Kernel* utiliza como relógio de referência, isto é, como base de tempo comum. O TrueTime 1.5 utiliza o Tempo de Simulação do Matlab/Simulink como sua referência.

A Figura 3.11 ajuda a entender os diversos tipos de relógios que estão sendo utilizados para chegar ao resultado desejado.

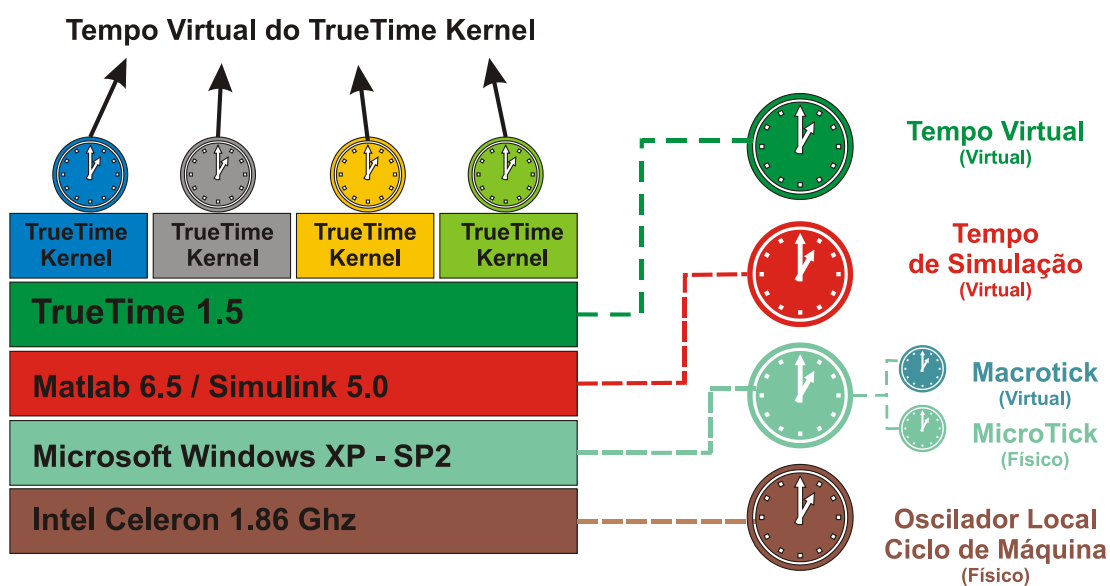


Figura 3.11 – Modelo de Simulação.

Com o bloco do TrueTime Kernel é possível variar os parâmetros de deriva (*Clock Drift*) e os parâmetros de viés inicial (*Clock Offset*) do relógio local (Tempo Virtual do *TrueTime Kernel*) através da interface de configuração, como pode ser observado na Figura 3.12. Os outros relógios da simulação, tais como o Tempo Virtual, o Tempo de Simulação, MacroTick, MicroTick e Ciclo de Máquina não foram modificados e tratados nesta simulação.

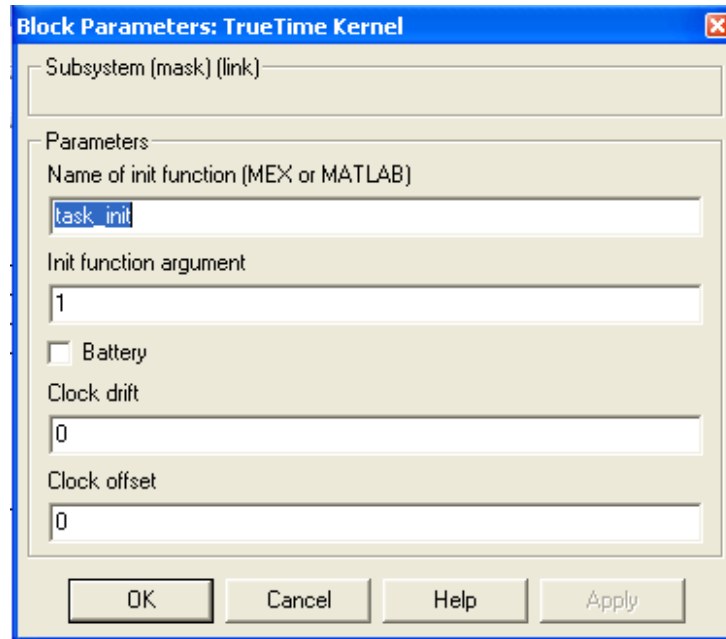


Figura 3.12 – Interface de configuração do bloco TrueTime Kernel.

Foi escolhido o algoritmo de sincronização de relógios FTM (*Welch-Lynch*) com arquitetura distribuída, vide Figura 2.26, e sincronização de relógios com Filtro de Kalman com arquitetura centralizada, vide Figura 2.27. O modelo de Filtro de Kalman e o modelo de FTM utilizados foram os mesmos modelos mostrados anteriormente nas seções 2.6 e 2.7 respectivamente.

Na simulação com sincronização de relógios com o uso do algoritmo FTM, todos os nós (Federados) trocam informações de relógios (Tempo Virtual do *TrueTime Kernel*), periodicamente, entre si pela rede de comunicação. Após o ciclo de troca de mensagens completar, a sincronização de relógios é feita pelo algoritmo FTM. A tarefa de sincronismo, de cada nó, chama o arquivo ***synccode.m***, ***synccode2.m***, ***synccode3.m*** e ***synccode4.m***. O Apêndice A descreve um exemplo da implementação destas funções. Nestes arquivos estão os códigos de implementação do algoritmo FTM para cada nó do sistema.

Na simulação com sincronização de relógios com o uso do Filtro de Kalman, o primeiro nó (Federado) envia periodicamente sua informação de relógio (Tempo Virtual do *TrueTime Kernel*) pela rede de comunicação via

broadcast. Todos os nós da rede recebem este dado e fazem a sincronização de relógios através do Filtro de Kalman. A tarefa de sincronismo chama o arquivo ***synccodeKalman.m***, ***synccodeKalman2.m***, ***synccodeKalman3.m*** e ***synccodeKalman4.m***. O Apêndice A descreve um exemplo da implementação destas funções. Nestes arquivos estão os códigos de implementação do algoritmo de Filtro de Kalman para cada nó.

### **3.2.5. Avaliação dos efeitos da sincronização**

Para visualizar os efeitos da variação do parâmetro tempo na simulação, modificou-se os valores de deriva (*Clock Drift*) e viés inicial (*Clock offset*) mostrado na Figura 3.12, de maneira a possibilitar avaliar a influência dos parâmetros de tempo no sistema de controle. Para várias simulações, observou-se as seguintes características do sistema de controle:

- Valor de pico;
- Instante de pico;
- % de sobre-elevação (*overshoot*);

### **3.3. Sincronização de Relógios em Sistemas de Controle sobre uma Arquitetura Integrada**

Para mostrar a influência da variação do tempo em sistemas de controle por rede e a eficiência dos algoritmos de sincronização de relógios, nesta seção foi escolhida uma arquitetura de controle integrada que diferentemente da arquitetura federada que possui seus componentes independentes, a arquitetura integrada, escolhida neste trabalho, possui todos os seus componentes, controles, atuadores e sensores interligados por um barramento de comunicação.

### **3.3.1. Metodologia**

A metodologia consiste em analisar um sistema de controle por rede de comunicação, isto é, todos os nós do sistema estão interligados por um barramento de comunicação. Na arquitetura federada foram utilizados 4 conjuntos de controle. Na arquitetura integrada utilizada por este trabalho são usados 2 conjuntos de controle, isto é, o sistema possui duas malhas de controle interligadas por um barramento de comunicação em comum. Todo o processo será controlado pelos blocos do TrueTime, que roda no MatLab/Simulink.

Os atuadores utilizados são os mesmo da Arquitetura Federada, dados pela equação 3.11. Os controladores implantados também foram os mesmos e são dados pelas equações 3.12 até a 3.17.

Os nós de Controle e os nós de Sensor possuem um relógio lógico, dado pelo computador e utilizam a rede de comunicação para trocar os dados de seus relógios. Os atuadores utilizam a rede de comunicação somente para receber os dados de controle.

Cada nó de controle além de implementar uma tarefa de controle periódica, também implementa uma tarefa de sincronização de relógios periódica. Cada nó de sensor além de implementar uma tarefa de envio dos dados medidos ao controlador, também implementa uma tarefa de sincronização de relógios periódica. Os atuadores são ativados por eventos a cada tarefa de controle que chega ao atuador pela rede de comunicação. Todos os nós possuem uma interrupção causada quando um dado é recebido pela rede de comunicação.

O objetivo é sincronizar os relógios lógicos dos nós de controle e sensores, utilizando-se de algoritmos de sincronização de relógios. Os nós de atuadores não são sincronizados, já que o efeito do tempo é descartado pois os atuadores são ativados por eventos.

Foi escolhido o algoritmos de sincronização de relógios FTM com arquitetura distribuída para realizar a simulação. Variou-se somente o protocolo de comunicação, fazendo um conjunto de simulações com TDMA e outro conjunto com o Ethernet.

### 3.3.2. Modelo do Sistema

O sistema distribuído integrado constituiu-se de 8 nós interligados a um barramento de comunicação. A Figura 3.13 mostra a arquitetura Integrada utilizada, onde os blocos em azul representam um conjunto de controle e os blocos em verde representam outro conjunto de controle. O bloco em laranja representa o barramento de comunicação. O nó de interferência é usado como forma de poder inserir uma interferência na rede de comunicação para se ter uma aproximação maior da realidade. Neste trabalho não utilizamos o parâmetro de interferência.

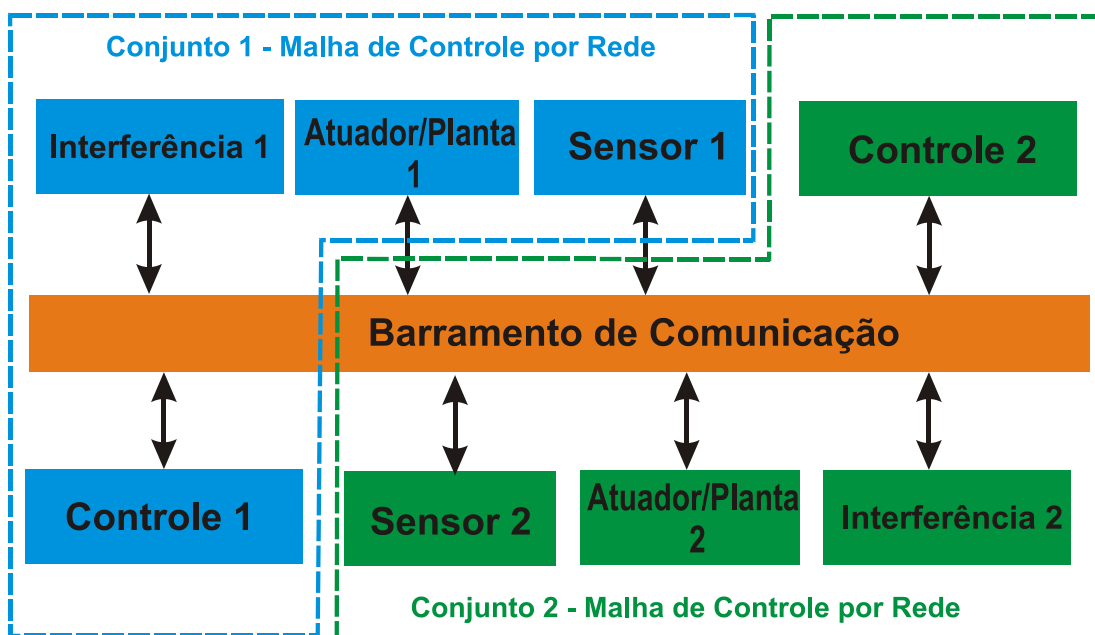


Figura 3.13 – Arquitetura Integrada.

Cada conjunto de controle por rede de comunicação implementa a seguinte malha de controle da Figura 3.14, com um simples controlador PID de um atuador eletro-hidráulico:

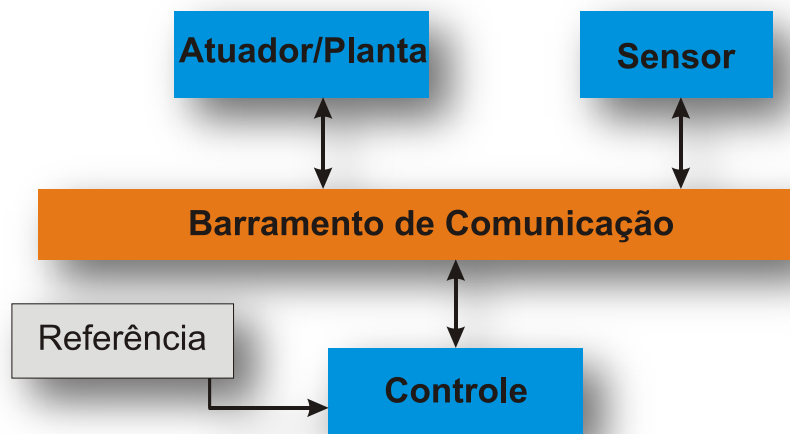


Figura 3.14 – Malha simples de controle por rede de comunicação do atuador eletro-hidráulico.

A Figura 3.15 mostra o modelo do controlador implementado com os blocos do TrueTime. O controle é realizado em um computador de tempo real simulado, dado pelo bloco do *TrueTime Kernel*. Neste bloco são agendadas 3 tarefas periódicas: 1) A tarefa do controle PID (das equações 3.12 até 3.17); 2) a tarefa de envio de dados do tempo na rede e; 3) a tarefa de sincronismo de relógio. Existe também agendada uma interrupção causada quando chega um dado de rede ao nó.

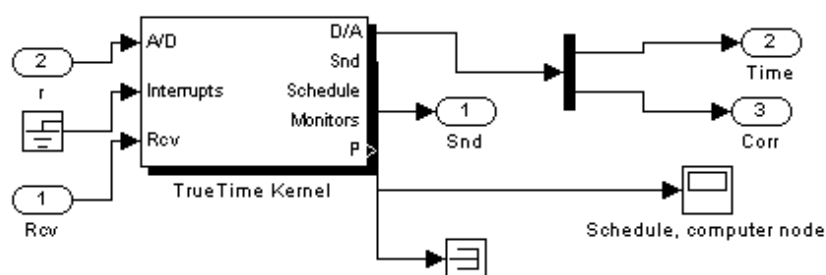


Figura 3.15 – Modelo do Controlador pelos blocos do TrueTime.

A Figura 3.16 mostra o modelo do atuador e sensor implementado com os blocos do TrueTime. O controle da atuação é realizado em um computador de tempo real simulado, dado pelo bloco do TrueTime Kernel. Neste bloco é

agendada 1 tarefa ativada por eventos. Todas as vezes que um dado de rede chega do controlador a atuação é realizada. A Figura 3.17 mostra o modelo do atuador no TrueTime.

O sensor é controlado em um computador de tempo real simulado, dado pelo bloco do *TrueTime Kernel*. Neste bloco são agendadas 3 tarefas periódicas a realizar. A tarefa de envio de dados medidos ao controlador, a tarefa de envio de dados do tempo na rede e a tarefa de sincronismo de relógio. Existe também agendada uma interrupção causada quando chega um dado de rede ao nó. A Figura 3.18 mostra o modelo do sensor no TrueTime.

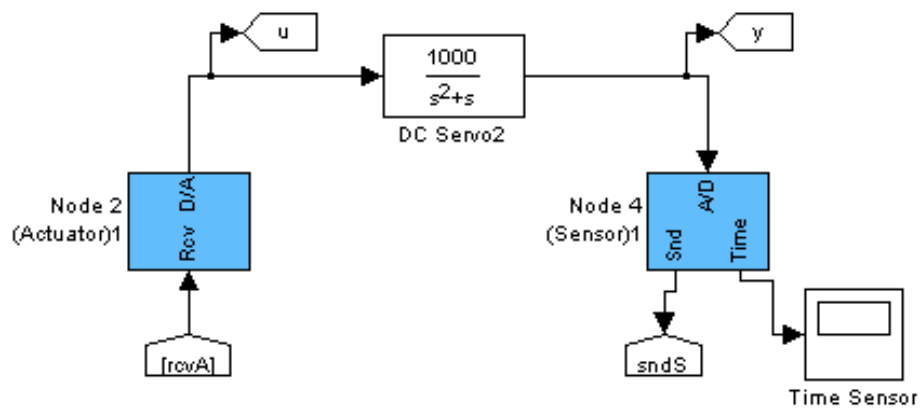


Figura 3.16 – Modelo do Atuador Eletro-Hidráulico no TrueTime.

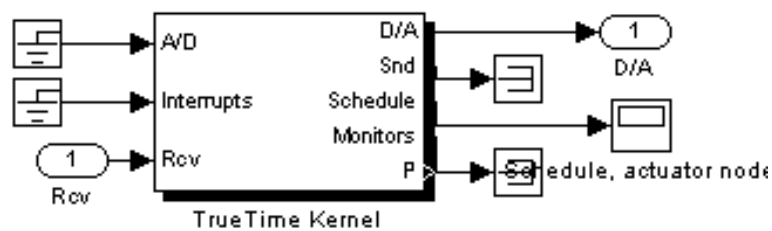


Figura 3.17 – Modelo do Atuador no TrueTime.



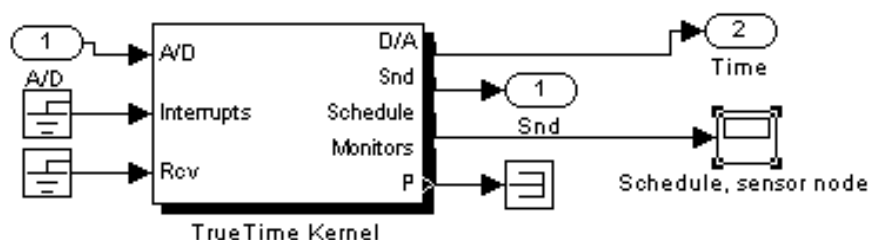


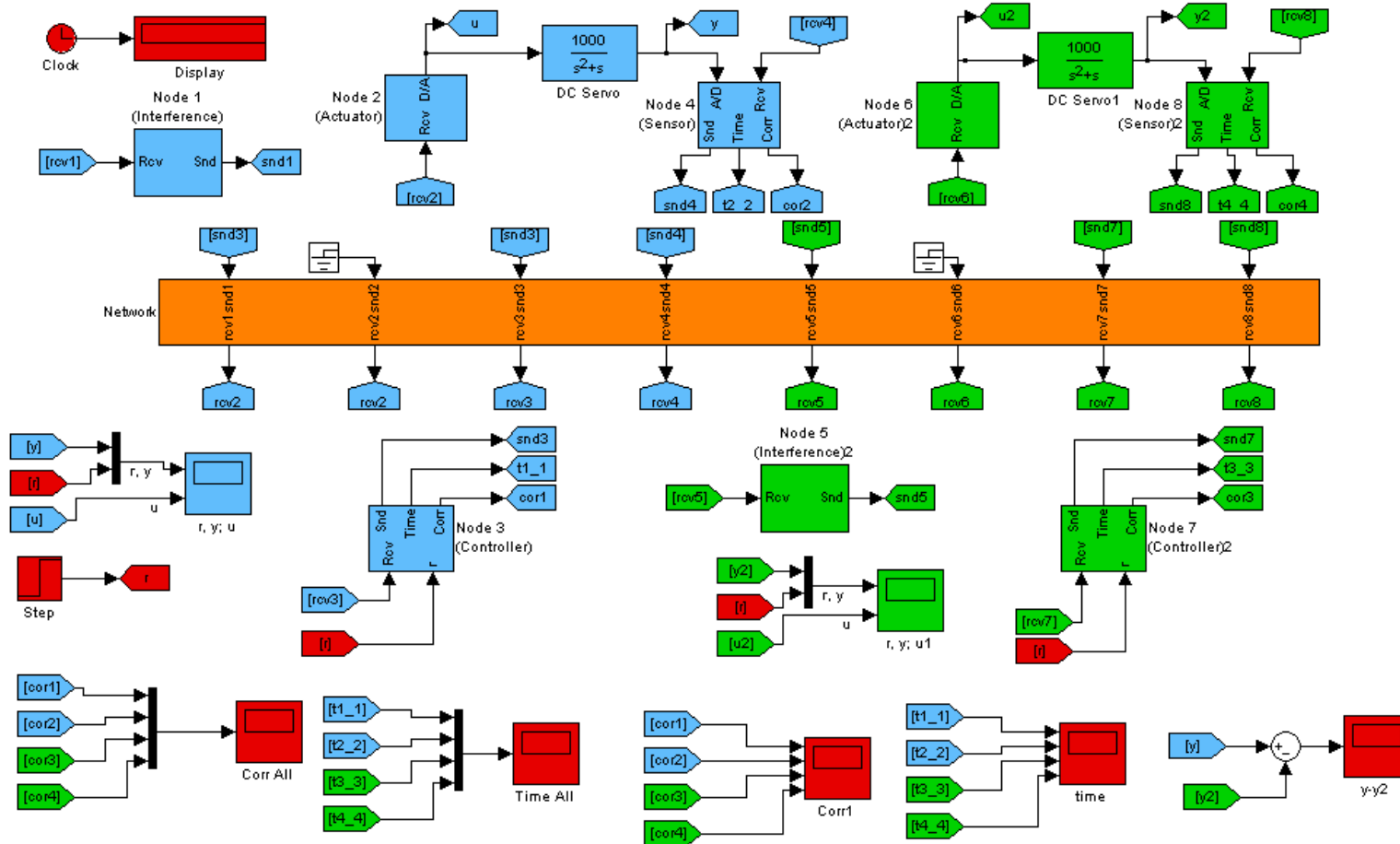
Figura 3.18 – Modelo do Sensor no TrueTime.

Cada nó do controlador no TrueTime, Figura 3.15, chama a função ***controller\_init.m***, ***controller\_init2.m***. O Apêndice B descreve um exemplo da implementação destas funções. Cada arquivo cria uma tarefa para o controlador, que calcula o sinal de controle periodicamente e o envia ao atuador; uma tarefa de sincronismo, que faz a sincronização de relógios; uma tarefa para envio dos dados de tempo pelo barramento de comunicação via *broadcast*; e configura a interrupção causada pelos dados provenientes da rede de comunicação, além de inicializar o núcleo ou (kernel) do TrueTime®.

Cada nó do sensor no TrueTime, Figura 3.18, chama a função ***sensor\_init.m***, ***sensor\_init2.m***. O Apêndice B descreve um exemplo da implementação destas funções. Cada arquivo cria uma tarefa para o envio periódico de dados ao controlador, que mede o sinal de saída do atuador; uma tarefa de sincronismo, que faz a sincronização de relógios; uma tarefa para envio dos dados de tempo pelo barramento de comunicação via *broadcast*; e configura a interrupção causada pelos dados provenientes da rede de comunicação, além de inicializar o núcleo ou (kernel) do TrueTime®.

Cada nó do atuador no TrueTime, Figura 3.18, chama a função ***actuator\_init.m***, ***actuator\_init2.m***. O Apêndice B descreve um exemplo da implementação destas funções. Cada arquivo cria uma tarefa não periódica de atuação na planta; e configura a interrupção causada pelos dados provenientes da rede de comunicação, além de inicializar o núcleo ou (kernel) do TrueTime®.

A tarefa de envio de dados de tempo pelo barramento de comunicação e de sincronização são tarefas condicionadas. Neste caso, o nó 1 possui as duas tarefas periódicas. O nó envia o broadcast, quando o dado do nó 1 chega ao nó 2, o nó 2 faz o seu broadcast, quando o dado do nó 2 chega ao nó 3, o nó 3 faz seu broadcast e quando o nó 4 recebe o dado de 3, ele faz o broadcast. Quando todos os nós recebem os dados de 4, aí sim a sincronização é autorizada. Foi feito esse loop lógico para que a simulação garanta que todos os nós do sistema tenham recebido os dados e assim minimizando o erro do algoritmo. A Figura 3.19 mostra o modelo de Arquitetura Integrada implementada no TrueTime®. Na figura 3.19, os blocos em azul representam a malha de controle 1, com sensor, interferência, atuador/planta e controlador. Os blocos em verde representam a malha de controle 2, com sensor, interferência, atuador/planta e controlador. Os blocos em vermelho são blocos auxiliares, para visualização gráfica e também os blocos que indicam a referência que no caso é um sinal degrau. O bloco em laranja representa o barramento de comunicação que é compartilhado pelas duas malhas de controle.



TrueTime 1.5 Arquitetura de Controle Integrada  
 Copyright (c) 2010  
 Eloy Martins de Oliveira Junior  
 Marcelo Lopes de Oliveira e Souza  
 INPE - Instituto Nacional de Pesquisas Espaciais, Brasil

Figura 3.19 – Modelo da Arquitetura Integrada no TrueTime.

### 3.3.3. Variação da Rede de Comunicação

Para esta simulação foram escolhidos dois protocolos de comunicação para análise de resultados. Os protocolos escolhidos são o Ethernet e um protocolo com acesso ao meio TDMA. A Figura 3.10 mostra a interface de configuração da rede de comunicação no TrueTime.

### 3.3.4. Modelo da Sincronização de Relógios

No TrueTime é possível trabalhar com sincronização de relógios, pois além de implementar um computador de tempo real virtual pelo bloco TrueTime Kernel da figura 3.8, é implementado também em cada bloco um relógio virtual independente, a qual neste trabalho foi denominado Tempo Virtual do *TrueTime Kernel*. O relógio do TrueTime 1.5, denominado Tempo Virtual, é o relógio que o bloco do *TrueTime Kernel* utiliza como relógio de referência, isto é, como base de tempo comum. O TrueTime 1.5 utiliza o Tempo de Simulação do Matlab/Simulink como sua referência.

O modelo desta de sincronização de relógios desta simulação com Arquitetura Integrada é igual ao modelo de sincronização da Arquitetura Federada dada pela Figura 3.11 que ajuda a entender os diversos tipos de relógios que estão sendo utilizados para chegar ao resultado e a simulação desejados.

Neste trabalho, somente utilizou-se dois relógios do modelo 3.11, o Tempo Virtual do *TrueTime Kernel* e o Tempo Virtual do TrueTime 1.5. O objetivo foi sincronizar os Tempos Virtuais do *TrueTime Kernel*. E o Tempo Virtual do TrueTime serviu como tempo de referência para a simulação.

Com o bloco do TrueTime Kernel é possível variar os parâmetros de deriva (*Clock Drift*) e os parâmetros de viés inicial (*Clock Offset*) do relógio local (Tempo Virtual do *TrueTime Kernel*) através da interface de configuração, como pode ser observado na Figura 3.12. Os outros relógios da simulação,

tais como o Tempo Virtual, o Tempo de Simulação, MacroTick, MicroTick e Ciclo de Máquina não foram modificados nesta simulação.

Foi escolhido o algoritmo de sincronização de relógios FTM (*Welch-Lynch*) com arquitetura distribuída, vide Figura 2.26. O modelo do algoritmo FTM utilizado foi o mesmo modelo mostrado anteriormente nas seção 2.6.

Na simulação com sincronização de relógios com o uso do algoritmo FTM, os nós do Controladores e dos Sensores trocam informações de relógios, periodicamente, entre si pela rede de comunicação. Após o ciclo de troca de mensagens completa, a sincronização de relógios é feita pelo algoritmo FTM. A tarefa de sincronismo, de cada nó, chama o arquivo ***synccode.m***, ***synccode2.m***, ***synccode3.m*** e ***synccode4.m***. O Apêndice B descreve um exemplo da implementação destas funções. Nestes arquivos estão os códigos de implementação do algoritmo FTM para cada nó do sistema.

### 3.3.5. Avaliação dos efeitos da sincronização

Para visualizar os efeitos da variação do parâmetro tempo na simulação, modificou-se os valores de deriva (*Clock Drift*) e viés inicial (*Clock offset*) mostrado na Figura 3.22, de maneira a possibilitar avaliar a influência dos parâmetros de tempo no sistema de controle. Para várias simulações, observou-se as seguintes características do sistema de controle:

- Valor de pico;
- Instante de pico;
- % de sobre-elevação (*overshoot*);



## 4 RESULTADOS E SIMULAÇÕES

### 4.1. Análise e Simulação dos Algoritmos de Sincronização de Relógios FTM (*Fault-Tolerant Mid-Point*) e FTA (*Fault-Tolerant Average*)

Os algoritmos FTM e FTA são muito parecidos, com exceção de suas funções de convergência, conforme dito anteriormente. Por isto, neste trabalho, foi feita uma simulação, utilizando-se do programa Microsoft Excel, para mostrar que os algoritmos provêm resultados diferentes na convergência e precisão final alcançada. Através destas simulações é possível mostrar que os algoritmos são diferentes em alguns casos e iguais em outros, além de prover um maior conhecimento do processo de sincronização de relógios.

Nesta simulação, através da comparação do 1º e 2º casos é possível verificar a influência que o atraso da rede de comunicação proporciona ao resultado da simulação. É possível também ver a robustez do algoritmo na presença de falhas maliciosas, tais com erro na taxa de deriva (*drift rate*) e erro no estado do relógio (*state rate*).

#### 4.1.1. Projeto dos Experimentos

O Projeto dos Experimentos realizados está sumarizado na Tabela 4.1 abaixo:

Tabela 4.1 – Tabela de Casos do Experimento dos Algoritmos FTA e FTA.

	Num. de Relógios (n)	Num. de Relógios com Falha (f)	Atraso de Rede (segundos)	Resultado
<b>1º Caso</b>	4	Relógio 1 com erro em deriva	0.005	FTM = FTA
<b>2º Caso</b>	7	Relógio 1 e 2 com erro de deriva	0.005	FTM ≠ FTA
<b>Outros Casos</b>	Apêndice P			

#### 4.1.2. Caso 1 – 4 Relógios, um relógio com erro de deriva

O objetivo deste caso é mostrar que os algoritmos FTM e FTA provêm o mesmo resultado. Isto é possível verificar matematicamente através das equações (3.6) e (3.7) que ambos os algoritmos FTM e FTA possuem o mesmo resultado. Com 4 relógios no conjunto suas funções de convergência são iguais. Ambos os algoritmos com isso alcançam o mesmo resultado final.

Dados do Experimento:

- Número de Relógios:  $n = 4$ ;
- Número de erros bizantinos:  $f = 1$ ;
- Taxa de deriva:  $\rho = 0.2$ ;
- Atraso de Rede:  $\delta = 0.005$  (s);

Neste 1º caso, foi aplicado ao relógio T1 um erro de taxa de deriva de 20% em relação aos outros relógios. O relógio T1 é o relógio com falha de deriva. Os relógios T2, T3 e T4 não estão com falha. Em azul mais escuro tem-se o relógio de referência.

A Tabela 4.2 mostra uma parte dos dados do experimento, realizado com o auxílio do Microsoft Excel. A Tabela completa está no Apêndice C, Tabela C.1.

As colunas T1, T2, T3 e T4 são as colunas dos valores de tempo dos relógios. As colunas Delta1, Delta2, Delta3 e Delta4 são as colunas onde a diferença dos relógios é realizada, seguindo a equação (3.2).

O ciclo é identificado pela coluna *ciclo*. A cada instante um relógio envia seus dados, como em uma comunicação TDMA.



A coluna maior delta e menor delta descarta o maior e o menor valor da linha de deltas e pega o maior e o menor delta restante de cada ciclo e faz o cálculo da média do maior e menor delta restante, no caso a cada 4 linhas. A coluna média faz a correção, isto é faz o cálculo da função de convergência que é dado neste caso pela equação 3.6. Após cada ciclo completo a correção é computada, sendo compensada pela constante que indica o atraso de rede. Após o ciclo completo, é adicionado o valor de correção ao relógio, atualizando assim as colunas T1, T2, T3 e T4 com seus respectivos valores. A coluna referência é o relógio de referência da simulação.

A Figura 4.2 mostra o graficamente o resultado da simulação do algoritmo de sincronização de relógios, isto é, com aplicação das equações 3.8 e 3.9. É possível ver que o relógio T1 (Linha Roxa) se afasta dos outros relógios que não estão com falha. Ao aplicar o algoritmo é possível ver uma forma de onda dente de serra (linha vermelha) se formando. Isto ocorre, pois o algoritmo de sincronização é aplicado periodicamente, aplicando a correção dos tempos. Com isto os relógios T1, T2, T3 e T4 ficam dentro de uma determinada precisão. A Figura 4.2 mostra o resultado da função de correção, isto é, mostra o gráfico de correção que o algoritmo aplica aos relógios. Neste gráfico da figura 4.2 é possível ver que o algoritmo FTA/FTM convergiu a um valor de sua correção. Este valor de correção indica que a cada ciclo, o relógio T1 precisa sofrer uma correção em torno de 0.4 (s), e os relógios T2, T3 e T4, em torno de -0,005 (s).

Análise dos Dados:

- Máximo viés instantâneo, de acordo com a equação (2.33):  $\Gamma = 0.8$  (s);
- Máximo viés instantâneo na simulação:  $\Gamma = 0.55$  (s);
- Precisão, de acordo com a equação 2.48:  $\Pi = 0.8$  (s);
- Precisão alcançada pela simulação:  $\Pi = 0.55$  (s);

Tabela 4.2 – Tabela de Dados do Experimento FTM/FTA – 1º Caso.

T1	T2	T3	T4	Delta 1	Delta 2	Delta 3	Delta 4	Menor Delta	Maior Delta	Média	Correção	Ciclo	Referência	T1 Sem Correção
0	0	0	0	0	0	0	0	0,1	0,2	0,15	0,145	T1	0	0
0,6	0,5	0,5	0,5	0,1	0	0	0	0	0	0	-0,005	T2	0,5	0,6
1,2	1	1	1	0,2	0	0	0	0	0	0	-0,005	T3	1	1,2
1,8	1,5	1,5	1,5	0,3	0	0	0	0	0	0	-0,005	T4	1,5	1,8
2,255	2,005	2,005	2,005	0	-0,25	-0,25	-0,25	0,35	0,45	0,4	0,395	T1	2	2,4
2,855	2,505	2,505	2,505	0,35	0	0	0	0	0	0	-0,005	T2	2,5	3
3,455	3,005	3,005	3,005	0,45	0	0	0	0	0	0	-0,005	T3	3	3,6
4,055	3,505	3,505	3,505	0,55	0	0	0	0	0	0	-0,005	T4	3,5	4,2
4,26	4,01	4,01	4,01	0	-0,25	-0,25	-0,25	0,35	0,45	0,4	0,395	T1	4	4,8
4,86	4,51	4,51	4,51	0,35	0	0	0	0	0	0	-0,005	T2	4,5	5,4
5,46	5,01	5,01	5,01	0,45	0	0	0	0	0	0	-0,005	T3	5	6
6,06	5,51	5,51	5,51	0,55	0	0	0	0	0	0	-0,005	T4	5,5	6,6
6,265	6,015	6,015	6,015	0	-0,25	-0,25	-0,25	0,35	0,45	0,4	0,395	T1	6	7,2
6,865	6,515	6,515	6,515	0,35	0	0	0	0	0	0	-0,005	T2	6,5	7,8
7,465	7,015	7,015	7,015	0,45	0	0	0	0	0	0	-0,005	T3	7	8,4
8,065	7,515	7,515	7,515	0,55	0	0	0	0	0	0	-0,005	T4	7,5	9
8,27	8,02	8,02	8,02	0	-0,25	-0,25	-0,25	0,35	0,45	0,4	0,395	T1	8	9,6
8,87	8,52	8,52	8,52	0,35	0	0	0	0	0	0	-0,005	T2	8,5	10,2

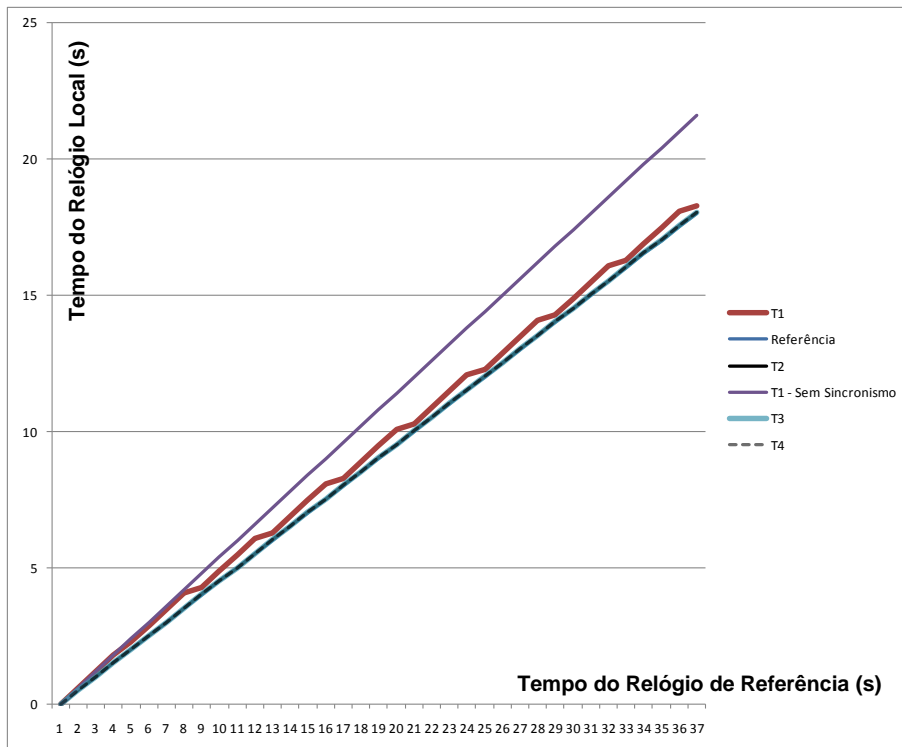


Figura 4.1 – FTM/FTA – 1º Caso Com Sincronismo.

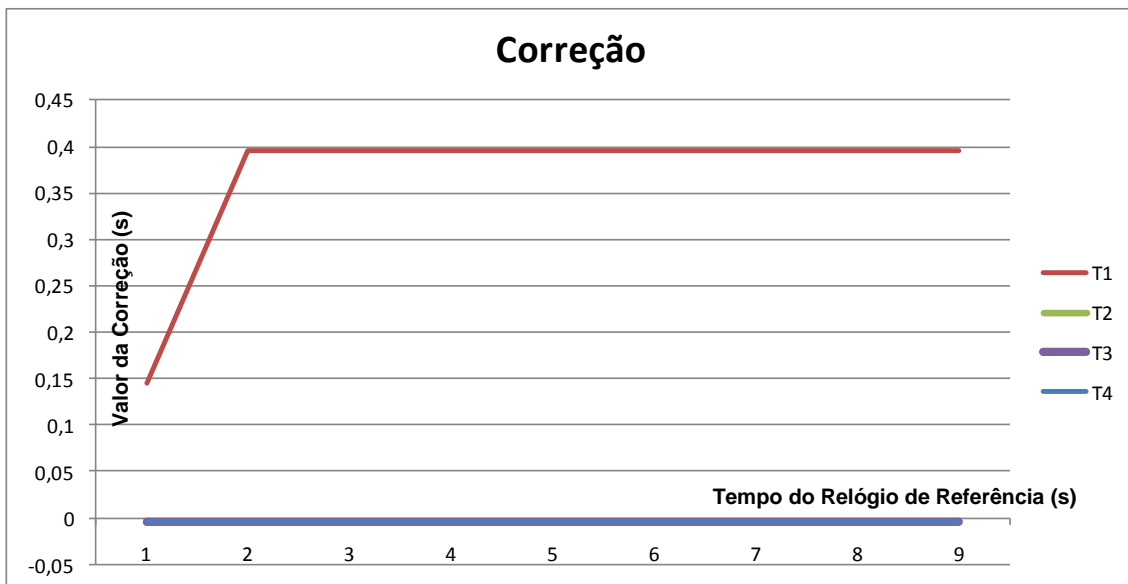


Figura 4.2 – FTM/ FTA – Correção do 1º Caso.

#### 4.1.3. Caso 2 – 7 Relógios - Erro de deriva

Neste caso é possível verificar através das equações (3.6) e (3.7) que os algoritmos FTM e FTA são diferentes, pois suas funções de convergência são diferentes. O resultado esperado é que os algoritmos apresentem resultado final diferente.

Dados do Experimento:

- Número de Relógios:  $n = 7$ ;
- Número de erros bizantinos:  $f = 2$ ;
- Taxa de deriva:  $\rho = 0.1$  e  $\rho = 0.2$ ;
- Atraso de Rede:  $\delta = 0.005$  (s);

Neste caso, foi aplicado ao relógio T1 um erro de taxa de deriva de 20% e ao relógio T2 um erro de taxa de deriva de 10% em relação aos outros relógios. Os relógios T1 e T2 são os relógios com falha de deriva. Os relógios T3, T4, T5, T6 e T7 não estão com falha.

A Tabela C.6 mostra os dados do experimento utilizando o algoritmo FTM e a Tabela C.7 mostra os dados do experimento utilizando o algoritmo FTA, realizado com o auxílio do Microsoft Excel. As colunas T1, T2, T3, T4, T5, T6 e T7 são as colunas de propagação dos relógios. As colunas Delta1, Delta2, Delta3, Delta4, Delta5, Delta6 e Delta7 são as colunas aonde a diferença dos relógios é realizada, de acordo com a equação 3.2. O ciclo é identificado pela coluna *ciclo*. A cada instante, um dos relógios do conjunto envia seus dados, como em uma comunicação TDMA.

No Algoritmo FTM, a coluna maior delta e menor delta descarta o maior e o menor valor da linha de deltas e pega o maior e o menor delta restante de cada

ciclo e calcula a média do maior e menor delta, no caso, a cada 7 linhas. No Algoritmo FTA, a coluna maior delta e menor delta, pega o maior e o menor delta de cada ciclo, e na hora de calcular a média o algoritmo retira os maiores valores e faz a média, no caso, a cada 7 linhas.

A coluna média, da Tabela C.6, faz a correção do algoritmo FTM, isto é faz o cálculo da função de convergência que é dado pela equação (3.6). A coluna média, da Tabela C.7, faz a correção do Algoritmo FTA, isto é faz o cálculo da função de convergência que é dado pela equação 3.7. Após cada ciclo completo a correção é computada, sendo compensada pela constante que indica o atraso de rede. Após o ciclo completo, é adicionado o valor de correção ao relógio, atualizando assim as colunas T1, T2, T3, T4, T5, T6 e T7 com seus respectivos valores. A coluna referência é o relógio de referência da simulação.

A Figura 4.3 mostra graficamente o resultado da simulação com aplicação do algoritmo FTM. A Figura 4.4 mostra o resultado da simulação aplicando o algoritmo FTA. Nas Figuras 4.3 e 4.4 é possível ver uma forma de onda dente de serra se formando. Isto ocorre, pois o algoritmo de sincronização é aplicado periodicamente, aplicando a correção dos tempos. Com isto os relógios T1, T2, T3, T4, T5, T6 e T7 ficam dentro de uma determinada precisão. Outro ponto é que os resultados das Figuras 4.3 e 4.4 não são iguais, apesar de serem muito parecidos, eles são diferentes. Isto ocorre, pois as equações 3.6 e 3.7 não são iguais e não possuem o mesmo resultado como nos casos anteriores.

- **Exemplo:**

Pegando o ultimo ciclo da Tabela C.6 e Tabela C.7, vamos fazer o calculo das médias. Será feito somente os cálculos com os vetores

delta1 e delta2, já que os vetores delta3, delta4, delta5, delta6 e delta7 são idênticos ao delta2.

**Tabela C.6:**

$$\Delta 1 = [-0.275; 0; 0.325; 0.375; 0.425; 0.475; 0,525]$$

$$\Delta 2 = [0; 0.225; 0.65; 0.75; 0.85; 0.95; 1.05]$$

**Caso com FTM:** Cálculo pela equação 3.6, eliminado o maior e menor delta, e pegando o maior e menor delta restante. Tem-se:

$$m\u00e9dia\Delta 1 = \frac{0 + 0.475}{2} = 0.2375 (s)$$

$$m\u00e9dia\Delta 2 = \frac{0.225 + 0.95}{2} = 0.5875 (s)$$

**Tabela C.7:**

$$\Delta 1 = [-0.26; 0; 0.31; 0.36; 0.41; 0.46; 0,51]$$

$$\Delta 2 = [0; 0.21; 0.62; 0.72; 0.82; 0.92; 1.02]$$

**Caso com FTA:** Cálculo pela equação 3.7, eliminado o maior e menor delta, e fazendo a média aritmética dos elementos restantes. Tem-se:

$$m\u00e9dia\Delta 1 = \frac{0 + 0.31 + 0.36 + 0.41 + 0.46}{5} = 0.308 (s)$$

$$m\u00e9dia\Delta 2 = \frac{0.21 + 0.62 + 0.72 + 0.82 + 0.92}{5} = 0.658 (s)$$

Conforme é possível ver no exemplo acima, além dos algoritmos FTM e FTA serem diferentes, eles possuem deltas e médias diferentes entre si. Conseqüentemente, as Figura 4.5 e 4.6, que mostram os resultados das funções de correção dos algoritmos FTM e FTA, respectivamente, são diferentes. Nestes gráficos das figuras 4.5 e 4.6 é possível ver que os algoritmos FTA e FTM convergiram a um valor e são diferentes. O Algoritmo FTA computa uma correção maior que o algoritmo FTM. Isso indica que o valor de correção do FTM a cada ciclo do relógio T1 precisa sofrer uma correção em torno de 0.2325 (s), o de T2 em torno de 0.5825 (s) e os relógios T3, T4, T5, T6 e T7, em torno de -0.1175. O valor de correção do FTA a cada ciclo do relógio T1 precisa sofrer uma correção em torno de 0.303, de T2 em torno de 0.653 e os relógios T3, T4, T5, T6 e T7, em torno de -0.047.

Análise dos Dados:

- **Algoritmo FTM:**

- Máximo viés instantâneo, de acordo com a eq. (2.33):  $\Gamma = 3.7$  (s);
- Máximo viés instantâneo na simulação:  $\Gamma = 1.05$  (s);
- Precisão, de acordo com a equação (2.45), multiplicando pelo termo do erro bizantino ( $\mu$ ) da equação 2.46:  $\Pi = 5.55$  (s);
- Precisão alcançada pela simulação:  $\Pi = 1.05$  (s);

- **Algoritmo FTA:**

- Máximo viés instantâneo, de acordo com a eq. (2.33):  $\Gamma = 3.7$  (s);
- Máximo viés instantâneo na simulação:  $\Gamma = 1.02$  (s);
- Precisão, de acordo com a equação 2.50:  $\Pi = 7.4$  (s);
- Precisão alcançada pela simulação:  $\Pi = 1.02$  (s);

No algoritmo FTM seu resultado é satisfatório já que a precisão ficou bem abaixo da pré-estabelecida pelas equações. No algoritmo FTA, a precisão também ficou bem abaixo da pré-estabelecida pelas equações. Apesar de

ambos os algoritmos obterem resultados coerentes com a parte teórica, a precisão final alcançada foi bem superior aos outros casos testados no apêndice. Isto ocorre por vários fatores, entre eles o intervalo de re-sincronização é muito maior que os outros casos; a deriva aplicada é muito grande; e isso prejudica a precisão final alcançada de ambos os algoritmos.

Outros casos são simulados no Apêndice P.

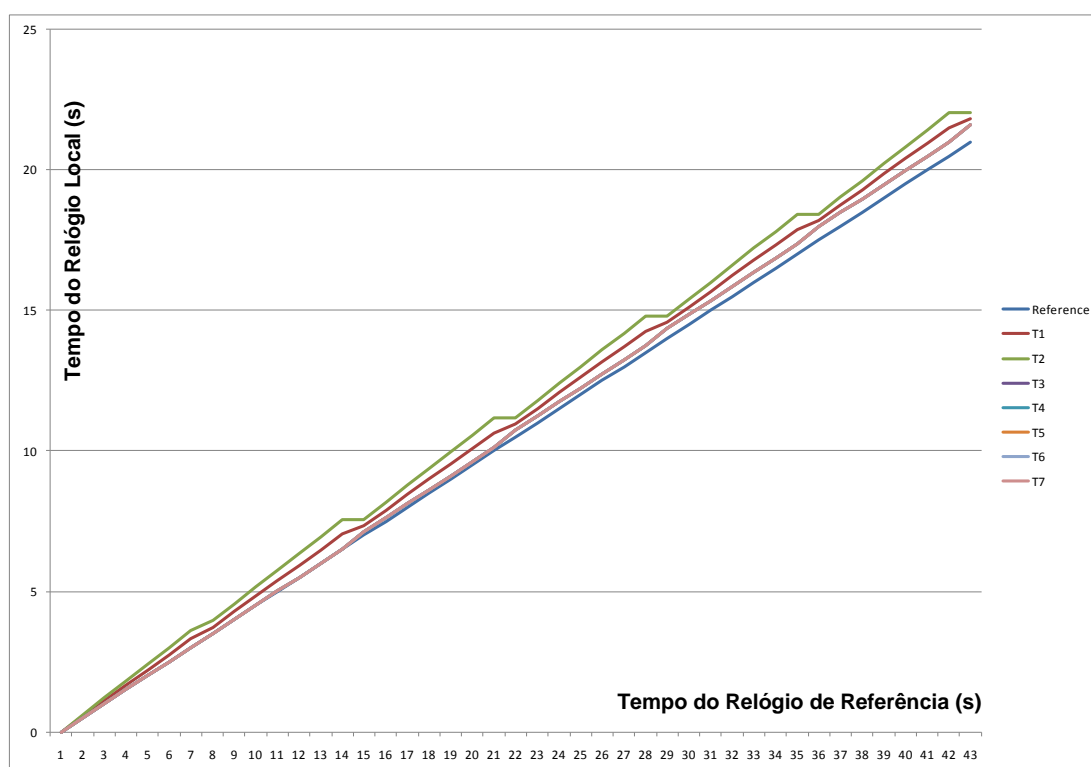


Figura 4.3 – FTM/FTA – 2º Caso Com Sincronismo pelo Algoritmo FTM.



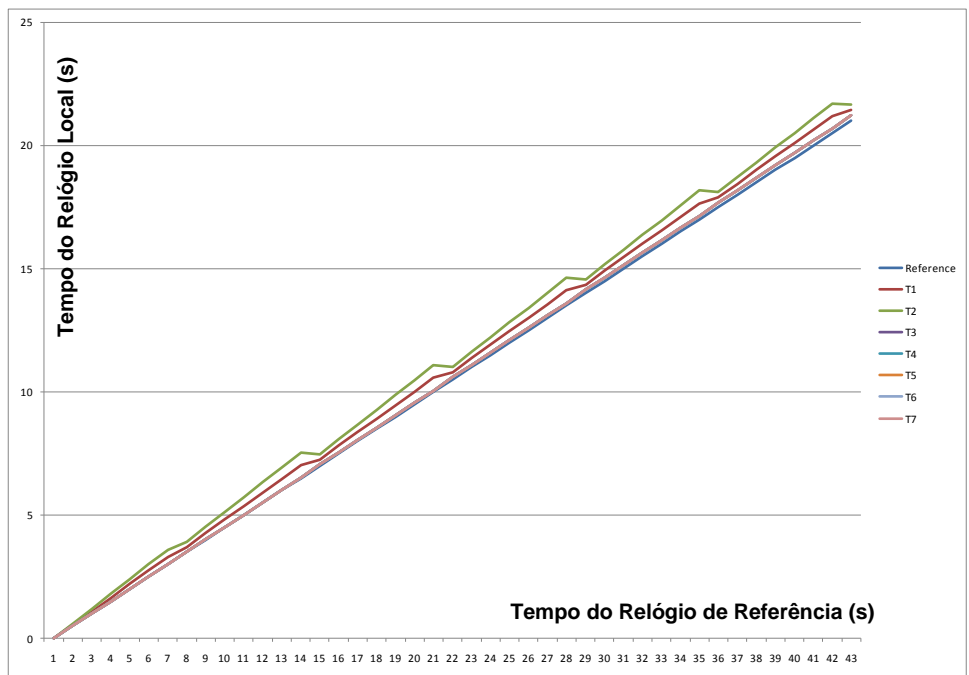


Figura 4.4 – FTM/FTA – 2º Caso Com Sincronismo pelo Algoritmo FTA.

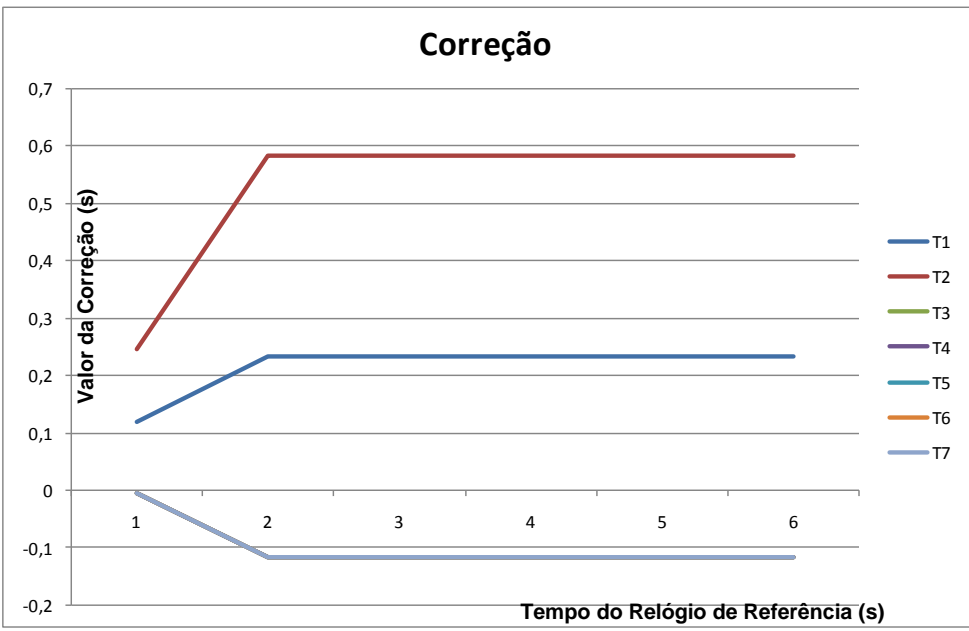


Figura 4.5 – FTM/FTA – Correção do 2º Caso pelo Algoritmo FTM.

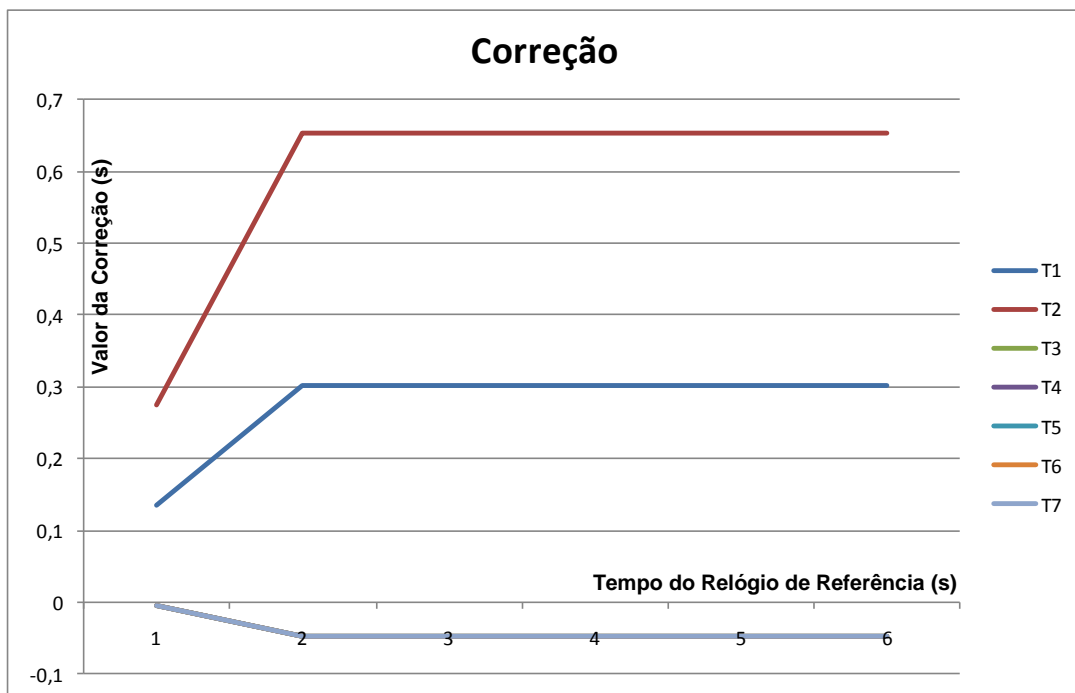


Figura 4.6 – FTM/FTA – Correção do 2º Caso pelo Algoritmo FTA.

## 4.2. Sincronização de Relógios em Sistemas de Controle sobre uma Arquitetura Federada

Neste capítulo é feita a simulação do sistema de controle distribuído com uma arquitetura Federada. Foi feita uma simulação com 4 subsistemas de controle. Os nós (Federados) estão conectados a um barramento de comunicação feita por um protocolo de comunicação com acesso ao meio TDMA. Em todos os casos da Arquitetura Federada, foi feita uma simulação em um sistema de controle com arquitetura federada, descrito na seção 3.2.

O objetivo deste conjunto de simulações é: 1) mostrar a influência que a dessincronização de relógios causa ao sistema de controle com arquitetura federada; 2) Utilizar dois métodos de sincronização de relógios, um com o algoritmo FTM e outro com Filtro de Kalman, comparar seus resultados e mostrar a influência no sistema de controle com arquitetura federada;

Esta simulação foi realizada com o Matlab/Simulink com o apoio do *TrueTime ToolBox*. Os quatro subsistemas de controle digitais foram implementados, isto é, a lei de controle PID foi implementada em um computador de tempo real simulado pelo *TrueTime ToolBox*.

Para **todos** os **casos** simulados, o **período de re-sincronização é de 0.0040 segundos**, isto é, a cada 0.0040 segundos o algoritmo Welch-Lynch (FTM) ou Filtro de Kalman (FK) atualiza os valores de relógio com a correção calculada.

### 4.2.1. Projeto dos Experimentos

O Projeto dos Experimentos realizados está sumarizado na Tabela 4.3 abaixo:

Tabela 4.3 – Tabela de Casos do Experimento com Arquitetura Federada.

	Num. de Nós (n)	Num. de Relógios com Falha (f)	Algoritmo de Sincronização	Rede
<b>1º Caso (C)</b>	4	Relógio 4 com erro em deriva	FTM (Welch-Lynch)	TDMA/Ethernet
<b>2º Caso (A)</b>	4	Relógio 4 com erro em viés inicial	FTM (Welch-Lynch)	TDMA/Ethernet
<b>3º Caso (B)</b>	4	Relógio 4 com erro de viés Inicial e Deriva	FTM (Welch-Lynch)	TDMA/Ethernet
<b>4º Caso (C)</b>	4	Relógio 4 com erro em deriva	Filtro de Kalman	TDMA/Ethernet
<b>5º Caso (A)</b>	4	Relógio 4 com erro em viés inicial	Filtro de Kalman	TDMA/Ethernet
<b>6º Caso (B)</b>	4	Relógio 4 com erro de viés Inicial e Deriva	Filtro de Kalman	TDMA/Ethernet
<b>Outros Casos</b>	Apêndices <b>D, E, F, G, H e I</b>			

A tabela 4.4 mostra os resultados dos dados simulados para cada caso. Na seqüência serão mostrados os 6 casos simulados, indicados na Tabela 4.3, seus gráficos e uma análise sobre os dados obtidos. Outros casos são mostrados no apêndice.

Tabela 4.4 – Tabela de Resultados da Arquitetura Federada.

	RF	AS	TD (%)	VI (s)	VP	O (%)	IP(s)	VP	O (%)	IP (s)	VP	O (%)	IP (s)
					Nó 1 - Sincronizado			Nó 4 - Sem Sincronização			Nó 4 - Com Sincronização		
1º Caso (A)	4	FTM	-10%	0	1.0448	4.4764%	0.3731	1.0496	4.9571%	0.3873	1.0446	4.4619%	0.3754
1º Caso (B)	4	FTM	1%	0	1.0465	4.6450%	0.3241	1.0461	4.6128%	0.3247	1.0464	4.6369%	0.3242
1º Caso (C)	4	FTM	10%	0	1.0472	4.7197%	0.3216	1.0505	5.0475%	0.3108	1.0473	4.7259%	0.3214
2º Caso (A)	4	FTM	0	-0.2	1.0459	4.5905%	0.3264	1.0451	4.5080%	0.4255	1.0458	4.5808%	0.3264
2º Caso (B)	4	FTM	0	0.39	1.0454	4.5355%	0.3288	1.0698	6.9792%	0.3009	37.7517	3x10 <sup>3</sup> %	0.4962
3º Caso (A)	4	FTM	-10%	-0.2	1.0448	4.4787%	0.3734	1.0495	4.9495%	0.5028	1.0446	4.4619%	0.3758
3º Caso (B)	4	FTM	10%	0.39	1.0471	4.7133%	0.3216	1.0950	9.4979%	0.2784	Este caso não convergiu		
4º Caso (A)	4	FK	-10%	0	1.0451	4.5057%	0.3318	1.0496	4.9571%	0.3873	1.0452	4.5159%	0.3283
4º Caso (B)	4	FK	1%	0	1.0451	4.5057%	0.3317	1.0461	4.6128%	0.3247	1.0453	4.5314%	0.3343
4º Caso (C)	4	FK	10%	0	1.0451	4.5057%	0.3318	1.0505	5.0475%	0.3108	1.0452	4.5204%	0.3296
5º Caso (A)	4	FK	0	-0.2	1.0451	4.5057%	0.3318	1.0451	4.5080%	0.4255	1.0452	4.5207%	0.3318
5º Caso (B)	4	FK	0	0.39	1.0451	4.5057%	0.3317	1.0698	6.9792%	0.3009	2.6838	168.37%	0.4905
6º Caso (A)	4	FK	-10%	-0.2	1.0451	4.5057%	0.3318	1.0495	4.9495%	0.5028	1.0440	4.3970%	0.3384
6º Caso (B)	4	FK	10%	0.39	1.0451	4.5057%	0.3317	1.0950	9.4979%	0.2784	2.6838	168.37%	0.4905

#### 4.2.2. Caso 1 – FTM - Relógio com Deriva

Nestes casos a sincronização foi feita através do algoritmo FTM (*Welch-Lynch*), com **arquitetura distribuída**, isto é, não existe um relógio de referência.. Nesta simulação variou-se o **parâmetro de deriva** do relógio do nó (subsistema) 4. O objetivo deste caso é: 1) mostrar a influência que a deriva de relógio tem sobre o sistema de controle com AF; 2) utilizar o algoritmo FTM para corrigir esta imperfeição do relógio e analisar o sistema de controle distribuído.

Este trabalho escolheu 3 taxas de deriva diferentes para variação do relógio:

- **Caso A: -10%** - O relógio está se atrasando em relação aos outros relógios do conjunto com taxa de 10%;
- **Caso B: 1%** - O relógio está se adiantando em relação aos outros relógios do conjunto com taxa de 1%;
- **Caso C: 10%** - O relógio está se adiantando em relação aos outros relógios do conjunto com taxa de 10%;

Os **casos A e B** são apresentados no **Apêndice D**. No **caso C**, a uma **taxa de deriva de 10%**, o relógio do nó 4 está se adiantando em relação ao relógio ideal e aos outros relógios dos outros nós, como mostra a Figura 4.7. Com isso, o sistema sofre uma pequena variação, por causa do relógio, no sistema de controle do nó 4 que está afetado pelo relógio. A Figura 4.8 mostra a resposta ao degrau do sistema de controle com a deriva de 10%, onde é possível ver que somente o nó 4 sofre variação em relação ao ideal e aos outros nós do sistema. A Figura 4.9 mostra as leis de controle aplicadas a cada nó do sistema e do sistema ideal. Com o impacto da taxa de deriva de 10%, aplicamos o algoritmo FTM de sincronização de relógios para corrigir a deriva. A Figura 4.11 mostra a linha do tempo com o algoritmo de sincronização FTM aplicado ao sistema. Na Figura 4.10 é possível ver o gráfico de correção dos

relógios, calculado pelo algoritmo, dos respectivos nós do sistema, além de mostrar que o algoritmo FTM convergiu rapidamente. Os nós 1, 2 e 3 não estão com erro de deriva, mas o nó 1 fez uma correção periódica em torno de 0.1 (ms) e os nós 2 e 3 fizeram uma correção periódica em torno de 0.15 (ms). O nó 4, com 10% de taxa de deriva, teve uma correção periódica em torno de 0.5 (ms). Pela Figura 4.10 que a precisão dos relógios do sistema ficou em torno de 0.7 (ms). A Figura 4.11 mostra que os relógios dos nós se sincronizaram entre si. O tempo global sofreu uma pequena deriva em relação ao tempo ideal, isto é, todos os nós adiantaram seu relógio em relação ao relógio de referência mas se mantiveram sincronizados entre si. A Figura 4.12 mostra a resposta ao degrau do sistema de controle. E a Figura 4.13 mostra as leis de controle do sistema. Através das Figuras 4.12 e 4.13 é possível ver que a sincronização de relógios minimizou a diferença da resposta ao degrau e das leis de controle dos nós em relação ao sistema ideal. Apesar do sistema ser um pouco adiantado em relação ao sistema ideal, as respostas ao degrau estão sincronizadas entre si, por causa da correção do algoritmo de sincronismo de relógios. Neste caso o algoritmo mostrou-se eficiente corrigindo o erro de deriva presente no sistema, aproximando os nós do sistema de controle de um sistema de controle ideal. A Tabela 4.4 mostra, na linha do *1º Caso (C)*, os dados de controle do nó 4 em comparação com o nó 1. A *% de sobre elevação*, com sincronização, é muito próxima e o *instante de pico* tem uma diferença de 0.2 (ms).

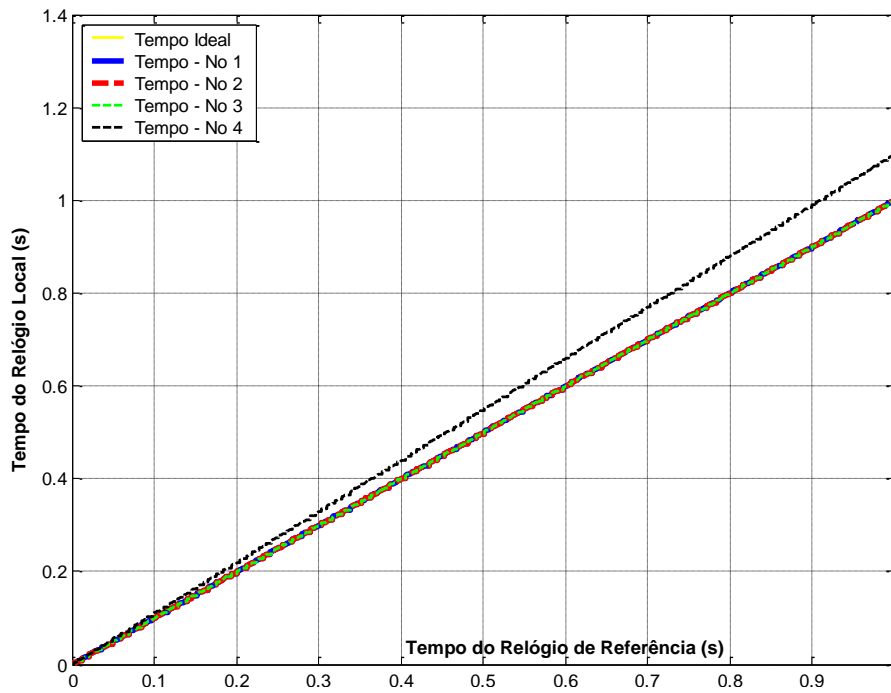


Figura 4.7 – Linha do Tempo com deriva de 10% - AF

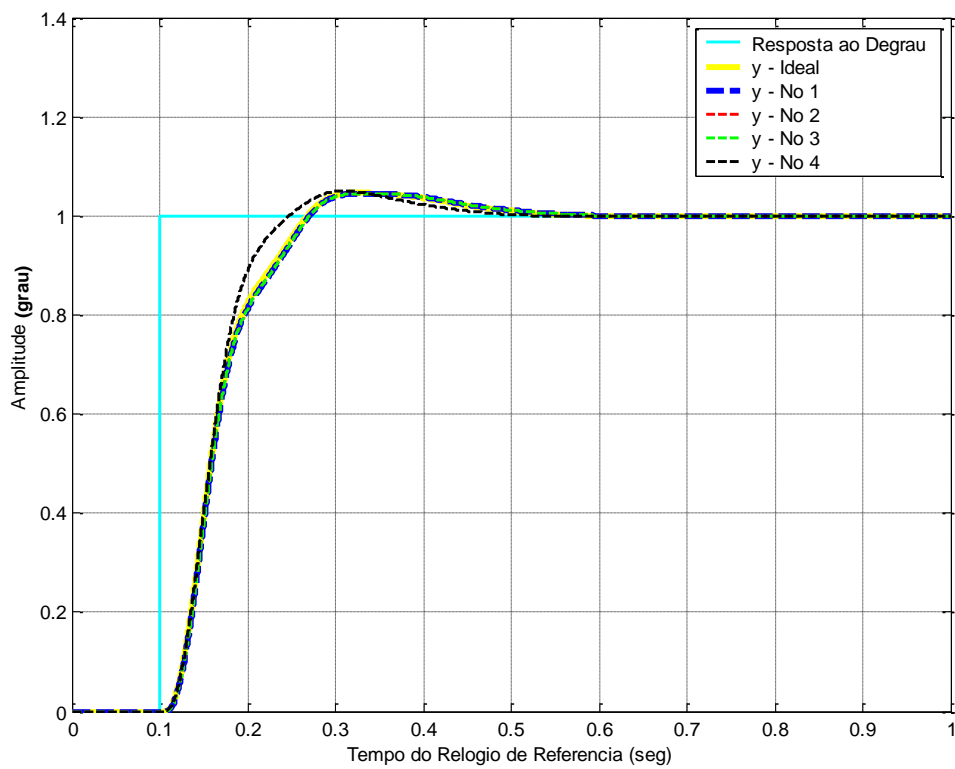


Figura 4.8 – Resposta ao Degrau com deriva de 10% - AF.



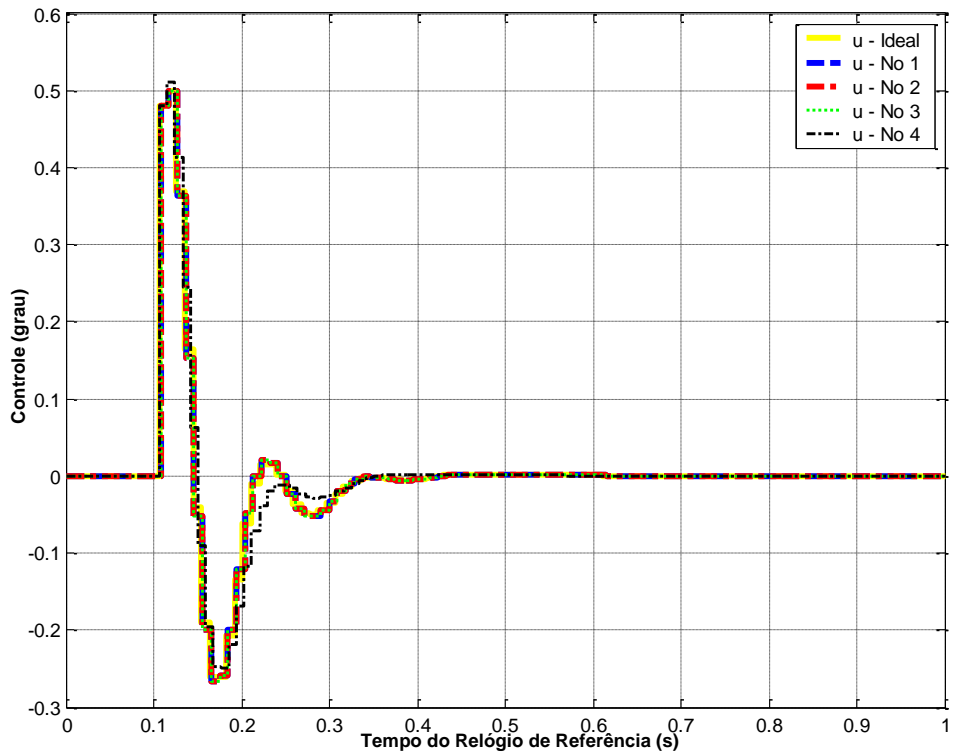


Figura 4.9 – Lei de Controle com deriva de 10% - AF.

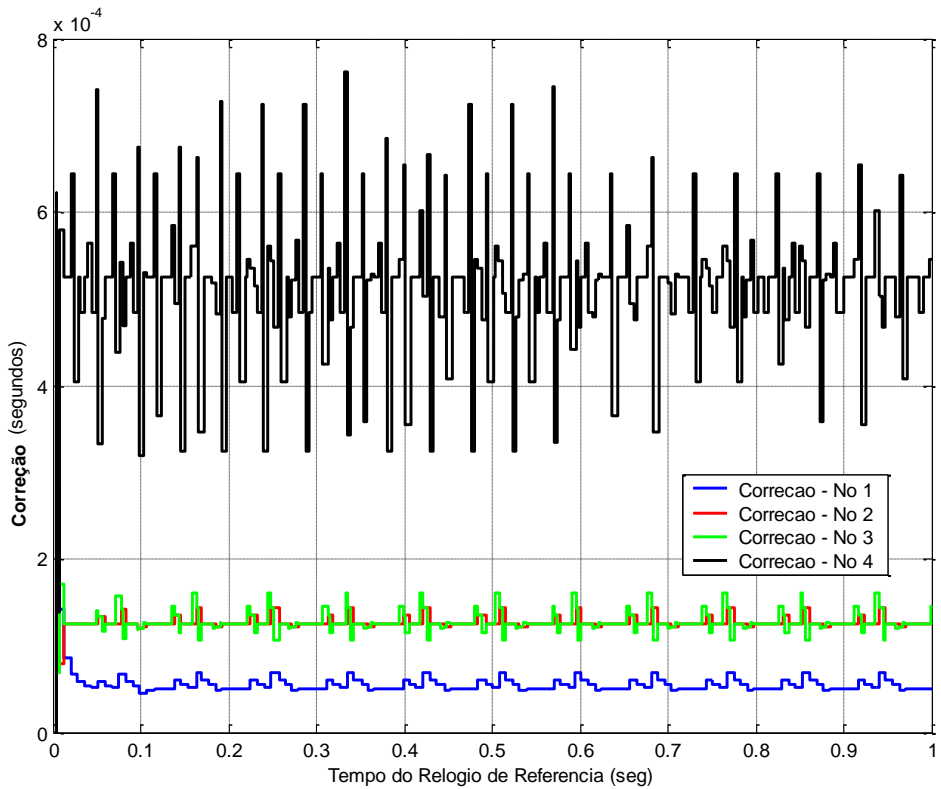


Figura 4.10 – Correção Algoritmo FTM com deriva de 10% - AF.

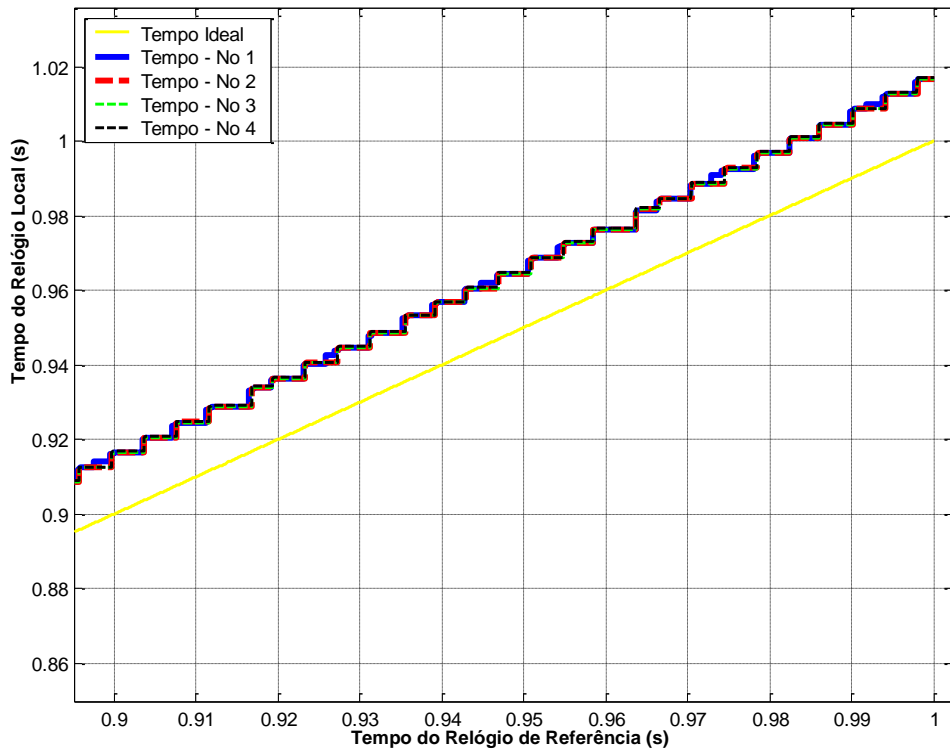


Figura 4.11 – Linha do Tempo Sincronizada com deriva de 10% - AF.

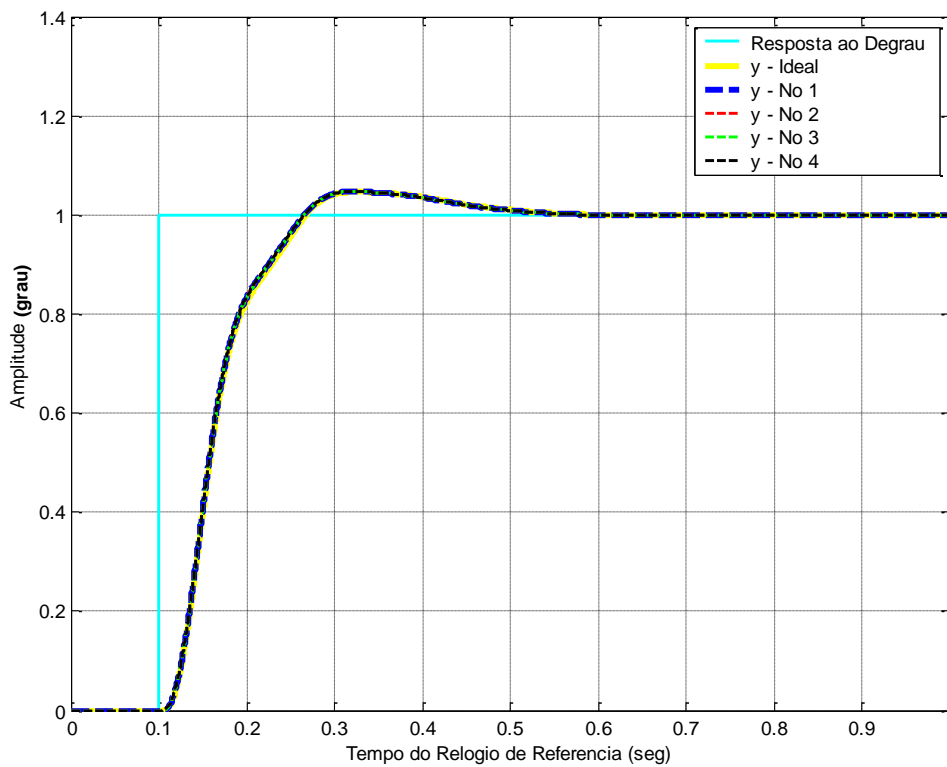


Figura 4.12 – Resposta ao Degrau com deriva de 10% Sincronizado - AF.

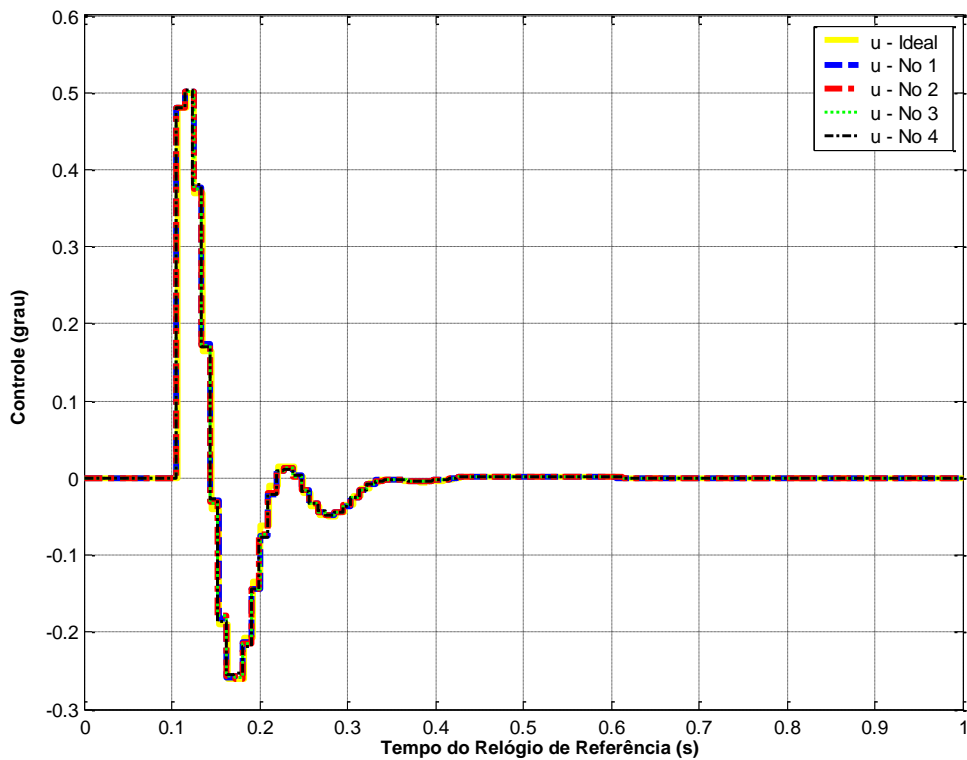


Figura 4.13 – Lei de Controle com deriva de 10% Sincronizado - AF.

#### 4.2.3. Caso 2 – FTM - Relógio com Viés Inicial

Utilizando a mesma simulação do caso 1, com a sincronização através do **algoritmo FTM**, neste caso, variou-se o **parâmetro de viés inicial** (*offset*) do relógio do nó (subsistema) 4. O objetivo deste caso é: 1) mostrar a influência que o parâmetro de viés inicial de um relógio tem sobre o algoritmo de sincronização e conseqüentemente sobre o sistema de controle com AF; 2) utilizar o algoritmo FTM para tentar corrigir esta imperfeição do relógio e analisar o sistema de controle distribuído.

Este trabalho escolheu 2 valores de viés inicial diferentes para variação do relógio:

- **Caso A: -0.2 (seg)** - O relógio inicia atrasado de 0.2 segundos em relação aos outros;
- **Caso B: 0.39 (seg)** - O relógio inicia adiantado de 0.39 segundos em relação aos outros;

Foram escolhidos estes dois valores de viés inicial pois foi constatado, por simulação, que os valores dentro dessa faixa não alteram significativamente os sistemas de controle com AF desta simulação.

O **caso B** é apresentado no **Apêndice E**. No **caso A**, com o **viés inicial de -0.2 segundos**, o relógio do nó 4 inicia atrasado em relação ao relógio ideal e aos outros relógios dos outros nós, como mostra a Figura 4.14. Com isso, o sistema de controle do nó 4 sofre uma pequena variação no overshoot, que está afetado pelo relógio. A Figura 4.15 mostra a resposta ao degrau do sistema de controle com o viés inicial de -0.2 segundos. É possível ver que somente o nó 4 sofre variação em relação ao ideal e aos outros nós do sistema. O nó 4 atrasa sua lei de controle em 0.1 segundos, i.e., o relógio do nó 4 inicia atrasado. A Figura 4.16 mostra as leis de controle aplicadas a cada nó do sistema e do sistema ideal onde somente o nó 4 difere sua resposta ao degrau e a lei de controle dos outros nós do sistema. O sistema de controle do nó 4 sofre o impacto do atraso, para tentar corrigir este problema vamos aplicar-se o algoritmo de sincronização de relógios FTM. A Figura 4.18 mostra a linha do tempo com o algoritmo de sincronização de relógios FTM aplicado ao sistema. Na Figura 4.17 é possível ver o gráfico de correção dos relógios dos respectivos nós do sistema. A Figura 4.17 mostra que o algoritmo FTM convergiu. Os nós 1, 2 e 3, que não estavam com erro de viés inicial, fizeram uma correção periódica em torno de 0.1 (ms) enquanto o nó 4 faz uma correção periódica em torno de 0.1 (ms). Com isso, é possível constatar graficamente pela Figura 4.17 que a precisão dos relógios do sistema ficou em

torno de 0.2 (ms). A Figura 4.18 mostra que os relógios dos nós se sincronizaram entre si.

O algoritmo de sincronização de relógios FTM tem em sua suposição que todos os relógios da rede devem iniciar sincronizados, isto é, sem viés inicial em nenhum relógio do conjunto. O interessante deste caso é que apesar deste caso desobedecer a regra, o algoritmo consegue sincronizar os relógios. Isto ocorre, pois o algoritmo de sincronização é tolerante a falhas e ao iniciar a sincronização em 0.0040 segundos o relógio do nó 4 esta indicando 0 segundos. Mesmo com este fato o algoritmo corrige o relógio. Como o controle em 0.0040 ainda é zero, o sistema corrige o relógio antes do controle atuar e com isso não prejudica o sistema. A Figura 4.19 mostra a resposta ao degrau do sistema de controle. E a Figura 4.20 mostra as leis de controle. Através das Figuras 4.19 e 4.20 é possível ver que a sincronização de relógios anulou o erro que o viés inicial do relógio 4 causou à resposta ao degrau e a lei de controle. E os relógios do sistema estão sincronizados entre si. Neste caso, em particular, o algoritmo mostrou-se muito eficiente corrigindo o erro de viés inicial do relógio 4. A Tabela 4.4 mostra, na linha do 2º Caso (A), os dados de controle do nó 4 em comparação com o nó 1. A **% de sobre elevação**: Nó 1 sincronizado: 4.5905%; Nó 4 sem sincronização: 4.5080%; Nó 4 sincronizado: 4.5808%. O **instante de pico**: Nó 1 sincronizado: 0.3264 (s); Nó 4 sem sincronização: 0.4255 (s); Nó 4 sincronizado: 0.3264 (s).

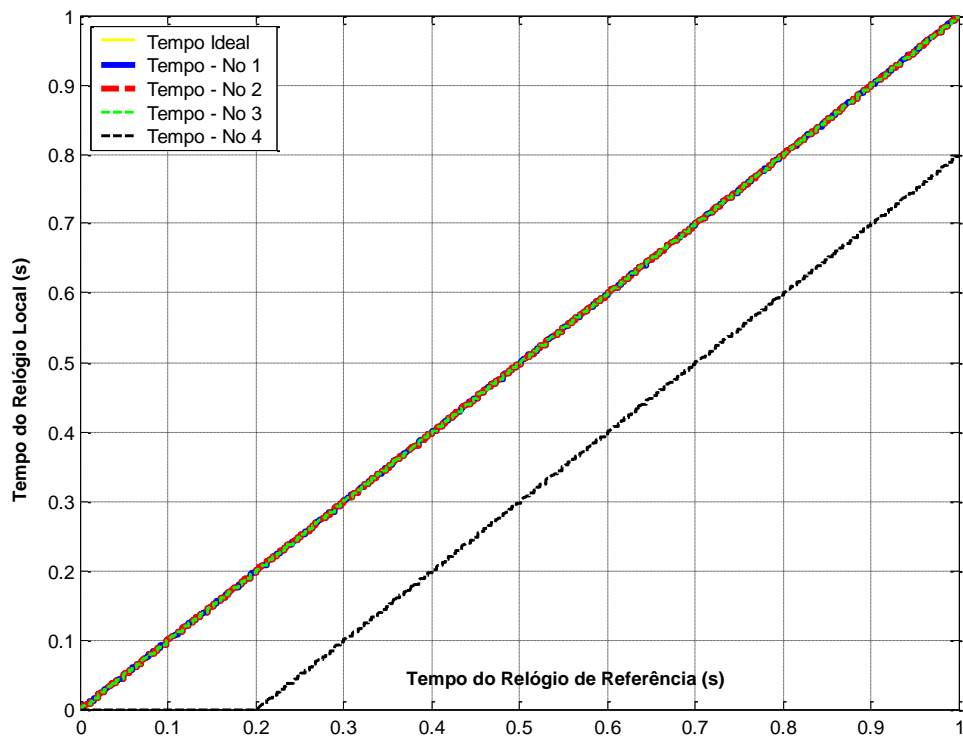


Figura 4.14 – Linha do Tempo com viés inicial de -0.2 segundos - AF.

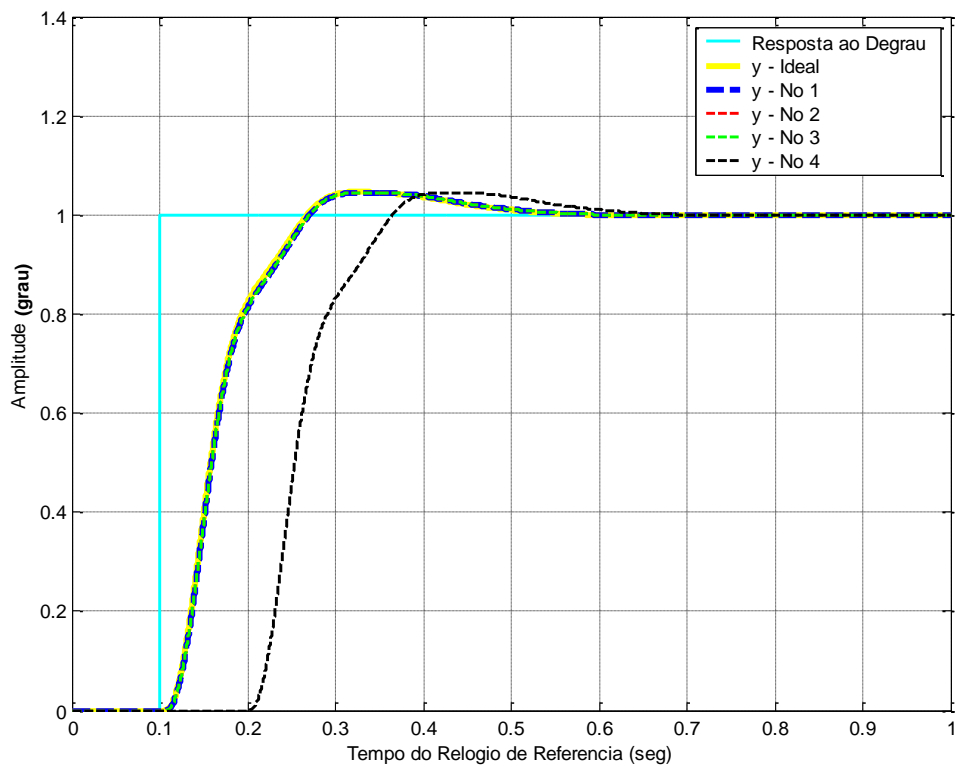


Figura 4.15 – Resposta ao Degrau com viés inicial de -0.2 segundos - AF.

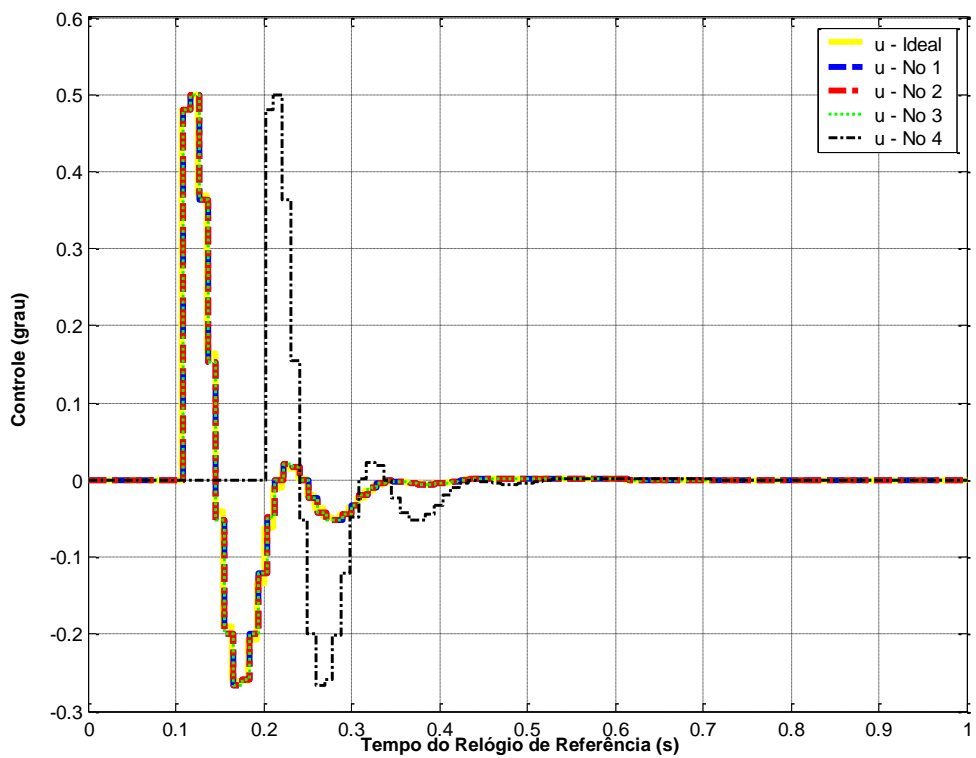


Figura 4.16 – Lei de Controle com viés inicial de -0.2 segundos - AF.

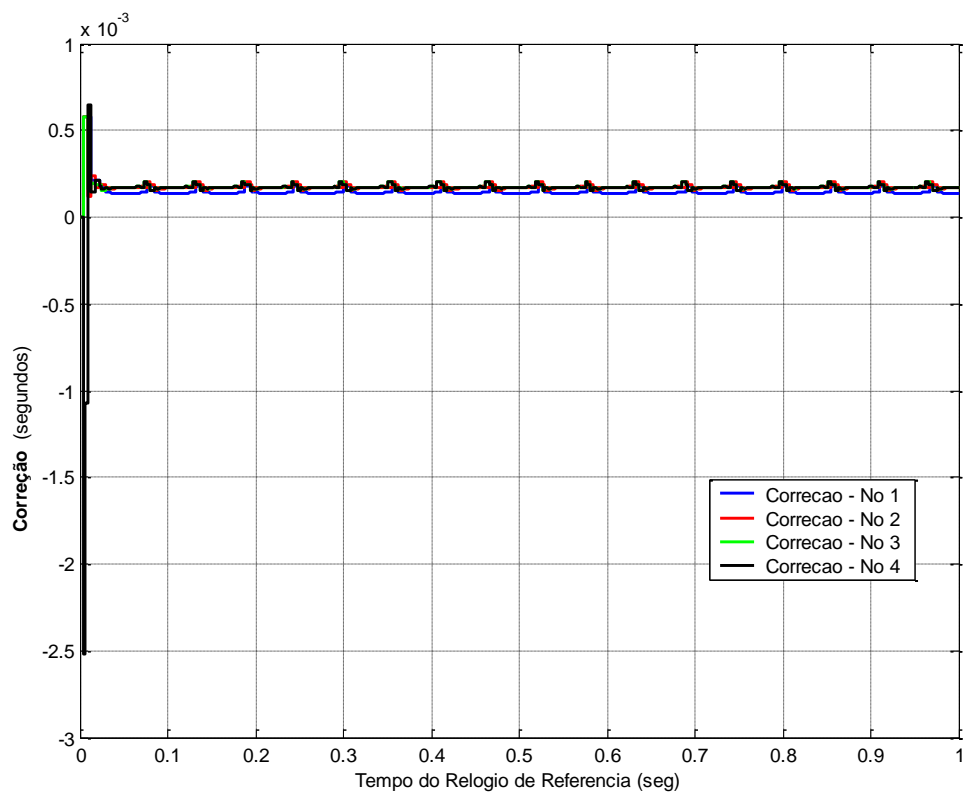


Figura 4.17 – Correção Algoritmo FTM com viés inicial de -0.2 segundos - AF.

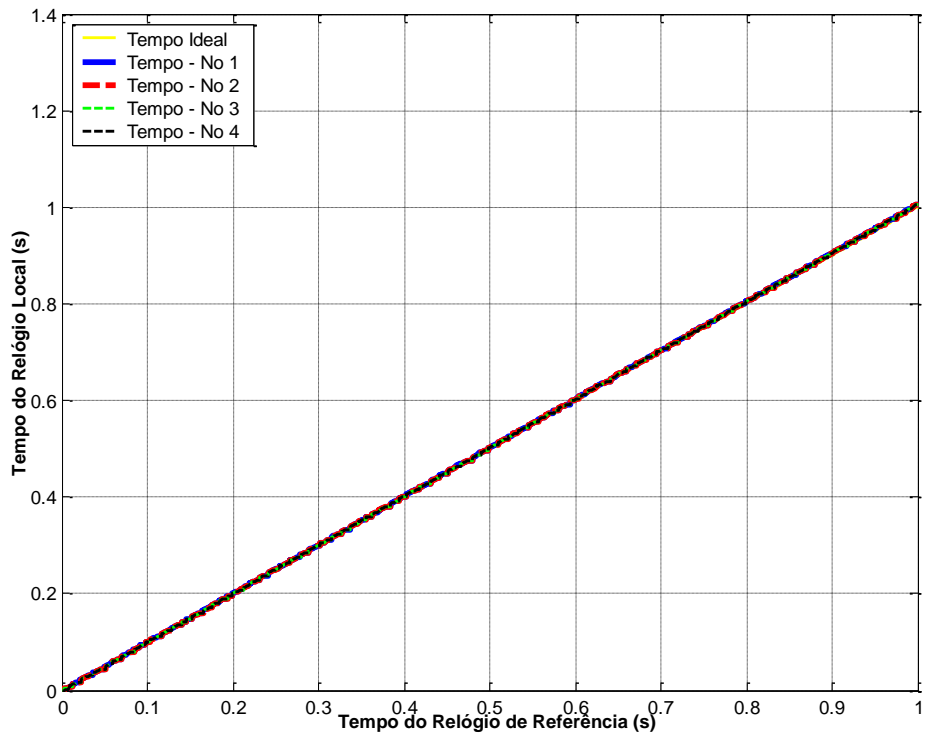


Figura 4.18 – Linha do Tempo com viés inicial de -0.2 segundos – Sincronizado - AF.

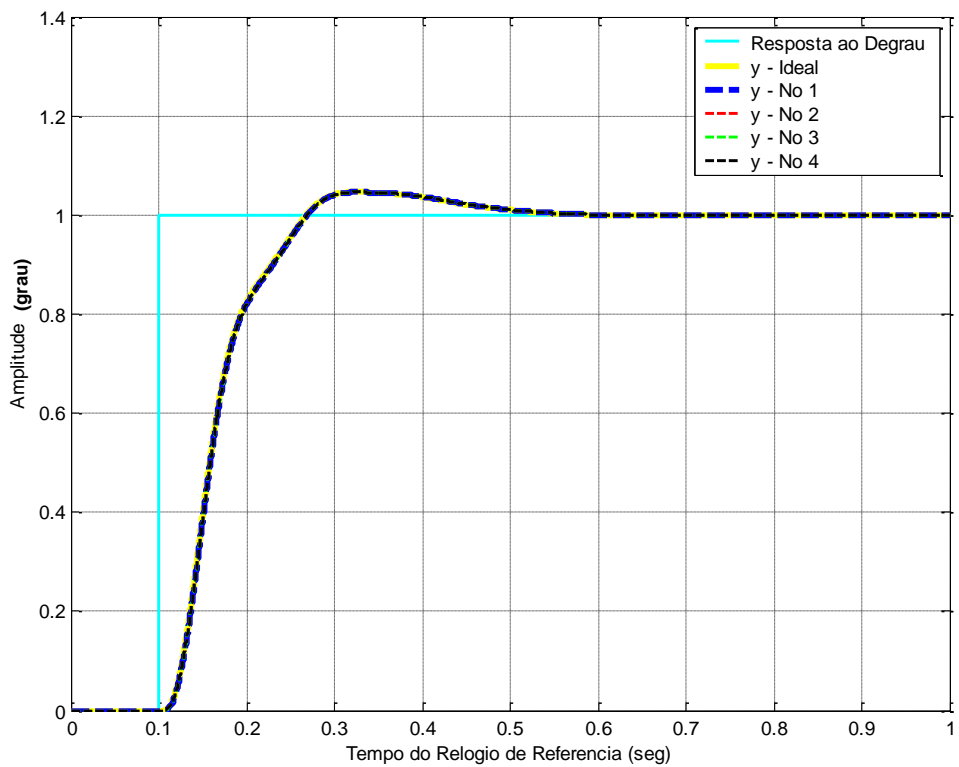


Figura 4.19 – Resposta ao Degrau com viés inicial de -0.2 seg. – Sincronizado -AF.



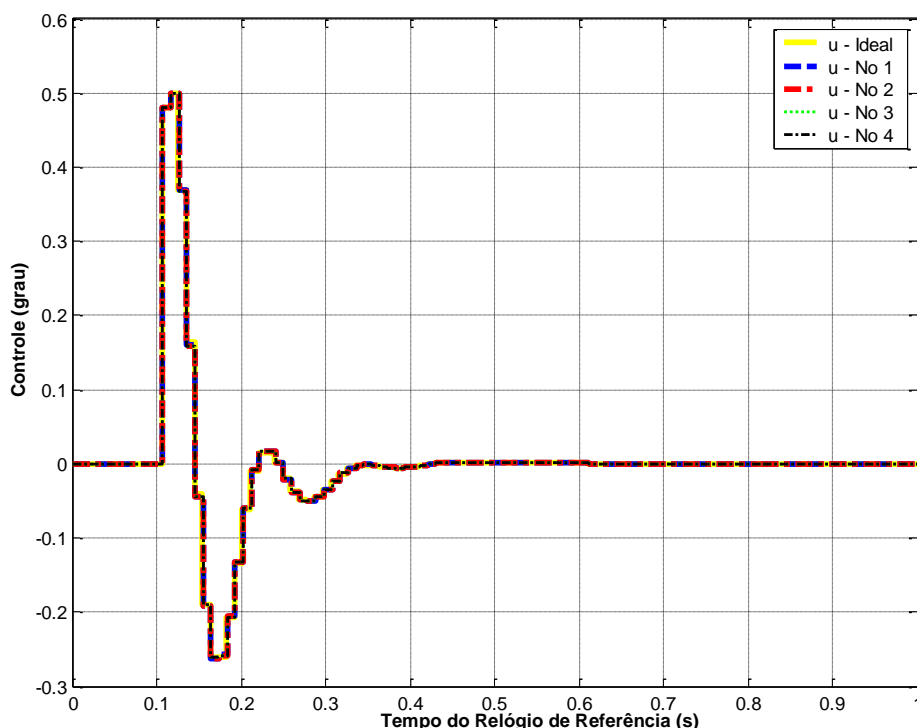


Figura 4.20 – Lei de Controle com viés inicial de -0.2 segundos – Sincronizado - AF.

#### 4.2.4. Caso 3 - FTM– Relógio com Deriva e Viés Inicial

Utilizando a mesma simulação do caso 1, neste caso, variou-se os parâmetros da **deriva** e de **viés inicial** (*offset*) do relógio do nó (subsistema) 4. O objetivo deste caso é: 1) mostrar a influência que os parâmetros de deriva e viés inicial de um relógio tem sobre o algoritmo de sincronização e conseqüentemente sobre o sistema de controle com AF; 2) utilizar o algoritmo FTM para tentar corrigir esta imperfeição do relógio e analisar o sistema de controle.

Este trabalho escolheu 2 valores deriva e de viés inicial diferentes para variação do relógio do nó 4:

- **Caso A: Deriva: -10%; Viés: -0.2 (seg)** - O relógio inicia atrasado de 0.2 segundos em relação aos outros e esta com uma taxa de deriva negativa, isto é, o relógio está atrasando em relação ao outros;

- **Caso B: Deriva: 10%; Viés: 0.39 (seg)** - O relógio inicia adiantado de 0.39 segundos em relação aos outros e esta com uma taxa de deriva positiva, isto é, o relógio está adiantando em relação ao outros;

Foram escolhidos estes dois valores de viés inicial, pois são uma mistura dos valores dos casos anteriores.

O **caso A** é apresentado no **Apêndice F**. No **caso B**, com um **viés inicial de 0.39 segundos e taxa de deriva de 10%**, o relógio do nó 4 inicia seu relógio adiantado e além disso o relógio está se adiantando em relação ao relógio ideal e aos outros relógios do conjunto, como mostra a Figura 4.21. Com isso, o sistema de controle do nó 4 que esta afetado pelo relógio sofre uma variação. A Figura 4.22 mostra a resposta ao degrau dos sistemas de controles. É possível ver que somente o nó 4 sofre variação. Isto ocorre pois o nó 4 adianta sua lei de controle em 0.39 segundos e também sofre de uma taxa de deriva de 10% ao longo de sua operação. Ainda na Figura 4.22 pode-se ver que o erro de relógio aumenta a *% de sobre elevação* e altera o instante de pico e o valor de pico da resposta ao degrau. A Figura 4.23 mostra as leis de controle aplicadas a cada nó do sistema e do sistema ideal. Somente o nó 4 difere sua resposta ao degrau e a lei de controle dos outros nós do sistema. O sistema de controle do nó 4 sofre o impacto do adiantamento inicial e da deriva e a lei de controle sofre um aumento na sua amplitude. Para corrigir este problema vamos aplicar o algoritmo FTM de sincronização de relógios. Na Figura 4.24 é possível ver o gráfico de correção dos relógios dos respectivos nós do sistema. A Figura 4.24 mostra que o algoritmo FTM **não convergiu**. Os nós 1, 2 e 3, que não estavam com erro não fizeram nenhuma correção. O nó 4, com viés inicial 0.39 segundos e deriva de 10% teve vários picos grandes de correção, mas durante o tempo de simulação o algoritmo não convergiu a nenhum valor. A Figura 4.25 mostra a linha do tempo com o algoritmo de sincronização de relógios FTM aplicado ao sistema. Os relógios dos nós 1, 2 e 3 se sincronizaram entre si, mas o relógio do nó 4 não conseguiu se sincronizar. O

algoritmo de sincronização de relógios FTM tem em sua suposição que todos os relógios da rede devem iniciar sincronizados, isto é, sem viés inicial em nenhum relógio do conjunto. Este caso desrespeita a regra e o algoritmo não consegue sincronizar todos os relógios do conjunto. Apesar do relógio 4 ser prejudicado, pelo fato do algoritmo de sincronização ser tolerante a falhas os outros relógios do conjunto, os nós 1, 2 e 3, não são afetados pelo erro no nó 4. A Figura 4.26 mostra a resposta ao degrau do sistema de controle. E a Figura 4.27 mostra as leis de controle. Através das Figuras 4.26 e 4.27 é possível ver que a sincronização de relógios aumentou o erro que o viés inicial e a deriva do relógio 4 causaram à resposta ao degrau e a lei de controle do nó 4, fazendo com que o nó 4 perdesse sua estabilidade. Os relógios dos nós 1, 2 e 3 estão sincronizados entre si e seus sistemas de controle não foram afetados. Neste caso o algoritmo mostrou-se ineficiente ao corrigir o erro de viés inicial e deriva do relógio 4, amplificando o efeito negativo na lei de controle, chegando a instabilizá-lo. A Tabela 4.4 mostra, na linha do 3º Caso (b), os dados de controle do nó 4 em comparação com o nó 1.

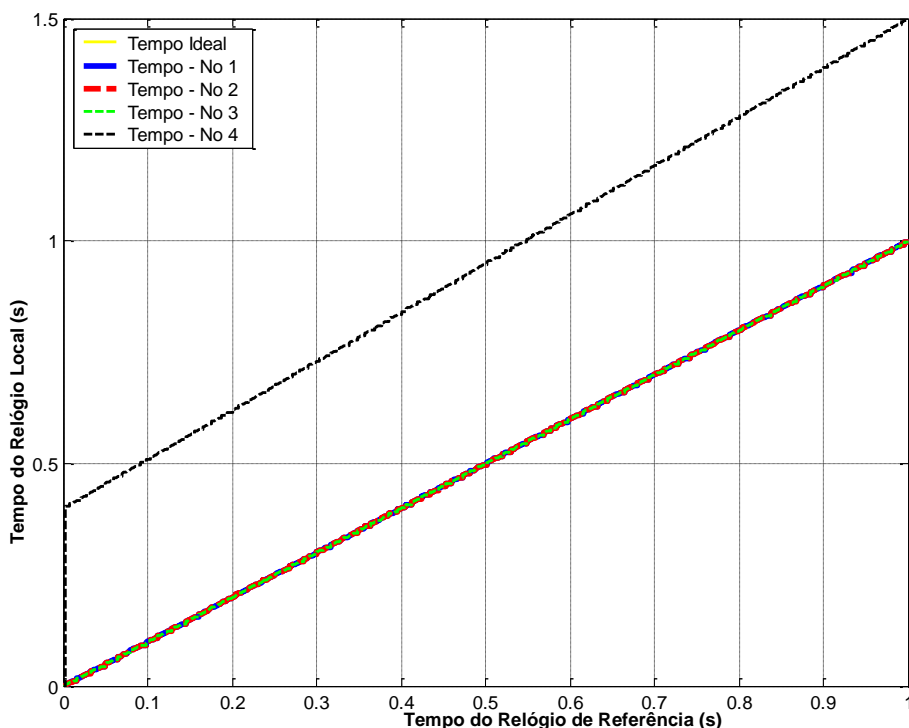


Figura 4.21 – Linha do Tempo com viés inicial de 0.39 s e deriva 10% - AF.

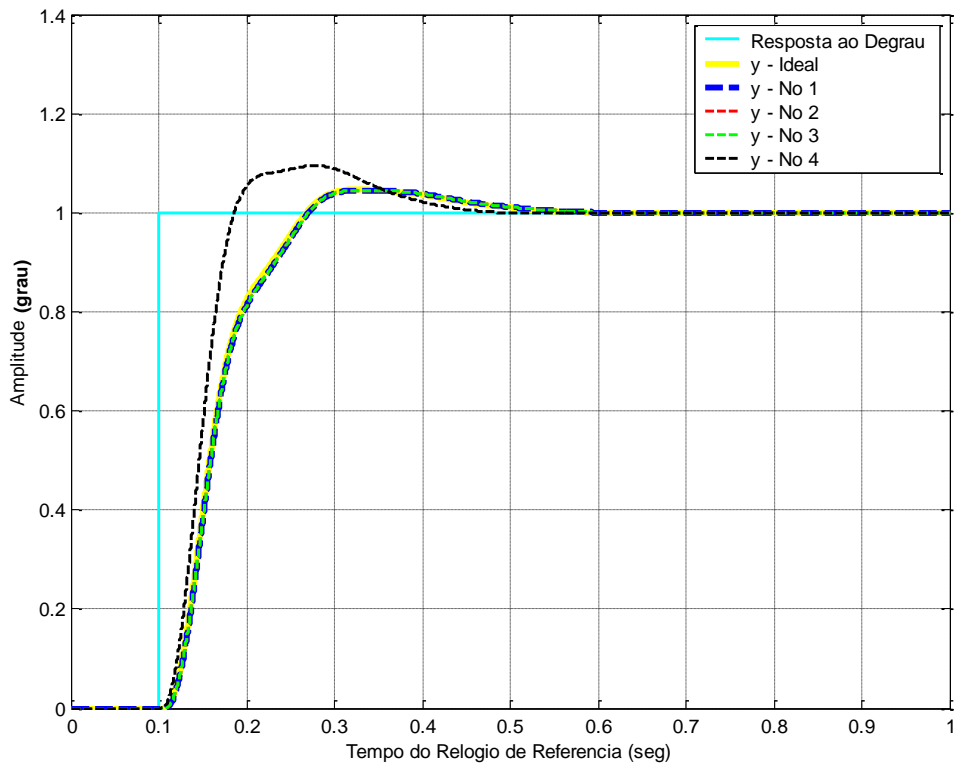


Figura 4.22 – Resposta ao Degrau com viés inicial de 0.39 s e deriva 10% - AF.

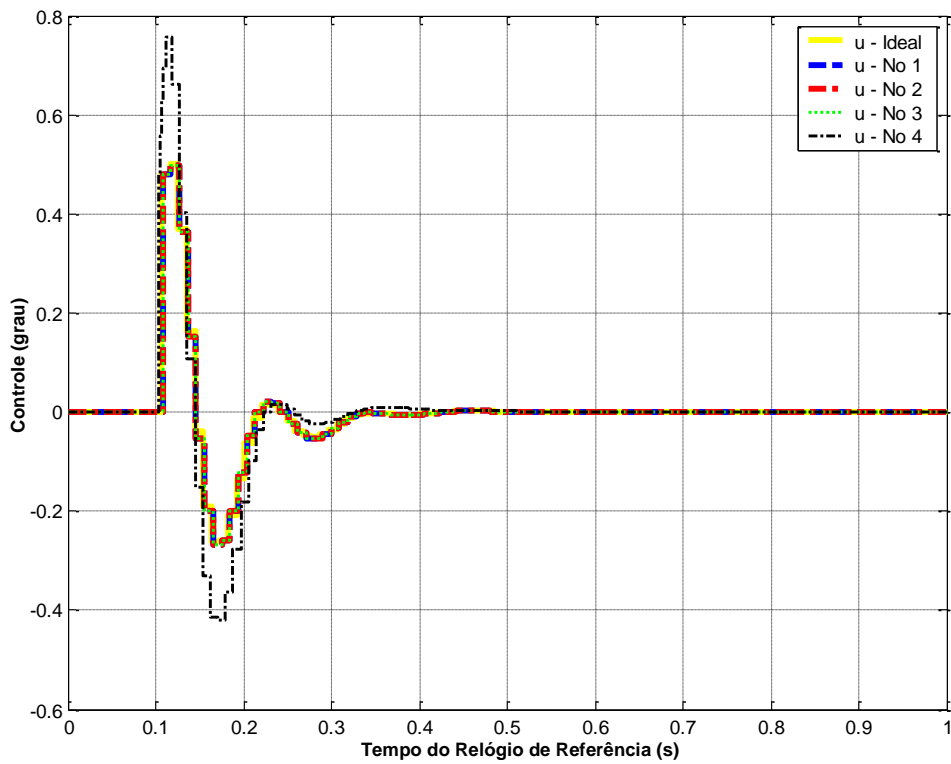


Figura 4.23 – Lei de Controle com viés inicial de 0.39 segundos e deriva 10% - AF.

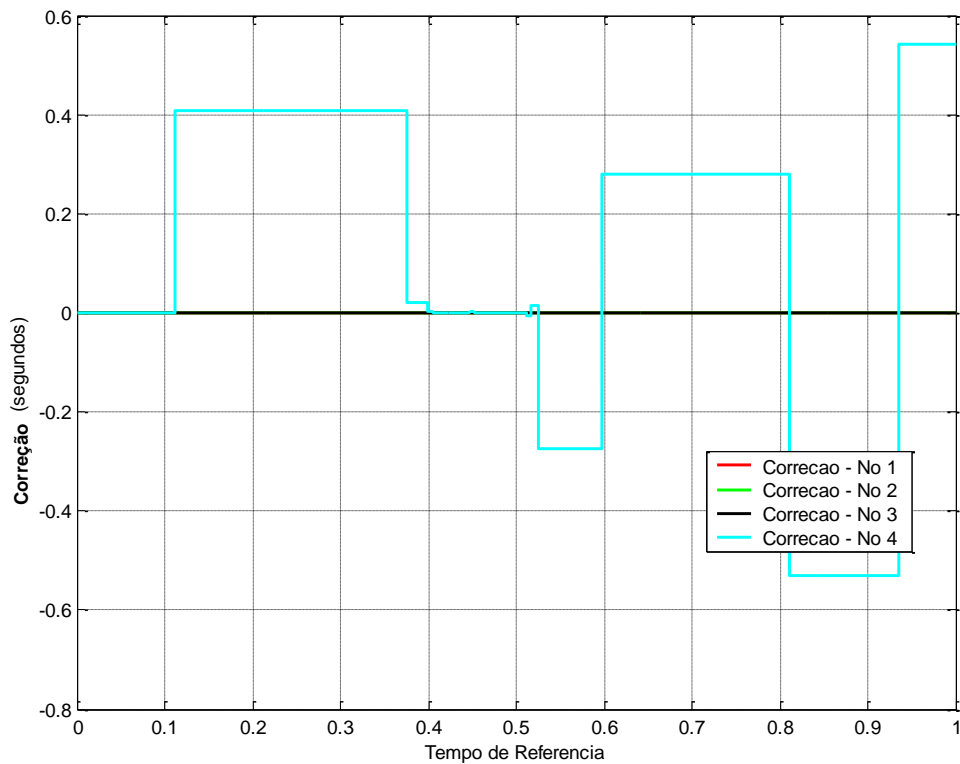


Figura 4.24 – Correção Algoritmo FTM: Viés inicial: 0.39 segundos e Deriva 10% - AF.

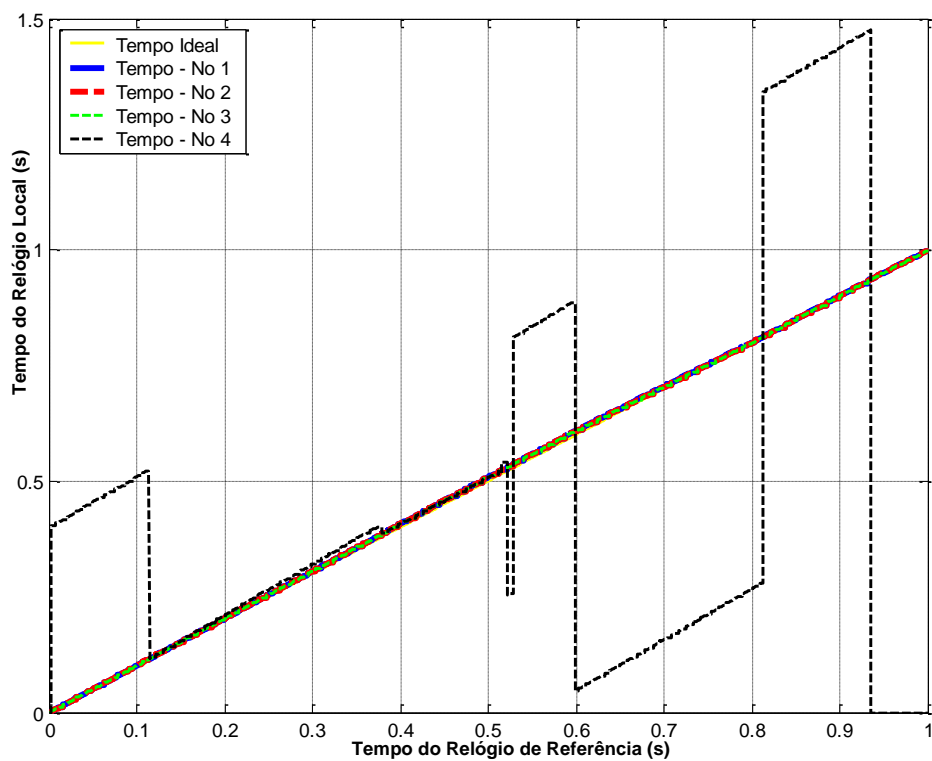


Figura 4.25 – Linha do Tempo Sincronizada: Viés inicial: 0.39 seg. e Deriva: 10% - AF.

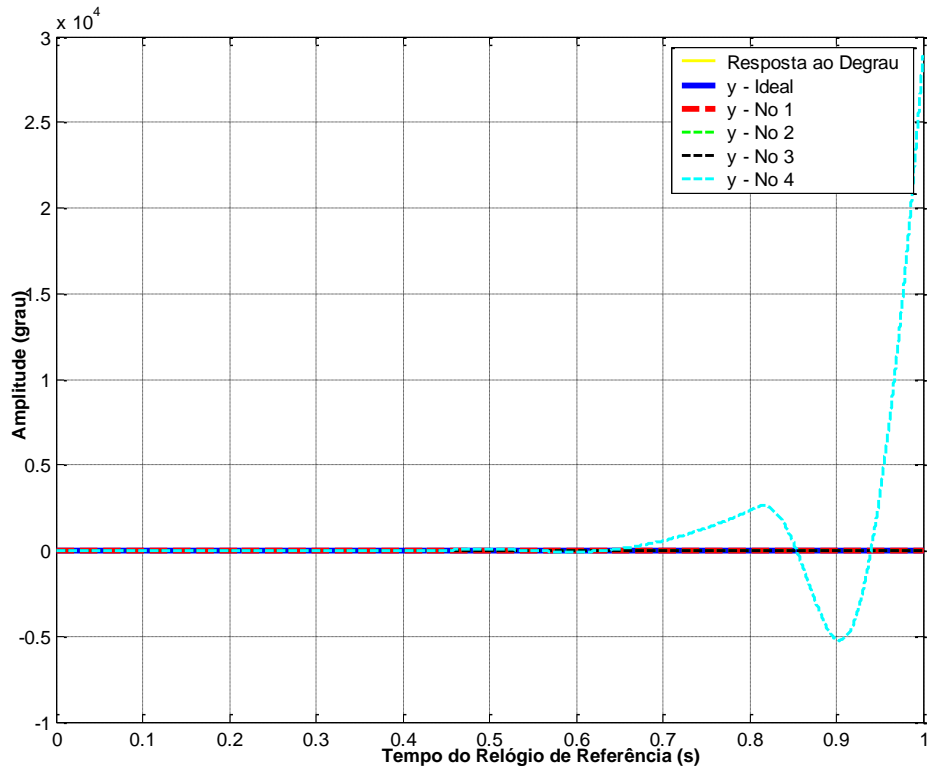


Figura 4.26 – Resposta ao Degrau Sincronizado: Viés inicial:0.39 seg, Deriva: 10% AF.

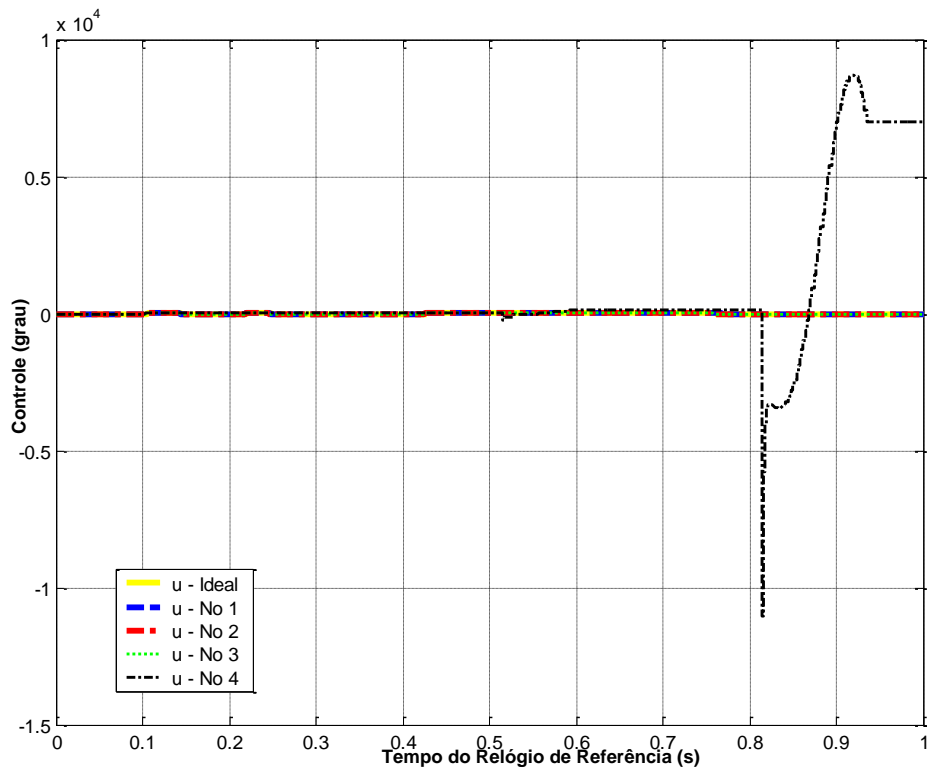


Figura 4.27 – Lei de Controle Sincronizada: Viés inicial: 0.39 seg., Deriva: 10% - AF.

#### 4.2.5. Caso 4 – Filtro de Kalman – Relógio com Deriva

Neste caso, foi feita uma simulação com o mesmo sistema de controle do 1º caso. A diferença agora é que a sincronização de relógios agora é feita através do **Filtro de Kalman (FK)**, com **arquitetura centralizada**, isto é, existe um relógio de referência. Para todos os casos simulados, o relógio escolhido como **relógio de referência** é o **relógio do nó 1**. Neste caso variou-se o **parâmetro de deriva** do relógio do nó (subsistema) 4. O objetivo deste caso é: 1) mostrar a influência que a deriva de relógio tem sobre o sistema de controle com AF; 2) utilizar o algoritmo FK para corrigir esta imperfeição do relógio e analisar o sistema de controle distribuído.

Este trabalho escolheu 3 taxas de deriva diferentes para variação do relógio:

- **Caso A: -10%** - O relógio 4 está se atrasando em relação ao relógio de referência a uma taxa de 10%;
- **Caso B: 1%** - O relógio 4 está se adiantando em relação ao relógio de referência com taxa de 1%;
- **Caso C: 10%** - O relógio 4 está se adiantando em relação ao relógio de referência com taxa de 10%;

Os **casos A e B** são apresentados no **Apêndice G**, com resultados diferentes. No **caso C**, a uma **taxa de deriva de 10%**, o relógio do nó 4 esta se adiantando em relação ao relógio de referência. Os outros relógios do conjunto não estão com erro de relógio. A Figura 4.7 mostra a linha do tempo com erro de deriva de 10% no relógio 4. Com isso, o sistema de controle do nó 4 sofre uma variação. A figura 4.8 mostra a resposta ao degrau com deriva de 10%. O nó 4 sofre uma variação pequena em relação ao nó 1 e aos outros nós do sistema. A Figura 4.9 mostra as leis de controle aplicadas a cada nó do sistema. Somente o nó 4, com erro de deriva, difere dos outros nós do conjunto

e do relógio 1 de referência. Com a variação de relógio o sistema de controle sofre um impacto pequeno com uma taxa de deriva de 10%. Com essa deriva o sinal de resposta ao degrau do nó 4 tem uma pequena elevação da % de sobre-elevação (*overshoot*).

Aplicando o Filtro de Kalman na sincronização de relógios, a Figura 4.28 mostra a linha do tempo sincronizada com o Filtro de Kalman aplicado ao sistema. Na Figura 4.29 é possível ver o gráfico de correção dos relógios dos respectivos nós do sistema. A Figura 4.29 mostra que o Filtro de Kalman convergiu. O nó 1 não corrige seu valor pois ele é a referência. Os nós 2 e 3, que não estavam com erro de deriva, fizeram uma correção periódica com o valor médio zero. O nó 4, com 10% de taxa de deriva, teve uma correção periódica com o valor médio 0.4 (ms). É possível constatar através do gráfico 4.29 que a precisão dos relógios do sistema ficou em torno de 2 (ms). A Figura 4.28 mostra que os relógios dos nós se sincronizaram com o relógio de referência do nó 1.

A Figura 4.30 mostra a resposta ao degrau do sistema de controle com o sistema sincronizado pelo filtro de Kalman. A Figura 4.31 mostra as leis de controle do sistema sincronizado com filtro de Kalman. Através das Figuras 4.30 e 4.31 é possível ver que a sincronização de relógios com o Filtro de Kalman minimizou a diferença do tempo de resposta e das leis de controle de cada nó. Apesar do nó 4 ser adiantado do nó 1, os tempos de respostas estão sincronizados entre si. Neste caso o algoritmo mostrou-se eficiente corrigindo o erro de deriva presente no nó 4 sistema. Com essa correção de relógios o algoritmo de sincronização com FK minimizou a diferença da porcentagem de sobre-elevação presente quando o relógio do nó 4 não está sincronizado. A Tabela 4.4 mostra, na linha do **4º Caso (C)**, os dados de controle do nó 4 em comparação com o nó 1. A **% de sobre elevação**: Nó 1 sincronizado: 4.5057%; Nó 4 sem sincronização: 5.0475%; Nó 4 sincronizado: 4.5204%. O **instante de**



**pico:** *Nó 1 sincronizado:* 0.33318 (s); *Nó 4 sem sincronização:* 0.3108 (s); *Nó 4 sincronizado:* 0.3296 (s).

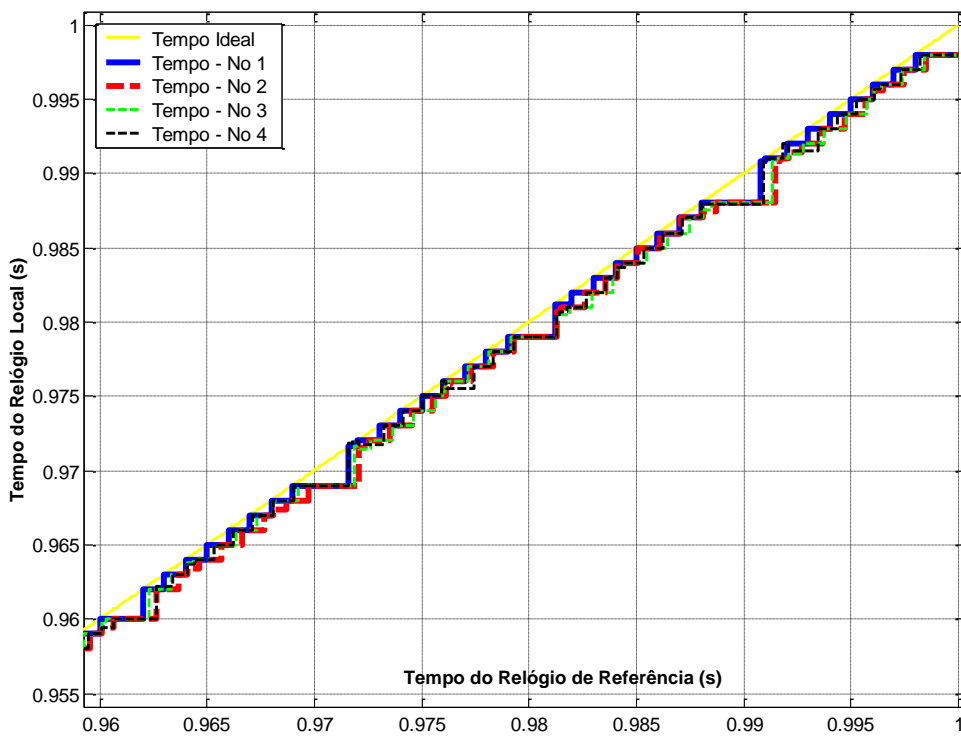


Figura 4.28 – Linha do Tempo Sincronizada com deriva de 10% - AF.

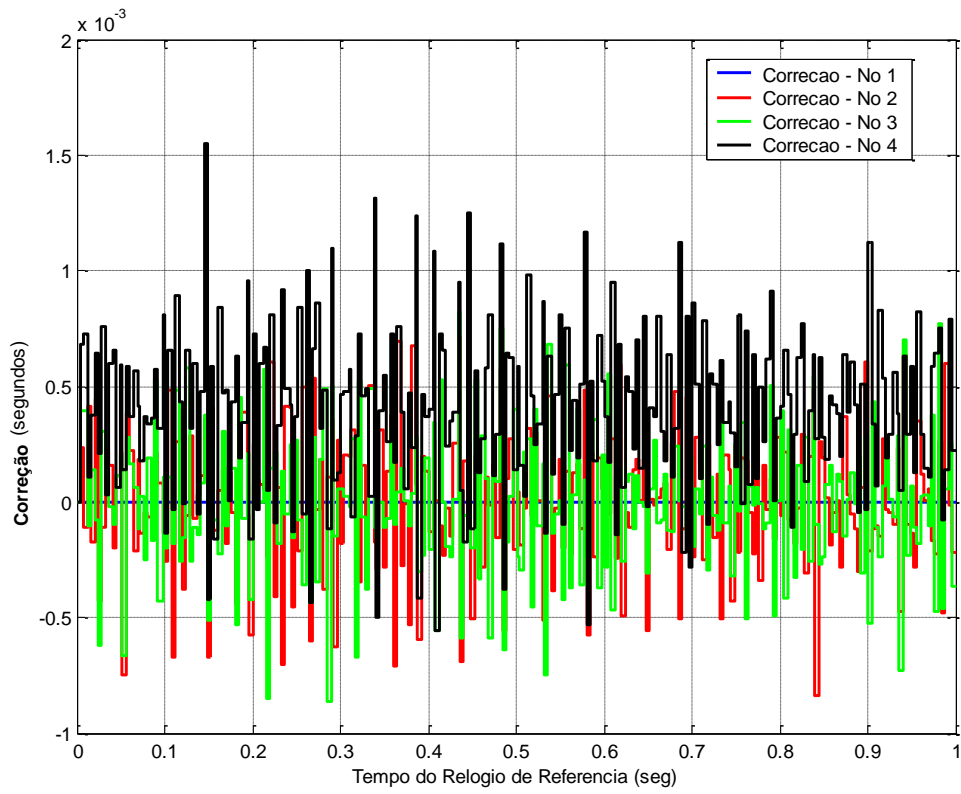


Figura 4.29 –Correção com Filtro de Kalman com deriva de 10% - AF.

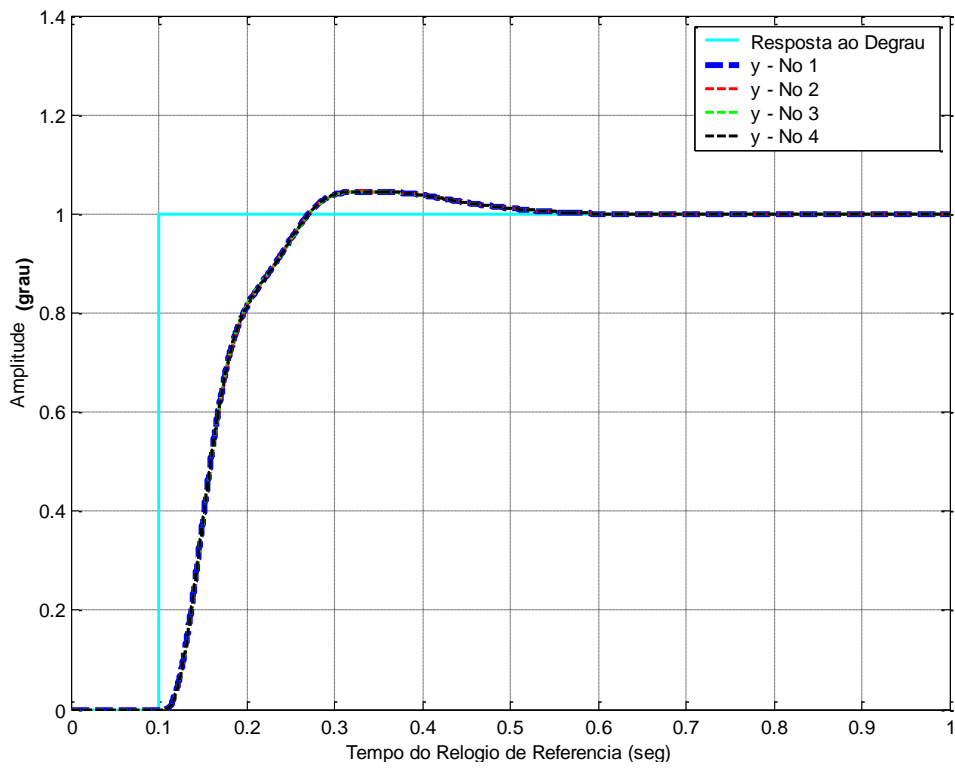


Figura 4.30 – Resposta ao Degrau com deriva de 10% Sincronizado - AF.

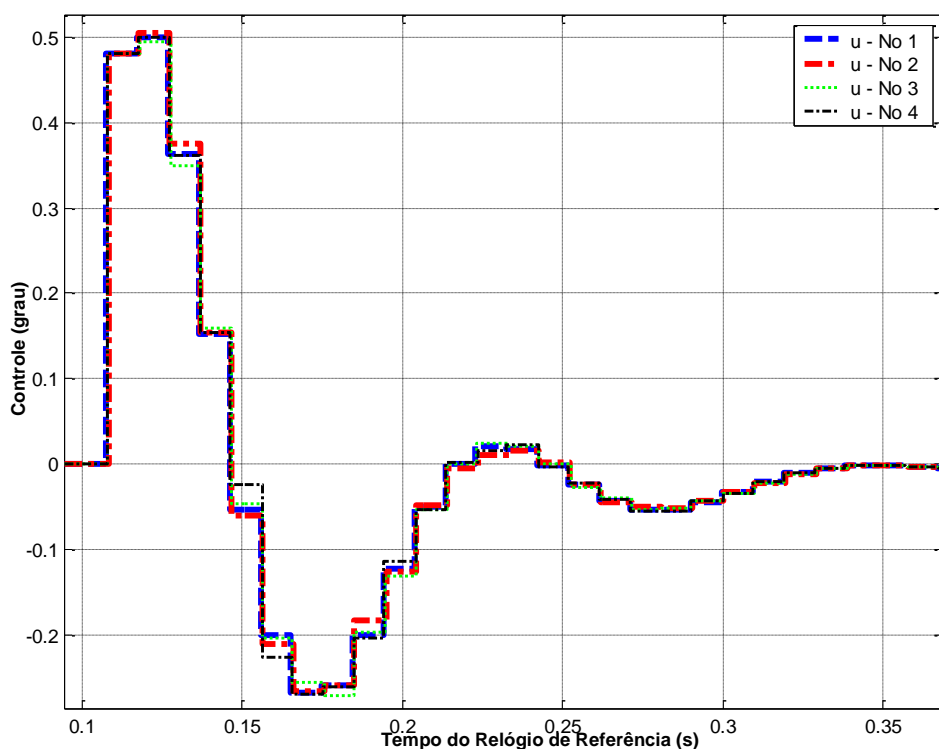


Figura 4.31 – Lei de Controle com deriva de 10% Sincronizado - AF.

#### 4.2.6. Caso 5 – Filtro de Kalman – Relógio com Viés Inicial

Neste caso, foi refeita a mesma simulação do 4º Caso, com a diferença que agora variou-se o **parâmetro de viés inicial** do relógio do nó (subsistema) 4.

Este trabalho escolheu 2 valores de viés inicial diferentes para variação do relógio:

- **Caso A: -0.2 (seg)** - O relógio inicia atrasado de 0.2 segundos em relação aos outros;
- **Caso B: 0.39 (seg)** - O relógio inicia adiantado de 0.39 segundos em relação aos outros;

O **caso B** é apresentado no **Apêndice H**, com resultados diferentes. No **caso A**, com o **viés inicial de -0.2 segundos**, o relógio do nó 4 inicia atrasado em

relação ao relógio de referência e aos outros relógios do conjunto, como mostra a Figura 4.14. A Figura 4.15 mostra a resposta ao degrau do sistema de controle com o **viés inicial de -0.2 segundos**. É possível ver que somente o nó 4 sofre variação em relação ao de referência e aos outros nós do sistema. O nó 4 atrasa sua lei de controle em 0.2 segundos. Isto ocorre pois o relógio do nó 4 inicia atrasado em relação ao de referência. A Figura 4.16 mostra as leis de controle aplicadas a cada nó do sistema e do sistema ideal. Somente o nó 4 difere sua resposta ao degrau e a lei de controle do nó 1, de referência, e dos outros nós do sistema. O sistema de controle do nó 4 sofre o impacto do atraso, para corrigir este problema vamos aplicar o algoritmo FK para sincronizar os relógios. A Figura 4.33 mostra a linha do tempo sincronizada com o FK aplicado ao sistema. Na Figura 4.32 é possível ver o gráfico de correção dos relógios dos respectivos nós do sistema. A Figura 4.32 mostra que o Filtro de Kalman convergiu. O nó 1 não faz correção pois é a referência, os nós 2 e 3, que não estavam com erro de viés inicial, fizeram uma correção periódica com valor médio zero. O nó 4, com viés inicial de -0.2 segundos, também faz uma correção periódica em torno de zero. Com isso, é possível constatar através do gráfico 4.32 que a precisão dos relógios do sistema ficou em torno de 2 (ms). A Figura 4.33 mostra que os relógios dos nós se sincronizaram com o relógio de referência. A Figura 4.34 mostra a resposta ao degrau do sistema de controle com sincronização com o uso do FK. E a Figura 4.35 mostra as leis de controle aplicadas aos nós. Através das Figuras 4.34 e 4.35 é possível ver que a sincronização de relógios com o FK minimizou o atraso do relógio do nó 4. Neste caso o algoritmo de FK mostrou-se eficiente corrigindo o erro de viés inicial presente no nó 4 do sistema. Com essa correção de relógios o algoritmo de sincronização com FK minimizou o atraso sobre a lei de controle do nó 4 causado pelo viés inicial do relógio. Na Tabela 4.4 temos, na linha do *5º Caso (A)*, os dados de controle: **A % de sobre elevação: Nó 1 sincronizado: 4.5057%; Nó 4 sem sincronização: 4.5080%; Nó**

4 sincronizado: 4.5207%. O instante de pico: Nó 1 sincronizado: 0.3318 (s);  
 Nó 4 sem sincronização: 0.4255 (s); Nó 4 sincronizado: 0.3318 (s).

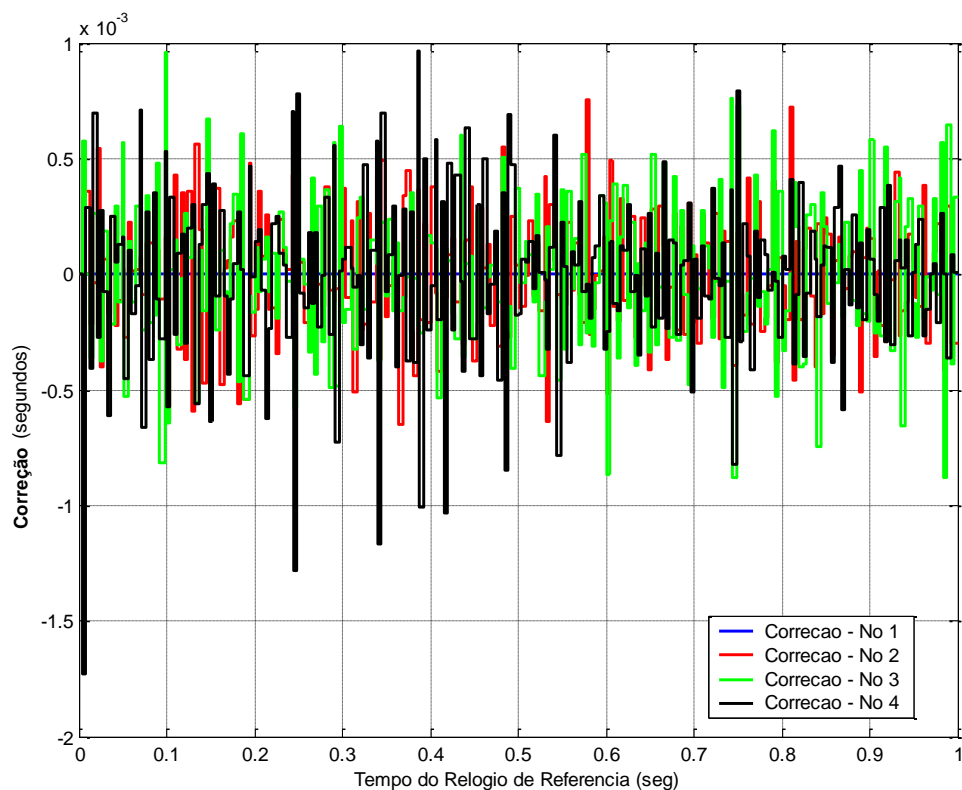


Figura 4.32 – Correção do Filtro de Kalman, Viés: -0.2 segundos - AF.

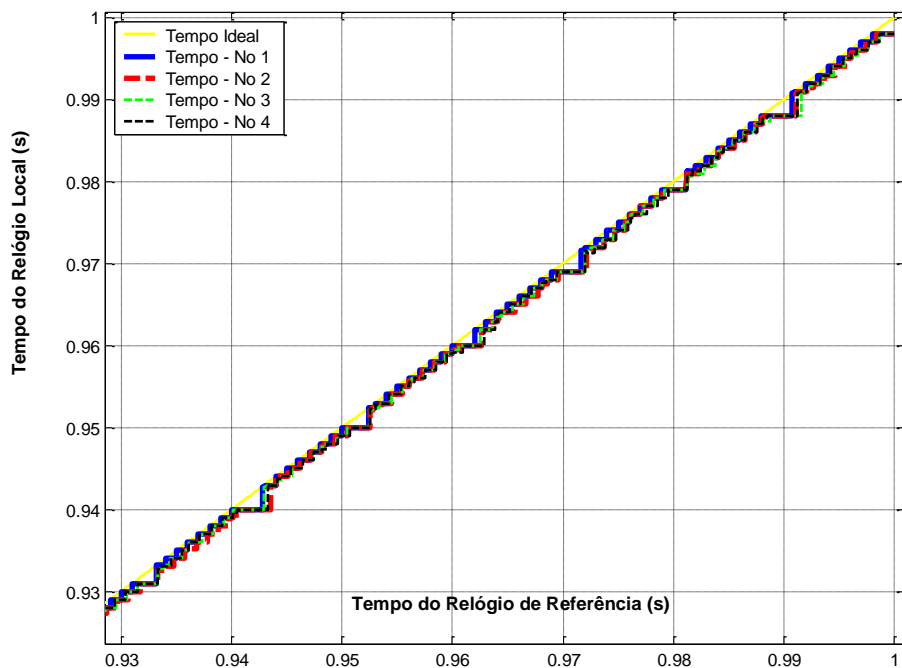


Figura 4.33 – Linha do Tempo Sincronizada - Viés inicial: -0.2 segundos - AF.

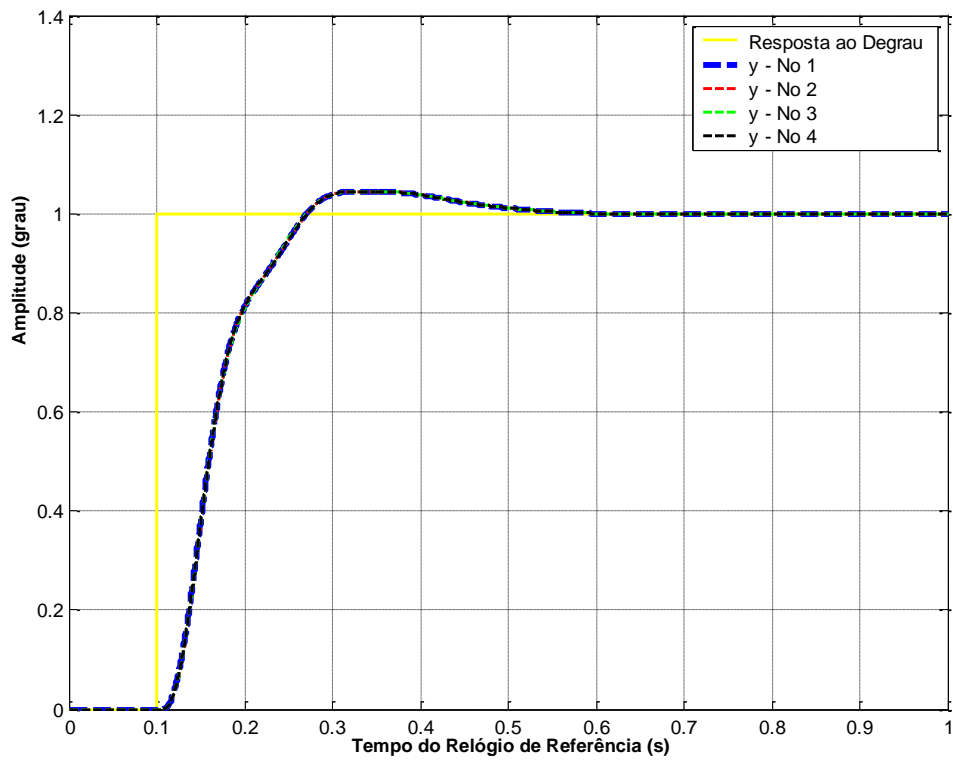


Figura 4.34 – Resposta ao Degrau com viés inicial de -0.2 (s) – Sincronizado - AF.

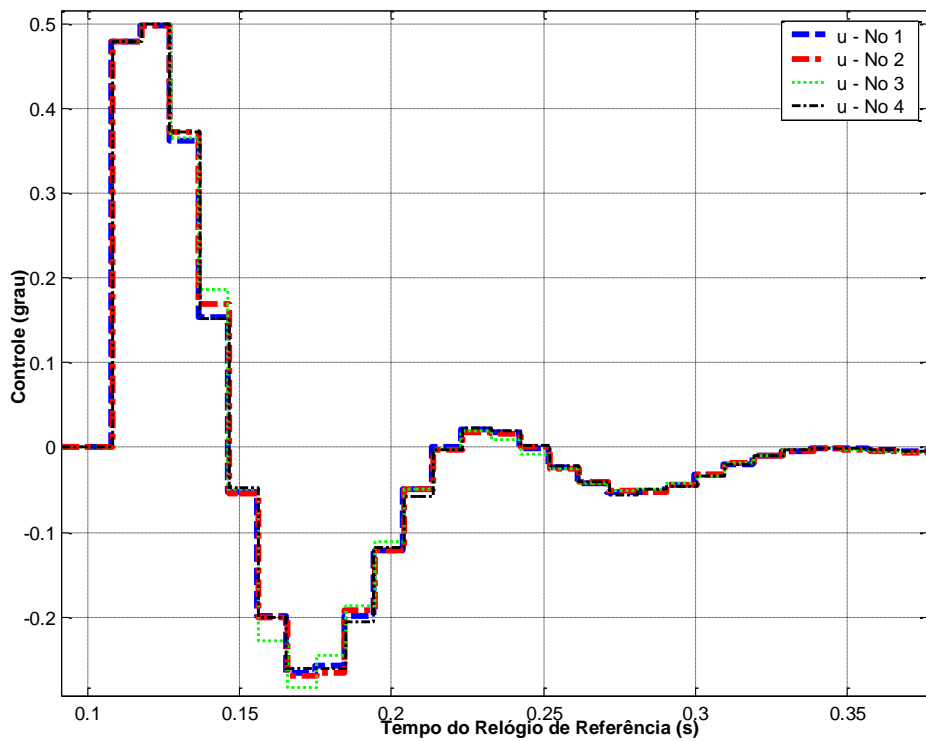


Figura 4.35 – Lei de Controle com viés inicial de -0.2 segundos – Sincronizado – AF.

#### 4.2.7. Caso 6 – Filtro de Kalman – Relógio com Deriva e Viés Inicial

Neste caso, foi refeita a mesma simulação do 4º Caso, com a diferença que agora variou-se o **parâmetro de viés inicial e deriva** do relógio do nó (subsistema) 4.

Este trabalho escolheu 2 valores deriva e de viés inicial diferentes para variação do relógio do nó 4:

- **Caso A: Deriva: -10%; Viés: -0.2 (seg)** - O relógio inicia atrasado de 0.2 segundos em relação ao relógio de referência e esta com uma taxa de deriva negativa, isto é, o relógio está se atrasando em relação ao relógio de referência;
- **Caso B: Deriva: 10%; Viés: 0.39 (seg)** - O relógio inicia adiantado de 0.39 segundos em relação ao relógio de referência e está com uma taxa de deriva positiva, isto é, o relógio está se adiantando em relação ao relógio de referência;

O **caso A** é apresentado no **Apêndice I**, com resultados diferentes. No **caso B**, com um **viés inicial de 0.39 segundos e taxa de deriva de 10%**, o relógio do nó 4 inicia adiantado e além disso o relógio está se adiantando em relação ao relógio de referência e aos outros relógios do conjunto, como mostra a Figura 4.21. A Figura 4.22 mostra a resposta ao degrau do sistema de controle com o viés inicial de 0.39 segundos e taxa de deriva de 10%. É possível ver que somente o nó 4 sofre variação em relação ao de referência e aos outros nós do sistema. Existe um aumento na porcentagem de sobre elevação da resposta ao degrau do nó 4. Isto ocorre pois o relógio do nó 4 inicia adiantado e com deriva positiva em relação ao relógio de referência. A Figura 4.23 mostra as leis de controle aplicadas a cada nó do sistema e do nó de referência. Somente o nó 4 difere sua resposta ao degrau e a lei de controle do nó 1, de referência, e dos outros nós do sistema. O sistema de controle do nó 4 sofre o impacto do

adiantamento e da deriva, a lei de controle do nó 4 sofre um aumento na sua amplitude. Para tentar corrigir este problema vamos aplicar **Filtro de Kalman** para sincronizar os relógios. A Figura 4.36 mostra a linha do tempo sincronizada com o FK aplicado ao sistema. Na Figura 4.37 é possível ver o gráfico de correção dos relógios dos respectivos nós do sistema. A Figura 4.37 mostra que o FK convergiu. O nó 1 não faz correção pois é a referência, os nós 2 e 3, que não estavam com erro de viés inicial e deriva, fizeram uma correção periódica com valor médio zero. O nó 4, com viés inicial 0.39 segundos e deriva de 10%, primeiramente faz uma correção grande de 0.39 segundos, para corrigir o viés inicial, após esta correção o nó converge e faz uma correção periódica em com valor médio em torno de 0.5 (ms). Com isso, é possível constatar, através do gráfico 4.37, que a precisão final dos relógios do sistema ficou em torno de 2 (ms). A Figura 4.36 mostra que os relógios dos nós se sincronizaram com o relógio de referência. A Figura 4.38 mostra a resposta ao degrau do sistema de controle com sincronização com o uso do Filtro de Kalman. A Figura 4.39 mostra as leis de controle. Através das Figuras 4.38 e 4.39 é possível ver que a sincronização de relógios  **aumentou o erro**  que o viés inicial e a deriva do relógio 4 causaram à resposta ao degrau e a lei de controle do nó 4. O FK causou um atraso ao sistema e a resposta ao degrau demorou mais tempo para corrigir. O sinal de controle do nó 4 ficou atrasado. Neste caso o algoritmo FK não mostrou-se eficiente corrigindo o erro de viés inicial e deriva presente no nó 4 do sistema. Com essa correção de relógios o algoritmo de sincronização com FK inseriu um atraso sobre a lei de controle do nó 4 causado pelo viés inicial do relógio. Os relógios dos nós 1, 2 e 3 estão sincronizados entre si e seus sistemas de controle não foram afetados. O algoritmo FK mostrou-se ineficiente ao corrigir o erro de viés inicial e deriva do relógio 4, amplificando o efeito negativo sobre a lei de controle. Na Tabela 4.4 temos, na linha do *6º Caso (B)*, os dados de controle: **A % de sobre elevação:** *Nó 1 sincronizado:* 4.5057%; *Nó 4 sem sincronização:* 9.4979%; *Nó 4*



sincronizado: 168.37%. O instante de pico: Nó 1 sincronizado: 0.3317 (s); Nó 4 sem sincronização: 0.2784 (s); Nó 4 sincronizado: 0.4905 (s).

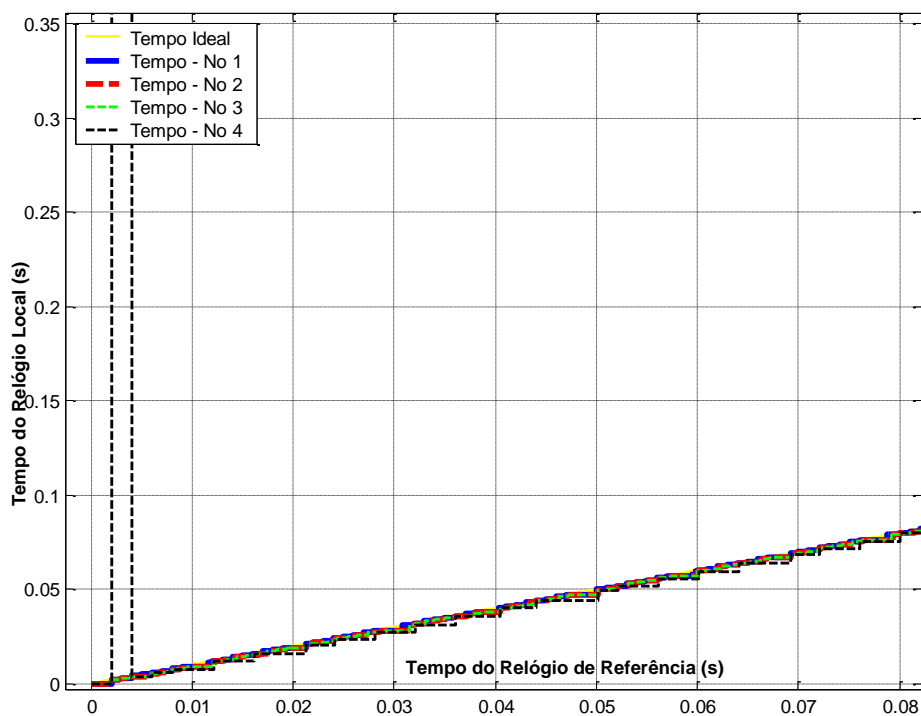


Figura 4.36 – Linha do Tempo Sincronizada - Viés inicial: 0.39 (s) e Deriva: 10% - AF.

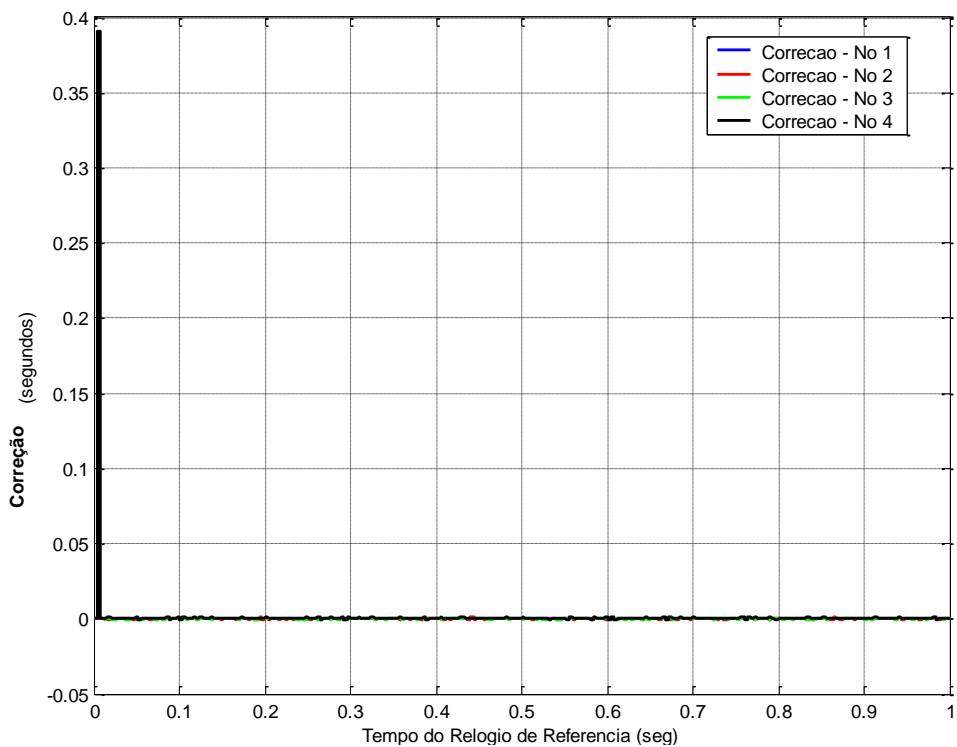


Figura 4.37 – Correção com Filtro de Kalman, Viés: 0.39 (s) e Deriva: 10% - AF.

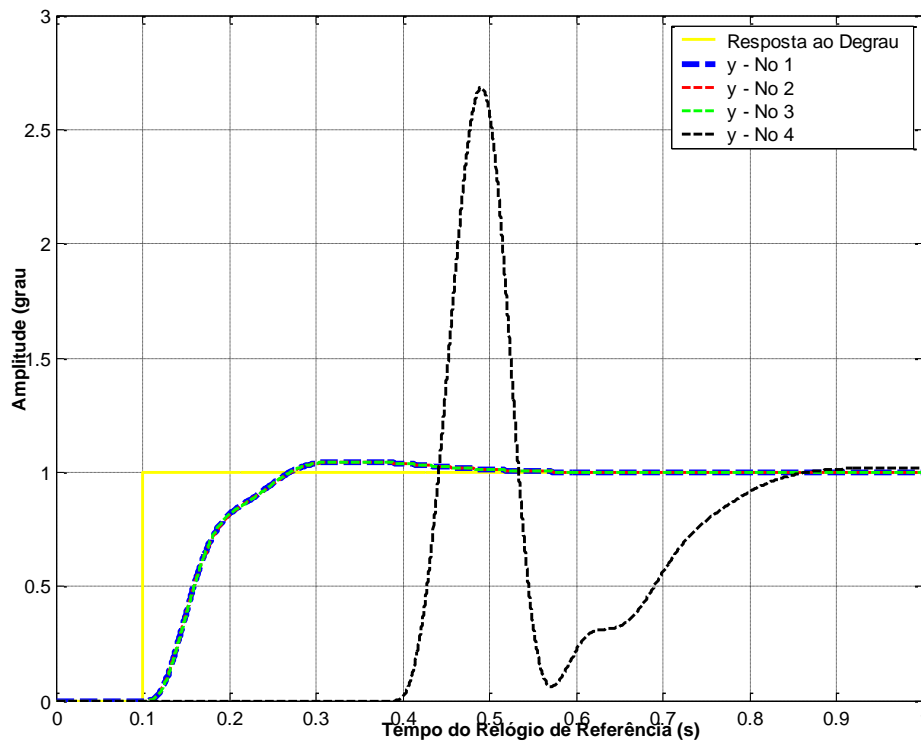


Figura 4.38 – Resposta ao Degrau com viés inicial de 0.39 (s) e Deriva: 10% - Sincronizado - AF.

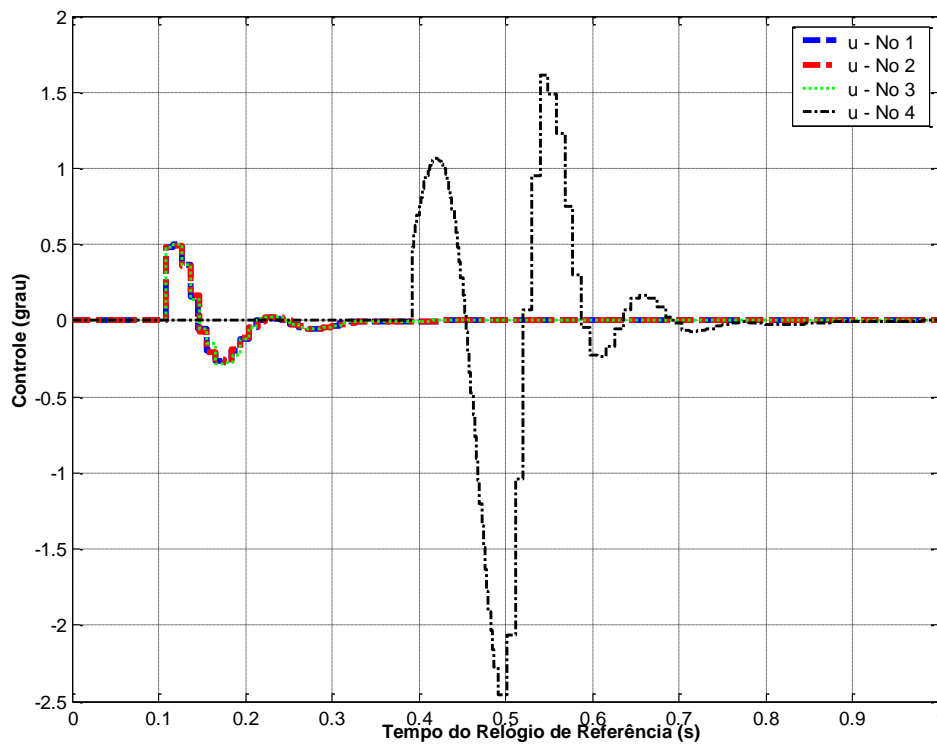


Figura 4.39 – Lei de Controle: Viés inicial de 0.39(seg) e Deriva:10% - Sincronizado - AF.

### 4.3.Sincronização de Relógios em Sistemas de Controle sobre uma Arquitetura Integrada

Neste capítulo é feita a simulação do sistema de controle distribuído com uma arquitetura Integrada. Foi feita uma simulação com 2 subsistemas de controle dividindo o mesmo barramento de dados. Os sensores, atuadores e controladores estão conectados pelo barramento de comunicação. Os protocolos de comunicação utilizados são o Ethernet e TDMA.

#### 4.3.1. Projeto dos Experimentos

O Projeto dos Experimentos realizados está sumarizado na Tabela 4.5 abaixo:

Tabela 4.5 – Tabela de Casos do Experimento com Arquitetura Integrada.

	Num. de Nós (n)	Num. de Relógios com Falha (f)	Algoritmo de Sincronização	Rede
<b>1º Caso</b>	4	Sensor 1 com erro em deriva	FTM (Welch-Lynch)	TDMA
<b>2º Caso</b>	4	Sensor 1 com erro em viés inicial	FTM (Welch-Lynch)	TDMA
<b>3º Caso</b>	4	Sensor 1 com erro de viés Inicial e Deriva	FTM (Welch-Lynch)	TDMA
<b>4º Caso</b>	4	Sensor 1 com erro em deriva	FTM (Welch-Lynch)	Ethernet
<b>5º Caso</b>	4	Sensor 1 com erro em viés inicial	FTM (Welch-Lynch)	Ethernet
<b>6º Caso</b>	4	Sensor 1 com erro de viés Inicial e Deriva	FTM (Welch-Lynch)	Ethernet
<b>Outros Casos</b>	Apêndices J, K, L, M, N e O.			

Esta simulação foi realizada com o Matlab/Simulink, com o apoio do *TrueTime ToolBox*. Nesta simulação foram implementados 2 subsistemas de controle por rede, isto é, o sensor, atuador e controlador estão conectados a um barramento de comunicação. A lei de controle PID foi implementada em um computador de tempo real simulado pelo *TrueTime ToolBox*. Foram implementados dois conjuntos de controle por rede integrados ao mesmo barramento de comunicação, como na Figura 3.13. Dois protocolos de comunicação foram usados, um com TDMA e o outro com Ethernet.

Na implementação TDMA, a arquitetura de controle integrada por rede sem efeitos de erro de relógio, isto é, todos os relógios são ideais, possui a resposta ao degrau vista na Figura 4.40. A lei de controle é vista na Figura 4.41. Como pode-se ver, as duas malhas de controle da rede, com protocolo de comunicação TDMA, possuem as respostas ao degrau e a lei de controle muito próximas. Na implementação Ethernet, a arquitetura de controle por rede integrada sem efeitos de erro de relógio, isto é, todos os relógios são ideais, possui a seguinte resposta ao degrau, dada pela Figura 4.42, e a lei de controle dada pela Figura 4.43. Como pode-se ver, as duas malhas de controle da rede, com um protocolo de comunicação Ethernet, possuem quase a mesma resposta ao degrau e lei de controle. Na seqüência serão inseridos erros do relógio no sensor 1 da malha de controle 1, e em seguida será usado o algoritmo FTM (*Welch-Lynch*) para corrigir o erro de relógio. Este caso será analisado variando os 2 protocolos de comunicação. Para **todos os casos** simulados, o **período de re-sincronização é de 0.0768 segundos** (este valor foi escolhido, pois bate com o valor do final de cada TDMA Round), isto é, a cada 0.0768 segundos o algoritmo FTM atualiza os valores de relógio com a correção calculada.

A tabela 4.6 mostra os resultados dos dados simulados para cada caso. Na seqüência serão mostrados os 6 casos simulados, indicados na Tabela 4.5,

seus gráficos e uma análise sobre os dados obtidos. Outros casos são mostrados nos Apêndices.

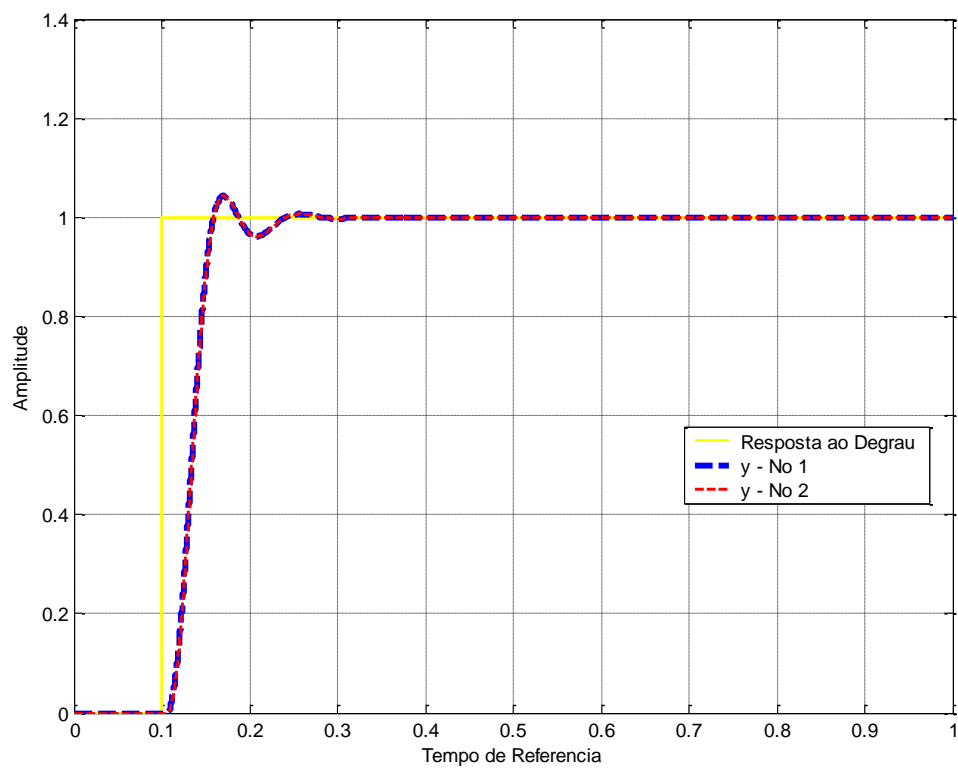


Figura 4.40 – Resposta ao Degrau – Arquitetura Integrada – TDMA.

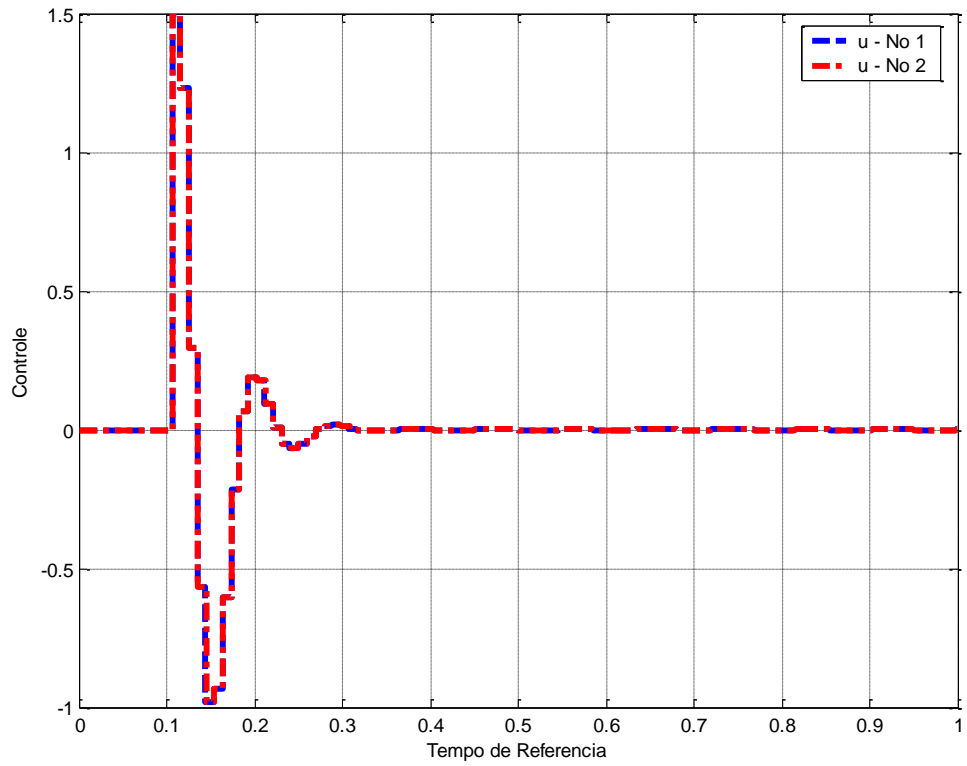


Figura 4.41 – Lei de Controle – Arquitetura Integrada - TDMA.

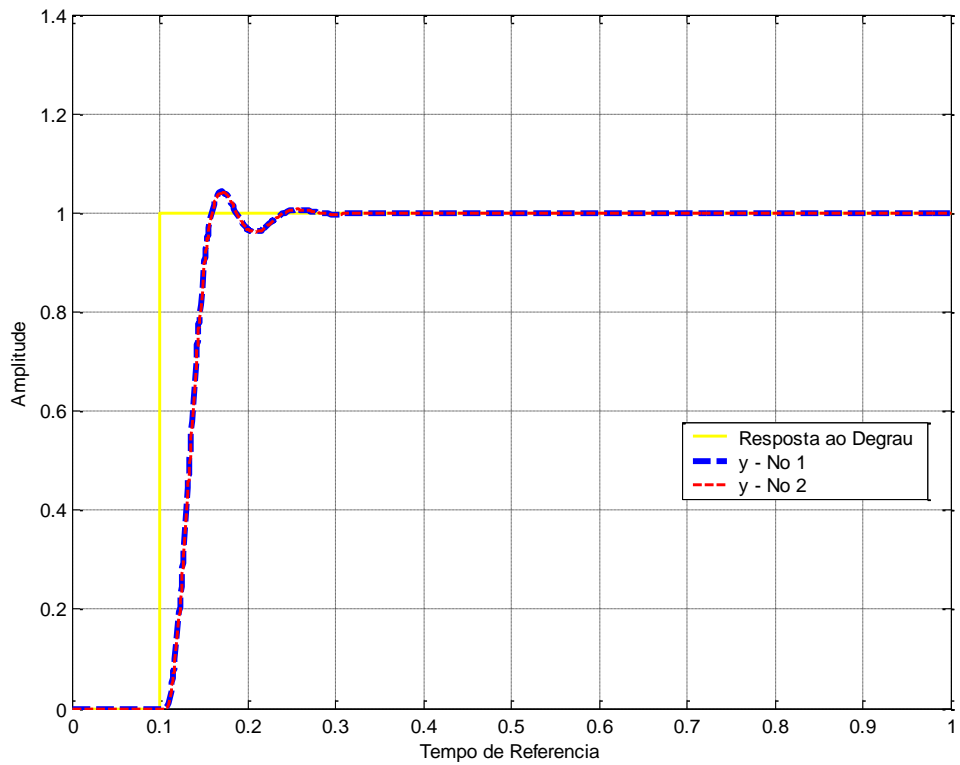


Figura 4.42 – Resposta ao Degrau – Arquitetura Integrada - Ethernet.

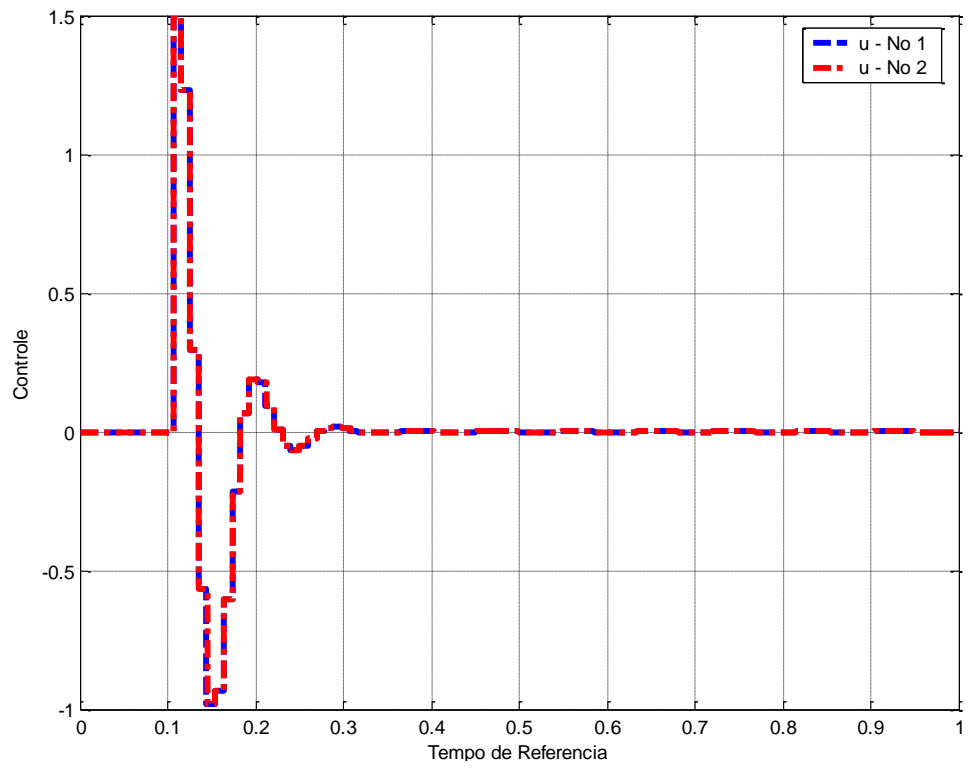


Figura 4.43 – Lei de Controle – Arquitetura Integrada - Ethernet.

Tabela 4.6 – Tabela de Resultados da Arquitetura Integrada.

Casos	Prot.	TD (%)	VI (s)	VP	O (%)	IP (s)	VP	O (%)	IP (s)	VP	(%)	IP (s)	VP	O (%)	IP (s)
				Malha 2 – Sem Sincronização			Malha 2 - Sincronizado			Malha 1 - Sem Sincronização			Malha 1- Sincronizado		
1º Caso (A)	TDMA	-10	0	1.0656	6.56%	0.1717	1.0874	8.73%	0.1703	1.3640	36.39	0.1643	1.2444	24.44	0.1837
1º Caso (B)	TDMA	1	0	1.0656	6.56%	0.1717	1.0874	8.74%	0.1703	1.1165	11.65	0.1689	1.0784	7.84	0.1701
1º Caso (C)	TDMA	10	0	1.0656	6.56%	0.1717	1.0874	8.74%	0.1703	1.2591	25.91	0.1701	1.2507	25.06	0.1773
2º Caso (A)	TDMA	0	-0.2	1.0656	6.56%	0.1717	1.0874	8.74%	0.1703	1.1245	12.45	0.2737	1.0470	4.7	0.1742
2º Caso (B)	TDMA	0	0.39	1.0656	6.56%	0.1717	1.0846	8.46%	0.1707	1.3776	37.76	0.1766	28.4681	2.7 10 <sup>3</sup>	0.4419
3º Caso (A)	TDMA	-10	-0.2	1.0656	6.56%	0.1717	1.0874	8.74%	0.1703	1.2154	21.54	0.2974	1.2307	23.07	0.1842
3º Caso (B)	TDMA	10	0.39	1.0656	6.56%	0.1717	1.0846	8.46%	0.1707	1.1449	14.49	0.1823	51.4431	5 10 <sup>3</sup>	0.4466
4º Caso (A)	Ethernet	-10	0	1.0656	6.56%	0.1717	1.0760	7.59%	0.1708	1.3640	36.39	0.1643	1.23	23	0.1842
4º Caso (B)	Ethernet	1	0	1.0656	6.56%	0.1717	1.0715	7.15%	0.1715	1.1165	11.65	0.1689	1.1365	13.65	0.1791
4º Caso (C)	Ethernet	10	0	1.0656	6.56%	0.1717	1.0721	7.21%	0.1715	1.2591	25.91	0.1701	1.1152	11.52	0.1839
5º Caso (A)	Ethernet	0	-0.2	1.0656	6.56%	0.1717	1.0743	7.43%	0.1711	1.1245	12.45	0.2737	1.0461	4.61	0.1761
5º Caso (B)	Ethernet	0	0.39	1.0656	6.56%	0.1717	1.0737	7.37%	0.1708	1.3776	37.76	0.1766	1.0593	5.93	0.4678
6º Caso (A)	Ethernet	-10	-0.2	1.0656	6.56%	0.1717	1.0760	7.59%	0.1708	1.2154	21.54	0.2974	1.2172	21.72	0.1842
6º Caso (B)	Ethernet	10	0.39	1.0656	6.56%	0.1717	1.0737	7.37%	0.1708	1.1449	14.49	0.1823	1.1165	11.64	0.4535



#### 4.3.2. Caso 1 – Relógio com Deriva - TDMA

Neste caso, foi feita uma simulação em um sistema de controle com arquitetura integrada, descrito na seção 3.3. A sincronização foi feita através do **algoritmo FTM (Welch-Lynch)**, com **arquitetura distribuída**, isto é, não existe um relógio de referência. Nesta simulação variou-se o **parâmetro de deriva do relógio do nó 2 (Sensor da Malha 1 de Controle por Rede)**. A rede de comunicação foi feita por um protocolo de comunicação **TDMA**. O objetivo deste caso é: 1) mostrar a influência que a deriva de relógio tem sobre o sistema de controle com AI; 2) utilizar o algoritmo FTM para corrigir a deriva do relógio e analisar a influência e a sincronização no sistema de controle.

Este trabalho escolheu 3 taxas de deriva diferentes para variação do relógio:

- **Caso A: -10%** - O relógio do sensor 1 está se atrasando em relação aos outros relógios do conjunto com taxa de 10%;
- **Caso B: 1%** - O relógio do sensor 1 está se adiantando em relação aos outros relógios do conjunto com taxa de 1%;
- **Caso C: 10%** - O relógio do sensor 1 está se adiantando em relação aos outros relógios do conjunto com taxa de 10%;

Os **casos A e C** são apresentados no **Apêndice J**. No **caso B**, a uma **taxa de deriva de 1%**, o **relógio do nó 2, sensor 1**, está se adiantando em relação ao relógio ideal e aos outros relógios do conjunto, como mostra a Figura 4.44. Com isto, o sistema de controle 1 é afetado pelo relógio e sofre uma variação em sua resposta ao degrau e na lei de controle. A Figura 4.45 mostra a resposta ao degrau do sistema de controle com a deriva de 1% no sensor 2. Essa variação no sensor 1 prejudica o sistema de controle 1, causando um aumento na porcentagem de sobre-elevação, mudança no valor de pico e instante de pico. Isto ocorre pois as suas medidas do sensor 1 chegam adiantadas ao controle 1. A Figura 4.46 mostra as leis de controle das duas

malhas integradas na rede de comunicação. O sistema de controle por rede 1 (azul) sofre um impacto com uma taxa de deriva de 1% no relógio do sensor 1. O que mostra que mesmo com uma deriva de relógio baixa, diferentemente da arquitetura federada, nesta arquitetura integrada o sistema é mais sensível à erros de relógio. Para reduzir/eliminar este impacto, aplicamos o **algoritmo FTM** para sincronizar os relógios. A Figura 4.47 mostra a linha do tempo com o algoritmo de sincronização FTM aplicado ao sistema, onde vemos que os relógios dos nós se sincronizaram entre si. Na Figura 4.48 é possível ver o gráfico de correção dos relógios dos respectivos nós do sistema, onde vemos que o algoritmo FTM convergiu. Os nós 1, 3 e 4 não estão com erro de deriva. Os relógios 1 e 4 não corrigiram fizeram nenhuma correção, o relógio 3 fez algumas pequenas correções em torno de 0.1 (ms) e -0.25 (ms). O nó 2, com 1% de taxa de deriva, teve uma correção periódica em torno de 0.2 (ms) e 0.5 (ms). Através da Figura 4.48 é possível constatar que a precisão dos relógios do sistema ficou em torno de 0.7 (ms). A Figura 4.49 mostra a resposta ao degrau do sistema de controle, e a Figura 4.50 mostra as leis de controle do sistema. Através das Figuras 4.49 e 4.50 é possível ver que a sincronização de relógios praticamente anulou o aumento da % de sobre elevação da resposta ao degrau e ao tempo de acomodação e praticamente anulou o atraso da lei de controle. As respostas ao degrau dos dois sistemas de controle são praticamente iguais. **Neste caso** o algoritmo **FTM** mostrou-se  **muito eficiente** corrigindo o erro de **deriva** presente no sistema. Na Tabela 4.6 temos, na linha do *1º Caso (B)*, os dados de controle de ambas as malhas.

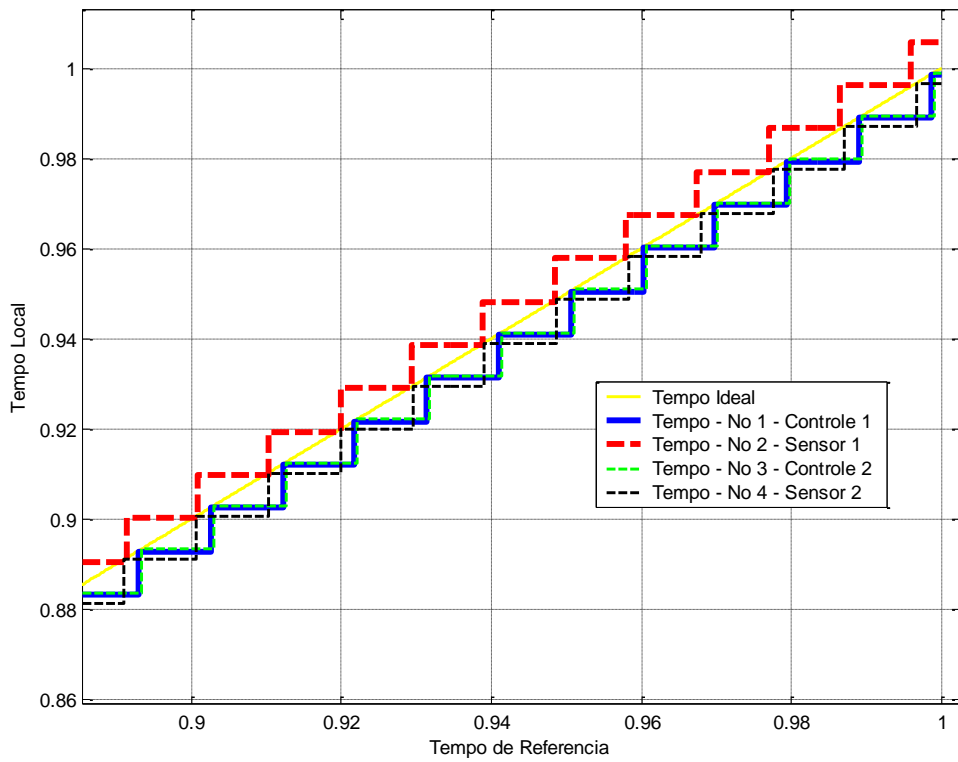


Figura 4.44 – Linha do Tempo com deriva de 1% - AI.

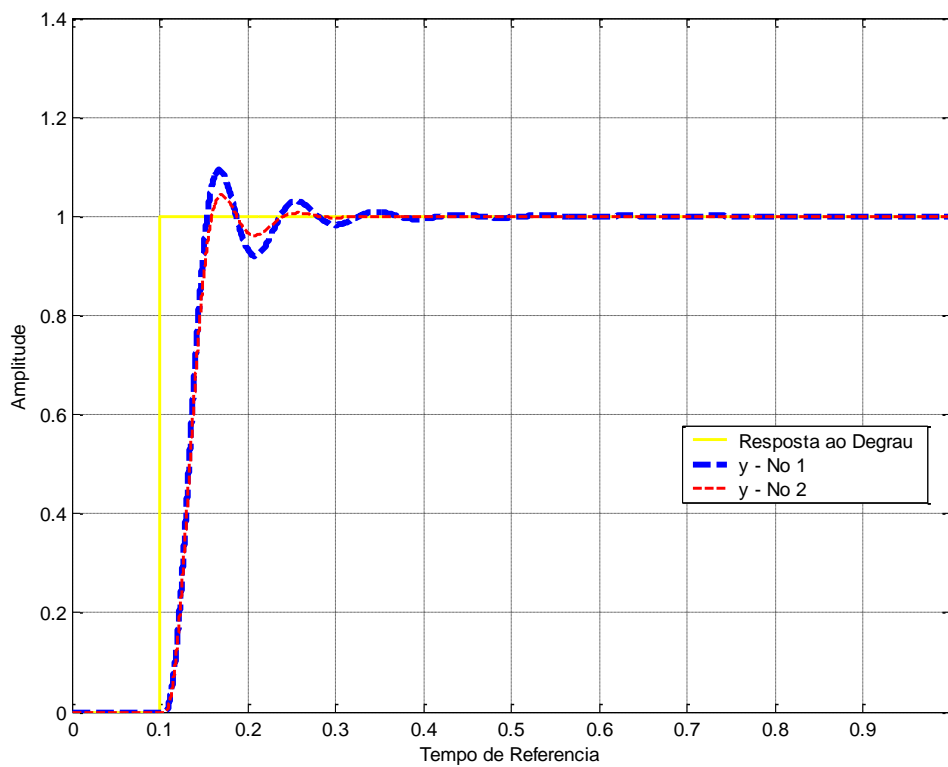


Figura 4.45 – Resposta ao Degrau com deriva de 1% - AI.

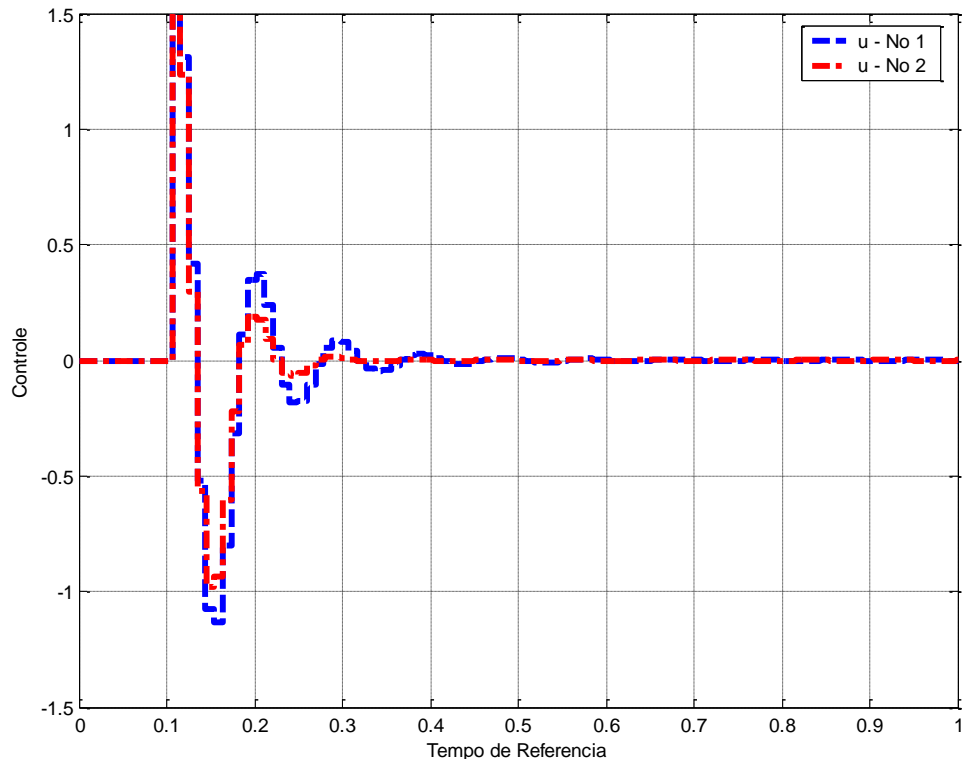


Figura 4.46 – Lei de Controle com deriva de 1% - AI.

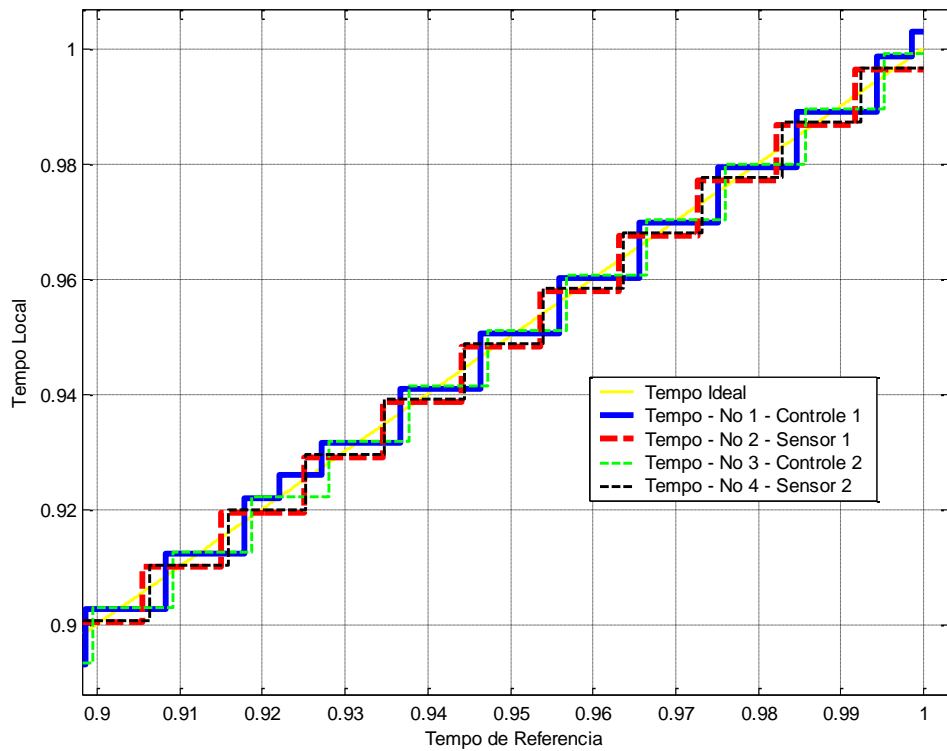


Figura 4.47 – Linha do Tempo Sincronizada com deriva de 1% - AI.

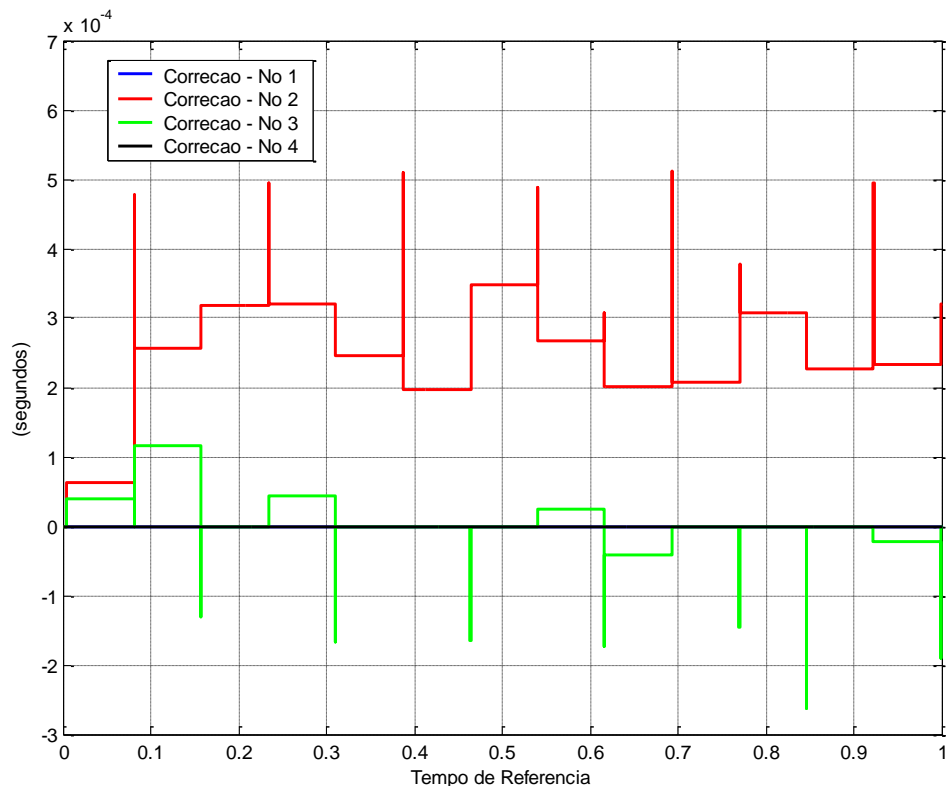


Figura 4.48 – Correção Algoritmo FTM com deriva de 1% - AI.

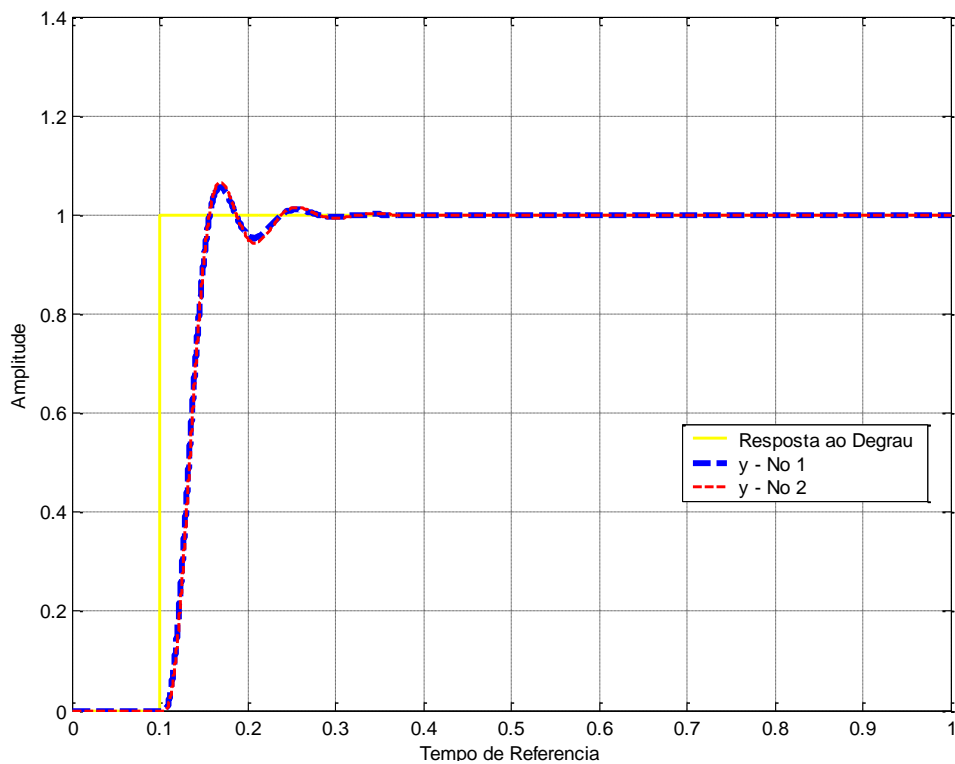


Figura 4.49 – Resposta ao Degrau com deriva de 1% - Sincronizado - AI.

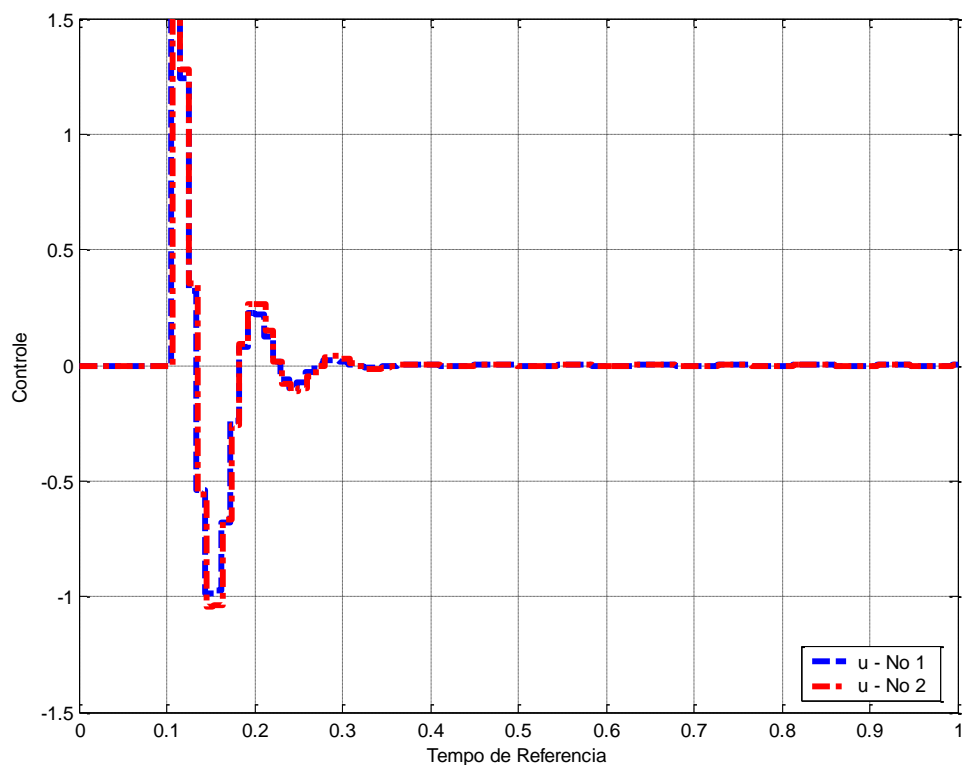


Figura 4.50 – Lei de Controle com deriva de 1% - Sincronizado - AI.

#### 4.3.3. Caso 2 – Relógio com Viés Inicial - TDMA

Utilizando a mesma simulação do caso 1, mas neste caso variou-se o **parâmetro de viés inicial** do relógio do **nó 2 (Sensor da Malha 1 de Controle por Rede)**. O objetivo deste caso é: 1) mostrar a influência que o viés inicial de relógio tem na sincronização de relógios e sobre o sistema de controle com AI; 2) utilizar o algoritmo FTM para corrigir o viés inicial do relógio e analisar a influência sobre a sincronização com o Algoritmo FTM e sobre o sistema de controle.

Este trabalho utilizou os 2 valores de viés inicial, utilizados na AF, para variação do relógio:

- **Caso A: -0.2 seg.** - O relógio do sensor 1 inicia atrasado de -0.2 seg. em relação aos outros relógios do conjunto;

- **Caso B: 0.39 seg.** - O relógio do sensor 1 inicia adiantado de 0.39 seg. em relação aos outros relógios do conjunto;

O **caso A** é apresentado no **Apêndice K**. No **caso B**, com um **viés inicial de 0.39 segundos**, o **relógio do nó 2, sensor 1**, inicia adiantado em relação ao relógio ideal e aos outros relógios do conjunto, como mostra a Figura 4.51. Com isso, o sistema de controle 1 sofre um atraso em sua resposta ao degrau e na lei de controle, pois é afetado pelo relógio. A Figura 4.52 mostra a resposta ao degrau do sistema de controle com o viés inicial de 0.39 seg. no sensor 2. É possível ver que esse adiantamento no sensor 1 prejudica muito o sistema de controle 1. Existe um aumento na % de sobre-elevação. Isto ocorre pois o relógio do sensor 2 está adiantado e com isso as suas medidas se adiantam ao controle, com isso o controle é feito com medidas atrasadas. A Figura 4.54 mostra as leis de controle das 2 malhas integradas na rede de comunicação. O sistema de controle por rede 1 (azul), Figura 4.52, sofre um impacto considerável com um viés inicial de 0.39 segundos no relógio do sensor 1. Para reduzir este impacto, vamos aplicar o algoritmo FTM para sincronizar os relógios. A Figura 4.54 mostra a linha do tempo com o algoritmo de sincronização FTM aplicado ao sistema. A Figura 4.54 mostra que os relógios dos nós se sincronizaram entre si. Na Figura 4.55, é possível ver o gráfico de correção dos relógios dos respectivos nós do sistema, ela ainda mostra que o algoritmo FTM convergiu. Os nós 1, 3 e 4 não estão com erro de viés inicial. Os relógios 1, 3 e 4 não corrigiram nada de seus relógios. O relógio 2, com 0.39 segundos de adiantamento inicial, teve uma correção grande em torno de 0.2 (s) e após essa correção manteve uma correção periódica praticamente nula. Com isso, é possível constatar através do gráfico 4.55 que a precisão dos relógios do sistema após a convergência ficou muito próxima de 0 (zero). A Figura 4.56 mostra a resposta ao degrau do sistema de controle. E a Figura 4.57 mostra as leis de controle do sistema. Através das Figuras 4.56 e 4.57 é possível ver que a sincronização de relógios prejudicou mais ainda o sistema de controle. O algoritmo FTM é feito principalmente para corrigir efeitos

de deriva e supõe que todos os relógios do conjunto estão sincronizados inicialmente. O erro ocorre pois o algoritmo FTM, ao aplicar sua correção, insere um **erro de estado** entre 0 e 0.39 segundos, isto é, o relógio fica parado, isto faz com que os relógios se sincronizem. Após esse período parado é que o sistema volta a atuar. Esta ação do algoritmo de parar o relógio para que os relógios se sincronizem, apesar de se mostrar eficiente ao sincronizar os relógios, acaba prejudicando o controle, tornando o sistema temporariamente instável durante o período sincronização. Neste caso o algoritmo FTM **não se mostrou eficiente**, corrigindo o **erro de viés inicial** presente no sistema. Na Tabela 4.6 temos, na linha do 2º Caso (B), os dados de controle de ambas as malhas. Nesta Tabela 4.6 é possível ver que a malha de controle 1 sincronizada tem uma % de sobre elevação de  $2,7 \cdot 10^3$  %.

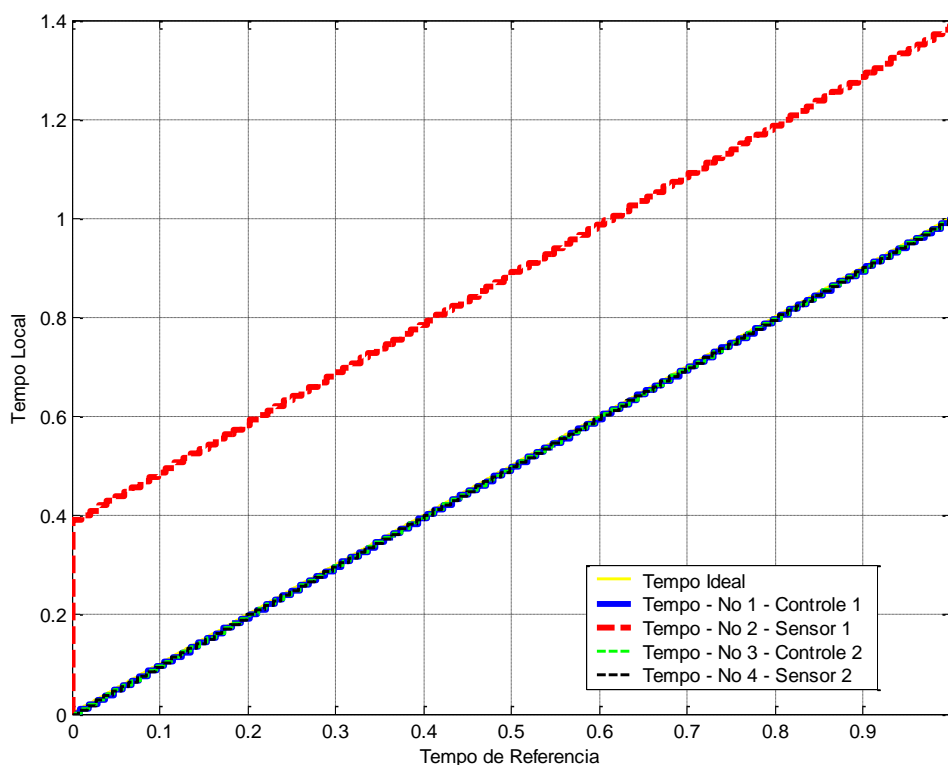


Figura 4.51 – Linha do Tempo com viés inicial de 0.39 segundos – A1.



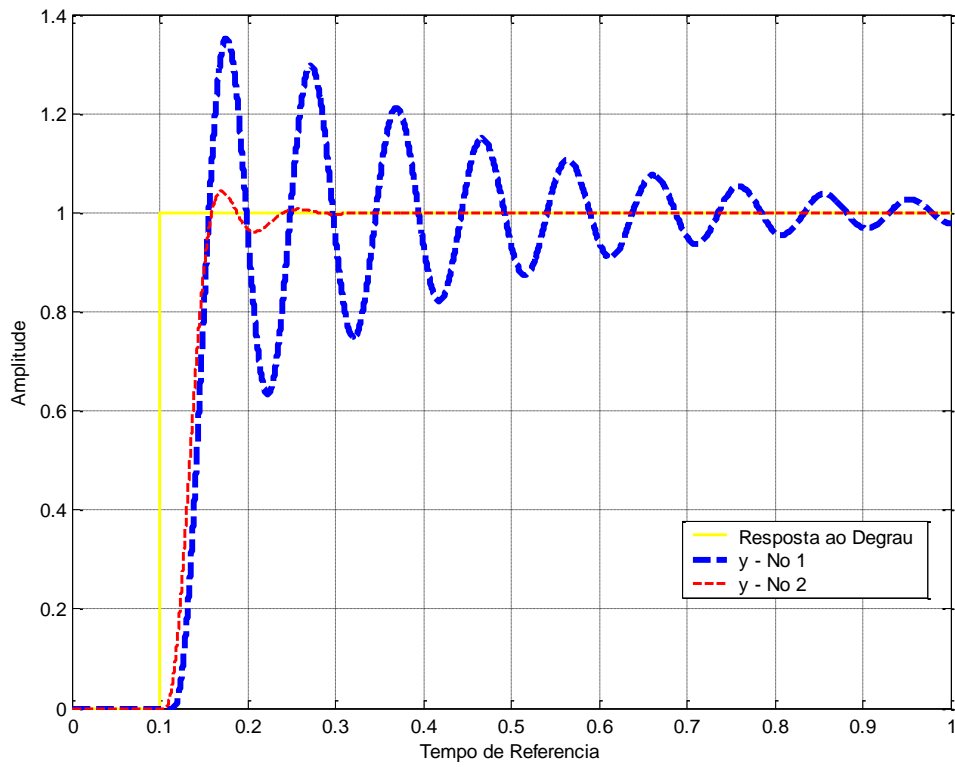


Figura 4.52 – Resposta ao Degrau com viés inicial de 0.39 segundos - AI.

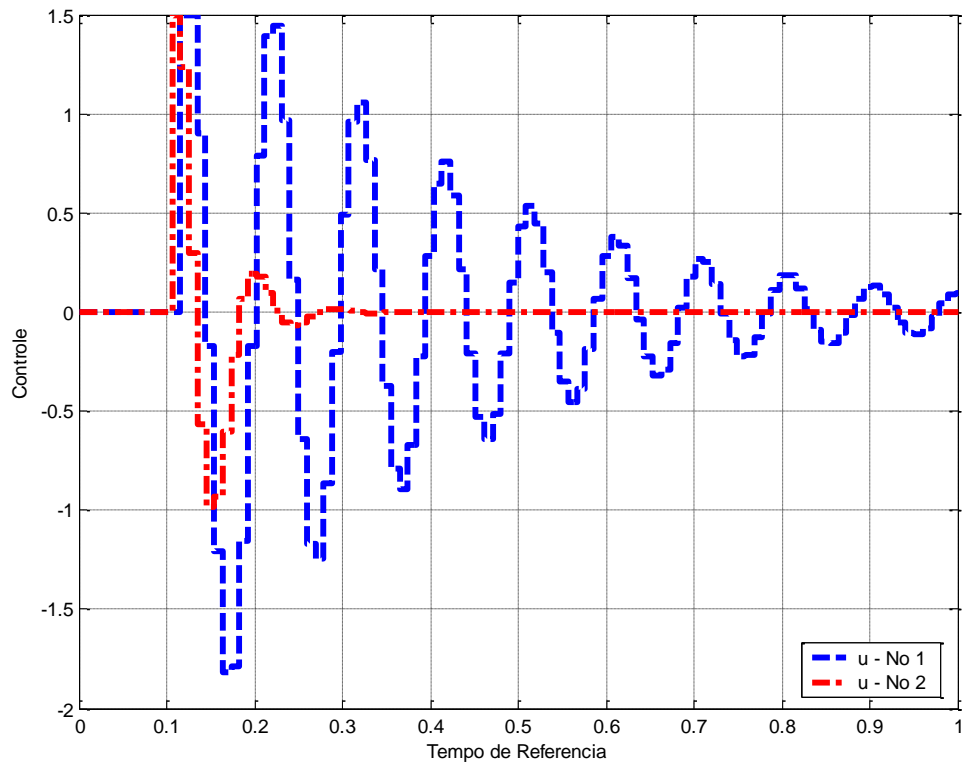


Figura 4.53 – Lei de Controle com viés inicial de 0.39 segundos - AI.

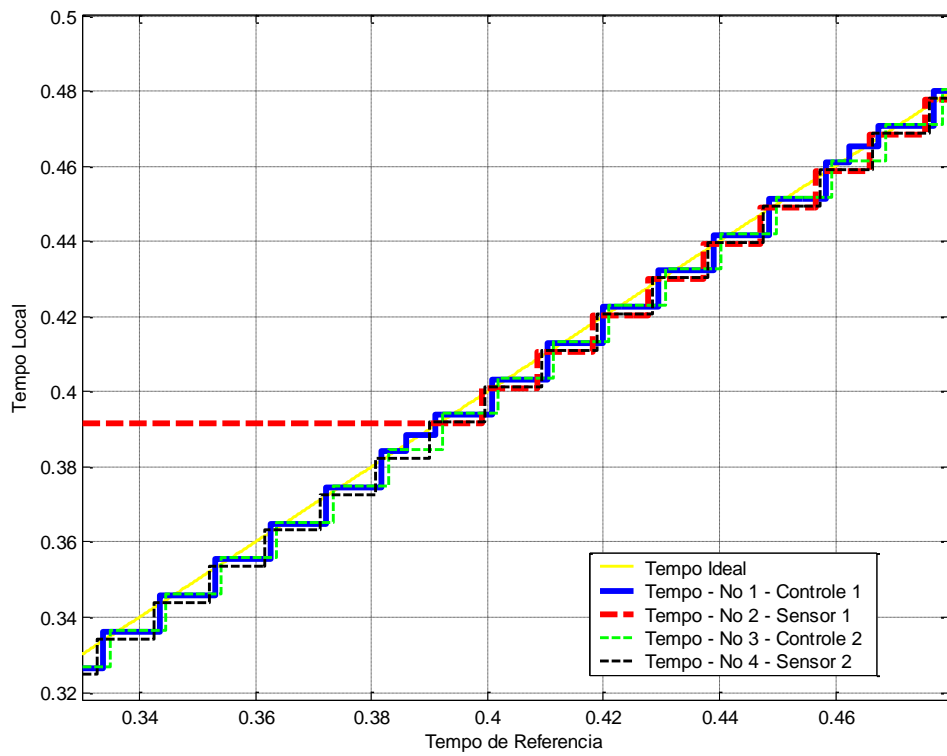


Figura 4.54 – Linha do Tempo Sincronizada: Viés de 0.39 s - AI.

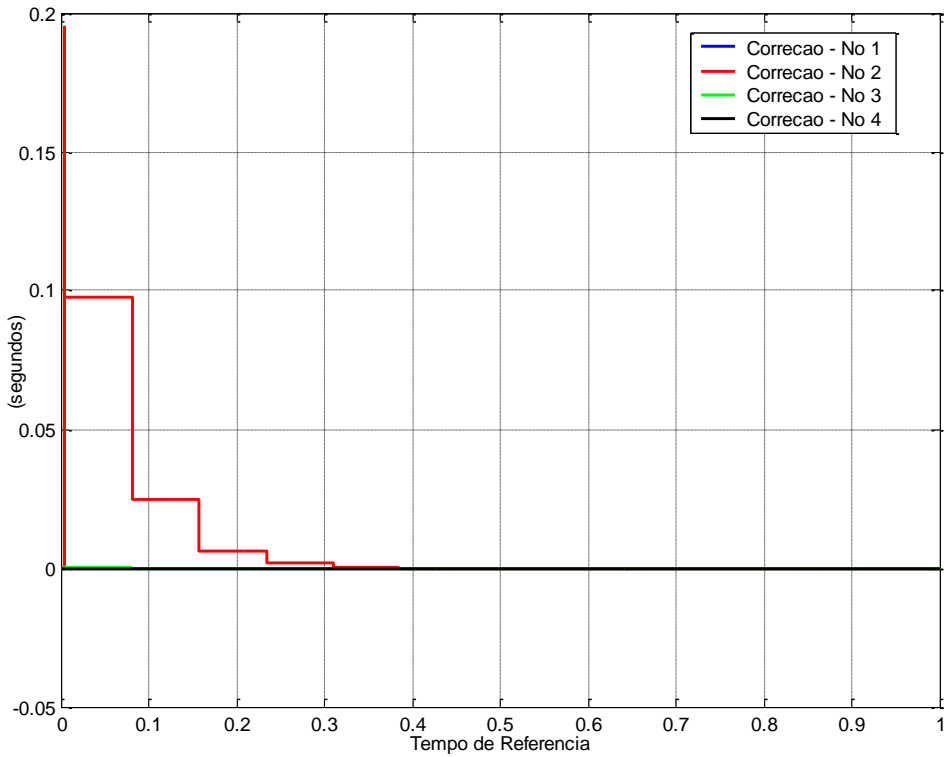


Figura 4.55 – Correção Algoritmo FTM com viés inicial de 0.39 s - AI.

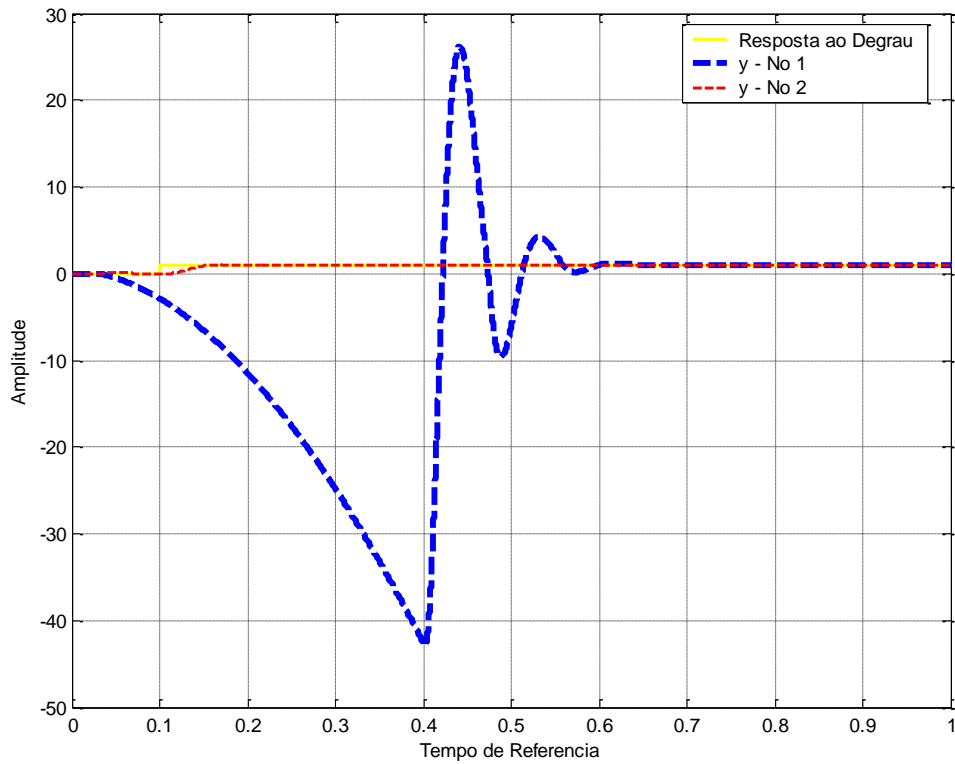


Figura 4.56 – Resposta ao Degrau com Viés de 0.39 s – AI - Sincronizado.

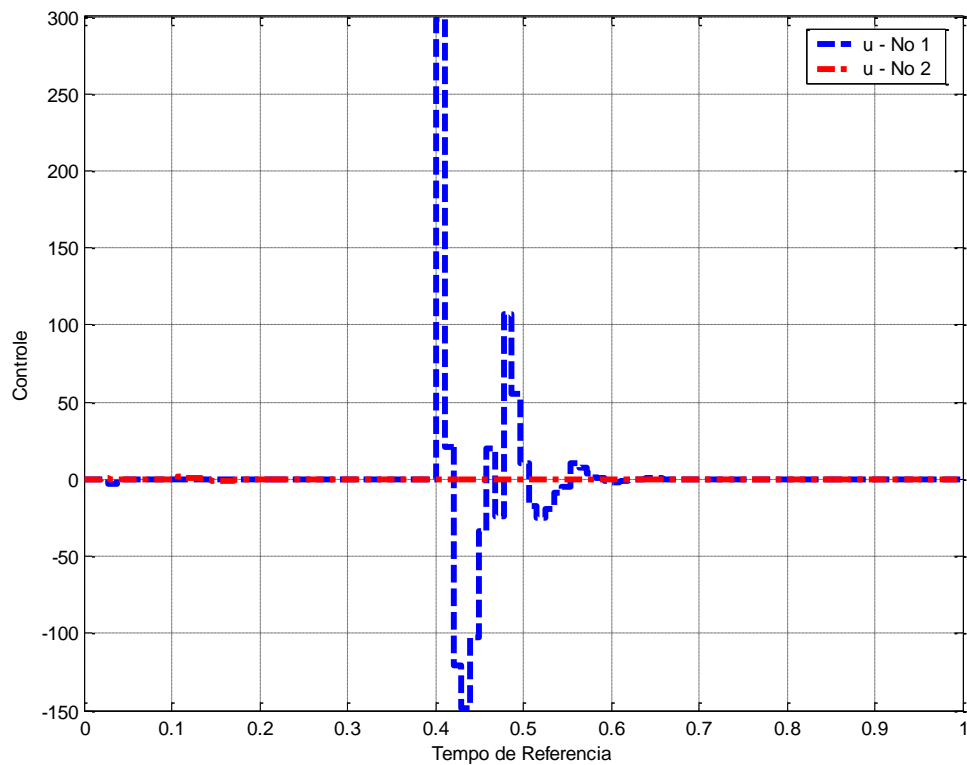


Figura 4.57 – Lei de Controle com Viés de 0.39 s – AI - Sincronizado.

#### 4.3.4. Caso 3 – Relógio com Deriva e Viés Inicial - TDMA

Utilizando a mesma simulação do caso 1, mas neste caso variou-se os **parâmetros de viés inicial e deriva** do relógio do **nó 2 (Sensor da Malha 1 de Controle por Rede)**. O objetivo deste caso é: 1) mostrar a influência que o viés inicial e a deriva de relógio tem na sincronização de relógios e sobre o sistema de controle com AI; 2) utilizar o algoritmo FTM para corrigir o viés inicial e a deriva do relógio e analisar a influência sobre a sincronização com o Algoritmo FTM e sobre o sistema de controle.

Este trabalho escolheu 2 valores de viés inicial diferentes para variação do relógio:

- **Caso A: Deriva: -10%, Viés inicial: -0.2 seg.** - O relógio do sensor 1 inicia atrasado de -0.2 seg. e ainda esta se atrasando em relação aos outros relógios do conjunto;
- **Caso B: Deriva: 10%, Viés Inicial: 0.39 seg.** - O relógio do sensor 1 inicia adiantado de 0.39 seg. e ainda esta se adiantando em relação aos outros relógios do conjunto;

O **caso A** é apresentado no **Apêndice I**. No **caso B**, com um **viés inicial de 0.39 segundos e deriva de 10%**, o **relógio do nó 2, sensor 1**, inicia adiantado e está se adiantando em relação ao relógio ideal e aos outros relógios do conjunto, como mostra a Figura 4.58. Com isso, o sistema de controle 1 sofre um adiantamento em sua resposta ao degrau e na lei de controle, pois é afetado pelo relógio. A Figura 4.59 mostra a resposta ao degrau do sistema de controle com o viés inicial de 0.39 seg. e deriva 10% no sensor 2. É possível ver que esse atraso no sensor 1 prejudica o sistema de controle 1. Existe um aumento na porcentagem de sobre-elevação e no instante de pico. Isto ocorre pois o relógio do sensor 2 inicia adiantado e esta se adiantando, e com isso as suas medidas são afetadas e conseqüentemente se adiantam à transmissão ao controle. A Figura 4.60 mostra as leis de controle

das 2 malhas integradas na rede de comunicação. Para reduzir este erro, vamos aplicar o algoritmo FTM para sincronizar os relógios. A Figura 4.61 mostra a linha do tempo com o algoritmo de sincronização FTM aplicado ao sistema. A Figura 4.61 mostra que os relógios dos nós se sincronizaram entre si. Na Figura 4.62, é possível ver o gráfico de correção dos relógios dos respectivos nós do sistema, e ainda mostra que o algoritmo FTM convergiu. Os nós 1, 3 e 4 não estão com erro de viés inicial e deriva. Os relógios 1, 3 e 4 não corrigiram nada de seus relógios. O nó 2 com 0.39 segundos de adiantamento inicial e taxa de deriva de 10%, manteve uma correção periódica em torno de 4 (ms). Através do gráfico 4.62 é possível constatar que a precisão dos relógios do sistema ficou em torno de 4 (ms). A Figura 4.63 mostra a resposta ao degrau do sistema de controle. E a Figura 4.64 mostra as leis de controle do sistema. Através das Figuras 4.63 e 4.64, assim como no Caso 2, é possível ver que a sincronização de relógios prejudicou mais ainda o sistema de controle. Pois além de ter o efeito de viés inicial o controle ainda possui o efeito da deriva. Assim como no caso 2, o erro ocorre pois o algoritmo FTM, ao aplicar sua correção, insere um **erro de estado** entre 0 e 0.39 segundos, isto é, o relógio fica parado, isto faz com que os relógios se sincronizem. Após esse período parado é que o sistema volta a atuar. Esta ação do algoritmo de parar o relógio para que os relógios se sincronizem, apesar de se mostrar eficiente ao sincronizar os relógios, acaba prejudicando o controle, tornando o sistema temporariamente instável durante o período sincronização. Neste caso o algoritmo FTM **não se mostrou eficiente**, corrigindo o **erro de viés inicial e deriva** presente no sistema. Na Tabela 4.6 temos, na linha do 3º Caso (B), os dados de controle de ambas as malhas. Nesta Tabela 4.6 é possível ver que a malha de controle 1 sincronizada tem uma % de sobre elevação de  $5 \times 10^3$  %, que é bem maior que o Caso 2.

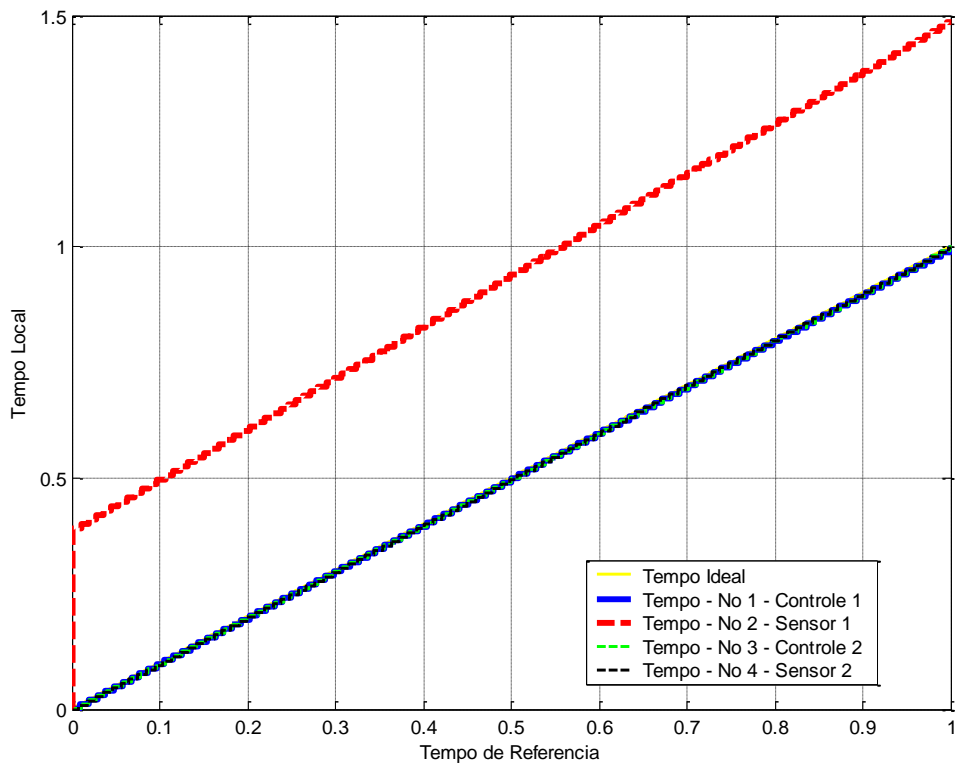


Figura 4.58 – Linha do Tempo com Viés inicial: 0.39 segundos, Deriva: 10% - AI.

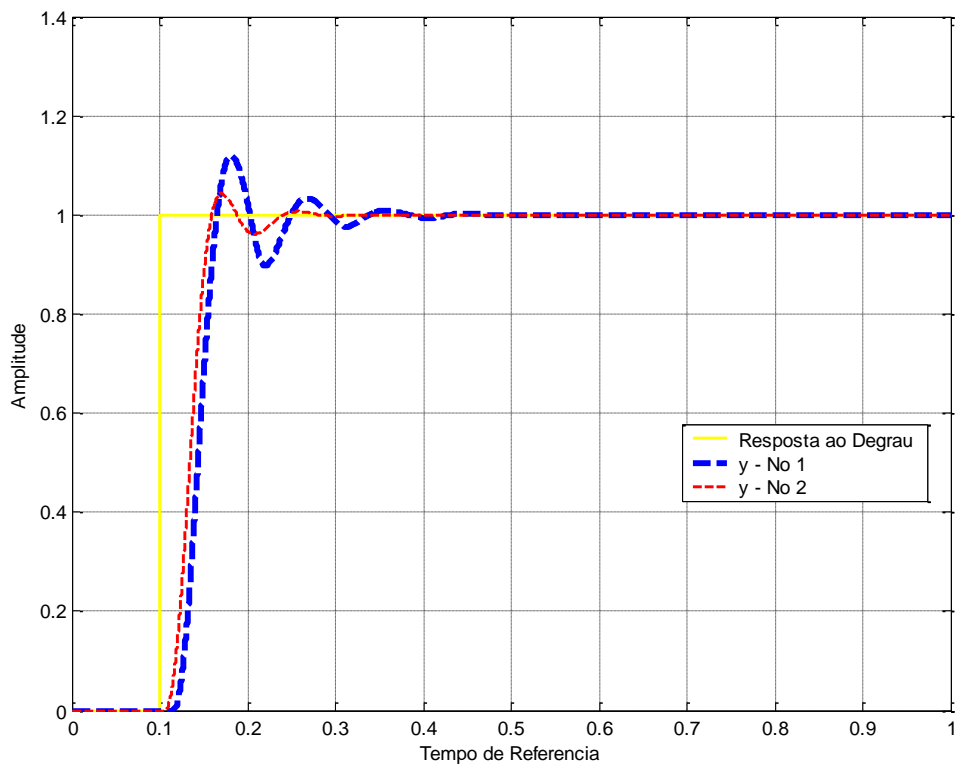


Figura 4.59 – Resposta ao Degrau: Viés inicial: 0.39 segundos, Deriva: 10% - AI.

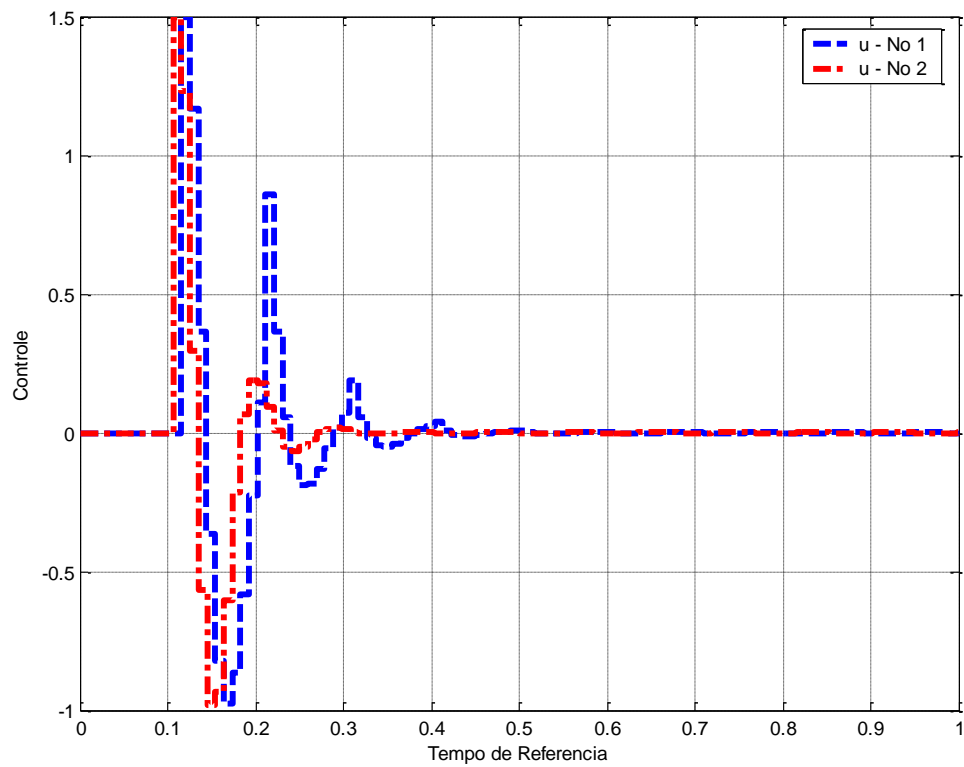


Figura 4.60 – Lei de Controle: Viés inicial: 0.39 segundos, Deriva: 10% - AI.

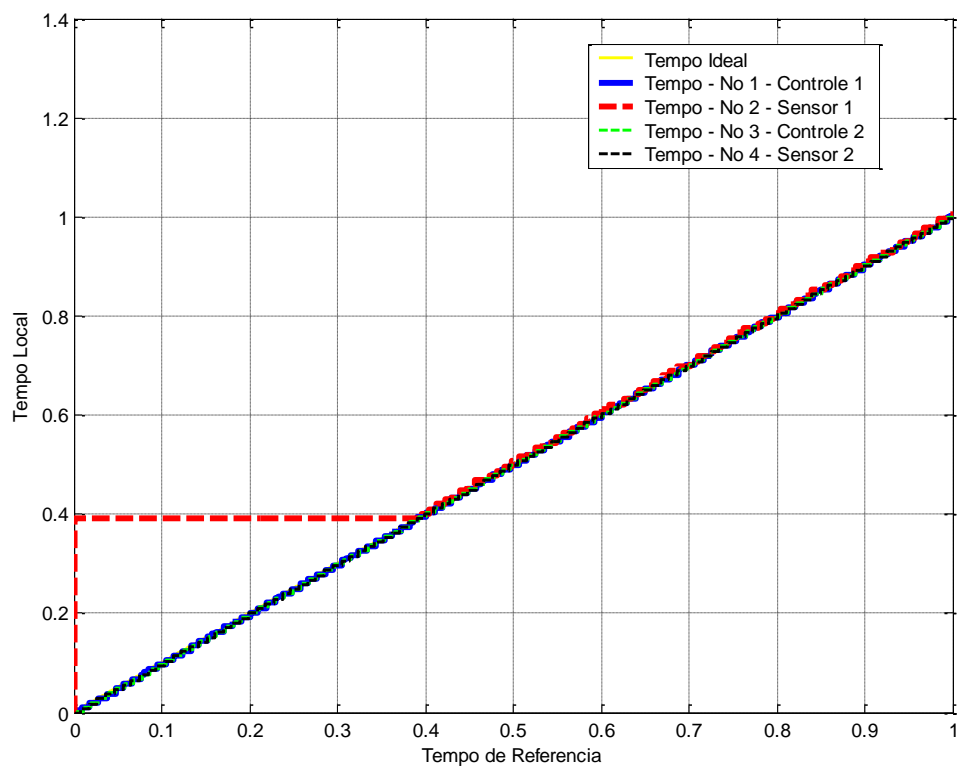


Figura 4.61 – Linha do Tempo Sincronizada: Viés inicial: 0.39(s), Deriva: 10% - AI.

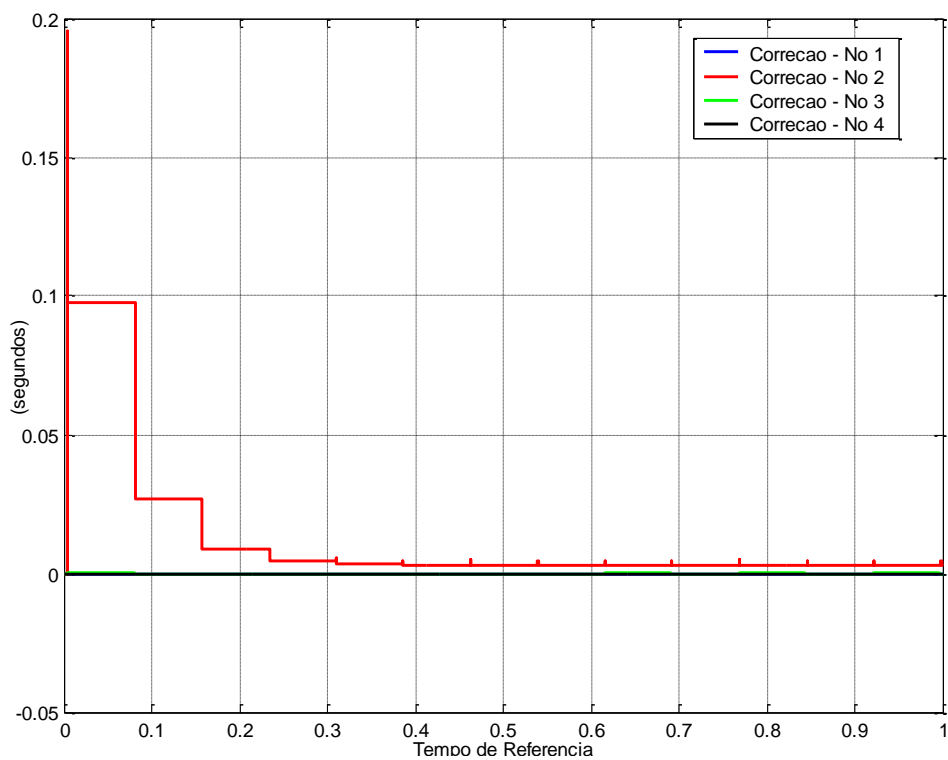


Figura 4.62 – Correção Algoritmo FTM: Viés inicial: 0.39 (s), Deriva: 10% - AI.

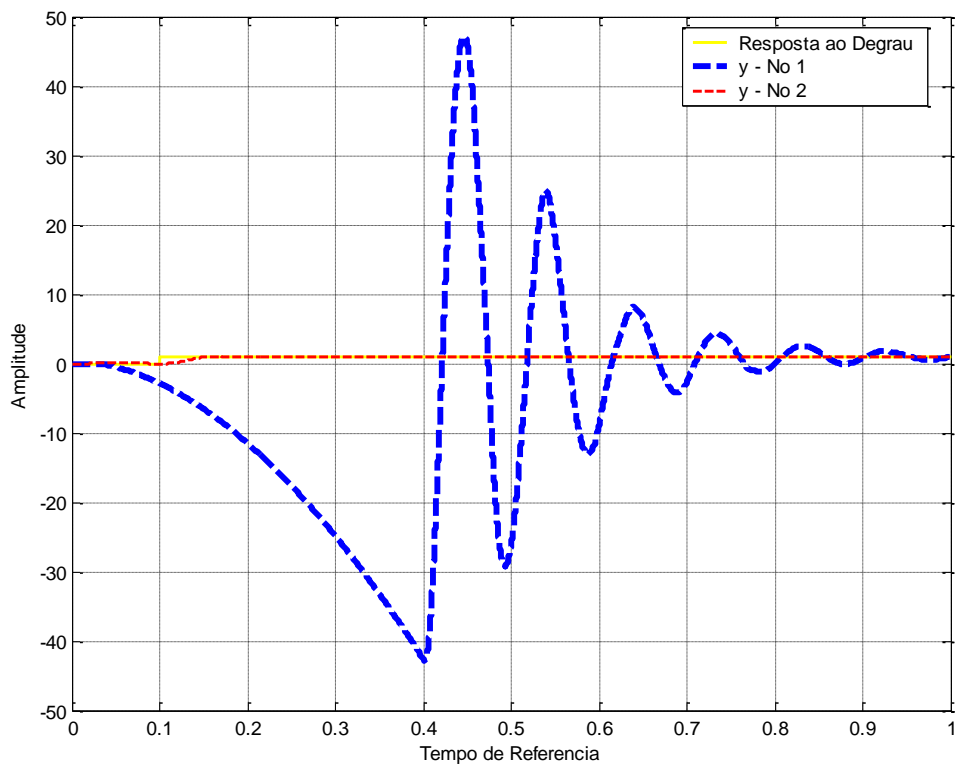


Figura 4.63 – Resposta ao Degrau Sincronizada: Viés inicial:0.39(s), Deriva:10% - AI.



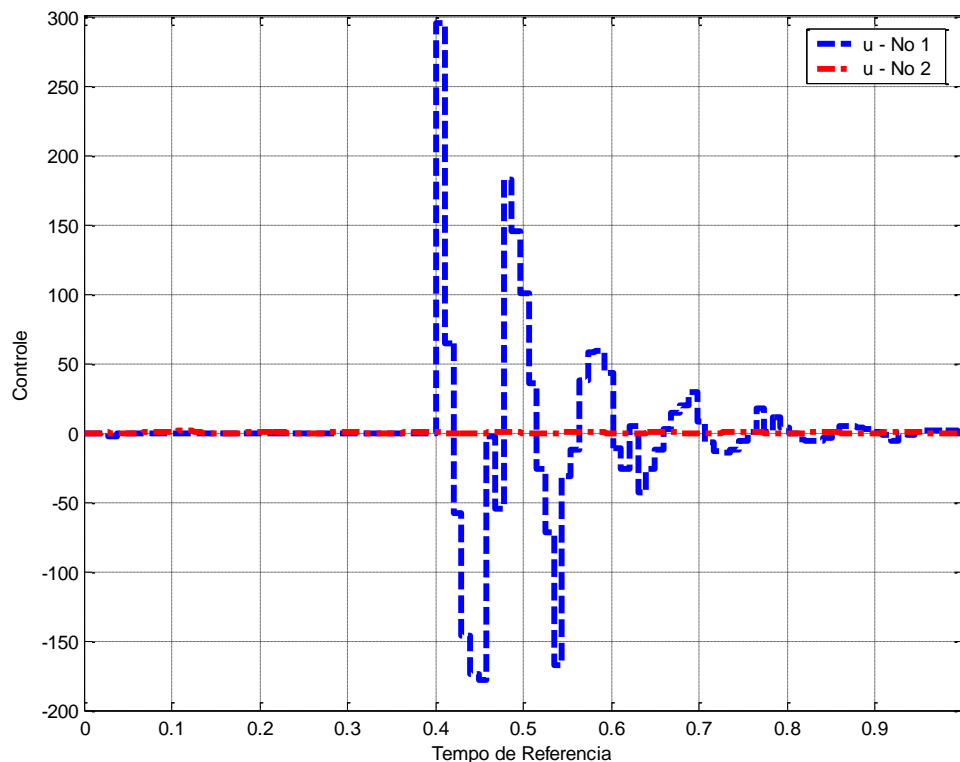
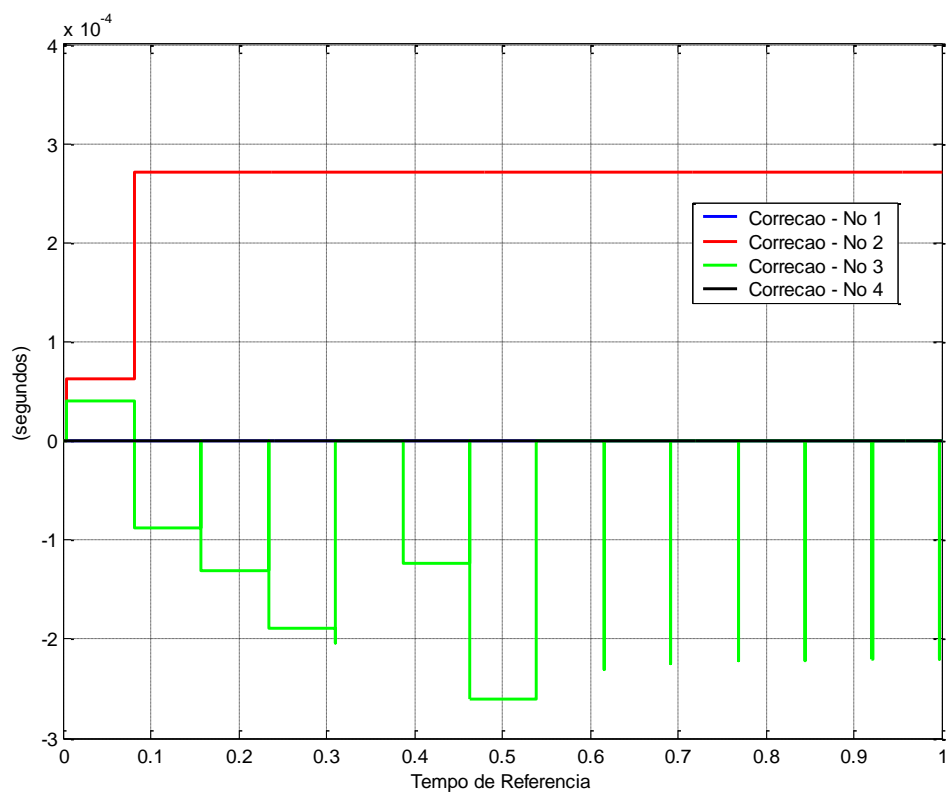


Figura 4.64 – Lei de Controle Sincronizada: Viés inicial: 0.39(s), Deriva: 10% - AI.

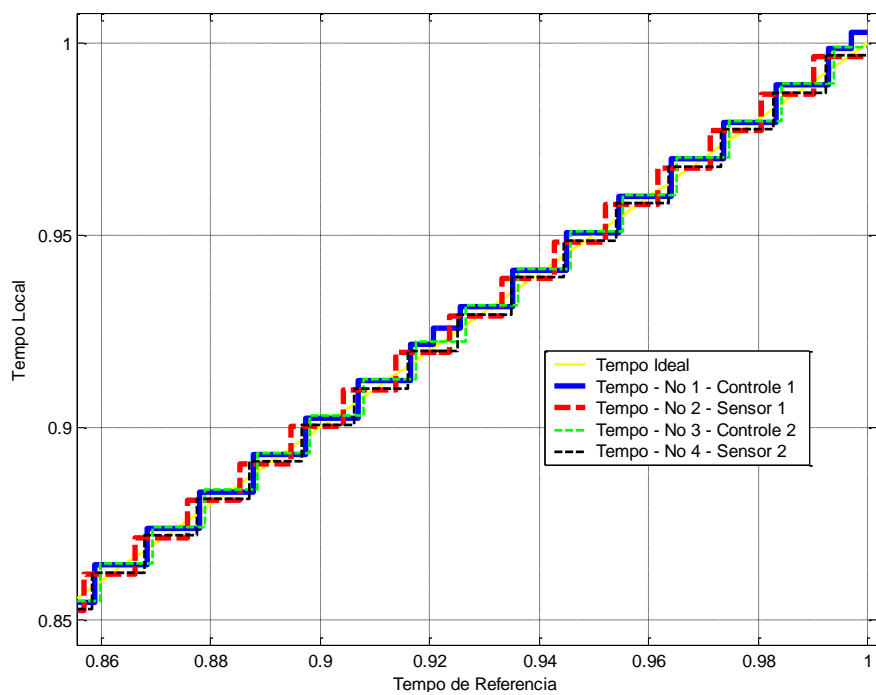
#### 4.3.5. Caso 4 – Relógio com Deriva - Ethernet

Neste caso, foi refeita a simulação de um sistema de controle com arquitetura integrada, igual ao 1º Caso. A sincronização foi feita através do algoritmo FTM (*Welch-Lynch*), com arquitetura distribuída, isto é, não existe um relógio de referência. Nesta simulação variou-se o **parâmetro de deriva** do relógio do **nó 2 (Sensor da Malha 1 de Controle por Rede)**. A rede de comunicação foi feita por um protocolo de comunicação com acesso ao meio **Ethernet**. A resposta ao degrau e Lei de controle ideais podem ser observadas na Figura 4.42 e 4.43. Foi realizado o mesmo experimento do 1º Caso.

Os **casos A e C** com **taxa de deriva com -10% e 10%** podem ser encontrados no **Apêndice M**. O **caso B**, com **taxa de deriva de 1%** os resultados da sincronização com Ethernet foram:

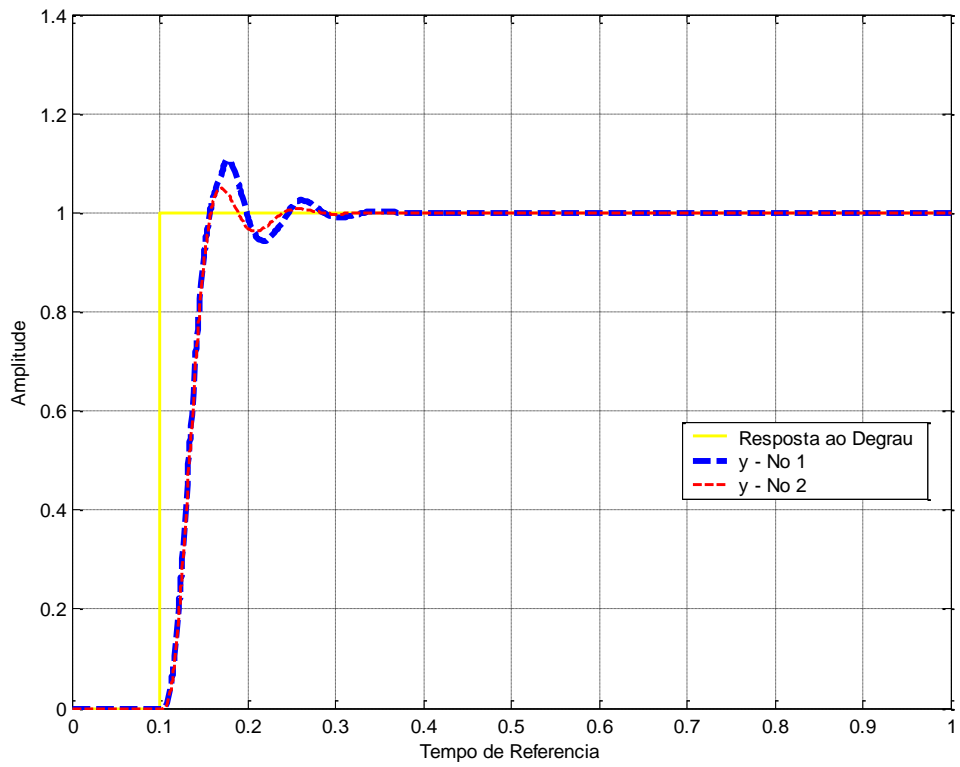


(a)

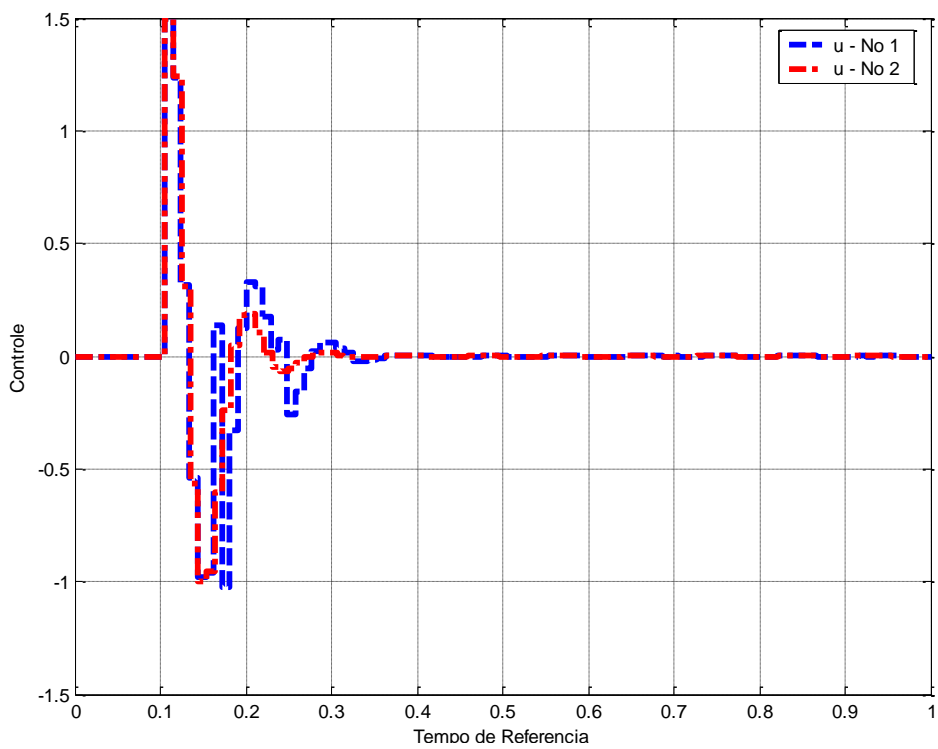


(b)

Figura 4.65 – (a) Correção: Rede Ethernet, Deriva 1% (b) Linha do Tempo - AI.



(a)



(b)

Figura 4.66 – (a) Resposta ao Degrau, Deriva 1% (b) Lei de Controle - AI.

#### 4.3.6. Caso 5 – Relógio com Viés Inicial - Ethernet

Neste caso, foi refeita a simulação do Caso 6. Nesta simulação variou-se o **parâmetro de viés inicial** do relógio do **nó 2 (Sensor da Malha 1 de Controle por Rede)**. No **caso B**, com o **viés inicial** de 0.39 segundos, o algoritmo FTM com rede Ethernet teve o mesmo desempenho, mas a lei de controle e a resposta ao degrau se modificaram. Na rede TDMA a resposta ao degrau e a lei de controle amplificaram suas amplitudes. Na rede Ethernet as amplitudes não se modificaram mas a resposta ao degrau e a lei de controle se atrasaram de 0.39 segundos. Isso mostra que o sistema com a rede Ethernet é menos suscetível a erros de relógio. O **caso A** com **viés inicial de -0,2 (s)** pode ser encontrado no **Apêndice N**. No caso onde o viés inicial é -0.2 segundos o algoritmo FTM com rede Ethernet teve um desempenho praticamente idêntico ao com rede TDMA. A Tabela 4.6 mostra os dados tabelados de controle. Com **viés inicial: 0.39(s)**, os resultados da sincronização com Ethernet foram:

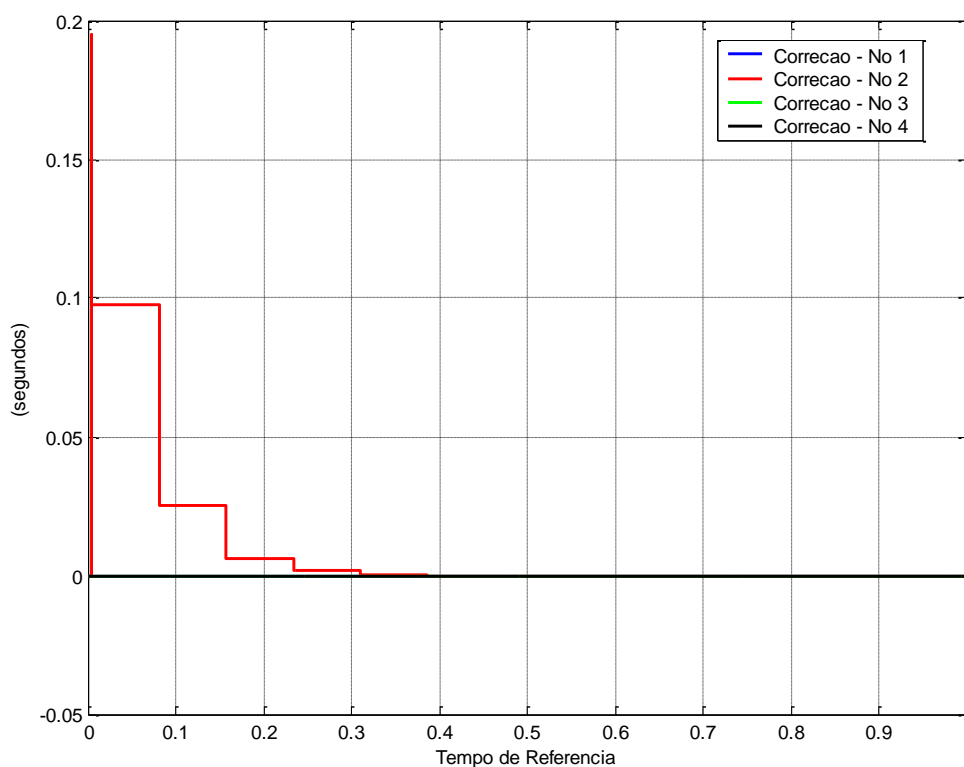


Figura 4.67 – Correção: Rede Ethernet, Viés 0.39s -Al.

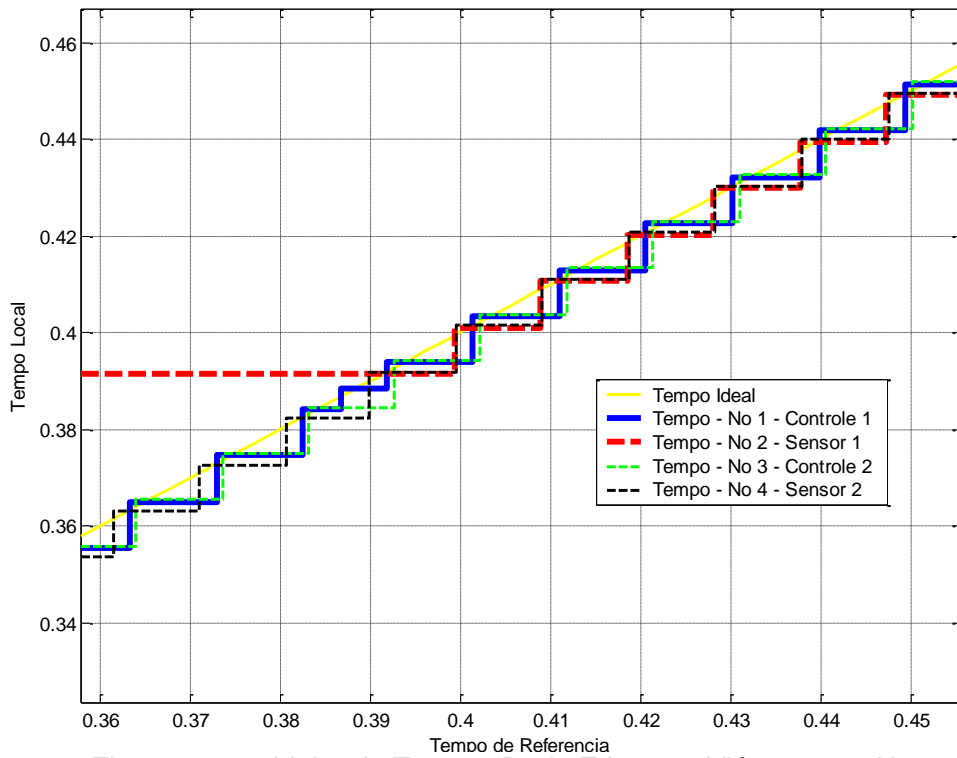


Figura 4.68 – Linha do Tempo: Rede Ethernet, Viés 0.39s - AI.

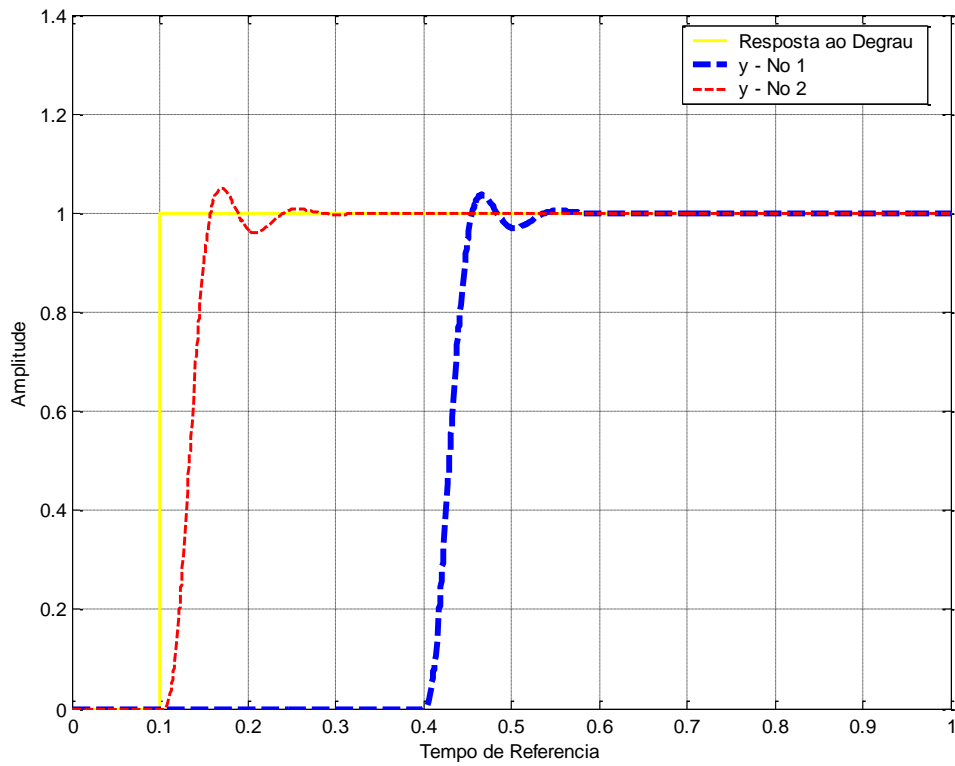


Figura 4.69 – Resposta ao Degrau, Ethernet - Viés 0.39s - AI.

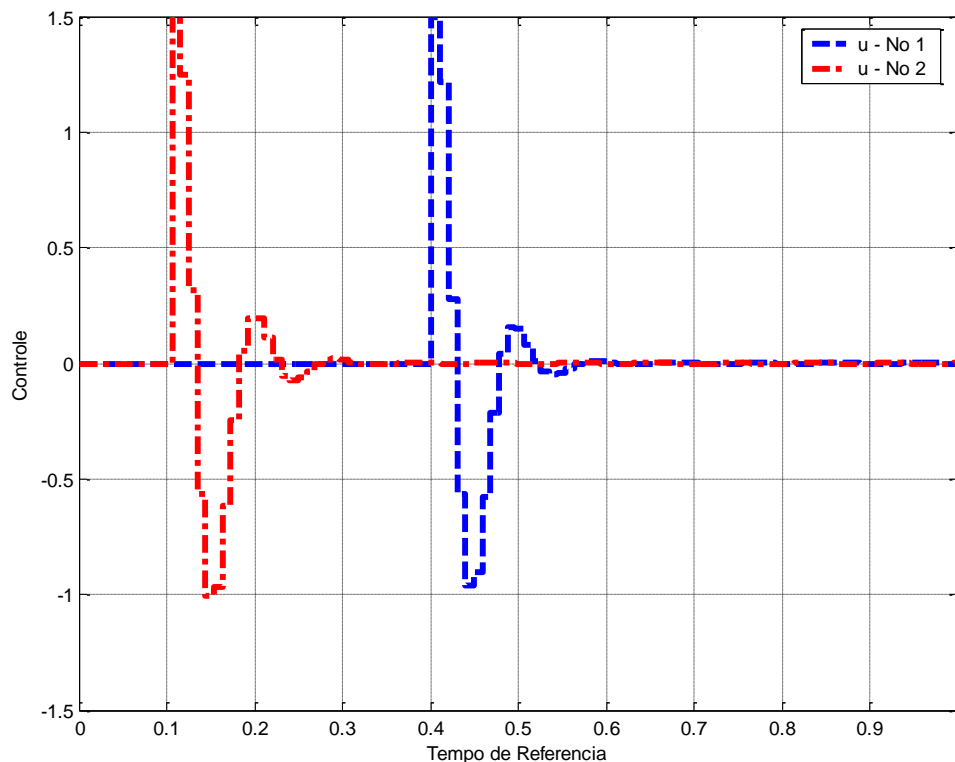


Figura 4.70 – Lei de Controle: Ethernet - Viés 0.39s - AI.

#### 4.3.7. Caso 6 – Relógio com Deriva e Viés Inicial - Ethernet

Neste caso, foi refeita a simulação do Caso 6. Nesta simulação variou-se o **parâmetro de viés inicial** do relógio do **nó 2 (Sensor da Malha 1 de Controle por Rede)**. No **caso B**, com o **viés inicial de 0.39 segundos e deriva 10%**, o algoritmo FTM com rede Ethernet teve o mesmo desempenho, mas a lei de controle e a resposta ao degrau modificaram. Na rede TDMA a resposta ao degrau e a lei de controle amplificaram suas amplitudes. Na rede Ethernet as amplitudes não se amplificaram tanto, mas a resposta ao degrau e a lei de controle se atrasaram de 0.39 segundos. A Tabela 4.6 mostra os dados tabelados de controle.

Com **viés inicial: 0.39(s) e Deriva 10%**, os resultados da sincronização com Ethernet foram:

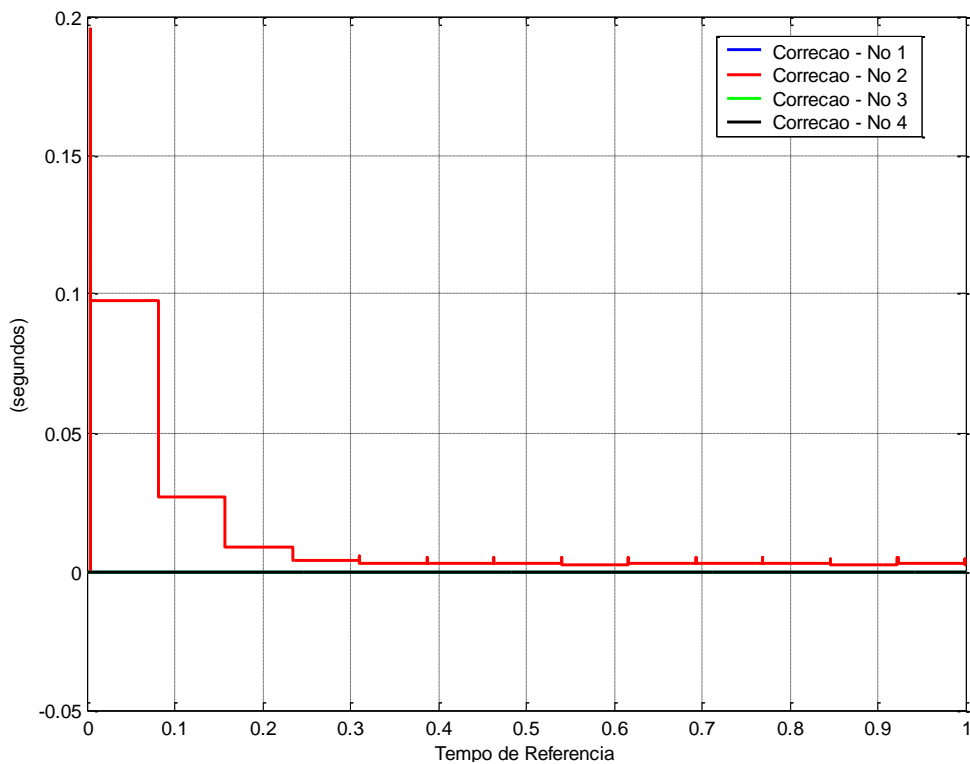


Figura 4.71 – Correção: Rede Ethernet, Viés 0.39s, Deriva: 10% -AI.

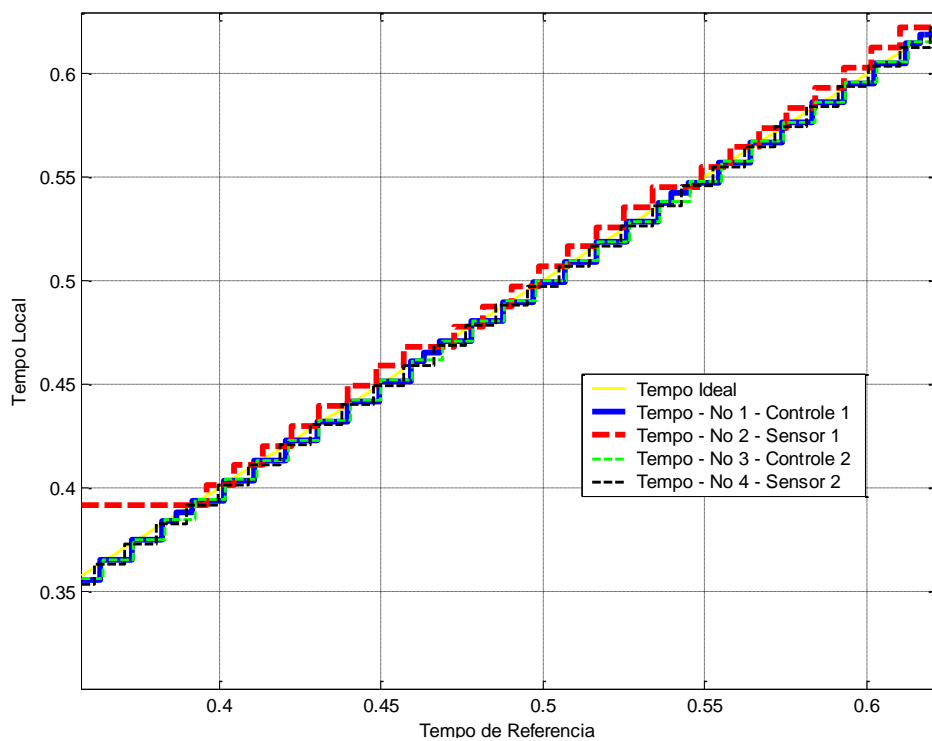


Figura 4.72 – Linha do Tempo: Rede Ethernet, Viés 0.39s, Deriva: 10% - AI.

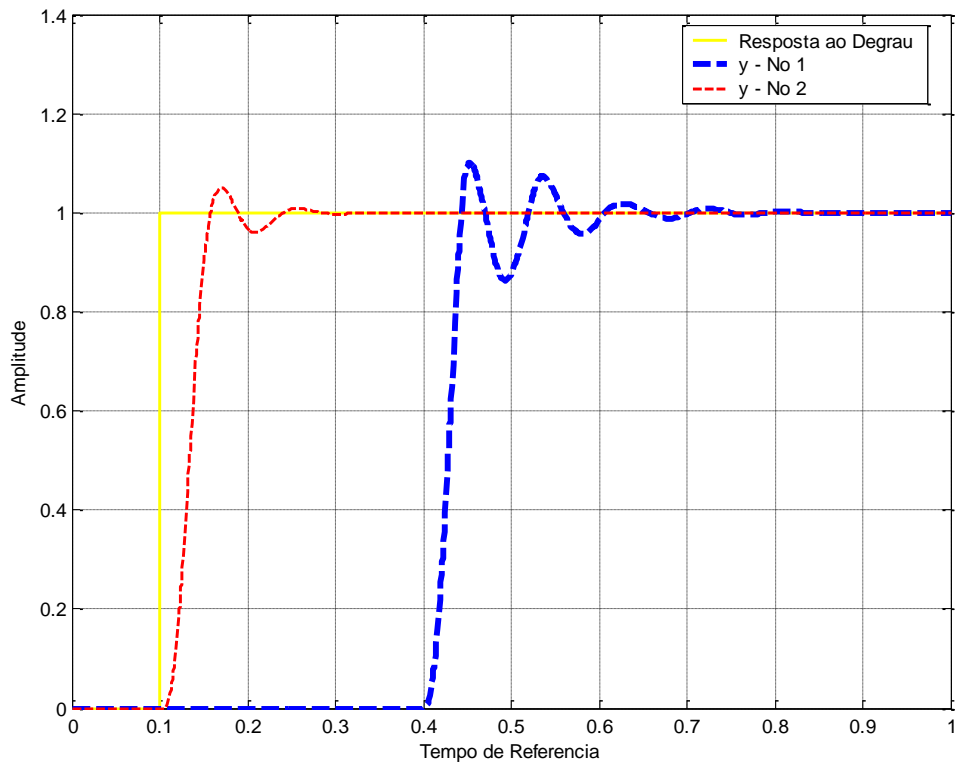


Figura 4.73 – Resposta ao Degrau, Ethernet - Viés 0.39s, Deriva: 10% - AI.

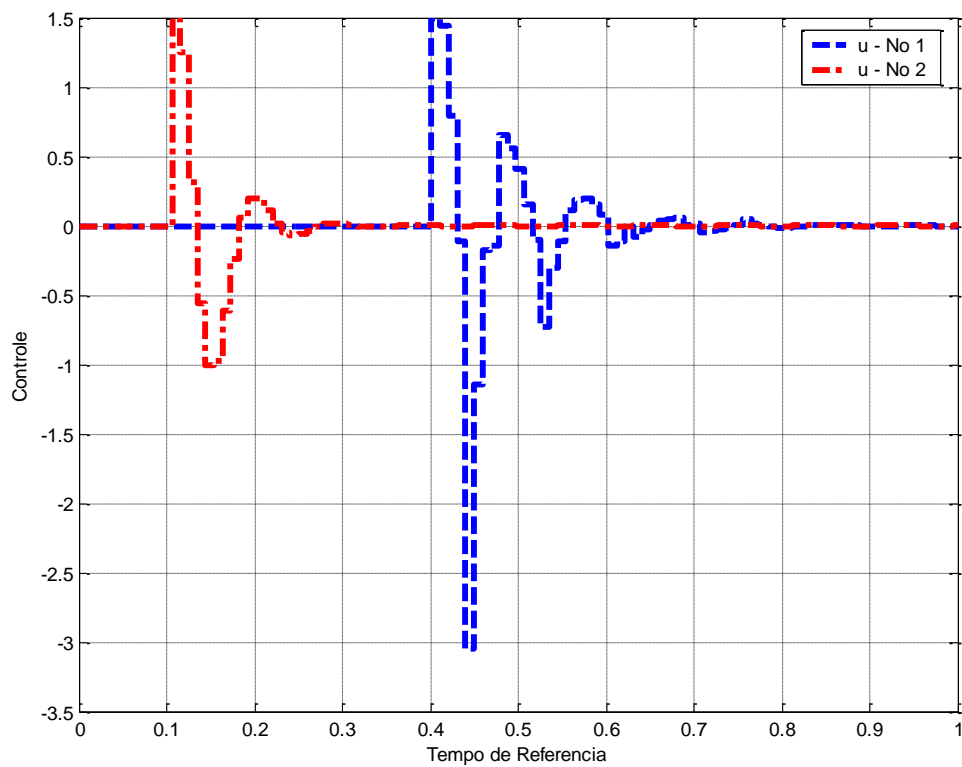


Figura 4.74 – Lei de Controle: Ethernet - Viés 0.39s, Deriva: 10% - AI.

#### 4.3.8. Caso 3 - TDMA versus Caso 6 - Ethernet



O caso 3 é um dos casos mais relevantes encontrados durante as simulações neste trabalho. A Figura 4.63 mostra que ao sincronizar os relógios do sistema com arquitetura integrada com um protocolo de comunicação TDMA, utilizando o algoritmo de sincronização FTM, a sincronização de relógios afeta as leis de controle. Já no caso 6, com um protocolo de comunicação Ethernet a sincronização de relógios com o algoritmo FTM não torna o sistema de controle temporariamente instável.

No caso 3, este fator de instabilidade temporária ocorre pois na simulação com o *TrueTime Toolbox* ao iniciar o sensor 1 com viés inicial positivo, i.e. quando o relógio do sensor está adiantado, conforme mostra a Figura 4.58, o computador virtual do sensor 1 identifica que está com tarefas atrasadas e com isso começa a realizar todas as tarefas atrasadas de uma só vez. A tarefa do sensor 1 é medir e enviar os dados via barramento de comunicação ao controle 1. Ao enviar os dados via barramento de comunicação TDMA, o volume de tarefas é muito maior que o espaço de tempo que o sensor 1 possui no barramento TDMA. Com isso as tarefas do sensor 1 vão sendo suspensas quando o seu espaço de tempo é superado. Esta suspensão de tarefas gera um atraso de aproximadamente 0.02 segundos. A Figura 4.75 mostra o agendador da rede de comunicação TDMA, onde é possível observar este fenômeno. Este atraso gerado pela rede de comunicação TDMA é suficientemente grande para que o sistema de controle invertesse a fase. Ao inverter a fase o sistema fica instável. A Figura 4.77 mostra a inversão de fase da Lei de controle e a Figura 4.78 mostra uma aproximação da Figura 4.63 no instante da inversão de fase. Como o algoritmo fez com que o relógio do sensor 1 parasse por aproximadamente 0.39 segundos, conforme visto na Figura 4.61, então a lei de controle durante este período sofre com a inversão de fase, fica instável e não recebe mais dados atualizados do sensor 1. Quando o relógio do sensor 1 volta a funcionar, os dados voltam a ser transmitidos normalmente ao controle 1 e com isso o sistema retorna a sua estabilidade.

Já no caso 6 com os parâmetros de relógio iguais ao caso 3 mas com uma rede de comunicação Ethernet a diferença para o caso 3 é que o barramento de comunicação Ethernet suporta o envio destas tarefas atrasadas acumuladas. Com isso, o atraso gerado pelo barramento é de 10 (ms), que não é suficiente para gerar uma inversão de fase. E com isso, apesar da malha 1 de controle se atrasar, conforme mostra a Figura 4.73, o sistema não sofre desta instabilidade temporária causada pelo barramento TDMA.

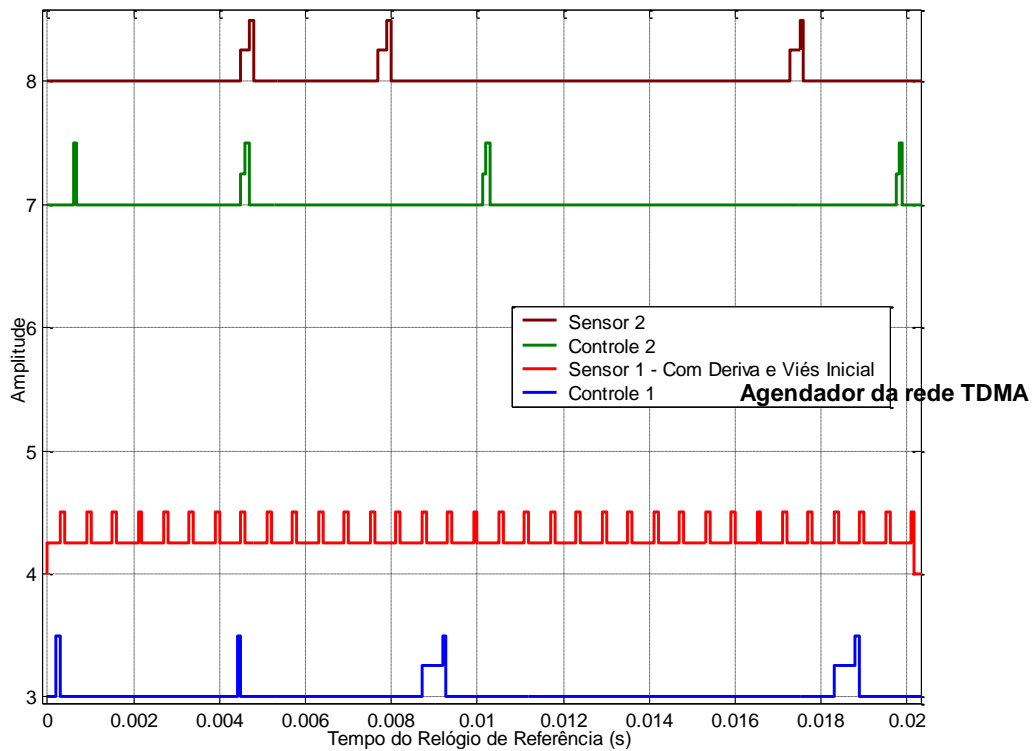


Figura 4.75 – Agendador da Rede TDMA Sincronizado - AI.

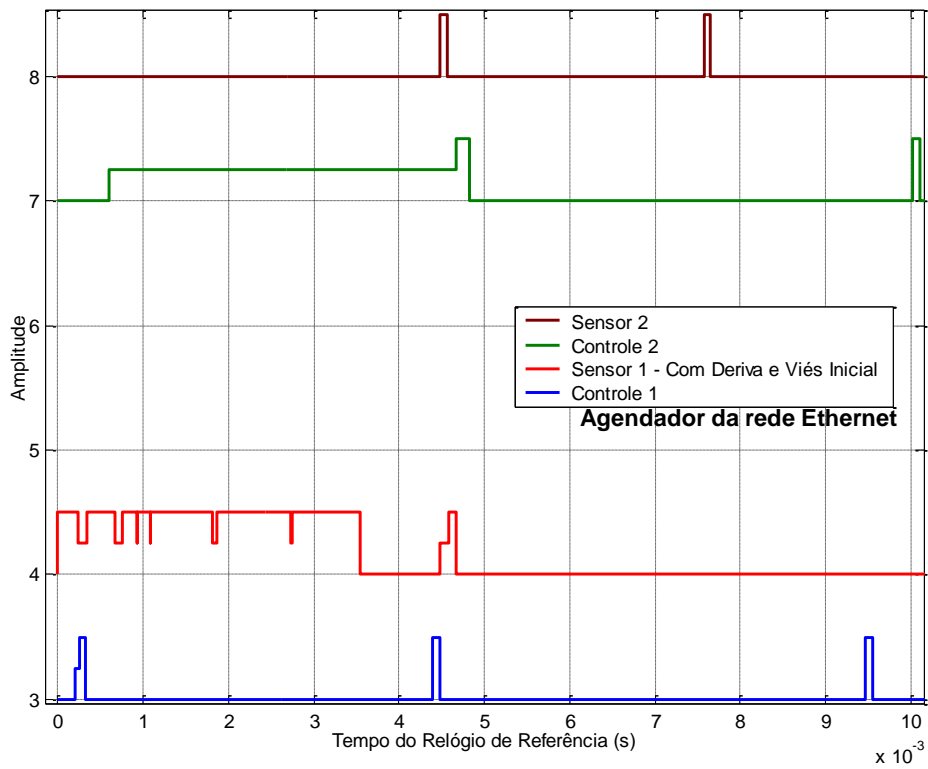


Figura 4.76 – Agendador da Rede Ethernet Sincronizado - AI.

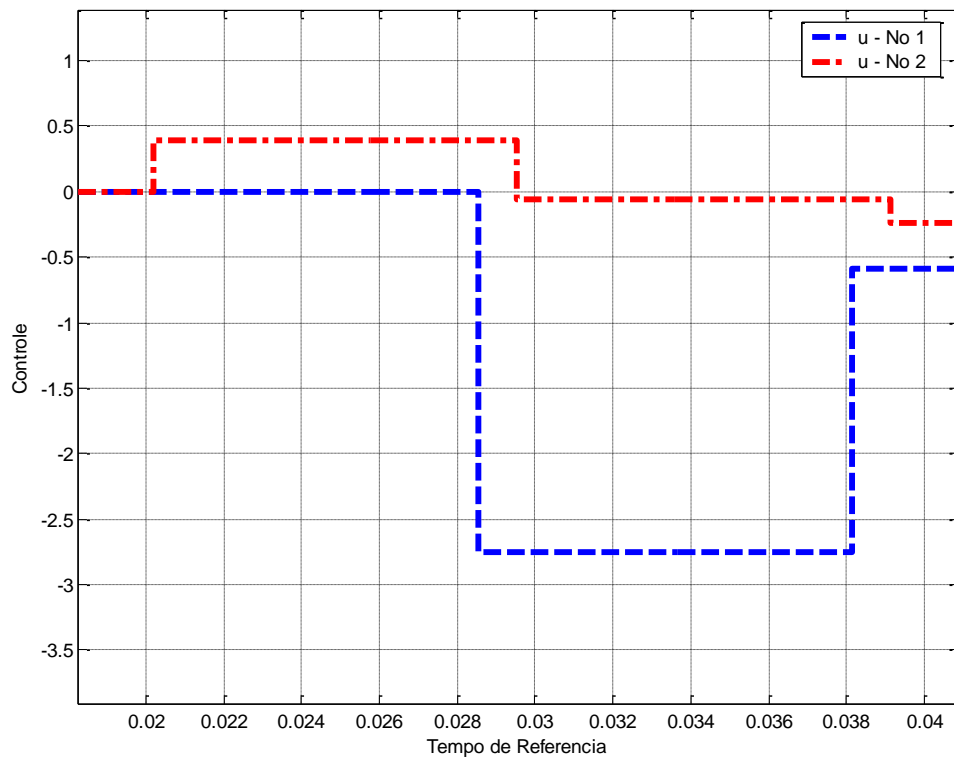


Figura 4.77 – Fase Invertida do Caso 3 - AI.

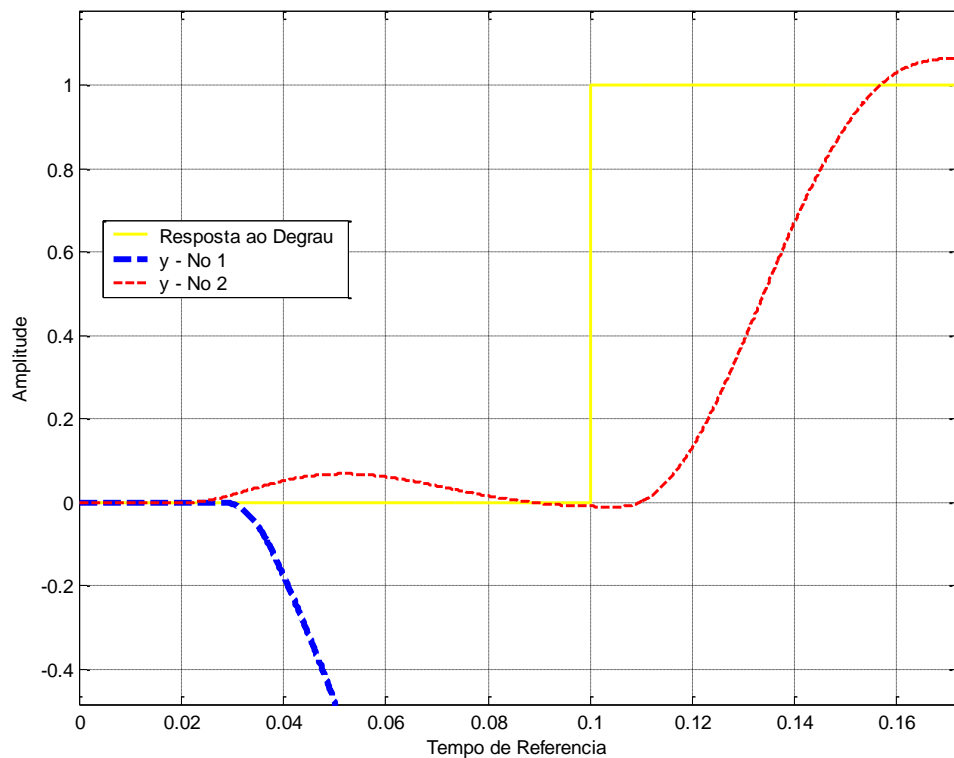


Figura 4.78 – Aproximação da Figura 4.63 no instante da inversão de fase - AI.

Desta comparação de casos, concluí-se que: 1) o barramento de comunicação TDMA é muito mais suscetível a erros de relógios que o um barramento de comunicação com Ethernet; 2) A Ethernet por não ser ativada por tempo aloca melhor as incertezas de relógio que o TDMA; 3) O Algoritmo de sincronização Welch-Lynch (FTM) não garante a correção de erros com viés inicial; 4) A Figura 4.78 mostra que o algoritmo de sincronização acopla as duas malhas de controle independentes.

## 5 CONCLUSÕES E SUGESTÕES PARA TRABALHOS FUTUROS

O problema de sincronização de relógios aparece em vários contextos e mundos de estudo diferentes, onde pode-se citar sistemas bancários, sistemas de tempo real, de tráfego aéreo, navegação, computação, comunicação, controle entre muitas outras áreas aonde um bom relógio é um fator determinante para o correto funcionamento do sistema.

Existem vários métodos para sincronização de relógios, métodos híbridos que envolvem soluções conjuntas de hardware e software, métodos somente por hardware e métodos por software. A sincronização híbrida é um dos métodos mais precisos, pois envolve a flexibilidade do software com a alta precisão do hardware. O problema é que para obter esta precisão, a sincronização de relógios híbridos tem um alto custo. Outra opção, a de hardware, tem um alto custo e boa precisão, mas perde em flexibilidade. A opção por software tem a menor precisão entre as outras opções de sincronização mas ganha em flexibilidade e no menor custo.

Este trabalho focou o estudo em sincronização por software. Foram estudadas as principais características dos algoritmos de sincronização de relógios Welch-Lynch e FTA, com um arquitetura de sincronização distribuída. Foram feitas simulações para analisar o desempenho do algoritmo Welch-Lynch sobre um sistema de controle por rede(Integrado) e sobre um sistema de controle com arquitetura federada. Foi estudado também o uso do Filtro de Kalman como método de sincronização de relógios em uma arquitetura de sincronização centralizada sobre um sistema de controle Federado. Foram usados dois protocolos de comunicação um com acesso ao meio TDMA que provê determinismo, tolerância a falhas e propriedades de tempo real da tecnologia *time-triggered*; e o Ethernet que provê flexibilidade, dinâmica e o legado de “melhor-esforço”.

Este trabalho separou os métodos de sincronização em 3 tipos de arquiteturas de sincronização padrão.

A arquitetura de sincronização distribuída faz uso do tempo global, onde não é utilizado um relógio de referência para sincronizar os relógios. Para simular a arquitetura distribuída, foram usados os algoritmos Welch-Lynch e FTA. Ambos os algoritmos de sincronização de relógios Welch-Lynch e FTA são muito parecidos, somente modificando a sua função de convergência e, em um caso geral, onde temos 4 relógios em um conjunto, os dois algoritmos são idênticos. Através de simulações foi possível constatar essa diferença. Apesar de serem diferentes em alguns casos, mesmo com o número de relógios no conjunto diferentes e maiores que 4, os algoritmos obtiveram o mesmo resultado. Estas simulações estão compiladas no Apêndice P. Estes casos, quando ambos os algoritmos possuem o mesmo resultado com número de relógios maior que 4, nos faz constatar que existe em certos casos particulares vantagens do Welch-Lynch sobre o FTA, pois o Welch-Lynch com menos dados alcança o mesmo resultado em menos tempo, isto é, em menos tempo de intervalo de transmissão TDMA. Portanto ao utilizar os algoritmos de sincronização de relógios Welch-Lynch ou FTA uma análise do tempo de convergência e número de dados (ou relógios) necessários para atingir a precisão requerida é importante para se conseguir a sincronização.

Na arquitetura centralizada, foi simulado o Filtro de Kalman como método de sincronização de relógios em um arquitetura Federada. A arquitetura centralizada tem a vantagem de não ser influenciada por erros dos relógios no conjunto, mas tem a desvantagem de ter um ponto comum de falha, onde se o relógio de referência falhar toda a sincronização do sistema é prejudicada. Formas de conter esse ponto comum de falha, como a estratégia multi-mestre, devem ser utilizadas em projeto de sistemas de controle em tempo real.

Outro fator importante sobre o método com Filtro de Kalman é que em todos os casos, independente da deriva de relógio e viés inicial, o filtro de Kalman manteve quase a mesma precisão final em todos os casos simulados. O Filtro de Kalman corrigiu melhor também erros de viés inicial.

Existe também a arquitetura com cálculo central, que é uma modificação da arquitetura centralizada para prover tolerância a falhas.

Este trabalho, simulou uma arquitetura Federada de controle, com o uso do *TrueTime Toolbox* em *Matlab/Simulink*, onde foram variados parâmetros do relógio de um dos nós do conjunto. Em nossa arquitetura federada, com a variação de relógios, as leis de controle e a resposta ao degrau se modificaram muito pouco. Com isso a arquitetura federada mostrou-se mais robusta em relação às imperfeições de relógios e influência da rede, pois a rede não interfere diretamente no sistema de controle. Com isso a congruência de dados dos quatro nós do conjunto não foram afetadas com uma taxa de deriva menor que  $\pm 10\%$ . Na arquitetura Federada com o uso do algoritmo *Welch-Lynch*, todos os casos convergiram a boas precisões, com exceção de um caso. O caso com viés inicial positivo, isto é, com o relógio iniciando adiantado, o algoritmo não teve um bom desempenho, era esperado, já que o Welch-Lynch em suas suposições, supõem que todos os relógios do conjunto devem iniciar sincronizados.

Na simulação com arquitetura integrada, isto é, com a malha de controle em rede, foi possível perceber que esta arquitetura não é muito robusta aos erros de relógio. Nestas simulações, apenas com 1% de deriva de relógio, o sistema de controle já é prejudicado, afetando a congruência de dados dos dois conjuntos de controle presentes ao barramento de dados. O Algoritmo Welch-Lynch nesta arquitetura integrada teve um ótimo desempenho, principalmente para pequenos erros de deriva. Em erros de viés inicial, no caso com o relógio atrasado, o algoritmo obteve um bom desempenho. No caso com o relógio

adiantado, em uma rede TDMA, o algoritmo não obteve bom desempenho prejudicando as leis de controle e a resposta ao degrau. Isto mostra que o algoritmo Welch-Lynch não garante que o sistema com viés inicial dos relógios seja corrigido. Outro fator importante desta simulação com a arquitetura integrada é que, no caso com o relógio adiantado, em uma rede Ethernet, o algoritmo conseguiu um desempenho melhor que no TDMA. Este fator, constata que o TDMA apesar de ser determinístico é mais suscetível aos erros de relógio. O Ethernet se mostrou mais flexível à erros de relógio.

Deste estudo conclui-se que em sistemas distribuídos de controle em tempo real, o uso de relógios é um fator muito importante. Mas o uso de relógios deve ser feito com cuidado, pois a medida que o sistema cresce, o número de relógios aumenta, e com isso os algoritmos de sincronização podem começar a influenciar e até prejudicar as leis de controle. Em sistemas de tempo real com comunicação TDMA, o sincronismo de relógios é fundamental para se obter um resultado lógico e temporal corretos. O Algoritmo Welch-Lynch e o Filtro de Kalman são dois métodos muito bons, desde que respeitadas as condições e restrições impostas pelo sistema e neste os seus resultados foram muito bons. Existem muitos assuntos para se analisar e estudar em sincronização de relógios. Este trabalho propôs estudar e simular sincronização por software utilizando-se de Filtro de Kalman e o Algoritmo Welch-Lynch, e analisar a influência da dessincronização de relógios em sistemas de controle distribuídos. Este trabalho também mostrou 3 possibilidades de arquiteturas de sincronização, e as 3 famílias de soluções. Dentro deste universo existem muitos outros assuntos a serem abordados.

### **Propostas para Trabalhos Futuros:**

O assunto sobre sincronização de relógios é muito extenso e várias linhas de pesquisas podem ser desenvolvidas. Abaixo existem algumas propostas para trabalhos futuros:



- Avaliar outros algoritmos de sincronização de relógios, como o IEEE 1588;
- Analisar os algoritmos estudados em outros sistemas de controle distribuído;
- Implementar e Validar o estudo em Hardware;
- Avaliar o efeito da sincronização Externa em conjunto com a Sincronização Interna;
- Simular algoritmos com outros tipos de arquitetura de sincronização de relógios, tal como a Arquitetura com Cálculo Central;
- Estudar e simular o efeito da flutuação do relógio na sincronização, verificar robustez do algoritmo;
- Estudar, verificar e analisar algoritmos para controle do tempo da CPU do computador;
- Explorar mais as arquiteturas de sincronização, identificando melhor os padrões existentes dos métodos de sincronização;
- Estudar e explorar a ferramenta de Simulação SIDERA;
- Estudar a sincronização de relógios em sistemas distribuídos na presença de mais de uma rede de comunicação;
- Introduzir o efeito do transporte na sincronização de relógios;
- Usar outros tipos de estimadores recursivos;
- Avaliar casos em que o FK não funciona corretamente e como otimizar a configuração do FK;

- Aplicar os algoritmos de sincronização em aplicações espaciais, como computadores de bordo, sensor de estrelas, GPS, câmeras de sensoriamento, e analisar seu funcionamento.

## REFERÊNCIAS BIBLIOGRÁFICAS

ALLAN D. W. ASHBY, N. HODGE, C. C. **The science of time keeping** - application note 1289. [S.l.]: Hewlett Packard, 1997.

ALLAN, D.W., **Clock Characterization Tutorial**, In: ANNUAL PRECISE TIME AND TIME INTERVAL (PTTI) SYSTEMS AND APPLICATIONS MEETING, 15., 1983, Washington, DC. **Proceedings...** , Washington: DC: Naval Research Lab, 1983.

CARVALHO, F. C. **Uma extensão do protocolo CAN para aplicações críticas em sistemas distribuídos**, 2006. 113p. Dissertação (Mestrado em Ciência da Computação) – Universidade Federal do Rio Grande do Sul, Porto Alegre.

BRIAND, L. P.; ROY, D. M. **Meeting Deadlines in Hard Real-time Systems: the Rate Monotonic Approach**, Los Alamitos, California: IEEE Computer Society, 2000.

DAHMAN, J. S.; FUJIMOTO, R. M.; WEATHERLY, R. M. The Department of Defense High Level Architecture. In: WINTER SIMULATION CONFERENCE, 29., 1998, Atlanta, GE, USA. **Proceedings...** Atlanta: WSC Foundation, 1998.

DUTERTRE, B. **The Welch-Lynch Clock Synchronization Algorithm**. London: Department of Computer Science Queen Mary and Westfield College, University of London, March, 1998, Technical Report 747.

EIDSON, J.; FISCHER, M.; WHITE, J. **IEEE-1588™ Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems**, In: ANNUAL PRECISE TIME AND TIME INTERVAL (PTTI), AND APPLICATIONS MEETING, p. 243-254, 2002, Reston, VA. **Proceedings...** , Reston: VA: Hyatt Regency Hotel, 2002.

EIDSON, J. **Measurement, control, and communication using IEEE 1588 - Advances in industrial control**, Germany: Springer, 2006.

FARINES, J.; FRAGA, J. S.; OLIVEIRA, R. S. **Sistemas de Tempo Real**, São Paulo: IME-USP, 2000.

GOBATO, M. F. **Controles monovariáveis e multivariáveis aplicados a sistemas aeroespaciais fracamente ou fortemente acoplados**. 2006. 388 p. (INPE-14494-TDI/1175). Dissertação (Mestrado em Mecânica Espacial e Controle) - Instituto Nacional de Pesquisas Espaciais, São José dos Campos. 2006. Disponível em: <<http://urlib.net/sid.inpe.br/MTCm13@80/2006/06.30.18.04>>. Acesso em: 15 jun. 2009.

GANSSELE, J. **Bus Cycle**, ESP – Embedded System Magazine, Março de 1995, Disponível em: < <http://www.ganssele.com/articles/abuscyc.htm> >. Acesso em: 16/12/2009.

GOODWIN, G.; GRAEBE, S.; SALGADO, M. **Control system design**. Valparaíso, Chile: Prentice Hall, 2000. 908p. ISBN: 10 0139586539.

GWALTNEY D. A.; BRISCOE J. M. **Comparison of communication architectures for spacecraft modular avionics systems**. Marshall Space Flight Center, AL 35812: NASA, pp. 32 (NASA/TM-2006-214431). 2006.

HENRIKSSON, D.; CERVIN A.; ÅRZÉN K.-E. **TrueTime: Simulation of control loops under shared computer resources**. In: IFAC WORLD CONGRESS ON AUTOMATIC CONTROL, 2002, Barcelona, ES. Proceedings..., Barcelona, ES, 2002.

HANZLIK, A. **Investigation of fault-tolerant multi-cluster clock synchronization strategies by means of simulation**. 2004. 143 p. Tese (Doutorado) - Vienna University of Technology Vienna, Austria. 2003. Disponível em: <<http://www.vmars.tuwien.ac.at/people/alexhanzlik.html>>. Acesso em: 10 dez. 2009.

HENRIKSSON, D.; CERVIN A.; ÅRZÉN K.-E. **TrueTime: real-time control system simulation with MATLAB/Simulink**. In: NORDIC MATLAB CONFERENCE, 2003, Copenhagen, DK. Proceedings..., Copenhagen, DK, 2003.

KOPETZ H. **Real time systems: design principles for distributed embedded applications**. Kluwer Academic Publishers, 1997.

KOPETZ, H.; BAUER, G. The Time-triggered architecture. **Proceedings of the IEEE**, v.91, n.1, p.112-126, 2003.

KOPETZ, H.; OCHSENREITER, W. **Clock Synchronization In Distributed Real-Time Systems**. Washington, DC.: IEEE Transactions Computer, v.36, n.8, p.933-940, 1987.

KOPETZ, H. HEXEL, R KRÜGER, A. MILLINGER, D. SCHEDL, A. **A Synchronization Strategy for a TTP/C Controller**. In: SAE CONGRESS AND EXHIBITION, 1997, Detroit, MI. **Proceedings...**, Detroit, MI, 1997.

KIHARA, M.; ONO, S.; ESKELINEN, P. **Digital Clocks for Synchronization and Communications**, Artech House Boston, Londres, 2003.

KUGA, H. K. **Noções Práticas de Técnicas de Estimação – Notas de Aula**, Instituto Nacional de Pesquisas Espaciais, INPE, São José dos Campos, 2005.

LAMPORT, L.; MELLIAR-SMITH, P. M. **Synchronizing Clocks in the Presence of Faults**. Journal of the ACM, [S.I.], v.32, n.1, p.52.78, 1985.

LAMPORT, L.; MELLIAR-SMITH, P. M. **Byzantine Clock Synchronization**, In: Annual ACM Symposium on Principles of Distributed Computing, Vancouver, B. C., **Proceedings...**, Vancouver, B. C: Canada, 1984.

LAMPORT, L. **Time, clocks, and the ordering of events in a distributed system**. Communications of the ACM, v.21, n. 7, p.558.565, ISSN:0001-0782, Julho, 1978.

LUNDELIUS, J.; LYNCH, N. **A New Fault-Tolerant Algorithm for Clock Synchronization**, Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, MA, Junho de 1984.

LALA, J. H.; HARPER, R. E. **Architectural Principles for Safety-Critical Real-Time Applications**, Proceedings of the IEEE, Janeiro 1, 1994.

LUSTOSA, H. D.; SOUZA, M. L. O. **Influences of Data Bus Protocols on an Aircraft Fly-By-Wire Networked Control System**. In: CONGRESSO 2008. SAE BRASIL - XVII CONGRESSO E EXPOSIÇÃO INTERNACIONAIS DE TECNOLOGIA DA MOBILIDADE, 2008c, São Paulo, SP - Brasil. Proceedings... 2008. v. 1, p. 54. CD-ROM. doi: <2008-36-0008>.

LUSTOSA, H. D. **Influência de Tipos de Barramentos e de suas Características de Alto Nível em Sistemas de Controle por Rede**. 2009. 247 p. (INPE- 15781-TDI/1524). Dissertação (Mestrado em Mecânica Espacial e Controle) - Instituto Nacional de Pesquisas Espaciais, São José dos Campos. 2009. Disponível em: <<http://urlib.net/sid.inpe.br/mtc-m18@80/2009/03.02.21.00>>. Acesso em: 08 fev. 2010.

MALLETE, L. A. **Atomic and quartz clock hardware for communication and navigation satellites**, In: ANNUAL PRECISE TIME AND TIME INTERVAL (PTTI) SYSTEMS AND APPLICATIONS MEETING, 2007, Long Beach, California **Proceedings...** , Long Beach: California: EUA, 2007.

MARQUES FILHO, E. A.; KUGA, H. K.; LOPES, R. V. F. **Real Time Estimation of GPS Receiver Clock Offset by the Kalman Filter**, In: INTERNATIONAL CONGRESS OF MECHANICAL ENGINEERING, 2003, São Paulo, SP. Proceedings..., São Paulo, SP, Nov., 10-14, 2003.

MAYBECK, P.S. **Stochastic models estimation and Control, Vol .1.** New York, NY, Academic Press, 1979.

MINER, P. S. **Verification of Fault-Tolerant Clock Synchronization Systems,** Novembro, 1993, NASA Technical Report, 3349.

MOREIRA, M. L. B. **Projeto e simulação de um controle discreto para a plataforma multi-missão e sua migração para um sistema operacional de tempo real.** 2006. 181 p. (INPE-14202-TDI/1103). Dissertação (Mestrado em Mecânica Espacial e Controle) - Instituto Nacional de Pesquisas Espaciais, São José dos Campos. 2006. Disponível em: <<http://urlib.net/sid.inpe.br/MTCm13@80/2006/07.10.13.42>>. Acesso em: 20 jun. 2009.

OBERMAISSER, R. PETI, P. TAGLIABO, F. , **An Integrated Architecture for Future Car Generations.** Real-Time Systems Journal, Volume 36, 2007, pages 101–133, Springer.

OGATA, K. **Engenharia de controle moderno.** 3. ed. Rio de Janeiro, RJ: Prentice-Hall do Brasil Ltda, 1997. 813p. ISBN: 85-7054-074-4.

RAJI R., **Smart network for control,** IEEE Spectrum pp 49-55, Junho 1994.

RAMANATHAN, P. SHIN, K.G., BUTLER, R. W., **Fault-Tolerant Clock Synchronization in Distributed Systems.** Computer, vol. 23, no. 10, pp. 33-42, Oct. 1990, doi:10.1109/2.58235.

REIS, A. M. **Simulação Distribuída em Tempo Real de um Sistema de Controle de Atitude e de Órbita para a Plataforma Multi-Missão utilizando a Arquitetura HLA.** 2009. 247 p. (INPE- 15872-TDI/1525). Dissertação (Mestrado em Mecânica Espacial e Controle) - Instituto Nacional de Pesquisas Espaciais, São José dos Campos. 2009. Disponível em: <<http://urlib.net/sid.inpe.br/mtc-m18@80/2009/03.02.22.17>>. Acesso em: 08 fev. 2010.

SCHNEIDER, F. B. **A Paradigm for Reliable Clock Synchronization.** In: ADVANCED SEMINAR OF LOCAL AREA NETWORK, 1986, Bandol, France **Proceedings...** Bandol, France, 1986.

SPITZER, C. R. **Avionics: Elements, Software and Functions. Digital Avionics Handbook.,** Williamsburg, Virginia, U.S., AvioniCon, Inc., 2007.

STANKOVIC J.A. Misconceptions about real-time computing – a serious problem for next-generation systems. **IEEE Computer,** vol.21, no.10, pp.10-19, Outubro 1988.

TÖRNGREN, M. Fundamentals of Implementing Real-Time Control Applications in Distributed Computer Systems. **Real-Time Systems**, 14, 219–250 (1998).

TTTECH. **Time-Triggered protocol TTP/C high-level specification document protocol version 1.1**, D–032–S–10–028, TTTech Computertechnik, Edition 1.4.3, Nov., 2003.

VALDIVIA, R. H. V. **Influência dos agendadores da computação (RMS) e da comunicação (TDMA) na estabilidade de um sistema de controle por rede**. 2009. 168 p. (INPE-15692-TDI/1466). Tese (Doutorado em Mecânica Espacial e Controle) - Instituto Nacional de Pesquisas Espaciais, São José dos Campos. 2009. Disponível em: <<http://urlib.net/sid.inpe.br/mtcm18@80/2009/02.06.17.10>>. Acesso em: 20 jun. 2009.

VARNUM, F. B. **Kalman Filtering with a Two-State Clock Model**, In: ANNUAL PRECISE TIME AND TIME INTERVAL (PTTI) SYSTEMS AND APPLICATIONS MEETING, 15., 1983, Washington, DC. **Proceedings...**, Washington: DC: Naval Research Lab, 1983.





## **GLOSSÁRIO**

Welch-Lynch – Algoritmo de Sincronização de Relógios, conhecido também com FTM (Fault-Tolerant Mid-Point).

broadcast – ato de enviar dados a todo os nós da rede de comunicação.



## APÊNDICE A - CÓDIGO DAS FUNÇÕES CHAMADAS PELA SIMULAÇÃO DA ARQUITETURA FEDERADA

### A.1 EXEMPLO DE UMA FUNÇÃO DE INICIALIZAÇÃO DO COMPUTADOR DE TEMPO REAL

*function task\_init(mode)*

global id

global flag1\_1

global flag1\_2

global flag1\_3

global flag1\_4

global t1\_1

global t1\_2

global t1\_3

global t1\_4

global flag

global tk

global tk1

global tmk

global tmk1

*% Initialize TrueTime kernel*

*ttlNitKernel(2, 7, 'prioFP'); % nbrOfInputs, nbrOfOutputs, fixed priority*

*%Configurando Canais de Saida*

*data.Chan1 = 1; %Tempo 1*

*data.Chan2 = 2; %Tempo 2*

*data.Chan3 = 3; %Tempo 3*

*data.Chan4 = 4; %Tempo 4*

*data.uChan = 5; %Controle*

*data.corChan = 6; %Correcao*

*data.TS1Chan = 7; %Tempo Corrigido*

*%Configurando Entradas*

```
data.rChan = 1; %Referencia
data.yChan = 2; %Realimentacao
%Inicializando Variaveis
data.time1 = 0;
data.time2 = 0;
data.time3 = 0;
data.time4 = 0;
data.teste = 0;
data.cor = 0;
data.sync = mode;
flag = 0;
tk = 0;
tk1 = 0;
tmk = 0;
tmk1 = 0;
flag1_1 = 0;
flag1_2 = 0;
flag1_3 = 0;
flag1_4 = 0;
t1_1 = 0;
t1_2 = 0;
t1_3 = 0;
t1_4 = 0;
flag = 0;
%-----Configurando Tarefa de Controle-----%
period = 0.0096; %Periodo
data.K = 0.96;
data.Ti = 0.12;
data.Td = 0.049;
data.beta = 0.5;
data.N = 10;
```

```

data.h = period;
data.u = 0;
data.lold = 0;
data.Dold = 0;
data.yold = 0;
deadline = period; %Deadline
offset = 0.0; % start of first task job
prio = 2; %PRIORIDADE
ttCreatePeriodicTask('pid_task', offset, period, prio, 'pidcode1', data);
%-----Configurando Tarefa de Sincronismo-----%
offset = 0;
period = 0.0010;
prio = 4; %PRIORIDADE
ttCreatePeriodicTask('sync_task', offset, period, prio, 'synccode', data);
offset = 0;
period = 0.1;
prio = 3; %PRIORIDADE
%-----Configurando Tarefa de Envio Periodica-----%
offset = 0;
period = 0.0040;
prio = 2; %PRIORIDADE
ttCreatePeriodicTask('send_task1', offset, period, prio, 'sendTask1code', data);
%-----Initialize network-----%
prio = 1; %PRIORIDADE
ttCreateInterruptHandler('nw_handler1', prio, 'msgRcvTask1', data);
ttlNitNetwork(1, 'nw_handler1'); % node #1 in the network

```

## **A.2 EXEMPLO DE UMA FUNÇÃO DO FILTRO DE KALMAN**

***function [exectime, data] = synccodeKalman(seg, data)***

global t1\_1

global t1\_2

```

global t1_3
global t1_4
global flag
global tk
global tk1
global tmk
global tmk1
global tt1
switch seg,
    case 1,
        tk = ttCurrentTime;
        tmk = ((t1_2 + t1_3 + t1_4)/4);
        kalmanTime = kalmanfilter(0.0004);
        kalmanTime;
        ttAnalogOut(data.corChan, kalmanTime(1));
        tk = ttCurrentTime;
        ttCurrentTime(tk - kalmanTime(1));
        tk = ttCurrentTime;
        exectime = 0.00001;
    case 2,
        %-----Finalizando Variaveis-----%
        tk1 = tk;
        tmk1 = tmk;
        exectime = -1;
end

```

## APÊNDICE B - CÓDIGO DAS FUNÇÕES CHAMADAS PELA SIMULAÇÃO DA ARQUITETURA INTEGRADA

### B.1 EXEMPLO DE FUNÇÃO DO SENSOR

#### ***function sensor\_init(arg)***

```
% Distributed control system: sensor node
% Initialize TrueTime kernel
ttlInitKernel(1, 2, 'prioFP'); % nbrOfInputs, nbrOfOutputs, fixed priority
rj = 0;
% Create sensor task
data.y = 0;
offset = 0.0075;
period = 0.0096;
prio = 1;
data.time = 0;
data.Chan1 = 1;
data.Chan2 = 2;
ttCreatePeriodicTask('sens_task', offset, period, prio, 'senscode', data);
% Initialize network
ttCreateInterruptHandler('nw_handler4', prio, 'msgRcvSensor', data);
ttlInitNetwork(4, 'nw_handler4'); % node #4 in the network
```

### B.2 EXEMPLO DE FUNÇÃO DO CONTROLADOR

#### ***function controller\_init(arg)***

```
global rr
global yy
global id
global t1_1
global t2_1
global t3_1
```

```

global t4_1
% Distributed control system: controller node
% Initialize TrueTime kernel
ttlNitKernel(1, 2, 'prioFP'); % nbrOfInputs, nbrOfOutputs, fixed priority
%iniciar Variaveis Globais
id = 0;
t1_1 = 0;
t1_2 = 0;
t1_3 = 0;
t1_4 = 0;
data.time1 = 0;
data.time2 = 0;
data.time3 = 0;
data.time4 = 0;
% Controller parameters
h = 0.0096;
N = 100000;
Td = 0.035;
K = 1.5;
% Create task data (local memory)
data.u = 0.0;
data.K = K;
data.ad = Td/(N*h+Td);
data.bd = N*K*Td/(N*h+Td);
data.Dold = 0.0;
data.yold = 0.0;
data.yo = 0.0;
yy = 0.0;
rr = 0.0;
data.r = 0.0;
data.time = 0;

```



```

data.Chan2 = 1;
data.Chan3 = 2;
offset = 0.0002;
period = 0.0096;
prio = 1;
% Create controller task
deadline = h;
prio = 2;
%ttCreateTask('pid_task', deadline, prio, 'ctrlcode', data);
ttCreatePeriodicTask('pid_task', offset, period, prio, 'ctrlcode', data);
%-----Tarefa de Envio do Tempo-----%
if arg==1
    offset = 0.0044;
    period = 0.0768;
    prio = 3;
    ttCreatePeriodicTask('send_time', offset, period, prio, 'send_time', data);
end
prio=1;
%-----Fim do Envio do Tempo-----%
% Initialize network
ttCreateInterruptHandler('nw_handler3', prio, 'msgRcvCtrl', data);
ttlNitNetwork(3, 'nw_handler3'); % node #3 in the network

```

### B.3 EXEMPLO DA FUNÇÃO DE RECEBIMENTO E DO ALGORITMO WELCH-LYNCH

```
function [exectime, data] = msgRcvCtrl(seg, data)
```

```

global rr
global yy

```

```
global id
```

```
global t1_1
```

```
global t1_2
```

```
global t1_3
```

```
global t1_4
```

```
switch seg,
```

```
case 1,
```

```
    t1_1 = ttCurrentTime;
```

```
    switch id
```

```
        case 1,
```

```
            disp('erro 1');
```

```
            xp = ttGetMsg;
```

```
            ttSetNextSegment(5);
```

```
            exectime = 0;
```

```
        case 2,
```

```
            t1_1 = ttCurrentTime;
```

```
            data.time2 = ttGetMsg;
```

```
            t1_2 = t1_1 - data.time2;
```

```
            ttSetNextSegment(5);
```

```
            exectime = 0;
```

```
        case 3,
```

```
            t1_1 = ttCurrentTime;
```

```
            data.time3 = ttGetMsg;
```

```
            t1_3 = t1_1 - data.time3;
```

```
            ttSetNextSegment(5);
```

```
            exectime = 0;
```

```
        case 4,
```

```
            t1_1 = ttCurrentTime;
```

```

    data.time4 = ttGetMsg;
    t1_4 = t1_1 - data.time4;
    ttSetNextSegment(3);
    exectime = 0;
case 5,
    %disp('5 1 ');
    ttSetNextSegment(4);
    exectime = 0;
end
case 2,
    xp =ttGetMsg;
    ttSetNextSegment(5);
    exectime = 0;
case 3,
    %SYNC%
    disp('SYNC 1');
    X = [(0) (t1_2) (t1_3) (t1_4)];
    X = sort(X);
    delay = 0.00011;
    corr = (X(2) + X(3))/2;
    ttAnalogOut(data.Chan3, corr);
    data.time = ttCurrentTime;
    data.time = data.time + delay - corr;
    ttCurrentTime(data.time);
    ttSetNextSegment(5);
    exectime = 0;
case 4,
    data.yo = ttGetMsg;      % Obtain sensor value
    yy = data.yo;
    data.r = ttAnalogIn(1); % Read reference value
    rr = data.r;

```

```
    exectime = 0.000005;  
case 5,  
    xp = ttGetMsg;  
    exectime = -1; % finished  
end
```

## APÊNDICE C – TABELAS DE DADOS DA SIMULAÇÃO COM EXCEL

### C.1 Tabela do 1º Caso do Experimento FTM/FTA:

Tabela C.1 – Tabela de Dados do Experimento FTM/FTA – 1º Caso.

T1	T2	T3	T4	Delta 1	Delta 2	Delta 3	Delta 4	Menor Delta	Maior Delta	Média	Correção	Ciclo	Referência	T1 Sem Correção
0	0	0	0	0	0	0	0	0,1	0,2	0,15	0,145	T1	0	0
0,6	0,5	0,5	0,5	0,1	0	0	0	0	0	0	-0,005	T2	0,5	0,6
1,2	1	1	1	0,2	0	0	0	0	0	0	-0,005	T3	1	1,2
1,8	1,5	1,5	1,5	0,3	0	0	0	0	0	0	-0,005	T4	1,5	1,8
2,255	2,005	2,005	2,005	0	-0,25	-0,25	-0,25	0,35	0,45	0,4	0,395	T1	2	2,4
2,855	2,505	2,505	2,505	0,35	0	0	0	0	0	0	-0,005	T2	2,5	3
3,455	3,005	3,005	3,005	0,45	0	0	0	0	0	0	-0,005	T3	3	3,6
4,055	3,505	3,505	3,505	0,55	0	0	0	0	0	0	-0,005	T4	3,5	4,2
4,26	4,01	4,01	4,01	0	-0,25	-0,25	-0,25	0,35	0,45	0,4	0,395	T1	4	4,8
4,86	4,51	4,51	4,51	0,35	0	0	0	0	0	0	-0,005	T2	4,5	5,4
5,46	5,01	5,01	5,01	0,45	0	0	0	0	0	0	-0,005	T3	5	6
6,06	5,51	5,51	5,51	0,55	0	0	0	0	0	0	-0,005	T4	5,5	6,6
6,265	6,015	6,015	6,015	0	-0,25	-0,25	-0,25	0,35	0,45	0,4	0,395	T1	6	7,2
6,865	6,515	6,515	6,515	0,35	0	0	0	0	0	0	-0,005	T2	6,5	7,8
7,465	7,015	7,015	7,015	0,45	0	0	0	0	0	0	-0,005	T3	7	8,4
8,065	7,515	7,515	7,515	0,55	0	0	0	0	0	0	-0,005	T4	7,5	9

(Continua)

Tabela C1 - Conclusão

8,27	8,02	8,02	8,02	0	-0,25	-0,25	-0,25	0,35	0,45	0,4	0,395	<b>T1</b>	8	9,6
8,87	8,52	8,52	8,52	0,35	0	0	0	0	0	0	-0,005	<b>T2</b>	8,5	10,2
9,47	9,02	9,02	9,02	0,45	0	0	0	0	0	0	-0,005	<b>T3</b>	9	10,8
10,07	9,52	9,52	9,52	0,55	0	0	0	0	0	0	-0,005	<b>T4</b>	9,5	11,4
10,275	10,025	10,025	10,025	0	-0,25	-0,25	-0,25	0,35	0,45	0,4	0,395	<b>T1</b>	10	12
10,875	10,525	10,525	10,525	0,35	0	0	0	0	0	0	-0,005	<b>T2</b>	10,5	12,6
11,475	11,025	11,025	11,025	0,45	0	0	0	0	0	0	-0,005	<b>T3</b>	11	13,2
12,075	11,525	11,525	11,525	0,55	0	0	0	0	0	0	-0,005	<b>T4</b>	11,5	13,8
12,28	12,03	12,03	12,03	0	-0,25	-0,25	-0,25	0,35	0,45	0,4	0,395	<b>T1</b>	12	14,4
12,88	12,53	12,53	12,53	0,35	0	0	0	0	0	0	-0,005	<b>T2</b>	12,5	15
13,48	13,03	13,03	13,03	0,45	0	0	0	0	0	0	-0,005	<b>T3</b>	13	15,6
14,08	13,53	13,53	13,53	0,55	0	0	0	0	0	0	-0,005	<b>T4</b>	13,5	16,2
14,285	14,035	14,035	14,035	0	-0,25	-0,25	-0,25	0,35	0,45	0,4	0,395	<b>T1</b>	14	16,8
14,885	14,535	14,535	14,535	0,35	0	0	0	0	0	0	-0,005	<b>T2</b>	14,5	17,4
15,485	15,035	15,035	15,035	0,45	0	0	0	0	0	0	-0,005	<b>T3</b>	15	18
16,085	15,535	15,535	15,535	0,55	0	0	0	0	0	0	-0,005	<b>T4</b>	15,5	18,6
16,29	16,04	16,04	16,04	0	-0,25	-0,25	-0,25	0,35	0,45	0,4	0,395	<b>T1</b>	16	19,2
16,89	16,54	16,54	16,54	0,35	0	0	0	0	0	0	-0,005	<b>T2</b>	16,5	19,8
17,49	17,04	17,04	17,04	0,45	0	0	0	0	0	0	-0,005	<b>T3</b>	17	20,4
18,09	17,54	17,54	17,54	0,55	0	0	0	0	0	0	-0,005	<b>T4</b>	17,5	21
18,295	18,045	18,045	18,045	0	-0,25	-0,25	-0,25						18	21,6

## C.2 Tabela do 2º Caso do Experimento FTM/FTA:

Tabela C.2 – Dados do Experimento FTM/FTA – 2º Caso.

T1	T2	T3	T4	Delta 1	Delta 2	Delta 3	Delta 4	Menor Delta	Maior Delta	Média	Correção	Ciclo	Referência	T1 Sem Correção
0	0	0	0	0	0	0	0	0,1	0,2	0,15	-0,35	T1	0	0
0,6	0,5	0,5	0,5	0,1	0	0	0	0	0	0	-0,5	T2	0,5	0,6
1,2	1	1	1	0,2	0	0	0	0	0	0	-0,5	T3	1	1,2
1,8	1,5	1,5	1,5	0,3	0	0	0	0	0	0	-0,5	T4	1,5	1,8
2,75	2,5	2,5	2,5	0	-0,25	-0,25	-0,25	0,35	0,45	0,4	-0,1	T1	2	2,4
3,35	3	3	3	0,35	0	0	0	0	0	0	-0,5	T2	2,5	3
3,95	3,5	3,5	3,5	0,45	0	0	0	0	0	0	-0,5	T3	3	3,6
4,55	4	4	4	0,55	0	0	0	0	0	0	-0,5	T4	3,5	4,2
5,25	5	5	5	0	-0,25	-0,25	-0,25	0,35	0,45	0,4	-0,1	T1	4	4,8
5,85	5,5	5,5	5,5	0,35	0	0	0	0	0	0	-0,5	T2	4,5	5,4
6,45	6	6	6	0,45	0	0	0	0	0	0	-0,5	T3	5	6
7,05	6,5	6,5	6,5	0,55	0	0	0	0	0	0	-0,5	T4	5,5	6,6
7,75	7,5	7,5	7,5	0	-0,25	-0,25	-0,25	0,35	0,45	0,4	-0,1	T1	6	7,2
8,35	8	8	8	0,35	0	0	0	0	0	0	-0,5	T2	6,5	7,8
8,95	8,5	8,5	8,5	0,45	0	0	0	0	0	0	-0,5	T3	7	8,4
9,55	9	9	9	0,55	0	0	0	0	0	0	-0,5	T4	7,5	9
10,25	10	10	10	0	-0,25	-0,25	-0,25	0,35	0,45	0,4	-0,1	T1	8	9,6
10,85	10,5	10,5	10,5	0,35	0	0	0	0	0	0	-0,5	T2	8,5	10,2

(Continua)

Tabela C2 - Conclusão

11,45	11	11	11	0,45	0	0	0	0	0	0	-0,5	<b>T3</b>	9	10,8
12,05	11,5	11,5	11,5	0,55	0	0	0	0	0	0	-0,5	<b>T4</b>	9,5	11,4
12,75	12,5	12,5	12,5	0	-0,25	-0,25	-0,25	0,35	0,45	0,4	-0,1	<b>T1</b>	10	12
13,35	13	13	13	0,35	0	0	0	0	0	0	-0,5	<b>T2</b>	10,5	12,6
13,95	13,5	13,5	13,5	0,45	0	0	0	0	0	0	-0,5	<b>T3</b>	11	13,2
14,55	14	14	14	0,55	0	0	0	0	0	0	-0,5	<b>T4</b>	11,5	13,8
15,25	15	15	15	0	-0,25	-0,25	-0,25	0,35	0,45	0,4	-0,1	<b>T1</b>	12	14,4
15,85	15,5	15,5	15,5	0,35	0	0	0	0	0	0	-0,5	<b>T2</b>	12,5	15
16,45	16	16	16	0,45	0	0	0	0	0	0	-0,5	<b>T3</b>	13	15,6
17,05	16,5	16,5	16,5	0,55	0	0	0	0	0	0	-0,5	<b>T4</b>	13,5	16,2
17,75	17,5	17,5	17,5	0	-0,25	-0,25	-0,25	0,35	0,45	0,4	-0,1	<b>T1</b>	14	16,8
18,35	18	18	18	0,35	0	0	0	0	0	0	-0,5	<b>T2</b>	14,5	17,4
18,95	18,5	18,5	18,5	0,45	0	0	0	0	0	0	-0,5	<b>T3</b>	15	18
19,55	19	19	19	0,55	0	0	0	0	0	0	-0,5	<b>T4</b>	15,5	18,6
20,25	20	20	20	0	-0,25	-0,25	-0,25	0,35	0,45	0,4	-0,1	<b>T1</b>	16	19,2
20,85	20,5	20,5	20,5	0,35	0	0	0	0	0	0	-0,5	<b>T2</b>	16,5	19,8
21,45	21	21	21	0,45	0	0	0	0	0	0	-0,5	<b>T3</b>	17	20,4
22,05	21,5	21,5	21,5	0,55	0	0	0	0	0	0	-0,5	<b>T4</b>	17,5	21
22,75	22,5	22,5	22,5	0	-0,25	-0,25	-0,25						18	21,6



**C.3 Tabela do 3º Caso do Experimento FTM/FTA:**

Tabela C.3 – Modelo do Experimento FTM x FTA – 3º Caso.

<b>T1</b>	<b>T2</b>	<b>T3</b>	<b>T4</b>	<b>Delta 1</b>	<b>Delta 2</b>	<b>Delta 3</b>	<b>Delta 4</b>	<b>Menor Delta</b>	<b>Maior Delta</b>	<b>Média</b>	<b>Correção</b>	<b>Ciclo</b>	<b>Referência</b>	<b>T1 Sem Correção</b>
0	0	0	0	0	0	0	0	-1	-0,5	-0,75	-0,755	T1	0	0
0	0,5	0,5	0,5	-0,5	0	0	0	0	0	0	-0,005	T2	0,5	0
0	1	1	1	-1	0	0	0	0	0	0	-0,005	T3	1	0
0	1,5	1,5	1,5	-1,5	0	0	0	0	0	0	-0,005	T4	1,5	0
0,755	2,005	2,005	2,005	0	1,25	1,25	1,25	-2,25	-1,75	-2	-2,005	T1	2	0
0,755	2,505	2,505	2,505	-1,75	0	0	0	0	0	0	-0,005	T2	2,5	0
0,755	3,005	3,005	3,005	-2,25	0	0	0	0	0	0	-0,005	T3	3	0
0,755	3,505	3,505	3,505	-2,75	0	0	0	0	0	0	-0,005	T4	3,5	0
2,76	4,01	4,01	4,01	0	1,25	1,25	1,25	-2,25	-1,75	-2	-2,005	T1	4	0
2,76	4,51	4,51	4,51	-1,75	0	0	0	0	0	0	-0,005	T2	4,5	0
2,76	5,01	5,01	5,01	-2,25	0	0	0	0	0	0	-0,005	T3	5	0
2,76	5,51	5,51	5,51	-2,75	0	0	0	0	0	0	-0,005	T4	5,5	0
4,765	6,015	6,015	6,015	0	1,25	1,25	1,25	-2,25	-1,75	-2	-2,005	T1	6	0
4,765	6,515	6,515	6,515	-1,75	0	0	0	0	0	0	-0,005	T2	6,5	0
4,765	7,015	7,015	7,015	-2,25	0	0	0	0	0	0	-0,005	T3	7	0
4,765	7,515	7,515	7,515	-2,75	0	0	0	0	0	0	-0,005	T4	7,5	0
6,77	8,02	8,02	8,02	0	1,25	1,25	1,25	-2,25	-1,75	-2	-2,005	T1	8	0

(Continua)

Tabela C3 - Conclusão

<b>6,77</b>	8,52	8,52	8,52	-1,75	0	0	0	0	0	0	-0,005	<b>T2</b>	8,5	0
<b>6,77</b>	9,02	9,02	9,02	-2,25	0	0	0	0	0	0	-0,005	<b>T3</b>	9	0
<b>6,77</b>	9,52	9,52	9,52	-2,75	0	0	0	0	0	0	-0,005	<b>T4</b>	9,5	0
<b>8,775</b>	10,025	10,025	10,025	0	1,25	1,25	1,25	-2,25	-1,75	-2	-2,005	<b>T1</b>	10	0
<b>8,775</b>	10,525	10,525	10,525	-1,75	0	0	0	0	0	0	-0,005	<b>T2</b>	10,5	0
<b>8,775</b>	11,025	11,025	11,025	-2,25	0	0	0	0	0	0	-0,005	<b>T3</b>	11	0
<b>8,775</b>	11,525	11,525	11,525	-2,75	0	0	0	0	0	0	-0,005	<b>T4</b>	11,5	0
<b>10,78</b>	12,03	12,03	12,03	0	1,25	1,25	1,25	-2,25	-1,75	-2	-2,005	<b>T1</b>	12	0
<b>10,78</b>	12,53	12,53	12,53	-1,75	0	0	0	0	0	0	-0,005	<b>T2</b>	12,5	0
<b>10,78</b>	13,03	13,03	13,03	-2,25	0	0	0	0	0	0	-0,005	<b>T3</b>	13	0
<b>10,78</b>	13,53	13,53	13,53	-2,75	0	0	0	0	0	0	-0,005	<b>T4</b>	13,5	0
<b>12,785</b>	14,035	14,035	14,035	0	1,25	1,25	1,25	-2,25	-1,75	-2	-2,005	<b>T1</b>	14	0
<b>12,785</b>	14,535	14,535	14,535	-1,75	0	0	0	0	0	0	-0,005	<b>T2</b>	14,5	0
<b>12,785</b>	15,035	15,035	15,035	-2,25	0	0	0	0	0	0	-0,005	<b>T3</b>	15	0
<b>12,785</b>	15,535	15,535	15,535	-2,75	0	0	0	0	0	0	-0,005	<b>T4</b>	15,5	0
<b>14,79</b>	16,04	16,04	16,04	0	1,25	1,25	1,25	-2,25	-1,75	-2	-2,005	<b>T1</b>	16	0
<b>14,79</b>	16,54	16,54	16,54	-1,75	0	0	0	0	0	0	-0,005	<b>T2</b>	16,5	0
<b>14,79</b>	17,04	17,04	17,04	-2,25	0	0	0	0	0	0	-0,005	<b>T3</b>	17	0
<b>14,79</b>	17,54	17,54	17,54	-2,75	0	0	0	0	0	0	-0,005	<b>T4</b>	17,5	0
<b>16,795</b>	18,045	18,045	18,045	0	1,25	1,25	1,25						18	0

#### C.4 Tabela do 4º Caso do Experimento FTM/FTA:

Tabela C.4 – Dados do Experimento FTM/FTA – 4º Caso.

T1	T2	T3	T4	T5	Delta 1	Delta 2	Delta 3	Delta 4	Delta 5	Menor Delta	Maior Delta	Média	Correção	Ciclo	Referência	T1 Sem Correção
0	0	0	0	0	0	0	0	0	0	0,1	0,3	0,2	0,195	T1	0	0
0,6	0,5	0,5	0,5	0,5	0,1	0	0	0	0	0	0	0	-0,005	T2	0,5	0,6
1,2	1	1	1	1	0,2	0	0	0	0	0	0	0	-0,005	T3	1	1,2
1,8	1,5	1,5	1,5	1,5	0,3	0	0	0	0	0	0	0	-0,005	T4	1,5	1,8
2,4	2	2	2	2	0,4	0	0	0	0	0	0	0	-0,005	T5	2	2,4
2,805	2,505	2,505	2,505	2,505	0	-0,3	-0,3	-0,3	-0,3	0,4	0,6	0,5	0,495	T1	2,5	3
3,405	3,005	3,005	3,005	3,005	0,4	0	0	0	0	0	0	0	-0,005	T2	3	3,6
4,005	3,505	3,505	3,505	3,505	0,5	0	0	0	0	0	0	0	-0,005	T3	3,5	4,2
4,605	4,005	4,005	4,005	4,005	0,6	0	0	0	0	0	0	0	-0,005	T4	4	4,8
5,205	4,505	4,505	4,505	4,505	0,7	0	0	0	0	0	0	0	-0,005	T5	4,5	5,4
5,31	5,01	5,01	5,01	5,01	0	-0,3	-0,3	-0,3	-0,3	0,4	0,6	0,5	0,495	T1	5	6
5,91	5,51	5,51	5,51	5,51	0,4	0	0	0	0	0	0	0	-0,005	T2	5,5	6,6
6,51	6,01	6,01	6,01	6,01	0,5	0	0	0	0	0	0	0	-0,005	T3	6	7,2
7,11	6,51	6,51	6,51	6,51	0,6	0	0	0	0	0	0	0	-0,005	T4	6,5	7,8
7,71	7,01	7,01	7,01	7,01	0,7	0	0	0	0	0	0	0	-0,005	T5	7	8,4
7,815	7,515	7,515	7,515	7,515	0	-0,3	-0,3	-0,3	-0,3	0,4	0,6	0,5	0,495	T1	7,5	9
8,415	8,015	8,015	8,015	8,015	0,4	0	0	0	0	0	0	0	-0,005	T2	8	9,6

(Continua)

Tabela C4 - Conclusão

9,015	8,515	8,515	8,515	8,515	0,5	0	0	0	0	0	0	0	-0,005	<b>T3</b>	8,5	10,2
9,615	9,015	9,015	9,015	9,015	0,6	0	0	0	0	0	0	0	-0,005	<b>T4</b>	9	10,8
10,215	9,515	9,515	9,515	9,515	0,7	0	0	0	0	0	0	0	-0,005	<b>T5</b>	9,5	11,4
10,32	10,02	10,02	10,02	10,02	0	-0,3	-0,3	-0,3	-0,3	0,4	0,6	0,5	0,495	<b>T1</b>	10	12
10,92	10,52	10,52	10,52	10,52	0,4	0	0	0	0	0	0	0	-0,005	<b>T2</b>	10,5	12,6
11,52	11,02	11,02	11,02	11,02	0,5	0	0	0	0	0	0	0	-0,005	<b>T3</b>	11	13,2
12,12	11,52	11,52	11,52	11,52	0,6	0	0	0	0	0	0	0	-0,005	<b>T4</b>	11,5	13,8
12,72	12,02	12,02	12,02	12,02	0,7	0	0	0	0	0	0	0	-0,005	<b>T5</b>	12	14,4
12,825	12,525	12,525	12,525	12,525	0	-0,3	-0,3	-0,3	-0,3	0,4	0,6	0,5	0,495	<b>T1</b>	12,5	15
13,425	13,025	13,025	13,025	13,025	0,4	0	0	0	0	0	0	0	-0,005	<b>T2</b>	13	15,6
14,025	13,525	13,525	13,525	13,525	0,5	0	0	0	0	0	0	0	-0,005	<b>T3</b>	13,5	16,2
14,625	14,025	14,025	14,025	14,025	0,6	0	0	0	0	0	0	0	-0,005	<b>T4</b>	14	16,8
15,225	14,525	14,525	14,525	14,525	0,7	0	0	0	0	0	0	0	-0,005	<b>T5</b>	14,5	17,4
15,33	15,03	15,03	15,03	15,03	0	-0,3	-0,3	-0,3	-0,3	0,4	0,6	0,5	0,495	<b>T1</b>	15	18
15,93	15,53	15,53	15,53	15,53	0,4	0	0	0	0	0	0	0	-0,005	<b>T2</b>	15,5	18,6
16,53	16,03	16,03	16,03	16,03	0,5	0	0	0	0	0	0	0	-0,005	<b>T3</b>	16	19,2
17,13	16,53	16,53	16,53	16,53	0,6	0	0	0	0	0	0	0	-0,005	<b>T4</b>	16,5	19,8
17,73	17,03	17,03	17,03	17,03	0,7	0	0	0	0	0	0	0	-0,005	<b>T5</b>	17	20,4
17,835	17,535	17,535	17,535	17,535	0	-0,3	-0,3	-0,3	-0,3						17,5	21

**C.5 Tabela do 5º Caso do Experimento FTM/FTA:**

Tabela C.5 – Dados do Experimento FTM/FTA – 5º Caso – 6 Relógios.

T1	T2	T3	T4	T5	T6	Delta 1	Delta 2	Delta 3	Delta 4	Delta 5	Delta 6	Menor Delta	Maior Delta	Média	Correção	Ciclo	Referência	T1 Sem Correção
0	0	0	0	0	0	0	0	0	0	0	0	0,1	0,4	0,25	0,245	T1	0	0
0,6	0,5	0,5	0,5	0,5	0,5	0,1	0	0	0	0	0	0	0	0	-0,005	T2	0,5	0,6
1,2	1	1	1	1	1	0,2	0	0	0	0	0	0	0	0	-0,005	T3	1	1,2
1,8	1,5	1,5	1,5	1,5	1,5	0,3	0	0	0	0	0	0	0	0	-0,005	T4	1,5	1,8
2,4	2	2	2	2	2	0,4	0	0	0	0	0	0	0	0	-0,005	T5	2	2,4
3	2,5	2,5	2,5	2,5	2,5	0,5	0	0	0	0	0	0	0	0	-0,005	T6	2,5	3
3,355	3,005	3,005	3,005	3,005	3,005	0	-0,35	-0,35	-0,35	-0,35	-0,35	0,45	0,75	0,6	0,595	T1	3	3,6
3,955	3,505	3,505	3,505	3,505	3,505	0,45	0	0	0	0	0	0	0	0	-0,005	T2	3,5	4,2
4,555	4,005	4,005	4,005	4,005	4,005	0,55	0	0	0	0	0	0	0	0	-0,005	T3	4	4,8
5,155	4,505	4,505	4,505	4,505	4,505	0,65	0	0	0	0	0	0	0	0	-0,005	T4	4,5	5,4
5,755	5,005	5,005	5,005	5,005	5,005	0,75	0	0	0	0	0	0	0	0	-0,005	T5	5	6
6,355	5,505	5,505	5,505	5,505	5,505	0,85	0	0	0	0	0	0	0	0	-0,005	T6	5,5	6,6
6,36	6,01	6,01	6,01	6,01	6,01	0	-0,35	-0,35	-0,35	-0,35	-0,35	0,45	0,75	0,6	0,595	T1	6	7,2
6,96	6,51	6,51	6,51	6,51	6,51	0,45	0	0	0	0	0	0	0	0	-0,005	T2	6,5	7,8
7,56	7,01	7,01	7,01	7,01	7,01	0,55	0	0	0	0	0	0	0	0	-0,005	T3	7	8,4
8,16	7,51	7,51	7,51	7,51	7,51	0,65	0	0	0	0	0	0	0	0	-0,005	T4	7,5	9
8,76	8,01	8,01	8,01	8,01	8,01	0,75	0	0	0	0	0	0	0	0	-0,005	T5	8	9,6
9,36	8,51	8,51	8,51	8,51	8,51	0,85	0	0	0	0	0	0	0	0	-0,005	T6	8,5	10,2
9,365	9,015	9,015	9,015	9,015	9,015	0	-0,35	-0,35	-0,35	-0,35	-0,35	0,45	0,75	0,6	0,595	T1	9	10,8
9,965	9,515	9,515	9,515	9,515	9,515	0,45	0	0	0	0	0	0	0	0	-0,005	T2	9,5	11,4
10,565	10,015	10,015	10,015	10,015	10,015	0,55	0	0	0	0	0	0	0	0	-0,005	T3	10	12
11,165	10,515	10,515	10,515	10,515	10,515	0,65	0	0	0	0	0	0	0	0	-0,005	T4	10,5	12,6

(Continua)

Tabela C5 - Conclusão

11,765	11,015	11,015	11,015	11,015	11,015	0,75	0	0	0	0	0	0	0	0	0	-0,005	T5	11	13,2
12,365	11,515	11,515	11,515	11,515	11,515	0,85	0	0	0	0	0	0	0	0	0	-0,005	T6	11,5	13,8
12,37	12,02	12,02	12,02	12,02	12,02	0	-0,35	-0,35	-0,35	-0,35	-0,35	0,45	0,75	0,6	0,595	T1	12	14,4	
12,97	12,52	12,52	12,52	12,52	12,52	0,45	0	0	0	0	0	0	0	0	0	-0,005	T2	12,5	15
13,57	13,02	13,02	13,02	13,02	13,02	0,55	0	0	0	0	0	0	0	0	0	-0,005	T3	13	15,6
14,17	13,52	13,52	13,52	13,52	13,52	0,65	0	0	0	0	0	0	0	0	0	-0,005	T4	13,5	16,2
14,77	14,02	14,02	14,02	14,02	14,02	0,75	0	0	0	0	0	0	0	0	0	-0,005	T5	14	16,8
15,37	14,52	14,52	14,52	14,52	14,52	0,85	0	0	0	0	0	0	0	0	0	-0,005	T6	14,5	17,4
15,375	15,025	15,025	15,025	15,025	15,025	0	-0,35	-0,35	-0,35	-0,35	-0,35	0,45	0,75	0,6	0,595	T1	15	18	
15,975	15,525	15,525	15,525	15,525	15,525	0,45	0	0	0	0	0	0	0	0	0	-0,005	T2	15,5	18,6
16,575	16,025	16,025	16,025	16,025	16,025	0,55	0	0	0	0	0	0	0	0	0	-0,005	T3	16	19,2
17,175	16,525	16,525	16,525	16,525	16,525	0,65	0	0	0	0	0	0	0	0	0	-0,005	T4	16,5	19,8
17,775	17,025	17,025	17,025	17,025	17,025	0,75	0	0	0	0	0	0	0	0	0	-0,005	T5	17	20,4
18,375	17,525	17,525	17,525	17,525	17,525	0,85	0	0	0	0	0	0	0	0	0	-0,005	T6	17,5	21
18,38	18,03	18,03	18,03	18,03	18,03	0,35	-0,35	-0,35	-0,35	-0,35	-0,35							18	21,6

### C.6 Tabela do 6º Caso do Experimento FTM/FTA:

Tabela C.6 – Dados do Experimento FTM – 6º Caso – 7 Relógios.

T1	T2	T3	T4	T5	T6	T7	Delta 1	Delta 2	Delta 3	Delta 4	Delta 5	Delta 6	Delta 7	Menor Delta	Maior Delta	Média	Correção	Ciclo	Referência
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0,25	0,125	0,12	T1	0
0,55	0,6	0,5	0,5	0,5	0,5	0,5	-0,05	0	-0,1	-0,1	-0,1	-0,1	-0,1	0	0,5	0,25	0,245	T2	0,5
1,1	1,2	1	1	1	1	1	0,1	0,2	0	0	0	0	0	0	0	0	-0,005	T3	1
1,65	1,8	1,5	1,5	1,5	1,5	1,5	0,15	0,3	0	0	0	0	0	0	0	0	-0,005	T4	1,5

(Continua)

Tabela C6 - Continuação

<b>2,2</b>	2,4	2	2	2	2	2	0,2	0,4	0	0	0	0	0	0	0	0	-0,005	<b>T5</b>	2
<b>2,75</b>	3	2,5	2,5	2,5	2,5	2,5	0,25	0,5	0	0	0	0	0	0	0	0	-0,005	<b>T6</b>	2,5
<b>3,3</b>	3,6	3	3	3	3	3	0,3	0,6	0	0	0	0	0	0	0	0	-0,005	<b>T7</b>	3
<b>3,73</b>	3,955	3,505	3,505	3,505	3,505	3,505	0	0,225	-	-	-	-	-	0	0,475	0,2375	0,2325	<b>T1</b>	3,5
<b>4,28</b>	4,555	4,005	4,005	4,005	4,005	4,005	-	0	-0,55	-0,55	-0,55	-0,55	-0,55	0,225	0,95	0,5875	0,5825	<b>T2</b>	4
<b>4,83</b>	5,155	4,505	4,505	4,505	4,505	4,505	0,325	0,65	0	0	0	0	0	-0,225	0	-	-0,1175	<b>T3</b>	4,5
<b>5,38</b>	5,755	5,005	5,005	5,005	5,005	5,005	0,375	0,75	0	0	0	0	0	-0,225	0	-	-0,1175	<b>T4</b>	5
<b>5,93</b>	6,355	5,505	5,505	5,505	5,505	5,505	0,425	0,85	0	0	0	0	0	-0,225	0	-	-0,1175	<b>T5</b>	5,5
<b>6,48</b>	6,955	6,005	6,005	6,005	6,005	6,005	0,475	0,95	0	0	0	0	0	-0,225	0	-	-0,1175	<b>T6</b>	6
<b>7,03</b>	7,555	6,505	6,505	6,505	6,505	6,505	0,525	1,05	0	0	0	0	0	-0,225	0	-	-0,1175	<b>T7</b>	6,5
<b>7,3475</b>	7,5725	7,1225	7,1225	7,1225	7,1225	7,1225	0	0,225	-	-	-	-	-	0	0,475	0,2375	0,2325	<b>T1</b>	7
<b>7,8975</b>	8,1725	7,6225	7,6225	7,6225	7,6225	7,6225	-	0	-0,55	-0,55	-0,55	-0,55	-0,55	0,225	0,95	0,5875	0,5825	<b>T2</b>	7,5
<b>8,4475</b>	8,7725	8,1225	8,1225	8,1225	8,1225	8,1225	0,325	0,65	0	0	0	0	0	-0,225	0	-	-0,1175	<b>T3</b>	8
<b>8,9975</b>	9,3725	8,6225	8,6225	8,6225	8,6225	8,6225	0,375	0,75	0	0	0	0	0	-0,225	0	-	-0,1175	<b>T4</b>	8,5
<b>9,5475</b>	9,9725	9,1225	9,1225	9,1225	9,1225	9,1225	0,425	0,85	0	0	0	0	0	-0,225	0	-	-0,1175	<b>T5</b>	9
<b>10,0975</b>	10,5725	9,6225	9,6225	9,6225	9,6225	9,6225	0,475	0,95	0	0	0	0	0	-0,225	0	-	-0,1175	<b>T6</b>	9,5

(Continua)

Tabela C6 - Continuação

<b>10,6475</b>	11,1725	10,1225	10,1225	10,1225	10,1225	10,1225	0,525	1,05	0	0	0	0	0	-0,225	0	-	-0,1175	<b>T7</b>	10
<b>10,965</b>	11,19	10,74	10,74	10,74	10,74	10,74	0	0,225	-	-	-	-	-	0	0,475	0,2375	0,2325	<b>T1</b>	10,5
<b>11,515</b>	11,79	11,24	11,24	11,24	11,24	11,24	-	0	-0,55	-0,55	-0,55	-0,55	-0,55	0,225	0,95	0,5875	0,5825	<b>T2</b>	11
<b>12,065</b>	12,39	11,74	11,74	11,74	11,74	11,74	0,325	0,65	0	0	0	0	0	-0,225	0	-	-0,1175	<b>T3</b>	11,5
<b>12,615</b>	12,99	12,24	12,24	12,24	12,24	12,24	0,375	0,75	0	0	0	0	0	-0,225	0	-	-0,1175	<b>T4</b>	12
<b>13,165</b>	13,59	12,74	12,74	12,74	12,74	12,74	0,425	0,85	0	0	0	0	0	-0,225	0	-	-0,1175	<b>T5</b>	12,5
<b>13,715</b>	14,19	13,24	13,24	13,24	13,24	13,24	0,475	0,95	0	0	0	0	0	-0,225	0	-	-0,1175	<b>T6</b>	13
<b>14,265</b>	14,79	13,74	13,74	13,74	13,74	13,74	0,525	1,05	0	0	0	0	0	-0,225	0	-	-0,1175	<b>T7</b>	13,5
<b>14,5825</b>	14,8075	14,3575	14,3575	14,3575	14,3575	14,3575	0	0,225	-	-	-	-	-	0	0,475	0,2375	0,2325	<b>T1</b>	14
<b>15,1325</b>	15,4075	14,8575	14,8575	14,8575	14,8575	14,8575	-	0	-0,55	-0,55	-0,55	-0,55	-0,55	0,225	0,95	0,5875	0,5825	<b>T2</b>	14,5
<b>15,6825</b>	16,0075	15,3575	15,3575	15,3575	15,3575	15,3575	0,325	0,65	0	0	0	0	0	-0,225	0	-	-0,1175	<b>T3</b>	15
<b>16,2325</b>	16,6075	15,8575	15,8575	15,8575	15,8575	15,8575	0,375	0,75	0	0	0	0	0	-0,225	0	-	-0,1175	<b>T4</b>	15,5
<b>16,7825</b>	17,2075	16,3575	16,3575	16,3575	16,3575	16,3575	0,425	0,85	0	0	0	0	0	-0,225	0	-	-0,1175	<b>T5</b>	16
<b>17,3325</b>	17,8075	16,8575	16,8575	16,8575	16,8575	16,8575	0,475	0,95	0	0	0	0	0	-0,225	0	-	-0,1175	<b>T6</b>	16,5

(Continua)



Tabela C6 - Conclusão

17,8825	18,4075	17,3575	17,3575	17,3575	17,3575	17,3575	0,525	1,05	0	0	0	0	0	-0,225	0	-	-0,1175	T7	17
18,2	18,425	17,975	17,975	17,975	17,975	17,975	0	0,225	-	-	-	-	-	0	0,475	0,2375	0,2325	T1	17,5
18,75	19,025	18,475	18,475	18,475	18,475	18,475	-	0	-0,55	-0,55	-0,55	-0,55	-0,55	0,225	0,95	0,5875	0,5825	T2	18
19,3	19,625	18,975	18,975	18,975	18,975	18,975	0,325	0,65	0	0	0	0	0	-0,225	0	-	-0,1175	T3	18,5
19,85	20,225	19,475	19,475	19,475	19,475	19,475	0,375	0,75	0	0	0	0	0	-0,225	0	-	-0,1175	T4	19
20,4	20,825	19,975	19,975	19,975	19,975	19,975	0,425	0,85	0	0	0	0	0	-0,225	0	-	-0,1175	T5	19,5
20,95	21,425	20,475	20,475	20,475	20,475	20,475	0,475	0,95	0	0	0	0	0	-0,225	0	-	-0,1175	T6	20
21,5	22,025	20,975	20,975	20,975	20,975	20,975	0,525	1,05	0	0	0	0	0	-0,225	0	-	-0,1175	T7	20,5
21,8175	22,0425	21,5925	21,5925	21,5925	21,5925	21,5925	0	0,225	-	-	-	-	-						21

Tabela C.7 – Dados do Experimento FTA – 6º Caso – 7 Relógios.

T1	T2	T3	T4	T5	T6	T7	Delta 1	Delta 2	Delta 3	Delta 4	Delta 5	Delta 6	Delta 7	Menor Delta	Maior Delta	Média	Correção	Ciclo	Referência
0	0	0	0	0	0	0	0	0	0	0	0	0	0	-0,05	0,3	0,14	0,135	T1	0
0,55	0,6	0,5	0,5	0,5	0,5	0,5	-0,05	0	-0,1	-0,1	-0,1	-0,1	-0,1	0	0,6	0,28	0,275	T2	0,5
1,1	1,2	1	1	1	1	1	0,1	0,2	0	0	0	0	0	-0,1	0	0	-0,005	T3	1
1,65	1,8	1,5	1,5	1,5	1,5	1,5	0,15	0,3	0	0	0	0	0	-0,1	0	0	-0,005	T4	1,5
2,2	2,4	2	2	2	2	2	0,2	0,4	0	0	0	0	0	-0,1	0	0	-0,005	T5	2

(Continua)

Tabela C7 - Continuação

2,75	3	2,5	2,5	2,5	2,5	2,5	0,25	0,5	0	0	0	0	0	-0,1	0	0	-0,005	T6	2,5
3,3	3,6	3	3	3	3	3	0,3	0,6	0	0	0	0	0	-0,1	0	0	-0,005	T7	3
3,715	3,925	3,505	3,505	3,505	3,505	3,505	0	0,21	-0,21	-0,21	-0,21	-0,21	-0,21	-0,26	0,51	0,308	0,303	T1	3,5
4,265	4,525	4,005	4,005	4,005	4,005	4,005	-0,26	0	-0,52	-0,52	-0,52	-0,52	-0,52	0	1,02	0,658	0,653	T2	4
4,815	5,125	4,505	4,505	4,505	4,505	4,505	0,31	0,62	0	0	0	0	0	-0,52	0	-0,042	-0,047	T3	4,5
5,365	5,725	5,005	5,005	5,005	5,005	5,005	0,36	0,72	0	0	0	0	0	-0,52	0	-0,042	-0,047	T4	5
5,915	6,325	5,505	5,505	5,505	5,505	5,505	0,41	0,82	0	0	0	0	0	-0,52	0	-0,042	-0,047	T5	5,5
6,465	6,925	6,005	6,005	6,005	6,005	6,005	0,46	0,92	0	0	0	0	0	-0,52	0	-0,042	-0,047	T6	6
7,015	7,525	6,505	6,505	6,505	6,505	6,505	0,51	1,02	0	0	0	0	0	-0,52	0	-0,042	-0,047	T7	6,5
7,262	7,472	7,052	7,052	7,052	7,052	7,052	0	0,21	-0,21	-0,21	-0,21	-0,21	-0,21	-0,26	0,51	0,308	0,303	T1	7
7,812	8,072	7,552	7,552	7,552	7,552	7,552	-0,26	0	-0,52	-0,52	-0,52	-0,52	-0,52	0	1,02	0,658	0,653	T2	7,5
8,362	8,672	8,052	8,052	8,052	8,052	8,052	0,31	0,62	0	0	0	0	0	-0,52	0	-0,042	-0,047	T3	8
8,912	9,272	8,552	8,552	8,552	8,552	8,552	0,36	0,72	0	0	0	0	0	-0,52	0	-0,042	-0,047	T4	8,5
9,462	9,872	9,052	9,052	9,052	9,052	9,052	0,41	0,82	0	0	0	0	0	-0,52	0	-0,042	-0,047	T5	9
10,012	10,472	9,552	9,552	9,552	9,552	9,552	0,46	0,92	0	0	0	0	0	-0,52	0	-0,042	-0,047	T6	9,5
10,562	11,072	10,052	10,052	10,052	10,052	10,052	0,51	1,02	0	0	0	0	0	-0,52	0	-0,042	-0,047	T7	10
10,809	11,019	10,599	10,599	10,599	10,599	10,599	0	0,21	-0,21	-0,21	-0,21	-0,21	-0,21	-0,26	0,51	0,308	0,303	T1	10,5
11,359	11,619	11,099	11,099	11,099	11,099	11,099	-0,26	0	-0,52	-0,52	-0,52	-0,52	-0,52	0	1,02	0,658	0,653	T2	11
11,909	12,219	11,599	11,599	11,599	11,599	11,599	0,31	0,62	0	0	0	0	0	-0,52	0	-0,042	-0,047	T3	11,5
12,459	12,819	12,099	12,099	12,099	12,099	12,099	0,36	0,72	0	0	0	0	0	-0,52	0	-0,042	-0,047	T4	12
13,009	13,419	12,599	12,599	12,599	12,599	12,599	0,41	0,82	0	0	0	0	0	-0,52	0	-0,042	-0,047	T5	12,5
13,559	14,019	13,099	13,099	13,099	13,099	13,099	0,46	0,92	0	0	0	0	0	-0,52	0	-0,042	-0,047	T6	13
14,109	14,619	13,599	13,599	13,599	13,599	13,599	0,51	1,02	0	0	0	0	0	-0,52	0	-0,042	-0,047	T7	13,5
14,356	14,566	14,146	14,146	14,146	14,146	14,146	0	0,21	-0,21	-0,21	-0,21	-0,21	-0,21	-0,26	0,51	0,308	0,303	T1	14
14,906	15,166	14,646	14,646	14,646	14,646	14,646	-0,26	0	-0,52	-0,52	-0,52	-0,52	-0,52	0	1,02	0,658	0,653	T2	14,5
15,456	15,766	15,146	15,146	15,146	15,146	15,146	0,31	0,62	0	0	0	0	0	-0,52	0	-0,042	-0,047	T3	15
16,006	16,366	15,646	15,646	15,646	15,646	15,646	0,36	0,72	0	0	0	0	0	-0,52	0	-0,042	-0,047	T4	15,5
16,556	16,966	16,146	16,146	16,146	16,146	16,146	0,41	0,82	0	0	0	0	0	-0,52	0	-0,042	-0,047	T5	16

(Continua)

Tabela C7 - Conclusão

17,106	17,566	16,646	16,646	16,646	16,646	16,646	0,46	0,92	0	0	0	0	0	0	-0,52	0	-0,042	-0,047	T6	16,5
17,656	18,166	17,146	17,146	17,146	17,146	17,146	0,51	1,02	0	0	0	0	0	0	-0,52	0	-0,042	-0,047	T7	17
17,903	18,113	17,693	17,693	17,693	17,693	17,693	0	0,21	-0,21	-0,21	-0,21	-0,21	-0,21	-0,26	0,51	0,308	0,303	T1	17,5	
18,453	18,713	18,193	18,193	18,193	18,193	18,193	-0,26	0	-0,52	-0,52	-0,52	-0,52	-0,52	0	1,02	0,658	0,653	T2	18	
19,003	19,313	18,693	18,693	18,693	18,693	18,693	0,31	0,62	0	0	0	0	0	-0,52	0	-0,042	-0,047	T3	18,5	
19,553	19,913	19,193	19,193	19,193	19,193	19,193	0,36	0,72	0	0	0	0	0	-0,52	0	-0,042	-0,047	T4	19	
20,103	20,513	19,693	19,693	19,693	19,693	19,693	0,41	0,82	0	0	0	0	0	-0,52	0	-0,042	-0,047	T5	19,5	
20,653	21,113	20,193	20,193	20,193	20,193	20,193	0,46	0,92	0	0	0	0	0	-0,52	0	-0,042	-0,047	T6	20	
21,203	21,713	20,693	20,693	20,693	20,693	20,693	0,51	1,02	0	0	0	0	0	-0,52	0	-0,042	-0,047	T7	20,5	
21,45	21,66	21,24	21,24	21,24	21,24	21,24	0	0,21	-0,21	-0,21	-0,21	-0,21	-0,21						21	

C.7 Tabela do 7º Caso do Experimento FTM/FTA:

Tabela C.8 – Dados do Experimento FTM – 7º Caso – 7 Relógios.

T1	T2	T3	T4	T5	T6	T7	Delta 1	Delta 2	Delta 3	Delta 4	Delta 5	Delta 6	Delta 7	Menor Delta	Maior Delta	Média	Correção	Ciclo	Referência
0	0	0	0	0	0	0	0	0	0	0	0	0	0	-2,5	0	-1,25	-1,255	T1	0
0	0	0,5	0,5	0,5	0,5	0,5	0	0	0,5	0,5	0,5	0,5	0,5	-2,5	0	-1,25	-1,255	T2	0,5
0	0	1	1	1	1	1	-1	-1	0	0	0	0	0	0	0	0	-0,005	T3	1
0	0	1,5	1,5	1,5	1,5	1,5	-1,5	-1,5	0	0	0	0	0	0	0	0	-0,005	T4	1,5
0	0	2	2	2	2	2	-2	-2	0	0	0	0	0	0	0	0	-0,005	T5	2
0	0	2,5	2,5	2,5	2,5	2,5	-2,5	-2,5	0	0	0	0	0	0	0	0	-0,005	T6	2,5
0	0	3	3	3	3	3	-3	-3	0	0	0	0	0	0	0	0	-0,005	T7	3
1,255	1,255	3,505	3,505	3,505	3,505	3,505	0	0	2,25	2,25	2,25	2,25	2,25	-4,75	0	-2,375	-2,38	T1	3,5

(Continua)

Tabela C8 - Continuação

1,255	1,255	4,005	4,005	4,005	4,005	4,005	0	0	2,75	2,75	2,75	2,75	2,75	-4,75	0	-2,375	-2,38	<b>T2</b>	4
1,255	1,255	4,505	4,505	4,505	4,505	4,505	-3,25	-3,25	0	0	0	0	0	0	2,25	1,125	1,12	<b>T3</b>	4,5
1,255	1,255	5,005	5,005	5,005	5,005	5,005	-3,75	-3,75	0	0	0	0	0	0	2,25	1,125	1,12	<b>T4</b>	5
1,255	1,255	5,505	5,505	5,505	5,505	5,505	-4,25	-4,25	0	0	0	0	0	0	2,25	1,125	1,12	<b>T5</b>	5,5
1,255	1,255	6,005	6,005	6,005	6,005	6,005	-4,75	-4,75	0	0	0	0	0	0	2,25	1,125	1,12	<b>T6</b>	6
1,255	1,255	6,505	6,505	6,505	6,505	6,505	-5,25	-5,25	0	0	0	0	0	0	2,25	1,125	1,12	<b>T7</b>	6,5
3,635	3,635	5,885	5,885	5,885	5,885	5,885	0	0	2,25	2,25	2,25	2,25	2,25	-4,75	0	-2,375	-2,38	<b>T1</b>	7
3,635	3,635	6,385	6,385	6,385	6,385	6,385	0	0	2,75	2,75	2,75	2,75	2,75	-4,75	0	-2,375	-2,38	<b>T2</b>	7,5
3,635	3,635	6,885	6,885	6,885	6,885	6,885	-3,25	-3,25	0	0	0	0	0	0	2,25	1,125	1,12	<b>T3</b>	8
3,635	3,635	7,385	7,385	7,385	7,385	7,385	-3,75	-3,75	0	0	0	0	0	0	2,25	1,125	1,12	<b>T4</b>	8,5
3,635	3,635	7,885	7,885	7,885	7,885	7,885	-4,25	-4,25	0	0	0	0	0	0	2,25	1,125	1,12	<b>T5</b>	9
3,635	3,635	8,385	8,385	8,385	8,385	8,385	-4,75	-4,75	0	0	0	0	0	0	2,25	1,125	1,12	<b>T6</b>	9,5
3,635	3,635	8,885	8,885	8,885	8,885	8,885	-5,25	-5,25	0	0	0	0	0	0	2,25	1,125	1,12	<b>T7</b>	10
6,015	6,015	8,265	8,265	8,265	8,265	8,265	0	0	2,25	2,25	2,25	2,25	2,25	-4,75	0	-2,375	-2,38	<b>T1</b>	10,5
6,015	6,015	8,765	8,765	8,765	8,765	8,765	0	0	2,75	2,75	2,75	2,75	2,75	-4,75	0	-2,375	-2,38	<b>T2</b>	11
6,015	6,015	9,265	9,265	9,265	9,265	9,265	-3,25	-3,25	0	0	0	0	0	0	2,25	1,125	1,12	<b>T3</b>	11,5
6,015	6,015	9,765	9,765	9,765	9,765	9,765	-3,75	-3,75	0	0	0	0	0	0	2,25	1,125	1,12	<b>T4</b>	12
6,015	6,015	10,265	10,265	10,265	10,265	10,265	-4,25	-4,25	0	0	0	0	0	0	2,25	1,125	1,12	<b>T5</b>	12,5
6,015	6,015	10,765	10,765	10,765	10,765	10,765	-4,75	-4,75	0	0	0	0	0	0	2,25	1,125	1,12	<b>T6</b>	13
6,015	6,015	11,265	11,265	11,265	11,265	11,265	-5,25	-5,25	0	0	0	0	0	0	2,25	1,125	1,12	<b>T7</b>	13,5
8,395	8,395	10,645	10,645	10,645	10,645	10,645	0	0	2,25	2,25	2,25	2,25	2,25	-4,75	0	-2,375	-2,38	<b>T1</b>	14
8,395	8,395	11,145	11,145	11,145	11,145	11,145	0	0	2,75	2,75	2,75	2,75	2,75	-4,75	0	-2,375	-2,38	<b>T2</b>	14,5
8,395	8,395	11,645	11,645	11,645	11,645	11,645	-3,25	-3,25	0	0	0	0	0	0	2,25	1,125	1,12	<b>T3</b>	15
8,395	8,395	12,145	12,145	12,145	12,145	12,145	-3,75	-3,75	0	0	0	0	0	0	2,25	1,125	1,12	<b>T4</b>	15,5
8,395	8,395	12,645	12,645	12,645	12,645	12,645	-4,25	-4,25	0	0	0	0	0	0	2,25	1,125	1,12	<b>T5</b>	16
8,395	8,395	13,145	13,145	13,145	13,145	13,145	-4,75	-4,75	0	0	0	0	0	0	2,25	1,125	1,12	<b>T6</b>	16,5
8,395	8,395	13,645	13,645	13,645	13,645	13,645	-5,25	-5,25	0	0	0	0	0	0	2,25	1,125	1,12	<b>T7</b>	17
10,775	10,775	13,025	13,025	13,025	13,025	13,025	0	0	2,25	2,25	2,25	2,25	2,25	-4,75	0	-2,375	-2,38	<b>T1</b>	17,5

(Continua)

Tabela C8 - Conclusão

10,775	10,775	13,525	13,525	13,525	13,525	13,525	0	0	2,75	2,75	2,75	2,75	2,75	-4,75	0	-2,375	-2,38	<b>T2</b>	18
10,775	10,775	14,025	14,025	14,025	14,025	14,025	-3,25	-3,25	0	0	0	0	0	0	2,25	1,125	1,12	<b>T3</b>	18,5
10,775	10,775	14,525	14,525	14,525	14,525	14,525	-3,75	-3,75	0	0	0	0	0	0	2,25	1,125	1,12	<b>T4</b>	19
10,775	10,775	15,025	15,025	15,025	15,025	15,025	-4,25	-4,25	0	0	0	0	0	0	2,25	1,125	1,12	<b>T5</b>	19,5
10,775	10,775	15,525	15,525	15,525	15,525	15,525	-4,75	-4,75	0	0	0	0	0	0	2,25	1,125	1,12	<b>T6</b>	20
10,775	10,775	16,025	16,025	16,025	16,025	16,025	-5,25	-5,25	0	0	0	0	0	0	2,25	1,125	1,12	<b>T7</b>	20,5
13,155	13,155	15,405	15,405	15,405	15,405	15,405	0	0	2,25	2,25	2,25	2,25	2,25						21

Tabela C.9 – Modelo do Experimento FTA – 7º Caso – 7 Relógios.

T1	T2	T3	T4	T5	T6	T7	Delta 1	Delta 2	Delta 3	Delta 4	Delta 5	Delta 6	Delta 7	Menor Delta	Maior Delta	Média	Correção	Ciclo	Referência
0	0	0	0	0	0	0	0	0	0	0	0	0	0	-3	0	-1,4	-1,405	<b>T1</b>	0
0	0	0,5	0,5	0,5	0,5	0,5	0	0	0,5	0,5	0,5	0,5	0,5	-3	0	-1,4	-1,405	<b>T2</b>	0,5
0	0	1	1	1	1	1	-1	-1	0	0	0	0	0	0	0,5	0	-0,005	<b>T3</b>	1
0	0	1,5	1,5	1,5	1,5	1,5	-1,5	-1,5	0	0	0	0	0	0	0,5	0	-0,005	<b>T4</b>	1,5
0	0	2	2	2	2	2	-2	-2	0	0	0	0	0	0	0,5	0	-0,005	<b>T5</b>	2
0	0	2,5	2,5	2,5	2,5	2,5	-2,5	-2,5	0	0	0	0	0	0	0,5	0	-0,005	<b>T6</b>	2,5
0	0	3	3	3	3	3	-3	-3	0	0	0	0	0	0	0,5	0	-0,005	<b>T7</b>	3
1,405	1,405	3,505	3,505	3,505	3,505	3,505	0	0	2,1	2,1	2,1	2,1	2,1	-5,1	0	-3,08	-3,085	<b>T1</b>	3,5
1,405	1,405	4,005	4,005	4,005	4,005	4,005	0	0	2,6	2,6	2,6	2,6	2,6	-5,1	0	-3,08	-3,085	<b>T2</b>	4
1,405	1,405	4,505	4,505	4,505	4,505	4,505	-3,1	-3,1	0	0	0	0	0	0	2,6	0,42	0,415	<b>T3</b>	4,5
1,405	1,405	5,005	5,005	5,005	5,005	5,005	-3,6	-3,6	0	0	0	0	0	0	2,6	0,42	0,415	<b>T4</b>	5
1,405	1,405	5,505	5,505	5,505	5,505	5,505	-4,1	-4,1	0	0	0	0	0	0	2,6	0,42	0,415	<b>T5</b>	5,5
1,405	1,405	6,005	6,005	6,005	6,005	6,005	-4,6	-4,6	0	0	0	0	0	0	2,6	0,42	0,415	<b>T6</b>	6
1,405	1,405	6,505	6,505	6,505	6,505	6,505	-5,1	-5,1	0	0	0	0	0	0	2,6	0,42	0,415	<b>T7</b>	6,5
4,49	4,49	6,59	6,59	6,59	6,59	6,59	0	0	2,1	2,1	2,1	2,1	2,1	-5,1	0	-3,08	-3,085	<b>T1</b>	7
4,49	4,49	7,09	7,09	7,09	7,09	7,09	0	0	2,6	2,6	2,6	2,6	2,6	-5,1	0	-3,08	-3,085	<b>T2</b>	7,5

(Continua)

Tabela C9 - Conclusão

4,49	4,49	7,59	7,59	7,59	7,59	7,59	-3,1	-3,1	0	0	0	0	0	0	2,6	0,42	0,415	T3	8
4,49	4,49	8,09	8,09	8,09	8,09	8,09	-3,6	-3,6	0	0	0	0	0	0	2,6	0,42	0,415	T4	8,5
4,49	4,49	8,59	8,59	8,59	8,59	8,59	-4,1	-4,1	0	0	0	0	0	0	2,6	0,42	0,415	T5	9
4,49	4,49	9,09	9,09	9,09	9,09	9,09	-4,6	-4,6	0	0	0	0	0	0	2,6	0,42	0,415	T6	9,5
4,49	4,49	9,59	9,59	9,59	9,59	9,59	-5,1	-5,1	0	0	0	0	0	0	2,6	0,42	0,415	T7	10
7,575	7,575	9,675	9,675	9,675	9,675	9,675	0	0	2,1	2,1	2,1	2,1	2,1	-5,1	0	-3,08	-3,085	T1	10,5
7,575	7,575	10,175	10,175	10,175	10,175	10,175	0	0	2,6	2,6	2,6	2,6	2,6	-5,1	0	-3,08	-3,085	T2	11
7,575	7,575	10,675	10,675	10,675	10,675	10,675	-3,1	-3,1	0	0	0	0	0	0	2,6	0,42	0,415	T3	11,5
7,575	7,575	11,175	11,175	11,175	11,175	11,175	-3,6	-3,6	0	0	0	0	0	0	2,6	0,42	0,415	T4	12
7,575	7,575	11,675	11,675	11,675	11,675	11,675	-4,1	-4,1	0	0	0	0	0	0	2,6	0,42	0,415	T5	12,5
7,575	7,575	12,175	12,175	12,175	12,175	12,175	-4,6	-4,6	0	0	0	0	0	0	2,6	0,42	0,415	T6	13
7,575	7,575	12,675	12,675	12,675	12,675	12,675	-5,1	-5,1	0	0	0	0	0	0	2,6	0,42	0,415	T7	13,5
10,66	10,66	12,76	12,76	12,76	12,76	12,76	0	0	2,1	2,1	2,1	2,1	2,1	-5,1	0	-3,08	-3,085	T1	14
10,66	10,66	13,26	13,26	13,26	13,26	13,26	0	0	2,6	2,6	2,6	2,6	2,6	-5,1	0	-3,08	-3,085	T2	14,5
10,66	10,66	13,76	13,76	13,76	13,76	13,76	-3,1	-3,1	0	0	0	0	0	0	2,6	0,42	0,415	T3	15
10,66	10,66	14,26	14,26	14,26	14,26	14,26	-3,6	-3,6	0	0	0	0	0	0	2,6	0,42	0,415	T4	15,5
10,66	10,66	14,76	14,76	14,76	14,76	14,76	-4,1	-4,1	0	0	0	0	0	0	2,6	0,42	0,415	T5	16
10,66	10,66	15,26	15,26	15,26	15,26	15,26	-4,6	-4,6	0	0	0	0	0	0	2,6	0,42	0,415	T6	16,5
10,66	10,66	15,76	15,76	15,76	15,76	15,76	-5,1	-5,1	0	0	0	0	0	0	2,6	0,42	0,415	T7	17
13,745	13,745	15,845	15,845	15,845	15,845	15,845	0	0	2,1	2,1	2,1	2,1	2,1	-5,1	0	-3,08	-3,085	T1	17,5
13,745	13,745	16,345	16,345	16,345	16,345	16,345	0	0	2,6	2,6	2,6	2,6	2,6	-5,1	0	-3,08	-3,085	T2	18
13,745	13,745	16,845	16,845	16,845	16,845	16,845	-3,1	-3,1	0	0	0	0	0	0	2,6	0,42	0,415	T3	18,5
13,745	13,745	17,345	17,345	17,345	17,345	17,345	-3,6	-3,6	0	0	0	0	0	0	2,6	0,42	0,415	T4	19
13,745	13,745	17,845	17,845	17,845	17,845	17,845	-4,1	-4,1	0	0	0	0	0	0	2,6	0,42	0,415	T5	19,5
13,745	13,745	18,345	18,345	18,345	18,345	18,345	-4,6	-4,6	0	0	0	0	0	0	2,6	0,42	0,415	T6	20
13,745	13,745	18,845	18,845	18,845	18,845	18,845	-5,1	-5,1	0	0	0	0	0	0	2,6	0,42	0,415	T7	20,5
16,83	16,83	18,93	18,93	18,93	18,93	18,93	0	0	2,1	2,1	2,1	2,1	2,1						21

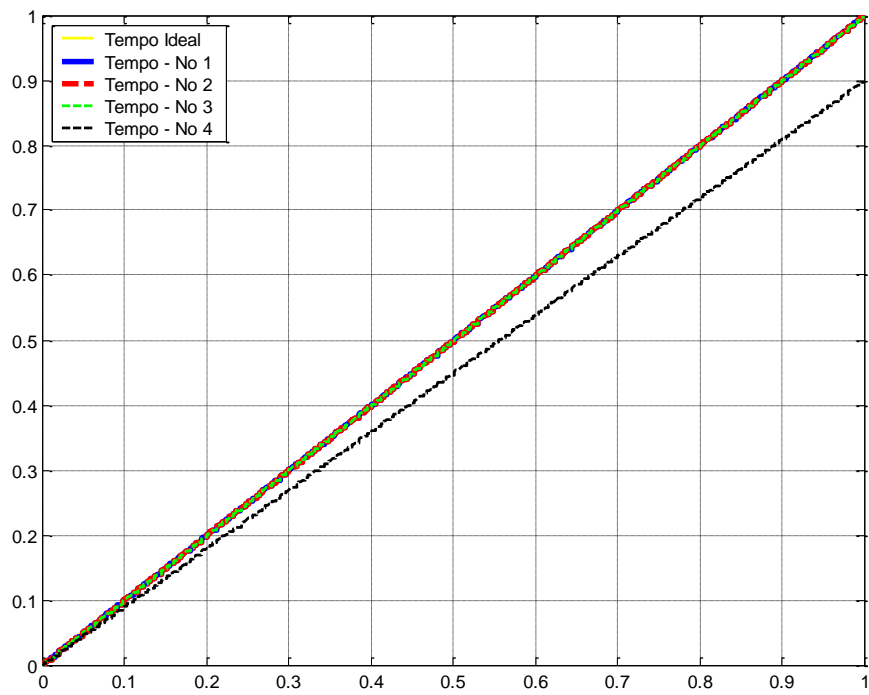
## **APÊNDICE D – SIMULAÇÃO DO 1º CASO – ARQUITETURA FEDERADA – ALGORITMO FTM**

Neste caso, foi feita uma simulação em um sistema de controle com arquitetura federada, descrito na seção anterior. A sincronização foi feita através do algoritmo FTM (*Welch-Lynch*), com arquitetura distribuída, isto é, não existe um relógio de referência. Nesta simulação variou-se o parâmetro de deriva do relógio do nó (subsistema) 4. A rede de comunicação foi feita por um protocolo de comunicação com acesso ao meio TDMA.

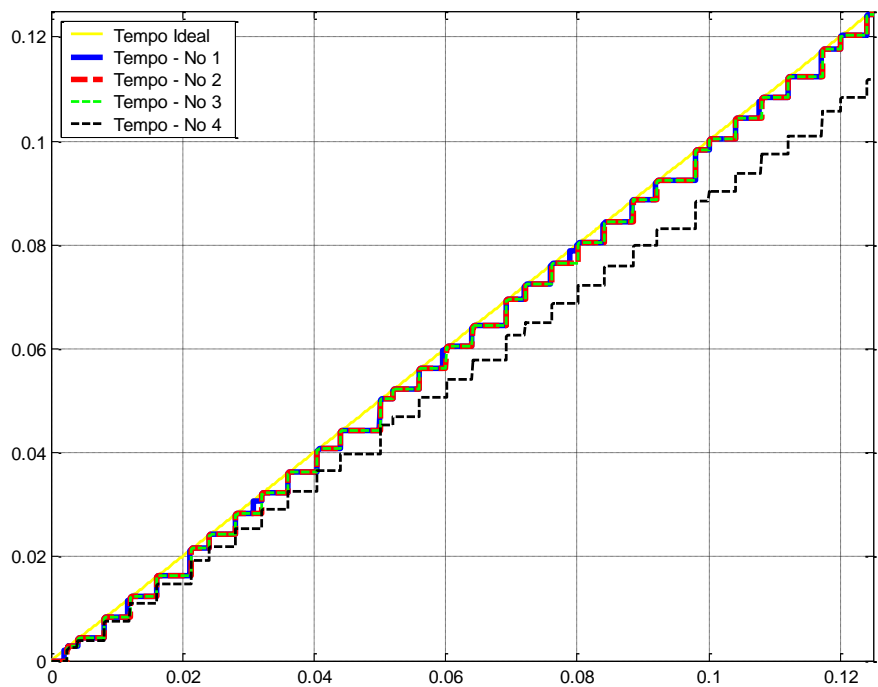
Este trabalho escolheu 3 taxas de deriva diferentes para variação do relógio:

- **Caso A: -10%** - O relógio está se atrasando em relação aos outros com taxa de 10%;
- **Caso B: 1%** - O relógio está se adiantando em relação aos outros com taxa de 1%;
- **Caso C: 10%** - O relógio está se adiantando em relação aos outros com taxa de 10%;

**No caso A, a uma taxa de deriva de -10%**, o relógio do nó 4 esta se atrasando em relação ao relógio ideal e aos outros relógios dos outros nós, como mostra a Figura D.1. Com isso, o sistema sofre uma pequena variação no sistema de controle do nó 4 que esta afetado pelo relógio.



(a)



(b)

Figura D.1 – (a) Linha do Tempo com deriva de -10%. (b) Aproximação da Linha do Tempo com deriva de -10%



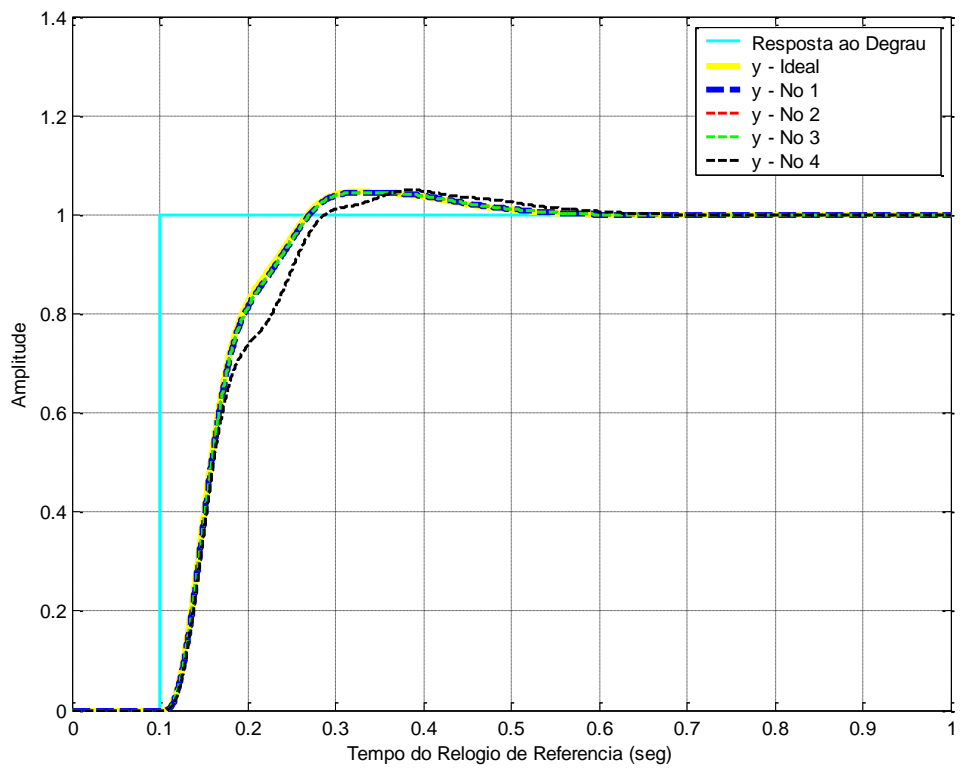


Figura D.2 – Resposta ao Degrau com deriva de -10%.

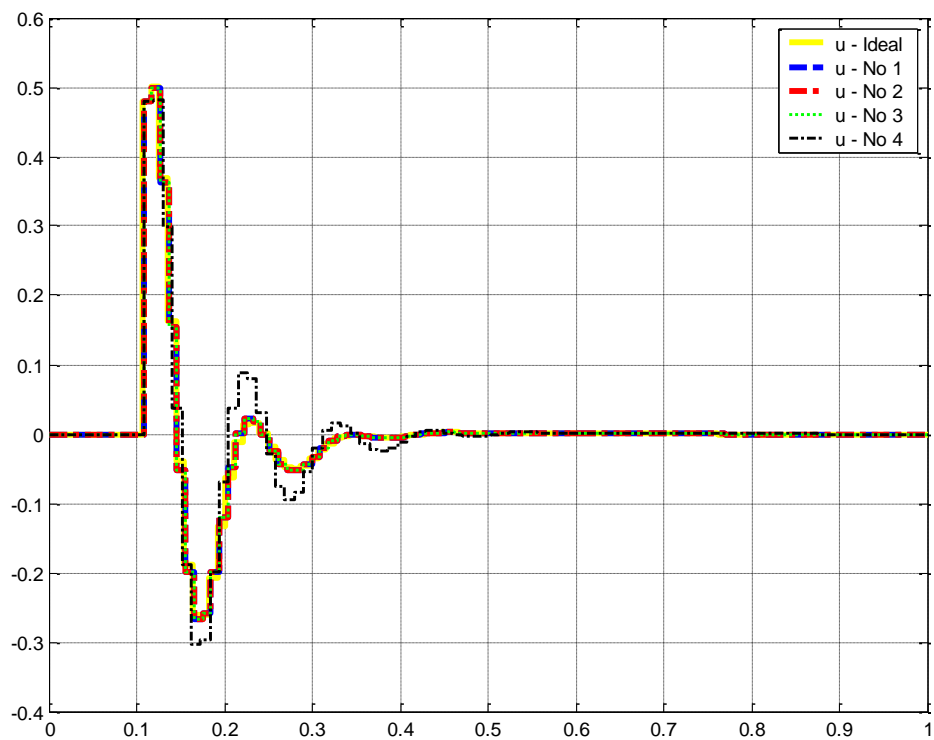


Figura D.3 – Lei de Controle com deriva de -10%.

A Figura D.2 mostra a resposta ao degrau do sistema de controle com a deriva de -10%. É possível ver que somente o nó 4 sofre variação em relação ao ideal e aos outros nós do sistema. Isto ocorre pois o relógio do nó 4 está atrasado em relação aos outros. A Figura D.3 mostra as leis de controle aplicadas a cada nó do sistema e do sistema ideal. Somente o nó 4 difere dos outros. Apesar de sofrer variação o sistema de controle não sofre grande impacto com uma taxa de deriva de -10%.

Mesmo com um pequeno impacto, vamos aplicar o algoritmo FTM de sincronização de relógios. A Figura D.5 mostra a linha do tempo com o algoritmo de sincronização FTM aplicado ao sistema. O período de resincronização é de 0.0040 segundos, isto é, a cada 0.0040 segundos o algoritmo FTM atualiza os valores de relógio com a correção calculada. Na Figura D.4 é possível ver o gráfico de correção dos relógios dos respectivos nós do sistema.

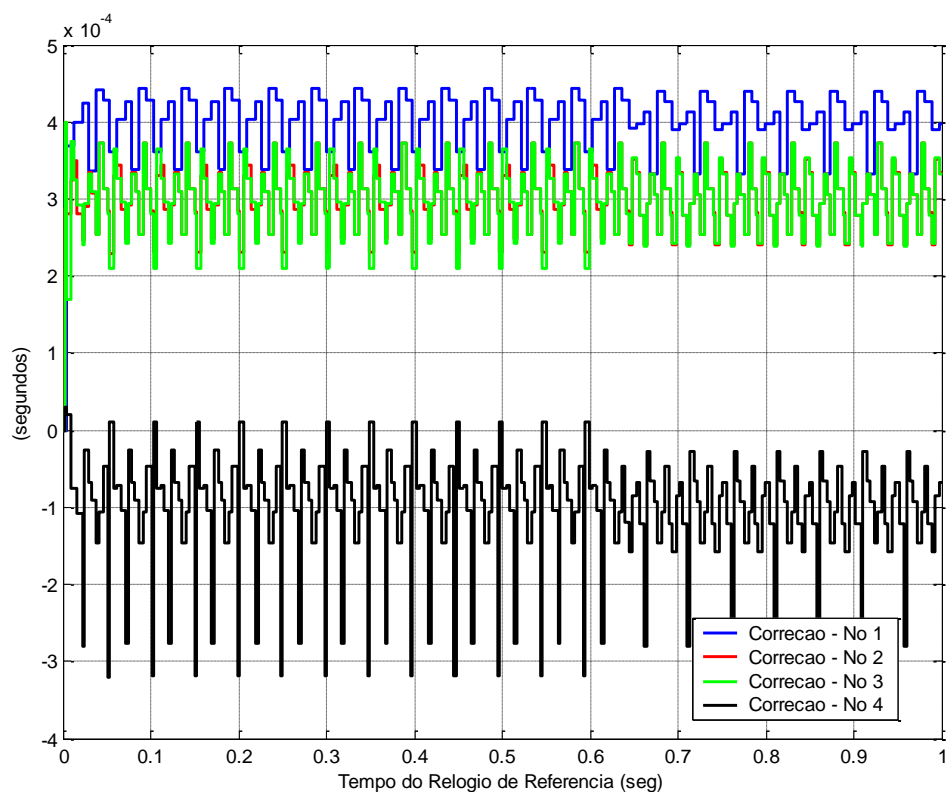
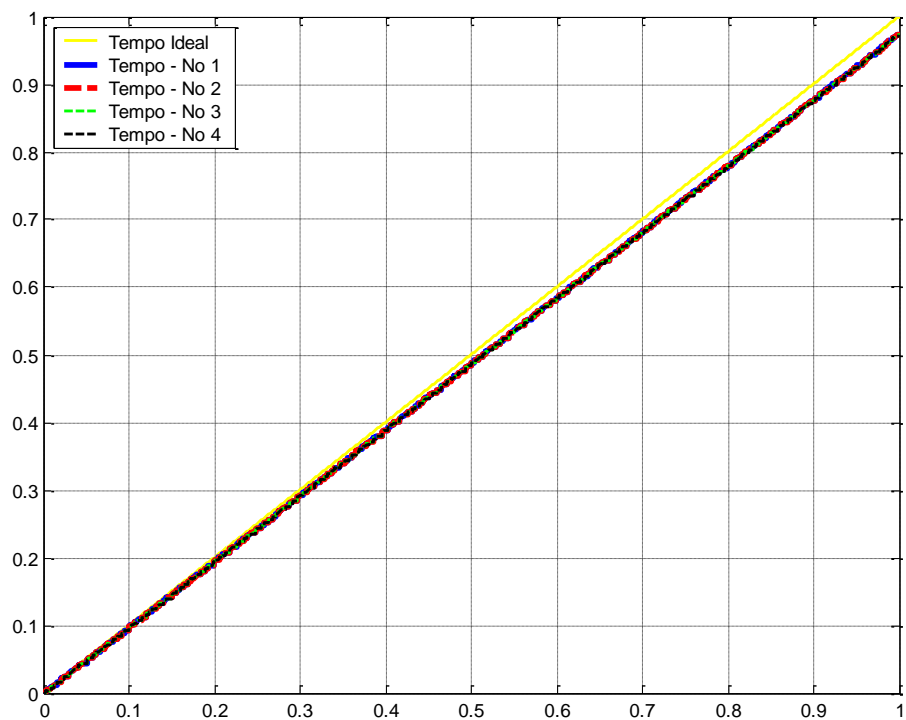
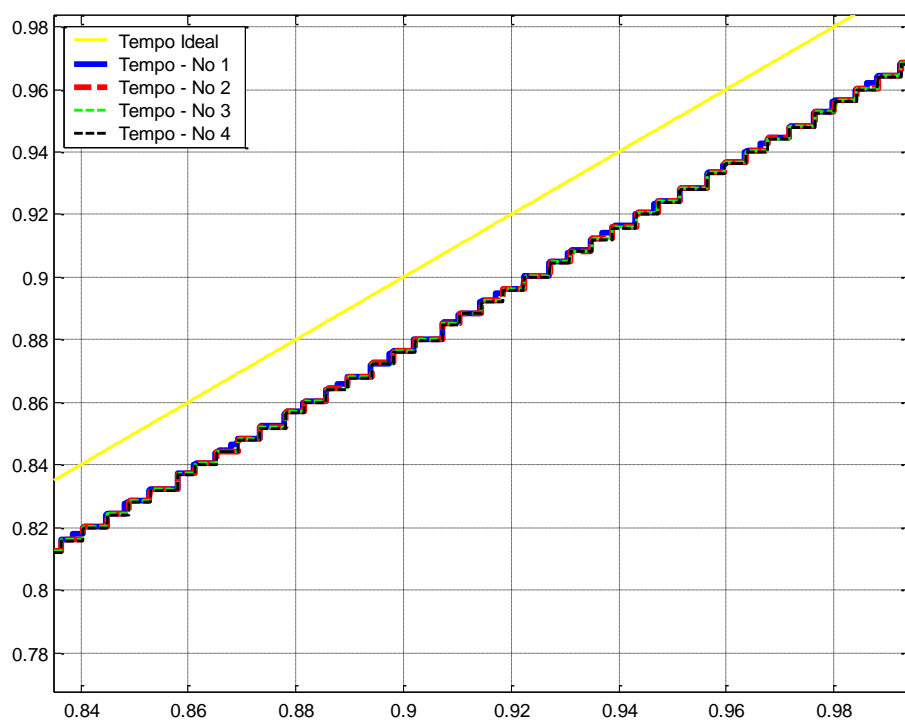


Figura D.4 – Correção Algoritmo FTM com deriva de -10%



(a)



(b)

Figura D.5 – (a) Linha do Tempo com deriva de -10%. Sincronizado (b) Aproximação.

A Figura D.4 mostra que o algoritmo FTM convergiu. Os nós 1, 2 e 3, que não estavam com erro de deriva, fizeram uma correção periódica em torno de 0.3 (ms) enquanto o nó 4, com -10% de taxa de deriva, teve uma correção periódica em torno de -0.1 (ms). Com isso, é possível analisar através do gráfico D.4 que a precisão dos relógios do sistema ficou em torno de 0.7 (ms). A Figura D.5 mostra que os relógios dos nós se sincronizaram entre si, apesar do tempo global sofrer uma pequena deriva em relação ao tempo ideal. A Figura D.6 mostra a resposta ao degrau do sistema de controle. E a Figura D.7 mostra as leis de controle do sistema.

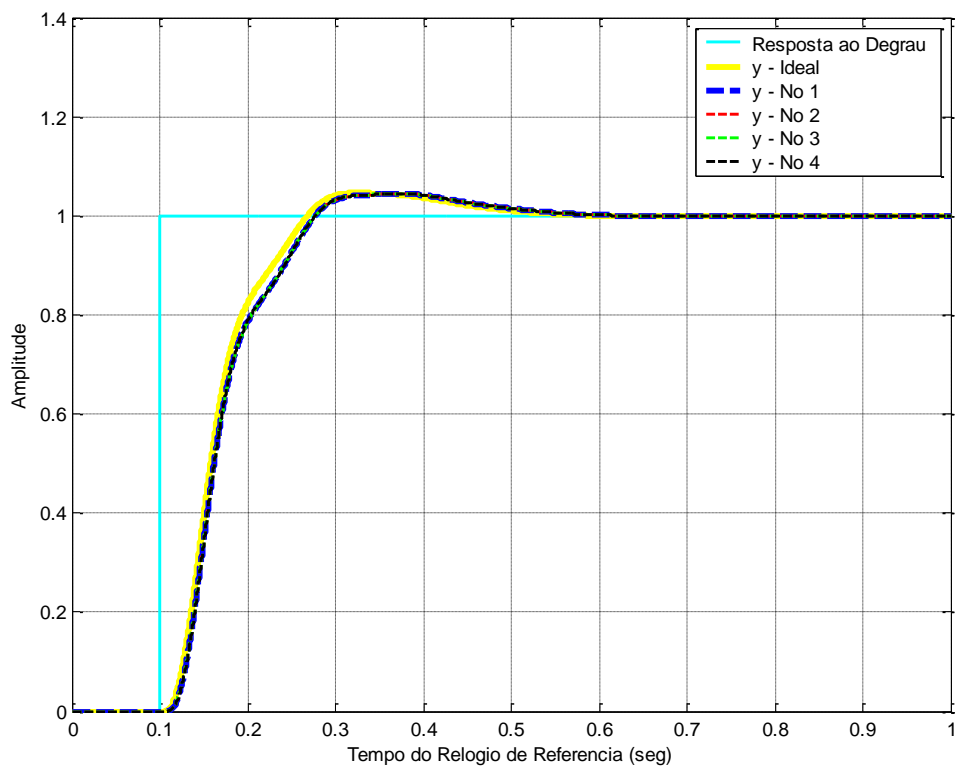


Figura D.6 – Resposta ao Degrau com deriva de -10% Sincronizado.

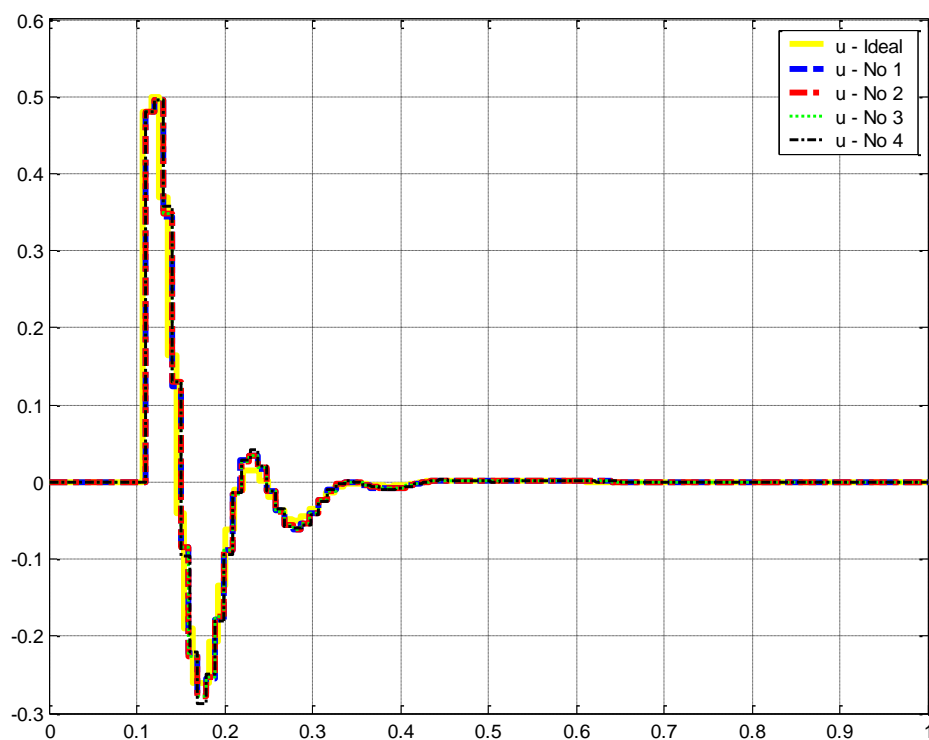
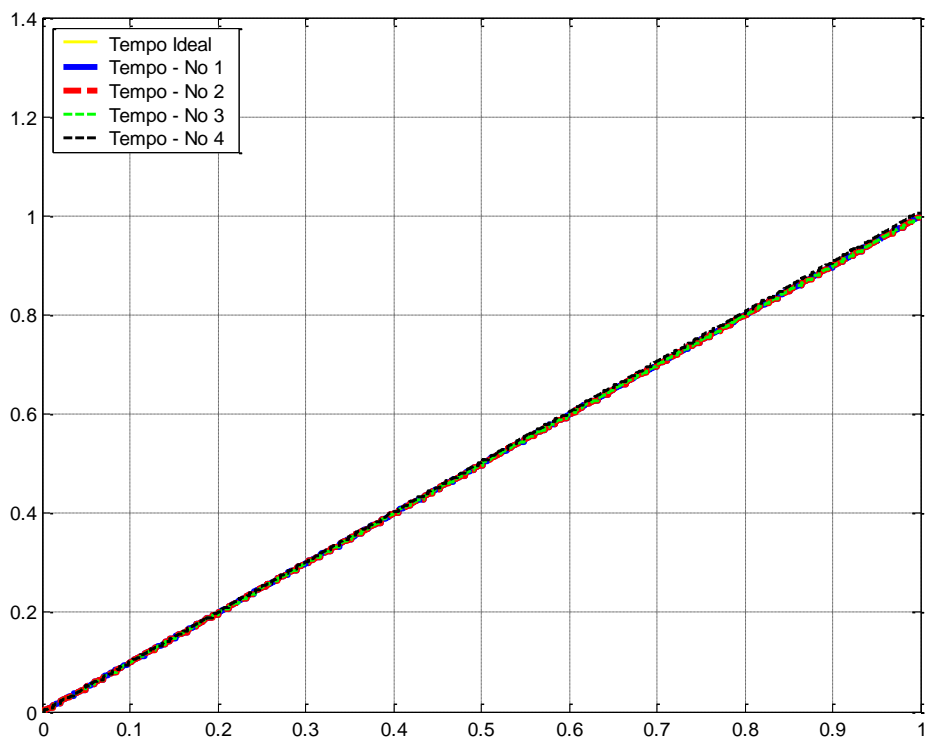


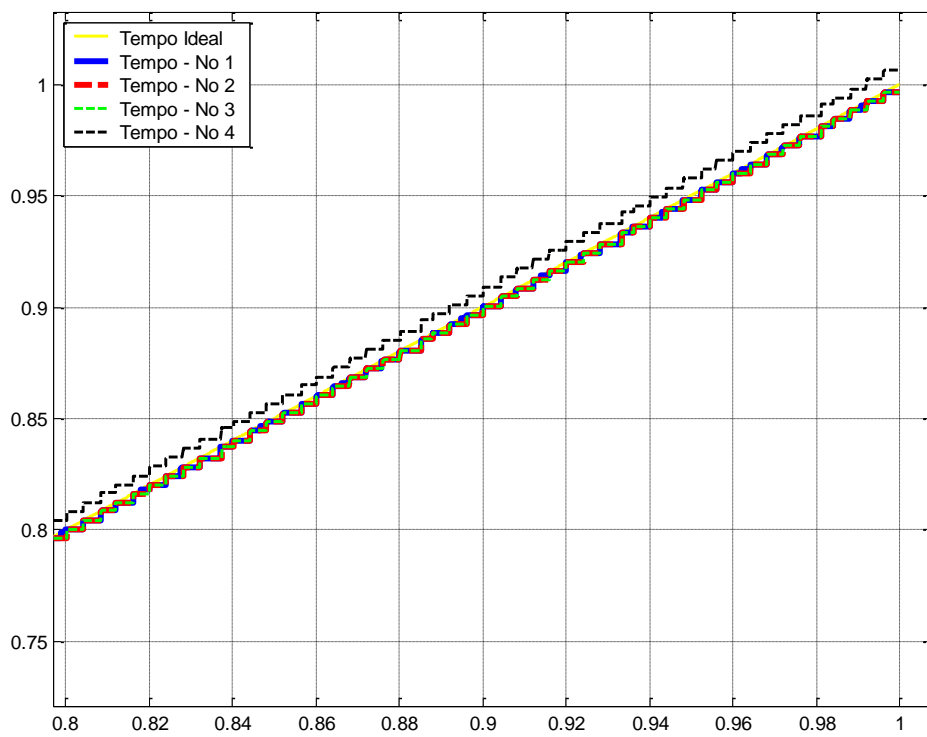
Figura D.7 – Lei de Controle com deriva de -10% Sincronizado.

Através das Figuras D.6 e D.7 é possível ver que a sincronização de relógios minimizou a diferença do tempo de resposta e das leis de controle de cada nó. Apesar do sistema ser um pouco atrasado do sistema ideal, os tempos de respostas estão sincronizados entre si. Neste caso o algoritmo mostrou-se muito eficiente corrigindo o erro de deriva presente no sistema.

A uma **taxa de deriva de 1%**, o relógio do nó 4 está se adiantando em relação ao relógio ideal e aos outros relógios dos outros nós, como mostra a Figura D.8. Com isso, o sistema sofre uma pequena variação, por causa do relógio, no sistema de controle do nó 4 que está afetado pelo relógio. Esta variação é muito pequena, e portanto é de se esperar que o sistema de controle não seja muito afetado.



(a)



(b)

Figura D.8 – (a) Linha do Tempo com deriva de 1%. (b) Aproximação

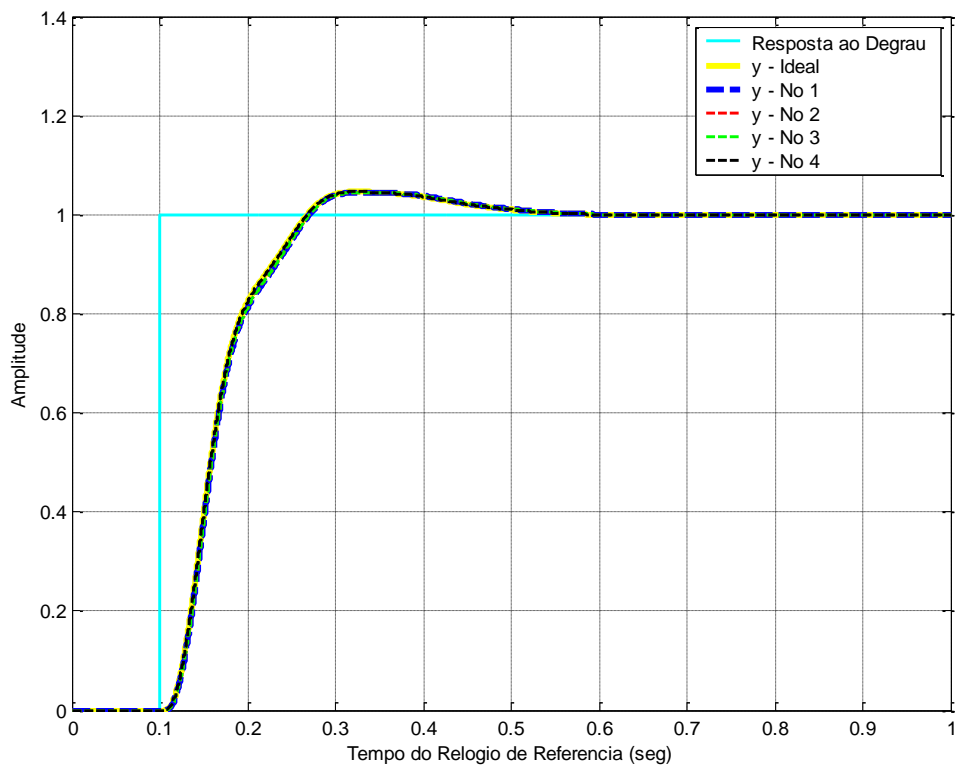


Figura D.9 – Resposta ao Degrau com deriva de 1%.

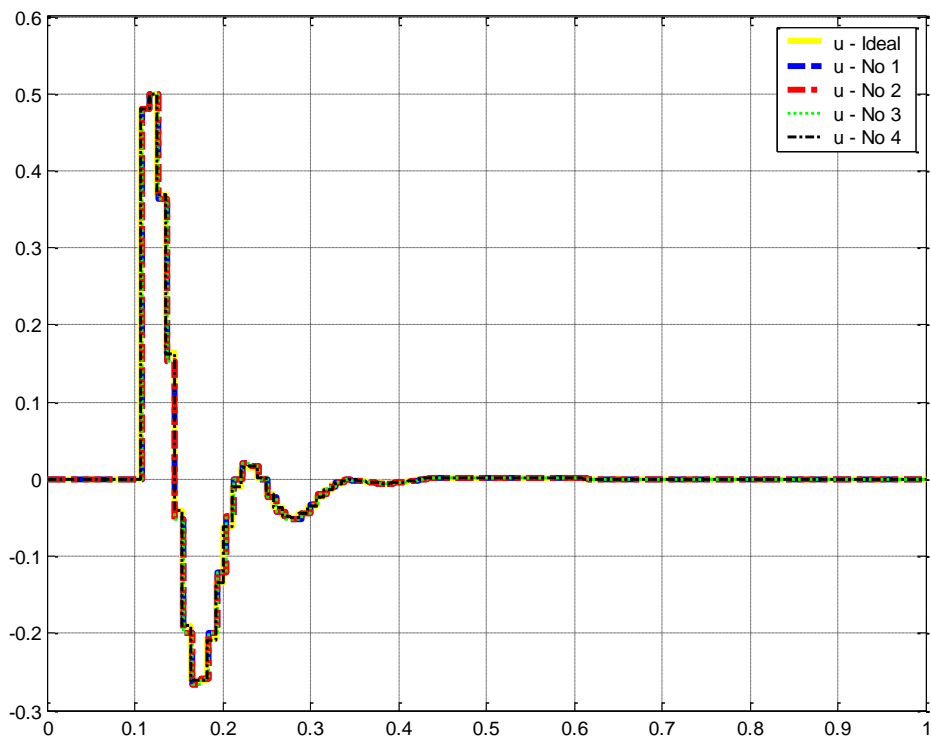


Figura D.10 – Lei de Controle com deriva de 1%.

A Figura D.9 mostra a resposta ao degrau do sistema de controle com a deriva de 1%. É possível ver que somente o nó 4 sofre variação muito pequena em relação ao ideal e aos outros nós do sistema. O relógio do nó 4 está adiantado em relação aos outros, mas o adiantamento é muito pequeno. A Figura D.10 mostra as leis de controle aplicadas a cada nó do sistema e do sistema ideal. Conforme visto, a variação do nó 4 é muito pequena.

Mesmo com o pequeno impacto da taxa de deriva de 1%, vamos aplicar o algoritmo FTM de sincronização de relógios. A Figura D.12 mostra a linha do tempo com o algoritmo de sincronização FTM aplicado ao sistema. O período de re-sincronização é de 0.0040 segundos, isto é, a cada 0.0040 segundos o algoritmo FTM atualiza os valores de relógio com a correção calculada. Na Figura D.10 é possível ver o gráfico de correção dos relógios dos respectivos nós do sistema.

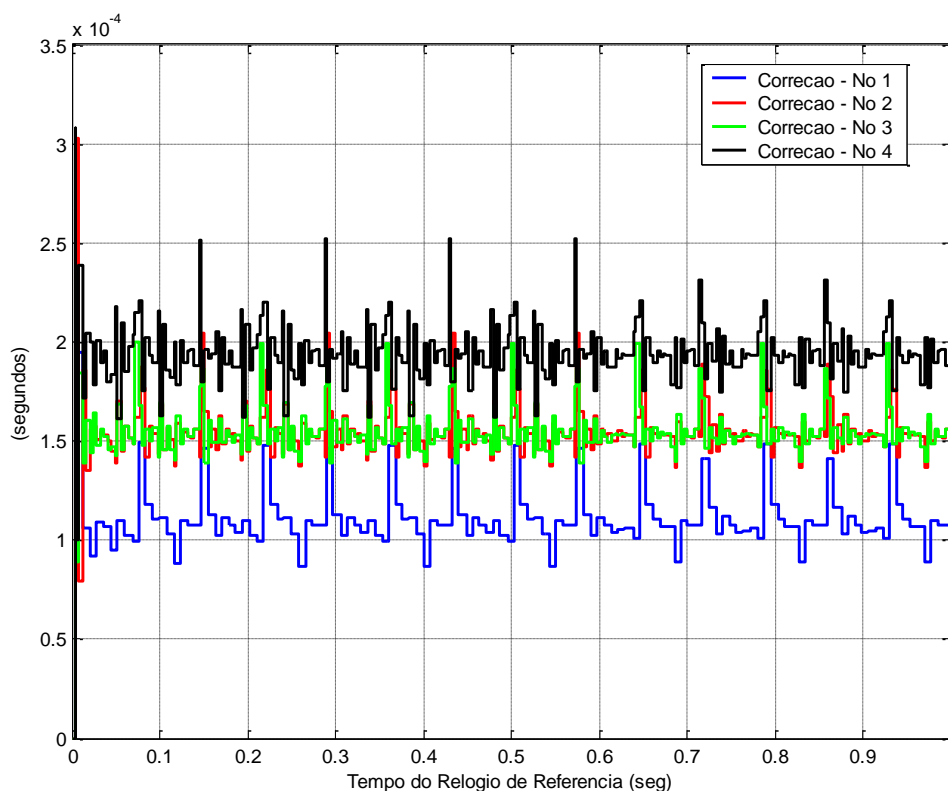
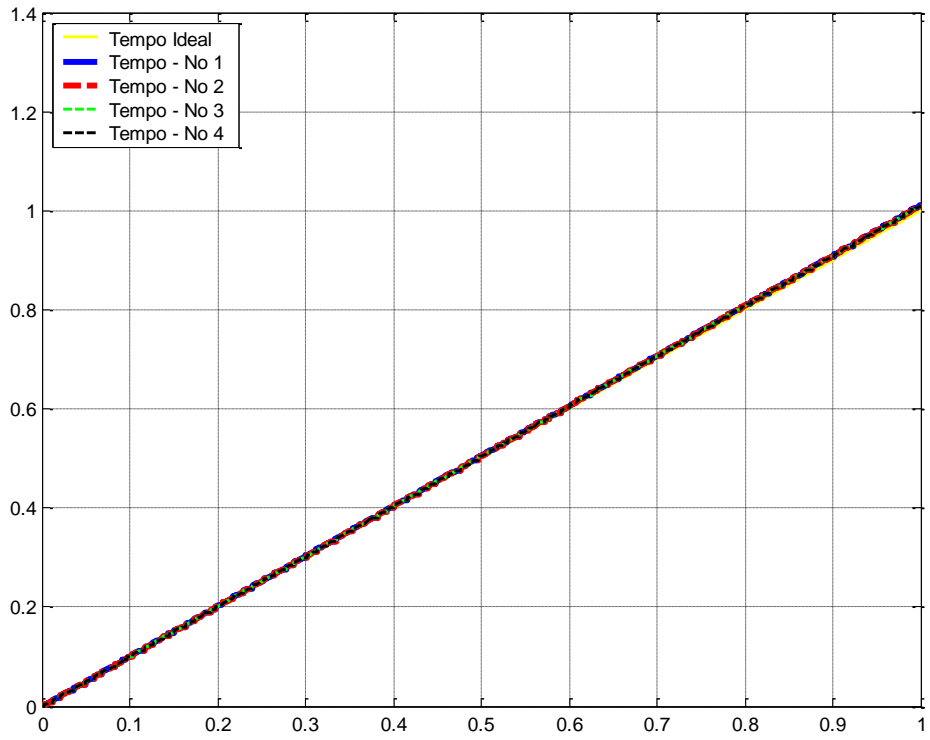
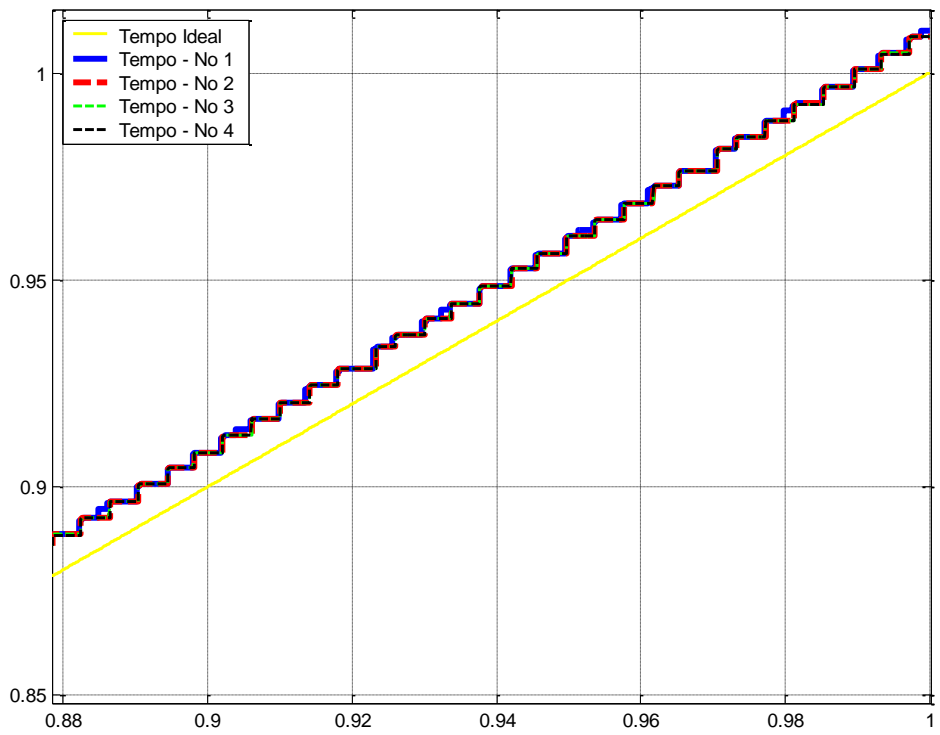


Figura D.11 – Correção Algoritmo FTM com deriva de 1%





(a)



(b)

Figura D.12 – (a) Linha do Tempo com deriva de 1%. Sincronizado (b) Aproximação.

A Figura D.11 mostra que o algoritmo FTM convergiu. Os nós 1, 2 e 3 não estão com erro de deriva. O nó 1 fez uma correção periódica em torno de 0.1 (ms), os nós 2 e 3 fizeram uma correção periódica em torno de 0.15 (ms) e o nó 4, com 1% de taxa de deriva, teve uma correção periódica em torno de 0.2 (ms). Com isso, é possível constatar através do gráfico D.11 que a precisão dos relógios do sistema ficou em torno de 0.135 (ms). A Figura D.12 mostra que os relógios dos nós se sincronizaram entre si, apesar do tempo global sofrer uma pequena deriva em relação ao tempo ideal. A Figura D.13 mostra a resposta ao degrau do sistema de controle. E a Figura D.14 mostra as leis de controle do sistema.

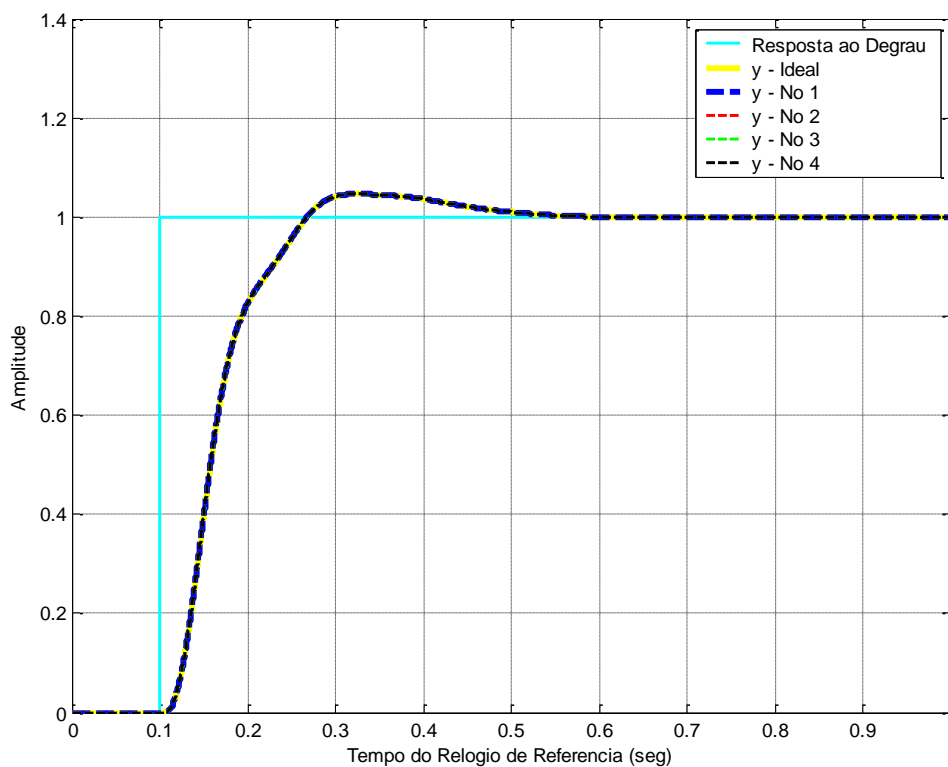


Figura D.13 – Resposta ao Degrau com deriva de 1% Sincronizado.

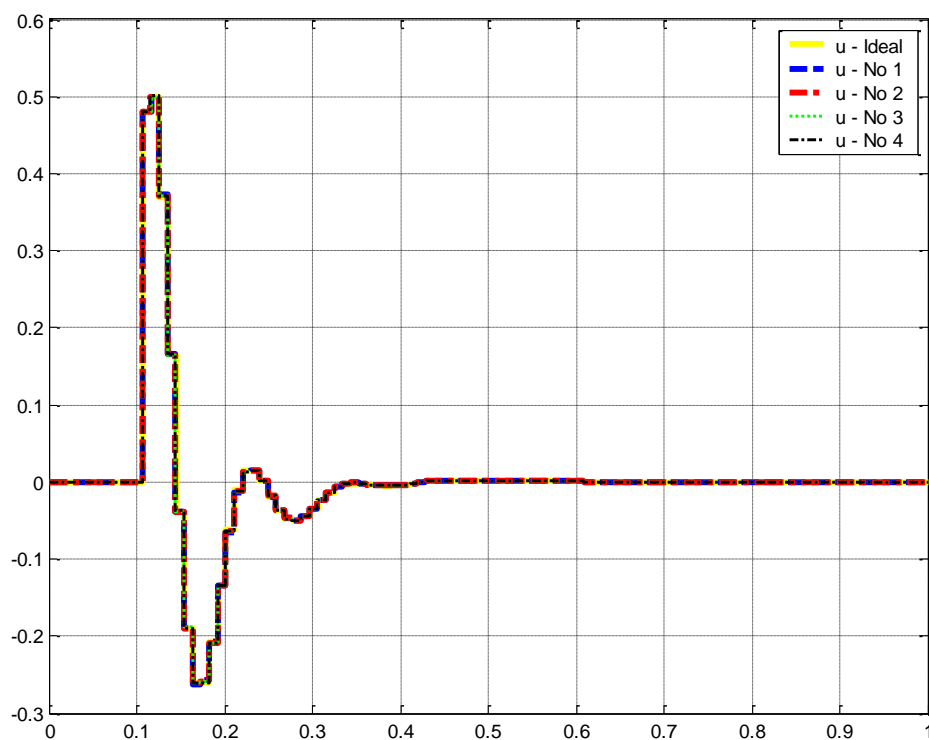
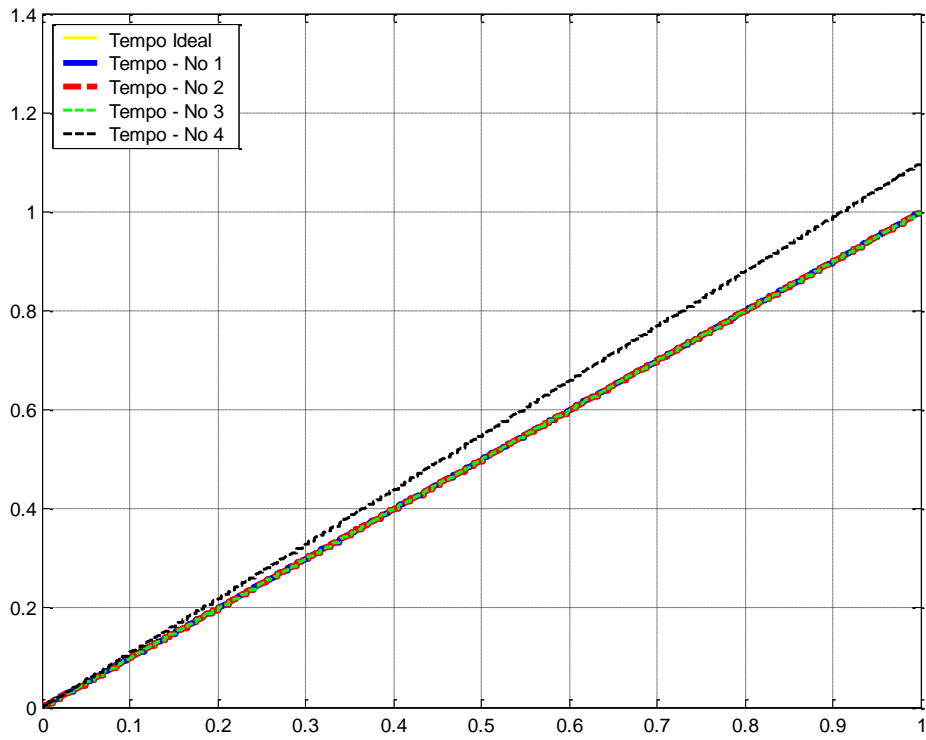


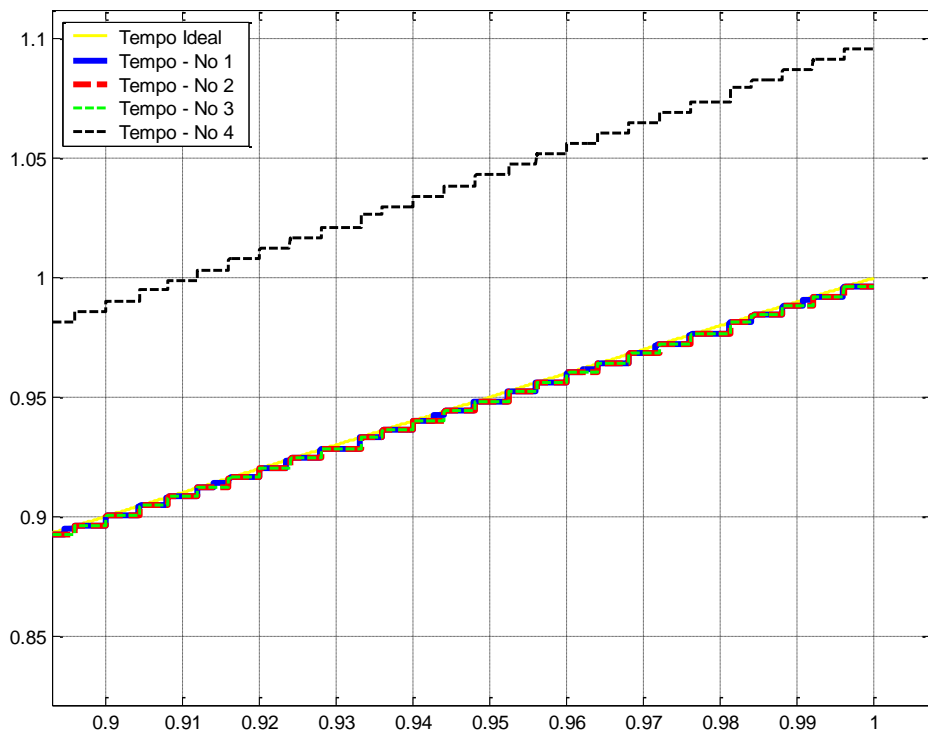
Figura D.14 – Lei de Controle com deriva de 1% Sincronizado.

Através das Figuras D.13 e D.14 é possível ver que a sincronização de relógios minimizou a diferença da resposta ao degrau e das leis de controle dos nós em relação ao sistema ideal. Apesar do sistema ser um pouco adiantado em relação ao sistema ideal, as respostas ao degrau estão sincronizadas entre si, graças ao sincronismo de relógios. Neste caso o algoritmo mostrou-se muito eficiente corrigindo o erro de deriva presente no sistema e aproximando o nós do sistema de controle ao sistema de controle ideal.

A uma **taxa de deriva de 10%**, o relógio do nó 4 esta se adiantando em relação ao relógio ideal e aos outros relógios dos outros nós, como mostra a Figura D.15. Com isso, o sistema sofre uma pequena variação, por causa do relógio, no sistema de controle do nó 4 que esta afetado pelo relógio. Com esta variação, se espera que o sistema de controle seja afetado.



(a)



(b)

Figura D.15 – (a) Linha do Tempo com deriva de 10%. (b) Aproximação

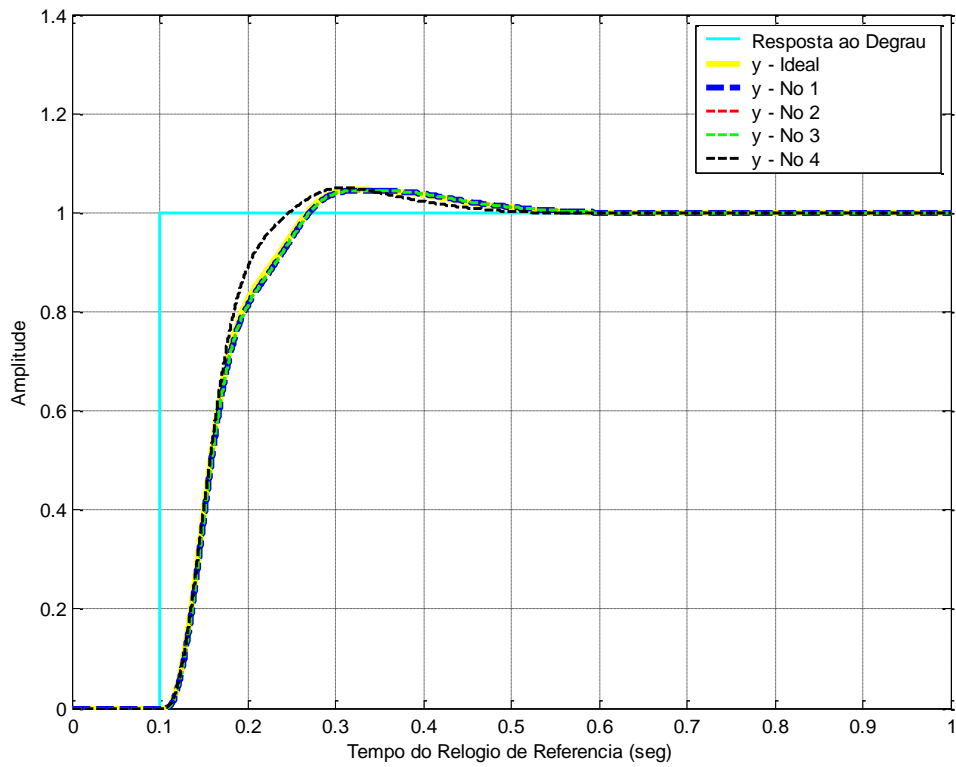


Figura D.16 – Resposta ao Degrau com deriva de 10%.

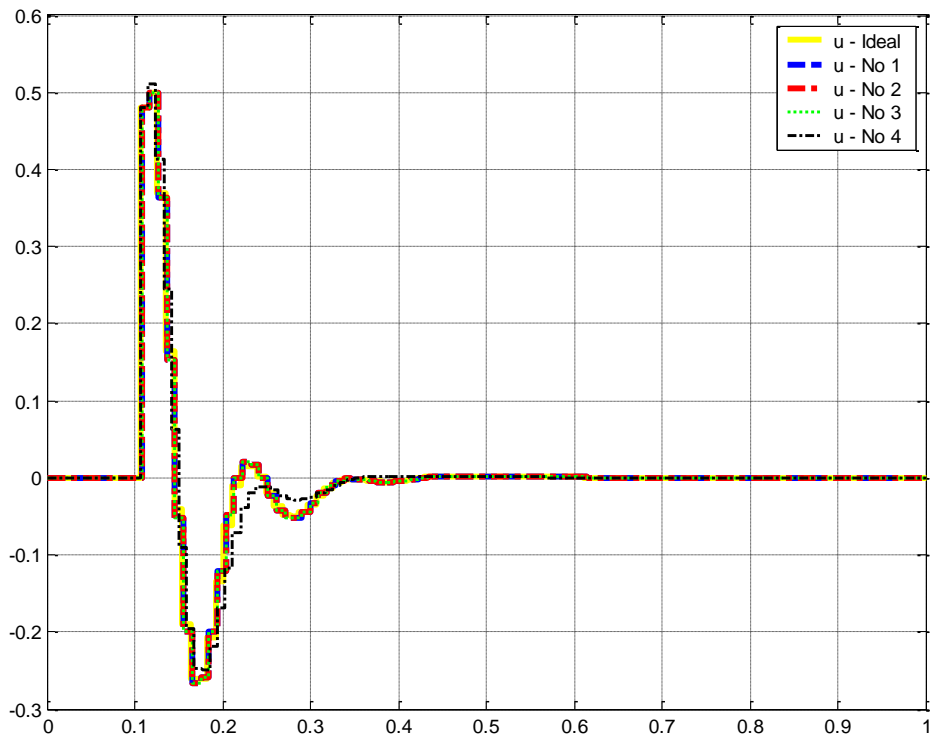


Figura D.17 – Lei de Controle com deriva de 10%.

A Figura D.16 mostra a resposta ao degrau do sistema de controle com a deriva de 10%. É possível ver que somente o nó 4 sofre variação em relação ao ideal e aos outros nós do sistema. O relógio do nó 4 está adiantado em relação aos outros. A Figura D.17 mostra as leis de controle aplicadas a cada nó do sistema e do sistema ideal. Com o impacto da taxa de deriva de 10%, vamos aplicar o algoritmo FTM de sincronização de relógios. A Figura D.19 mostra a linha do tempo com o algoritmo de sincronização FTM aplicado ao sistema. O período de re-sincronização é de 0.0040 segundos, isto é, a cada 0.0040 segundos o algoritmo FTM atualiza os valores de relógio com a correção calculada. Na Figura D.18 é possível ver o gráfico de correção dos relógios dos respectivos nós do sistema.

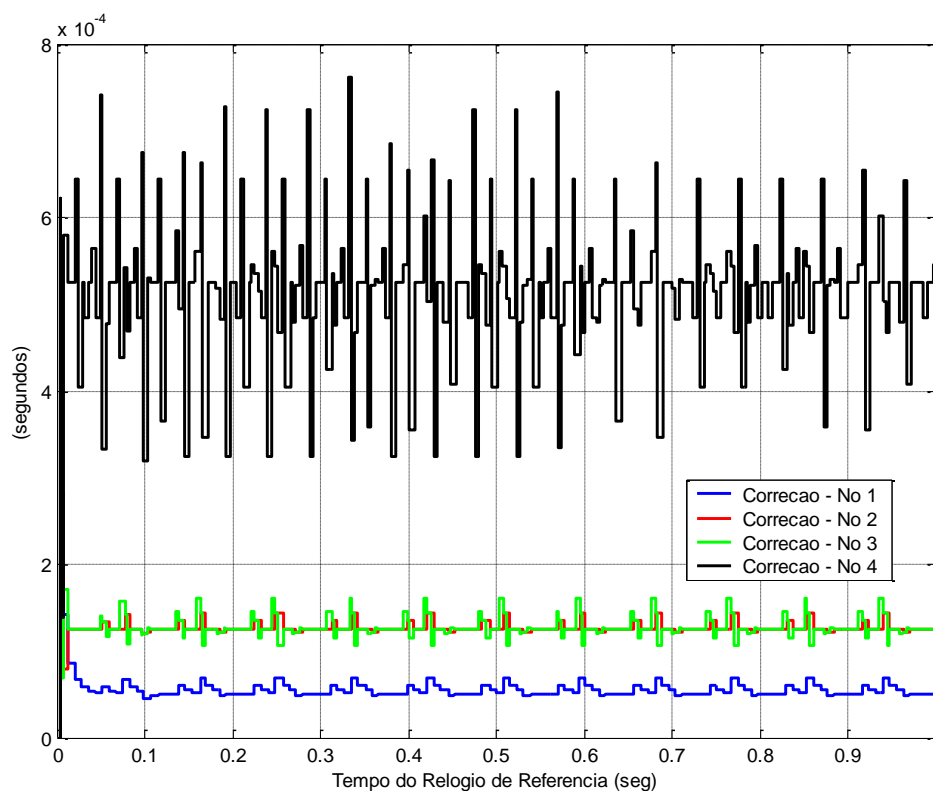
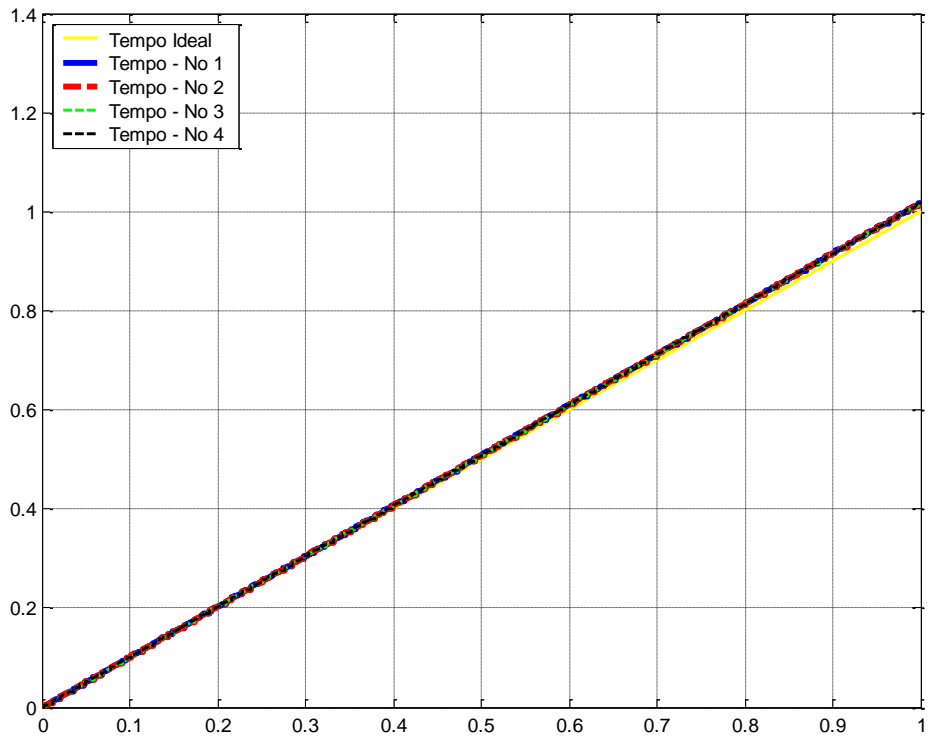
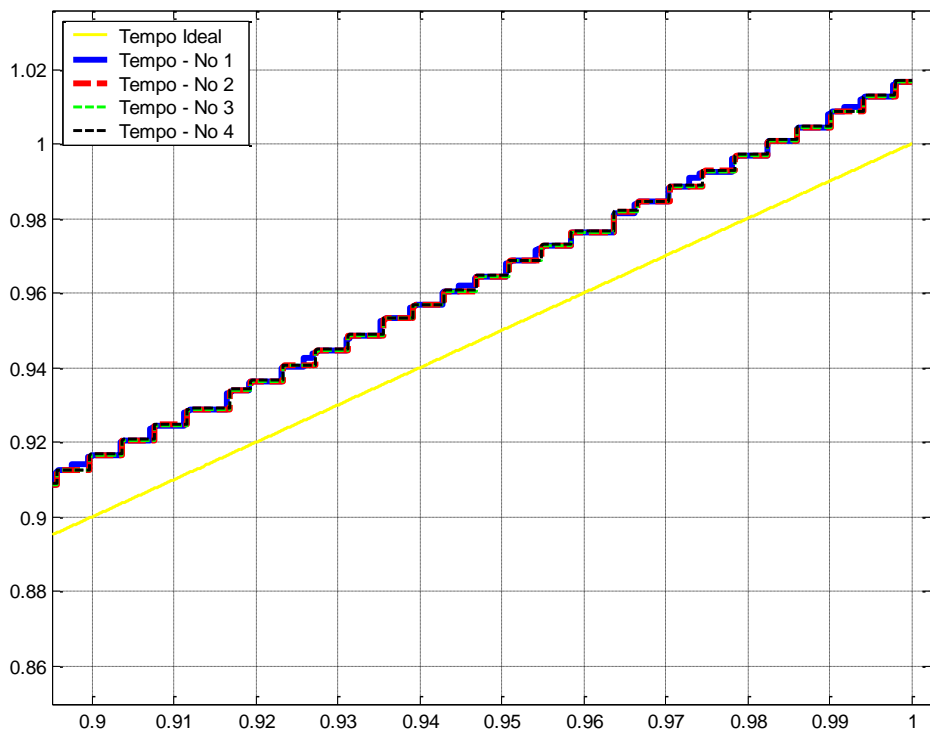


Figura D.18 – Correção Algoritmo FTM com deriva de 10%



(a)



(b)

Figura D.19 – (a) Linha do Tempo com deriva de 10%. Sincronizado (b) Aproximação.

A Figura D.18 mostra que o algoritmo FTM convergiu. Os nós 1, 2 e 3 não estão com erro de deriva. O nó 1 fez uma correção periódica em torno de 0.1 (ms), os nós 2 e 3 fizeram uma correção periódica em torno de 0.15 (ms) e o nó 4, com 10% de taxa de deriva, teve uma correção periódica em torno de 0.5 (ms). Com isso, é possível analisar através do gráfico D.18 que a precisão dos relógios do sistema ficou em torno de 0.7 (ms). A Figura D.19 mostra que os relógios dos nós se sincronizaram entre si, apesar do tempo global sofrer uma pequena deriva em relação ao tempo ideal.

A Figura D.20 mostra a resposta ao degrau do sistema de controle. E a Figura D.21 mostra as leis de controle do sistema.

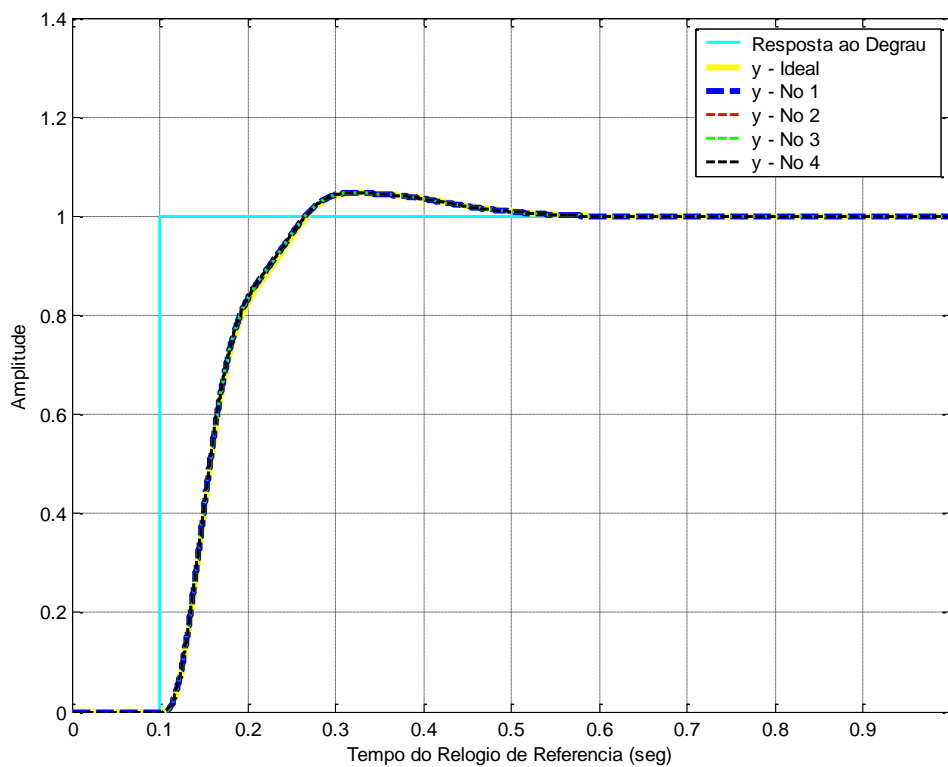


Figura D.20 – Resposta ao Degrau com deriva de 10% Sincronizado.



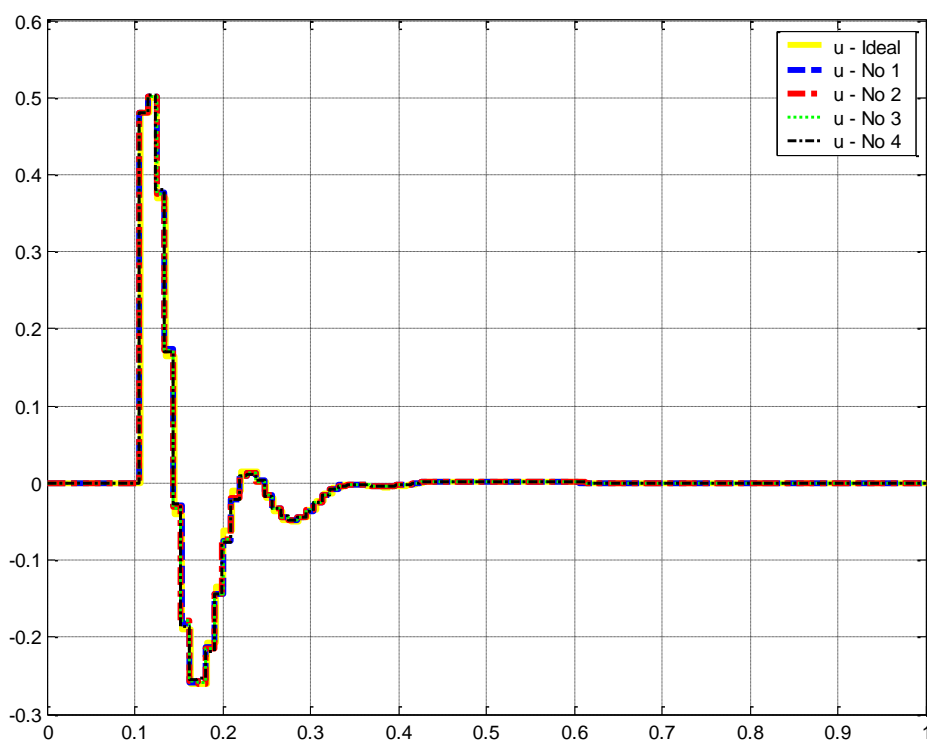


Figura D.21 – Lei de Controle com deriva de 10% Sincronizado.

Através das Figuras D.20 e D.21 é possível ver que a sincronização de relógios minimizou a diferença da resposta ao degrau e das leis de controle dos nós em relação ao sistema ideal. Apesar do sistema ser um pouco adiantado em relação ao sistema ideal, as respostas ao degrau estão sincronizadas entre si, graças ao sincronismo de relógios. Neste caso o algoritmo mostrou-se muito eficiente corrigindo o erro de deriva presente no sistema e aproximando o nós do sistema de controle ao sistema de controle ideal.



## APÊNDICE E – SIMULAÇÃO DO 2º CASO – ARQUITETURA FEDERADA – ALGORITMO FTM

Neste caso, foi feita uma simulação em um sistema de controle com arquitetura federada, descrito na seção anterior. A sincronização foi feita através do algoritmo FTM (*Welch-Lynch*), com arquitetura distribuída, isto é, não existe um relógio de referência. Nesta simulação variou-se o parâmetro de viés inicial (*offset*) do relógio do nó (subsistema) 4. A rede de comunicação foi feita por um protocolo de comunicação com acesso ao meio TDMA.

Este trabalho escolheu 2 valores de viés inicial diferentes para variação do relógio:

- **Caso A: -0.2 (seg)** - O relógio inicia atrasado de 0.2 segundos em relação aos outros;
- **Caso B: 0.39 (seg)** - O relógio inicia adiantado de 0.39 segundos em relação aos outros;

Foram escolhidos estes dois valores de viés inicial, pois em valores menores, os sistemas de controle sofrem muito pouco, ou quase nenhuma variação.

No **caso A**, Com um **viés inicial de -0.2 segundos**, o relógio do nó 4 esta atrasado em relação ao relógio ideal e aos outros relógios dos outros nós, como mostra a Figura E.1. Com isso, o sistema de controle do nó 4 sofre uma pequena variação que esta afetado pelo relógio.

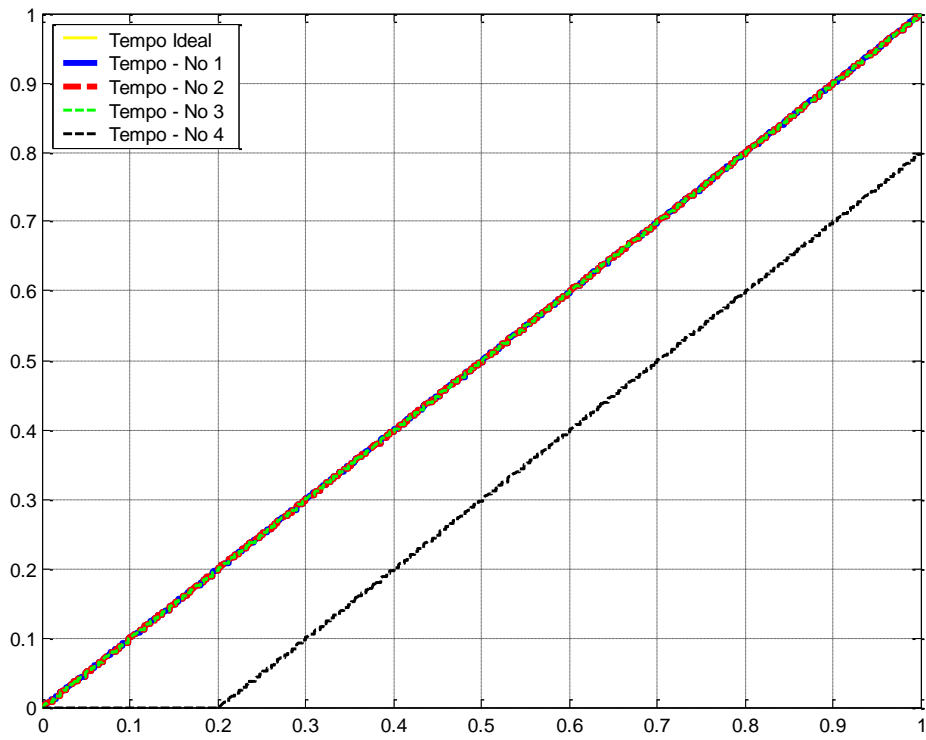


Figura E.1 – Linha do Tempo com viés inicial de -0.2 segundos.

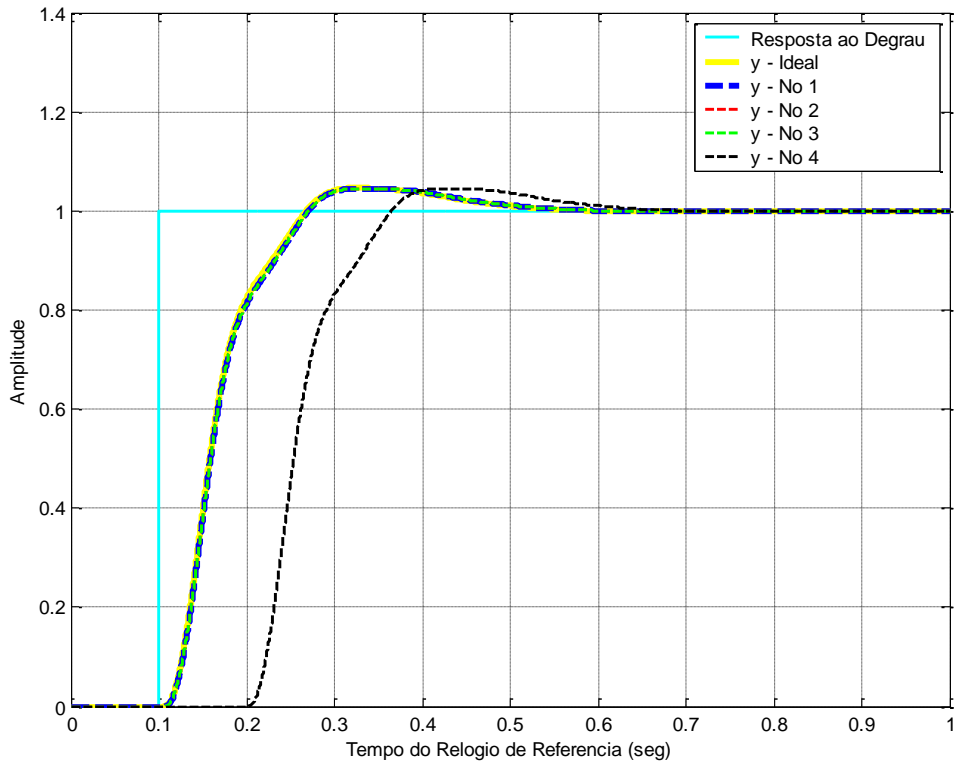


Figura E.2 – Resposta ao Degrau com viés inicial de -0.2 segundos.

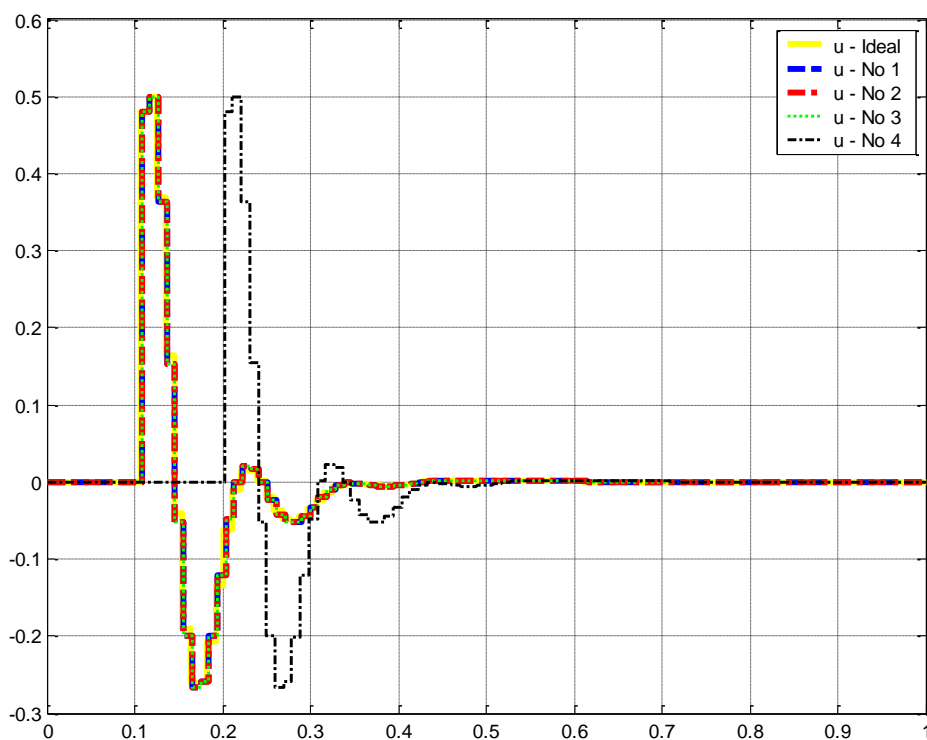


Figura E.3 – Lei de Controle com viés inicial de -0.2 segundos.

A Figura E.2 mostra a resposta ao degrau do sistema de controle com o viés inicial de -0.2 segundos. É possível ver que somente o nó 4 sofre variação em relação ao ideal e aos outros nós do sistema. O nó 4 atrasa sua lei de controle em 0.2 segundos. Isto ocorre pois o relógio do nó 4 inicia atrasado em relação aos outros. A Figura E.3 mostra as leis de controle aplicadas a cada nó do sistema e do sistema ideal. Somente o nó 4 difere sua resposta ao degrau e a lei de controle dos outros nós do sistema. O sistema de controle do nó 4 sofre o impacto do atraso, para tentar corrigir este problema vamos aplicar o algoritmo FTM de sincronização de relógios. A Figura E.5 mostra a linha do tempo com o algoritmo de sincronização de relógios FTM aplicado ao sistema. O período de re-sincronização é de 0.0040 segundos, isto é, a cada 0.0040 segundos o algoritmo FTM atualiza os valores de relógio com a correção calculada. Na Figura E.4 é possível ver o gráfico de correção dos relógios dos respectivos nós do sistema.

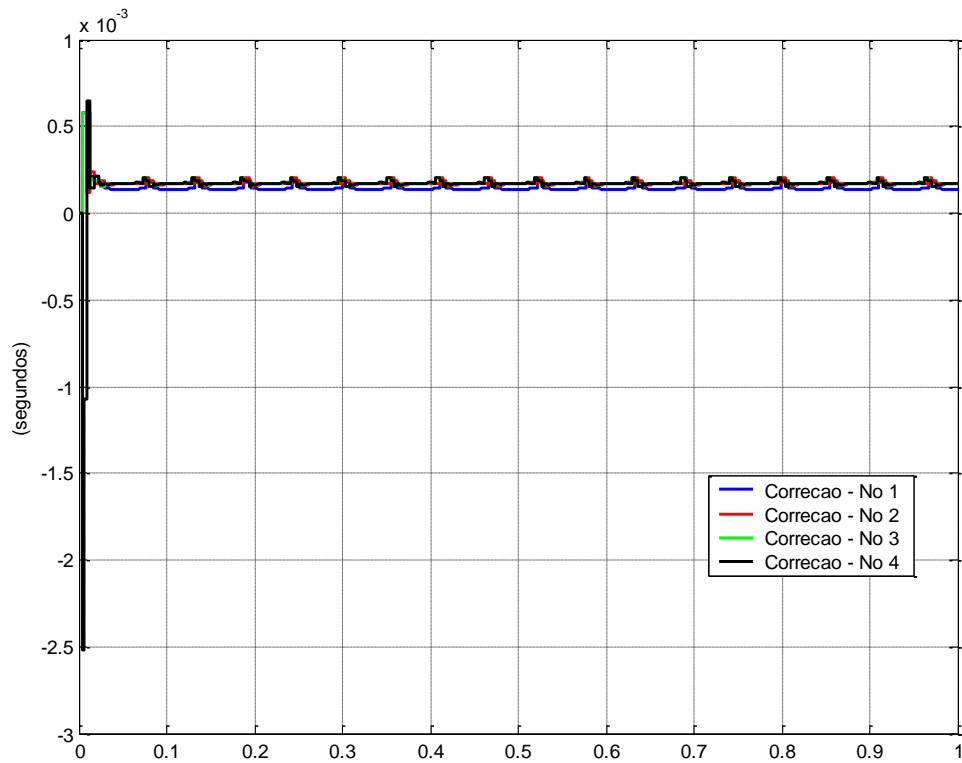


Figura E.4 – Correção Algoritmo FTM com viés inicial de -0.2 segundos.

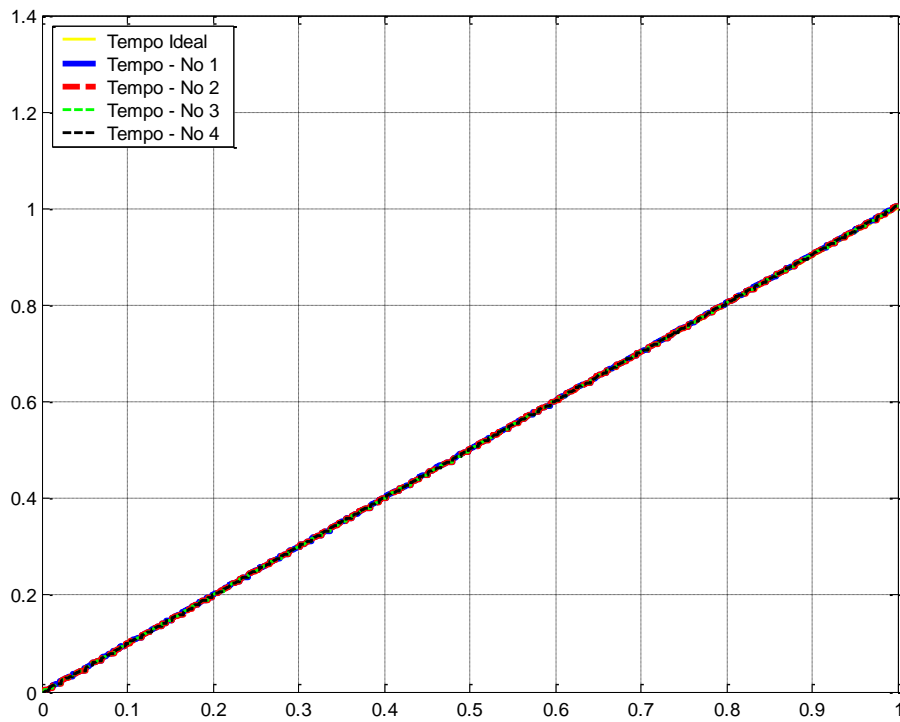


Figura E.5 – Linha do Tempo com viés inicial de -0.2 segundos - Sincronizado.

A Figura E.4 mostra que o algoritmo FTM convergiu. Os nós 1, 2 e 3, que não estavam com erro de viés inicial, fizeram uma correção periódica em torno de 0.1 (ms) enquanto o nó 4, com viés inicial -0.2 segundos teve também uma correção periódica em torno de 0.1 (ms). Com isso, é possível analisar através do gráfico E.4 que a precisão dos relógios do sistema ficou em torno de 0.2 (ms). A Figura E.5 mostra que os relógios dos nós se sincronizaram entre si. Um fato interessante é que o algoritmo de sincronização de relógios FTM tem em sua suposição que todos os relógios da rede devem iniciar sincronizados, isto é, sem viés inicial em nenhum relógio. Apesar deste caso desobedecer a regra, o algoritmo consegue sincronizar os relógios. Este fato ocorre, pois o algoritmo de sincronização é tolerante a falhas, ao iniciar a sincronização em 0.0040 segundos, o relógio do nó 4 esta indicando 0 segundos. Mesmo com este fato o algoritmo corrige o relógio. Como o sistema de controle em 0.0040 ainda é zero, o sistema corrige o relógio antes do controle atuar.

A Figura E.6 mostra a resposta ao degrau do sistema de controle. E a Figura E.7 mostra as leis de controle.

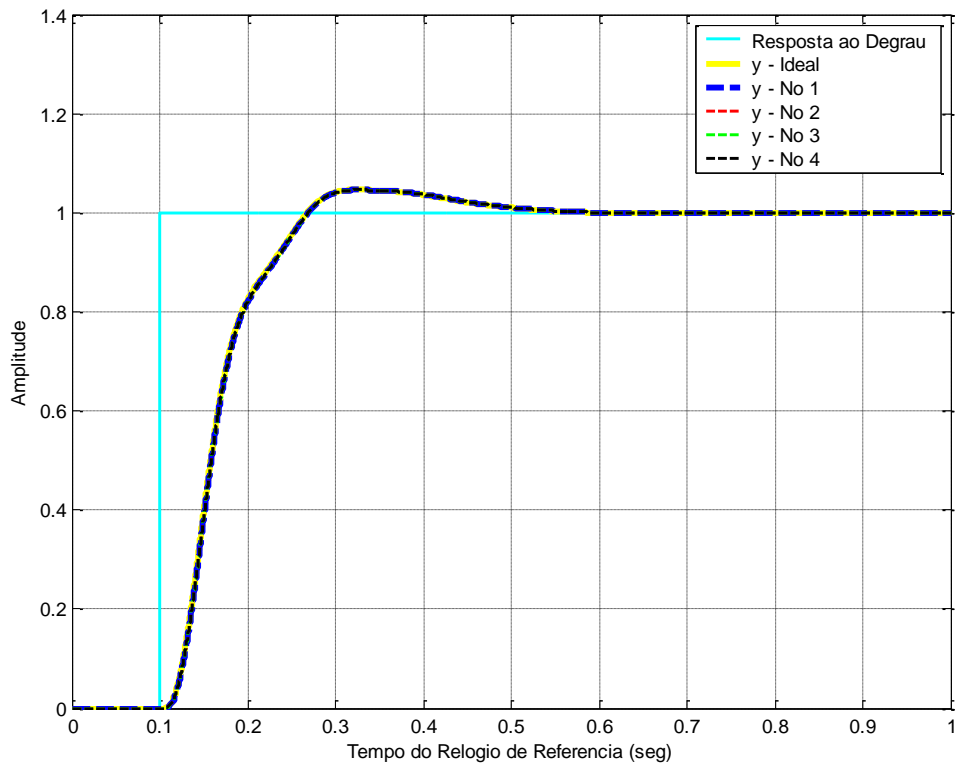


Figura E.6 – Resposta ao Degrau com viés inicial de -0.2 segundos - Sincronizado.

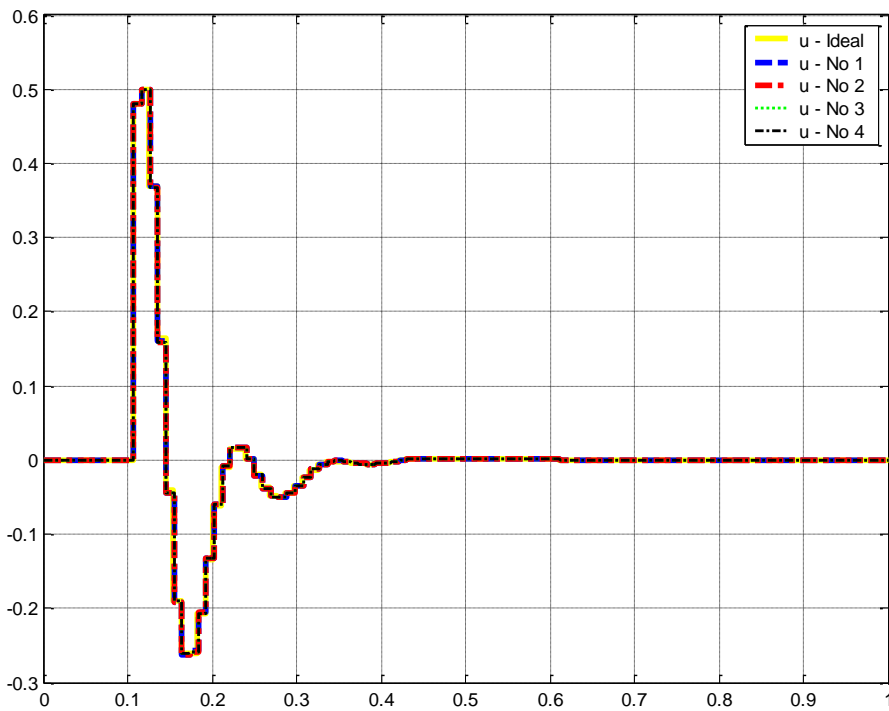


Figura E.7 – Lei de Controle com viés inicial de -0.2 segundos - Sincronizado.



Através das Figuras 4. 50 e 4.51 é possível ver que a sincronização de relógios anulou o erro que o viés inicial do relógio 4 causou à resposta ao degrau e a lei de controle. E os relógios do sistema estão sincronizados entre si. Neste caso o algoritmo mostrou-se muito eficiente corrigindo o erro de viés inicial do relógio 4.

**No caso B**, com um **viés inicial de 0.39 segundos**, o relógio do nó 4 esta adiantado em relação ao relógio ideal e aos outros relógios dos outros nós, como mostra a Figura E.8. Com isso, o sistema de controle do nó 4 que é afetado pelo relógio e sofre uma pequena variação na resposta ao degrau e na lei de controle.

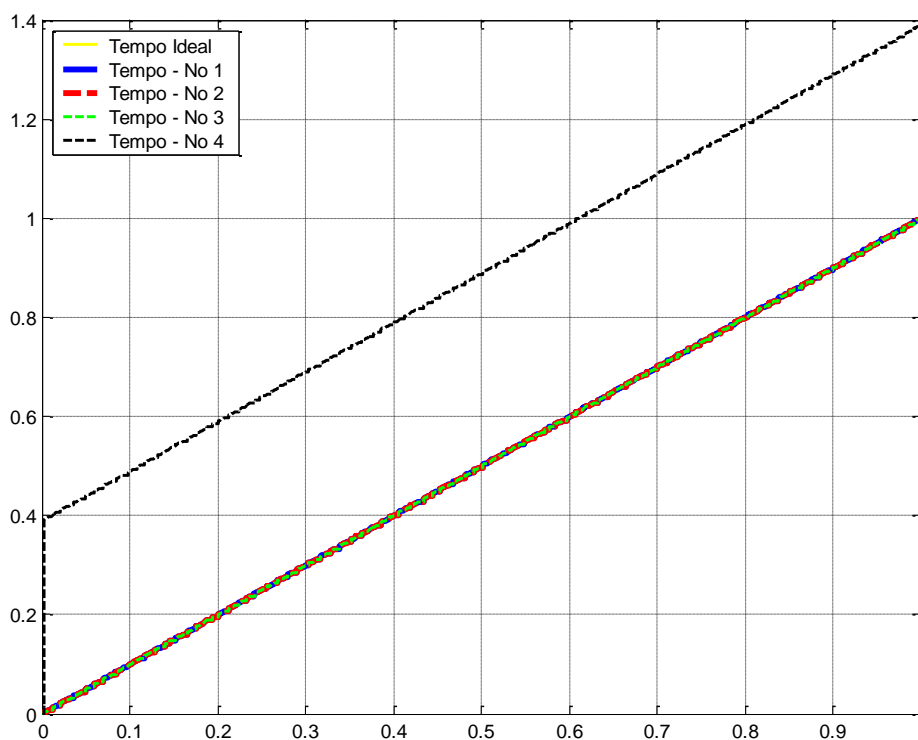


Figura E.8 – Linha do Tempo com viés inicial de 0.39 segundos.

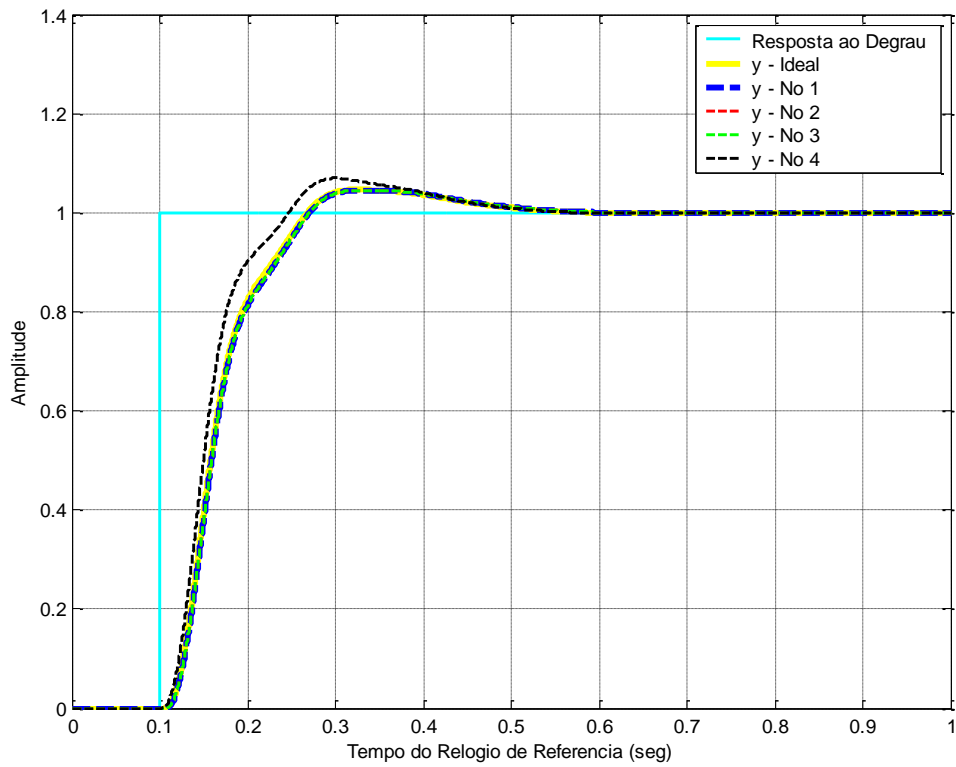


Figura E.9 – Resposta ao Degrau com viés inicial de 0.39 segundos.

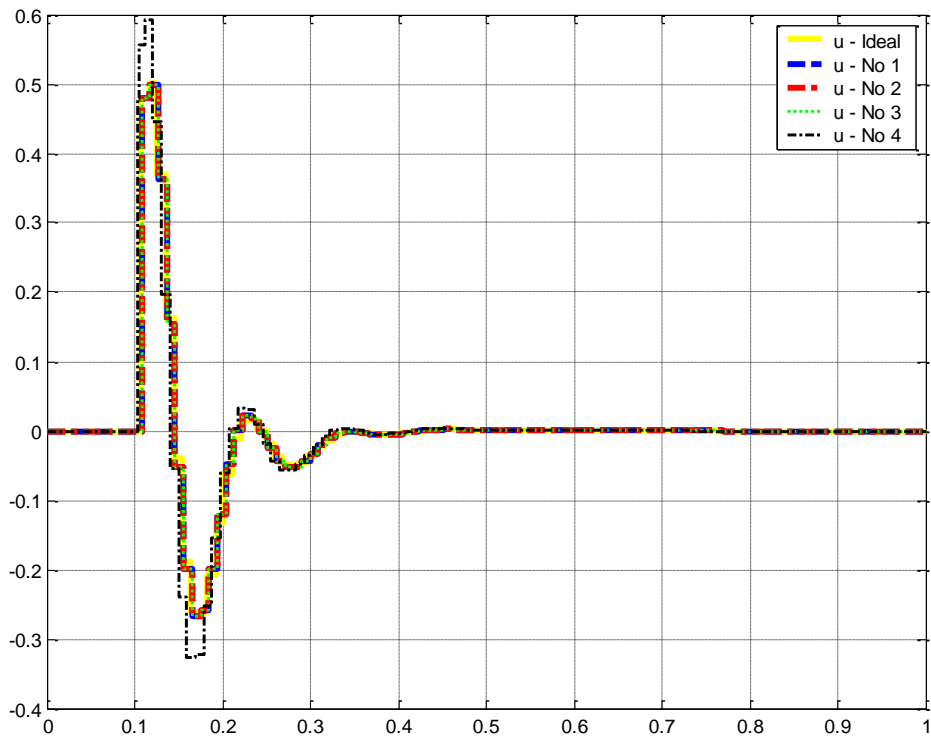


Figura E.10 – Lei de Controle com viés inicial de 0.39 segundos.

A Figura E.9 mostra a resposta ao degrau do sistema de controle com o viés inicial de 0.39 segundos. É possível ver que somente o nó 4 sofre variação em relação ao ideal e aos outros nós do sistema. Isto ocorre pois o relógio do nó 4 inicia adiantado em relação aos outros. A Figura E.10 mostra as leis de controle aplicadas a cada nó do sistema e do sistema ideal. Somente o nó 4 difere sua resposta ao degrau e a lei de controle dos outros nós do sistema. O sistema de controle do nó 4 sofre o impacto do adiantamento, para tentar corrigir este problema vamos aplicar o algoritmo FTM de sincronização de relógios. A Figura 4.56 mostra a linha do tempo com o algoritmo de sincronização de relógios FTM aplicado ao sistema. O período de re-sincronização é de 0.0040 segundos, isto é, a cada 0.0040 segundos o algoritmo FTM atualiza os valores de relógio com a correção calculada. Na Figura E.11 é possível ver o gráfico de correção dos relógios dos respectivos nós do sistema.

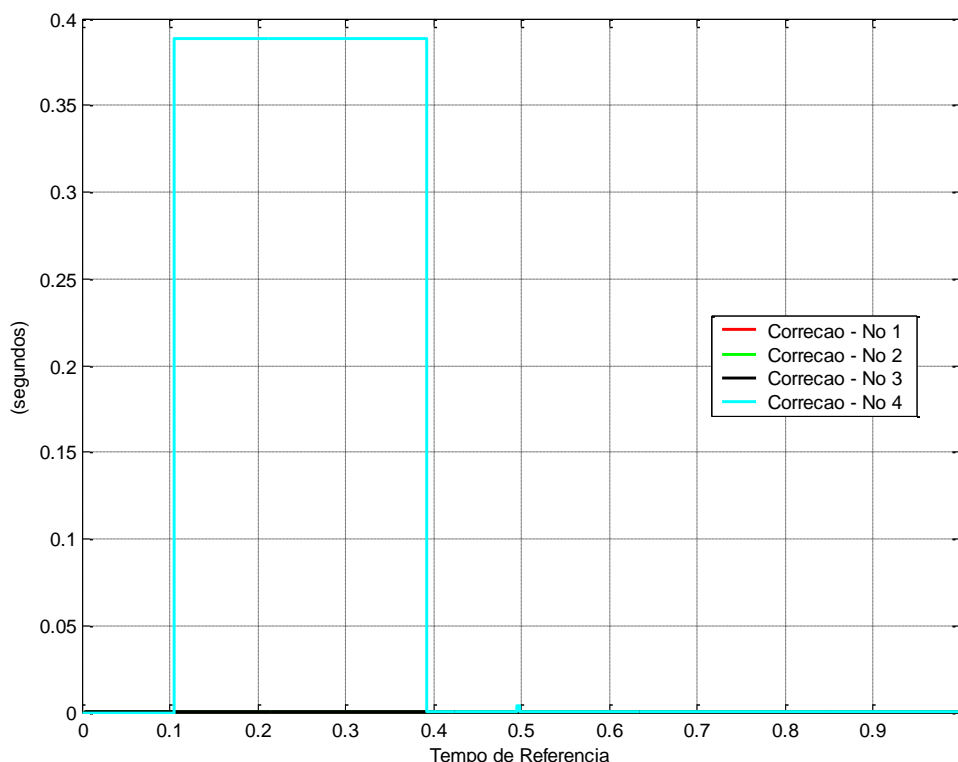


Figura E.11 – Correção Algoritmo FTM com viés inicial de -0.2 segundos.

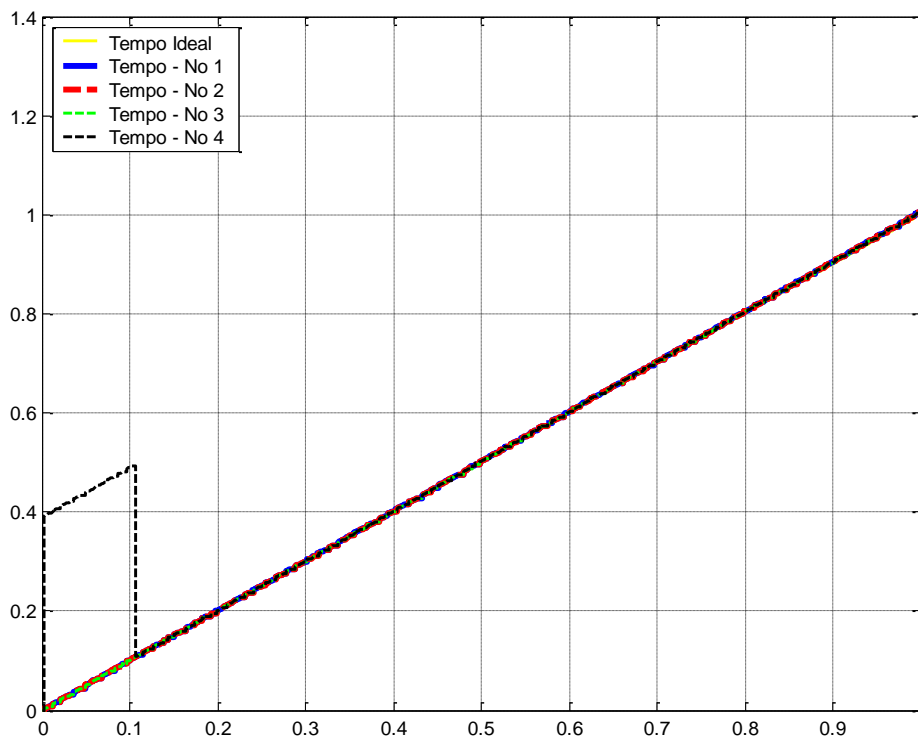


Figura E.12 – Linha do Tempo com viés inicial de 0.39 segundos - Sincronizado.

A Figura E.11 mostra que o algoritmo FTM convergiu. Os nós 1, 2 e 3, que não estavam com erro de viés inicial, fizeram uma correção periódica praticamente nula, enquanto o nó 4, com viés inicial 0.39 segundos teve uma correção periódica grande em torno de 0.39 segundos e em seguida praticamente nula. Com isso, é possível analisar através do gráfico E.11 que a precisão final dos relógios do sistema ficou quase em zero. A Figura E.12 mostra que os relógios dos nós se sincronizaram entre si. Um fato interessante é que o algoritmo de sincronização de relógios FTM tem em sua suposição que todos os relógios da rede devem iniciar sincronizados, isto é, sem viés inicial em nenhum relógio. Apesar deste caso desobedecer a regra, o algoritmo consegue sincronizar os relógios. Mesmo com o algoritmo corrigindo o relógio, o sistema de controle do nó 4 é muito afetado pelo adiantamento. Pois ao corrigir o algoritmo inseriu um atraso no controle fazendo o sistema de controle desestabilizar e depois recuperar o controle.

A Figura E.13 mostra a resposta ao degrau do sistema de controle. E a Figura E.14 mostra as leis de controle.

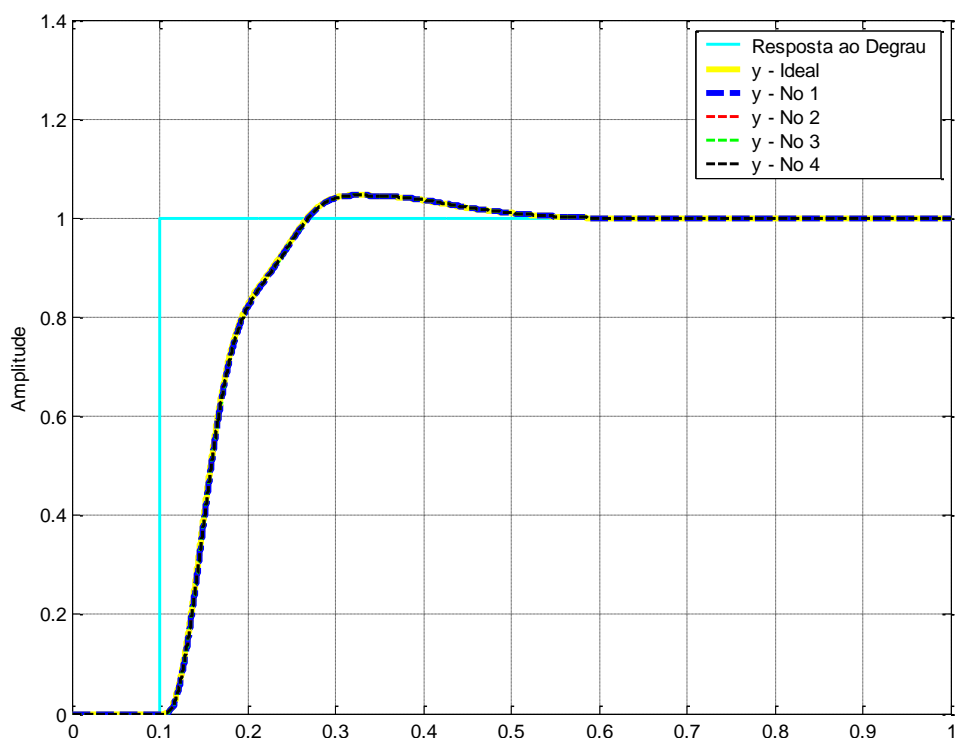


Figura E.13 – Resposta ao Degrau com viés inicial de 0.39 segundos - Sincronizado.

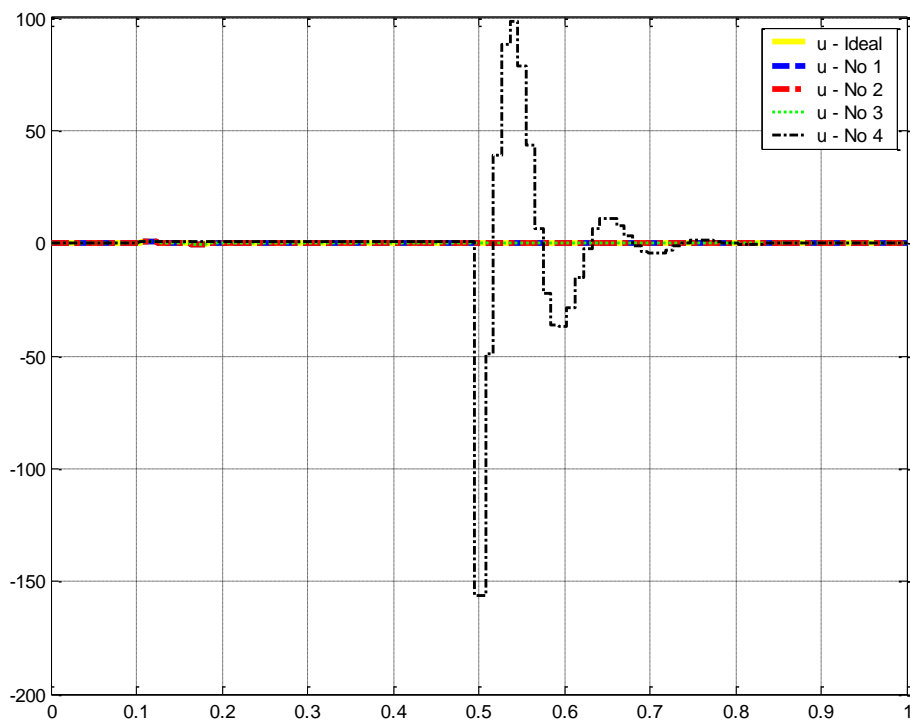


Figura E.14 – Lei de Controle com viés inicial de 0.39 segundos - Sincronizado.

Através das Figuras E.13 e E.14 é possível ver que a sincronização de relógios anulou o erro que do viés inicial do relógio 4. Mas não sincronizou as ações de controle. E com isso causou um erro muito grande à resposta ao degrau e a lei de controle, mesmo com os relógios do sistema sincronizados entre si. Neste caso o algoritmo não mostrou-se muito eficiente, mesmo corrigindo o erro de viés inicial do relógio 4, a lei de controle e a resposta ao degrau foi muito prejudicada.

## APÊNDICE F – SIMULAÇÃO DO 3º CASO – ARQUITETURA FEDERADA – ALGORITMO FTM

Neste caso, foi feita uma simulação em um sistema de controle com arquitetura federada, descrito na seção anterior. A sincronização foi feita através do algoritmo FTM (*Welch-Lynch*), com arquitetura distribuída, isto é, não existe um relógio de referência. Nesta simulação variou-se o parâmetro da deriva e de viés inicial (*offset*) do relógio do nó (subsistema) 4. A rede de comunicação foi feita por um protocolo de comunicação com acesso ao meio TDMA.

Este trabalho escolheu 2 valores deriva e de viés inicial diferentes para variação do relógio do nó 4:

- **Caso A: Deriva: -10%; Viés: -0.2 (seg)** - O relógio inicia atrasado de 0.2 segundos em relação aos outros e esta com uma taxa de deriva negativa, isto é, o relógio está atrasando em relação ao outros;
- **Caso B: Deriva: 10%; Viés: 0.39 (seg)** - O relógio inicia adiantado de 0.39 segundos em relação aos outros e esta com uma taxa de deriva positiva, isto é, o relógio está adiantando em relação ao outros;

Foram escolhidos estes dois valores de viés inicial, pois são os valores dos casos anteriores.

No **caso A**, com um **viés inicial de -0.2 segundos e taxa de deriva de -10%**, o relógio do nó 4 inicia atrasado e além disso o relógio está se atrasando em relação ao relógio ideal e aos outros relógios dos outros nós, como mostra a Figura F.1. Com isso, o sistema de controle do nó 4 que esta afetado pelo relógio sofre uma pequena variação.

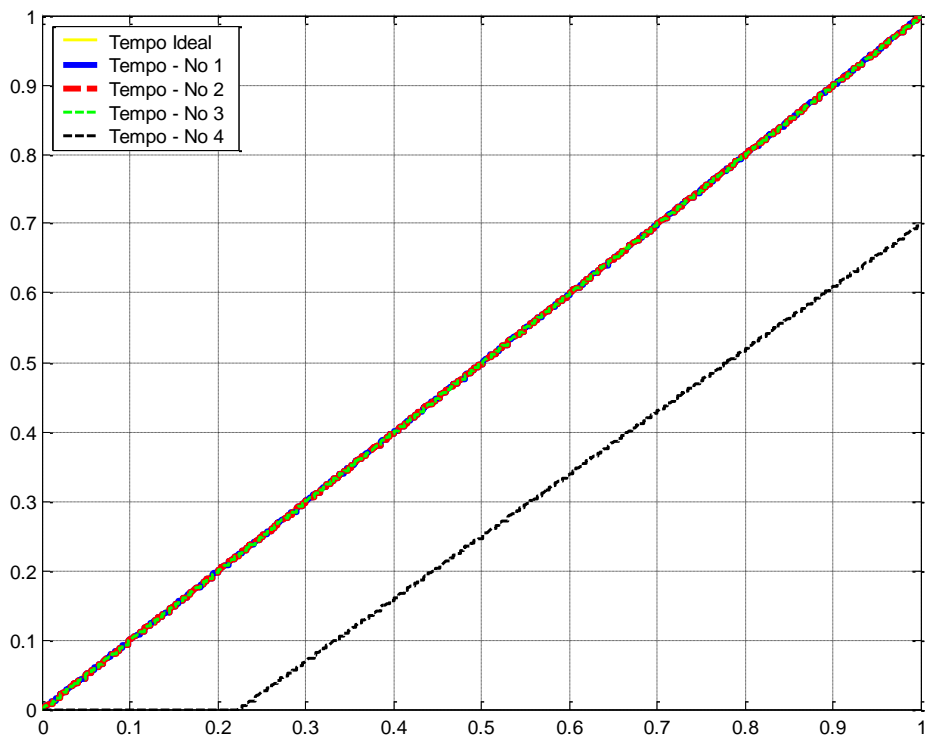


Figura F.1 – Linha do Tempo com viés inicial de -0.2 segundos e deriva -10%.

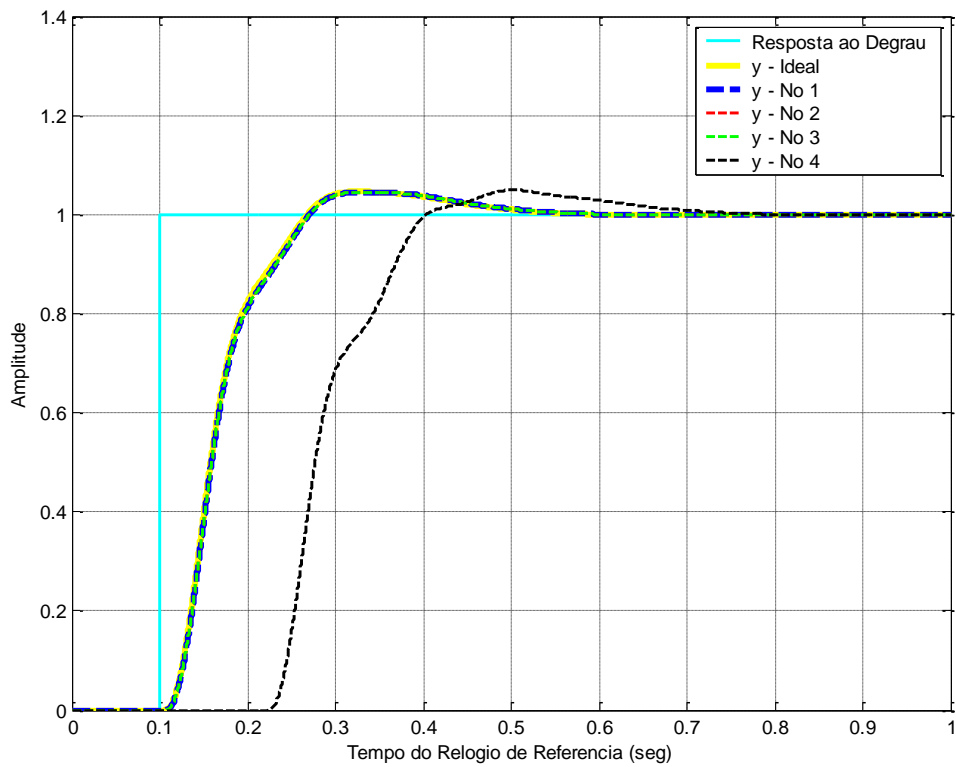


Figura F.2 – Resposta ao Degrau com viés inicial de -0.2 segundos e deriva -10%.



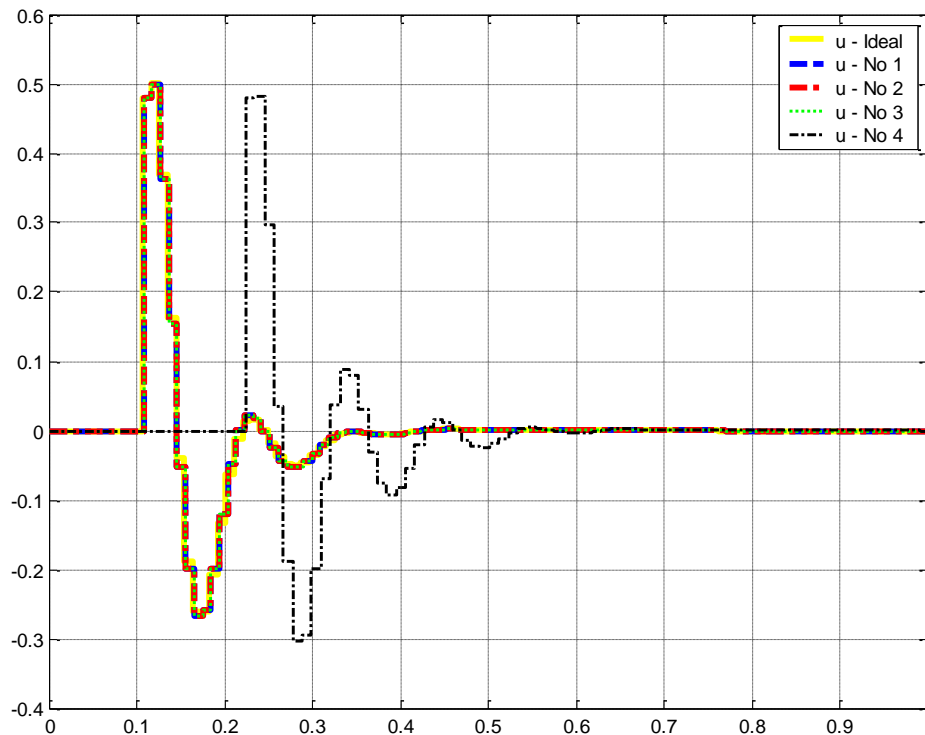


Figura F.3 – Lei de Controle com viés inicial de -0.2 segundos e deriva -10%..

A Figura F.2 mostra a resposta ao degrau do sistema de controle com o viés inicial de -0.2 segundos e deriva de -10%. É possível ver que somente o nó 4 sofre variação em relação ao ideal e aos outros nós do sistema. O nó 4 atrasa sua lei de controle em 0.2 segundos e também ao longo de sua operação o relógio 4 esta se atrasando a uma taxa de deriva de -10%. Isto ocorre pois o relógio do nó 4 inicia atrasado e esta com deriva em relação aos outros relógios do conjunto. A Figura F.3 mostra as leis de controle aplicadas a cada nó do sistema e do sistema ideal. Somente o nó 4 difere sua resposta ao degrau e a lei de controle dos outros nós do sistema. O sistema de controle do nó 4 sofre o impacto do atraso e da deriva atrasando sua lei de controle. Para tentar corrigir este problema vamos aplicar o algoritmo FTM de sincronização de relógios.

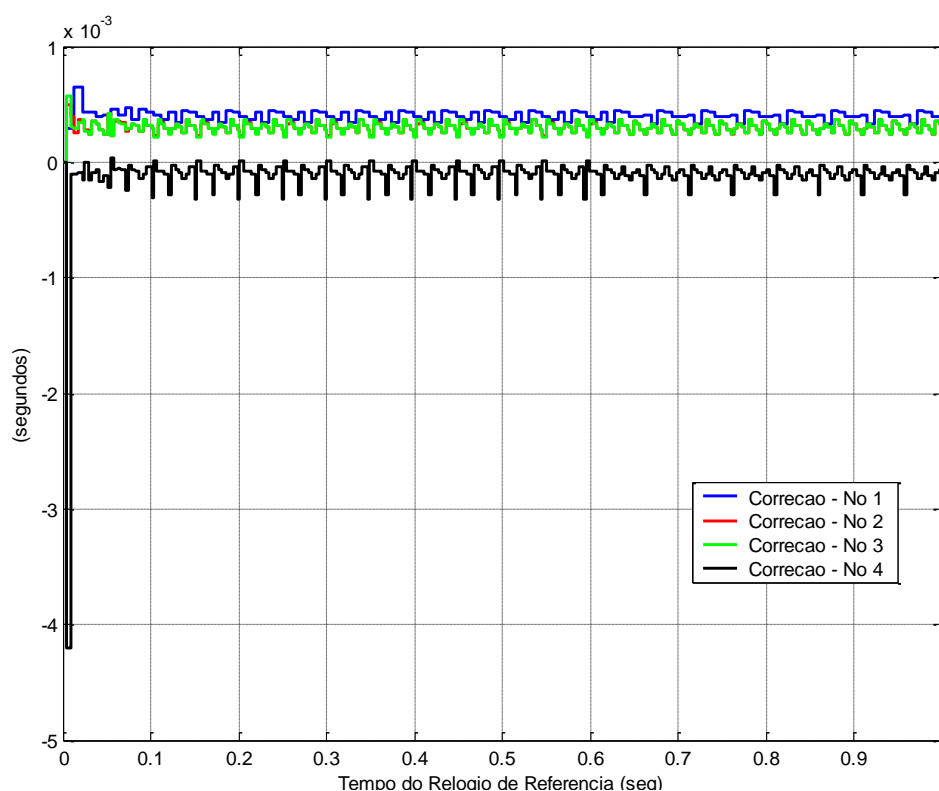
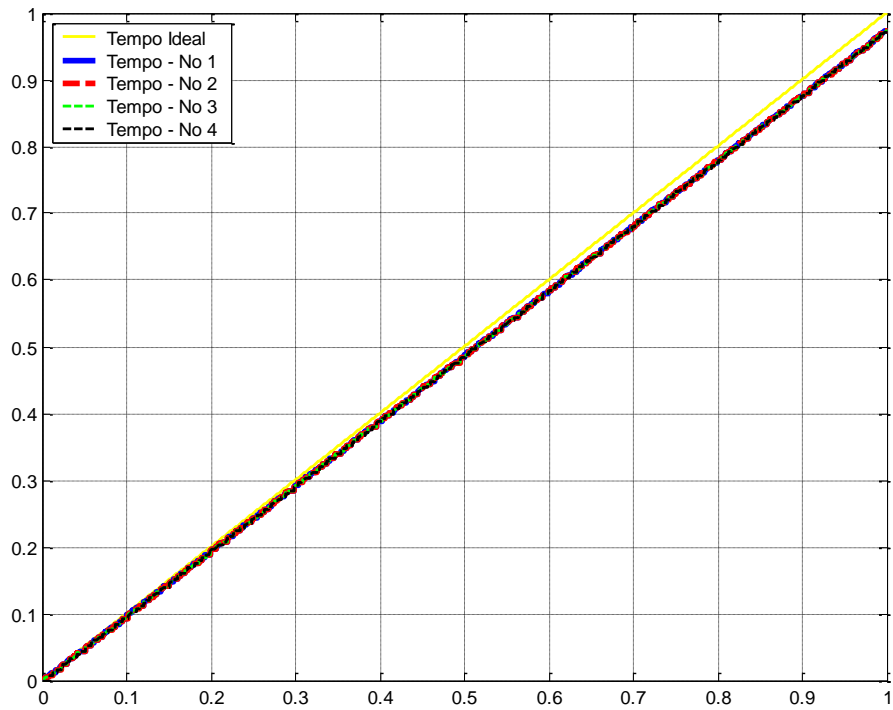


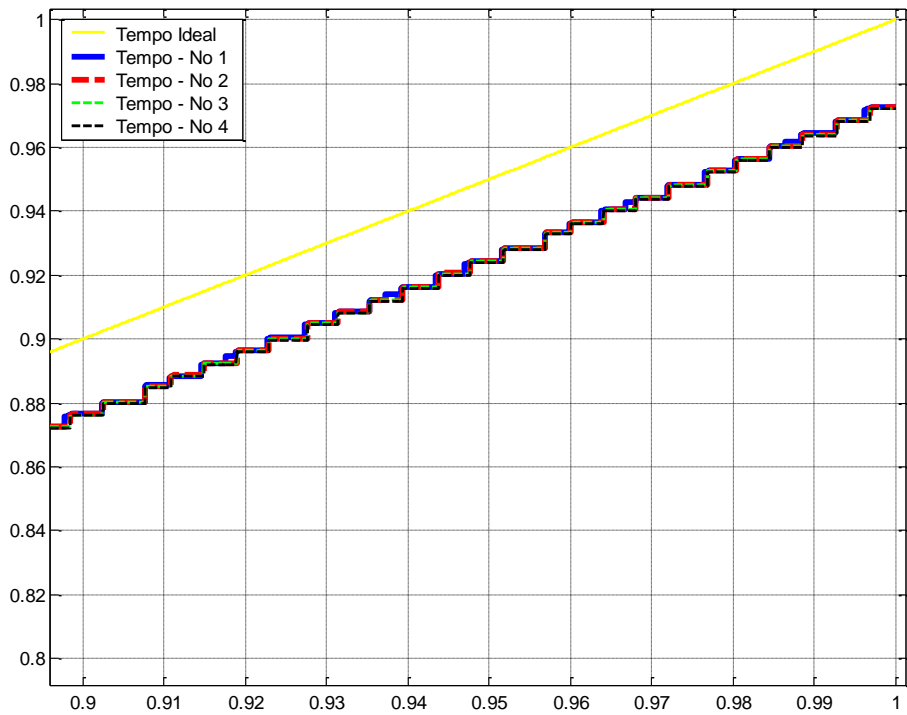
Figura F.4 – Correção Algoritmo FTM: Viés inicial de -0.2 segundos e Deriva -10%.

Na Figura F.4 é possível ver o gráfico de correção dos relógios dos respectivos nós do sistema. A Figura F.4 mostra que o algoritmo FTM convergiu. Os nós 1, 2 e 3, que não estavam com erro de viés inicial, fizeram uma correção periódica em torno de 0.25 (ms). O nó 4, com viés inicial -0.2 segundos e deriva de -10%, teve um pico grande de correção de -0.4 ms e depois convergiu a uma correção periódica em torno de 0.1 (ms). É possível constatar através do gráfico F.4 que a precisão dos relógios do sistema ficou em torno de 1 (ms).

A Figura F.5 mostra a linha do tempo com o algoritmo de sincronização de relógios FTM aplicado ao sistema. O período de re-sincronização é de 0.0040 segundos, isto é, a cada 0.0040 segundos o algoritmo FTM atualiza os valores de relógio com a correção calculada.



(a)



(b)

Figura F.5 – (a) Linha do Tempo Sincronizada: Viés inicial de -0.2 segundos e Deriva - 10%. (b) Aproximação.

A Figura F.5 mostra que os relógios dos nós se sincronizaram entre si. Um fato interessante é que o algoritmo de sincronização de relógios FTM tem em sua suposição que todos os relógios da rede devem iniciar sincronizados, isto é, sem viés inicial em nenhum relógio. Apesar deste caso desobedecer a regra, o algoritmo consegue sincronizar os relógios. Este fato ocorre, pois o algoritmo de sincronização é tolerante a falhas, ao iniciar a sincronização em 0.0040 segundos, o relógio do nó 4 esta indicando 0 segundos. Mesmo com este fato o algoritmo corrige o relógio. Como o sistema de controle em 0.0040 ainda é zero, o sistema corrige o relógio antes do controle atuar.

A Figura F.6 mostra a resposta ao degrau do sistema de controle. E a Figura F.7 mostra as leis de controle.

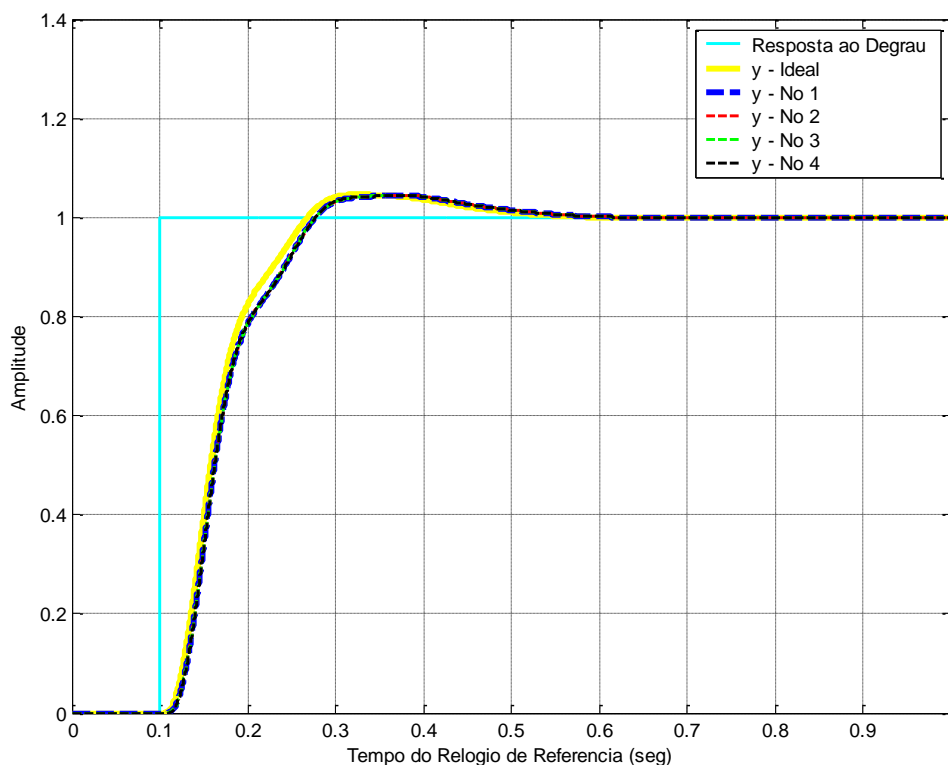


Figura F.6 – Resposta ao Degrau Sincronizado: Viés inicial: -0.2 segundos, Deriva: - 10%.

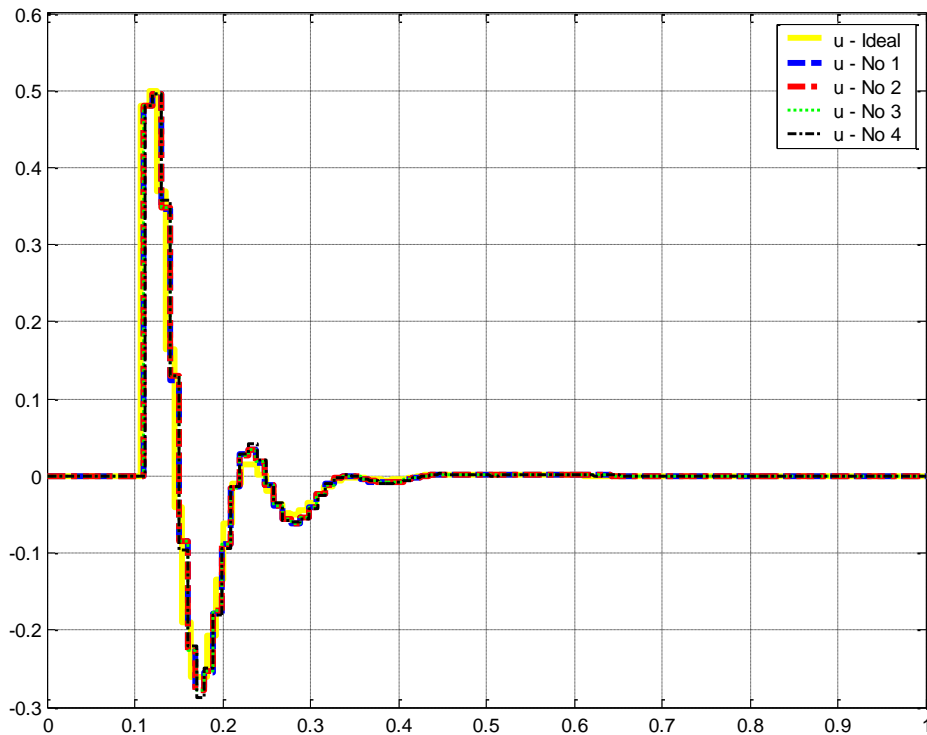


Figura F.7 – Lei de Controle Sincronizada: Viés inicial: -0.2 segundos, Deriva: -10%.

Através das Figuras F.6 e F.7 é possível ver que a sincronização de relógios anulou o erro que o viés inicial e a deriva do relógio 4 causaram à resposta ao degrau e a lei de controle do nó 4. Os relógios do sistema estão sincronizados entre si. Neste caso o algoritmo mostrou-se muito eficiente corrigindo o erro de viés inicial e deriva do relógio 4, eliminando o efeito negativo na lei de controle.

No **caso B**, com um **viés inicial de 0.39 segundos e taxa de deriva de 10%**, o relógio do nó 4 inicia adiantado e além disso o relógio está se adiantando em relação ao relógio ideal e aos outros relógios do conjunto, como mostra a Figura F.8. Com isso, o sistema de controle do nó 4 que está afetado pelo relógio sofre uma variação.

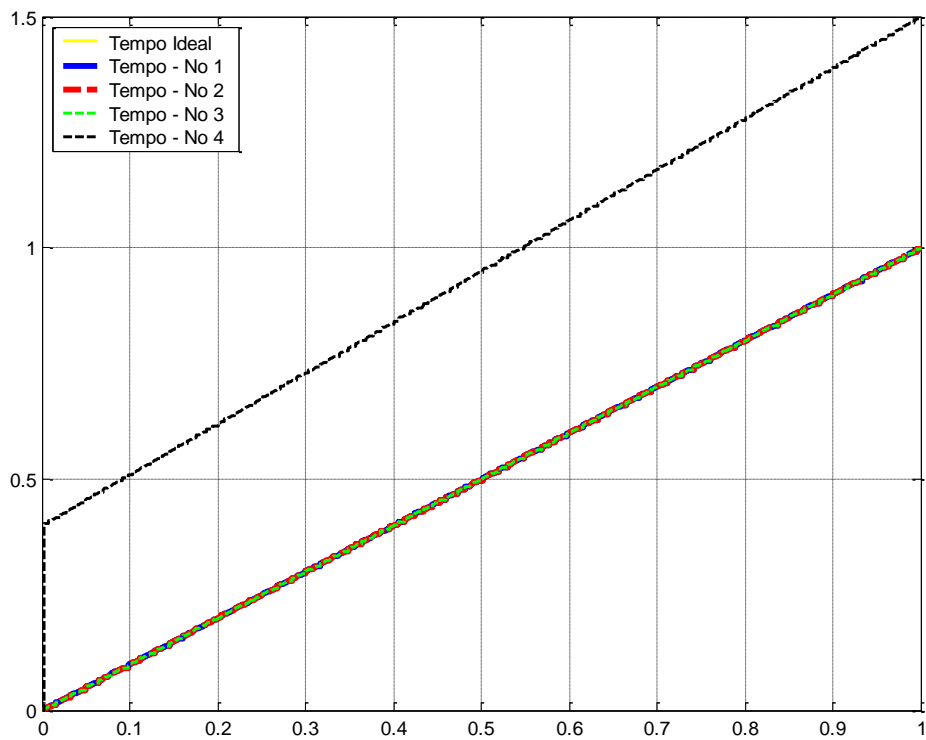


Figura F.8 – Linha do Tempo com viés inicial de 0.39 segundos e deriva 10%.

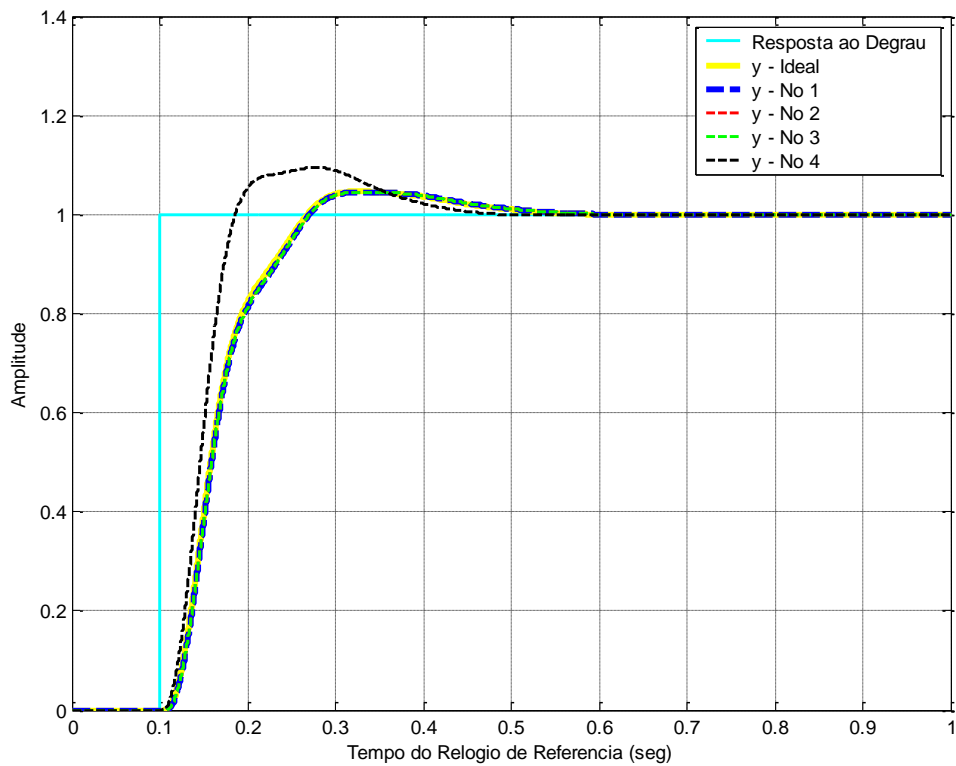


Figura F.9 – Resposta ao Degrau com viés inicial de 0.39 segundos e deriva 10%.

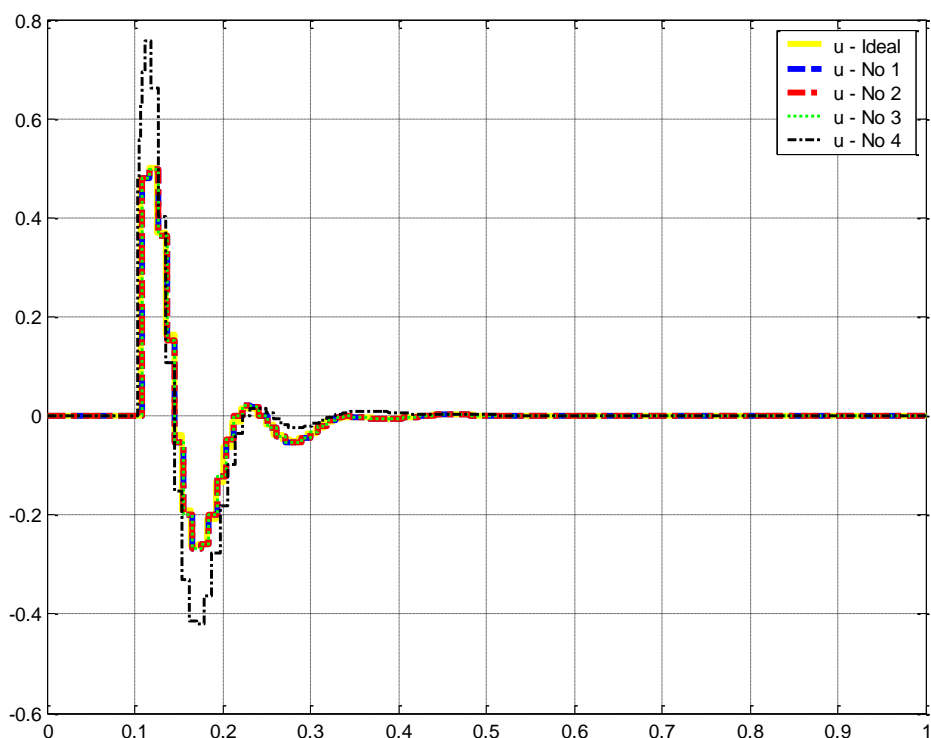


Figura F.10 – Lei de Controle com viés inicial de 0.39 segundos e deriva 10%.

A Figura F.9 mostra a resposta ao degrau do sistema de controle com o viés inicial de 0.39 segundos e deriva de 10%. É possível ver que somente o nó 4 sofre variação em relação ao ideal e aos outros nós do sistema. O nó 4 atrasa sua lei de controle em 0.39 segundos e também ao longo de sua operação o relógio 4 esta se atrasando a uma taxa de deriva de 10%. Isto ocorre pois o relógio do nó 4 inicia atrasado e esta com deriva em relação aos outros relógios do conjunto. Ainda na Figura F.9 pode-se ver que o erro de relógio aumenta o pico de subida da resposta ao degrau. A Figura F.10 mostra as leis de controle aplicadas a cada nó do sistema e do sistema ideal. Somente o nó 4 difere sua resposta ao degrau e a lei de controle dos outros nós do sistema. O sistema de controle do nó 4 sofre o impacto do adiantamento e da deriva, a lei de controle do nó 4 sofre um aumento na sua amplitude. Para tentar corrigir este problema vamos aplicar o algoritmo FTM de sincronização de relógios.

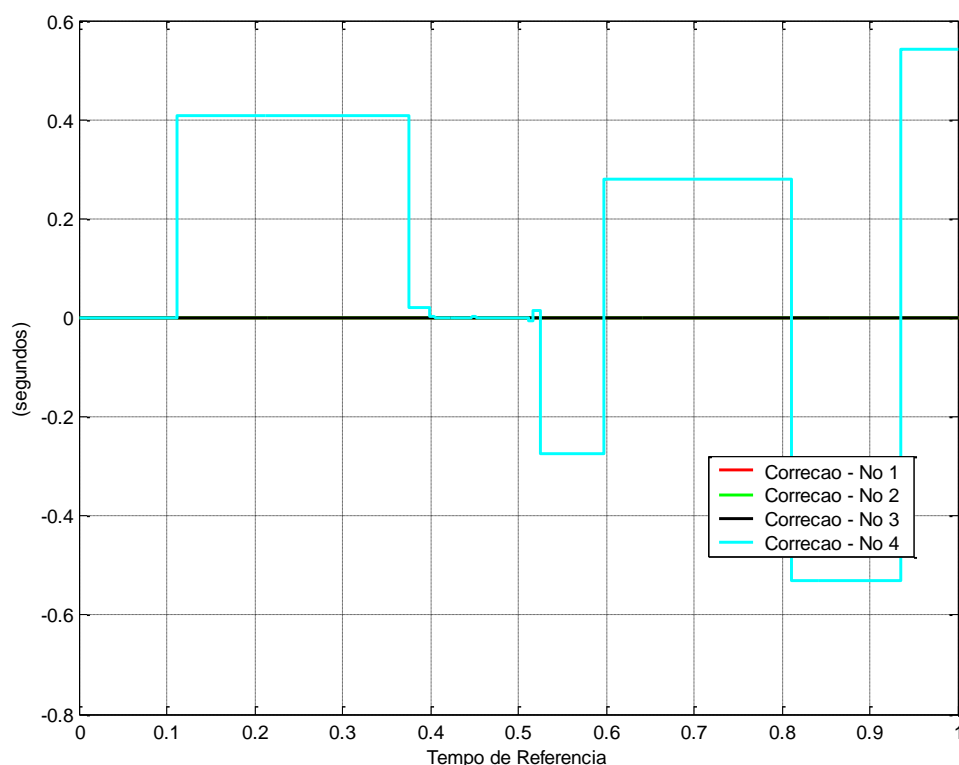


Figura F.11 – Correção Algoritmo FTM: Viés inicial de 0.39 segundos e Deriva 10%.

Na Figura F.11 é possível ver o gráfico de correção dos relógios dos respectivos nós do sistema. A Figura F.11 mostra que o algoritmo FTM não convergiu. Os nós 1, 2 e 3, que não estavam com erro de viés inicial, fizeram uma correção periódica praticamente nula. O nó 4, com viés inicial 0.39 segundos e deriva de 10%, teve vários picos grandes de correção, mas no tempo de simulação o algoritmo não convergiu a um valor.

A Figura F.12 mostra a linha do tempo com o algoritmo de sincronização de relógios FTM aplicado ao sistema. O período de re-sincronização é de 0.0040 segundos, isto é, a cada 0.0040 segundos o algoritmo FTM atualiza os valores de relógio com a correção calculada.



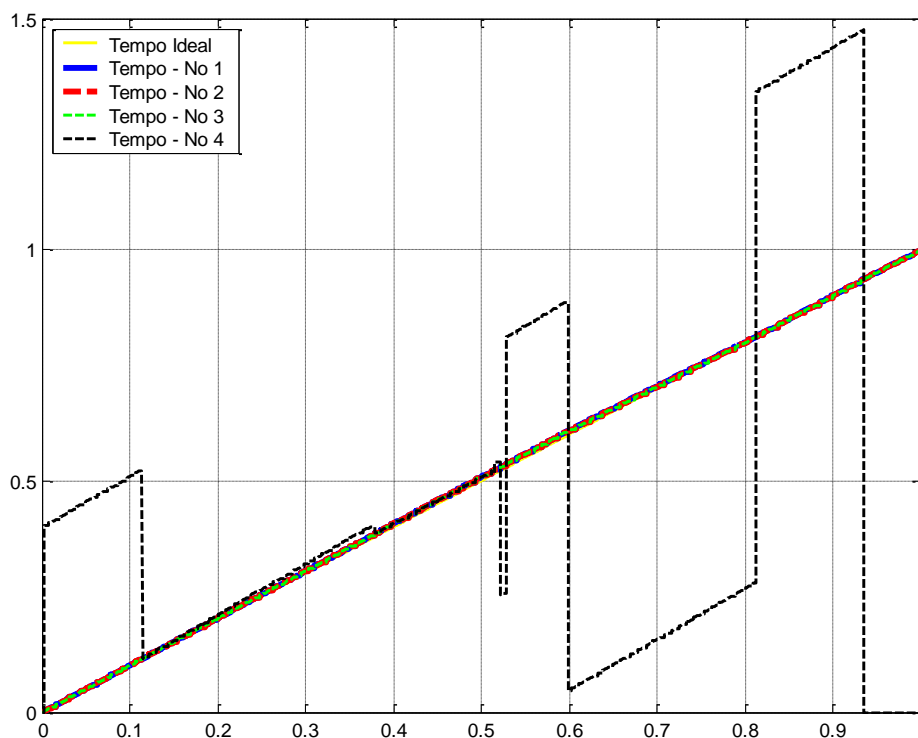


Figura F.12 – Linha do Tempo Sincronizada: Viés inicial de 0.39 segundos e Deriva: 10%.

A Figura F.12 mostra que os relógios dos nós 1, 2 e 3 se sincronizaram entre si, mas o relógio do nó 4 não conseguiu se sincronizar. O algoritmo de sincronização de relógios FTM tem em sua suposição que todos os relógios da rede devem iniciar sincronizados, isto é, sem viés inicial em nenhum relógio do conjunto. Este caso desobedece a regra e o algoritmo não consegue sincronizar todos os relógios do conjunto. Apesar do relógio 4 ser prejudicado, pelo fato do algoritmo de sincronização ser tolerante a falhas, os outros nós do conjunto, os nós 1, 2 e 3, não são afetados pelo erro no nó 4.

A Figura F.13 mostra a resposta ao degrau do sistema de controle. E a Figura F.14 mostra as leis de controle.

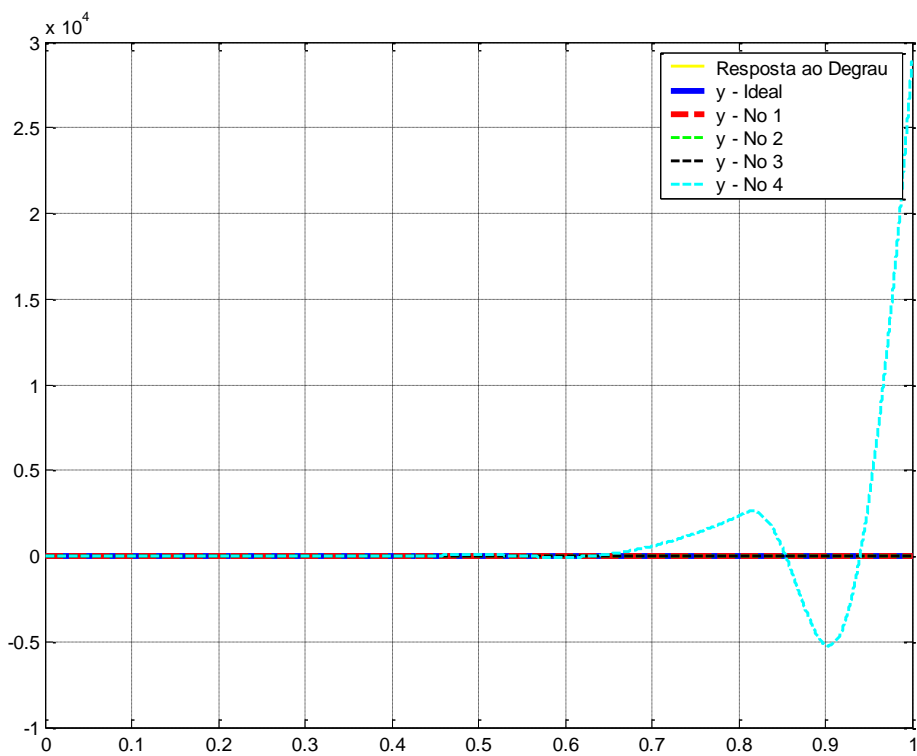


Figura F.13 – Resposta ao Degrau Sincronizado: Viés inicial: 0.39 segundos, Deriva: 10%.

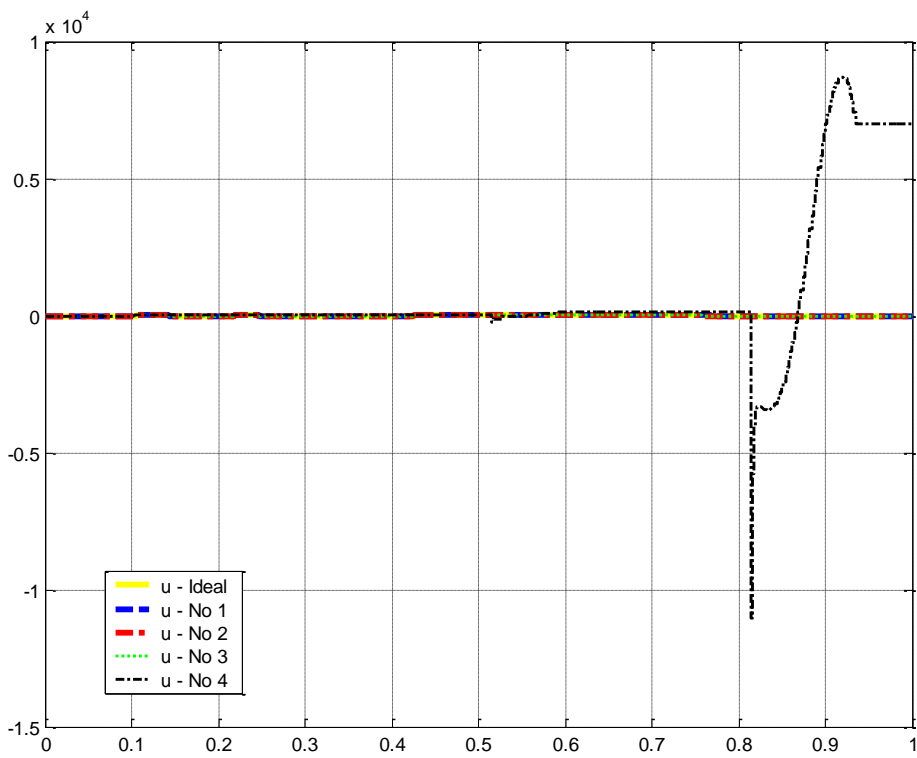


Figura F.14 – Lei de Controle Sincronizada: Viés inicial: 0.39 segundos, Deriva: 10%.

través das Figuras F.13 e F.14 é possível ver que a sincronização de relógios aumentou o erro que o viés inicial e a deriva do relógio 4 causaram à resposta ao degrau e a lei de controle do nó 4, fazendo com que o nó 4 perca sua estabilidade. Os relógios dos nós 1, 2 e 3 estão sincronizados entre si e seus sistemas de controle não foram afetados. Neste caso o algoritmo mostrou-se ineficiente ao corrigir o erro de viés inicial e deriva do relógio 4, amplificando o efeito negativo na lei de controle.



## **APÊNDICE G – SIMULAÇÃO DO 4º CASO – ARQUITETURA FEDERADA – ALGORITMO FK**

Neste caso, foi feita a mesma simulação de um sistema de controle do 1º Caso, com a diferença que a sincronização de relógios agora é feita através do **Filtro de Kalman**, com arquitetura centralizada, isto é, existe um relógio de referência. O relógio escolhido como relógio correto é o relógio do nó 1. Nesta simulação variou-se o parâmetro de deriva do relógio do nó (subsistema) 4. A rede de comunicação foi feita por um protocolo de comunicação com acesso ao meio TDMA.

Este trabalho escolheu 3 taxas de deriva diferentes para variação do relógio:

- **Caso A: -10%** - O relógio está se atrasando em relação ao relógio de referência a uma taxa de 10%;
- **Caso B: 1%** - O relógio está se adiantando em relação ao relógio de referência com taxa de 1%;
- **Caso C: 10%** - O relógio está se adiantando em relação ao relógio de referência com taxa de 10%;

No **caso A**, a uma **taxa de deriva de -10%**, o relógio do nó 4 esta se atrasando em relação ao relógio de referência. Os outros relógios do conjunto não estão com erro de relógio. A Figura D.1 mostra a linha do tempo com erro de deriva de -10% no relógio 4. Com isso, o sistema sofre uma pequena variação no sistema de controle do nó 4 que esta afetado pelo relógio. A figura D.2 mostra a resposta ao Degrau com deriva de -10%. O nó 4 sofre variação em relação ao nó 1 e aos outros nós do sistema. Isto ocorre pois o relógio do nó 4 esta atrasado em relação ao relógio de referência e aos outros relógios do conjunto. A Figura D.3 mostra as leis de controle aplicadas a cada nó do sistema. Somente o nó 4 difere dos outros nós do conjunto.

Apesar de sofrer variação o sistema de controle não sofre grande impacto com uma taxa de deriva de -10%.

Mesmo com um pequeno impacto, vamos aplicar o Filtro de Kalman na sincronização de relógios. A Figura G.3 mostra a linha do tempo sincronizada com o Filtro de Kalman aplicado ao sistema. O período de re-sincronização é de 0.0040 segundos, isto é, a cada 0.0040 segundos o Filtro de Kalman atualiza os valores de relógio com a correção calculada. Na Figura G.1 é possível ver o gráfico de correção dos relógios dos respectivos nós do sistema e na Figura G.2 uma aproximação da figura G.1.

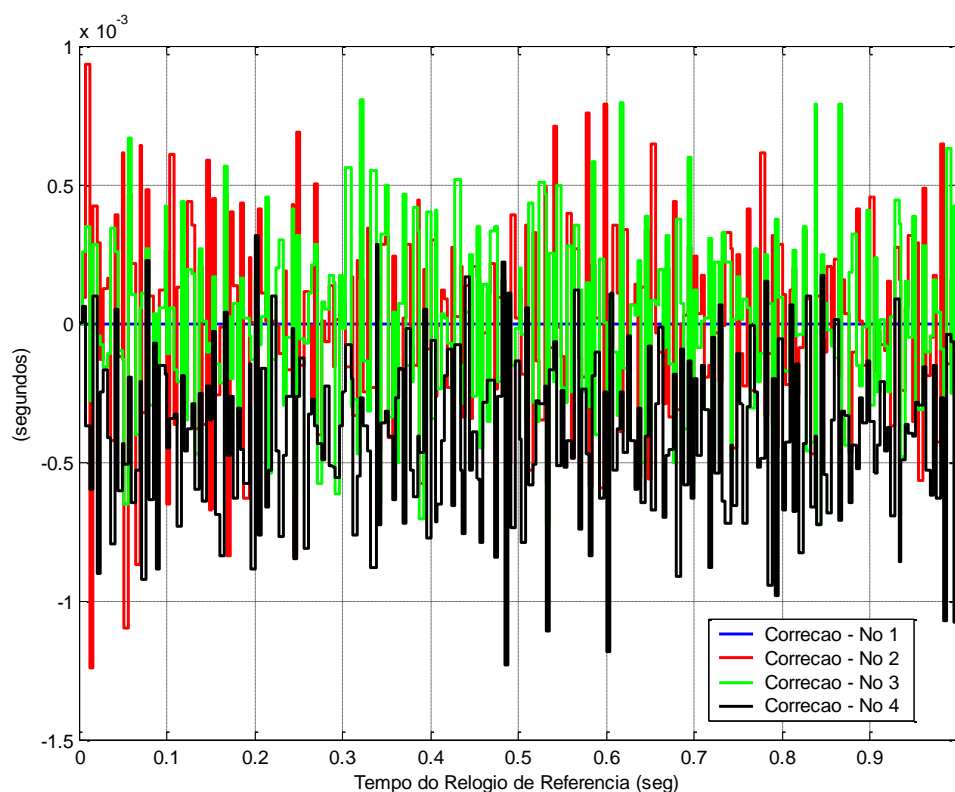


Figura G.1 – Correção com Filtro de Kalman com deriva de -10%.

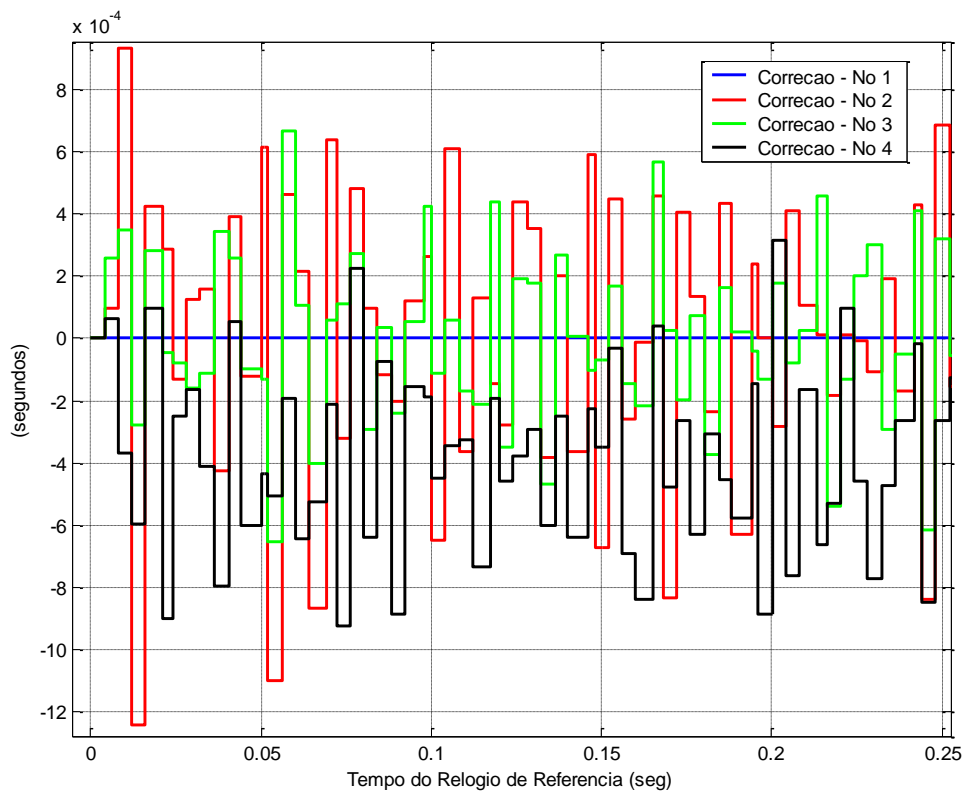
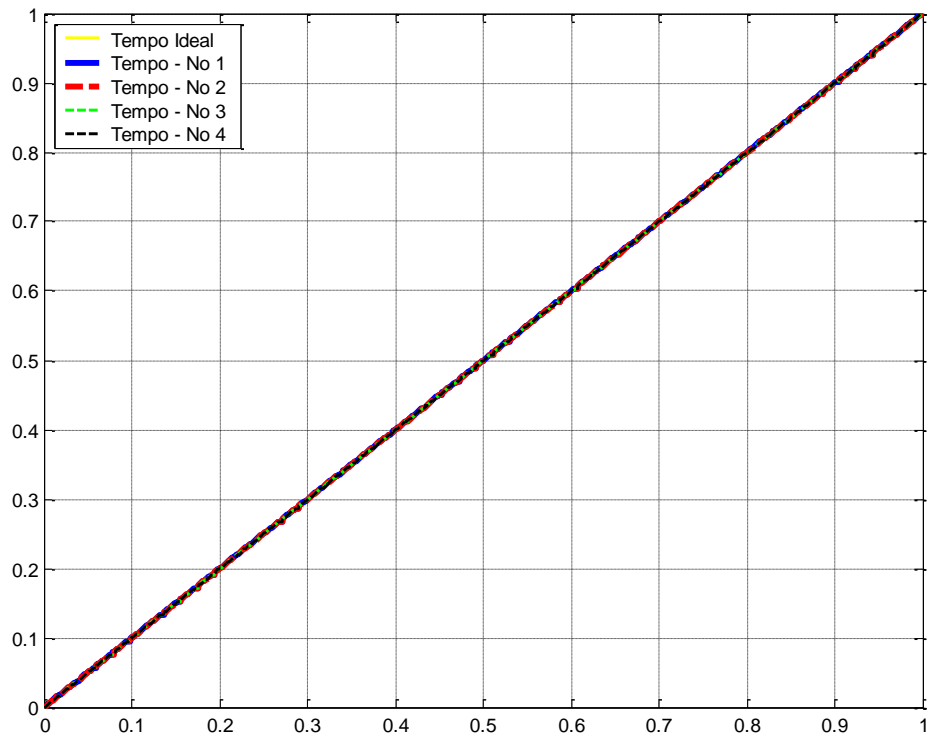


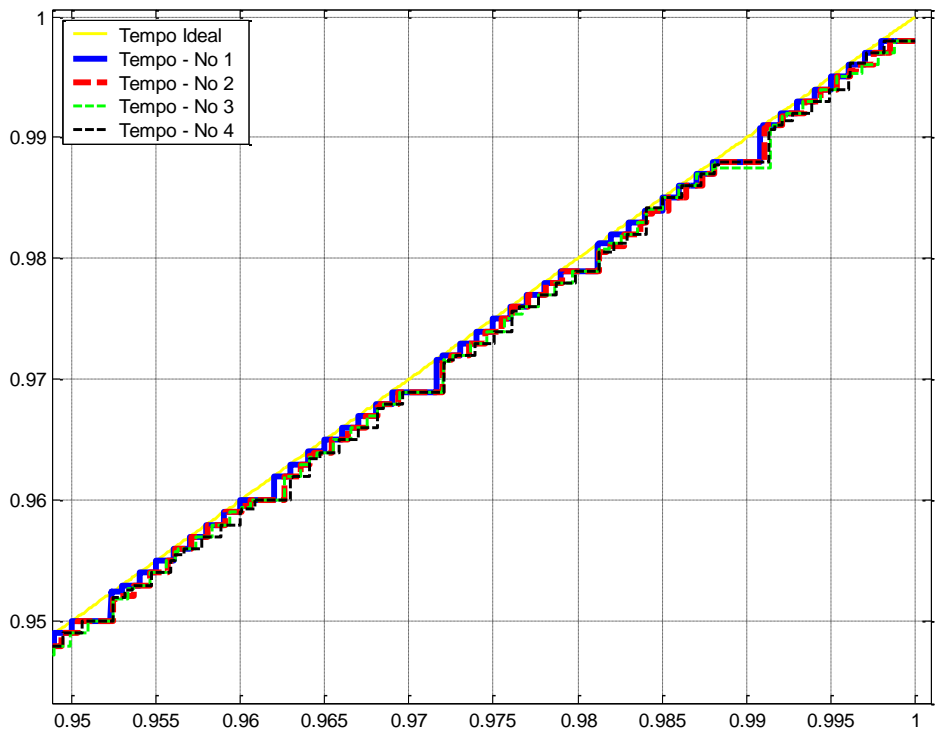
Figura 4.2 – Aproximação - Correção Filtro de Kalman com deriva de -10%.

A Figura G.1 mostra que o Filtro de Kalman convergiu. Os nó 1 não corrige seu valor pois ele é a referência. Os nós 2 e 3, que não estavam com erro de deriva, fizeram uma correção periódica com o valor médio de zero. O nó 4, com -10% de taxa de deriva, teve uma correção periódica com o valor médio em torno de -0.4 (ms). É possível analisar através do gráfico G.1 e G.2 que a precisão dos relógios do sistema ficou em torno de 2 (ms).

A Figura G.3 mostra que os relógios dos nós se sincronizaram com o relógio de referência do nó 1.



(a)



(b)

Figura G.3 – (a) Linha do Tempo com deriva de -10%. Sincronizado (b) Aproximação.



A Figura G.4 mostra a resposta ao degrau do sistema de controle com o sistema sincronizado pelo filtro de Kalman. E a Figura G.5 mostra as leis de controle do sistema sincronizado com filtro de Kalman.

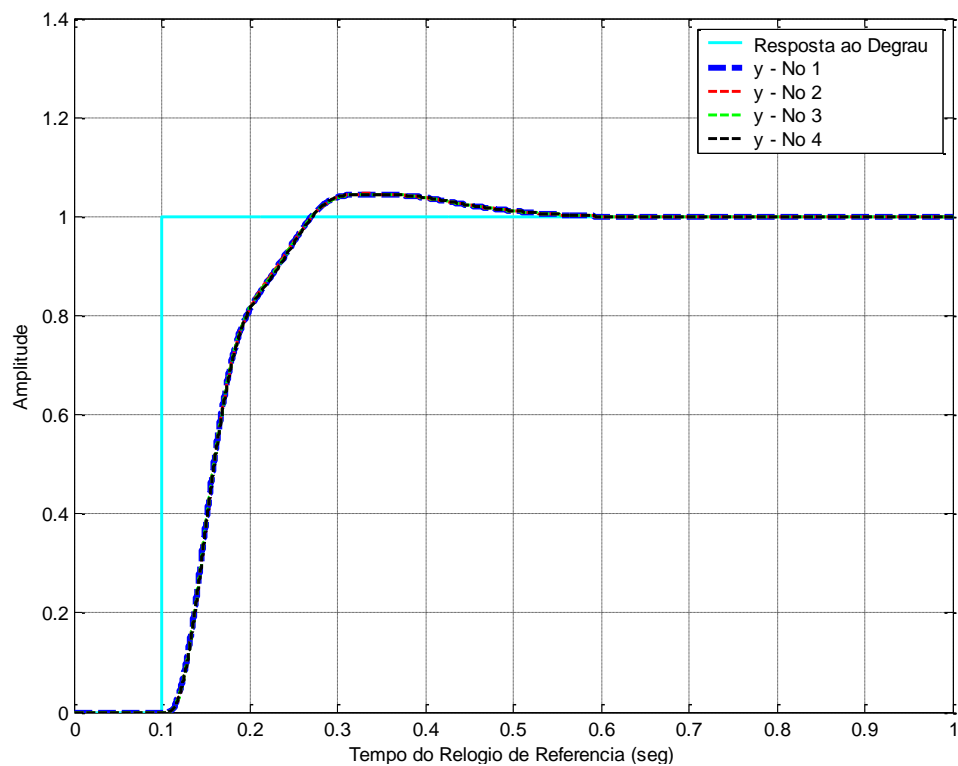
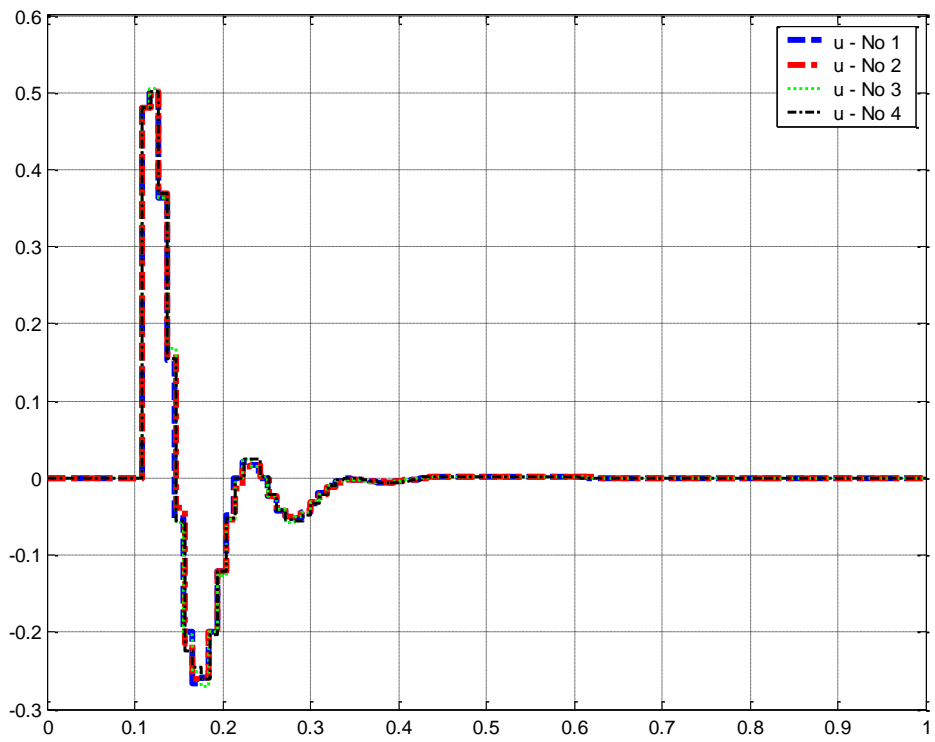
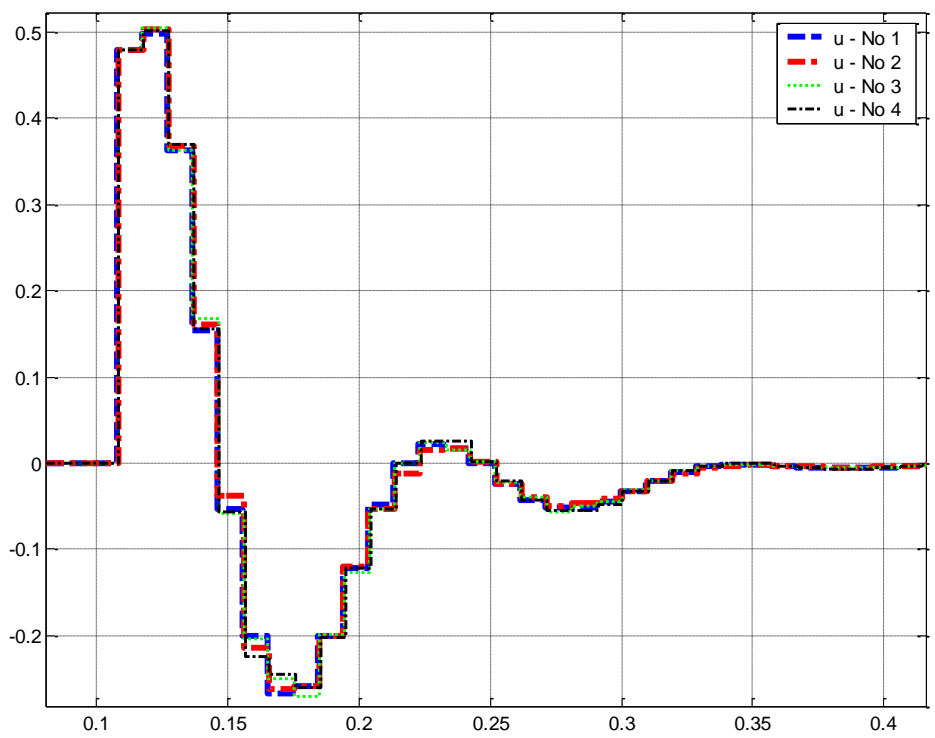


Figura G.4 – Resposta ao Degrau com derivada de -10% Sincronizado.



(a)



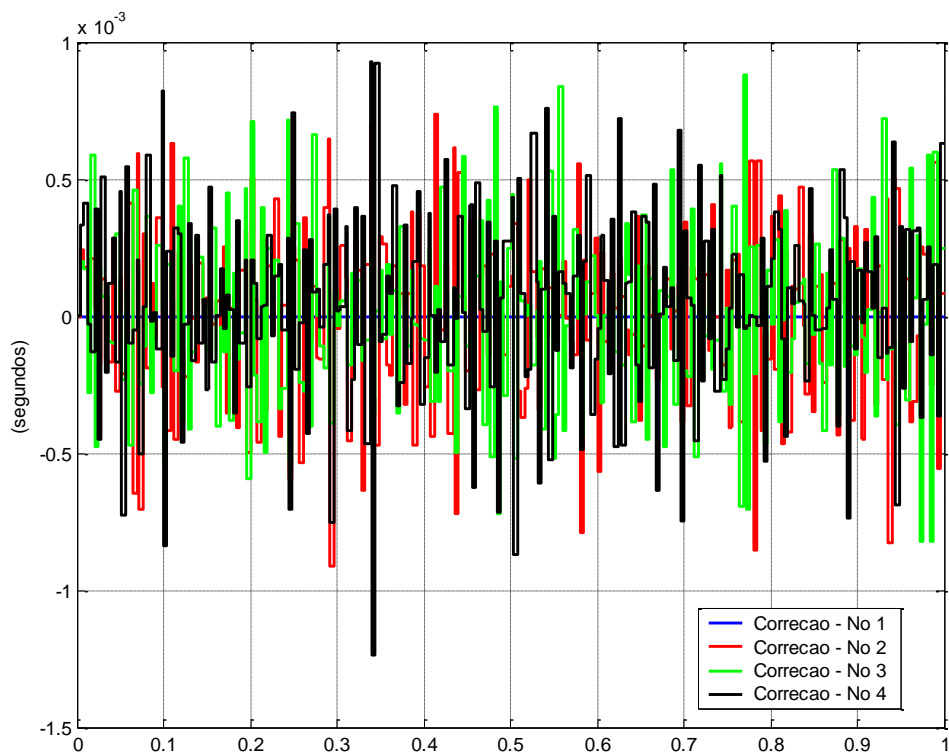
(b)

Figura G.5 – (a) Lei de Controle com deriva de -10% Sincronizado. (b) Aproximação.

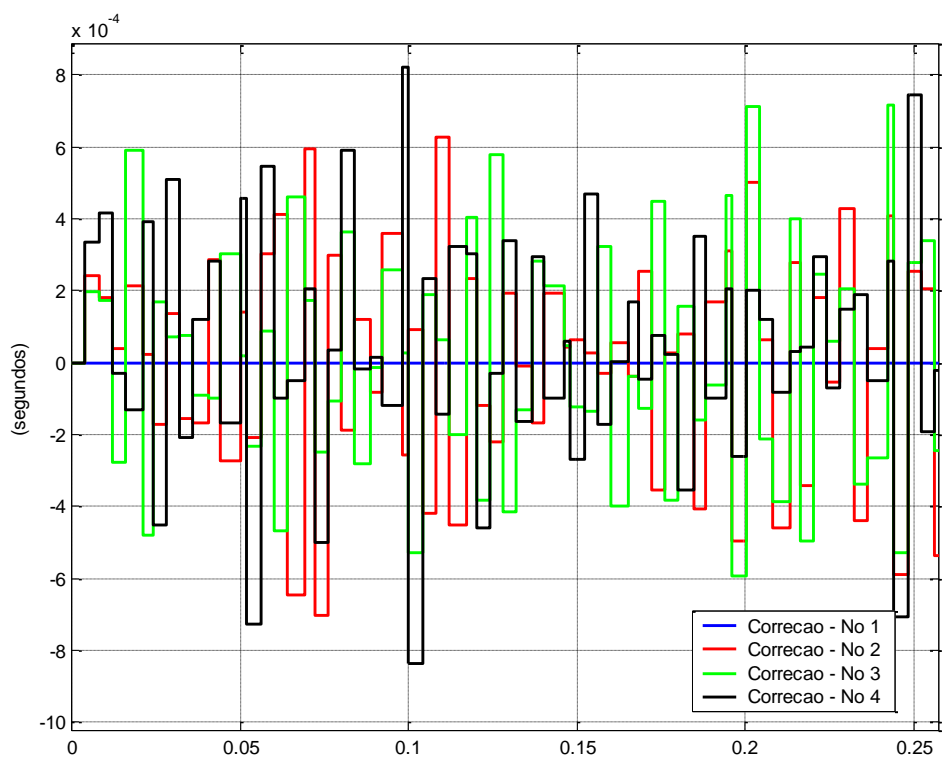
Através das Figuras G.4 e G.5 é possível ver que a sincronização de relógios com o Filtro de Kalman minimizou a diferença do tempo de resposta e das leis de controle de cada nó. Apesar do nó 4 ser um pouco atrasado do nó 1, os tempos de respostas estão sincronizados entre si. Neste caso o algoritmo mostrou-se muito eficiente corrigindo o erro de deriva presente no nó 4 sistema.

No **caso B**, a uma **taxa de deriva de 1%**, o relógio do nó 4 esta se atrasando em relação ao relógio de referência. Os outros relógios do conjunto não estão com erro de relógio. A Figura D.8 mostra a linha do tempo com erro de deriva de 1% no relógio 4. Com isso, o sistema sofre uma pequena variação no sistema de controle do nó 4 que esta afetado pelo relógio. A figura D.9 mostra a resposta ao Degrau com deriva de 1%. O nó 4 sofre uma variação muito pequena em relação ao nó 1 e aos outros nós do sistema. Isto ocorre pois o relógio do nó 4 esta se adiantando em relação ao relógio de referência. A Figura D.10 mostra as leis de controle aplicadas a cada nó do sistema. Somente o nó 4 difere muito pouco dos outros nós do conjunto. Apesar de sofrer variação de relógio o sistema de controle não sofre grande impacto com uma taxa de deriva de 1%.

Mesmo com um pequeno impacto, vamos aplicar o Filtro de Kalman na sincronização de relógios. A Figura G.7 mostra a linha do tempo sincronizada com o Filtro de Kalman aplicado ao sistema. O período de re-sincronização é de 0.0040 segundos, isto é, a cada 0.0040 segundos o Filtro de Kalman atualiza os valores de relógio com a correção calculada. Na Figura G.6 é possível ver o gráfico de correção dos relógios dos respectivos nós do sistema.

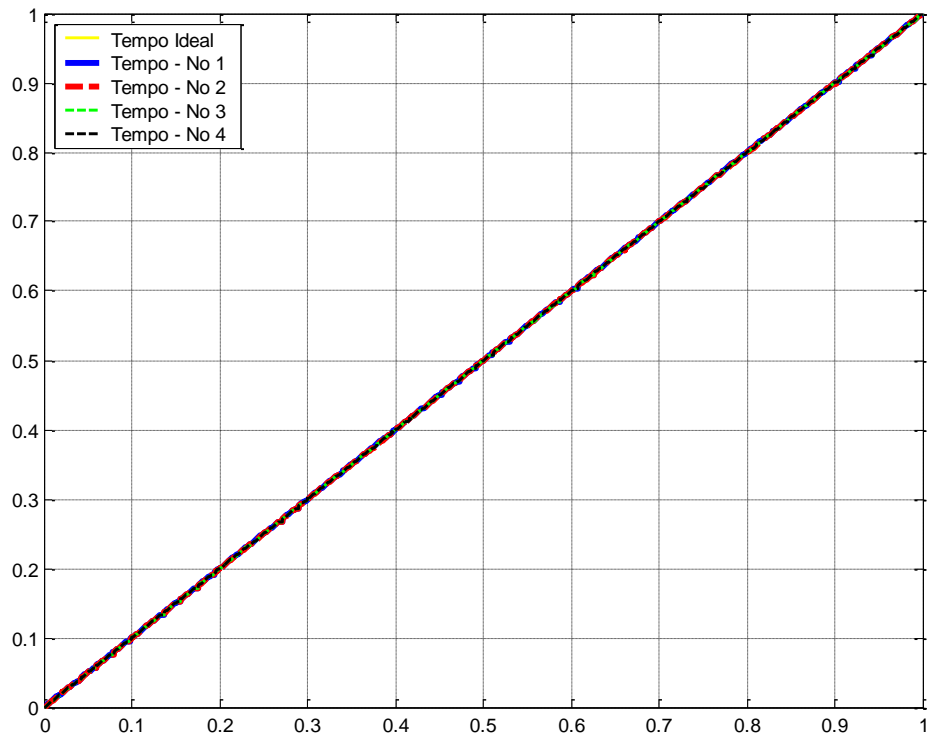


(a)

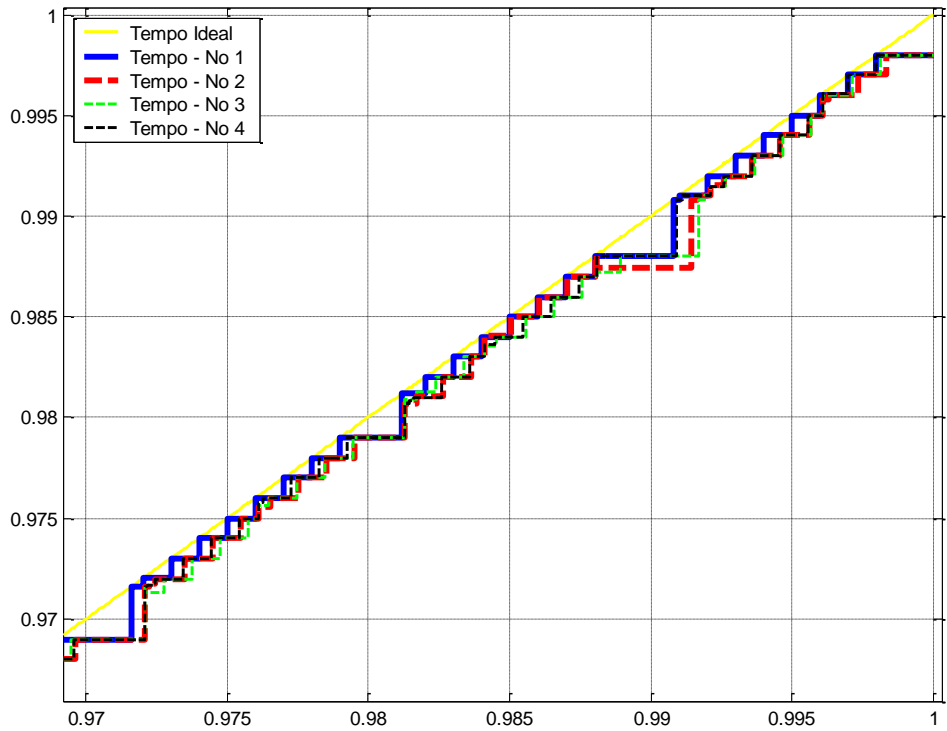


(b)

Figura G.6 – (a) Correção com Filtro de Kalman com deriva de 1%. (b) Aproximação.



(a)



(b)

Figura G.7 – (a) Linha do Tempo com deriva de 1%. Sincronizado (b) Aproximação.

A Figura G.6 mostra que o Filtro de Kalman convergiu. Os nó 1 não corrige seu valor pois ele é a referência. Os nós 2 e 3, que não estavam com erro de deriva, fizeram uma correção periódica com o valor médio de zero. O nó 4, com 1% de taxa de deriva, teve uma correção periódica com o valor médio muito próxima de zero. É possível analisar através do gráfico G.6 que a precisão dos relógios do sistema ficou em torno de 2 (ms).

A Figura G.7 mostra que os relógios dos nós se sincronizaram com o relógio de referência do nó 1. A Figura G.8 mostra a resposta ao degrau do sistema de controle com o sistema sincronizado pelo filtro de Kalman.

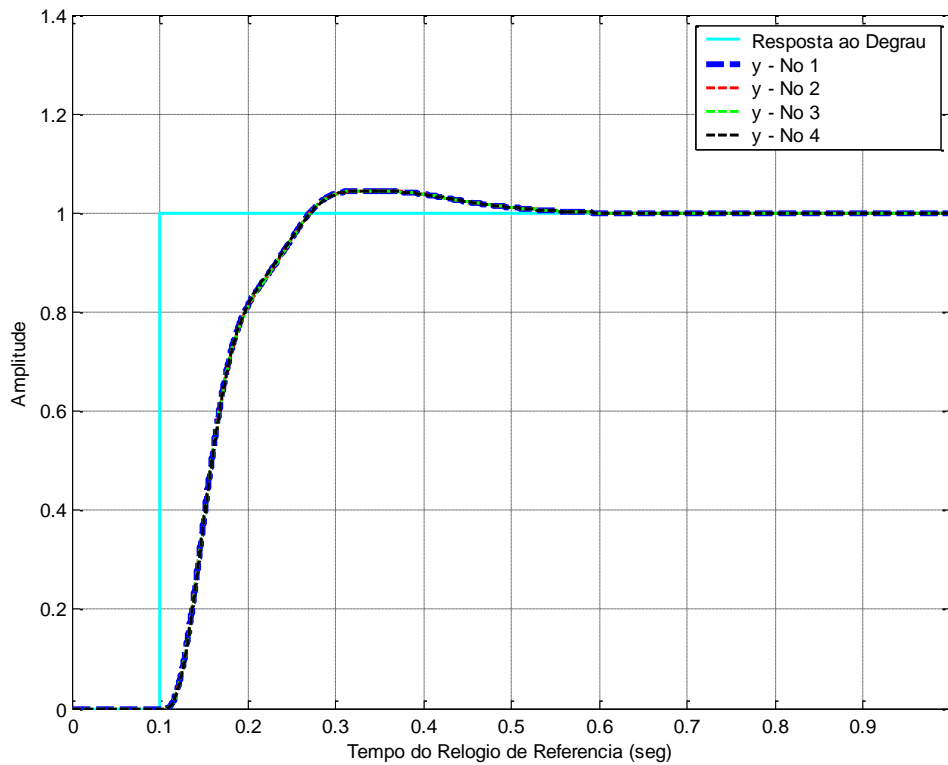
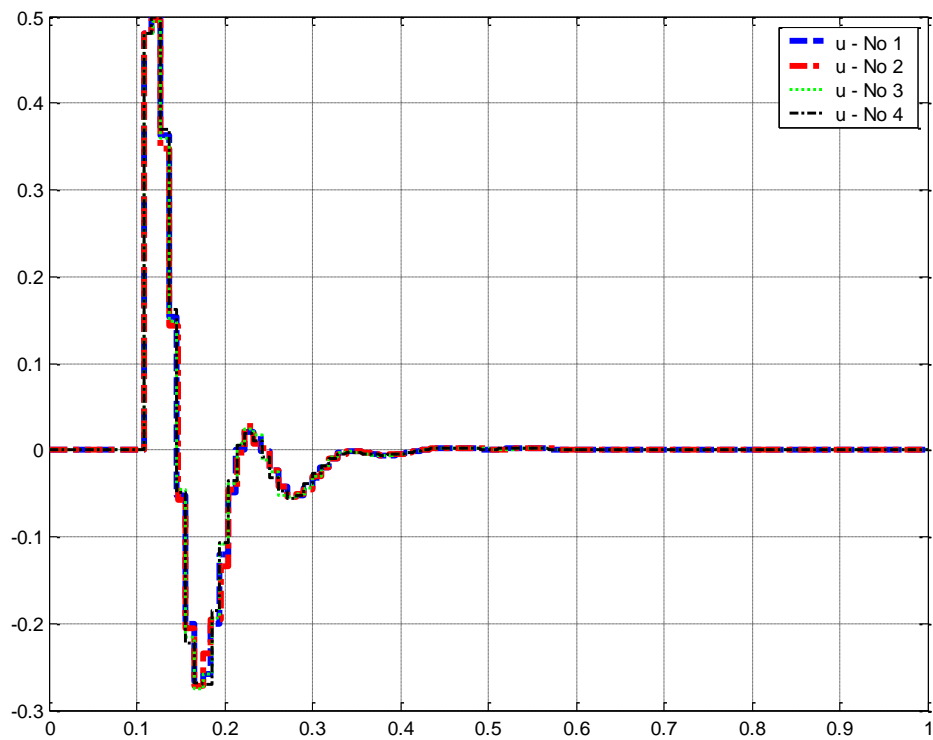
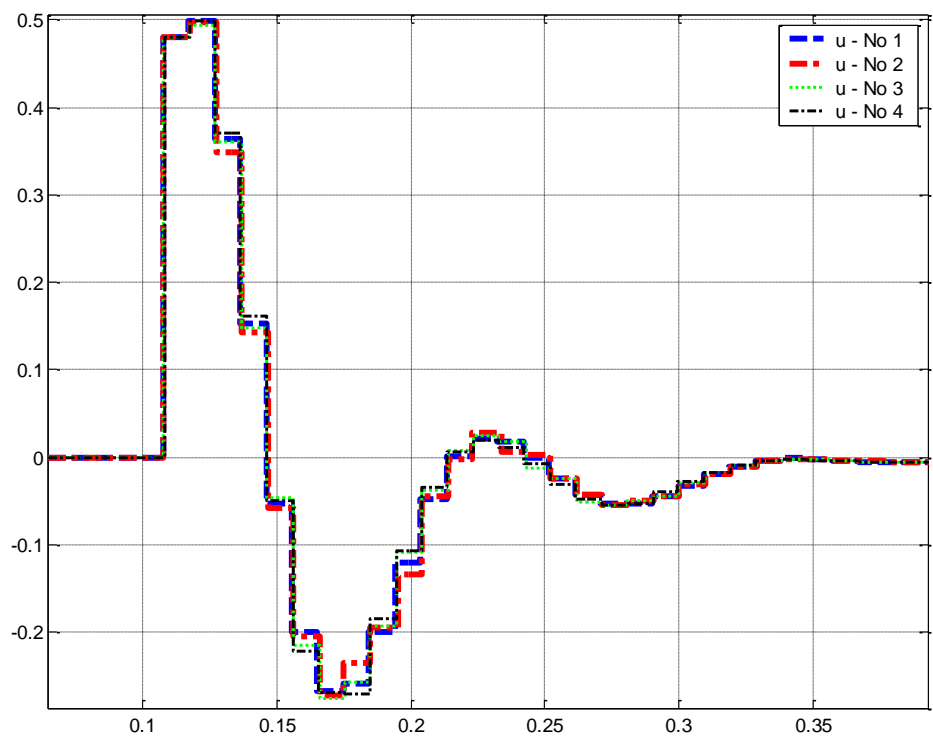


Figura G.8 – Resposta ao Degrau com deriva de 1% Sincronizado.

A Figura G.9 mostra as leis de controle do sistema sincronizado com filtro de Kalman.



(a)



(b)

Figura G.9 – (a) Lei de Controle com deriva de 1% Sincronizado. (b) Aproximação.

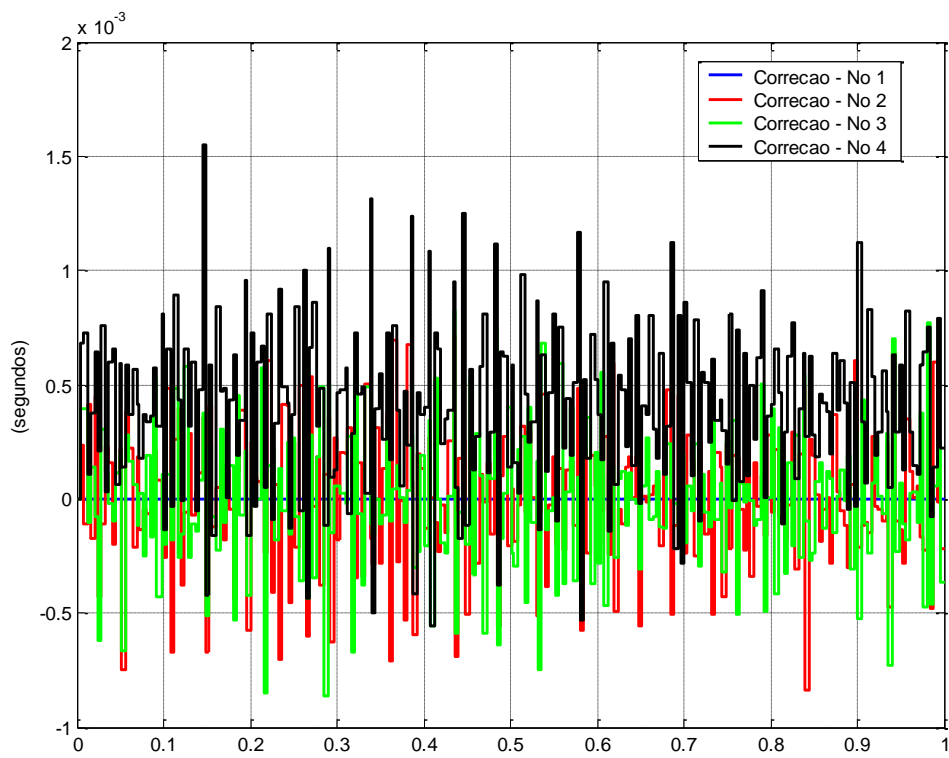
Através das Figuras G.8 e G.9 é possível ver que a sincronização de relógios com o Filtro de Kalman minimizou a diferença do tempo de resposta e das leis de controle de cada nó. Apesar do nó 4 ser um pouco adiantado do nó 1, os tempos de respostas estão sincronizados entre si. Neste caso o algoritmo mostrou-se eficiente corrigindo o erro de deriva presente no nó 4 sistema, apesar do erro inicial do relógio praticamente não influenciar a lei de controle.

No **caso C**, a uma **taxa de deriva de 10%**, o relógio do nó 4 esta se atrasando em relação ao relógio de referência. Os outros relógios do conjunto não estão com erro de relógio. A Figura D.15 mostra a linha do tempo com erro de deriva de 10% no relógio 4. Com isso, o sistema sofre uma variação no sistema de controle do nó 4 que esta afetado pelo relógio. A figura D.16 mostra a resposta ao Degrau com deriva de 10%. O nó 4 sofre uma variação pequena em relação ao nó 1 e aos outros nós do sistema. Isto ocorre pois o relógio do nó 4 esta se adiantando em relação ao relógio de referência. A Figura D.17 mostra as leis de controle aplicadas a cada nó do sistema. Somente o nó 4, com erro de deriva, difere dos outros nós do conjunto e do relógio 1 de referência.

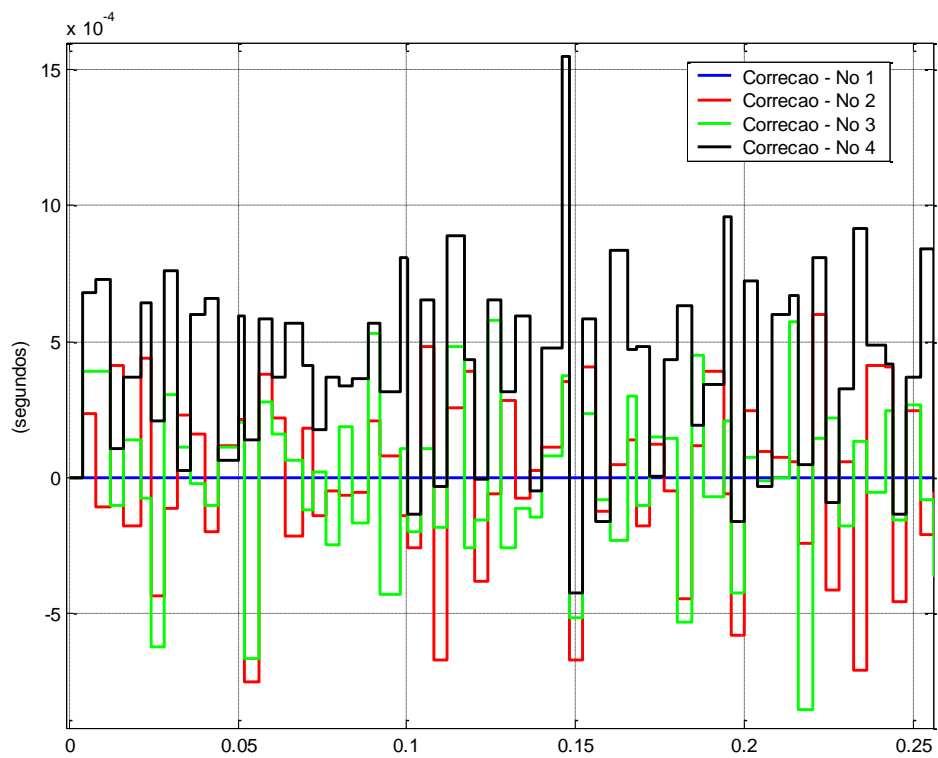
Apesar de sofrer variação de relógio o sistema de controle não sofre grande impacto com uma taxa de deriva de 10%. Com essa deriva o sinal de resposta ao degrau do nó 4 tem uma pequena elevação da % de sobre-elevação.

Aplicando o Filtro de Kalman na sincronização de relógios. A Figura G.11 mostra a linha do tempo sincronizada com o Filtro de Kalman aplicado ao sistema. O período de re-sincronização é de 0.0040 segundos, isto é, a cada 0.0040 segundos o Filtro de Kalman atualiza os valores de relógio com a correção calculada. Na Figura G.10 é possível ver o gráfico de correção dos relógios dos respectivos nós do sistema.



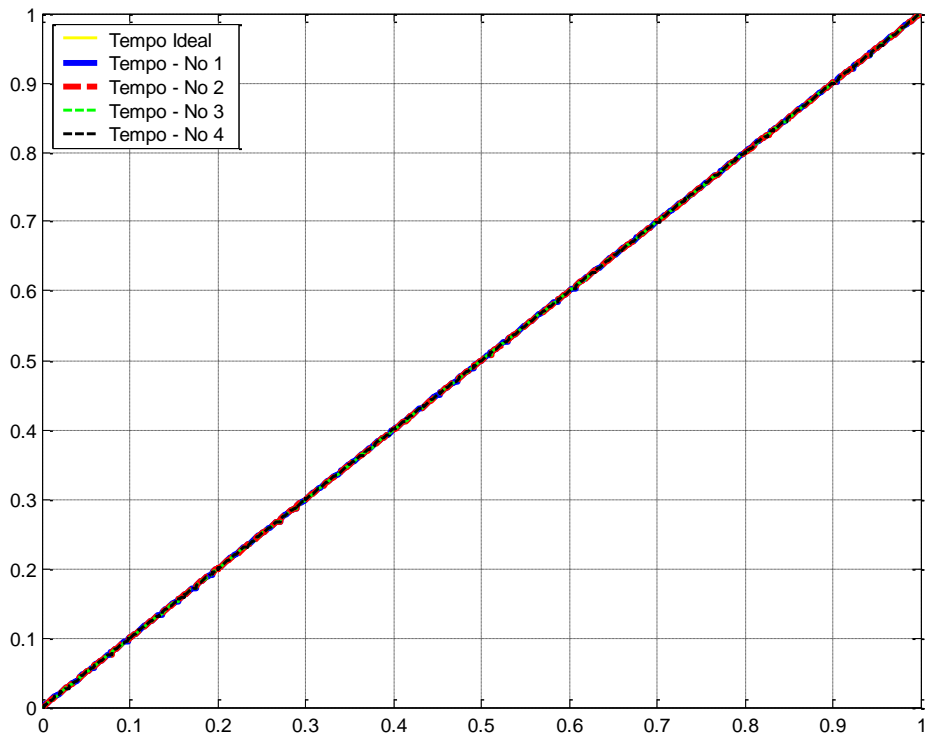


(a)

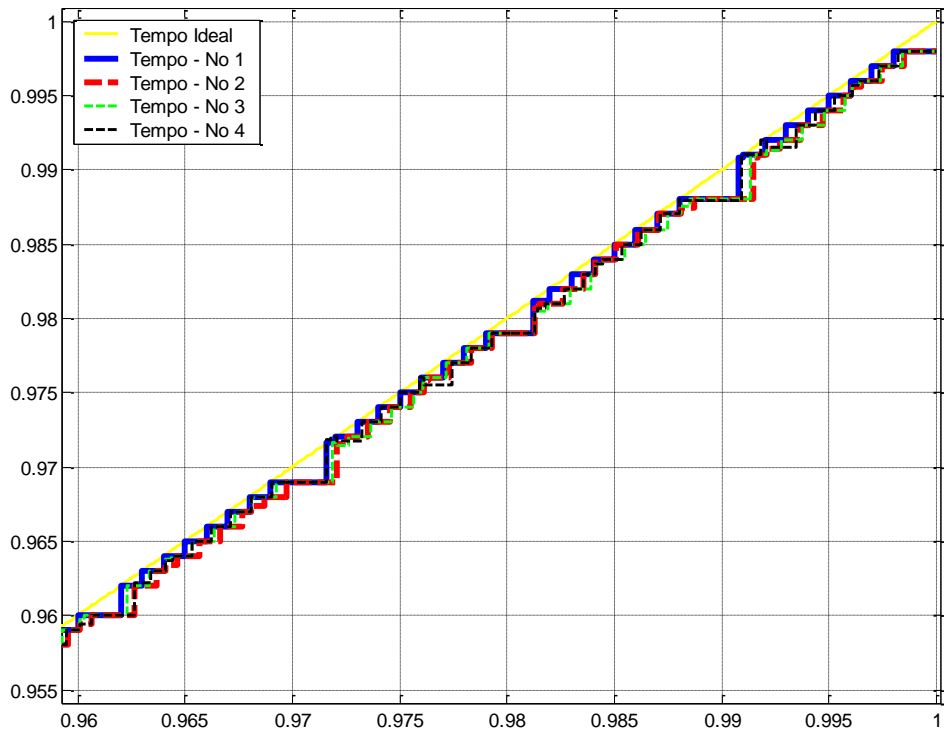


(b)

Figura G.10 – (a) Correção com Filtro de Kalman com deriva de 10%.(b) Aproximação.  
289



(a)



(b)

Figura G.11 – (a) Linha do Tempo com deriva de 10%. Sincronizado (b) Aproximação.

A Figura G.10 mostra que o Filtro de Kalman convergiu. Os nó 1 não corrige seu valor pois ele é a referência. Os nós 2 e 3, que não estavam com erro de deriva, fizeram uma correção periódica com o valor médio de zero. O nó 4, com 10% de taxa de deriva, teve uma correção periódica com o valor médio 4 (ms). É possível analisar através do gráfico G.10 que a precisão dos relógios do sistema ficou em torno de 2 (ms).

A Figura G.11 mostra que os relógios dos nós se sincronizaram com o relógio de referência do nó 1. A Figura G.12 mostra a resposta ao degrau do sistema de controle com o sistema sincronizado pelo filtro de Kalman.

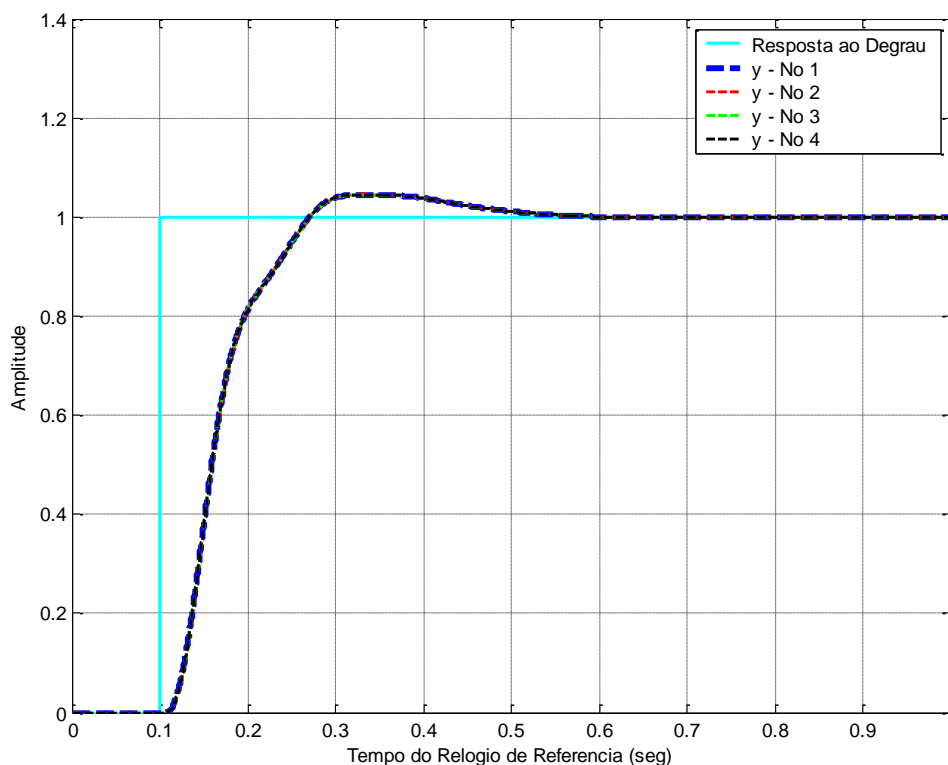
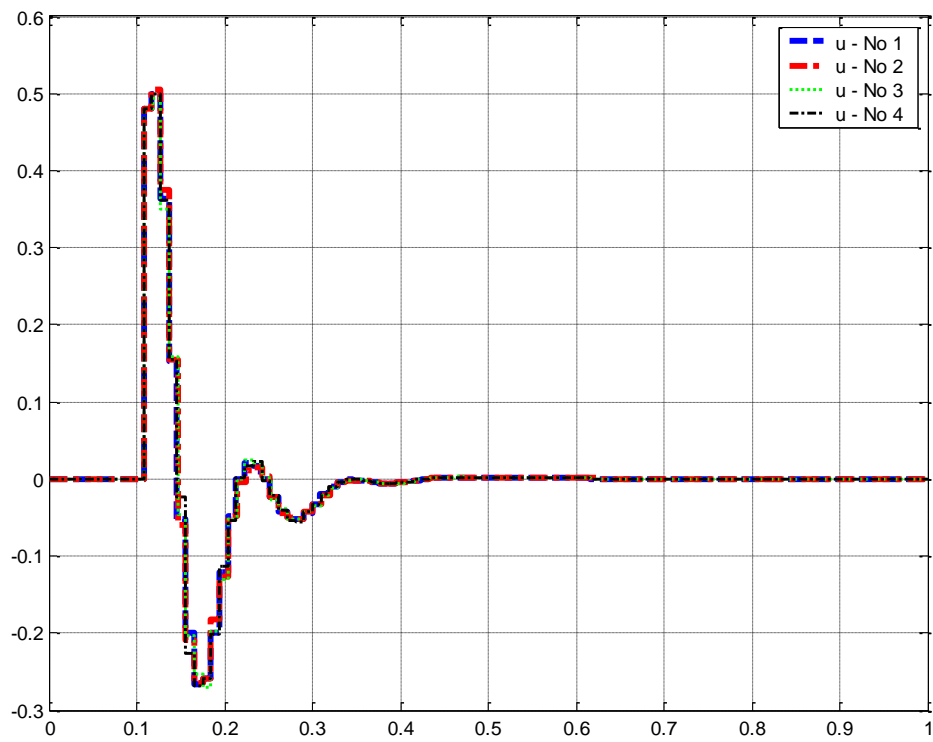
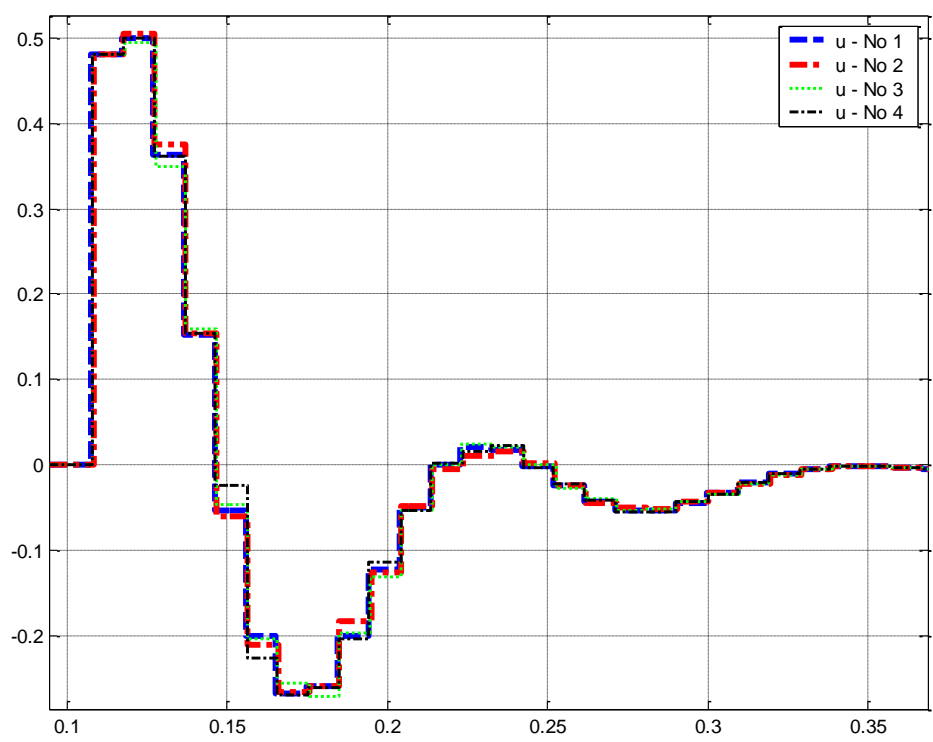


Figura G.12 – Resposta ao Degrau com deriva de 10% Sincronizado.

A Figura G.13 mostra as leis de controle do sistema sincronizado com filtro de Kalman.



(a)



(b)

Figura G.13 – (a) Lei de Controle com deriva de 10% Sincronizado. (b) Aproximação.

Através das Figuras G.12 e G.13 é possível ver que a sincronização de relógios com o Filtro de Kalman minimizou a diferença do tempo de resposta e das leis de controle de cada nó. Apesar do nó 4 ser um pouco adiantado do nó 1, os tempos de respostas estão sincronizados entre si. Neste caso o algoritmo mostrou-se eficiente corrigindo o erro de deriva presente no nó 4 sistema. Com essa correção de relógios o algoritmo de sincronização com Filtro de Kalman eliminou a % de sobre-elevação presente quando o relógio do nó 4 não esta sincronizado.



## APÊNDICE H – SIMULAÇÃO DO 5º CASO – ARQUITETURA FEDERADA – ALGORITMO FK

Neste caso, foi feita a mesma simulação de um sistema de controle do 2º Caso, com a diferença que a sincronização de relógios agora é feita através do Filtro de Kalman, com arquitetura centralizada, isto é, existe um relógio de referência. O relógio escolhido como relógio correto é o relógio do nó 1. Nesta simulação variou-se o parâmetro de viés inicial do relógio do nó (subsistema) 4. A rede de comunicação foi feita por um protocolo de comunicação com acesso ao meio TDMA.

Este trabalho escolheu 2 valores de viés inicial diferentes para variação do relógio:

- **Caso A: -0.2 (seg)** - O relógio inicia atrasado de 0.2 segundos em relação aos outros;
- **Caso B: 0.39 (seg)** - O relógio inicia adiantado de 0.39 segundos em relação aos outros;

Foram escolhidos estes dois valores de viés inicial, pois em valores menores, os sistemas de controle sofrem muito pouco, ou quase nenhuma variação.

No **caso A**, com um **viés inicial de -0.2 segundos**, o relógio do nó 4 está atrasado em relação ao relógio de referência e aos outros relógios do conjunto, como mostra a Figura E.1. Com isso, o sistema de controle do nó 4 sofre uma pequena variação que está afetado pelo relógio.

A Figura E.2 mostra a resposta ao degrau do sistema de controle com o viés inicial de -0.2 segundos. É possível ver que somente o nó 4 sofre variação em relação ao de referência e aos outros nós do sistema. O nó 4 atrasa sua lei de controle em 0.2 segundos. Isto ocorre pois o relógio do nó 4 inicia atrasado em relação aos outros e ao de referência. A Figura E.3 mostra as leis de controle

aplicadas a cada nó do sistema e do sistema ideal. Somente o nó 4 difere sua resposta ao degrau e a lei de controle do nó 1, de referência, e dos outros nós do sistema. O sistema de controle do nó 4 sofre o impacto do atraso, para tentar corrigir este problema vamos aplicar Filtro de Kalman para a sincronização de relógios.

Aplicando o Filtro de Kalman na sincronização de relógios. A Figura H.3 e H.4 mostra a linha do tempo sincronizada com o Filtro de Kalman aplicado ao sistema. O período de re-sincronização é de 0.0040 segundos, isto é, a cada 0.0040 segundos o Filtro de Kalman atualiza os valores de relógio com a correção calculada. Na Figura H.1 e H.2 é possível ver o gráfico de correção dos relógios dos respectivos nós do sistema.

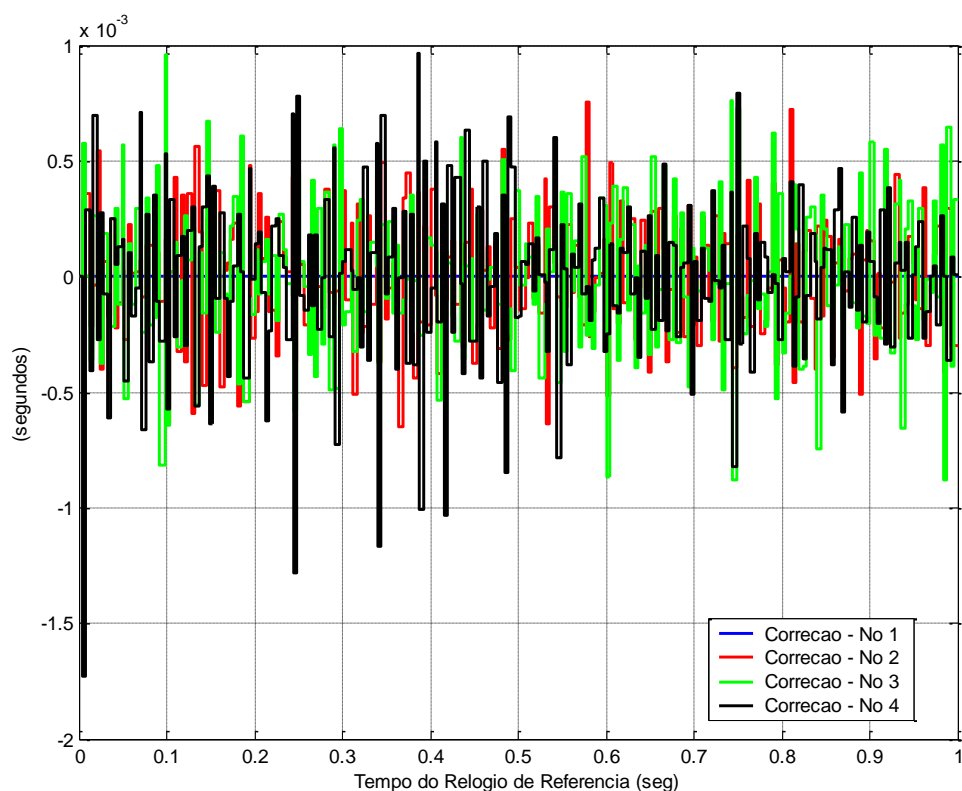


Figura H.1 – (a) Correção do Filtro de Kalman, Viés: -0.2 segundos.



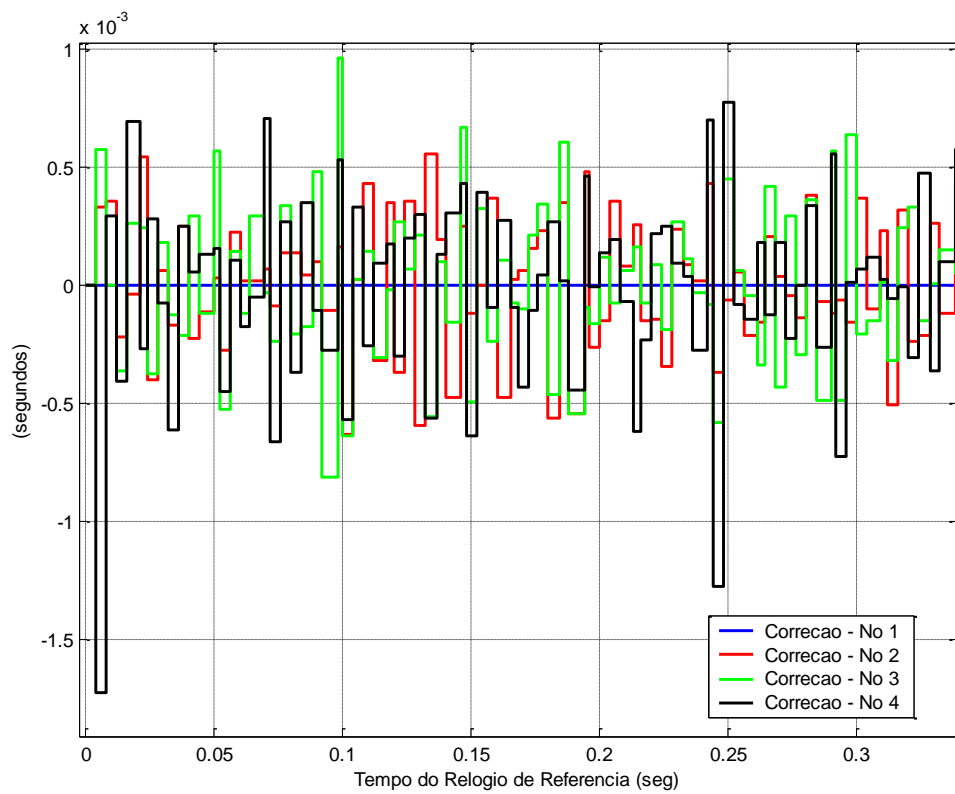


Figura H.2 – Aproximação: Correção do Filtro de Kalman, Viés: -0.2 segundos.

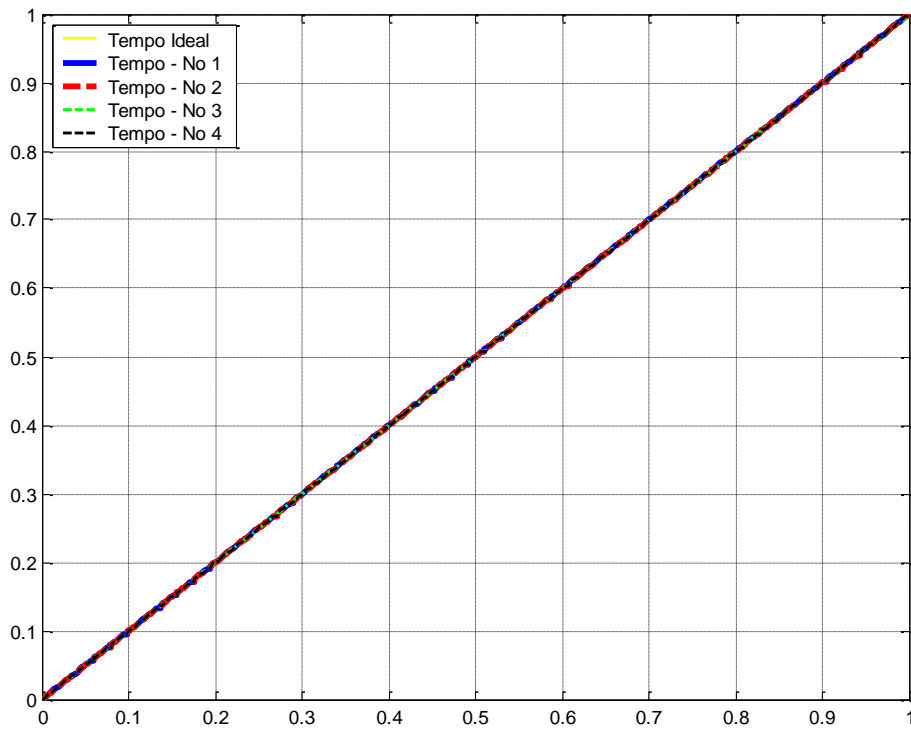


Figura H.3 – Linha do Tempo Sincronizada - Viés inicial: -0.2 segundos.

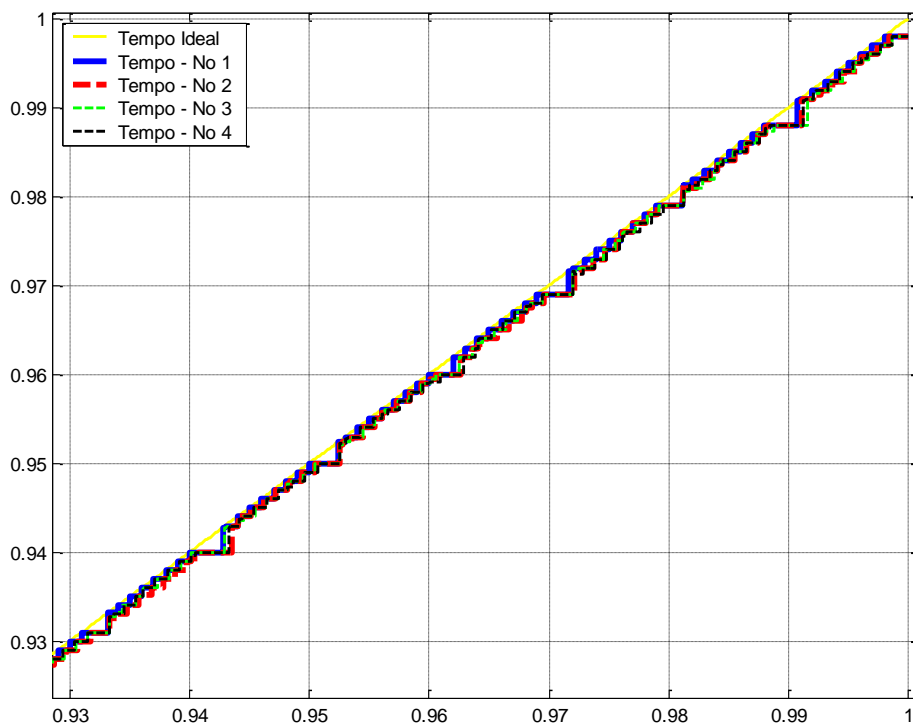


Figura H.4 – Aproximação Linha do Tempo Sincronizada - Viés inicial: -0.2 segundos.

As Figuras H.1 e H.2 mostram que o Filtro de Kalman convergiu. O nó não faz correção pois é a referência, os nós 2 e 3, que não estavam com erro de viés inicial, fizeram uma correção periódica com valor médio de zero. O nó 4, com viés inicial -0.2 segundos, também faz uma correção periódica em torno de zero. Com isso, é possível analisar através do gráfico H.1 que a precisão dos relógios do sistema ficou em torno de 2 (ms).

As Figuras H.3 e H.4 mostram que os relógios dos nós se sincronizaram com o relógio de referência. A Figura H.5 mostra a resposta ao degrau do sistema de controle com sincronização com o uso do Filtro de Kalman. E as Figuras H.6 E H.7 mostram as leis de controle aplicadas aos nós.

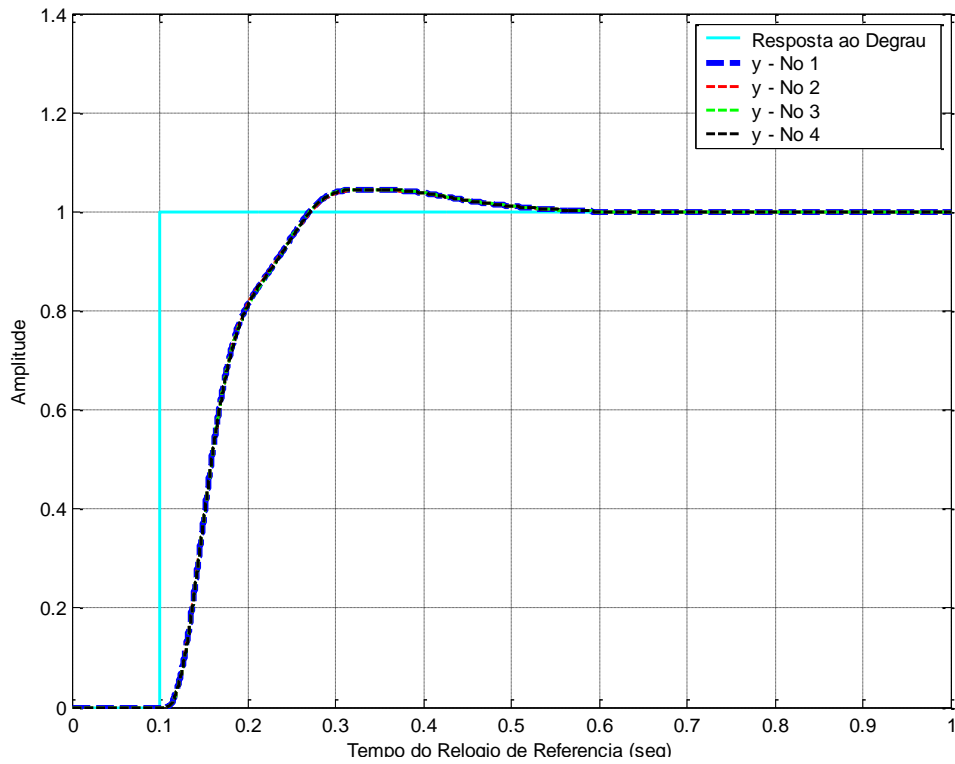


Figura H.5 – Resposta ao Degrau com viés inicial de -0.2 segundos - Sincronizado.

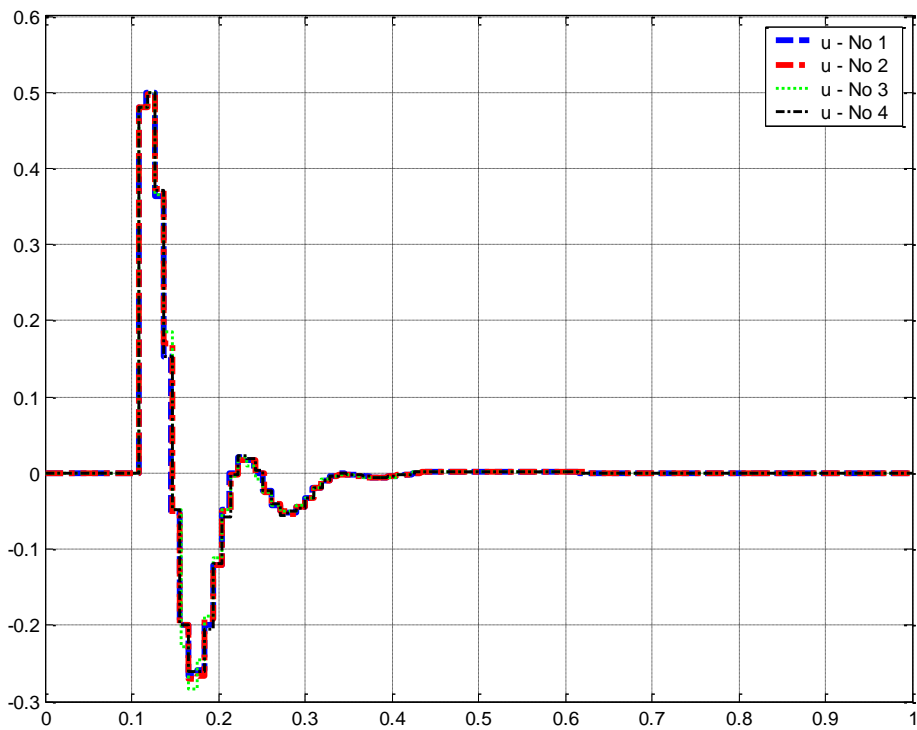


Figura H.6 – Lei de Controle com viés inicial de -0.2 segundos - Sincronizado.

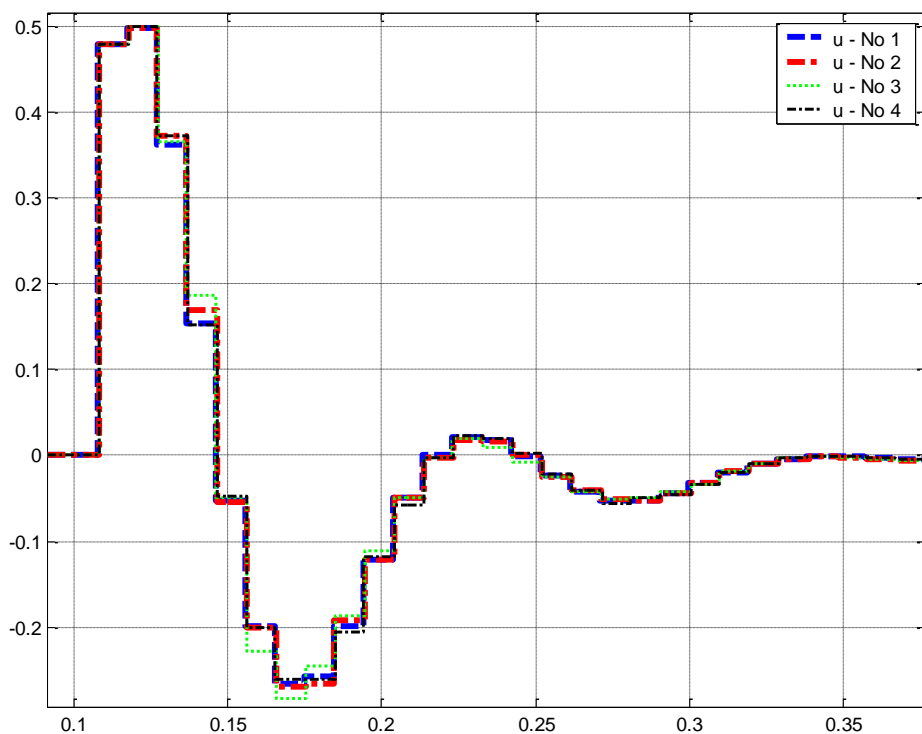


Figura H.7 – Aproximação Lei de Controle com viés inicial de -0.2 segundos - Sincronizado.

Através das Figuras H.5, H.6 e H.7 é possível ver que a sincronização de relógios com o Filtro de Kalman praticamente anulou o efeito do atraso do relógio do nó 4. Neste caso o algoritmo de Filtro de Kalman mostrou-se eficiente corrigindo o erro de viés inicial presente no nó 4 sistema. Com essa correção de relógios o algoritmo de sincronização com Filtro de Kalman eliminou o atraso sobre a lei de controle do nó 4 causado pelo viés inicial do relógio.

No **caso B**, com um **viés inicial de 0.39 segundos**, o relógio do nó 4 está adiantado em relação ao relógio de referência e aos outros relógios do conjunto, como mostra a Figura E.8. Com isso, o sistema de controle do nó 4 sofre uma pequena variação na % de sobre-elevação, causado pelo erro de relógio.

A Figura E.9 mostra a resposta ao degrau do sistema de controle com o viés inicial de 0.39 segundos. É possível ver que somente o nó 4 sofre variação em relação ao de referência e aos outros nós do sistema. O nó 4 adianta sua lei de controle em 0.39 segundos. Isto ocorre pois o relógio do nó 4 inicia adiantado em relação aos outros nós e ao nó de referência. A Figura E.10 mostra as leis de controle aplicadas a cada nó do sistema e do sistema de referência. Somente o nó 4 difere sua resposta ao degrau e a lei de controle do nó 1, nó de referência, e dos outros nós do sistema. O sistema de controle do nó 4 sofre o impacto do adiantamento. Este impacto é um aumento na % de sobre-elevação. Para tentar corrigir este problema vamos aplicar Filtro de Kalman para a sincronização de relógios.

Aplicando o Filtro de Kalman na sincronização de relógios. As Figuras H.10 e H.11 mostram a linha do tempo sincronizada com o Filtro de Kalman aplicado ao sistema. O período de re-sincronização é de 0.0040 segundos, isto é, a cada 0.0040 segundos o Filtro de Kalman atualiza os valores de relógio com a correção calculada. Nas Figuras H.8 e H.9 é possível ver o gráfico de correção dos relógios dos respectivos nós do sistema.

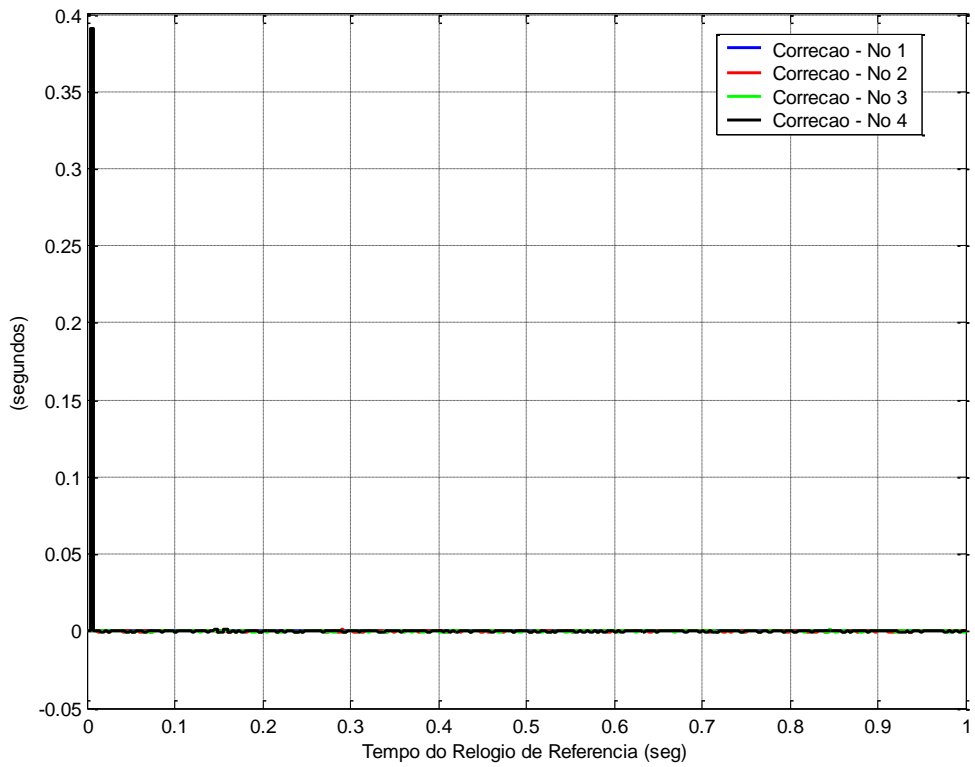


Figura H.8 – Correção do Filtro de Kalman, Viés: 0.39 segundos.

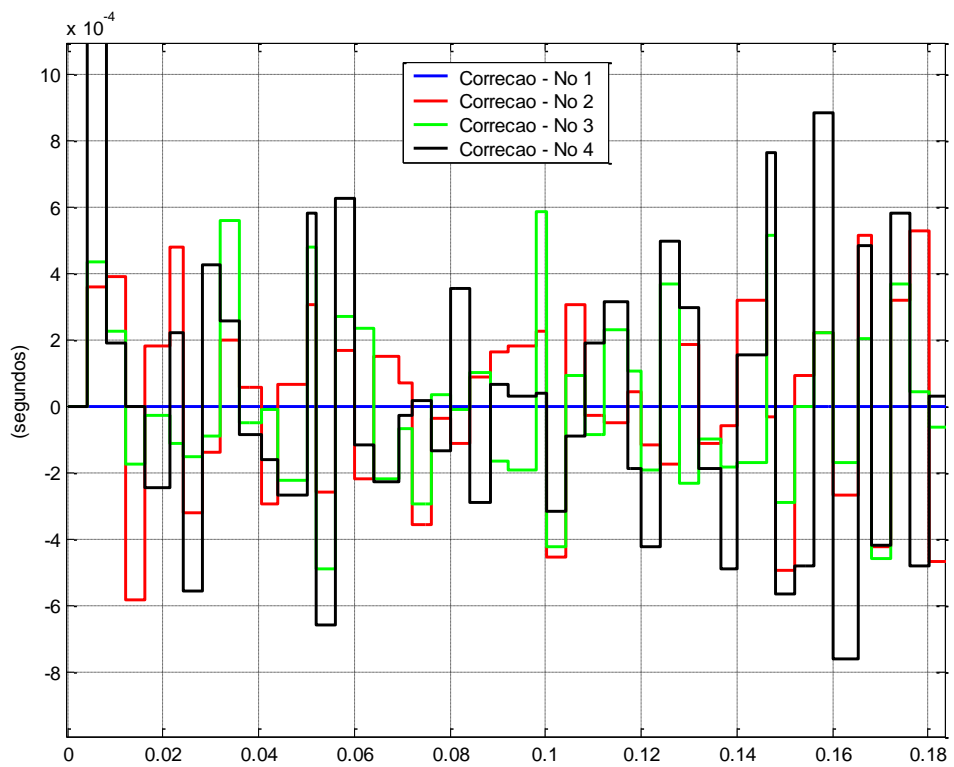


Figura H.9 – Aproximação: Correção do Filtro de Kalman, Viés: 0.39 segundos.

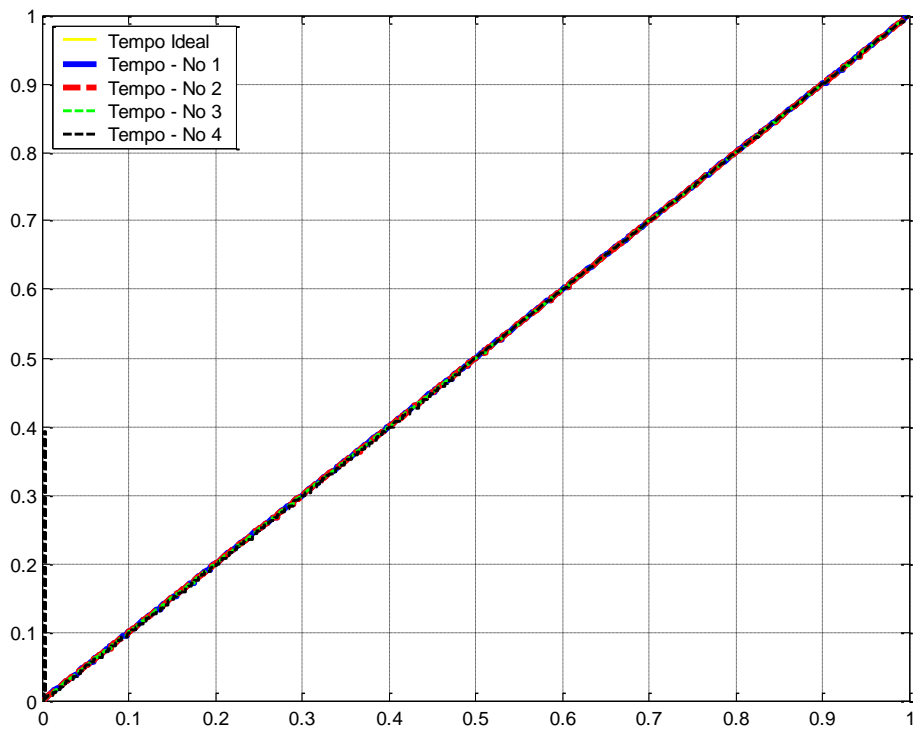


Figura H.10 – Linha do Tempo Sincronizada - Viés inicial: 0.39 segundos.

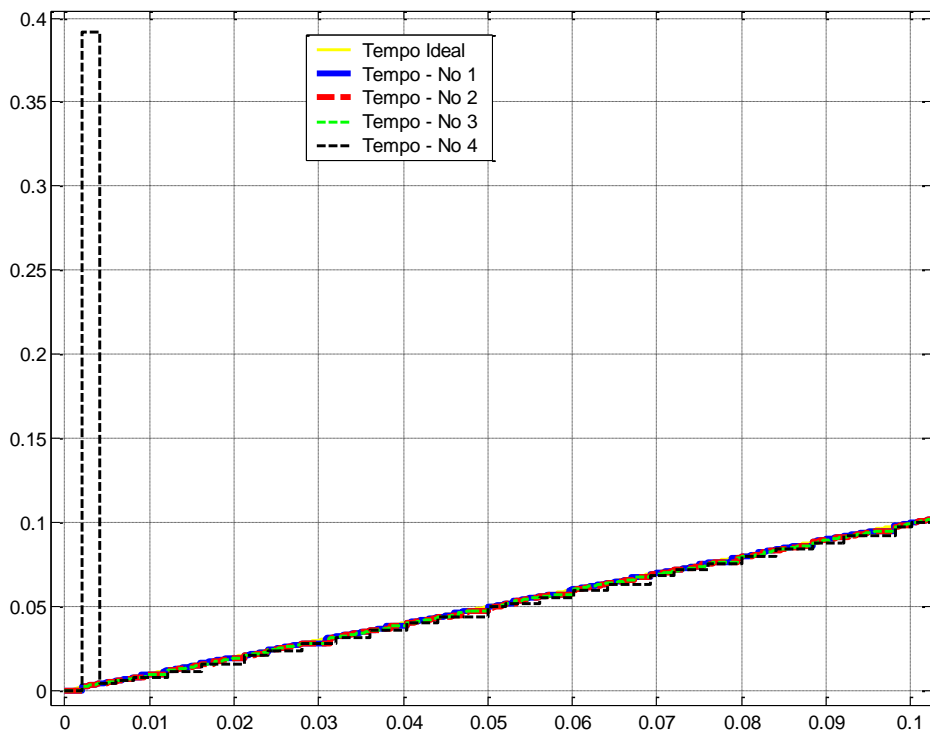


Figura H.11 – Aproximação: Linha do Tempo Sincronizada - Viés inicial: 0.39 segundos.

As Figuras H.8 e H.9 mostram que o Filtro de Kalman convergiu. O nó 1 não faz correção pois é a referência, os nós 2 e 3, que não estavam com erro de viés inicial, fizeram uma correção periódica com valor médio de zero. O nó 4, com viés inicial de 0.39 segundos, também faz uma correção periódica em torno de zero. É interessante observar na Figuras H.8, que existe um pico bem grande que é o pico que corrige o viés do relógio 4, após isso o Filtro de Kalman converge. Com isso, é possível analisar através do gráfico H.9 que a precisão dos relógios, após a convergência, do sistema ficou em torno de 2 (ms). As Figuras H.10 e H.11 mostram que os relógios dos nós se sincronizaram com o relógio de referência. A Figura H.12 mostra a resposta ao degrau do sistema de controle com sincronização com o uso do Filtro de Kalman.

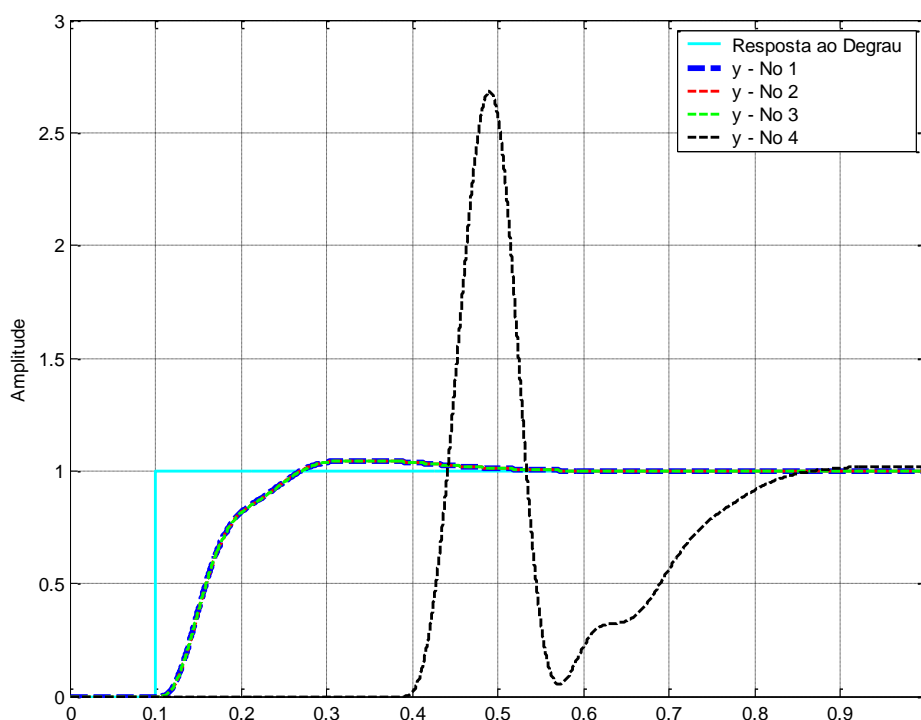


Figura H.12 – Resposta ao Degrau com viés inicial de 0.39 segundos - Sincronizado.

E a Figura H.13 mostra as leis de controle aplicadas aos nós.



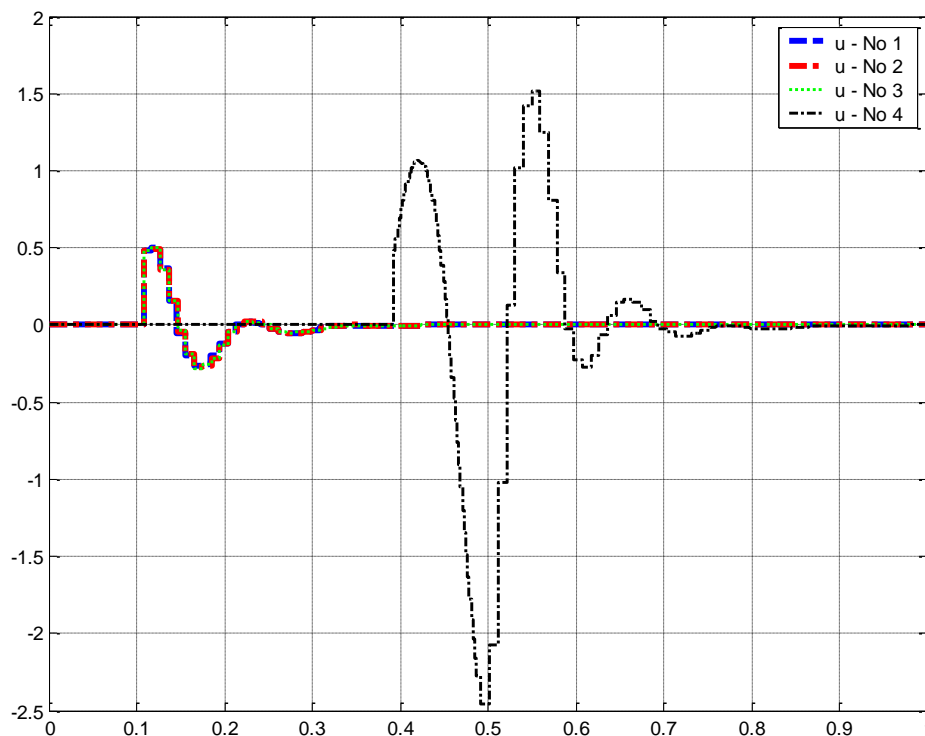


Figura H.13 – Lei de Controle com viés inicial de 0.39 segundos – Sincronizado.

Através das Figuras H.12 e H.13 é possível ver que a sincronização de relógios com o Filtro de Kalman amplificou o efeito do erro do relógio do nó 4. O Filtro de Kalman causou um atraso ao sistema e a resposta ao degrau demorou mais tempo para corrigir. O sinal de controle do nó 4 ficou atrasado. Neste caso o algoritmo de Filtro de Kalman não mostrou-se eficiente corrigindo o erro de viés inicial presente no nó 4 sistema. Com essa correção de relógios o algoritmo de sincronização com Filtro de Kalman eliminou o adiantamento, mas em contrapartida inseriu um atraso sobre a lei de controle do nó 4 causado pelo viés inicial do relógio.



## APENDICE I – SIMULAÇÃO DO 6º CASO – ARQUITETURA FEDERADA – ALGORITMO FK

Neste caso, foi feita a mesma simulação de um sistema de controle do 3º Caso, com a diferença que a sincronização de relógios agora é feita através do Filtro de Kalman, com arquitetura centralizada, isto é, existe um relógio de referência. O relógio escolhido como relógio correto é o relógio do nó 1. Nesta simulação variou-se o parâmetro de viés inicial e deriva do relógio do nó (subsistema) 4. A rede de comunicação foi feita por um protocolo de comunicação com acesso ao meio TDMA.

Este trabalho escolheu 2 valores deriva e de viés inicial diferentes para variação do relógio do nó 4:

- **Caso A: Deriva: -10%; Viés: -0.2 (seg)** - O relógio inicia atrasado de 0.2 segundos em relação ao relógio de referência e esta com uma taxa de deriva negativa, isto é, o relógio está se atrasando em relação ao relógio de referência;
- **Caso B: Deriva: 10%; Viés: 0.39 (seg)** - O relógio inicia adiantado de 0.39 segundos em relação ao relógio de referência e esta com uma taxa de deriva positiva, isto é, o relógio está se adiantando em relação ao relógio de referência;

Foram escolhidos estes dois valores de viés inicial, pois são os valores dos casos anteriores.

No **caso A**, com um **viés inicial de -0.2 segundos e taxa de deriva de -10%**, o relógio do nó 4 inicia atrasado e além disso o relógio está se atrasando em relação ao relógio de referência e aos outros relógios do conjunto, como mostra a Figura F.1. Com isso, o sistema de controle do nó 4 que esta afetado pelo relógio sofre uma pequena variação. A Figura F.2 mostra a resposta ao degrau

do sistema de controle com o viés inicial de -0.2 segundos e taxa de deriva de -10%. É possível ver que somente o nó 4 sofre variação em relação ao de referência e aos outros nós do sistema. O nó 4 atrasa sua lei de controle em mais de 0.2 segundos. Isto ocorre pois o relógio do nó 4 inicia atrasado e com deriva negativa em relação ao de referência. A Figura F.3 mostra as leis de controle aplicadas a cada nó do sistema e do nó de referência. Somente o nó 4 difere sua resposta ao degrau e a lei de controle do nó 1, de referência, e dos outros nós do sistema. O sistema de controle do nó 4 sofre o impacto do atraso e da deriva, para tentar corrigir este problema vamos aplicar Filtro de Kalman para a sincronização de relógios. Aplicando o Filtro de Kalman na sincronização de relógios. A Figura I.3 mostra a linha do tempo sincronizada com o Filtro de Kalman aplicado ao sistema. O período de re-sincronização é de 0.0040 segundos, isto é, a cada 0.0040 segundos o Filtro de Kalman atualiza os valores de relógio com a correção calculada. Nas Figuras I.1 e I.2 é possível ver o gráfico de correção dos relógios dos respectivos nós do sistema.

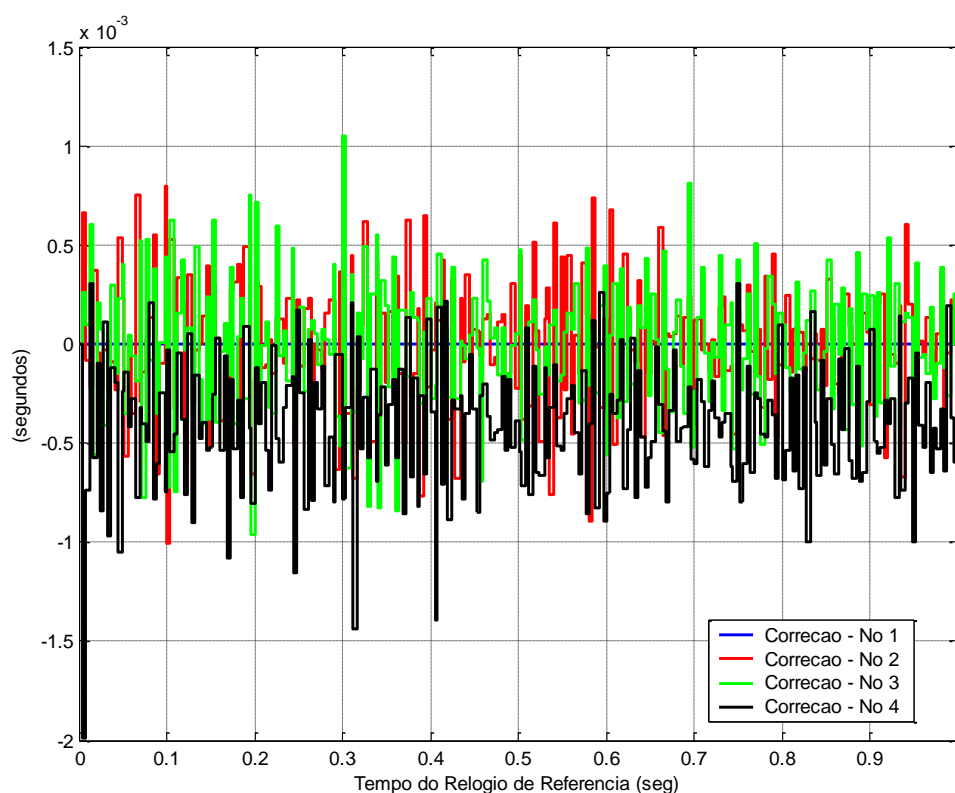


Figura I.1 – Correção com Filtro de Kalman, Viés: -0.2 segundos e Deriva: -10%.

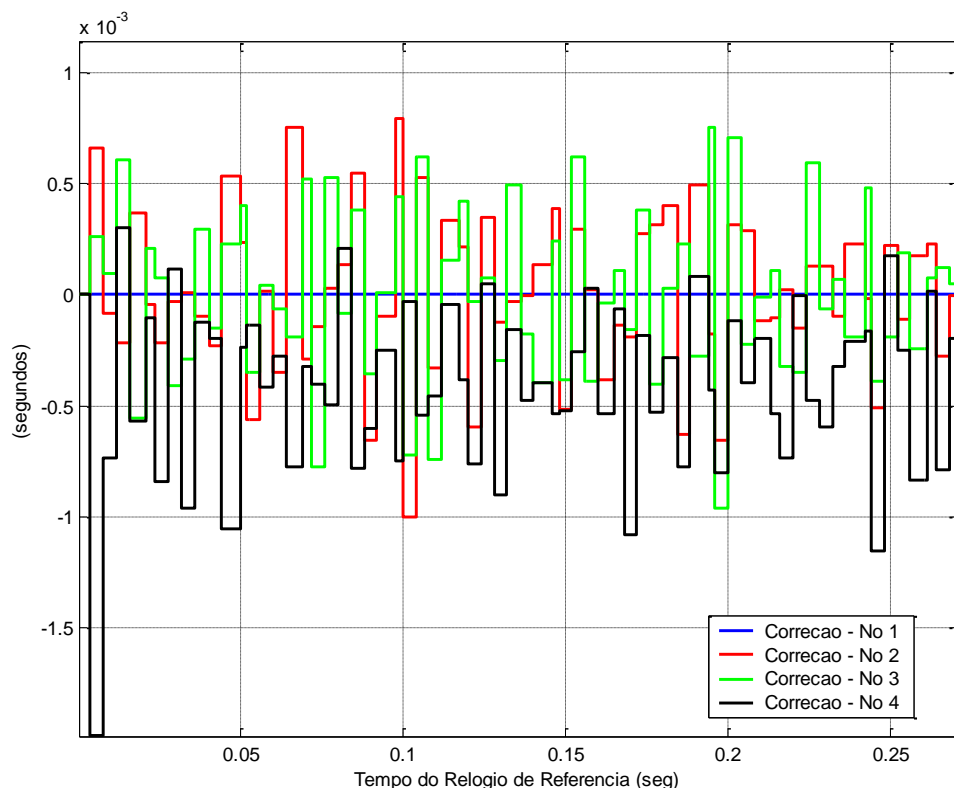
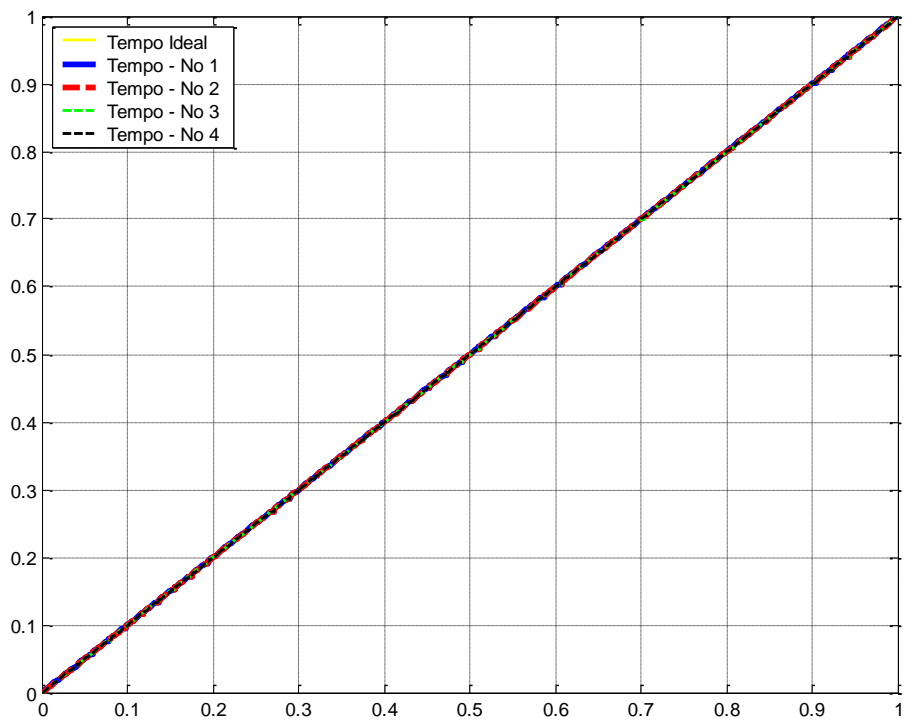
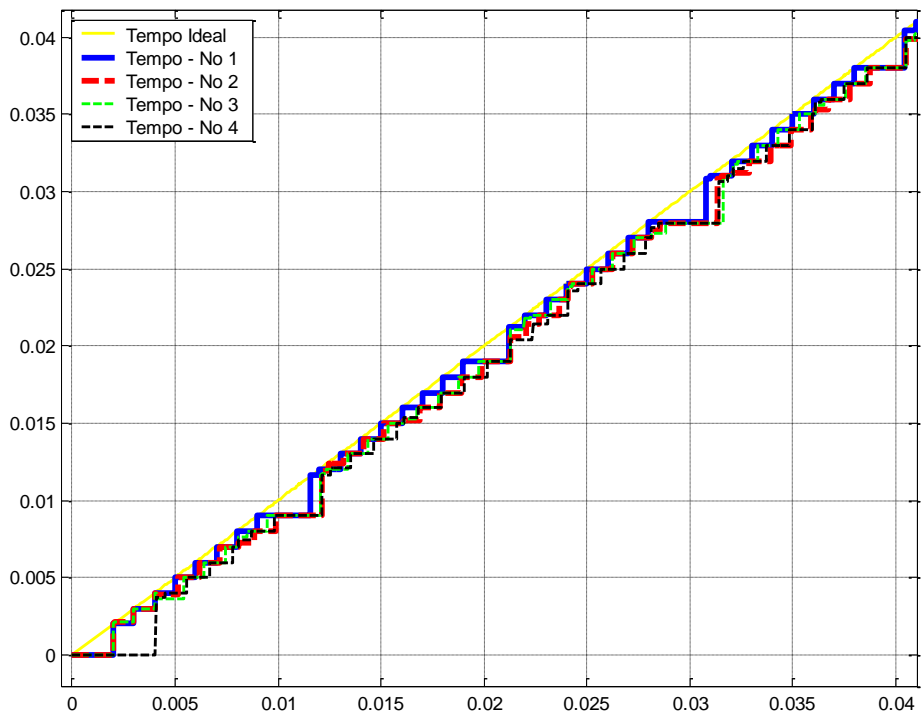


Figura I.2 –Aproximação - Correção com Filtro de Kalman, Viés: -0.2 segundos e Deriva: -10%.

As Figuras I.1 e I.2 mostram que o Filtro de Kalman convergiu. O nó não faz correção pois é a referência, os nós 2 e 3, que não estavam com erro de viés inicial, fizeram uma correção periódica com valor médio de zero. O nó 4, com viés inicial -0.2 segundos e deriva de -10%, faz uma correção periódica em com valor médio em torno de -0.5 (ms). Com isso, é possível analisar através do gráfico I.1 que a precisão dos relógios do sistema ficou em torno de 2 (ms). A Figura I.3 mostra que os relógios dos nós se sincronizaram com o relógio de referência.



(a)



(b)

Figura I.3 – (a) Linha do Tempo Sincronizada - Viés inicial: -0.2 segundos e Taxa de Deriva: -10%. (b) Aproximação.

A Figura I.4 mostra a resposta ao degrau do sistema de controle com sincronização com o uso do Filtro de Kalman.

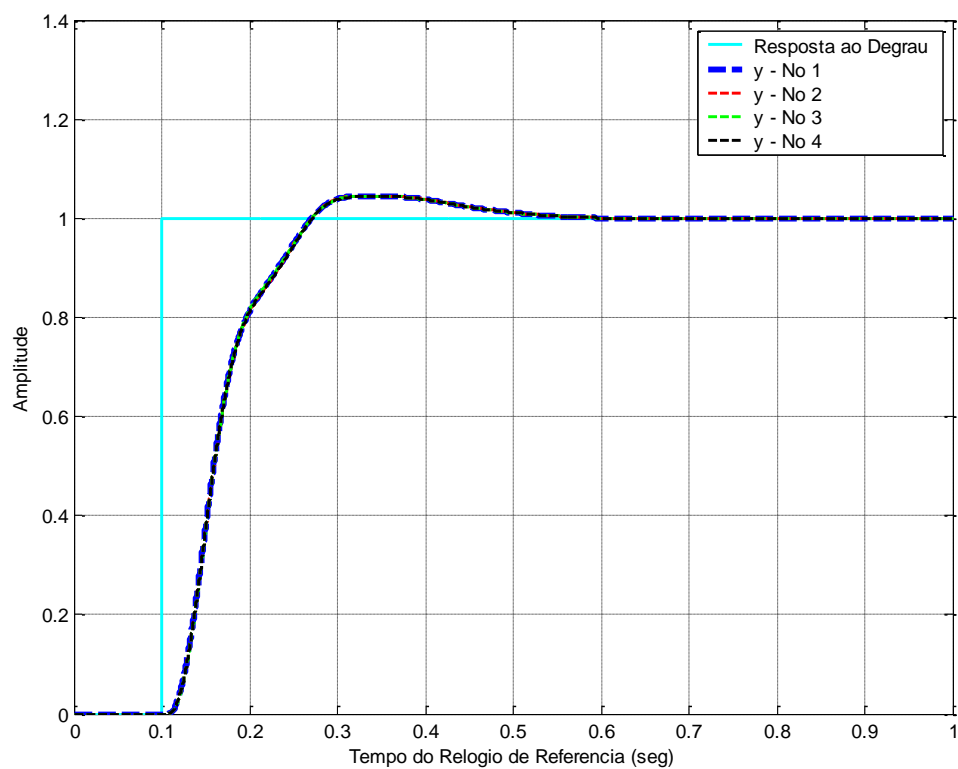
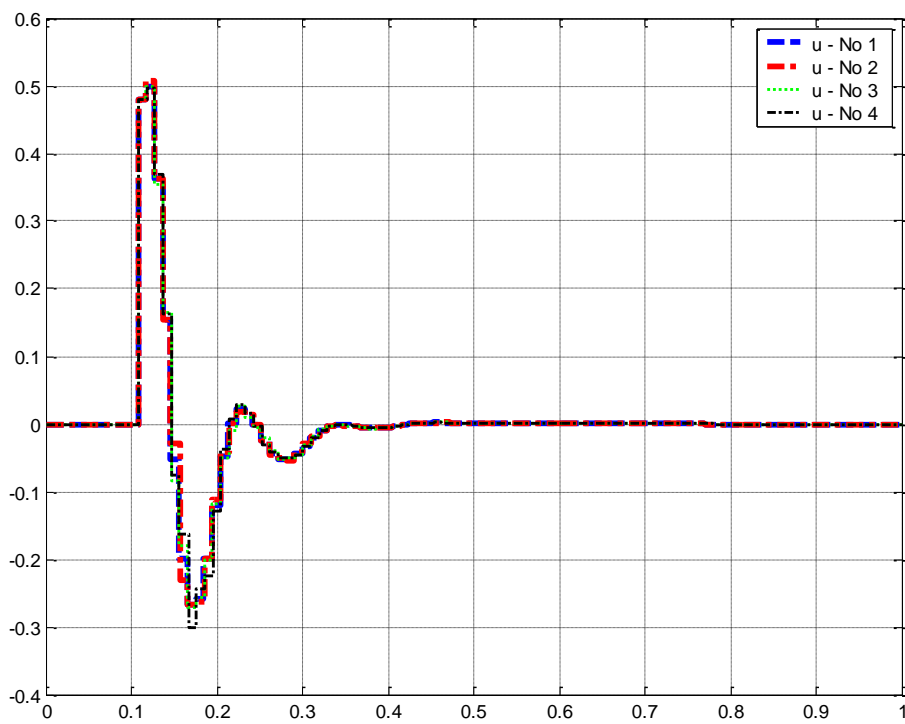
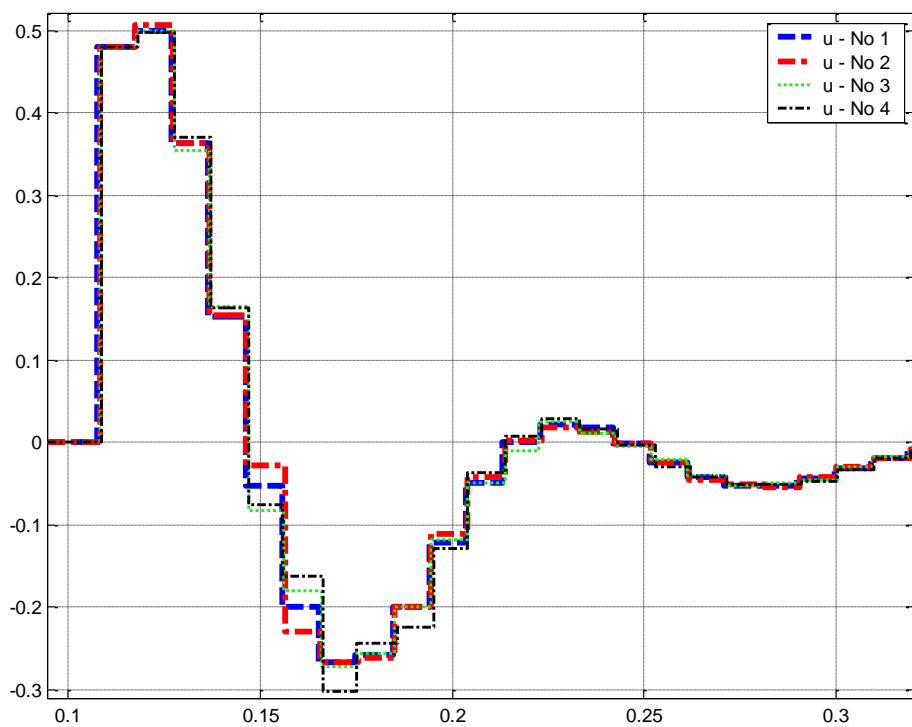


Figura I.4 – Resposta ao Degrau com viés inicial de -0.2 segundos - Sincronizado.

E a Figura I.5 mostra as leis de controle aplicadas aos nós.



(a)



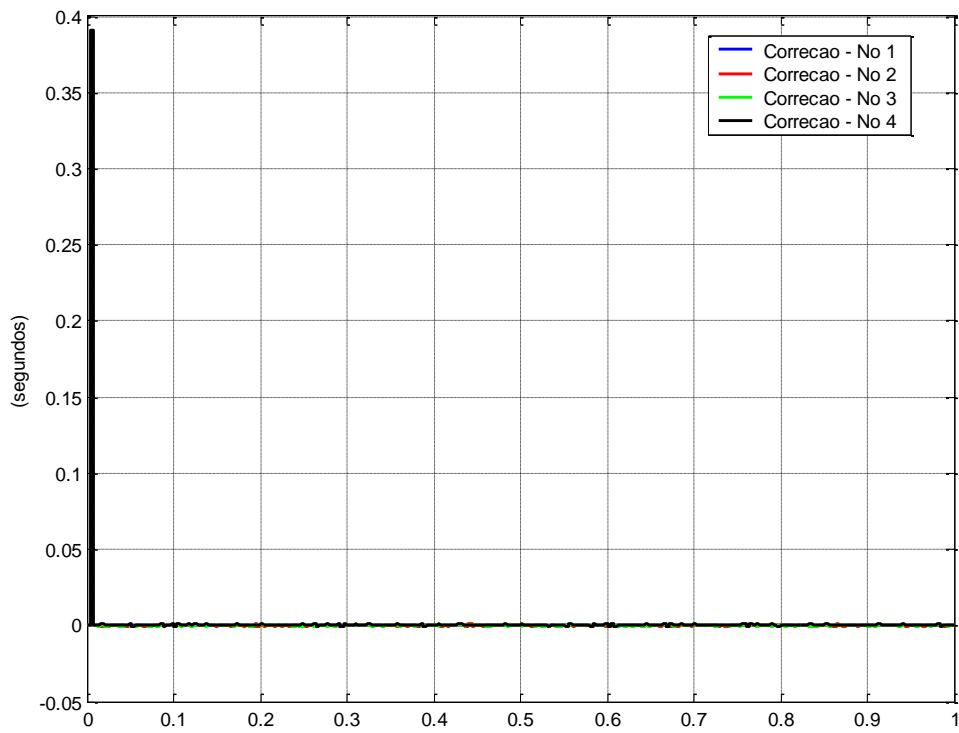
(b)

Figura I.5 – (a) Lei de Controle: Viés inicial de -0.2 segundos e Taxa de Deriva: -10% - Sincronizado. (b) Aproximação.

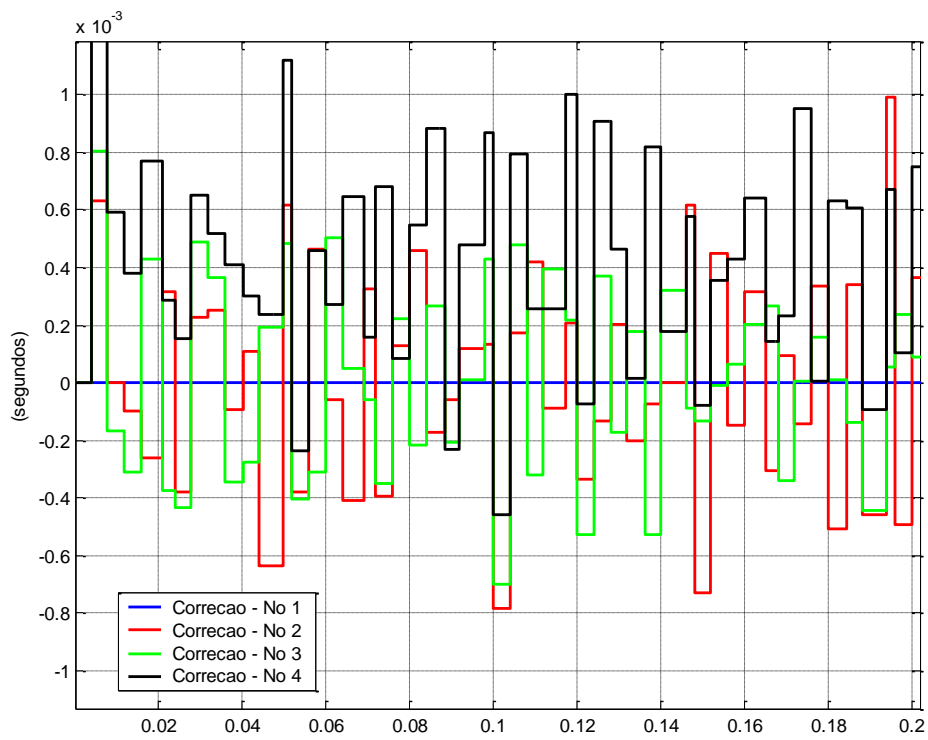


Através das Figuras I.4 e I.5 é possível ver que a sincronização de relógios com o Filtro de Kalman praticamente anulou o efeito do atraso e da deriva do relógio do nó 4. Neste caso o algoritmo de Filtro de Kalman mostrou-se eficiente corrigindo o erro de viés inicial e a taxa de deriva do relógio presente no nó 4 do sistema. Com essa correção de relógios o algoritmo de sincronização com Filtro de Kalman eliminou o atraso sobre a lei de controle do nó 4 causado pelo viés inicial e da deriva negativa do relógio 4.

No **caso B**, com um **viés inicial de 0.39 segundos e taxa de deriva de 10%**, o relógio do nó 4 inicia adiantado e além disso o relógio está se adiantando em relação ao relógio de referência e aos outros relógios do conjunto, como mostra a Figura F.8. Com isso, o sistema de controle do nó 4 que está afetado pelo relógio sofre uma pequena variação. A Figura F.9 mostra a resposta ao degrau do sistema de controle com o viés inicial de 0.39 segundos e taxa de deriva de 10%. É possível ver que somente o nó 4 sofre variação em relação ao de referência e aos outros nós do sistema. A uma aumento na % de sobre elevação da resposta ao degrau do nó 4. Isto ocorre pois o relógio do nó 4 inicia adiantado e com deriva positiva em relação ao relógio de referência. A Figura F.10 mostra as leis de controle aplicadas a cada nó do sistema e do nó de referência. Somente o nó 4 difere sua resposta ao degrau e a lei de controle do nó 1, de referência, e dos outros nós do sistema. O sistema de controle do nó 4 sofre o impacto do adiantamento e da deriva. Para tentar corrigir este problema vamos aplicar Filtro de Kalman para a sincronização de relógios. Aplicando o Filtro de Kalman na sincronização de relógios. A Figura I.7 mostra a linha do tempo sincronizada com o Filtro de Kalman aplicado ao sistema. O período de re-sincronização é de 0.0040 segundos, isto é, a cada 0.0040 segundos o Filtro de Kalman atualiza os valores de relógio com a correção calculada. Na Figura I.6 é possível ver o gráfico de correção dos relógios dos respectivos nós do sistema.



(a)



(b)

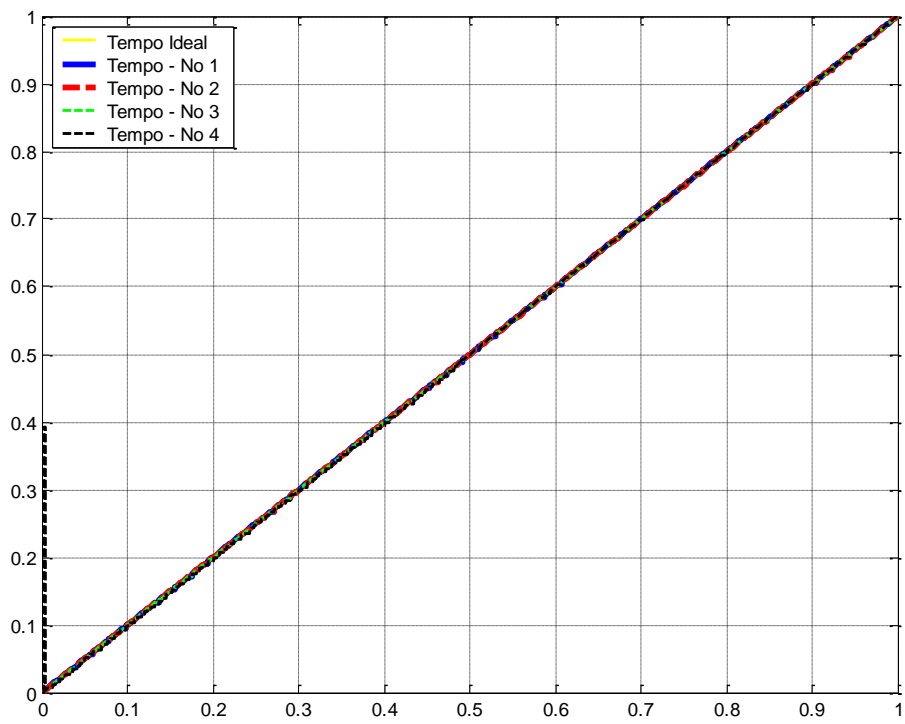
Figura I.6 –Aproximação - Correção com Filtro de Kalman, Viés: 0.39 segundos e

Deriva: 10%.

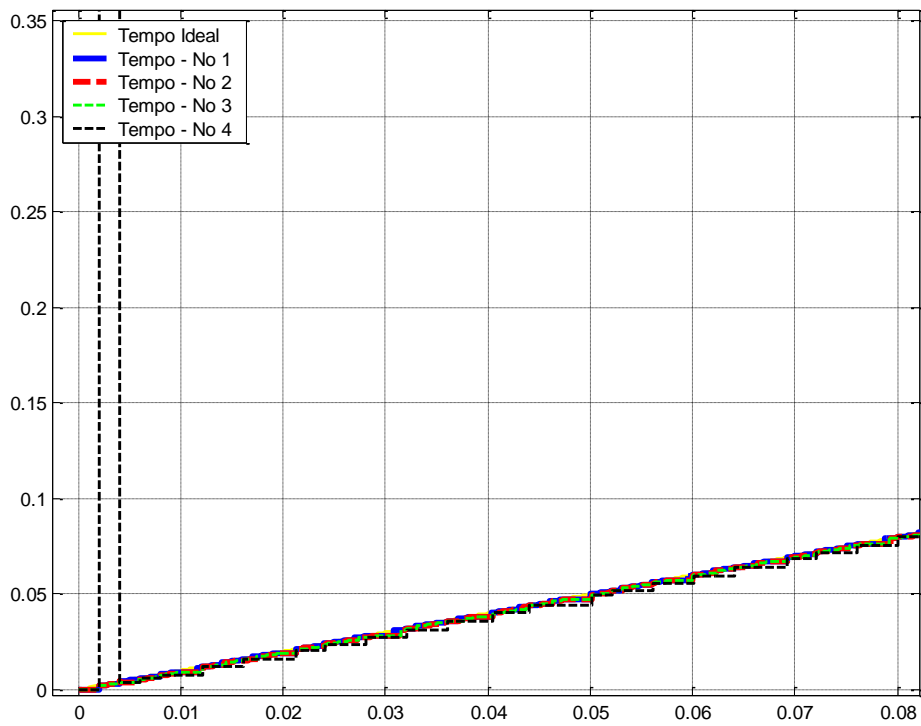
A Figura I.6 mostra que o Filtro de Kalman convergiu. O nó 1 não faz correção pois é a referência, os nós 2 e 3, que não estavam com erro de viés inicial e deriva, fizeram uma correção periódica com valor médio de zero. O nó 4, com viés inicial 0.39 segundos e deriva de 10%, faz uma correção periódica em com valor médio em torno de 0.5 (ms). Com isso, é possível analisar através do gráfico I.6 que a precisão final dos relógios do sistema ficou em torno de 2 (ms).

A Figura I.7 mostra que os relógios dos nós se sincronizaram com o relógio de referência.

A Figura I.8 mostra a resposta ao degrau do sistema de controle com sincronização com o uso do Filtro de Kalman.



(a)



(b)

Figura I.7 – (a) Linha do Tempo Sincronizada - Viés inicial: 0.39 segundos e Taxa de Deriva: 10%. (b) Aproximação.

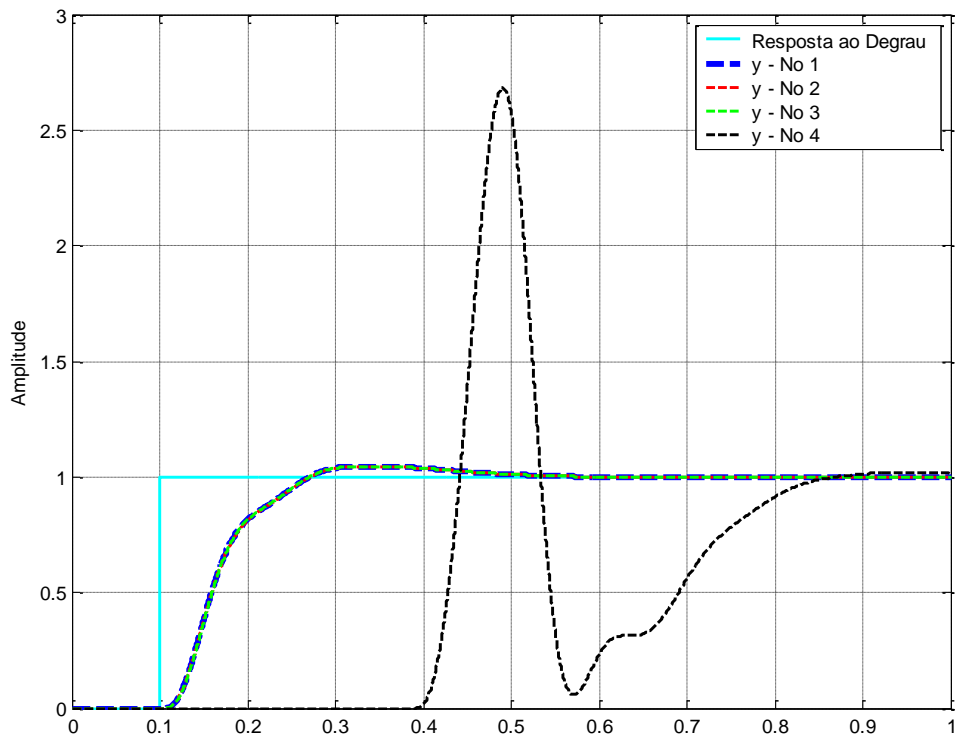


Figura I.8 – Resposta ao Degrau com viés inicial de 0.39 segundos e Taxa de Deriva: 10% - Sincronizado.

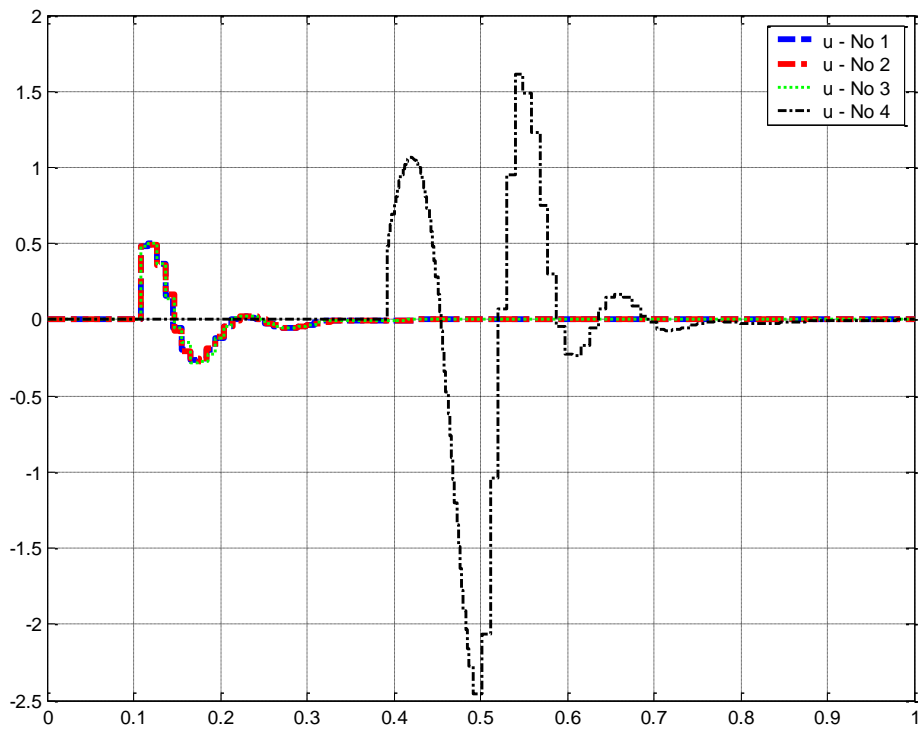


Figura I.9 – Lei de Controle: Viés inicial de 0.39(seg) e Deriva:10% - Sincronizado.

Através das Figuras 1.8 e 1.9 é possível ver que a sincronização de relógios aumentou o erro que o viés inicial e a deriva do relógio 4 causaram à resposta ao degrau e a lei de controle do nó 4. O Filtro de Kalman causou um atraso ao sistema e a resposta ao degrau demorou mais tempo para corrigir. O sinal de controle do nó 4 ficou atrasado. Neste caso o algoritmo de Filtro de Kalman não mostrou-se eficiente corrigindo o erro de viés inicial e deriva presente no nó 4 sistema. Com essa correção de relógios o algoritmo de sincronização com Filtro de Kalman eliminou o adiantamento, mas em contrapartida inseriu um atraso sobre a lei de controle do nó 4 causado pelo viés inicial do relógio.

Os relógios dos nós 1, 2 e 3 estão sincronizados entre si e seus sistemas de controle não foram afetados. Neste caso o algoritmo mostrou-se ineficiente ao corrigir o erro de viés inicial e deriva do relógio 4, amplificando o efeito negativo na lei de controle.

## APÊNDICE J – SIMULAÇÃO DO 1º CASO – ARQUITETURA INTEGRADA – REDE TDMA

Neste caso, foi feita uma simulação em um sistema de controle com arquitetura integrada, descrito na seção 3.3. A sincronização foi feita através do algoritmo FTM (*Welch-Lynch*), com arquitetura distribuída, isto é, não existe um relógio de referência. Nesta simulação variou-se o parâmetro de deriva do relógio do nó 2 (Sensor da Malha 1 de Controle por Rede). A rede de comunicação foi feita por um protocolo de comunicação com acesso ao meio TDMA.

Este trabalho escolheu 3 taxas de deriva diferentes para variação do relógio:

- **Caso A: -10%** - O relógio do sensor 1 está se atrasando em relação aos outros com taxa de 10%;
- **Caso B: 1%** - O relógio do sensor 1 está se adiantando em relação aos outros com taxa de 1%;
- **Caso C: 10%** - O relógio do sensor 1 está se adiantando em relação aos outros com taxa de 10%;

No **caso A**, a uma **taxa de deriva de -10%**, o relógio do nó 2, sensor 1, está se atrasando em relação ao relógio ideal e aos outros relógios do conjunto, como mostra a Figura J.1. Com isso, o sistema de controle 1 sofre uma variação em sua resposta ao degrau e na lei de controle, pois é afetado pelo relógio.

A Figura J.2 mostra a resposta ao degrau do sistema de controle com a deriva de -10% no sensor 2. É possível ver que essa variação no nó 2 prejudica o sistema de controle 1. Existe um aumento na % de sobre-elevação e no tempo de acomodação. Isto ocorre pois o relógio do sensor 2 está atrasado e com isso as suas medidas chegam atrasadas ao controle. A Figura J.3 mostra as leis de controle das 2 malhas integradas na rede de comunicação.

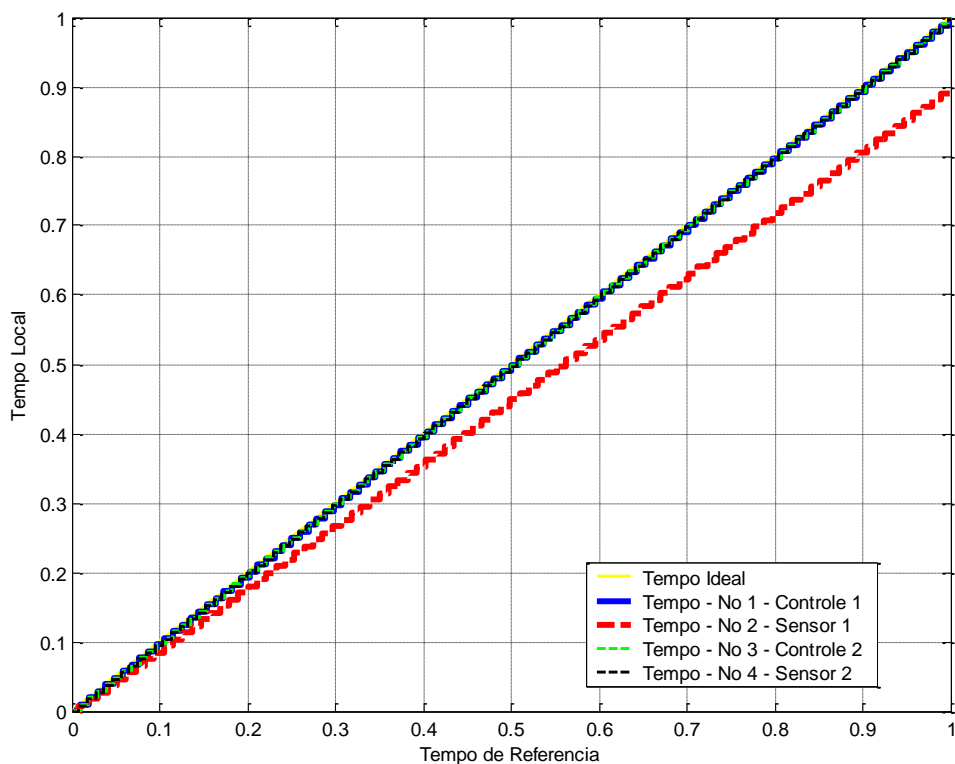
O sistema de controle por rede 1 (azul) sofre um impacto considerável com uma taxa de deriva de -10% no relógio do sensor 1.

Para reduzir este impacto, vamos aplicar o algoritmo FTM de sincronização de relógios. A Figura J.4 mostra a linha do tempo com o algoritmo de sincronização FTM aplicado ao sistema. O período de re-sincronização é de 0.0768 segundos (este valor foi escolhido, pois bate com o valor do final de cada TDMA Round), isto é, a cada 0.0768 segundos o algoritmo FTM atualiza os valores de relógio com a correção calculada. A Figura J.4 mostra que os relógios dos nós se sincronizaram entre si.

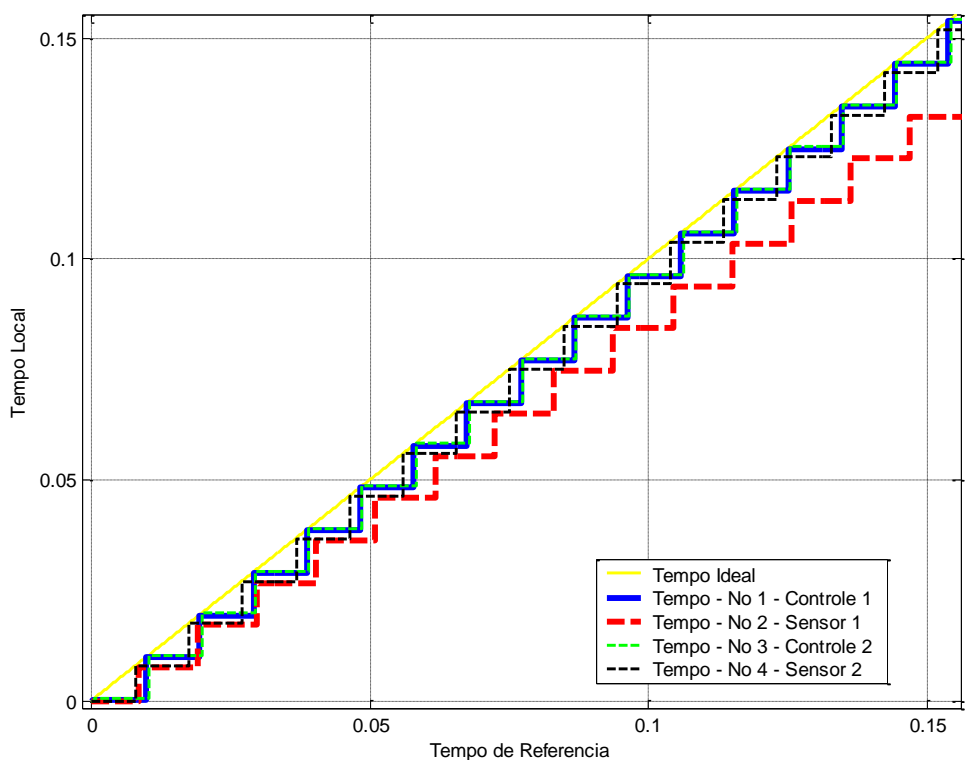
Na Figura J.5 é possível ver o gráfico de correção dos relógios dos respectivos nós do sistema. A Figura J.5 mostra que o algoritmo FTM convergiu. Os nós 1, 3 e 4 não estão com erro de deriva. O relógio 1 e 4 não corrigiram nada de seus relógios, o relógio 3 fez algumas pequenas correções. O nó 2 com -10% de taxa de deriva, teve uma correção periódica em entre -0.2 (ms) e -0.5 (ms). Com isso, é possível analisar através do gráfico 4.111 que a precisão dos relógios do sistema ficou em torno de 0.5 (ms).

A Figura J.6 mostra a resposta ao degrau do sistema de controle. E a Figura J.7 mostra as leis de controle do sistema.





(a)



(b)

Figura J.1 – (a) Linha do Tempo com deriva de -10%. (b) Aproximação.  
321

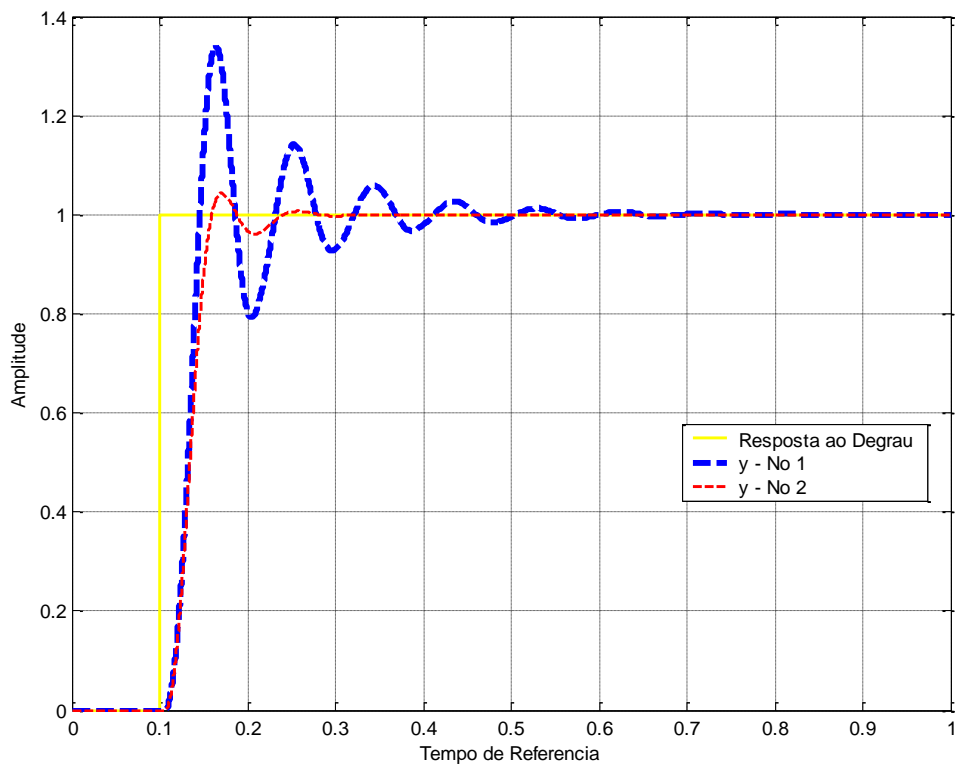


Figura J.2 – Resposta ao Degrau com deriva de -10%.

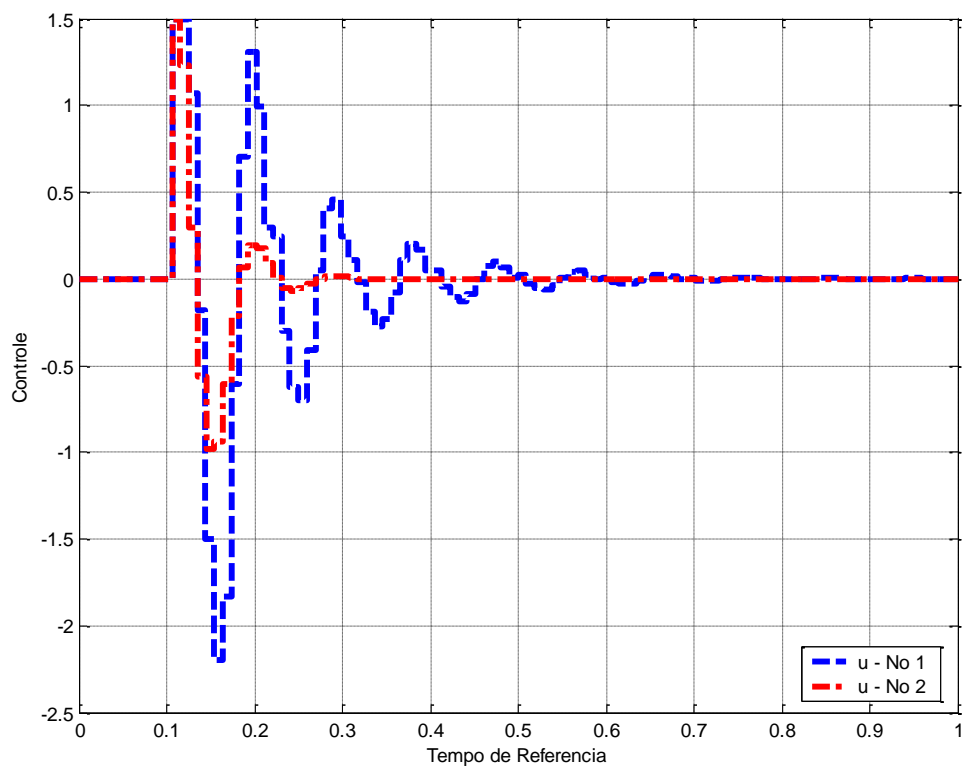
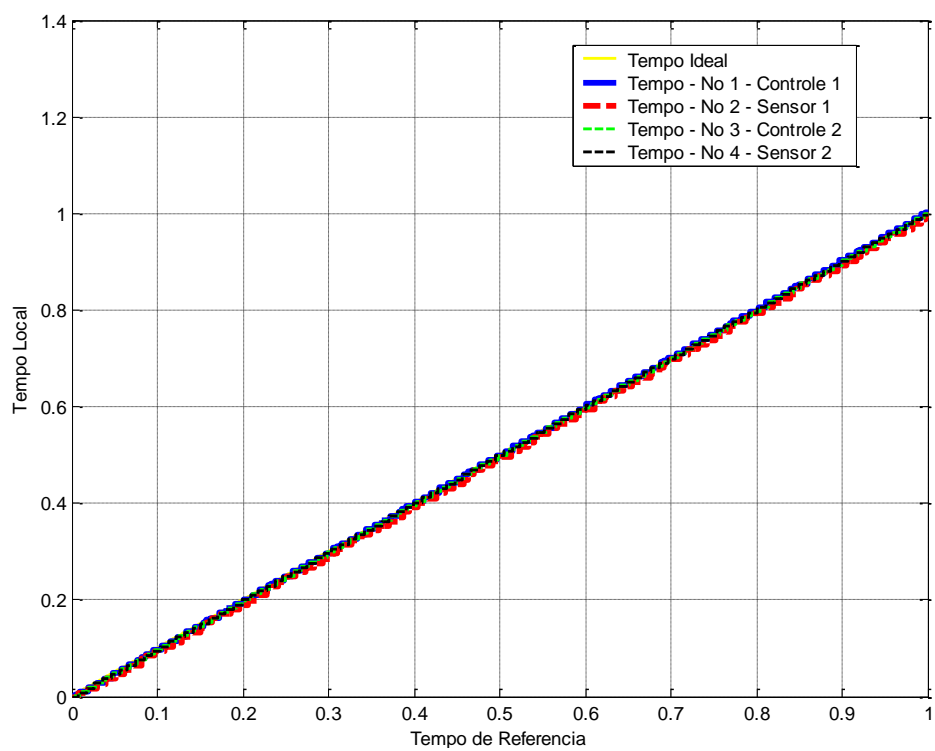
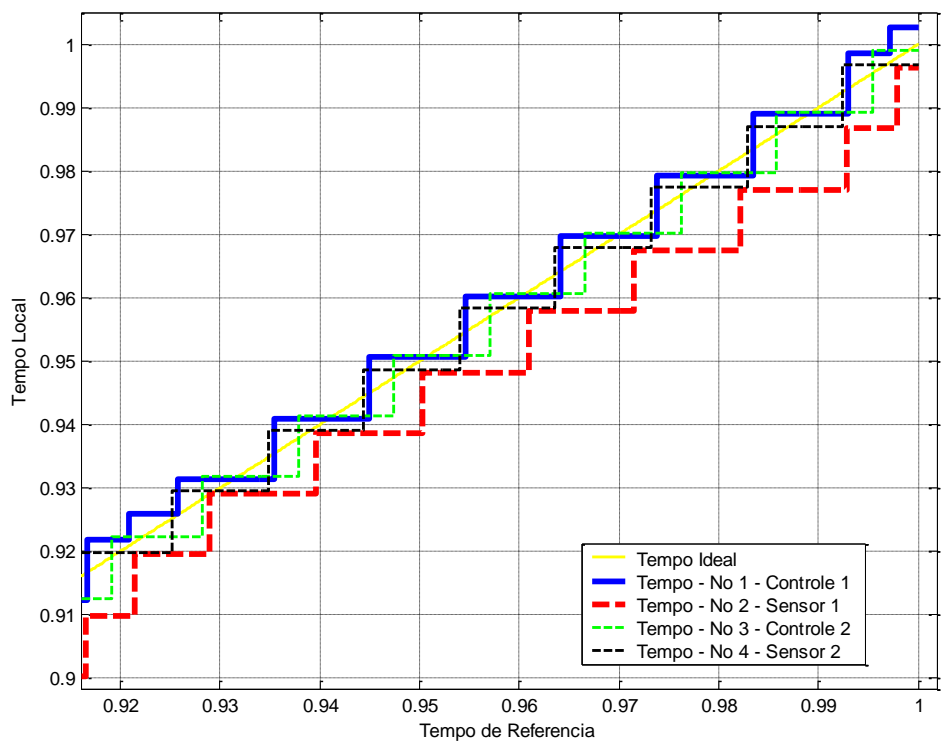


Figura J.3 – Lei de Controle com deriva de -10%.



(a)



(b)

Figura J.4 – (a) Linha do Tempo Sincronizada com deriva de -10%.(b) Aproximação.

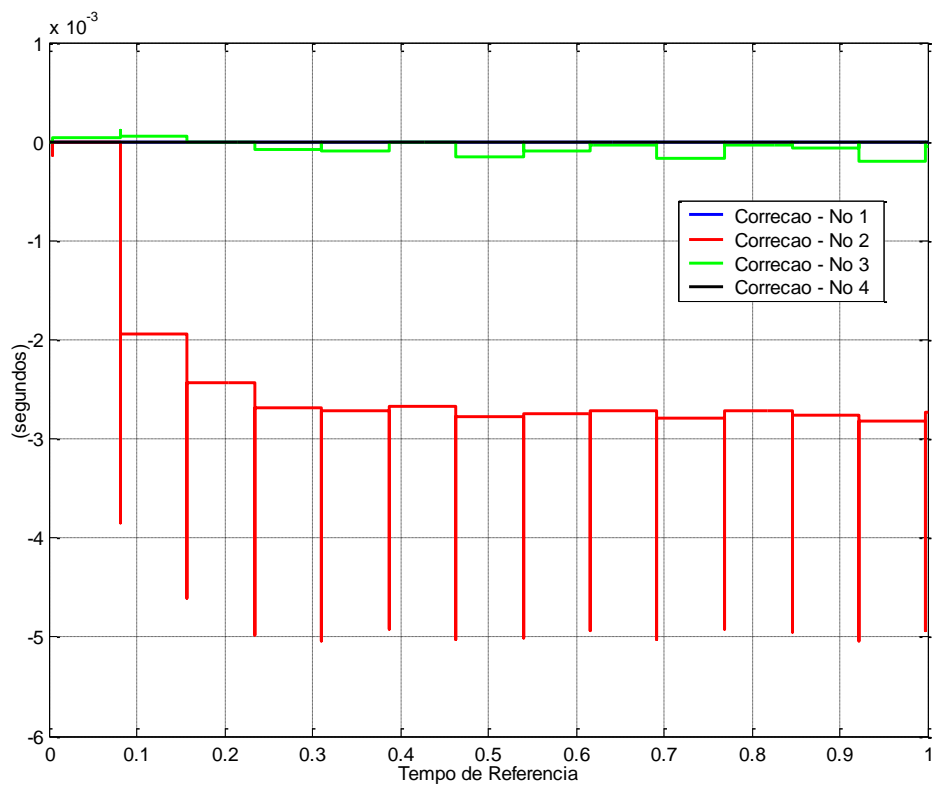


Figura J.5 – Correção Algoritmo FTM com deriva de -10%.

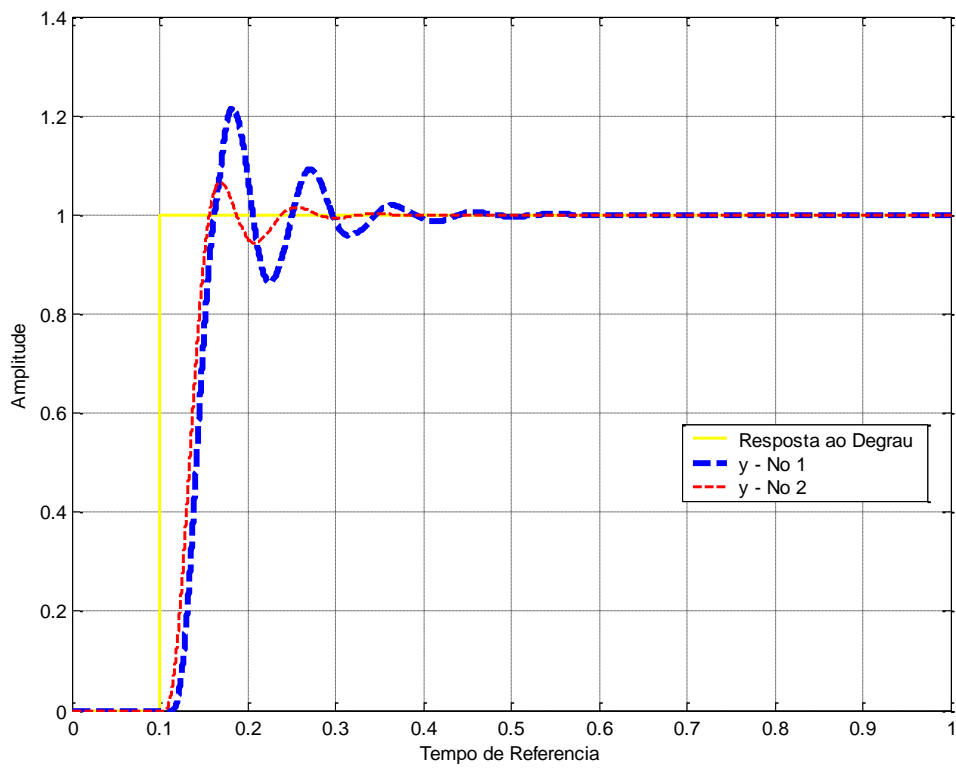


Figura J.6 – Resposta ao Degrau com deriva de -10% Sincronizado.

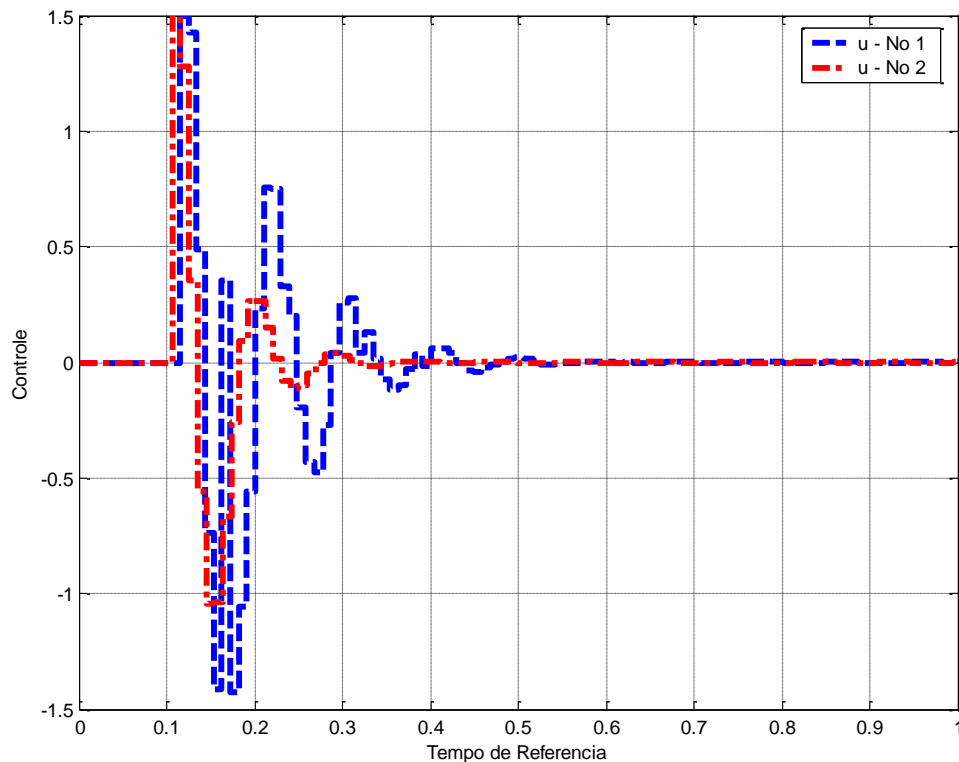


Figura J.7 – Lei de Controle com derivada de -10% - Sincronizado.

Através das Figuras J.6 e J.7 é possível ver que a sincronização de relógios minimizou a % de sobre elevação da resposta ao degrau e o tempo de acomodação. E minimizou o atraso da lei de controle. As respostas ao degrau dos dois sistemas de controle ainda desiguais. Neste caso o algoritmo não mostrou-se muito eficiente corrigindo o erro de deriva presente no sistema. Apesar da correção, o erro entre as leis de controle ainda é grande.

No **caso B**, A uma **taxa de deriva de 1%**, o relógio do nó 2, sensor 1, está se adiantando em relação ao relógio ideal e aos outros relógios do conjunto, como mostra a Figura J.8. Com isso, o sistema de controle 1 sofre uma variação em sua resposta ao degrau e na lei de controle, pois é afetado pelo relógio.

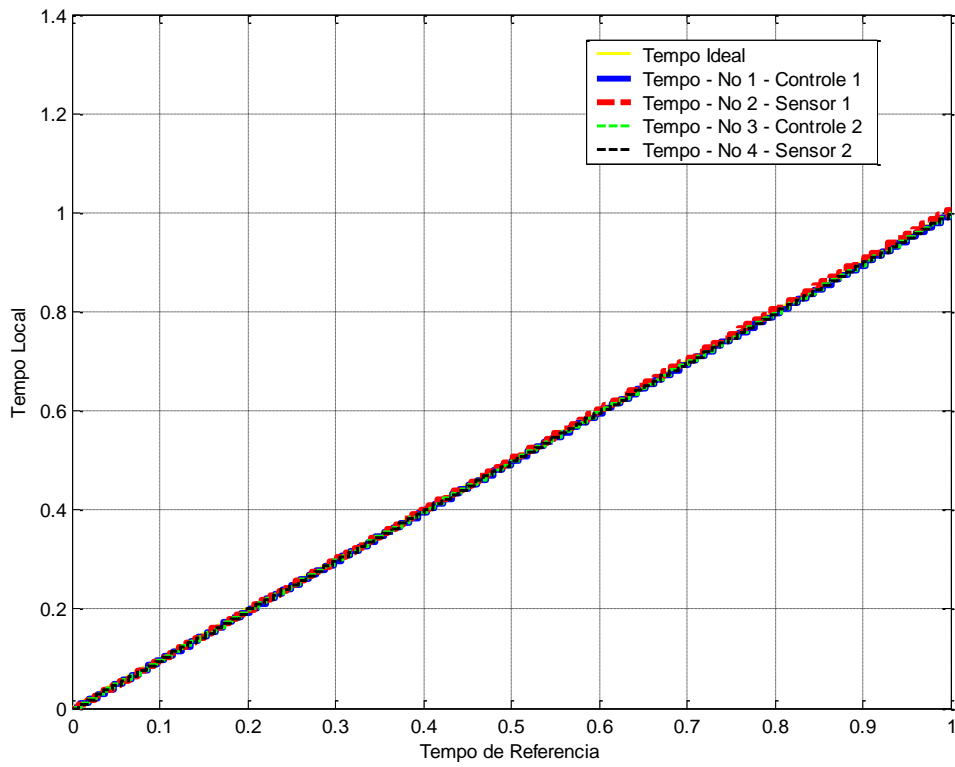
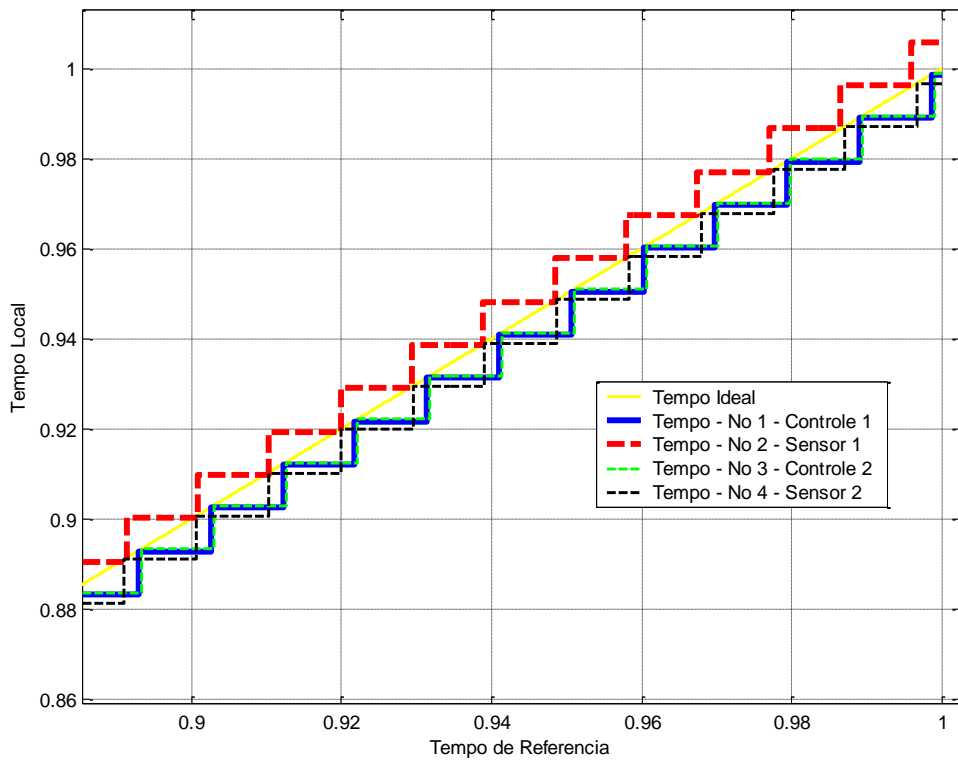


Figura J.8 –Linha do Tempo com deriva de 1%.



(b)

Figura J.9 – Aproximação: Linha do Tempo com deriva de 1%.

A Figura J.10 mostra a resposta ao degrau do sistema de controle com a deriva de 1% no sensor 2. É possível ver que essa variação no nó 2 prejudica um pouco o sistema de controle 1. Existe um pequeno aumento na % de sobre-elevação e no tempo de acomodação. Isto ocorre pois o relógio do sensor 2 está adiantado e com isso as suas medidas chegam adiantadas ao controle. A Figura J.11 mostra as leis de controle das 2 malhas integradas na rede de comunicação.

O sistema de controle por rede 1 (azul) sofre um impacto pequeno com uma taxa de deriva de 1% no relógio do sensor 1. O que mostra que mesmo com uma deriva de relógio baixa diferentemente da arquitetura integrada, nesta arquitetura o sistema é mais sensível aos erros de relógio.

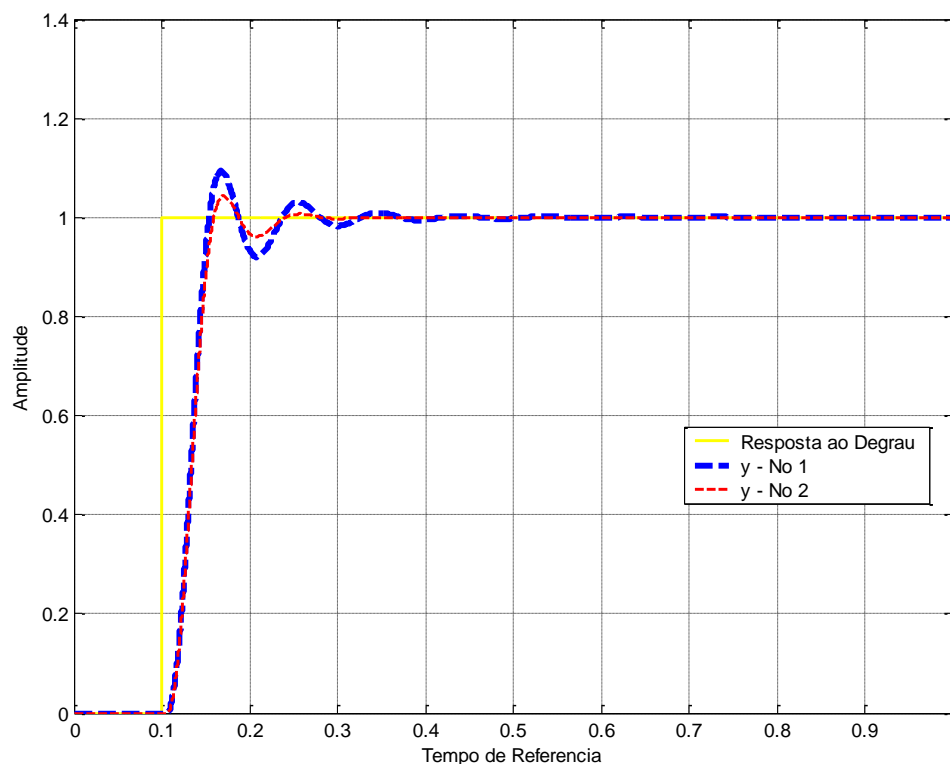


Figura J.10 – Resposta ao Degrau com deriva de 1%.

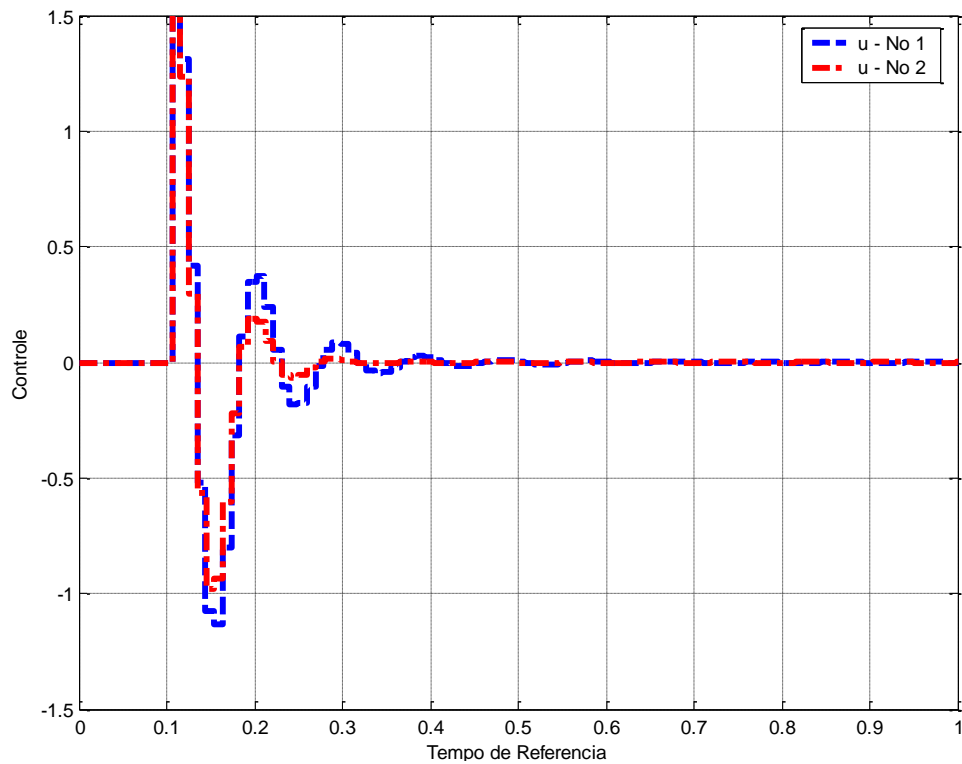
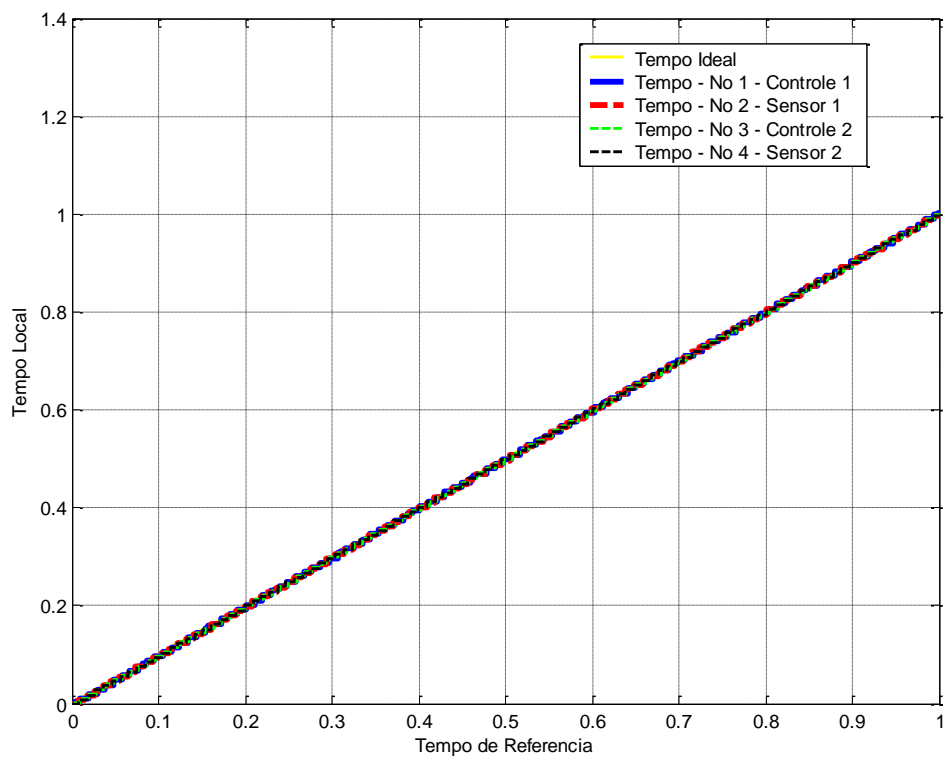


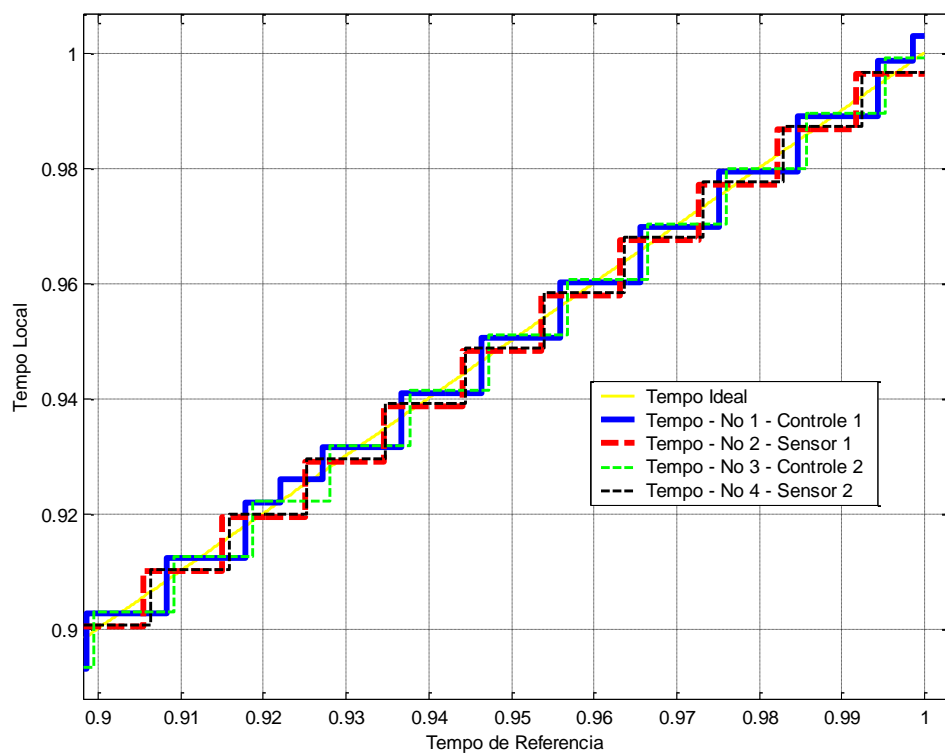
Figura J.11 – Lei de Controle com derivada de 1%.

Para reduzir este impacto, vamos aplicar o algoritmo FTM de sincronização de relógios. A Figura J.12 mostra a linha do tempo com o algoritmo de sincronização FTM aplicado ao sistema. O período de re-sincronização é de 0.0768 segundos (este valor foi escolhido, pois bate com o valor do final de cada TDMA Round), isto é, a cada 0.0768 segundos o algoritmo FTM atualiza os valores de relógio com a correção calculada. A Figura J.12 mostra que os relógios dos nós se sincronizaram entre si.





(a)



(b)

Figura J.12 – (a) Linha do Tempo Sincronizada com deriva de 1%.(b) Aproximação.

Na Figura J.13 é possível ver o gráfico de correção dos relógios dos respectivos nós do sistema.

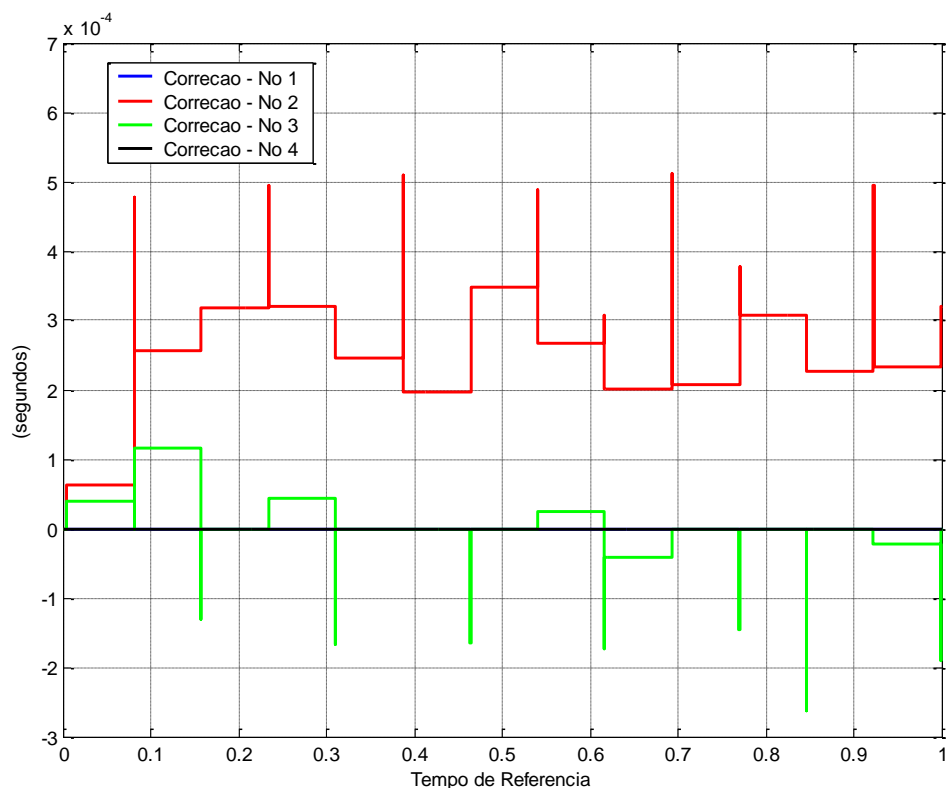


Figura J.13 – Correção Algoritmo FTM com deriva de 1%.

A Figura J.13 mostra que o algoritmo FTM convergiu. Os nós 1, 3 e 4 não estão com erro de deriva. O relógio 1 e 4 não corrigiram nada de seus relógios, o relógio 3 fez algumas pequenas correções em torno de 0.1 (ms) e -0.25 (ms). O nó 2 com 1% de taxa de deriva, teve uma correção periódica em entre 0.2 (ms) e 0.5 (ms). Com isso, é possível analisar através do gráfico J.13 que a precisão dos relógios do sistema ficou em torno de 0.7 (ms).

A Figura J.14 mostra a resposta ao degrau do sistema de controle. E a Figura J.15 mostra as leis de controle do sistema.

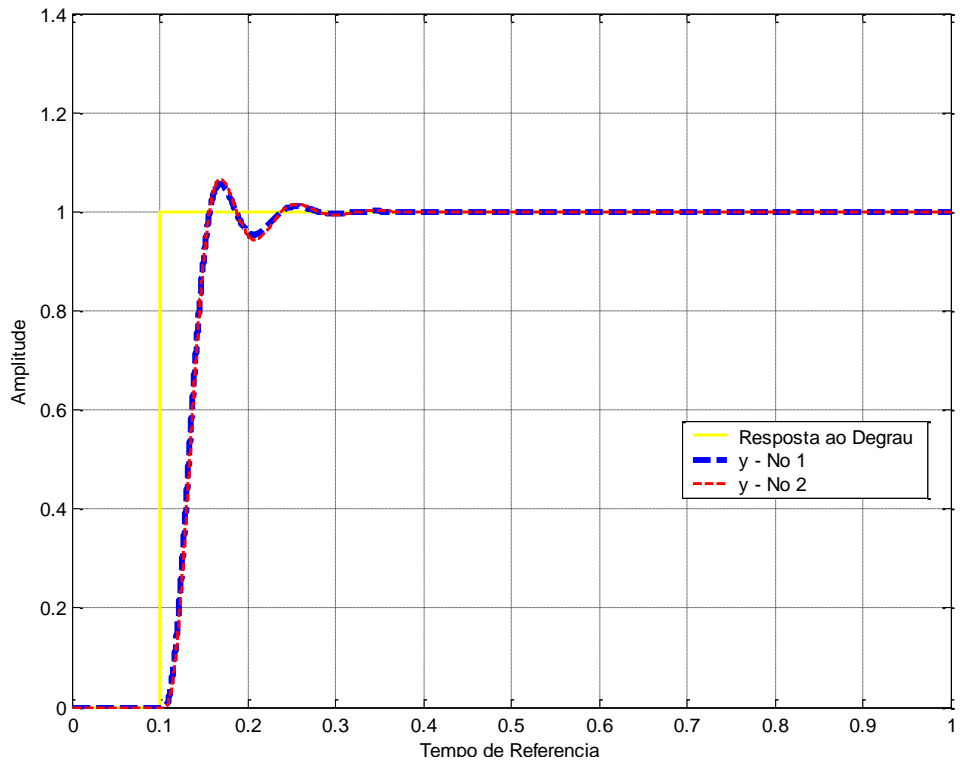


Figura J.14 – Resposta ao Degrau com deriva de 1% - Sincronizado.

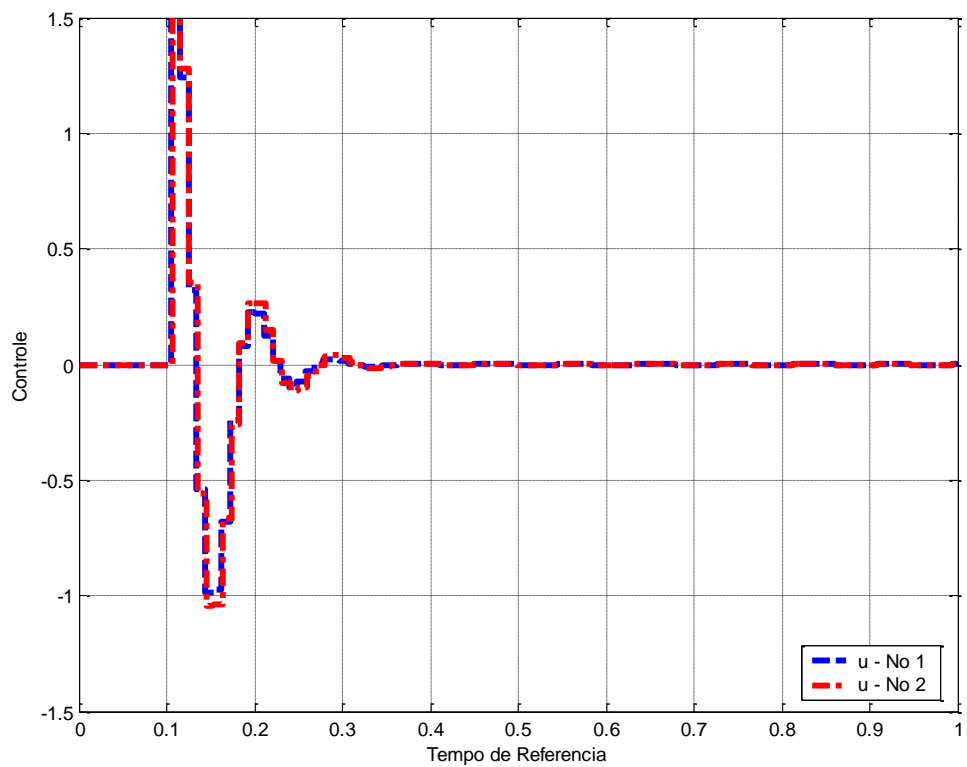


Figura J.15 – Lei de Controle com deriva de 1% - Sincronizado.

Através das Figuras J.14 e J.15 é possível ver que a sincronização de relógios praticamente anulou o aumento da % de sobre elevação da resposta ao degrau e ao tempo de acomodação. E praticamente anulou o atraso da lei de controle. As respostas ao degrau dos dois sistemas de controle são praticamente iguais. Neste caso o algoritmo FTM mostrou-se muito eficiente corrigindo o erro de deriva presente no sistema.

No **caso C**, a uma **taxa de deriva de 10%**, o relógio do nó 2, sensor 1, está se adiantando em relação ao relógio ideal e aos outros relógios do conjunto, como mostra a Figura J.16. Com isso, o sistema de controle por rede 1 sofre uma variação em sua resposta ao degrau e na lei de controle, pois é afetado pelo relógio.

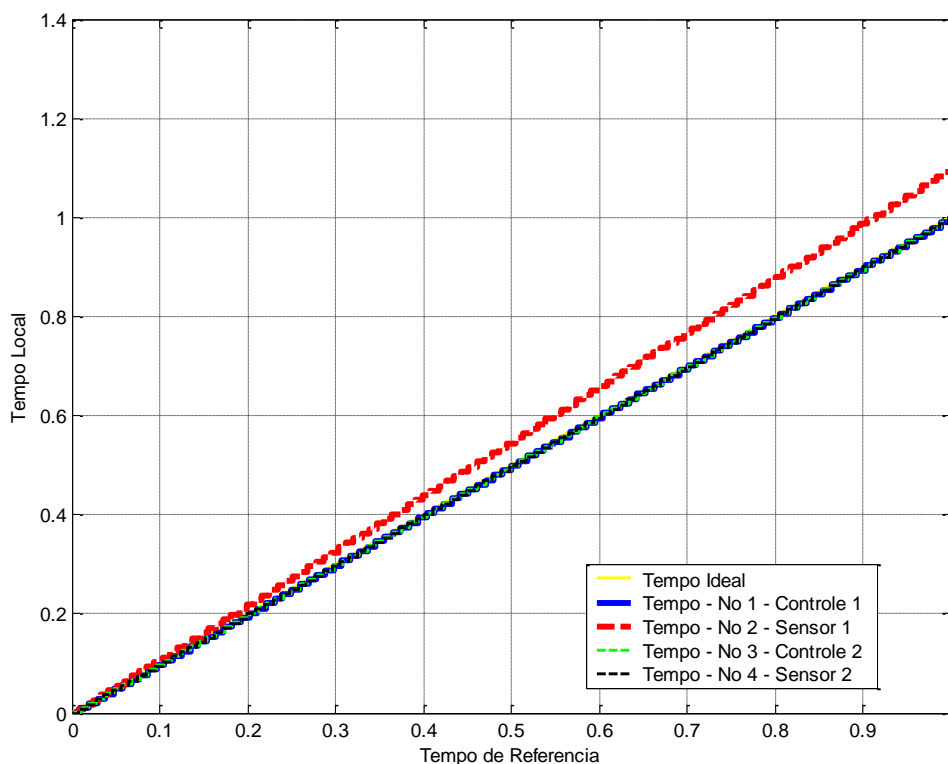


Figura J.16 – Linha do Tempo com deriva de 10%.

A Figura J.17 mostra a resposta ao degrau do sistema de controle com a deriva de 10% no sensor 2. É possível ver que essa variação no nó 2 prejudica o sistema de controle 1. Existe um aumento na % de sobre-elevação e no tempo de acomodação. Isto ocorre pois o relógio do sensor 2 está adiantado e com isso as suas medidas chegam adiantadas ao controle.

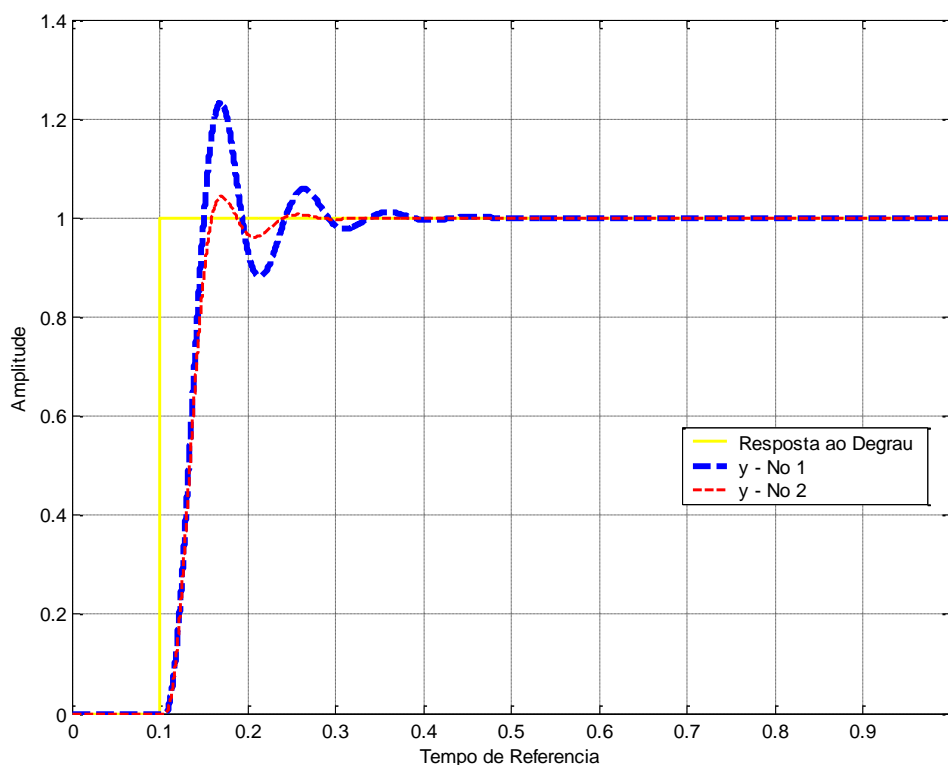


Figura J.17 – Resposta ao Degrau com deriva de 10%.

A Figura J.18 mostra as leis de controle das 2 malhas integradas na rede de comunicação. O sistema de controle por rede 1 (azul) sofre um impacto considerável com uma taxa de deriva de 10% no relógio do sensor 1. Ainda sim o impacto é menor do que com a taxa de deriva de -10%.

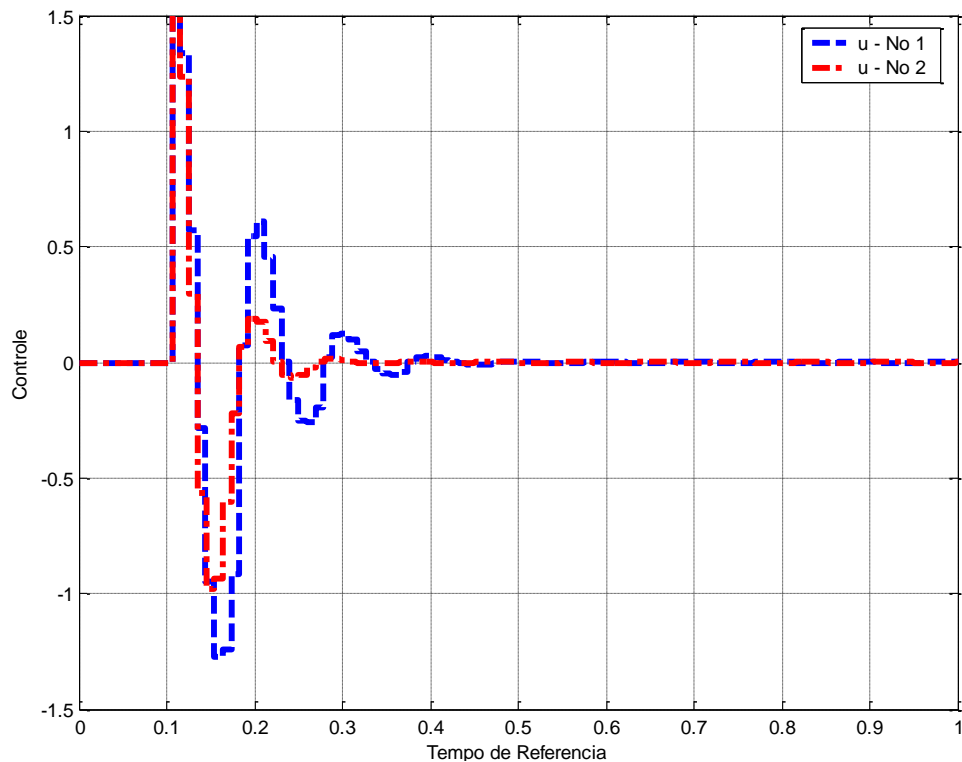
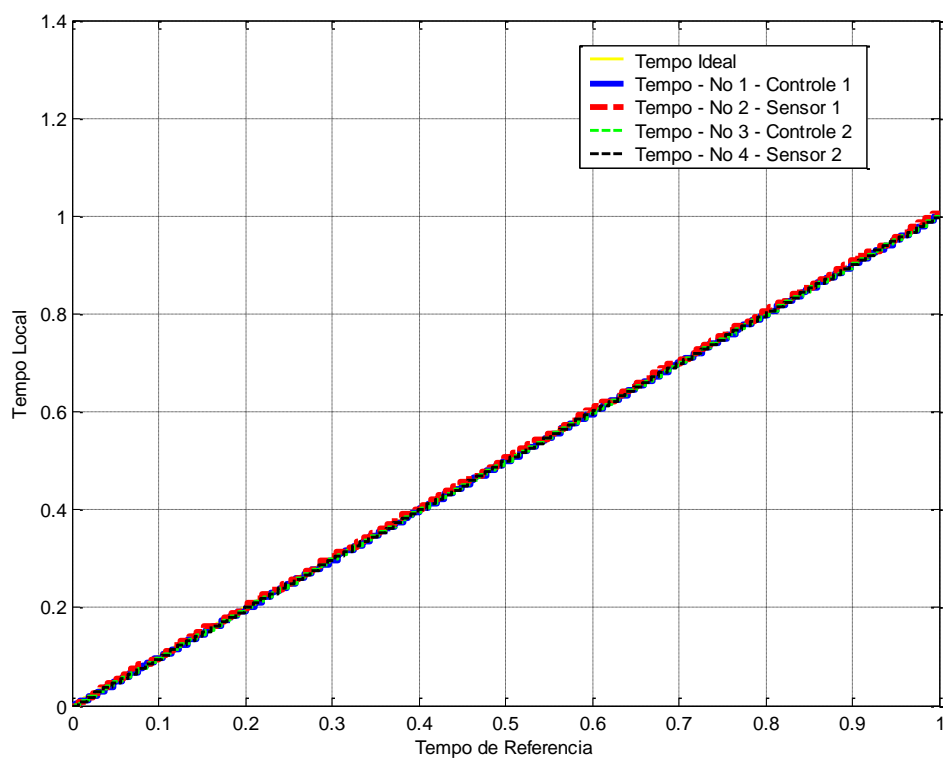
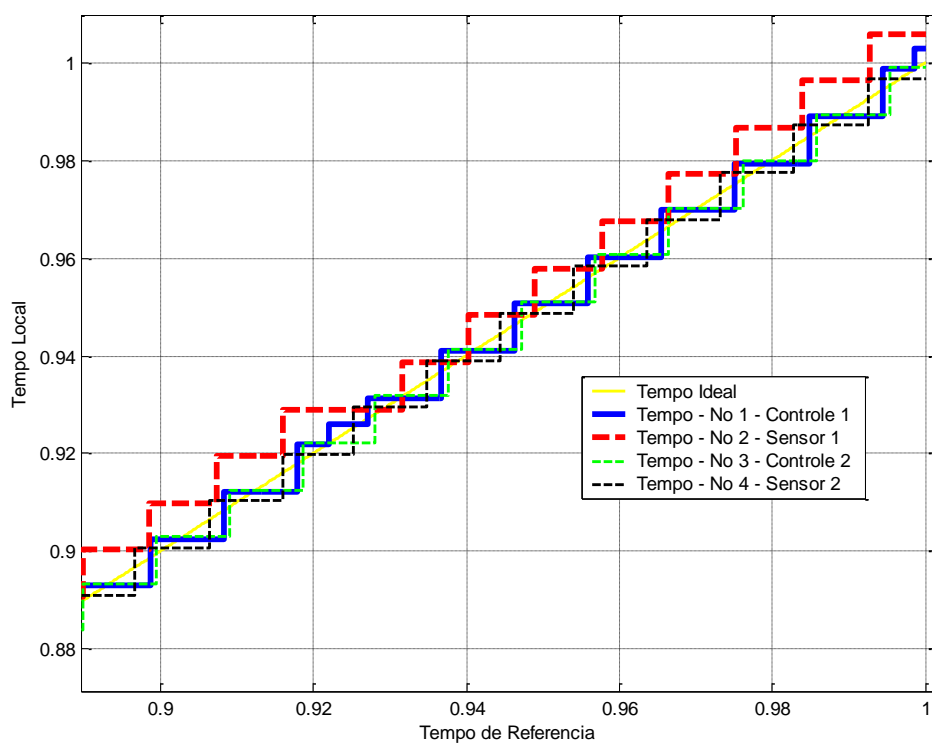


Figura J.18 – Lei de Controle com derivada de 10%.

Para reduzir este impacto, vamos aplicar o algoritmo FTM de sincronização de relógios. A Figura J.19 mostra a linha do tempo com o algoritmo de sincronização FTM aplicado ao sistema. O período de re-sincronização é de 0.0768 segundos (este valor foi escolhido, pois bate com o valor do final de cada TDMA Round), isto é, a cada 0.0768 segundos o algoritmo FTM atualiza os valores de relógio com a correção calculada. A Figura J.19 mostra que os relógios dos nós se sincronizaram entre si.



(a)



(b)

Figura J.19 – (a) Linha do Tempo Sincronizada com deriva de 10%. (b) Aproximação.

Na Figura J.20 é possível ver o gráfico de correção dos relógios dos respectivos nós do sistema.

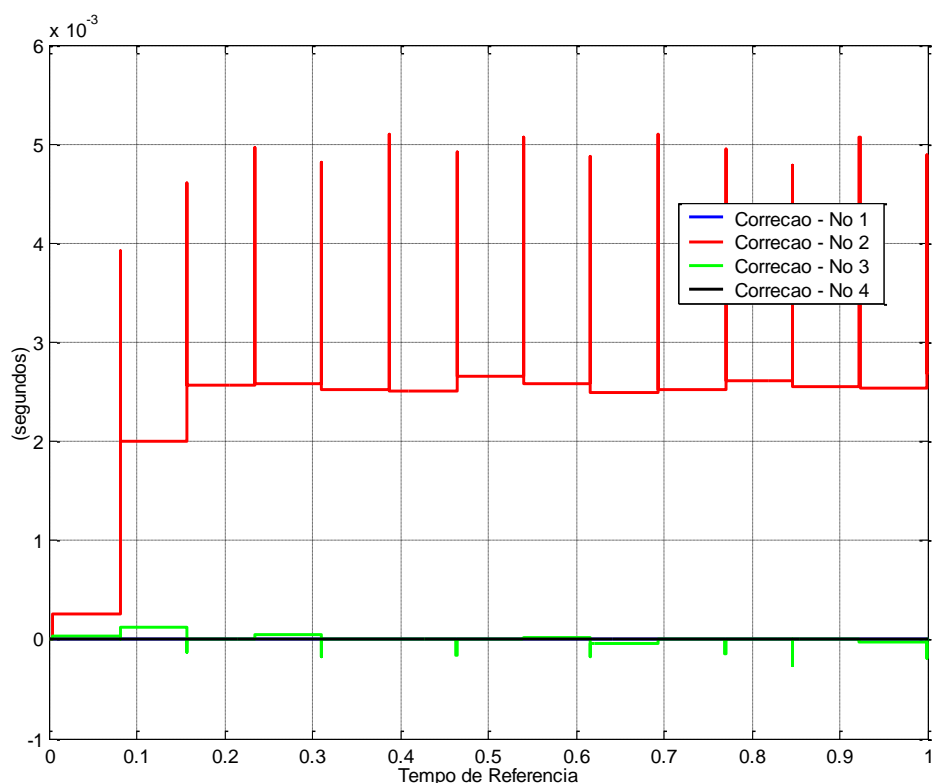


Figura J.20 – Correção Algoritmo FTM com deriva de 10%.

A Figura J.20 mostra que o algoritmo FTM convergiu. Os nós 1, 3 e 4 não estão com erro de deriva. O relógio 1 e 4 não corrigiram nada de seus relógios, o relógio 3 fez algumas pequenas correções. O nó 2 com 10% de taxa de deriva, teve uma correção periódica em entre 2 (ms) e 5 (ms). Com isso, é possível analisar através do gráfico J.20 que a precisão dos relógios do sistema ficou em torno de 5 (ms).

A Figura J.21 mostra a resposta ao degrau do sistema de controle. E a Figura J.22 mostra as leis de controle do sistema, com o algoritmo de sincronização de relógios FTM.



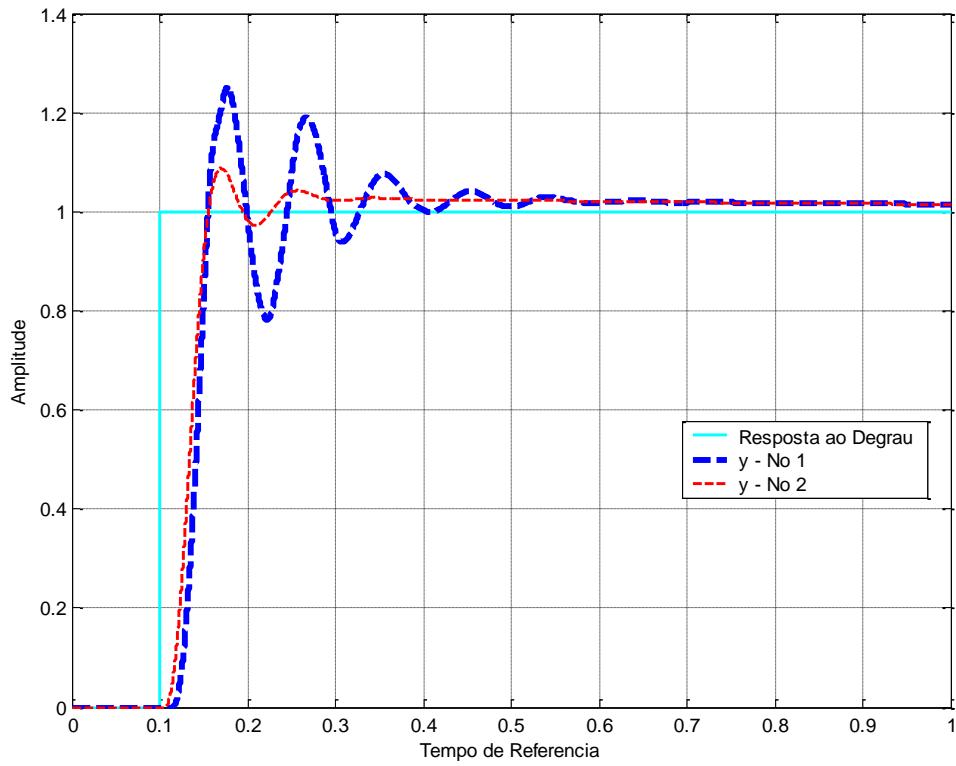


Figura J.21 – Resposta ao Degrau com deriva de 10% Sincronizado.

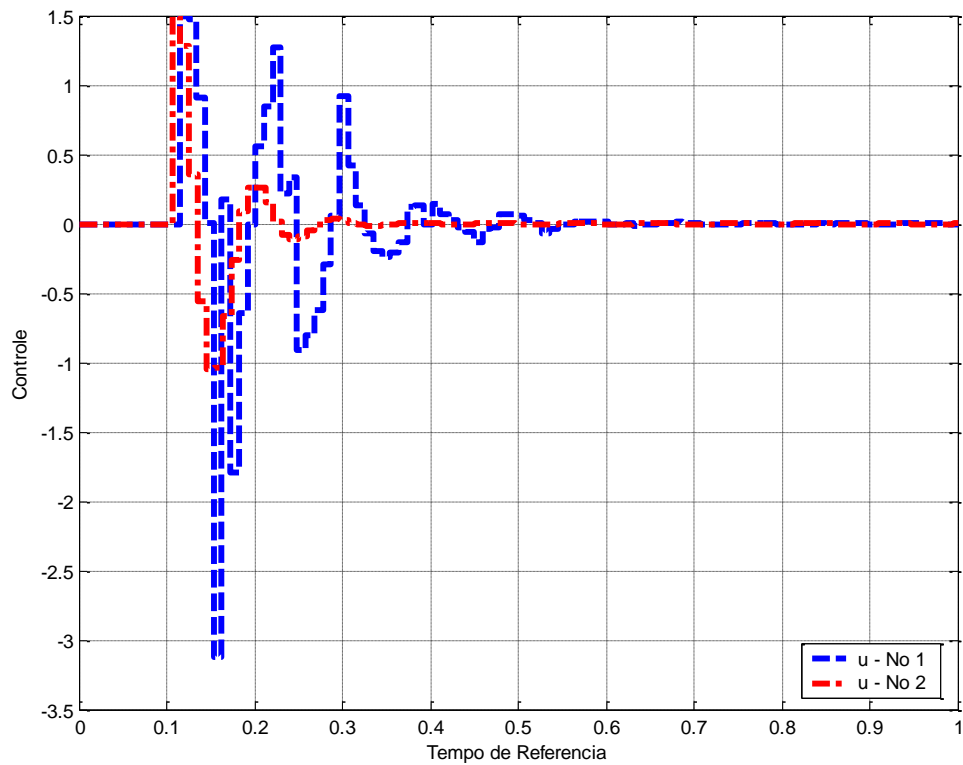


Figura J.22 – Lei de Controle com deriva de 10% - Sincronizado.

Através das Figuras J.21 e J.22 é possível ver que a sincronização de relógios minimizou a % de sobre elevação da resposta ao degrau e praticamente não modificou o tempo de acomodação. E aumentou um pouco o atraso na lei de controle. As respostas ao degrau dos dois sistemas de controle ainda são desiguais. Neste caso o algoritmo FTM mostrou-se razoavelmente eficiente corrigindo o erro de deriva presente no sistema e minimizando a % de sobre elevação e a diferença entre as duas malhas de controle. Apesar da correção, o erro entre as leis de controle ainda é grande.

## APÊNDICE K – SIMULAÇÃO DO 2º CASO – ARQUITETURA INTEGRADA – REDE TDMA

Neste caso, foi feita uma simulação em um sistema de controle com arquitetura integrada, descrito na seção 3.3. A sincronização foi feita através do algoritmo FTM (*Welch-Lynch*), com arquitetura distribuída, isto é, não existe um relógio de referência. Nesta simulação variou-se o parâmetro de viés inicial do relógio do nó 2 (Sensor da Malha 1 de Controle por Rede). A rede de comunicação foi feita por um protocolo de comunicação com acesso ao meio TDMA. Este trabalho escolheu 2 valores de viés inicial diferentes para variação do relógio:

- **Caso A: -0.2 seg.** - O relógio do sensor 1 inicia atrasado de -0.2 seg. em relação aos outros relógios do conjunto;
- **Caso B: 0.39 seg.** - O relógio do sensor 1 inicia adiantado de 0.39 seg. em relação aos outros relógios do conjunto;

No **caso A**, com um **viés inicial de -0.2 segundos**, o relógio do nó 2, sensor 1, inicia atrasado como mostra a Figura K.1.

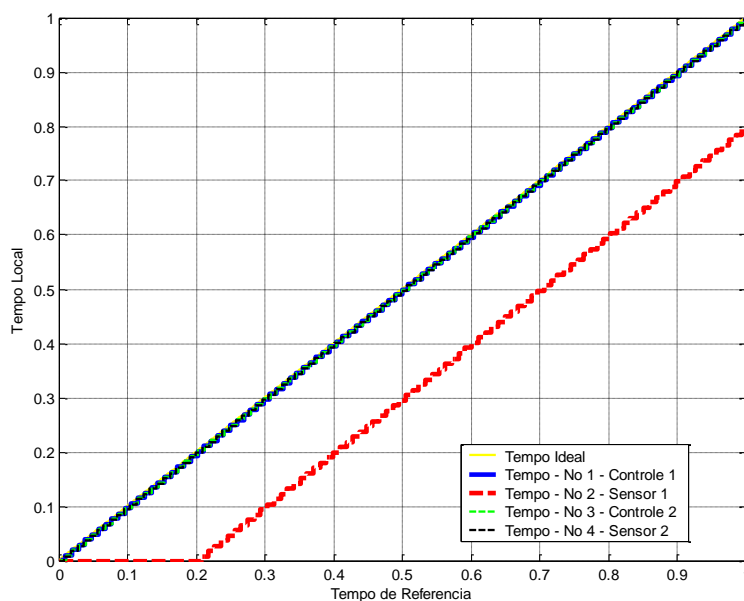


Figura K.1 – Linha do Tempo com viés inicial de -0.2 segundos.

Com isso, o sistema de controle 1 sofre um atraso em sua resposta ao degrau e na lei de controle, pois é afetado pelo relógio. A Figura K.2 mostra a resposta ao degrau do sistema de controle com o viés inicial de -0.2 seg. no sensor 2. É possível ver que esse atraso no nó 2 prejudica o sistema de controle 1. Existe um aumento pequeno na % de sobre-elevação e no tempo de acomodação, pois a lei de controle atrasa de -0.2. seg. Isto ocorre pois o relógio do sensor 2 está atrasado e com isso as suas medidas se atrasam para iniciar a transmissão ao controle. A Figura K.3 mostra as leis de controle das 2 malhas integradas na rede de comunicação. O sistema de controle por rede 1 (azul) sofre um atraso considerável com um viés inicial de -0.2 segundos no relógio do sensor 1.

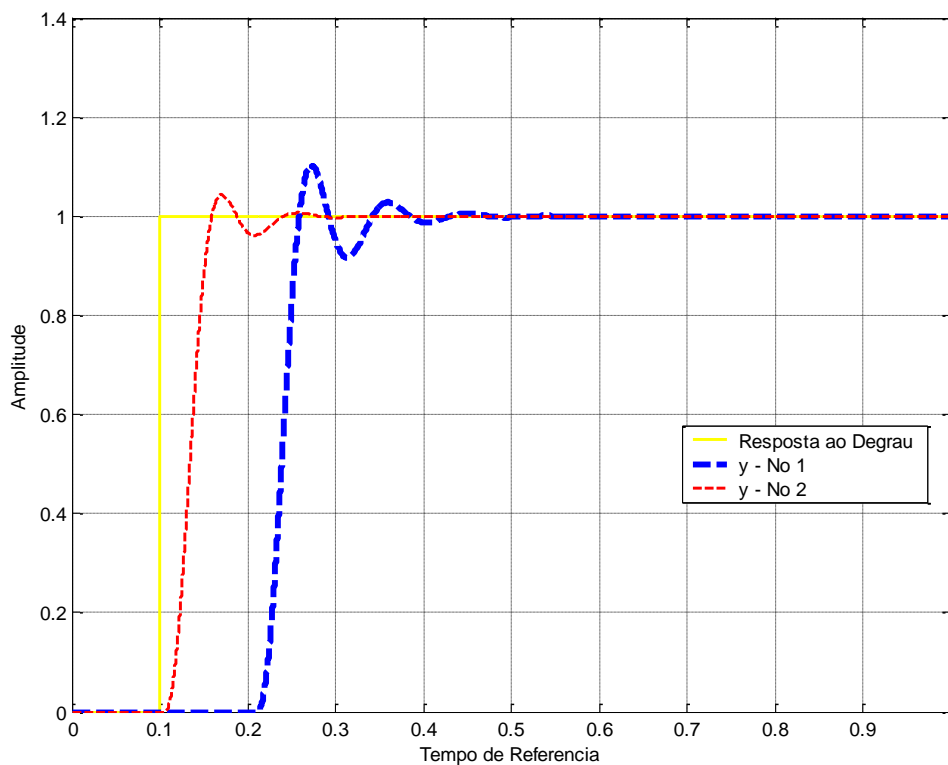


Figura K.2 – Resposta ao Degrau com viés inicial de -0.2 segundos - A1.

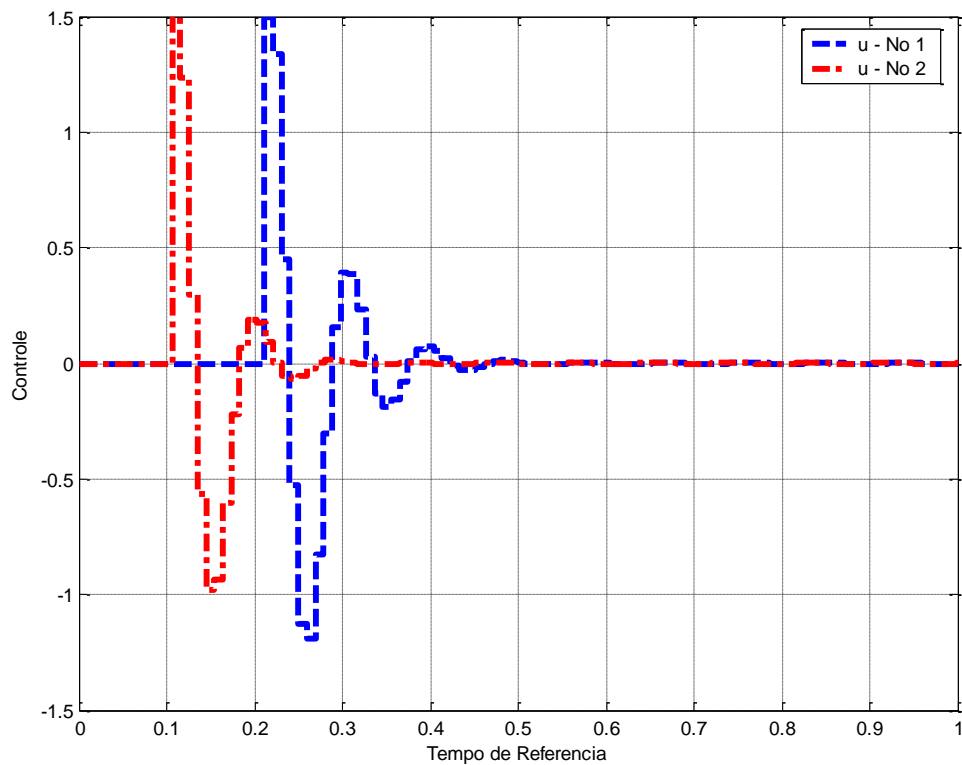
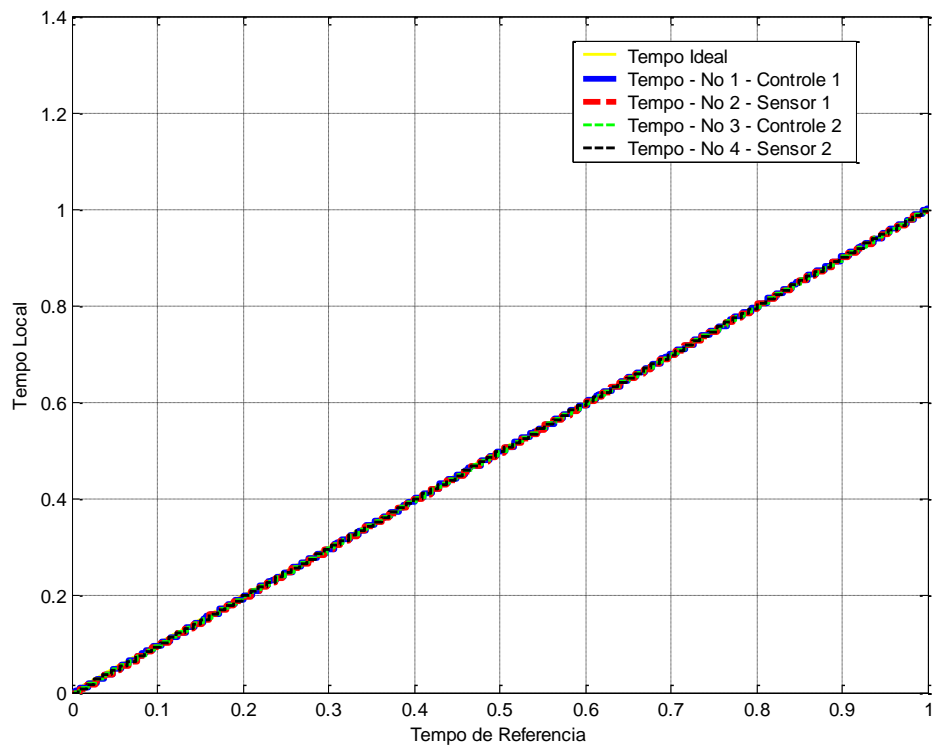
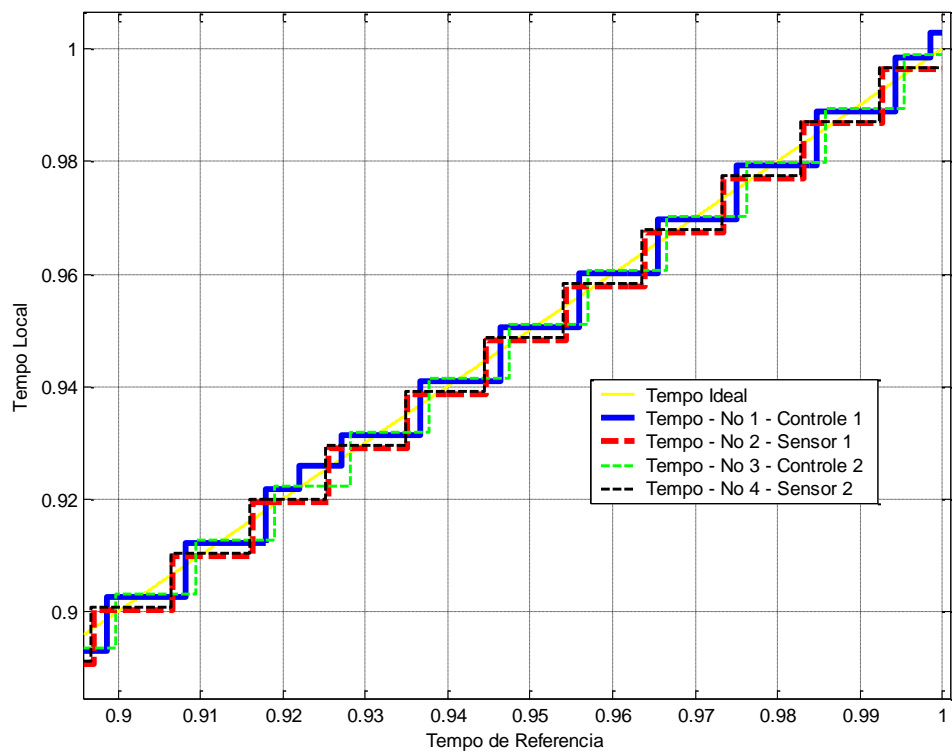


Figura K.3 – Lei de Controle com viés inicial de -0.2 segundos - AI.

Para reduzir este atraso, vamos aplicar o algoritmo FTM de sincronização de relógios. A Figura K.4 mostra a linha do tempo com o algoritmo de sincronização FTM aplicado ao sistema. O período de re-sincronização é de 0.0768 segundos (este valor foi escolhido, pois bate com o valor do final de cada TDMA Round), isto é, a cada 0.0768 segundos o algoritmo FTM atualiza os valores de relógio com a correção calculada. A Figura K.4 mostra que os relógios dos nós se sincronizaram entre si.



(a)



(b)

Figura K.4 – (a) Linha do Tempo Sincronizada: Viés de -0.2 s - AI. (b) Aproximação.

Na Figura K.5 é possível ver o gráfico de correção dos relógios dos respectivos nós do sistema.

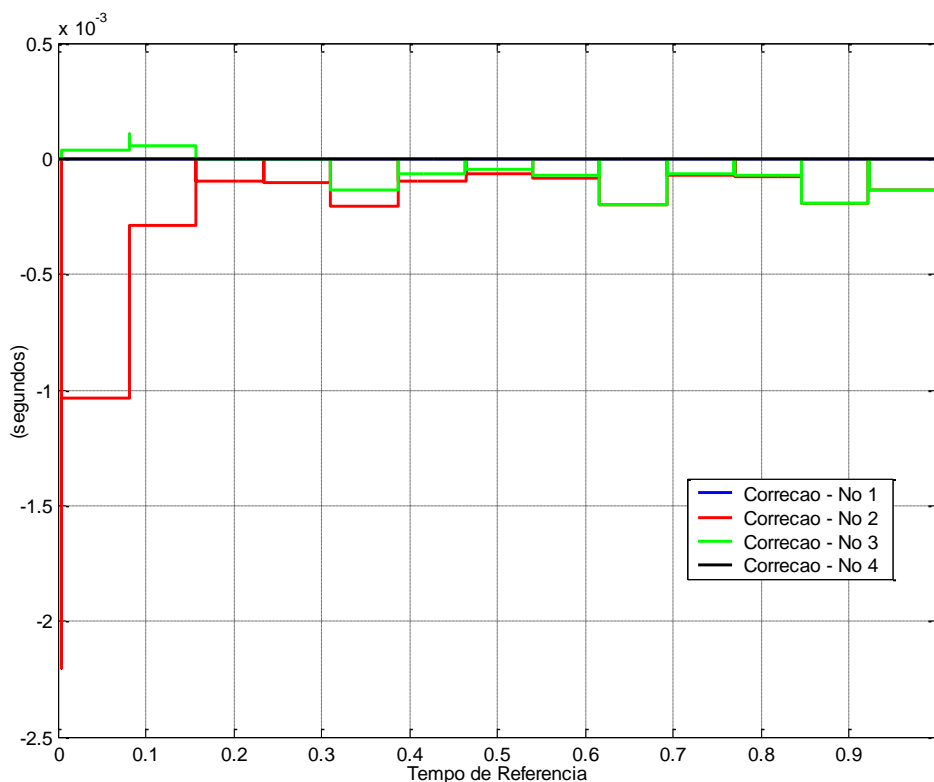


Figura K.5 – Correção Algoritmo FTM com viés inicial de -0.2 segundos.

A Figura K.5 mostra que o algoritmo FTM convergiu. Os nós 1, 3 e 4 não estão com erro de viés inicial. O relógio 1 e 4 não corrigiram nada de seus relógios, o relógio 3 fez algumas pequenas correções. O nó 2 com -0.2 segundos de atraso inicial, teve uma correção grande em torno de -2 (ms) e após essa correção, manteve uma correção periódica entre 0 e -0.5 (ms). Com isso, é possível analisar através do gráfico K.5 que a precisão dos relógios do sistema ficou em torno de 0.5 (ms).

A Figura K.6 mostra a resposta ao degrau do sistema de controle. E a Figura K.7 mostra as leis de controle do sistema.

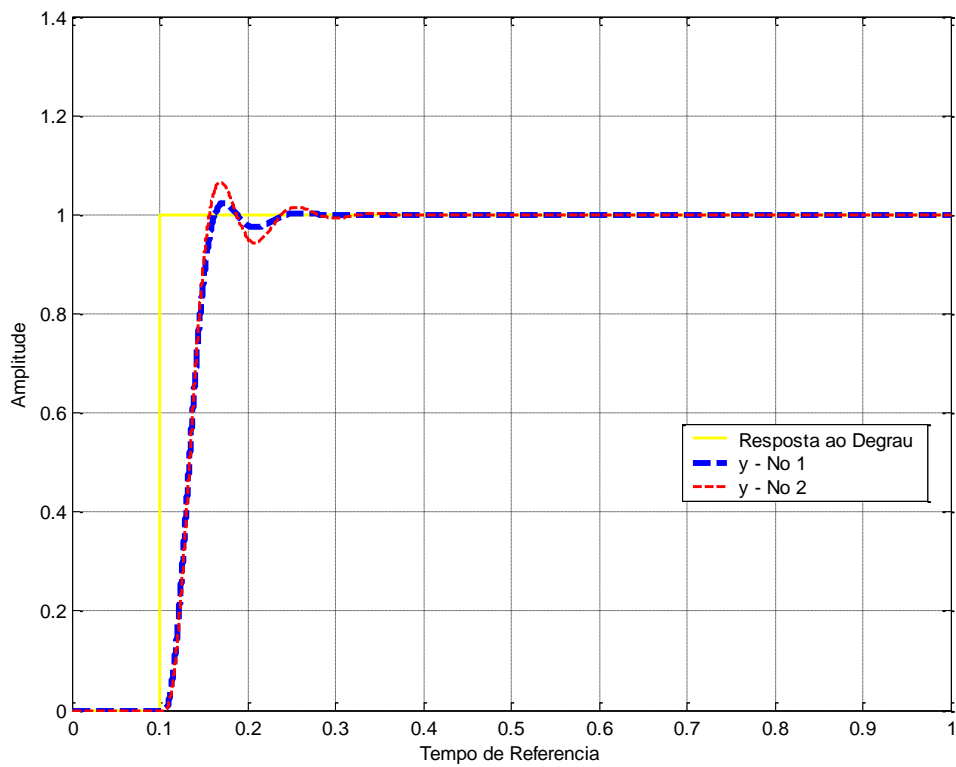


Figura K.6 – Resposta ao Degrau com Viés de -0.2 seg. – AI - Sincronizado.

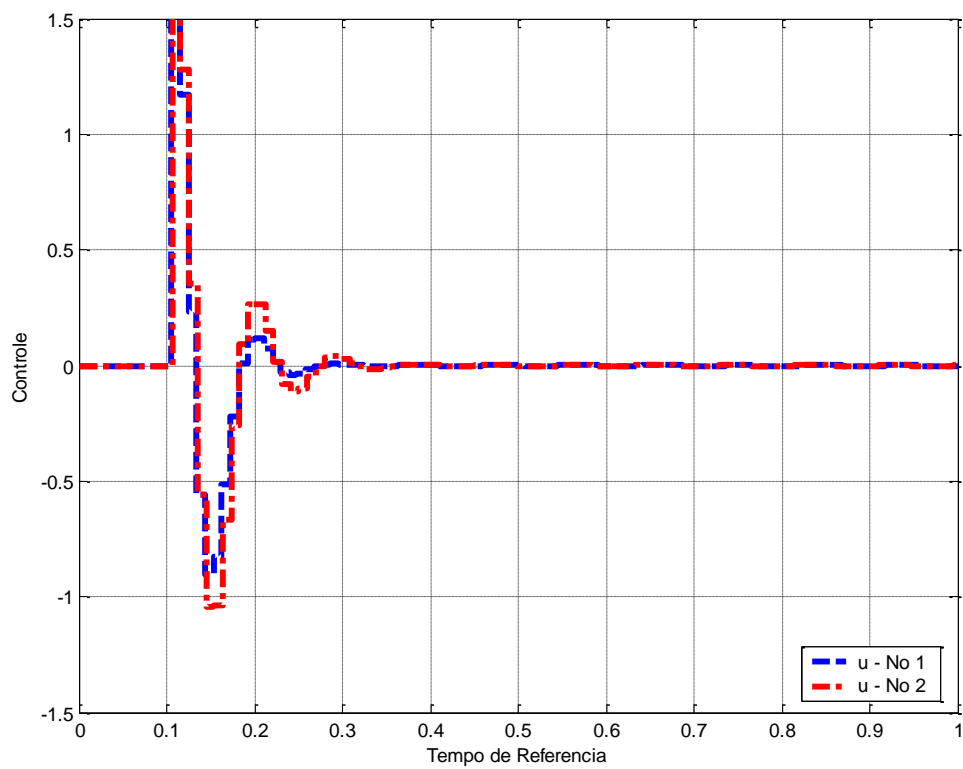


Figura K.7 – Lei de Controle com Viés de -0.2 seg. – AI - Sincronizado.



Através das Figuras K.6 e K.7 é possível ver que a sincronização de relógios minimizou a % de sobre elevação da resposta ao degrau e o tempo de acomodação. A % de sobre elevação com o algoritmo de sincronização é ainda menor que o valor inicial. O algoritmo ainda, praticamente anulou o atraso da lei de controle. As respostas ao degrau dos dois sistemas de controle são muito próximas. Neste caso o algoritmo mostrou-se muito eficiente corrigindo o erro de viés inicial presente no sistema.

No **caso B**, com um **viés inicial de 0.39 segundos**, o relógio do nó 2, sensor 1, inicia adiantado em relação ao relógio ideal e aos outros relógios do conjunto, como mostra a Figura K.8. Com isso, o sistema de controle 1 sofre um atraso em sua resposta ao degrau e na lei de controle, pois é afetado pelo relógio.

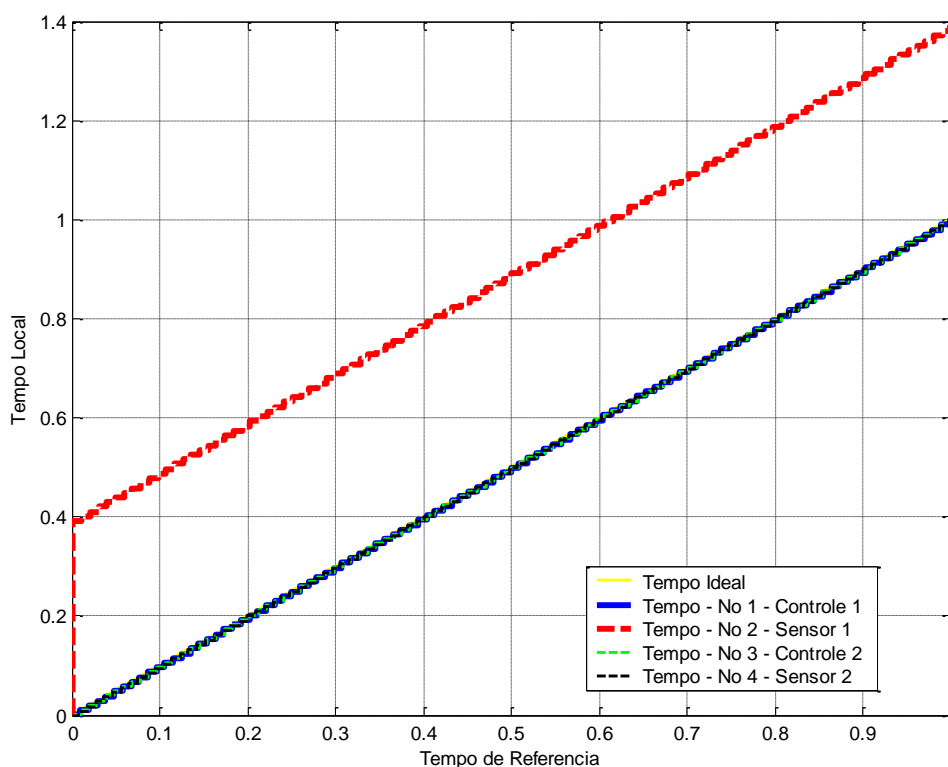


Figura K.8 – Linha do Tempo com viés inicial de 0.39 segundos – AI.

A Figura K.9 mostra a resposta ao degrau do sistema de controle com o viés inicial de 0.39 seg. no sensor 2. É possível ver que esse adiantamento no nó 2 prejudica muito o sistema de controle 1. Existe um aumento na % de sobre-elevação e no tempo de acomodação, pois a lei de controle se adianta de 0.39. seg. Isto ocorre pois o relógio do sensor 2 está adiantado e com isso as suas medidas se adiantam ao controle, com isso o controle é feito com medidas um pouco atrasadas. A Figura K.10 mostra as leis de controle das 2 malhas integradas na rede de comunicação.

O sistema de controle por rede 1 (azul) sofre um impacto considerável com um viés inicial de 0.39 segundos no relógio do sensor 1.

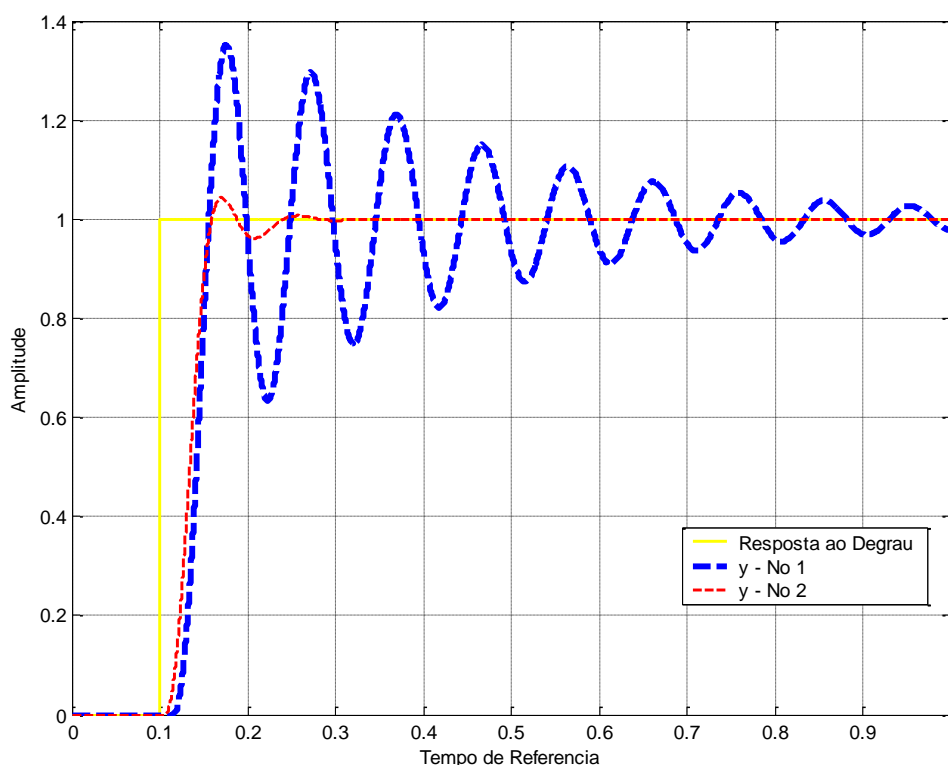


Figura K.9 – Resposta ao Degrau com viés inicial de 0.39 segundos - Al.

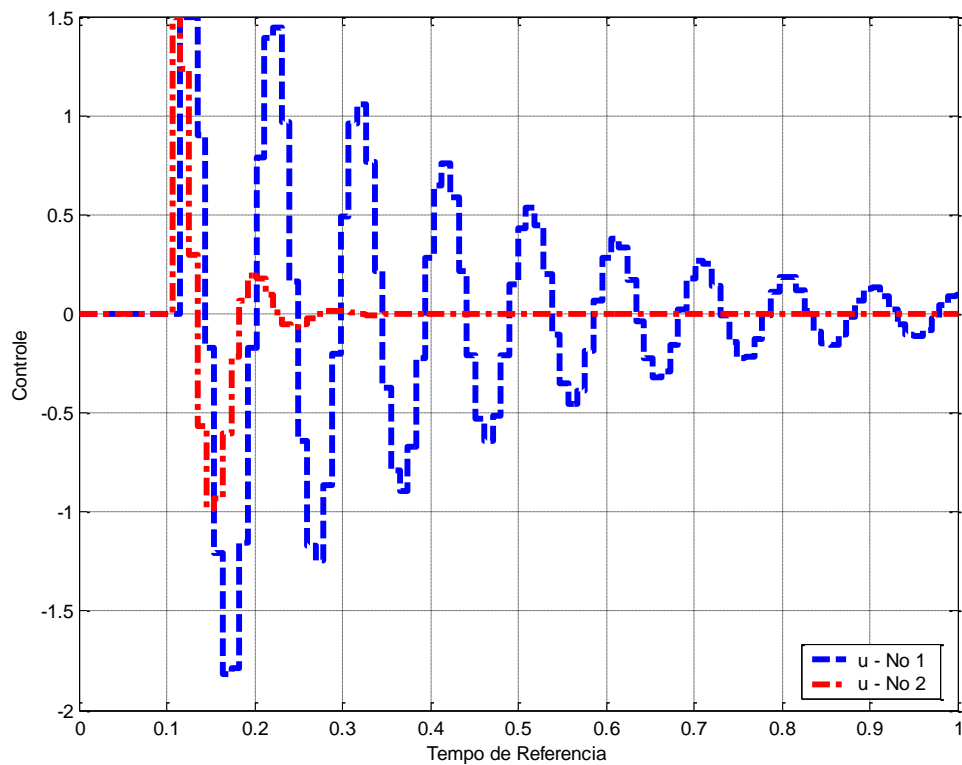
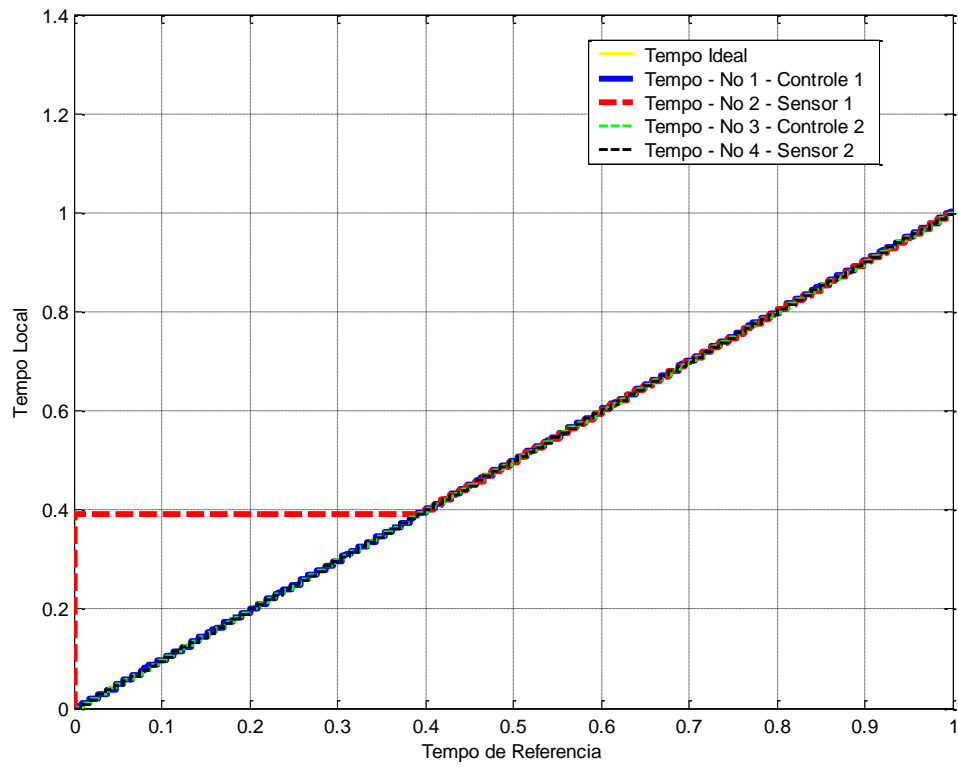
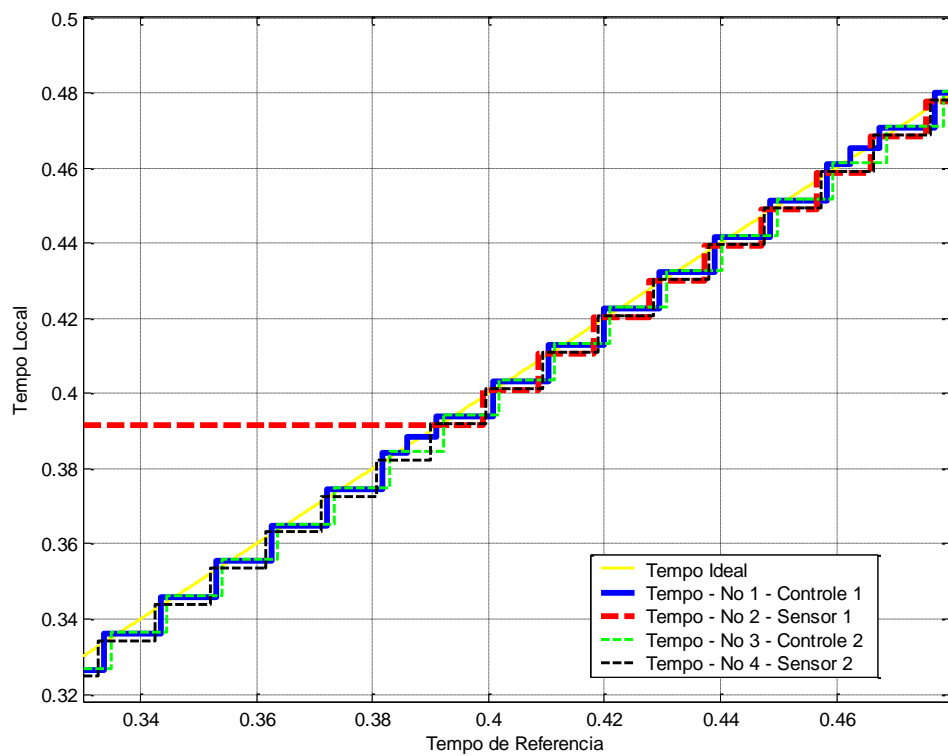


Figura K.10 – Lei de Controle com viés inicial de 0.39 segundos - AI.

Para reduzir este impacto, vamos aplicar o algoritmo FTM de sincronização de relógios. A Figura K.11 mostra a linha do tempo com o algoritmo de sincronização FTM aplicado ao sistema. O período de re-sincronização é de 0.0768 segundos (este valor foi escolhido, pois bate com o valor do final de cada TDMA Round), isto é, a cada 0.0768 segundos o algoritmo FTM atualiza os valores de relógio com a correção calculada. A Figura K.11 mostra que os relógios dos nós se sincronizaram entre si.



(a)



(b)

Figura K.11 – (a) Linha do Tempo Sincronizada: Viés de 0.39 s - AI. (b) Aproximação.

Na Figura K.12 é possível ver o gráfico de correção dos relógios dos respectivos nós do sistema.

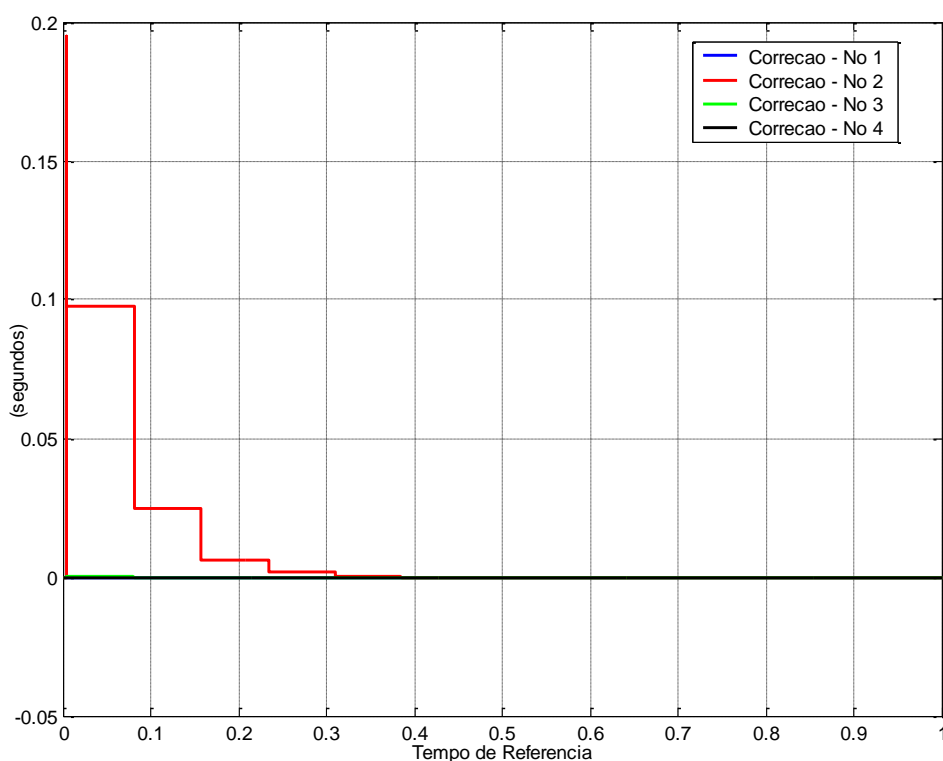


Figura K.12 – Correção Algoritmo FTM com viés inicial de 0.39 seg. - A1.

A Figura K.12 mostra que o algoritmo FTM convergiu. Os nós 1, 3 e 4 não estão com erro de viés inicial. O relógio 1, 3 e 4 não corrigiram nada de seus relógios. O nó 2 com 0.39 segundos de adiantamento inicial, teve uma correção grande em torno de 0.2 (s) e após essa correção, manteve uma correção periódica praticamente nula. Com isso, é possível analisar através do gráfico K.12 que a precisão dos relógios do sistema após a convergência ficou muito próxima de 0.

A Figura K.13 mostra a resposta ao degrau do sistema de controle. E a Figura K.14 mostra as leis de controle do sistema.

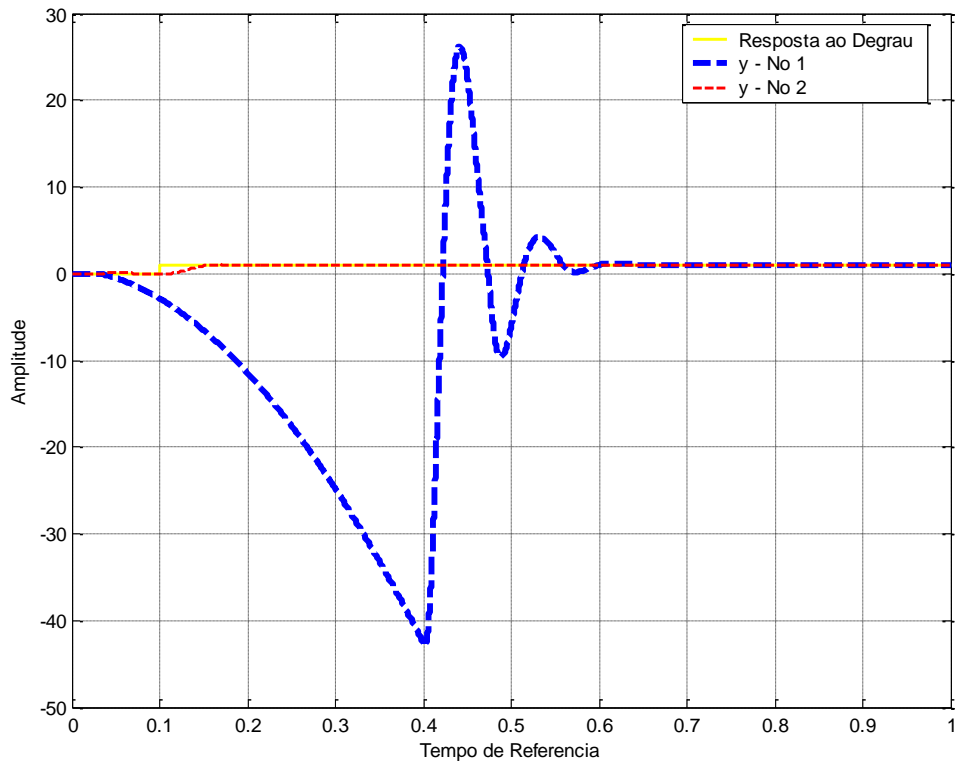


Figura K.13 – Resposta ao Degrau com Viés de 0.39 seg. – AI - Sincronizado.

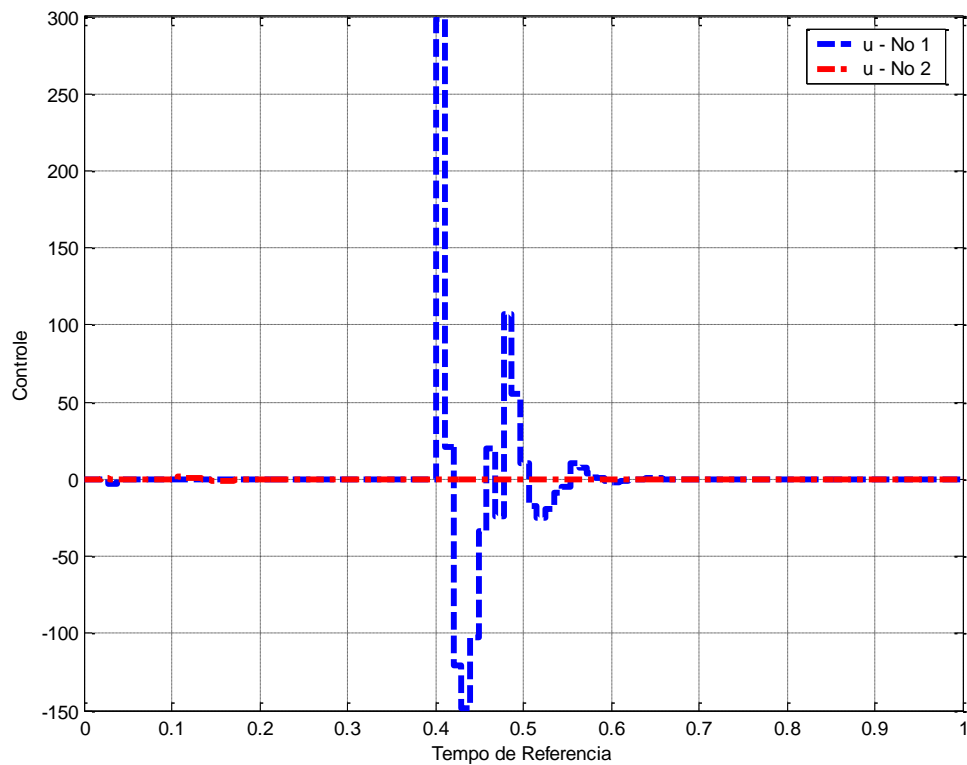


Figura K.14 – Lei de Controle com Viés de 0.39 seg. – AI - Sincronizado.

Através das Figuras K.13 e K.14 é possível ver que a sincronização de relógios prejudicou mais ainda o sistema de controle. Isto ocorre pois o algoritmo FTM é feito principalmente para corrigir efeitos de deriva e supõe que todos os relógios do conjunto estão sincronizados inicialmente. O aumento de erro ocorre pois o algoritmo insere um erro de estado entre 0 e 0.39 segundos, isto é, o relógio fica parado. Após esse período que o sistema volta a atuar. Neste caso o algoritmo não mostrou-se eficiente corrigindo o erro de viés inicial presente no sistema.





## **APÊNDICE L – SIMULAÇÃO DO 3º CASO – ARQUITETURA INTEGRADA – REDE TDMA**

Neste caso, foi feita uma simulação em um sistema de controle com arquitetura integrada, descrito na seção 3.3. A sincronização foi feita através do algoritmo FTM (*Welch-Lynch*), com arquitetura distribuída, isto é, não existe um relógio de referência. Nesta simulação variou-se os parâmetros de viés inicial e deriva do relógio do nó 2 (Sensor da Malha 1 de Controle por Rede). A rede de comunicação foi feita por um protocolo de comunicação com acesso ao meio TDMA.

Este trabalho escolheu 2 valores de viés inicial diferentes para variação do relógio:

- **Caso A: Deriva: -10%, Viés inicial: -0.2 seg.** - O relógio do sensor 1 inicia atrasado de -0.2 seg. e ainda esta se atrasando em relação aos outros relógios do conjunto;
- **Caso B: Deriva: 10%, Viés Inicial: 39 seg.** - O relógio do sensor 1 inicia adiantado de 0.39 seg. e ainda esta se adiantando em relação aos outros relógios do conjunto;

No **caso A**, com um **viés inicial de -0.2 segundos e deriva de -10%**, o relógio do nó 2, sensor 1, inicia atrasado e esta atrasando em relação ao relógio ideal e aos outros relógios do conjunto, como mostra a Figura L.1. Com isso, o sistema de controle 1 sofre um atraso em sua resposta ao degrau e na lei de controle, pois é afetado pelo relógio.

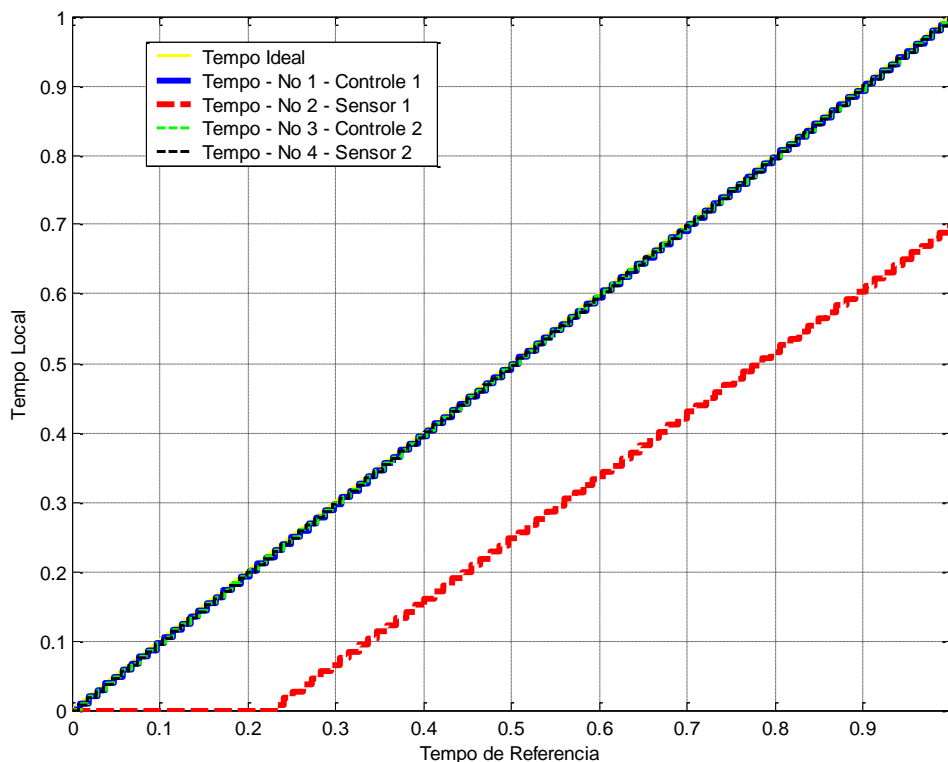


Figura L.1 – Linha do Tempo com Viés inicial: -0.2 segundos, Deriva: -10% - AI.

A Figura L.2 mostra a resposta ao degrau do sistema de controle com o viés inicial de -0.2 seg. e deriva -10% no sensor 2. É possível ver que esse atraso no nó 2 prejudica o sistema de controle 1. Existe um aumento na % de sobre-elevação e no tempo de acomodação, além de atrasar a lei de controle em um pouco mais de -0.2. seg. Isto ocorre pois o relógio do sensor 2 inicia atrasado e esta se atrasando, e com isso as suas medidas são afetadas e conseqüentemente se atrasam para iniciar a transmissão ao controle. A Figura L.3 mostra as leis de controle das 2 malhas integradas na rede de comunicação.

O sistema de controle por rede 1 (azul) sofre um atraso considerável com um viés inicial de -0.2 segundos e deriva de -10% no relógio do sensor 1.

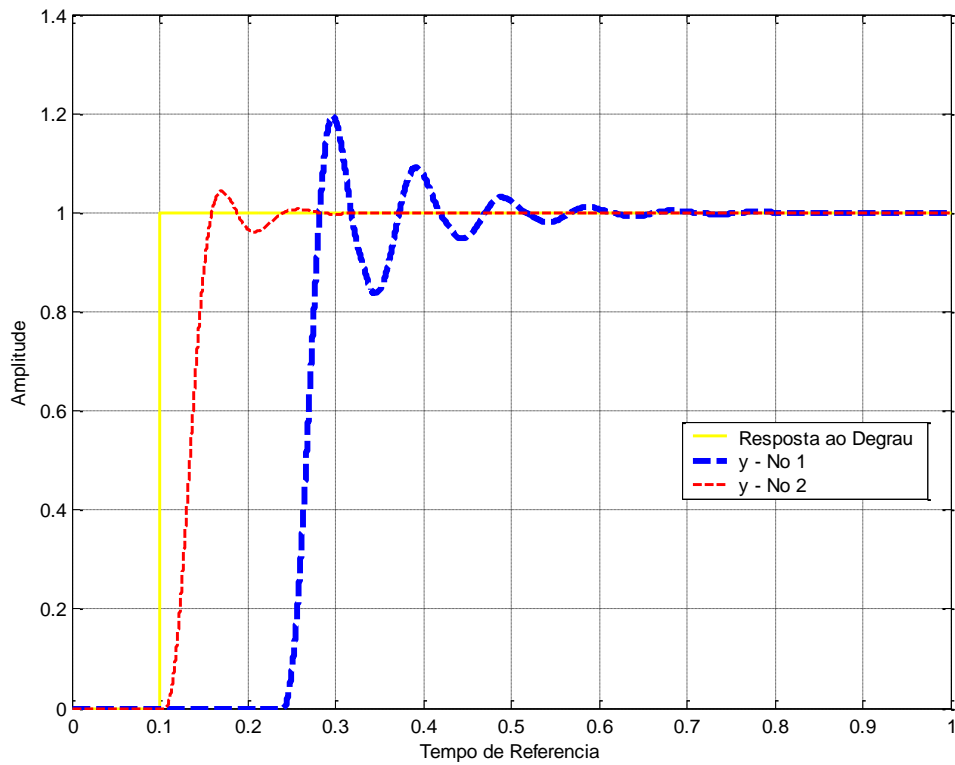


Figura L.2 – Resposta ao Degrau: Viés inicial: -0.2 segundos, Deriva: -10% - AI.

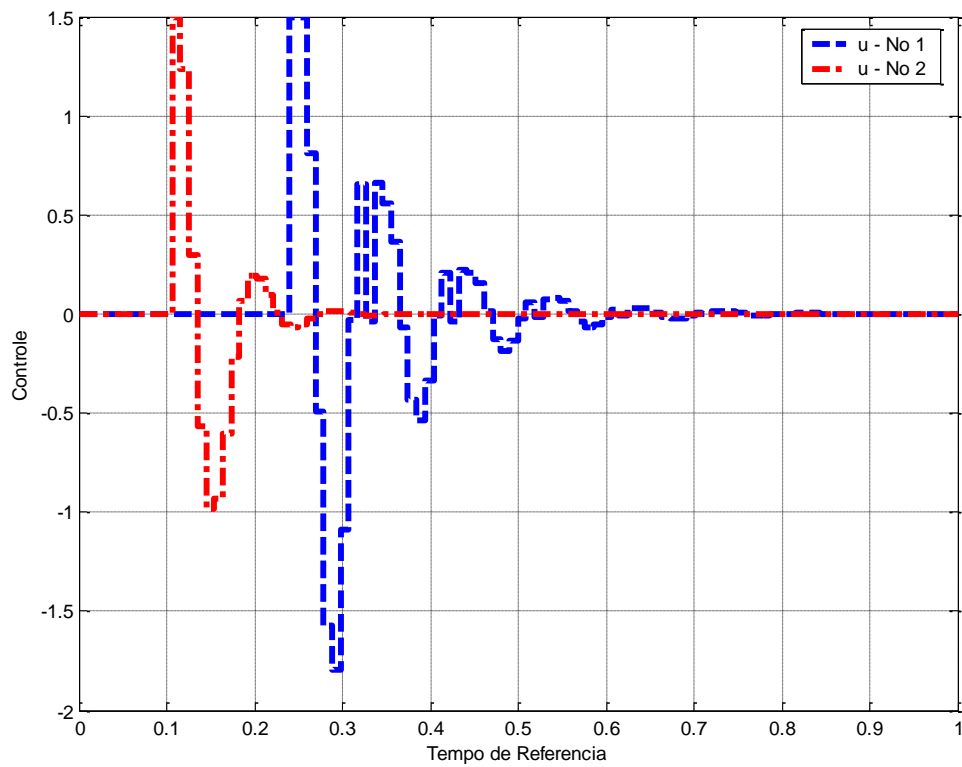


Figura L.3 – Lei de Controle: Viés inicial: -0.2 segundos, Deriva: -10% - AI.

Para reduzir este atraso, vamos aplicar o algoritmo FTM de sincronização de relógios. A Figura L.5 mostra a linha do tempo com o algoritmo de sincronização FTM aplicado ao sistema. O período de re-sincronização é de 0.0768 segundos (este valor foi escolhido, pois bate com o valor do final de cada TDMA Round), isto é, a cada 0.0768 segundos o algoritmo FTM atualiza os valores de relógio com a correção calculada. A Figura L.4 mostra que os relógios dos nós se sincronizaram entre si. Na Figura L.4 é possível ver o gráfico de correção dos relógios dos respectivos nós do sistema.

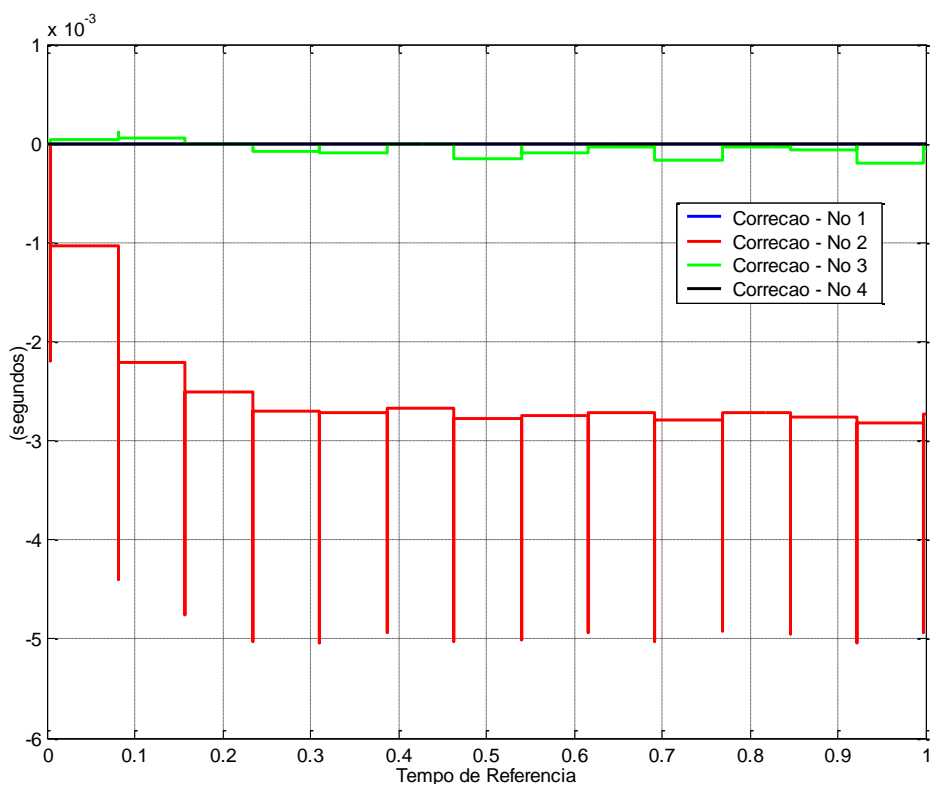
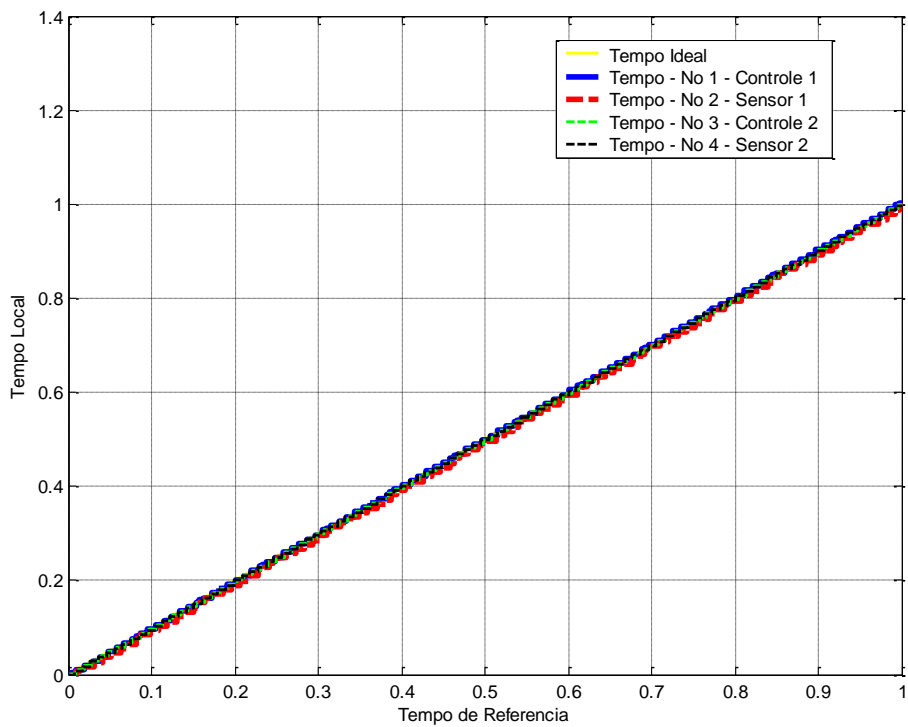
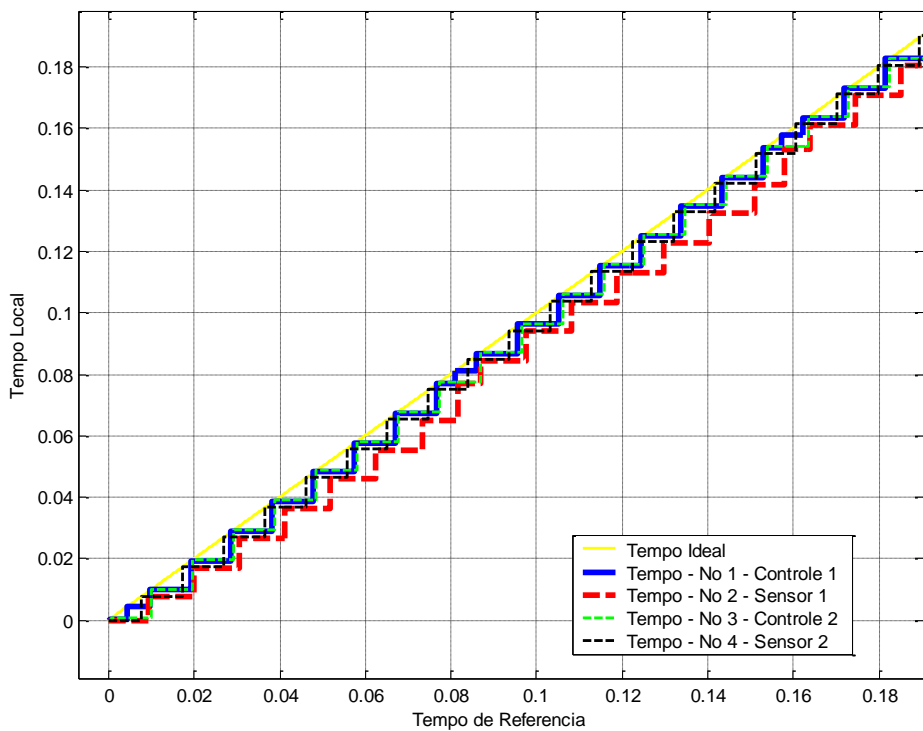


Figura L.4 – Correção Algoritmo FTM: Viés inicial: -0.2 segundos, Deriva: -10% - AI.



(a)



(b)

Figura L.5 – (a) Linha do Tempo Sincronizada: Viés inicial: -0.2(s), Deriva: -10% - AI.

(b) Aproximação.

A Figura L.4 mostra que o algoritmo FTM convergiu. Os nós 1, 3 e 4 não estão com erro de viés inicial e deriva. O relógio 1 e 4 não corrigiram nada de seus relógios, o relógio 3 fez algumas pequenas correções. O nó 2 com -0.2 segundos de atraso inicial e taxa de deriva de -10%, manteve uma correção periódica entre -2 e -5 (ms). Com isso, é possível analisar através do gráfico L.4 que a precisão dos relógios do sistema ficou em torno de 5 (ms).

A Figura L.6 mostra a resposta ao degrau do sistema de controle. E a Figura L.7 mostra as leis de controle do sistema.

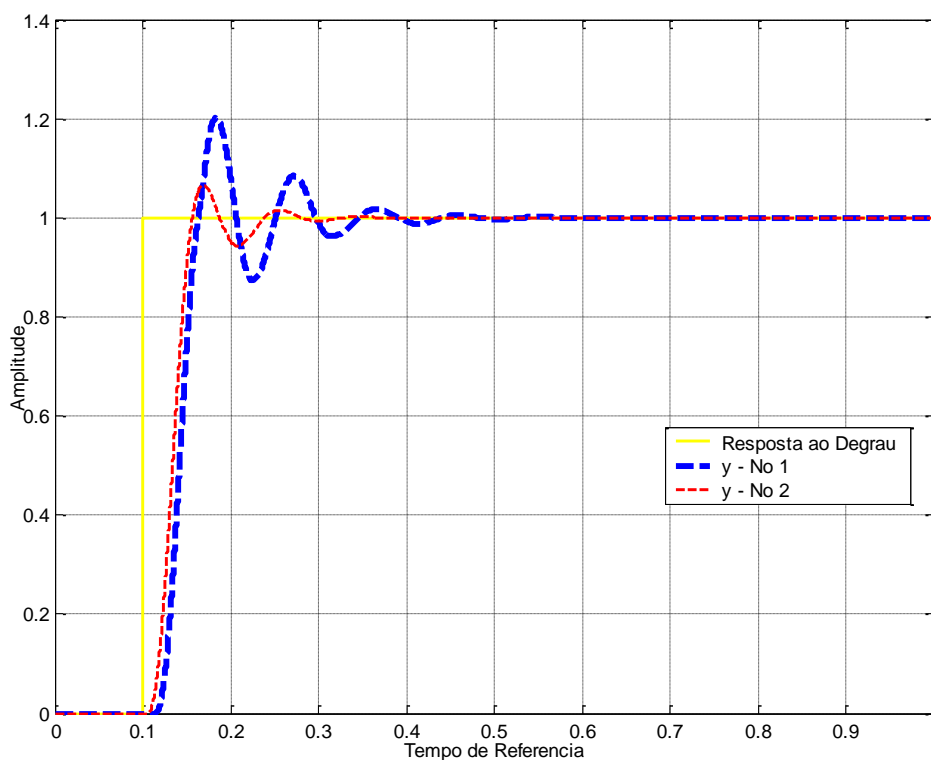


Figura L.6 – Resposta ao Degrau Sincronizada: Viés inicial: -0.2(s), Deriva: -10% - AI.

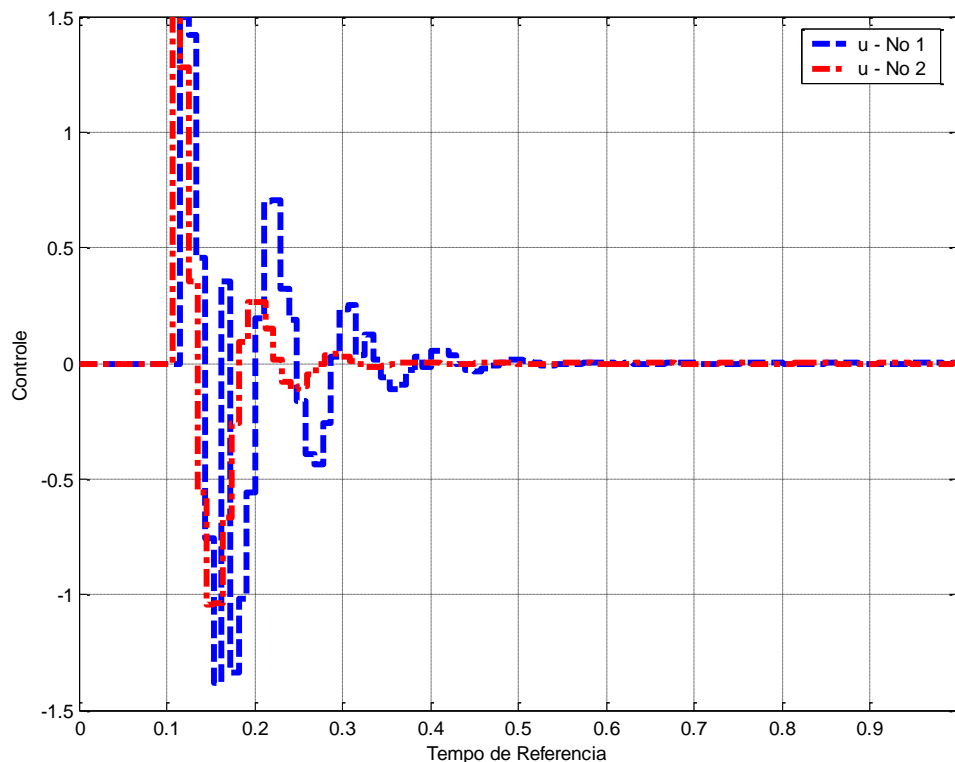


Figura L.7 – Lei de Controle Sincronizada: Viés inicial: -0.2(s), Deriva: -10% - AI.

Através das Figuras L.6 e L.7 é possível ver que a sincronização de relógios reduziu o atraso causado pelo viés inicial, mas o atraso da deriva de relógio ainda continuou presente. A % de sobre elevação da resposta ao degrau e o tempo de acomodação praticamente não modificaram. As respostas ao degrau dos dois sistemas de controle são diferentes. Neste caso o algoritmo mostrou-se muito eficiente corrigindo o erro de viés inicial mas não corrigiu muito bem o erro de deriva do relógio, presentes no sistema.

No **caso B**, com um **viés inicial de 0.39 segundos e deriva de 10%**, o relógio do nó 2, sensor 1, inicia adiantado e esta se adiantando em relação ao relógio ideal e aos outros relógios do conjunto, como mostra a Figura L.8. Com isso, o sistema de controle 1 sofre um adiantamento em sua resposta ao degrau e na lei de controle, pois é afetado pelo relógio.

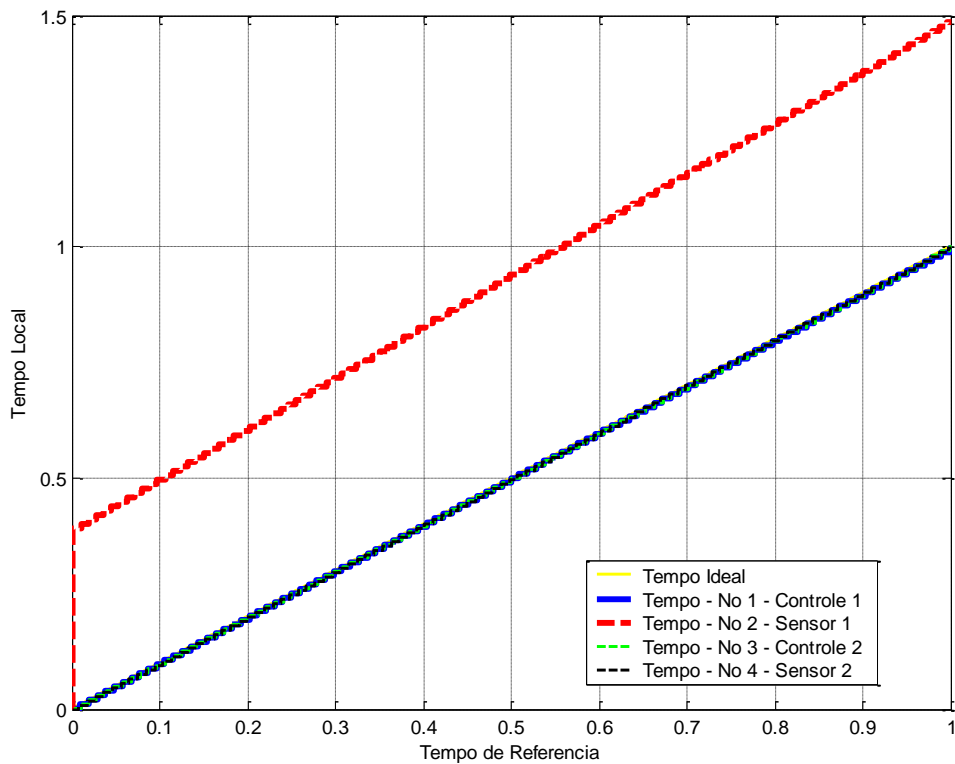


Figura L.8 – Linha do Tempo com Viés inicial: 0.39 segundos, Deriva: 10% - AI.

A Figura L.9 mostra a resposta ao degrau do sistema de controle com o viés inicial de 0.39 seg. e deriva 10% no sensor 2. É possível ver que esse atraso no nó 2 prejudica o sistema de controle 1. Existe um aumento na % de sobre-elevação e no tempo de acomodação. Isto ocorre pois o relógio do sensor 2 inicia adiantado e esta se adiantando, e com isso as suas medidas são afetadas e conseqüentemente se adiantam à transmissão ao controle. A Figura L.10 mostra as leis de controle das 2 malhas integradas na rede de comunicação.



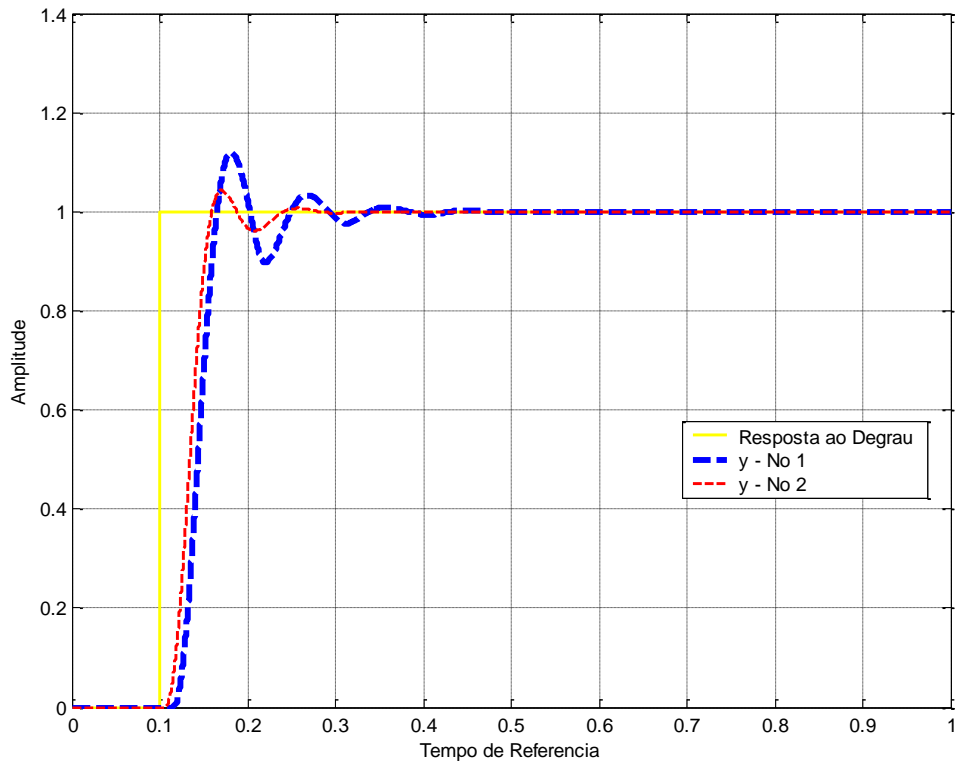


Figura L.9 – Resposta ao Degrau: Viés inicial: 0.39 segundos, Deriva: 10% - AI.

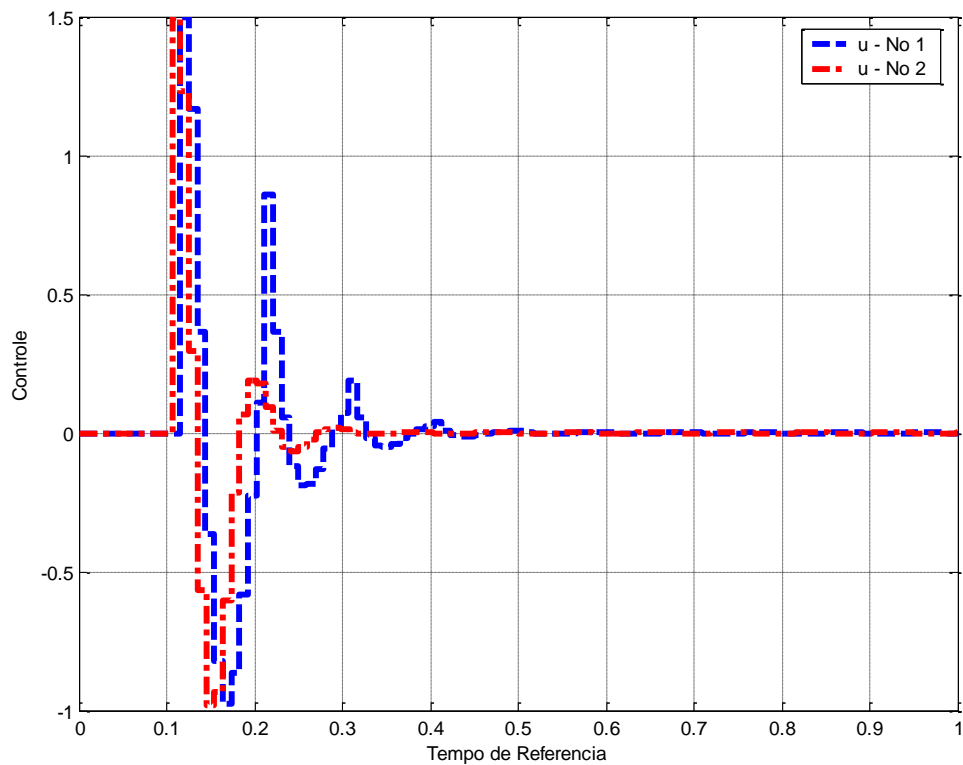


Figura L.10 – Lei de Controle: Viés inicial: 0.39 segundos, Deriva: 10% - AI.

Para reduzir este erro, vamos aplicar o algoritmo FTM de sincronização de relógios. A Figura L.12 mostra a linha do tempo com o algoritmo de sincronização FTM aplicado ao sistema. O período de re-sincronização é de 0.0768 segundos (este valor foi escolhido, pois bate com o valor do final de cada TDMA Round), isto é, a cada 0.0768 segundos o algoritmo FTM atualiza os valores de relógio com a correção calculada. A Figura L.12 mostra que os relógios dos nós se sincronizaram entre si. Na Figura L.11 é possível ver o gráfico de correção dos relógios dos respectivos nós do sistema.

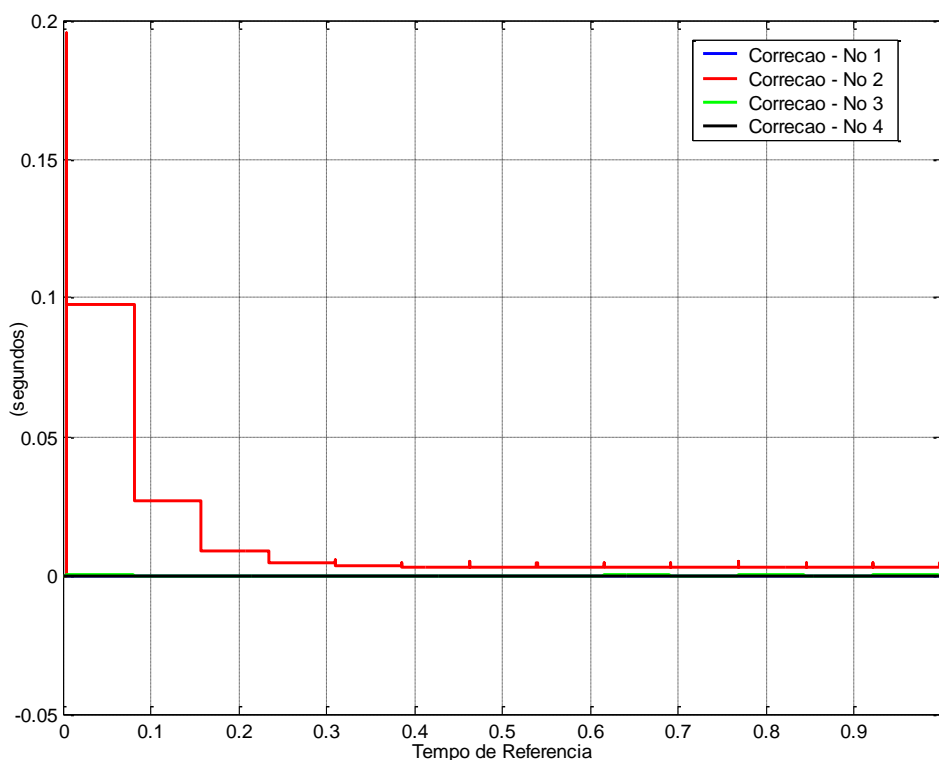


Figura L.11 – Correção Algoritmo FTM: Viés inicial: 0.39 segundos, Deriva: 10% - Al.

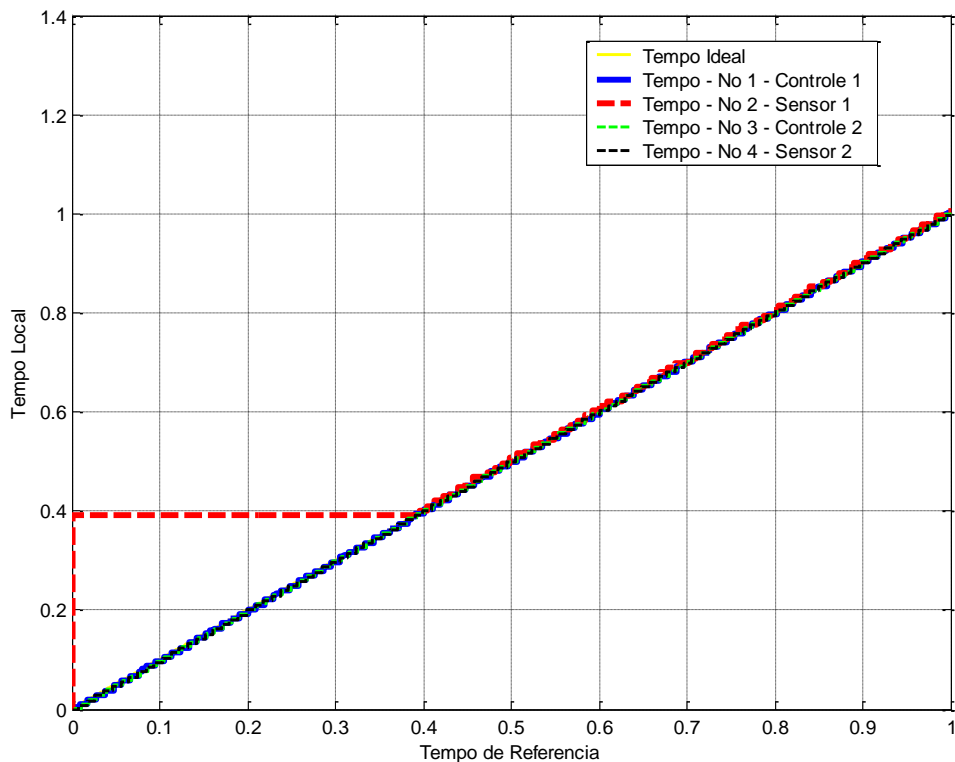


Figura L.12 – (a) Linha do Tempo Sincronizada: Viés inicial: 0.39(s), Deriva: 10% - AI.

A Figura L.11 mostra que o algoritmo FTM convergiu. Os nós 1, 3 e 4 não estão com erro de viés inicial e deriva. Os relógios 1, 3 e 4 não corrigiram nada de seus relógios. O nó 2 com 0.39 segundos de adiantamento inicial e taxa de deriva de 10%, manteve uma correção periódica em torno de 4 (ms). Com isso, é possível analisar através do gráfico L.11 que a precisão dos relógios do sistema ficou em torno de 4 (ms).

A Figura L.13 mostra a resposta ao degrau do sistema de controle. E a Figura L.14 mostra as leis de controle do sistema.

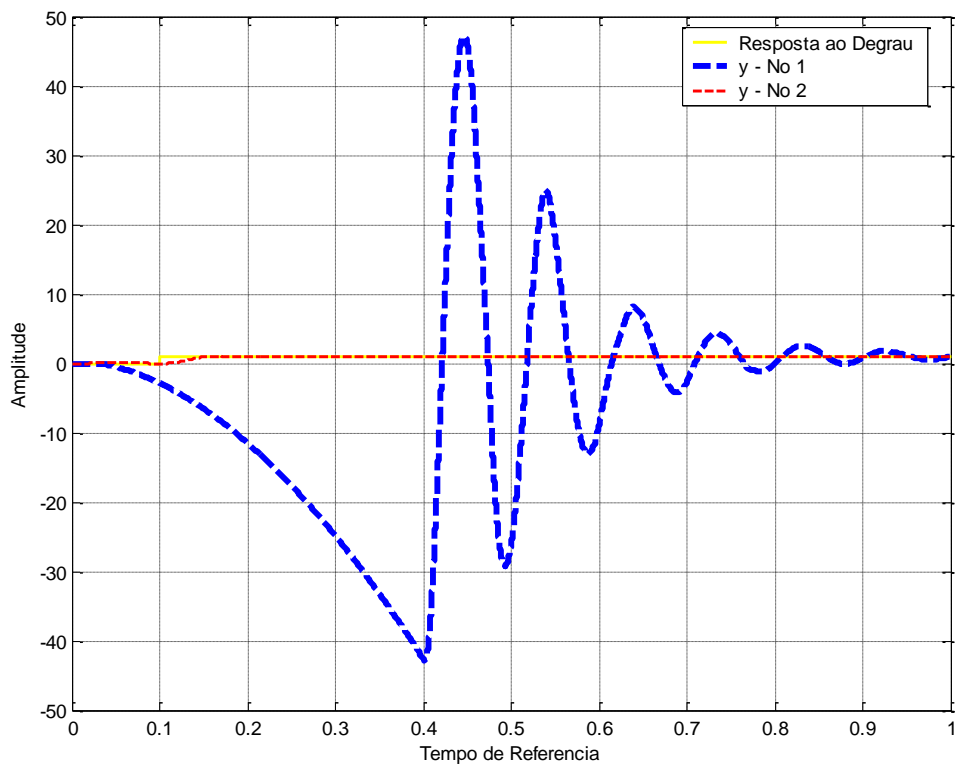


Figura L.13 – Resposta ao Degrau Sincronizada: Viés inicial:0.39(s), Deriva:10% - AI.

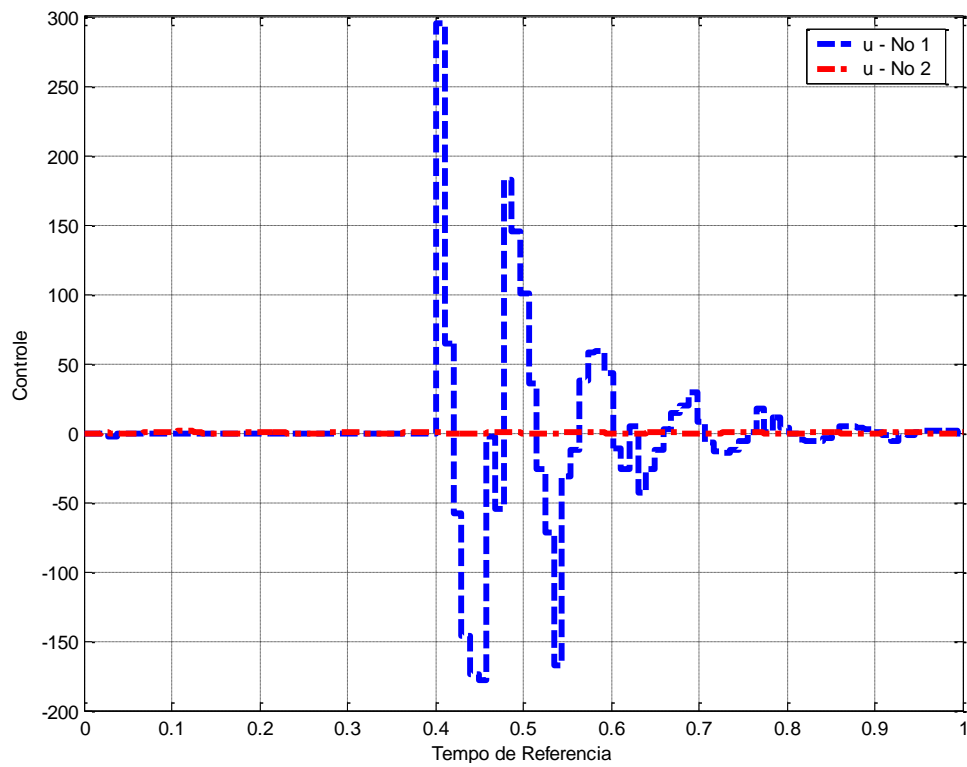


Figura L.14 – Lei de Controle Sincronizada: Viés inicial: 0.39(s), Deriva: 10% - AI.

Através das Figuras L.13 e L.14 é possível ver que a sincronização de relógios prejudicou mais ainda o sistema de controle. Isto ocorre pois o algoritmo FTM é feito principalmente para corrigir efeitos de deriva e supõe que todos os relógios do conjunto estão sincronizados inicialmente. O aumento de erro ocorre pois o algoritmo insere um erro de estado entre 0 e 0.39 segundos, isto é, o relógio fica parado. Após esse período que o sistema volta a atuar. Neste caso o algoritmo não mostrou-se eficiente corrigindo o erro de viés inicial, mas mostrou-se eficiente corrigindo o erro de deriva presente no sistema.



## APÊNDICE M – SIMULAÇÃO DO 4º CASO – ARQUITETURA INTEGRADA – REDE ETHERNET

Neste caso, foi feita refeita a simulação de um sistema de controle com arquitetura integrada, igual ao 1º Caso. A sincronização foi feita através do algoritmo FTM (*Welch-Lynch*), com arquitetura distribuída, isto é, não existe um relógio de referência. Nesta simulação variou-se o parâmetro de deriva do relógio do nó 2 (Sensor da Malha 1 de Controle por Rede). A rede de comunicação foi feita por um protocolo de comunicação com acesso ao meio **Ethernet**. A resposta ao degrau e Lei de controle ideais podem ser observadas na Figura 4.42 e 4.43.

Foi realizado o mesmo experimento do 1º Caso. No **caso A**, com **taxa de deriva de -10%** os resultados da sincronização com Ethernet foram:

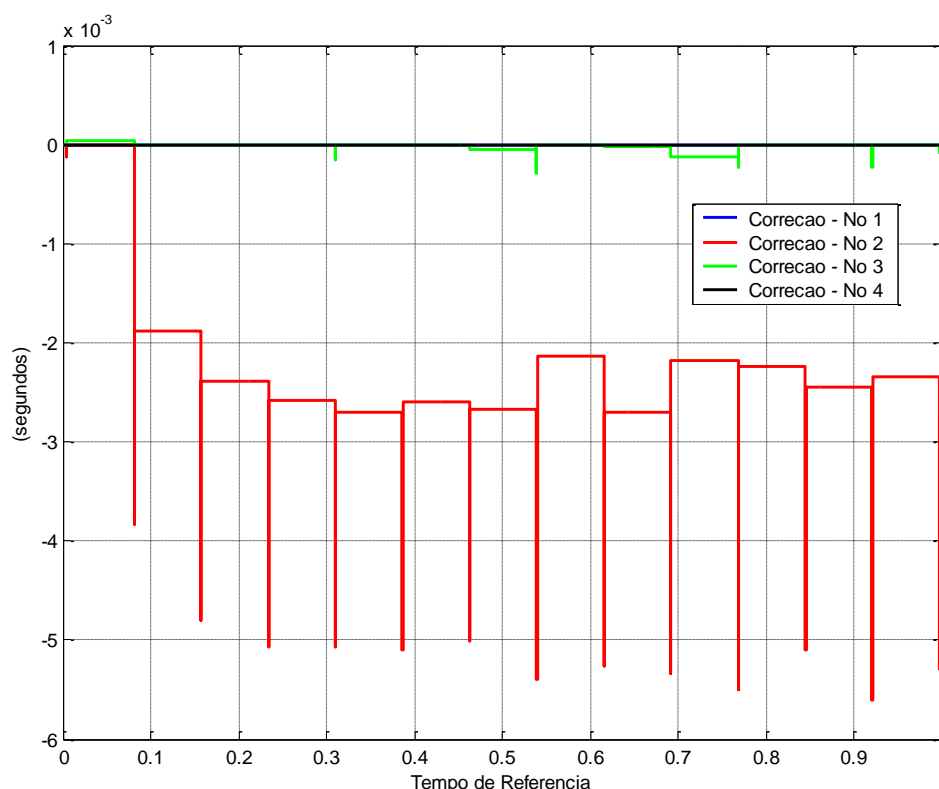


Figura M.1 –Correção com Rede Ethernet, Deriva -10% - AI.

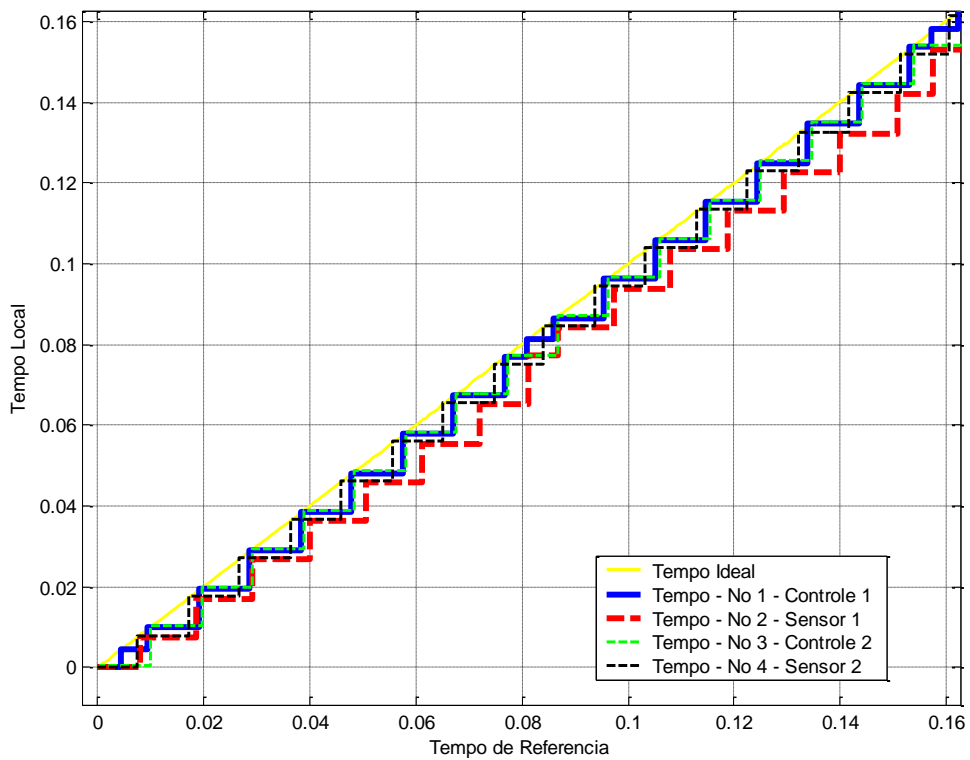


Figura M.2 – Linha do Tempo corrigida - AI.

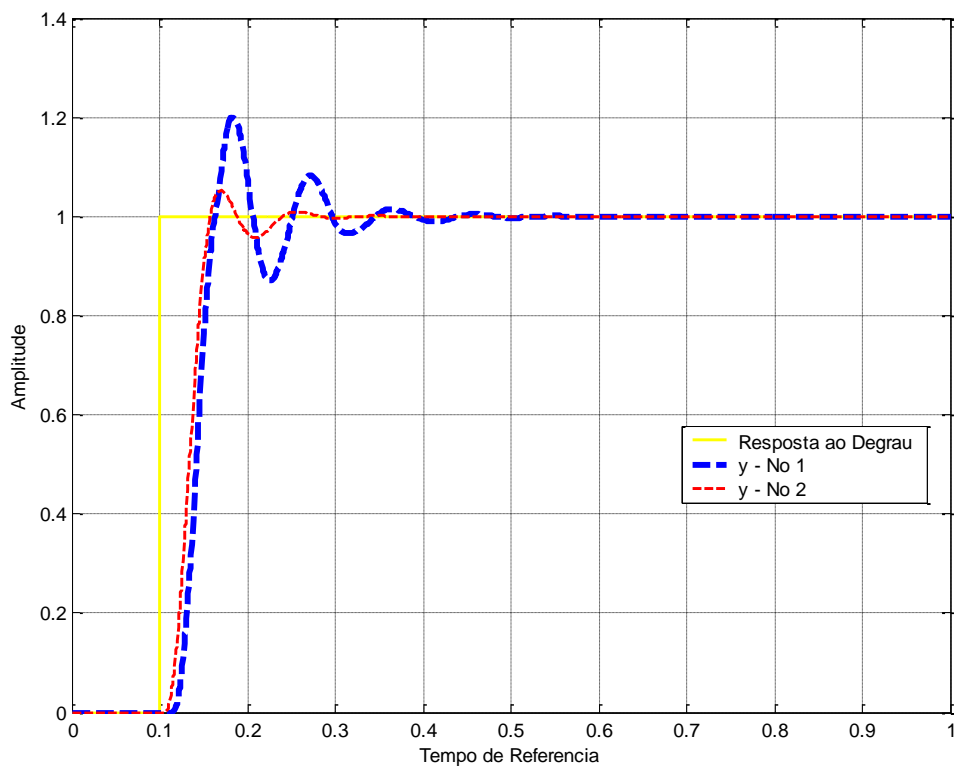
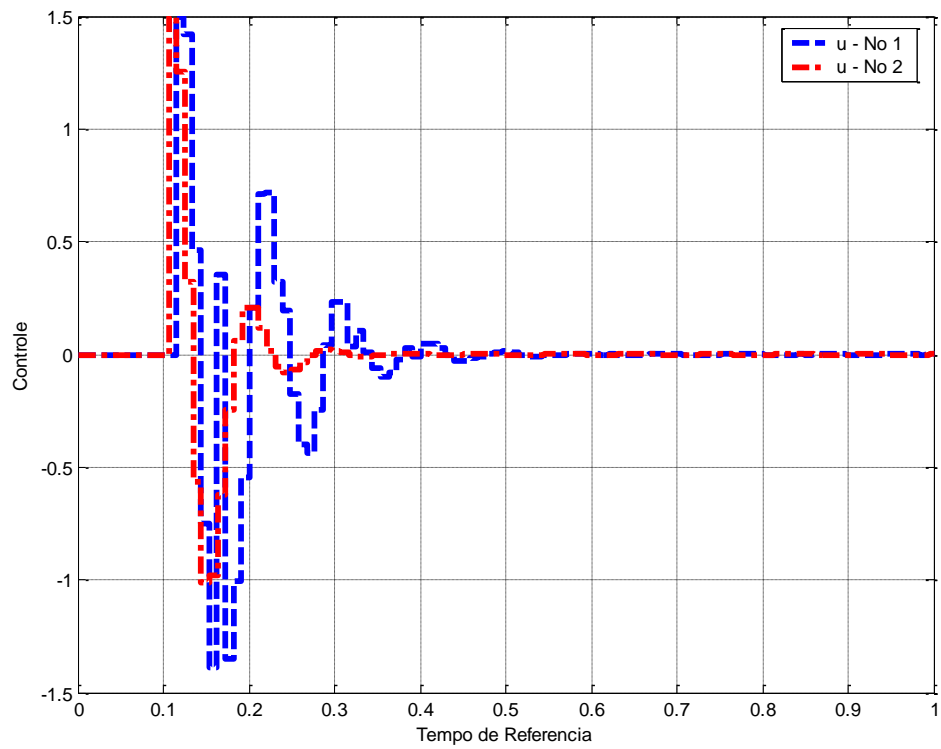


Figura M.3 –Resposta ao Degrau, Deriva -10% - AI.

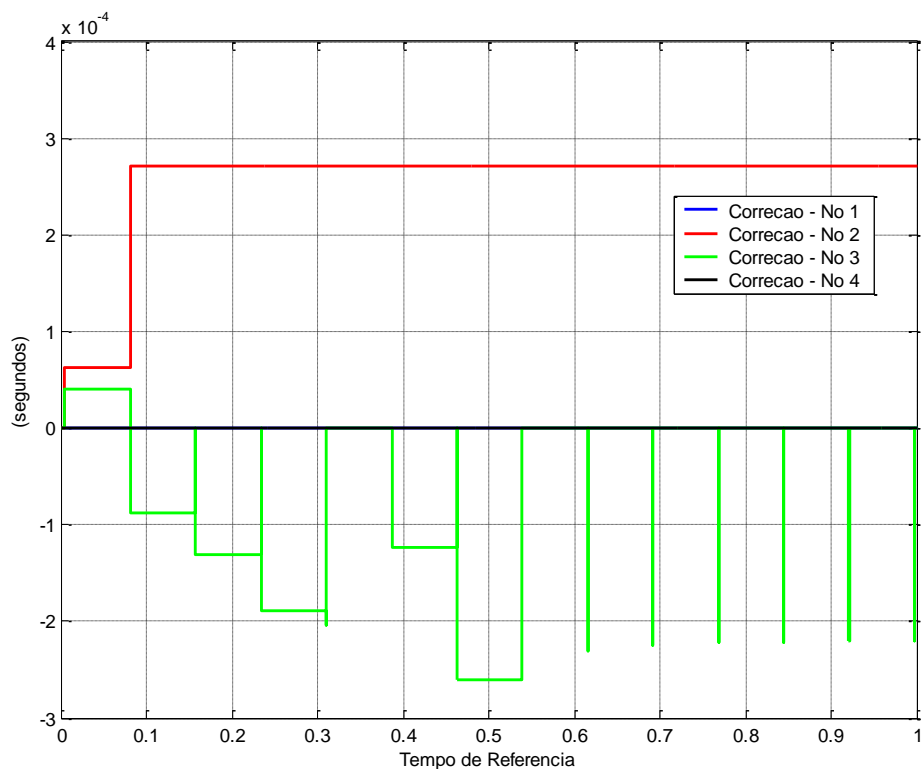




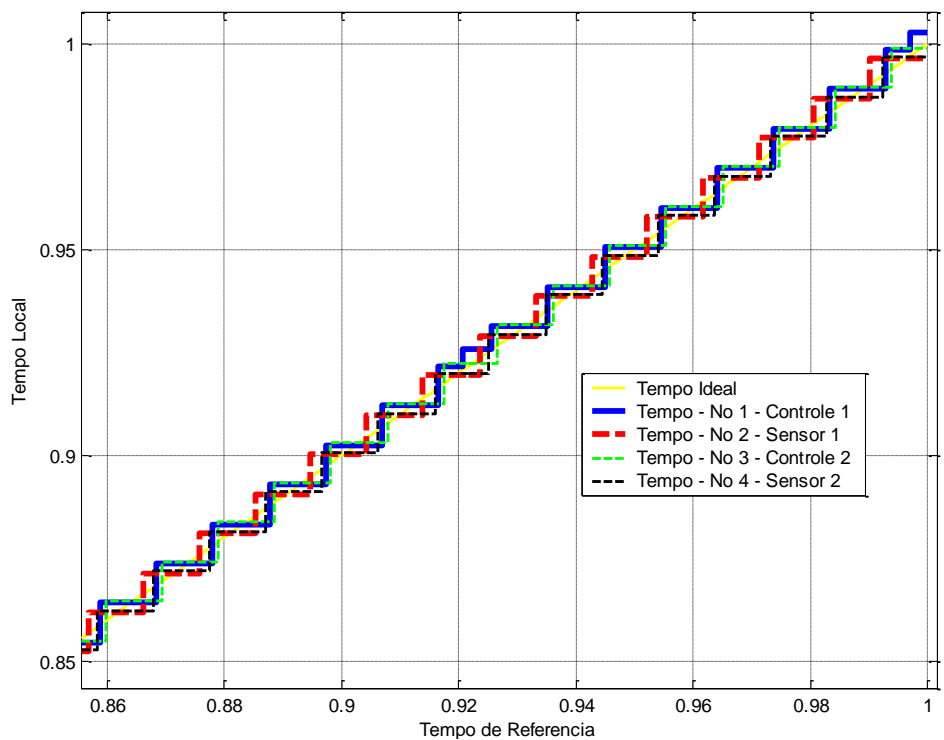
(b)

Figura M.4 – Deriva -10% - Lei de Controle - AI.

No **caso B**, com **taxa de deriva de 1%** os resultados da sincronização com Ethernet foram:

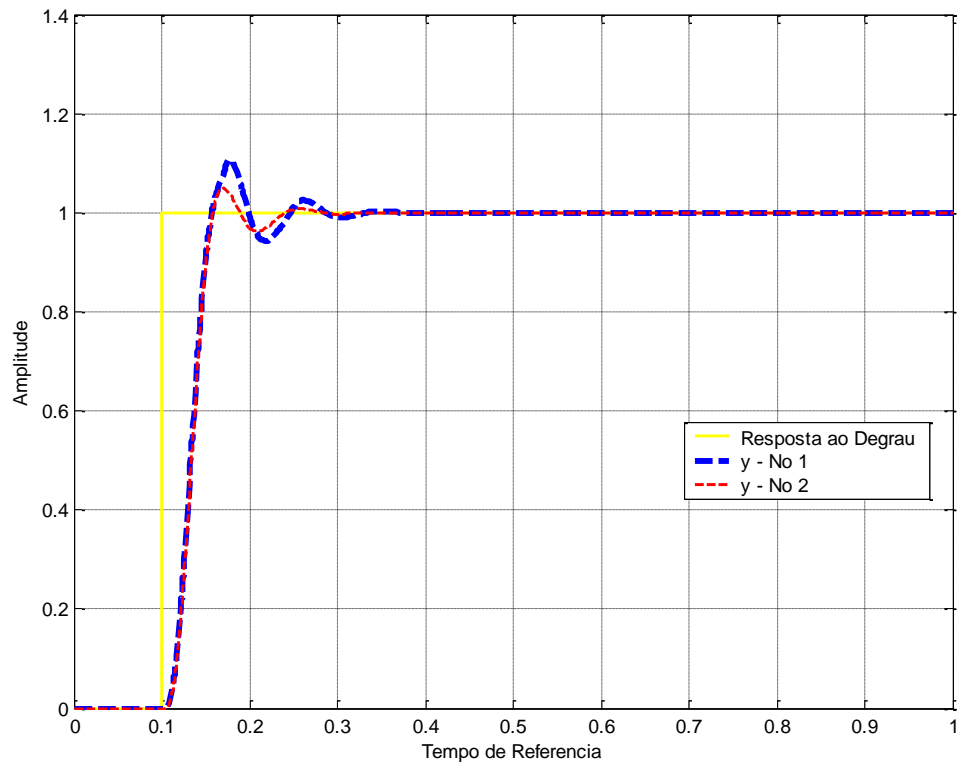


(a)

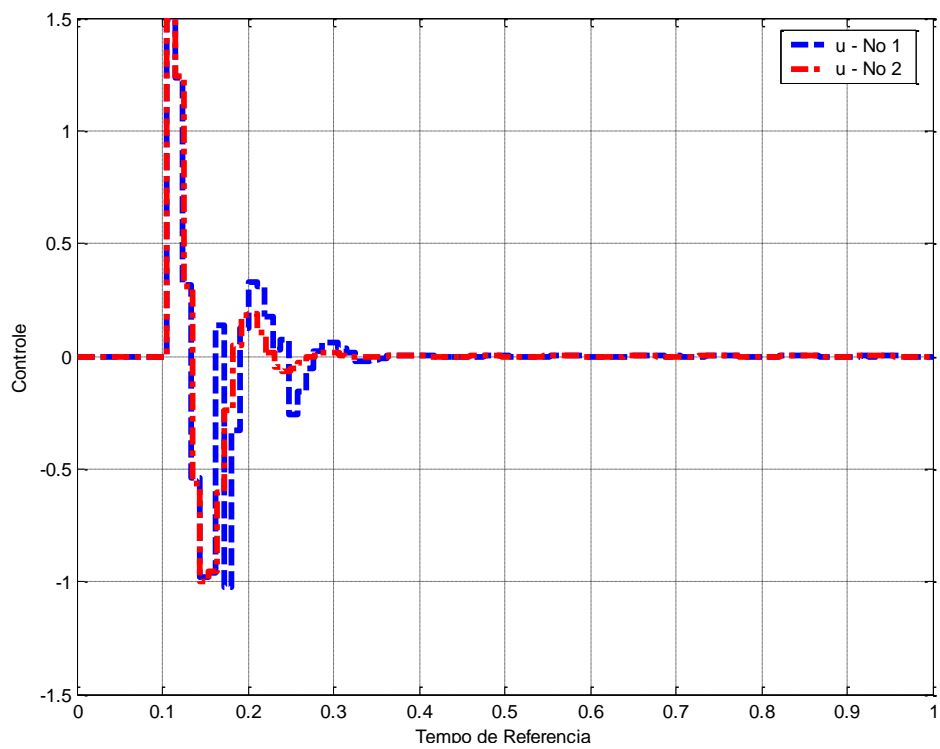


(b)

Figura M.5 – (a) Correção: Rede Ethernet, Deriva 1% (b) Linha do Tempo - AI.



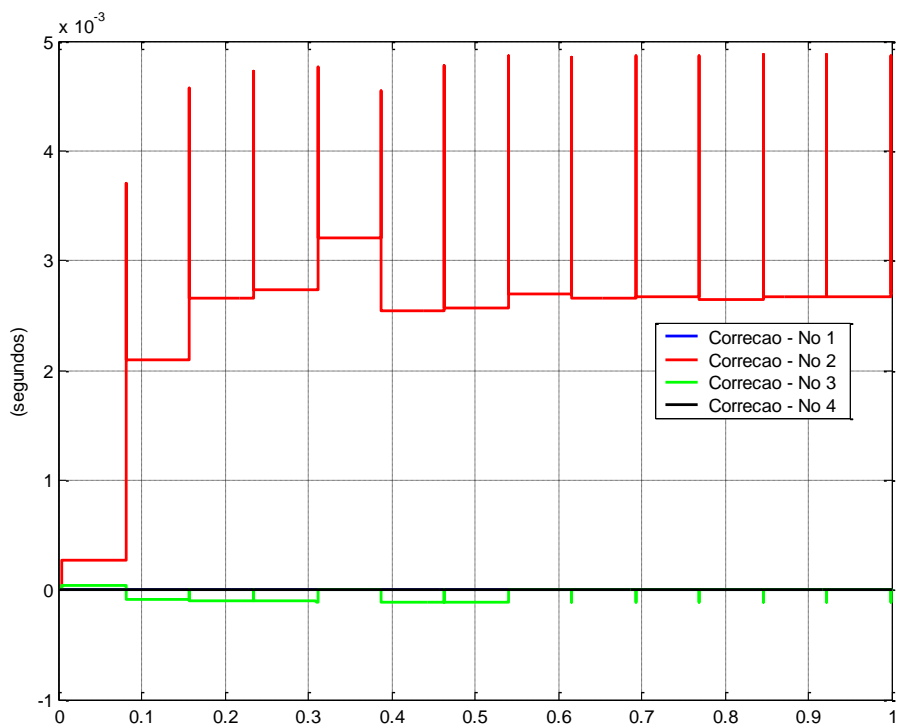
(a)



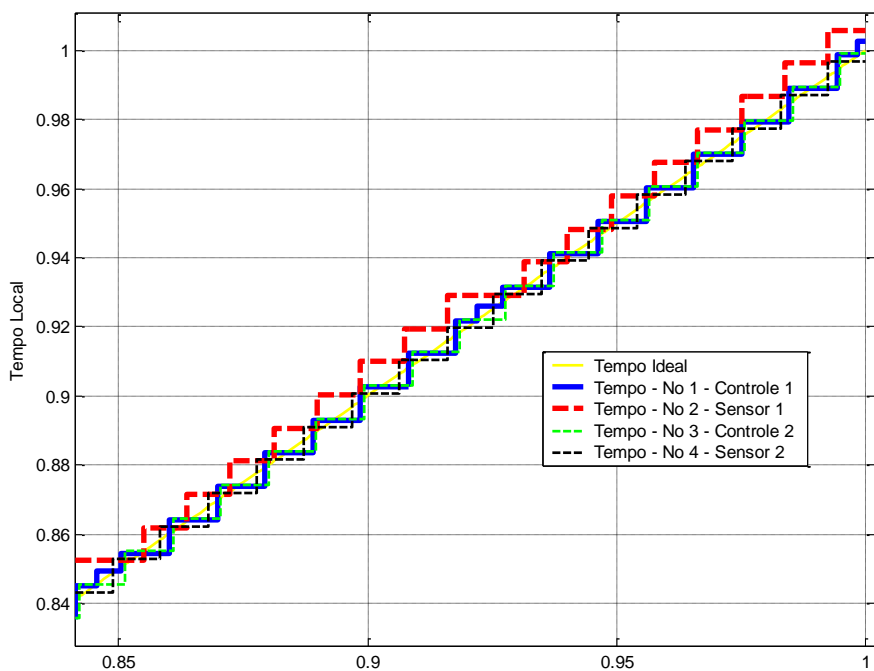
(b)

Figura M.6 – (a) Resposta ao Degrau, Deriva 1% (b) Lei de Controle - AI.

No caso C, com taxa de deriva de 10% os resultados da sincronização com Ethernet foram:

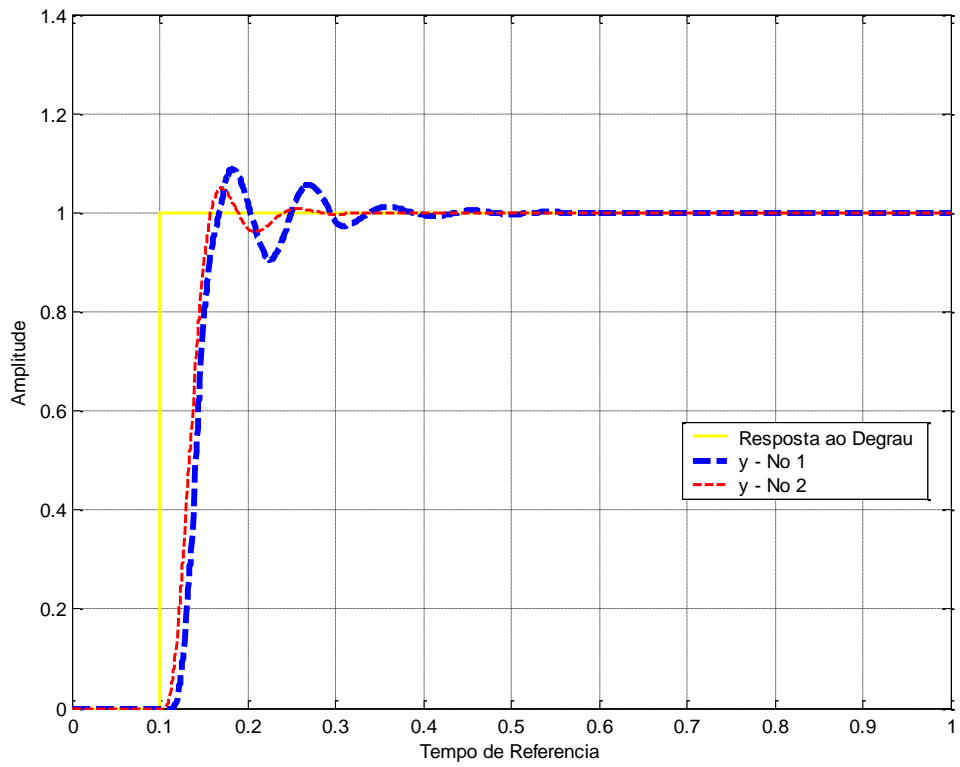


(a)

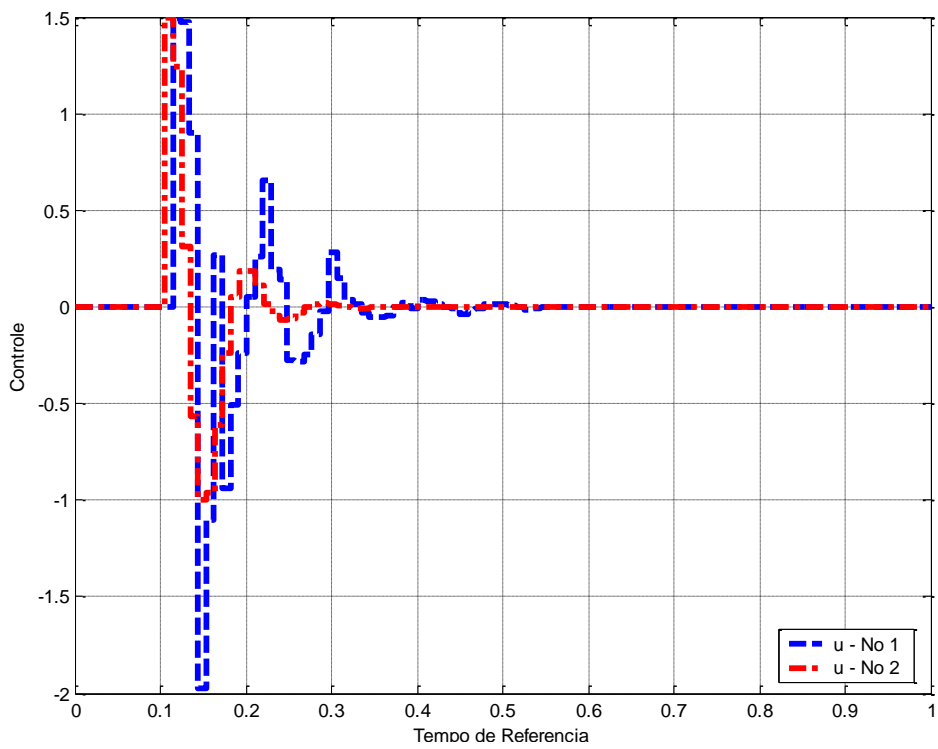


(b)

Figura M.7 – (a) Correção: Rede Ethernet, Deriva 10% (b) Linha do Tempo - Al.



(a)



(b)

Figura M.8 – (a) Resposta ao Degrau, Deriva 10% (b) Lei de Controle - AI.

Em todos os casos a taxa de deriva o algoritmo FTM com rede Ethernet teve um desempenho pior, pois sua precisão ficou ligeiramente maior. E com isso obteve na resposta ao degrau e na lei de controle resultados um pouco piores que comparado com a rede TDMA.

## APÊNDICE N – SIMULAÇÃO DO 5º CASO – ARQUITETURA INTEGRADA – REDE ETHERNET

No **caso A**, com **viés inicial: -0.2(s)**, os resultados da sincronização com Ethernet foram:

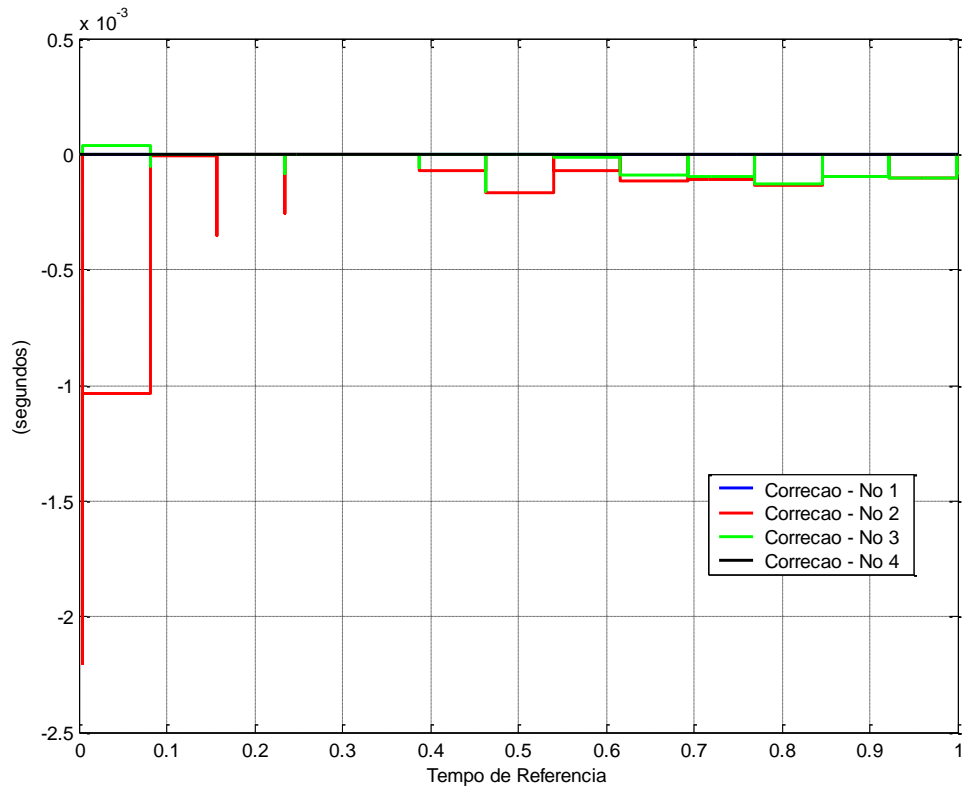


Figura N.1 – Correção: Rede Ethernet, Viés -0.2s -AI.

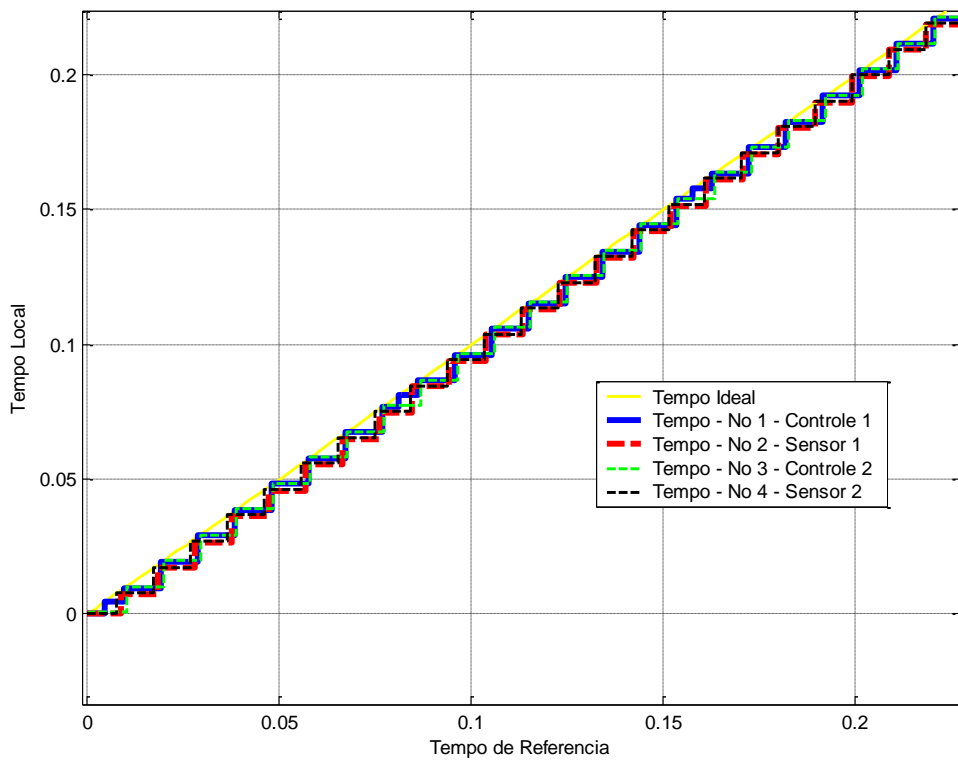


Figura N.2 – Linha do Tempo: Rede Ethernet, Viés -0.2s - AI.

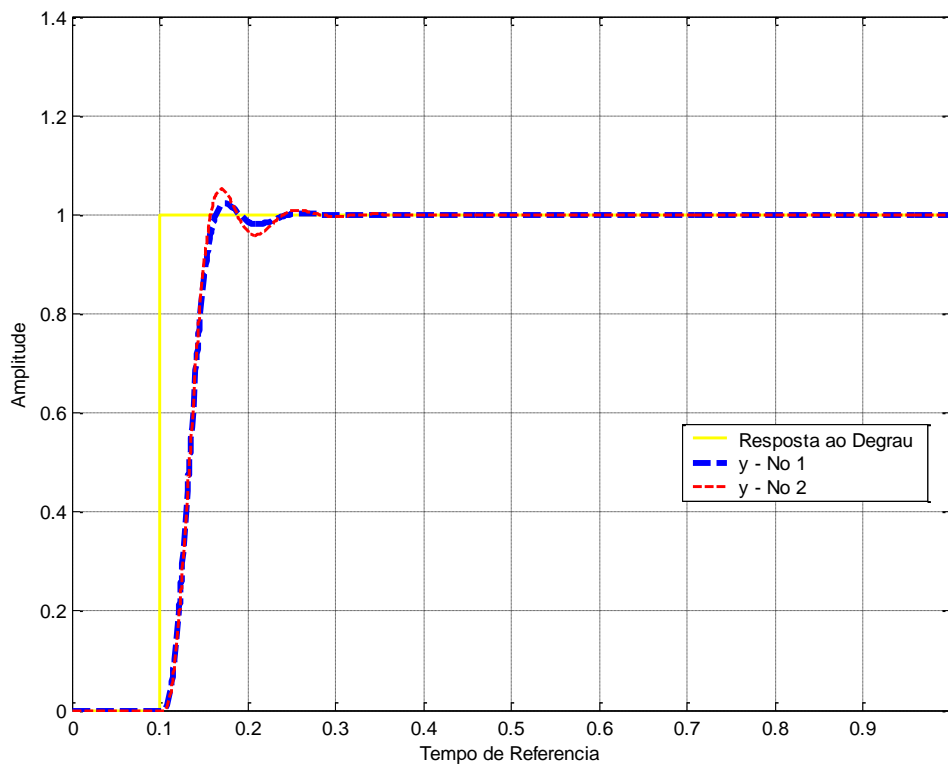


Figura N.3 – Resposta ao Degrau, Ethernet - Viés -0.2s - AI.



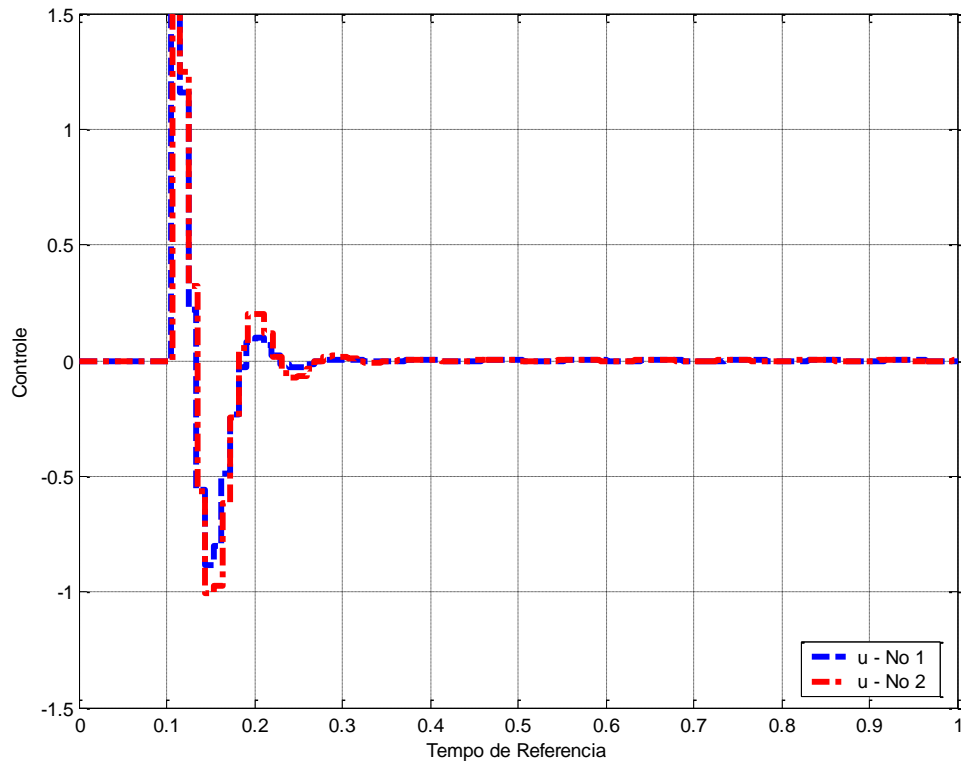


Figura N.4 – Lei de Controle: Ethernet - Viés -0.2s - AI.

No **caso B**, com **viés inicial: 0.39(s)**, os resultados da sincronização com Ethernet foram:

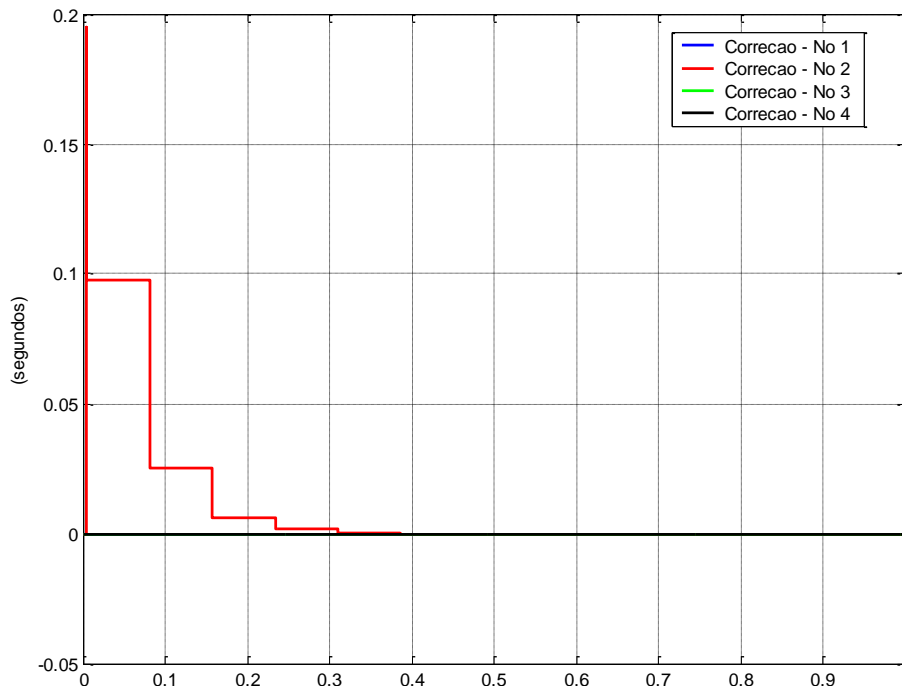


Figura N.5 – Correção: Rede Ethernet, Viés 0.39s -AI.

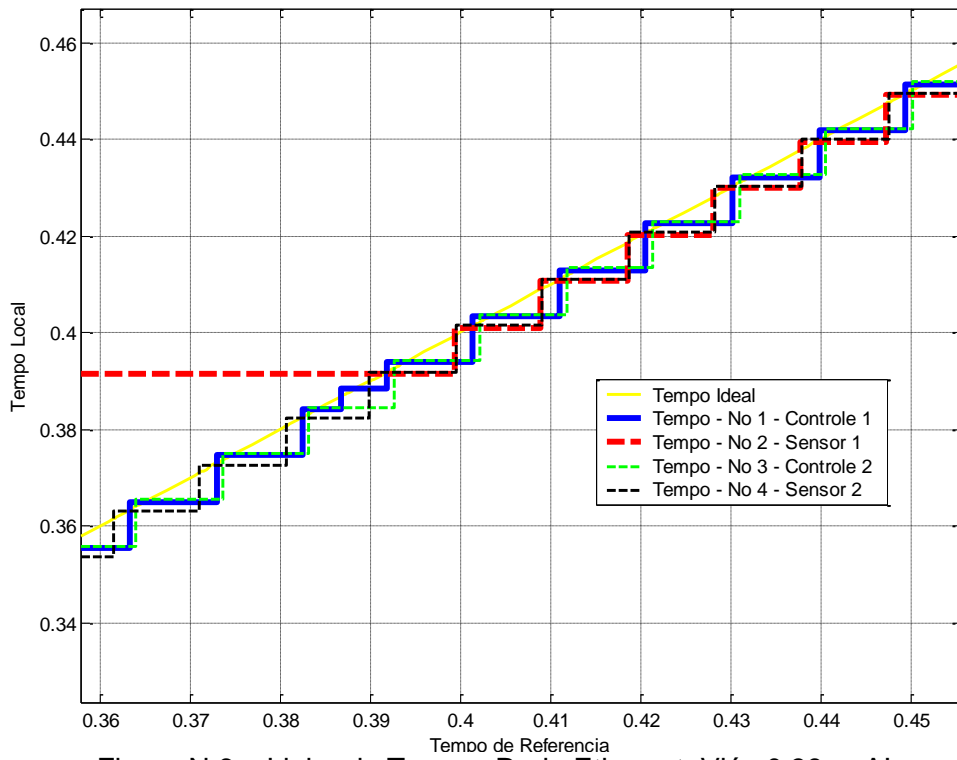


Figura N.6 – Linha do Tempo: Rede Ethernet, Viés 0.39s - AI.

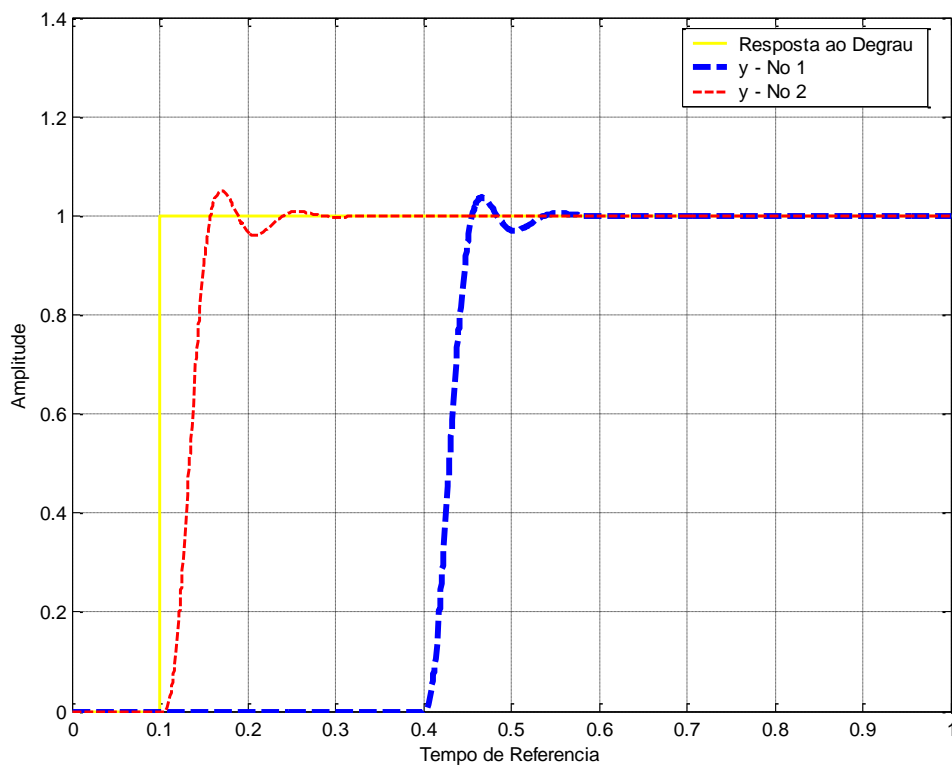


Figura N.7 – Resposta ao Degrau, Ethernet - Viés 0.39s - AI.

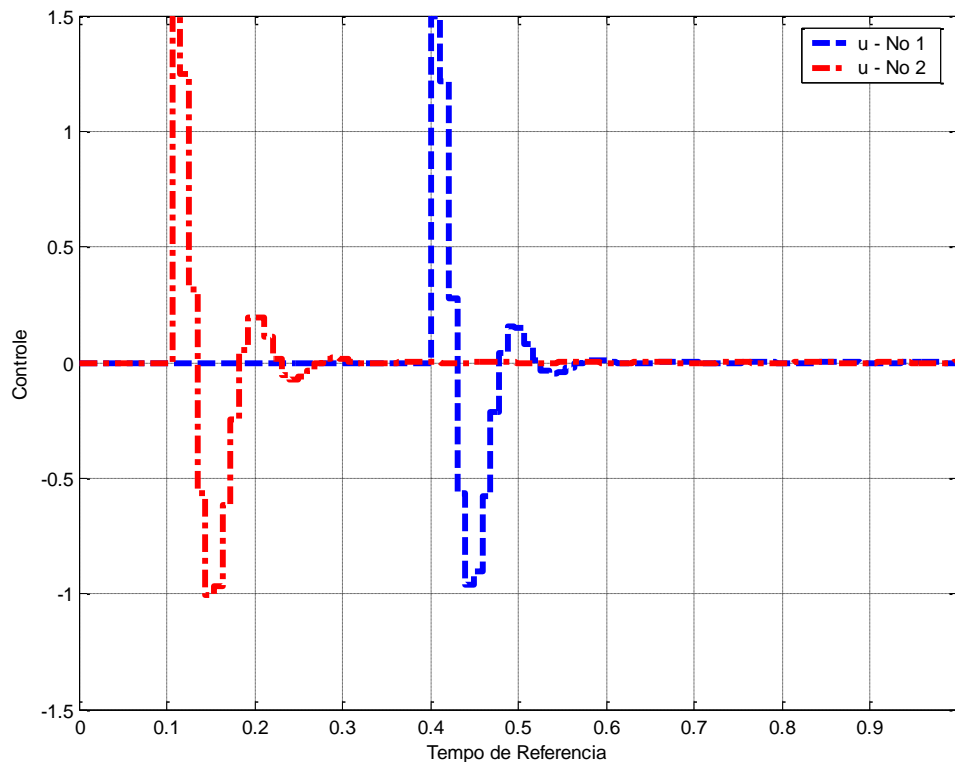


Figura N.8 – Lei de Controle: Ethernet - Viés 0.39s - AI.

No caso onde o viés inicial é -0.2 segundos o algoritmo FTM com rede Ethernet teve um desempenho praticamente idêntico ao com rede TDMA.

Com o viés em 0.39 segundos, o algoritmo FTM com rede Ethernet teve o mesmo desempenho, mas a lei de controle e a resposta ao degrau modificaram. Na rede TDMA a resposta ao degrau e a lei de controle amplificaram suas amplitudes. Na rede Ethernet as amplitudes não se modificaram mas a resposta ao degrau e a lei de controle se atrasaram de 0.39 segundos.



## APÊNDICE O – SIMULAÇÃO DO 6º CASO – ARQUITETURA INTEGRADA – REDE ETHERNET

No **caso A**, com **viés inicial: -0.2(s)** e **deriva: -10%**, os resultados da sincronização com Ethernet foram:

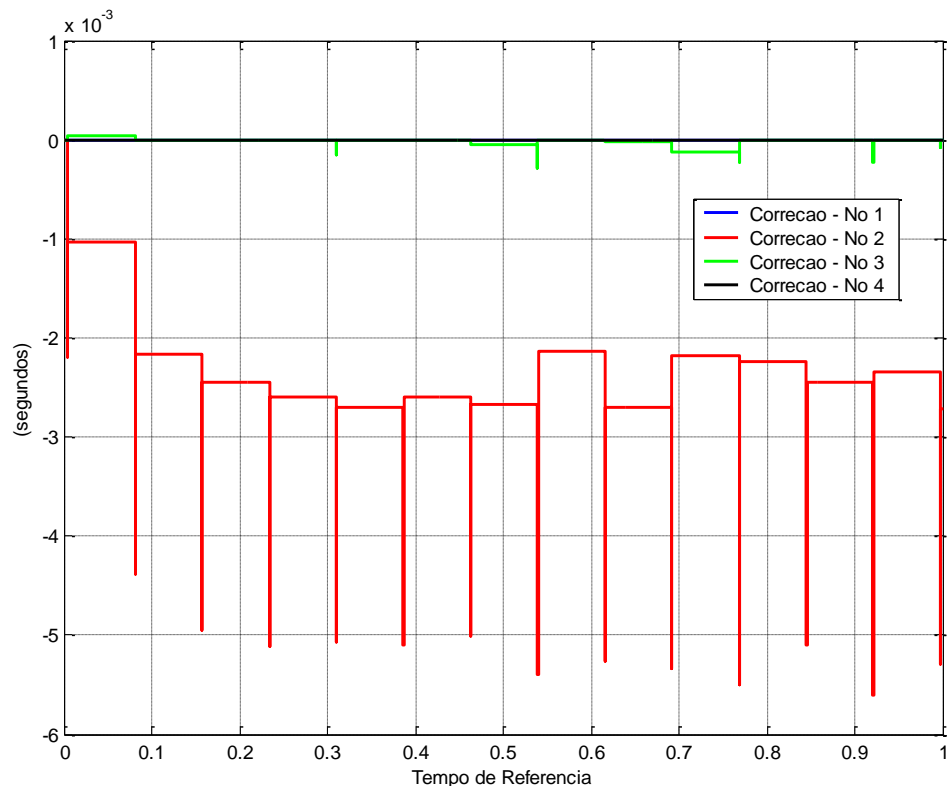


Figura O.1 – Correção: Rede Ethernet, Viés -0.2s, Deriva: -10% -AI.

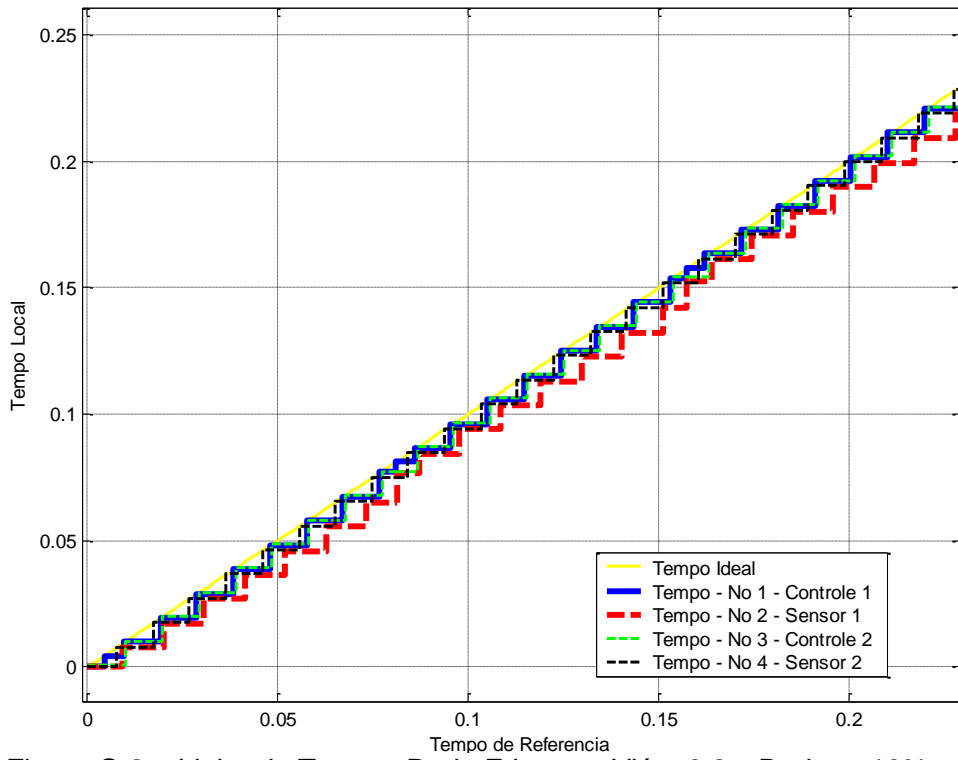


Figura O.2 – Linha do Tempo: Rede Ethernet, Viés -0.2s, Deriva: -10% - AI.

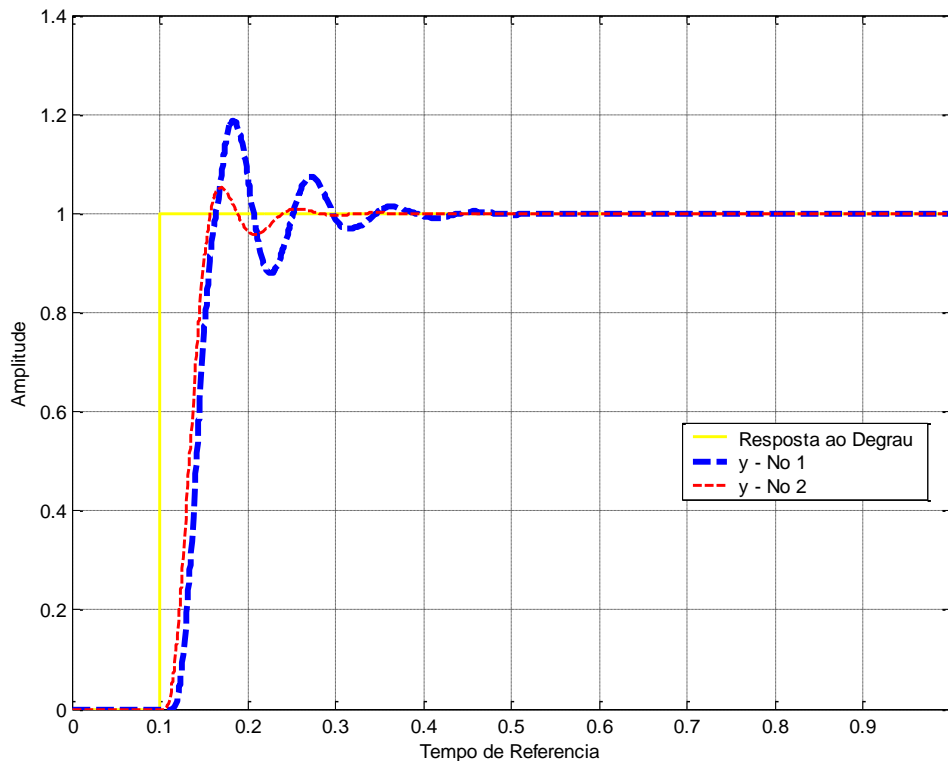


Figura O.3 – Resposta ao Degrau, Ethernet - Viés -0.2s, Deriva: -10% - AI.

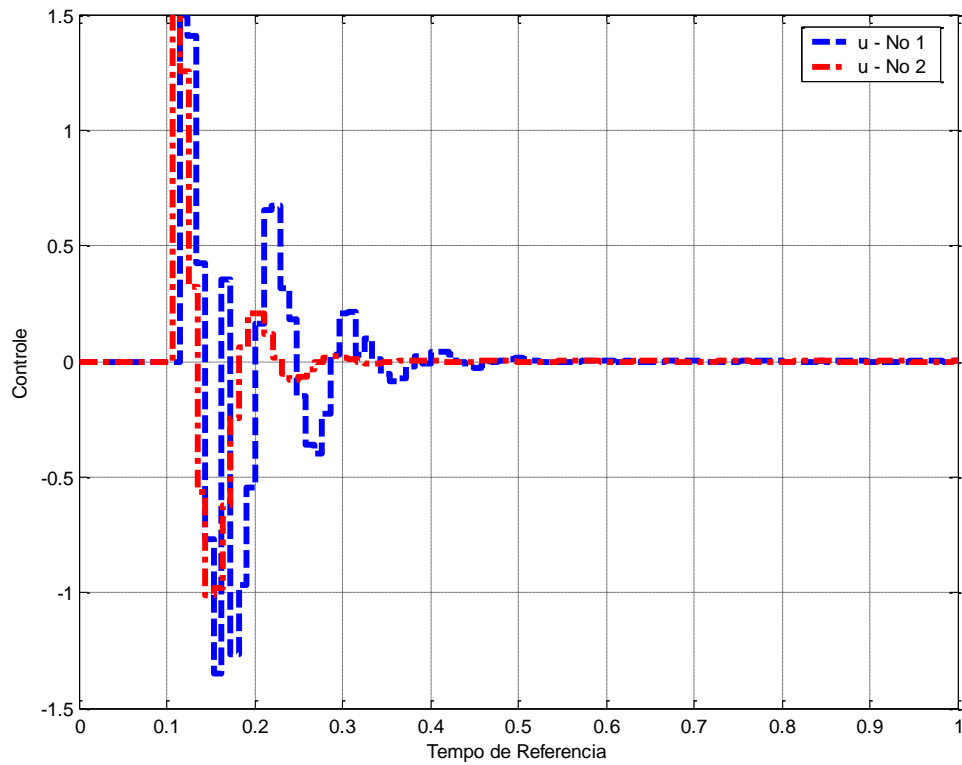


Figura O.4 – Lei de Controle: Ethernet - Viés -0.2s, Deriva: -10% -AI.

No **caso B**, com **viés inicial: 0.39(s)** e **Deriva 10%**, os resultados da sincronização com Ethernet foram:

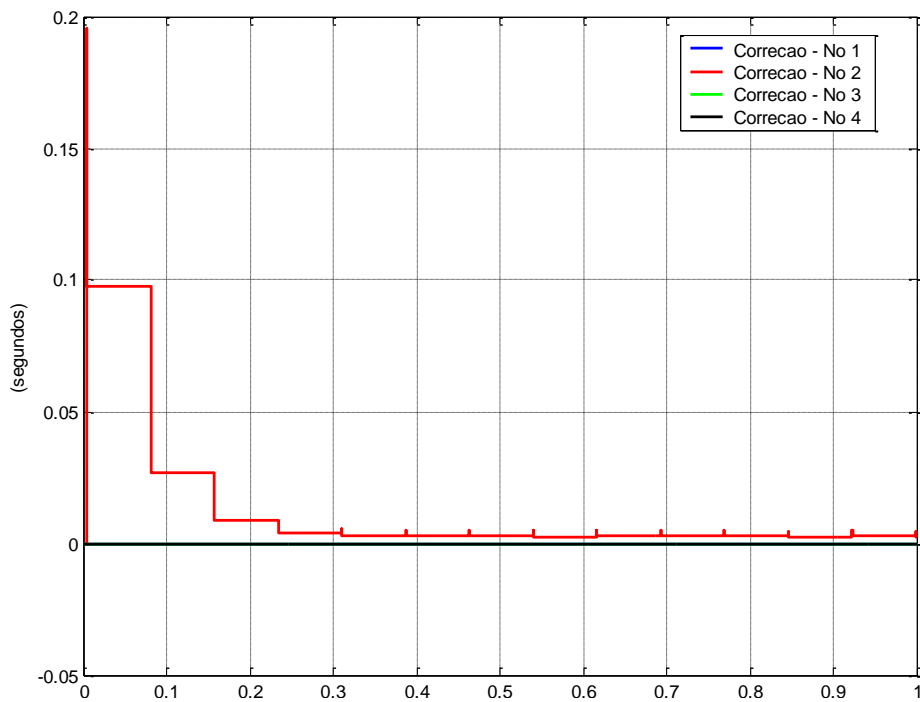
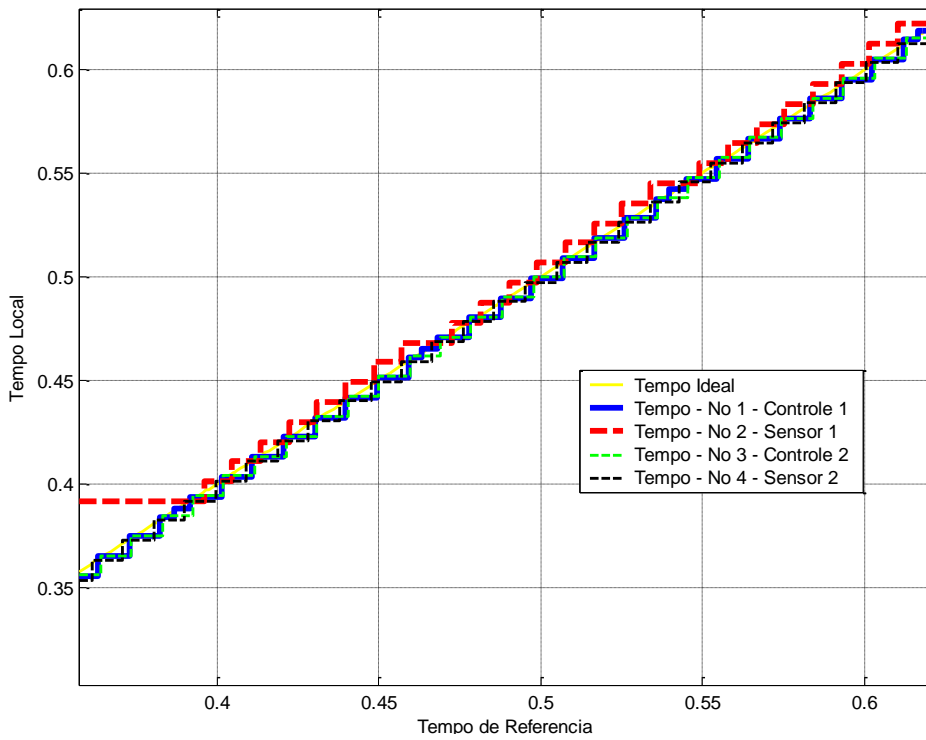


Figura O.5 – Correção: Rede Ethernet, Viés 0.39s, Deriva: 10% -AI.



Fi

Figura O.6 – Linha do Tempo: Rede Ethernet, Viés 0.39s, Deriva: 10% - AI.

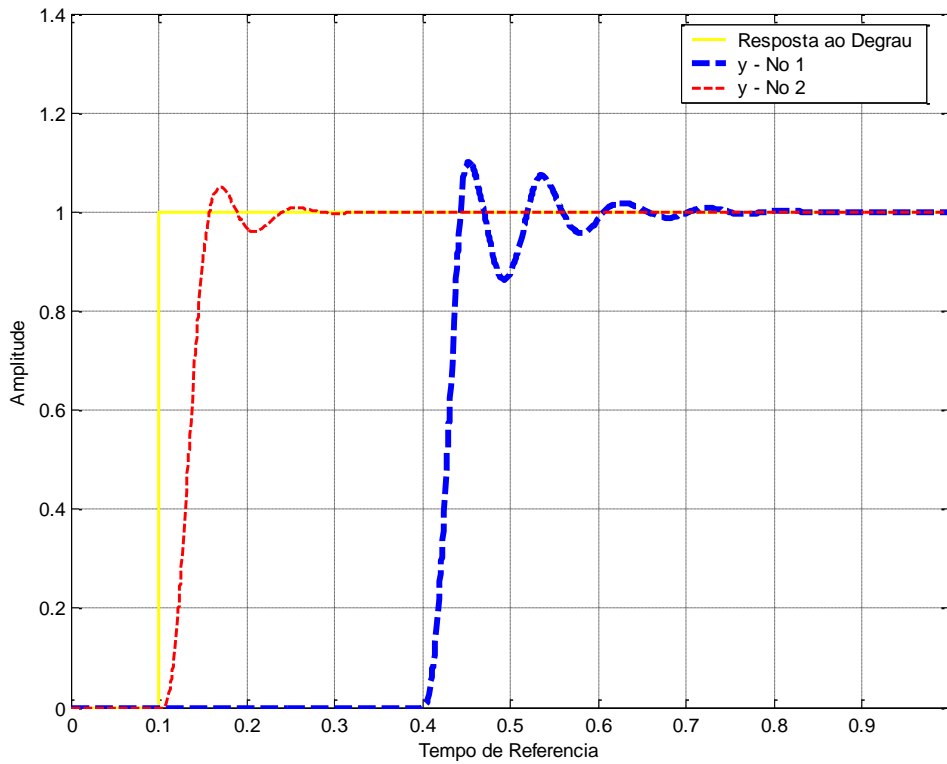


Figura O.7 – Resposta ao Degrau, Ethernet - Viés 0.39s, Deriva: 10% - AI.



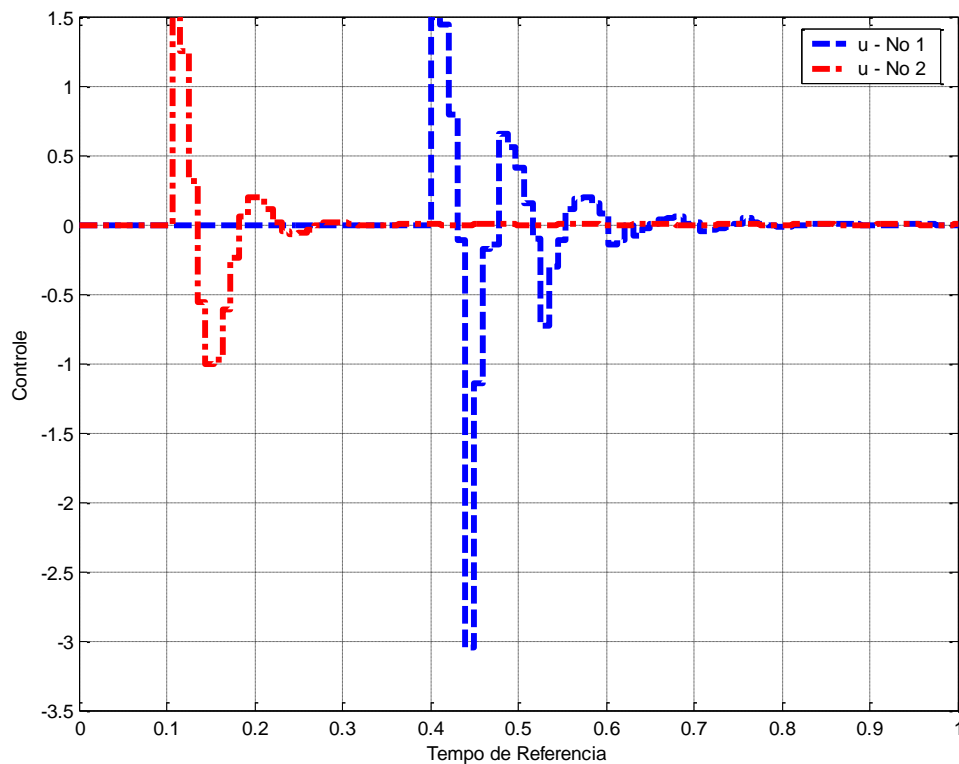


Figura O.8 – Lei de Controle: Ethernet - Viés 0.39s, Deriva: 10% - AI.

No caso onde o viés inicial é -0.2 segundos e a deriva -10% o algoritmo FTM com rede Ethernet teve um desempenho praticamente idêntico ao com rede TDMA.

Com o viés em 0.39 segundos e deriva 10%, o algoritmo FTM com rede Ethernet teve o mesmo desempenho, mas a lei de controle e a resposta ao degrau modificaram. Na rede TDMA a resposta ao degrau e a lei de controle amplificaram suas amplitudes chegando a instabilizar o controle. Na rede Ethernet as amplitudes não se amplificaram tanto, mas a resposta ao degrau e a lei de controle se atrasaram de 0.39 segundos, mas a lei de controle se manteve estável.



## APÊNDICE P – SIMULAÇÃO DO ALGORITMO FTM/FTA

Os algoritmos FTM e FTA são muito parecidos, com exceção de suas funções de convergência, conforme dito anteriormente. Por isto, neste trabalho, foi feita uma simulação, utilizando-se do programa Microsoft Excel, para mostrar que os algoritmos provêm resultados diferentes na convergência e precisão final alcançada. Através destas simulações é possível mostrar que os algoritmos são diferentes, além de prover um maior conhecimento do processo de sincronização de relógios.

Nesta simulação, através da comparação do 1º e 3º casos é possível verificar a influência que o atraso da rede de comunicação proporciona ao resultado da simulação. É possível também ver a robustez do algoritmo na presença de falhas maliciosas, tais com erro na taxa de deriva (*drift rate*) e erro no estado do relógio (*state rate*).

Tabela P.1 – Tabela de Casos do Experimento dos Algoritmos FTA e FTA.

	Num. de Relógios (n)	Num. de Relógios com Falha (f)	Atraso de Rede (segundos)	Resultado
<b>3º Caso</b>	4	Relógio 1 com erro em deriva	0.5	FTM = FTA
<b>4º Caso</b>	4	Relógio 1 com erro de estado	0,005	FTM = FTA
<b>5º Caso</b>	5	Relógio 1 com erro em deriva	0.005	FTM = FTA
<b>6º Caso</b>	6	Relógio 1 com erro em deriva	0.005	FTM = FTA
<b>7º Caso</b>	7	Relógio 1 e 2 com erro de estado	0.005	FTM ≠ FTA

### **Caso 3 – 4 Relógios, com variação do atraso de rede**

Neste caso, foi reproduzido novamente o 1º Caso, mas dessa vez modificando somente o parâmetro de atraso da rede.

Dados do Experimento:

- Número de Relógios:  $n = 4$ ;
- Número de erros bizantinos:  $f = 1$ ;
- Taxa de deriva:  $\rho = 0.2$ ;
- Atraso de Rede:  $\delta = 0.5$  (s);

Neste 3º caso, também foi aplicado ao relógio T1, um erro de taxa de deriva de 0.2 em relação aos outros relógios. O relógio T1 é o relógio com falha de deriva. Os relógios T2, T3 e T4 não estão com falha.

A Tabela C.2, encontrada no Apêndice C, mostra os dados do experimento, realizado com o auxílio do Microsoft Excel. As colunas T1, T2, T3 e T4 são as colunas de propagação dos relógios. As colunas Delta1, Delta2, Delta3 e Delta4 são as colunas aonde a diferença dos relógios é realizada. Este caso segue a mesma seqüência do processo do 1º Caso.

A Figura P.1 mostra o graficamente o resultado da simulação da Tabela C.2 com aplicação do algoritmo de sincronização de relógios.

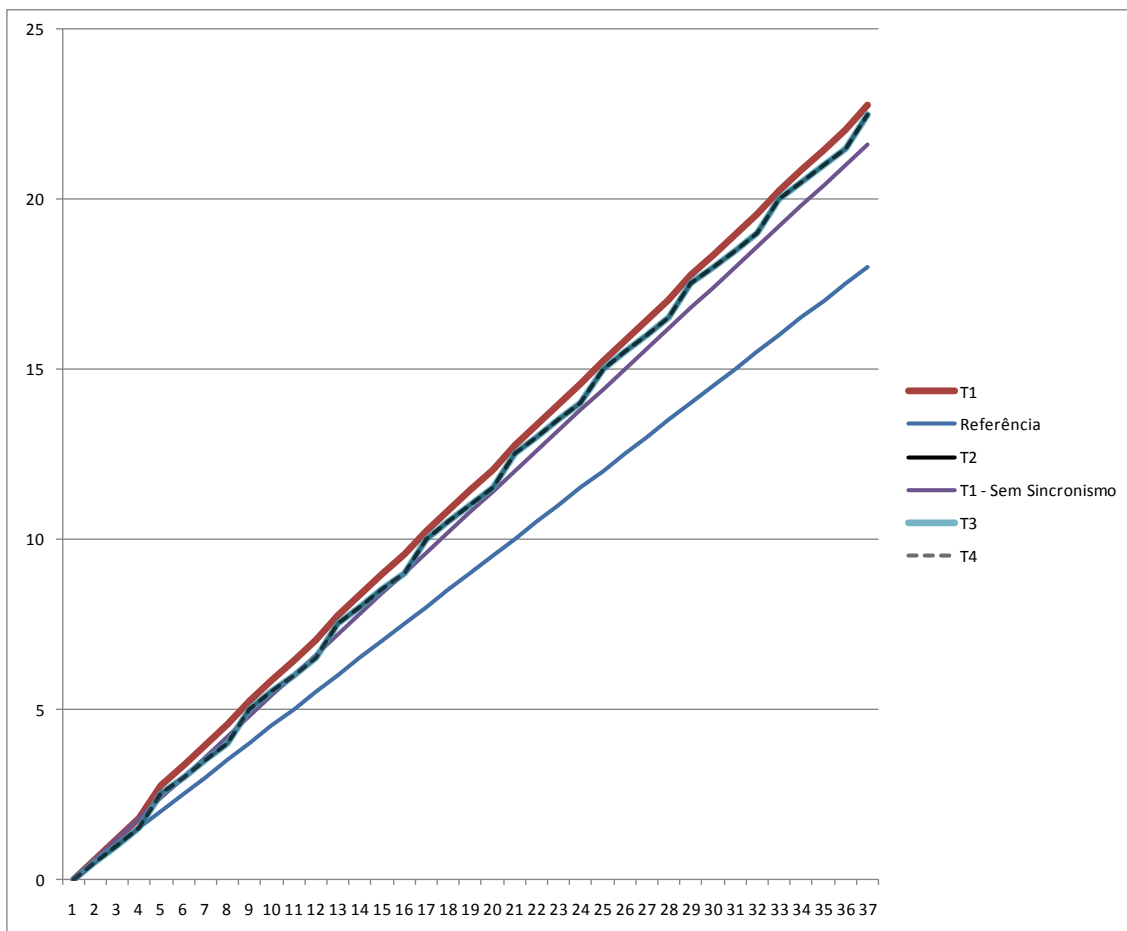


Figura P.1 – FTM/FTA – 3º Caso Com Sincronismo.

Este caso se mostrou muito interessante, pois é possível ver que os relógios do conjunto (T1, T2, T3 e T4) estão sincronizados entre si como no primeiro caso, mas com a diferença de que o conjunto se deriva (*drift*) em relação ao relógio de referência.

A Figura P.2 mostra o resultado da função de correção, isto é, mostra o gráfico de correção que o algoritmo aplica aos relógios.

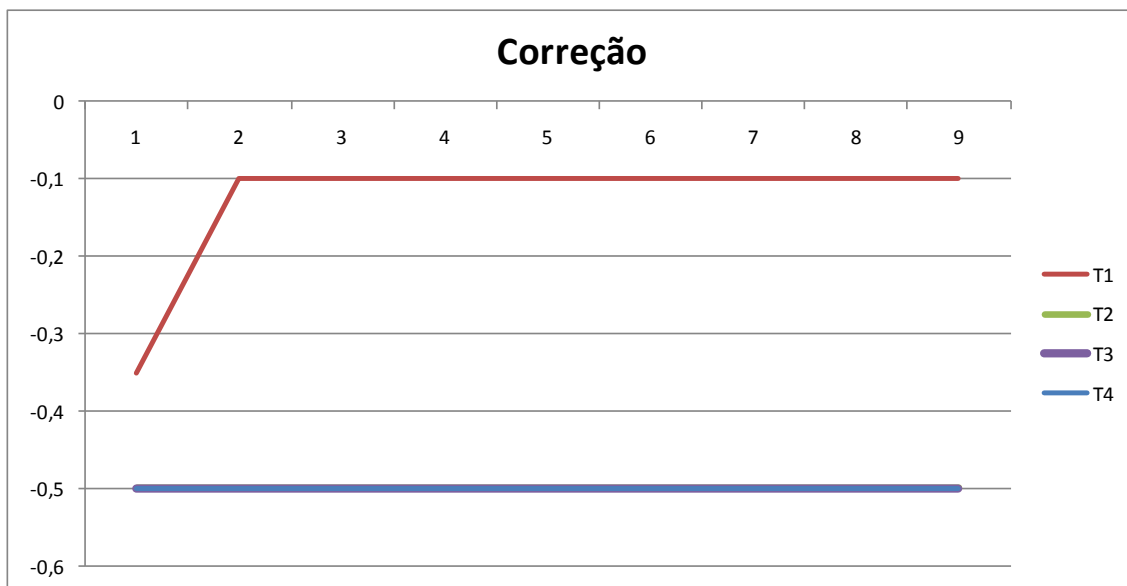


Figura P.2 – FTM/FTA – Correção do 3º Caso.

Neste gráfico da figura P.2 é possível ver que o algoritmo FTA/FTM convergiu a valores diferentes do 1º caso, Figura 4.1. Este valor de correção indica que a cada ciclo, o relógio T1 precisa sofrer uma correção em torno de -0.1 (s) e os relógios T2, T3, T4 em torno de -0.5 (s). O interessante é que apesar deste caso sofrer modificações em sua correção e na convergência final, os valores de máximo viés instantâneo e precisão continuaram iguais.

Análise dos Dados:

- Máximo viés instantâneo, de acordo com a equação (2.33):  $\Gamma = 0.8$ ;
- Máximo viés instantâneo na simulação:  $\Gamma = 0.55$ ;
- Precisão, de acordo com a equação 2.50:  $\Pi = 0.8$ ;
- Precisão alcançada pela simulação:  $\Pi = 0.55$ ;

#### **Caso 4 – 4 Relógios, um com erro de estado**

Neste caso, foi reproduzido novamente o 1º Caso, mas dessa vez inserindo um erro de estado, ao invés de erro de deriva.

Dados do Experimento:

- Número de Relógios:  $n = 4$ ;
- Número de erros bizantinos:  $f = 1$ ;
- Erro de Estado,  $\rho = -1$ ;
- Atraso de Rede:  $\delta = 0.005$  (s);

Neste 4º caso, foi aplicado ao relógio T1, um erro de estado, isto é, o relógio T1 está parado. O relógio T1 é o relógio com falha de deriva. Os relógios T2, T3 e T4 não estão com falha. A Tabela C.3 mostra o modelo do experimento, realizado com o auxílio do Microsoft Excel. As colunas T1, T2, T3 e T4 são as colunas de propagação dos relógios. As colunas Delta1, Delta2, Delta3 e Delta4 são as colunas aonde a diferença dos relógios é realizada. Este caso segue a mesma seqüência do processo do 1º Caso e 3º Caso. A Figura P.3 mostra o graficamente o resultado da simulação da Tabela C.3 com aplicação do algoritmo de sincronização de relógios.

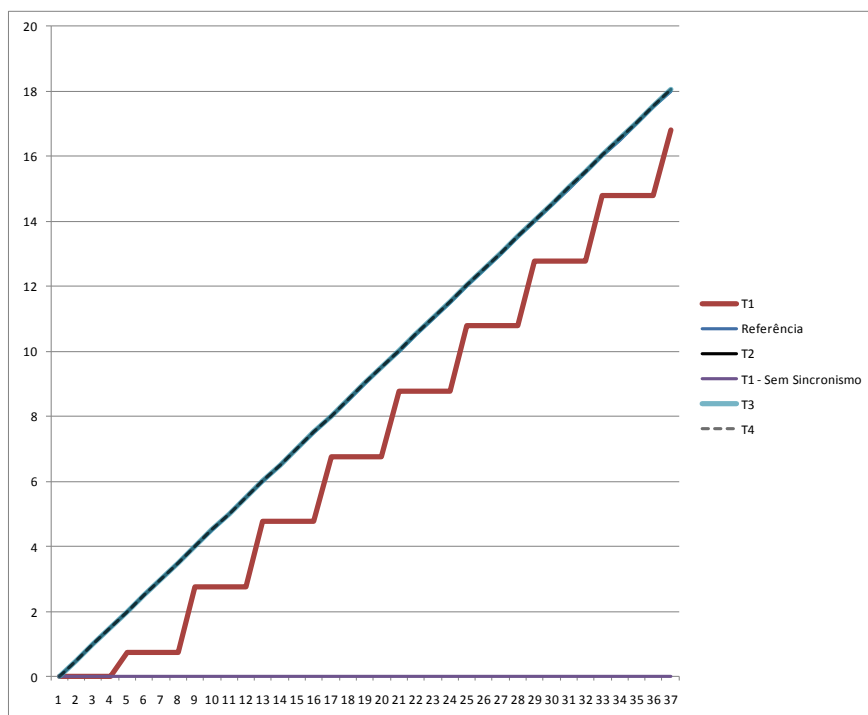


Figura P.3 – FTM/FTA – 4º Caso Com Sincronismo.

Neste caso, o relógio T1 está parado, mas mesmo assim foi aplicado a ele a correção dos relógios. Este caso é interessante, pois é possível ver que o algoritmo FTM/FTA foi robusto o bastante para isolar o problema causado pelo relógio T1 do resto dos relógios do conjunto (T2, T3 e T4) que continuam sincronizados entre si, pois sua convergência não se modificou. Somente o T1 foi prejudicado com o erro de estado. A Figura P.4 mostra o resultado da função de correção, isto é, mostra o gráfico de correção que o algoritmo aplica aos relógios.





Figura P.4 – FTM x FTA – Correção do 4º Caso.

Neste gráfico da figura P.4 é possível ver que o algoritmo FTA/FTM foi mantido os mesmos valores de convergência dos relógios T2, T3 e T4 em relação ao 1º caso, Figura 4.2. Somente o relógio T1 foi realmente prejudicado, modificando drasticamente seus valores de convergência. Neste caso os valores finais de máximo viés instantâneo e precisão sofreram modificações, conseqüentemente também em sua correção e na convergência final.

Análise dos Dados:

- Máximo viés instantâneo, de acordo com a equação (2.33):  $\Gamma = 0.8$ ;
- Máximo viés instantâneo na simulação:  $\Gamma = 2.75$ ;
- Precisão, de acordo com a equação 2.50:  $\Pi = 0.8$ ;
- Precisão alcançada pela simulação:  $\Pi = 2$ ;

Neste caso podemos concluir que o erro de estado causado no relógio T1, degrada e prejudica o relógio, atrapalha a função de convergência e prejudica a precisão final, mas em compensação isola os outros relógios do problema.

## Caso 5 – 5 Relógios, um com erro de deriva

Neste caso é possível verificar através das equações (3.6) e (3.7) que os algoritmos FTM e FTA são diferentes, pois suas funções de convergência são diferentes. O resultado esperado é que os algoritmos apresentem resultado final um pouco diferente.

Dados do Experimento:

- Número de Relógios:  $n = 5$ ;
- Número de erros bizantinos:  $f = 1$ ;
- Taxa de deriva:  $\rho = 0.2$ ;
- Atraso de Rede:  $\delta = 0.005$  (s);

Neste caso, foi aplicado ao relógio T1, um erro de taxa de deriva de 0.2 em relação aos outros relógios. O relógio T1 é o relógio com falha de deriva. Os relógios T2, T3, T4 e T5 não estão com falha. A Tabela C.4 mostra o modelo do experimento utilizando o algoritmo FTM e FTA, realizado com o auxílio do Microsoft Excel. As colunas T1, T2, T3, T4 e T5 são as colunas de propagação dos relógios. As colunas Delta1, Delta2, Delta3, Delta4 e Delta5 são as colunas aonde a diferença dos relógios é realizada, seguindo a equação (3.2). O ciclo é identificado pela coluna ciclo. A cada instante um relógio envia seus dados, como em uma comunicação TDMA. A coluna maior delta e menor delta, pega o maior e o menor delta de cada ciclo, no caso a cada 5 linhas. A coluna média faz a correção do algoritmo FTM, isto é faz o cálculo da função de convergência que é dado pela equação (3.6). A coluna média, da Tabela C.4, faz a correção do Algoritmo FTA, isto é faz o cálculo da função de convergência que é dado pela equação (3.7). Após cada ciclo completo a correção é computada, sendo compensada pela constante que indica o atraso de rede. Após o ciclo completo, é adicionado o de correção valor ao relógio,

atualizando assim as colunas T1, T2, T3, T4 e T5 com seus respectivos valores. A coluna referência é o relógio de referência da simulação.

A Figura P.5 mostra o graficamente o resultado da simulação com aplicação do algoritmo FTM. A Figura P.6 mostra o resultado da simulação aplicando o algoritmo FTA.

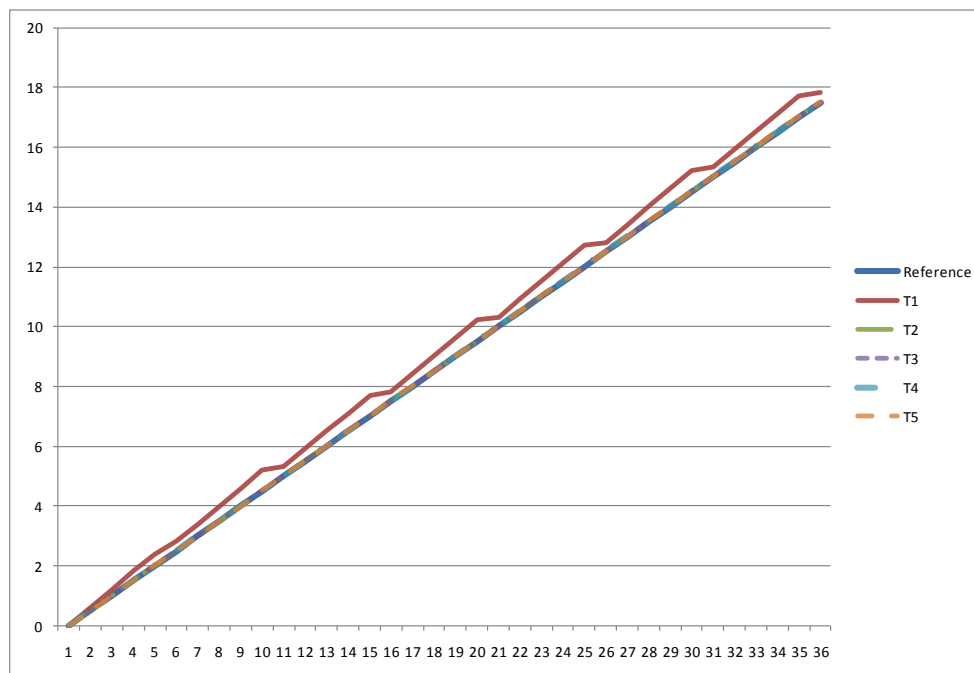


Figura P.5 – FTM/FTA – 5º Caso Com Sincronismo - FTM.

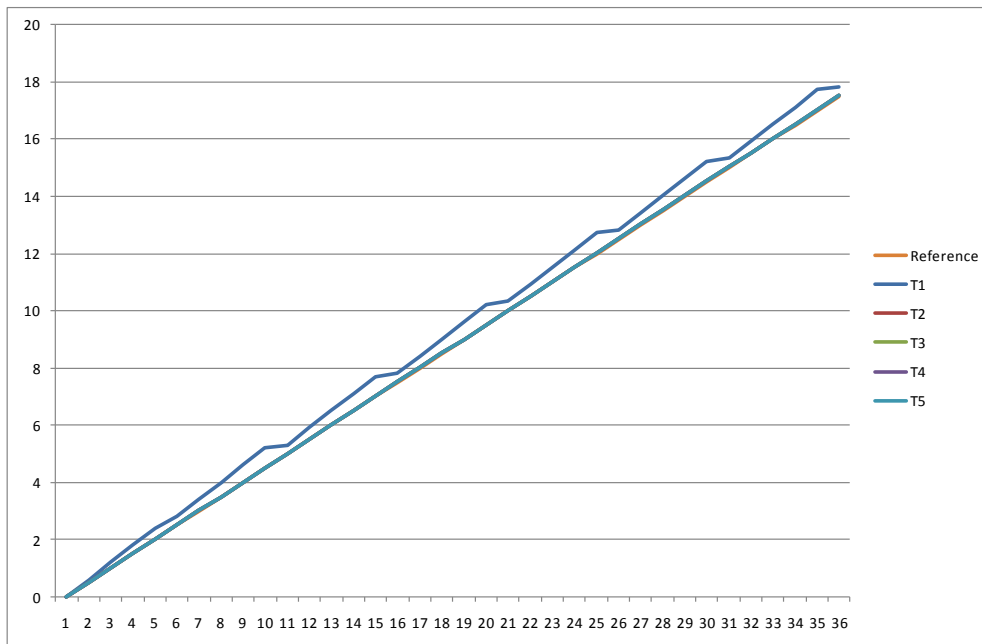


Figura P.6 – FTM/ FTA – 5º Caso Com Sincronismo - FTA.

Nas Figuras P.5 e P.6 é possível ver uma forma de onda dente de serra se formando. Isto ocorre, pois o algoritmo de sincronização é aplicado periodicamente, aplicando a correção dos tempos. Com isto os relógios T1, T2, T3, T4 e T5 ficam dentro de uma determinada precisão. Outro ponto é que os resultados das Figuras P.5 e P.6 são iguais, apesar de serem aplicados algoritmos de correção diferentes. Isto ocorre, pois apesar dos algoritmos serem diferentes, as equações 3.6 e 3.7 possuem o mesmo resultado.

• **Exemplo:**

Pegando o ultimo ciclo da Tabela C.4, vamos fazer o cálculo das médias. Será feito somente os cálculos com os vetores delta1 e delta2, já que os vetores delta3, delta4 e delta5 são idênticos ao delta2.

$$\Delta 1 = [0; 0.4; 0.5; 0.6; 0.7]$$

$$\Delta 2 = [-0.3; 0; 0; 0; 0]$$

**Caso com FTM:** Cálculo pela equação 3.6, eliminado o maior e menor delta, e pegando o maior e menor delta restante. Tem-se:

$$m\u00e9diaDelta1 = \frac{0.4 + 0.6}{2} = 0.5$$

$$m\u00e9diaDelta2 = \frac{0 + 0}{2} = 0$$

**Caso com FTA:** Cálculo pela equação 3.7, eliminado o maior e menor delta, e fazendo a média aritm\u00e9tica dos elementos restantes. Tem-se:

$$m\u00e9diaDelta1 = \frac{0.4 + 0.5 + 0.6}{3} = 0.5$$

$$m\u00e9diaDelta1 = \frac{0 + 0 + 0}{3} = 0$$

Conforme \u00e9 poss\u00edvel ver no exemplo acima, apesar de os algoritmos FTM e FTA serem diferentes, neste caso quando variamos somente a deriva do rel\u00f3gio T1, os algoritmos alcan\u00e7am o mesmo resultado. Conseq\u00fcentemente, as Figura P.7 e P.8 que mostram o resultado da fun\u00e7\u00e3o de corre\u00e7\u00e3o dos algoritmos FTM e FTA, respectivamente, s\u00e3o iguais.

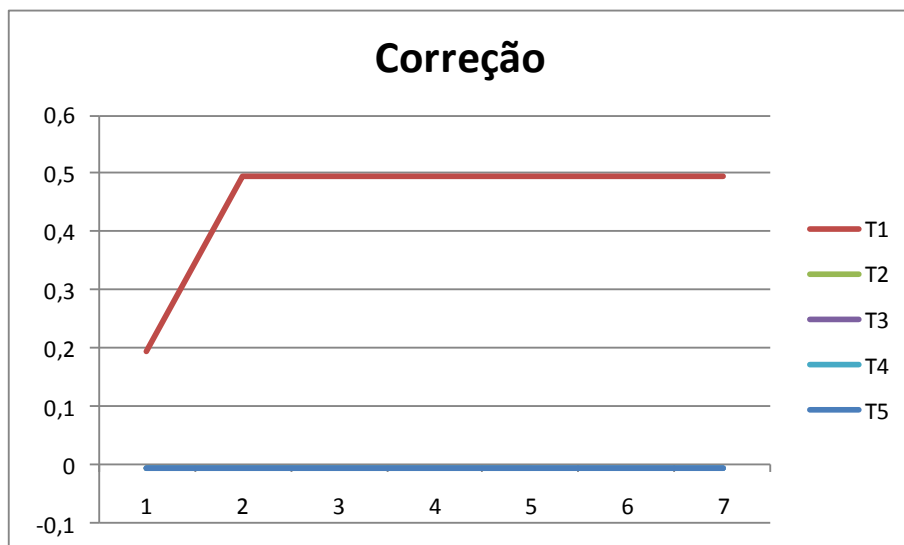


Figura P.7 – FTM/FTA – Correção do 5º Caso pelo Algoritmo FTM.

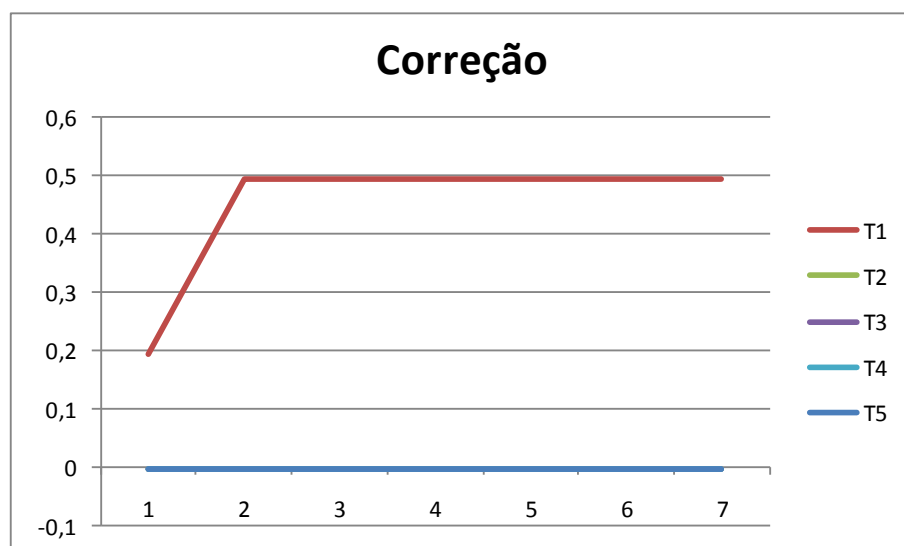


Figura P.8 – FTM/FTA – Correção do 5º Caso pelo Algoritmo FTA.

Nestes gráficos das figuras P.7 e P.8 é possível ver que os algoritmos FTA e FTM convergiram a um valor. Este valor de correção indica que a cada ciclo, o relógio T1 precisa sofrer uma correção em torno de 0.5, e os relógios T2, T3, T4 e T5, em torno de -0,005.

Análise dos Dados:

- **Algoritmo FTM:**

- Máximo viés instantâneo, de acordo com a eq. (2.33):  $\Gamma = 1.0$ ;
- Máximo viés instantâneo na simulação:  $\Gamma = 0.7$ ;
- Precisão, de acordo com a equação (2.45), multiplicando pelo termo de do erro bizantino ( $\mu$ ) da equação 2.49:  $\Pi = 0.75$ ;
- Precisão alcançada pela simulação:  $\Pi = 0.7$ ;

- **Algoritmo FTA:**

- Máximo viés instantâneo, de acordo com a eq. (2.33):  $\Gamma = 1.0$ ;
- Máximo viés instantâneo na simulação:  $\Gamma = 0.7$ ;
- Precisão, de acordo com a equação 2.50:  $\Pi = 0.5$ ;
- Precisão alcançada pela simulação:  $\Pi = 0.7$ ;

Neste caso podemos concluir que ambos os algoritmos chegam a um mesmo resultado, tanto em máximo viés instantâneo quanto na precisão final alcançada. Isto ocorre, pois o número de relógios do algoritmo FTA não é suficientemente maior para que as médias dos vetores delta se afastem da mediana. Isto é, a mediana é igual à média. Este é um caso particular e uma coincidência, para outros valores de deriva, ou outros relógios com deriva isto não é necessariamente uma verdade.

No algoritmo FTM seu resultado é satisfatório já que a precisão ficou dentro da pré-estabelecida pelas equações. No algoritmo FTA, a precisão ficou acima da pré-estabelecida pelas equações. Isto pode ter ocorrido por vários fatores, entre eles o intervalo de re-sincronização é muito grande, o número de dados (número de relógios) não é satisfatório, devendo assim aumentar o número de dados, ou também a deriva aplicada é muito grande, prejudicando assim a precisão final alcançada.

## Caso 6 – 6 Relógios, um com erro de deriva

Neste caso também é possível verificar através das equações (3.6) e (3.7) que os algoritmos FTM e FTA são diferentes, pois suas funções de convergência são diferentes. O resultado esperado é que os algoritmos apresentem resultado final um pouco diferente.

Dados do Experimento:

- Número de Relógios:  $n = 6$ ;
- Número de erros bizantinos:  $f = 1$ ;
- Taxa de deriva:  $\rho = 0.2$ ;
- Atraso de Rede:  $\delta = 0.005$  (s);

Neste caso, foi aplicado ao relógio T1, um erro de taxa de deriva de 0.2 em relação aos outros relógios. O relógio T1 é o relógio com falha de deriva. Os relógios T2, T3, T4, T5 e T6 não estão com falha. A Tabela C.5 mostra o modelo do experimento utilizando o algoritmo FTM e FTA, realizado com o auxílio do Microsoft Excel. As colunas T1, T2, T3, T4, T5 e T6 são as colunas de propagação dos relógios. As colunas Delta1, Delta2, Delta3, Delta4, Delta5 e Delta6 são as colunas aonde a diferença dos relógios é realizada, seguindo a equação 3.2. O ciclo é identificado pela coluna ciclo. A cada instante um relógio envia seus dados, como em uma comunicação TDMA. A coluna maior delta e menor delta, pega o maior e o menor delta de cada ciclo, no caso a cada 6 linhas. A coluna média faz a correção do algoritmo FTM, isto é faz o cálculo da função de convergência que é dado neste caso é dado pela equação 3.6. A coluna média, da Tabela C.5 faz a correção do Algoritmo FTA, isto é faz o cálculo da função de convergência que é dado neste caso pela equação 3.7. Após cada ciclo completo a correção é computada, sendo compensada pela constante que indica o atraso de rede. Após o ciclo completo, é adicionado o



de correção valor ao relógio, atualizando assim as colunas T1, T2, T3, T4, T5 e T6 com seus respectivos valores. A coluna referência é o relógio de referência da simulação.

A Figura P.9 mostra o graficamente o resultado da simulação com aplicação do algoritmo FTM. A Figura P.10 mostra o resultado da simulação aplicando o algoritmo FTA.

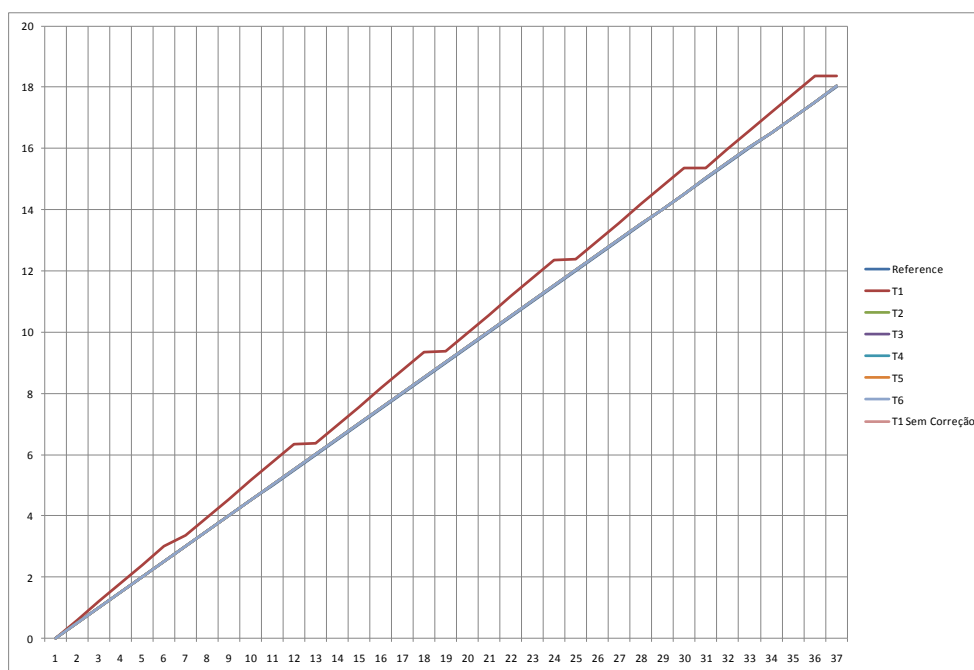


Figura P.9 – FTM/FTA – 6º Caso Com Sincronismo pelo Algoritmo FTM.

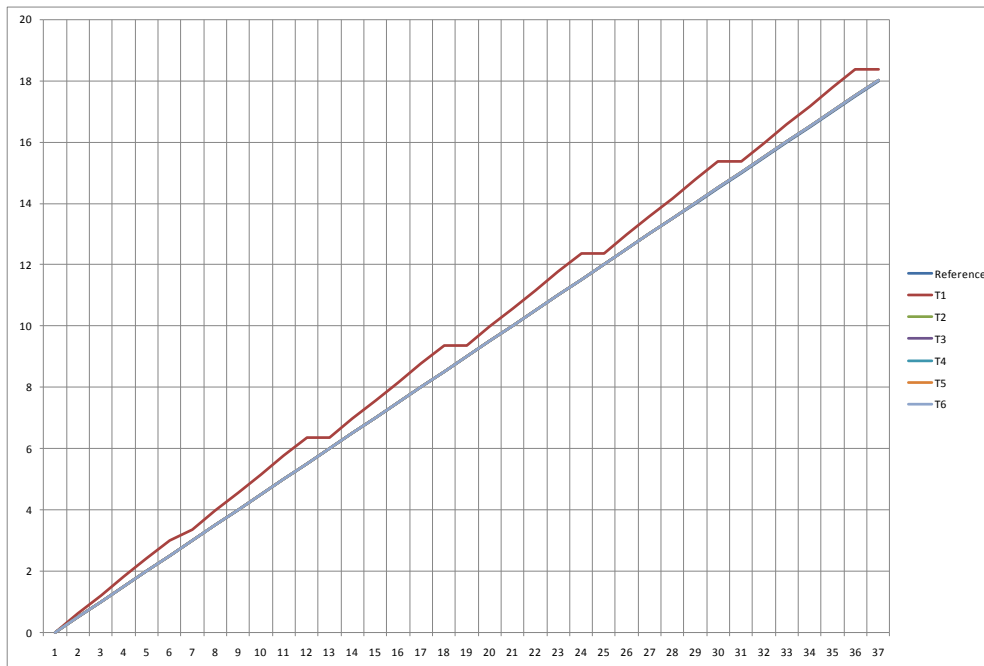


Figura P.10 – FTM/FTA – 6º Caso Com Sincronismo pelo Algoritmo FTA.

Nas Figuras P.9 e P.10 é possível ver uma forma de onda dente de serra se formando. Isto ocorre, pois o algoritmo de sincronização é aplicado periodicamente, aplicando a correção dos tempos. Com isto os relógios T1, T2, T3, T4, T5 e T6 ficam dentro de uma determinada precisão.

Outro ponto é que os resultados das Figuras P.9 e P.10 são iguais, apesar de serem aplicados algoritmos de correção diferentes. Isto ocorre, pois apesar dos algoritmos serem diferentes as equações 3.6 e 3.7 possuem o mesmo resultado.

• **Exemplo:**

Pegando o ultimo ciclo da Tabela C.5 vamos fazer o calculo das médias. Será feito somente os cálculos com os vetores delta1 e delta2, já que os vetores delta3, delta4, delta5 e delta6 são idênticos ao delta2.

$$\Delta 1 = [0; 0.45; 0.55; 0.65; 0.75; 0,85]$$

$$\Delta_2 = [-0.35; 0; 0; 0; 0; 0]$$

**Caso com FTM:** Cálculo pela equação 3.6, eliminado o maior e menor delta, e pegando o maior e menor delta restante. Tem-se:

$$m\u00e9dia\Delta_1 = \frac{0.4 + 0.6}{2} = 0.6$$

$$m\u00e9dia\Delta_2 = \frac{0 + 0}{2} = 0$$

**Caso com FTA:** Cálculo pela equação 3.7, eliminado o maior e menor delta, e fazendo a média aritmética dos elementos restantes. Tem-se:

$$m\u00e9dia\Delta_1 = \frac{0.45 + 0.55 + 0.65 + 0.75}{4} = 0.6$$

$$m\u00e9dia\Delta_2 = \frac{0 + 0 + 0 + 0}{4} = 0$$

Conforme é possível ver no exemplo acima, apesar de os algoritmos FTM e FTA serem diferentes, neste caso quando variamos somente a deriva do relógio T1, os algoritmos alcançam o mesmo resultado. Conseqüentemente, as Figura P.11 e P.12 que mostram o resultado da função de correção dos algoritmos FTM e FTA, respectivamente, são iguais.

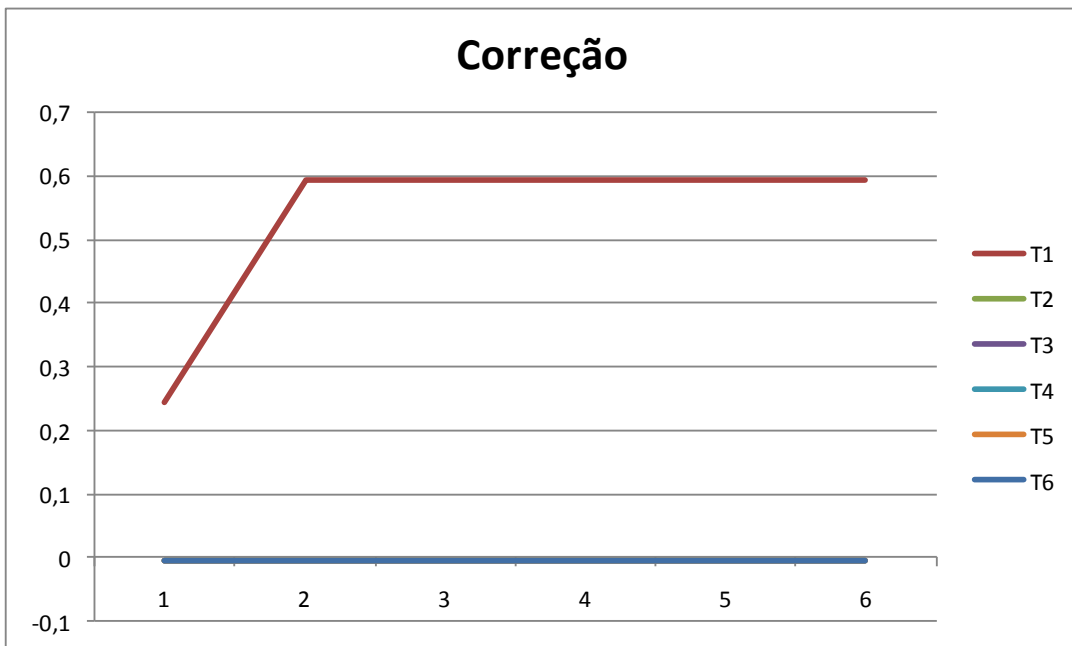


Figura P.11 – FTM/FTA – Correção do 6º Caso pelo Algoritmo FTM.

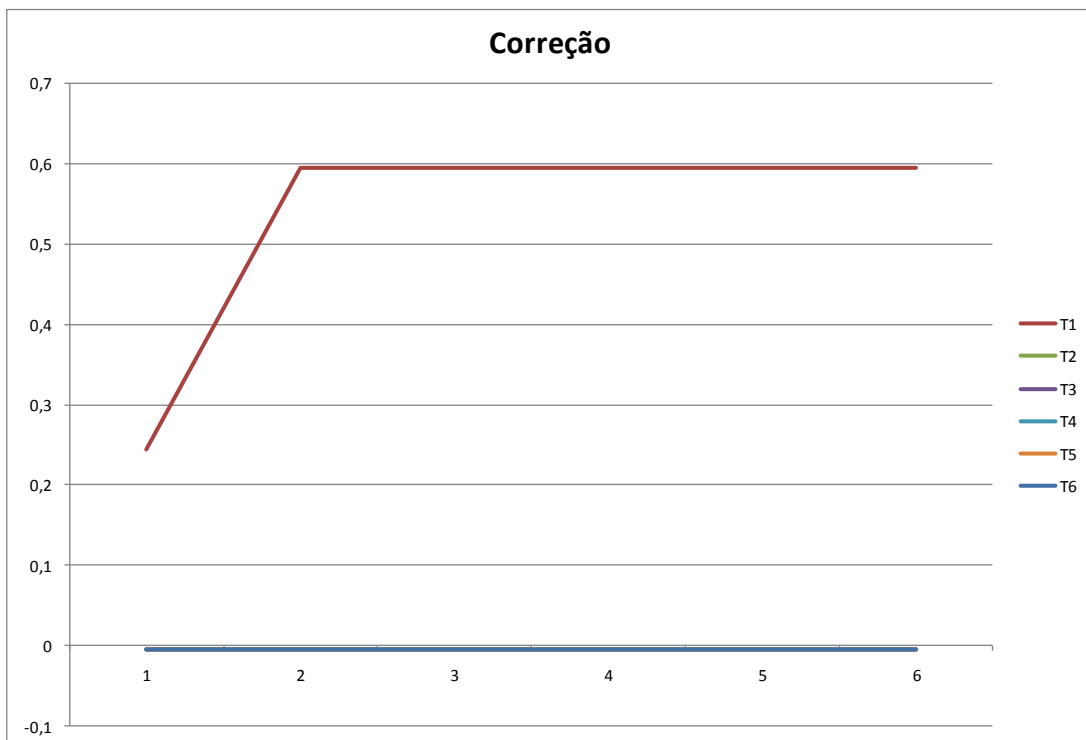


Figura P.12 – FTM/FTA – Correção do 6º Caso pelo Algoritmo FTA.

Nestes gráficos das figuras P.11 e P.12 é possível ver que os algoritmos FTA e FTM convergiram a um valor. Este valor de correção indica que a cada ciclo, o relógio T1 precisa sofrer uma correção em torno de 0.6, e os relógios T2, T3, T4, T5 e T6, em torno de -0,005.

Análise dos Dados:

- **Algoritmo FTM:**

- Máximo viés instantâneo, de acordo com a eq. (2.33):  $\Gamma = 1.2$ ;
- Máximo viés instantâneo na simulação:  $\Gamma = 0.85$ ;
- Precisão, de acordo com a equação (2.45), multiplicando pelo termo de do erro bizantino ( $\mu$ ) da equação 2.49:  $\Pi = 0.798$ ;
- Precisão alcançada pela simulação:  $\Pi = 0.85$ ;

- **Algoritmo FTA:**

- Máximo viés instantâneo, de acordo com a eq. (2.33):  $\Gamma = 1.2$ ;
- Máximo viés instantâneo na simulação:  $\Gamma = 0.85$ ;
- Precisão, de acordo com a equação 2.50:  $\Pi = 0.5$ ;
- Precisão alcançada pela simulação:  $\Pi = 0.399$ ;

Neste caso podemos concluir que ambos os algoritmos chegam a um mesmo resultado, tanto em máximo viés instantâneo quanto na precisão final alcançada. Isto ocorre, pois o número de relógios do algoritmo FTA não é suficientemente maior para que as médias dos vetores delta se afastem da mediana dos vetores. Isto é, a mediana é igual à média. Este é um caso particular e uma coincidência, para outros valores de deriva, ou outros relógios com deriva isto não é necessariamente uma verdade.

No algoritmo FTM seu resultado não é satisfatório já que a precisão ficou acima da pré-estabelecida pelas equações. No algoritmo FTA, a precisão também ficou acima da pré-estabelecida pelas equações. Isto ocorre por vários fatores, entre eles o intervalo de re-sincronização é muito grande; o número de dados

(número de relógios) não é satisfatório no caso FTA, devendo assim aumentar o número de dados; a deriva aplicada é muito grande; tudo isso em conjunto prejudica a precisão final alcançada.

### **Caso 7– 7 Relógios, dois com erro de estado**

Neste caso é possível verificar através das equações (3.6) e (3.7) que os algoritmos FTM e FTA são diferentes, pois suas funções de convergência são diferentes. O resultado esperado é que os algoritmos apresentem resultado final diferente.

Dados do Experimento:

- Número de Relógios:  $n = 7$ ;
- Número de erros bizantinos:  $f = 2$ ;
- Erro de estado;
- Atraso de Rede:  $\delta = 0.005$  (s);

Neste caso, foi aplicado ao relógio T1 e ao relógio T2, um erro de estado, isto é, os relógios T1 e T2 estão parados. Os relógios T1 e T2 são os relógios com falha de estado. Os relógios T3, T4, T5, T6 e T7 não estão com falha. A Tabela C.8 mostra o modelo do experimento com o algoritmo FTM e a Tabela C.9 mostra o modelo do experimento com o algoritmo FTA, ambas simulações realizadas com o auxílio do Microsoft Excel. As colunas T1, T2, T3, T4, T5, T6 e T7 são as colunas de propagação dos relógios. As colunas Delta1, Delta2, Delta3, Delta4, Delta5, Delta6 e Delta7 são as colunas aonde a diferença dos relógios é realizada, de acordo com a equação (3.2). O ciclo é identificado pela coluna ciclo. A cada instante um relógio envia seus dados, como em uma comunicação TDMA. A coluna maior delta e menor delta, pega o maior e o menor delta de cada ciclo, no caso a cada 7 linhas. A coluna média, da Tabela

C.8, faz a correção do algoritmo FTM, isto é faz o cálculo da função de convergência que é dado pela equação 3.6. A coluna média, da Tabela C.9, faz a correção do Algoritmo FTA, isto é faz o cálculo da função de convergência que é dado neste caso pela equação 3.7. Após cada ciclo completo a correção é computada, sendo compensada pela constante que indica o atraso de rede. Após o ciclo completo, é adicionado o de correção valor ao relógio, atualizando assim as colunas T1, T2, T3, T4, T5, T6 e T7 com seus respectivos valores. A coluna referência é o relógio de referência da simulação.

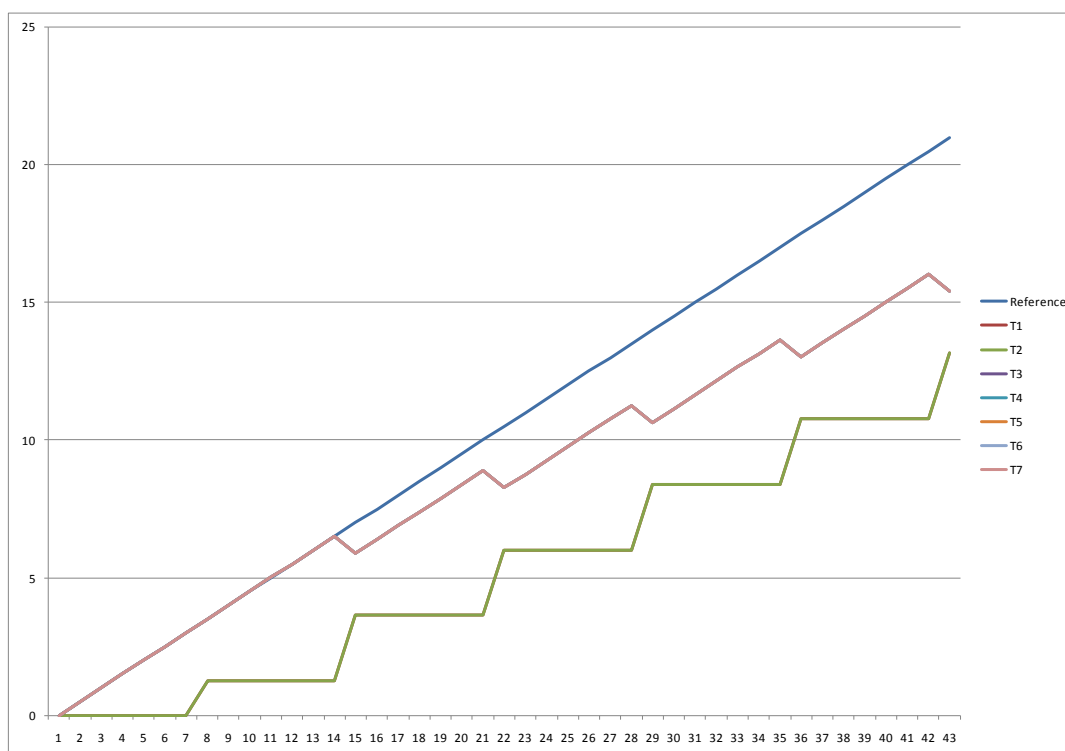


Figura P.13 – FTM/FTA – 7º Caso Com Sincronismo pelo Algoritmo FTM.

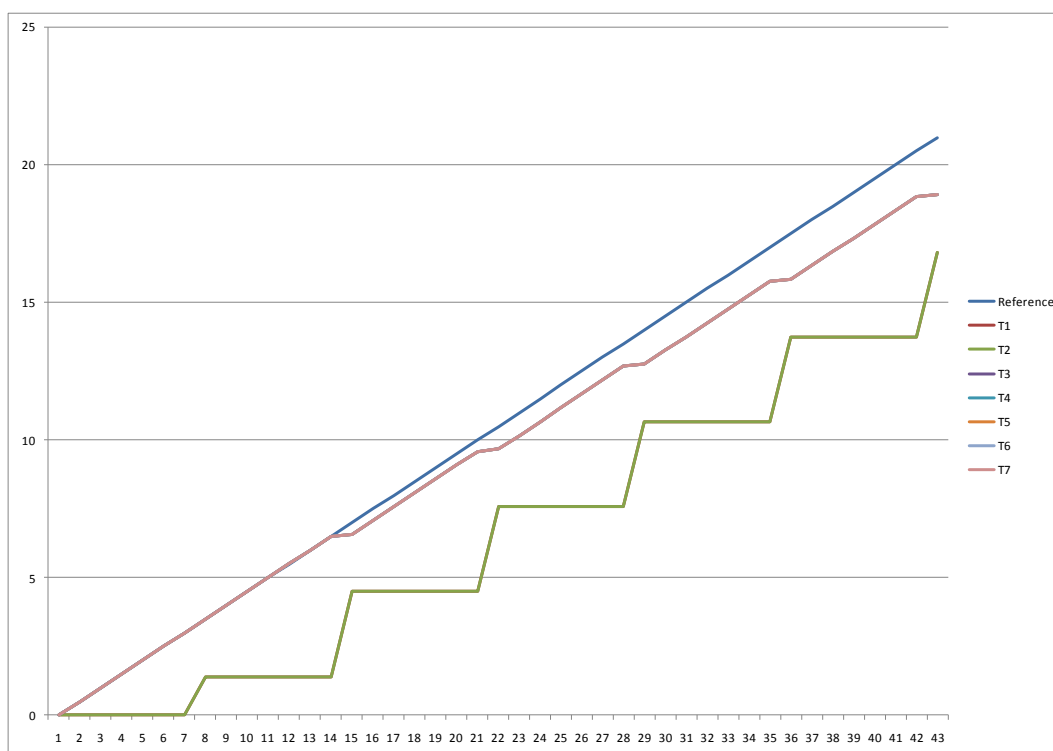


Figura P.14 – FTM/FTA – 7º Caso Com Sincronismo pelo Algoritmo FTA.

A Figura P.13 mostra o graficamente o resultado da simulação com aplicação do algoritmo FTM. A Figura P.14 mostra o resultado da simulação aplicando o algoritmo FTA. Nas Figuras P.13 e P.14 é possível ver uma forma de onda de serra se formando. Isto ocorre, pois o algoritmo de sincronização é aplicado periodicamente, aplicando a correção dos tempos. Neste caso, o relógio T1 e T2 estão parados, mas mesmo assim foi aplicado a ele a correção dos relógios. Este caso é interessante, pois é possível ver que o algoritmo FTM e o FTA foram robustos o bastante para isolarem o problema causado pelo relógio T1 e T2 do resto dos relógios do conjunto (T3, T4, T5, T6 e T7) que continuam sincronizados entre si, pois sua convergência não se modificou. Somente o T1 e T2 foram prejudicados com o erro de estado. É interessante também o fato de que o conjunto dos relógios (T3, T4, T5, T6, T7) apesar de sincronizados entre si, o conjunto se afasta do valor do relógio de referência, neste caso o conjunto tanto do algoritmo FTM quanto do algoritmo FTA está se atrasando em relação ao tempo de referência.



**Exemplo:**

Pegando o ultimo ciclo da Tabela C.8 e Tabela C.9, vamos fazer o cálculo das médias. Será feito somente os cálculos com os vetores delta1 e delta3, já que os vetores delta4, delta5, delta6 e delta7 são idênticos ao delta3 e o vetor delta2 idêntico ao vetor delta1.

**Tabela C.8:**

$$\text{Delta1} = [-5.25; -4.75; -4.25; -3.75; -3.25; 0; 0]$$

$$\text{Delta3} = [0; 0; 0; 0; 0; 2.25; 2.75]$$

**Caso com FTM:** Cálculo pela equação 3.6, eliminado o maior e menor delta, e pegando o maior e menor delta restante. Tem-se:

$$\text{médiaDelta1} = \frac{-4.75 + 0}{2} = -2.375$$

$$\text{médiaDelta3} = \frac{0 + 2.25}{2} = 1.125$$

**Tabela C.9:**

$$\text{Delta1} = [-5.1; -4.6; -4.1; -3.6; -3.1; 0; 0]$$

$$\text{Delta3} = [0; 0; 0; 0; 0; 2.1; 2.6]$$

**Caso com FTA:** Cálculo pela equação 3.7, eliminado o maior e menor delta, e fazendo a média aritmética dos elementos restantes. Tem-se:

$$\text{médiaDelta1} = \frac{-4.6 - 4.1 - 3.6 - 3.1 + 0}{5} = -3.08$$

$$\text{médiaDelta3} = \frac{0 + 0 + 0 + 0 + 2.1}{5} = 0.42$$

No exemplo acima é possível ver que ambos os algoritmos computam valores de correção diferentes. Nas Figuras P.15 e 4.16 é possível ver os gráficos de convergência do algoritmo.

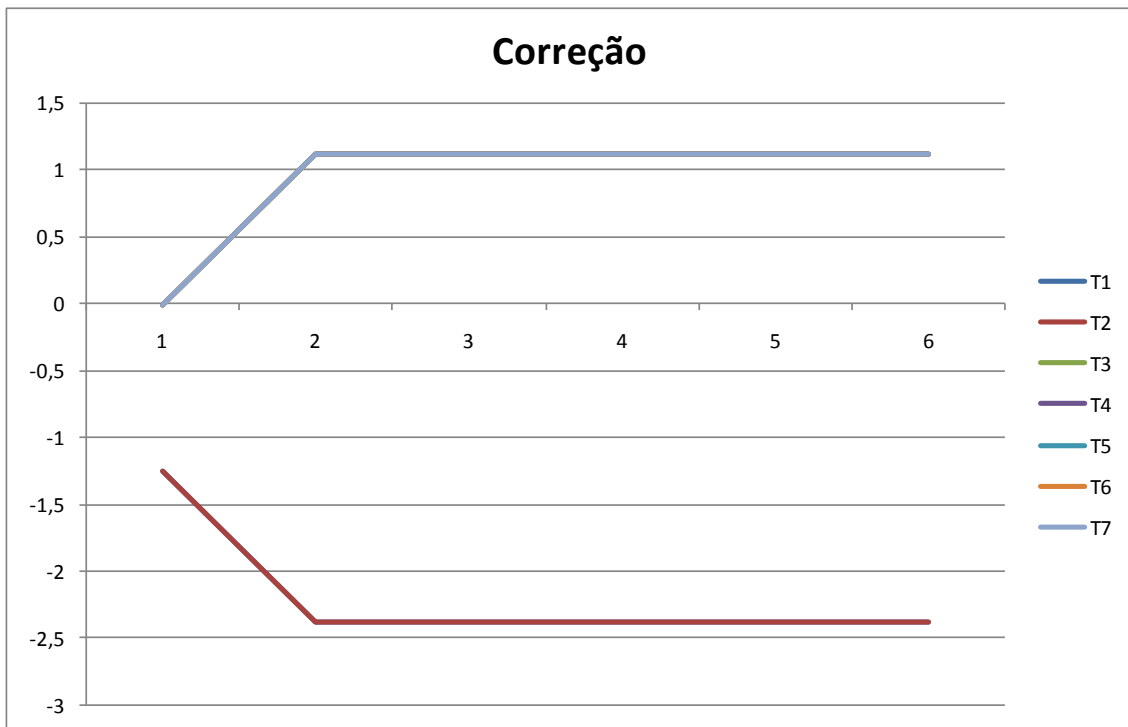


Figura P.15 – FTM/FTA – Correção do 7º Caso pelo Algoritmo FTM.

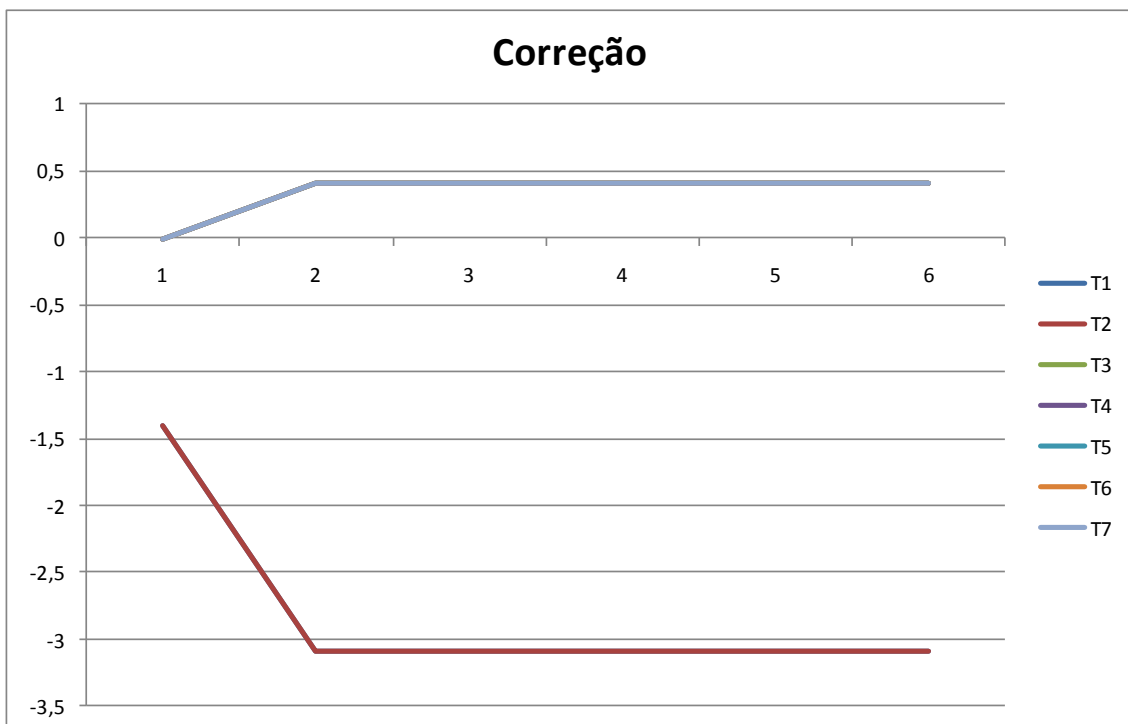


Figura P.16 – FTM/FTA – Correção do 7º Caso pelo Algoritmo FTA.

Nestes gráficos das figuras P.15 e P.16 é possível ver que os algoritmos FTA e FTM convergiram a um valor e são diferentes. O Algoritmo FTA computa uma correção maior para os relógios T1 e T2 e uma correção menor para os relógios T3, T4, T5, T6 e T7. Isso indica que o valor de correção do FTM, a cada ciclo, do relógio T1 e T2 precisa sofrer uma correção em torno de -2.375, e os relógios T3, T4, T5, T6 e T7, em torno de 1.125. O valor de correção do FTA, a cada ciclo, do relógio T1 e T2 precisa sofrer uma correção em torno de -3.08, e os relógios T3, T4, T5, T6 e T7, em torno de 0.42.

Análise dos Dados:

- **Algoritmo FTM:**

- Máximo viés instantâneo na simulação:  $\Gamma = -5.25$ ;
- Precisão alcançada pela simulação:  $\Pi = 5.25$ ;

- **Algoritmo FTA:**

- Máximo viés instantâneo na simulação:  $\Gamma = 5.1$ ;
- Precisão alcançada pela simulação:  $\Pi = 5.1$ ;

No algoritmo FTM e FTA as precisões foram prejudicadas por causa da presença dos relógios T1 e T2. Mas os outros relógios T3, T4, T5, T6 e T7 ficaram isolados do problema e não prejudicaram a sua precisão, mantendo-se sincronizados entre si. O algoritmo FTA teve uma precisão ligeiramente melhor que o algoritmo FTM, o que explica o seu menor afastamento nas Figura P.14.



## APÊNDICE Q – MODELO DO FILTRO DE KALMAN

Existem muitos modelos para se representar um relógio. O modelo mais simples de um relógio, baseado em Varnum (1983), é definido por duas variáveis de estado, o viés instantâneo e a deriva. Este Apêndice desenvolve as equações formais do Filtro de Kalman(FK) utilizadas neste trabalho.

O filtro de Kalman é dividido em duas etapas: 1) Propagação ou predição; 2) Atualização ou correção. A fase de propagação propaga o estado e a covariância do instante. A fase de atualização corrige o estado e a covariância para o instante, devido à medida.

O **modelo de dois estados** para representar o relógio, é dado por:

$$x_k = \begin{bmatrix} T_k \\ D_k \end{bmatrix} \quad (\text{Q.1})$$

$$\begin{bmatrix} \dot{T} \\ \dot{D} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} T \\ D \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$$
$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \quad (\text{Q. 2})$$
$$G = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = N(0, Q_k)$$

Onde  $x_k$  é o vetor de 2 estados. )  $A$  é a matriz 2x2 que relaciona o estado e sua derivada linearmente,  $G$  é uma matriz 2x2 de adição do ruído dinâmico, e  $w$  é o ruído dinâmico modelado por um processo branco,  $Q_k$  é matriz 2x2 de covariância do ruído branco.

A **Fase de Propagação** é dada por:

$$\bar{x}_k = \Phi_{k,k-1} \hat{x}_{k-1} \quad (\text{Q.3})$$

$$\bar{P}_k = \Phi_{k,k-1} \hat{P}_{k-1} \Phi_{k,k-1}^t + \Gamma_k Q_k \Gamma_k^t \quad (\text{Q.4})$$

Onde  $\bar{x}_k$  e  $\bar{P}_k$  representam o estado e a covariância propagada no instante k.

Para solucionar este modelo, vamos dividir em duas partes. A parte determinística e a parte não determinística.

### Solução da parte determinística:

**Matriz de Transição:** A solução da matriz de transição, necessária para propagar os estados e as covariâncias é dada por:

$$\Phi_{k+1,k} = e^{A\Delta t} \cong I + A\Delta t + \frac{1}{2!} A^2 \Delta t^2 + \dots + \frac{1}{n!} A^n \Delta t^n \quad (\text{Q.5})$$

$$\Phi_{k+1,k} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & t_k - t_{k-1} \\ 0 & 1 \end{bmatrix} \quad (\text{Q.6})$$

Onde a matriz  $A$  é dada pela equação Q.2 e  $\Delta t$  é o tempo decorrido (*elapsed time*). Portanto a solução de Q.2 é dada por:

$$\begin{bmatrix} T_k \\ D_k \end{bmatrix} = \Phi_{k+1,k} \begin{bmatrix} T_{k-1} \\ D_{k-1} \end{bmatrix} \quad (\text{Q.7})$$

As condições iniciais do nosso Filtro de Kalman são dadas por:

$$\begin{aligned} \bar{x}_k &= \hat{x}_k \\ \bar{P}_k &= \hat{P}_k \end{aligned} \quad (\text{Q.8})$$

Com isto, agora é necessário apresentar a solução da parte não-determinística do modelo. A solução é dada por uma equação matricial de Riccati em Q.4. Para este trabalho utilizamos a seguinte equação:

$$\bar{P}_k = \Phi_{k+1,k} \hat{P}_{k-1} \Phi_{k+1,k}^t + \int_k^{k+1} \Phi_{\tau,k} G(\tau) Q(\tau) G^t(\tau) \Phi_{\tau,k} d\tau \quad (\text{Q.9})$$

Neste trabalho iremos utilizar as matrizes G e Q constantes no tempo. Portanto a solução da equação Q.9 é dada por:

$$\bar{P}_k = \Phi_{k+1,k} \hat{P}_{k-1} \Phi_{k+1,k}^t + \begin{bmatrix} 1 & \frac{\Delta t}{2} \\ 0 & 1 \end{bmatrix} Q \begin{bmatrix} 1 & \frac{\Delta t}{2} \\ 0 & 1 \end{bmatrix} \Delta t \quad (\text{Q.10})$$

Onde a matriz Q é dada por:

$$Q = \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix} \quad (\text{Q.11})$$

Neste trabalho foi usado a seguinte variância:

$$\sigma^2 = 0.0002 \quad (\text{Q.12})$$

Portanto a equação Q.10 é dada por:

$$\bar{P}_k = \Phi_{k+1,k} \hat{P}_{k-1} \Phi_{k+1,k}^t + \begin{bmatrix} 1 & \frac{\Delta t}{2} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0.0002 & 0 \\ 0 & 0.0002 \end{bmatrix} \begin{bmatrix} 1 & \frac{\Delta t}{2} \\ 0 & 1 \end{bmatrix} \Delta t \quad (\text{Q.13})$$

$$\bar{P}_k = \Phi_{k+1,k} \hat{P}_{k-1} \Phi_{k+1,k}^t + \begin{bmatrix} 0.0002 & 0.0002\Delta t \\ 0 & 0.0002 \end{bmatrix} \Delta t \quad (\text{Q.14})$$

A matriz de covariância P, da solução não determinística do modelo do relógio, especifica a incerteza na indica na medida do relógio. A matriz Q incorpora o

ruído dinâmico para cada estado do filtro. Antes de realizar a fase de atualização é necessário definir o modelo de medidas. Que é dado por:

$$T_k = t_k^2 - t_k^1 \quad (\text{Q.15})$$

$$D_k = \frac{t_k^2 - t_{k-1}^2}{t_k^1 - t_{k-1}^1} - 1 \quad (\text{Q.16})$$

Onde,  $t_k^2$  é o valor do tempo no relógio local, definido como 2, no instante k e o  $t_k^1$  é o valor do tempo no relógio de referência, definido como 1, no instante k. O modelo de medidas do filtro de Kalman, neste trabalho é um modelo discreto. O modelo de medidas proposto, também baseado em Varnum (1983), é dado por:

$$y_k = H_k x_k + v_k \quad (\text{Q.17})$$

Com:

$$v = N(0, R_k(t)) \quad (\text{Q.18})$$

Onde  $y_k$  é o vetor de  $m$ -medidas,  $H_k$  é a matriz de observação que relaciona as medidas com os estados e  $x_k$  é o vetor de  $n$ -estados do sistema,  $v_k$  é a matriz de ruídos gaussianos  $m \times m$  onde utilizaremos a mesma variância de Q.12, todos os termos estão no instante  $t_k$ .

Com isso o modelo proposto é dado por:

$$x_k = \begin{bmatrix} T_k \\ D_k \end{bmatrix} \quad (\text{Q.19})$$

$$H_k = [1 \quad \Delta t] \quad (\text{Q.20})$$

$$y_k = [1 \quad \Delta t] \begin{bmatrix} T_k \\ D_k \end{bmatrix} + v_k \quad (\text{Q.21})$$

Onde  $\Delta t$  é o tempo decorrido.



Fazendo as equações (Q.15) e (Q.16) em (Q.21), tem-se o modelo de medidas:

$$y_k = [1 \quad \Delta t] \begin{bmatrix} t_k^2 - t_k^1 \\ \frac{t_k^2 - t_{k-1}^2}{t_k^1 - t_{k-1}^1} - 1 \end{bmatrix} + v_k \quad (\text{Q.22})$$

Com isso é a **Fase de Atualização** do Filtro de Kalman é dada por:

$$K_k = \bar{P}_k H_k^t (H_k \bar{P}_k H_k^t + R_k)^{-1} \quad (2.53)$$

$$\hat{P}_k = (I - K_k H_k) \bar{P}_k \quad (2.54)$$

$$\hat{x}_k = \bar{x}_k + K_k (y_k - H_k \bar{x}_k) \quad (2.55)$$

Onde  $K_k$  é o ganho de Kalman, e  $\hat{x}_k$  e  $\hat{P}_k$  são o estado e a covariância atualizada no instante  $k$ .