

KAIQUE RHUAN DE AZEVEDO ALBUQUERQUE

**DESENVOLVIMENTO DE UM SISTEMA DE  
CONTROLE E ORDENAÇÃO DE  
CONVERSORES PONTE H EM CASCATA  
UTILIZANDO REDE CAN**

João Pessoa

2017



KAIQUE RHUAN DE AZEVEDO ALBUQUERQUE

**DESENVOLVIMENTO DE UM SISTEMA DE  
CONTROLE E ORDENAÇÃO DE CONVERSORES  
PONTE H EM CASCATA UTILIZANDO REDE CAN**

Trabalho de Conclusão de Curso apresentado  
ao Curso de Engenharia Elétrica da UFPB  
como requisito parcial para obtenção do título  
de Engenheiro Eletricista

UNIVERSIDADE FEDERAL DA PARAÍBA - UFPB

Orientador: Carlos Alberto de Sousa Filho

Coorientador: Lucas Vinícius Hartmann

João Pessoa

2017

KAIQUE RHUAN DE AZEVEDO ALBUQUERQUE

**DESENVOLVIMENTO DE UM SISTEMA DE  
CONTROLE E ORDENAÇÃO DE CONVERSORES  
PONTE H EM CASCATA UTILIZANDO REDE CAN**

Trabalho de Conclusão de Curso apresentado  
ao Curso de Engenharia Elétrica da UFPB  
como requisito parcial para obtenção do  
título de Engenheiro Eletricista

Orientador: Prof. Carlos Alberto de  
Sousa Filho

Coorientador: Prof. Lucas Vinícius Hartmann

Banca Examinadora

---

**Prof. Dr. Carlos Alberto de Sousa  
Filho**  
Orientador

---

**Prof. Dr. Lucas Vinícius Hartmann**  
Coorientador

---

**Prof. Dr. Ademar Virgolino da Silva  
Netto**  
Prof. Convidado 1

---

**Prof. Dr. Edgard Luiz Lopes Fabrício**  
Prof. Convidado 2

João Pessoa  
2017

# RESUMO

Neste Trabalho de Conclusão de Curso foi realizada a implementação de um algoritmo de ordenação para atuar na estratégia de controle de um grupo de inversores multinível ponte H em cascata. O algoritmo de ordenação se baseia na propriedade do protocolo CAN que trata da arbitração de mensagens, sendo esta utilizada para definir prioridades entre os nós da rede, onde cada nó representa um conversor ponte H. Assim como a implementação do modelo de um painel fotovoltaico e de um conversor ponte H no nó conversor, e simulação de um controle de corrente no mestre. O algoritmo de ordenação foi validado através do monitoramento das mensagens em uma rede CAN com um nó mestre e 3 nós escravos simulando conversores acoplados a painéis fotovoltaicos. Foi verificado que os três nós conversores fornecem a tensão requisitada pelo nó mestre, alternando seu estado de operação para se manter dentro do ponto de operação ótimo. A cada instante de requisição da tensão e corrente pelo nó mestre, foi constatado que apenas os nós conversores com maior necessidade de fornecer ou receber energia da rede entraram em condução. Ao fim da ordenação os nós conversores alternam os estados de suas chaves adequadamente para se manter em torno do ponto de operação ótimo. O controle de corrente da rede foi executado pelo nó mestre, sendo validado através da mudança do sinal de referência e monitoramento do comportamento da corrente até atingir o regime permanente, onde é mostrado curvas que representam o comportamento desta corrente com relação à um valor de referência. Como conclusão há a constatação de que é possível utilizar a rede CAN para realizar o algoritmo de ordenação de conversores ponte H.

**Palavras-chaves:** *Controller Area Network* (CAN). Algoritmo de Ordenação. Conversor Ponte H. Nó Conversor. Nó Mestre.



# ABSTRACT

This work proposes the development of a sorting algorithm used on control of multilevel cascaded H-bridge inverters. The sorting algorithm is based on CAN protocol property that deals with message arbitration, which is used to define priorities among the nodes of the network, where each node represents a H-bridge converter. Also the development of a master node that coordinates the converter nodes execution. As well as the implementation of the model of a photovoltaic panel and an inverter in the converter node, and simulation of a current control in the master. The sorting algorithm was validated by monitoring the messages in a CAN network with a master node and 3 slave nodes simulating inverters coupled to photovoltaic panels. It has been found that the three converter nodes provide the voltage required by the master node, alternating its operating state to keep within the optimum operating point. At each instant of requesting the voltage and current through the master node, it was found that only the converter nodes with the greatest need to supply or receive power from the network were conducting. At the end of sorting, the converter nodes alternate the states of their switches appropriately to keep themselves around the optimal operating point. The control of current of the network was executed by the master node, being validated by the change of the reference signal and monitoring of the behavior of the current until reaching the steady-state, where it is shown curves that represent the behavior of this current with respect to a reference value. As conclusion, there is the realization that it is possible to use the CAN network to perform the ordering algorithm for bridge converters.

**Keywords:** Controller Area Network. Sorting Algorithm. H-Bridge Converter.



# Agradecimentos

Primeiramente ao Cristo Jesus.

Aos meus pais Francisco de Assis Cartaxo e Francisca Neta por sempre me guiarem pelo melhor caminho, nunca duvidarem da minha capacidade de ir sempre mais longe e me apoiarem incondicionalmente.

Ao meus irmãos Kássio e Kaio pelo apoio e companheirismo.

Aos meus amigos Jonathan, Ramilo, Higo Thaian, João, Leonardo, André Higo, Pedro, Anderson, Gabriel, dentre outros que de alguma forma me ajudaram durante o curso.

Ao professor Carlos Alberto por me guiar no decorrer deste trabalho e não me permitir desistir do mesmo. E ao professor Lucas Hartmann por apresentar a ideia deste trabalho, além de todo o apoio ao longo dele.



# Lista de ilustrações

|  |    |
|--|----|
| Figura 1 – Inversores Ponte H em cascata ligados à rede . . . . .                      | 17 |
| Figura 2 – Dominância de bit . . . . .   | 23 |
| Figura 3 – Mensagem de Dados . . . . .   | 25 |
| Figura 4 – Campo de Arbitração no Formato Padrão . . . . .                             | 26 |
| Figura 5 – Campo de Arbitração no Formato Estendido . . . . .                          | 26 |
| Figura 6 – Campo de controle de uma mensagem de dados . . . . .                        | 26 |
| Figura 7 – Campo de arbitração e controle CAN FD padrão . . . . .                      | 27 |
| Figura 8 – Campo de arbitração e controle CAN FD estendido . . . . .                   | 27 |
| Figura 9 – Exemplo do critério de arbitração . . . . .                                 | 29 |
| Figura 10 – Ilustração do circuito de potência . . . . .                               | 31 |
| Figura 11 – Circuito inversor ponte completa . . . . .                                 | 32 |
| Figura 12 – Chaves $S_{PN}$ e $S_{NP}$ fechadas . . . . .                              | 33 |
| Figura 13 – Chaves $S_{PP}$ e $S_{NN}$ fechadas . . . . .                              | 34 |
| Figura 14 – Planta do Sistema e Controlador . . . . .                                  | 37 |
| Figura 15 – Planta do sistema e controlador estendido . . . . .                        | 37 |
| Figura 16 – Arquitetura do sistema proposto . . . . .                                  | 41 |
| Figura 17 – Diagrama de blocos do nó conversor . . . . .                               | 44 |
| Figura 18 – Diagrama de blocos do nó mestre . . . . .                                  | 45 |
| Figura 19 – Tiva C <i>Series</i> . . . . .   | 47 |
| Figura 20 – Fluxo de mensagens e condições de parada . . . . .                         | 53 |
| Figura 21 – Corrente do <i>grid</i> $I_L$ sob o efeito do controlador . . . . .        | 55 |
| Figura 22 – Fluxograma do conversor . . . . .  | 57 |
| Figura 23 – Fluxograma do mestre . . . . .   | 58 |
| Figura 24 – Representação de dois ciclos de mensagem . . . . .                         | 60 |
| Figura 25 – Primeira Mensagem Comparada ao Barramento CAN . . . . .                    | 61 |
| Figura 26 – Segunda Mensagem Comparada ao Barramento CAN . . . . .                     | 61 |
| Figura 27 – Terceira Mensagem Comparada ao Barramento CAN . . . . .                    | 62 |
| Figura 28 – Condições antes do funcionamento do sistema . . . . .                      | 63 |
| Figura 29 – Referência de corrente inicial $I_{Lref} = 4.54$ A . . . . .               | 64 |
| Figura 30 – Controle do sistema do mestre com referência $I_{Lref} = 8$ A . . . . .    | 64 |
| Figura 31 – Controle do sistema do mestre com referência $I_{Lref} = 12$ A . . . . .   | 65 |
| Figura 32 – Controle do sistema do mestre com referência $I_{Lref} = 10$ A . . . . .   | 65 |
| Figura 33 – Controle do sistema do mestre com referência $I_{Lref} = 4.54$ A . . . . . | 66 |
| Figura 34 – Controle do sistema do mestre com referência $I_{Lref} = 6$ A . . . . .    | 66 |
| Figura 35 – Fluxo do Algoritmo de Ordenação da CAN . . . . .                           | 68 |
| Figura 36 – Curva de tensão armazenada no capacitor . . . . .                          | 70 |

Figura 37 – Estados da chave  $S_{pp}$  . . . . . 71

# Lista de tabelas

|   |    |
|---|----|
| Tabela 1 – Relação Comprimento do Barramento × Taxa de transferência de dados | 23 |
| Tabela 2 – Lógica das Chaves de um Conversor Ponte Completa . . . . .         | 32 |
| Tabela 3 – Objeto disponíveis na rede CAN . . . . .                           | 50 |



# Sumário

|            |   |           |
|------------|---|-----------|
| <b>1</b>   | <b>INTRODUÇÃO</b>   | <b>15</b> |
| <b>1.1</b> | <b>Objetivos</b>  | <b>18</b> |
| 1.1.1      | Objetivos Específicos   | 18        |
| <b>1.2</b> | <b>Divisão do Documento</b>   | <b>18</b> |
| <b>2</b>   | <b>REFERENCIAL TEÓRICO</b>  | <b>21</b> |
| <b>2.1</b> | <b><i>Controller Area Network (CAN)</i></b>                                       | <b>21</b> |
| 2.1.1      | Características Gerais  | 21        |
| 2.1.2      | Definição da Camada Física  | 22        |
| 2.1.3      | Camada de Enlace  | 23        |
| 2.1.4      | Configuração e Transferência de Mensagens   | 24        |
| 2.1.4.1    | Tipos de Mensagens  | 24        |
| 2.1.4.2    | Mensagem de Dados   | 25        |
| 2.1.5      | <i>CAN with Flexible Data-Rate (CAN FD)</i>                                       | 27        |
| 2.1.6      | Critério de Arbitração  | 28        |
| 2.1.6.1    | Filtro de Mensagem  | 29        |
| <b>2.2</b> | <b>Conversor Multinível Ponte Completa em Cascata para Sistemas Fotovoltaicos</b> | <b>30</b> |
| 2.2.1      | Tensão de Saída em um Inversor Ponte Completa Ligado a um Painel Fotovoltaico     | 31        |
| 2.2.2      | Corrente no Filtro  | 35        |
| 2.2.3      | Algoritmo de Ordenação para Controle de Comutação das Chaves                      | 35        |
| <b>2.3</b> | <b>Estratégia de Controle Para o Sistema de Potência</b>                          | <b>36</b> |
| <b>3</b>   | <b>DESCRIÇÃO DA PROPOSTA</b>  | <b>41</b> |
| <b>3.1</b> | <b>Arquitetura do Sistema</b>   | <b>41</b> |
| 3.1.1      | Número de Nós Conversores   | 42        |
| <b>3.2</b> | <b>Descrição dos Componentes Físicos</b>  | <b>44</b> |
| 3.2.1      | Nós conversores   | 44        |
| 3.2.2      | Nó Mestre   | 45        |
| 3.2.3      | Materiais e Ferramentas Utilizados no Desenvolvimento do Sistema                  | 45        |
| 3.2.3.1    | Microcontrolador e Plataforma de Desenvolvimento                                  | 46        |
| <b>3.3</b> | <b>Desenvolvimento do <i>Firmware</i></b>   | <b>46</b> |
| 3.3.1      | Estabelecimento da Rede CAN   | 48        |
| 3.3.2      | Implementação do Algoritmo de Ordenação   | 50        |
| 3.3.3      | Proposta de Implementação do Simulador  | 52        |

|            |   |           |
|------------|---|-----------|
| 3.3.3.1    | Simulação da Corrente no Mestre . . . . .             | 52        |
| 3.3.3.2    | Cálculo do Controlador para a Corrente . . . . .      | 54        |
| 3.3.3.3    | Simulação da Tensão Armazenada no Capacitor . . . . . | 54        |
| 3.3.4      | Fluxograma . . . . .                                  | 55        |
| 3.3.4.1    | Fluxograma do Nó Conversor . . . . .                  | 56        |
| 3.3.4.2    | Fluxograma do Mestre . . . . .                        | 56        |
| <b>4</b>   | <b>RESULTADOS OBTIDOS . . . . .</b>                   | <b>59</b> |
| <b>4.1</b> | <b>Funcionamento da Rede CAN . . . . .</b>            | <b>59</b> |
| <b>4.2</b> | <b>Validação da Estratégia de Controle . . . . .</b>  | <b>62</b> |
| <b>4.3</b> | <b>Validação da Algoritmo de Ordenação . . . . .</b>  | <b>67</b> |
| <b>4.4</b> | <b>Funcionamento do Sistema Proposto . . . . .</b>    | <b>69</b> |
| <b>5</b>   | <b>CONCLUSÃO . . . . .</b>                            | <b>73</b> |
| <b>5.1</b> | <b>Propostas de Trabalhos Futuros . . . . .</b>       | <b>74</b> |
|            | <b>REFERÊNCIAS . . . . .</b>                          | <b>75</b> |

# 1 Introdução

Sustentabilidade e preservação do meio ambiente há um bom tempo vem sendo prioridade no desenvolvimento científico. A produção de energia, devido ao crescimento populacional e desenvolvimento da tecnologia, é cada vez mais necessária e em maiores quantidades. Como a pesquisa a cerca de combustíveis fósseis, que estão mais escassos e afetam negativamente o meio ambiente, e de meios convencionais para produção de energia elétrica (hidrelétricas, termelétricas) estão sendo desencorajadas com o passar dos anos, a produção de energia por fontes renováveis se firma como tendência no meio científico e econômico. Como parte integrante desse grupo de energias renováveis, a energia fotovoltaica desponta como a fonte de energia que mais cresceu nos últimos 6 anos, segundo SAWIN et al., 2011.

Um dos problemas envolvendo a utilização de energia fotovoltaica é a otimização da sua integração à rede de distribuição de energia elétrica. Como consequência, o interesse em pesquisas acerca da integração de sistemas de conversão de energia fotovoltaica tem crescido exponencialmente; tendo como foco a redução dos custos de produção e aumento da capacidade de potência que se consegue entregar à rede (COPPOLA et al., 2016b).

O conversor é um dos principais componentes do sistema fotovoltaico. É ele quem faz o intermédio da célula que produz energia e a rede de distribuição. Suas principais tarefas são a de atuar como inversor, convertendo a corrente contínua gerada pelo painel fotovoltaico em alternada, que é a utilizada pela rede. Além de garantir que o sistema sempre entregue a máxima potência permitida à rede ajustando dinamicamente a carga efetiva conectada. Esta segunda é alcançada por intermédio de algoritmos de busca do ponto máximo de potência MPPT (*Maximum Power Point Tracking*) (NAPOLI et al., 2015), o ponto máximo de potência está relacionado a menor distorção harmônica e a menor perda possível para o sistema, enquanto se garante a máxima confiabilidade (COPPOLA et al., 2016b).

No trabalho COPPOLA et al., 2016a é mencionado classificações típicas de conversores utilizados em sistemas fotovoltaicos, como por exemplo inversor central, inversor em cadeia, inversor multi-cadeia e o conversor CC-AC, também conhecido como micro-inversor (para uma descrição comercial do micro inversor é recomendado conferir SOLAR, 2017). Dentre estas topologias destaca-se a do conversor CC-CA, principalmente por seu alto desempenho com relação à máxima transferência de potência do sistema fotovoltaico e a carga ao inversor conectada. Seu melhor desempenho justifica-se devido ao inversor CC-CA dedicar um único conversor á um único módulo fotovoltaico. É mais abrangente em aplicações para sistemas de baixa potência e domésticos. Sua principal desvantagem é a

necessidade do uso de um transformador elevador ou de um conversor cc-cc para aumentar sua capacidade, o que aumenta seu custo em relação à outra topologia, a em cadeia por exemplo (KOURO et al., 2009).

Como o interesse é integrar a energia fotovoltaica para sistemas de alta potência, conversores multinível são o foco de interesse de estudos, como o apresentado em (COPPOLA et al., 2016a, COELHO, 2016, PEREIRA, 2008, FUENTES et al., 2017) por exemplo, pelo fato de essa configuração permitir que sejam conectados diversos conversores de mesma topologia aumentando o nível de tensão que se pode entregar à rede e ainda mantendo as características intrínsecas de cada conversor. Assim, é possível associar cada célula a um conversor e uni-los na topologia multinível aumentando a capacidade de energia entregue.

O conversor multinível em cascata ponte H (CHB) vêm ganhando destaque nessa aplicação, este é representado na Figura 1. A principal razão é a necessidade que cada ponte possui de uma fonte DC isolada, que é facilmente provida por painéis fotovoltaicos. Outro benefício é eliminar a necessidade do uso de transformadores elevadores ou conversores cc-cc do tipo *boost* em aplicações de alta potência, conectando quantos módulos ponte-H em série forem necessários. E não só isso, com a melhora na qualidade de energia, que é característica inerente ao conversor, há a redução do tamanho do filtro e da frequência de chaveamento, aumentando a eficiência do sistema (KOURO et al., 2009). É ilustrado na Figura 1 uma ideia da conexão dos inversores CHB com o sistema fotovoltaico e a rede.

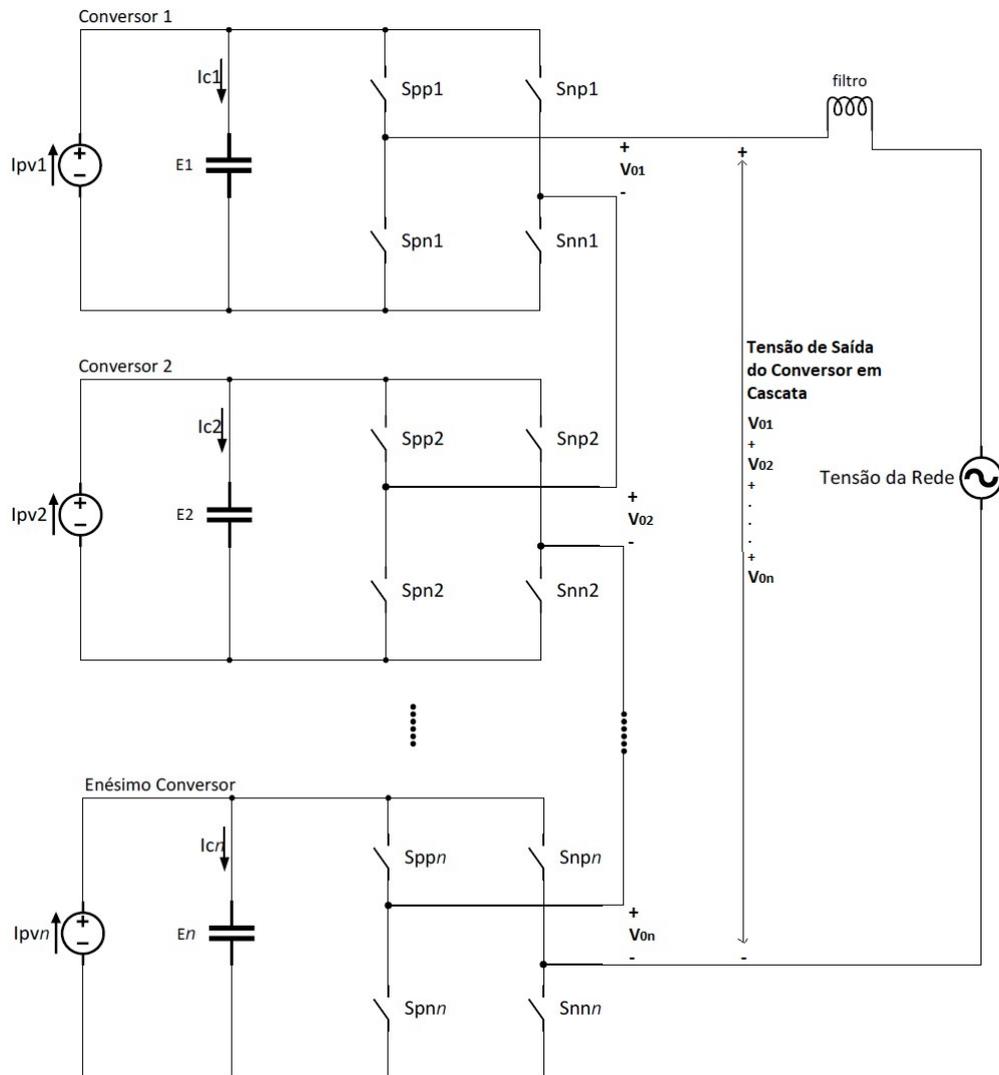
Para o estudo que será realizado é adotado a topologia de inversor CC-CA, onde são ligados em cascata conversores ponte H, como representado na Figura 1. A vantagem de haver a ligação destes conversores em cascata é que a junção dos mesmo elimina a necessidade do uso de um conversor CC-CC por parte do conversor CC-CA para elevação de sua tensão. Pois os conversores em cascata produzem um somatório de tensão na sua saída.

Na Figura 1 o painel fotovoltaico é representado como sendo uma fonte de corrente contínua. Os conversores ponte H estão conectados em série, sendo que cada um é responsável por um painel e pelo seu fluxo de energia, que se acumula no capacitor.

Controlar individualmente cada conversor é o principal obstáculo enfrentado nessa topologia. Ainda mais para aplicações utilizando energia solar, que é afetada pela temperatura e pelo nível de radiação sobre cada painel solar, tornando a produção de energia fotovoltaica desigual para cada célula do sistema. Esta desigualdade por sua vez afeta a distribuição de potência ao longo dos conversores ponte H (COPPOLA et al., 2016a). Em resumo, cada célula apresenta um diferente nível de energia, e cabe ao sistema de controle decidir quais células liberarão energia para a rede.

Levando em consideração o nível energético presente em cada célula, o controlador

Figura 1 – Inversores Ponte H em cascata ligados à rede



é responsável por decidir quais células irão compor a tensão que vai para o barramento, seja injetando ou recebendo energia, e quais irão pular o ciclo dentro de um determinado período de tempo. Para se determinar o estado de cada conversor é realizado um processo de ordenação que se baseia na necessidade energética do sistema e no ponto ótimo de tensão (MPPT). Lembrando que uma das missões da estratégia de controle é sempre garantir a operação estável do sistema (COPPOLA et al., 2016b).

Para que ocorra a ordenação é necessário que os nós do sistema fotovoltaico estejam em constante comunicação entre si. O controle pode ser localizado, sendo que cada nó conversor é controlado e transmite seus dados pertinentes por uma rede, ou centralizado, onde todas as informações vão para um único controlador, e este escolhe o estado de cada nó individualmente. Neste trabalho é proposto a realização de um controle localizado, cada nó comunica seu nível energético aos outros nós e, através de um critério de arbitragem, ocorre o processo de ordenação. Com a ordenação definida cada nó controla seu sistema de acordo com a sua necessidade - entregar energia à rede elétrica, receber energia ou ficar

inativo num dado período de amostragem.

Neste trabalho é proposto que a comunicação entre os nós conversores se dê através de um barramento serial. Este barramento é padronizado por um protocolo CAN (*Controller Area Network*). O protocolo CAN possui uma característica decisiva para sua escolha, seu critério de arbitragem. O barramento onde trafegam as mensagens da rede CAN possui um critério próprio de decisão de quem pode enviar mensagens, definindo prioridades para as mensagens que por ele trafegam baseadas num identificador que é inerente à cada mensagem, desta forma é possível realizar a ordenação dos painéis conversores baseando-se no sistema de prioridade de mensagem da rede CAN.

## 1.1 Objetivos

O objetivo deste trabalho de conclusão de curso é a síntese de uma rede CAN que executa um algoritmo de ordenação para controle de carga e descarga energética de painéis fotovoltaicos ligados a conversores ponte H em cascata. Deve-se com isso alcançar uma otimização da troca de energia entre os painéis e qualquer que seja o sistema ligado no outro lado dos conversores em cascata.

### 1.1.1 Objetivos Específicos

Para que seja possível montar e validar o funcionamento do proposto, pretende-se: i) identificar e otimizar o número de conversores que podem ser conectados em cascata; ii) desenvolver os meios necessários para implementação da rede CAN, sendo possível empregar o algoritmo de ordenação a partir desta implementação ; iii) simular o comportamento tanto do sistema, quanto dos painéis fotovoltaicos; iv) garantir o funcionamento do algoritmo de ordenação.

## 1.2 Divisão do Documento

Este trabalho de conclusão de curso tem seu desenvolvimento separado em cinco capítulos:

- a) Capítulo 1 - Neste capítulo é abordado uma introdução sobre o que será desenvolvido no trabalho.
- b) Capítulo 2 - Neste se encontra os conceitos tomados por base para desenvolvimento da proposta.
- c) Capítulo 3 - No Capítulo 3 está a descrição de como foi pensado e construído o sistema, contém sua arquitetura, a descrição dos equipamentos utilizados, um detalhamento do *firmware* utilizado e como se dá a simulação do sistema de potência.

- d) Capítulo 4 - Usa-se o Capítulo 4 para apresentar os resultados que puderam ser observados do trabalho desenvolvido.
- e) Capítulo 5 - Este é usado para comentários sobre o trabalho, assim como apresenta propostas de trabalhos futuros.



## 2 Referencial Teórico

### 2.1 *Controller Area Network (CAN)*

No começo da década de 1980 a utilização de componentes eletrônicos e de telemetria em veículos cresceu em números absurdos (BARBOSA, 2003). Gerando assim o aumento da quantidade de cabos responsáveis pela interconexão desses componentes, aumentando o custo de produção e o peso do automóvel. Pensando nisso a multinacional Robert Bosch GmbH apresentou, em fevereiro de 1986, perante a Sociedade de Engenheiros Automotivos - ou na sigla em inglês SAE - a ideia do que viria a ser um dos mais influentes protocolos de comunicação serial do mundo, o CAN (AUTOMATION, 2013).

O protocolo de rede CAN é padronizado mundialmente pela resolução da *International Society of Organization* ISO 11898, e possui atualmente os níveis de especificação 2.0A, 2.0B (ambas padronizadas pela ISO 11898-1) e CAN FD. A especificação 2.0A se propõe a garantir compatibilidade entre quaisquer dois nós que compõem uma rede. A especificação 2.0B vem como uma melhoria para o identificador de mensagens da especificação 2.0A, trazendo agora 29 bits de identificador, superando os 11 bits suportado pela 2.0A (TEAM, 2004). Em 2011 é apresentado então o padrão CAN FD, que chega com uma melhoria na taxa de transferência de dados pelo barramento; o campo de dados da mensagem é transferido em até 5 mega bits por segundo (Mbps) (TEAM, 2012), quando nas outras especificações a taxa de transferência máxima é de até 1 Mbps. A aplicabilidade destas especificações são regulamentadas pela SAE e utilizam como referência o modelo de camadas das rede de dados OSI (acrônimo de *Open System Interconnection*).

#### 2.1.1 Características Gerais

Criada para suportar controle em tempo real distribuído com um alto nível de confiabilidade se firmou desde muito tempo como opção para montagem de redes veiculares, e está ganhando cada vez mais espaço em aplicações industriais. Foi proposta para ambientes com alto nível de ruído induzido e grandes oscilações de alimentação e possui as seguintes características:

- a) Prioridade de mensagens - o protocolo possui um critério de arbitragem de mensagem. Nele mensagens de maior prioridade tem domínio sobre o barramento e transmitem primeiro que mensagens de menor prioridade. Um mesmo nó no barramento CAN pode enviar mensagens com diferentes níveis de prioridade, o que é uma vantagem sobre outros protocolos (I2C por exemplo) que arbitram sobre o nó.

- b) Garantia do tempo de latência.
- c) Flexibilidade de Configuração - esta característica está ligada à camada física da rede CAN. Pois a mesma pode assumir qualquer tipo de topologia (estrela ou cascata, por exemplo), tem uma taxa de transferência variável (62.5 kbps à 1 Mbps), comporta quantos nós for fisicamente possível, e qualquer novo nó pode ser adicionado à rede sem alterar a configuração de *software* ou *hardware* dos outros já existentes.
- d) Multimestre - a rede não possui um mestre definido, qualquer nó pode ter o domínio do barramento a qualquer momento, tudo depende do identificador de sua mensagem.
- e) Sinalização e detecção de erro - todo erro apresentado por um nó ou pelo próprio barramento é comunicado à todos os nós. Mensagens sinalizadas com erro são reenviadas assim que o barramento estiver ocioso.
- f) Distinção entre erros temporários e permanentes - erros sucessivos de um mesmo nó se somam e todo o barramento toma conhecimento; dependendo da quantidade de erros assinaladas, a rede tem autonomia para desativar o nó defeituoso. Caso o erro seja somente temporário, o nó poderá se recuperar e voltar a mandar mensagens (BARBOSA, 2003).

A partir destas características gerais é possível desenvolver os conceitos básicos que envolvem o protocolo de comunicação serial CAN, como: camadas física e de enlace, configuração e transferência de mensagens e critério de arbitração.

### 2.1.2 Definição da Camada Física

Compreende o meio físico de transmissão das mensagens, suas características são definidas pelo padrão ISO/OSI e conforme diz (CARISSIMI; ROCHOL; GRANVILLE, 2009) tem por função:

- a) Ativar e desativar uma conexão física;
- b) concatenar nós para a formação de uma conexão física entre todos eles;
- c) codificar e decodificar o canal de transmissão;
- d) controle e sincronização da transmissão e recepção de dados;
- e) supervisão, manutenção e controle da qualidade de enlaces físicos e conexões físicas.

É a camada física que limita o número de nós componentes de uma rede CAN, assim como a mínima distância possível que um dado pode percorrer mantendo uma conexão ativa e também a taxa de transferência de dados no barramento. A Tabela 1

Tabela 1 – Relação Comprimento do Barramento  $\times$  Taxa de transferência de dados

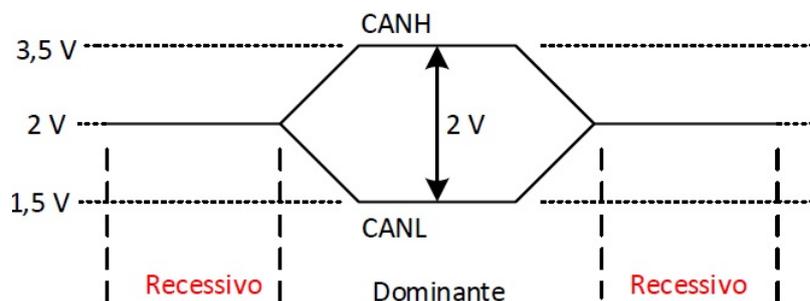
| Comprimento do Barramento (m) | Taxa de Transferência (kbps) |
|-------------------------------|------------------------------|
| 30                            | 1000                         |
| 100                           | 500                          |
| 250                           | 250                          |
| 500                           | 125                          |
| 1000                          | 62.5                         |

relaciona a taxa de transferência com o comprimento do barramento segundo (BARBOSA, 2003).

O cabo utilizado tem impedância característica de  $120 \Omega$  e resistores de terminação de  $120 \Omega$ , configurado para ser par trançado, com blindagem opcional.

Um dos principais conceitos que definem a camada física é o de dominância de bit. A dominância ocorre quando o estado de repouso de uma rede, representado por um bit recessivo, é interrompido por um bit dominante, fazendo assim com que o barramento saia do estado ocioso. A transmissão de dados pela rede CAN ocorre pelo valor da diferença de tensão entre dois fios o CAN *High* (CANH) e o CAN *Low* (CANL). Para que haja um bit dominante o CANH força uma tensão de  $3,5 \text{ V}$  e o CANL força uma tensão de  $1,5 \text{ V}$ , fazendo com que o bit dominante ocorra quando há uma tensão aproximada de  $2 \text{ V}$ ; o bit dominante tem nível lógico baixo (0). Na rede CAN o bit recessivo é mantido por dois resistores de *pull-up* conectados aos fios CANH e CANL, fazendo com que o nível lógico alto (1) represente o valor recessivo (BARBOSA, 2003). A Figura 2 é usada para ilustrar como é a dominância de bit.

Figura 2 – Dominância de bit



### 2.1.3 Camada de Enlace

A camada de enlace é subdividida em duas outras camadas, de acordo com a especificação 2.0B e são citadas:

- Logical Link Control* (LLC) - tem por função fornecer serviços para transferência de mensagens e por requisição remota de dados, decidir quais mensagens recebidas

devem ser aceitas e prover meios para gerenciamento de procedimentos de recuperação e sobrecarga de notificações.

- b) *Medium Acces Control* (MAC) - trata do protocolo de transferência de dados, controlando o enquadramento das mensagens, realizando arbitragem, checagem e sinalização de erro e confinamento de falhas. É na subcamada MAC que é decidido quando o barramento está livre para uma nova transmissão ou se está sendo recebida uma mensagem. Além de funcionalidades específicas da CAN como o *bit stuffing*, que atua enviando um valor de bit inverso à cada 5 bits consecutivos com mesmo valor.

Com relação ao confinamento de erros a camada de enlace se responsabiliza por confinar ou recuperar um nó defeituoso na rede CAN, enquanto que a camada física cuida do erro relacionado ao barramento.

#### 2.1.4 Configuração e Transferência de Mensagens

A mensagens na CAN tem formatos diferentes quanto às suas especificações, dispositivos que se baseiam na especificação 2.0A tem o formato de mensagem chamado padrão, dispositivos que se baseiam na especificação 2.0B tem a possibilidade de trabalhar com mensagens no formato padrão ou no estendido. A diferença entre os dois formatos é definido pelo tamanho do identificador da mensagens, o formato padrão envia mensagens com identificadores de até 11 bits, enquanto o formato estendido proporciona o envio de mensagens com identificadores de até 29 bits (TEAM, 2004).

##### 2.1.4.1 Tipos de Mensagens

De acordo com a especificação 2.0B existem quatro tipos de mensagens que podem se manifestar ou serem controladas dentro da rede CAN:

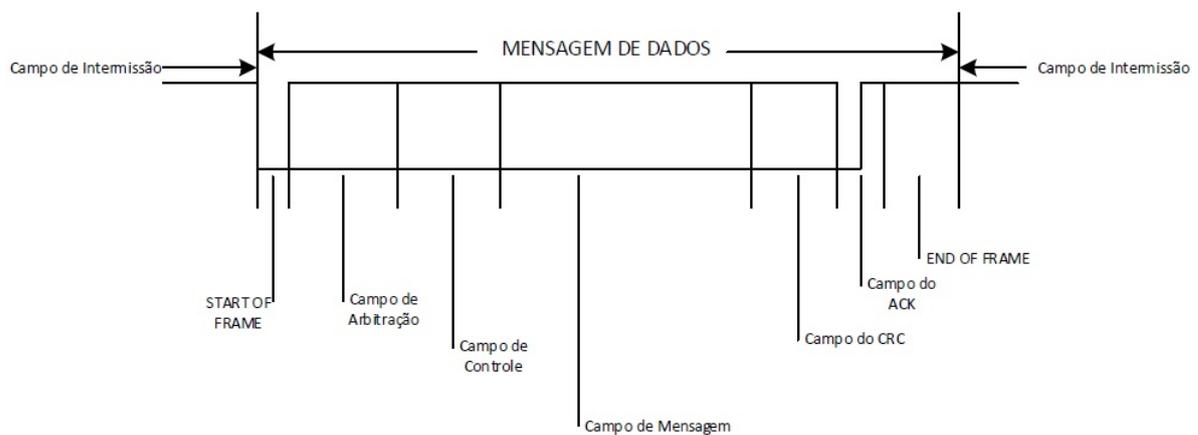
- a) Mensagem de dados - carrega dados de um transmissor até um receptor;
- b) Mensagem de requisição remota - é transmitido por um nó requisitando a transmissão de uma mensagem de dados com um mesmo identificador;
- c) Mensagem de erro - é enviado por um nó que detectou um erro no barramento;
- d) Mensagem de sobrecarga - é utilizado quando uma recepção necessita que próximas mensagens de dados ou requisição remotas atrasem e quando é detectado um bit dominante no primeiro e segundo bits do campo de intermissão de mensagens.

As mensagens de erro e de sobrecarga tem formato definido, enquanto as de requisição remota e de dados podem utilizar tanto o formato padrão quanto o estendido. Somente a mensagem de dados é de interesse para o trabalho realizado por este autor, portanto é o único tipo de mensagem abordado neste texto.

### 2.1.4.2 Mensagem de Dados

Uma mensagem de dados é utilizada quando um nó, que nesse caso é denominado transmissor, deseja enviar alguma informação para  $n$  quaisquer nós na rede, denominados nós receptores. A mensagem é composta de sete diferentes campos de bits: Início da mensagem (*Start of Frame*), campo de arbitração, campo de controle, campo do CRC (*Cyclic Redundancy Check*), campo do ACK (*Acknowledgement*) e fim da mensagem (*End of Frame*). Entre uma mensagem de dados e qualquer outra mensagem posterior ou anterior ainda existe o campo de intermissão, composto de bits recessivos que marcam a ociosidade do sistema. A Figura 3 é uma ilustração de como uma mensagem de dados é organizada (TEAM, 2004).

Figura 3 – Estrutura de uma Mensagem de dados.



Fonte:(TEAM, 2004)

O campo de Início da mensagem é comum tanto para a mensagem de dados quanto para a requisição remota, e marca o início de uma transmissão. Consiste de apenas um único bit dominante, e só pode ser enviado se o barramento for detectado como ocioso. Se comporta como um bit de sincronia, fazendo com que todos os nós componentes da rede o tomem como referencia.

O campo de Arbitração é o responsável pelo identificador da mensagem, e tem tamanho variável para os formatos padrão e estendido. Para o formato padrão o campo de arbitração possui 11 bits do identificador e o bit RTR. No formato estendido o campo de arbitração consiste de 29 bits do identificador, o bit SRR e o bit RTR. A partir da especificação 2.0B o bit reservado r1 do campo de controle passa a se chamar IDE e distingue o formato padrão do estendido. A Figura 4 e a Figura 5 são usadas para ilustrar a estrutura dos bits componentes e as diferenças entre os dois campos de arbitração.

Composto por 6 bits, o campo de controle possui diferenças entre o formato padrão e o estendido. Quadros no formato padrão possuem os bits de tamanho do campo de dados (*Data length code*), IDE, que é transmitido como dominante e os bit reservados r1 e r0.

Figura 4 – Campo de Arbitração no Formato Padrão.

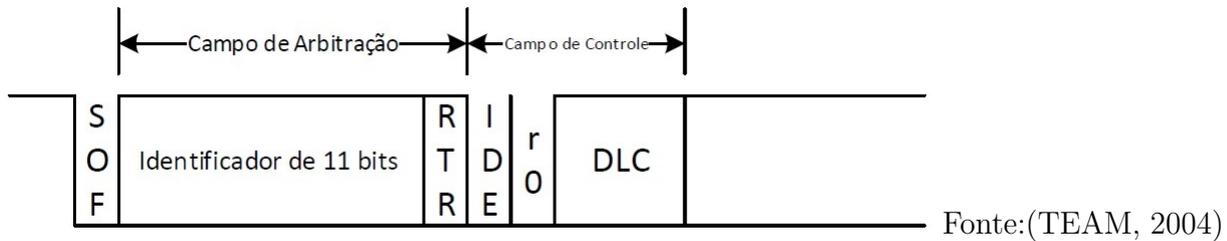


Figura 5 – Campo de Arbitração no Formato Estendido.

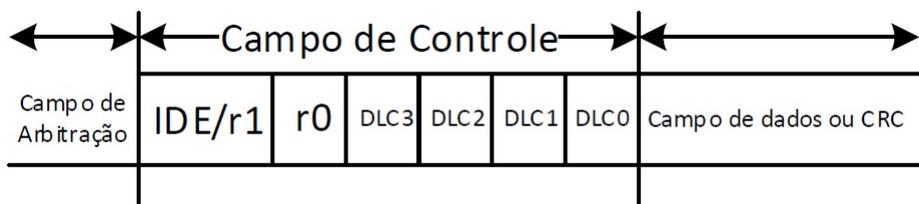


Fonte: (TEAM, 2004)

Os reservados devem ser enviados ambos dominantes, mas aceitos pelo receptor como dominante e recessivo em qualquer combinação.

O data length code é composto de 4 bits e indica quantos bytes estão sendo enviados no campo de dados, e varia de 0 a 8 bytes. Na Figura 6 são ilustrados os bits componentes do campo de controle.

Figura 6 – Campo de controle de uma mensagem de dados.



2004)

O campo de dados contém a mensagem a ser transmitida por uma mensagem de dados. Varia de tamanho dependendo da mensagem a ser enviada, podendo conter até 8 bytes. O formato da mensagem é *little endian*, ou seja, o bit menos significativo é enviado primeiro.

O campo de CRC trata do método de identificação de erros, consiste de 16 bits e checa a consistência da mensagem comparando o polinômio formador enviado pelo transmissor com o recebido no receptor.

O próximo campo é o de ACK e contém dois bits, o ACK SLOT e o ACK DELIMITER. O nó transmissor envia 2 bits recessivos, e quando há confirmação no envio da mensagem o nó receptor envia um bit dominante no ACK SLOT.

E por fim o *End of Frame* consiste numa sequência de 7 bits recessivos, que marcam

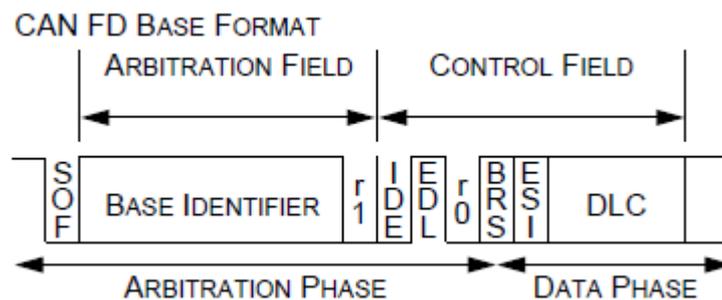
o fim da mensagem de dados.

Somando a quantidade de bits em cada campo da mensagem de dados esta possui, que são: 1 bit do campo de início de mensagem, 11 (formato padrão) ou 29 (formato estendido) bits do campo de arbitração, 6 bits do campo de controle, de 0 a 64 bits do campo de dados, 16 bits do campo de CRC, 2 bits do campo ACK, e 7 bits do *end of frame*, uma mensagem de dados da CAN possui de 44 à 108 bits o seu formato padrão e de 62 a 126 bits em seu formato estendido. Este valor, não leva em consideração os bits adicionados pelo bit *stuffing*.

### 2.1.5 CAN with Flexible Data-Rate(CAN FD)

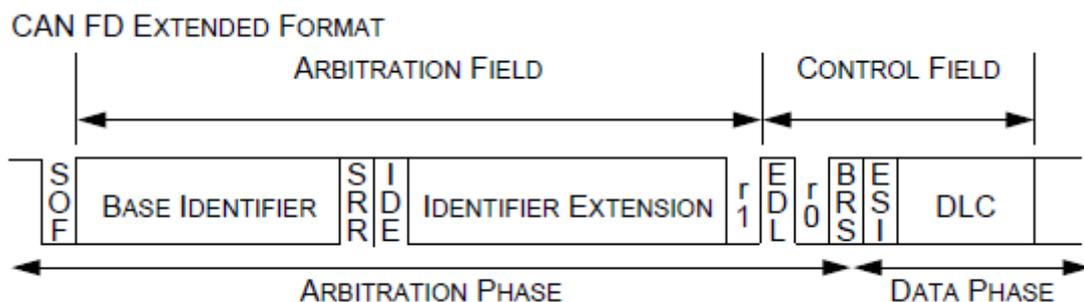
Baseado no protocolo CAN padrão, o CAN FD surge com o objetivo de aumentar a velocidade de transmissão de mensagens, mantendo a configuração física do protocolo em que é baseado (HARTWICH et al., 2012). De forma resumida, o protocolo CAN FD altera principalmente o formato das mensagens, adicionando bits para que se tenha um aumento no tamanho da mensagem de dados (chegando até 64 bytes). Aliado a isso, também proporciona um aumento na taxa de transferência, podendo chegar até 5 Mbps no campo de dados. Nas Figuras 7 e 8 são ilustrados os campos de arbitração e controle para os formatos de mensagem padrão estendido no CAN FD.

Figura 7 – Campo de arbitração e controle CAN FD no formato padrão.



Fonte: (TEAM, 2012)

Figura 8 – Campo de arbitração e controle CAN FD no formato estendido.



Fonte: (TEAM, 2012)

Com relação aos campos de arbitração o CAN FD não apresenta diferenças com o protocolo CAN que serviu de base, muito embora no campo de controle temos a adição dos bits *Extended Data Length* (EDL), *Bit Rate Switch* (BRS) e *Error State Indicator* (ESI), como também o aumento do alcance do *Data Length Code* (DLC).

O EDL é um bit recessivo (TEAM, 2012), e só existe para fazer a distinção entre o formato de mensagem do CAN padrão e o CAN FD. Vem sempre antes do bit dominante r0, que é um bit reservado para futuras expansões no protocolo.

O BRS determina a taxa de transferência do campo de dados. Se transmitido como recessivo há a troca da taxa de transferência no campo de dados, o valor da nova taxa de transferência é previamente configurado. O bit BRS sendo enviado como dominante indica que não há troca da velocidade de transmissão na fase de dados. O ESI é uma *flag* que indica erros. O DLC ainda possui 4 bits, mas agora varia de  $'0000_2'$  até  $'1111_2'$ , quando no CAN base variava de  $'0000_2'$  até  $'1000_2'$ .

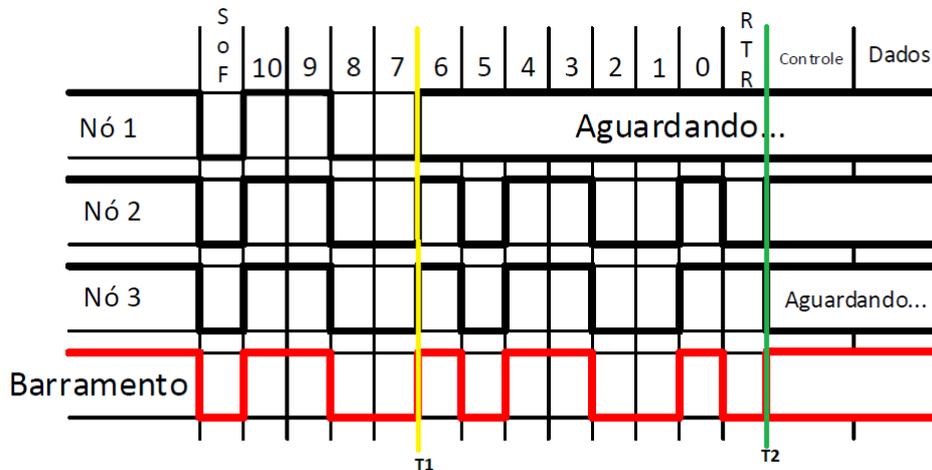
Este protocolo não é utilizado no trabalho apresentado, mas entra como uma proposta de melhoria no mesmo, pois o aumento da taxa de transferência dos dados é interessante para a expansão da rede, possibilitando um aumento no número de nós comunicadores que podem integrar o sistema.

### 2.1.6 Critério de Arbitração

O critério de arbitração é um dos principais diferenciais da rede CAN. O acesso ao meio por um nó da rede é baseado no conceito de CSMA/CD com NBDA (*Carrier Sense Multiple Access/ Collision Detection with Non-Destructive Bitwise Arbitration*), significando que os nós sempre verificam a disponibilidade do barramento antes de uma transmissão e também sua arbitração é não destrutiva, significando que a mensagem não é perdida durante o processo de arbitração. Ao verificar que o *status* da rede sinaliza o barramento como ocioso, as unidades pertencentes enviam suas mensagens, e tendo verificado que há mais de uma mensagem sendo enviada num mesmo instante, há o início do critério de arbitração.

O critério começa com os nós pertencentes à rede tentando escrever seus identificadores no barramento, e pelo princípio da dominância de bit, o identificador que escrever um bit dominante vai sobrescrever os bits recessivos que estiverem sendo enviados ao mesmo tempo, este procedimento é feito bit a bit do identificador. Ao fim do critério de arbitração, o barramento irá conter o identificador de menor valor (o bit 0 é dominante na rede CAN), fazendo que o critério de arbitração não desperdice a mensagem. Os nós que perderam a arbitração aguardam a ociosidade do barramento para reenviarem sua mensagem, dando início a um novo critério. Ilustra-se na Figura 9 um exemplo do funcionamento da arbitração na rede.

Figura 9 – Exemplo do critério de arbitração na rede CAN



Há na Figura 9 dois indicadores denominados T1 e T2, estes definem dois instantes. O instante T1 (barra vertical amarela) representa o instante onde o Nó 1 perde a arbitração, este fato ocorre devido ao Nó 1 enviar um bit recessivo, que é sobrescrito pelo bit dominante presente no Nó 2 e no Nó 3. O instante T2 (barra vertical amarela) representa o momento que o Nó 3 perde a arbitração, e conseqüentemente o Nó 2 vence. A partir deste instante o Nó 2 faz o envio de sua mensagem, enquanto os nós 3 e 1 aguardam o momento em que o barramento fica livre novamente.

#### 2.1.6.1 Filtro de Mensagem

Outra funcionalidade interessante com relação a troca de mensagens na rede é o filtro de mensagem. Receptores podem ser programados para receber mensagens com  $n$  identificadores diferentes, ou seja, um mesmo objeto pode receber um único identificador específico ou aceitar qualquer mensagem enviada pelo barramento. Esta propriedade só é possível por causa do filtro de mensagens.

A lógica de funcionamento é a seguinte:

1. O identificador do objeto de recepção é definido;
2. o filtro é definido através de uma máscara;
3. uma mensagem chega até o nó pelo barramento CAN;
4. é feita uma lógica "E" entre o identificador da mensagem transmitida e a máscara;
5. o resultado é comparado com o identificador do objeto de recepção, se for igual a mensagem é aceita pelo nó, senão é rejeitada.

Dessa forma, se o identificador do nó receptor for 0 e a máscara for 0, o nó receptor pode receber qualquer mensagem que lhe enviarem. E se o valor da máscara for em binário '111111111111<sub>2</sub>' (assumindo o formato padrão de mensagem), o nó receptor só aceita identificadores que sejam iguais ao seu.

## 2.2 Conversor Multinível Ponte Completa em Cascata para Sistemas Fotovoltaicos

Conversores multinível são assim denominados por serem a associação de células de comutação, dispostos em série e/ou paralelo, a fim de superar suas limitações físicas. São mais utilizados em sistemas de alta potência e tem como característica a melhoria da qualidade da tensão de saída, baixa distorção harmônica, o que diminui os componentes de filtragem e possibilita operação com uma alta eficiência (COELHO, 2016).

A topologia ponte H em cascata surgiu em 1975 e foi uma das primeiras topologias propostas. Utiliza conversores monofásicos em ponte completa para formar uma tensão senoidal de saída com vários níveis (PEREIRA, 2008). No geral é uma topologia mais simples. Atualmente, o foco está em aperfeiçoar estratégias de controle para essa tecnologia, seja para melhoria de sua eficiência seja na prevenção de falhas.

O conversor é responsável por adequar à rede elétrica a energia gerada por painéis fotovoltaicos, transformando em alternada a corrente contínua que sai dos painéis. Um exemplo de sua topologia é mostrado na Figura 1. Pela ilustração é possível notar que estão conectados em série  $n$  conversores ponte completa. Cada conversor é alimentado por um capacitor que serve como célula de armazenamento de energia produzida pelo painel fotovoltaico (PV). Cada conversor isoladamente é um inversor que gera uma tensão de saída  $V_{0n}$  igual a  $E_n$ , 0 ou  $-E_n$  (KOURO et al., 2009).

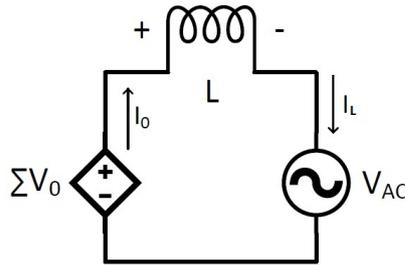
A tensão gerada por um inversor está dependente ao posicionamento das chaves  $S_{pp}$ ,  $S_{pn}$ ,  $S_{np}$  e  $S_{nn}$ . Com  $S_{pp}$  e  $S_{pn}$  fechadas ao mesmo tempo, toda a corrente  $I_{PV}$  do painel fotovoltaico circula por um braço em curto circuito, causando mal-funcionamento do sistema; o mesmo efeito ocorre para o braço negativo ( $S_{np}$  e  $S_{nn}$ ). Quando se opta por somente uma chave de cada braço abrindo, ocorre o processo de condução natural.

O somatório da tensão de saída de todos os inversores é uma fonte de tensão AC controlada. Um filtro é empregado entre a fonte controlada e a rede elétrica, sendo representada por um indutor. A tensão da rede elétrica é modelada como sendo uma fonte de tensão senoidal. Na Figura 10 ilustra-se o circuito descrito.

Onde:

- $\Sigma V_0$  é o somatório das tensões na saída de cada inversor individual em volts (V);

Figura 10 – Ilustração do circuito de potencia



- $I_0$  é a corrente de saída dos inversores em Ámpères (A);
- $L$  é a indutância do filtro em Henrys (H);
- $I_L$  é a corrente no lado do *grid* em Ámpères (A);
- $V_{AC}$  é a tensão da fonte que modela o *grid* de potência em volts (V);

Certos países exigem isolamento entre o sistema fotovoltaico e a rede, para isso pode ser considerado a adição de um transformador de baixa frequência com relação 1:1 no lado do *grid* de potencia (KOURO et al., 2009).

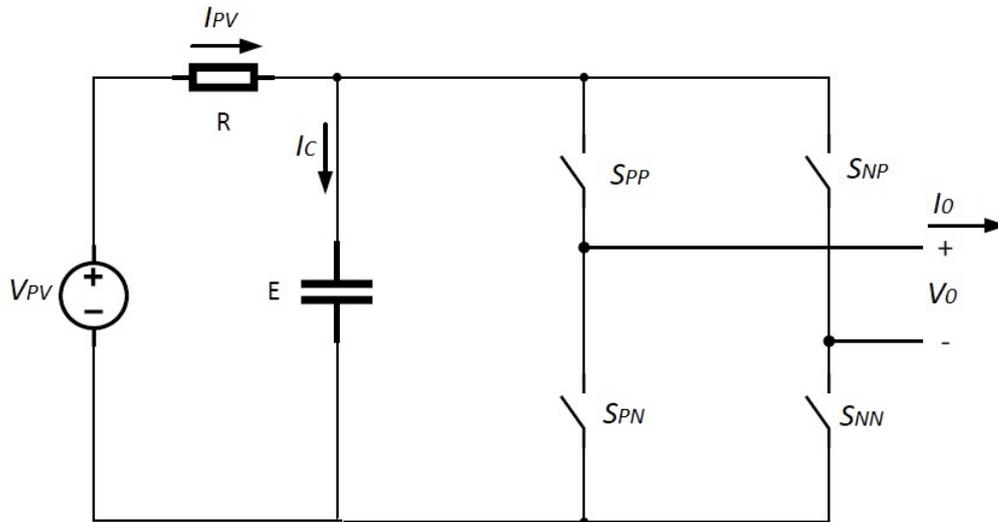
### 2.2.1 Tensão de Saída em um Inversor Ponte Completa Ligado a um Painel Fotovoltaico

O inversor ponte completa ligado a um painel fotovoltaico pode ser modelado pelo circuito ilustrado na Figura 11 o circuito de um inversor ponte completa ligado a um painel fotovoltaico. O painel fotovoltaico é modelado como sendo uma fonte de tensão em série com uma resistência, gerando uma corrente que varia de acordo com a carga do capacitor. A resistência é natural pois o painel não se configura como uma fonte ideal. Modelar o painel desta forma limita a tensão que pode ser armazenada no capacitor à uma tensão de saturação  $V_{sat}$  dependente do valor da resistência e a corrente que o painel consegue fornecer.

Onde:

- $V_{PV}$  é a tensão do painel fotovoltaico em Volt (V);
- $R$  é a resistência natural do painel em Ohms ( $\Omega$ );
- $I_{PV}$  é a corrente gerada no painel fotovoltaico em ámpere (A);
- $I_C$  é a corrente que passa pelo capacitor em ámpere (A);
- $E$  é a tensão no capacitor em volt (V);
- $S_{PP}$  é a chave positiva do braço positivo do conversor;

Figura 11 – Circuito inversor ponte completa ligada a um painel fotovoltaico



- $S_{NP}$  é a chave positiva do braço negativo do conversor;
- $S_{PN}$  é a chave negativa do braço positivo do conversor;
- $S_{NN}$  é a chave negativa do braço negativo do conversor;
- $V_0$  é a tensão de saída do inversor em volt (V);
- $I_0$  é a corrente de saída do inversor em ampere (A).

A corrente que sai do painel  $I_{PV}$  pode ser modelada através da equação 2.1.

$$I_{PV} = \frac{V_{Sat} - E}{R} \quad (2.1)$$

Assim, como já foi discutido no início da seção 2.2, a posição em que se encontram as chaves determinam o valor da tensão de saída do inversor. Elas se relacionam de maneira lógica, de acordo com a Tabela 2.

Tabela 2 – Lógica das Chaves de um Conversor Ponte Completa

| Combinação de Chaves   | Resultado |
|------------------------|-----------|
| $S_{PP} \times S_{PN}$ | 0         |
| $S_{NP} \times S_{NN}$ | 0         |
| $S_{PP} + S_{PN}$      | 1         |
| $S_{NP} + S_{NN}$      | 1         |

Por definição de tensão e corrente no capacitor tem-se  $I_C$ , que é a corrente que flui pelo capacitor sendo calculada na equação 2.2.

$$I_C = \frac{CdE}{dt} \quad (2.2)$$

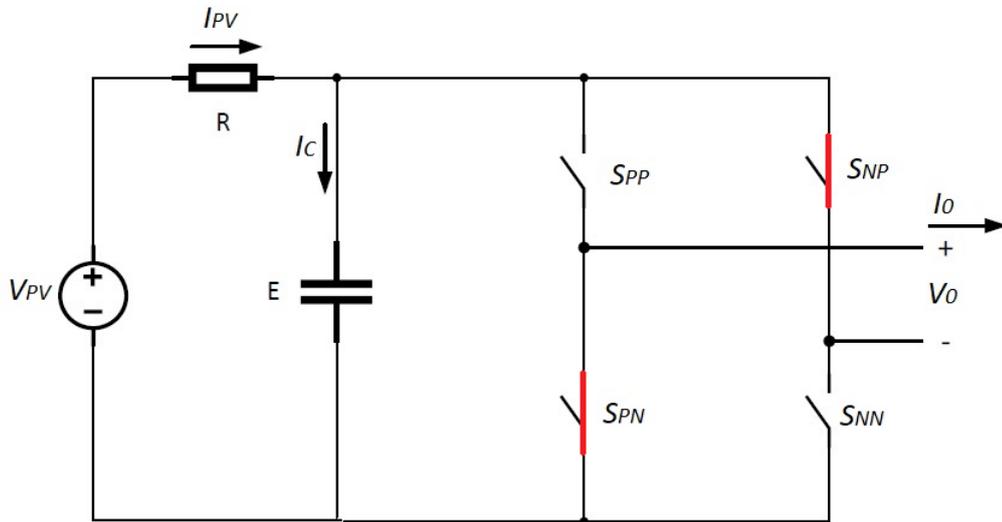
Onde  $C$  é a capacitância do capacitor. Outra forma de se obter o valor da corrente  $I_C$  é analisando o fluxo de potência do circuito da Figura 11. É possível observar que  $I_C$  é composto pelas contribuições das correntes  $I_{PV}$  e  $I_0$ . O valor de  $I_{PV}$  é definido pelo que o painel produz. No caso de  $I_0$  ele assume quatro valores com relação ao posicionamento das chaves e o fluxo de potência (carga ou descarga). Como o processo de condução só ocorre quando somente uma chave em cada braço está conduzindo, para avaliar o valor de  $I_0$  é necessário apenas conhecer o estado de uma chave de cada braço. Para a análise apresentada a seguir são considerados apenas os valores de  $S_{PP}$  e  $S_{NP}$ .

Quando  $S_{PP} = 0$  e  $S_{NP} = 0$ , para que haja manutenção da configuração dos painéis em cascata, há a necessidade que  $S_{PN}$  e  $S_{NN}$  sejam fechadas (representado pelo valor lógico 1). Nesta configuração não há abertura para a corrente  $I_0$  conduzir no conversor, logo seu valor é  $I_0 = 0$ . A situação inversa ( $S_{PN} = S_{NN} = 1$  e  $S_{PP} = S_{NP} = 0$ ) segue a mesma lógica e  $I_0 = 0$  novamente.

Para  $S_{PP} = 0$  e  $S_{NP} = 1$  há a condução nas chaves  $S_{PN}$  e  $S_{NP}$ . Analisando a Figura 12 pode-se concluir, a partir do sentido da corrente  $I_0$  na saída do painel que esta conduz para dentro do circuito conversor, fazendo com que o valor de  $I_C$  seja:

$$I_C = I_{PV} + I_0 \quad (2.3)$$

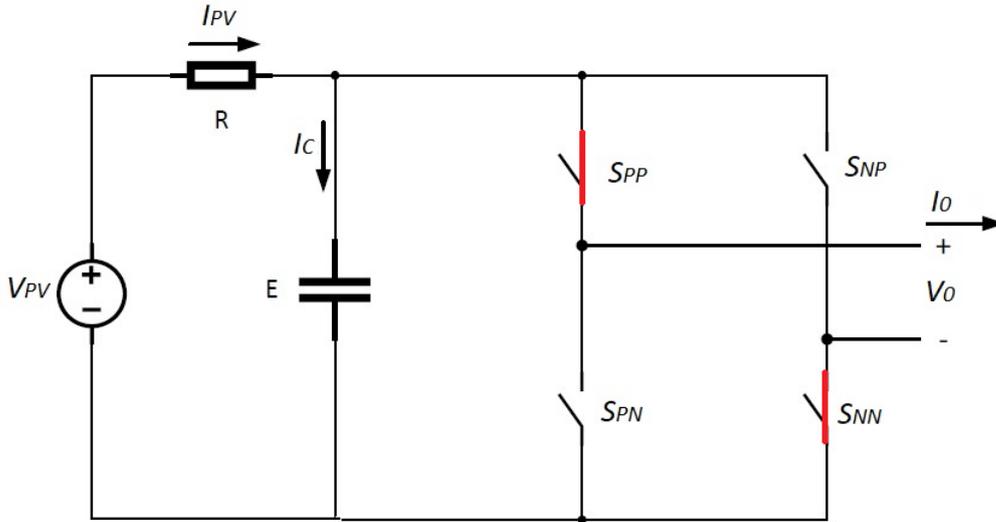
Figura 12 – Chaves  $S_{PN}$  e  $S_{NP}$  fechadas



Para  $S_{PP} = 1$  e  $S_{NP} = 0$  há a condução nas chaves  $S_{PP}$  e  $S_{NN}$ . Analisando a Figura 13 adotando o mesmo sentido do fluxo de potência da situação anterior, pode-se concluir que  $I_0$  conduz para fora do circuito conversor, fazendo com que o valor de  $I_C$  seja:

$$I_C = I_{PV} - I_0 \quad (2.4)$$

Quando a corrente  $I_0$  está circulando pelo circuito do conversor no sentido contrário ao adotado nas situações acima, ocorre uma inversão do valor de  $I_C$  com relação a posição

Figura 13 – Chaves  $S_{PP}$  e  $S_{NN}$  fechadas

das chaves. Neste caso, o valor das chaves sendo  $S_{PP} = 0$  e  $S_{NP} = 1$  a corrente que passa pelo capacitor é como na equação 2.4. Da mesma forma, no momento que as chaves tiverem o valor  $S_{PP} = 1$  e  $S_{NP} = 0$  a corrente  $I_C$  é calculada pela equação 2.3.

A partir das equações 2.3 e 2.4 é possível representar o valor de  $I_C$  em função das chaves  $S_{PP}$  e  $S_{NP}$  como é mostrado na equação 2.5.

$$I_C = I_{PV} + (S_{NP} - S_{PP}) \cdot I_0 \quad (2.5)$$

Substituindo a equação 2.5 na equação 2.2 e isolando o valor de  $\frac{dE}{dt}$  obtém-se:

$$\frac{dE}{dt} = \frac{1}{C} \cdot (I_{PV} + (S_{NP} - S_{PP}) \cdot I_0) \quad (2.6)$$

A tensão na saída ( $V_0$ ) é calculada em função da tensão  $E$  e do posicionamento das chaves do conversor. A análise é feita de forma análoga a  $I_0$ . Aqui quando  $S_{NP}$  e  $S_{PN}$  estão fechadas o terminal negativo de  $V_0$  é o mesmo que o positivo de  $E$ , logo:

$$V_0 = -E \quad (2.7)$$

De forma contrária, quando  $S_{PP}$  e  $S_{NN}$  estão fechados o terminal positivo de  $V_0$  é comum ao terminal positivo de  $E$ , então:

$$V_0 = E \quad (2.8)$$

Seguindo esta lógica, a tensão na saída do inversor  $V_0$  em termos da posição das chaves e do valor instantâneo de  $E$  é definido na equação 2.9.

$$V_0 = (S_{PP} - S_{NP}) \cdot E \quad (2.9)$$

### 2.2.2 Corrente no Filtro

Tanto o cálculo da tensão na saída do inversor quanto o da corrente que passa no indutor são vitais para o controle da corrente no *grid*. É com o estudo destes valores que pode-se estimar o comportamento do sistema que se pretende controlar. Como o trabalho inicialmente não envolve a construção física do circuito inversor, é necessário de algum modo modelar seu comportamento e simular uma aproximação do real. As equações definidas neste trabalho servem a esse propósito.

A tensão no indutor ( $V_L$ ) é expressa como:

$$V_L = L \cdot \frac{dI_L}{dt} \quad (2.10)$$

onde  $L$  é a indutância do filtro e  $I_L$  é a corrente que passa pelo indutor em Ampere (A).

Seguindo o sentido da corrente adotado na Figura 10, o valor da tensão  $V_L$  é escrito como a diferença de potencial entre  $\Sigma V_0$  e  $V_{AC}$ , na forma

$$V_L = \Sigma V_0 - V_{AC} \quad (2.11)$$

Isolando  $\frac{dI_L}{dt}$  na equação 2.10 e substituindo o valor encontrado na equação 2.11, obtêm-se:

$$\frac{dI_L}{dt} = \frac{V_L}{L} = \frac{\Sigma V_0 - V_{AC}}{L} \quad (2.12)$$

### 2.2.3 Algoritmo de Ordenação para Controle de Comutação das Chaves

O algoritmo de ordenação é a estratégia de controle adotada para o sistema fotovoltaico apresentado neste trabalho. Segundo NAPOLI et al., 2015 o controle do sistema fotovoltaico deve estar de acordo com os seguintes requerimentos:

- (i) controle independente de cada capacitor de enlace DC de forma a realizar o processo de MPPT em cada célula; (ii) entregar potência dos painéis fotovoltaicos para a rede elétrica ( $\Sigma V_0$  sempre maior que  $V_{AC}$ ); (iii) ter uma taxa de distorção harmônica (THD) adequada às normas da rede elétrica; (iv) a síntese correta da forma de onda gerada por  $\Sigma V_0$ ; (v) uma operação estável mesmo em condições de descasamento.

A cada um determinado período de amostragem do sistema, o erro de tensão ( $\Delta V_{PV}$ ), definido como a diferença entre a tensão ideal localizada pelo algoritmo de MPPT - aqui denominada como  $V_{PVref}$  - e a tensão atual armazenada no capacitor  $E$ , é calculado para cada célula componente do sistema. Os valores são então mapeados em ordem crescente. A primeira célula no vetor ordenado possui o maior erro positivo, significa que a tensão  $E$  medida nos terminais da célula armazenadora é maior que o valor de referência do MPPT  $V_{PVref}$  e por isso a célula precisa liberar mais energia para a rede do que as outras. De modo análogo ocorre de a última célula enfileirada possuir o menor erro

negativo, neste caso a célula que armazena energia deve ser carregada para que o seu valor de tensão medido se iguale com o de referência.

O algoritmo de ordenação também define um segundo vetor composto pelas tensões de referência apontadas pelo MPPT. Os elementos são posicionados na mesma ordem que o vetor de erros. Através de cálculos iterativos, que envolve a tensão final do inversor ( $\Sigma V_0$ ), o algoritmo define  $n$  células, onde as  $n - 1$  células do vetor ordenado são conectados diretamente à rede (entram no modo de condução); enquanto que a  $n$ ésima célula é forçada a operar chaveando em frequência (no modo PWM). Os inversores que não participarem desta etapa ficam inativos (não conduzem). O motivo de apenas uma célula estar modulada em PWM é para conseguir taxas reduzidas de perda por comutação. Além do mais, o algoritmo permite que os painéis tenham liberdade quanto à sua produção de energia, pois o algoritmo de MPPT é realizado individualmente em cada célula, blindando o sistema de falhas graves sob condições de descasamento ou falha do funcionamento de uma célula.

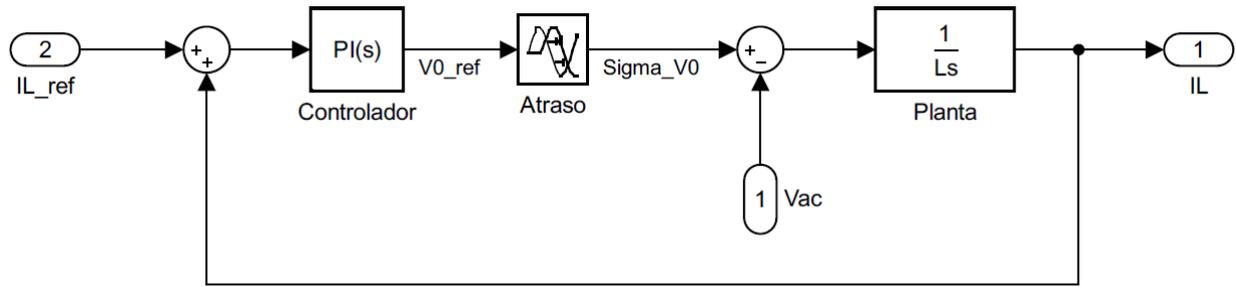
Se for necessário o tratamento da energia reativa na rede, em certos momentos o capacitor de armazenamento recebe energia da rede. Caso isso ocorra, ambos os vetores comutam suas posições, sendo que a primeira posição em cada vetor ocupado pelo nó com o menor erro negativo.

Para o sistema proposto neste trabalho, o algoritmo de ordenação vai ser realizado seguindo o critério de arbitragem existente na CAN. Como está explanado mais acima, o critério de arbitragem permite que o menor identificador ganhe o controle do barramento de comunicação e transmita sua mensagem; trazendo este conceito para o sistema fotovoltaico, a célula com a maior prioridade de descarga tem o menor identificador de mensagem. Dessa forma, o algoritmo de ordenação por *hardware* acontece devido ao critério de arbitragem juntamente com o de dominância de bit, que definem o controle do barramento.

## 2.3 Estratégia de Controle Para o Sistema de Potência

Nesta seção é tratado da forma como controlar a corrente que passa pelo barramento, para que a mesma atinja um nível de referência que ajuste a potência do barramento em função da tensão proveniente dos painéis fotovoltaicos em cascata. O controlador usa ganhos proporcional e integral para ajustar o erro de corrente, proveniente da diferença entre o valor de referência da corrente  $I_{Lref}$  e a corrente calculada no simulador  $I_L$ . O erro de corrente passa por um integrador e gera um valor de referência em tensão  $V0_{ref}$ . Este valor é o que os painéis devem produzir e entregar para o barramento. Após um atraso ocorrido devido o envio da referência e o recebimento do valor proveniente do sistema fotovoltaico, recalcula-se o valor de  $I_L$  através da equação 2.12 para ser realimentado e gerar um novo erro. A planta do sistema assim como o controlador são ilustrados na Figura 14.

Figura 14 – Planta do Sistema e Controlador

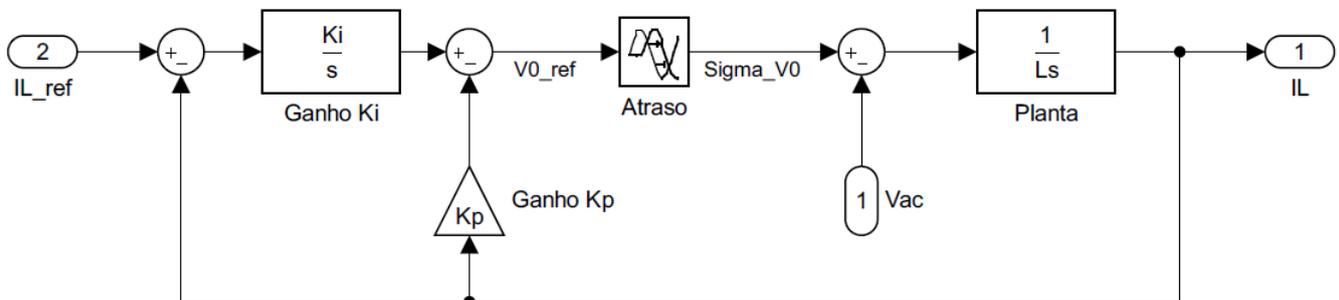


Onde:

- $IL\_ref$  é a referência de corrente  $I_{Lref}$ ;
- $V0\_ref$  é a tensão de referência para os painéis;
- $Sigma\_V0$  é a tensão produzida no sistema fotovoltaico;
- $Vac$  é a tensão no *grid*;
- $L$  é o indutor no filtro;
- $IL$  é a corrente no filtro.

O controlador age integrando o erro que é gerado entre o valor real da corrente  $I_L$  e sua referência  $I_{Lref}$ , multiplicando esta integração por um ganho integral  $K_i$ , que por sua vez irá subtrair com o valor de  $I_L$ , sendo que este está multiplicado por um ganho proporcional  $K_p$ . O controlador pode ser representado na forma ilustrada através da Figura 15.

Figura 15 – Planta do sistema e controlador estendido



Onde:

- $K_p$  é o ganho proporcional;

- $K_i$  é o ganho integral;

O cálculos de ambos os ganhos é descrito equacionando a malha de controle, conforme é mostrado a seguir.

Primeiramente, a relação entre a saída  $I_L$  da Figura 15 e a tensão calculada na planta ( $Sigma\_V0 - V_{AC}$ ), sem levar em consideração o atraso ( $V0\_ref = Sigma\_V0$ ) é como na equação 2.13.

$$\frac{I_L}{(Sigma\_V0 - V_{AC})} = \frac{\frac{1}{L \cdot s}}{1 + \frac{K_p}{L \cdot s}} \quad (2.13)$$

Reorganizando esta equação obtém-se a equação 2.14.

$$\frac{I_L}{(Sigma\_V0 - V_{AC})} = \frac{\frac{1}{L}}{s + \frac{K_p}{L}} \quad (2.14)$$

A partir da equação 2.14, têm-se a função de transferência  $\frac{I_L}{I_{Lref}}$  como é desenvolvido.

$$\begin{aligned} \frac{I_L}{I_{Lref}} &= \frac{\frac{1}{L \cdot \left(s + \frac{K_p}{L}\right)} \cdot \frac{K_i}{s}}{1 + \frac{1}{L \cdot \left(s + \frac{K_p}{L}\right)} \cdot \frac{K_i}{s}} \\ \frac{I_L}{I_{Lref}} &= \frac{\frac{1}{L} \cdot K_i}{s^2 + s \cdot \frac{1}{L} \cdot K_p + \frac{1}{L} \cdot K_i} \end{aligned} \quad (2.15)$$

Na forma descrita pela equação 2.15 a função de transferência toma o formato de um sistema de segunda ordem (NISE; SILVA, 2002), que é descrito na equação 2.16.

$$G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (2.16)$$

Onde:

- $\zeta$  é o amortecimento do sistema;
- $\omega_n$  é a frequência natural de operação do sistema em  $rad/s$ ;

Relacionando a equação 2.15 e a 2.16, pode-se obter os valores dos ganhos  $K_p$  e  $K_i$  através das equações 2.17 e 2.18 respectivamente.

$$\begin{aligned} \frac{K_p}{L} &= 2\zeta\omega_n \\ K_p &= \frac{2 \cdot \zeta \cdot \omega_n}{\frac{1}{L}} \end{aligned} \quad (2.17)$$

Como também:

$$\frac{K_i}{L} = \omega_n^2$$

$$K_i = \frac{\omega_n^2}{\frac{1}{L}} \quad (2.18)$$

Com as equações desenvolvidas nesta seção 2.3 e na seção 2.2 é possível montar todo o sistema de equações que simula o comportamento da rede proposta neste trabalho.



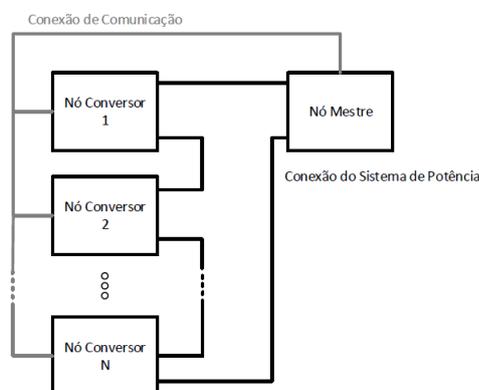
## 3 Descrição da Proposta

Para que este trabalho possa ser avaliado e a proposta inicial validada é necessário que sejam desenvolvidas as etapas mostradas ao longo deste capítulo. Elas estão divididas em: Arquitetura do Sistema, Descrição dos Componentes Físicos e Desenvolvimento do *Firmware*. Em Arquitetura do Sistema está uma descrição do diagrama de blocos do sistema, assim como o número de nós que podem compor a rede mantendo seu funcionamento regular e a descrição detalhada de cada bloco citado no diagrama. A Descrição dos Componentes Físicos trata de explanar os equipamentos utilizados para a construção física do sistema proposto. E o Desenvolvimento do *Firmware* mostra as etapas para concepção do programa que são executados nos microcontroladores em cada nó além da descrição de uma proposta de implementação do simulador do circuito de potência.

### 3.1 Arquitetura do Sistema

O sistema é composto por  $n$  nós conversores e um nó tido como mestre. Estes nós estão conectados entre si de duas formas, através da rede CAN e da ligação do circuito de potência, onde os nós conversores são ligado em cascata e o terminal positivo do primeiro conversor está ligado ao terminal positivo do nó mestre. O terminal negativo do  $n$ -ésimo conversor por sua vez se liga ao terminal negativo do nó mestre. O nó mestre é responsável principalmente por monitorar a situação da rede elétrica, enquanto os nós conversores monitoram seus respectivos sistemas fotovoltaicos. Na rede CAN todos os nós se comunicam pelo mesmo barramento, para trocarem mensagens entre si. Na Figura 16 é ilustrada uma ideia de como é a arquitetura proposta.

Figura 16 – Arquitetura do sistema proposto



Lembrando que a parte de alta potência não foi construída, sendo que da arquitetura proposta tem-se somente a ligação do barramento CAN, o sistema do *grid* foi simulado.

O número de nós que compõe a rede são definidos em torno da taxa de transferência do barramento CAN e o período de amostragem das grandezas do circuito. O cálculo desse número ( $n$ ) é discutido na seção 3.1.1. Todo o desenvolvimento mostrado leva em consideração que o tamanho físico do barramento CAN é desprezível, dessa forma não há queda na taxa de transferência devido à distância entre os nós.

### 3.1.1 Número de Nós Conversores

Para se definir o número de nós que podem ser adicionados na rede CAN, é preciso primeiro determinar como acontece o fluxo de mensagens num período de amostragem. O fluxo ocorre da seguinte forma:

- a) O nó mestre envia sua mensagem determinando o valor da tensão requerida  $V_{req.}$  e a corrente no *grid*  $I_L$ ;
- b) Os nós conversores enviam suas mensagens, contendo o valor da tensão  $E$  armazenada em seus capacitores;
- c) Inicia-se o critério de arbitração e a mensagem com menor identificador (maior erro positivo ou negativo, a depender do sentido da corrente e o sinal da tensão) ganha o controle do barramento;
- d) O valor do somatório das tensões na saída do inversor  $\Sigma V_0$  é atualizado em todos os nós;
- e) Compara-se  $\Sigma V_0$  com o valor de tensão requerida  $V_{req.}$ , e  $\Sigma V_0$  se for maior ou igual acaba a troca de mensagens naquele intervalo de amostragem;
- f) Se não, os nós restantes iniciam outro processo de arbitração e os passos c) à e) se repetem.
- g) Ao fim da troca de mensagens o barramento CAN fica ocioso e aguarda a próxima mensagem do nó mestre.

Com isso, dependendo do valor de  $V_{req.}$  é necessário que várias mensagens circulem pelo barramento. Então se chegar ao fim do período de amostragem e a condição descrita no item e) não se cumprir, o sistema vai falhar por confusão na troca de informações. Desta forma é possível definir que o sistema proposto opera sobre a condição teórica de tempo real. Para que não haja falha, é necessário definir um limite teórico do número de nós compondo o barramento.

O que se deve definir em primeiro lugar é o número de bits que podem trafegar no barramento durante um período de amostragem, aqui definido como  $Q$  em bits. Este valor depende da taxa de transferência do barramento, aqui definido como  $t_f$  em bits por

segundo (bits/s), e da frequência de amostragem definido pelo nó mestre, definido como  $f_s$  em Hertz (1/s). A relação se dá pela equação 3.1 mostrada.

$$Q = \frac{t_f}{f_s} \quad (3.1)$$

O número de conversores são representados pelo número de mensagens que elas podem enviar dentro de um período de amostragem. Sendo que cada mensagem representa um pacote de bits com tamanho  $L$ . Considerando que cada nó conversor só precisa enviar uma mensagem por critério de arbitração, e que o nó mestre sempre envia uma mensagem, o número de conversores  $n$  se relaciona com  $Q$  de acordo com a equação 3.2.

$$n \leq \frac{Q}{L} - 1,5 \quad (3.2)$$

Onde esta constante 1,5 que é subtraída do valor de  $n$  representa a mensagem do nó mestre, é considerado assim pois o campo de dados da mensagem do mestre é maior que o do nó conversor, considerando assim uma margem maior para seu envio. O número de conversores ser menor ou igual ao valor calculado é devido ao fato de  $n$  ser um número natural. Algum número de nós conversores maior que o calculado causaria mal funcionamento. Substituindo a equação 3.1 na equação 3.2 chega-se a relação entre o número de nós, a taxa de transferência do barramento e o período de amostragem, como é mostrado na Figura 3.3.

$$n = \frac{t_f}{f_s L} - 1,5 = \frac{t_f}{f_s L} - 1,5 \quad (3.3)$$

Num exemplo prático, para uma frequência de amostragem de 1 kHz, com uma taxa de transferência de 1 Mbps. É considerando a mensagem enviado por um nó conversor, com formato padrão e um campo de dados de 4 bytes, ou seja  $L = 76$  bits, o número de nós conversores é calculado como :

$$\begin{aligned} n &\leq \frac{t_f}{f_s L} - 1,5 \\ &\leq \frac{1 \times 10^6}{(1000) \cdot (76)} - 1,5 \\ &\leq \frac{1 \times 10^6}{0,76 \times 10^5} - 1,5 \\ &\leq 13,157 - 1,5 \\ n &\leq 11,657 \\ n &= 11 \end{aligned}$$

Neste caso é possível a comunicação de 11 nós conversores, mais o nó mestre pelo barramento CAN. Com este conceito apresentado, um detalhamento sobre o que compõe os nós é apresentado na seção 3.2.

## 3.2 Descrição dos Componentes Físicos

A descrição do funcionamento dos nós conversores, o nó mestre e como foi montado a rede CAN é mostrado nesta seção. Além de uma descrição dos materiais que foram utilizados para o desenvolvimento dos nós e da rede CAN.

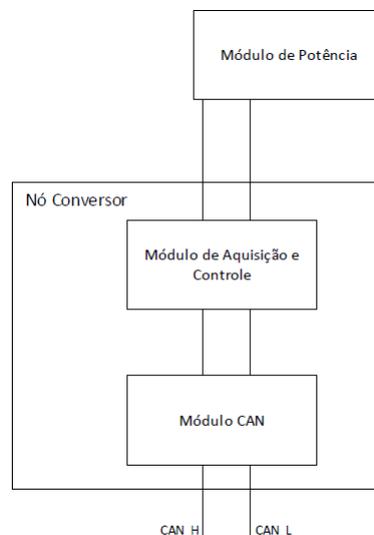
### 3.2.1 Nós conversores

Os nós conversores tratam de calcular o valor da tensão armazenada no capacitor ( $E$ ), como o circuito físico do conversor não está disponível, esta tensão é fornecida através do simulador. De posse desse valor, o nó conversor, é responsável por calcular o erro de tensão com relação ao valor de referência  $V_{PVref}$  e gerar um identificador para a mensagem que é transmitida pelo barramento CAN. Nesta mensagem está contido também o seu valor de  $E$ . Todos esses conceitos serão discutidos na seção 3.3.

O bloco nó conversor é composto de dois sub-blocos: o módulo de aquisição e controle e o módulo CAN, além de se comunicar com o módulo de potência. O sub-bloco de aquisição e controle é responsável por se comunicar com o circuito de potência, adquirindo as grandezas físicas necessárias para gerar o sinal de controle para este mesmo circuito. Além de definir as mensagens que devem ser enviadas pelo módulo CAN.

No módulo CAN está todo o controle da rede, ele trata da transmissão e recepção das mensagens do barramento, além do tratamento e sinalização de erros na rede CAN. É ilustrado na Figura 17 a estrutura do bloco nó conversor.

Figura 17 – Diagrama de blocos do nó conversor



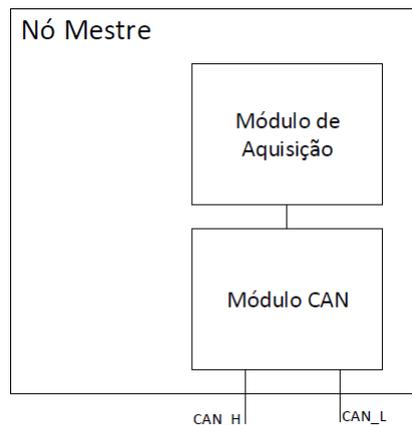
### 3.2.2 Nó Mestre

O nó mestre é responsável por fazer a aquisição da corrente que passa pelo filtro e o módulo da tensão da rede. Devido a indisponibilidade do circuito de potência, pelo fato do seu desenvolvimento não ser o foco deste trabalho, a corrente é calculada através de uma simulação e a tensão para este trabalho é considerada como em uma fonte ideal CC.

O bloco mestre é composto dois sub-blocos: módulo de aquisição e módulo CAN. O módulo de aquisição é responsável por gerar os valores das grandezas no simulador, e enviar estes valores para o módulo CAN. O módulo CAN por sua vez é responsável por estabelecer a comunicação com a rede CAN e através desta enviar suas mensagens.

O nó mestre é quem gera a mensagem principal da rede, pois é ela quem define o início do algoritmo de ordenação, fornecendo para os nós conversores o valor da tensão requerida  $V_{req}$ . O nó mestre também é responsável por definir o tempo de amostragem do sistema. A Figura 18 é usada para ilustrar o diagrama de blocos do nó mestre.

Figura 18 – Diagrama de blocos do nó mestre



### 3.2.3 Materiais e Ferramentas Utilizados no Desenvolvimento do Sistema

Os materiais utilizados para o desenvolvimento do sistema foram o microcontrolador TM4C123GH6PM, da *Texas Instruments*, montado sob a plataforma de desenvolvimento Tiva C Series, também da *Texas Instruments*. Embora tenha suporte para CAN, a plataforma não consegue gerar os sinais de CANH e CANL, que são os sinais que circulam pelo barramento serial da rede. Para que seja possível a geração destes sinais é necessário o uso de um *transceiver* CAN, que "traduz" os sinais de transmissão e recepção que circula pela Tiva em sinais de CANH e CANL. O *transceiver* SN65HVD231D, da *Texas Instruments* é então escolhido para desempenhar este papel. Para um aprofundamento melhor sobre o funcionamento do *transceiver*, recomenda-se ler seu *data-sheet* disponível em (INSTRUMENTS, 2001).

Acerca das ferramentas utilizadas destaca-se o uso do ambiente de programação disponibilizado pela *Texas Instruments*, o *Code Composer Studio* (CCS). É multiplataforma (sendo disponível para Windows, MacOs e distribuições Linux), gratuito e suporta a maioria dos microcontroladores produzidos pela empresa. O ambiente é baseado no Eclipse, e possui ferramentas semelhantes, com a adição de se poder gravar o *firmware* no microcontrolador a partir dele, além de ser possível uma depuração em nível de *hardware*. Outra ferramenta vital foi a interface de programação de aplicações (do inglês *Application Programming Interface*, API) TivaWare, que disponibiliza uma biblioteca de funções para programação e manipulação de todos os periféricos da Tiva. A TivaWare proporciona bibliotecas de funções para lidar com os periféricos do microcontrolador de forma mais intuitiva. No caso do periférico da CAN, a TivaWare trabalha com um sistema de objetos de mensagens. Neste sistema é possível criar objetos para os tipos de mensagens da CAN, onde é definido para cada objeto o valor do identificador, da máscara do identificador, do tamanho do campo de dados, da informação contida no campo de dados, entre outros. Há objetos de transmissão e recepção, além de transmissão e recepção remota.

### 3.2.3.1 Microcontrolador e Plataforma de Desenvolvimento

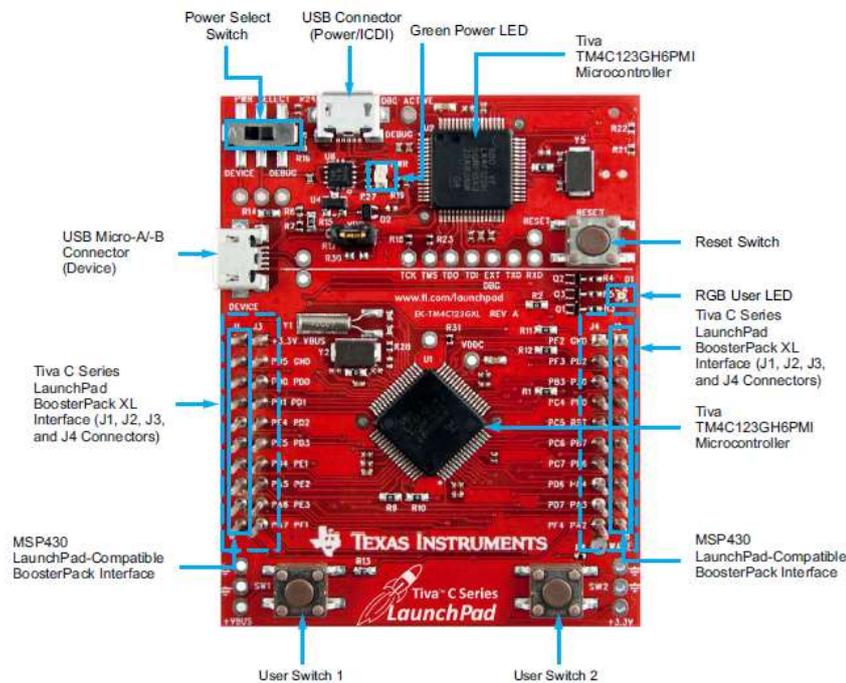
O microcontrolador TM4C123GH6PM é baseado no processador ARM Cortex-M4 de 32-bits, com 256 kB de memória *Flash*, 32 kB de SRAM, e opera à uma frequência de chaveamento de 80 MHz. À cerca dos periféricos pode-se citar seu controlador CAN, implementado com a versão CAN 2.0 A e B, além de 32 objetos de mensagem e filtro de mensagem individual para cada um. Outros periféricos utilizados neste trabalho são os temporizadores de propósito geral. O microcontrolador possui seis *timers* de 16/32 bits (dois *subtimers* de 16 bits, ou um *timer* de 32 bits) e mais seis de 32/64 bits. Para mais informações sobre o microcontrolador utilizado é recomendado checar seu *data-sheet* disponível em (INSTRUMENTS, 2013).

A plataforma de desenvolvimento Tiva C *Series* é configurada de forma que a inicialização do microcontrolador, além de seu manuseio seja o mais simples possível. A figura 19 abaixo é utilizada para mostrar a plataforma de desenvolvimento Tiva C *Series*, além de indicações sobre suas funcionalidades.

## 3.3 Desenvolvimento do *Firmware*

*Firmware* é o conjunto de instruções operacionais programadas diretamente no hardware de um equipamento eletrônico, é a designação clássica dos programas que são executados em microcontroladores. Todo o desenvolvimento do *firmware* está centralizado em dois objetivos: o estabelecimento da rede CAN, junto da implementação do algoritmo de ordenação e o desenvolvimento do simulador do *grid* de potência controlado. Dentro do

Figura 19 – Tiva C Series



Fonte:(INSTRUMENTS, 2013)

primeiro objetivo é preciso se trabalhar na ideia de construção dos objetos de mensagem, onde é fundamental definir qual a função de cada objeto, assim como qual o identificador de mensagem que qualquer objeto deve possuir. Além dos objetos, é necessário descrever as características da rede CAN, como taxa de transmissão e o número de nós. O algoritmo de ordenação é somente implementado nos nós conversores. São os nós conversores quem disputam o controle do barramento uns com os outros. Além disso são responsáveis por determinar o estado das chaves no conversor e o ciclo de trabalho do mesmo. Ao nó mestre fica a responsabilidade de determinar a tensão de referência  $V_{req}$  e calcular a corrente do *grid*, além de enviar estes dois valores via barramento CAN para os nós conversores.

O segundo objetivo é necessário para se discutir o funcionamento da estratégia de controle, mas é opcional para validação do algoritmo de ordenação. Para se desenvolver o simulador é necessário calcular o valor da corrente a partir da equação 2.12, onde o valor de  $I_L$  é calculado usando-se o valor da tensão  $\Sigma V_0$  enviada via rede CAN. A indutância de  $L$  como também a tensão no barramento  $V_{AC}$  são constantes. O valor de  $I_L$  é então utilizado junto de um valor de referência  $I_{Lref}$  para gerar um erro  $I_{error}$ , que passa pelo controlador descrito na seção 2.3 para gerar a tensão de referência que é enviado via rede CAN para os nós escravos  $V_{req}$ .

Todo o desenvolvimento citado na seção 2.2 para a simulação do sistema de potência deve ocorrer no nó mestre. Os nós conversores calculam a tensão  $E$  armazenada

nos capacitores, além de operarem a comutação das chaves do inversor, para que operem de acordo com a necessidade energética do sistema (armazenar ou fornecer energia para o *grid*).

Nesta seção serão descritos os três procedimentos citados: estabelecimento da rede CAN, implementação do algoritmo de ordenação e implementação do simulador, discutindo o desenvolvimento de cada procedimento no nó mestre e no nó conversor. Ao final, é mostrado um fluxograma do *firmware* desenvolvido para cada nó.

### 3.3.1 Estabelecimento da Rede CAN

Para validar o funcionamento do sistema são utilizados quatro microcontroladores, onde três deles representam nós escravos e Um o mestre. Define-se a taxa de transferência de todos os nós como sendo 1 Mbps. É preciso dizer também que Um único *firmware* foi utilizado para todos os nós escravos, salvo alguns valores de constantes adotadas (no que tange valores de referência principalmente).

Tomando por base o que foi definido para o nó mestre, tem-se que sua mensagem é a mais importante do ciclo, por isso é criado um objeto de transmissão com identificador '00000000000<sub>2</sub>', o identificador com maior prioridade sobre a rede. Este objeto de transmissão tem também uma campo de dados de 8 bytes, que é suficiente para fornecer o valor da soma da tensão requerida na saída dos inversores, que neste caso é  $V_{req}$ . e a corrente no *grid*  $I_L$ , onde são reservados 4 bytes para a tensão e 4 bytes para a corrente. Com o objetivo de se alcançar uma melhor precisão dos dados enviados, é multiplicado o valor real por um ganho de 10000 (dez mil). Desta forma são conservados cinco casas decimais do valor original, pois o dado que trafega pelo barramento é inteiro sem sinal. Para indicar o sinal (positivo ou negativo) do dado enviado é reservado o bit mais significativo do campo da tensão e da corrente, quando este último bit está com valor 1, significa que o sinal é negativo.

O mestre também possui um objeto de recepção responsável por adquirir as mensagens enviadas pelos nós conversores. O objeto de recepção possui identificador '00000000000<sub>2</sub>' e máscara também com este valor, garantindo que este objeto receba qualquer mensagem enviada por outro nó pelo barramento.

O objeto de transmissão no nó mestre envia mensagens em intervalos de 1 ms, desta forma assume-se que o conversor opere numa frequência de 1 kHz. Para que isso seja possível é definido um módulo temporizador no microcontrolador, que a cada período habilita um sinal para que o objeto possa transmitir uma mensagem. Antes que seja efetivamente enviada, uma função define o valor da tensão  $V_{req}$  e da corrente  $I_L$  que devem ser informados e atribui esse valor ao campo de dados da mensagem de transmissão. Todo esse procedimento considera que o tempo de processamento é desprezível, pois está sendo

utilizado um *clock* no sistema de 80 Mega Hertz (MHz). Quando uma mensagem chega com um identificador que é compatível com o filtro do objeto de recepção, é utilizado o valor da tensão informado pela mensagem para atualizar o valor de  $\Sigma V_0$ . Ao final do ciclo de transmissão de mensagens pelos nós conversores, o valor de  $\Sigma V_0$  é utilizado no cálculo da corrente  $I_L$ .

O nó conversor possui um objeto de transmissão e dois de recepção. O gerenciador da rede CAN no microcontrolador (TivaWare) define uma prioridade própria para seus objetos, onde o objeto zero é o de maior prioridade. Dessa forma, se uma mensagem chegar em um nó e dois objetos de recepção estiverem aptos à recebê-la, o menor objeto fica com a mensagem. Esta informação é importante para diferenciar os dois objetos de recepção do nó conversor. O objeto zero vai ser responsável por receber a mensagem que chega do nó mestre, e tem um identificador com valor  $'0000000000_2'$ , e uma máscara com valor  $'1111111111_2'$ , significando que este objeto só pode receber mensagens com identificador zero. É o objeto zero que recebe a mensagem do mestre, ele quem sinaliza o início da transmissão. Quando este objeto recebe uma mensagem, através de uma rotina de interrupção (que aliás sempre ocorre quando mensagens são recebidas ou transmitidas em qualquer nó) é iniciado um temporizador, com um valor de aproximadamente 1% do período de amostragem, para que haja uma sincronização de todos os nós conversores na etapa de transmissão. Ao fim da contagem do temporizador, através de uma rotina de interrupção pelo estouro deste, é solicitado ao objeto de transmissão um envio de mensagem.

O objeto 1 é de transmissão, ele trás consigo no campo de dados o valor de tensão atual armazenada no capacitor  $E$ . O identificador do objeto de transmissão é definido pelo erro percentual entre a tensão de referência MPPT e a tensão armazenada  $E$ . O nó transmite pela primeira vez quando o temporizador iniciado pelo objeto de recepção do mestre estoura, havendo assim a solicitação de envio da mensagem, é necessário que nesta etapa ocorra sincronia de todos os nós conversores. Com o envio da mensagem é iniciado o processo de arbitragem no barramento. Quando o envio é bem sucedido, o objeto fica inativo e aguarda o próximo período de amostragem. No caso de não conseguir enviar sua mensagem ao fim do processo de ordenação, o objeto também é desativado e aguarda a próxima mensagem chegar do mestre.

O segundo objeto de recepção é o de número 2, e recebe as mensagens dos nós conversores que ganham o critério de arbitragem, este tem a menor prioridade sobre o barramento CAN. Para o objeto de recepção 2 não há filtro de mensagem, podendo receber qualquer mensagem que esteja no barramento. Foi definido este objeto como o de menor prioridade para que não haja a possibilidade de o mesmo receber a mensagem enviada pelo nó mestre, garantindo que a mensagem enviada pelo mestre seja recebida pelo objeto de recepção 0 dentro do nó conversor. Este nível de prioridade é definido a

Tabela 3 – Objeto disponíveis na rede CAN

| Tipo de Nó | Tipo de Objeto | Número do Objeto | Identificador (hexa)                                      | Filtro de Mensagem  | Função   |
|------------|----------------|------------------|---|---|--|
| Mestre     | Transmissão    | 1                | 0x00  | –   | Mensagem principal do sistema, define o início do algoritmo de arbitração                          |
| Mestre     | Recepção       | 0                | –   | Filtro de Mensagem desligado                                  | Recebe os valores do objeto de transmissão dos nós conversores para compor o valor de $\Sigma V_0$ |
| Conversor  | Transmissão    | 1                | Depende do erro de tensão definido nas equações 3.4 e 3.5 | –   | Mensagem que disputa a arbitração e contém o valor da tensão armazenada no capacitor               |
| Conversor  | Recepção       | 0                | –   | Definido para receber somente mensagens do identificador 0x00 | Objeto que recebe a mensagem principal do sistema  |
| Conversor  | Recepção       | 2                | –   | Filtro desligado, assim qualquer mensagem é aceita            | Objeto responsável por receber as mensagens enviadas pelos nós conversores                         |

nível de registrador no microcontrolador, mas é gerenciado de forma mais simples pela API TivaWare. Basicamente a função deste objeto de recepção é somente receber as mensagens dos nós conversores quando elas chegarem e salvar seu campo de dados em alguma variável, para que seja recalculado o valor necessário. A Tabela 3 é usada para resumir as informações apresentadas sobre os objetos presentes nos dois tipos de nós.

### 3.3.2 Implementação do Algoritmo de Ordenação

O algoritmo inicia quando é definido o valor de  $V_{req.}$  pelo nó mestre. Esse então é enviado a partir do objeto de transmissão do mestre para todos os nós conversores presentes. Desta forma, o valor é armazenado pelos nós escravos e serve como ponto de parada do algoritmo de ordenação. Todos os nós escravos carregam suas mensagens no barramento ao mesmo tempo. O campo de dados desta mensagem deve ser preenchido com o valor da tensão armazenada no capacitor  $E$ , no mesmo instante o identificador da mensagem é carregado respeitando a equação 3.4. Esta se baseia no erro relativo entre a tensão do capacitor  $E$  e sua tensão de ponto ótimo  $V_{PVref}$ . Aqui a constante 1000 é utilizada como um ganho para o valor percentual, já que o identificador utiliza números inteiros e é necessário aproveitar casas decimais do valor calculado para o erro. A constante 1001, por sua vez, é uma forma de garantir que o menor identificador da mensagem do objeto de recepção seja 1. Desta forma, é preservado o valor do identificador do mestre, que é 0.

$$ID = 1001 + \left( \frac{1000 \cdot (V_{PVref} - E)}{V_{PVref}} \right) \quad (3.4)$$

Note que o valor calculado na equação 3.4 considera o fluxo de potência saindo dos conversores. Desta forma, se a tensão armazenada no capacitor for maior que o valor de referência do ponto ótimo da mesma  $V_{PVref}$ , a tendência é diminuir o valor do identificador,

dando maior prioridade para este nó. O nó possuir a maior prioridade no barramento significa que o mesmo está com a maior tensão armazenada, e portanto com a maior necessidade de descarregar energia. Não foi implementado neste trabalho o cálculo do algoritmo de MPPT, portanto o ponto ótimo é definido como sendo a metade da tensão de saturação do capacitor seguindo o modelo do painel fotovoltaico. Embora seguindo o que é mostrado na seção 2.2, é possível que haja outra forma de se calcular o identificador da mensagem. Esta se refere à situação em que o sentido da corrente e o sinal da tensão são opostos, o que caracteriza um fluxo de potência entrando nos conversores. Considerando esta situação, o identificador da mensagem é como na equação 3.5.

$$ID = 1001 - \left( \frac{1000 \cdot (V_{PVref} - E)}{V_{PVref}} \right) \quad (3.5)$$

Adotando este sentido, o nó com a maior prioridade é aquele que mais necessita receber energia, que acaba sendo o com menor identificador.

Voltando para o início do critério de arbitração, o nó que vence a arbitração aguarda o fim de todo o critério para então comutar sua chave fornecendo o sentido da tensão  $\Sigma V_0$  necessário ( $S_{np} = 0$  e  $S_{pp} = 1$ , ou  $S_{np} = 1$  e  $S_{pp} = 0$ ), não realizando mais nenhuma tentativa de envio até o próximo ciclo. No caso específico de o conversor ser o último à ganhar a arbitração, o chaveamento do mesmo é coordenado por um pulso PWM ciclo de trabalho igual à fração do que o nó conversor necessita enviar. Por exemplo, se a tensão requerida ( $V_{req}$ ) for 100 V e o somatório de tensões  $\Sigma V_0$  for 90 V, e o último nó à ganhar o processo de arbitração tiver 50 V, ele fornece sua tensão como uma onda PWM com ciclo de trabalho de 20%, dessa forma ele entrega somente 1/5 da sua tensão que são os 10 V restantes para finalizar o somatório da tensão necessária.

Na rede CAN, uma vez iniciada a transmissão de uma mensagem por um nó, este tentará o envio até conseguir. Logo, se o mesmo perde a arbitração, no instante em que detecta a ociosidade do barramento o nó tenta a re-transmissão da mensagem. Quando não mais é necessário o envio de mensagens, qualquer mensagem enviada pelos nós conversores devem ser apagadas, para que não haja uma nova tentativa de envio.

Quando é transmitida uma mensagem pelo barramento, os nós que perderam a arbitração atualizam o valor do somatório de tensão na saída com o valor do  $E$  informado na mensagem enviada pelo conversor que ganhou a arbitração. Na hipótese de haver a necessidade de mais uma mensagem, os nós aguardam a ociosidade do barramento para re-enviar a sua, dando início a mais uma arbitração. Não sendo mais exigido mensagens pelo barramento, ou seja  $\Sigma V_0 \geq V_{req}$ , os nós restantes limpam suas mensagens que iriam para o barramento, forçam o estado das suas chaves para  $S_{np} = 0$ ,  $S_{pp} = 0$  e aguardam um novo ciclo do algoritmo de ordenação, iniciado por uma nova mensagem do mestre.

Havendo a possibilidade de todos os nós conversores enviarem suas mensagens e ainda assim  $\Sigma V_0 < V_{req}$  é conveniente formular outra forma de determinar o fim do critério

de arbitração. É proposto que um temporizador compute o tempo que o barramento CAN fica ocioso, no momento que o temporizador alcance um intervalo fixo correspondente ao campo de intermissão, 3 bits para nós ativos e 11 bits para nós passivos, mais uma taxa de tolerância. Este intervalo fixo então é adotado como o tempo de envio de 15 bits pelo barramento CAN.

Fisicamente, a condição onde o somatório de tensões não satisfazem a tensão requerida sinaliza mal funcionamento do sistema, acarretando em danos ao mesmo. Na simulação aqui proposta todos os nós conversores forçarão suas chaves para  $S_{np} = 0$  e  $S_{pp} = 0$  e aguardarão um novo ciclo. Pelo lado do mestre é onde ocorre o problema, pois desta forma o valor do somatório das tensões não é atualizado fazendo com que o a corrente aumente o seu erro, levando mais tempo para se estabilizar. A Figura 20 é usada para ilustrar como é o fluxo de mensagens num ciclo, assim como as duas formas onde se sinaliza o fim do algoritmo de ordenação.

Na situação 1 o somatório de tensões dos conversores consegue ser maior que a tensão requerida fazendo com que o algoritmo de ordenação chegue ao seu fim. Ao contrário da situação 2, onde após o envio da mensagem pelo escravo 3 o requisito há um intervalo de tempo sem mensagens sendo enviadas, quando o temporizador estoura e conclui-se que o requisito  $\Sigma V_0 \geq V_{req}$  não é atingido. Desta forma, chega-se também ao fim do algoritmo de ordenação, sendo que aqui todos os nós tendem a forçar o estado de suas chaves a zero.

### 3.3.3 Proposta de Implementação do Simulador

#### 3.3.3.1 Simulação da Corrente no Mestre

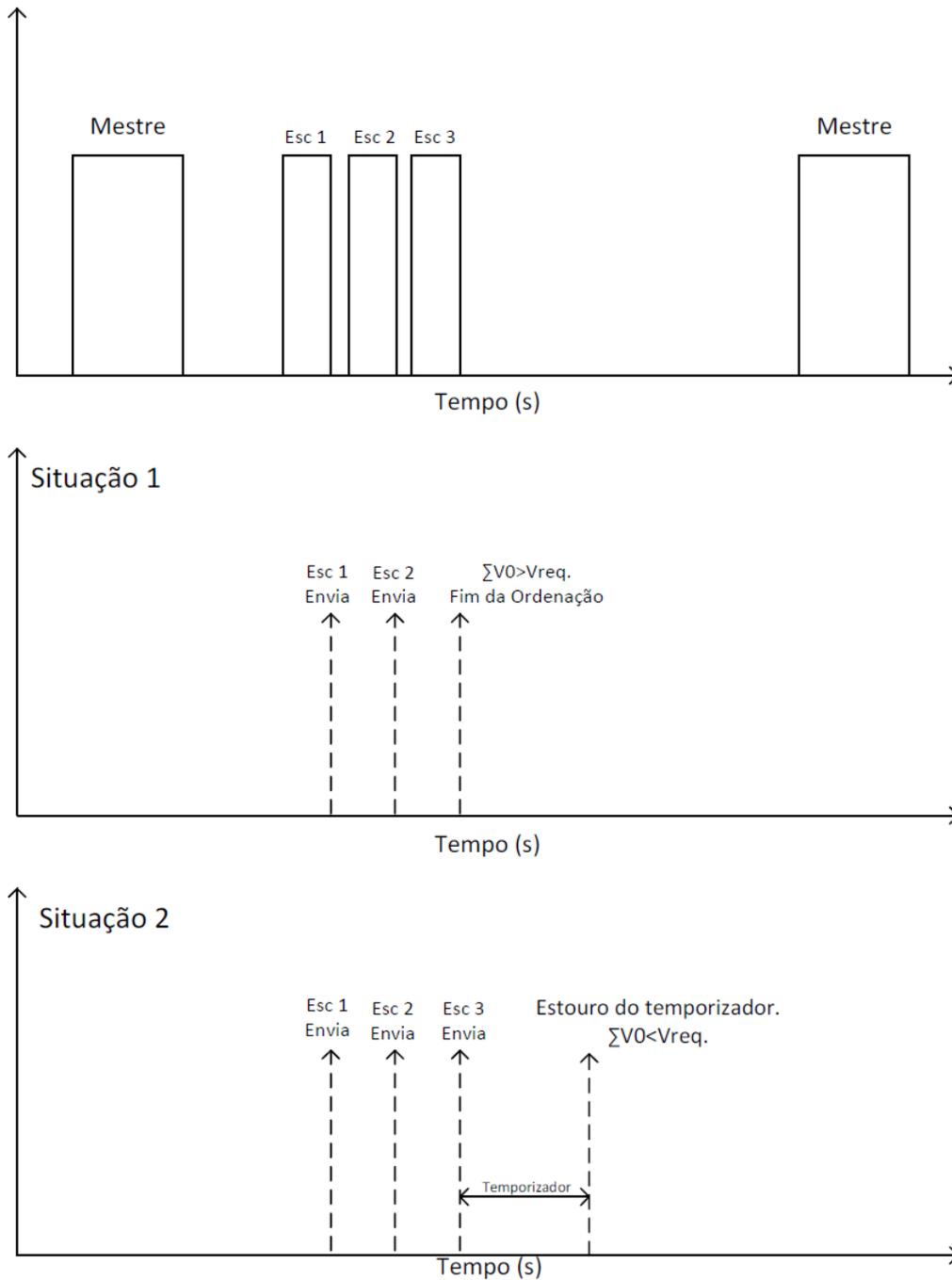
A proposta do simulador toma como base as equações definidas na seção 2.2, onde é definido os valores da corrente no filtro e a tensão armazenada no capacitor. O nó mestre é responsável por simular a corrente que passa no filtro, que de toda forma é a corrente circulando por todo o circuito. Na equação 2.12 está definido o valor da corrente  $I_L$ , mas considerando um tempo infinitesimal. Para que seja possível sua utilização no microcontrolador o tempo deve ser finito. É necessário então que se use uma aproximação, é adotado então a definição da derivada aproximada pelo Método de Euler, ou o método de Diferenças Diretas (*Forward Differences*). Como é calculado na equação 3.6.

$$\frac{\delta I_L}{\delta t} = \frac{I_L(n+1) - I_L(n)}{\Delta t} \quad (3.6)$$

Onde  $\Delta t$  é o intervalo de amostragem da corrente  $I_L$ , neste caso adotado como sendo 1 ms.

Adotar o valor futuro como na equação 3.6 não é prático. Então leva-se em consideração um atraso de 1 na amostra ( $n = n - 1$ ). Desta forma, o numerador da equação 3.6 assume o valor  $I_L(n) - I_L(n - 1)$ . Então a equação 2.12 pode ser expressa na

Figura 20 – Fluxo de mensagens e condições de parada



forma da equação 3.7.

$$I_L(n) = I_L(n - 1) + \Delta t \left( \frac{\Sigma V_0 - V_{AC}}{L} \right) \quad (3.7)$$

O valor simulado dessa corrente é utilizado para definir a tensão requerida  $V_{req}$ , que é enviada pelo mestre e que dá início ao algoritmo de ordenação. A implementação da equação no *firmware* é simples, mas para que se possa ser usada necessita de uma estratégia de controle atrelada à mesma, esta estratégia está descrita na seção 2.3 e é implementada na subseção 3.3.3.2.

### 3.3.3.2 Cálculo do Controlador para a Corrente

O primeiro passo é calcular a resposta do erro de corrente ao integrador. O erro  $e$  é a diferença entre a corrente de referência  $I_{Lref}$  e a corrente calculada na equação 3.7. A variável de controle  $U_{c1}$  é desenvolvida na equação 3.8, considerando o intervalo de cálculo  $\Delta t$ .

$$U_{c1}(n) = U_{c1}(n - 1) + K_i \cdot I_{erro} \cdot \Delta t \quad (3.8)$$

$U_{c1}$  é a resposta do sistema ao integrador. A resposta da saída ao ganho proporcional é  $U_{c2}$ , basicamente é a corrente  $I_L$  vezes o ganho proporcional  $K_p$ . A equação 3.9 é usada para mostrar este cálculo.

$$U_{c2} = K_p \cdot I_L \quad (3.9)$$

Como é ilustrado na Figura 15, o valor gerado pelo controlador proporcional é subtraído do integral, para gerar o valor de referência da tensão  $V_{PVref}$ , na forma mostrada na equação 3.10.

$$V_{PVref} = U_{c1} - U_{c2} \quad (3.10)$$

Este valor é enviado via rede CAN, junto da corrente que circula pelo *grid* atualmente  $I_L$ , para os nós conversores. Ao final de todo o algoritmo de ordenação nos nós conversores, o que retorna ao mestre é o somatório de tensão dos conversores  $\Sigma V_0$ , que é aplicado à equação 3.7, reiniciando assim o ciclo de controle.

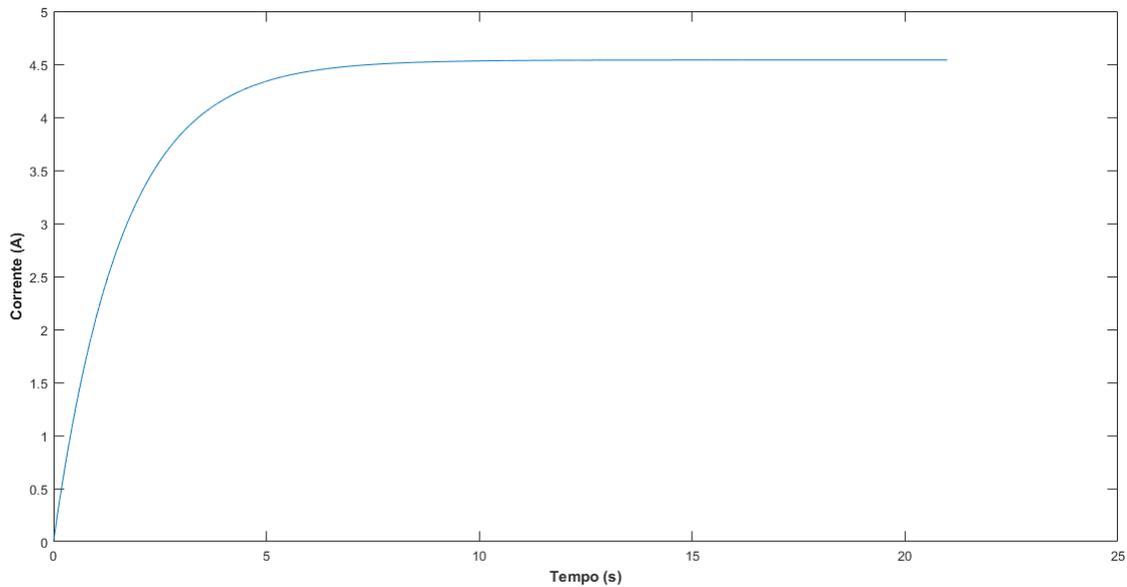
Utilizando as equações 3.7, 3.8, 3.9, 3.10, as equações para o cálculo dos ganhos  $K_p$  e  $K_i$  (2.17 e 2.18, respectivamente) é possível obter a curva da corrente  $I_L$  com o controlador atuando sobre ela para que a mesma atinja seu valor de referência. A Figura 21 apresenta a curva da corrente  $I_L$  indo para o seu valor de referência  $I_{Lref}$ , os ganhos  $K_p$  e  $K_i$  são calculados de forma a se obter uma resposta superamortecida. É utilizado o *software* Matlab para plotagem da curva.

### 3.3.3.3 Simulação da Tensão Armazenada no Capacitor

Nos nós conversores é simulado o valor da tensão armazenada no capacitor ( $E$ ). É esse valor que preenche o campo de dados da mensagem transmitida pelos conversores no barramento CAN. Está definido na equação 2.6, e assim como é o caso da corrente  $I_L$  não pode ser expressa de forma infinitesimal, portanto seguindo seguindo o mesmo desenvolvimento, baseado na aproximação de Euler que foi usado para a corrente  $I_L$ , o valor da tensão armazenada é apresentado na equação 3.11. Onde neste caso é adicionado o valor do ciclo de trabalho  $D$ .

$$E(n) = E(n - 1) + \frac{\Delta t}{C} (I_{PV} + (S_{NP} \cdot (1 - D) - S_{PP} \cdot D) I_0) \quad (3.11)$$

Onde novamente  $\Delta t$  é o tempo de amostragem e  $I_{PV}$  é calculado utilizando-se a equação 2.1. Outra tarefa que o nó escravo deve executar é a comutação das chaves.

Figura 21 – Corrente do *grid*  $I_L$  sob o efeito do controlador

Deve respeitar a tensão que se quer impor na saída do inversor  $V_0$ . Quando o nó ganha a arbitração suas chaves devem comutar de acordo com o fluxo de potência. Sendo ele positivo - tanto com  $V_0$  positivo e  $I_0$  positivo, ou  $V_0$  negativo e  $I_0$  negativo - é necessário que o capacitor descarregue, ou seja  $S_{NP} = 0$  e  $S_{PP} = 1$ . Sendo ele negativo, ou seja os sentidos de  $V_0$   $I_0$  alternados (um positivo e outro negativo), há a necessidade de carga do capacitor, desta forma  $S_{NP} = 1$  e  $S_{PP} = 0$ .

O valor do ciclo de trabalho  $D$  depende de quanto se é necessário ter na saída do conversor, e é calculado somente no último nó a conduzir no ciclo. Caso o nó que ganhe a arbitração não seja o último, o mesmo deve entregar toda sua potência ao barramento ( $D = 1$ ). Quando o conversor é o último a conduzir, ele entrega ao barramento somente o que é necessário para que  $\Sigma V_0 = V_{PVref}$ , dessa forma o nó deve entrar em estado de PWM. O cálculo do ciclo de trabalho então deve ser realizado considerando o valor da tensão armazenada  $E$  e o quanto falta para que  $\Sigma V_0$  seja igual a  $V_{PVref}$ . A equação 3.12 é usada para se encontrar o valor do ciclo de trabalho  $D$ .

$$D = \frac{\Sigma V_0 - V_{req}}{E} \quad (3.12)$$

D então está dentro da faixa de 0 a 1 de acordo com a necessidade de tensão. Com isso é descrito a simulação do sistema tanto pelo nó mestre, quanto pelo nó conversor. Desta forma pode-se montar um fluxograma com os passos a serem seguidos para a implementação do *firmware* nos dois tipos de nó.

### 3.3.4 Fluxograma

A subseção 3.3.4 é usada para apresentar o fluxograma dos nós conversores e do nó mestre. A ideia é apresentar uma visão melhor da execução do *firmware* apresentado.

### 3.3.4.1 Fluxograma do Nó Conversor

O fluxograma do conversor começa com a configuração do periférico da CAN e de dois temporizadores. Um temporizador é utilizado para o cálculo da tensão armazenada no capacitor  $E$ , enquanto o outro é responsável por sinalizar o final do algoritmo de ordenação. Quando é detectada uma mensagem do mestre, o conversor é informado quanto de tensão é necessário ser fornecido  $V_{req}$ , além de ser habilitado a transmissão da mensagem do nó conversor. Então o nó calcula seu identificador, dependendo do sentido da corrente  $I_L$  a partir das equações 3.5 ou 3.4. Em seguida carrega o campo de dados do seu objeto de transmissão com o valor da sua tensão acumulada  $E$ , associado ao identificador calculado anteriormente. O nó então envia a mensagem pelo barramento e monitora se ganhou a arbitragem, caso tenha o menor identificador e por consequência ganhe, o nó aguarda o temporizador indicar o fim do algoritmo de ordenação, comutando sua chave para o modo de carga ou descarga (de acordo com o fluxo de potência) e atualizando seu ciclo de trabalho para  $D = 1$ , aguardando então uma nova mensagem vinda do mestre.

Se o nó perdeu a arbitragem, este recebe a mensagem enviada pelo nó que ganhou e atualiza o valor de  $\Sigma V_0$ . Se  $\Sigma V_0 < V_{req}$ , há a necessidade de uma nova mensagem, dessa forma o nó aguarda o barramento ficar ocioso e tenta reenviar sua mensagem. Para a situação em que  $\Sigma V_0 = V_{req}$ , não há a necessidade do envio de uma outra mensagem, então o nó conversor não tenta um reenvio. Ao fim do algoritmo de ordenação este atualiza seu ciclo de trabalho para  $D = 1$ , força os estados das chaves para  $S_{np} = 0$ ,  $S_{pp} = 0$  e aguarda uma nova mensagem do mestre.

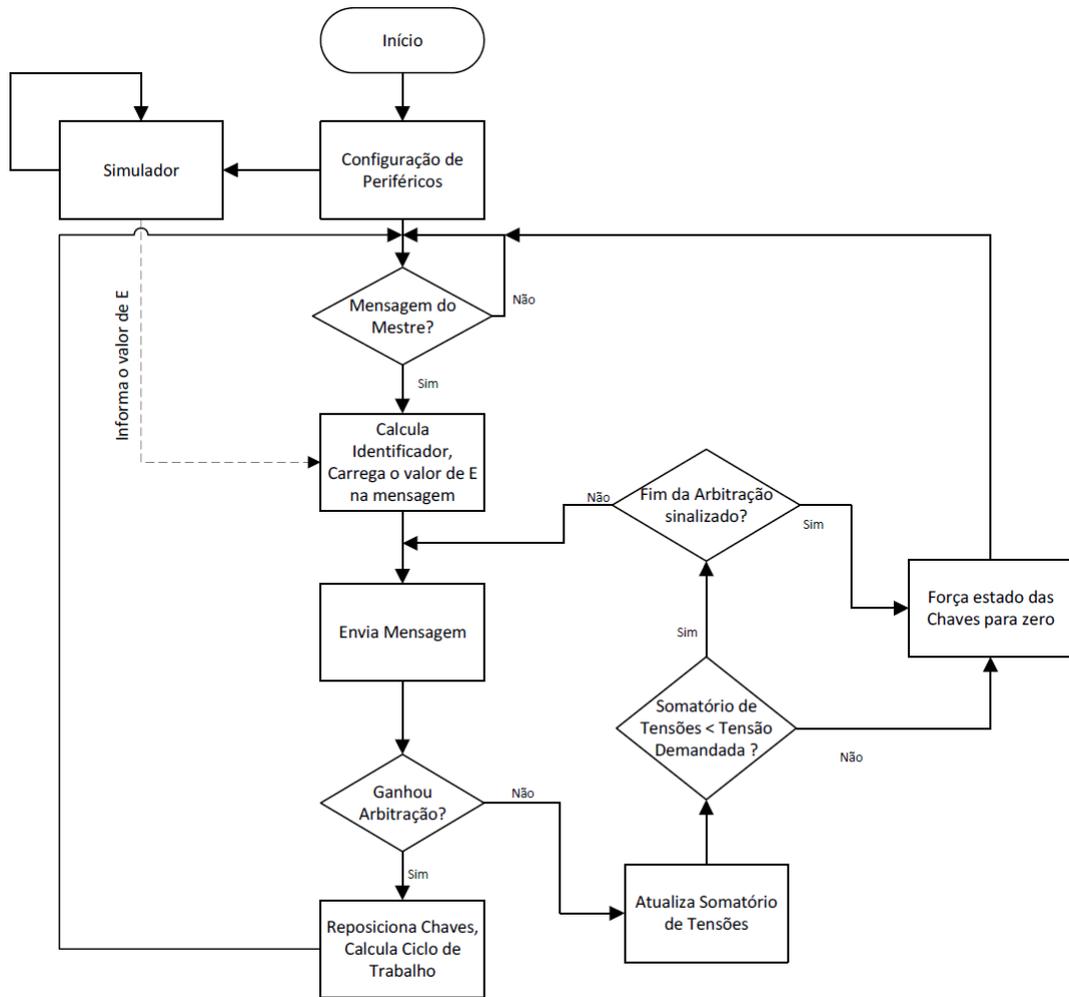
No caso específico de ser detectado que o nó está enviando a última mensagem (se  $\Sigma V_0 + E \geq V_{req}$ ), após o nó ganhar a arbitragem este deve atualizar o valor de  $D$  de acordo com a equação 3.12 quando for sinalizado o fim da arbitragem.

Ocorrendo de todos enviarem suas mensagens mas o somatório de tensão não corresponder ao que foi requisitado pelo mestre, ao estouro do temporizador que determina o fim da arbitragem, todos os nós tem seu estado forçado ao de não conduzir neste ciclo. O intervalo para que seja afirmado que não há mais mensagens do barramento corresponde ao tempo de envio do campo de intermissão da mensagem, com mais alguns bits de tolerância.

### 3.3.4.2 Fluxograma do Mestre

Os fluxogramas dos *firmwares* implementados nos nós mestre e conversor são ilustrados nas Figuras 23 e 22. O fluxograma do mestre inicia com a configuração dos periféricos, onde se destaca a habilitação do periférico da CAN e dois *Timers*, sendo um para o cálculo da simulação da corrente dentro do microcontrolador e o outro que informa quando a mensagem que carrega o valor de  $V_{req}$  e  $I_L$  deve ser enviada.

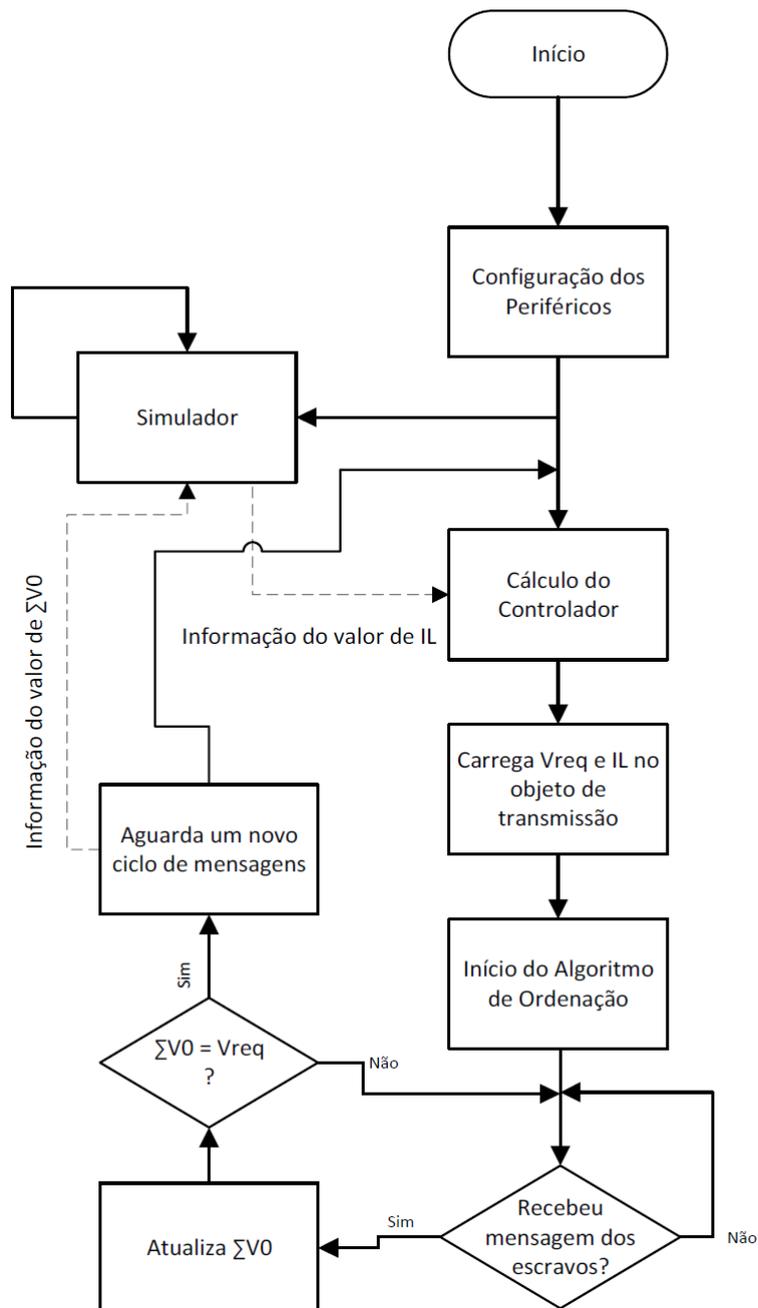
Figura 22 – Fluxograma do conversor



O controle da corrente está sob o mesmo *timer* que envia a mensagem via CAN, pois é este controle que fornece a referência de tensão para os nós conversores  $V_{req}$ . Após o envio da mensagem pelo mestre, os nós escravos iniciam seu envio. A cada mensagem recebida o mestre atualiza o valor do somatório das tensões disponíveis nos nós conversores  $\Sigma V_0$ , até que a confirmação do recebimento da última mensagem, quando  $\Sigma V_0 \geq V_{req}$ .

Recebida a última mensagem, atualiza-se o valor de  $\Sigma V_0$  usado para o cálculo da corrente no *grid* na equação 3.7. Na hipótese da condição não ser cumprida, o valor de  $\Sigma V_0$  não é atualizado. A corrente é recalculada, realimentada à malha de controle para que possa ser gerado uma nova referência  $V_{req}$  para os nós conversores, que é novamente enviada pelo mestre via barramento CAN, dando início a um novo ciclo de mensagens assim que o *Timer* estourar novamente.

Figura 23 – Fluxograma do mestre



## 4 Resultados Obtidos

Este capítulo é utilizado para avaliar e validar todos os pontos propostos para o funcionamento do sistema, como a validação da ordenação via *hardware*, o controle de corrente no mestre e o funcionamento da rede CAN. Para isso foi criada uma rede CAN com quatro nós, sendo cada nó uma plataforma de desenvolvimento Tiva TM4C123GH6PM. São três nós conversores e um nó mestre, ou dois conversores como em alguns testes. A rede foi montada como descrito na proposta: taxa de transferência de 1 Mbit/s, os nós escravos com um objeto de transmissão e dois de recepção, um correspondendo à mensagem que vem do mestre e o outro às transmissões dos escravos. O mestre possui um objeto de transmissão e outro de recepção.

### 4.1 Funcionamento da Rede CAN

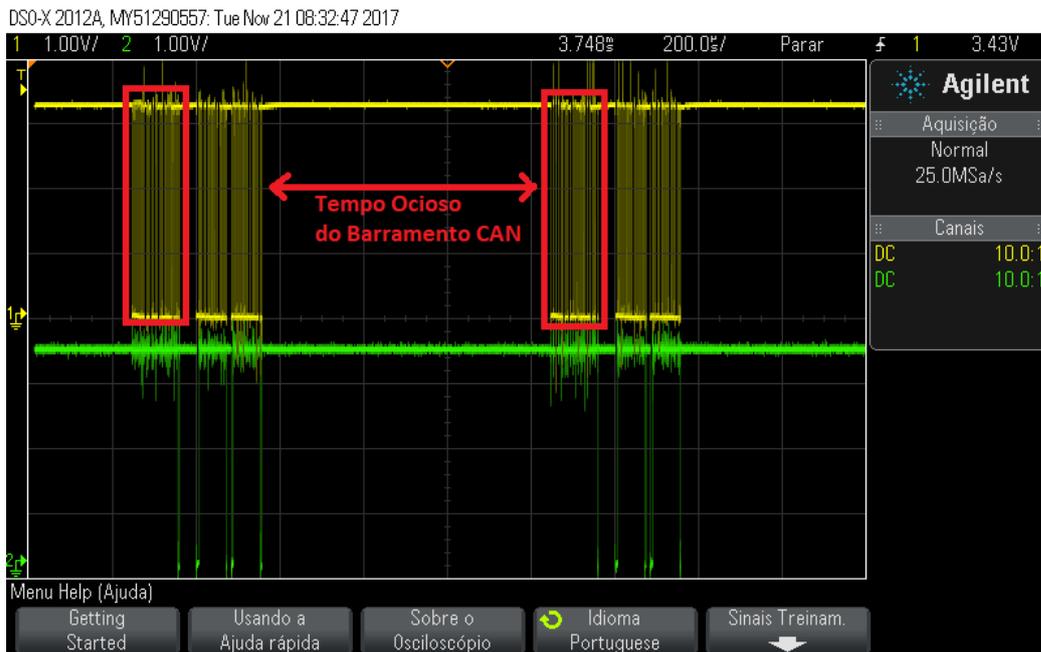
A rede CAN é considerada como operando normalmente quando pelo menos dois nós conectados a ela conseguem transmitir e receber mensagens sem erros provenientes do barramento. Para o levantamento dos resultados quanto ao funcionamento da rede, foi montado uma rede CAN com quatro nós, que dentro da dinâmica deste trabalho são três nós escravos e um mestre. O teste consiste em o mestre enviar sua mensagem, e os três conversores, cada um com um identificador de valor arbitrário diferente enviarem uma resposta. A mensagem do mestre contém a tensão de referência  $V_{req}$  que o mesmo deveria informar para que sua estratégia de controle ocorresse da forma correta, e os escravos enviam no campo de dados das suas mensagens 0.53 vezes o valor informado pelo mestre (para evitar problemas com aproximações). Consequentemente, seriam necessárias duas mensagens por parte dos escravos para suprir a necessidade do nó mestre.

O ciclo de mensagens é reiniciado à cada 1 ms, tempo suficiente para que haja toda a troca de mensagens e processamento das informações. A Figura 24 mostra dois ciclos de mensagem, sendo perceptível o excedente de tempo ocioso entre os dois.

As marcações em vermelho evidenciam as duas mensagens do mestre, que são os indicadores de um novo ciclo.

Devido à implementação utilizada, os nós conversores enviam suas mensagens em sincronia, então haverá disputa de arbitração. São realizadas duas disputas, o nó que perder em ambas é forçado a limpar sua mensagem do barramento. Após seu envio, o nó entra em estado ocioso no aguardo da próxima mensagem do mestre, o mesmo ocorre com o nó que perde ambas as arbitrações. Portanto, para haver o funcionamento do mestre como proposto é necessário que ocorra a seguinte sequência de mensagens:

Figura 24 – Representação de dois ciclos de mensagem



- O mestre envia sua mensagem;
- Há a primeira arbitração, o nó com o menor identificador ganha a disputa, envia sua mensagem e entra em modo ocioso;
- Ocorre então a segunda arbitração, dos dois nós disputando aquele com o menor identificador vence a arbitração, envia sua mensagem e entra em modo ocioso;
- O terceiro nó não envia mensagem nenhuma, e os três nós aguardam o reinício do ciclo com uma nova mensagem do mestre.

As Figuras 25, 26 e 27 são usadas para mostrar respectivamente o primeiro nó escravo que envia sua mensagem em comparação com o barramento, o segundo nó ganhando a arbitração e enviando sua mensagem e o terceiro nó perdendo todas as mensagens e não enviando nada.

Figura 25 – Primeira Mensagem Comparada ao Barramento CAN

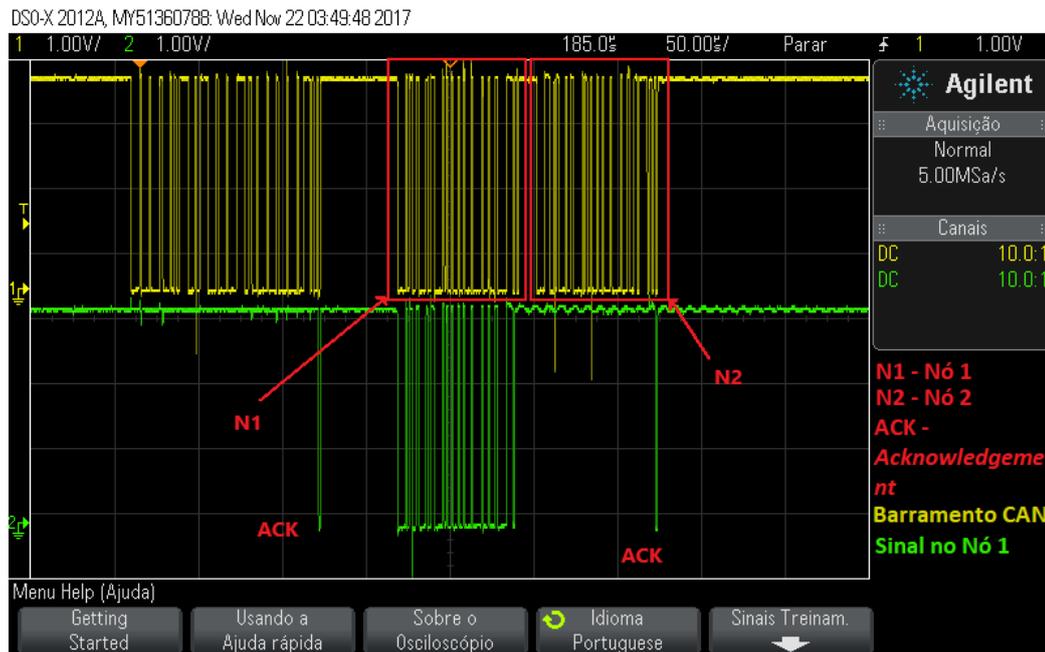
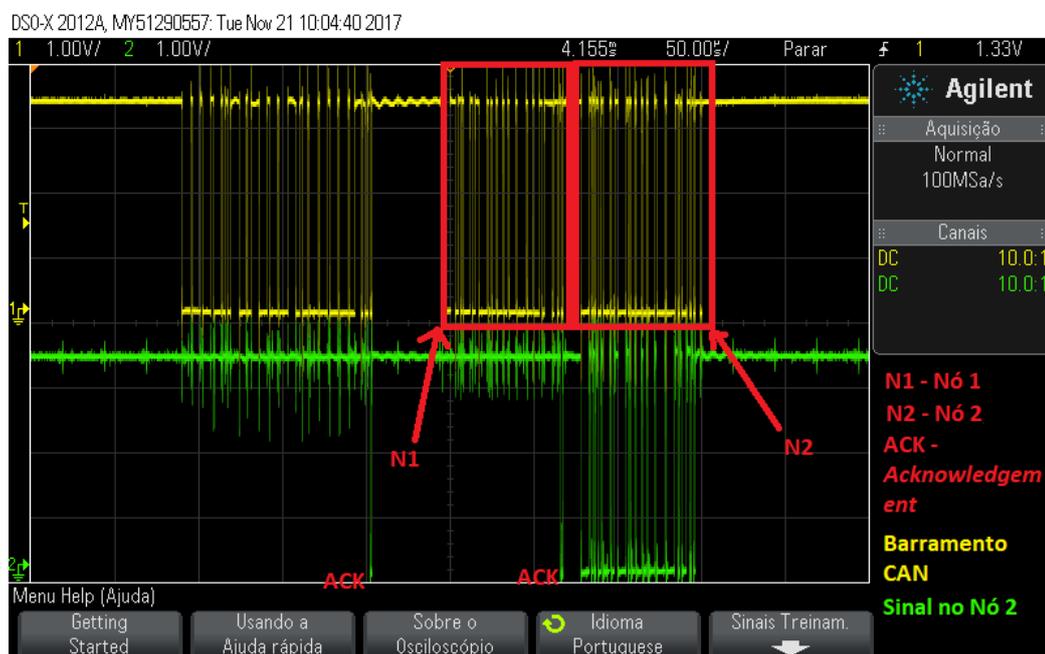
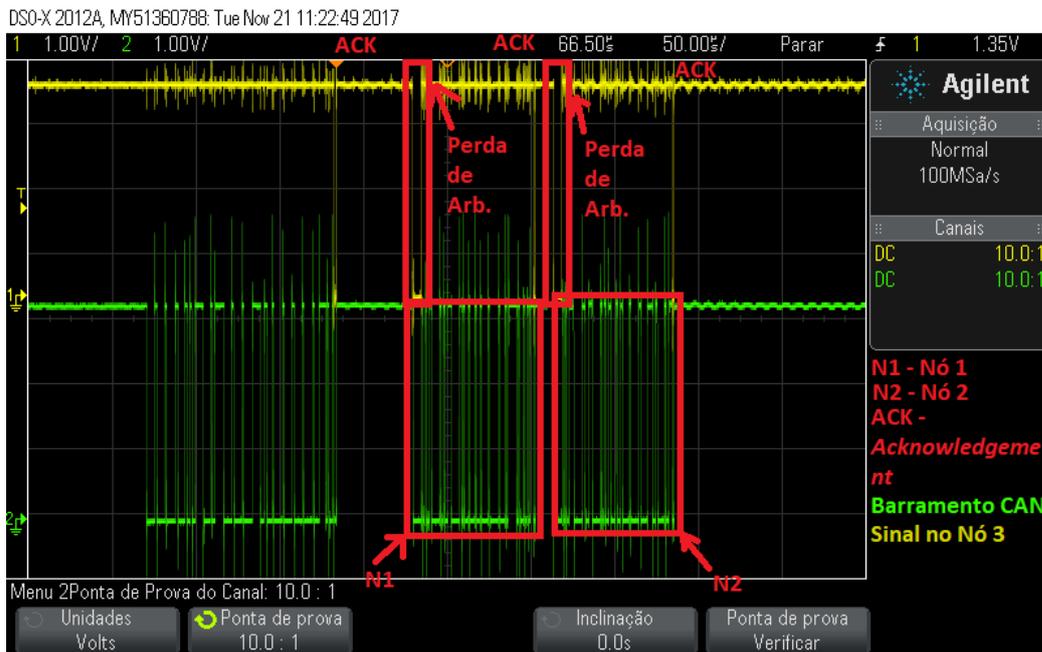


Figura 26 – Segunda Mensagem Comparada ao Barramento CAN



É possível notar pelas Figuras 25, 26 e 27 que a rede CAN funciona como proposta, o barramento monitorado na Figura 25 (em amarelo) é o mesmo para as Figuras 26 (em amarelo) e 27 (em verde). Ainda pontuando o funcionamento do nó que perde todas as arbitrações, ele está representado na Figura 27, as marcações em vermelho indicam quando o nó perde a arbitração, nota-se que ele tenta imprimir sua mensagem, mas perde pois seu identificador é o maior entre os três.

Figura 27 – Terceira Mensagem Comparada ao Barramento CAN



## 4.2 Validação da Estratégia de Controle

Para se validar a estratégia de controle é necessário mostrar que utilizando as equações 3.7, 3.8, 3.9 e 3.10 presentes na subseção 3.3.2 é possível chegar no valor da corrente de referência  $I_{Lref}$  desejada. Partindo do pressuposto de que estas equações controlam a saída da corrente com relação à uma referência, é montado um sistema com quatro nós CAN, sendo um nó mestre e três escravos. De forma que os nós conversores (escravos) recebem o valor de referência do mestre e carregam sua mensagem com uma fração deste mesmo valor no campo de dados, com um identificador como calculado na equação 3.4. Por o mestre aguardar a resposta dos nós conversores há um certo atraso entre o envio do valor de referência e seu recebimento, para atenuar quaisquer erros provenientes de atraso foi respeitado o mesmo período de 1 ms tanto para envio da mensagem quanto para cálculo do simulador de corrente. Além do valor de  $\Delta t = 1$  ms, fez-se uso das seguintes constantes:

- $V_{AC} = 220$  V, pois é considerado a média da tensão de uma rede monofásica (embora seja considerado um barramento DC);
- $L = 5$  mH;
- $K_p = 3,016$ , este valor é obtido através da equação 2.17, considerando  $\zeta = 0,8$  e  $\omega_n = 377$  rad/s;
- $K_i = 1,885$ , este valor é obtido através da equação 2.18, considerando  $\omega_n = 377$  rad/s;

- $I_{Lref} = 4,54$  A, este sendo o valor de referência da corrente para o valor de  $V_{AC}$  e uma potência de 1000 kW, este é o valor inicial para esta variável, mas é alterado ao longo do teste.

O teste consiste em monitorar, com o auxílio do ambiente de programação *Code Composer Studio*, o valor da corrente no *grid*  $I_L$  ao longo do funcionamento do sistema. Para isso o ambiente dispõe de uma ferramenta gráfica que plota os valores de variáveis contidas no *firmware* que está sendo depurado que se atualizam dentro de ciclos regulares definidos pelo usuário. O valor da corrente de referência  $I_{Lref}$  é atualizado para que se possa mostrar o funcionamento do controlador. São utilizadas as Figuras 28, 29, 30, 31, 32, 33 e 34 para mostrar o funcionamento do sistema sob várias correntes de referências diferentes. Usa-se a Figura 28 para mostrar a condição inicial antes do sistema entrar em funcionamento, sendo necessário fazer a relação de que na figura  $I_g$  representa o valor de  $I_L$  e  $I_{g\_ref}$  o de  $I_{Lref}$ . Esta relação é válida para todas as outras figuras. Seguindo, a Figura 29 é utilizada para mostrar o funcionamento para a referência inicial  $I_{Lref} = 4,54$  A, a Figura 30 com uma referência  $I_{Lref} = 8$  A, a Figura 31  $I_{Lref} = 12$  A, a Figura 32  $I_{Lref} = 10$  A, a Figura 33 retorna ao valor inicial para após a Figura 34 forçar a referência para  $I_{Lref} = 6$  A.

Figura 28 – Condições antes do funcionamento do sistema

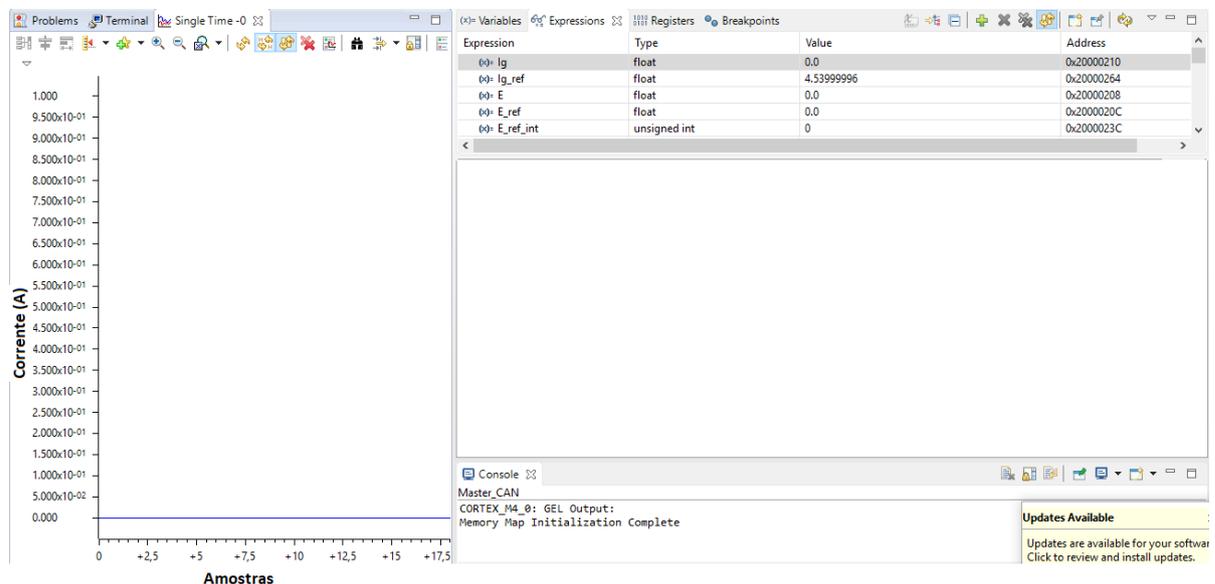


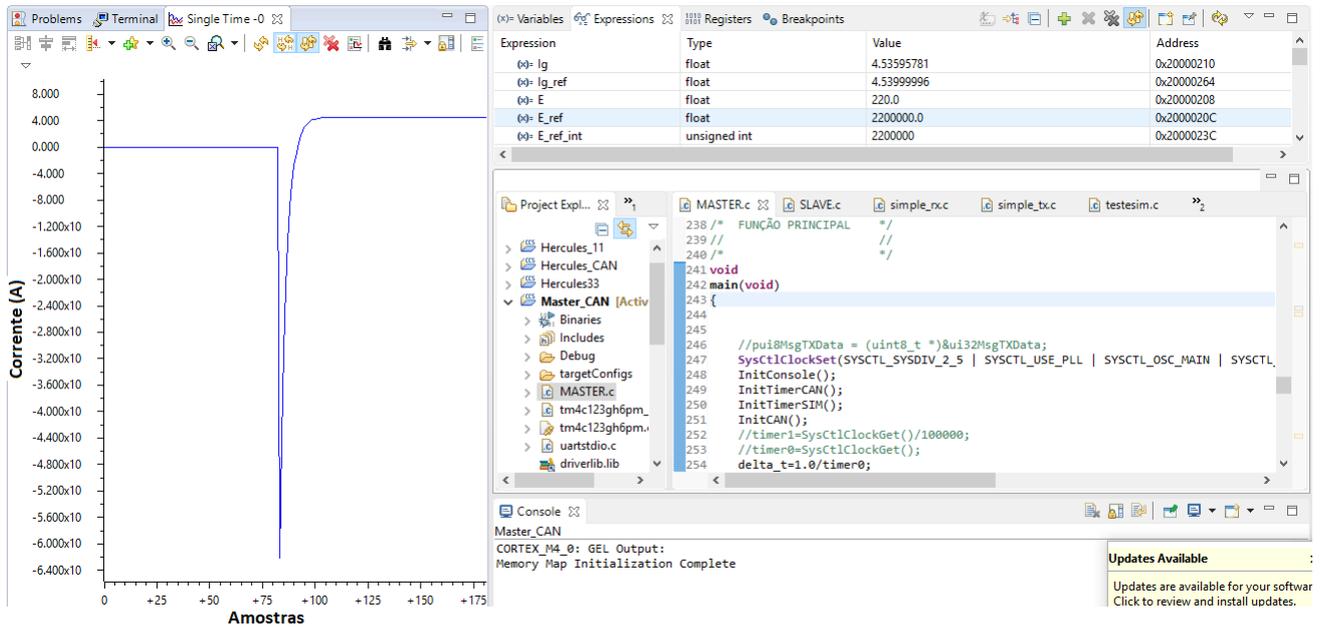
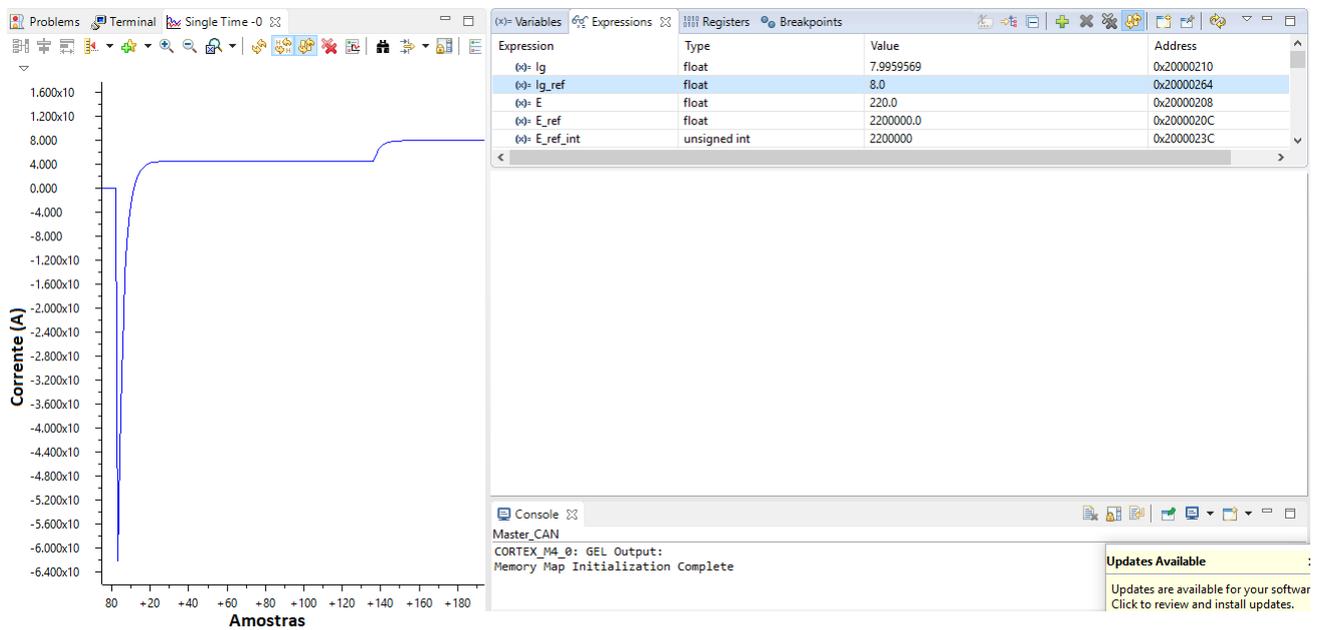
Figura 29 – Referência de corrente inicial  $I_{Lref} = 4.54$  AFigura 30 – Controle do sistema do mestre com referência  $I_{Lref} = 8$  A

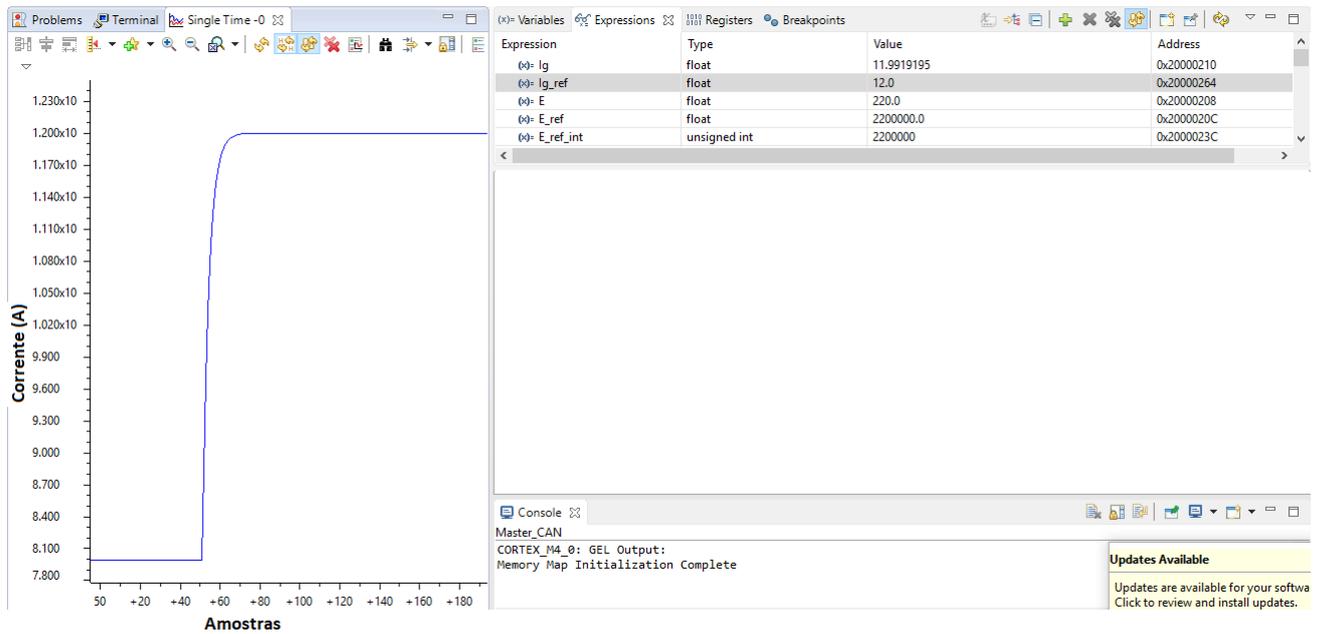
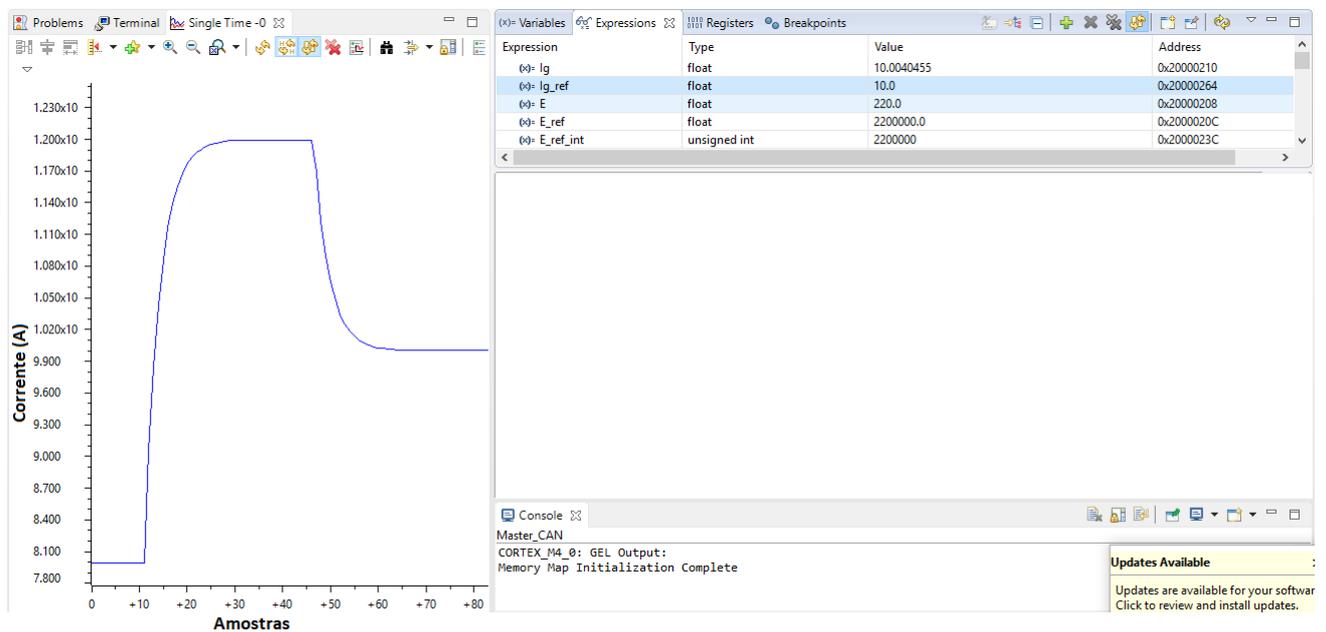
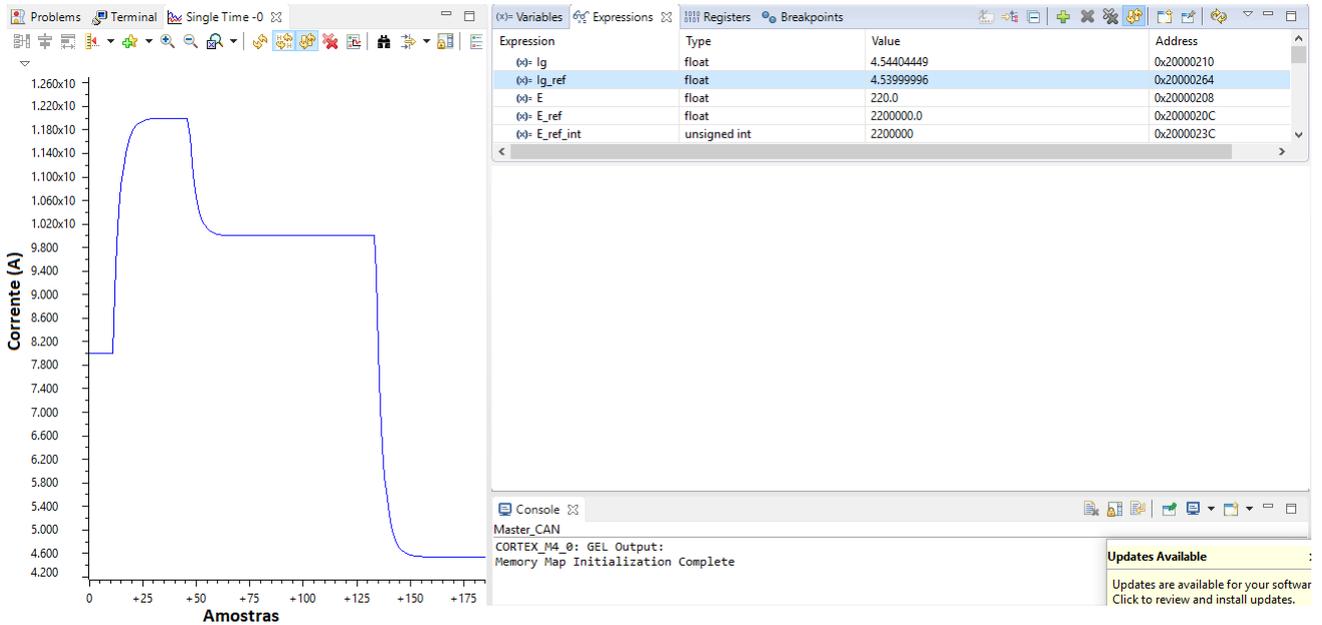
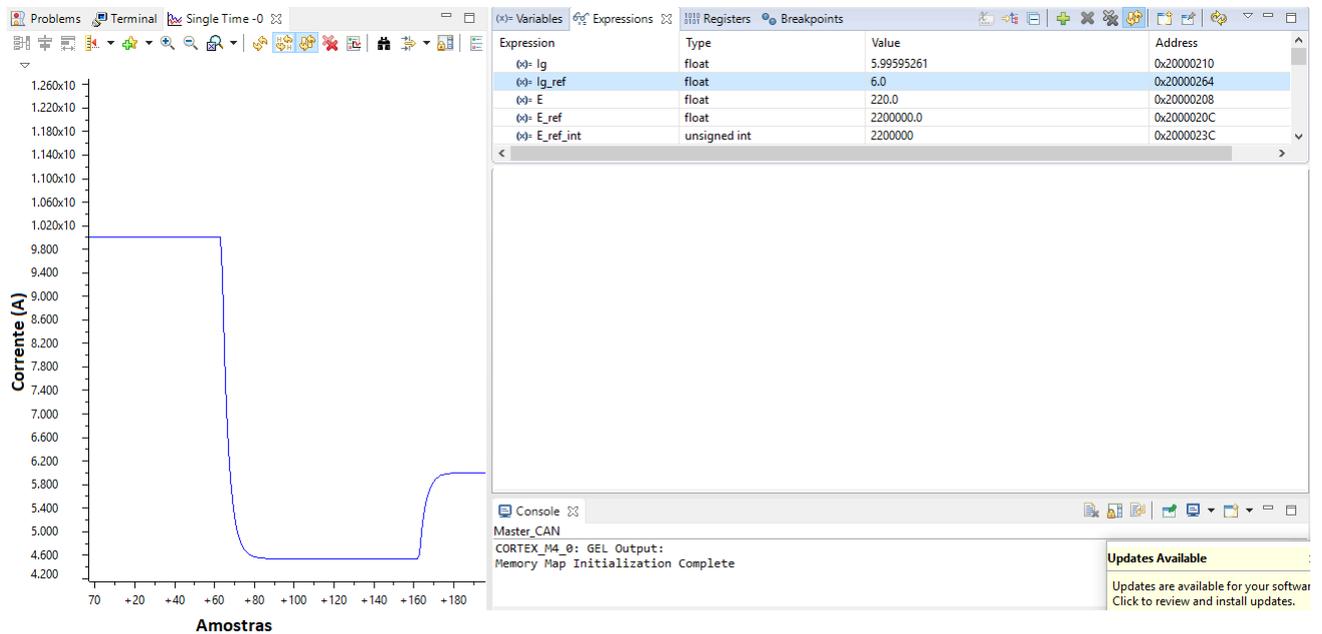
Figura 31 – Controle do sistema do mestre com referência  $I_{Lref} = 12$  AFigura 32 – Controle do sistema do mestre com referência  $I_{Lref} = 10$  A

Figura 33 – Controle do sistema do mestre com referência  $I_{Lref} = 4.54$  AFigura 34 – Controle do sistema do mestre com referência  $I_{Lref} = 6$  A

Dessa forma, a partir dos ganhos  $K_p$  e  $K_i$  calculados, pode-se notar uma resposta superamortecida do sistema. O controle do sistema assim como descrito na subseção 3.3.2 é então validado.

### 4.3 Validação da Algoritmo de Ordenação

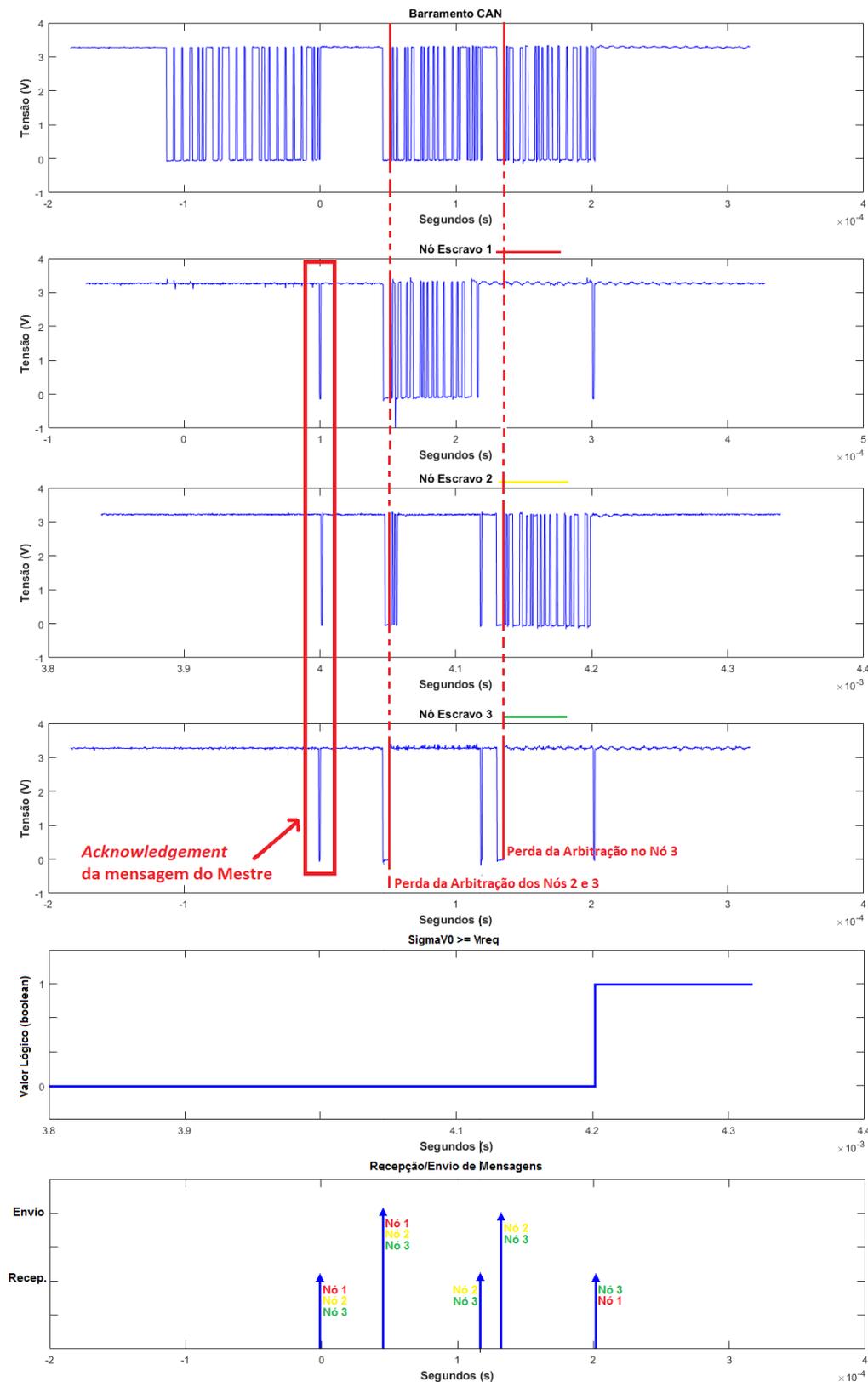
Para atestar o funcionamento do algoritmo de ordenação como proposto é necessário mostrar que as mensagens dos nós conversores são enviados ao mesmo tempo, em sincronia, para que haja a disputa pelo barramento e conseqüentemente processo de arbitração. A arbitração no barramento é quem vai ordenar os nós conversores, pois o identificador das mensagens é baseado no erro entre a tensão armazenada no capacitor  $E$  e sua tensão de ponto ótimo  $V_{PVref}$ . Desta forma, o capacitor mais carregado envia primeiro, seguido pelo segundo capacitor mais carregado e assim por diante.

Outra afirmação que precisa ser provada é que o barramento CAN não pode receber mensagens de nós conversores após a condição  $\Sigma V_0 \geq V_{req}$  for cumprida. Tendo isto posto, os nós conversores no sistema que não conseguirem enviar suas mensagens devem limpá-las do barramento. É repetido então o mesmo sistema mencionado na seção 4.1, mas por motivos de melhor entendimento do processo é empregado o *software* Matlab para geração das curvas que mostram o barramento CAN, e os sinais de transmissão dos três nós conversores do sistema. A Figura 35 é usada para mostrar um ciclo de mensagem pelo barramento, a resposta de cada nó conversor dentro deste ciclo e ilustra o fluxo de mensagens e o momento onde os nós devem parar de enviar dados pelo barramento. Na Figura 35 o Nó 1 representa o menor identificador, o nó 2 aquele com o segundo menor e o nó 3 o que sempre perde o processo de arbitração.

O fluxo de mensagens ocorre como nas etapas descritas:

- a) O mestre envia sua mensagem, e ao final do envio os três nós enviam um bit de *acknowledgement* confirmando seu recebimento, isto está explícito no sinal de todos os conversores e é sinalizado no fluxo de mensagens;
- b) Há então a primeira tentativa de envio, onde há a ocorrência de arbitração, sendo possível notar nos nós 2 e 3 o momento em que perdem a arbitração e o nó 1 envia sua mensagem. No momento posterior ao envio da mensagem completa pelo nó 1, os nós que perderam a arbitração enviam seus bits de *acknowledgement* pelo barramento sinalizando o recebimento correto da mensagem;
- c) Logo após vem a segunda tentativa de controle do barramento CAN, onde os nós escravos 2 e 3 re-enviam suas mensagens e iniciam outro processo de arbitração. Dessa forma o nó 2 vence e envia sua mensagem, o nó escravo 3 tenta enviar mas não consegue por ter maior identificador e após o envio da mensagem pelo nó 2 manda um *acknowledgement* pelo barramento. É possível notar que o nó 1 também recebe esta segunda mensagem, mas pela maneira como foi implementado o algoritmo esta segunda mensagem não faz diferença para ele;

Figura 35 – Fluxo do Algoritmo de Ordenação da CAN



- d) Quando ocorre o recebimento da segunda mensagem é sinalizado a condição de que  $\Sigma V_0 \geq V_{req}$ , por este motivo não há necessidade do envio da terceira mensagem, os três nós entram em modo ocioso e estão no aguardo do reinício do ciclo pelo mestre.

É atestado o funcionamento do algoritmo de ordenação pela ocorrência de ambos os fatos: disputa do barramento e conseqüentemente processo de arbitração, onde este fator é observado nos dois períodos em que ocorre transmissão de mensagens. Desta forma é validado também sincronismo no envio pelos nós conversores. O outro fato que valida o algoritmo de ordenação é o terceiro nó não enviar sua mensagem quando o barramento CAN fica livre após a sinalização da condição de parada do envio de mensagens  $\Sigma V_0 \geq V_{req}$ . Em conclusão, o algoritmo de ordenação, principal objetivo deste trabalho, está validado e funciona como esperado.

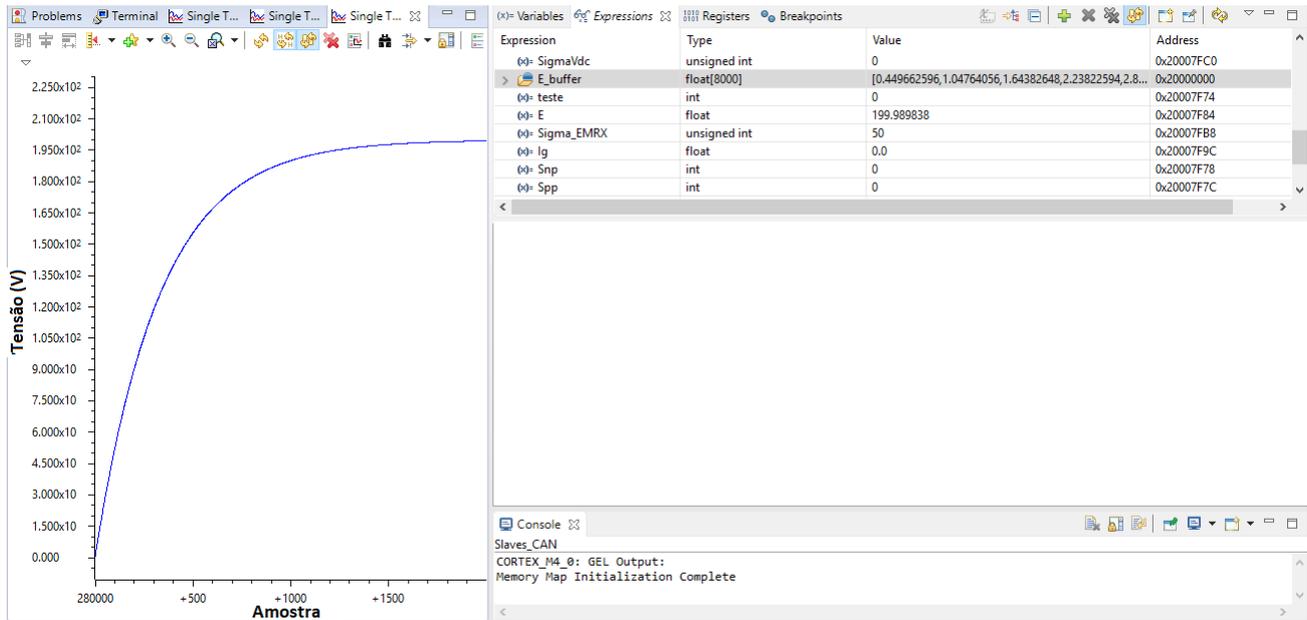
## 4.4 Funcionamento do Sistema Proposto

Analisar e provar o funcionamento do sistema significa que cada componente formador do mesmo está funcionando em conjunto com os outros. Neste caso deve-se atestar que o nó mestre envia sua mensagem para os nós conversores, recebe devidamente suas respostas e atualiza de maneira correta a malha de controle nele implementado. Para os nós conversores é necessário provar que os mesmos recebem a mensagem do mestre, realizam entre si o algoritmo de ordenação com ajuda do barramento CAN, e fazem o cálculo correto da tensão armazenada em seus capacitores, que neste trabalho são simulados dentro do microcontrolador. O funcionamento da rede CAN está validado, assim como o algoritmo de ordenação e a estratégia de controle, faltando agora a utilização do simulador da tensão armazenada no capacitor  $E$  para fornecer os valores da mensagem enviada pelos nós conversores.

Com base no proposto na subseção 3.3.3, a tensão armazenada no capacitor é implementada nos nós conversores. O teste consiste somente monitorar o valor de  $E$  enquanto as equações que simulam seu comportamento são executadas no microcontrolador. Neste teste não é considerado a integração com o conversor e conseqüentemente o *grid*, desta forma, seguindo a equação 3.11 o valor de  $I_0$  é zero fazendo com que o estado das chaves e o valor do ciclo de trabalho  $D$  não importem. No mais os demais valores são escolhidos arbitrariamente,  $V_{sat} = 200$  V e  $R = 66.7$   $\Omega$ , desta maneira  $I_{PV} = 3$  A. A capacitância de  $C$  é de 0.005 F e o passo de cálculo  $\Delta t$  é 25  $\mu s$ . Na Figura 36 mostra-se a curva de carga do capacitor, ela foi adquirida com a ajuda da ferramenta de depuração do *Code Composer Studio*.

Pela Figura 36 é possível notar o formato característico da curva de carga de um capacitor, então é possível atestar o funcionamento da equação que simula seu comportamento. Com este último elemento pode ser realizado o teste de todo o sistema. Inicialmente é montado a rede CAN com um mestre e dois escravos. Os escravos são montados garantindo que ambos consigam suprir a necessidade de tensão do mestre com somente um deles conduzindo, significando também que sempre haverá um nó em estado

Figura 36 – Curva de tensão armazenada no capacitor



de não condução. É monitorado um dos nós escravos, na medida em que o *firmware* é executado a corrente  $I_L$  vai para seu valor de referência - devido ao controle no mestre - e a tensão  $E$  armazenada no capacitor vai mudando de acordo com a necessidade do sistema.

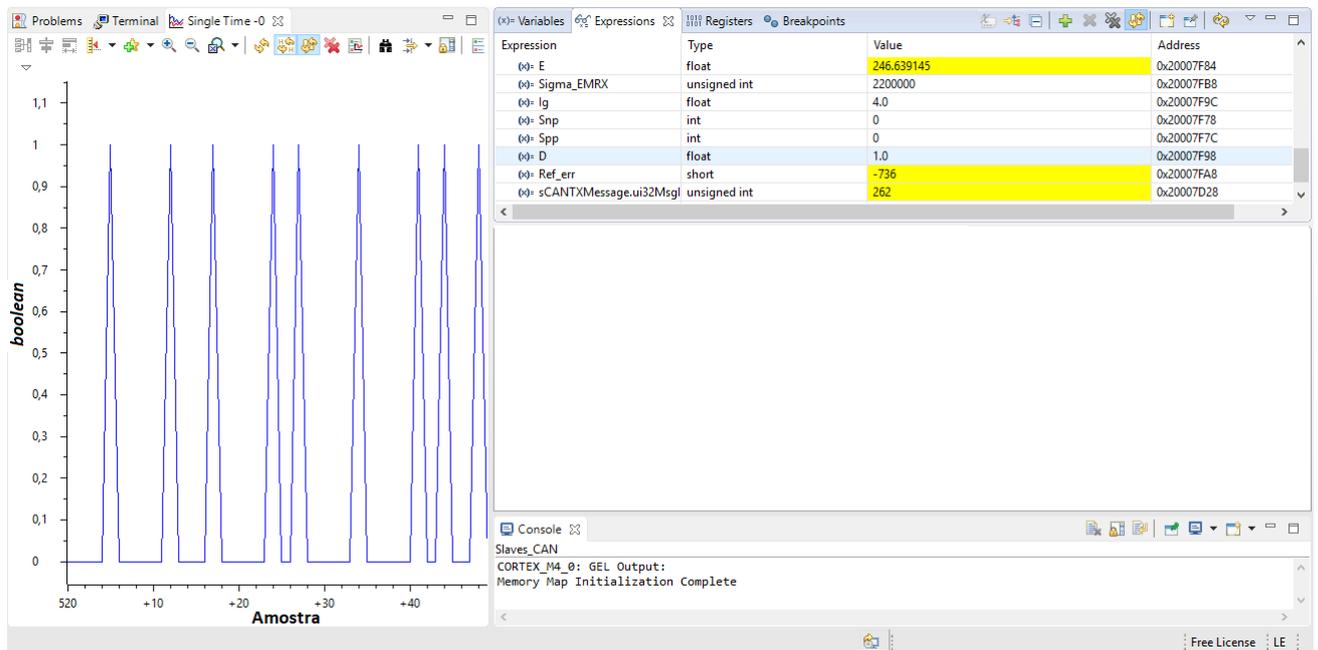
Partindo do ponto que o nó monitorado está com o maior erro (com maior necessidade de descarregar energia), este vai ganhando as arbitrações do barramento, e como é garantido que a condição  $\Sigma V_0 > V_{ref}$  a posição de suas chaves garante que ele descarregue energia no *grid*. Na medida em que a tensão armazenada vai diminuindo, este nó vai apresentando um erro menor com relação ao outro e por consequência uma hora perde o controle do barramento CAN. Nesta hora, suas chaves mudam para o estado de não condução ( $S_{np} = 0$  e  $S_{pp} = 0$ ) e o capacitor começa a carregar novamente. Em contra-partida o outro nó conversor está conduzindo e consequentemente descarregando sua tensão armazenada no capacitor  $E$ , até novamente o nó monitorado estar mais carregado e voltar a ganhar a arbitração, comutando suas chaves para o modo de descarga ( $S_{np} = 0$  e  $S_{pp} = 1$ ).

Seguindo esse raciocínio, é admissível testar a comutação da chave  $S_{pp}$  para validar o funcionamento do sistema. Pelo seguinte motivo: como a suposição para todos os testes realizados é que o sinal da tensão no *grid* não muda (sempre considerado como sendo o mesmo valor e sinal de  $\Sigma V_0$ ) as chaves que fecham são  $S_{pp}$  e  $S_{nn}$ . Na equação 3.11  $S_{pp}$  é quem define o sinal de  $I_0$ , que é a parcela da equação que determina em quais instantes o valor de  $E$  aumenta ou decai. Quando  $S_{pp} = 0$ ,  $I_0 = 0$  e o capacitor tende a aumentar o valor da tensão armazenada. Quando  $S_{pp} = 1$ ,  $I_0 = -I_L$  e o capacitor tende a descarregar

sua tensão armazenada.

Resumindo, quando a chave  $S_{pp}$  conduz, significa que o nó ganhou a arbitração, a condição de que  $\Sigma V_0 \geq V_{req}$  está sendo cumprida, a tensão armazenada no capacitor encontra-se descarregando e o sistema funciona como proposto. Do outro lado, quando  $S_{pp}$  está aberta, o nó perdeu a arbitração e o capacitor está carregando. A Figura 37 é usada para mostrar os estados da chave  $S_{pp}$  durante o funcionamento do sistema. Novamente é utilizado a ferramenta de depuração existente no *Code Composer Studio*.

Figura 37 – Estados da chave  $S_{pp}$





## 5 Conclusão

Neste trabalho foi desenvolvido um sistema de controle e ordenação para conversores multinível ponte H cascata, onde o foco caiu sobre o desenvolvimento do algoritmo de ordenação responsável por decidir o estado de cada conversor individualmente. O conversor é uma peça chave na integração de painéis fotovoltaicos à rede elétrica, como também à cargas diversas.

O algoritmo de ordenação proposto se utiliza do protocolo CAN, de forma que o critério de arbitração existente no barramento era responsável por ordenar os nós que faziam parte do sistema dos conversores de acordo com sua necessidade de liberar ou receber energia. Por este motivo pode-se considerar que a ordenação é feita por *hardware*. Além disso o trabalho também contém uma proposta de implementação de um simulador para alguns parâmetros chave da rede, para que a mesma possa ser simulada e controlada sem a necessidade da construção do sistema físico em si.

A utilização da ordenação é uma proposta relativamente nova, sendo citada por (COPPOLA et al., 2016b) por exemplo. O trabalho aqui apresentado traz como diferencial a proposta de monitoramento distribuído, onde existe um microcontrolador diferente controlando cada conversor, diferente do que ocorre em (COPPOLA et al., 2016b) que faz a ordenação por *software* em somente uma plataforma, que nesse caso é uma FPGA (*Field Programmable Gate Array*). Este controle distribuído só é possível graças a rede CAN. Com relação aos resultados obtidos foi-se possível observar o funcionamento da rede CAN e do algoritmo de ordenação através do critério de arbitração, assim como do funcionamento da rede proposta sob a forma de simulação da mesma.

Como vantagens do sistema proposto pode-se citar o monitoramento e controle distribuído dos nós, podendo cada conversor operar individualmente, sendo possível, dependendo da montagem do sistema, que uma falha crítica em um conversor não afete o funcionamento geral do sistema. Ainda mais, se é detectado um nó defeituoso o próprio tratamento de erro da rede CAN desabilita a comunicação com o mesmo. Outra vantagem de utilizar a rede CAN é possível adicionar novos nós conversores ao sistema sem mudar sua estrutura. Como desvantagem há a limitação no número de conversores que podem compor o sistema, pois muitas mensagens podem sobrecarregar o barramento, ocasionando mal funcionamento. Além disso, muitos nós conectados na rede implica no aumento do tamanho físico do sistema, ocasionando uma redução na taxa de transferência de dados pelo barramento.

## 5.1 Propostas de Trabalhos Futuros

Como proposta de continuidade do trabalho desenvolvido, pretende-se realizar as atividades seguintes:

- a) Melhoria na taxa de transferência da rede utilizando o protocolo CAN FD;
- b) Simulação do circuito de potência através de *softwares* especializados e utilização do sistema desenvolvido neste trabalho para testes da estratégia de controle;
- c) Construção do sistema real, utilizando dispositivos e componentes reais.

# Referências

- AUTOMATION, C. C. in. *History of CAN technology*. 2013. Disponível em: <<https://www.can-cia.org/can-knowledge/can/can-history/>>. Citado na página 21.
- BARBOSA, L. R. G. Rede can. *Escola de Engenharia da UFMG. Universidade Federal de Minas Gerais, Belo Horizonte*, 2003. Citado 3 vezes nas páginas 21, 22 e 23.
- CARISSIMI, A. S.; ROCHOL, J.; GRANVILLE, L. Z. *Redes de Computadores-Volume 20 da Série Livros didáticos informática UFRGS*. [S.l.]: Bookman, 2009. Citado na página 22.
- COELHO, C. A. e. o. *Estudo e Implementação de um Inversor Multinível Monofásico Controlado por FPGA Para Aplicações de Ensino e Pesquisa*. 2016. 1-150 p. Citado na página 30.
- COPPOLA, M. et al. Modulation technique for grid-tied pv multilevel inverter. In: IEEE. *Power Electronics, Electrical Drives, Automation and Motion (SPEEDAM), 2016 International Symposium on*. [S.l.], 2016. p. 923–928. Citado na página 16.
- COPPOLA, M. et al. An fpga-based advanced control strategy of a gridtied pv chb inverter. *IEEE Transactions on Power Electronics*, IEEE, v. 31, n. 1, p. 806–816, 2016. Citado 3 vezes nas páginas 15, 17 e 73.
- FUENTES, C. D. et al. Experimental validation of a single dc bus cascaded h-bridge multilevel inverter for multistring photovoltaic systems. *IEEE Transactions on Industrial Electronics*, IEEE, v. 64, n. 2, p. 930–934, 2017. Nenhuma citação no texto.
- HARTWICH, F. et al. Can with flexible data-rate. In: *Proc. iCC*. [S.l.: s.n.], 2012. p. 1–9. Citado na página 27.
- INSTRUMENTS, T. *SN65HVD23x 3.3-V CAN Bus Transceivers*. [S.l.]: datasheet, 2001. Citado na página 45.
- INSTRUMENTS, T. *Tiva™ C Series TM4C123G Launchpad Evaluation Board-User's Guide*. [S.l.]: Texas Instruments Incorporated, 2013. Citado na página 47.
- INSTRUMENTS, T. *Tiva TM4C123GH6PM microcontroller*. [S.l.]: datasheet, 2013. Citado na página 46.
- KOURO, S. et al. Control of a cascaded h-bridge multilevel converter for grid connection of photovoltaic systems. In: IEEE. *Industrial Electronics, 2009. IECON'09. 35th Annual Conference of IEEE*. [S.l.], 2009. p. 3976–3982. Citado 3 vezes nas páginas 16, 30 e 31.
- NAPOLI, F. D. et al. Effective control strategy for photovoltaic cascaded h-bridge inverters developed on fpga platform. In: IEEE. *Ph. D. Research in Microelectronics and Electronics (PRIME), 2015 11th Conference on*. [S.l.], 2015. p. 49–52. Citado na página 15.
- NISE, N. S.; SILVA, F. R. da. *Engenharia de sistemas de controle*. [S.l.]: LTC, 2002. v. 3. Citado na página 38.

PEREIRA, I. F. B. de F. *Projectar, Simular e Implementar um Inversor Multinível*. Tese (Doutorado) — Universidade do Porto, 2008. Citado na página 30.

TEAM, R. B. G. v2. 0. *Common Public Radio Interface (CPRI)*, p. 1–75, 2004. Citado 4 vezes nas páginas 21, 24, 25 e 26.

TEAM, R. B. G. Can fd. *Common Public Radio Interface (CPRI)*, p. 1–34, 2012. Citado 3 vezes nas páginas 21, 27 e 28.