

**UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
PROJETO DE GRADUAÇÃO**



JOSÉ RICARDO DA SILVA SANTOS

**SIMULAÇÃO HARDWARE-IN-THE-LOOP DE UM
CONVERSOR APLICADO A UMA PLANTA DE GERAÇÃO
EÓLICA**

VITÓRIA – ES
AGOSTO/2014

JOSÉ RICARDO DA SILVA SANTOS

SIMULAÇÃO HARDWARE-IN-THE-LOOP DE UM CONVERTOR APLICADO A UMA PLANTA DE GERAÇÃO EÓLICA

Parte manuscrita do Projeto de Graduação do aluno **José Ricardo da Silva Santos**, apresentado ao Departamento de Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Engenheiro Eletricista.

Orientadora: Profa. MSc. Tiara Rodrigues Smarssaro de Freitas

Coorientador: Prof. Dr. Paulo J. M. Menegáz

VITÓRIA – ES
AGOSTO/2014

JOSÉ RICARDO DA SILVA SANTOS

SIMULAÇÃO HARDWARE-IN-THE-LOOP DE UM CONVERTOR APLICADO A UMA PLANTA DE GERAÇÃO EÓLICA

Parte manuscrita do Projeto de Graduação do aluno **José Ricardo da Silva Santos**, apresentado ao Departamento de Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Engenheiro Eletricista.

Aprovada em 08, de agosto de 2014.

COMISSÃO EXAMINADORA:

Profa. MSc. Tiara Rodrigues Smarssaro de Freitas
Universidade Federal do Espírito Santo
Orientadora

Prof. Dr. Paulo J. M. Menegáz
Universidade Federal do Espírito Santo
Coorientador

Prof. Dr. Lucas Frizera Encarnação
Universidade Federal do Espírito Santo
Examinador

Eng. Odair de Barros Júnior
Universidade Federal do Espírito Santo
Examinador

À família da Silva Santos.

AGRADECIMENTO

Agradeço primeiramente a Deus pela graça de poder escrever estas linhas para agradecer esta conquista tão sonhada por mim e por toda a minha família. Agradeço profundamente também a Rosa da Silva Santos, José Luiz da Silva Santos e José Rogério da Silva Santos, que são minha família, meu porto seguro e minha maior força nos momentos de provação pelos quais passei. Agradeço também aos meus grandes amigos pela lealdade e pela amizade sem interesse, com a qual é possível acreditar que podemos sim mudar o mundo. Agradeço também aos professores com os quais tive o prazer de conviver ao longo destes anos e as suas contribuições para minha formação sólida e plena como Engenheiro Eletricista e como pessoa.

RESUMO

Este projeto trata do estudo e simulação *hardware-in-the-loop* de um processo de geração eólica. Devido ao avanço deste tipo de geração de energia elétrica, seja pela demanda crescente das cargas instaladas no Sistema Interligado Nacional ou pela opção por um processo que agrida menos o meio ambiente, os estudos voltados a este tema são de grande relevância para indústrias e também para o meio acadêmico. O projeto trata do aprimoramento da simulação de um sistema de geração eólica desenvolvido no ambiente MATLAB/SIMULINK com o intuito de realizar o controle de tensão de um conversor CA-CC. A simulação foi adaptada de um modelo puramente computacional para uma nova técnica onde um elemento físico é inserido no processo simulado. O dispositivo físico em questão trata-se do sistema microcontrolado Explorer16, que conta com um grande número de periféricos e um microcontrolador com elevado grau de processamento, o dsPIC33FJ256GP710A, e cabe a este sistema estabelecer o fechamento de uma malha de controle implementada.

LISTA DE FIGURAS

Figura 1 - Principais Componentes de um Sistema de Geração Eólica	14
Figura 2 - Principais Funções dos Conversores Eletrônicos	15
Figura 3 - Diagrama de Blocos de um Sistema com Microprocessador	17
Figura 4 - Diagrama Geral de Blocos de um Microcontrolador	18
Figura 5 - Esquemático de um Sistema de Simulação <i>Hardware-In-the-Loop</i>	20
Figura 6 - Diagrama de Blocos de uma Malha de Controle Para Simulação <i>Hardware-In-the-Loop</i>	21
Figura 7 - Diagrama de Forças e Elementos de Posicionamento de uma Aeronave	22
Figura 8 - Esquema da HILS de um Piloto Automático	23
Figura 9 - Evolução dos Microcontroladores da Família PIC	24
Figura 10 - Pinagem do dsPIC33FJ256GP710A	25
Figura 11 - A Explorer16	26
Figura 12 - A Programadora ICD2	27
Figura 13 - Estrutura da Transmissão Serial	29
Figura 14 - Sistema de Geração Eólica Simulado	30
Figura 15 - Bloco de Geração	31
Figura 16 - Estrutura Interna do Bloco de Simulação da Geração	31
Figura 17 - O Conversor CA-CC	32
Figura 18 - Estrutura Interna do Conversor CA-CC	32
Figura 19 - Bloco do Controlador no Sistema Simulado	33
Figura 20 - Estrutura Interna do Controlador Simulado	33
Figura 21 - Bloco do PWM	35
Figura 22 - Estrutura Interna do Bloco do PWM.....	35
Figura 23 - Janela de Configuração do Bloco <i>Triangle</i>	36
Figura 24 - Tensão na Saída do Conversor e Tensão de Fase Gerada	37
Figura 25 - Razão Cíclica Durante a Simulação.....	37
Figura 26- Sistema Para Simulação HIL	39
Figura 27 - O Bloco de Adequação de Sinal	40
Figura 28 - Estrutura Interna do Bloco de Adequação de Sinal	40
Figura 29 - O Bloco do dsPIC33FJ256GP710A.....	42
Figura 30 - Estrutura Interna do Bloco dsPIC33FJ256GP710A	42
Figura 31 - Configuração do Bloco Serial Configuration	43

Figura 32 - Configuração do Bloco <i>Serial Send</i>	44
Figura 33 - Configuração do Bloco <i>Serial Receive</i>	45
Figura 34 - Criação de Um Novo Projeto Para dsPIC no CCS C Compiler	46
Figura 35 - Janela de Configuração do Microcontrolador	47
Figura 36 - Configuração da Comunicação Serial no Microcontrolador	48
Figura 37 - Configuração da Interrupção	50
Figura 38 - Aproximação de Função Por Trapézio.....	52
Figura 39 - Fluxograma do Anti <i>Windup</i>	55
Figura 40 - Estrutura Interna do Bloco de Geração do PWM	56
Figura 41 - Montagem do sistema Hardware-In-The-Loop	57
Figura 42 - Resposta da Simulação HIL e Tensão de Fase Gerada	58

LISTA DE QUADROS

Quadro 1 - Principais Características do dsPIC33FJ256GP710A.....	25
Quadro 2 - Principais Recursos da Explorer16.....	26
Quadro 3 - Constantes do Controlador PI do Sistema Simulado.....	34
Quadro 4 - Resultados Obtidos da Simulação Puramente Computacional	38
Quadro 5 - Resolução dos Tipos de Variáveis Mais Utilizados	41
Quadro 6 - Constantes do Controlador PI Digital.....	55
Quadro 7 - Resultados Obtidos da Simulação HIL.....	58

SUMÁRIO

1	INTRODUÇÃO.....	11
2	EMBASAMENTO TEÓRICO	14
2.1	Energia Eólica	14
2.2	Conversores Eletrônicos de Potência	15
2.3	Processadores Digitais de Sinais	16
2.3.1	Microprocessador	17
2.3.2	Microcontrolador	18
2.4	<i>Hardware-In-the-Loop Simulation</i>	18
3	O MICROCONTROLADOR DSPIC33FJ256GP710A.....	24
3.1	A Placa Explorer16	26
3.2	O Compilador CCS C Compiler e MPLAB ICD2	27
3.3	A Comunicação Serial RS232	28
4	SIMULAÇÃO PURAMENTE COMPUTACIONAL.....	30
4.1	O Sistema de Geração	30
4.2	O Conversor CA-CC	32
4.3	O Sistema de Controle de Tensão	33
4.4	O Sinal PWM.....	34
4.5	A Simulação do Sistema.....	36
5	A SIMULAÇÃO DO SISTEMA COM <i>HARDWARE-IN-THE-LOOP</i>	39
5.1	O Bloco de Adequação de Sinal	40
5.2	O Controlador Digital no dsPIC33FJ256GP710A	41
5.2.1	A Programação do dsPIC33FJ256GP710A	45
5.2.2	O Código	50
5.2.2.1	Aproximação Integral Pela Transformação de Tustin.....	51
5.2.2.2	A Equação Final do Controlador	54
5.2.2.3	O Sistema Anti <i>Windup</i>	55
5.3	O Sinal PWM.....	56
5.4	Simulação do Sistema Final.....	57
5.5	Discussão dos Resultados Obtidos.....	59
6	CONCLUSÃO GERAL E TRABALHOS FUTUROS	61
6.1	Conclusão Geral.....	61
6.2	Trabalhos Futuros.....	61

BIBLIOGRAFIA.....	62
APÊNDICE A – ROTINA PRINCIPAL DO MICROCONTROLADOR.....	65

1 INTRODUÇÃO

Projetos realizados no campo da Engenharia Elétrica são dotados de riscos físicos e econômicos para quem os desenvolve. Mais especificamente, projetos no campo de geração de energia elétrica envolvem trabalhos com tensões elevadas e componentes que requerem cuidados na sua operação. Particularmente, com a complexidade que os sistemas têm se desenvolvido, os microcontroladores têm sido utilizados com muita frequência como parte do processo. Tais componentes são vulneráveis às falhas das etapas de testes, pondo em risco o alto investimento feito no desenvolvimento do projeto. A fim de minimizar estes riscos e custos, é realizada uma simulação prévia do sistema projetado. Grandes empresas mantêm seus focos em projetar simuladores de sistemas que possam oferecer aos projetistas um grande grau de liberdade para manipular as diferentes etapas do processo, assim como, ter um cenário de operação tão fidedigno quanto o real sistema em operação, e o microcontrolador inserido neste sistema é uma parte essencial a ser simulada e verificada antes de ser colocado em operação.

Especificamente, este projeto trata do estudo e simulação do processo de geração eólica, que consiste no aproveitamento da força dos ventos para que esta seja convertida em energia elétrica. Dentro da matriz energética brasileira, a geração eólica não faz frente à principal forma de geração no país, a hidráulica. Porém com a grande demanda de energia impulsionada pelo crescimento da economia, a implantação de parques de geração eólica ganha força tanto no Brasil como no Mundo. Um grande exemplo deste panorama vem da China, um país que é marcado por um forte crescimento industrial e econômico nos últimos anos e também pela grande poluição em sua capital Pequim. Os Chineses contam hoje com uma carga instalada de geração eólica de 75 gigawatts (GW) e lançaram um plano de impulso às formas renováveis de geração de energia elétrica e com isso pretendem chegar em 2020 a marca de 200 GW na geração eólica (1). Todos os esforços feitos para que se tenham processos industriais limpos vão além da consciência ambiental e ganham contorno cada vez mais econômico. É uma preocupação das grandes marcas terem atreladas ao seu nome um marketing de produção verde, ou seja, a produção de seus bens faz uso de tecnologias que preservam a natureza. Além destas grandes marcas, passou a ser preocupação dos governos nacionais que as emissões de carbono em seus países fossem reduzidas. Porém, trata-se de um

grande desafio diminuir a emissão de poluentes sem comprometer o desenvolvimento econômico.

Visando atender o Protocolo de Kyoto, que é um acordo internacional entre os países desenvolvidos para a diminuição da emissão de gases que geram o efeito estufa, surgiu o Mercado de Crédito de Carbono. Todos os países que são membros signatários deste protocolo devem reduzir suas emissões de gases poluentes ano após ano. Recentemente, os níveis de corte de emissões destes gases foram reforçados para uma diminuição de 25% a 40%, em 2020, dos níveis que haviam sido registrados em 1990 (2). Assim, países que conseguem se manter abaixo das emissões exigidas pelo protocolo vendem a cota de carbono que deixaram de emitir para outras nações que não conseguiram atingir suas metas.

Assim, o mercado desafia as equipes de engenharia e planejamento a propor soluções de formas de produção e desenvolvimento de equipamentos cada vez mais econômicos e ecologicamente limpos. No desenvolvimento de equipamentos é muito importante e comum fazer uso de métodos de simulação para que seja possível verificar o seu comportamento em diferentes cenários de operação. Sem um primeiro modelo virtual do processo ou equipamento desenvolvido, aumentam as chances de ocorrência de falhas quando este sistema for posto em real operação. A utilização de sistemas de simulação minimiza as chances de erros na execução do projeto, poupando assim recursos financeiros e otimizando o tempo gasto para projetar e implantar um processo. A utilização de sistemas microcontrolados para simular o funcionamento de processos é uma ideia bem difundida no meio da pesquisa, uma vez que os microcontroladores apresentam grande versatilidade em seu uso, alto poder de processamento e um baixo custo para aquisição.

Um estudo já foi realizado anteriormente pelo Eng.º Odair de Barros Junior, no qual apresentou a utilização de um sistema microcontrolado para simular a operação de um conversor *Buck-Boost* (3). Porém, este estudo fora feito com o uso de um modelo puramente virtual do Processador Digital de Sinais (DSP) TMS 320F28335 operando em conjunto com uma malha de controle desenvolvida no PSCAD/EMTDC.

Visando contribuir com as pesquisas já existentes sobre o assunto, este projeto trata da implementação de um sistema de geração de energia eólica através de uma técnica de

simulação denominada *hardware-in-the-loop*, que faz uso do sistema microcontrolado Explorer16 para estabelecer o fechamento de uma malha de controle, diferentemente do que é feito hoje, onde toda a simulação do sistema transcorre de uma forma puramente computacional.

Como a finalidade de conseguir os resultados esperados da operação do sistema simulado, foi necessário utilizar os recursos de programação disponibilizados pela Explorer16 para que os parâmetros do sistema eólico simulado fossem tratados pelo controlador nela implementado. Foi necessário também estabelecer uma forma de comunicação entre o sistema eólico simulado no computador por meio do *software* MATLAB/SIMULINK e o microcontrolador. A forma de comunicação escolhida foi a serial e esta será melhor detalhada na seção 3.3. Assim, com o microcontrolador devidamente configurado e com a comunicação entre os dispositivos efetivada, o sistema foi simulado utilizando a técnica *hardware-in-the-loop* utilizando um tempo virtual de simulação, inerente ao *software* MATLAB/SIMULINK, tendo visto que o sistema microcontrolado possui uma elevada capacidade de processamento e suporta esta condição de operação.

Para que seja entendido como transcorreu todo o trabalho executado este relatório conta, primeiramente, no Capítulo 2 com um embasamento teórico que traz as bases teóricas sobre toda a ciência empregada na construção deste projeto. São expostos assuntos relativos à geração eólica, às partes integrantes deste sistema, como os conversores CA-CC e a técnica de simulação empregada. No Capítulo 3 é descrito o sistema microcontrolado Explorer16 que é utilizado no processo de simulação do sistema de geração eólica. No Capítulo 4 é abordado como a técnica de simulação é aplicado no sistema desenvolvido no *software* MATLAB/SIMULINK. Ao longo do Capítulo 5 é explicado cada parte do sistema de geração eólica simulado, e como estes podem afetar as respostas esperadas. Também durante o Capítulo 5 serão apresentados os resultados que foram obtidos empregando a nova técnica de simulação. Por fim, o Capítulo 6 apresentará a conclusão de todo o projeto executado e os trabalhos futuros que poderão ser desenvolvidos a fim de complementar esta aplicação.

2 EMBASAMENTO TEÓRICO

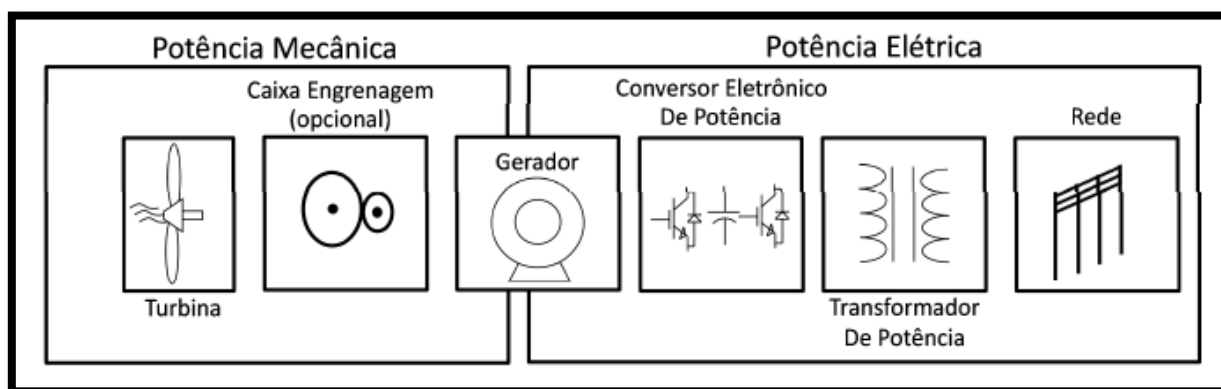
2.1 Energia Eólica

A crescente demanda de energia no planeta aliada à necessidade cada vez maior de tornar-se independente do petróleo intensificou as pesquisas para descobrir e explorar fontes alternativas de energia. As duas principais características procuradas nestas novas fontes de energia é que estas sejam renováveis e não degradem o meio ambiente. Neste contexto está inserida a energia eólica.

Atualmente já é possível ver que mesmo fontes energéticas que são renováveis, caso das fontes hídricas no Brasil, apresentam uma grande dificuldade em suprir a demanda existente (4). Esta dificuldade é devida às restrições ambientais para construção de novas barragens, ao aumento das cargas presentes no sistema elétrico e também aos grandes períodos de seca atravessados pelas bacias que represam as águas para geração de energia nas usinas hidrelétricas. Em março de 2014, as principais usinas hidrelétricas do país já operavam com baixos níveis de armazenamento em seus reservatórios, e ainda estava por vir o período de seca (4). Sendo assim, a geração de energia eólica se apresenta como uma boa forma de complementar o sistema de geração hidrelétrica, uma vez que a maior incidência de ventos ocorre no mesmo período onde há pouca ocorrência de chuva.

De forma sucinta, tem-se um sistema de geração de energia eólica descrito na Figura 1 pelos seus principais componentes.

Figura 1 - Principais Componentes de um Sistema de Geração Eólica



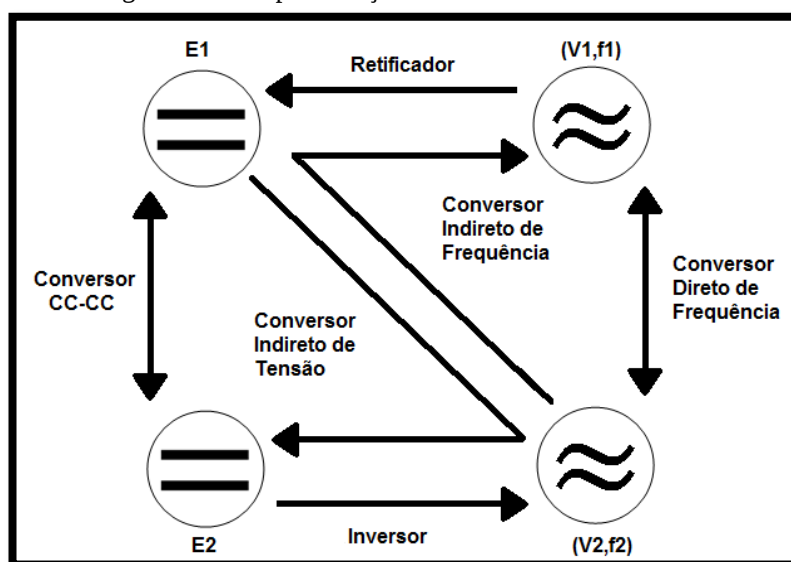
Fonte: Alternativas de Topologias Retificadoras Para Geração Eólica Com Geradores Síncronos A Ímã Permanente de Baixa Potência (5).

O processo de geração tem início com incidência dos ventos na turbina eólica, onde tem-se a geração de uma energia mecânica. Esta energia mecânica, em algumas situações, é transmitida para uma caixa de marchas para adequar a velocidade de mais baixa do giro da turbina à alta velocidade de rotação do gerador. O gerador por sua vez está encarregado de transformar a energia mecânica recebida em seu eixo para uma energia elétrica em seus terminais. A potência elétrica gerada deve ser então adequada para a sua devida finalidade que se deseja, seja atender consumidores diretamente, ser armazenada em um banco de baterias, dentre outras. Cabe então ao Conversor Eletrônico de Potência adequar a energia que é gerada.

2.2 Conversores Eletrônicos de Potência

Os Conversores Eletrônicos de Potência têm a finalidade de promover uma interface entre uma ou mais partes de um sistema elétrico, no qual ele é o responsável por tratar e controlar o fluxo de potência circulante, mantendo-o em seus níveis nominais mesmo diante de variações de operação do gerador. A Figura 2 traz as principais funções dos conversores eletrônicos aplicados na indústria.

Figura 2 - Principais Funções dos Conversores Eletrônicos



Fonte: Eletrônica de Potência (6).

Dentro de todos estes tipos de aplicações possíveis para os conversores estáticos, este estudo trata especificamente de um conversor CA-CC, o qual, pela Figura 2, trata-se de um retificador.

Visando um maior aproveitamento da energia que é gerada e desejando aumentar os níveis de potência dos sistemas de geração, intensificaram-se os estudos sobre a eletrônica de potência, que é a base destes tipos de conversores. Tais estudos têm levado ao desenvolvimento de dispositivos semicondutores cada vez mais modernos, capazes de suportar elevados níveis de tensão associado a uma operação em altas frequências, o que permite que sejam utilizados conversores eletrônicos em sistema associados a potências cada vez mais elevadas.

Sendo assim, estando os conversores eletrônicos presentes em grandes sistemas de geração de energia, é necessário que seja feito junto a ele um sistema de controle. Visando uma operação em conjunto do sistema de controle e conversor, é possível obter sistemas mais robustos e com uma melhor qualidade da energia gerada.

Baseado nesta necessidade da elaboração de um sistema de controle para o conversor, se faz necessária a utilização de um dispositivo capaz de realizar análises das condições de operação do sistema de geração e proporcionar uma alteração sobre ele, visando uma operação adequada. Este novo dispositivo é o Processador Digital de Sinais (DSP, do inglês *Digital Signal Processor*). No DSP são embarcadas lógicas de programação que sintetizam controladores digitais, responsáveis por gerir o comportamento do sistema em um ponto de operação.

Assim, são necessários alguns esclarecimentos a respeito alguns dos diferentes tipos de DSP disponíveis e como estes podem operar em um sistema de controle, mediante suas características.

2.3 Processadores Digitais de Sinais

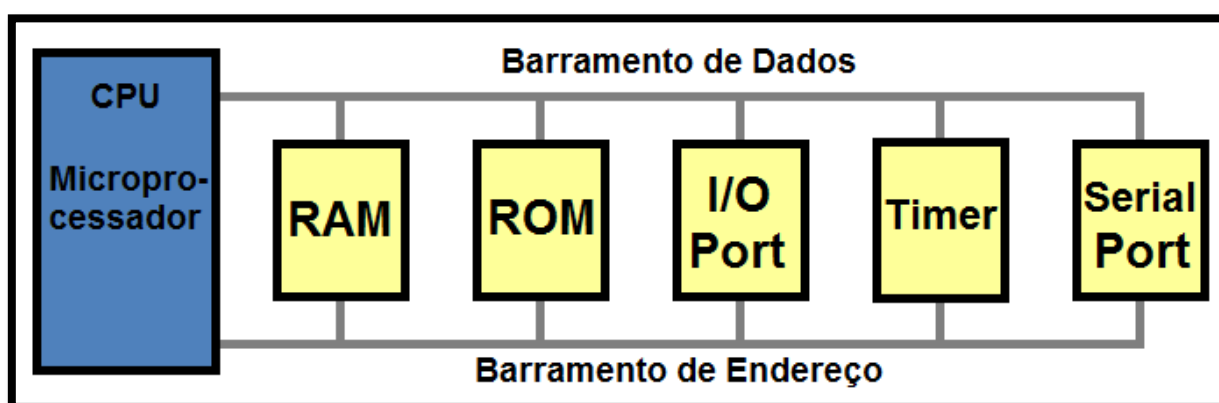
O desenvolvimento tecnológico no ramo da eletrônica não ficou restrito somente ao campo da eletrônica de potência. Com avanços alcançados nas pesquisas de materiais semicondutores, é possível cada vez mais alocar em um mesmo *chip* uma quantidade imensa de transistores

(dezenas de milhares). Isso levou ao desenvolvimento de sistemas que possuem um alto poder de processamento de dados, sendo mais compactos em seu tamanho físico e possuindo um valor agregado mais baixo. Assim nasceram os DSPs, que proporcionam a possibilidade de realizar um tratamento de dados de um sistema com um bom desempenho de processamento aliado a um custo reduzido. Dentre os diversos tipos de DSPs existentes, os mais comuns são compostos por sistemas microprocessados e microcontrolados. Sendo assim, nas seções seguintes serão melhores detalhadas as funcionalidades e características dos microcontroladores e microprocessadores a fim de sanar qualquer dúvida que haja em distinguir estes dois dispositivos. Isso fundamenta a escolha por um sistema microprocessado para compor o sistema de controle desta aplicação.

2.3.1 Microprocessador

Os microprocessadores são voltados para trabalhar com um grande volume de dados e executar cálculos com uma elevadíssima velocidade de processamento. Assim, os microprocessadores apresentam em sua constituição apenas uma *Central Process Unit* (CPU) e todos os outros periféricos necessários para a aplicação devem ser anexados a ele (7). Uma visão genérica da arquitetura de um microprocessador pode ser vista na Figura 3. Nesta figura fica evidenciada a estrutura de uma placa mãe onde um bloco é a CPU e, externamente a ela, são inseridas as memórias RAM e ROM, portas de entrada e saída de dados, dispositivos de contagem, portas de comunicação com outros dispositivos e quaisquer outros elementos que se façam necessários na aplicação.

Figura 3 - Diagrama de Blocos de um Sistema com Microprocessador

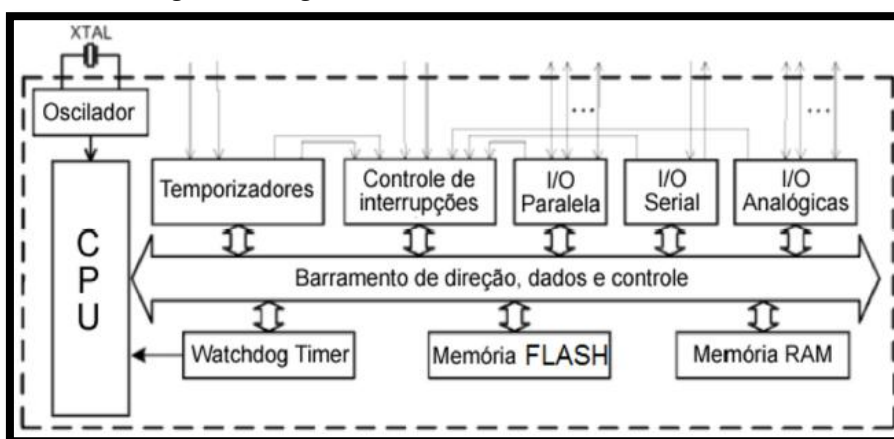


Fonte: Adaptada de Microcontroladores (8).

2.3.2 Microcontrolador

Os microcontroladores são dispositivos nos quais todos os periféricos necessários para o seu correto funcionamento já estão imersos dentro de um mesmo circuito integrado. A utilização dos microcontroladores ganhou força com o grande avanço da tecnologia e com a possibilidade de integrar milhões e até bilhões de transistores em um mesmo chip chegou-se a um barateamento na fabricação de circuitos integrados e também a sistemas com um bom poder de processamento de dados. Assim, passou-se a usar estes circuitos que contam com um menor poder de processamento (quando comparado aos processadores mais modernos e sofisticados) para tarefas mais específicas, que não demandavam tantos recursos como controle de impressora, reguladores de velocidade e acionadores de motores de passos (8). Na Figura 4 pode-se ter uma visão geral dos elementos que compõem um microcontrolador.

Figura 4 - Diagrama Geral de Blocos de um Microcontrolador



Fonte: Microcontroladores (8).

2.4 Hardware-In-the-Loop Simulation

A técnica de simulação *Hardware-In-the-Loop* (*Hardware-in-the-loop Simulation HILS*) é um método já utilizado há aproximadamente 20 anos pela indústria para o desenvolvimento e implantação de projetos (9). Trata-se de uma técnica que visa auxiliar no desenvolvimento e testes de sistemas de controle para processos de complexidade elevada. O primeiro segmento industrial a utilizar a simulação *hardware-in-the-loop* foi o setor de aviação. Hoje, este tipo de simulação está auxiliando no aperfeiçoamento dos sistemas da indústria de motores, no setor de desenvolvimento de equipamentos médico-hospitalar, em sistemas mecânicos e elétricos (com foco em energia renovável) (10), na linha automotiva, dentre outros.

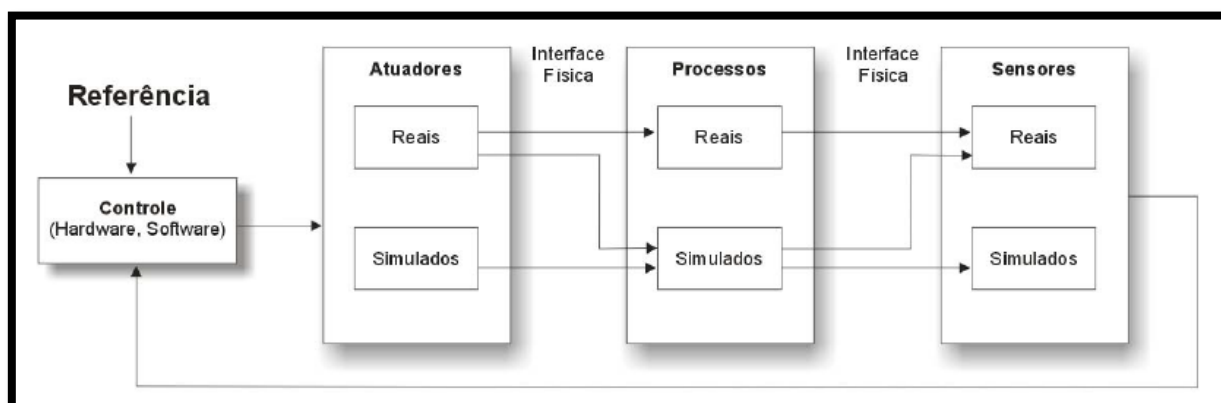
Operar equipamentos sem antes ter uma boa noção de sua condição em operação é um grande risco. Tanto riscos à vida para os especialistas que operam no processo como para os equipamentos nele envolvido. Grandes cifras são envolvidas em processos de grande porte, como os sistemas de geração de energia. Mesmo sendo realizados os devidos cálculos e dimensionamentos das partes do sistema é conveniente que seja possível validar os resultados obtidos com o formalismo matemático. Por isso, sistemas de simulação são tão difundidos no meio industrial.

Há uma grande variedade de simuladores nos quais é possível sintetizar todo um processo industrial. Este processo de síntese é realizado pelo levantamento de modelos matemáticos que descrevem o sistema em questão em uma determinada condição de operação. De fato, estes métodos de simulação no qual são utilizados componentes estritamente virtuais levam o projetista a ter boas aproximações da operação real da planta em desenvolvimento. Porém, sistemas reais são dotados de não-linearidades e características próprias de funcionamento que não são possíveis de ser sintetizadas em um simulador, ou a sua modelagem neste ambiente torna-se inviável tamanha a complexidade inerente a esta característica.

Sendo assim, a fim de chegar-se a um sistema simulado que opere de forma mais próxima de uma operação real de um sistema complexo, é possível adotar uma estratégia de simulação baseada em *hardwares*. O sistema real completo pode ser subdividido em sistemas menores e, por sua vez, estes subsistemas serem substituídos por *hardwares* de simulação mais simples e de menor risco, de forma que a conexão estabelecida entre eles possa resultar em uma operação similar a do sistema real. Porém, elaborar estes hardwares que representem fielmente o funcionamento dos equipamentos reais pode ser tão ou mais complexo que lidar com o próprio equipamento, levando a um aumento considerável no tempo de execução do projeto e gastos desnecessários. Dessa forma, estes equipamentos e sistemas complexos são substituídos por modelos matemáticos e são sintetizados em softwares simuladores.

Uma vez que os sistemas mais complexos de um processo já foram posto em simulação via um sistema computacional, é conveniente que seja elaborado um sistema de controle para que este processo possa ter uma operação mais próxima das condições ideais de funcionamento. Neste cenário que se apresenta a viabilidade da técnica de simulação *hardware-in-the-loop*.

Figura 6 - Diagrama de Blocos de uma Malha de Controle Para Simulação *Hardware-In-the-Loop*



Fonte: Simulação com Hardware-In-the-Loop Aplicada a Veículos Submarinos Semi-Autônomos (11).

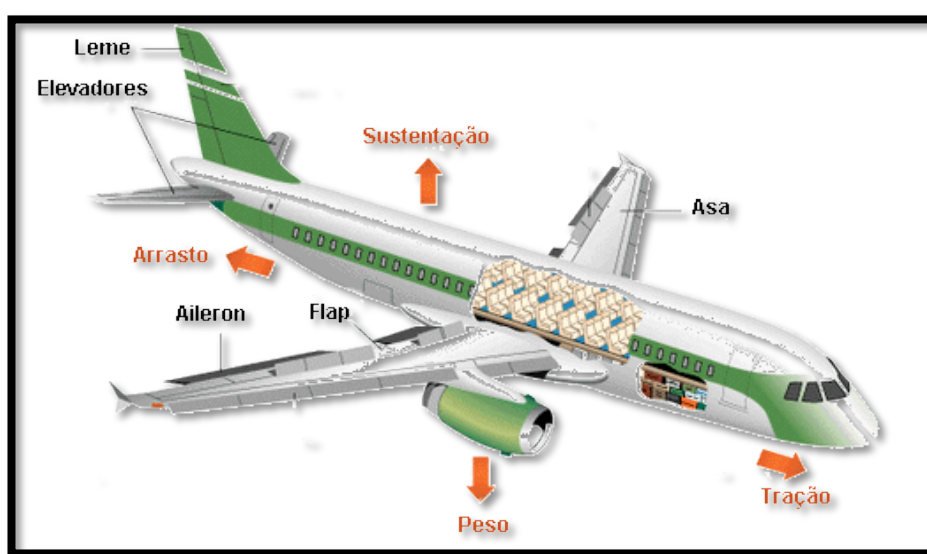
Analisando o diagrama retratado na Figura 6, é possível perceber diferentes formas de aplicação da simulação *hardware-in-the-loop* em uma malha de controle. É muito comum para este método de simulação que o sistema de controle da planta esteja em um hardware, a fim de que este seja testado antes de sua implantação em um sistema real. Porém, podem haver situações onde o sistema de controle não seja o foco da simulação. Exemplo disso, é se há a necessidade de conhecer o comportamento de um determinado atuador do sistema. Assumindo que o controle é implementado em um sistema simulado, mediante uma interface de potência que propicie o seu funcionamento, este atuador real recebe um sinal de controle e pode se comunicar com um processo real ou simulado, dependendo também do interfaceamento entre estes subsistemas. Sendo o processo simulado bem como os sensores, estabelece-se o fechamento da malha de controle onde o *hardware* é o atuador e todo o resto do sistema é implementado via *software* constituindo a técnica chamada de *Load Simulation* (11). Há também a possibilidade do *hardware* da simulação *hardware-in-the-loop* ser o sensor da planta, e todo o resto do sistema ser implementado em *software*. Apesar de não ser uma simulação muito usual, é possível a sua implementação desde que os devidos cuidados no interfaceamento entre a parte real e simulada seja feito de modo adequado. Porém, fica evidenciado na Figura 6 que há estratégias de simulação que não são possíveis de realizar. Exemplo disso é um atuador simulado operar sobre um processo real, ou ainda, um processo real ter alguma variável mensurada por um sensor simulado.

Usualmente, a simulação *hardware-in-the-loop* vem para auxiliar no desenvolvimento de controladores físicos para uso em sistemas reais. Uma vez que o dispositivo ou planta que será simulada já foi sintetizado em um simulador computacional, tem-se muitas vezes a presença de uma plataforma física microcontrolada a qual será incumbida de realizar o

controle da operação do sistema em simulação. A fim de exemplificar este modo de operação da simulação *hardware-in-the-loop*, segue a metodologia utilizada para a realização de um teste de um piloto automático para um sistema de aviação.

A ideia fundamental da ação de um piloto automático está em realizar o comando do posicionamento do *flap* presente nas asas do avião, que são responsáveis por fazer que a aeronave tenha um ângulo com relação à horizontal, como pode ser visto na Figura 7.

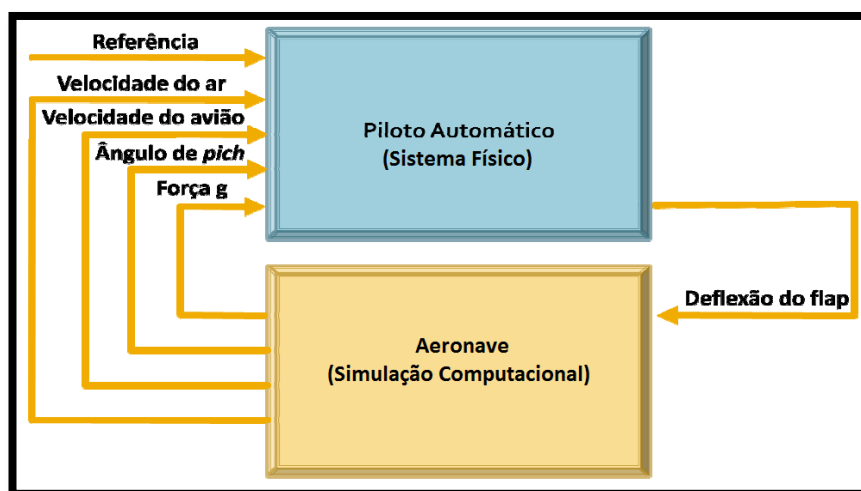
Figura 7 - Diagrama de Forças e Elementos de Posicionamento de uma Aeronave



Fonte: Instituto de Física da UFRGS (12).

No sistema descrito na Figura 8, deseja-se simular o comportamento do piloto automático de uma aeronave mediante as informações recebidas dos medidores que indicam as condições de voo atuais. Sendo assim, o Piloto Automático encontra-se embarcado em um sistema de processamento de sinais e recebe os dados vindos do sistema simulado em ambiente virtual da aeronave em voo. Como resposta, o sistema embarcado fornece a deflexão do flap o qual posiciona a aeronave em um ângulo desejado com relação à horizontal.

Figura 8 - Esquema da HILS de um Piloto Automático



Fonte: Adaptado de Hardware-In-The-Loop Para Desenvolvimento de Software Embarcado em DSPs Utilizando Ambiente PSCAD/EMTDC (10).

Assim, conclui-se que a principal função deste método de simulação é fazer com que seja possível testar a função de um controlador antes que este venha a ser colocado em funcionamento no processo real. Uma vez que o modelo que descreve o processo real apresenta uma grande fidelidade, é possível ajustar os ganhos do controlador via a simulação HIL e chegar-se a melhores resultados práticos do que uma aplicação direta no processo.

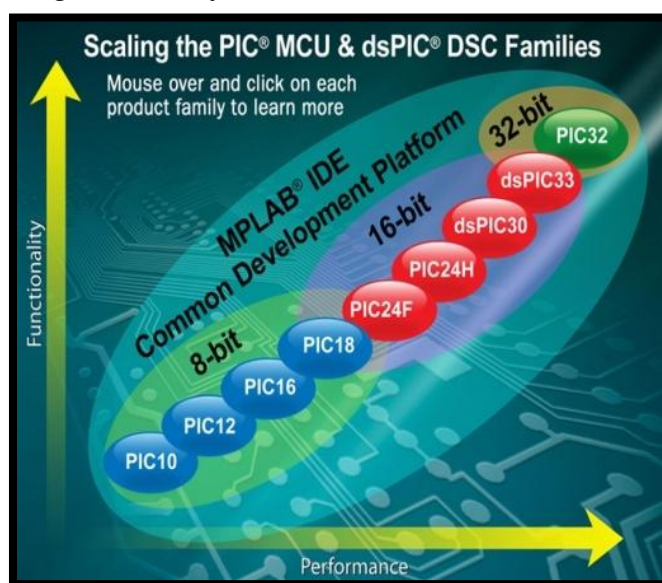
Diante de todo esse quadro de aplicações, este trabalho vem apresentar a utilização desta técnica empregando o *software* MATLAB/SIMULINK e o *hardware* dsPIC33FJ256GP710A. Uma simulação de um sistema de geração de energia eólica já foi desenvolvida nesta plataforma de simulação; sendo assim, parte da simulação será mantida e parte será implementada pelo *hardware*, permitindo a utilização desta técnica inovadora.

3 O MICROCONTROLADOR DSPIC33FJ256GP710A

A fim de entender as funcionalidades da placa Explorer16 é necessário conhecer o microcontrolador que é responsável por processar toda a informação nesta placa.

Há diversos tipos de microcontroladores disponíveis no mercado, cada qual com suas características construtivas e funcionais disponibilizadas pelos seus desenvolvedores. Estes estão divididos em famílias, e estas famílias foram surgindo ao longo dos anos em virtude do avanço tecnológico e da disponibilização de cada vez mais recursos de processamento e facilidades para sua programação. Especificamente, este projeto utiliza um microcontrolador que foi desenvolvido pela empresa Microchip. A evolução dos microcontroladores da Microchip pode ser vista na Figura 9.

Figura 9 - Evolução dos Microcontroladores da Família PIC



Fonte: EDN China (13).

O dsPIC33F256GP710A, como é mostrado na Figura 9, é um microcontrolador de 16 bits e pertence à família dsPIC33, uma das mais modernas deste fabricante, apresentando uma alta performance aliada a uma alta funcionalidade, ideal para aplicações que se baseiam em uma computação repetitiva de alta velocidade. Suas principais características podem ser vistas no Quadro 1 e sua pinagem na Figura 10.

3.1 A Placa Explorer16

O dsPIC33FJ256GP710A foi embarcado em uma placa de desenvolvimento elaborada pela Microchip, denominada Explorer16. Esta placa tem por objetivo permitir a exploração das funcionalidades disponíveis no microcontrolador de forma mais simples, uma vez que são disponibilizados vários dispositivos já interligados aos seus pinos. Os principais recursos disponíveis nesta placa podem ser vistos no Quadro 2 e a Figura 11 mostra a Explorer16.

Quadro 2 - Principais Recursos da Explorer16

Recursos
Teclas e LEDs (4 teclas e 8 LEDs);
Memória serial EEPROM 24WC256 (protocolo I ² C);
Memória serial EEPROM 25LC256 (protocolo SPI);
Sensor de temperatura MCP9700 (saída analógica);
Comunicação serial RS232;
Comunicação CAN;
Comunicação Ethernet;
Botão de reset manual;
Possibilidade de trabalhar com LCD 16x2 (alfanumérico) e LCD 128 x 64 (gráfico).
Compatível com os gravadores ICD2BR, ICD2 Microchip, PICkit e Real ICE Microchip.

Fonte: Guia do Usuário Explorer16BR dsPIC33F (16).

Figura 11 - A Explorer16



Fonte: Produção do próprio autor.

3.2 O Compilador CCS C Compiler e MPLAB ICD2

A elaboração do programa responsável pelo funcionamento do microcontrolador na placa é feita em um compilador que utiliza a linguagem C. Existem diversos tipos de compiladores que podem ser utilizados, cada um deles podendo suportar a programação de uma grande família de microcontroladores e podendo oferecer mais ou menos recursos para a manipulação das funcionalidades presentes nos dispositivos programados. São exemplos de compiladores o C30 que é disponibilizado pela Microchip e trabalha junto com o MPLAB, o MikroC e o CCS C Compiler.

Neste projeto, foi utilizado o compilador CCS C Compiler, devido a sua grande facilidade na programação dos recursos do dsPIC33FJ256GP710A. Com uma linguagem de programação e uma interface muito amigável para o trabalho, este compilador permite que o esforço do projetista seja direcionado para a elaboração da lógica de programação da aplicação e não para observação e manipulação minuciosa de cada registrador interno que é feito em uma linguagem de nível mais baixo.

Ao fim da programação da aplicação, é gerado um arquivo hexadecimal (.hex), que será gravado no dsPIC com o auxílio de uma programadora. Neste Projeto, foi utilizada a programadora da Microchip MPLAB ICD2. Esta programadora é capaz de programar os dispositivos das famílias PIC, dsPIC e PIC32 (17), possuindo assim uma larga aplicação e sendo também muito flexível a sua utilização, uma vez que permite a conexão ao computador mediante uma porta serial ou por uma comunicação USB. Na Figura 12 pode-se ver a programadora utilizada no projeto.

Figura 12 - A Programadora ICD2



Fonte: Produção do próprio autor.

3.3 A Comunicação Serial RS232

Dos recursos disponibilizados na placa Explorer16, foi explorada a comunicação serial RS232, a qual foi utilizada para estabelecer a comunicação entre o controlador programado na placa e a planta do sistema em simulação no Simulink.

A comunicação entre o dsPIC e o Simulink é estabelecida utilizando uma interface serial assíncrona denominada UART, do inglês *Universal Asynchronous Receiver Transmitter*. A placa possui um *driver* para adequar os níveis de tensão do microcontrolador que são do padrão TTL, onde o nível lógico “baixo” (definido eletricamente por uma tensão que varia entre 0 e 0,8 volts) e o nível lógico “alto” (de 2 a 5 volts) para o padrão RS232 onde os sinais válidos variam para o nível lógico “baixo” de -12 a -3V e para nível lógico alto, de 3 a 12V (16).

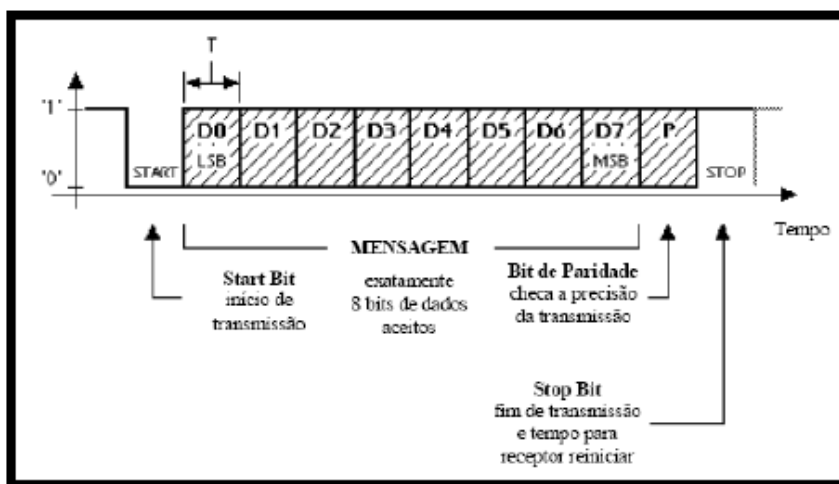
O dsPIC utiliza para a transmissão dos dados o pino RF5 chamado TX2 e para a recepção de dados utiliza o pino RF4, chamado RX2. Estes pinos são conectados a uma porta DB9 presente na placa. O computador utilizado no projeto não contava com uma porta serial, que são mais comuns em equipamentos mais antigos, assim, foi necessário utilizar um cabo de conversão do DB9 presente na placa para o padrão USB com a finalidade de estabelecer a conexão entre placa e computador. Para um correto funcionamento é necessário atentar para a necessidade de instalação de um *driver* para que o computador possa reconhecer este cabo de conversão.

Os dados são enviados e recebidos pelo dsPIC mediante aos respectivos *buffers* de envio e recebimento. É possível enviar mensagens que contenham até 8 bits de dados, e juntamente com a mensagem são enviados alguns bits de sinalização. Para que seja estabelecida a comunicação serial, deve-se ficar atento a configuração dos seguintes parâmetros (18):

- Bits de dados: 7 ou 8 bits;
- Paridade: Não, Par ou Ímpar;
- Stop bits: 1 ou 2 bits
- *Baud Rate* – 110, 300, 600, 1200, 4800 bits/s.

Quanto à paridade, pode ser escolhido se este bit estará ativo ou inativo. Caso esteja ativo, indicará se o número da mensagem é par ou ímpar. Para que seja sinalizado onde a informação acaba, são utilizados os *stop bits*. A taxa com a qual os bits são transmitidos é denominada *Baud Rate*. Além dos *bits* já citados, há também o bit que inicializa a transmissão, chamado *Start Bit*. A disposição destes bits nos buffers de recepção e envio pode ser visto na Figura 13.

Figura 13 - Estrutura da Transmissão Serial



Fonte: A Interface Serial e o padrão RS-232 (18).

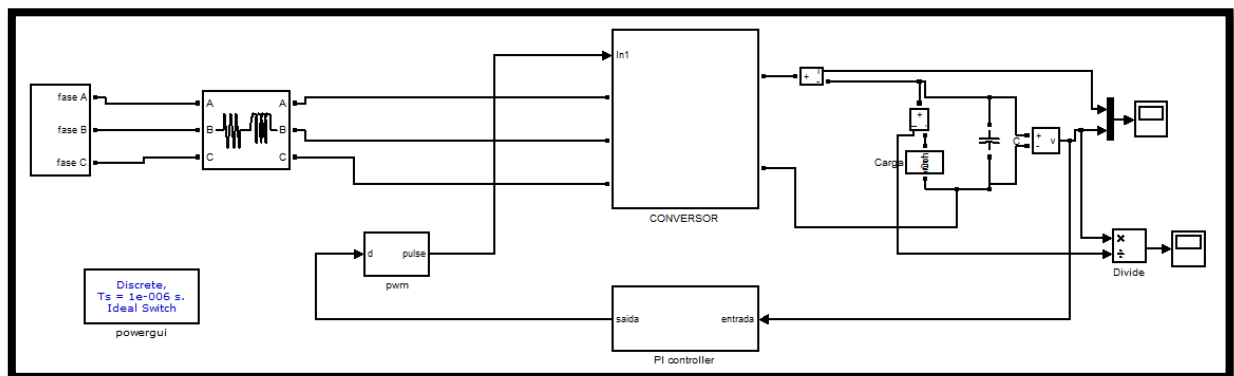
Uma vez definida qual será a taxa de transmissão a ser utilizada na aplicação, é possível conhecer o tempo T , em segundos, para que um bit do *buffer* seja transmitido, dado pela Equação (1).

$$T = \frac{1}{\text{Baud Rate}} [s] \quad (1)$$

4 SIMULAÇÃO PURAMENTE COMPUTACIONAL

A implementação realizada no Simulink trata-se da simulação de um sistema de geração eólica que tem início no sistema de geração e chega até um barramento CC de 400V de tensão. Os principais dispositivos inseridos neste caminho são tratados separadamente nas seções que seguem para que seja possível ter o entendimento do funcionamento da simulação como um todo. O sistema completo pode ser visto na Figura 14.

Figura 14 - Sistema de Geração Eólica Simulado



Fonte: Produção do próprio autor.

4.1 O Sistema de Geração

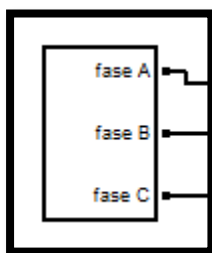
O sistema implementado no Simulink consiste primeiramente de uma estrutura que representa a geração da energia eólica, que se dá pela ação da força dos ventos nas pás que movimentam o rotor do gerado elétrico. A estrutura que simula a geração eólica (Figura 15) possui três saídas de tensão, sendo estas as fases V_a , V_b e V_c apresentadas nas Equações (2), (3) e (4), respectivamente.

$$V_a(t) = \sqrt{2} V_f \cos (wt) \quad (2)$$

$$V_b(t) = \sqrt{2} V_f \cos (wt + 120^\circ) \quad (3)$$

$$V_c(t) = \sqrt{2} V_f \cos (wt - 120^\circ) \quad (4)$$

Figura 15 - Bloco de Geração

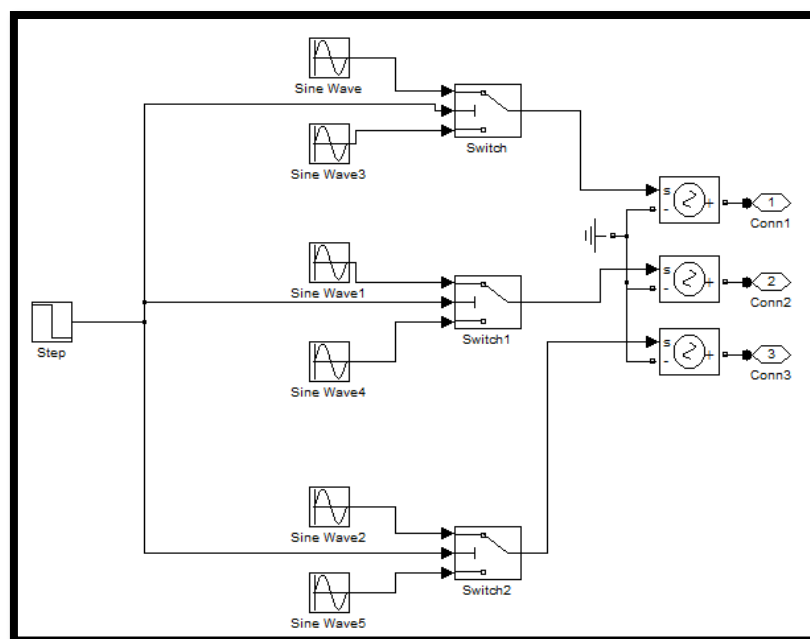


Fonte: Produção do próprio autor.

Onde V_f é o valor RMS da tensão gerada em seu interior.

A Figura 16 apresenta a estrutura interna do bloco de geração das tensões.

Figura 16 - Estrutura Interna do Bloco de Simulação da Geração



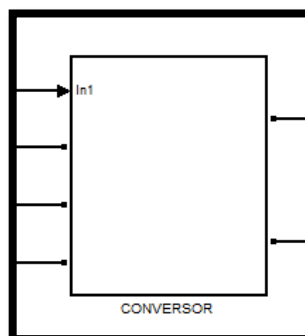
Fonte: Produção do próprio autor.

A frequência das tensões geradas varia conforme a velocidade do vento que incide nas pás da turbina. Assim, considerando as velocidades típicas deste tipo de gerador, teremos que para um vento com velocidade de 3m/s, as tensões geradas possuirão uma frequência de 10Hz e para uma velocidade de 12m/s tem-se uma frequência de 40Hz (5). Pode-se variar a tensão de saída do bloco gerador em um instante de tempo especificado no bloco "Step". Ao variar a saída deste bloco seleciona-se uma tensão de saída específica, simulando uma variação do vento na entrada da turbina.

4.2 O Conversor CA-CC

No centro desta aplicação, encontra-se o conversor CA-CC com o qual se deseja realizar o controle de tensão. Seu bloco no sistema e sua estrutura interna podem ser vistos respectivamente nas Figuras 17 e 18.

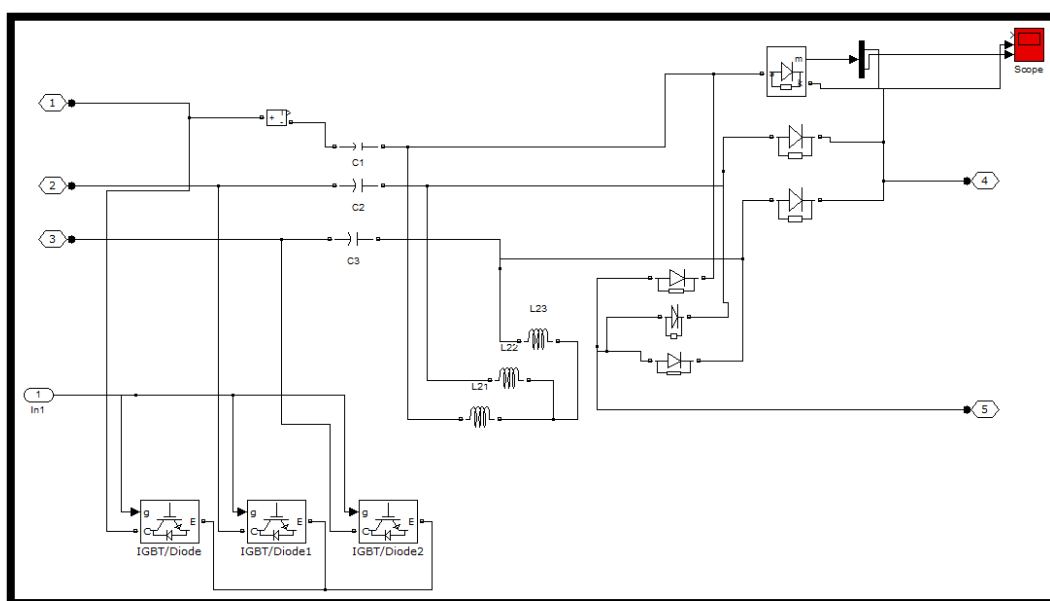
Figura 17 - O Conversor CA-CC



Fonte: Produção do próprio autor.

Como é possível ver, na Figura 17, o conversor conta com 4 entradas (situadas à esquerda do bloco) onde a entrada “In1” o sinal o PWM que comandará o acionamento dos IGBTs no interior do conversor e as 3 entradas subsequentes são as fases A, B e C oriundas do bloco de geração do sistema simulado. As duas saídas (situadas do lado direito) do bloco são referentes aos pólos positivos e negativos da tensão contínua obtida.

Figura 18 - Estrutura Interna do Conversor CA-CC



Fonte: Produção do próprio autor.

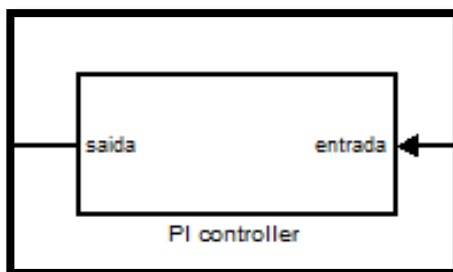
A topologia deste conversor é a SEPIC, do inglês *Single-Ended Primary-Inductor Converter*, e sua descrição detalhada pode ser conferida em (5). Este conversor por sua vez, alimenta uma carga resistiva constante durante toda a simulação.

O objetivo deste conversor é manter a saída CC constante, (no caso 400V) apesar das variações de tensão CA na sua entrada devido às oscilações de vento. Sendo assim, ao efetuar o controle do disparo na chave semicondutora (através do ajuste da razão cíclica) ajusta-se o mesmo valor da tensão de saída para as diversas entradas de tensão.

4.3 O Sistema de Controle de Tensão

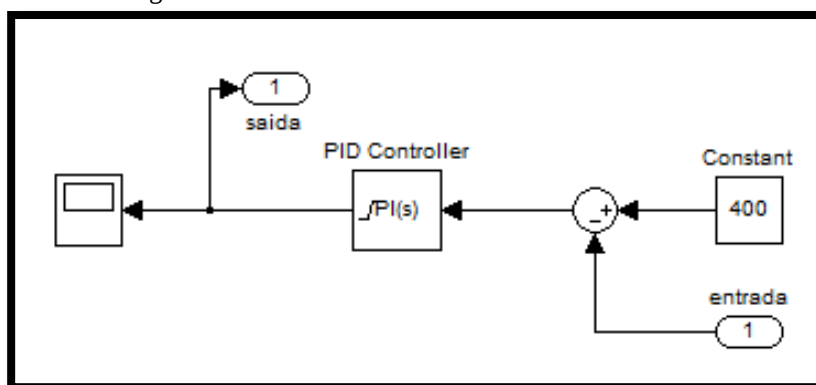
Como resultado da operação deste sistema de geração de energia, deseja-se obter uma tensão de 400V contínua no barramento de saída. Para que este objetivo seja alcançado faz-se necessária a presença de um controlador no sistema. O bloco referente ao controlador no sistema simulado pode ser visto na Figura 19 e sua estrutura interna na Figura 20.

Figura 19 - Bloco do Controlador no Sistema Simulado



Fonte: Produção do próprio autor.

Figura 20 - Estrutura Interna do Controlador Simulado



Fonte: Produção do próprio autor.

O controlador utilizado nesta aplicação trata-se do Proporcional Integral (PI), no qual em sua entrada é colocado o sinal do erro apresentado pelo sistema que é a subtração do sinal de referência de tensão do sistema pelo valor da tensão de saída do conversor no instante atual. Como pode ser visto na Figura 20, tem-se a presença do bloco PID do Simulink, porém, a sua parte derivativa é suprimida, uma vez que é atribuído um ganho da constante derivativa k_d igual a zero.

A saída do controlador irá atuar de forma que seja possível estabilizar o sistema no nível de tensão indicado pelo valor de referência. Sendo assim, a resposta obtida na saída do controlador será a razão cíclica que é aplicada ao sinal do bloco PWM, que por sua vez atua no disparo dos IGBTs que estão presentes na entrada do conversor CA-CC.

Uma vez que não se tem o modelo matemático que descreve o conversor, não foram utilizados métodos clássicos para obter a sintonia do controlador, chegando-se aos valores das constantes k_p e k_i de forma empírica. Para a simulação realizada com este controlador, têm-se os valores descritos no Quadro 3.

Quadro 3 - Constantes do Controlador PI do Sistema Simulado

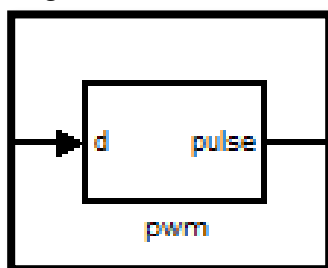
Constante	Valor
k_p	0,5
k_i	100

Fonte: produção do próprio autor.

4.4 O Sinal PWM

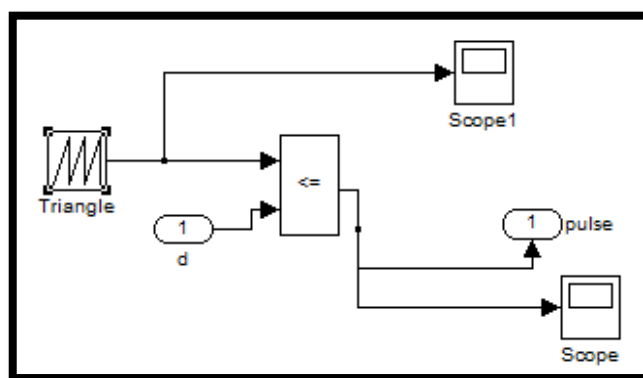
Uma vez que o sistema de controle do conversor atuou, o mesmo forneceu um valor de razão cíclica para o subsistema que gera um sinal PWM adequado ao nível de tensão que foi medido no barramento de saída do conversor. O bloco responsável pela geração do PWM pode ser visto na Figura 21 e a sua estrutura interna na Figura 22.

Figura 21 - Bloco do PWM



Fonte: Produção do próprio autor.

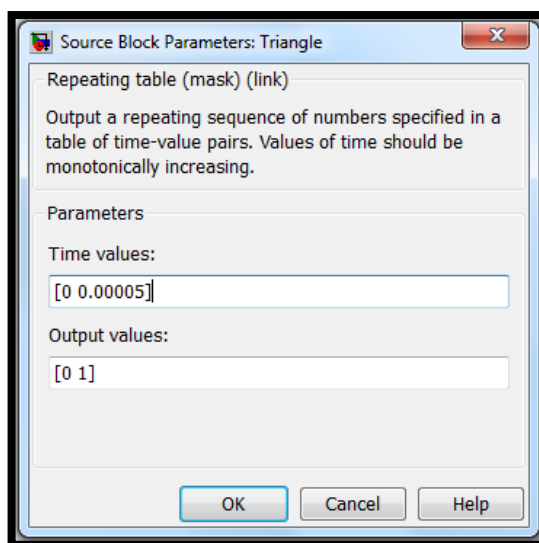
Figura 22 - Estrutura Interna do Bloco do PWM



Fonte: Produção do próprio autor.

O bloco do PWM recebe o valor da razão cíclica, que é um número real, positivo e pode, para esta aplicação, variar entre 0,25 e 0,5 dependendo da dinâmica apresentada pelo sistema (5). Este valor indica a qual a porcentagem do período do sinal PWM está em nível lógico “alto”, permitindo assim o fechamento dos IGBTs presentes no conversor.

A geração da onda de saída do bloco do PWM se dá por meio da comparação de uma onda triangular do tipo dente-de-serra com o valor da razão cíclica vinda do controlador PI. Esta onda dente-de-serra é gerada pelo bloco *Triangle* do Simulink e possui uma amplitude que varia de 0 a 1 e um período de oscilação de 50µs, o que a leva a ter uma frequência 20kHz, como pode ser vista em sua janela de configuração na Figura 23.

Figura 23 - Janela de Configuração do Bloco *Triangle*

Fonte: Produção do próprio autor.

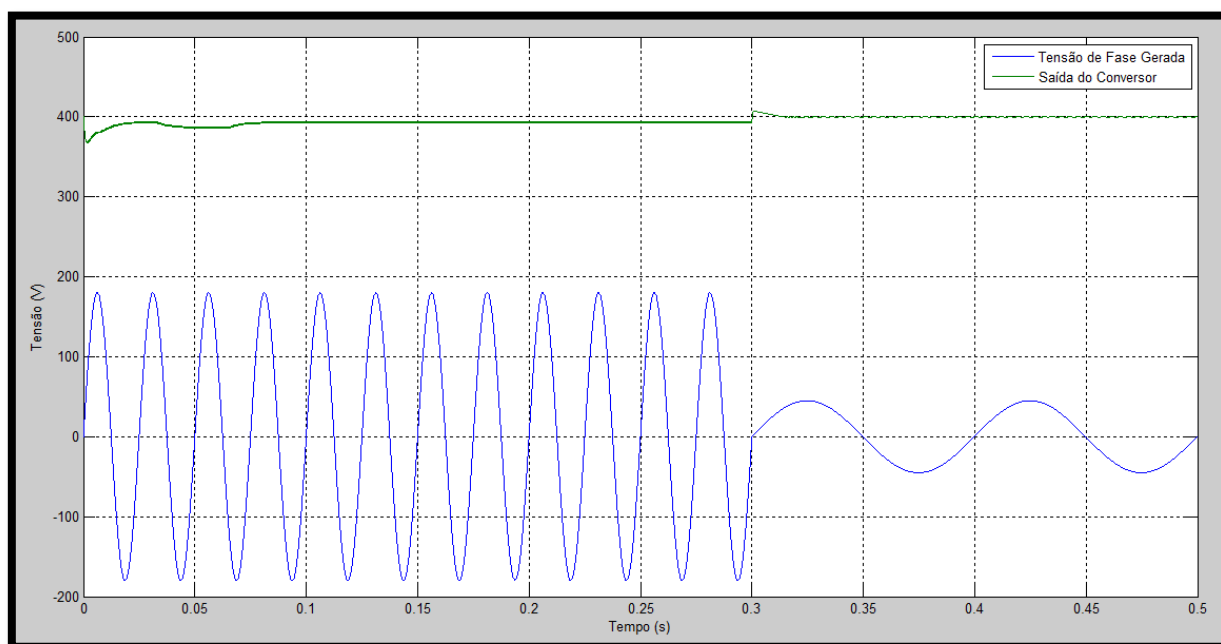
Como resultado da comparação, é obtida uma onda quadrada, com a mesma frequência de oscilação da onda triangular, com uma amplitude apresentando um valor binário que é consequência da comparação que é realizada. Enquanto a amplitude da onda triangular é menor ou igual ao valor da razão cíclica, tem-se uma amplitude da onda de saída igual a 1, caso contrário, a saída assumirá um valor igual a zero. Com isso, sempre será mantida constante a frequência da onda de saída e será variada a largura do pulso desta onda, como é definido este tipo de modulação.

Por fim, o sinal PWM gerado é aplicado aos IGBTs do conversor, permitindo que seja possível o funcionamento do mesmo.

4.5 A Simulação do Sistema

O resultado da simulação, apresentada na Figura 24, é de um sistema que opera nominalmente com velocidade do vento de 12m/s gerando tensões cuja frequência é de 40Hz.

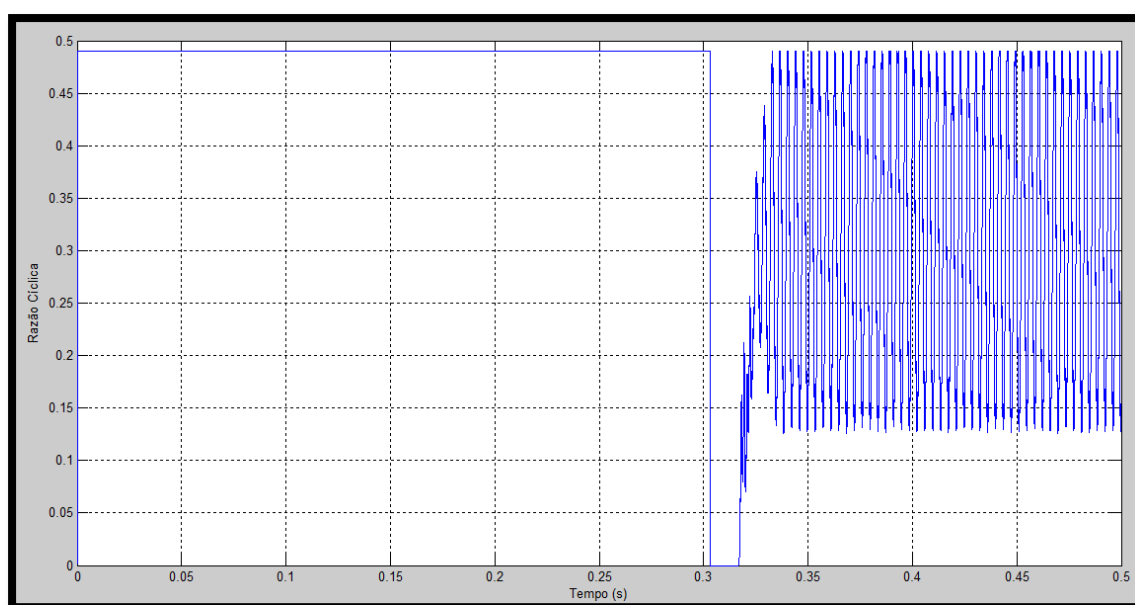
Figura 24 - Tensão na Saída do Conversor e Tensão de Fase Gerada



Fonte: Produção do próprio autor.

O tempo de simulação é de 0,5 segundos, sendo que até o instante de tempo $t = 0,3$ segundos as condições são as nominais; em $t = 0,3$ segundos há uma queda na velocidade do vento para 3m/s, resultando em tensões cujas frequências é de 10Hz. A razão cíclica para a simulação em questão pode ser vista na Figura 25.

Figura 25 - Razão Cíclica Durante a Simulação



Fonte: Produção do próprio autor.

O resultado da atuação do controlador pode ser observado pois mesmo com a variação de tensão de entrada, a tensão de saída do conversor manteve-se em 400V.

As medições realizadas para esta simulação levaram aos valores apresentados no Quadro 4.

Quadro 4 - Resultados Obtidos da Simulação Puramente Computacional

Frequência	Parâmetro	Valor
40Hz	Tempo de Acomodação [s]	0,1
	Tensão Final [V]	393
10Hz	Tempo de Acomodação [s]	0,015
	Valor de Pico [V]	407
	Tensão Final [V]	400

Fonte: Produção do próprio autor.

Analisando o quadro acima, é possível chegar a algumas conclusões. O Erro estacionário apresentando pelo sistema antes da mudança de velocidade do vento é dado por:

$$Erro_{40Hz} [\%] = \left(1 - \frac{393}{400}\right) 100$$

$$Erro_{40Hz} = 1,75 \%$$

Outra informação importante que pode ser retirada dos dados do Quadro 4 é o sobressinal apresentado pelo sistema quando ocorre a transição de velocidade do vento.

$$Sobressinal [\%] = \left(\frac{407}{400} - 1\right) 100$$

$$Sobressinal = 1,75 \%$$

Estes resultados são referência para a nova forma de simulação que é descrito no capítulo a seguir.

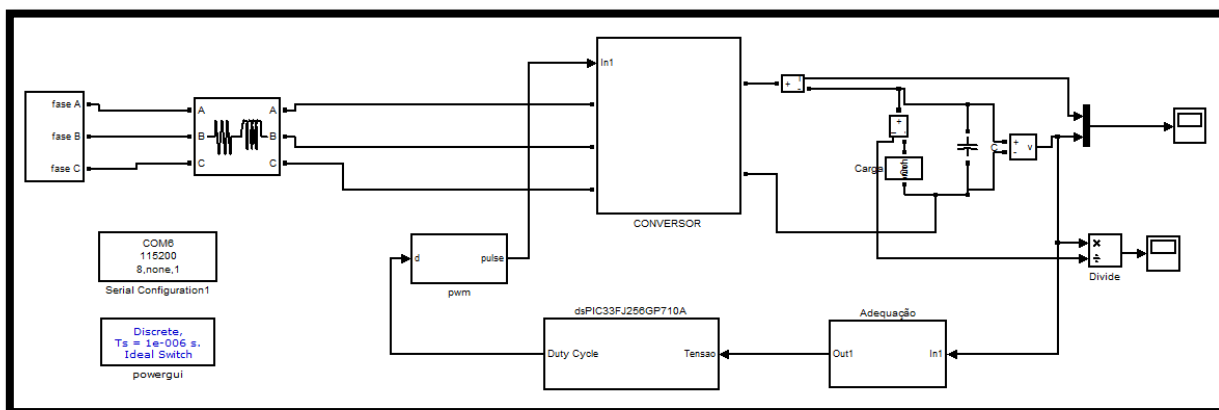
5 A SIMULAÇÃO DO SISTEMA COM *HARDWARE-IN-THE-LOOP*

Com o objetivo de realizar a simulação *hardware-in-the-loop* no sistema apresentado na Figura 14, foi inserido o dsPIC33FJ256GP710A que está presente na placa Explorer16 que opera substituindo o bloco controlador PI, dessa forma, ficando a cargo desde sistema microcontrolado o controle do conversor.

É sabido que o PWM no dsPIC33FJ256GP710A é implementado em hardware. Todas as características desejadas para o sinal PWM são previamente estabelecidas no momento da configuração do dispositivo microcontrolado. O PWM no dsPIC opera segundo um *timer*, que é um contador interno do microcontrolador. Este *timer* opera por meio de uma interrupção interna do dispositivo e a cada estouro de sua contagem, realiza uma ação determinada. Este *timer* define então qual será a frequência de operação do PWM. Esta frequência permanece constante, sendo variada a sua razão cíclica conforme a necessidade da planta. Assim, não há um código que sintetize um PWM presente na lógica de programação. Assim, o PWM que comanda o conversor desta aplicação é implementado via um sistema de comparação entre uma onda portadora e a razão cíclica, que é fornecida como resposta do controlador implementado no código do dsPIC.

Para realizar então o controle do conversor, foram necessárias algumas alterações no sistema apresentado anteriormente. O sistema que realiza a simulação *hardware-in-the-loop* pode ser visto na Figura 26.

Figura 26- Sistema Para Simulação HIL



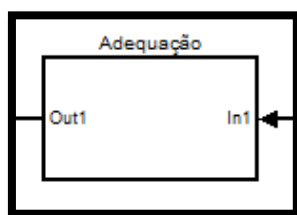
Fonte: Produção do próprio autor.

As alterações que foram realizadas são citadas e explicadas ao longo deste capítulo.

5.1 O Bloco de Adequação de Sinal

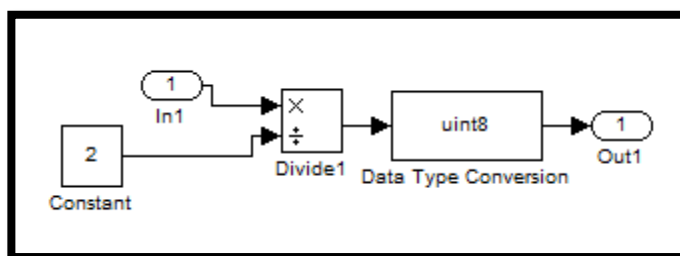
A primeira alteração realizada no sistema foi a inserção de um bloco de adequação do sinal de tensão que será tratado posteriormente pelo sistema microcontrolado. O bloco citado pode ser visto na Figura 27 e sua estrutura interna na Figura 28.

Figura 27 - O Bloco de Adequação de Sinal



Fonte: Produção do próprio autor.

Figura 28 - Estrutura Interna do Bloco de Adequação de Sinal



Fonte: Produção do próprio autor.

A necessidade da presença deste bloco na entrada do sistema microcontrolado vem do modo de comunicação serial RS232 suportar transmitir dados de até 8 bits, ou seja, 1 byte. Tendo o *buffer* de transferência e recepção de dados a capacidade de transmissão de 1 byte de resolução, é possível transmitir dados que variam de 0 a 255. Como a tensão medida na saída do conversor atinge valores próximos ou exatamente 400V, dividiu-se este valor de tensão por 2, fazendo seja possível transmitir os dados com o 1 byte disponível no buffer para informação.

A preocupação com o tipo de variável de cada uma das conexões do sistema é de suma importância para o correto funcionamento do sistema. Muitos erros são ocasionados pela falha na forma de tratar uma variável presente no sistema. Os problemas podem ir desde a perda de informações e o comprometimento dos dados do sistema até a incapacidade de colocar a

simulação em funcionamento. Assim, é importante ter o conhecimento da resolução suportada por cada tipo de variável que aparece no sistema. As resoluções mostradas no Quadro 5 auxiliam a definir as variáveis tanto na programação do código embarcado no microcontrolador quanto no tratamento dos dados que circulam na simulação.

Quadro 5 - Resolução dos Tipos de Variáveis Mais Utilizados

Tipos de Dados	Tamanho em bits	Descrição
int	8	variáveis inteiras sem sinal, capacidade de 0 a 255
signed int	8	variáveis inteiras com sinal, capacidade de -128 a 127
long	16	variáveis inteiras sem sinal, capacidade de 0 a 65535
signed long	16	variáveis inteiras com sinal, capac. de -32768 a 32767
float	32	variáveis reais com sinal. Representação aproximada
short ou boolean	1	variáveis lógicas, de um bit, podendo valer 0 ou 1
int32	32	variáveis inteiras sem sinal, podendo valer de 0 a 4294967295
signed int32	32	variáveis inteiras com sinal, podendo armazenar valores de -2147483648 a 2147483647
char	8	variáveis que armazenam caracteres em forma de bytes

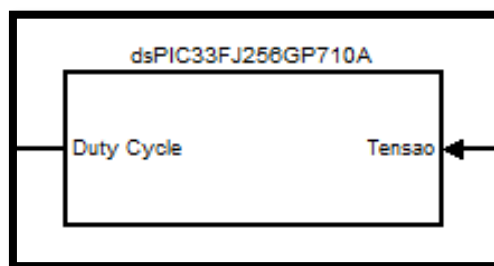
Fonte: Funções Mais Utilizadas da Linguagem C Padrão CCS (19).

A variável de tensão do barramento CC está em um formato *float* que possui 32 bits de tamanho e a constante que a divide está em um formato inteiro que possui 16 bits de resolução, sendo assim, o resultado desta divisão está em um formato *float* de 32 bits, sendo necessário utilizar um bloco de conversão de tipo de dado (bloco uint8) para que seja possível disponibilizar um dado de 1 byte para que este seja transmitido.

5.2 O Controlador Digital no dsPIC33FJ256GP710A

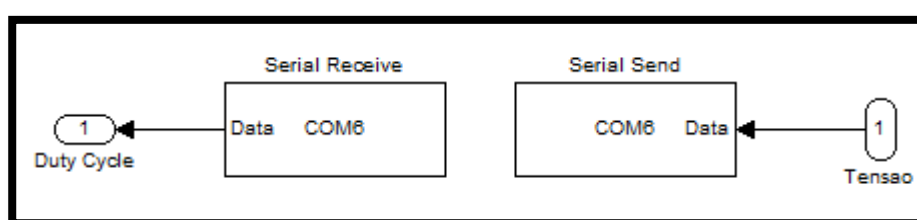
Após a adequação do tamanho dos dados que são transmitidos, os mesmos são transmitidos de forma serial para o dsPIC. A ação do dsPIC é similar a ação do bloco do controlador PI que era utilizado no sistema simulado de forma puramente computacional, de forma a fornecer em sua saída um valor de razão cíclica que é utilizada para produzir o sinal PWM. O bloco do dsPIC33FJ256GP710A pode ser visto na Figura 29 e sua estrutura interna na Figura 30.

Figura 29 - O Bloco do dsPIC33FJ256GP710A



Fonte: Produção do próprio autor.

Figura 30 - Estrutura Interna do Bloco dsPIC33FJ256GP710A

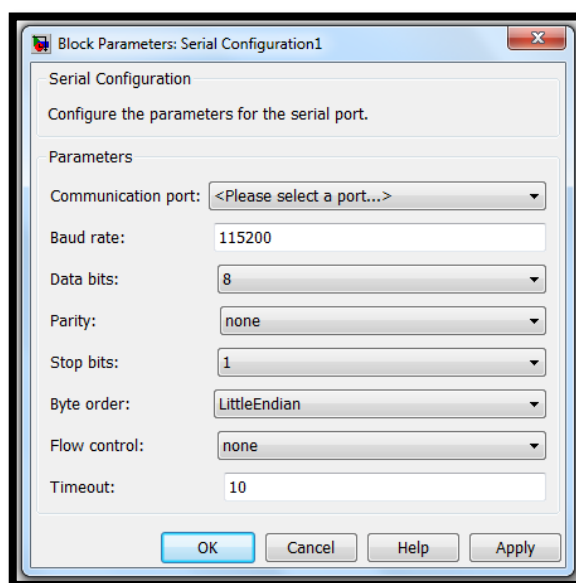


Fonte: Produção do próprio autor.

Como pode ser visto na Figura 30, a estrutura interna do bloco dsPIC33FJ256GP710A é extremamente simples, contendo apenas os blocos de transmissão do sinal do sistema simulado para o dsPIC e o de recebimento do valor fornecido como resposta do controlador digital implementado no microcontrolador.

Para a utilização da biblioteca de comunicação serial que está disponível no Simulink é necessário antes configurar os parâmetros para o seu correto funcionamento. Todos os blocos citados na comunicação entre o dispositivo e o Simulink encontram-se disponíveis na biblioteca *Instrument Control Toolbox*. O primeiro bloco a ser configurado é o *Serial Configuration*, no qual estão presentes todos os parâmetros que regem como deve se dar a comunicação serial quanto ao tamanho da mensagem que é transmitida e recebida, o tipo da variável, bits de parada, dentre outros. A janela de configuração do bloco *Serial Configuration* pode ser vista na Figura 31.

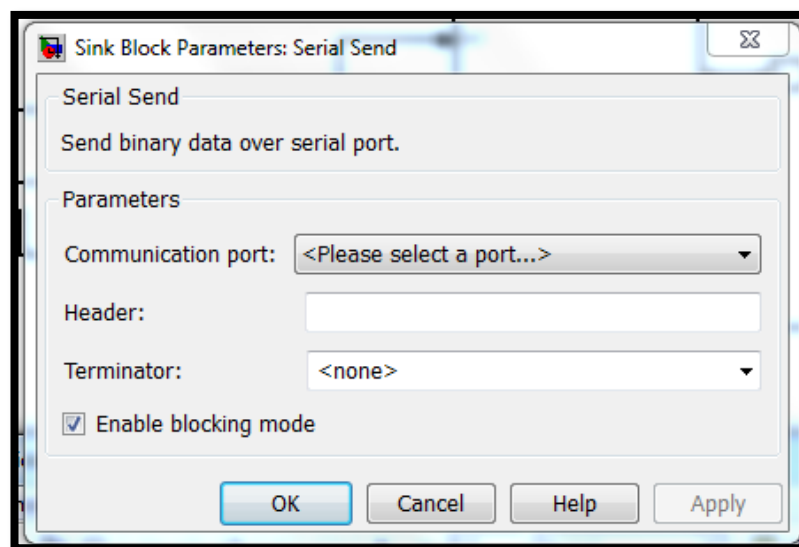
Figura 31 - Configuração do Bloco Serial Configuration



Fonte: Produção do próprio autor.

Como pode ser visto na Figura 31, o valor de *baud rate* escolhido para esta aplicação é de 115200 bits/s, um dos maiores valores disponíveis para este padrão de comunicação, visando que seja possível ter uma transmissão e recebimentos dos dados o mais veloz possível, com a intenção de minimizar a lentidão característica deste tipo de comunicação. O tamanho da mensagem transmitida é de 8 bits (o maior tamanho de palavra disponível), não possuindo bit de paridade, tendo um bit de parada, um bit de início de transmissão e tendo uma organização dos bits da mensagem seguindo uma sequência *LittleEndian*, onde o bit menos significativo ocupa a primeira posição e o mais significativo, a última. O número da porta serial a qual o dispositivo está conectado também deve ser informado. Assim, toda a configuração que realizada para a comunicação serial foi com a intenção de transmitir o maior número de informação possível, aliando alta velocidade de transferência e boa resolução das mensagens trocadas entre simulação e hardware.

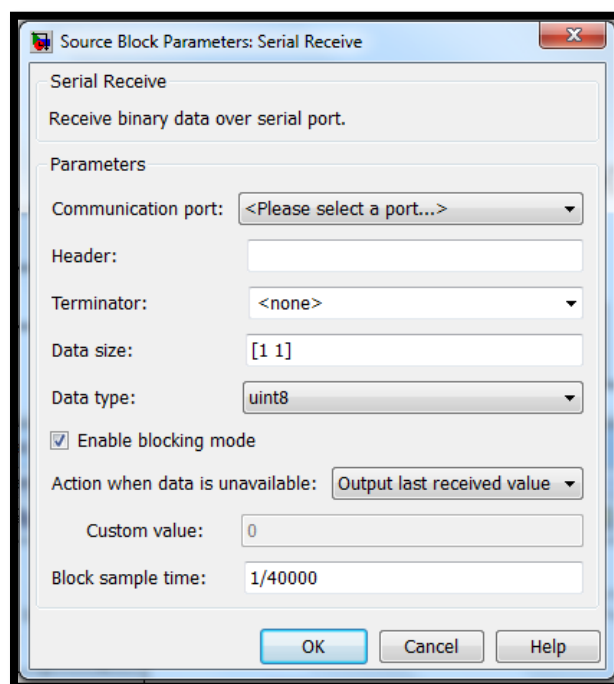
Depois de terminada esta configuração preliminar da comunicação serial, tem-se ainda que configurar cada um dos blocos de envio e recebimento de informação. Tratando primeiramente o bloco *Serial Send*, tem-se para ele a janela de configuração mostrada na Figura 32.

Figura 32 - Configuração do Bloco *Serial Send*

Fonte: Produção do próprio autor.

O bloco *Serial Send* é responsável pelo envio da tensão, que foi medida e tratada pelo bloco de adequação, para o dsPIC. Sua configuração é muito simples como é possível ver, alterando somente para este bloco qual o número da porta serial do computador está sendo usada pela aplicação.

O último bloco do sistema de comunicação serial configurado foi o bloco *Serial Receive*. Este bloco realiza o recebimento da informação que foi tratada pelo dsPIC. Assim como o *Serial Send*, este bloco não possui uma configuração complexa como mostra a Figura 33.

Figura 33 - Configuração do Bloco *Serial Receive*

Fonte: Produção do próprio autor.

Na janela de configuração mostrada acima, são alterados três parâmetros. O primeiro deles, a exemplo dos outros dois blocos já citados, é a porta serial utilizada para a comunicação. O segundo diz respeito ao tipo de dado que é recebido na comunicação. Já o terceiro parâmetro é o *sample time* do sistema. O *sample time* é o tempo de execução de cada passo do Simulink, agindo como uma amostragem do sistema.

A saída então do bloco *Serial Receive* é a razão cíclica necessária para o PWM operar, que foi calculado segundo a lógica definida dentro do controlador embarcado no dsPIC. Para efetuar a programação do microcontrolador seguiram-se os procedimentos descritos na seção 5.2.1.

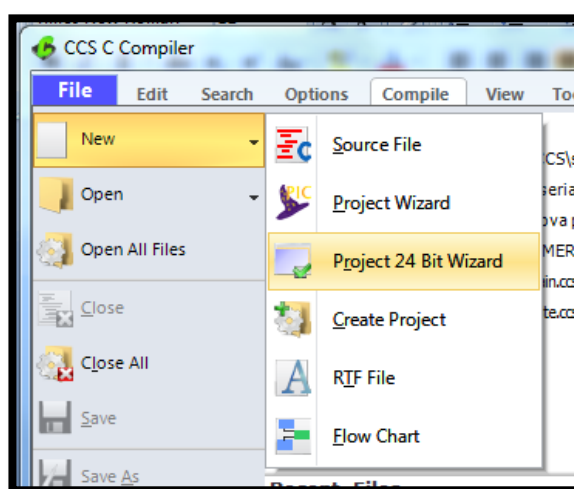
5.2.1 A Programação do dsPIC33FJ256GP710A

Como mencionado anteriormente, para gerar o código em linguagem C para a programação do dsPIC foi utilizado compilador CCS C Compiler. Inicialmente, a programação do dsPIC se dá com a configuração do modo de operação que este dispositivo deve ter. Uma grande virtude do compilador que foi adotado é a facilidade proporcionada ao programador, trazendo

uma interface muito amigável e direta, tanto para a configuração inicial do dispositivo quanto na construção do código em si.

Para realizar a configuração inicial do dsPIC, deve-se criar um projeto, no qual será possível fazer com que vários arquivos, como bibliotecas e cabeçalhos dos recursos utilizados na programação, possam comunicar-se entre si para o funcionamento do programa desenvolvido. O procedimento para a criação de um novo projeto é visto na Figura 34.

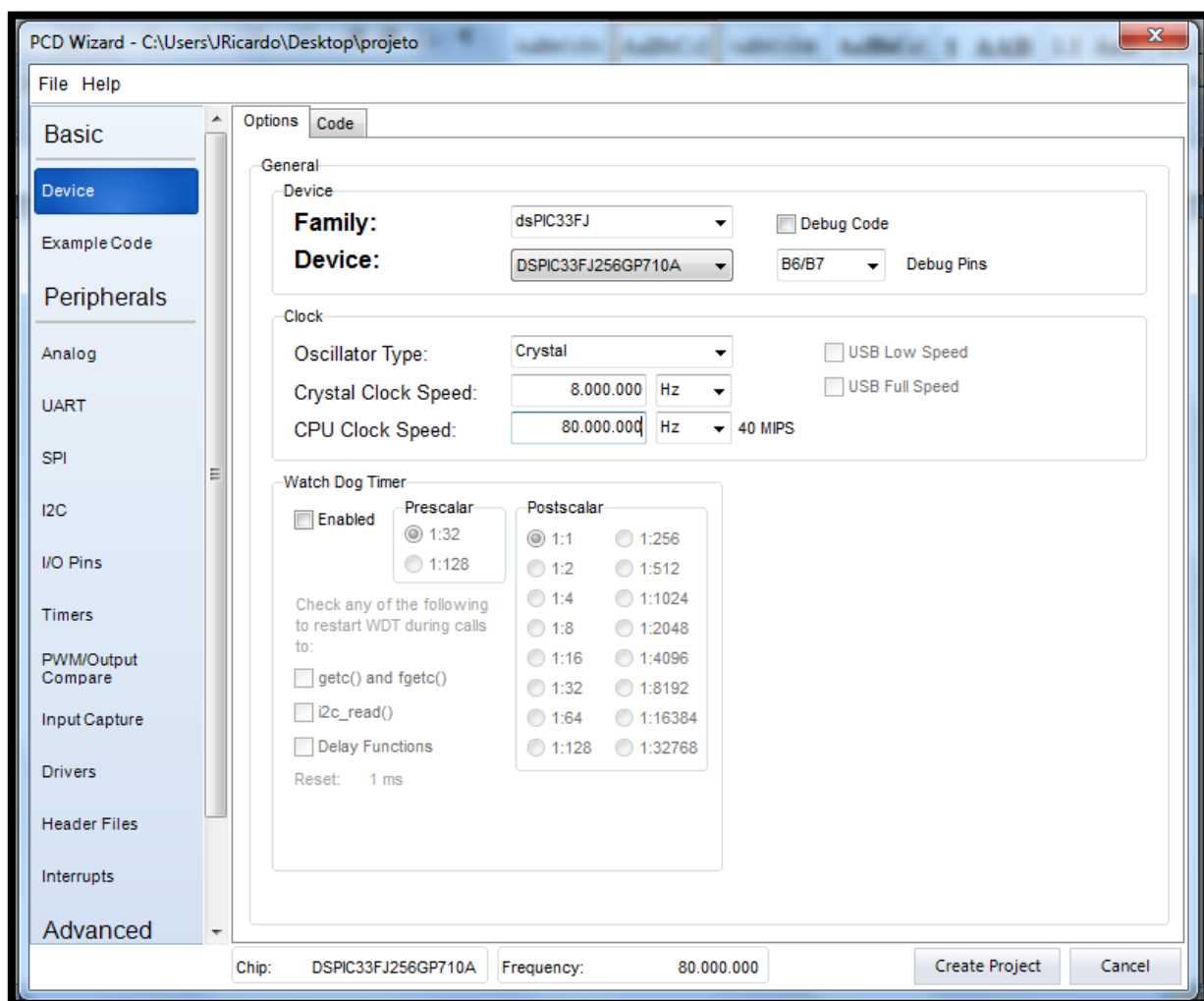
Figura 34 - Criação de Um Novo Projeto Para dsPIC no CCS C Compiler



Fonte: Produção do próprio autor.

Será aberta uma janela para que possa ser dado um nome ao projeto e salvá-lo em um diretório. Após isso, será aberta uma nova janela onde será possível realizar a configuração do microcontrolador. Esta janela pode ser vista na Figura 35.

Figura 35 - Janela de Configuração do Microcontrolador

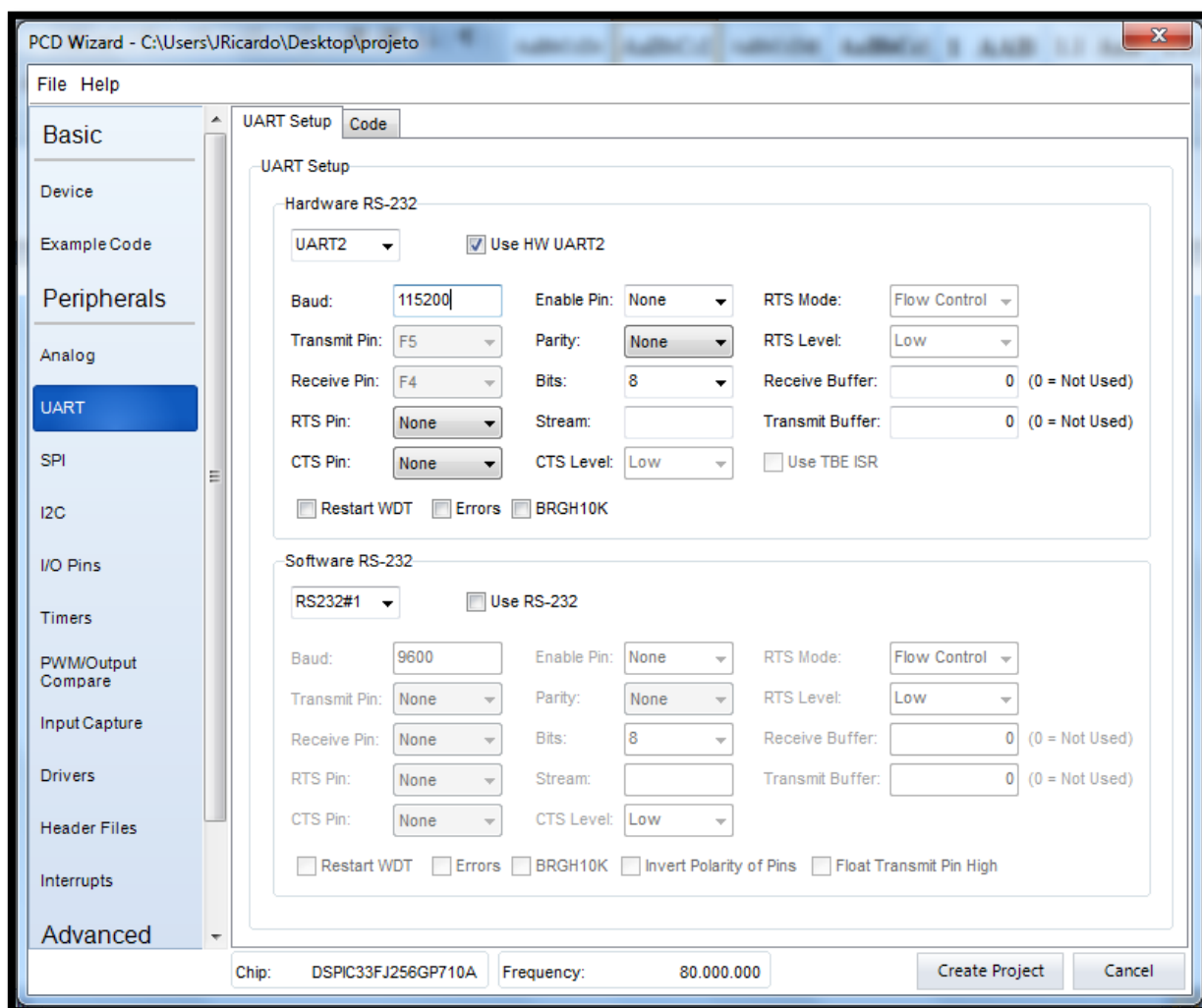


Fonte: Produção do próprio autor.

Na janela de configuração é escolhido o modelo do microcontrolador a ser programado e como será o *clock* que rege o funcionamento do dispositivo. Segundo informações do fabricante, a placa Explorer16 conta com um cristal oscilador de 8MHz e o dsPIC pode trabalhar com uma velocidade de 40 milhões de instruções por segundo, ou 40 MIPS, o que permite que o microcontrolador opere em uma frequência de 80MHz.

Passada a configuração inicial do dispositivo, é possível também realizar a configuração de itens periféricos do microcontrolador. Para isso, basta na coluna ao lado esquerda da janela de configuração, selecionar qual o periférico desejado e configurá-lo. Neste projeto foi utilizado o periférico de comunicação serial do microcontrolador e a sua configuração pode ser vista na Figura 36.

Figura 36 - Configuração da Comunicação Serial no Microcontrolador



Fonte: Produção do próprio autor.

Para que haja o correto funcionamento da comunicação, as configurações realizadas no Simulink e no periférico do microcontrolador devem ser idênticas. Para isso basta selecionar uma das duas opções de comunicação UART disponíveis no microcontrolador e informar o *baud rate* desejado, pois as demais configurações já vêm como *default* igual àquelas realizadas no Simulink. Sendo assim, foi selecionada em *Hardware RS-232* na janela de configuração a comunicação UART2 do microcontrolador e informado o *baud rate* de 115200 bits/s.

Outra funcionalidade do microcontrolador que foi explorada nesta aplicação é o uso de Interrupções. As interrupções são sinalizações de acontecimentos ocorridos, as quais são percebidas e tratadas pelo microcontrolador. Este recurso dos microcontroladores pode ser de

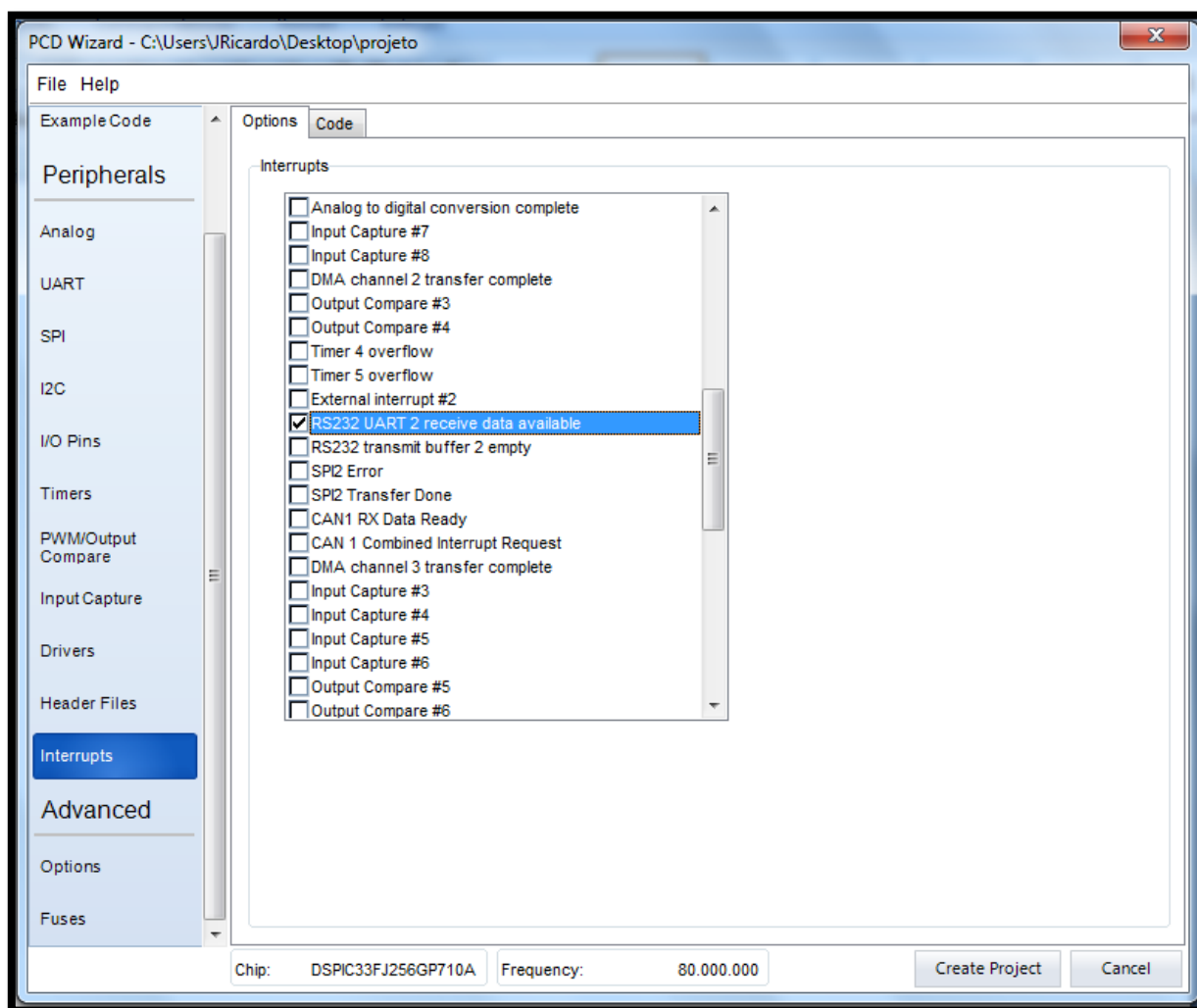
grande importância para algumas aplicações. Quando o microcontrolador recebe um sinal de ocorrência de um determinado evento é possível desviar, de forma muito veloz, a execução do programa principal para uma sub-rotina que irá tratar o acontecimento ocorrido e voltará ao programa principal para retomar a execução do ponto exato em que parou. Sendo assim, esta capacidade do microcontrolador é aplicada de forma a aperfeiçoar o código do ponto de vista que se torna desnecessário checar, a cada ciclo de execução do programa principal, se um determinado evento ocorreu, diminuindo o esforço computacional do dispositivo o que permite que se tenham execuções mais velozes dos códigos implementados.

Existem diversos tipos de interrupções suportadas pelo dsPIC33FJ256GP710A. Elas podem ser interrupções externas, as quais são originadas por excitações em seus pinos de entrada advindas, por exemplo, de sensores; podem também ser internas ao microcontrolador, caso das interrupções originadas do estouro de contadores internos, ocorrência de erros de processamento, dentre uma infinidade de acontecimentos.

Um exemplo do uso das interrupções em um projeto com microcontroladores pode ser um sistema de segurança. Supondo que um sensor piroelétrico de presença seja conectado a uma porta do microcontrolador e esta esteja configurada para gerar uma interrupção quando receber um pulso de corrente. Assim, quando ocorrer a detecção de algum movimento, uma sirene pode ser disparada. O microcontrolador não precisa então checar a todo ciclo de execução do programa principal se o sensor detectou alguma presença indesejada. Quando ocorrer alguma movimentação, o microcontrolador perceberá a ativação da interrupção e então a lógica implementada para o disparo da sirene, que foi implementada em uma sub-rotina, será então executada.

Dessa forma, para a atual aplicação, é interessante utilizar a interrupção gerada quando se tem um dado disponível no *buffer* de entrada da porta serial. Implementando o controlador digital dentro da sub-rotina desta interrupção, tem-se que o controlador somente atuará quando o Simulink enviar um dado para o dsPIC e este estiver pronto para ser usado, evitando assim uma possível perda de sincronia da informação que é calculada e transmitida. A interrupção utilizada foi configurada como mostra a Figura 37.

Figura 37 - Configuração da Interrupção



Fonte: Produção do próprio autor.

Como pode ser visto na Figura 37, a configuração da interrupção é a mais simples possível, bastando simplesmente escolher dentre as inúmeras opções disponíveis e marcar a sua caixa de seleção. Para esta aplicação utilizou-se a interrupção *RS232 UART2 receive data available*.

5.2.2 O Código

Após o término de todas as configurações o compilador gera um código base, em linguagem C, onde as configurações realizadas anteriormente já se encontram escritas, necessitando somente executar a programação para a aplicação desejada.

O código implementado para esta aplicação consiste basicamente da recepção e transmissão de dados pela porta serial e do cálculo de um controlador PI. Para a implementação do código do controlador em linguagem C, foi utilizado o mesmo procedimento para a construção de um controlador PID, porém, suprimindo a parte derivativa no momento em que se assume que o seu ganho k_d é igual a zero, assim como realizado no Simulink.

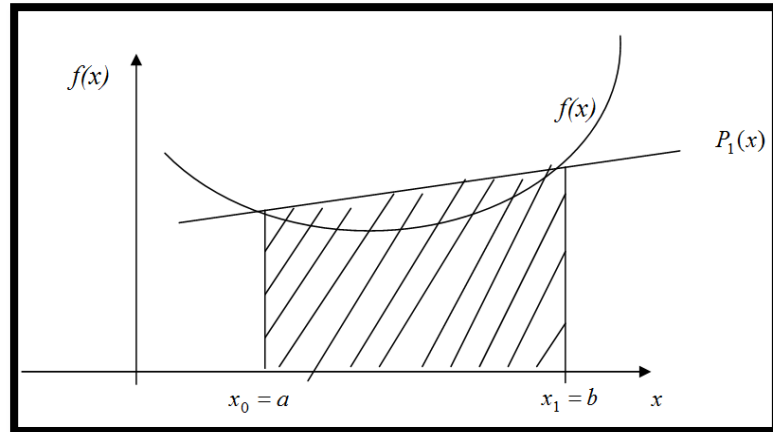
A resolução da equação do controlador PID contínuo é muito complexa e quando há a necessidade de executar a operação em tempo real, tem-se um grande esforço computacional. Os controladores comerciais espalhados pela indústria não contam com toda a capacidade de processamento necessária para a execução destas operações em tempo real, pelo fato dos altos custos agregados à sua produção. Logo, é feito uso de aproximações nos termos integrais e derivativos da equação, com a finalidade de diminuir o esforço de processamento requerido, o que possibilita embarcar a lógica em dispositivos como os microcontroladores.

Existem muitas formas de realizar as aproximações dos termos da equação do PID. Como este projeto não faz uso do termo derivativo, o método de aproximação de termos com essa natureza é suprimido deste texto. Para realizar a aproximação do termo integral da equação do PID foi utilizado o método da aproximação pela transformação de Tustin.

5.2.2.1 Aproximação Integral Pela Transformação de Tustin

A Transformação de Tustin consiste em aproximar o termo integral da equação pela regra trapezoidal (20). Nesta regra, a função a ser integrada $f(x)$ é aproximada por um polinômio interpolador de ordem 1 $P_1(x)$. Assim, necessita-se de dois pontos para interpolação: $[x_0, x_1] = [a, b]$. Graficamente, tem-se a Figura 38.

Figura 38 - Aproximação de Função Por Trapézio



Fonte: Integração Numérica (20).

Para a o cálculo da área hachurada A, tem-se:

$$A = \int_a^b f(x)dx \approx \int_a^b P_1(x)dx \quad (5)$$

Escrevendo a função $P_1(x)$ como série de Taylor até o primeiro termo, tem-se:

$$P_1(x) = f(x_0) + \frac{(x - x_0)}{h} \Delta f(x_0) \quad (6)$$

O que resulta em:

$$A = \int_a^b f(x)dx \approx \int_a^b P_1(x)dx = \int_{x_0}^{x_1} \left[f(x_0) + \frac{(x - x_0)}{h} \Delta f(x_0) \right] dx \quad (7)$$

Para facilitar a integração, é feita uma mudança de variáveis:

$$y_0 = f(x_0) \quad (8)$$

$$z = \frac{(x - x_0)}{h} \quad (9)$$

$$\Delta y_0 = \Delta f(x_0) = f(x_1) - f(x_0) \quad (10)$$

Resultando em:

$$A \approx \int_a^b [y_0 + z\Delta y_0] dx \quad (11)$$

Derivando z em relação à x e mudando-se os limites de integração, tem-se:

$$\frac{dz}{dx} = \frac{1}{h} \rightarrow dx = h dz \quad (12)$$

$$a = x_0 \rightarrow z = \frac{(x_0 - x_0)}{h} = 0 \quad (13)$$

$$b = x_1 \rightarrow z = \frac{(x_1 - x_0)}{h} = \frac{h}{h} = 1 \quad (14)$$

Resultando em:

$$A \approx \int_0^1 [y_0 + z\Delta y_0] h dz \quad (15)$$

Integrando a Equação (15), tem-se:

$$A \approx \int_0^1 [y_0 + z\Delta y_0] h dz = h \left[zy_0 + \frac{z^2}{2} \Delta y_0 \right]_0^1 = h \left[y_0 + \frac{\Delta y_0}{2} \right] \quad (16)$$

O que nos leva a ter finalmente:

$$A = \int_a^b f(x) dx \approx h \left[f(x_0) + \frac{f(x_1) - f(x_0)}{2} \right] = \frac{h}{2} [f(x_0) + f(x_1)] \quad (17)$$

Que é a área do trapézio sob a curva $P_1(x)$.

5.2.2.2 A Equação Final do Controlador

Para a integral do erro do sistema, pela Transformação de Tustin e definindo $u(kT)$ como a área acumulada dos trapézios até o instante kT , tem-se:

$$\int_0^{kT} e(t)dt = \int_0^{kT-T} e(t)dt + \int_{kT-T}^{kT} e(t)dt \quad (18)$$

Que resulta em:

$$u(kT) = u(kT - T) + \frac{e(kT - T) + e(kT)}{2}T \quad (19)$$

Aplicando a aproximação da integral a equação do PI contínuo, é possível chegar à equação do PI digital embarcado no microcontrolador dada por (21):

$$PI(t) = Kp.e(kt) + ui(kT - T) + Ki \frac{e(kT - T) + e(kt)}{2}T \quad (20)$$

Onde:

Kp : ganho proporcional do sistema;

Ki : ganho integral do sistema;

T : período de amostragem do sistema;

$e(kT)$: erro atual do sistema;

$e(kT - T)$: erro no instante anterior ao erro atual do sistema;

$ui(kT - T)$: valor da ação integral no instante anterior ao instante atual do sistema;

$PI(t)$: saída do controlador do sistema transmitido para o Simulink.

Para o controlador digital implementado foram utilizadas as constantes descritas no Quadro 6.

Quadro 6 - Constantes do Controlador PI Digital

Constante	Valor
k_p	18
k_i	150

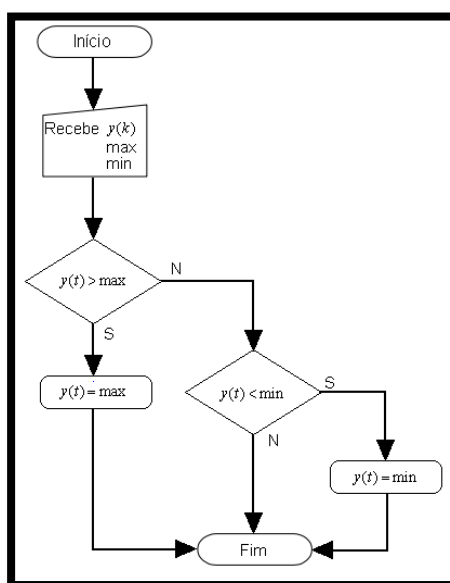
Fonte: Produção do próprio autor.

5.2.2.3 O Sistema Anti Windup

Por fim, na programação do microcontrolador, foi adicionada uma rotina para a prevenção do efeito *windup*, caracterizando uma saturação do sistema.

Este efeito é muito comum em sistemas que apresentam uma componente integral. A saturação proveniente do sinal de controle ocorre devido à oscilação da variável entre os seus valores máximos e mínimos no atuador. Em um curto intervalo de tempo pode ocorrer uma primeira saturação do sistema, no caso desta aplicação, ocorrendo um máximo erro na entrada do controlador PI, decrescendo logo após até o seu valor mínimo, sendo somada durante todo este processo oscilatório o termo integral. Assim, o sistema anti *windup* age de forma a minimizar a ação do termo integral impondo limites de saturação do sistema (22). Considerando a resposta de um controlador PI digital como $y(k)$, tem-se o fluxograma da Figura 39 que descreve a lógica do sistema anti *windup* programado no microcontrolador.

Figura 39 - Fluxograma do Anti Windup



Fonte: Sistema de Auto-Sintonia Para Controlador PID Aplicado a Plantas Térmicas (22).

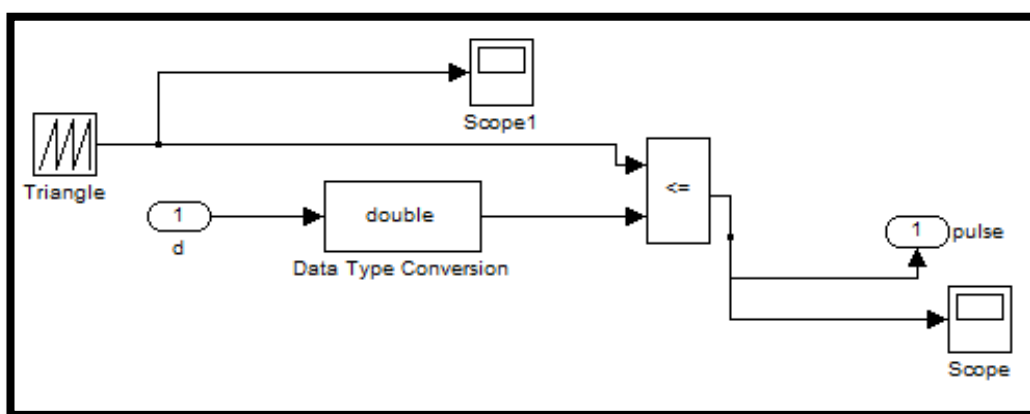
Assim, esta subseção finaliza a lógica implementada em linguagem C que rege o comportamento do microcontrolador no sistema. O código pode ser encontrado no Apêndice A.

5.3 O Sinal PWM

Por fim, o gerador do sinal PWM foi alterado do sistema original para o funcionamento da simulação *hardware-in-the-loop*.

Dentro do bloco de PWM do sistema anterior, foram realizadas alterações com relação à adequação dos tipos de variáveis, bem como da característica da onda dente-de-serra. A Figura 40 mostra a estrutura interna do bloco de PWM usada nesta simulação.

Figura 40 - Estrutura Interna do Bloco de Geração do PWM



Fonte: Produção do próprio autor.

Para o sistema atual, houve a necessidade de transformação do tipo de variável da razão cíclica que chega à simulação via transmissão serial, pois para realizar-se a sua comparação com a onda dente-de-serra, ambos os sinais devem possuir o mesmo tipo de variável. Assim, a razão cíclica que antes era uma variável do tipo inteira de oito bits, passou a ser do tipo *double*.

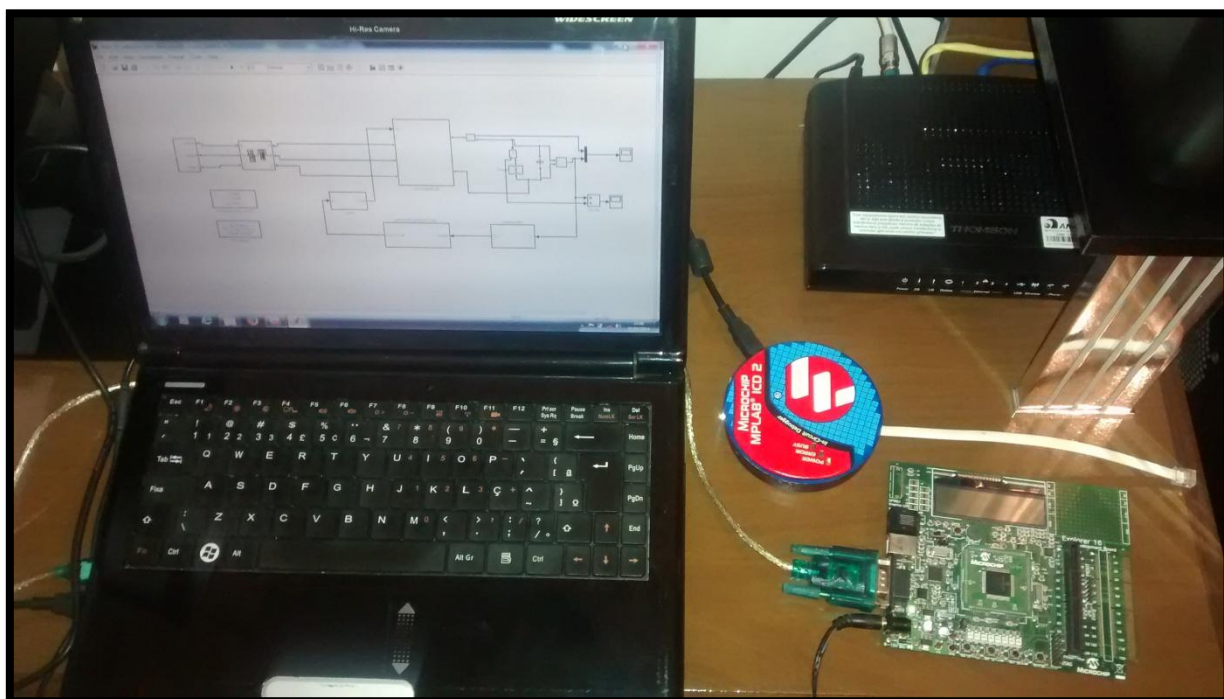
Já a onda dente-de-serra teve alterado o seu valor de amplitude, que passou de 1 para 100. Este artifício foi utilizado para que fosse possível realizar a comparação desta onda com um valor que pudesse ser escrito no formato inteiro de oito bits, que é a resolução da mensagem transmitida serialmente. Antes, a onda era comparada com um valor de ponto flutuante

representado por uma variável *float* de 32 bits. Para que fosse possível transmitir este tipo de variável pela porta serial, seria necessário criar um *array* de bytes na saída do controlador e na recepção do sistema simulado, o que demanda tempo e esforço computacional. Sendo assim, esta forma de simulação trata-se de um artifício válido para fins de simulação.

5.4 Simulação do Sistema Final

A montagem completa do sistema *hardware-in-the-loop* simulado pode ser visto na Figura 41.

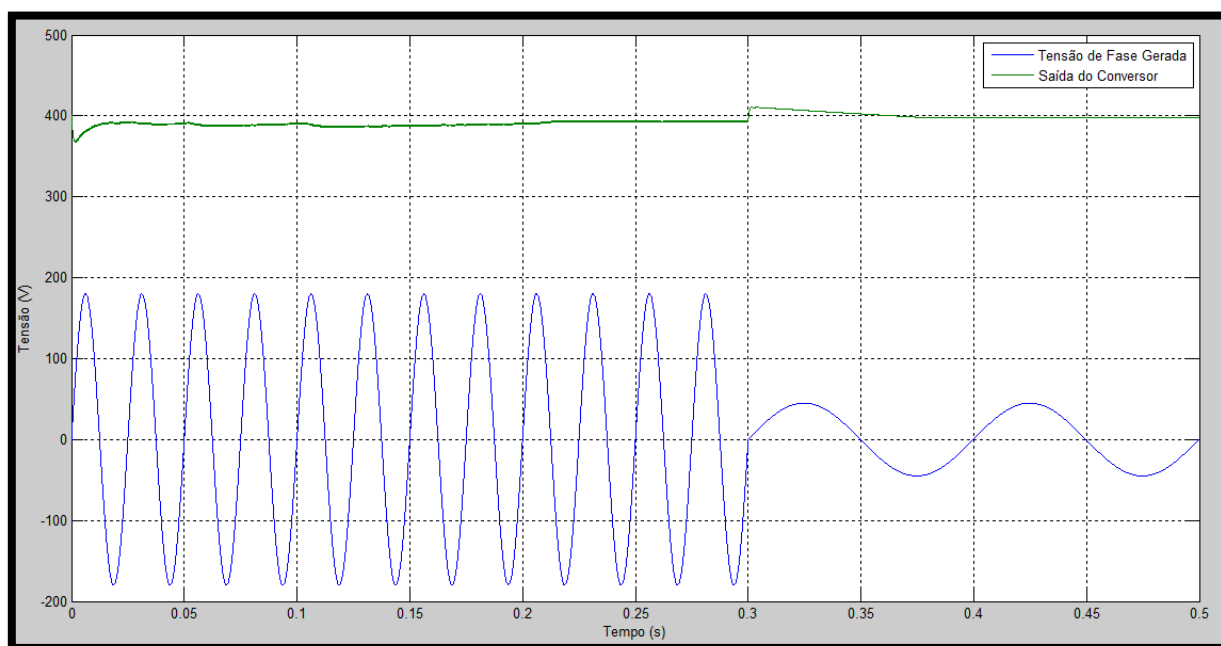
Figura 41 - Montagem do sistema Hardware-In-The-Loop



Fonte: Produção do próprio autor.

Assim como na simulação do sistema puramente computacional, a operação nominal deste sistema de geração se dá com uma velocidade de vento de 12m/s, gerando tensões com uma frequência de 40Hz. O tempo de simulação permanece em 0,5 segundos e depois de transcorridos 0,3 segundos de simulação, a velocidade do vento cai para 3m/s gerando assim tensões com uma frequência de 10Hz. Considerando os parâmetros do controlador digital descritos no Quadro 6, chegou-se à resposta obtida na Figura 42.

Figura 42 - Resposta da Simulação HIL e Tensão de Fase Gerada



Fonte: Produção do próprio autor.

Assim, com base na análise da resposta obtida na curva da Figura 41, é obtido o Quadro 7.

Quadro 7 - Resultados Obtidos da Simulação HIL

Frequência	Parâmetro	Valor
40Hz	Tempo de Acomodação [s]	0,218
	Tensão Final [V]	393
10Hz	Tempo de Acomodação [s]	0,077
	Valor de Pico [V]	410
	Tensão Final [V]	398

Fonte: Produção do próprio autor.

Analisando o Quadro 7, é possível chegar a algumas conclusões, assim como para a simulação puramente computacional. O erro estacionário apresentando pelo sistema antes da mudança de velocidade do vento é dado por:

$$Erro_{40Hz} [\%] = \left(1 - \frac{393}{400}\right) 100$$

$$Erro_{40Hz} = 1,75 \%$$

O sobressinal apresentado pelo sistema quando ocorre a transição de velocidade do vento é:

$$Sobressinal [\%] = \left(\frac{410}{400} - 1 \right) 100$$

$$Sobressinal = 2,50 \%$$

Para a simulação da velocidade do vento de 3m/s, foi obtido um erro estacionário diferente de zero, como se segue:

$$Erro_{10Hz} [\%] = \left(1 - \frac{398}{400} \right) 100$$

$$Erro_{10Hz} = 0,50 \%$$

5.5 Discussão dos Resultados Obtidos

Depois de decorrido todo o processo de simulação tanto do sistema computacional como para a simulação HIL, é possível analisar as respostas obtidas e tirar algumas conclusões.

Quando comparada com a simulação puramente computacional, a simulação HIL, para a questão de tempos de acomodação, fornece uma resposta mais lenta, uma vez que o tempo gasto pelo sistema para atingir o regime estacionário é maior que o anterior. Para a condição de operação anterior a transição de velocidade, a diferença entre os tempos de acomodação já se mostrou mais severa, girando em torno de 120ms de diferença entre os tempos da simulação puramente computacional e da HIL.

Tais diferenças podem ser explicadas pelo fato da inserção do hardware no meio do sistema. Apesar de o sistema microcontrolado contar com um poder de processamento elevado, os dispositivos físicos estão sujeitos a ruídos, oscilações de tensão, dentre outros distúrbios possíveis.

Especificamente para este projeto, uma fonte de distúrbios é a comunicação serial, que vem a ser um tipo mais lento de comunicação, realizando o controle de um sistema de tensão simulado, que vem a ser um sistema rápido por natureza. Nesta comunicação, por algumas

vezes podem ocorrer algumas perdas de pacotes de informação na hora da aquisição dos dados que são tratados no controlador, podendo gerar algumas oscilações e um erro associado.

O fato da estratégia de trabalho de abrir mão de tratar a aquisição e envio de dados de ponto flutuante, que possuem uma resolução de 32 bits muito superior aos 8 bits de dados da comunicação serial, também é um agente que interfere no sistema, tendo em vista que as casas decimais das variáveis transmitidas são rejeitadas.

Aliada às aproximações já feitas para que não fosse gasto um esforço computacional com o tratamento de dados com uma resolução superior a da comunicação escolhida, está a aproximação realizada para a síntese do controlador digital no microcontrolador. Uma vez que quando há a necessidade de trabalhar com dados que possuem uma resolução superior a da comunicação escolhida, a informação deve ser dividida em blocos e transmitida. No destino, por sua vez, a mensagem deve ser remontada e só então estará disponível para o uso. Quando comparado a um controlador executado em um software como o MATLAB/SIMULINK, que conta com um poder de processamento de informação muito mais elevado que um microcontrolador, é aceitável e normal que as respostas obtidas por ambos sejam distintas e que as do controlador sejam mais vulneráveis a erros. Dessa forma, para corrigir os erros inseridos no sistema, foi necessário obter ganhos mais elevados para o controlador digital do que para o controlador que era simulado anteriormente. Ganhos mais elevados tornam o sistema mais oscilatório e o faz ter sobressinais mais elevados. Mas, como foi possível verificar, o sobressinal apresentado pelo sistema na presença do controlador digital foi bem próximo ao do sistema puramente simulado, tendo sido 0,75% maior apenas.

Assim, mesmo com um cenário de recursivas aproximações, os resultados não apresentaram um comprometimento da sua aceitação. Ao analisar que o erro estacionário para uma velocidade do vento de 12m/s foi de 1,75%, o mesmo para os dois sistemas colocados em simulação, e que o erro estacionário para a simulação HIL, tendo uma velocidade do vento de 3m/s é de apenas 0,5% quando comparado com a referência, dá a dimensão da grande precisão do sistema implementado.

6 CONCLUSÃO GERAL E TRABALHOS FUTUROS

6.1 Conclusão Geral

Neste trabalho foram abordados temas que são de grande relevância na atualidade no que diz respeito à geração de energia limpa e estratégias de projetos que visam diminuir custos e impactos ambientais.

Especificamente, dentro do caráter de pesquisa e implementação deste desenvolvimento, foi proposta a realização de uma simulação *hardware-in-the-loop* de um sistema de geração eólica no qual um microcontrolador seria o controlador de um conversor CA-CC e a posterior validação de sua aplicação comparada a um sistema simulado de forma puramente computacional. Conseguiu-se então conhecer mais sobre a técnica *hardware-in-the-loop*, e chegou-se a resultados satisfatórios quando comparados aos de um sistema puramente simulado, o que valida a aplicação desta técnica no caso estudado.

6.2 Trabalhos Futuros

Visando explorar cada vez mais a utilização de sistemas microcontrolados para aplicações industriais, é de grande importância que, futuramente, este controlador possa ser implantado em um sistema real a fim de validar a sua eficácia dentro de uma malha de controle.

É interessante que antes de colocar este controlador em funcionamento dentro de um sistema de grande porte possa ser medido o tempo de execução de todo o código que é executado. Isso levará a uma sintonia mais fina das constantes deste controlador. É também interessante que seja implementado um conversor eletrônico de pequeno porte, e feito junto a ele um sistema de controle. Neste controle é conveniente que seja explorado o PWM disponível no sistema microprocessado, situação não explorada na realização da simulação *hardware-in-the-loop*. Outra funcionalidade muito importante que será explorada nesta aplicação futura é a utilização de conversores AD para realizar a leitura das medições dos sensores presentes no sistema. Assim, um teste deste controlador em um sistema físico de dimensões reduzidas contribuirá ainda mais para que sejam obtidos resultados mais precisos quando este controlador for implementado em uma planta real e complexa.

BIBLIOGRAFIA

- (1) BBC BRASIL. **China Dá Maior Impulso à Energia Eólica Já Visto no Mundo.**
Disponível em:
< http://www.bbc.co.uk/portuguese/noticias/2014/01/140108_china_eolica_mdb.shtml >.
Acesso em 6 maio 2014.

- (2) PORTAL BRASIL. **Entenda Como Funciona o Mercado de Crédito de Carbono.**
Disponível em: < <http://www.brasil.gov.br/meio-ambiente/2012/04/entenda-como-funciona-o-mercado-de-credito-de-carbono> >. Acesso em 6 maio 2014.

- (3) BARROS, O. Jr. **Bancada Digital Para Programação de Microcontroladores.** 2013.
Trabalho de Conclusão de Curso (Graduação em Engenharia Elétrica) – Colegiado do Curso de Engenharia Elétrica, Vitória, 2013.

- (4) JORNAL O EXPRESSO. **Situação da Geração de Energia Não é Boa. E Estação da Seca Vem Aí.** Disponível em:
< <http://jornaloexpresso.wordpress.com/2014/03/05/61283/>>. Acesso em 30 jul. 2014.

- (5) FREITAS, T. R. S. **Alternativas de Topologias Retificadoras Para Geração Eólica Com Geradores Síncronos A Ímã Permanente de Baixa Potência.** 2014. Proposta de Tese de Doutorado (Doutorado em Engenharia Elétrica) – Programa de Pós Graduação em Engenharia Elétrica, Vitória, 2014.

- (6) BARBI, I. **Eletrônica de Potência.** 6 ed. Florianópolis: Edição do Autor. 2006.

- (7) SILVA, B. C. R., CÂNDIDO, L. A. A. **Sistema de Controle Residencial Baseado na Plataforma Arduino.** 2011. Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) – Instituto Unificado de Ensino Superior Objetivo, Goiânia, 2011.

- (8) BASTOS, A. V. **Notas de Aula -Microcontroladores.** Colegiado de Engenharia de Controle e Automação, Universidade Federal de Ouro Preto.

- (9) HALVORSEN, H. P. **Introduction to Hardware-in-the-Loop Simulation.** 1 ed.
Porsgrunn: Edição do Autor.

- (10) FRANÇA, B. W. **Hardware-In-The-Loop Para Desenvolvimento de Software Embarcado em DSPs Utilizando Ambiente PSCAD/EMTDC.** 2009. Trabalho de Conclusão de Curso (Graduação em Engenharia Elétrica com Ênfase em Eletrotécnica) – Departamento do Curso de Engenharia Elétrica, Rio de Janeiro, 2009.

- (11) SILVA, H. M. **Simulação com Hardware-In-The-Loop Aplicada a Veículos Submarinos Semi-Autônomos**. 2008. Dissertação (Mestrado em Engenharia Elétrica) – Programa de Pós Graduação em Engenharia Elétrica, São Paulo, 2008.
- (12) INSTITUTO DE FÍSICA DA UFRGS. **O Que Faz Um Avião Voar?**. Disponível em: < <http://www.if.ufrgs.br/tex/fis01043/20031/Andre/>>. Acesso em 31 jul. 2014.
- (13) EDN CHINA. **dsPIC Controlador Digital de Sinais (DSC) e PIC24H Microcontrolador (MCU)**. Disponível em: < http://bbs.ednchina.com/BLOG_ARTICLE_185059.HTM#>. Acesso em 20 maio 2014.
- (14) MICROCHIP. **dsPIC33FJ256GP710A**. Disponível em: <<https://www.microchip.com/wwwproducts/Devices.aspx?product=dsPIC33FJ256GP710A>>. Acesso em 20 maio 2014.
- (15) MICROCHIP. **dsPIC33FJXXXGPX06A/X08A/X10A Data Sheet**. Disponível em: <<http://ww1.microchip.com/downloads/en/DeviceDoc/70593B.pdf>>. Acesso em 25 jul. 2014.
- (16) MOSAICO. **Guia do Usuário Explorer16BR dsPIC33F**. Disponível em: <[http://www.mosaico.com.br/Midias/Documentacao/Manual%20Explorer%2016%20BR%20\(dsPIC33F\)_rev_05.pdf](http://www.mosaico.com.br/Midias/Documentacao/Manual%20Explorer%2016%20BR%20(dsPIC33F)_rev_05.pdf)>. Acesso em 25 jul. 2014.
- (17) MICROCHIP. **MPLAB ICD2 In-Circuit Debugger User's Guide**. Disponível em: < <http://ww1.microchip.com/downloads/en/devicedoc/51331b.pdf> >. Acesso em 25 jul. 2014.
- (18) PINHEIRO, G. **Notas de Aula – A Interface Serial e o Padrão RS-232**. Departamento de Eletrônica e Telecomunicações, Universidade do Estado do Rio de Janeiro.
- (19) MECATRÔNICA. **Funções Mais Utilizadas da Linguagem C Padrão CCS**. Disponível em: <http://www.mecatronica.org.br/disciplinas/programacao/CARTAO_REFERENCIA.pdf>. Acesso em 18 jul. 2014.
- (20) CAMPAGNOLO, J. M. **Notas de Aula – Integração Numérica**. Departamento de Engenharia Elétrica, Universidade Federal de Santa Catarina.

- (21) MARCELO MACIEL. **Controle PID com Aproximação Digital Para Utilização no PIC**. Disponível em: < <http://www.marcelomaciel.com/2012/05/pid-digital-para-pic.html>>. Acesso em 25 jul. 2014.
- (22) SCHEID, M. A.; DE BRITO, R. M. **Sistema de Auto-Sintonia Para Controlador PID Aplicado a Plantas Térmicas**. 2011. Trabalho de Conclusão de Curso (Graduação em Engenharia Industrial Elétrica) – Colegiado do Curso de Engenharia Industrial Elétrica, Curitiba, 2011.

APÊNDICE A – ROTINA PRINCIPAL DO MICROCONTROLADOR

```
#include <codigo.h>
```

```
// Declaração de Variáveis Globais
```

```
float kp = 18;
```

```
float ki = 150;
```

```
int tensao;
```

```
float setpoint = 200;
```

```
float erro_atual;
```

```
float erro_anterior = 0;
```

```
float acao_int_anterior = 0;
```

```
int PI;
```

```
//ROTINA DE INTERRUPTO
```

```
#INT_RDA2
```

```
void rda2_isr(void)
```

```
{
```

```
    tensao = getc(); //Recebe Tensão do Barramento
```

```
    erro_atual = setpoint - tensao; //calcula o erro
```

```
    // EQUAÇÃO DO CONTROLADOR
```

```
    PI = kp*erro_atual + acao_int_anterior + ki*((erro_anterior + erro_atual)*(1/11520))/2;
```

```
    // Salva valores para o próximo ciclo
```

```
    acao_int_anterior = ki*((erro_anterior + erro_atual)*(1/11520))/2;
```

```
    erro_anterior = erro_atual;
```

```
    // Anti Windup
```

```
    if(PI>48)
```

```
    {
```

```
        PI = 48;
```

```
    }
```

```
    else
```

```
    {
```

```
        if(PI<25)
```

```
        {
```

```
            PI = 25;
```

```
        }
```

```
    }
```

```
    // Envia o Duty Cycle para o Simulink
```

```
    putc(PI);
```

```
}
```

```
void main()
```

```
{
```

```
    enable_interrupts(INT_RDA2);
```

```
enable_interrupts(INTR_GLOBAL);  
  
while(TRUE) //LAÇO INFINITO  
{  
    //TODO: User Code  
}  
  
}
```