

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE ELÉTRICA
TECNOLOGIA EM AUTOMAÇÃO INDUSTRIAL**

BRUNO MASAHARU SHIMADA

**DESENVOLVIMENTO DE UMA BANCADA DIDÁTICA PARA
CONTROLE DE MOTORES CC UTILIZANDO MATLAB/SIMULINK
EM TEMPO REAL**

TRABALHO DE CONCLUSÃO DE CURSO

CORNÉLIO PROCÓPIO

2016

BRUNO MASAHARU SHIMADA

**DESENVOLVIMENTO DE UMA BANCADA DIDÁTICA PARA
CONTROLE DE MOTORES CC UTILIZANDO MATLAB/SIMULINK
EM TEMPO REAL**

Trabalho de Conclusão de Curso apresentado ao Departamento Acadêmico de Elétrica da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do título de “Tecnólogo em Automação Industrial”.

Orientador: Prof. Dr. Marcio Aurelio Furtado Montezuma

CORNÉLIO PROCÓPIO

2016

BRUNO MASAHARU SHIMADA

DESENVOLVIMENTO DE UMA BANCADA DIDÁTICA PARA CONTROLE DE MOTORES CC UTILIZANDO MATLAB/SIMULINK EM TEMPO REAL

Trabalho de conclusão de curso apresentado às 15:50h do dia 21 de Junho de 2016 como requisito parcial para a obtenção do título de Tecnólogo em Automação Industrial da Universidade Tecnológica Federal do Paraná. O candidato foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Prof. Dr. Marcio Aurelio Furtado Montezuma
Professor Orientador
UTFPR / Campus Cornélio Procópio

Prof. Dr. Fabian Andres Lara Molina
Professor Convidado
UTFPR / Campus Cornélio Procópio

Prof. Dr. Marcelo Favoretto Castoldi
Professor Convidado
UTFPR / Campus Cornélio Procópio

A Folha de Aprovação assinada encontra-se na Coordenação do Curso.

Dedico este trabalho à minha avó
Tsuyako (in memoriam).

AGRADECIMENTOS

Agradeço ao Prof. Dr. Marcio Aurelio Furtado Montezuma, orientador deste trabalho, pela disponibilidade oferecida, paciência, incentivo e por todo conhecimento que me transmitiu para que alcançasse meus objetivos.

À minha família, em especial ao meu pai Yukio, minha mãe Lilia e minha irmã Karina, grandes batalhadores, fontes da minha inspiração e determinação que, mesmo estando longe, sempre me apoiaram e ensinaram o significado de respeito e integridade.

À minha noiva Priscila, por todo amor, companheirismo, compreensão e colaboração.

A todos os amigos de república da RepOur, por todos estes anos compartilhando momentos de descontração, companheirismo e lealdade.

Aos amigos do laboratório LaSisC, Lucas Niro, Marcus, Wagner, Takahashi, Carlos Matheus, Fabrizzio e tanto outros que passaram pelo laboratório, pelas trocas de conhecimento que foram fundamentais para minha formação.

Aos amigos de graduação.

A todos os professores da UTFPR, campus Cornélio Procópio, pelo ensino de qualidade e excelência.

“O sucesso nasce do querer, da
determinação e persistência em se chegar
a um objetivo. Mesmo não atingindo o
alvo, quem busca e vence obstáculos, no
mínimo fará coisas admiráveis.”

(José de Alencar)

RESUMO

SHIMADA, Bruno Masaharu. **Desenvolvimento de uma bancada didática para controle de motores CC utilizando MATLAB/Simulink em tempo real**. 2016. 86 f. Trabalho de Conclusão de Curso (Graduação em Tecnologia em Automação Industrial) – Departamento Acadêmico de Elétrica, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2016.

Este trabalho propõe o desenvolvimento de uma bancada didática para o estudo de sistemas controle com baixo custo. O objetivo foi controlar a velocidade de um Motor de Corrente Contínua (CC) utilizando a técnica de controle PID e lógica Fuzzy. Para isto utilizou-se um sistema de aquisição de dados baseado em um microcontrolador da família dsPIC, responsável por acionar o drive de potência do motor CC, realizar a leitura do *encoder* para determinar a velocidade do motor e enviar estes dados em valores digitais para serem manipulados pelo computador através do software *MATLAB/Simulink*. A troca de dados do microcontrolador e o software é realizada através de um protocolo de comunicação via comunicação serial RS-232. No software *MATLAB/Simulink* são implementados os controladores PID e Fuzzy através da técnica *hardware-in-the-loop*. Um gerador, com várias cargas, acoplado ao motor e controlado via software, simulou distúrbios no sistema. O *MATLAB/Simulink* facilita a mudança da técnica de controle sem ter que reprogramar o microcontrolador e faz com que seja possível implementar várias técnicas de controle, tais como clássica, moderna e não convencional, além de fornecer um sistema supervisor para visualização da resposta do sistema e alteração dos parâmetros de controle em tempo real, o que torna essa aplicação interessante para o ensino de técnicas de controle.

Palavras-chaves: Sistema de controle; Bancada didática; Tempo Real; Comunicação Serial.

ABSTRACT

SHIMADA, Bruno Masaharu. **Development of a didactic workbench for control of DC motors using MATLAB/Simulink in Real-time.** 2016. 86 f. Trabalho de Conclusão de Curso (Graduação em Tecnologia em Automação Industrial) – Departamento Acadêmico de Elétrica, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2016.

This work proposes the development of a didactic workbench for control systems study with low cost. Initial goal is to control the speed of a Direct Current Motor (DC) using the PID and Fuzzy Logic control technique. It will use a data acquisition system based on a dsPIC microcontroller family, which will be responsible for activating the motor DC power driver, perform the reading of the encoder to determine the motor speed and send this data into digital values to be handled the computer using the *MATLAB/Simulink* software. Data exchange between microcontroller and *MATLAB/Simulink* will be performed by RS-232 serial communication. In *MATLAB/Simulink* occur closing the loop and implementation of PID and Fuzzy control through hardware-in-the-loop technique. A generator with multiple loads coupled to the motor and controlled via software simulate disturbances in the system. The *MATLAB/Simulink* facilitates change of control technique without having to reprogram the microcontroller and makes it possible to implement various control techniques such as classical, modern and unconventional, in addition to a supervisory system to preview the system response and change control parameters in real-time, which makes this interesting application for teaching control techniques.

Keywords: Control system; Workbench; Real-time; Serial communication.

LISTA DE ILUSTRAÇÕES

Figura 1 – Esquemático de um Motor de corrente contínua de ímã permanente	21
Figura 2 - Diagrama de um motor de corrente	23
Figura 3 - Diagrama de um motor CC controlado por armadura	24
Figura 4 - Circuito de controle de carga	25
Figura 5 - Gráfico "Tensão x Tempo" na carga.....	26
Figura 6 - Controle de carga pela variação de <i>duty cycle</i>	27
Figura 7 - Braço robótico utilizando encoders rotativos	28
Figura 8 - Diagrama básico de funcionamento de um encoder	29
Figura 9 – Diferença de discos de um encoder incremental a direita e absoluto a esquerda ...	30
Figura 10 - Decodificação por quadratura de um encoder incremental	30
Figura 11 - Resposta do filtros Butterworth passa-baixa de ordens de 1 a 5,	36
Figura 12 - Bancada com motor CC utilizada Fonte: Autoria própria	38
Figura 13 – Ilustração dos componentes da bancada	39
Figura 14 - Pinos do dsPIC30F2010	42
Figura 15 - Foto de um FTDI à esquerda e diagrama com descrição dos pinos à direita	46
Figura 16 – Foto de um L298 à esquerda e diagrama com descrição dos pinos à direita	48
Figura 17 - Diagrama de blocos internos do L298	49
Figura 18 - Plataforma HIL (Hardware-In-the-Loop)	50
Figura 19 - Plataforma RCP (Rapid Control Prototyping)	50
Figura 20 - Toolbox RTWT para acesso a sistemas de aquisição (DAQ).....	51
Figura 21 - Diagrama da bancada didática para controle de motor CC	52
Figura 22 - Diagrama com componentes da placa de aquisição.....	53
Figura 23 – Esquemático do circuito principal do dsPIC30F2010 e da fonte reguladora.....	54
Figura 24 - Esquemático do conversor USB / UART para comunicação	54
Figura 25 - Esquemático do <i>drive</i> de potência para acionamento do motor	55
Figura 26 - Placa principal - vista superior e inferior.....	56
Figura 27 - Esquemático do controle de carga	57
Figura 28 - Placa de acionamento do controle de carga	57
Figura 29 - Placa de aquisição e bancada de motor CC	58
Figura 30 - Fluxograma do <i>firmware</i>	59
Figura 31 - Diagrama de troca de dados entre o MATLAB/Simulink e a Placa de aquisição. 62	

Figura 32 – Interface desenvolvida no <i>MATLAB/Simulink</i> para leitura e controle em tempo real da bancada com motor CC	64
Figura 33 - Curva de resposta do Filtro Butterworth	65
Figura 34 - Configuração do bloco <i>Packet Input</i> e da porta serial	65
Figura 35 - Subsistema de tratamento dos dados recebidos	66
Figura 36 - Subsistema de tratamento dos dados enviados	67
Figura 37 - Bancada didática montada	67
Figura 38 - Sistema de controle PID implementado na bancada didática	69
Figura 39 - Entrada degrau de 2500RPM a 2 segundos com controlador PID	70
Figura 40 - Entrada degrau de 2500RPM a 2 segundos e inversão de velocidade com controlador PID	70
Figura 41 - Entrada senoidal de 0,5Hz com inversão de sentido no controlador PID.....	71
Figura 42 - Entrada trapezoidal de 0,5Hz no controlador PID	71
Figura 43 - Resposta do sistema a variação de carga com controlador PID	72
Figura 44 - Base de Regras Fuzzy	73
Figura 45 - Sistema de controle Fuzzy implementado na bancada didática.....	74
Figura 46 - Entrada degrau de 2500RPM a 2 segundos com controlador Fuzzy	75
Figura 47 - Entrada degrau de 2500RPM a 2 segundos e inversão de velocidade com controlador Fuzzy	75
Figura 48 - Entrada senoidal de 0,5Hz com inversão de sentido no controlador Fuzzy	76
Figura 49 - Entrada trapezoidal de 0,5Hz no controlador Fuzzy.....	76
Figura 50 - Resposta do sistema a variação de carga com controlador Fuzzy	77

LISTA DE TABELAS

Tabela 1 – Efeitos individual dos termos do PID na resposta de malha fechada	32
Tabela 2 - Base de regras de MacVicar-Whelan	34
Tabela 3 - Trabalhos relacionados ao controle de motor de corrente continua	37
Tabela 4 - Descrição das portas dsPIC30F2010	42
Tabela 5 - Fonte de interrupções	45
Tabela 6 - Descrição dos pinos FTDI	46
Tabela 7 - Descrição dos pinos L298	48
Tabela 8 – Custos de projeto	77

LISTA DE SIGLAS

A/D	Analógico / Digital
CC	Corrente Contínua
CLF	Controlador de Lógica Fuzzy
DSC	Controlador Digital de Sinal
DSP	Processador Digital de Sinal
E/S	Entradas / Saídas
EEPROM	<i>Electrically Erasable Programmable Read-Only Memory</i>
FTDI	<i>Future Technology Devices International Ltd.</i>
I2C	<i>Inter-Integrated Circuit</i>
LaSisC	Laboratório de Sistemas Automatizados e Controle
MATLAB	<i>Matrix Laboratory</i>
MCU	<i>Microcontroller Unit</i>
MIPS	Milhões de Instruções por Segundo
NG	Negativo Grande
NM	Negativo Médio
NP	Negativo Pequeno
PG	Positivo Grande
PID	Proporcional Integral Derivativo
PM	Positivo Médio
PP	Positivo Pequeno
PWM	<i>Pulse Width Modulation</i>
RAM	<i>Random Access Memory</i>
SPI	<i>Serial Peripheral Interface</i>
UART	<i>Universal Asynchronous Receiver/Transmitter</i>
USB	<i>Universal Serial Bus</i>
Z	Zero
FET	Field Effect Transistor
IGBT	Insulated Gate Bipolar Transistor
SCR	Silicon Controlled Rectifier
TTL	Lógica Transistor-Transistor

HIL	<i>Hardware-In-the-Loop</i>
RCP	<i>Rapid Control Prototyping</i>
ECU	<i>Engine Control Unit</i>
DAQ	<i>Data Acquisition</i>
EAGLE	<i>Easy Applicable Graphical Layout Editor</i>

LISTA DE SÍMBOLOS

K_D	Constante derivativo
K_I	Constante integrativo
K_P	Constante proporcional
T_D	Tempo derivativo
T_S	Período do sinal PWM
T_i	Tempo integrativo
V_O	Valor médio da tensão
V_S	Tensão na carga
q_0, q_1, q_2	Constantes do controlador PID discreto
t_{off}	Tempo do interruptor aberto
t_{on}	Tempo do interruptor fechado
ω_c	Frequência de corte
\emptyset	Fluxo de entreferro
δ	<i>Duty cycle</i>
G	Ganho do filtro
H	Função transferência
K_f	Constante de campo
T_d	Torque perturbador
T_m	Torque do motor
V_b	Tensão contra-eletromotriz
$e(t)$	Sinal de erro em função do tempo
$e(k), e(k - \dots)$	Sinal do erro no tempo discreto
i_f	Corrente de campo
j	Número imaginário
n	Ordem do filtro
$u(k)$	Ação de controle em função do tempo discreto
$u(t)$	Ação de controle em função do tempo
ω	Frequência angular do sinal em radianos por segundo

SUMÁRIO

1 INTRODUÇÃO.....	16
1.1 PROBLEMA.....	17
1.2 JUSTIFICATIVA.....	17
1.3 OBJETIVO.....	18
1.3.1 Objetivos específicos.....	18
1.4 ORGANIZAÇÃO DO TRABALHO.....	19
2 FUNDAMENTAÇÃO TEÓRICA.....	21
2.1 MOTOR DE CORRENTE CONTINUA DE IMÃ PERMANENTE.....	21
2.2 PULSE-WIDTH MODULATION (PWM).....	25
1.3 ENCODER INCREMENTAL.....	28
1.4 CONTROLADOR PID.....	31
2.4 CONTROLADOR DE LÓGICA FUZZY.....	33
2.5 FILTRO BUTTERWORTH.....	35
2.6 TRABALHOS RELACIONADOS.....	36
3 IMPLEMENTAÇÃO DO PROJETO.....	38
3.1 MATERIAIS.....	38
3.1.1 Bancada com motor CC.....	38
3.1.2 Controlador Digital de Sinais dsPIC30F2010.....	39
3.1.2.1 Memórias.....	41
3.1.2.2 Interfaces de entrada e saída (portas).....	41
3.1.2.3 Temporizadores e contadores.....	44
3.1.2.4 Interrupções.....	44
3.1.3 Conversor FTDI FT232R.....	45
3.1.4 Drive de potência L298.....	47
3.1.5 MATLAB/SIMULINK e Real-Time Windows Target.....	49
3.2 DESENVOLVIMENTO.....	51
3.2.1 Elaboração da placa de aquisição.....	52
3.2.2 Desenvolvimento de um Firmware para Placa de Aquisição.....	58
3.2.2 Desenvolvimento da Interface MATLAB/Simulink.....	62
4 RESULTADOS.....	68
4.1 CONTROLE DE VELOCIDADE UTILIZANDO PID.....	68
4.2 CONTROLE DE VELOCIDADE UTILIZANDO FUZZY.....	73
4.3 CUSTO DO PROJETO.....	77
5 CONSIDERAÇÕES FINAIS.....	79
REFERÊNCIAS.....	80
APENDICE A – CÓDIGO DE FONTE DO <i>FIRMWARE</i>.....	83

1 INTRODUÇÃO

O Motor de Corrente Contínua (CC) é um dispositivo atuador de potência que converte energia elétrica de corrente contínua em energia mecânica rotativa (DORF, BIRSHOP, 2001). O motor CC possui como principais características: alto torque na partida e em baixas rotações, ampla faixa de variação de velocidade, além da relativa simplicidade dos seus sistemas de acionamento para controle de posição e velocidade quando comparado à máquinas de corrente alternada. Devido a estas características, sistemas de motores CC ainda são largamente utilizados em numerosas aplicações industriais, principalmente onde necessita-se manter o torque, mesmo com a variação de carga e de velocidade. Sendo assim, os motores CC são a melhor escolha em aplicações como: manipuladores robóticos, máquinas de papel, máquinas-ferramentas, laminadoras, impressoras, entre outros. (FITZGERALD et al, 2006).

A relativa simplicidade de acionamento elétrico de motores CC permite a adaptabilidade a vários métodos de controle, tanto sendo em malha aberta ou fechada, e implementados em controladores analógicos ou em modernos controladores digitais. Com a popularização e o aumento do poder de processamentos de sistemas digitais a baixos custos, o número total de sistemas de controle auxiliados por computador ou sistemas embarcados instalados na indústria aumentou nas últimas três décadas, ocasionando a substituição dos controladores analógicos de motores CC pela implantação de técnicas de controle digital (DORF, BIRSHOP, 2001).

Atualmente, os dispositivos comercializados para controle digital de motores CC, envolvem estruturas dedicadas ao controle do motor para fins industriais, resultando em equipamentos complexos, sofisticados e de elevado custo para fins educacionais. Isto tem dificultado o acesso aos estudantes a introdução e ao estudo de novas tecnologias para melhoria do desempenho e implementação de novas técnicas de controle (PATANÉ, 2008).

Este cenário tem exigido a construção de plantas didáticas, que sejam acessíveis em termos de custo e sejam capazes de demonstrar o funcionamento e a aplicações dos conceitos lecionados em sala de aula para o aluno compreender o que lhe foi ensinado. Neste contexto, laboratórios que permitam realizar testes e implementação de um sistema de controle a fim de validar dados sobre uma planta real, tornam-se interessantes.

Portanto, com as dificuldades encontradas para o estudo de técnicas de controles digitais, o objetivo deste trabalho propôs a construção de uma bancada para implementação do controle de velocidade em malha fechada de um motor de corrente contínua. O objetivo

principal é desenvolver uma solução de baixo custo para que estudantes e pesquisadores universitários possam implementar diferentes técnicas de sistemas de controle digitais em uma planta real.

1.1 PROBLEMA

A maioria dos dispositivos para acionamento de motores CC comercializados atualmente, envolvem estruturas dedicadas ao controle do motor para fins industriais, sendo equipamentos complexos, sofisticados e de elevado custo. Além disso, estes produtos disponíveis no mercado são limitados e pouco flexíveis, ou seja, não permitem o estudo do sistema de controle que está por trás do equipamento. Geralmente, nos casos de controladores PID industriais, somente são permitidas a alterações de constantes dos ganhos do PID, impedindo o estudo mais aprofundado do sistema de controle, pois estes não permitem a visualização das variáveis do sistema e nem a possibilidade de aplicação de outros métodos de controle para o estudo.

Existem algumas fabricantes de kits didáticos para o estudo de sistema de controle. Podemos citar a fabricante Quanser (2015), que possui a venda plantas para estudo de sistema de controle em servomotores, pêndulos invertidos, braços robóticos, entre outros. Porém, por se tratar de uma fabricante internacional, tende-se a tornar alto o investimento para a compra destes equipamentos, dificultando assim o acesso de alunos e profissionais da área à introdução e o estudo de novas tecnologias para melhoria do desempenho e implementação de novas técnicas de controle.

1.2 JUSTIFICATIVA

Com base no problema apresentado, a realização deste Trabalho de Conclusão de Curso torna-se interessante devido aos alunos da graduação tecnológica, profissionais da área e afins, poderem estudar o controle de um motor de corrente contínua através das facilidades do ambiente de programação *MATLAB/Simulink* utilizando para isto uma bancada didática de baixo custo. A estrutura do sistema de controle, as variáveis da planta, os ganhos de controle

serão de fácil alteração e visualização, facilitando assim ao aluno, observar em tempo real, a resposta do sistema a cada modificação no sistema de controle.

O estudo dos sistemas de controle é importante tanto no meio acadêmico, como também no meio industrial, por ser utilizado em sistemas automatizados, consolidando assim os conceitos teóricos apresentados em sala de aula sobre técnicas de controle com a prática, auxiliando o professor na demonstração e aplicação dos conceitos de controle digital dentro de disciplinas que abordem este assunto, de uma forma fácil e com custo acessível.

Outro ponto importante no desenvolvimento deste trabalho é devido as etapas cumpridas para se atingir o objetivo final, já que a construção da bancada envolve vários conceitos multidisciplinares na sua fase de desenvolvimento, tais como: programação em microcontroladores, sistemas de aquisição e controle, desenvolvimento de protocolos de comunicação, desenvolvimento e confecção de hardwares, máquinas elétricas, eletrônica para acionamento de motores, programação em *MATLAB/Simulink* e conceitos de sistemas de controles. Desta forma, este trabalho foi importante para a aprendizagem multidisciplinar durante o desenvolvimento do projeto.

1.3 OBJETIVO

Este trabalho tem como objetivo principal desenvolver uma bancada didática de baixo custo para o estudo de sistema controle de velocidade aplicada a um motor de corrente contínua. Utilizou-se um sistema de aquisição de dados baseado em um microcontrolador dsPIC30F2010 e uma interface gráfica computadorizada para o processamento do algoritmo de controle utilizando o software *MATLAB/Simulink*. A interface gráfica permite a visualização em tempo real das variáveis controle da bancada didática, tais como a velocidade e o sentido de rotação, além de permitir realizar os ajustes de parâmetros do controle.

1.3.1 Objetivos específicos

- Desenvolver uma placa eletrônica baseado em microcontrolador dsPIC com interface USB para aquisição de dados e controle.

- Desenvolver um *firmware* do microcontrolador para comunicação e tratamento de todas as variáveis do processo;
- Utilizar o software *MATLAB/Simulink* como ambiente de desenvolvimento de programas de controle;
- Desenvolver uma interface em *MATLAB/Simulink* que contenha um diagrama de blocos com acesso às todas variáveis de processo da planta;
- Desenvolver o sistema de modo que o tempo de discretização seja pequeno e possa ser executado em tempo real;
- Implementar um controlador PID e um controlador Fuzzy utilizando a bancada desenvolvida para validação de funcionamento de todo o sistema;
- Utilizar metodologias de confecção que tornem a bancada acessível em termos de custo.

1.4 ORGANIZAÇÃO DO TRABALHO

O trabalho foi dividido em 5 capítulos e para melhor compreensão e organização são descritos a seguir:

- **Capítulo 1:** Introdução do tema abordado, os problemas enfrentados, a justificativa deste trabalho e seus objetivos.
- **Capítulo 2:** Apresenta descrição dos conceitos teóricos fundamentais para realização deste trabalho.
- **Capítulo 3:** Inicialmente o capítulo apresenta os principais materiais e equipamentos que foram utilizados para o desenvolvimento do trabalho. Posteriormente, é apresentando o desenvolvimento do trabalho. O tópico desenvolvimento foi dividido entre o desenvolvimento da placa eletrônica e do desenvolvimento da interface em *MATLAB/Simulink*. A discussão é focada em fazer a integração entre o hardware e o software para funcionamento da bancada didática.
- **Capítulo 4:** Apresenta os resultados de vários tipos de entradas no sistema com controlador PID e lógica Fuzzy que foram implementados na bancada didática desenvolvida. O objetivo deste capítulo é validar o funcionamento da bancada didática

e além disto, apresentar uma breve discussão do desempenho dos sistemas de controle implementados.

- **Capítulo 5:** Apresenta as considerações finais sobre o trabalho e também indicações para o desenvolvimento de trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo é realizada de uma forma resumida uma revisão teórica dos fundamentos que foram importantes para o desenvolvimento deste projeto.

2.1 MOTOR DE CORRENTE CONTÍNUA DE IMÃ PERMANENTE

Os motores de corrente contínua (CC) de ímã permanente são, geralmente, a primeira abordagem de máquina CC com a qual os alunos de graduação têm contato. Um motor de corrente contínua é constituído por três partes principais, sendo elas: o rotor, o estator e comutador, como é apresentado na Figura 1.

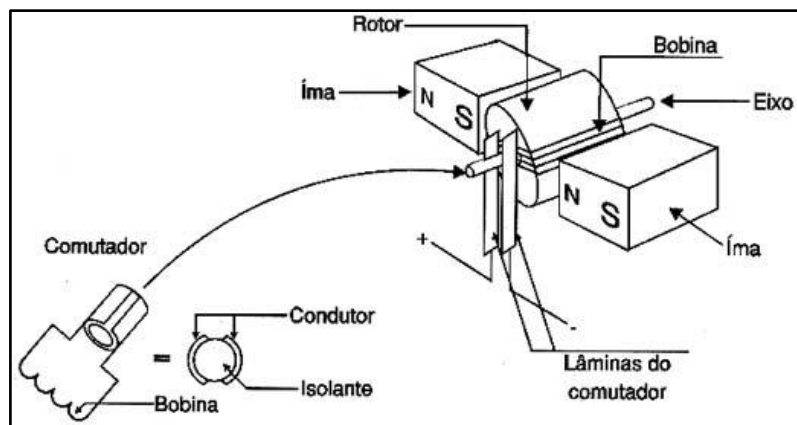


Figura 1 – Esquemático de um Motor de corrente contínua de ímã permanente
Fonte: Adaptado de Braga (2016)

O rotor é a parte girante do motor, montado sobre o eixo da máquina, construído de um material ferromagnético envolto de uma bobina chamado de enrolamento de armadura. O enrolamento de armadura é o principal circuito por transformar a energia proveniente da fonte elétrica em campo magnético, que por sua vez, é responsável pelo movimento mecânico do motor.

O estator, como o próprio nome já diz, é a parte estática do motor, ou seja, é a carcaça montada ao redor do rotor, de forma que o mesmo possa girar internamente. É composto de um material ferromagnético, podendo conter um enrolamento denominado de enrolamento de campo ou um ímã permanente. A função do estator é apenas de produzir um campo magnético

fixo para que haja uma força contrária com o campo magnético da armadura e assim possa produzir movimento do eixo.

Já o comutador, é responsável por conduzir energia ao enrolamento de armadura e assegurar que a corrente que circula pelo enrolamento de armadura flua na direção correta para que se haja o movimento do motor.

Fundamentalmente, o comutador é constituído de um anel segmentado que é fixado e gira conjuntamente no mesmo eixo do rotor, e também, de escovas de contato, que são fixados no estator. O anel é constituído de um material condutor, segmentado por um material isolante, que quando este entra em contato com as escovas durante o movimento, tem a função de alimentar alternadamente o enrolamento de armadura, para que assim gere forças de repulsão ou atração entre o campo magnético do enrolamento de armadura e o campo magnético do ímã permanente, gerando assim um binário de forças e conseqüentemente o giro no eixo do motor.

Os motores de corrente contínua de ímã permanente, geralmente são utilizados em aplicações de baixa potência, por terem como características um tamanho reduzido, construção simples, baixo custo e simplicidade no controle quando comparado a motores de excitação do enrolamento de campo externa. O ímã permanente normalmente é fabricado de materiais magnéticos como samário-cobalto e o neodímio-ferro-boro. (FITZGERALD et al, 2006).

O controle de um motor de ímã permanente, pode ser realizado apenas pela variação de tensão aplicada no enrolamento de armadura, podendo assim, utilizar facilmente sistemas eletrônicos para realizar o controle de velocidade e posição, sejam o controle tanto em sistemas de malha aberta ou fechada, como utilizando componentes analógicos ou digitais através de modernas técnicas de sistemas de controle onde permite-se obter resultados de controle com extrema precisão.

A Figura 2 apresenta o diagrama de fiação de um motor de corrente contínua, onde:

R_a = Resistência de armadura;

L_a = Indutância da bobina de armadura;

i_a = Corrente de armadura;

V_f = Tensão de campo;

R_f = Resistência de campo;

L_f = Indutância da bobina de campo;

i_f = Corrente de campo;

θ = Posição angular;

ω = Velocidade angular.

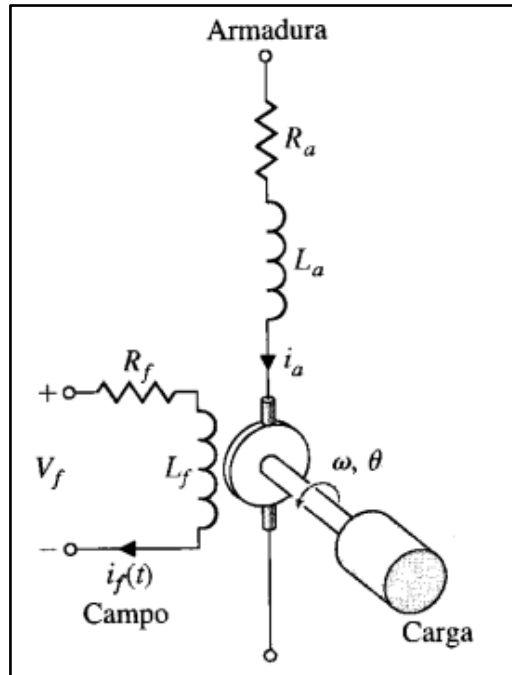


Figura 2 - Diagrama de um motor de corrente Contínua.

Fonte: DORF, BIRSHOP (2001)

Segundo Dorf e Birshop (2001) a função transferência do motor de CC pode ser deduzida por meio de uma aproximação linear do motor real. Desprezando os efeitos como histerese e queda de tensão nas escovas, o fluxo no entreferro do motor é proporcional à corrente de campo, desde que o campo não esteja saturado, ou seja:

$$\Phi = K_f i_f \quad (1)$$

Onde Φ é o fluxo de entreferro, K_f é a constante de campo e i_f é a corrente de campo.

O torque desenvolvido T_m pelo motor é admitido como sendo relacionado linearmente ao fluxo no entreferro Φ e à corrente de armadura i_a , como mostrado na equação:

$$T_m = K_1 \Phi i_a(t) = K_1 K_f i_f(t) i_a(t) \quad (2)$$

A partir da equação (2), nota-se que, para ter um elemento linear, uma das correntes deve ser mantida constante enquanto a outra se torna a corrente de entrada. Para um motor de CC de ímã permanente, a corrente de campo é definida como constante e varia-se a corrente de armadura, em consequência, o torque do motor CC de ímã permanente controlado pela armadura representada por transformada de Laplace é:

$$T_m(s) = (K_1 K_f I_f) I_a(s) = K_m I_a(s) \quad (3)$$

A tensão de armadura pode ser expressa como:

$$V_a(s) = (R_a + L_a s) I_a(s) + V_b(s) \quad (4)$$

$V_b(s)$ é a tensão devida à força contraeletromotriz proporcional à velocidade do motor, sendo:

$$V_b(s) = K_b \omega(s) \quad (5)$$

E a corrente de armadura é:

$$I_a(s) = \frac{V_a(s) - K_b \omega(s)}{(R_a + L_a s)} \quad (6)$$

A Figura 3 apresenta o modelo de em diagrama de blocos do motor CC controlado pela armadura.

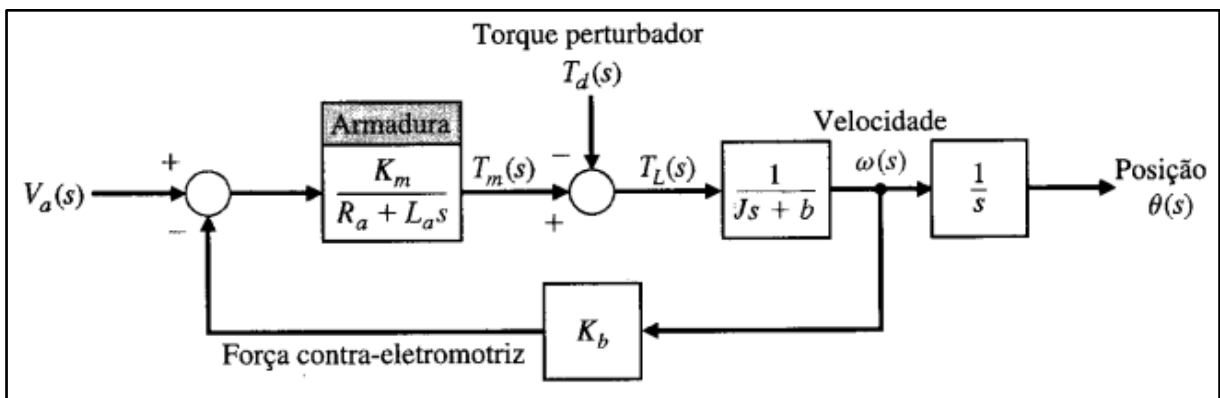


Figura 3 - Diagrama de um motor CC controlado por armadura
Fonte: Dorf e Birshop (2001).

Utilizando o diagrama de blocos e considerando o torque perturbador $T_d(s)$ igual a zero e desprezando o tempo de armadura, obtém-se a função de transferência:

$$G(s) = \frac{\theta(s)}{V_a(s)} = \frac{K_m}{s[R_a(Js + b) + K_b K_m]} = \frac{\left[\frac{K_m}{R_a b + K_b K_m} \right]}{s(\tau_1 s + 1)} \quad (7)$$

Onde a constante de tempo equivalente é:

$$\tau_1 = \frac{R_a J}{R_a b + K_b K_m}. \quad (8)$$

2.2 PULSE-WIDTH MODULATION (PWM)

PWM é a abreviação de *Pulse Width Modulation* ou Modulação de Largura de Pulso. A técnica de PWM é empregada em diversas áreas da eletrônica, e sua utilização é comum em fontes chaveadas, em controle de luminosidade, no controle de cargas resistivas, e é largamente encontrada em aplicações onde é necessário o controle de velocidade de motores, controle de servomotores e diversas outras aplicações. O PWM vem em substituição de técnicas para variação de carga em sistemas que utilizavam para isto, componentes como por exemplo, os resistores variáveis, que por desvantagem, tinham grande perda de potência sobre o próprio resistor de controle.

Para que se entenda o funcionamento de controle de carga através da técnica PWM, considere um circuito como apresentado na Figura 4, formado por um interruptor e uma carga que se deseja controlar sua potência, alimentados por uma fonte de tensão V_E .

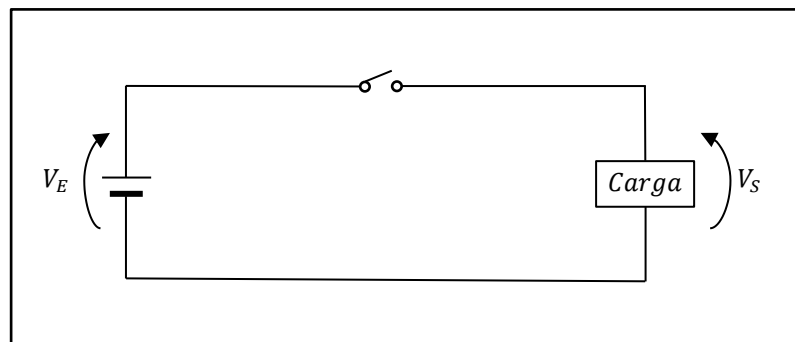


Figura 4 - Circuito de controle de carga
Fonte: Adaptado de Braga (2016)

Observando a figura, percebe-se que, quando o interruptor está aberto não há corrente na carga e a potência aplicada é nula. Ao contrário, no instante em que o interruptor é fechado, a carga recebe a tensão total da fonte e a potência aplicada é máxima (BRAGA, 2016).

Considerando uma variação entre o interruptor aberto e fechado, variando a uma frequência fixa, temos a tensão na carga V_S como observado na Figura 5:

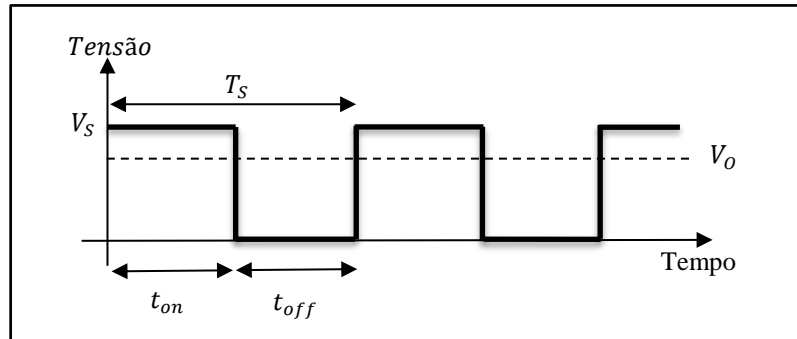


Figura 5 - Gráfico "Tensão x Tempo" na carga
Fonte: Adaptado de Braga (2016)

Onde T_s é o período de alternância da chave, t_{on} é o tempo do interruptor fechado e t_{off} é o tempo do interruptor em aberto, e V_o é o valor médio da tensão. Considerando-se uma alta frequência de chaveamento do interruptor, pode-se considerar em maioria dos casos, a potência média da carga através da tensão média da carga (OLIVEIRA, V. et al, 2013). A tensão média na carga, segundo Aguiar (2004), pode ser expressa pela equação (9).

$$V_o = \frac{1}{T_s} \int_0^{t_{on}} V_S \cdot dt = \frac{1}{T_s} \cdot V_S \cdot t_{on} = V_S \cdot \delta \quad (9)$$

Considerando:

$$\delta = \frac{t_{on}}{T_s} \quad (10)$$

Onde δ é a razão cíclica ou comumente denominado de *duty cycle*, que é caracterizado pela porcentagem em que o interruptor fica fechada, ou seja, que a carga está ativa.

Observando a equação (9), percebe-se que mantendo o período, conseqüentemente a frequência de chaveamento em um valor fixo, porém variando-se o *duty cycle*, podemos controlar a potência média aplicada a uma carga. Assim, quando o *duty cycle* varia de zero até o máximo, a potência também varia na mesma proporção (BRAGA, 2016). A Figura 6 mostra dois exemplos onde o *duty cycle* é de 1% e outro de 99%.

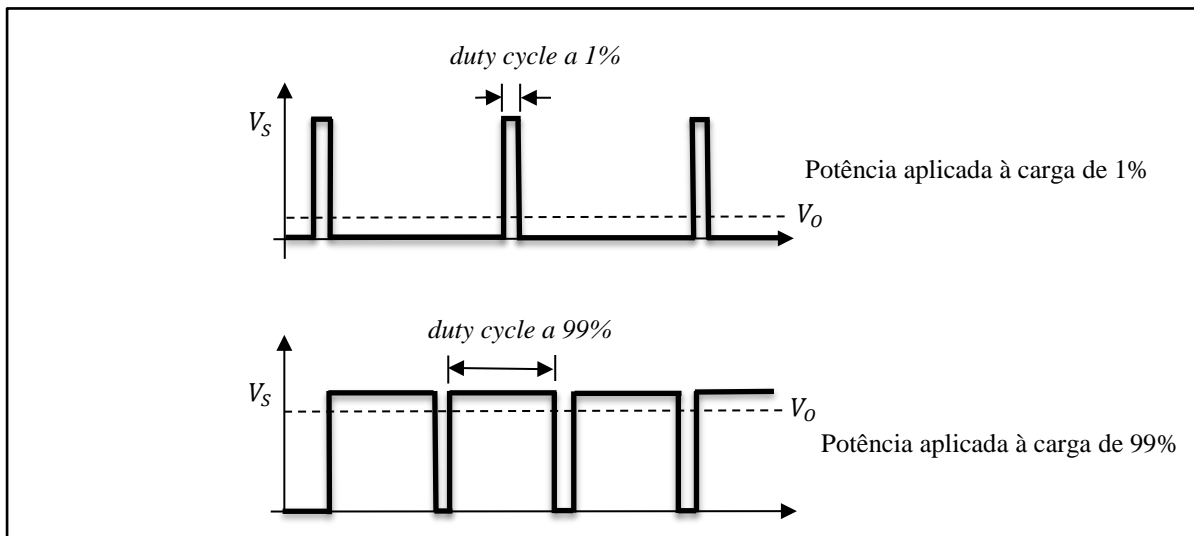


Figura 6 - Controle de carga pela variação de duty cycle
Fonte: Adaptado de Braga (2016)

Na prática, substitui-se o interruptor por um dispositivo de chave de estado sólido que possa abrir e fechar o circuito rapidamente como, por exemplo, um transistor bipolar, um FET de potência, um IGBT ou até mesmo um SCR. Como a potência instantânea dissipada pela chave é o produto da corrente pela tensão elétrica a um dado instante, isso significa que nenhuma potência é dissipada considerando que as chaves de estados sólidos sejam “ideais”, porém, em dispositivos reais, esta potência dissipada é pequena se comparada aos resistores variáveis, o que torna interessante a técnica PWM no controle de motores elétricos e outros tipos de carga (BRAGA, 2016). A baixa potência dissipada permite a utilização de encapsulamento de tamanhos menores e a ausência de grandes dissipadores, principalmente em aplicações onde é necessário o controle de sistemas de grande potência, reduzindo assim custos e o espaço ocupado.

Adicionalmente, uma outra vantagem em se utilizar o PWM no controle de carga, é devido a maioria dos sistemas microcontrolados ou microprocessados, terem suporte nativo à geração de sinal de PWM através de hardware programados por softwares, isto facilita a integração do controle de carga à maioria dos sistemas de controles ou sistemas automatizados utilizados atualmente.

1.3 ENCODER INCREMENTAL

Encoders são dispositivos eletromecânicos que convertem uma movimentação linear ou angular em um sinal digital, que tratado, fornece informações sobre o movimento efetuado, indicando com precisão informações como velocidade, aceleração, distância e posição (DYNAPAR, 2016).

Estes são largamente utilizados em robôs industriais, máquinas ferramentas CNC, servomotores, impressoras, radares, telescópios, antenas, entre outros (MECATRÔNICA ..., 2016).

Existem encoders lineares e rotativos. Encoders lineares, como o próprio nome já diz, obtém dados de movimentos lineares. Já encoders do tipo rotativo, obtém-se dados onde se deseja mensurar movimentos rotativos, como por exemplo de eixos de servomotores, juntas de braços robóticos, como observado na Figura 7.

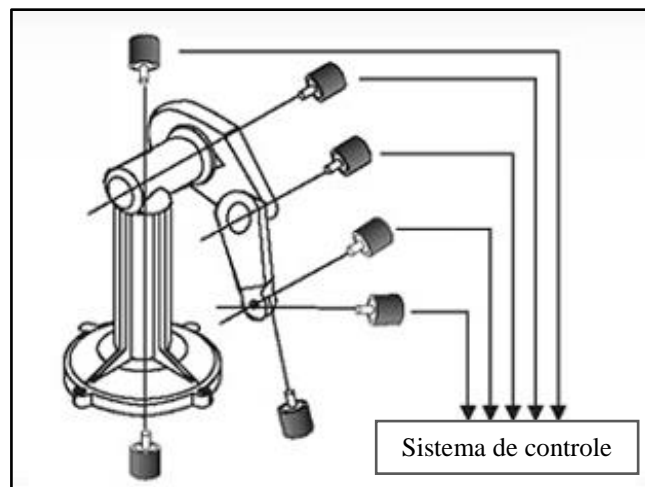


Figura 7 - Braço robótico utilizando encoders rotativos
Fonte: Adaptado de Autonics Sensors & Controllers (2011)

O sistema de leitura de um encoder é baseado em um sinal emitido por um fotoemissor que é projetado a um disco interno contendo janelas transparentes e opacas alternadas ao longo de sua circunferência. As janelas transparentes geram pulsos luminosos em seu lado oposto quando o disco é movimentado, e estes são lidos por um fotorreceptor e enviados em forma de pulsos elétricos à saída do encoder (ETEC, 2012).

A Figura 8 apresenta os componentes básicos de um encoder.

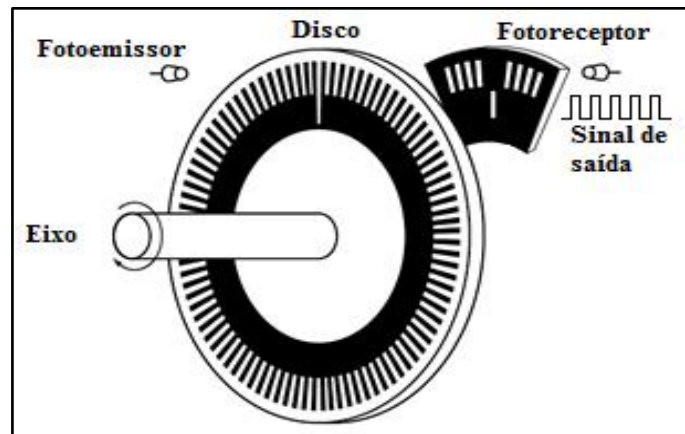


Figura 8 - Diagrama básico de funcionamento de um encoder
Fonte: Adaptado de Tamagawa (2016)

Os encoders podem conter mais de uma trilha de geração de pulsos em um único disco, aumentando assim sua precisão e permitindo conhecer seu sentido de rotação. Para cada trilha, denomina-se à saída do encoder, um canal de saída.

Os encoders rotativos podem fornecer leituras incrementais ou absolutos. Os encoders incrementais geram um pulso para cada unidade de deslocamento, portanto, para saber a posição deve-se definir um ponto de zero e a cada pulso, incrementar ou decrementar da posição atual. Os encoders incrementais podem conter somente um canal, o canal A, fornecendo assim apenas a posição e velocidade de rotação. Mas, também existem encoders incrementais que podem conter dois canais, A e B, defasados a 90° , comumente chamados de encoder de quadratura, que permitem conhecer seu sentido de rotação e também aumentar sua precisão. Existe também encoder com um terceiro canal, Z, que gera um zero absoluto para utiliza-se como uma referência única de posicionamento (ETEC, 2012).

Já os encoders absolutos possuem trilhas em seu disco que, para cada posição lida, é gerado um código único, podendo assim através de uma tabela saber a posição real do encoder. Portanto este não perde referência da posição real em caso de desligamento do sistema, pois não necessita a contagem de posições como no encoder incremental. Porém este sistema possui um custo mais alto em comparação ao incremental e em muitas vezes o incremental satisfaz a necessidade da aplicação.

A Figura 9 apresenta um diagrama de encoder incremental e um absoluto.

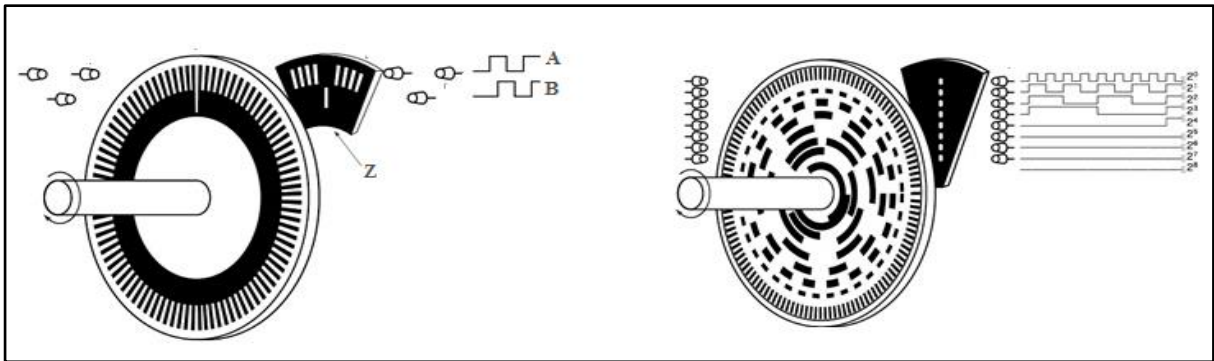


Figura 9 – Diferença de discos de um encoder incremental a direita e absoluto a esquerda
Fonte: Adaptado de Tamagawa (2016)

A resolução dos encoders é definida pelos pulsos por revolução (PPR), ou seja, pela quantidade de pulsos por volta mecânica no eixo. Quanto maior o PPR maior será a precisão obtida na leitura da posição. Como exemplo, um encoder de 1000 PPR, contém mil pulsos em uma volta de 360° , ou seja, a cada pulso equivale a $0,36^\circ$ de deslocamento angular no eixo.

Existe uma técnica digital que possibilita multiplicar os valores de incremento por revolução de um encoder, permitindo assim uma maior precisão na leitura de posição do encoder, e este, denomina-se decodificador por quadratura (MICROCHIP, 2007). Esta técnica consiste em detectar todas as bordas de subida e descida dos canais A e B como um incremento de posição. Portanto, cada pulso por canal contém uma borda de subida e outra de descida, conseqüentemente, multiplica-se em quatro vezes a quantidade de incremento de posição em relação aos PPR's existente em um encoder. Desta maneira, um encoder com 1000 PPR, pode-se ler 4000 posições diferentes em uma revolução, ou seja, $0,09^\circ$ por incremento de posição. A Figura 10 mostra os sinais dos canais A e B e a decodificação por quadratura.

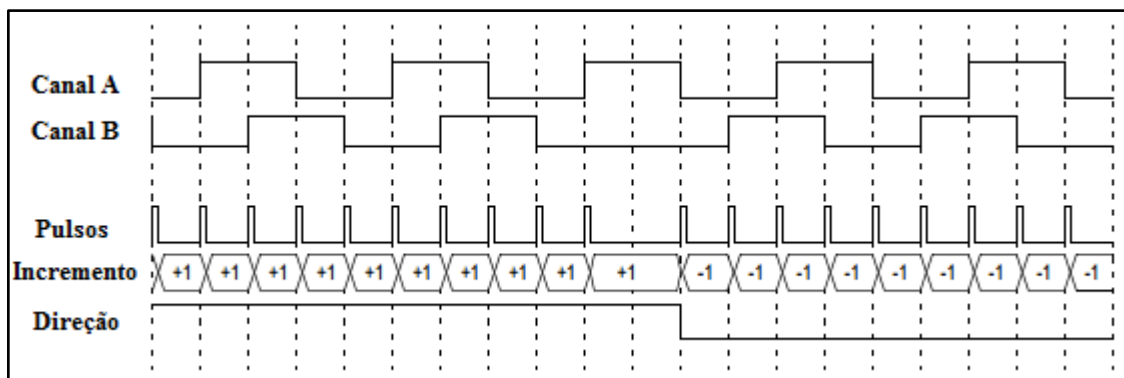


Figura 10 - Decodificação por quadratura de um encoder incremental
Fonte: Microchip (2007)

1.4 CONTROLADOR PID

Uma técnica de controle, segundo Ogata (1998), presente em mais da metade dos controladores industriais, são os controladores que empregam algoritmos do tipo proporcional, integral e derivativo (PID) para controle de variáveis (posição, velocidade, pressão, vazão, etc.). Estes tipos de controladores são os mais populares algoritmos de controle em malha fechada utilizados, tendo em vista que os mesmos são aplicados à maioria dos sistemas de controles de forma satisfatória, mesmo quando se desconheça as propriedades particulares de cada planta.

O funcionamento dos controladores PID se baseia no cálculo inicial do erro entre a variável controlada (medida no processo) e seu valor desejado (*setpoint*). O algoritmo PID usa o erro em três módulos distintos (proporcional, integral e derivativo) para produzir o sinal de saída de forma a estabilizar e manter estável o sistema da melhor forma possível.

Um controlador PID pode ser utilizado para controle de velocidade de um motor de corrente contínua. O controlador PID calcula o valor de tensão média a partir das informações da velocidade desejada (*setpoint*), do valor obtido da realimentação da velocidade e de constantes de proporcionalidade, integral e derivativo (K_p , K_i e K_D respectivamente) para que a velocidade do final motor seja igual ao valor de *setpoint*. É utilizado um sinal PWM com variação do *duty cycle* para obter a tensão média, e assim acionar o motor através de um de circuito de potência.

Segundo Ogata (1998) no caso contínuo, a ação de controle de um controlador PID ideal é representado como:

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(\tau) dt + K_p T_D \frac{de(t)}{dt} \quad (11)$$

onde:

$u(t)$ = Ação de controle em função do tempo

K_p = Constante proporcional

$e(t)$ = Sinal de erro

T_i = Tempo integral

T_D = Tempo derivativo

Considerando

$$K_i = \frac{K_p}{T_i} \quad (12)$$

e

$$K_D = K_p T_D \quad (13)$$

A função transferência do controlador PID é:

$$\frac{u(s)}{e(s)} = C(s) = K_p + \frac{K_I}{s} + K_D s \quad (14)$$

O controlador fornece um termo proporcional, um termo integrativo e um termo derivativo. O efeito individual dos termos na resposta do sistema, são resumidos na Tabela 1.

Tabela 1 – Efeitos individual dos termos do PID na resposta de malha fechada

Ganho	Tempo de subida	Sobressinal	Tempo de resposta	Erro	Estabilidade
$\uparrow K_p$	Diminui	Aumenta	Aumenta pouco	Diminui	Piora
$\uparrow K_I$	Diminui Pouco	Aumenta	Aumenta	Diminui muito	Piora
$\uparrow K_D$	Diminui pouco	Diminui	Diminui	Muda pouco	Melhora

Fonte: Oliveira, V. et al. (2013).

No caso discreto, supondo um tempo de amostragem pequeno, utilizando pela aproximação da integral pela fórmula retangular anterior, segundo Oliveira V. et al (2013) a ação de controle pode ser descrita como:

$$u(k) = u(k-1) + q_0 e(k) + q_1 e(k-1) + q_2 e(k-2) \quad (15)$$

Onde:

$$q_0 = K_p \left(1 + \frac{T_D}{T_s} \right) \quad (16)$$

$$q_1 = -K_p \left(1 + 2 \frac{T_D}{T_S} - \frac{T_S}{T_i} \right) \quad (17)$$

$$q_2 = K_p \frac{T_D}{T_S} \quad (18)$$

Existem métodos para sintonias das constantes do controlador, como por exemplo o método de Ziegler-Nichols, porém no momento este não faz parte do objetivo de trabalho, portanto os valores das constantes serão definidos de forma heurística.

2.4 CONTROLADOR DE LÓGICA FUZZY

Controles de Lógica Fuzzy são sistemas baseados em regras que usa variáveis de linguagem difusas (por exemplo, pequenas, médias e grandes, etc) em abordagens para resolver problemas (PASSINO, YURKOVICH, 1997). O controle executado por lógica fuzzy imita um comportamento em regras em vez de um controle explicitamente restritos a modelos matemáticos, como equações diferenciais. A base de regra é a parte mais importante de um controlador fuzzy. Ele é formado por um regras lógicas, que descreve as relações entre entradas e saídas do controlador. O Controlador fuzzy têm sido aplicado em diversos problemas de controle com sucesso, porque não há modelagem matemática envolvida, porém, neste caso, é necessário o conhecimento de funcionamento da planta.

Desta forma, para a construção de base de regra é baseada em um modelo de base de regra (padrão) utilizando o modelo de base de regra pode ser considerada como uma ferramenta básica, ligando o senso comum de engenharia e experiência com lógica fuzzy.

A base de regras sugerida por MacVicar-Whelan (1976) é um bom exemplo de base de regra modelo. Sendo desenvolvida completando a base de regras introduzida no primeiro controlador fuzzy. A base de regras MacVicar-Whelan (1976) resume as regras usadas na base de regras desses CLF's, e mais situações especiais. A expansão da base de regras original é estabelecida seguindo-se estas três regras fundamentais:

- Regra 1: Se o erro $e(k)$ e a variação $\Delta e(k)$ são zero, então mantém-se o presente controle.
- Regra 2: Se o erro $e(k)$ tende para zero numa taxa satisfatória, então mantém-se o presente controle.
- Regra 3: Se o erro $e(k)$ não está se autocorrigindo, então a ação de controle $\Delta u(k)$ não é zero e depende do sinal e da magnitude de $e(k)$ e $\Delta e(k)$.

O erro é definido como a diferença entre o valor desejado e o valor de saída

$$e(k) = w(k) - y(k) \quad (19)$$

a variação do erro é dada por:

$$\Delta e(k) = e(k) - e(k - 1) \quad (20)$$

e a variação da ação de controle

$$\Delta u(k) = u(k) - u(k - 1) \quad (21)$$

Será utilizada como base de regras a de MacVicar-Whelan apresentada na matriz abaixo:

Tabela 2 - Base de regras de MacVicar-Whelan							
$e(k) \backslash \Delta e(k)$	NG	NM	NP	Z	PP	PM	PG
NG	NG	NG	NG	NG	NM	NP	Z
NM	NG	NG	NM	NM	NP	Z	PP
NP	NG	NM	NP	NP	Z	PP	PM
Z	NG	NM	NP	Z	PP	PM	PG
PP	NM	NP	Z	PP	PP	PM	PG
PM	NP	Z	PP	PM	PM	PG	PG
PG	Z	PP	PM	PG	PG	PG	PG

Fonte: MacVicar-Whelan (1976)

Onde:

- PG = Positivo Grande;
- PM = Positivo Médio;
- PP = Positivo Pequeno;
- Z = Zero;
- NP = Negativo Pequeno;
- NM = Negativo Médio;
- NG = Negativo Grande.

Baseado nestas regras, será implementado um controlador de Lógica Fuzzy com auxílio da biblioteca oferecida pelo *MATLAB/Simulink* para controlar a bancada didática.

2.5 FILTRO BUTTERWORTH

Filtro Butterworth é um tipo de filtro de processamento de sinal projetado para ter a resposta em frequência o mais plano possível na banda de passagem. Também é referido como filtro de máxima magnitude plana. Foi descrito em 1930 pelo físico e engenheiro Stephen Butterworth o seu artigo “*On the Theory of Filter Amplifiers*” (BUTTERWORTH, 1930).

A resposta em frequência de um filtro Butterworth é muito plana (não possui *ripple*, ou ondulações) na banda passante, e se aproxima do zero na banda rejeitada.

Para um filtro de primeira ordem, a resposta varia em -6 dB por oitava. Para um filtro Butterworth de segunda ordem, a resposta em frequência varia em -12 dB por oitava, em um filtro de terceira ordem a variação é de -18 dB , e assim por diante. Os filtros Butterworth possuem uma queda na sua magnitude como uma função linear com ω , que quando observado em um gráfico logarítmico, esta resposta decresce linearmente até o infinito negativo. A Figura 11, apresenta a resposta do filtro para uma frequência de corte de 1 rad/s e várias ordens.

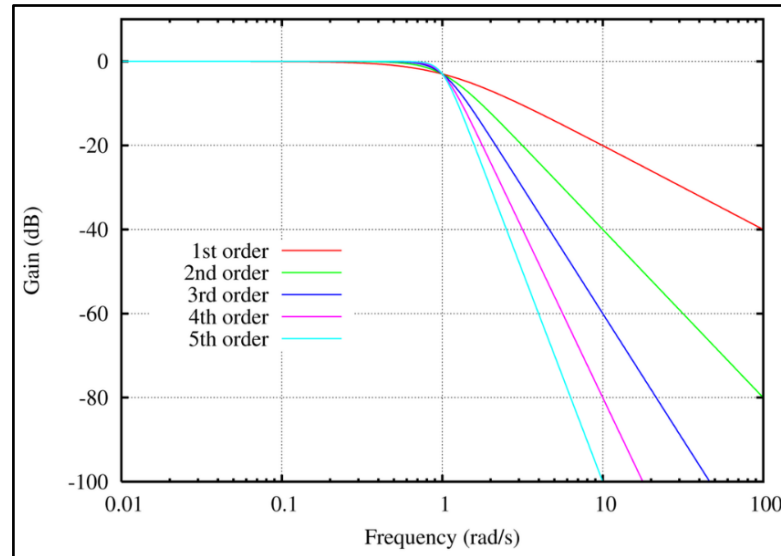


Figura 11 - Resposta dos filtros Butterworth passa-baixa de ordens de 1 a 5, com frequência de corte $\omega = 1$ rad/s.

A forma mais comum do filtro Butterworth é na utilização de filtros passa-baixa, onde a magnitude da resposta em frequência de ordem n pode ser definida matematicamente como mostrado na equação (22).

$$G_n(\omega) = |H_n(j\omega)| = \frac{1}{\sqrt{1 + \left(\frac{\omega}{\omega_c}\right)^{2n}}} \quad (22)$$

onde:

G = ganho do filtro;

H = função de transferência;

j = número imaginário;

n = ordem do filtro;

ω = frequência angular do sinal em radianos por segundo;

ω_c = frequência de corte.

2.6 TRABALHOS RELACIONADOS

É possível citar alguns trabalhos relacionados à motores de continua e que tiveram o objetivo de controlar a velocidade destes motores. Para melhor visualização foi apresentado os

trabalhos desenvolvidos através da Tabela 3, onde foi dividido entre o hardware, o software e o sistema de controle que cada autor utilizou para realizar o controle do motor.

Tabela 3 - Trabalhos relacionados ao controle de motor de corrente continua

Autor	Hardware	Software	Sistema de controle
Patane (2008)	DAQ National Instruments Elvis PCI 6251	LabVIEW	PID
Silva (2008)	Microcontrolador ARM NXP LPC2148	Próprio em C++	PID
Jabo (2008)	DAQ Advantech USB 4761	Próprio em Visual Basic	PID
Guimalhões (2011)	DAQ National Instruments NI AT-MO-16E-10	<i>MATLAB</i> <i>/Simulink</i>	PID Dead-Beat Realimentação de estados
Bertachi (2011)	Microcontrolador PIC18F4550	LabVIEW	PID
Oliveira, C. (2013)	DAQ National Instruments MyDaq	LabVIEW	PID

Fonte: Autoria Própria

Patane (2008), Jabo (2008) e Oliveira C. (2013) utilizaram placas de aquisição de dados que possuem bibliotecas prontas para a aquisição nos softwares por eles utilizados. Já Silva (2008), Bertachi (2011), utilizam-se de microcontroladores para fazer a aquisição de dados, tendo como parte do trabalho fazer a programação de um protocolo de comunicação para comunicação com software.

O presente trabalho, diferentemente dos trabalhos citados, irá utilizar um controlador digital de sinais (DSC) dsPIC30F2010 como hardware, e o *MATLAB/Simulink* como software, além de implementar duas técnicas de controle, o controlar PID e também um controlador Fuzzy.

3 IMPLEMENTAÇÃO DO PROJETO

Neste capítulo é apresentado os principais recursos que foram utilizados para atingir o objetivo final do trabalho, que é a construção de uma bancada didática para controle de motor CC de imã permanente. Este capítulo foi dividido em uma seção de materiais e outra de desenvolvimento.

3.1 MATERIAIS

Nesta seção é apresentado os principais equipamentos e materiais que foram utilizados no desenvolvimento do trabalho.

3.1.1 Bancada com motor CC

O projeto utilizou uma bancada de motor de corrente contínua de imã permanente e esta se encontra no Laboratório de Sistemas Automatizados e Controle (LaSisC) da Universidade Tecnológica Federal do Paraná, Campus Cornélio Procópio. A bancada pode ser vista na Figura 12:



Figura 12 - Bancada com motor CC utilizada
Fonte: Autoria própria

A bancada contém um motor CC de ímã permanente marca Pittman com tensão de trabalho máxima de 19,1V e rotação máxima de 10.000 rpm. Acoplado ao mesmo eixo do motor CC, a bancada contém um encoder incremental de quadratura modelo HEDS-5310, marca Avago Technologies, que possui 512 PPR e 3 canais.

Um gerador CC de ímã permanente é acoplado por meio de uma correia ao motor CC, e é empregado para reproduzir e/ou simular cargas mecânicas e perturbações no eixo do motor CC.

Todos os componentes são montados sobre uma placa de fibra de carbono e suportes emborrachados para garantir estabilidade e rigidez física ao projeto.

A Figura 13 ilustrativa uma imagem da bancada que permite melhor visualização dos componentes apresentados e a disposição entre eles.

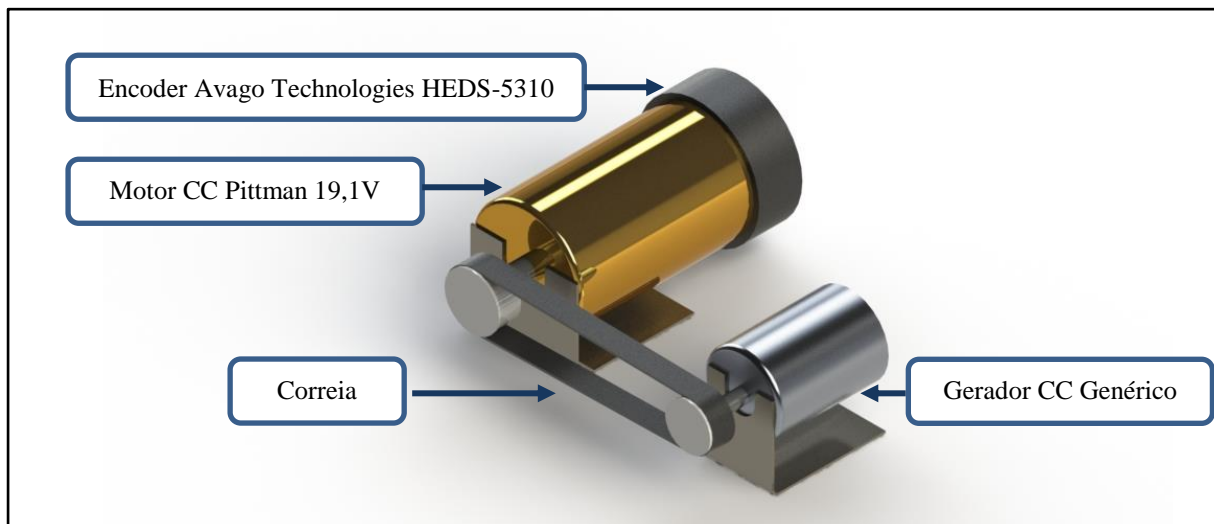


Figura 13 – Ilustração dos componentes da bancada
Fonte: Autoria Própria

3.1.2 Controlador Digital de Sinais dsPIC30F2010

O dsPIC30F2010 é produzido pela Microchip e faz parte da família de Controladores Digitais de Sinais (DSC) da família dsPIC30F. É um dispositivo de 16 bits com arquitetura Harvard modificada, que ao mesmo tempo combina as vantagens de um microcontrolador (MCU) de 16 bits com um Processador Digital de Sinais (DSP) obtendo-se assim uma solução completa para aplicações como controle de motores, conversão de potência, sensores de alta

velocidade, processamento de sinais de áudio e voz, sistema de comunicação, encriptação e aplicações automotivas.

As principais características da família dsPIC30F estão relacionados abaixo:

- MCU de 16 bits;
- Poder de processamento de um DSP, utilizando a arquitetura e instruções dedicadas;
- Velocidade de processamento de até 30MIPS;
- 86% das instruções são executadas em 1 ciclo de máquina;
- Alimentação 2,5 a 5,0V;
- 84 instruções;
- 3 Fontes de interrupção externa;
- 8 níveis de prioridades para cada interrupção;
- 24 fontes de interrupções, com 7 níveis de prioridade;
- Temporizadores de 16 bits, mas podem ser combinados para gerar temporizadores de 32 bits;
- O preço equivalente aos microcontroladores da família 18;
- Conversor A/D de 10 bits a 1Msps ou 12 bits a 200ksps;
- Entrada para leitura de Encoders de quadratura, direto de motores, para controle de posição/velocidade;
- Geração de PWM para motores;
- Comunicação SPI, I²C, UART, DCI, CAN2.0;
- WDT de 2ms a 16s com 1% de precisão;
- Proteção contra falha no cristal, (*Main clock failure*);
- Programação por ICSP (*In-Circuit Serial Programming*).

Os dsPICs estão disponíveis em três famílias, sendo estas as famílias de Controle de motores, sensores e uso geral. O Controlador de sinal digital dsPIC30F2010 está incluso na família de controle de motores, possuindo assim para esta família a leitura de encoder por quadratura, seis saídas de PWM complementares ou independentes e conversor analógico com sincronia ao PWM.

3.1.2.1 Memórias

O Controlador Digital de Sinais possui internamente três tipos de memória internas, sendo elas: memória Flash, memória RAM e memória EEPROM.

A memória Flash, também chamado de memória do programa, é o local onde se armazena o programa a ser executado pelo microcontrolador. Sendo somente de leitura quando em execução do programa, pode ser programado e reprogramado até milhares de vezes por meio de gravadores específicos, como, por exemplo, o PICKIT 2. Devido à facilidade na regravação de programas, esses componentes constituem a melhor opção para testes de programas e implementação final dos projetos. O dsPIC30F2010 possui 12 kBytes de memória Flash.

A memória RAM ou memória de dados é utilizada para armazenamento temporário de informações, que são manipuladas pelo programa automaticamente. Este tipo de memória é uma memória do tipo volátil, ou seja, quando o microcontrolador é desligado, os dados contidos na memória RAM são automaticamente perdidos. O dsPIC30F2010 possui 512 Bytes de memória RAM.

A memória EEPROM também é utilizada para armazenamento de dados temporários, mas diferentemente da memória RAM, é uma memória não volátil, permitindo assim manter as informações mesmo sem alimentação. O DSC estudado possui 1024 Bytes de memória EEPROM.

3.1.2.2 Interfaces de entrada e saída (portas)

O Controlador de Sinais Digitais dsPIC30F2010 possui 28 pinos, sendo todos os pinos (exceto V_{dd}, V_{ss}, MCLR e OSC1/CLKIN) são compartilhados entre os periféricos e as portas paralelas de entrada e saída. Todas as portas de entrada e saída tem a característica de serem *Schmitt Trigger*, isto é, permite uma maior estabilidade evitando ruídos na leitura de dados.

Os pinos de alimentação V_{dd} e V_{ss} suportam correntes máxima de 300mA e 250mA respectivamente. Enquanto os demais pinos suportam 25mA de corrente, tanto fornecendo (pino configurado como saída) quanto drenando a corrente (pino configurado como entrada).

Na Figura 14 é apresentado a localização física de cada pino dsPIC30F2010 com encapsulamento do tipo SDIP e quais funções os pinos podem assumir, já na Tabela 4 são descritas de forma detalhada a descrição de cada função.

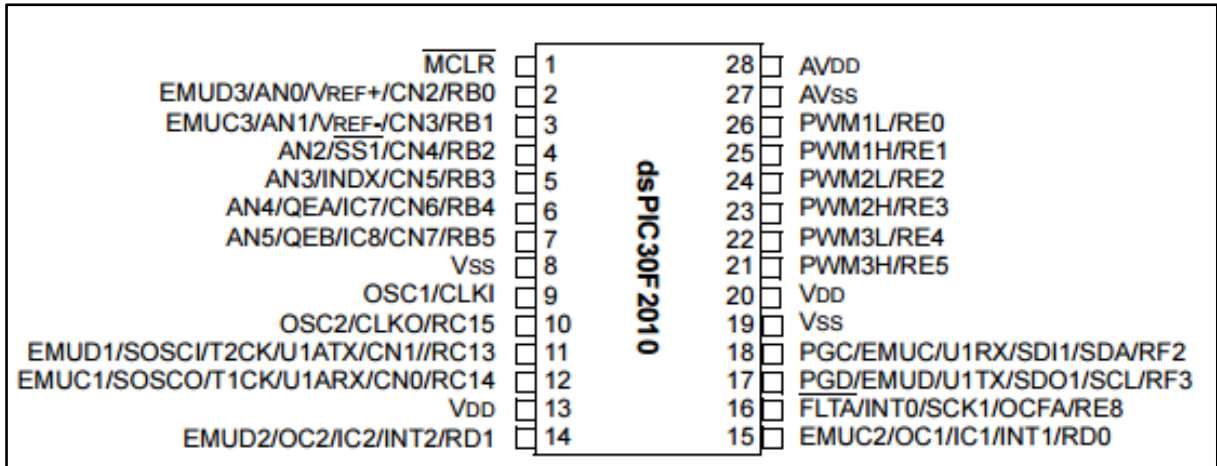


Figura 14 - Pinos do dsPIC30F2010
Fonte: Microchip (2011)

Tabela 4 - Descrição das portas dsPIC30F2010

(continua)

Função	Tipo do Pino	Descrição
AN0-AN5	E	Entrada analógica
AVDD	A	Alimentação positiva do módulo analógico
AVSS	A	Alimentação negativa do módulo analógico
CLKI	E	Entrada de fonte de clock
CLKO	S	Saída de fonte de clock
CN0-CN7	E	Entrada de interrupção de mudança de estado
EMUD	E/S	Primeira entrada de dados do ICD**
EMUC	E/S	Primeira saída de dados do ICD**
EMUD1	E/S	Segunda entrada de dados do ICD**
EMUC1	E/S	Segunda saída de dados do ICD**
EMUD2	E/S	Terceira entrada de dados do ICD**
EMUC2	E/S	Terceira saída de dados do ICD**
EMUD3	E/S	Quarta entrada de dados do ICD**
EMUC3	E/S	Quarta saída de dados do ICD**
IC1,IC2,IC7,IC8	S	Entrada do módulo de captura
INDX	E	Entrada do canal Index do Encoder por Quadratura
QEA	E	Entrada do canal A do Encoder por Quadratura
QEB	E	Entrada do canal B do Encoder por Quadratura
INT0	E	Entrada da interrupção externa 0

Tabela 4 - Descrição das portas dsPIC30F2010**(conclusão)**

Função	Tipo do Pino	Descrição
INT1	E	Entrada da interrupção externa 1
INT2	E	Entrada da interrupção externa 2
FLTA	E	Entrada de Falha no PWM
PWM1L	S	Saída de PWM 1 de baixa
PWM1H	S	Saída de PWM 1 de alta
PWM2L	S	Saída de PWM 2 de baixa
PWM2H	S	Saída de PWM 2 de alta
PWM3L	S	Saída de PWM 3 de baixa
PWM3H	S	Saída de PWM 3 de alta
MCLR	E/A	Master Clear (Reset) ou entrada de tensão de programação
OCFA	E	Entrada de falta A (para Comparação 1,2,3 e 4)
OC1-OC2	S	Saídas de comparação
OSC1	E	Entrada de fonte de clock
OSC2	E/S	Entrada de fonte de clock
PGD	E/S	Pino de dados da gravação ICSP
PGC	E	Pino de clock da gravação ICSP
RB0-RB5	E/S	Portas bidirecionais do PORTB
RC13-RC14	E/S	Portas bidirecionais do PORTC
RD0-RD1	E/S	Portas bidirecionais do PORTD
RE0-RE5,RE8	E/S	Portas bidirecionais do PORTE
RF2,RF3	E/S	Portas bidirecionais do PORTF
SCK1	E/S	Pino de clock para comunicação SPI
SDI1	E	Entrada de dados SPI
SD01	S	Saída de dados SPI
SS1	E	Seleção do escravo SPI
SCL	E/S	Pino de clock para comunicação I2C
DAS	E/S	Pino de dados para comunicação I2C
SOSCO	S	Saída para cristal oscilador de 32kHz
SOSCI	E	Entrada para cristal oscilador de 32kHz
T1CK	E	Entrada de clock externo para o Timer 1
T2CK	E	Entrada de clock extremo para o Timer 2
U1RX	E	Recepção do UART1
U1TX	E	Transmissão do UART1
U1ARX	E	Recepção alternativa da UART1
U1ATX	S	Transmissão alternativa da UART1
VDD	A	Alimentação positiva
VSS	A	Terra da alimentação
VREF+	A	Entrada de referência analógica positiva
VREF-	A	Entrada de referência analógica negativa

Fonte: Microchip (2011)

*E= Entradas S= Saídas A=Alimentação **ICD = In-Circuit Debugger

3.1.2.3 Temporizadores e contadores

Os temporizadores, ou *timers* contam tempo e os contadores contam eventos. Os *timers* variam conforme o ciclo de máquina, enquanto os contadores variam a cada pulso do clock externo. Ambos podem ser lidos e escritos e funcionam apenas de forma incremental. É bem útil onde se deseja contar o tempo, como relógios e eventos e também gerar atrasos no código.

O dsPIC possui três *timers* de 16 bits cada um, o que quer dizer que estes podem contar de 0 a 65535, sendo eles: *timer1*, *timer2* e *timer3*. Os *timers* 2 e 3 podem ser combinados para formarem um *timer* de 32 bits e podem contar até 4294967296.

Os *timers* e os contadores podem gerar interrupções, isto é, quanto atingirem um valor definido pelo programador (PRX), gera uma interrupção.

Uma função dos *timers* denominado de Prescaler é uma forma de aumentar o tempo do através de divisões especificadas do valor do *timer*. Como exemplo, escolhendo um valor de Prescaler igual a 8, tem-se que a cada 8 ciclos de máquina é que ocorrerá 1 incremento no timer. Os valores de Prescaler possível são 1, 8, 64 ou 256.

$$Temp_de_estouro = Ciclo_de_maquina * PRESCALER * PRX \quad (23)$$

3.1.2.4 Interrupções

Segundo Souza (2008): “Interrupções são estímulos externos ou internos que no momento em que ocorre, e dependendo da sua prioridade, devem ser tratadas naquele instante.”

O dsPIC possui 24 interrupções das quais podem ser tratadas com 7 níveis de prioridade, de 1 a 7. Quanto maior o valor numérico da prioridade, maior é a prioridade da interrupção.

Quando uma interrupção é gerada, o programa salva o endereço de memória em que o programa parou e começa a executar o tratamento da interrupção. Após terminada o tratamento da interrupção, o programa volta a executar de onde parou anteriormente.

No dsPIC30F2010, as interrupções são identificadas através dos vetores, como numerados na tabela abaixo:

Vetor de tratamento	Fonte de interrupção
0	INT0 - Interrupção externa 0
1	IC1 - Entrada Captura 1
2	OC1 - Saída Comparação 1
3	T1 - Timer 1
4	IC2 - Entrada Captura 2
5	OC2 - Saída Comparação 2
6	T2 - Timer 2
7	T3 - Timer 3
8	SPI1
9	U1RX - UART1 Recepção
10	U1TX - UART1 Transmissão
11	ADC - ADC Final Conversão
12	NVM - NVM Escrita completa
13	SI2C - I2C Interrupção do escravo
14	MI2C - I2C Interrupção do Master
15	Entrada de interrupção de mudança
16	INT1 - Interrupção Externa 1
17	IC7 - Entrada Capture 7
18	IC8 - Entrada Capture 8
19	INT2 - Interrupção Externa 2
20	INT3 - Interrupção Externa 3
21	PWM - PWM Partida de período
22	QEI - Interrupção QEI
23	FLTA - Falta A de PWM

Fonte: Microchip (2011)

3.1.3 Conversor FTDI FT232R

O FT232R é um circuito integrado com função de conversor USB (*Universal Serial Bus*) – UART, fabricado pela FTDI (*Future Technology Devices International Ltd.*), é utilizado para transmitir dados recebidos da UART pela USB ou vice-versa. É largamente utilizada em conversores USB para RS232, RS422, RS485, em transferência de dados para celular, sistemas

embarcados com microcontroladores, câmeras USB, controle industrial USB, leitores de código de barras USB, entre outras várias aplicações.

O circuito integrado FT232R permite interligar os pinos Rx e Tx das portas de comunicação UART diretamente à uma porta USB dos computadores, necessitando apenas instalar um *driver* no sistema operacional. O FT232R é reconhecido pelo sistema operacional como uma porta serial COM equivalente à comunicação RS232 que utilizavam o conector conectores DB9, que, atualmente está cada vez mais difícil se encontrar em computadores, pois está se tornando obsoleto.

O FT232R possui 28 pinos totais, pode ser alimentado com tensões de 3,3V a 5,25V e permite comunicação com velocidade variando de 300 Bd (baud) até 3 Mbd (Megabaud).

A Figura 15 mostra a disposição dos pinos do circuito integrado FT232R:

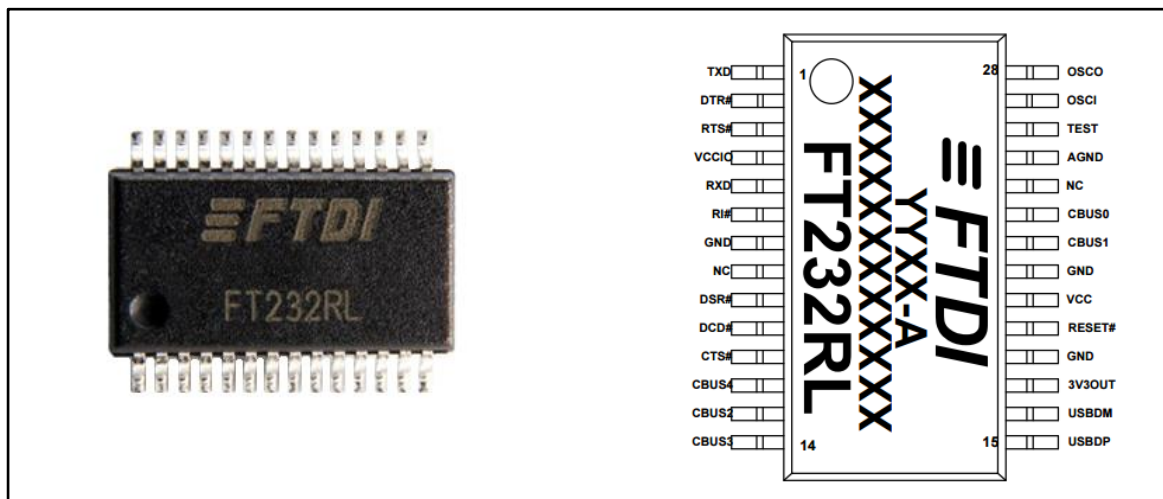


Figura 15 - Foto de um FTDI à esquerda e diagrama com descrição dos pinos à direita
Fonte: FTDI (2015)

Tabela 6 - Descrição dos pinos FTDI

(continua)

Função	Tipo de pino	Descrição
USBDP	E/S	Sinal de dados USB+
USBDM	E/S	Sinal de dados USB-
VCCIO	A	Alimentação +1.8V a +5.25V
GND	A	GND
3V3OUT	O	Saída 3.3V regulada
VCC	A	Alimentação +3.3V a 5.25V
AGND	A	GND analógico

Tabela 6 - Descrição dos pinos FTDI

(conclusão)

Função	Tipo de pino	Descrição
NC	NC	Não utilizado
RESET#	E	Pino de reset
TEST	E	Coloca em modo teste
OSCI	E	Entrada oscilador 12Mhrz opcional
OSCO	S	Saída oscilador 12Mhrz opcional
TXD	S	Transmissão de dados
DTR#	S	Terminal de dados pronto
RTS#	S	Pronto para enviar
RXD	E	Recepção de dados
RI#	E	Indicador de telefone tocando
DSR#	E	Conjunto de dados pronto
DCD#	E	Portadora detectada
CTS#	E	Controle de apagar envio
CBUS4	E/S	E/S configurável
CBUS2	E/S	E/S configurável
CBUS3	E/S	E/S configurável
CBUS1	E/S	E/S configurável
CBUS0	E/S	E/S configurável

Fonte: FTDI (2015).

*E= Entradas S= Saídas A=Alimentação

3.1.4 Drive de potência L298

O L298 é um circuito integrado utilizado para acionamento cargas de potência através de sinais lógicos de baixa potência, onde circuitos com estas características, podem ser chamados de *drive* de potência. O *drive* L298 é recomendado para acionamentos de cargas indutivas, tais como: relés, solenoides, motores de corrente contínua e motores de passo. Pode ser utilizado em aplicações automobilísticas, médicas, industriais, entre outros. Este *drive* é capaz de ser acionados por níveis de tensões padrão TTL (Lógica Transistor-Transistor), ou seja, tensões de 0 e 5V, e fornecer tensões de saída de até 46V, com correntes de até 4A tendo como configuração interna a forma de duas pontes H, permitindo assim, o acionamento de dois motores de corrente contínua nos dois sentidos de rotação. O controle de sentido de rotação é realizado apenas comutando seus pinos através de sinais digitais.

A Figura 16 apresenta uma foto real e disposição dos pinos do L298 com encapsulamento *Multiwatt15* e logo abaixo, na Tabela 7, a descrição das funções dos pinos.

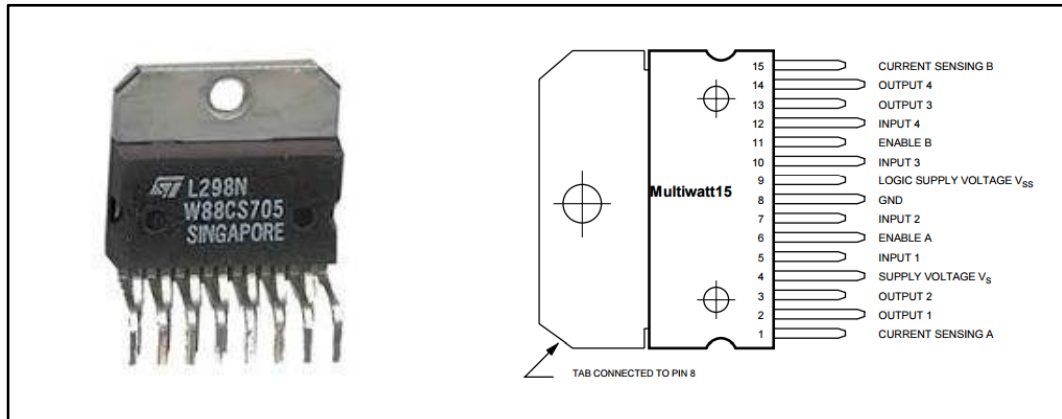


Figura 16 – Foto de um L298 à esquerda e diagrama com descrição dos pinos à direita
Fonte: STMicroelectronics (2000)

Tabela 7 - Descrição dos pinos L298

Função	Descrição
Sense A ; Sense B	Entre este pino e o GND pode ser ligado um resistor de sensor de corrente
Out 1; Out 2	Saída de potência da Ponte A
VS	Tensão de alimentação das Saídas de Potência
Input 1; Input 2	Entradas TLL compatíveis da Ponte A
Enable A; Enable B	Entradas TLL compatíveis para ativar ou desativar as pontes A e/ou B
GND	GND
VSS	Tensão de alimentação dos circuitos lógicos
Input 3; Input 4	Entradas TLL compatíveis da Ponte B
Out 3 ; Out 4	Saída de potência da Ponte A

Fonte: STMicroelectronics (2000)

A Figura 17 mostra o diagrama de blocos do esquema elétrico interno do L298.

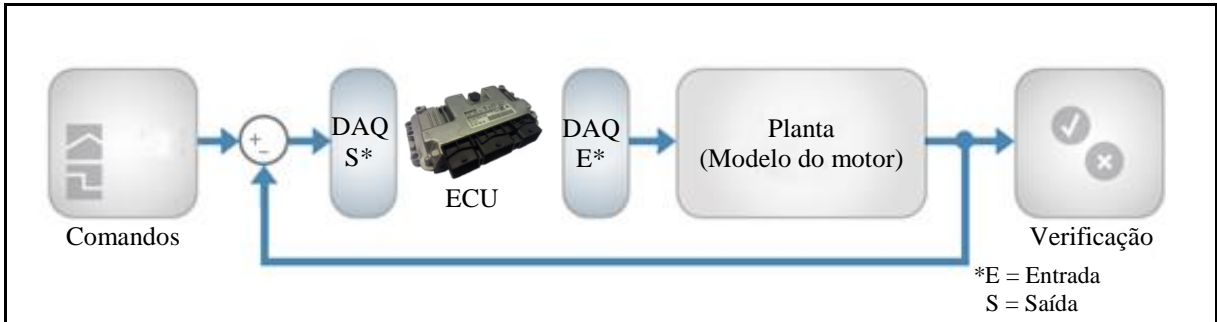


Figura 18 - Plataforma HIL (Hardware-In-the-Loop)
 Fonte: Adaptado de MathWorks (2016)

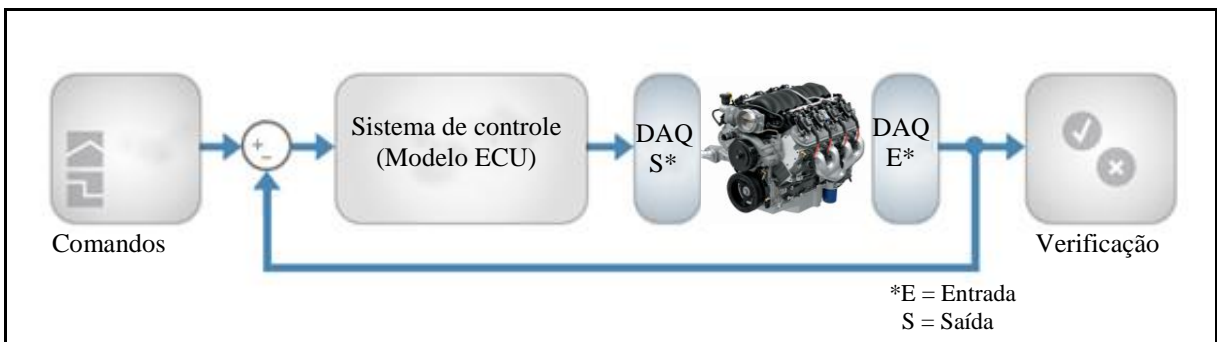


Figura 19 - Plataforma RCP (Rapid Control Prototyping)
 Fonte: Adaptado de MathWorks (2016)

Percebe-se através da Figura 18, que a plataforma HIL, o hardware dedicado de controle (ECU) é usado no experimento e a planta (Motor) é substituída por um modelo de simulação em tempo real. Esta abordagem é clássica em testes e análises de comportamento quando se deseja alterar a planta. Já nos sistemas RCP, como apresentado pela Figura 19, a planta real (motor) é utilizada no experimento e o hardware dedicado de controle (ECU) é substituído por um modelo simulado do controlador, executado em um *software* de propósito geral, como o *MATLAB/Simulink*. Esta abordagem é empregada com frequência quando se deseja testar novos algoritmos de controle diretamente na planta física (Mecatrônica Atual, 2007).

Tanto para plataforma HIL quanto para o RCP, o *Simulink* utiliza-se blocos do RTWT para acessar os pinos de entrada e saída, bem como configurar as funções dos dispositivos externos. Estes dispositivos externos são baseados em sistemas de aquisição (DAQ) que podem ser integrados diretamente no computador, por exemplo, através de placas com slots PCI, PCI-Express ou também podem ser conectados através de conexões USB, RS232, CAN, Ethernet ou até mesmo rede sem fio. Os blocos do *Toolbox* RTWT no *Simulink* utilizados para acesso à dispositivos externos são mostrados na Figura 20.

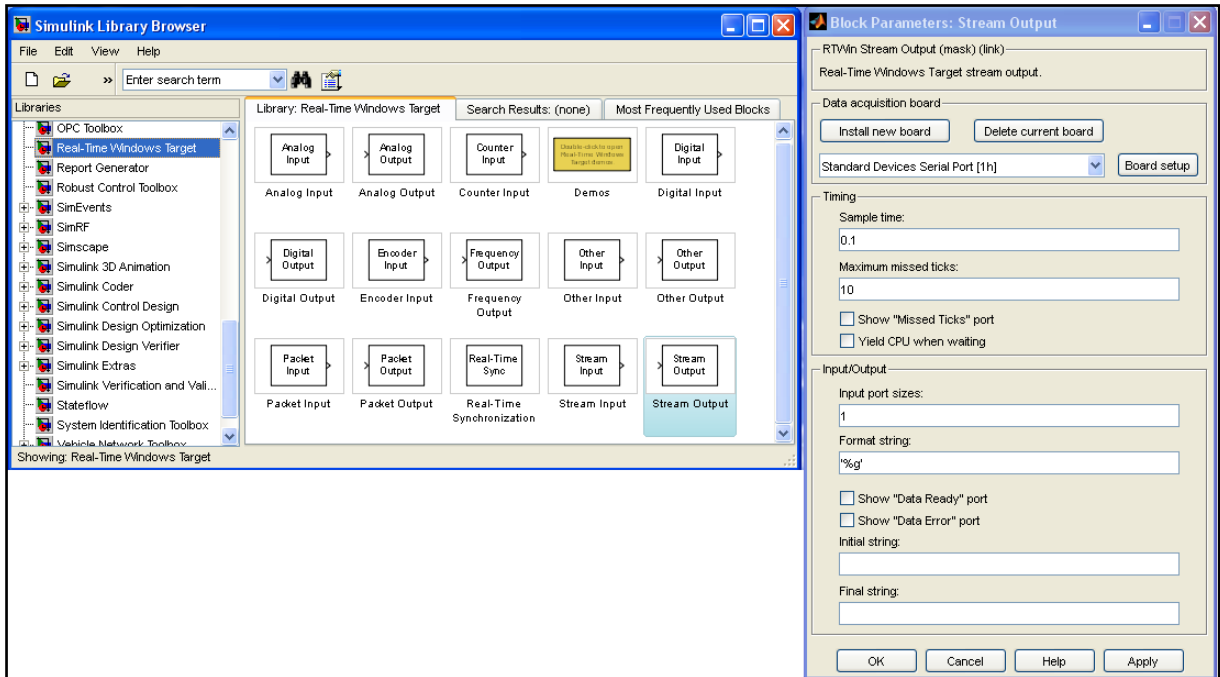


Figura 20 - Toolbox RTWT para acesso a sistemas de aquisição (DAQ)
Fonte: Autoria Própria

Portanto, através destes blocos apresentados, é possível realizar a interligação com sistemas do mundo real com todas as demais funcionalidades oferecidas pelo ambiente *MATLAB/Simulink*, disponibilizando assim, uma ferramenta poderosa para estudo de sistemas de controle ou vários outros tipos de sistemas.

3.2 DESENVOLVIMENTO

Esta seção tem como objetivo apresentar o desenvolvimento da bancada didática a partir dos materiais que foram apresentados na seção anterior.

De modo a facilitar o entendimento, a bancada didática para controle de um motor CC, foi dividido em três partes principais, sendo elas: o ambiente de desenvolvimento gráfico, o sistema de aquisição de dados e a planta a ser controlada. Um diagrama contendo as três partes e a forma como são integrados entre si é apresentado pela Figura 21.

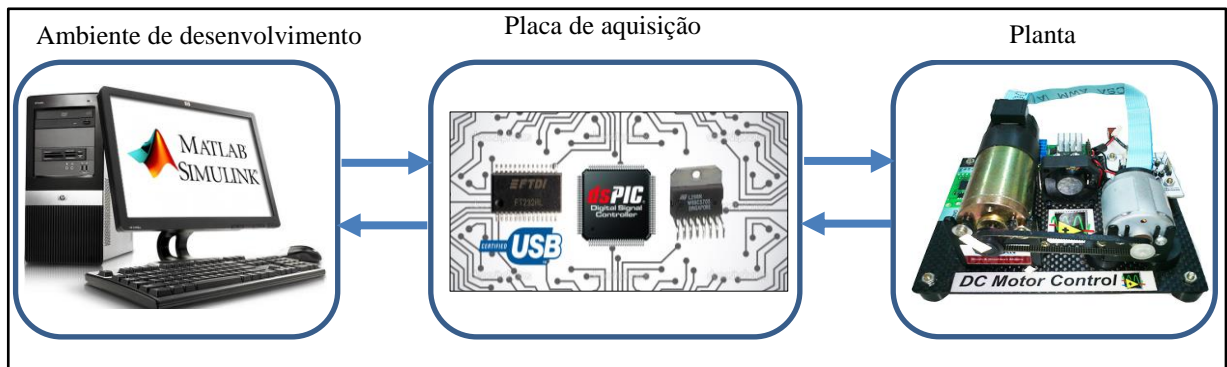


Figura 21 - Diagrama da bancada didática para controle de motor CC

Fonte: Autoria Própria

O modelo deste trabalho é baseado na plataforma de prototipagem rápida de controle (RCP), ou seja, utilizou-se o software *MATLAB/Simulink* como ambiente gráfico para o estudo de sistemas de controles sobre uma planta real, que neste trabalho, é a bancada com Motor de CC. O sistema de aquisição entre o software *MATLAB/Simulink* e a bancada de Motor CC foi utilizada uma placa baseada no Controlador Digital de Sinais dsPIC30F2010.

Neste trabalho, das três partes apresentadas anteriormente, foi necessário desenvolver duas partes delas: a placa de aquisição de dados e uma interface no ambiente *MATLAB/Simulink* para acesso às variáveis da bancada e para implementação do sistema de controle. A planta contendo o motor CC de imã permanente não necessitou ser desenvolvida, pois utilizou-se a bancada que havia disponível no Laboratório de Sistemas Automatizados e Controle, conforme foi detalhada anteriormente.

Toda a discussão do desenvolvimento das duas partes, é essencialmente, focada em fazer a integração entre o hardware e o software para funcionamento da bancada didática.

Portanto, foi necessário desenvolver uma placa de aquisição de dados, que contém um *firmware* e uma interface em *MATLAB/Simulink* para acesso às variáveis da bancada de motor CC e implementação dos sistemas de controles. Os desenvolvimentos destes itens são apresentados a seguir.

3.2.1 Elaboração da placa de aquisição

Para iniciar o desenvolvimento de uma bancada didática para controle do motor CC, primeiramente deve haver uma interface entre hardware e software, que permita a integração

entre a bancada de motor CC e o computador com software *MATLAB/Simulink*. Esta interface é realizada pela placa de aquisição.

A placa de aquisição foi projetada com base no Controlador Digital de Sinais dsPIC30F2010, pois nele estão presentes características que foram essenciais para o desenvolvimento do projeto, que são: alto poder de processamento, módulo de leitura de encoder (QEI), módulos de geração de PWM, comunicação UART e outras características foram descritas na seção anterior.

A placa foi confeccionada através do software EAGLE (*Easy Applicable Graphical Layout Editor*), produzida pela empresa CadSoft, que é um software de projetos para circuitos impressos. Para simplificar a confecção da placa no software, foi dividida em quatro esquemas elétricos, sendo eles: circuito principal do dsPIC30F2010 e fonte reguladora, comunicação, acionamento do motor e acionamento de cargas. A Figura 22 mostra o diagrama dos circuitos presentes nas placas desenvolvidas.

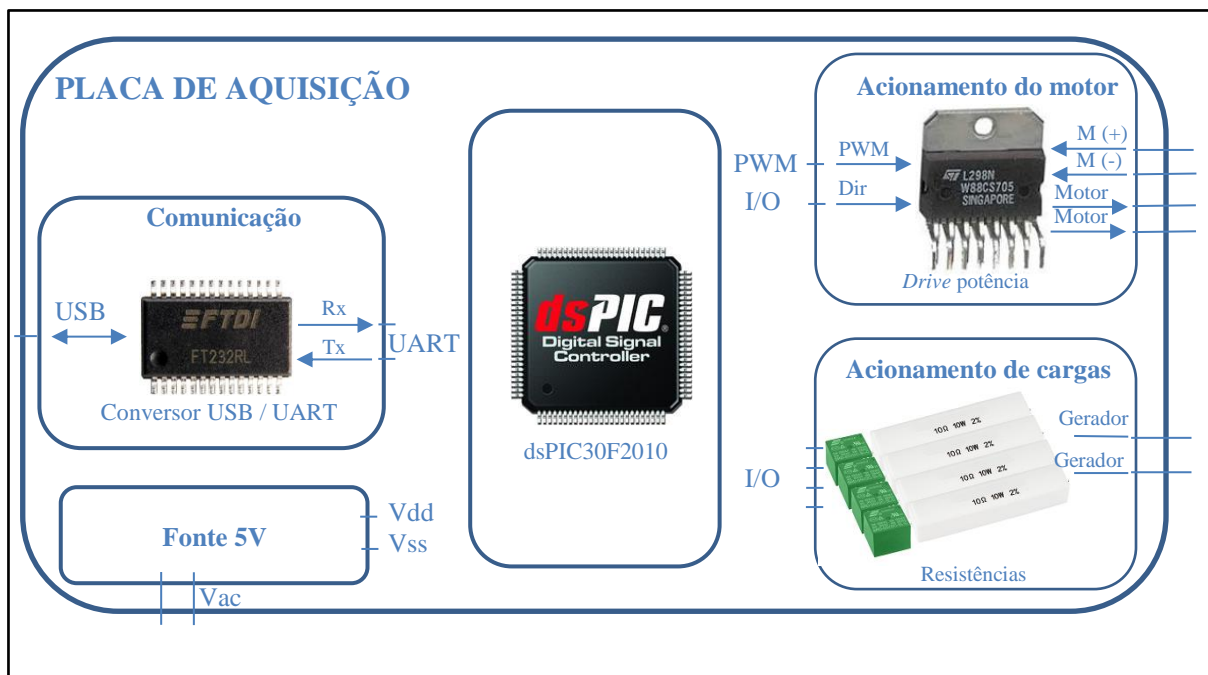


Figura 22 - Diagrama com componentes da placa de aquisição
Fonte: Autoria Própria

Cada circuito foi projetado em função dos componentes necessários para funcionamento com base na folha de dados de cada componente, sendo assim, o esquema elétrico desenvolvido no software EAGLE para cada uma das partes é mostrada nas Figura 23, Figura 24 e Figura 25.

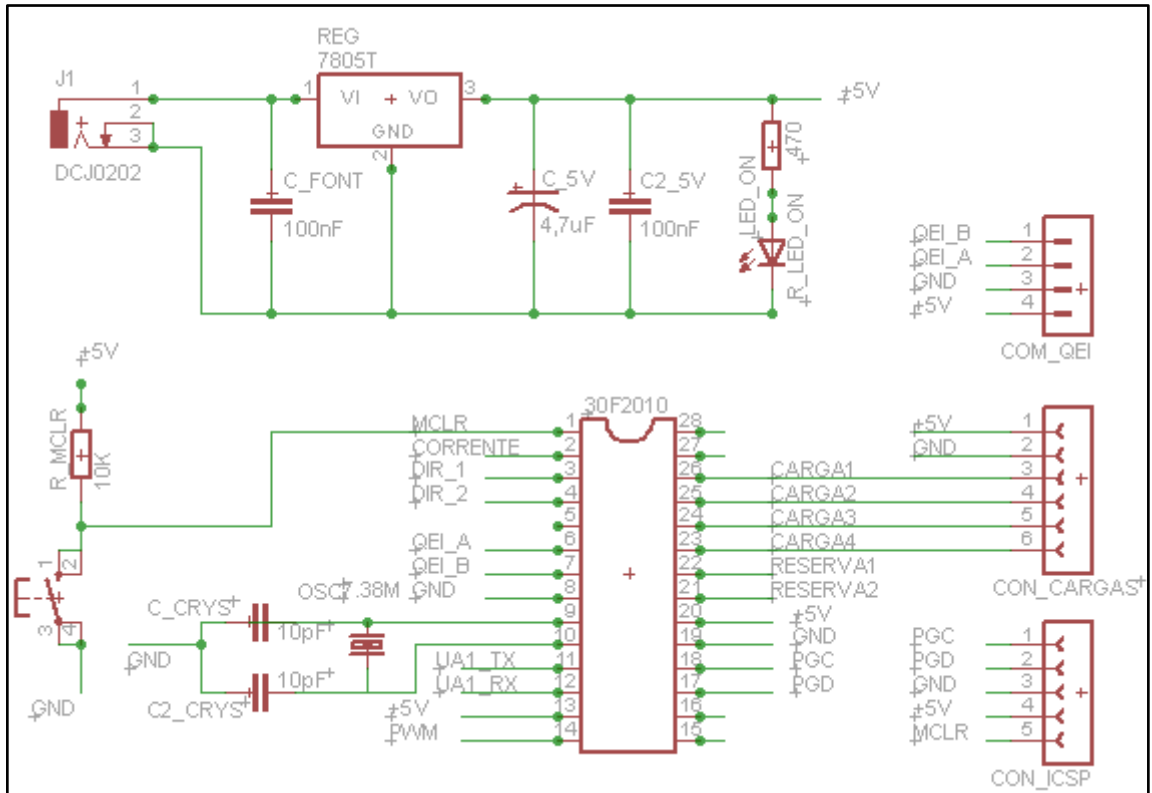


Figura 23 – Esquemático do circuito principal do dsPIC30F2010 e da fonte reguladora
Fonte: Autoria Própria

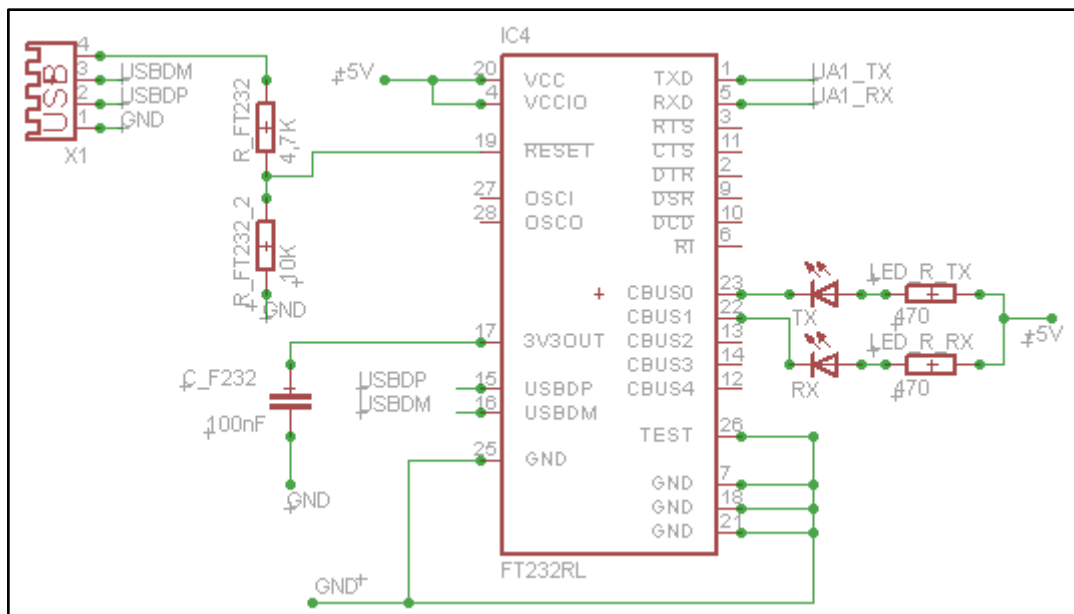


Figura 24 - Esquemático do conversor USB / UART para comunicação
Fonte: Autoria Própria

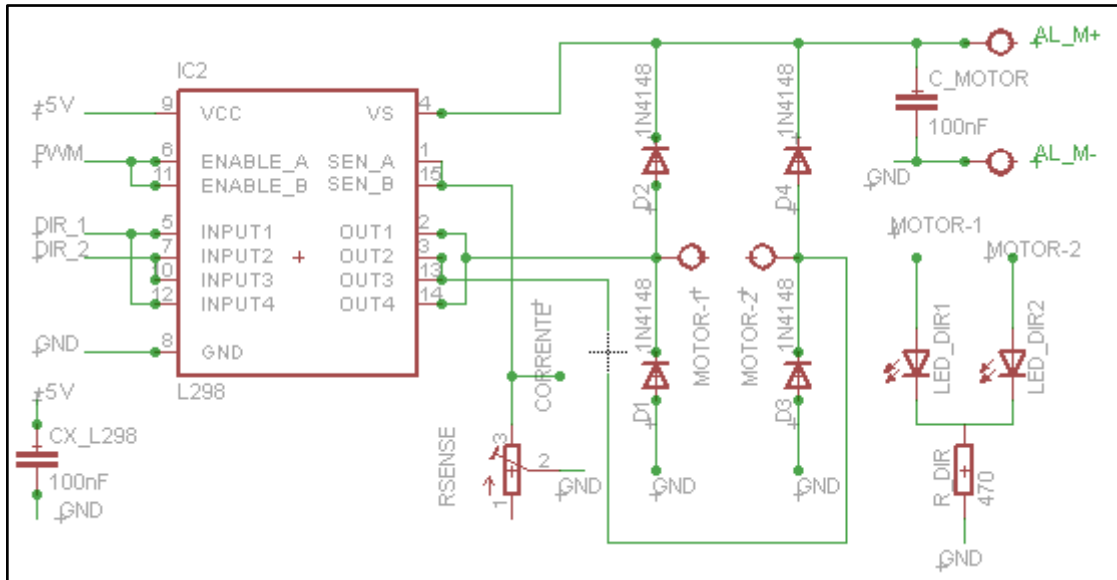


Figura 25 - Esquemático do *drive* de potência para acionamento do motor
Fonte: Autoria Própria

Feito os esquemáticos no software EAGLE, a placa foi confeccionada pelo método da transferência térmica, que consiste em imprimir o layout onde contém as trilhas da placa sobre um papel de transferência com uma impressora a laser. O toner da impressora a laser depositado no papel, é transferido do sobre o cobre presente na placa de circuito através de uma prensa termina, formando assim, uma camada de toner sobre o cobre da placa, com formato do layout que se deseja obter. Posteriormente, a placa é colocada em uma solução química onde ocorre a corrosão do cobre que não está protegido pelo toner, restando assim somente o cobre com formato do layout desejado. A Figura 26 apresenta a vista superior e inferior da placa, já com os componentes montados após confecção por transferência térmica.

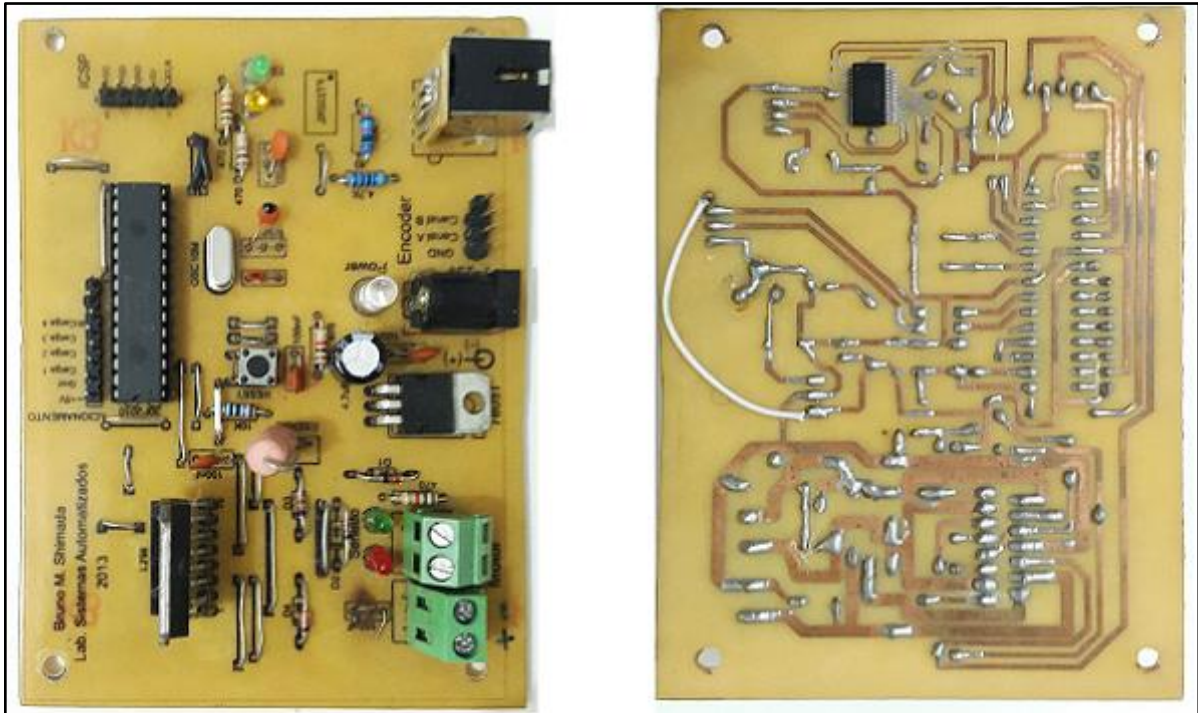


Figura 26 - Placa principal - vista superior e inferior
Fonte: Autoria Própria

A máscara dos componentes e textos nas placas, é feito pelo mesmo processo de transferência térmica do toner, porém não é realizado a corrosão através da solução química.

O circuito para controle de carga foi confeccionado em uma placa separada da placa principal, de modo que as duas placas fossem fixadas uma a cima da outra, a fim de manter uma bancada didática de tamanho reduzido.

O circuito de controle de carga consiste em resistores que podem ser ligados em série com enrolamento de armadura do gerador CC de imã permanente, com o objetivo de testar a eficácia dos sistemas de controle no motor CC. Foram utilizadas quatro resistências de $4R7\Omega$ 10W, que podem ser combinados em paralelo para gerar diferentes resistências equivalentes, ocasionando diferentes intensidades de carga mecânica no eixo do gerador. A combinação dos resistores poderá ser alterada pelo *MATLAB/Simulink*.

A Figura 27 mostra o esquemático do circuito de acionamento de controle de carga, e a Figura 28, a placa já finalizada.

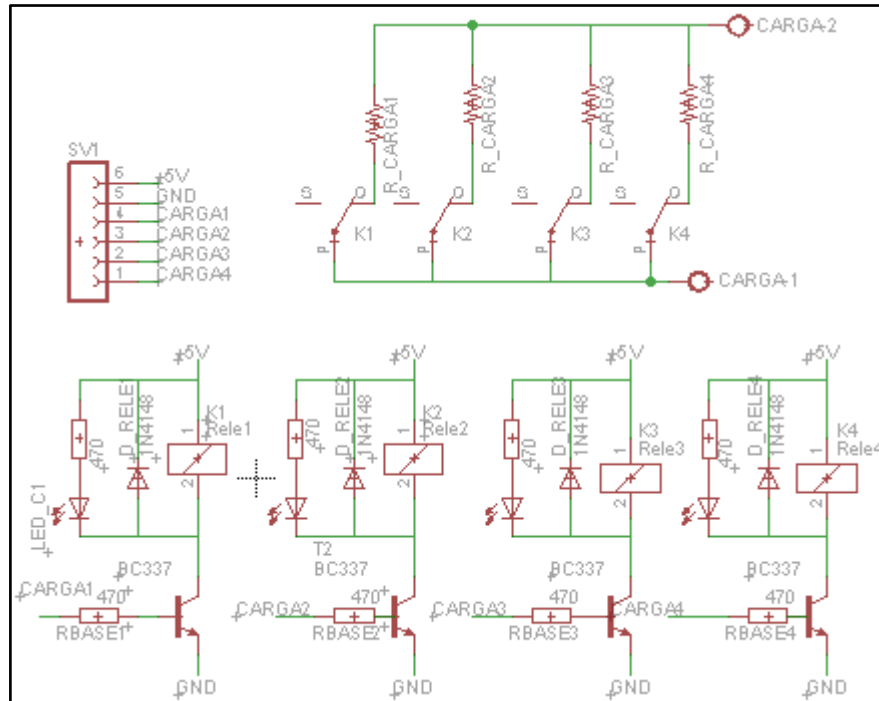


Figura 27 - Esquemático do controle de carga
Fonte: Autoria Própria

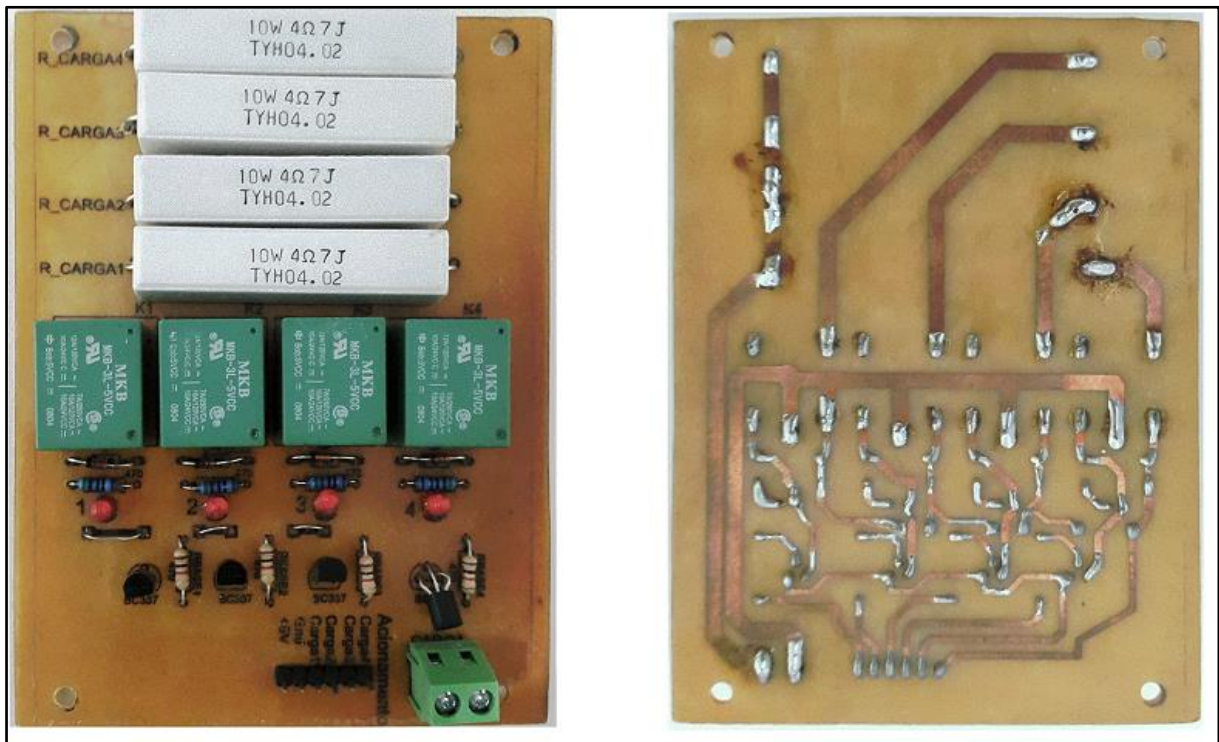


Figura 28 - Placa de acionamento do controle de carga
Fonte: Autoria Própria

A Figura 29 apresenta a placa de aquisição finalizada com a placa principal e a placa de acionamento de carga e ao lado, a bancada de motor CC.

O tamanho do da placa de aquisição após finalizada possui 11,5 cm de comprimento, 8 cm de profundidade e aproximadamente 6 cm de altura. Já a bancada de possui 13 cm de comprimento, 11 cm de profundidade e 6 cm de altura.

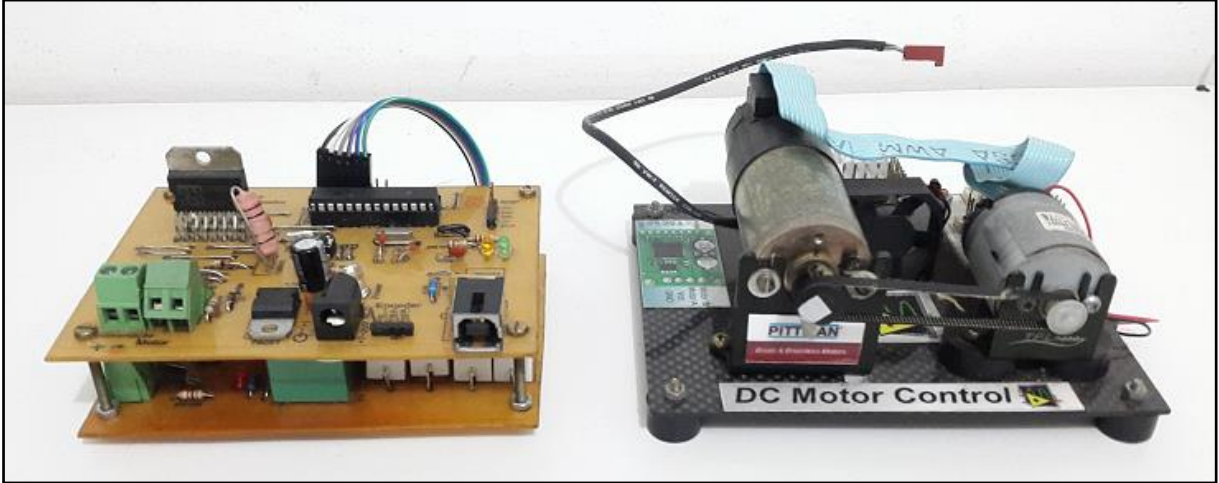


Figura 29 - Placa de aquisição e bancada de motor CC
Fonte: Autoria Própria

3.2.2 Desenvolvimento de um Firmware para Placa de Aquisição

Para que a placa de aquisição confeccionada anteriormente funcione corretamente, foi necessário desenvolver um programa contendo um conjunto de instruções diretamente no dsPIC30F2010 para que exista a integração entre software (*MATLAB/Simulink*) e o hardware (Bancada com motor CC). Neste trabalho, este programa foi chamado de *firmware* da placa de aquisição.

Dentre as principais funções, o *firmware* é responsável por gerenciar tarefas do dsPIC30F2010 para que possa realizar cálculos matemáticos, interpretar pulsos elétricos do encoder e converte-los em medidas de velocidade, gerar sinais PWM para o *drive* do motor, controlar o acionamento dos resistores de cargas, e principalmente, realizar a troca destas informações para o software *MATLAB/Simulink* através de um protocolo de comunicação.

O *firmware* foi desenvolvido em Linguagem C, utilizando como ambiente de programação o software MPLAB IDE da *Microchip Technology* juntamente com compilador CCS C para geração do arquivo em linguagem de máquina, e que foi programado no dsPIC30F2010 através de um gravador PICkit 2, também da *Microchip Technology*. O código fonte do *firmware* desenvolvido encontra-se no apêndice A deste final do trabalho.

O fluxograma apresentado pela Figura 30, mostra de forma simplificada, a rotina das tarefas realizadas pelo *firmware* programado no dsPIC3020F10.

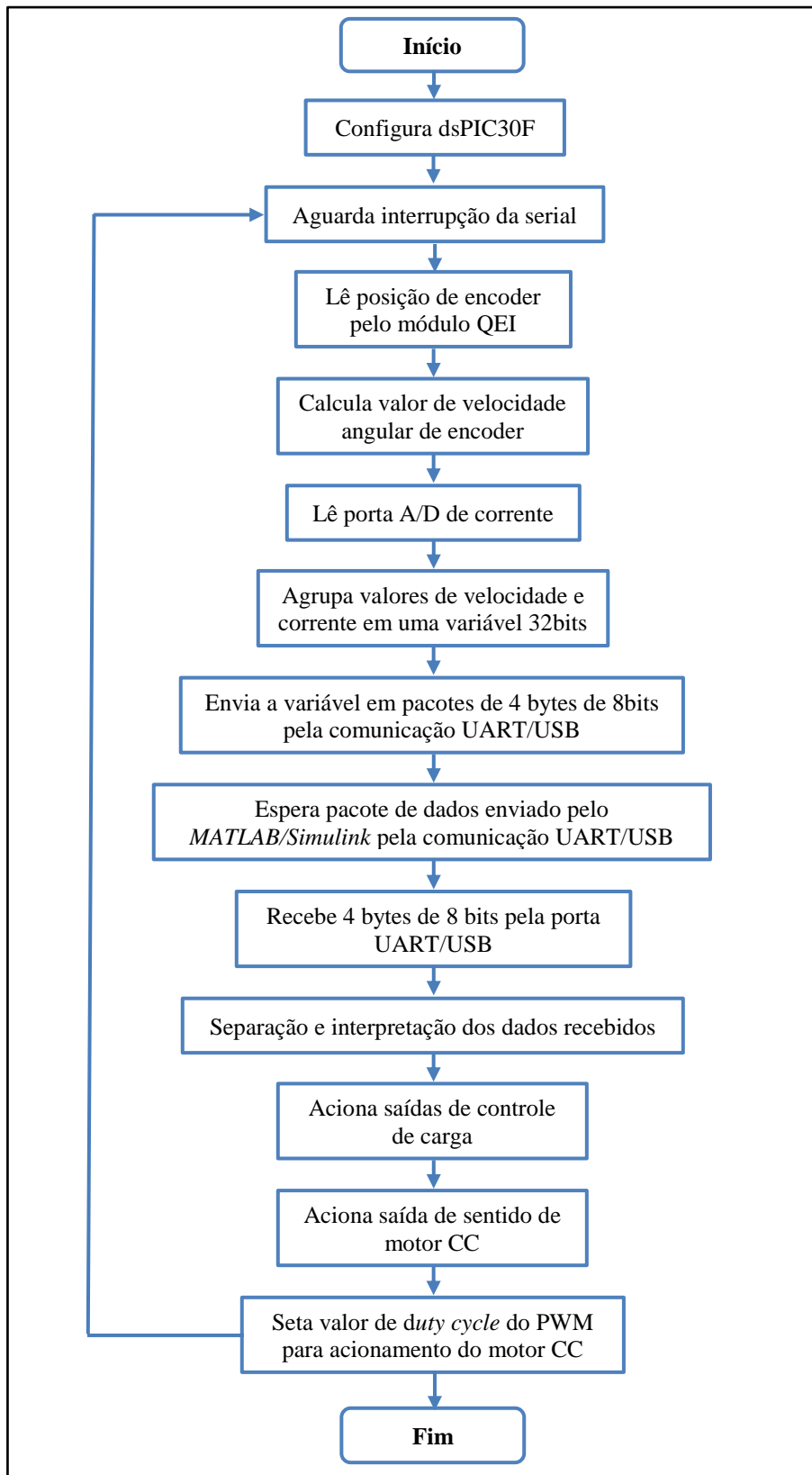


Figura 30 - Fluxograma do *firmware*
Fonte: Autoria Própria

Como observado pelo fluxograma, a rotina implementada pelo *firmware* é iniciada pela configuração dos parâmetros de funcionamento do dsPIC30F2010, tais como: inicialização de variáveis, configuração dos *timers*, configuração do PWM, configuração do módulo A/D, configuração do módulo QEI e configuração do módulo UART. Posteriormente, o programa entra em um laço de repetição, onde é realizada a leitura da posição do encoder através do módulo QEI de 16 bits, em seguida é calculada a velocidade angular a partir do valor da posição lido anteriormente e por último é realizada uma leitura da porta A/D de 16bits, que informa a tensão proporcional a corrente que circula pelo Motor CC. Estes valores, são agrupados em uma variável de 32bits, que é enviada através da comunicação UART/USB ao computador executando o software *MATLAB/Simulink*, sob a forma de 4 bytes de 8 bits.

```

.....
#int_rda
void serial_isr()
{
.....
    if (Recebido == '}')
        {
            Flag=0;
            buffer_r[0]=Texto[0];
            buffer_r[1]=Texto[1];
            buffer_r[2]=Texto[2];
            buffer_r[3]=Texto[3];
            buffer[0]= valor_enviar3 & 0b00000000000000000000000011111111;
            valor_enviar3 = valor_enviar3>>8;
            buffer[1]= valor_enviar3 & 0b00000000000000000000000011111111;
            valor_enviar3 = valor_enviar3>>8;
            buffer[2]= valor_enviar3 & 0b00000000000000000000000011111111;
            valor_enviar3 = valor_enviar3>>8;
            buffer[3]= valor_enviar3 & 0b00000000000000000000000011111111;

            putc(buffer[0]); //Envia byte 1
            putc(buffer[1]); //Envia byte 2
            putc(buffer[2]); //Envia byte 3
            putc(buffer[3]); //Envia byte 4
        }
    }
.....

```

Os valores enviados ao computador são processados por um sistema de controle implementado no software *MATLAB/Simulink*, onde, posteriormente é retornado via comunicação UART/USB para o dsPIC30F2010 por um pacote de dados formado por 4 bytes de 8 bits conjuntamente com um byte de início representado pelo caractere “{” e um byte de término representado pelo caractere “}” .

A recepção destes 4 bytes no dsPIC30F2010 é feito pelos seguintes comandos:

```

.....
#int_rda
void serial_isr()
{
    Recebido = getc();
    if (Recebido == '{')
        {
            Flag=1;
            Ponteiro = &Texto;
        }
    else if (Flag==1)
        {
            *Ponteiro = Recebido;
            Ponteiro++;
        }
    ....
}
.....

```

Recebidos os valores de 4 bytes de 8 bits, o mesmo são unidos de modo a formar uma variável de 32bits. Este pacote de 32 bits contém valores do *duty cycle* de 16 bits, valor da direção do motor com 1 bit, valor do estado do motor com 1 bit e valor do acionamento de carga do gerador CC com 14 bits. Estes valores são extraídos do pacote de 32bits por uma técnica de mascaramento implementados pelas linhas de comando abaixo :

```

.....
value_received= buffer_r[0]+256*buffer_r[1]+65536*buffer_r[2]+16777216*buffer_r[3];
pwm = value_received & 0xFFFF0000;
pwm = pwm >> 16 ;
direcao = value_received & 0x20;
direcao = direcao >> 5;
enable_motor = value_received & 0x10;
enable_motor = enable >> 4;
loads= value_received & 0xF;
load1= loads & 0b1000;
load1 = load1 >> 3;
load2= loads & 0b0100;
load2 = load2 >> 2;
load3= loads & 0b0010;
load3 = load3 >> 1;
load4= loads & 0b0001;
.....

```

Com os valores extraídos do pacote, estes são utilizados para que o dsPIC30F2010 acione o *drive* de potência do motor CC com valor de *duty cycle* recebido através do módulo PWM, define o sentido de rotação, o estado de funcionamento do motor CC, isto é, se ligado ou desligado, e aciona as devidas saídas das cargas do gerador CC. Terminado estas tarefas, o laço de repetição do *firmware* inicia-se novamente, repetindo todo o ciclo a cada tempo de

discretização. A frequência de oscilação do PWM foi definida em 1kHz, devido a esta frequência servir para a maior parte dos motores de corrente contínua.

A figura abaixo mostra o conteúdo dos pacotes de dados de 32bits trocados entre o *MATLAB/Simulink* e a placa de aquisição através da comunicação o UART/USB.

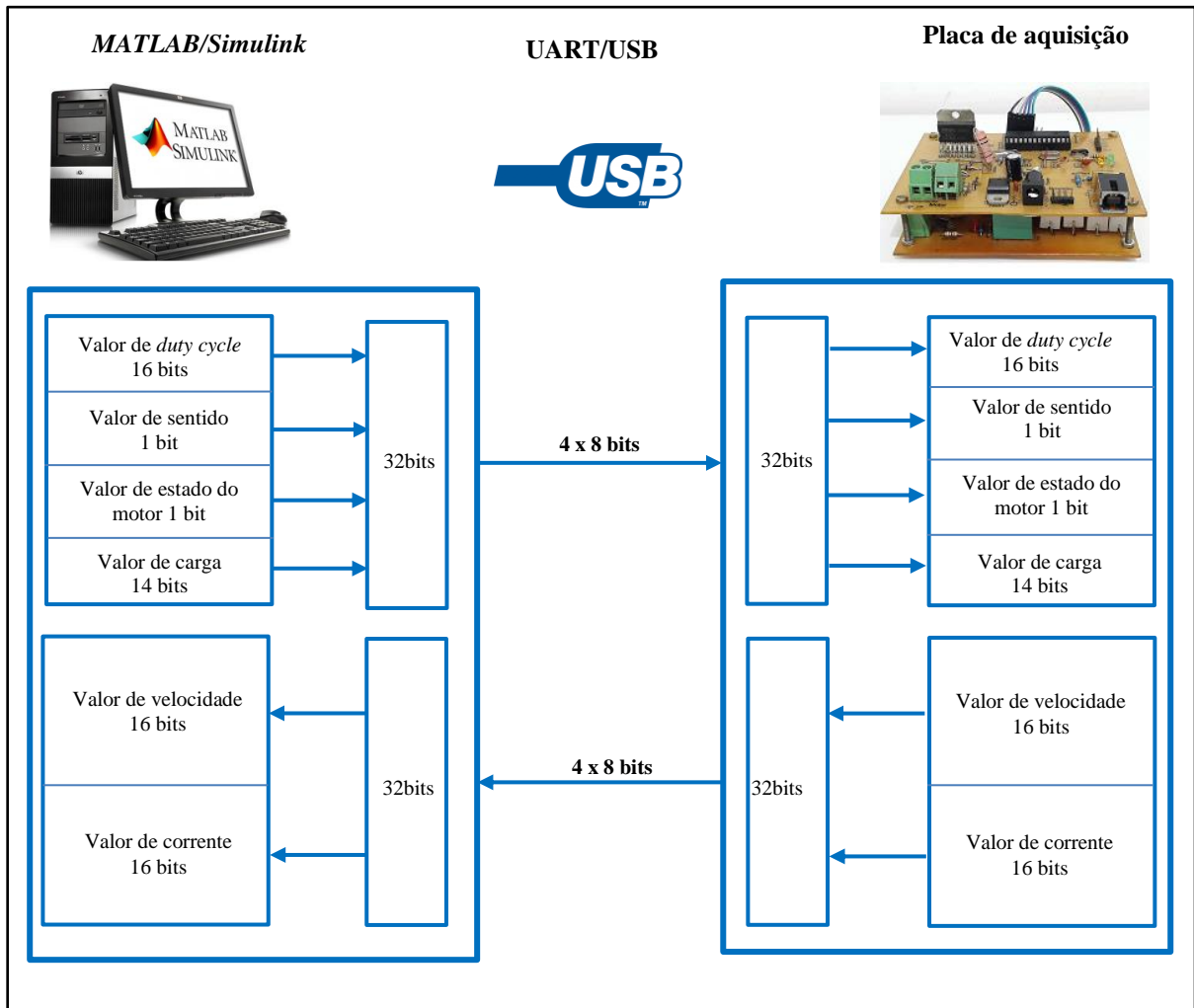


Figura 31 - Diagrama de troca de dados entre o MATLAB/Simulink e a Placa de aquisição.

Fonte: Autoria própria

3.2.2 Desenvolvimento da Interface MATLAB/Simulink

Realizado a placa de aquisição e seu *firmware*, para completar o desenvolvimento de uma bancada didática de motor CC, foi necessário criar uma interface no software *MATLAB/Simulink* para possibilitar a leitura e controle em tempo real dos parâmetros da bancada do motor CC.

Busca-se desenvolver uma interface simplificada e de fácil utilização para o usuário da bancada didática, de tal forma que, através dos blocos de comunicação sejam possíveis implementar diferentes técnicas de sistemas de controle, desde que se utilizem as mesmas variáveis de entradas e saídas, sem que seja necessário alterar ou dedicar tempo na configuração da placa de aquisição, facilitando assim, a análise de diferentes técnicas de controle sobre a mesma planta.

Para desenvolver esta interface, utilizou-se os blocos *Packet Input* e *Packet Output* presente no *Toolbox* RTWT do software *MATLAB/Simulink*. Estes blocos permitem o recebimento e envio de dados entre dispositivos com comunicações seriais, TCP/IP, CAN, entre outros. Desta maneira, estes blocos foram utilizados para realizar a troca de dados entre o software *MATLAB/Simulink* e a placa de aquisição através da comunicação UART/USB e o protocolo que foi implementado pelo *firmware*. A versão utilizada do *MATLAB/Simulink* foi a versão 2012a, rodando sobre o sistema operacional *Windows XP Service Pack 3* em um computador com processador *AMD Athlon II X3 3,2GHz* com 4GB de memória RAM.

A Figura 32 apresenta a interface desenvolvida utilizando os blocos *Packet Input* e *Packet Output*, juntamente com os blocos de tratamento de dados e os blocos de acesso aos parâmetros da bancada de motor CC. Observando a figura, as funções dos blocos e suas unidades de medida utilizadas são:

- **Blocos de variáveis de entrada da planta (cor azul):**
 - **RPM:** Velocidade angular do motor em Rotações por minuto;
 - **Corrente:** Valor da corrente do motor CC medido em Volts proporcional a corrente.
- **Blocos de variáveis de saída da planta (cor verde):**
 - **PWM:** Valor do *duty cycle* para acionamento do *drive* do motor, que podem assumir valores de 0 a 254, equivalente à *duty cycles* de 0 a 100%;
 - **Dir:** Valor do sentido do motor, sendo o valor 0 para um sentido e 1 para o sentido o sentido inverso;
 - **En:** Valor do estado do motor, sendo 0 para desligado e 1 para ligado;
 - **Carga:** Referente ao acionamento dos quatros resistores do gerador CC, que podem assumir valores de 0 até 4.
- **Real-Time Sync:** Garante a sincronização de todos os blocos do programa para o funcionamento em tempo real.

- **Tratamento Recebido:** Subsistema que contém tratamento de dados para os dados recebidos pela comunicação;
- **Tratamento de envio:** Subsistema que contém tratamento de dados para o envio de dados enviados pela comunicação;
- **Filtro Buttherworth:** Filtro passa baixa da entrada dos valores de RPM.

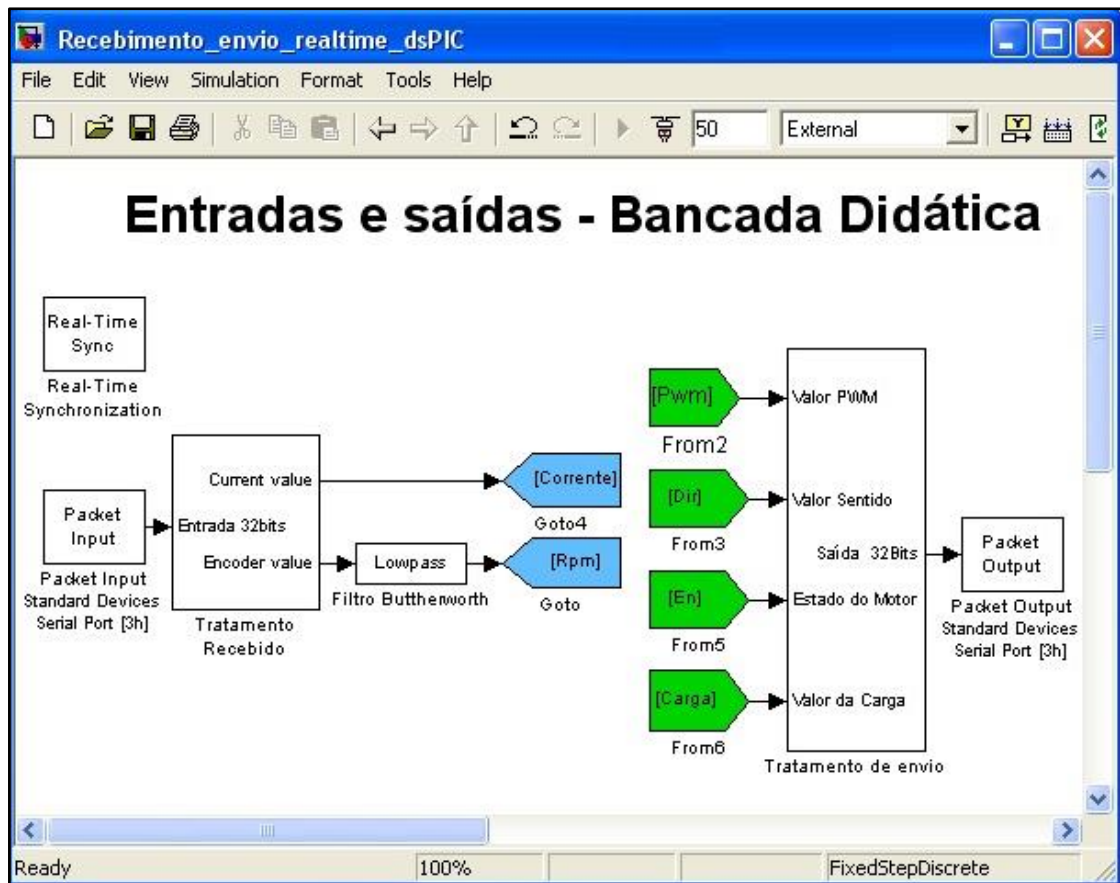


Figura 32 – Interface desenvolvida no *MATLAB/Simulink* para leitura e controle em tempo real da bancada com motor CC

Fonte: Autoria própria

Na variável RPM é utilizado um filtro Buttherworth, implementado através de um biblioteca pré-configurada do *MALTA*B/Simulink para filtrar ruídos de alta frequência características na leitura do encoder. Utilizou-se uma frequência de corte em passa baixa de 15hz. A Figura 33 mostra a curva de resposta do filtro implementado.

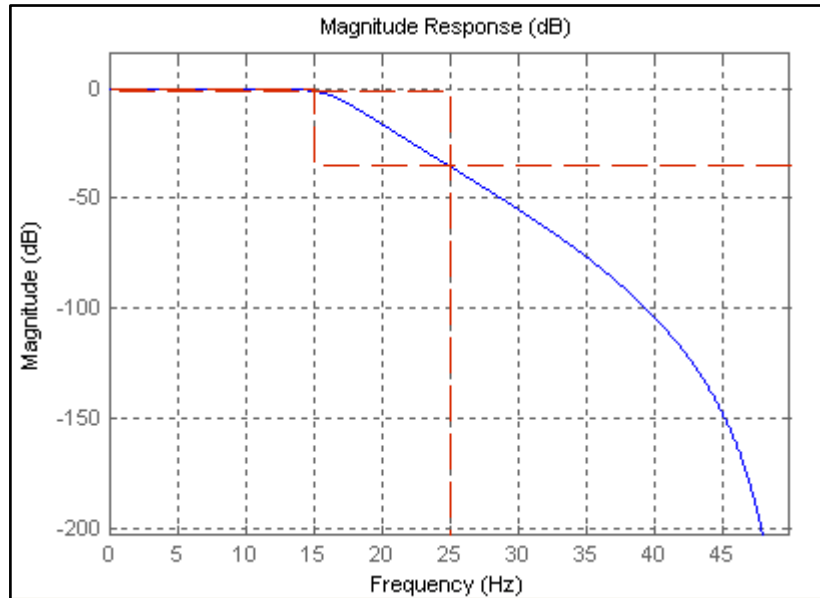


Figura 33 - Curva de resposta do Filtro Butterworth
Fonte: Autoria Própria

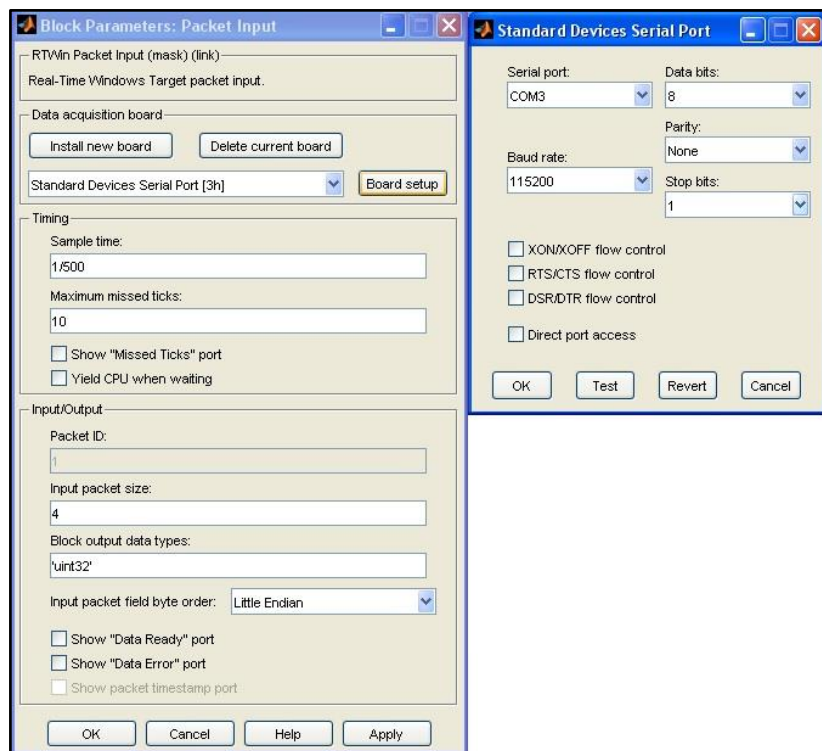


Figura 34 - Configuração do bloco *Packet Input* e da porta serial
Fonte: Autoria Própria

A Figura 34 apresenta as configurações relevantes do bloco *Packet Input* e da comunicação da porta Serial, que dá acesso à comunicação UART/USB. Foi utilizado o tempo de amostragem, ou, também chamado de *Sample Time* de 2 milissegundos, ou seja, a comunicação e o fechamento da malha é realizado 500 vezes durante um segundo. Na comunicação, utilizou-se *Baud Rate* de 115000bps com dados de 8bits e sem paridade.

Da mesma forma que foi implementado o tratamento de dados para os pacotes de comunicação de 32 bits no *firmware*, o tratamento de dados também é realizado dentro dos subsistemas criados no *MATLAB/Simulink*. O tratamento dos dados de 32bits é realizada através de blocos *Extract Bits*, *Shift Arithmetic*, *BitWise OR* e *Data Converter* do *Simulink* que são utilizados para tratamento dos dados através de operações de separação, deslocamentos de bits, união ou conversão do tipo de dados. Estas operações têm como objetivo separar ou formar os pacotes de 32 bits que é enviado ou recebido à placa de aquisição. A figura 35 mostra o diagrama de blocos do subsistema para o tratamento dos dados recebidos pelo *MATLAB/Simulink*.

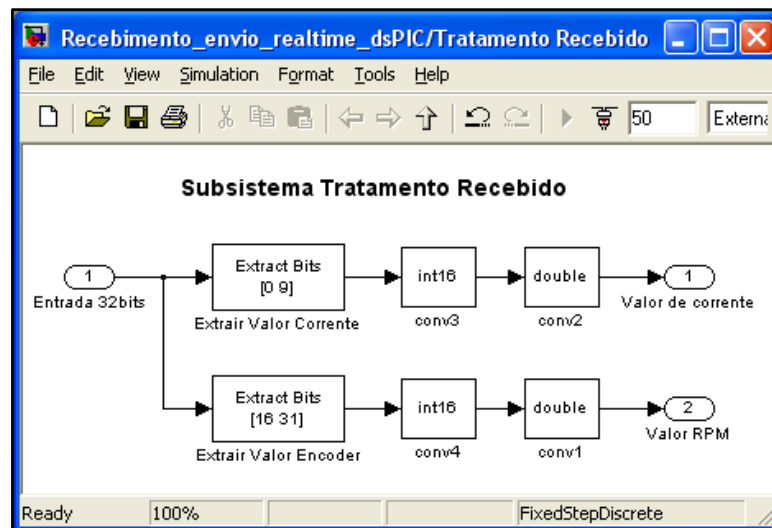


Figura 35 - Subsistema de tratamento dos dados recebidos
Fonte: Autoria Própria

Para os dados de 32 bits recebidos como mostrado na figura anterior, é realizada apenas a separação entre os primeiros 16 bits mais significativos que representa o valor de RPM e os outros 16 bits referente ao valor de corrente.

Já a Figura 36 mostra o diagrama de blocos do subsistema para o tratamento dos dados enviados pelo *MATLAB/Simulink* para placa de aquisição.

O subsistema dos dados enviados, basicamente recebe os valores do PWM, do sentido, do estado do motor e do valor de carga e os unem a um dado de 32 bits, para posteriormente dividi-los em quatro variáveis de 8 bits e uni-los precedente com o dado de início “{” e com o dado de termino “}” que foram definidos pelo protocolo de comunicação.

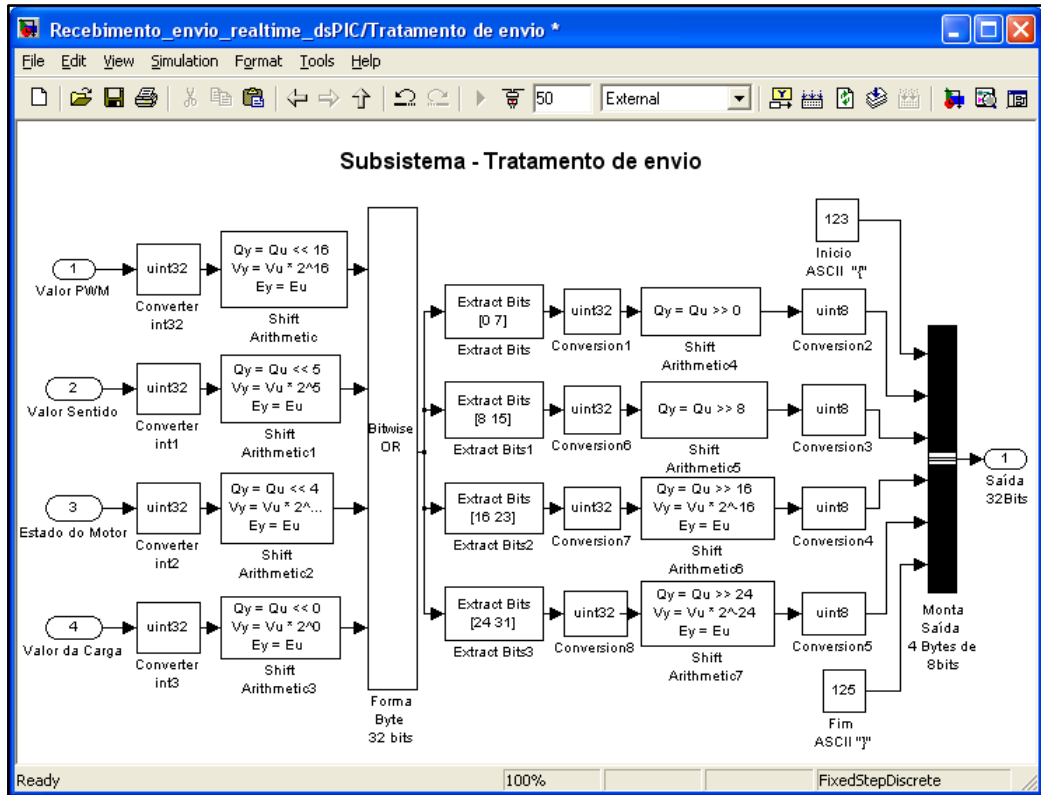


Figura 36 - Subsistema de tratamento dos dados enviados
Fonte: Autoria Própria

Portanto, através da interface em MATLAB/Simulink desenvolvida apresentada pela Figura 32, onde se encontram todos os parâmetros para leitura e controle da bancada de motor CC, completa-se todos os itens necessário da bancada didática para controle de motores CC. A bancada didática pronta para funcionamento, composta pelo ambiente de desenvolvimento, pela placa de aquisição e pela planta é apresentada na Figura 37.

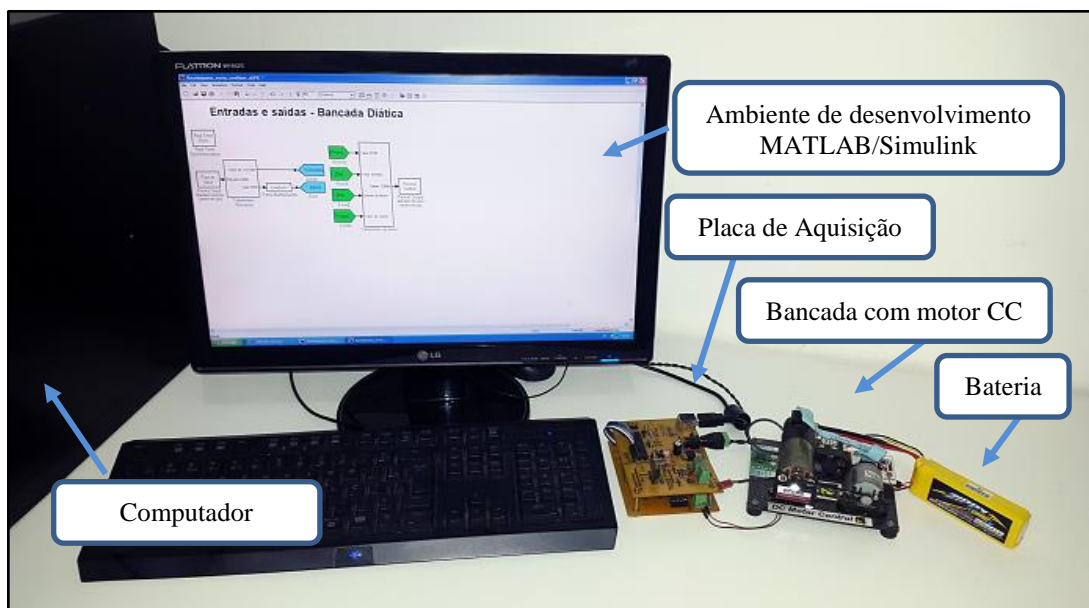


Figura 37 - Bancada didática montada
Fonte: Autoria Própria

4 RESULTADOS

Neste capítulo, é apresentado a implementação de um controlador PID e um controlador Fuzzy na bancada didática desenvolvida. Esta implementação tem como objetivo validar o funcionamento da bancada didática, de forma a testar a eficiência da placa de aquisição e do ambiente de desenvolvido para verificar se os mesmos atendem o objetivo previsto no início do trabalho, que é o desenvolvimento da bancada didática para ensino de controle. Além disto, estas implementações, servirão de exemplos a quem deseja realizar o estudo de sistemas de controle na bancada didática. Ao final do capítulo, é apresentado uma análise de custo da bancada, mostrando os custos totais para desenvolvimento do projeto.

Para implementar os sistemas de controle, foram utilizados os blocos de entradas e saídas da interface desenvolvida no software *MATLAB/Simulink*. Os controladores PID e Fuzzy foram implementados através de bibliotecas presentes no *MATLAB/Simulink*, agilizando e facilitando assim a implementação destes sistemas de controle, já que o objetivo real deste trabalho não é o estudo aprofundado do sistema de controle, mas sim o desenvolvimento de uma bancada para o estudo de sistema controle.

Nos experimentos foram utilizados como fonte de alimentação para acionamento dos motores uma bateria de 11,1V de LiPo (Polímero de Lítio) com capacidade de corrente instantânea máxima de 55A e uma fonte chaveada de 12Vcc de 1A para alimentar a placa de aquisição. O tempo de amostragem utilizado foi de 500Hz.

Para cada sistema de controle foi realizado experimentos com uma entrada degrau, uma entrada de pulso, uma entrada senoidal, uma entrada trapezoidal e por fim uma variação de carga no gerador CC.

4.1 CONTROLE DE VELOCIDADE UTILIZANDO PID

Um dos controles usados para validar o funcionamento da bancada didática foi p Proporcional Integral Derivativo (PID). O controlador PID calcula o valor da ação de controle dado em forma de *duty cycle* de PWM através do erro entre velocidade instantânea e a velocidade desejada (*Setpoint*) utilizando as constantes de proporção, integral e derivativo (K_P , K_I e K_D , respectivamente). Estas constantes do controlador foram definidas heurísticamente

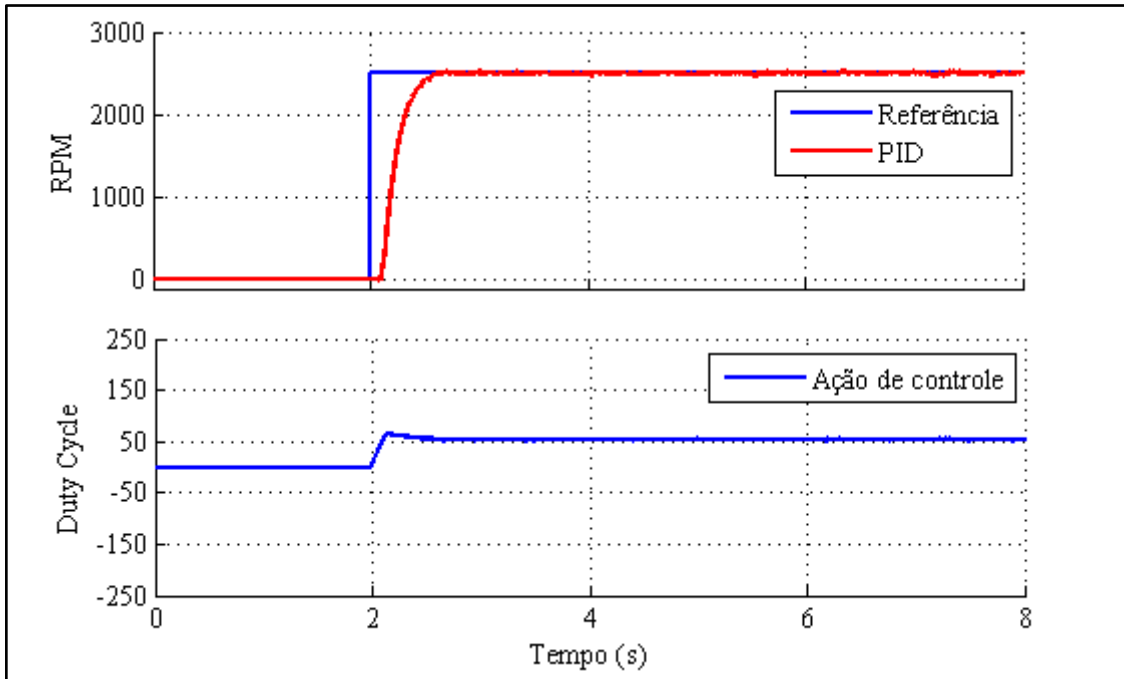


Figura 39 - Entrada degrau de 2500RPM a 2 segundos com controlador PID
 Fonte: Autoria Própria

A Figura 40 mostra o comportamento da bancada didática para inversão do sentido de rotação do motor. Foi realizada uma entrega degrau aos 2 segundos de 2500 RPM e após estabilizado, em 5 segundos foi realizada a inversão do sentido de rotação do motor com a mesma rotação. Em 8 segundos o motor retorna em repouso. O teste apresentou ser capaz de inverter o sentido e estabilizar-se, comprovando o funcionamento da função de inversão do sentido de velocidade.

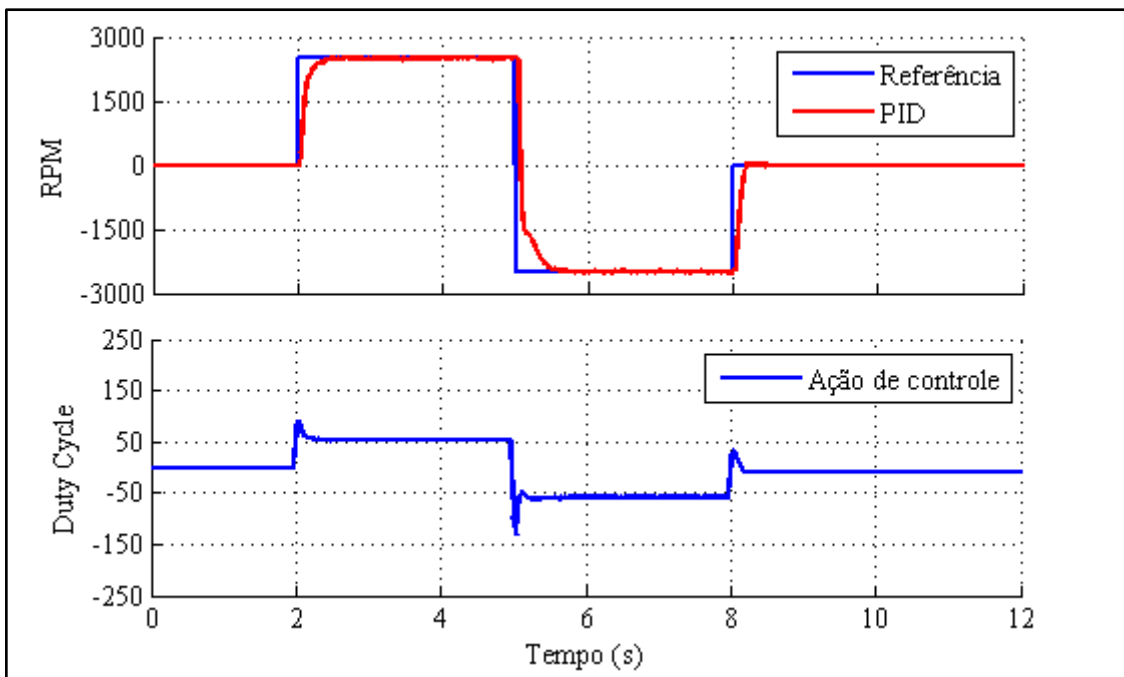


Figura 40 - Entrada degrau de 2500RPM a 2 segundos e inversão de velocidade com controlador PID
 Fonte: Autoria própria

A Figura 41 apresenta o comportamento do sistema a uma entrada senoidal de 0,5Hz com amplitude de 2000 RPM, com inversão de velocidade.

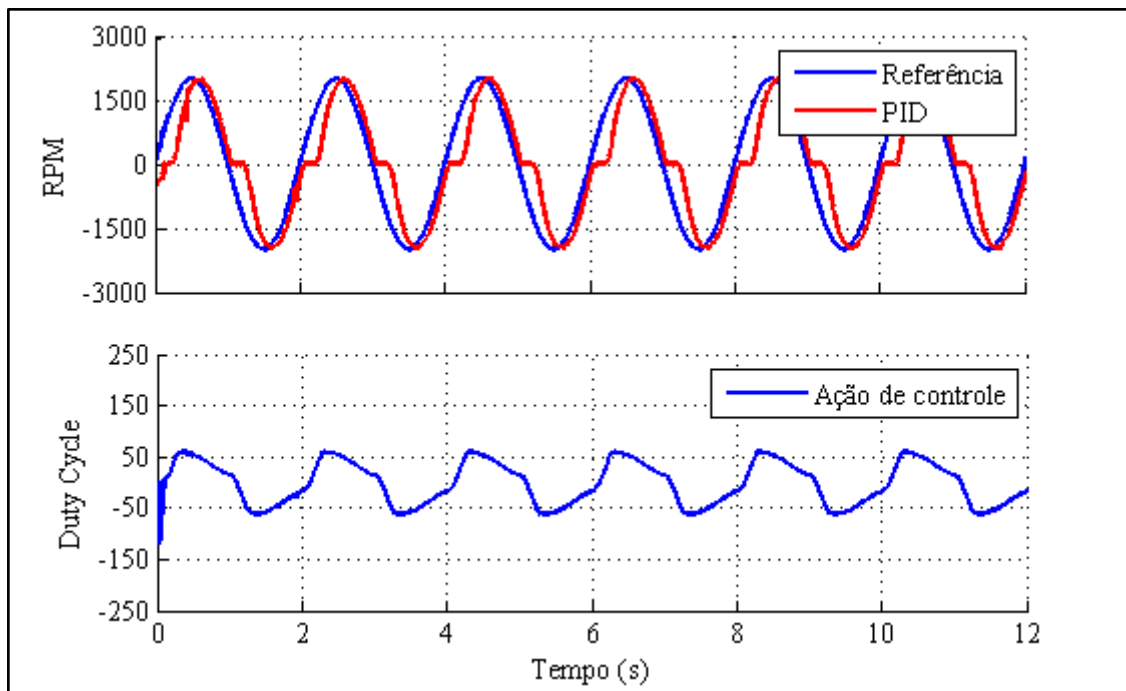


Figura 41 - Entrada senoidal de 0,5Hz com inversão de sentido no controlador PID
 Fonte: Autoria Própria

Nota-se uma descontinuidade durante o cruzamento por zero do sinal de velocidade, isto se deve a não linearidade do motor de corrente contínua, principalmente a baixas rotações, porém este problema é um problema de controle e pode ser corrigido através de ajustes da faixa de operação do *duty cycle*.

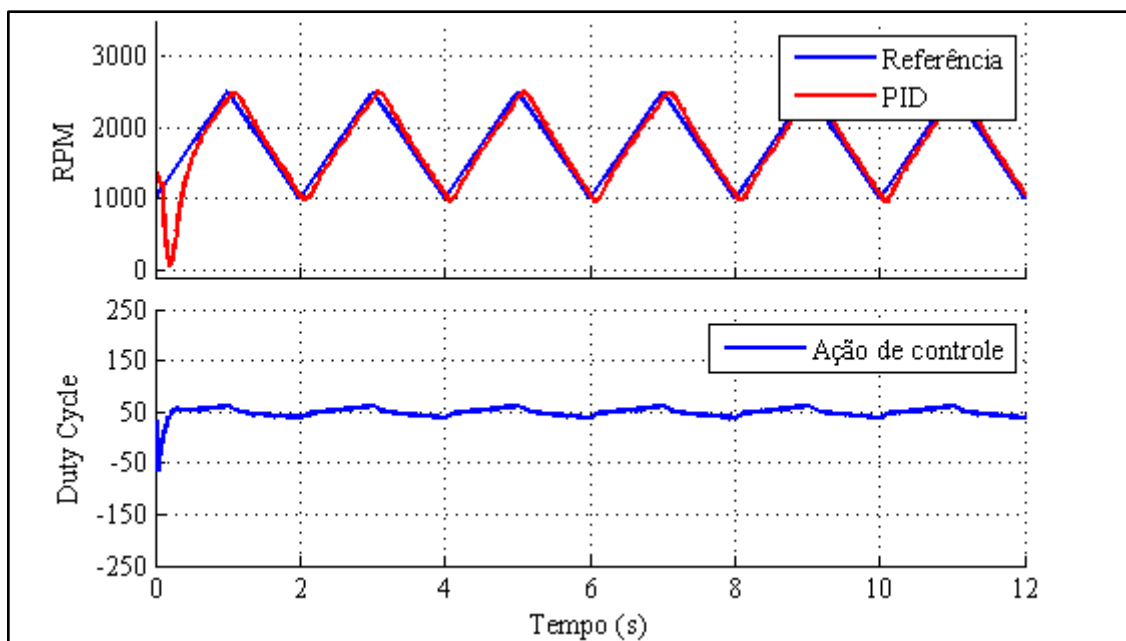


Figura 42 - Entrada trapezoidal de 0,5Hz no controlador PID
 Fonte: Autoria Própria

A Figura 42 mostra o comportamento da bancada à uma entrada trapezoidal de 0,5Hz com amplitude de 1500 RPM e sem inversão do sentido de rotação. O sistema apresentou bons resultados.

A Figura 43 demonstra a resposta do sistema em relação à variação da carga no gerador CC. A velocidade do motor é fixada em 1500 RPM e no instante 0 segundos, é acionada a carga 1, no instante de 4 segundos, a carga 2, no instante de 8 segundos, a carga 3 e por fim, no instante de 12 segundos é acionado a carga 4.

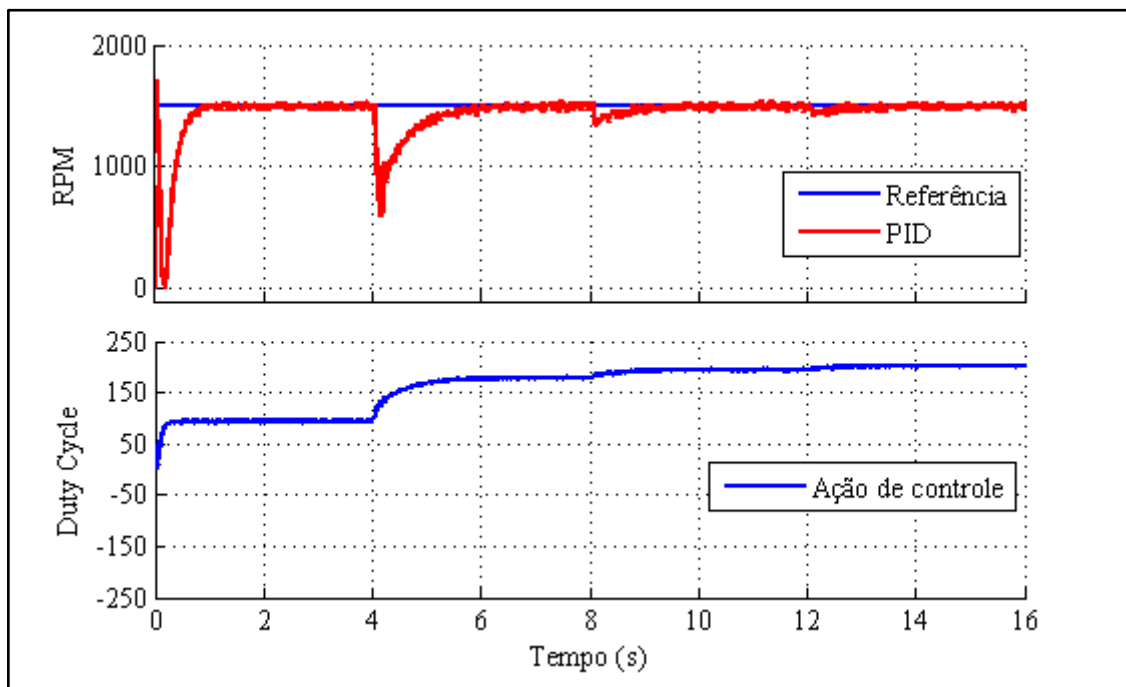


Figura 43 - Resposta do sistema a variação de carga com controlador PID
Fonte: Autoria Própria

Nota-se pela figura que o valor de *duty cycle* aumenta conforme o número de cargas acionadas, conforme previsto, pois quanto maior a carga no gerador, maior a energia necessária para manter uma velocidade no motor CC. A terceira e quarta carga, houve pouca diferença na variação de *duty cycle*. Isto se deve aos valores de resistências utilizados e tem-se a hipótese que utilizando valores de resistências diferentes e maiores tende-se distinguir os valores de carga geradas por cada resistor e assim pode-se melhorar a capacidade de análise das influencias do gerador CC, o que pode ser uma sugestão para trabalhos futuros.

4.2 CONTROLE DE VELOCIDADE UTILIZANDO FUZZY

No software *MATLAB/Simulink* há uma biblioteca denominada de "*Lógica Fuzzy Toolbox*" que contém ferramentas para criar, editar e implementar um controlador Fuzzy de forma simplificada. A biblioteca permite a implementação de regras através do de um editor de regras e implementar este sistema de controle através do bloco "*Fuzzy Logic Control*".

Neste experimento foi utilizada a base de regras sugerida por MacVicar-Whelan como apresentada anteriormente. A Figura 44 mostra as regras básicas implementadas a partir do editor de regras e também as curvas de interferência destas regras.

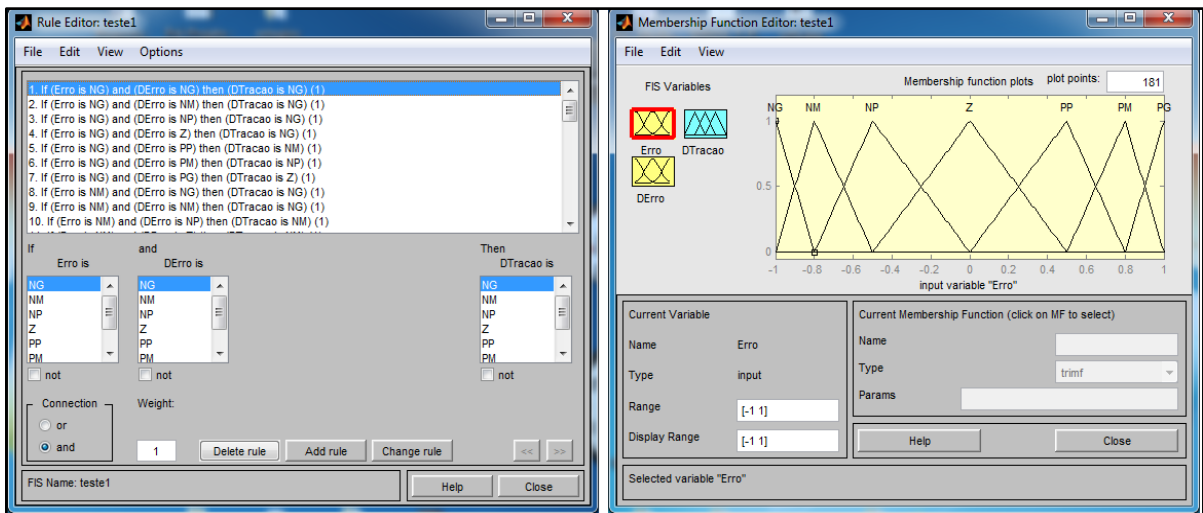


Figura 44 - Base de Regras Fuzzy
Fonte: Autoria Própria

A Figura 45 apresenta o sistema de controle Fuzzy implementado na bancada didática.

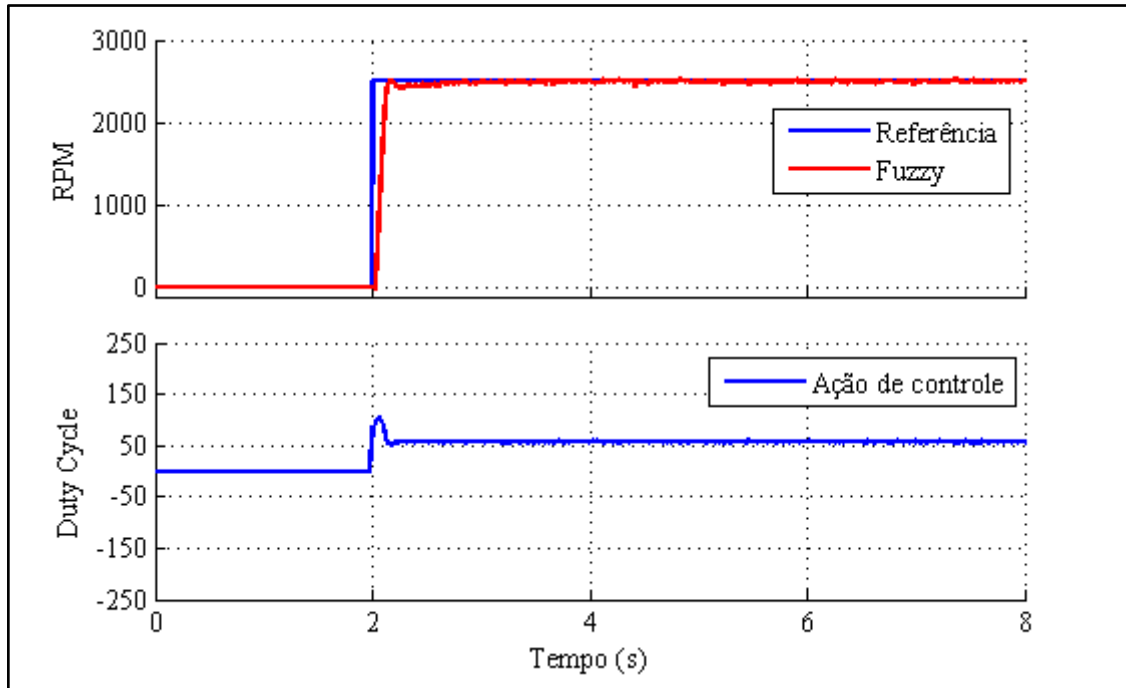


Figura 46 - Entrada degrau de 2500RPM a 2 segundos com controlador Fuzzy
 Fonte: Autoria Própria

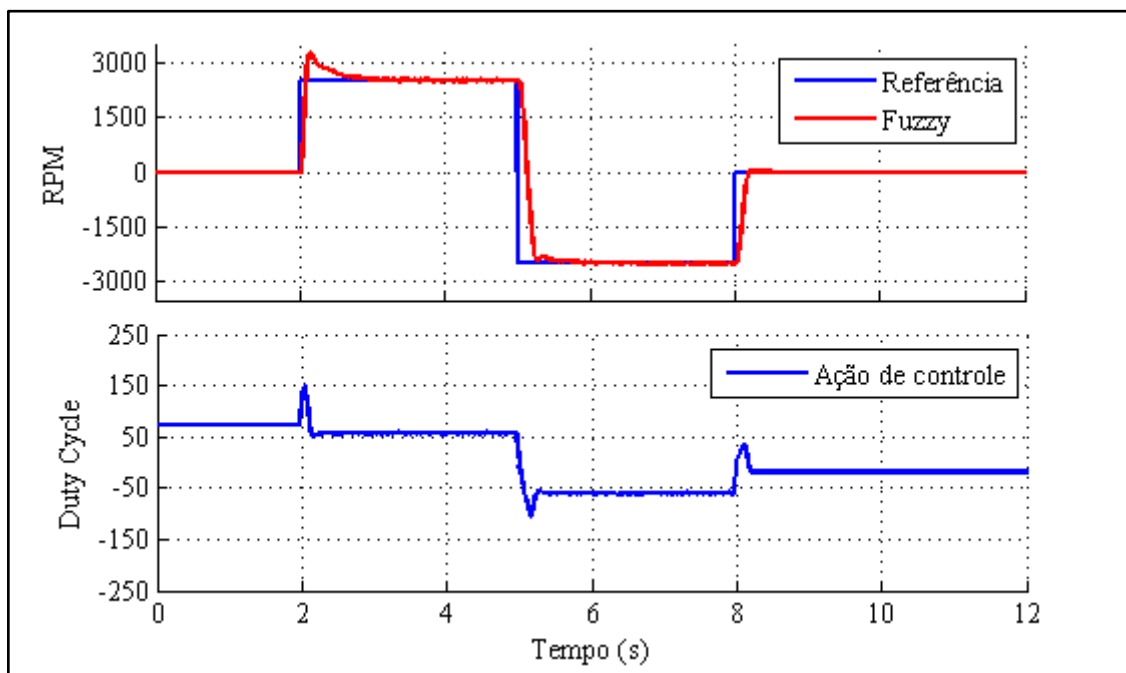


Figura 47 - Entrada degrau de 2500RPM a 2 segundos e inversão de velocidade com controlador Fuzzy
 Fonte: Autoria Própria

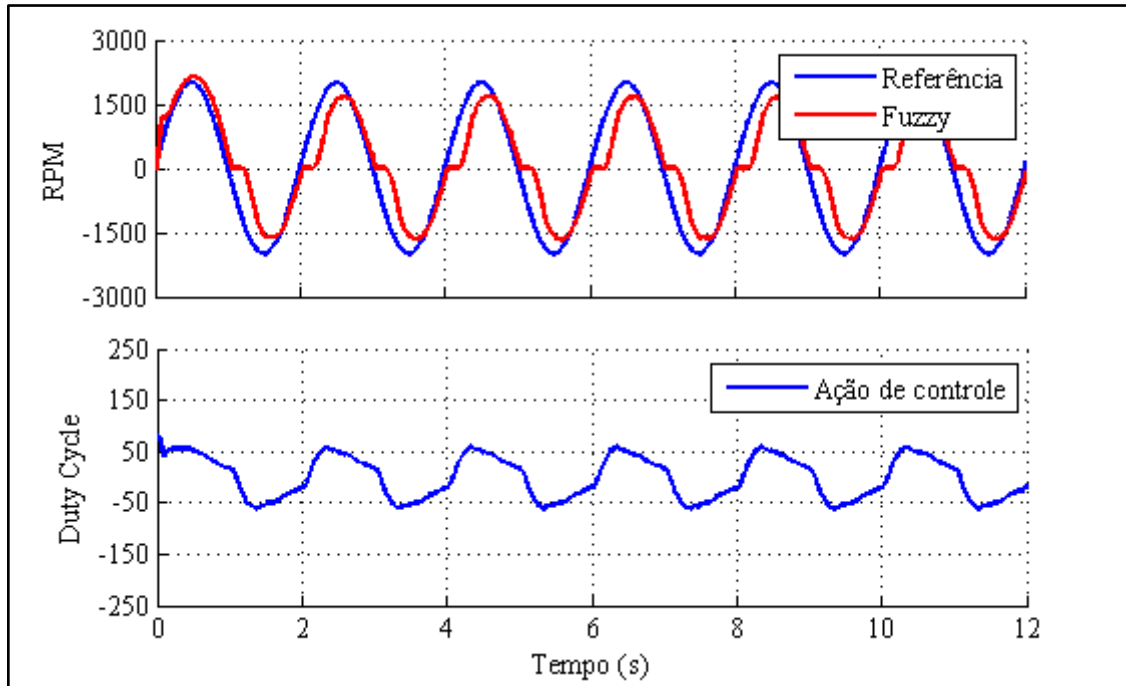


Figura 48 - Entrada senoidal de 0,5Hz com inversão de sentido no controlador Fuzzy
Fonte: Autoria Própria

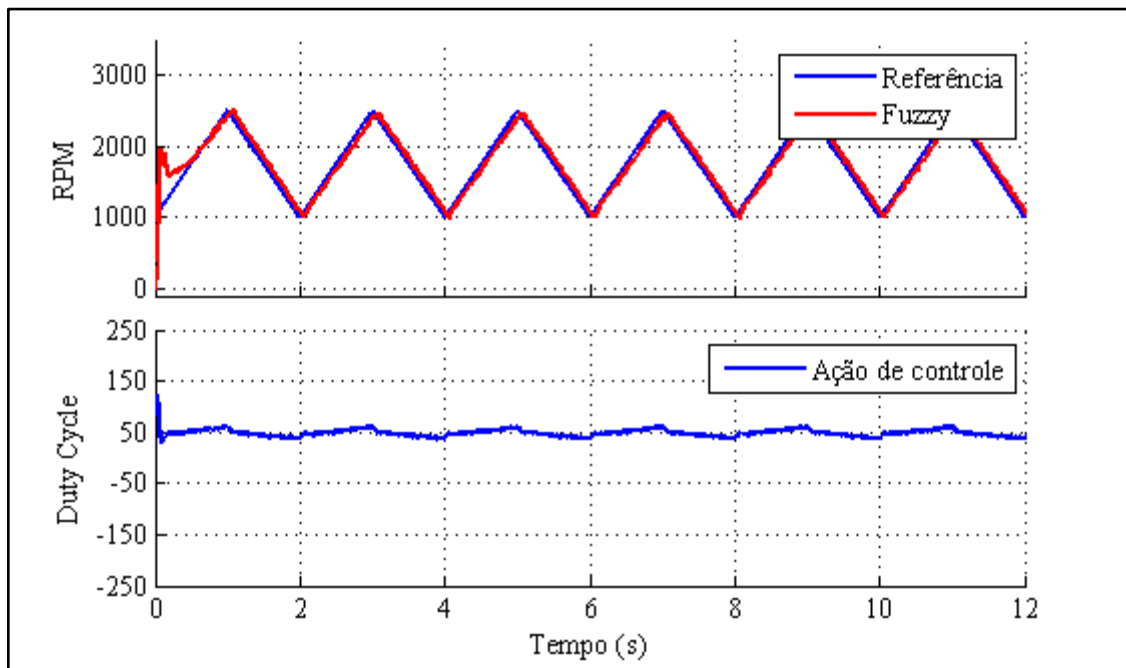


Figura 49 - Entrada trapezoidal de 0,5Hz no controlador Fuzzy
Fonte: Autoria Própria

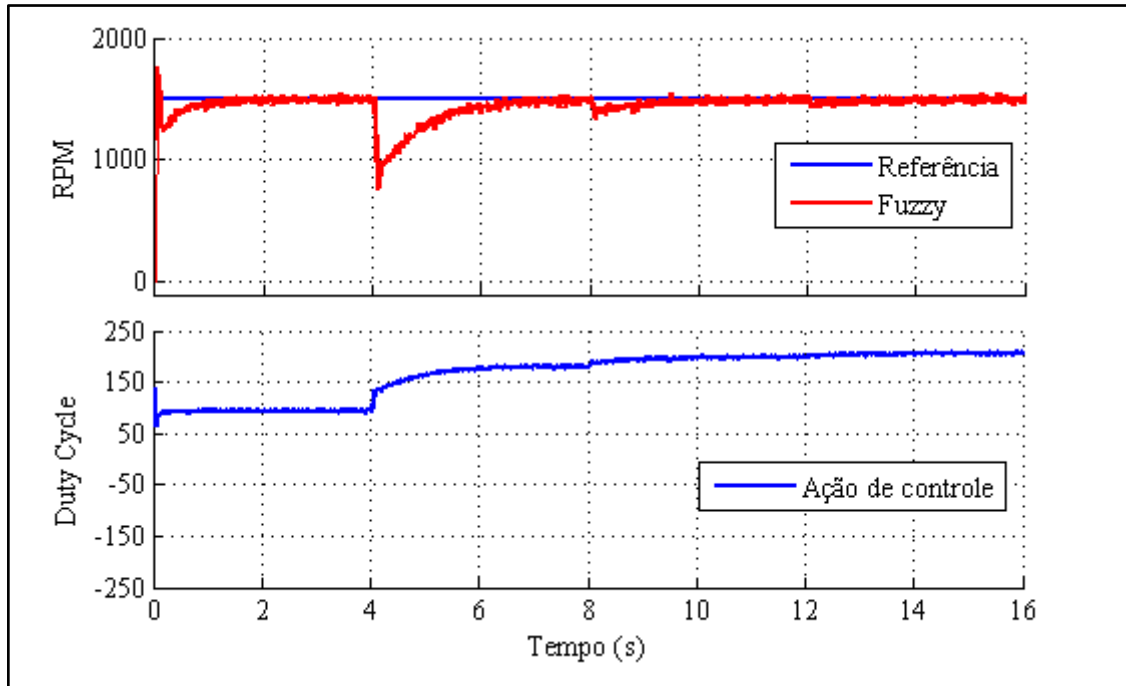


Figura 50 - Resposta do sistema a variação de carga com controlador Fuzzy
Fonte: Autoria Própria

Analisando os testes realizados com o controlador Fuzzy PD+I em comparação com o controlador PID convencional, ambos apresentaram resultados semelhantes, tanto em termos de resposta dinâmica do motor CC às diversas entradas solicitados, quanto em termo de funcionalidade da bancada didática.

4.3 CUSTO DO PROJETO

O valor aproximado total para desenvolvimento da bancada didática, listando os principais componentes utilizados, é mostrado pela Tabela 8.

Tabela 8 – Custos de projeto	
(continua)	
Componente	Valor
Motor CC Pitman com Encoder	R\$ 300,00
Gerador CC genérico	R\$ 10,00
Correia	R\$ 5,00
dsPIC 30F2010	R\$ 30,00

Tabela 8 – Custos de projeto**(continuação)**

L298	R\$ 11,00
FTDI FT232	R\$ 15,00
4 x Relés 5V	R\$ 10,00
4 x Resistores 4R7 10W	R\$ 4,00
Conector USB Fêmea	R\$ 7,00
Cabo USB	R\$ 5,00
Placa de cobre 30x20cm	R\$9,00
Outros (conectores, resistores, capacitores, LED, parafusos, estanho, diodos, etc.)	R\$ 100,00
Total aproximado	R\$ 506,00

O custo total aproximado do projeto ficou em torno de R\$ 506 reais. Os valores foram cotados através da média de valores entre lojas físicas e lojas virtuais em sites de internet. Este valor, quando compara a preço de bancadas didáticas comercializáveis, é um preço acessível em termo de custo, visto que grande parte das bancadas chegam a custar milhares de reais, pois normalmente envolvem sistemas importados e placas de aquisição de preço elevado.

5 CONSIDERAÇÕES FINAIS

Após término dos experimentos, a bancada didática atendeu os objetivos para o ensino de sistemas de controle aplicado a um motor de corrente contínua de imã permanente.

A bancada didática foi idealizada, para ter uma interface simplificada, de custo baixo e de fácil utilização para que qualquer pessoa que deseje estudar sistemas de controle, possa utiliza-la sem que seja necessário um conhecimento aprofundado de sistemas de aquisições.

Deste modo, a bancada tornou-se um sistema versátil, permitindo a alteração da técnica de controle, sem que seja necessário alterar as configurações do sistema de aquisição, ou seja, as variáveis de entradas e saídas da bancada de motor CC estão disponíveis através de blocos do *MATLAB/Simulink*, que contém o acesso a todas as variáveis já condicionadas para o uso. Assim, é possível implementar diferentes técnicas de controle, beneficiando-se das facilidades de programação oferecidas pelo *MATLAB/Simulink*, sem que seja necessário despender novamente tempo na elaboração de um novo sistema de aquisição de dados e controle.

Os experimentos com uso do controlador PID e com controlador Fuzzy para o controle de velocidade do motor CC, mostrou-se eficiente quando sujeito às variações de referência e carga do gerador. Além disto, o grande benefício é a implementação de uma um sistema de controle sobre uma planta real.

O projeto também possibilitou a integração de conhecimentos multidisciplinares tais como: programação de microcontroladores, projeto de sistema de controle, desenvolvimento de protocolos de comunicação, fabricação de placas PCI, utilização de comunicação serial entre hardwares, entre outros.

Para trabalhos futuros, sugere-se a implementação do controle de posição do motor CC na mesma interface de desenvolvimento, realizar a modelagem matemática do motor CC e implementar outros sistemas de controle, tais como: Controladores Robustos, Redes Neurais Artificiais, Mapas cognitivos, Neuro-Fuzzy, entre outros. Propõe-se também a alterar a escala de valor do *duty cycle* de PWM, que foi utilizado a faixa de valores entre 0 e 254, alterando para valores em porcentagem de 0 a 100%, de maneira a tornar a visualização mais intuitiva por outros utilizadores da bancada.

REFERÊNCIAS

AGUIAR, M. L. **Notas de Aula da Disciplina de Controle de Máquinas Elétricas.**

Programa de Pós-graduação em Engenharia Elétrica. Universidade de São Paulo, Escola de Engenharia de São Carlos, São Carlos, São Paulo, 2004.

AUTONICS SENSORS & CONTROLLERS. **Sensors Encoders.** 2011. Disponível em <http://www.autonics.com/products/products_2.php?big=01&mid=01/06>. Acesso em 13 mai. 2016.

BERSON, David. **C What Happens:** using PIC microcontroller and The CCS C compiler. 1 ed. Hayden: Square Eletronics, 2008.

BERTACHI, Arthur H. **Controle de servomecanismo utilizando microcontrolador PIC.** 2011. 104f. Trabalho de conclusão de curso (Engenharia Elétrica) – Universidade Tecnológica Federal do Paraná, Cornélio Procópio, 2011.

BRAGA, Newton. **Como funciona o motor de corrente contínua (ART476).** Disponível em <<http://www.newtonbraga.com.br/index.php/como-funciona/3414-art476a>>. Acesso em 13 mar. 2016.

_____. **O que é PWM (MEC071).** Disponível em <<http://www.newtonbraga.com.br/index.php/robotica/5169-mec071a>>. Acesso em 13 mar. 2016.

BUTTERWORTH, Stephen. Theory of Filter Amplifiers. **Experimental wireless and the Wireless Engineer.** v. 7, p. 536–541, 1930.

DORF, Richard C; BISHOP, Robert H. **Sistemas de controle modernos.** 8.ed. Rio de Janeiro: LTC, 2001.

DYNAPAR ENCODERS. **Para que serve um encoder?.** Disponível em <<http://www.dynaparencoders.com.br/perguntas-frequentes>>. Acesso em 13 mar. 2016.

ETEC. **Apostila de Automação Industrial II.** Disponível em <http://www.trajano.com.br/wp-content/uploads/2012/05/Controle_e_-Automacao_Industrial_II.pdf>. Acesso em 13 mar. 2016.

FITZGERALD, A. E. et al. **Máquinas elétricas:** com introdução à eletrônica de potência. 6. ed. Porto Alegre: Bookman, 2006.

FTDI, Chip. **Ft232R USB Uart IC.** 2.13 ed. Disponível em <http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT232R.pdf>. Acesso em 05 nov. 2015.

GUIMALHAES, Daniel V. **Kit didático para o aprendizado de controladores em tempo real utilizando MATLAB e Toolbox Realtime Workshop**. 2011. 70f. Trabalho de Conclusão de Curso (Engenharia Elétrica) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2011.

JABO, Nurul I. B. P. **Speed control of dc motor using pid controller implementation with visual basic**. 2008. Thesis (Bachelor of Electrical Engineering). Faculty of Electrical & Electronics Engineering. Universiti Malaysia Pahang, 2008.

KIM, Bong J. et al. Design of fuzzy PD + I controller for tracking control. **American Control Conference**, Anchorage, v. 3, p. 2124 - 2129, May 2002.

MACVICAR-WHELAN, P. J. Fuzzy Sets for Man-Machine Interaction. **International Journal of Man-Machine Studies**, Toulouse, v. 8, p. 687-697, June 1976.

MATHWORKS. **Simulate And Test Simulink Models With Hardware In Real Time**. Disponível em <<http://www.mathworks.com/discovery/real-time-simulation.html>>. Acesso em: 08 abr. 2016.

MECATRONICA ATUAL. **A técnica de Hardware-in-the-loop no auxílio de projetos mecatrônicos**. Ano 6, 35 ed, ago./set. 2007. Disponível em <<http://www.mecatronicaatual.com.br/educacao/770-a-tnica-de-hardware-in-the-loop-no-auxlio-de-projetos-mecatrnicos>>. Acesso em 30 jun. 2016.

_____. **Encoders**. Disponível em <www.mecatronicaatual.com.br/educacao/1027-encoders>. Acesso em 13 mar. 2016.

MICROCHIP. **dsPIC30F2010 DataSheet: high-performance 16-bit digital signal controllers**. 2011. Disponível em <<http://ww1.microchip.com/downloads/en/DeviceDoc/70118e.pdf>>. Acesso em 05 nov. 2015.

_____. **Quadrature E coder Interface (QEI)**. 2007. Disponível em <<http://ww1.microchip.com/downloads/en/DeviceDoc/70208A.pdf>>. Acesso em 14 mai. 2016.

OGATA, Katsuhiko. **Engenharia de Controle Moderno**. 3. ed. Rio de Janeiro: Prentice Hall, 1998.

OLIVEIRA, Carlos M. R. **Interface gráfica aplicada ao controle de velocidade de um motor de corrente contínua**. 2013. 80f. Trabalho de conclusão de curso (Bacharel em Engenharia Elétrica) - Coordenação Acadêmica de Eletrotécnica, Universidade Tecnológica Federal do Paraná, Cornélio Procópio, 2013.

OLIVEIRA, Vilma A. et al. **Sistema de controle**: aulas de laboratório. 2 ed. São Carlos: EESC/USP, 2013.

PASSINO, Kevin M.; YURKOVICH, Stephen. **Fuzzy control**. Menlo Park: Addison Wesley Longman, 1997.

PATANÉ, Edson. J. **Implementação de controle de velocidade em malha fechada para motores de corrente contínua utilizando sistema de aquisição de dados**. 2008. 127f. Dissertação (Mestre em Engenharia de Processos Químicos e Bioquímicos) - Escola de Engenharia de Mauá, Centro Universitário do Instituto Mauá de Tecnologia, São Caetano do Sul, 2008.

PEREIRA, Fábio. **Microcontroladores PIC**: programação em C. 2 ed. São Paulo: Érica, 2003.

QUANSER. **Real-time control experiments for education and research**. Disponível em <<http://www.quanser.com>>. Acesso em 12 dez. 2015

SIEMENS. **Motores de Corrente Contínua**: guia rápido para uma especificação precisa 2006. Disponível em <http://www.deg.ufla.br/site/_adm/upload/arquivos/mot_orcc-siemens.pdf>. Acesso em 18/09/2013.

SOUZA, Vitor. A. **Programação em C para o dsPIC**: fundamentos. São Paulo: Ensino Profissional, 2008.

STMICROELECTRONICS. **Datasheet L298**: dual-bridge driver. 2000. Disponível em <https://www.sparkfun.com/datasheets/Robotics/L298_H_Bridge.pdf>. Acesso em 14 mai. 2016.

TAMAGAWA, Seiki. **Rotary Encoders Fa-Coder®**. Disponível em <<http://www.tamagawa-seiki.com/english/encoder/>>. Acesso em 14 mai. 2016

APENDICE A – CÓDIGO DE FONTE DO *FIRMWARE*

```

////////////////////////////////////
//
//   Descrição de Pinos
//       B0 = Leitura do Sensor de corrente
//       B1 = Direcao 1 -IN1
//       B2 = Direcao 2 -IN2
//       B4 = Encoder A
//       B5 = Encoder B
//       D0 = Enable
//       D1 = PWM
//       E0 = Carga 1
//       E1 = Carga 2
//       E2 = Carga 3
//       E3 = Carga 4
////////////////////////////////////
#include <30F2010.h>
#fuses HS,NOWDT,NOBROWNOUT ,NOMCLR
#use delay(clock=7372800)
#use rs232(baud=115200, UART1A, xmit=PIN_C13,rcv=PIN_C14)

//////////////////////////////////Variaveis Globais //////////////////////////////////
int i;

//////////////////////////////////Variáveis PWM //////////////////////////////////
#define PRx 3685 //Periodo do PRx para estouro do timer
#define Divisao_timer TMR_DIV_BY_1
int iniciopwm=35;

//////////////////////////////////Variáveis para envio //////////////////////////////////
int32 valor_enviar=0;
int32 valor_enviar2=0;
int32 valor_enviar3=0;
int32 valor_enviar4=0;

int32 valor_encoder=0;
int32 valor_encoder2=0;
int16 temp_prog=0;
int32 valor_corrente=0;
int buffer_r[4], buffer[4];

//////////////////////////////////Variáveis para recepção //////////////////////////////////
int32 valor;
int32 pwm;
int32 cargas;
int32 carga1;
int32 carga2;
int32 carga3;
int32 carga4;
int32 enable;
int32 direcao;
boolean n;

//////////Para calculo de velocidade //////////////////////////////////
int passos16=0;
int32 passos32=0;
int32 tempo=0;

int Flag=0;
int Contador=0;
unsigned char *Ponteiro;
unsigned char Texto[4];
unsigned char Recebido;
int led;

```

```

#int_rda
void serial_isr()
{
    Recebido = getc();
    if (Recebido == '{')
    {
        Flag=1;
        Ponteiro = &Texto;
    }
    else if (Flag==1)
    {
        *Ponteiro = Recebido;
        Ponteiro++;
    }
    if (Recebido == '}')
    {
        Flag=0;
        buffer_r[0]=Texto[0];
        buffer_r[1]=Texto[1];
        buffer_r[2]=Texto[2];
        buffer_r[3]=Texto[3];

        buffer[0]= valor_enviar3 & 0b00000000000000000000000011111111;
        valor_enviar3 = valor_enviar3>>8;
        buffer[1]= valor_enviar3 & 0b00000000000000000000000011111111;
        valor_enviar3 = valor_enviar3>>8;
        buffer[2]= valor_enviar3 & 0b00000000000000000000000011111111;
        valor_enviar3 = valor_enviar3>>8;
        buffer[3]= valor_enviar3 & 0b00000000000000000000000011111111;

        putc(buffer[0]);
        putc(buffer[1]);
        putc(buffer[2]);
        putc(buffer[3]);
    }
}

void main()
{
    ////////////Configuração PWM ////////////
    //Configura Timer 2 como interno e prescaler 64, e Timer2 máximo
    setup_timer2(TMR_INTERNAL | Divisao_timer,PRx);
    // Comparador 1, modo PWM , base de tempo TIMER2
    setup_compare(2, COMPARE_PWM | COMPARE_TIMER2);

    ////////////Configuração Encoder e Timer para
    //calculo de Velocidade, Taxa de amostragem 50Hz//////////
    setup_qei(QEI_MODE_X4 |QEI_TIMER_INTERNAL);
    qei_set_count(0);
    //prescaler dividido por 1 e clock interno
    setup_timer1(TMR_INTERNAL| TMR_DIV_BY_8);
    set_timer1(0); //(47150); //inicia o timer 0 em 1311

    ////////////Configuração da leitura analogica//////////
    setup_adc_ports(sAN0); //Porta B0
    setup_adc(ADC_CLOCK_DIV_2);
    set_adc_channel(0);
    delay_us(20);
    ////////////////////////////////////////////
    delay_ms(10);

    while (TRUE)
    {
        enable_interrupts(int_rda);
        enable_interrupts(intr_global);

        ////////////PARTE DE ENVIO PARA O SERIAL ////////////

        valor_enviar=0; //Limpa o valor a enviar
    }
}

```

```

////////////////////////////////////
tempo=get_timer1();
passos16=qei_get_count();
set_timer1(0);
qei_set_count(0); //para velocidade
passos32=passos16;
valor_encoder = (6750*passos32)/tempo; //para velocidade
//valor_encoder=passos32; //para controle de posicao
if (valor_encoder>=65535) {valor_encoder=65535;}
valor_encoder2 = valor_encoder << 16;
valor_enviar2 = valor_encoder2 | valor_enviar;

////////////////////////////////////
valor_corrente = read_adc();//0b1111000011110000; //valor de 1023
valor_corrente = valor_corrente >> 6;
//Garantir que o valor da corrente nunca ultrapasse 10bits, apesar da variavel ser de 16bits
if (valor_corrente>=65535) {valor_corrente=65535;}
valor_enviar3 = valor_corrente | valor_enviar2;

////////////////////////////////////PARTE DO RECEBIMENTO DE DADOS////////////////////////////////////

valor= buffer_r[0]+256*buffer_r[1]+65536*buffer_r[2]+16777216*buffer_r[3];

////////////////////////////////////Valor PWM////////////////////////////////////
pwm = valor & 0xFFFF0000;
pwm = pwm >> 16 ;

//Dead band
//pwm= pwm*0.92+20;
//if (pwm <iniociopwm) {pwm=iniociopwm;}
//if (pwm >254) {pwm=254;}

pwm=((PRx+1)*pwm)/255; ///Mapeia o PWM para o valor Maximo de estouro
set_pwm_duty(2,pwm); ///Seta o Valor do PWM

////////////////////////////////////Valor Direcao////////////////////////////////////
direcao = valor & 0x20;
direcao = direcao >> 5;

if (direcao==0)
    {output_high(PIN_B1);
    output_low(PIN_B2);}
else
    {output_low(PIN_B1);
    output_high(PIN_B2);}

////////////////////////////////////Valor Enable////////////////////////////////////
enable = valor & 0x10;
enable = enable >> 4;

if (enable==0)
    {output_low(PIN_D0);}
else
    {output_high(PIN_D0);}

////////////////////////////////////Valor Cargas////////////////////////////////////
cargas= valor & 0xF;
carga1= cargas & 0b1000;
carga1 = carga1 >> 3;
if (carga1==1)
    {output_low(PIN_E3);}
else
    {output_high(PIN_E3);}

carga2= cargas & 0b0100;
carga2 = carga2 >> 2;
if (carga2==1)
    {output_low(PIN_E2);}
else
    {output_high(PIN_E2);}

```

```
carga3= cargas & 0b0010;
carga3 = carga3 >> 1;
if (carga3==1)
    {output_low(PIN_E1);}
else
    {output_high(PIN_E1);}

carga4= cargas & 0b0001;
if (carga4==1)
    {output_low(PIN_E0);}
else
    {output_high(PIN_E0);}
}
```

```
}
```