

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA
CURSO SUPERIOR DE TECNOLOGIA EM MECATRÔNICA INDUSTRIAL

EDUARDO FIGUEIREDO PRADO

SISTEMA DE CONTROLE DE MOTOR PARA INSTRUMENTOS
CIRÚRGICOS

TRABALHO DE CONCLUSÃO DE CURSO

CURITIBA

2011

EDUARDO FIGUEIREDO PRADO

**SISTEMA DE CONTROLE DE MOTOR PARA INSTRUMENTOS
CIRÚRGICOS**

Trabalho de Conclusão de Curso de graduação, apresentado à disciplina de Trabalho de Diplomação, do Curso Superior de Tecnologia em Mecatrônica Industrial do Departamento Acadêmico de Eletrônica – DAELN – e do Departamento Acadêmico de Mecânica – DAMEC – da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para obtenção do título de Tecnólogo.

Orientador: Prof Dr. Gilson Yukio Sato

CURITIBA

2011



Ministério da Educação

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Departamentos Acadêmicos de Eletrônica e de Mecânica



EDUARDO FIGUEIREDO PRADO

SISTEMA DE CONTROLE DE MOTOR PARA INSTRUMENTOS CIRÚRGICOS

Esta Monografia foi julgada e aprovada como requisito parcial para obtenção do título de Tecnólogo em Mecatrônica Industrial pela Universidade Tecnológica Federal do Paraná.

Curitiba, 08 de julho de 2011.

Prof. D. Eng. Milton Luiz Polli

Coordenador de Curso

Departamento Acadêmico de Mecânica

Prof. Décio Estevão do Nascimento

Responsável pelo Trabalho de Conclusão de Curso

Departamento Acadêmico de Eletrônica

BANCA EXAMINADORA

Prof. Dr. Gilson Yukio Sato (Orientador)

Prof. Dr. Bertoldo Schneider Jr.

Prof. M. Sc. Guilherme



UTFPR / DAMEC/ DAELN

Av. Sete de Setembro, 3165

80.230-901

Curitiba Paraná Brasil

Telefone: (41)3310-4616

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

AGRADECIMENTOS

- À **Deus** – pela Graça e Amor. Não apenas por dar-me forças e capacitar-me para chegar até aqui, mas por ser a razão e motivação para tal.
- À minha família – que sempre me apoiou e incentivou. Ao meu pai, minha mãe e minha irmã, por serem exemplos em minha vida, sendo muito importantes em toda minha caminhada.
- À minha namorada – que está sempre ao meu lado, apoiando-me e incentivando-me mesmo nos momentos difíceis. E além de tudo, por ter me ajudado na elaboração deste projeto.
- À Safesoft Ltda – por me ajudar e disponibilizar tempo e equipamentos para a realização deste projeto. Ao Sr. Edipo Lazo, que me ajudou em cada etapa do projeto.
- Ao professor Dr. Gilson Yukio Sato – pela dedicação e disponibilidade e pela orientação do projeto.
- Aos meus amigos – que me acompanharam durante todo o curso.

RESUMO

PRADO, Eduardo. Sistema de controle de motor para instrumentos cirúrgicos. 2011. 56 f. Trabalho de Conclusão de Curso (Curso Superior de Tecnologia em Mecatrônica Industrial) – Universidade Tecnológica Federal do Paraná. Curitiba, 2011.

Este trabalho apresenta o desenvolvimento de um sistema de controle de motor para instrumentos cirúrgicos. O projeto consiste em um sistema microcontrolado para controle de motor CC com um controle Dedicado. O sistema possibilita o controle de rotação de motores CC de até 40.000 rpm e controle automático de torque. O sistema apresenta as características necessárias para o controle de instrumentos cirúrgicos, tais como controle preciso de rotação e torque e rampas de acionamento. Discute-se o desenvolvimento completo do projeto, desde a elaboração do conceito, passando pelo desenvolvimento do circuito e placas eletrônicas, do programa de controle até a obtenção e interpretação dos resultados. O sistema permitiu conduzir estudos do controle, pois foi possível traçar gráficos de resposta do motor por meio do envio de dados por comunicação serial. Além disso, o trabalho apresenta resultados da resposta do sistema em diversas condições de teste.

Palavras-chave: Instrumento cirúrgico. Sistema microcontrolado. Motor CC. Controle Dedicado.

ABSTRACT

PRADO, Eduardo. A motor control system for surgical instruments. 2011. 56 f. Trabalho de Conclusão de Curso (Curso Superior de Tecnologia em Mecatrônica Industrial) – Universidade Tecnológica Federal do Paraná. Curitiba, 2011.

This report presents the development of a motor control system for surgical instruments. The system implements a Dedicated Adaptive control strategy using a micro-controller to control a DC motor. The system can control speed (up to 40.000 rpm) and torque. It presents characteristics that are required to control motors for surgical instruments, such as precise speed and torque control and actuation ramps. This report discusses the development of the project, from the concept elaboration, passing through electronic circuit and boards, and control program development, to the results and their interpretation. The system allowed me to study the controller by tracing graphics using data sent through serial communication. The report presents the response of the system in several circumstances.

Keywords: Surgical instrument. Micro-controlled system. DC motor. Dedicated control.

LISTA DE FIGURAS

Figura 1 – Motor INTRA LUX K200 e ponteira.....	22
Figura 2 – Diagrama de Blocos do Circuito.....	23
Figura 3 – Circuito da fonte do sistema.	24
Figura 4 – Circuito dos sensores, e acionamento do motor.	25
Figura 5 – Circuito do controle – conexões do PIC.....	26
Figura 6 – Interface: conector, leds e botões.....	27
Figura 7 – Conexão dos displays.....	27
Figura 8 – Chaveamento do motor.	28
Figura 9 – Chaveamento dos displays.	28
Figura 10 – Placas do sistema.	29
Figura 11 – Layout do circuito principal.....	30
Figura 12 – Fluxograma simplificado.....	31
Figura 13 – Exemplo de variação de <i>duty-cycle</i>	33
Figura 14 – Arquitetura do <i>firmware</i>	35
Figura 15 – Decodificação dos segmentos do display.....	37
Figura 16 – Diagrama de blocos da malha de controle.....	40
Figura 17 – Exemplo das áreas de controle.....	42

LISTA DE GRÁFICOS

Gráfico 1 – Resposta ao degrau para 40.000 rpm referente ao Teste 1.	45
Gráfico 2 – Teste 1 alterando a rotação desejada (20.00 – 30.000 – 40.000 – 20.000 rpm)....	46
Gráfico 3 – Resposta ao degrau para 40.000 rpm referente ao Teste 2.	47
Gráfico 4 – Resposta ao degrau para 40.000rpm referente ao Teste 3.	48
Gráfico 5 – Teste 3 variando a rotação desejada (20.000 – 30.000 – 40.000 – 20.000rpm). ..	48
Gráfico 6 – Resposta ao degrau para 40.000 rpm referente ao Teste 4.	49
Gráfico 7 - Teste 4 variando a rotação desejada (20.000 – 30.000 – 40.000 – 20.000rpm). ...	50
Gráfico 8 – Resposta ao degrau para 40.000rpm referente ao Teste 5.	51
Gráfico 9 – Teste 5 variando a rotação desejada (20.000 – 30.000 – 40.000 – 20.000 rpm)...	51
Gráfico 10 – Teste 5 com inversão do sentido de rotação.	52

SUMÁRIO

1 INTRODUÇÃO	9
2 PROBLEMA	11
3 JUSTIFICATIVA	12
4 OBJETIVOS	13
4.1 OBJETIVO GERAL	13
4.2 OBJETIVOS ESPECÍFICOS.....	13
5 MÉTODO DE PESQUISA	14
5.1 ELETRÔNICA.....	14
5.2 PROGRAMAÇÃO.....	14
5.3 CONTROLE.....	15
5.4 ANÁLISE DO SISTEMA DE CONTROLE PROPOSTO.....	15
6 FUNDAMENTAÇÃO TEÓRICA.....	17
6.1 ELETRÔNICA.....	17
6.1.1 MICROCONTROLADOR.....	17
6.1.2 RESISTOR SHUNT	17
6.1.3 INTERFACE SERIAL.....	18
6.1.4 CONVERSOR ANALÓGICO/DIGITAL	18
6.2 MECÂNICA	19
6.2.1 MOTOR DE CORRENTE CONTÍNUA.....	19
6.3 CONTROLE.....	20
6.3.1 CONTROLE SERVO E REGULATÓRIO.....	20
6.3.2 PWM.....	20
6.3.3 MEDIÇÃO DE ROTAÇÃO POR FCEM	21
6.4 PROGRAMAÇÃO.....	21
6.4.1 LINGUAGEM C.....	21
7 DESENVOLVIMENTO.....	22
7.1 DEFINIÇÕES	22
7.2 CIRCUITO ELETRÔNICO.....	23
7.2.1 FONTE.....	24
7.2.2 SENSORES	25
7.2.3 CIRCUITO DE CONTROLE	26
7.2.4 INTERFACE	26
7.2.5 CHAVEAMENTO.....	27
7.2.6 CONECTORES.....	28
7.2.7 PLACAS.....	29
7.3 FIRMWARE.....	30
7.3.1 FUNCIONAMENTO DO SISTEMA.....	31
7.3.2 PROGRAMAÇÃO	33
7.4 CONTROLE	39
7.4.1 IMPLEMENTAÇÃO	41
7.4.2 EFEITOS DAS AÇÕES DE CONTROLE.....	43
8 CONSIDERAÇÕES FINAIS.....	54
REFERÊNCIAS	55
APÊNDICE A – TABELA DE CONVERSÃO DE VALORES PARA TENSÃO.....	56

1 INTRODUÇÃO

Desde o surgimento dos motores elétricos no final do século XIX (KOSOW, 1982), o motor de corrente contínua (motor cc) tem sido amplamente utilizado em diversas áreas, como na robótica, em ferramentas e equipamentos eletrônicos, na automação industrial e até na medicina. Com o avanço da tecnologia, sistemas buscando soluções para novos problemas, assim como maior eficiência na resolução de problemas conhecidos, têm sido desenvolvidos. A complexidade de alguns desses sistemas têm exigido sistemas de controle de motores cc mais precisos e eficientes. Essa exigência tem sido cumprida por meio de uma maior integração entre a mecânica, a eletrônica e a informática.

Por meio da integração de motores cc com circuitos eletrônicos e com *softwares* de controle, tem-se desenvolvido sistemas e equipamentos com controle de motores com maior precisão. Como tais sistemas incluem desde brinquedos e ferramentas até instrumentos cirúrgicos, são necessários sistemas de controle de motor cc de diferentes níveis de complexidade e precisão.

Uma aplicação relativamente recente que tem ganhado espaço no mercado é a utilização de controle de motor cc em instrumentos cirúrgicos para odontologia, mais especificamente em instrumentos para perfuração de ossos. O projeto desse tipo de sistema deve ser cuidadosamente elaborado, pois o sistema será utilizado em cirurgias delicadas (como implantes dentários) que requerem alta precisão (maior que 98%). Cirurgias odontológicas que envolvem perfurações ósseas apresentam dificuldades e complicações tanto para o dentista quanto para o paciente, pois elas são realizadas em áreas de difícil acesso ao instrumental cirúrgico. Quanto maior a rotação do motor do instrumento de perfuração óssea, mais fácil é a perfuração do osso. Além disso, o torque do motor deve ser constante durante o processo, mesmo com a variação da carga (maior esforço para perfurar devido à profundidade ou resistência, por exemplo). Caso contrário, o processo fica mais difícil para o dentista, podendo resultar no fracasso da cirurgia, além de gerar maior dor e desconforto ao paciente (MISCH, 2009).

Desta maneira, para aplicação de um motor cc em instrumentos cirúrgicos é necessário um sistema que proveja controle preciso da rotação com controle de torque do motor e uma resposta rápida à variação de carga (KAVO, 2008). O desenvolvimento deste controle requer métodos para monitorar em tempo real a rotação e a corrente aplicada no motor. A partir disto precisa-se controlar a tensão no motor, para atingir a rotação desejada e

conseguir-se manter a corrente constante, para que o torque do motor, por sua vez, seja constante. Este tipo de controle pode ser implementado por meio de sistemas microcontrolados, por isso é muitas vezes chamado de “ μ Motor” (micro-Motor), principalmente no caso de equipamentos odontológicos.

Apesar de já existirem sistemas que realizam tal controle, no Brasil o tema ainda não foi devidamente explorado. Os equipamentos aqui utilizados normalmente são importados, o que faz com que eles tenham um custo elevado (de 3.000 a 10.000 reais em média). Desta maneira é bastante pertinente o estudo e implementação de um sistema de controle de motor cc para instrumentação cirúrgica no Brasil. O desenvolvimento de um sistema nacional pode reduzir custos, trazer benefícios para cirurgias odontológicas e o domínio da tecnologia envolvida, o que permitiria possíveis aprimoramentos e futuras inovações.

2 PROBLEMA

Cirurgias que envolvem corte e perfuração óssea, como por exemplo, algumas cirurgias odontológicas de implante, são realizadas em regiões bastante delicadas e apresentam vários complicadores que devem ser considerados. Se tais cirurgias não forem realizadas da maneira correta e com instrumentos adequados, elas podem causar dor e desconforto aos pacientes, além de poder acarretar conseqüências mais graves. Caso o instrumento não tenha uma rotação adequada, ou ocorra uma variação na sua rotação ou torque durante a operação, ele pode gerar mais vibrações e ruídos que, por sua vez, podem causar maior dor e aumentar a sensação de desconforto do paciente, além de poder acarretar o fracasso da cirurgia (MISCH, 2009).

Para facilitar a operação, trazer segurança e conforto, causar menos dor e propiciar uma melhor recuperação do paciente, os instrumentos utilizados devem possuir uma rotação elevada (até 40.000 rpm dependendo da operação) e constante, com um torque também constante, para facilitar o corte ou perfuração de ossos (KAVO, 2008). Além disso, um travamento do motor pode ocasionar a ocorrência de erros e acidentes por parte do operador.

Nessas circunstâncias, faz-se necessário o desenvolvimento de um controle preciso para motores de instrumentos cirúrgicos. Precisa-se de um sistema capaz de controlar precisamente a rotação de um motor cc, mesmo em altas rotações, mantendo o torque constante por meio de uma resposta rápida às alterações na carga. Tais sistemas de controle já têm sido estudados e desenvolvidos, principalmente na Europa, por empresas alemãs (como a Kavo, por exemplo). Porém, no Brasil o desenvolvimento desses sistemas de controle não é devidamente explorado e os equipamentos que os utilizam são normalmente importados e de custo elevado.

Em função do contexto exposto, propõe-se desenvolver um sistema nacional de controle de motores cc para instrumentação cirúrgica capaz de suprir às necessidades específicas de tal aplicação, reduzindo custos e possibilitando o domínio de uma tecnologia de interesse ao desenvolvimento industrial regional e nacional.

3 JUSTIFICATIVA

O desenvolvimento de um sistema nacional capaz de controlar precisamente a rotação de um motor cc, mantendo o torque constante, mesmo em altas rotações e com resposta rápida a alterações na carga é pertinente por diversas razões. O estudo e implementação de tal sistema pode, primeiramente, trazer benefícios a profissionais da área de saúde e a pacientes, facilitando e melhorando as cirurgias que envolvem perfuração óssea, diminuindo a dor, o desconforto e as complicações cirúrgicas.

O desenvolvimento de um sistema que facilite e possibilite um estudo mais sistemático deste tipo de controle pode agregar conhecimentos tecnológicos na área de eletrônica, controle e acionamento de motores. Mais especificamente, esse estudo pode contribuir na compreensão de processos como: monitoramento da corrente e da rotação de um motor cc; implementação de um controle contínuo de rotação por meio da modulação por largura de pulso; exploração de alternativas para obter-se uma resposta rápida com sistemas microcontrolados. Além disso, o desenvolvimento de tal sistema pode incentivar um estudo mais aprofundado do assunto em busca de aprimoramentos, novas soluções e melhorias nas diversas áreas envolvidas.

Outra motivação à realização deste projeto é o fato de que esse tipo de sistema de controle ainda não é muito difundido no Brasil. Desta maneira, o desenvolvimento de um sistema nacional pode incentivar a melhor exploração dessa tecnologia e estimular o desenvolvimento de novos equipamentos nacionais, com menores custos e até com melhorias, aprimoramentos e novos recursos.

4 OBJETIVOS

Nesta seção serão especificados o objetivo geral e os objetivos específicos deste projeto.

4.1 OBJETIVO GERAL

Desenvolver um sistema de acionamento e controle de motor cc que seja capaz de controlar precisamente a rotação do motor, mesmo em altas rotações, mantendo constante o torque, independentemente de alterações na carga.

4.2 OBJETIVOS ESPECÍFICOS

Os objetivos específicos para a realização do projeto são:

- Elaborar um circuito eletrônico para o acionamento de motor cc;
- Projetar o *hardware*, prevendo a alimentação do sistema, a placa do circuito, os botões e display;
- Utilizar um microcontrolador de família PIC 18 para controlar o sistema;
- Desenvolver um controle por meio do monitoramento de tensão e corrente.
- Realizar o controle utilizando modulação por largura de pulso (PWM – *Pulse Width Modulation*);
- Implementar rampas de subida e de descida para o acionamento do motor;
- Permitir a seleção da rotação desejada para motor;
- Indicar a rotação do motor por meio de um display;
- Permitir a alteração do sentido de rotação do motor por meio de um botão;
- Enviar dados relevantes do sistema (rotação, corrente, por exemplo) em tempo real via serial para o computador;
- Plotar gráficos a partir dos dados enviados pela serial, e a partir destes, analisar e estudar a atuação do controle;

5 MÉTODO DE PESQUISA

Para a realização do projeto foi necessária a utilização de diversos métodos e técnicas de diferentes áreas. Assim, o projeto como um todo depende da realização e integração das diversas partes.

5.1 ELETRÔNICA

O circuito eletrônico foi desenvolvido com o auxílio de ferramentas computacionais. A confecção do esquemático do circuito e do layout da placa foi feita utilizando o *software* Altium Designer 2008. Primeiramente, foram levantados os requisitos de um sistema de controle para motores cc operando em elevada rotação e que necessitem de rotação e torque constantes. A partir desses requisitos foram desenvolvidos os circuitos que compõe esse sistema. O esquemático do circuito foi realizado levando-se em consideração os componentes utilizados. O esquemático representa, além dos componentes e suas devidas ligações, a forma como foi feita a alimentação do circuito, a ligação do motor cc ao circuito e a utilização dos botões e displays.

Com o esquemático pronto, foi feito o layout para a placa de circuito impresso (PCI). A partir do esquemático, fez-se o layout do posicionamento dos componentes e das respectivas ligações, buscou-se uma utilização eficiente do espaço físico.

Com o layout pronto, imprimiu-se o esquemático na PCI e montou-se o circuito eletrônico. A compra de componentes e materiais foi realizada mediante pesquisas de preço, procurando-se manter uma relação de custo/benefício adequada.

5.2 PROGRAMAÇÃO

O controle do sistema foi realizado por um *firmware* armazenado em um microcontrolador. O *firmware* de controle foi escrito em linguagem C, sendo implementado e compilado utilizando-se os *softwares* Dxp e MPLab. O programa foi estruturado em blocos, em que cada bloco é responsável por uma parte específica do programa, e que se comunicam por meio de variáveis globais.

5.3 CONTROLE

O controle do sistema é realizado por um microcontrolador no qual foi implementado um *firmware* de controle. Para a implementação deste, foi feito primeiramente um estudo aprofundado de como deveria funcionar e atuar o controle do sistema. O controle consiste em variar a tensão aplicada no motor para se atingir a rotação desejada, mantendo-se a corrente constante.

Para isso foi preciso desenvolver técnicas para monitorar em tempo real a tensão e a corrente aplicadas no motor. Para medir a corrente utilizou-se um resistor *shunt* em série com o motor. A tensão sobre esse *shunt* foi condicionada servindo como dado de entrada para o microcontrolador. Já o monitoramento da tensão foi feito por meio da medição da força contra-eletromotriz gerada pelo motor que foi condicionada e serve como entrada para o microcontrolador. Foram necessários conversores analógico/digital para converter esses dois dados (tensão sobre o *shunt* e força contra-eletromotriz gerada pelo motor) fornecidos ao microcontrolador.

A partir dos dados de tensão e corrente recebidos, foi implementado um controle Dedicado, que permite que o sistema microcontrolado atue no motor controlando sua rotação e torque. O sistema requer um controle que apresente respostas satisfatórias tanto para um comportamento servo (mudança de *set-point*), quanto para um comportamento regulatório (distúrbios e alteração de carga), principalmente durante uma cirurgia, quando ocorre constante alteração de carga. A atuação direta do controle no motor foi implementada por meio de modulação por largura de pulso (PWM). O controle também inclui rampas de subida e de descida para o acionamento do motor, evitando partidas e paradas abruptas. As rampas de acionamento do motor também foram implementadas com PWM.

O programa de controle também controla os displays de indicação de rotação e trata os botões para seleção de velocidade e sentido de rotação do motor. Além disso, dados referentes à corrente e tensão no motor são enviados em tempo real via cabo serial.

5.4 ANÁLISE DO SISTEMA DE CONTROLE PROPOSTO

A análise do sistema de controle proposto foi feita por meio dos dados enviados pelo *firmware* do microcontrolador para um programa em um microcomputador pessoal via interface serial. A partir dos dados recebidos, foram plotados gráficos mostrando a ação do controle, podendo-se assim analisar e observar a atuação do controle no sistema.

Para uma análise efetiva do controle por meio dos dados de tensão e corrente, foram realizados testes simulando o funcionamento do sistema. Durante os testes foi simulada alteração da carga no motor, pressionando-se o eixo contra uma superfície. Por meio dos gráficos com dados relevantes do sistema, como a variação da tensão e da corrente em relação ao tempo, foi possível observar características do controle, como tempo de resposta e precisão, e assim obter conclusões significativas sobre o sistema.

6 FUNDAMENTAÇÃO TEÓRICA

A realização do projeto requer um estudo de seus diversos elementos. Abaixo se encontram as referências e teorias que dão o suporte teórico ao projeto.

6.1 ELETRÔNICA

Para a realização deste projeto são necessários vários elementos da área eletrônica, como microcontrolador, resistor *shunt*, conversor analógico digital e interface serial.

6.1.1 MICROCONTROLADOR

Um microcontrolador é um dispositivo que possibilita desenvolver projetos e sistemas com funções dedicadas como, por exemplo, alarmes, controladores lógico programáveis, sistemas de automação, etc. Ele não visa à execução de tarefas genéricas (edição de texto, navegação Internet) como um computador pessoal, mas pode ser programado para executar várias funções específicas, sendo empregado nas mais diversas aplicações.

De acordo com Martins (2005), um microcontrolador consiste basicamente em um encapsulamento que incorpora um microprocessador, uma memória de programa, uma memória de dados, uma série de pinos de entrada/saída e vários periféricos. Assim, o que diferencia um microcontrolador do outro é a quantidade de memória interna tanto para armazenar dados quanto instruções de um programa, a velocidade de processamento, a quantidade de entradas e saídas, a tensão de alimentação, quantidade e tipos de periféricos, a arquitetura e o conjunto de instruções que pode ser utilizado para programação.

Desta maneira pode-se controlar um sistema com um microcontrolador por meio da integração deste com outros componentes e de operações e relações matemáticas estabelecidas em um programa, utilizando os dados das entradas para controlar as saídas.

6.1.2 RESISTOR *SHUNT*

O resistor *shunt* é um sensor de corrente bastante difundido devido a sua boa precisão e baixo custo. Ele é um dispositivo utilizado para medir a corrente que passa em um determinado ponto do circuito.

Segundo Lepkowski (2003), o funcionamento de um resistor *shunt* baseia-se no monitoramento da tensão sobre um resistor de valor conhecido colocado em série com a carga, o que permite determinar a corrente que passa através dela. Se sua resistência for pequena em relação à da carga, a queda de tensão também será, de modo que a medição não terá maiores efeitos sobre o circuito. Contudo, devido à dissipação de potência, resistores *shunt* não devem ser usados para medir correntes muito altas (a partir de aproximadamente 20 A).

6.1.3 INTERFACE SERIAL

A interface serial é uma interface de transmissão de dados, bastante utilizada para conectar diversos dispositivos ao computador, como por exemplo, impressoras e scanners. Na comunicação serial os bits são enviados em série, um após o outro como se eles estivessem em uma fila.

Cada bit é transmitido seqüencialmente e assim, a duração de um bit é definida pelo período e a velocidade é dada em bits por segundo. Existem padrões de bits que são definidos para indicar o início e o fim de uma palavra enviada.

Para indicar o início ou fim de uma palavra utilizam-se os chamados *start-bit* e *stop-bit*, respectivamente bit de início o bit de fim. Assim, o sistema que está recebendo a informação pode identificar quando alguma palavra está chegando e quando a mesma acabou. Isto serve para realizar a sincronização dos dois sistemas que estão conectados, possibilitando a comunicação entre eles (PAZOS, 2002, p. 116).

De acordo com Pazos (2002, p. 116-117), outra característica importante da comunicação serial é o chamado bit de paridade. “Este é uma redundância acrescentada à palavra a transmitir para facilitar a detecção de erros na transmissão”, assim quando ocorrem erros as palavras são reenviadas, diminuindo erros na transmissão de dados.

6.1.4 CONVERSOR ANALÓGICO/DIGITAL

Para que um sistema digital, tal como computadores e microcontroladores, possa ler sinais analógicos, estes devem ser convertidos em sinais digitais. Para isto existe o conversor analógico/digital, ou conversor A/D.

Portanto a função dos conversores A/D é converter sinais do mundo analógico para o mundo digital, de maneira tal que estes possam ser lidos, processados,

monitorados ou armazenados adequadamente por dispositivos digitais (PAZOS, 2002, p. 127).

Assim, consegue-se trabalhar em sistemas digitais com sinais do mundo analógico, como tensão e corrente. Contudo a qualidade e a fidelidade do sinal digitalizado dependem da resolução de conversão e da taxa de amostragem. A resolução é definida pelo número de bits e pela faixa de valores que podem ser aplicados na entrada do conversor A/D. Segundo Pazos (2002), um fator crítico é o tempo de conversão “pois delimita o menor tempo de amostragem possível do sinal de entrada” (PAZOS, 2002, p. 129).

6.2 MECÂNICA

O principal elemento mecânico, e de fundamental importância, utilizado no sistema que foi implementado é o motor de corrente contínua.

6.2.1 MOTOR DE CORRENTE CONTÍNUA

O motor de corrente contínua, ou motor cc, é um tipo de motor elétrico bastante utilizado em diversas aplicações. Um motor elétrico é um dispositivo que transforma energia elétrica em energia mecânica.

Os motores cc são compostos basicamente por duas partes: a armadura e o estator. O estator é a parte fixa que produz um campo magnético constante, por meio de um eletroímã ou um ímã permanente. Já a armadura, também conhecida como rotor, compõe a parte rotatória que possui uma bobina, pela qual circula a corrente. “Quando uma corrente circula através da bobina do rotor, se produz uma força que tende a movimentar os condutores da bobina do rotor (PAZOS, 2002, p. 191)”, esta força pode ser representada pelo produto vetorial $F = l \cdot i \wedge B$, em que a força F em cada segmento do condutor é o produto do comprimento do condutor l e a corrente i elevada pelo campo magnético B .

Uma característica importante do motor cc é que sua rotação é proporcional à tensão nele aplicada e seu sentido de rotação é determinado pela polaridade da fonte. Contudo, sua rotação depende fortemente da carga nele aplicada. Como ele pode ser facilmente acionado aplicando-lhe uma corrente contínua, o motor cc é bastante utilizado em sistemas eletrônicos de modo geral.

6.3 CONTROLE

Para a implementação do controle Dedicado usado no sistema são utilizadas ações de controle dedicadas, modulação por largura de pulso e a medição da rotação do motor por meio da força contra eletromotriz. O controle visa manter o sistema estável tanto em um comportamento servo quanto em um regulatório.

6.3.1 CONTROLE SERVO E REGULATÓRIO

O controle de um processo contínuo consiste essencialmente em estabilizar uma determinada variável em um valor desejado, chamado de *set-point* (OGATA, 1982). Para isto existem basicamente dois problemas a serem considerados: o problema Servo; e o problema Regulatório. O sistema em questão requer um controle que apresente respostas adequadas tanto para o problema Servo quanto para o Regulatório.

O problema Servo consiste basicamente em uma mudança do *set-point* do sistema, de maneira que após esta alteração ele deve se estabilizar no novo valor estabelecido. Já o problema Regulatório se refere às perturbações de carga com um *set-point* constante (SCHNEIDER, 2010).

6.3.2 PWM

A modulação por largura de pulso, ou PWM (*Pulse Width Modulation*), consiste basicamente em variar a largura de uma série de pulsos em função de um sinal de entrada. É esse sinal de entrada que carrega a informação a ser transmitida.

Uma das variações de PWM, como explica Montgomery (2008, p. 21), utiliza um controle por dosagem de tempo. A saída corresponde a uma série de pulsos de liga e desliga, porém o tempo de cada ciclo é fixo e o que varia é a porcentagem do tempo ligado e do tempo desligado, ou seja, a largura dos pulsos.

Assim, controlando a porcentagem do tempo ligado e desligado de cada pulso do sinal, pode-se controlar o valor da alimentação entregue a uma carga, sem que ocorram perdas devido à queda de tensão, como ocorre em sistemas resistivos.

6.3.3 MEDIÇÃO DE ROTAÇÃO POR FCEM

Segundo Rylee (2003, p. 4), um método de baixo custo para medir a rotação de um motor cc é o método de medição da Força Contra-Eletromotriz (FCEM ou BEMF do inglês *Back Electro-Motive Force*) do motor. Ele afirma que a rotação do motor é diretamente proporcional à FCEM.

Uma vez que um motor cc de escovas pode ser modelado como uma carga indutiva, pode-se considerar que a tensão no motor é proporcional a indutância das bobinas. Assim, a FCEM surge quando se corta o sinal do motor, após a aplicação de uma FEM induzida, e as bobinas continuam girando dentro do campo magnético, gerando uma tensão de polaridade inversa (KOSOW, p, 29, 1982).

6.4 PROGRAMAÇÃO

A programação do *firmware* de controle é feita em linguagem C.

6.4.1 LINGUAGEM C

A linguagem de programação C é uma linguagem compilada de propósito geral, que originalmente foi criada para o desenvolvimento do sistema operacional UNIX, por Dennis Ritchie (Schildt, 1996, p. 3).

Segundo Schildt (1996), C é uma linguagem considerada de médio nível, pois combina elementos de linguagens de alto nível com a funcionalidade da linguagem assembly, considerada de baixo nível. O C trabalha com os elementos básicos com os quais computadores, micro-computadores e microcontroladores funcionam, permitindo a manipulação de bits, bytes e endereços.

A linguagem C pode ser considerada uma linguagem estruturada, que permite a compartimentalização do código e dos dados e diversas possibilidades de programação. De acordo com Schildt (1996), o principal componente estrutural da linguagem C é a função:

“Em C, funções são os blocos de construção em que toda a atividade do programa ocorre. Elas admitem que você defina e codifique separadamente as diferentes tarefas de um programa, permitindo, então, que seu programa seja modular.” (SCHILDT, 1996, p. 6).

7 DESENVOLVIMENTO

Nesta seção é descrito o desenvolvimento de cada etapa do projeto até a sua conclusão.

7.1 DEFINIÇÕES

Para iniciar o projeto foi fundamental definir claramente algumas questões, principalmente aquelas relacionadas ao *hardware*, como qual motor seria utilizado e qual microcontrolador seria empregado. Estes elementos foram definidos no início do projeto, pois todo o desenvolvimento do *hardware* e do *firmware* dependia dessas definições.

Primeiramente, foi necessário definir qual motor seria utilizado no sistema. Foi escolhido um motor de corrente contínua que atinge até 40.000 rpm, adequado para aplicações cirúrgicas, o motor INTRA LUX K200 da Kavo (Figura 1). Este é um motor cirúrgico utilizado na odontologia, de pequeno porte e adequado para a aplicação. É um motor de corrente contínua (24VCC), pneumático, que funciona com ar comprimido, feito para atingir até 40.000 rpm.



Figura 1 – Motor INTRA LUX K200 e ponteira.

Fonte: autoria própria.

Definido o motor, outra definição importante foi a do microcontrolador a ser utilizado. O microcontrolador escolhido para o projeto foi o PIC18LF4620 da Microchip. Este

microcontrolador foi escolhido, pois é robusto e possui várias características favoráveis ao projeto. O PIC18LF4620 é um microprocessador de oito bits que possui uma memória *flash* de 64KB, cinco portas totalizando 36 I/Os programáveis, podendo operar com tensão de 3,3 V. Ele possui quatro *timers* internos, dois conversores analógico/digital (A/D) que foram utilizados para a leitura de tensão e de corrente, uma porta com *pull-up* interno e uma rotina interna de PWM (MICROCHIP TECHNOLOGY INC., 2010).

7.2 CIRCUITO ELETRÔNICO

O circuito eletrônico do sistema é dividido basicamente em Fonte, Sensores, Controle, Interface, Chaveamento e Conectores (Figura 2). O circuito foi separado em duas placas: uma placa maior com o circuito principal e outra menor apenas com a interface (displays, leds e botões).

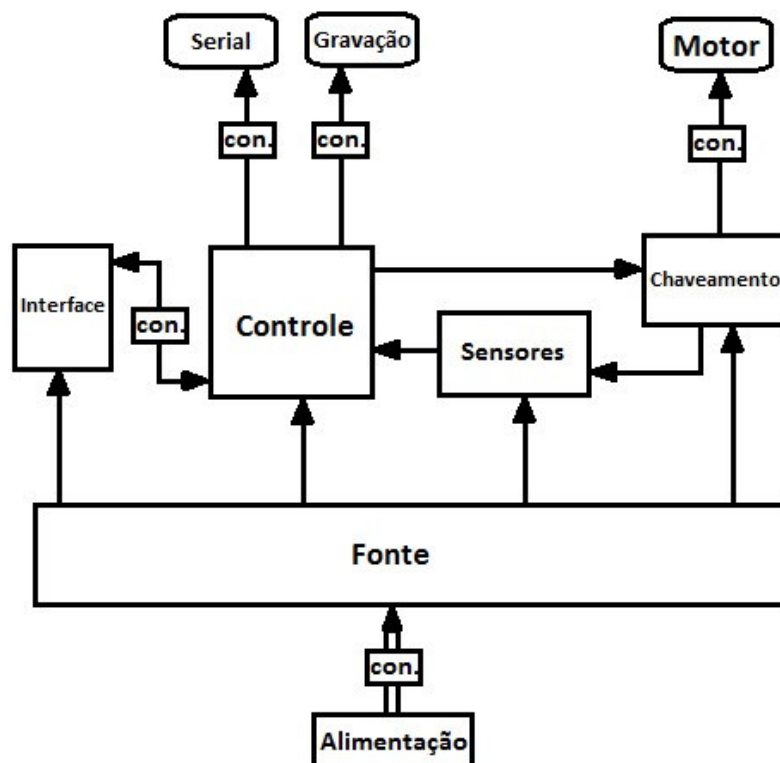


Figura 2 – Diagrama de Blocos do Circuito.

Fonte: autoria própria.

7.2.1 FONTE

Na entrada da fonte é utilizado um transformador que fornece 24 e 12V em corrente alternada (24VAC e 12VAC). Foi necessário desenvolver um circuito que fornecesse ao sistema as seguintes tensões DC: 24V para o acionamento do motor; 12V para o chaveamento do motor e dos displays e para os sensores; 3,3V para a alimentação do PIC e -1,2V para os filtros dos sensores.

De acordo com as necessidades apresentadas foi desenvolvido um circuito (Figura 3) com retificadores utilizando diodos para fornecer 12V, 24V e -1,2V e um regulador para fornecer 3,3V. Para obter-se 24V em corrente contínua foi implementada uma ponte retificadora de onda completa, utilizando-se quatro diodos 1N5408, que são diodos que suportam a corrente exigida pelo motor, que pode chegar à 4A. Já para obter-se 12V foi usada uma ponte retificadora com dois diodos 1N4007. Para se obter o referencial negativo (que deve ser de pelo menos -1,2V), usado nos amplificadores operacionais dos filtros, foram colocados quatro diodos de sinal 1N4148 entre dois pontos de GND com capacitores. Também foi implementado na entrada da fonte um circuito de segurança, com fusível de 6A.

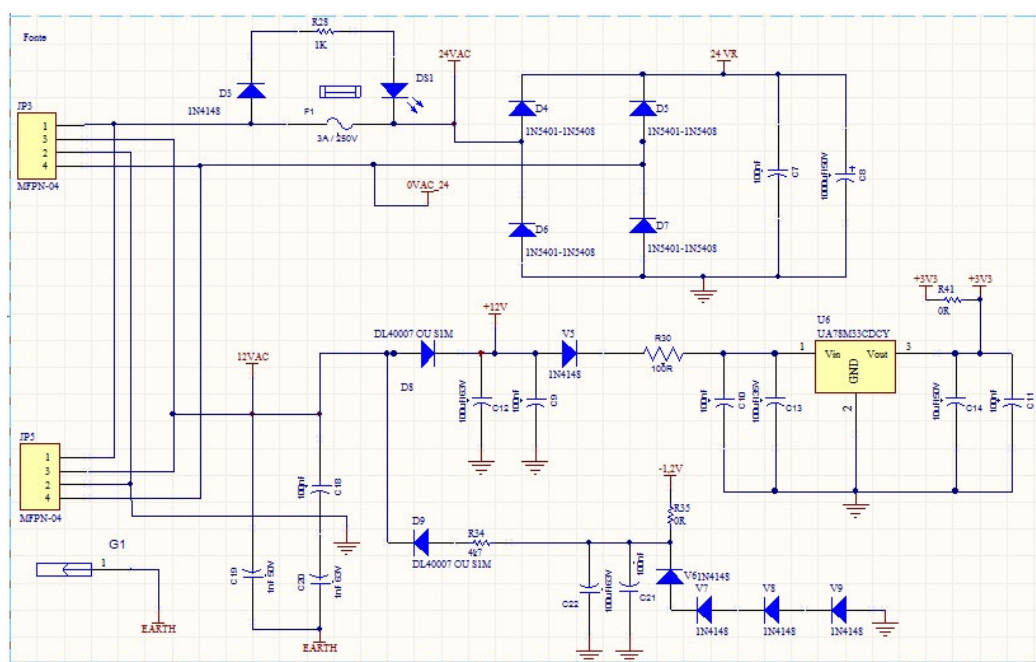


Figura 3 – Circuito da fonte do sistema.

Fonte: autoria própria.

7.2.2 SENSORES

Os sensores do sistema consistem basicamente de dois circuitos que se encontram nas entradas dos conversores A/D do PIC, para a leitura de tensão e corrente do motor.

A leitura da corrente é feita por meio de um resistor *shunt* de $0,1\Omega$ e $5W$ colocado em série com o motor (Figura 4). A leitura da tensão no motor é feita por meio da Força Contra-Eletromotriz (FCEM), medida diretamente no motor (Figura 4). Entre os pontos de medição de corrente e tensão no circuito e as entradas A/D do PIC existem dois filtros passa-baixa, um para o sensor de corrente outro para o sensor de tensão (Figura 4).

O filtro utilizado é conhecido como *Sallen-Key Low-pass filter*, um filtro passa-baixa que possibilita uma alta taxa de atenuação, que serve para atenuar os ruídos do sistema sem interferir no sinal, com o intuito de obter um sinal filtrado para uma medição mais precisa. É um filtro ativo de dois pólos, que consiste de um amplificador operacional com realimentação negativa e com um determinado ganho.

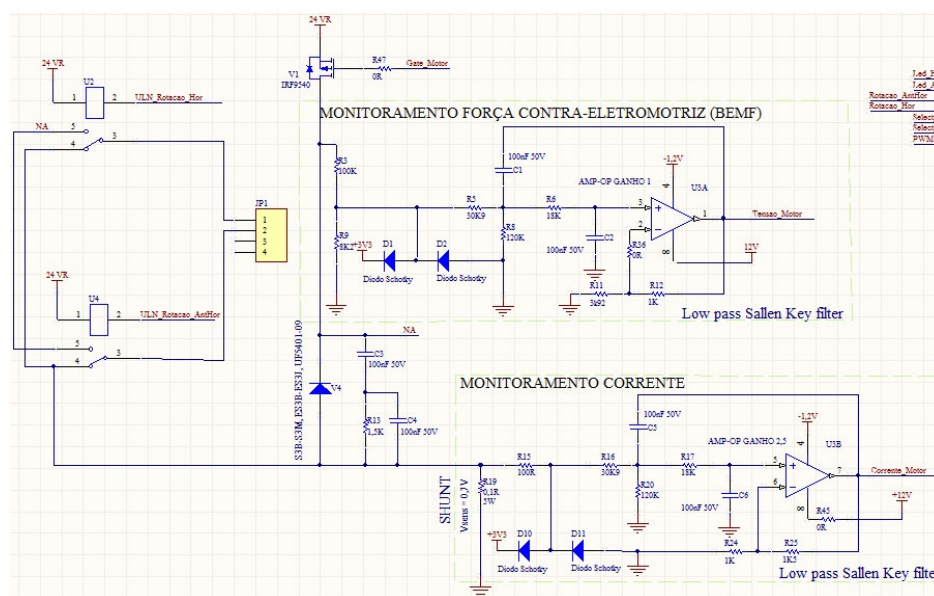


Figura 4 – Circuito dos sensores, e acionamento do motor.

Fonte: autoria própria.

7.2.3 CIRCUITO DE CONTROLE

O circuito de controle é de fundamental importância e seu projeto consistiu basicamente em definir o que estaria conectado em cada pino do microcontrolador. Portanto, para realizar o esquemático foi preciso definir previamente a função de cada porta e o que deveria estar conectado ao PIC. Além disso, foi necessário prever o circuito para a alimentação do PIC e para o oscilador (Figura 5).

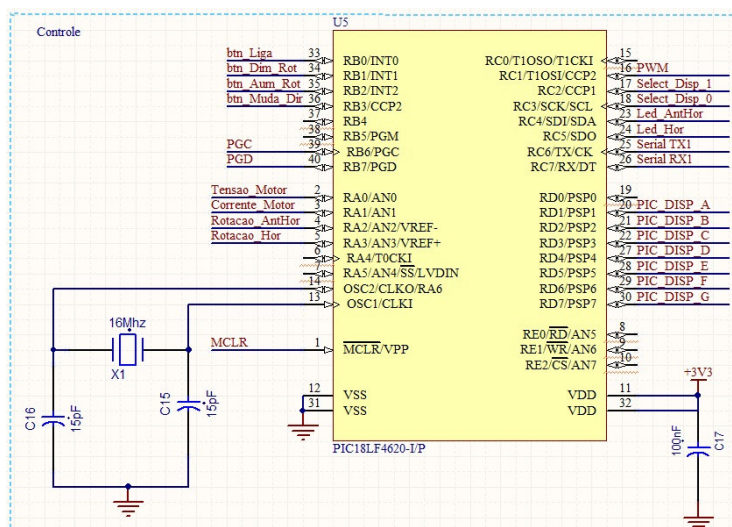


Figura 5 – Circuito do controle – conexões do PIC.

Fonte: autoria própria.

7.2.4 INTERFACE

O circuito da Interface contém os displays, leds e botões que formam a interface do sistema com o usuário. Por ficar em uma placa separada, este circuito foi feito separadamente do restante e é interconectado com a placa principal através de um conector de 16 pinos.

Este circuito (Figuras 6 e 7) consiste de um conector; dois displays de sete segmentos catodo comum, para indicar a rotação desejada do motor; dois leds, para indicar o sentido de rotação do motor e quatro botões tipo *push-button*: botão liga-desliga; botão aumenta rotação; botão diminui rotação e botão inverte rotação. Para poupar portas do PIC, os dois displays são multiplexados e ligados em paralelo na porta D, passando por um *driver* (ULN2003) e por resistores limitadores de corrente (Figura 7).

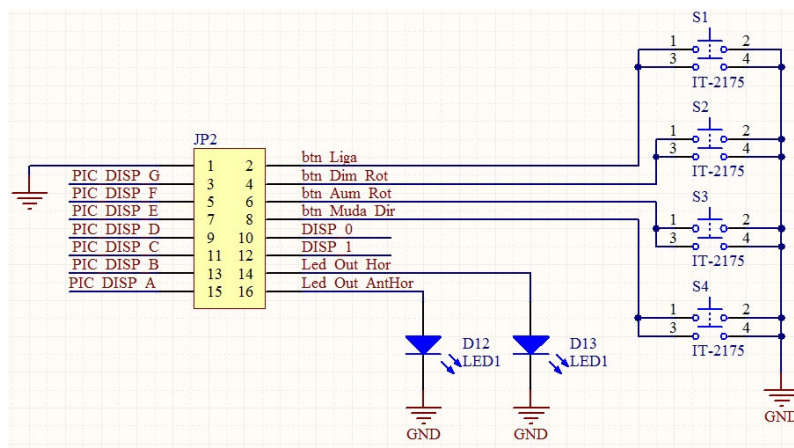


Figura 6 – Interface: conector, leds e botões.

Fonte: autoria própria.

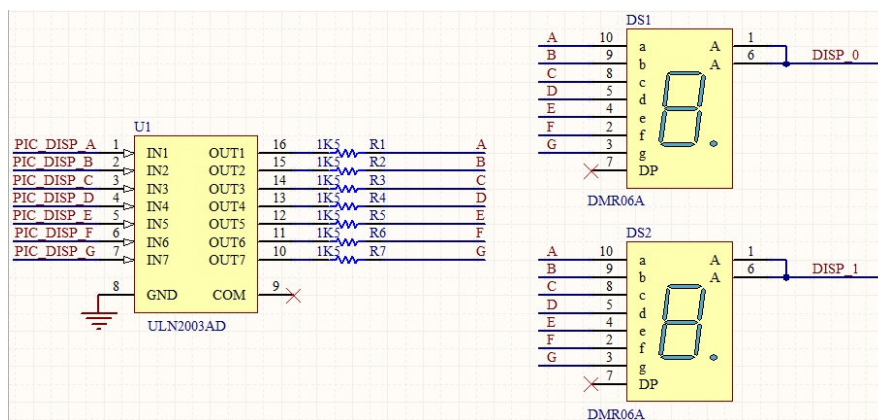


Figura 7 – Conexão dos displays.

Fonte: autoria própria.

7.2.5 CHAVEAMENTO

O sistema contém dois circuitos de chaveamento: o chaveamento do motor e o chaveamento dos displays. O chaveamento do motor é feito através de um MOSFET (Figura 3), que recebe o sinal (Gate_Motor) do PWM amplificado e invertido por transistores (BC548) e pelo driver ULN2003 (Figura 8).

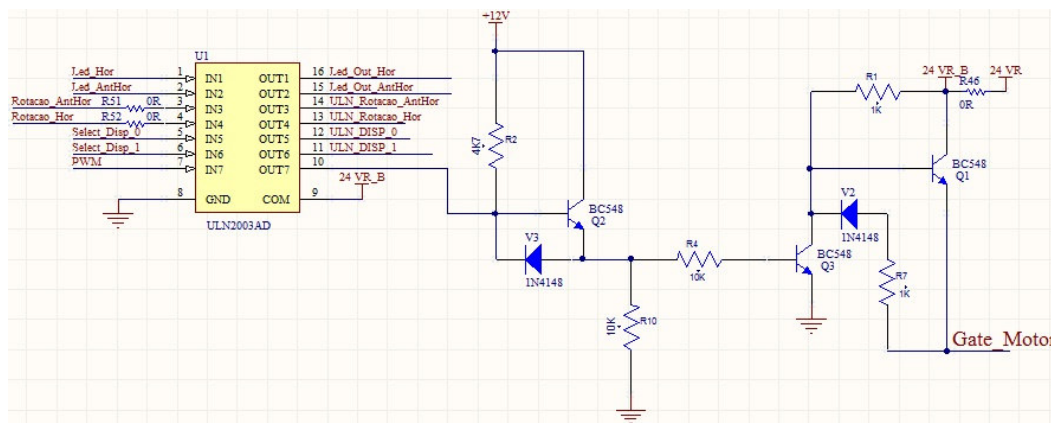


Figura 8 – Chaveamento do motor.

Fonte: autoria própria.

Já o chaveamento dos displays é feito por transistores (BC558), que recebem o sinal (ULN_DISP_1 e ULN_DISP_0) do PIC invertido pelo ULN2003, para selecionar qual display estará ligado (Figura 9).

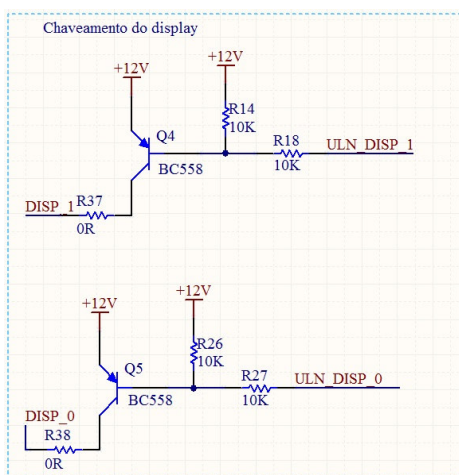


Figura 9 – Chaveamento dos displays.

Fonte: autoria própria.

7.2.6 CONECTORES

O circuito conta com seis conectores. Existe um conector para o motor, que está ligado aos relés, que possui quatro pinos com dois pinos não conectados. Estes pinos servem apenas para que o motor não seja ligado incorretamente. Há também um conector para a gravação do

firmware no PIC e um conector para a serial, que envia os dados para o computador. O transformador responsável pela alimentação do sistema é conectado à placa por meio de dois conectores Mini-Fit. O outro conector é o conector de 16 pinos que faz a comunicação entre as duas placas do sistema.

7.2.7 PLACAS

O circuito do sistema está dividido em duas placas (Figura 10), que se comunicam por meio de um *flat-cable* de 16 vias. A placa maior contém o circuito principal, que inclui fonte, sensores, controle, chaveamentos e conectores. A placa menor é a interface do sistema com o usuário, na qual se encontram os displays, leds e botões.

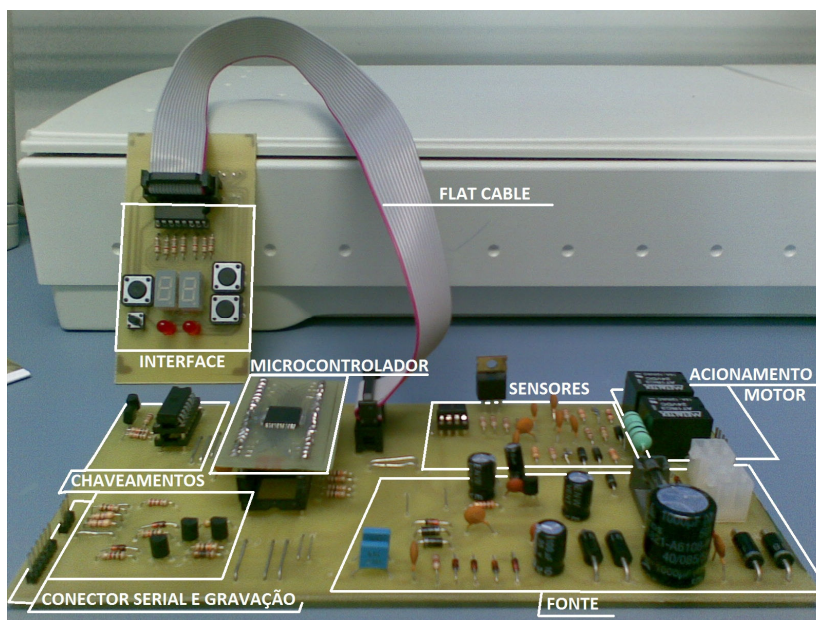


Figura 10 – Placas do sistema.

Fonte: autoria própria.

As placas foram confeccionadas manualmente. Primeiramente, a partir do esquemático eletrônico, foi projetado o layout do circuito das placas. Por ser de menor tamanho e possuir menos componentes, o layout da placa da interface foi mais simples. O layout da placa do circuito principal foi bastante complexo (Figura 11). Como a placa engloba diferentes partes do sistema, o posicionamento dos componentes precisou ser estudado minuciosamente antes que as trilhas fossem roteadas. O circuito ficou dividido em blocos: uma área da placa para a

fonte, outra para os sensores e assim por diante. Posteriormente cada bloco foi interligado pelas trilhas.

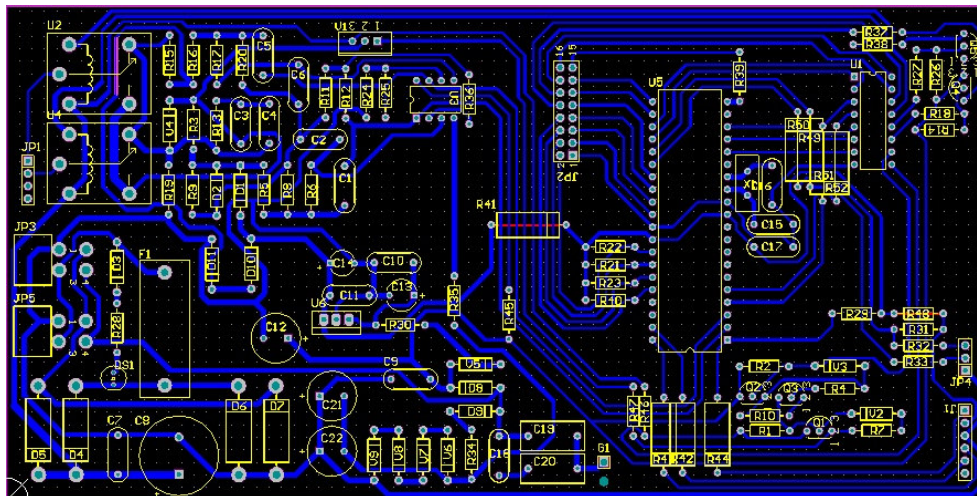


Figura 11 – Layout do circuito principal.

Fonte: autoria própria.

Depois de prontos, os layouts foram transferidos para as placas por meio de filme fotossensível e em seguida foi feita a corrosão do cobre. As placas foram perfuradas utilizando uma furadeira de mão. Finalmente, os componentes foram inseridos e soldados. Com as placas prontas, foi necessário conferir se o esquema havia sido respeitado e se não havia curtos-circuitos. Para estes foram realizados testes de continuidade utilizando um multímetro. Em seguida, a placa foi alimentada e utilizando-se um multímetro foram conferidas as tensões.

7.3 FIRMWARE

A implementação do *firmware* de controle do sistema pode ser dividida em duas etapas principais: a definição do funcionamento do sistema e a programação propriamente dita. O *firmware* é o principal responsável pelo funcionamento do sistema. Gravado no microcontrolador, ele controla todas as variáveis do sistema, contudo ele depende diretamente do *hardware*, de como cada elemento está conectado ao PIC.

7.3.1 FUNCIONAMENTO DO SISTEMA

O primeiro passo para a implementação do *firmware* de controle foi a definição detalhada do funcionamento do sistema. O controle de motor para instrumentos cirúrgicos consiste basicamente de um sistema que verifica a rotação desejada e atua no motor fazendo-o atingi-la. Como o sistema se propõe a manter a rotação mesmo diante variações na carga, deve existir uma verificação da corrente no sistema, para que o motor não seja danificado em caso de cargas muito elevadas. Assim, é possível resumir o funcionamento do sistema de acordo com o fluxograma da Figura 12.

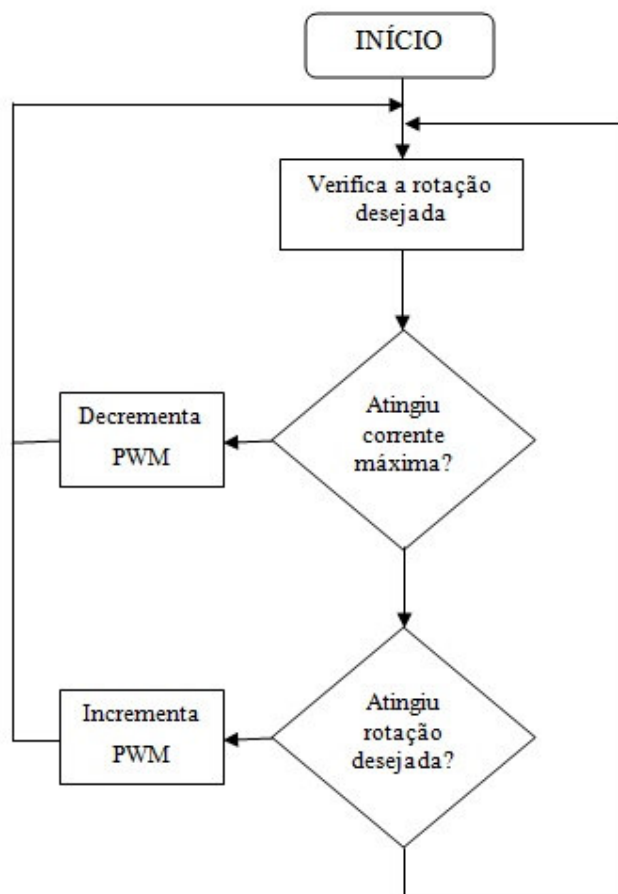


Figura 12 – Fluxograma simplificado.

Fonte: autoria própria.

Ao iniciar o sistema, verifica-se a rotação desejada, que será o *set-point*, e busca-se atingi-la. Neste processo verifica-se prioritariamente a corrente à tensão e a partir destes

dados verifica-se a rotação do motor, que é a variável controlada do sistema. Caso a corrente tenha atingido ou ultrapassado o valor máximo decrementa-se o PWM, que atua na rotação do motor. Se a corrente estiver dentro do limite aceitável e a rotação desejada ainda não tiver sido atingida incrementa-se o PWM, buscando atingi-la. O PWM controla a tensão, que é a variável manipulada do sistema, que será aplicada ao motor. A cada ciclo verifica-se novamente a rotação desejada, a corrente e a rotação atual, pois cada um destes parâmetros pode variar.

Durante este processo o usuário pode, por meio da interface, alterar a rotação a ser atingida e o seu sentido. O usuário pode escolher a rotação entre 2.000 e 40.000 rpm, em incrementos de 1.000 rpm. No display, o valor é mostrado em dois dígitos (de 02 a 40), ou seja, dividido por 1000. O sistema verifica constantemente se o usuário alterou a velocidade ou o sentido da rotação. Existe um sentido de rotação padrão (sentido horário) quando o sistema é iniciado, a partir disto o sentido é invertido cada vez que o usuário pressionar o botão para alterar o sentido. Quando o usuário altera a rotação a ser atingida, acionando os botões para aumentar ou diminuir a rotação, ocorre uma mudança de *set-point* e o sistema altera o valor indicado nos displays para este novo *set-point* e busca atingir a nova rotação.

Outra questão pertinente ao funcionamento do sistema é a forma de acionamento e desligamento do motor, assim como sua mudança de sentido de rotação. Uma vez que se trata de um sistema de controle para instrumentos cirúrgicos, o motor não deve sofrer alterações bruscas de velocidade e muito menos do sentido de rotação. Isso não deve ocorrer não apenas para não prejudicar o próprio motor, mas também para que este não trave e ofereça riscos de comprometer a cirurgia. Para isso foram implementadas rampas de subida e de descida para o acionamento do motor, para que, quando o sistema for ligado, este não passe do estado de repouso instantaneamente para a rotação desejada, mas sua rotação seja incrementada progressivamente. Assim, caso se deseje inverter a rotação durante uma cirurgia, o motor irá reduzir a velocidade progressivamente até parar completamente, para então inverter o sentido de rotação e acelerar progressivamente até atingir a rotação desejada novamente.

As rampas de subida e descida de acionamento do motor também são implementadas por meio do PWM. O PWM consiste de um sinal composto por pulsos de largura fixa no qual se varia o tempo em que cada pulso fica em nível alto (24V) ou baixo (0V). O *duty-cycle* (Figura 13) representa a porcentagem do tempo que o sinal fica em nível alto em relação ao tempo total do pulso, ou seja, é tempo que o sinal fica em nível alto dividido pelo tempo total do pulso multiplicado por 100. Assim as rampas de acionamento são obtidas incrementando ou decrementando progressivamente o *duty-cycle* do PWM.

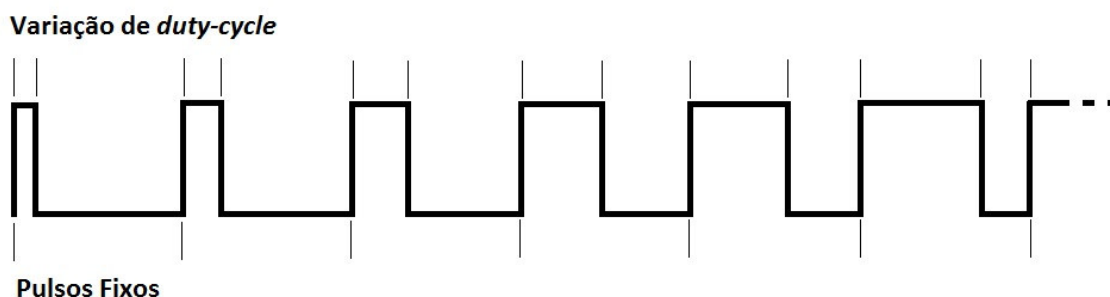


Figura 13 – Exemplo de variação de *duty-cycle*.

Fonte: autoria própria.

7.3.2 PROGRAMAÇÃO

A programação do *firmware* de controle foi toda realizada em linguagem C. O programa é todo estruturado em blocos e cada bloco é responsável por uma parte específica do programa. Os blocos comunicam-se e interligam-se por meio de funções e variáveis globais.

O primeiro passo da programação foi definir o microcontrolador utilizado e definir o *hardware* do sistema. Para a definição do *hardware* existe um bloco no qual é especificado o que está conectado em cada pino do PIC e se o pino foi utilizado como entrada ou saída (*input* ou *output*). De acordo com o esquemático, que pode ser observado na figura 5, as conexões das portas do PIC são as seguintes:

Bit\Porta	PORTA A	PORTA B	PORTA C	PORTA D
bit 0	Tensão no Motor	Botão Liga e Desliga	-	Display 7 Segmentos
bit 1	Corrente no Motor	Botão Diminui Rotação	PWM	
bit 2	Rotação Sentido Anti-Horário	Botão Aumenta Rotação	Seleciona Display 1	
bit 3	Rotação Sentido Horário	Botão Inverte Rotação	Seleciona Display 0	
bit 4	-	-	Led Anti-Horário	
bit 5	-	-	Led Horário	
bit 6	Osciladores	-	Serial TX1	
bit 7		-	Serial RX1	

Quadro 1 – Portas do PIC.

Fonte: autoria própria.

O programa possui uma estrutura (Figura 14) composta pelos blocos principais que juntos englobam todo o *firmware*. São eles: Inic.c; Main.c; TimersInterface.c; Serial.c; Debounce.c; Display.c; MotorMotion.c; e AdConverter.c. O primeiro bloco chamado pelo programa é o Inic.c, bloco responsável pela inicialização do programa. Neste bloco é definida a configuração principal do PIC (os *configuration bits*), são definidas as entradas e saídas do PIC e se elas são digitais ou analógicas. O bloco também inicializa as variáveis modulares e globais do *firmware*, inicializa a serial e os elementos de *hardware*. Este bloco é fundamental para o funcionamento adequado do sistema e para evitar problemas a cada vez que ele é inicializado.

Depois de feita a inicialização, é chamado o bloco principal do programa, o Main.c, que contém o loop principal do programa (a função MAIN_LOOP()). O loop principal do programa é um loop eterno responsável por chamar as tarefas principais na ordem correta. O MAIN_LOOP() consiste em:

```
#define MAIN_LOOP() do
{
    ProcessTimers();
    MainLoopSerial();
    SignalDebounceMainLoop();
    RefreshDisplay();
    MotorMotionMainLoop();
    AdConverterMainLoop();
}
while(0);
```

A função MAIN_LOOP() chama as funções na seqüência determinada, de forma ininterrupta. As funções chamadas pelo MAIN_LOOP() são os loops principais dos outros blocos que compõe a arquitetura do *firmware* (Figura 12), que executam as tarefas dos demais blocos do programa.

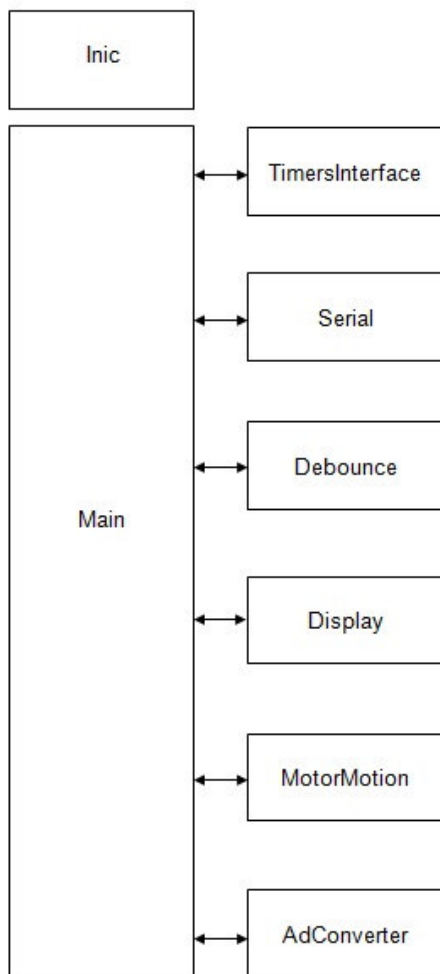


Figura 14 – Arquitetura do *firmware*.

Fonte: autoria própria.

O bloco `TimersInterface` é responsável pelo tratamento dos timers do sistema. Ele inicializa e configura os timers que são utilizados por outras funções. Os timers são utilizados da seguinte forma: um timer para cada sensor, um timer para a serial, um timer para o acionamento dos relés e um para a aquisição dos dados do conversor A/D. Além desses, são utilizados timers auxiliares para uso interno do programa, como um timer com período de 500 ms para a máquina principal do programa (a `MainMachine`) e outro de 1 ms para o controle do PWM.

`Serial` é bloco que faz o tratamento da comunicação serial. Ele configura e habilita a serial para enviar o pacote de dados para o computador. A velocidade de transmissão da serial (`SERIAL_BAUD_RATE`) é configurada para 57.600 bps. O pacote de dados a ser transmitido é definido no bloco que faz o tratamento da rotação do motor (`MotorMotion.c`), pois neste bloco são utilizadas todas as variáveis pertinentes ao controle do motor. Em cada ciclo é

enviada a corrente do motor, a tensão do motor e o *duty-cycle* e para indicar o fim do pacote de dados são enviados bytes com os valores 00001101 e 00001010 (equivalentes as letras “D” e “A” em hexadecimal). Como os vetores nos quais são armazenados os valores de corrente, tensão e *duty-cycle* são do tipo ‘word’, ou seja possuem dois bytes, ele são divididos em HIBYTE e LOBYTE (byte com os bits mais significativos e byte com os bits menos significativos do vetor respectivamente) para serem transmitidos. O bloco MotorMotion será melhor explicado mais adiante.

O bloco Debounce é responsável pelo *debounce* dos sinais de entrada. Ele inicializa a máquina de estados de *debounce* e realiza o *debounce* dos sensores. O *debounce* dos sinais de entrada serve para evitar que o sinal seja interpretado erroneamente pelo microcontrolador. No instante em que um botão é acionado, por exemplo, vários ruídos aleatórios são gerados durante alguns milissegundos, porém, isto pode ser o suficiente para que o PIC interprete como se o botão tivesse sido acionado várias vezes seguidas, apesar de ter sido acionado uma única vez. O *debounce* dos sinais de entrada ocorre da seguinte forma: no instante em que ocorre uma variação em um sinal de entrada, que estava em nível baixo e foi para nível alto, por exemplo, um timer de 20 ms (tempo definido pelo DEBOUNCE_PERIOD) é acionado. Se após este tempo o sinal permanecer em nível alto, a variação do sinal é confirmada. Caso o sinal volte para nível baixo antes de 20 ms a variação do sinal é interpretada como sendo apenas ruído.

A interface e o tratamento dos displays de sete segmentos é realizado pelo bloco Display. Neste bloco os dígitos de zero a nove são decodificados para valores em hexadecimal (Figura 15), que, quando são colocados na porta D do PIC, mostram o respectivo dígito no display selecionado. A função RefrshDisplay() é responsável por atualizar constantemente os valores dos displays. A partir da velocidade desejada para o motor, os valores que devem ser mostrados nos displays são decodificados. Como os dois displays estão conectados na mesma porta, o display 0 é desativado (DESELECT_DISPLAY_0_PORT()), limpa-se a porta D, o display 1 é selecionado (SELECT_DISPLAY_1_PORT()) e escreve-se o valor na porta, em seguida repete-se o processo, porém desativando-se o display 1 e selecionando-se o display 0.

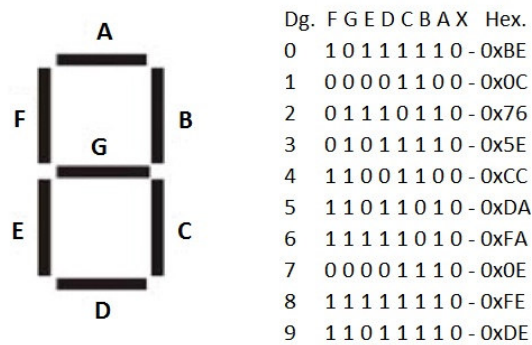


Figura 15 – Decodificação dos segmentos do display.

Fonte: autoria própria.

O bloco MotorMotion pode ser considerado um dos blocos mais importantes do sistema, pois é ele que controla os movimentos do motor e faz o tratamento do acionamento, das rampas e do travamento do motor. Também é neste bloco que os dados de corrente, tensão e *duty-cycle* são enviados para a serial.

Para realizar a lógica de controle dos movimentos do motor primeiramente são definidos os estados em que ele pode se encontrar:

```
typedef enum uMOTOR_STATES tag
{
    MOTOR_IDLE,           // Motor está parado.
    MOTOR_UNLOCKING,     // Destravando o motor.
    MOTOR_RUNNING,       // Motor está rodando.
    MOTOR_RISING_RAMP,   // Motor está em rampa de subida.
    MOTOR_FALLING_RAMP,  // Motor está em tampa da descida.
    MOTOR_WAITING_STOP,  // Motor aguardando parada.
    MOTOR_LOCKING,       // Travando o motor.
}
uMOTOR_STATES;
```

Assim, a partir do estado em que o motor se encontra são definidas as tarefas a serem executadas. Primeiramente, é feita a inicialização da movimentação do motor, na qual ele é colocado em estado de repouso (MOTOR_IDLE). Foi definido que o motor começará rodando em sentido horário, configura-se o PWM e inicializam-se as variáveis do módulo, que fica aguardando um comando para ligar o motor.

O tratamento de toda a movimentação do motor é feito na função MotorMotionMainLoop(), função principal responsável pelo funcionamento do sistema.

Nesta função são considerados todos os casos possíveis para cada estado em que o motor pode se encontrar e para cada um dos casos são definidas as tarefas a serem realizadas para as diferentes situações. Portanto, a função `MotorMotionMainLoop()` consiste de uma extensa seqüência de comandos condicionais sobrepostos para cada estado do motor, de modo a abranger todas as combinações e situações em que o motor possa se encontrar. Por exemplo, caso o motor esteja parado (“case `MOTOR_IDLE:`”), verifica-se se deve-se ligar o motor. Se sim, deve-se verificar se está em processo de inversão do sentido de rotação, se sim, verifica-se se estava no sentido horário, se sim, deve-se ligar o motor no sentido anti-horário, e assim por diante. Vale lembrar que para cada “se sim” existe também a opção “se não”, para abranger todas as opções.

Para saber se é preciso aumentar ou diminuir a rotação do motor (incrementar ou decrementar o *duty-cycle* do PWM) o módulo `MotorMotion` recebe os valores de tensão e corrente do módulo `AdConverter`, que faz a leitura e tratamento dos conversores A/D e calcula se o valor da rotação está próximo ou não do valor desejado. Para a interpretação dos valores lidos nos conversores A/D existem tabelas de decodificação que indicam a que rotação o valor lido se refere. A tabela de conversão para a tensão (APÊNDICE A) possui os valores em hexadecimal referentes a cada rotação, de 2.000 rpm até 40.000 rpm em incrementos de mil rpm. Desse modo, 2.000 rpm será lido como $2F^h$ e 40.000 rpm como $24F^h$. Na tabela de conversão para corrente, o valor em hexadecimal indica a corrente limite para cada rotação, para 2.000 rpm é 150^h e a partir de 16.000 até 40.000 rpm o limite é 80^h . A rotação escolhida pelo usuário é decodificada de acordo com as tabelas, assim é possível saber qual o valor esperado de leitura para a tensão e o valor máximo da corrente, para se fazer o controle da rotação do motor. A partir disso, dentro dos comandos condicionais para cada situação, no bloco `MotorMotion` são chamadas as rotinas de PWM, que estão no bloco `PwmControl`.

Como o sistema é feito visando motores para instrumentos cirúrgicos, existe uma preocupação com movimentos bruscos do motor. Em casos de cirurgias odontológicas, por exemplo, o instrumento não pode “dar trancos”, nem variar a velocidade bruscamente. Para evitar estes problemas, foram implementadas as rampas de acionamento. Toda vez que se precisa incrementar ou decrementar o PWM, seja para aumentar, diminuir ou inverter a rotação, no próprio módulo `MotorMotion` são chamadas as rotinas das rampas de acionamento. Nestas rotinas é verificado se o motor se encontra em alguma rampa e é analisado que tipo de caso deve ser tratado. A partir destes dados, o *duty-cycle* do PWM é variado progressivamente em pequenos passos, para realizar uma rampa suave.

A conversão dos dados analógicos para digitais é tratada no bloco AdConverter. Neste bloco é feita a aquisição dos valores de tensão e corrente do motor. Estes valores são tratados e depois são usados pelo bloco MotorMotion para controlar a rotação do motor. No início da conversão dos dados, as portas do PIC são reconfiguradas como portas analógicas, as variáveis que armazenam temporariamente os valores de tensão e corrente (variáveis do tipo WORD: mwVoltageVector; e mwCurrentVector;) são zeradas, então a conversão é feita. Primeiramente é adquirido o valor da tensão e então o da corrente. Esses valores não são tratados pelo bloco MotorMotion imediatamente após a conversão, existe uma variável booleana que indica quando há novos dados a serem tratados. No início da conversão esta variável é definida como falsa, indicando que não há novos dados a serem tratados. Conforme ocorrem as conversões A/D, são feitas médias entre as amostras dos valores de tensão e corrente medidos. Existem quantidades de amostras diferentes para realizar a média da tensão e da corrente. Depois de feita a média com todas as amostras disponíveis, a variável que indica se há um novo dado a ser tratado é definida como verdadeira e somente então os dados são tratados pelo bloco MotorMotion. A quantidade de amostras utilizadas para realizar as médias de cada um dos valores influencia no efeito integral do controle do motor, como será explicado na seção seguinte.

Para realizar a conversão A/D dos valores de tensão é necessário gerar uma interrupção no sistema. As interrupções do sistema são tratadas pelo bloco Interrupts, no qual são definidas as interrupções e as suas prioridades. Antes de realizar a conversão da tensão é gerada uma interrupção, de modo que o sistema pára de atuar no motor. Neste instante o motor continua girando devido à inércia e passa funcionar como um gerador. É neste momento que se consegue medir a força contra-eletromotriz, então a conversão é feita e o sistema volta a atuar no motor.

7.4 CONTROLE

O sistema possui um controle implementado via *firmware*. É um controle Dedicado de processo contínuo com realimentação. O controle consiste basicamente em manter o motor na velocidade desejada, que seria o *set-point* (SP). Desta forma pode-se dizer que a variável controlada (VC) do sistema é a velocidade. Contudo, a velocidade do motor é proporcional a tensão nele aplicada e depende também, devido à possibilidade de variação de carga, da corrente que passa por ele. Assim, podemos dizer que a tensão e a corrente também são

variáveis controladas (ou monitoradas), uma vez que existe um controle do valor da tensão, para se atingir a rotação desejada, e um controle da corrente, que serve como um limitador, limitando a corrente caso a carga seja muito elevada.

A malha de controle pode ser representada de acordo com a Figura 16. O fato de existir uma variável controlada e outra limitadora, caracteriza uma estratégia de controle com intertravamento, realizado pelo limitador de corrente.

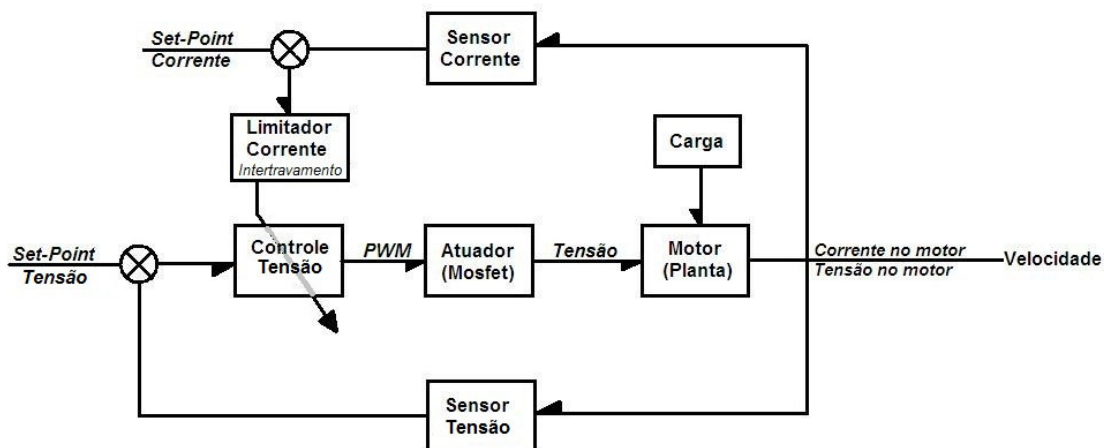


Figura 16 – Diagrama de blocos da malha de controle.

Fonte: autoria própria.

O sistema é composto basicamente pelo motor (Planta), pelos controladores e sensores de tensão e corrente. A carga aplicada no motor pode ser considerada como o distúrbio do sistema, pois irá alterar o esforço do motor, influenciando sua rotação. Pode-se dizer que o SP principal é a velocidade de rotação desejada, selecionada pelo usuário. Essa velocidade desejada é convertida no sistema (por meio de tabelas) em um SP para a tensão e um SP para a corrente. Da mesma maneira a velocidade do motor seria a VC principal, contudo o controle atua a partir da tensão e da corrente do motor. O controle atua no *duty-cycle* do PWM, que é a variável manipulada (VM) do controle de tensão, que por sua vez atua na tensão aplicada ao motor por meio do chaveamento do motor (MOSFET). Neste caso, a tensão também pode ser considerada como variável manipulada em relação ao atuador, representando, portanto, a variável manipulada do controle de velocidade.

O controle da tensão lê o valor recebido pelo sensor e compara com o valor do SP, obtendo um erro e atua na VM, buscando eliminá-lo. O intertravamento é feito pelo controle de corrente, que atua basicamente como um limitador. Caso o valor da corrente lido pelo

sensor seja maior que o valor do SP, ele irá limitar o controle de tensão, para que a VM não seja incrementada, impedindo que a corrente ultrapasse o valor máximo.

7.4.1 IMPLEMENTAÇÃO

O controle do sistema é implementado via *firmware*, programado em linguagem C. O controle é chamado de Dedicado por ter sido desenvolvido especificamente para o sistema em questão. O controle é composto por três ações de controle, inspiradas nas ações de controle proporcional, integral e derivativa, adaptadas e implementadas em partes específicas do programa para se conseguir o efeito desejado. São elas o Ganho Variável, efeito Amostral e de Atraso.

O Ganho Variável é feito diretamente no bloco que faz o controle do motor, o MotorMotion, de modo que ele é o efeito de controle que atua mais diretamente no motor. A ação do Ganho Variável atua diretamente no erro (diferença entre *duty-cycle* desejado e *duty-cycle* atual). Ele é definido basicamente por três parâmetros:

```
#define DUTY_CYCLE_ACCURACY_LEVEL
#define WIDE_VARIATION_DUTY_CYCLE
#define LOW_VARIATION_DUTY_CYCLE
```

O controle é dividido em duas partes: *Wide Variation*, área de grande variação no *duty-cycle*; e *Low Variation*, área de pequena variação no *duty-cycle*. Em cada área é definida a quantidade de passos que serão dados no incremento ou decremento do PWM de cada vez.. A área de pequena variação é determinada a partir da *define* DUTY_CYCLE_ACCURACY_LEVEL, que define o nível de precisão da área de pequena variação (Figura 17). Ele define o erro aceitável, para mais ou para menos, entre o *duty-cycle* desejado e o *duty-cycle* atual. O valor da *define* WIDE_VARIATION_DUTY_CYCLE define qual será o passo do *duty-cycle* na área de grande variação. Por exemplo, se o valor for cinco o *duty-cycle* será incrementado ou decrementado de cinco em cinco passos. Já a *define* LOW_VARIATION_DUTY_CYCLE define o passo para a pequena variação, que preferencialmente deve ser um, para se obter a maior precisão possível.

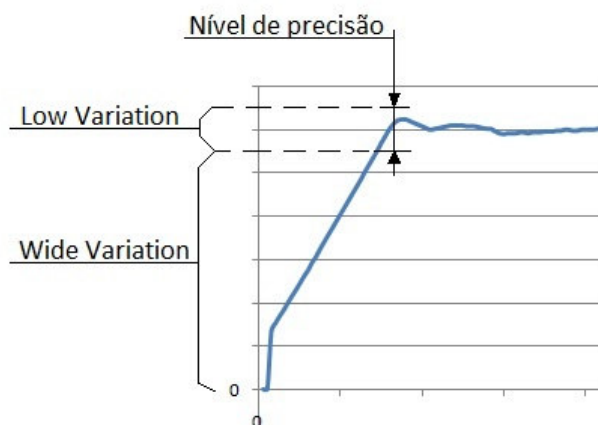


Figura 17 – Exemplo das áreas de controle.

Fonte: autoria própria.

Primeiramente, o *duty-cycle* atual é comparado com o desejado. Caso o valor seja muito diferente do desejado, ou seja, fora da área de pequena variação, o *duty-cycle* é incrementado de acordo com o `WIDE_VARIATION_DUTY_CYCLE`, para se atingir mais rapidamente o valor desejado. Se o *duty-cycle* já estiver próximo do desejado, o controle entra na área de pequena variação, na qual é feito um controle mais preciso, incrementando o *duty-cycle* de acordo com o `LOW_VARIATION_DUTY_CYCLE`, evitando-se um *overshoot*.

O efeito de ação de controle Amostral é obtido no bloco `AdConverter`. Pode-se afirmar que a ação Amostral funciona como a ação integral, que leva em consideração o “histórico” da variável controlada para eliminar o erro. Para isso é feita uma média dos erros de várias amostras das variáveis (tensão e corrente), antes que elas sejam tratadas. No bloco `AdConverter` é feita a média dos erros a partir das leituras dos sensores de tensão e corrente e essa média compõe o valor que será tratado pelo bloco `MotorMotion`. Desta maneira, o que define o efeito da ação Amostral é a quantidade de amostras da variável que é utilizada para compor a média. A quantidade de amostras é determinada pelos seguintes parâmetros:

```
#define CURRENT_SENS_COUNTER
#define VOLTAGE_SENS_COUNTER
```

A *define* `CURRENT_SENS_COUNTER` determina a quantidade de amostras que compõe o valor de corrente e a *define* `VOLTAGE_SENS_COUNTER` determina a quantidade de amostras que compõe o valor da tensão.

Aumentando-se o número de amostras, aumenta-se o efeito da ação Amostral, deixando o controle mais preciso, buscando-se eliminar o erro. Contudo, aumentar a quantidade de amostras também deixa a resposta do sistema mais lenta, pois é necessário aguardar a conversão A/D de todas as amostras e o cálculo da média antes do valor ser tratado no bloco MotorMotion.

A ação de Atraso atua juntamente com a ação de Ganho Variável na parte transitória, ou seja, enquanto o sinal está variando. Ela procura antecipar a ação de controle para deixar a resposta mais rápida e aumentar a estabilidade do sistema, diminuindo o *overshoot*. Além do controle de Ganho Variável com a área de precisão, existe um parâmetro no bloco Interrupts que define a quantidade de intervalos para a medição da FCEM:

```
#define BEMF_MEASURE_INTERVAL
```

Este parâmetro influi na quantidade de intervalos entre a medição da FCEM do motor e a ação do PWM, de modo que diminuindo estes intervalos antecipa-se a ação do controle. Aumentando este valor gera-se um atraso para a medição, de modo que a medição será mais significativa, pois a ação de controle já terá um efeito maior no valor medido. Deste modo pode-se alterar a estabilidade do sistema e sua resposta, contudo também se altera a sua precisão.

7.4.2 EFEITOS DAS AÇÕES DE CONTROLE

A partir dos dados enviados via serial é possível levantar gráficos e analisar a resposta do sistema. Assim, alterando os parâmetros de controle e levantando gráficos de diversas sintonias pode-se analisar o efeito das ações de controle no sistema.

Os gráficos mostram o valor das variáveis (eixo y) em relação às amostras coletadas (eixo x). Como o *baud rate* de comunicação serial está definido em 57.600 bps, e em cada pacote são enviados 7 bits de dados para cada uma das 3 variáveis, obtêm-se uma média de 46 amostras por segundo, ou seja, cada amostra representa aproximadamente 21 ms. Em cada gráfico a série um representa a corrente no motor, a série dois, o valor da tensão no motor, a série três representa o *duty-cycle*, que varia de acordo com a carga aplicada ao motor e a série quatro representa o valor desejado para a tensão, o seja, o *set-point*.

A sintonia do controle para a planta em questão, o motor INTRA LUX K200 da Kavo, foi feita de maneira predominantemente empírica, utilizando tentativas sistemáticas.

Conhecendo-se basicamente cada ação de controle, os parâmetros eram definidos e eram realizados testes verificando-se a resposta do sistema, até que se encontrasse uma resposta que fosse satisfatória. Foram estabelecidos alguns valores de parâmetros desejados para que a sintonia fosse considerada satisfatória: precisão maior que 98%, tempo de acomodação menor que 3 segundos (para degrau de 40.000 rpm), sobre-valor (*overshoot*) menor que 5%. Além destes requisitos é de fundamental importância, considerando a aplicação em cirurgias, que o sistema se mantenha a rotação constante mesmo com variação de carga, que representa o problema regulatório. A sintonia que obteve a resposta mais satisfatória apresenta os parâmetros de controle configurados da seguinte maneira:

```

TESTE 1:
GANHO VARIÁVEL:
#define DUTY_CYCLE_ACCURACY_LEVEL    5
#define WIDE_VARIATION_DUTY_CYCLE    3
#define LOW_VARIATION_DUTY_CYCLE     1
AMOSTRAL:
#define CURRENT_SENS_COUNTER         100
#define VOLTAGE_SENS_COUNTER         20
ATRASSO:
#define BEMF_MEASURE_INTERVAL        1

```

Com estes parâmetros, obteve-se uma precisão de aproximadamente 99,7%, tempo de acomodação de aproximadamente 2,5s e *overshoot* menor que 1,2% (Gráfico 1). O gráfico 1 mostra uma resposta ao degrau na qual a rotação desejada é de 40.000 rpm. De acordo com a tabela de conversão, o valor digital lido deve ser de 24F em hexadecimal, ou seja, de 591 em decimal. Após a estabilização do sistema, foi aplicada uma carga ao motor indicada no gráfico com o aumento do *duty-cycle*, simulando um distúrbio. Pode-se observar que mesmo com o aumento da carga, a tensão no motor manteve-se constante, contudo, no momento em que a carga é retirada, ocorre um aumento de tensão. Isso ocorre devido ao fato da carga ter variado muito rapidamente, contudo, em menos de 700 ms (tempo equivalente a aproximadamente 33 amostras, no caso entre as amostras 359 e 392 no gráfico) esse *overshoot* foi corrigido e a rotação voltou ao valor desejado.

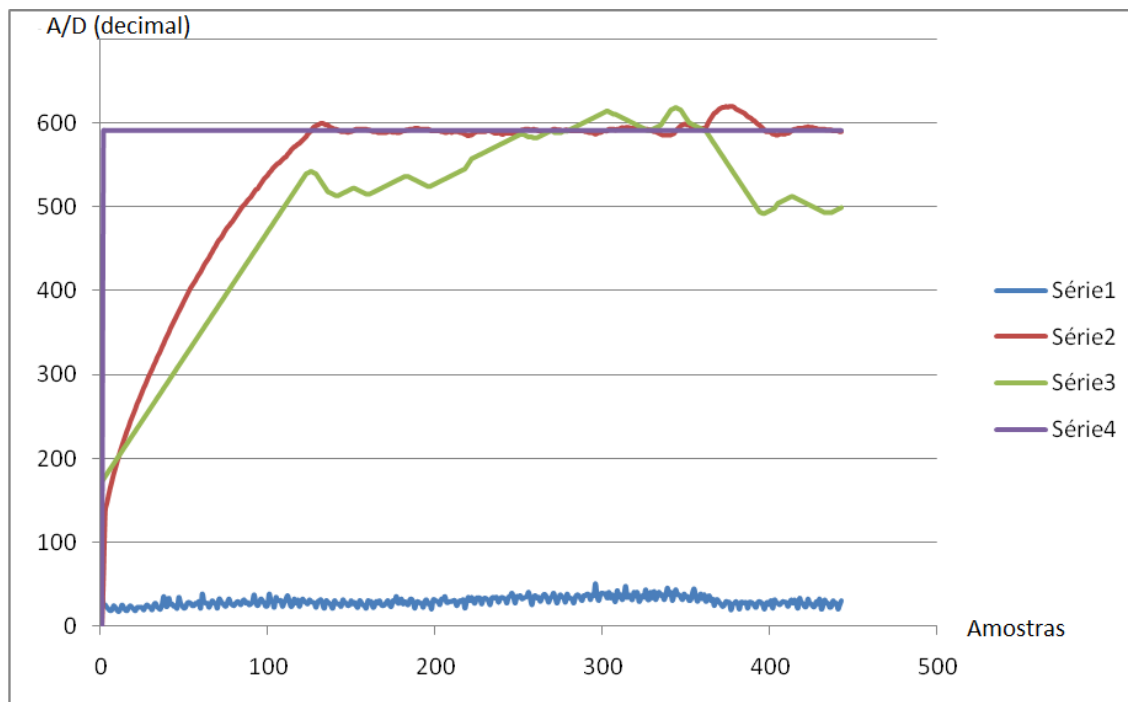


Gráfico 1 – Resposta ao degrau (servo) para 40.000 rpm e alteração de carga (regulatório) - Teste 1.

Fonte: autoria própria.

Ainda utilizando os mesmos parâmetros, foi realizado outro teste variando-se a rotação desejada do motor. Iniciou-se com 20.000 rpm, aumentando-se para 30.000 rpm, em seguida para 40.000 rpm e finalmente retornou-se para 20.000 rpm, obtendo-se a resposta mostrada no Gráfico 2:

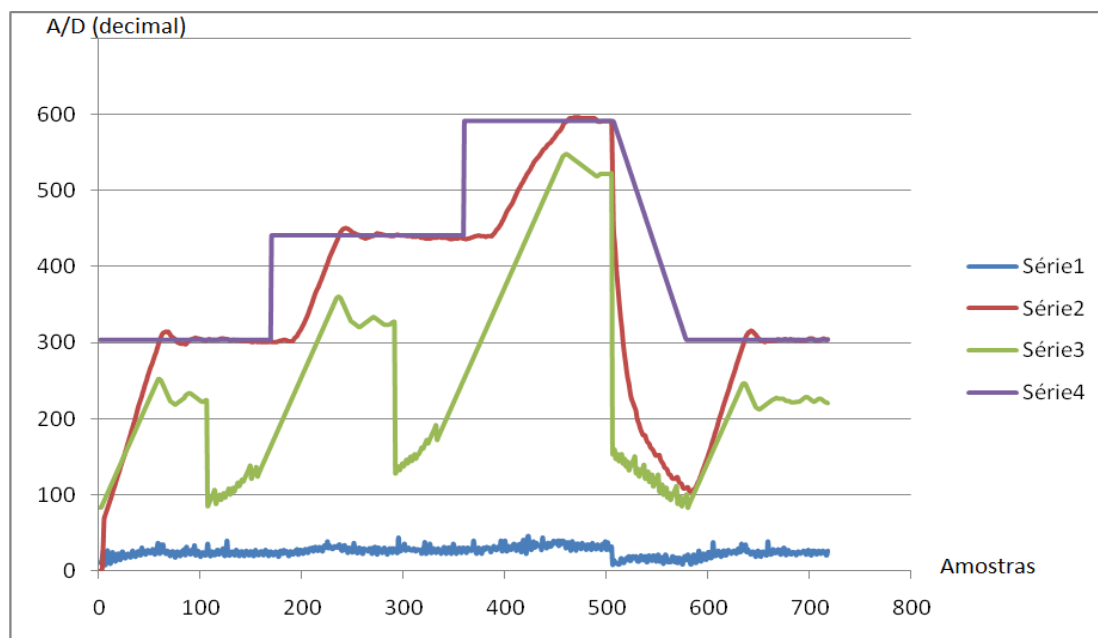


Gráfico 2 – Teste 1 alterando a rotação desejada (20.000 – 30.000 – 40.000 – 20.000 rpm) – problema servo.

Fonte: autoria própria.

Pode-se observar que em cada variação de rotação existe um tempo morto, ou seja, o tempo que o sistema leva para começar a reagir à mudança de *set-point*. Entre 20.000 e 30.000 rpm, por exemplo, existe um tempo morto de aproximadamente 420 ms (entre as amostras 170 e 190). Também há um *overshoot* em cada degrau, que chega até 3,6% (para 20.000 rpm).

Tendo como base o Teste 1, foram realizados diversos outros testes, alterando-se os parâmetros de controle, com o intuito de analisar o efeito das ações de controle sobre a resposta do sistema. Um segundo teste foi realizado alterando-se o parâmetro BEMF_MEASURE_INTERVAL. Com esta alteração, o intervalo entre as medições da FCEM é zerado, ou seja, em todos os ciclos a FCEM era medida, diminuindo um pouco o efeito de Atraso. Desta maneira o *overshoot* foi eliminado, como mostra o Gráfico 3. Contudo, no Teste 2 o sistema ficou um pouco mais sensível à variação de carga, pois ele precisa de mais ciclos para perceber as variações na FCEM.

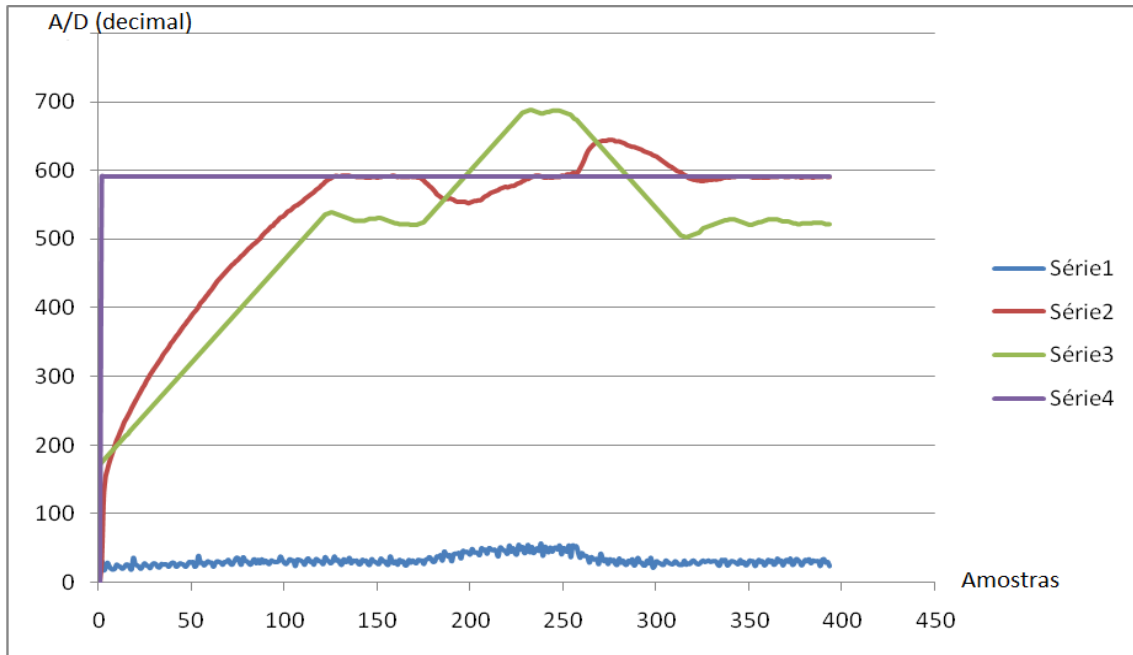


Gráfico 3 – Resposta ao degrau (servo) para 40.000 rpm e alteração de carga (regulatório) - Teste 2.

Fonte: autoria própria.

Também foram realizados testes alterando a quantidade de amostras utilizadas para o cálculo das médias, alterando assim o efeito Amostral. Dobrando a quantidade de amostras, no Teste 3 obtiveram-se os resultados mostrados nos Gráficos 4 e 5, de acordo com a seguinte configuração:

```

TESTE 3:
GANHO VARIÁVEL:
#define DUTY_CYCLE_ACCURACY_LEVEL 5
#define WIDE_VARIATION_DUTY_CYCLE 3
#define LOW_VARIATION_DUTY_CYCLE 1
AMOSTRAL:
#define CURRENT_SENS_COUNTER 200
#define VOLTAGE_SENS_COUNTER 40
ATRASSO:
#define BEMF_MEASURE_INTERVAL 1

```

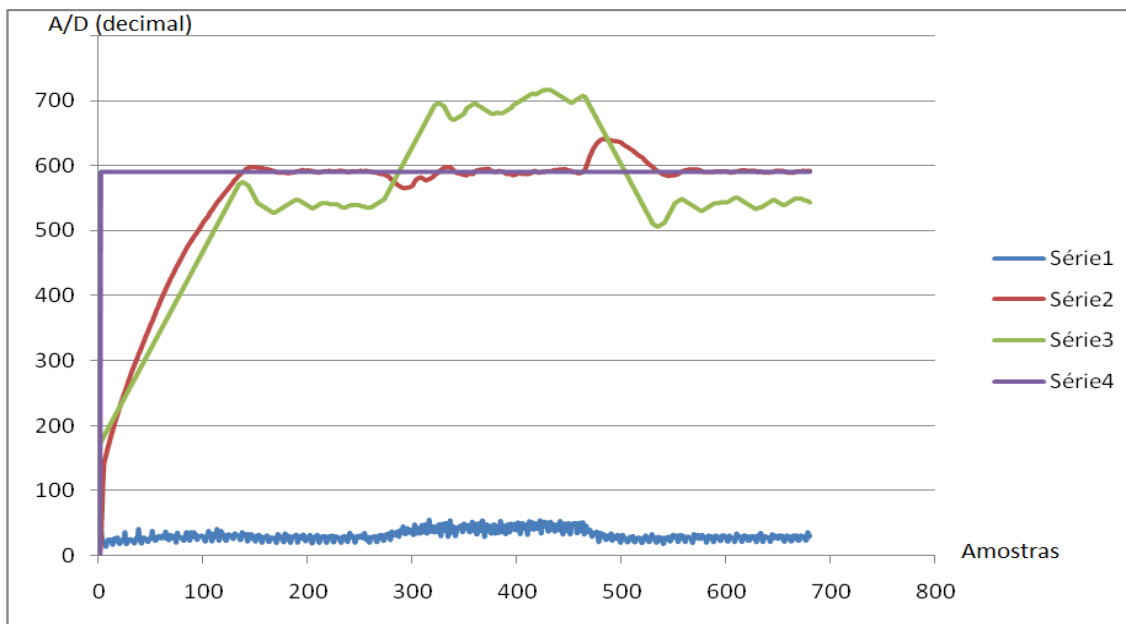



Gráfico 4 – Resposta ao degrau (servo) para 40.000 rpm e alteração de carga (regulatório) - Teste 3.

Fonte: autoria própria.

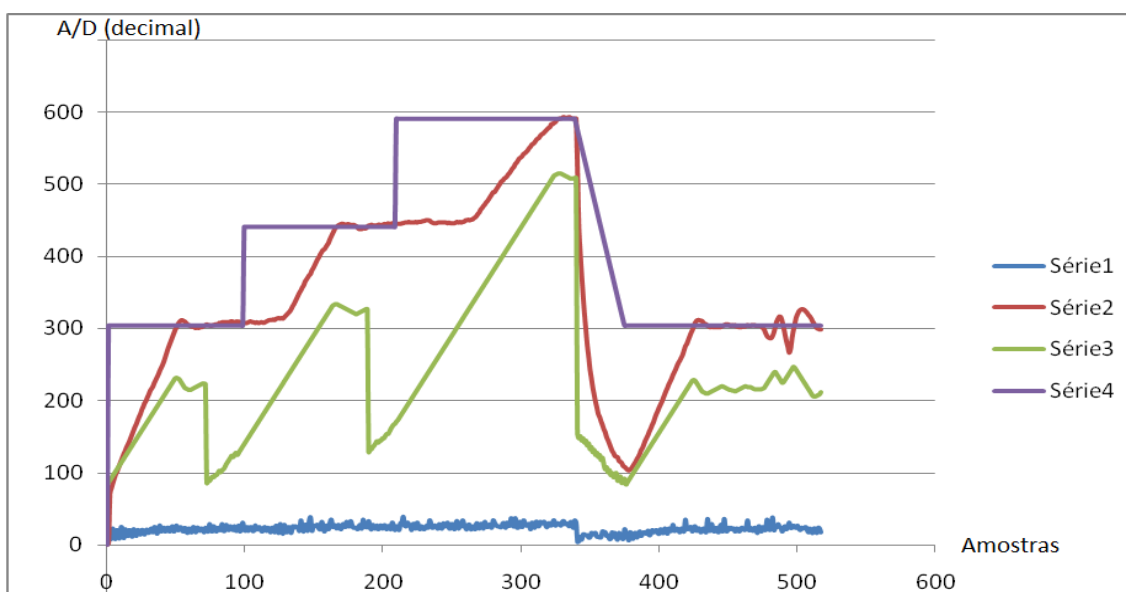


Gráfico 5 – Teste 3 variando a rotação desejada (20.000 – 30.000 – 40.000 – 20.000rpm) – problema servo.

Fonte: autoria própria.

No Gráfico 4 pode-se observar que a resposta apresenta uma melhora quanto a precisão, mesmo após a variação da carga. Depois da estabilização do sistema e antes do aumento da carga, obteve-se uma precisão de aproximadamente 99,8%. Entretanto, o Gráfico 5 mostra um aumento no tempo morto (entre as amostras 95 e 125, e amostras 205 e 255). O sistema demora mais para responder à variação de *set-point*, precisando aproximadamente de

1s para atingir o novo *set-point*. Isso se deve ao fato de que ao se dobrar a quantidade de amostras, dobra-se também o tempo para se obter essas amostras e calcular a média, fazendo com que a resposta à variação do *set-point* fique mais lenta. Também é possível observar no Gráfico 5 uma diminuição no *overshoot* para menos de 1%, se comparado ao 1,2 % de *overshoot* do Teste 1.

Um quarto teste foi realizado alterando-se os parâmetros do controle de Ganho Variável, aumentando-se o efeito do ganho. Foi diminuída a área de precisão e aumentou-se o passo da área de grande variação de *duty-cycle*, da seguinte maneira:

```

TESTE 4:
GANHO VARIÁVEL:
#define DUTY_CYCLE_ACCURACY_LEVEL 3
#define WIDE_VARIATION_DUTY_CYCLE 5
#define LOW_VARIATION_DUTY_CYCLE 1
AMOSTRAL:
#define CURRENT_SENS_COUNTER 100
#define VOLTAGE_SENS_COUNTER 20
ATRASSO:
#define BEMF_MEASURE_INTERVAL 1

```

Com esta configuração conseguiu-se aumentar a ação do ganho, de modo que a resposta do sistema ficou mais rápida.

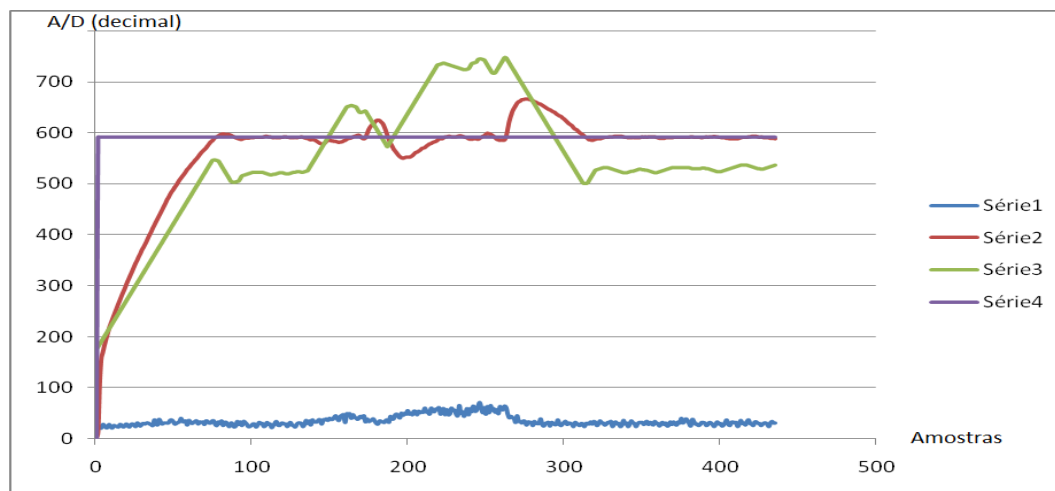


Gráfico 6 – Resposta ao degrau (servo) para 40.000 rpm e alteração de carga (regulatório) - Teste 4.

Fonte: autoria própria.

Como se pode observar no Gráfico 6, o sistema estabilizou-se mais rapidamente, com o tempo de acomodação de aproximadamente 1,6 s, praticamente 1s mais rápido do que no Teste 1. O Gráfico 7, todavia, mostra que em relação à variação de *set-point*, a resposta

apresenta um *overshoot* um pouco maior em cada degrau, chegando a 4,9% (para 20.000 rpm), 1,3% maior que no Teste 1. O tempo morto que era próximo de 420 ms no Teste 1, agora é de aproximadamente 315 ms (entre as amostras 115 e 129).

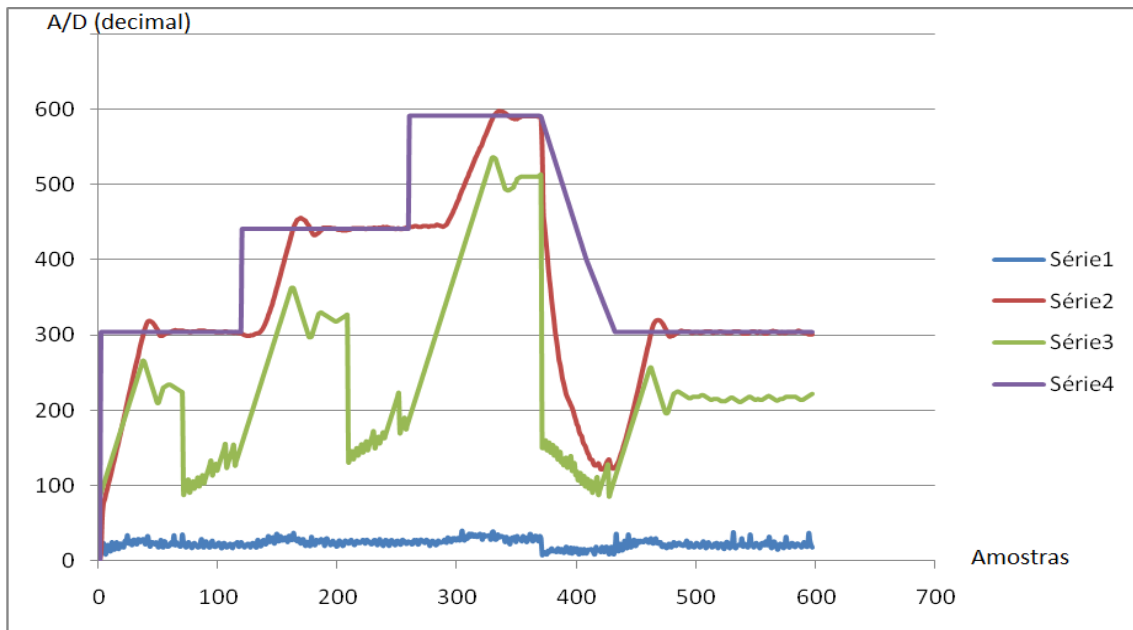


Gráfico 7 - Teste 4 variando a rotação desejada (20.000 – 30.000 – 40.000 – 20.000rpm) – problema servo.
Fonte: autoria própria.

Analisando as respostas obtidas nos diferentes testes realizados, procurou-se configurar os parâmetros de controle de modo a melhorar a resposta do sistema. Buscando-se uma resposta mais rápida, com mínimo de *overshoot* e de precisão adequada, foi feita a seguinte configuração:

```

TESTE 5:
GANHO VARIÁVEL:
#define DUTY_CYCLE_ACCURACY_LEVEL          3
#define WIDE_VARIATION_DUTY_CYCLE         5
#define LOW_VARIATION_DUTY_CYCLE          1
AMOSTRAL:
#define CURRENT_SENS_COUNTER              200
#define VOLTAGE_SENS_COUNTER              40
ATRASSO:
#define BEMF_MEASURE_INTERVAL             1

```

Procurou-se assim aumentar o efeito do Ganho Variável, para obter uma resposta mais rápida. Para compensar a instabilidade aumentou-se também o efeito amostral. Com o Teste 5 foram obtidas as seguintes respostas:

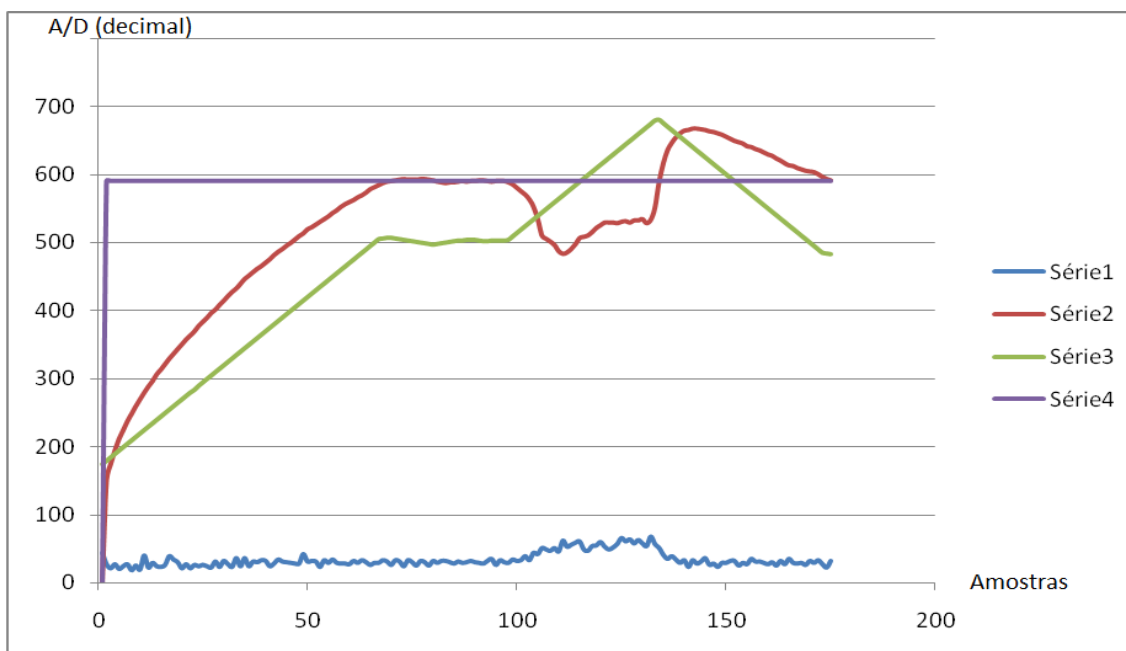


Gráfico 8 – Resposta ao degrau (servo) para 40.000 rpm e alteração de carga (regulatório) - Teste 5.

Fonte: autoria própria.

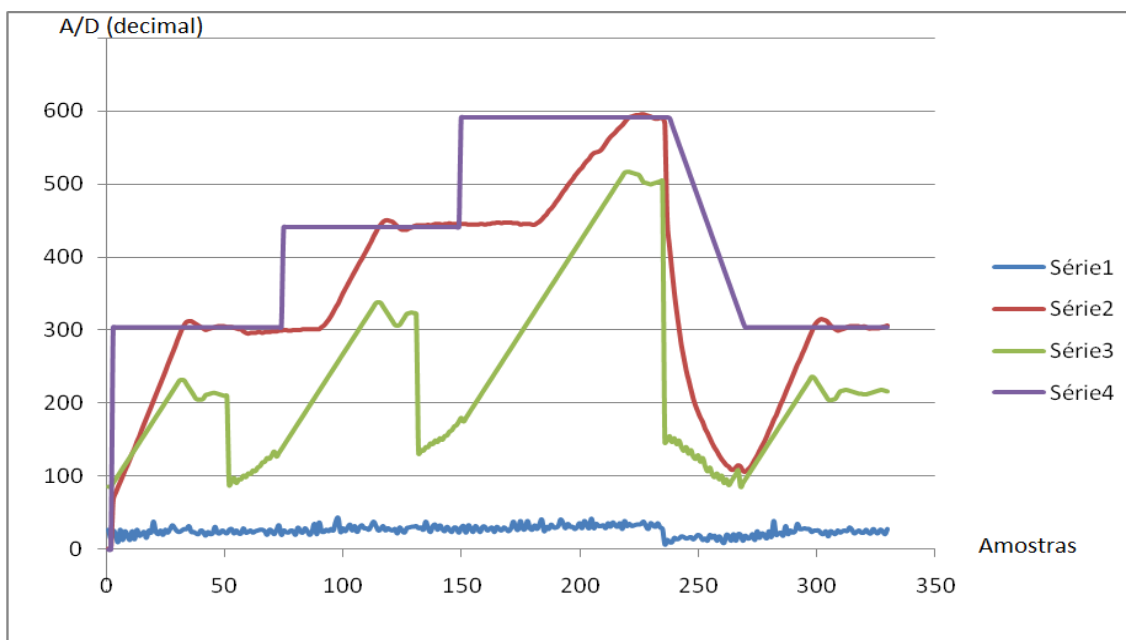


Gráfico 9 – Teste 5 variando a rotação desejada (20.000 – 30.000 – 40.000 – 20.000 rpm) – problema servo.

Fonte: autoria própria.

Com estes parâmetros, obteve-se no Teste 5 uma resposta mais rápida, com tempo de acomodação menor que 1,4s, sem *overshoot*, e com precisão de aproximadamente 99,5%, para o degrau para 40.000 rpm (Gráfico 8). Já a resposta em relação à variação de *set-point*, apresenta um *overshoot* de até 2,6% e tempo morto aproximado de 315 ms (entre as amostras 72 e 87), de acordo com o Gráfico 9. Contudo, apesar da resposta ser mais rápida, não apresentar *overshoot* e manter uma boa precisão, o Teste 5 não apresentou uma resposta tão satisfatória quanto à do Teste 1 à perturbação de carga, elemento muito importante para o funcionamento adequado do sistema.

Ainda com a sintonia do Teste 5, realizou-se mais um teste para avaliar a resposta do sistema à inversão do sentido de rotação do motor (Gráfico 10). Neste gráfico é possível observar que mesmo com a inversão de rotação o tempo de acomodação é de aproximadamente 1,4 s, praticamente sem *overshoot*. Também se pôde observar as rampas de acionamento do motor quando se inverte a rotação.

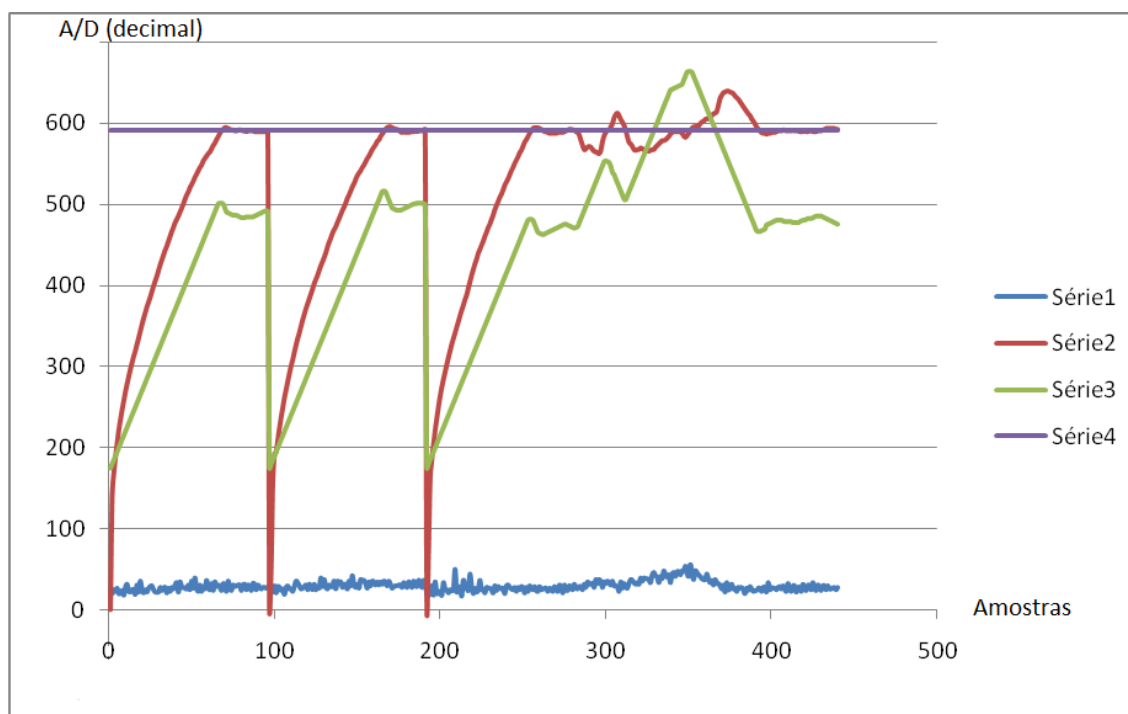


Gráfico 10 – Teste 5 com inversão do sentido de rotação.

Fonte: autoria própria.

Após a realização dos testes, pode-se observar que cada diferente sintonia apresenta aspectos positivos e negativos, de modo que a melhor sintonia, ou a mais adequada, irá depender da planta utilizada e das exigências da sua aplicação. Pode-se observar também que

os efeitos das ações de controle são bastante peculiares, não só devido à planta utilizada, mas principalmente pela forma de implementação do controle.

A maneira como foi implementado o controle Dedicado no *firmware*, fez com que as ações de controle tivessem reações diferentes e interferissem umas nas outras e em outros aspectos do controle. É possível constatar que a ação de controle com maior influência no sistema é a ação do Ganho Variável, enquanto a ação Amostral e de Atraso, apesar de também serem fundamentais, são mais sutis, possibilitando um ajuste fino. Contudo, configurando os parâmetros de controle é possível obter uma resposta com as características desejadas.

8 CONSIDERAÇÕES FINAIS

O desenvolvimento de um sistema de controle de motor para instrumentos cirúrgicos demonstrou ser um projeto complexo, envolvendo diversas áreas de conhecimento, no qual cada detalhe é significativo. A partir dos resultados obtidos pode-se concluir que o sistema de controle de motor cc microcontrolado oferece uma resposta bastante satisfatória, além de possibilitar uma análise mais profunda do controle.

Visando a aplicação cirúrgica do sistema, pode-se concluir que este oferece um funcionamento seguro, que corresponde às especificações propostas. Desenvolvido tendo em vista tal aplicação, o sistema alcançou um resultado satisfatório, apresentando uma resposta rápida e estável, tanto em relação ao problema servo quanto regulatório. Foram atingidas características importantes para instrumentação cirúrgica, como alta precisão, controle de torque e rampas de acionamento que evitam trancos e alterações bruscas.

Contudo, este sistema também possibilita outras aplicações, podendo ser utilizado para qualquer tipo de controle de motor de corrente contínua de 24VCC. O sistema pode ser utilizado para outros motores cc, sendo necessário apenas configurá-lo para a planta que será utilizada, o que se mostrou possível nos testes realizados. Para controlar outro motor seria necessário levantar a tabela de conversão de valores para a rotação e alterar os parâmetros de controle para encontrar uma sintonia adequada.

Com base nos resultados obtidos e nos gráficos levantados, pode-se concluir também que o sistema dá margem a estudos mais aprofundados do controle. O controle Dedicado, pela forma que foi implementado, apresentou respostas satisfatórias, atingindo o objetivo com êxito. Assim, o sistema também pode ser utilizado como planta para estudos e experiências referentes a formas de controle através *softwares* e *firmwares* em linguagem C.

REFERÊNCIAS

KAVO. **Integrated Technology for Implantology and Endodontics**. Alemanha, Biberach/Riss: KaVo Dental GmbH, 2008. (disponível em www.kavo.com).

KOSOW, Irving L. **Máquinas Elétricas e Transformadores**. 4ª ed. Porto Alegre, Rio de Janeiro: Editora Globo, 1982.

LEPKOWSKI, Jim. **DS00894A, Motor Control Sensor Feedback Circuits**. USA: Microchip Technology Inc., 2003.

MARTINS, Nardênio Almeida. **SISTEMAS MICROCONTROLADOS**. São Paulo: Novatec Editora, 2005.

MICROCHIP TECHNOLOGY INC. **DS41303G, PIC18FxxK20 Data Sheet**. USA: Microchip Technology Inc., 2010.

MISCH, Carl E. **Implantes Dentais Contemporâneos**. 3ª edição São Paulo: Elsevier, 2009.

MONTGOMERY, Ross. MCDOWALL, Robert. **Fundamentals of HVAC Control Systems**. Burlington, USA: Elsevier Science, 2008.

OGATA, Katsuhiko. **Engenharia de Controle Moderno**. Rio de Janeiro: Editora Prentice/Hall do Brasil Ltda., 1982.

PAZOS, Fernando. **Automação de Sistemas e Robótica**. Rio de Janeiro: Axcel Books do Brasil Editora, 2002.

RYLEE, Mike. DS00893A, **Low-Cost Bidirectional Brushed DC Motor Control Using The PIC16F684**. USA: Microchip Technology Inc., 2003.

SCHILDT, Herbert. **C Completo e Total** – 3ª edição Revista e Atualizada. São Paulo: Makron Books do Brasil Editora Ltda., 1996.

SCHNEIDER, Guilherme. **Apostila e notas de aula de Controle de Processos Contínuos**. Tecnologia em Mecatrônica Industrial, Universidade Tecnológica Federal do Paraná, 2010.

APÊNDICE A – Tabela de conversão de valores para tensão.

Tabela de conversão do valor no conversor A/D para a rotação		
HEXA	DECIMAL	ROTAÇÃO
2F	47	2000rpm
3F	63	3000rpm
4F	79	4000rpm
5F	95	5000rpm
6C	108	6000rpm
7D	125	7000rpm
87	135	8000rpm
93	147	9000rpm
A0	160	10000rpm
B1	177	11000rpm
BE	190	12000rpm
C5	197	13000rpm
DA	218	14000rpm
EC	236	15000rpm
FC	252	16000rpm
100	256	17000rpm
10F	271	18000rpm
118	280	19000rpm
130	304	20000rpm
138	312	21000rpm
149	329	22000rpm
154	340	23000rpm
166	358	24000rpm
172	370	25000rpm
183	387	26000rpm
191	401	27000rpm
19D	413	28000rpm
1AD	429	29000rpm
1B9	441	30000rpm
1C9	457	31000rpm
1D9	473	32000rpm
1E4	484	33000rpm
1F7	503	34000rpm
204	516	35000rpm
211	529	36000rpm
21F	543	37000rpm
22C	556	38000rpm
238	568	39000rpm
24F	591	40000rpm