



**UNIVERSIDADE FEDERAL DE OURO PRETO
ESCOLA DE MINAS
COLEGIADO DO CURSO DE ENGENHARIA DE
CONTROLE E AUTOMAÇÃO - CECAU**



MATHEUS ALVES E FARNESE

**CONTROLE PID DE UM MOTOR DE INDUÇÃO
UTILIZANDO MICROCONTROLADORES: APLICAÇÕES
PRÁTICAS DIDÁTICAS EM TECNOLOGIA INDUSTRIAL**

**MONOGRAFIA DE GRADUAÇÃO EM ENGENHARIA
DE CONTROLE E AUTOMAÇÃO**

Ouro Preto

Julho/2014

MATHEUS ALVES E FARNESE

CONTROLE PID DE UM MOTOR DE INDUÇÃO
UTILIZANDO MICROCONTROLADORES: APLICAÇÕES
PRÁTICAS DIDÁTICAS EM TECNOLOGIA INDUSTRIAL

Monografia apresentada ao Curso de Engenharia de Controle e Automação da Universidade Federal de Ouro Preto como parte dos requisitos para obtenção do Grau de Engenheiro de Controle e Automação.

Orientadora: Adrielle de Carvalho Santana

Ouro Preto

Escola de Minas – UFOP

Julho/2014

RESUMO

O presente trabalho discorre sobre a construção de uma estrutura para controle de velocidade PID de motor de indução trifásico. A pretensão é utilizar os recursos desenvolvidos para fins didáticos nas aulas do curso de Engenharia de Controle e Automação da Escola de Minas/UFOP. Para tal, foi utilizado um inversor de frequência controlado por uma placa de circuito microcontrolada. O trabalho apresenta uma discussão acerca de sistemas de controle e controladores PID bem como uma revisão bibliográfica sobre microcontroladores.

O trabalho apresenta também, detalhadamente, o desenvolvimento de uma placa de circuito impresso microcontrolada por um PIC18f4550, utilizada para enviar um sinal analógico para um inversor de frequência, responsável por realizar o acionamento de potência do motor. Para que seja possível, a aplicação foi utilizada a porta *CCP* para gerar um pulso modulado para controle de velocidade e a porta *Timer1* para reconhecer e contar os pulsos de um PWM gerado por um *encoder* acoplado ao motor. Além disso, é discutido nesse texto toda a eletrônica necessária para viabilizar a aplicação, como circuitos ceifadores, filtros e circuitos de amplificação.

Relata-se também no trabalho sobre o desenvolvimento de um sistema supervisor, na linguagem *C#* na plataforma Visual Studio, para supervisão do controle de velocidade por meio da comunicação USB HID. A comunicação USB permite ao usuário do sistema de supervisão plotar gráficos e analisar a variação da velocidade em velocímetros.

Por fim, é discutido sobre projeto de controladores PID pelo método empírico de Ziegler-Nichols e sua necessária discretização para implementação em microcontroladores, por meio do método de Tustin.

Este trabalho traz um conteúdo vasto dentro da área de controle e automação, passando pelos conteúdos de eletrotécnica, eletrônica, programação e sistemas de controle.

Palavras Chave: microcontroladores, controle de velocidade, PID, sistemas de controle, eletrônica, *C#*

ABSTRACT

The present work talks about the construction of a structure to do a PID speed control of a three-phase induction motor. The goal is use the resources developed, for teaching purposes in the classes of the course of Control and Automation Engineering from the School of Mines/UFOP. For this, it was used a frequency inverter controlled by a micro controlled circuit board. The work presents a discussion about control systems and PID controllers as well as a literature review about microcontrollers.

The work presents also, in detail, the development of a printed circuit board micro controlled by a PIC18f4550, used to send an analogic signal to a frequency inverter, responsible for perform the power actuation of the motor. For the application to be possible, a *CCP* gate was used to generate a modulated pulse for the speed control and the *Timer 1* gate to recognize and count the pulses from a PWM generated by an encoder linked to the motor. Besides, it is discussed in this text all the electronics necessary to enable the application, like the limiter circuits, filters and amplification circuits.

It is reported also in the work about the development of a supervisory system, in the C# language in the Visual Studio platform, to the supervision of the speed control through USB HID communication. The USB communication allows the supervision system user plot graphics and analyze the speed variation in speedometers.

Lastly, it is discussed about the PID controllers project using the Ziegler-Nichols empirical method and its necessary discretization to the implementation in microcontrollers, through Tustin method.

This work brings a wide content inside the control and automation field, going through the contents of electrotechnical, electronics, programming and control systems.

Keywords: microcontrollers, speed control, PID, control systems, electronics, C#.

LISTA DE FIGURAS

Figura 2.1 Diagrama de Blocos de um Sistema em Malha Aberta	16
Figura 2.2 Diagrama de Blocos de um Sistema em Malha Fechada.....	17
Figura 2.3 Descrição Gráfica da Estabilidade do Sistema	178
Figura 3.1 Sinal de Controle	21
Figura 3.2 Diagrama de Blocos do Controlador Proporcional.....	22
Figura 3.3 Diagrama de Blocos do Controlador Proporcional-Integral	23
Figura 3.4 Diagrama de Blocos do Controlador Proporcional-Derivativo	23
Figura 3.5 Sinal de Resposta de um Controlador PID Sintonizado	24
Figura 4.1 Onda PWM	29
Figura 5. 1 Desenho Esquemático do Inversor de Frequência.....	33
Figura 5.2 Sinal de Saída do Inversor	33
Figura 5.3 Inversor de Frequência	334
Figura 5.4 Curva de Ganho Analógico.....	335
Figura 5.5 Representação do Circuito Desenvolvido no Software Proteus	337
Figura 5.6 Configuração do Amplificador Não-Inversor	40
Figura 5.7 Pinos de Ligação do Inversor de Frequência.....	41
Figura 5.8 Construção da Placa de Circuito Impresso	44
Figura 8.1 Placa de Circuito Impresso	56
Figura 8.2 Trilhas da Placa de Circuito Impresso.....	57
Figura 8.3 Tela do Supervisório Durante Simulação.....	58

Sumário

RESUMO	3
ABSTRACT	4
1 INTRODUÇÃO	9
1.1 Um Pouco de História.....	11
1.2 Objetivo.....	12
1.3 Justificativa.....	12
1.4 Metodologia	13
2 CONTROLE DE SISTEMA	14
2.1 Definições Básicas.....	15
2.2 Controle em Malha Aberta e Fechada	16
2.3 Estabilidade de Sistemas de Controle.....	17
2.4 Ações de Controle e Controladores	19
3 CONTROLE PID.....	20
3.1 Controle Proporcional.....	21
3.2 Controle Proporcional-Integral	22
3.3 Controle Proporcional-Derivativo.....	23
3.4 Controle Proporcional-Integral-Derivativo.....	24
3.5 Sintonia de um Controlador PID.....	25
3.5.1 Ziegler-Nichols.....	25
4 MICROCONTROLADORES	27
4.1 Interrupções.....	28
4.2 Portas	28

4.3	Comunicação Serial.....	28
4.4	Modulação por Largura de Pulso (PWM).....	29
5	MONTAGEM PROPOSTA.....	30
5.1	O Moto de Indução Trifásico	30
5.2	Inversores de Frequência	32
5.2.1	Parametrização do Inversor	33
5.3	Elemento Sensor de Velocidade.....	35
5.4	Circuito de Controle.....	36
5.4.1	Circuito Ceifador de Onda	38
5.4.2	Filtro Passa-Baixa	39
5.4.3	Circuito de Amplificação	40
5.4.4	Comunicação USB	41
5.4.5	Placa de Circuito Impresso	42
6	LÓGICA DE CONTROLE E DESENVOLVIMENTO DE SUPERVISÓRIO	45
6.1	Programação do PIC.....	45
6.1.1	Sinal PWM	45
6.1.2	Contador de Giros	46
6.1.3	Fuses.....	48
6.1.4	Gravação do PIC.....	48
6.2	Sistema Supervisório	49
6.2.1	Interface Homem-Máquina (IHM).....	51
6.2.2	Comunicação USB-Serial.....	52
7	PROJETO DO CONTROLADOR	53

7.1 Método Empírico.....	53
7.2 Controlador Digital.....	53
8 RESULTADOS ALCANÇADOS	56
8.1 Sugestão de Trabalhos Futuros	58
9 CONCLUSÃO.....	60
REFERÊNCIAS BIBLIOGRAFICAS.....	61

1 INTRODUÇÃO

Controlar é uma expressão muito presente em nosso cotidiano. Controlar processos, controlar variáveis, controlar índices e parâmetros, são hábitos comuns ao ser humano para que possa lidar com as mais diversas situações, mantendo-as sempre de acordo com o esperado. É simples notarmos essa prática nas mais rotineiras atividades, por exemplo, quando uma pessoa controla uma TV, nada mais do que estipula ao aparelho uma variável chamada canal, para que a antena capte o sinal correspondente ao valor estipulado e transmita ao usuário. Outro exemplo, ainda mais simples que evidencia o hábito do controle, se encontra no simples fato de abirmos ou fecharmos um chuveiro de forma a manter a água numa temperatura que julgamos adequada ao banho.

Baseado nesses princípios básicos surge uma das mais emergentes ciências da atualidade, a Engenharia de Controle. Com um leque de atuação bem mais amplo do que situações rotineiras, essa ciência propõe técnicas de controle se utilizando de recursos tecnológicos para controlar diversos tipos de processos ou atividades. Dentro desse contexto o ato de controlar se assemelha a definição já citada, contudo ganha uma conotação bem mais vasta. Portanto, para a engenharia de controle, controlar processos significa manter o valor de determinada variável dentro de uma faixa aceitável de operação, e dentro dessa faixa, atingir valores convenientes ao funcionamento desejado. E em muitos casos vai além, buscando o valor ótimo de operação, com o qual atingirá melhores resultados, mais próximos do esperado.

Contudo, realizar o controle de processos nem sempre é tão simples como pode parecer. Existe uma sucessão de fatores físicos inerentes ao processo ou externo a ele que acabam por interferir nos resultados deste, afastando a variável que se deseja controlar do valor esperado. E é este exatamente o maior trabalho de especialistas da área, desenvolver estruturas que funcionem como um comandante da malha de controle, que tratem os distúrbios inerentes fazendo com que a saída do sistema se mantenha dentro do valor esperado. Essas estruturas que se utilizam de recursos computacionais e eletroeletrônicos para controlar o processo são chamados de controladores, e quanto mais bem elaborados estes forem melhor a resposta obtida. Ou seja, controladores bem sintonizados acabam por nos retornar uma resposta mais satisfatória e livre de erros.

E para que o sistema de controle funcione dentro do esperado, tratando o sinal de entrada do sistema de forma a leva-lo ao valor que desejamos, três elementos são fundamentais, os

sensores, os transmissores e controladores. Os sensores são os responsáveis por enxergar o estado do sistema, detectando o valor da variável controlada, e gerando um sinal elétrico que realimentará o sistema. O transmissor por sua vez realiza a transmissão desse sinal detectado pelo sensor até o controlador, elemento que comandará o sistema para que este atue no sentido de sempre corrigir o sinal para valores que tendem a se estabilizar em valores próximos ou iguais ao valor desejado. A incumbência de corrigir a variável medida, baseado no resultado do cálculo realizado pelo controlador cabe aos elementos finais de controle, que atuarão sobre o sistema físico.

É importante saber que a existência em si do sistema de controle não garante em nada que o sistema estará realmente controlado e estável. Para tal precisamos de uma estratégia de controle bem elaborada e com parâmetros bem dimensionados, de acordo com as necessidades de nosso processo. Existem diversos métodos já elaborados para a realização de controle tais como, controle ON/OFF, controle por antecipação, controle por lógica fuzzy e umas das mais clássicas e com maior aplicação no mercado atual, o controle Proporcional-Integral-Derivativo(PID). Estes métodos permitem um bom funcionamento do controlador, para que possa atuar sobre o processo de maneira acertada, levando a variável ao valor desejado no menor tempo possível e com baixo overshoot. O escopo do presente trabalho trará um estudo acerca desta última estratégia de controle citada, discorrendo sobre seus princípios de funcionamento, sua construção e aplicações, analisando os custos-benefícios de sua implementação, bem como os seus resultados.

A realização deste estudo se baseará em microcontroladores, que são hardwares passíveis de programação em alto nível para implementação de função específica. Esses componentes eletrônicos transistorizados, são estruturas robustas utilizadas na área de controle, que permitem trabalhar no desenvolvimento de aplicações diversas programadas. Segundo IBRAHIM (2002),

"Microcontrolador é um sistema composto no mínimo por um microprocessador, memória de programa, memória de dados, e dispositivos de entrada e saída. Alguns microcontroladores incluem componentes adicionais como temporizadores, contadores e conversores analógico digital".

O projeto apresentado apesar de não ser algo inovador, tem alta aplicabilidade em diversos tipos de processos, principalmente em processos industriais. Por essa razão mais

estudos e testes nessa área nunca serão exagerados. Além disso, o projeto é multidisciplinar e abordará várias das mais diversas áreas da Engenharia de Controle e Automação. Então aos interessados em processos de automação, em estratégias de controle, ou até mesmo em princípios de instrumentação e eletrotécnica, esse trabalho é uma excelente referência.

1.1 Um Pouco de História

Controlar processos e variáveis sempre foi uma necessidade.

"Em tempos passados, todo controle de processos era realizado por sinais pneumáticos, se utilizando de válvulas e cilindros para captação e transmissão de sinais pneumáticos oriundos do processo. Estes controles eram extremamente difíceis de serem feitos e muitos suscetíveis a erros" (BEGA, 2000).

O primeiro trabalho na área de controle automático foi desenvolvido por James Watt, o Regulador Centrífugo, utilizado para o controle de velocidade de uma máquina de vapor, ainda no século XVIII. A partir de então, em 1922, Minorsky provou que a estabilidade dos sistemas poderia ser descrita com equações diferenciais. Dez anos depois, Nyquist desenvolveu uma técnica para determinação da estabilidade de um sistema em malha fechada. E nos meados da década de 30, Hazrn introduziu a estes sistemas o conceito dos servomecanismos. E a evolução da área de sistemas de controle nunca mais foram freadas, pois com o advento da evolução tecnológica novas técnicas e conceitos foram sendo introduzidas no estudo destes sistemas.

"Ocorreu então um aprimoramento do controle de processos que começou a contar com técnicas de instrumentação para alcançar seus objetivos. Gradativamente foram sendo inseridos componentes eletrônicos, relés, sensores e atuadores eletromecânicos ao sistema de controle" (BEGA, 2000).

E mais recentemente recursos computacionais ganharam força e ocuparam todo o espaço dentro da área da engenharia de controle, sendo impossível desvincular o controle de processos ao desenvolvimento de recursos computacionais. Dessa forma os sistemas de controle perderam em tamanho e ganharam em robustez, além de passarem a ser menos suscetíveis a erros.

Mas ainda que os processos de controle estejam em um patamar bem elevado, ainda há muito caminho a ser trilhado. A cada dia surgem novos estudos principalmente na área de sistemas inteligentes, redes neurais e robótica, o que acrescenta em muito no potencial de desenvolvimento de controladores e automatismos. Por isso, estar sempre atento a novas tecnologias e novidades é essencial, contudo nunca se deve esquecer da origem desse estudo que forma uma base muito bem fundamentada e ainda atual.

1.2 Objetivo

O objetivo deste trabalho é desenvolver, auxiliado em recursos computacionais, um estudo sobre a técnica de controle clássica PID aplicada ao controle de sistemas lineares, desenvolvendo um controlador desse gênero utilizando um microcontrolador PIC18f4550 da Microchip.

Posteriormente pretende-se aplicar o controlador desenvolvido no controle de velocidade de um sistema motor trifásico. Avaliando o desempenho do controlador de forma a verificar sua eficácia na precisão. Para tanto pretende-se desenvolver uma placa de circuito impresso com um microcontrolador acoplado e um software de supervisão para averiguação do comportamento do motor mediante a implementação do controlador PID.

1.3 Justificativa

A proposta do projeto surgiu da necessidade de ampliar o conhecimento acerca de controladores, buscando entender um pouco melhor o seu funcionamento e a determinação de seus parâmetros. Além de entender o desenvolvimento do controlador PID em microcontroladores, conceito este que tem uma enorme aplicabilidade nos mais diversos tipos de indústrias.

Além disso o projeto surge com o intuito de contribuir com o desenvolvimento acadêmico do curso de Engenharia de Controle e Automação da Universidade Federal de Ouro Preto, considerando que a montagem desenvolvida servirá futuramente para complementar as aulas das disciplinas relacionados com controle, microcontroladores, eletrônica e eletrotécnica.

1.4 Metodologia

A proposta de desenvolvimento do trabalho é dada em etapas essenciais para que se atinja o resultado final que atenda o objetivo proposto. Abaixo são descritas sequencialmente as etapas do trabalho:

- Revisão Bibliográfica acerca dos conteúdos propostos pelo trabalho;
- Estruturação de uma placa de circuito impresso em software, estruturando todos os elementos necessários para a realização do controle proposto;
- Confecção da placa de circuito impresso;
- Desenvolvimento de sistema supervisorio, que permita supervisionar o controle da velocidade do motor;
- Obtenção dos parâmetros do motor e modelagem dinâmica deste equipamento e posterior discretização da função de transferência obtida para programação no microcontrolador;
- Programação do microcontrolador para a realização do controle de velocidade proposto;
- Discussão dos resultados;
- Conclusões.

2 CONTROLE DE SISTEMA

Existem no mundo diversas tarefas que não podem ser realizadas de forma simplista, com uma simples ação do homem. Mas ainda assim essas tarefas necessitam ser realizadas pelos mais diversos motivos. Para que seja possível a realização dessas tarefas é que surgem os sistemas que nada mais são que um conjunto de componentes interconectados com o objetivo de realizar uma dada tarefa. Por exemplo ao analisarmos um funcionamento de um carro, um motor de combustão sozinho jamais levaria algo ou alguém a lugar algum. Entretanto quando interconectamos vários elementos com funções específicas podemos criar um veículo automotor. Contudo nem sempre esses sistemas ficam restritos a conceitos físicos, muitas vezes podemos aplica-los a fenômenos abstratos ou dinâmicos como os encontrados na economia. Mas para efeitos do estudo aqui realizados nos restringiremos a sistemas de atuação física.

Definido sistemas é fácil perceber o que são sistemas de controle, os quais são sistemas que executam uma tarefa de forma a manter uma variável desejada dentro de uma faixa de operação determinada. Ou seja, sistemas de controle agem de forma a enxergar uma variável de entrada externa ao processo e atuar de forma a manter a variável de saída correspondendo ao valor desejado independente de distúrbios. Podemos citar como exemplo os equipamentos de ar-condicionado que recebem a temperatura desejada pelo usuário, compara com a temperatura ambiente e age de forma a resfriar ou esquentar o ar, mantendo a temperatura de acordo com o valor estipulado.

Olhando assim de forma simples parece bem mais fácil do que as coisas realmente são, isso porque existem diversos fenômenos físicos que interferem para que o controle de processos seja perfeito tal qual é esperado. Seja problemas na medição de variáveis ou na transmissão destas, ou até mesmo no acúmulo de energia pelo sistema, a saída do sistema nunca está exatamente como o esperado. Cabe ao responsável pela malha tratar esses problemas de forma a aprimorar o controle do processo.

A responsabilidade da malha cabe justamente ao elemento controlador, cuja tarefa é compensar esses distúrbios que acabam por afetar a resposta do sistema. Esse acaba por ser todo o segredo do sistema, controladores bem sintonizados acabam por nos dar respostas mais satisfatórias e livres de erro.

Na sequência será tratado com um pouco mais de cuidado essas malhas de controle, sua estabilidade, a forma de controle e o projeto de controladores. Entender esses conceitos é de suma importância para que o um controle seja alcançado com sucesso.

2.1 Definições Básicas

Antes de aprofundarmos mais no assunto de sistemas de controle é importante lançarmos algumas definições básicas necessárias para um bom entendimento da teoria.

- **Malha de Controle:** Conjunto de elementos de controle interconectados responsável pela ação sob determinada variável, para que esta trabalhe próximo ao seu Set Point.
- **Variável Controlada:** É a condição que se deseja manter em um determinado estado, ou seja, variável envolvida no processo, tal qual Nível, Vazão, Pressão, Temperatura, etc;
- **Valor Desejado (Set Point):** É o valor de referência para cada variável controlada.
- **Variável Manipulada:** É a variável que será interferida cuja a alteração influencia a variável controlada.
- **Distúrbio:** é um sinal que tende a afetar adversamente o valor da variável controlada. Este podem ser internos ou externos ao sistema.
- **Sinal de Erro:** É a comparação entre o valor da variável controlada com o Set Point estipulado, que mostra o erro que deve ser tratado.
- **Realimentação:** representa a relação direta entre a entrada e a saída do sistema. Ou seja, a introdução do sinal de erro a entrada do processo para que esse possa ser corrigido.

Com estes conceitos em mente, a discussão acerca de Malhas de controle e suas estratégias de controle é facilitada.

2.2 Controle em Malha Aberta e Fechada

Malhas de Controle podem ser construídas de duas maneiras distintas de acordo com o comportamento que o nosso sistema apresenta, podendo ser Malhas Abertas ou Malhas Realimentadas (Malha Fechada):

- **Malhas Abertas:** Neste tipo de malha a saída não influem em nada nas ações de controle. O sinal de erro não é comparado com a entrada de forma a reajustar o sistema, ou seja, não possui realimentação. Esses tipos de malhas são recomendados quando se conhece a relação entre entrada e saída e o sistema não estará exposto a distúrbios, pois qualquer variação que venha a atrapalhar a ação de controle alterando a saída do sistema (Variável Controlada), é impossível de ser corrigido. Este tipo de malha é bastante utilizada em sistemas que de acordo com a variação do tempo e não há variação na entrada do sistema (Variável Manipulada). Podemos citar como exemplo de sistemas que utilizam malhas abertas de controle as máquinas de lavar e os semáforos, que executam a mesma ação constantemente independentemente do valor da saída do sistema, ou seja, a máquina de lavar não analisa se as roupas realmente estão limpas e nem semáforos analisam se todos os carros já passaram do cruzamento ou não. A figura 2.1 ilustra esse tipo de sistema.



Figura 2.1

Figura 2.1 Diagrama de Blocos de um Sistema em Malha Aberta

- **Malhas Fechadas:** Já estes tipos de Malhas possuem realimentação, ou seja, a saída do sistema interfere diretamente nas ações de controle tomadas pelo controlador. Para estes casos as variáveis de saídas são medidas comparadas com o valor desejado, o erro obtido é levado novamente a entrada do sistema fazendo que o controlador instantaneamente se ajuste ao processo de forma a estabilizar a variável controlada o mais próximo possível do valor desejado estipulado. Estes tipos de montagem são menos suscetíveis a erros por distúrbio externos ou

internos do processo, uma vez que o distúrbio venha a afetar o sistema, o controlador automaticamente reconhece a variação e age de forma a estabilizar novamente a variável de saída. Estes tipos de malhas são usadas quando se é necessário um controle mais preciso ou quando o processo está exposto a muitos distúrbios. Casos como o do ar-condicionado, já citado anteriormente, se encaixa em sistemas de malha fechada. A figura 2.2 ilustra esse tipo de sistema.

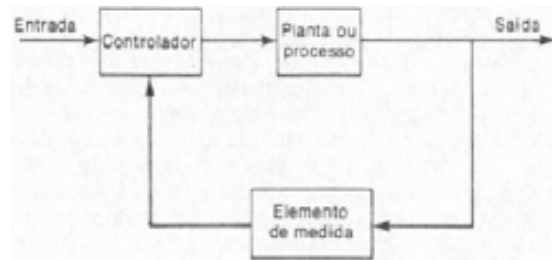


Figura 2.2 Diagrama de Blocos de um Sistema em Malha Fechada

FONTE: OGATA, 1985

Contudo, controladores em malha fechada possuem a necessidade de se trabalhar em cima da estabilidade do sistema, ao contrário dos controladores de Malha Aberta onde a estabilidade não chega a ser um problema. É importante na hora de se projetar controladores de malha fechada, se atentar para a estabilidade para que o sistema de fato convirja para o valor desejado, dentro do menor tempo possível e sem grandes oscilações. Por esse motivo a seguir será focado um pouco mais o problema da estabilidade de sistemas e suas aplicações.

2.3 Estabilidade de Sistemas de Controle

Para sistemas em malhas fechadas torna-se extremamente necessário a análise sobre a estabilidade do sistema, para que se saiba se a resposta do sistema converge a um valor determinado ou diverge para valores tendendo ao infinito fora da faixa de operação. Mais especificamente "Um sistema é dito estável se a resposta temporal for limitada para qualquer sinal de entrada também limitado" (PALHARES, 1998). Isso corresponde por exemplo a um sistema que tem um impulso como sinal de entrada, para que esse sistema seja considerado estável a saída deve convergir a zero à medida que o tempo tende a infinito. Graficamente um sistema só é considerável estável se todos os polos da função de transferência (equação no

domínio de Laplace que descreve o sistema a ser controlado) tem parte real negativa (semi-plano esquerdo do plano de Argand-Gauss).

Considerando o conceito de estabilidade demonstrado podemos definir os sistemas como Sistemas Estáveis, Sistemas Instáveis e Sistemas Marginalmente Estáveis. Ao contrário dos sistemas estáveis, os sistemas instáveis possuem pelo menos um dos polos com parte real positiva, e o valor de sua saída diverge para uma dada entrada, não se estabilizando em ponto algum. Já os sistemas marginalmente estáveis têm comportamento limítrofe, possuem um polo no 0, e estão no limite da estabilidade. A figura 2.3 descreve graficamente o comportamento de sistemas estáveis, marginalmente estáveis e instáveis.

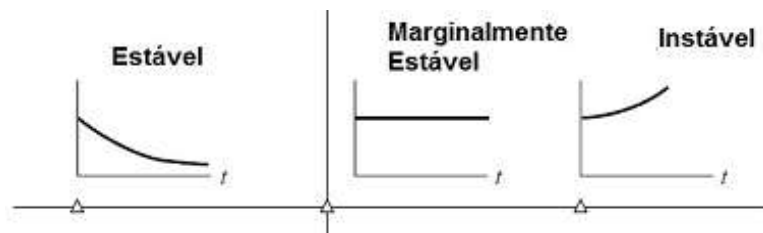


Figura 2.3 Descrição Gráfica da Estabilidade do Sistema

FONTE: PALHARES, 1998

Para que o controle seja realizado com perfeição devemos considerar que o sistema se estabilize em torno do valor desejado, no menor tempo possível e sem muita oscilação do sinal de saída. E os controladores se tornam importantes justamente para que este trabalho seja realizado. Uma vez que o controlador seja bem projetado, com os parâmetros estipulados de forma correta, o sistema terá um comportamento dentro do esperado se estabilizando em tempos menores, sem grandes oscilações e bem próximos ou no ponto de operação desejado. Por isso uma abordagem mais bem detalhada sobre esses equipamentos, estudando sua construção e sintonia, se torna de suma importância para este projeto.

2.4 Ações de Controle e Controladores

Os controladores são, por assim dizer, o “cérebro do sistema de controle”. Estas estruturas que na maioria das vezes são eletrônicas, mas podem também ser hidráulicas e pneumáticas, são responsáveis por receber o sinal de entrada ou o sinal de erro, interpreta-lo, e mandar sinais para que o sistema atue de forma a manter a saída sempre de acordo com o valor estipulado, este processo é denominado ação de controle. Contudo, para que este controlador atinja o seu objetivo de funcionamento ele deve ser construído de acordo com uma estratégia adequada ao funcionamento do sistema. Essas estratégias influenciarão diretamente na ação de controle tomada, provocando as variações desejadas na saída. Podemos citar como possíveis ações de controle tomadas por um controlador, o controle ON-OFF, o controle por lógica *fuzzy* e o controle PID.

Nos controladores ON-OFF, o atuador tem somente duas posições, isto é, Ligado - Desligado. Por esta razão apresenta um custo relativamente baixo, aliado a simplicidade. Neste tipo de controlador, a saída permanece num valor máximo ou num valor mínimo, dependendo do sinal do erro atuante, isto é, positivo ou negativo. Já o controle por meio de lógica *fuzzy*, é uma espécie de controlador baseado em aprendizado, que realiza o controle por meio da inferência de variáveis linguísticas, utilizando comando lógicos do tipo *if* e *then*. Mas o controlador PID (Proporcional-Integral-Derivativo) sem dúvidas é o mais tradicional e mais utilizado nos mais diversos tipos de controle, e é o mesmo que utilizaremos para construção do projeto proposto.

3 CONTROLE PID

Neste capítulo será abordado com mais detalhes sobre a ação de controle PID, sua especificação de parâmetros, além de suas vantagens e desvantagens. A revisão bibliográfica a seguir acerca de controladores PID é essencial para entendimento futuro da abordagem prática da qual este projeto se trata.

Ao estruturarmos o controlador com dada estratégia de controle, existe vinculada a essa estratégia uma ação de controle básica que atuará sobre o sistema. Essa ação de controle tem como princípio básico comandar os atuadores de forma que o sinal de saída do sistema convirja para o valor desejado, isento de erro em regime permanente, no menor tempo possível, com baixa oscilação e baixo *overshoot*. Para tal, as ações de controle desempenham funções específicas para que esse objetivo seja alcançado.

Especificamente para o controle PID é necessário a estipulação de três parâmetros fundamentais o K_p referente ao controle proporcional, que permite o sistema ter rápida resposta às variações do sinal de saída. O K_i referente ao controle integral, que elimina o erro em regime permanente. E finalmente o K_d referente ao controle derivativo, que leva o sistema a um comportamento mais estável. Cada um desses parâmetros influenciará de sua própria maneira sobre o sistema, alterando seu tempo de resposta, seu erro e sua oscilação. Devido a importância de cada parâmetro, abaixo abordaremos mais detalhadamente cada uma dessas técnicas de controle, que podem ser usadas separadas ou conjuntamente de acordo com as características que se espera do sistema, para que a resposta esteja de acordo com os padrões desejados à sua aplicação. A figura 3.1 ilustra um sinal controlado genérico.

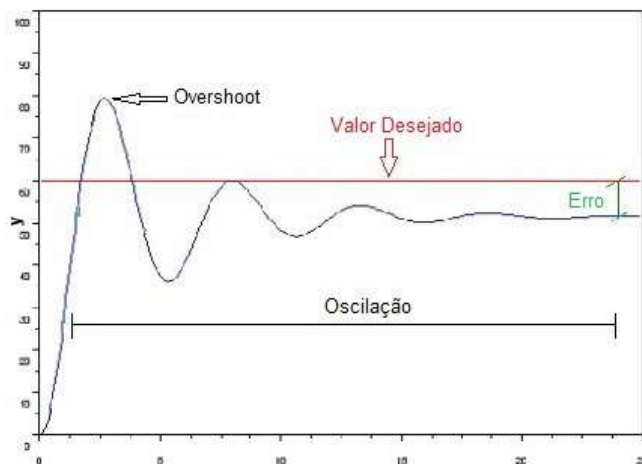


Figura 3.5.1 Sinal de Controle

3.1 Controle Proporcional

O controle proporcional puro é normalmente aplicado em muitos processos contendo constante de tempo simples, e responde rapidamente tanto aos distúrbios como às alterações do ponto de ajuste. Possui, entretanto a característica normalmente indesejável de apresentar um erro residual no estado estacionário (*offset*). A sintonia é relativamente fácil de ser obtida, pelo ajuste de um único parâmetro ($K_p = \text{Ganho Proporcional}$).

Para controle proporcional, quanto maior o K_p menor o erro (offset) na saída do sistema. Contudo quanto maior o ganho maior as oscilações e maior o tempo gasto para estabilização, portanto um K_p muito grande pode levar o sistema à instabilidade. Logo, o controle proporcional utilizado isoladamente não é suficiente para levar o sistema ao valor desejado, pois neste tipo de ação sempre vai haver um erro em regime permanente. O controle proporcional pode ser descrito pela equação (3.1) e representado pelo seguinte diagrama de blocos na figura 3.2:

$$\frac{R(s)}{C(s)} = \frac{K_p G(s)}{1 + K_p G(s)} \quad (3.1)$$

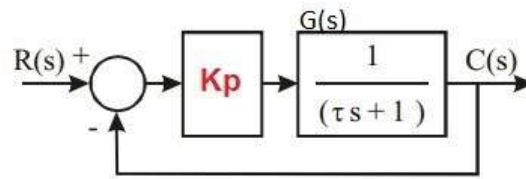


Figura 3.2 Diagrama de Blocos do Controle Proporcional

3.2 Controle Proporcional-Integral

É, com certeza, a ação de controle mais utilizada em controle de processos. Não apresenta offset associado ao controle proporcional puro, com vantagem de aumentar a velocidade de resposta em relação à ação integral isoladamente. Devido à presença da ação integral, a estabilidade da malha de controle diminui. Há riscos da saturação pelo modo integral, sendo que, neste caso, o controlador continua integrando o erro, mesmo sem haver correção efetiva, o que prejudica o controle do processo. É muito utilizado no controle de nível, vazão, pressão e outras variáveis que não apresentam atrasos muito grandes.

A ação PI devido à sua característica de variar a saída sempre que houver erro, faz com que se elimine o *offset* quando o processo atingir um estado estável.

Para tanto devemos ajustar o parâmetro K_i para que o sistema se comporte assim como o esperado, e para um K_i grande temos uma melhor aproximação da resposta à variável desejada contudo a velocidade de resposta se torna lenta. Já para um K_i muito pequeno, o comportamento da saída passa a oscilar demais e o sistema acaba demorando para se estabilizar. Considerando $e(t)$ como a entrada do controlador e $u(t)$ como a saída, seguem as equações 3.2 e 3.3 e a figura 3.3 com a representação do controle integral:

$$u(t) = K_p(e(t) + \frac{1}{T_i} \int_0^t e(T)d(T)) \quad \text{onde} \quad K_i = \frac{K_p}{T_i} \quad (3.2)$$

$$U(s) = (K_p + \frac{K_i}{s})E(s) \quad (3.3)$$

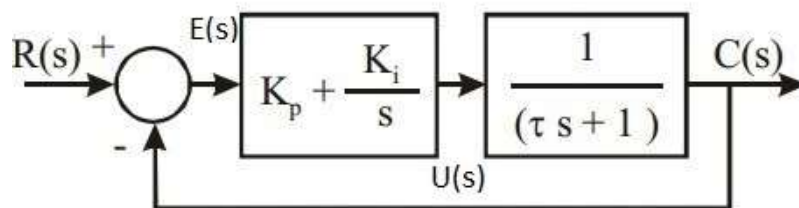


Figura 3.3 Diagrama de Blocos do Controle Proporcional-Integral

3.3 Controle Proporcional-Derivativo

Trata-se de um controle efetivo quando se tem sistemas com algumas constantes de tempo. Resulta numa resposta mais rápida, com menor offset do que o controle proporcional puro, mas ainda assim, este desvio persiste. Em geral, ação derivativa aumenta a estabilidade da malha de controle. Quando se tem processos rápidos, como controle de vazão, não se recomenda a utilização de ação derivativa, a não ser que acompanhada da proporcional e da integral, resultando num controle PID.

Para a sintonia desse tipo de controlador devemos regular o parâmetro K_d . O ajuste desse parâmetro provoca uma melhor estabilidade do sistema mas não diminui o erro em regime permanente inerente ao controle proporcional. Podemos modelar o controle proporcional de acordo com as equações 3.4 e 3.5. Seu diagrama de blocos é ilustrado na figura 3.4.

$$u(t) = K_p(e(t) + T_d \left(\frac{d}{dt}\right)e(t)) \quad \text{onde} \quad K_d = K_p T_d \quad (3.4)$$

$$U(s) = (K_p + K_d s)E(s) \quad (3.5)$$

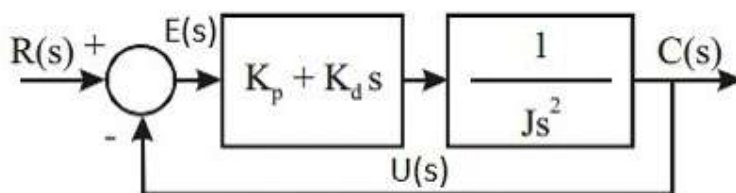


Figura 3.4 Diagrama de Blocos do Controle Proporcional-Derivativo

3.4 Controle Proporcional-Integral-Derivativo

Trata-se da ação mais complexa dentre as técnicas de controle convencional. Teoricamente, o controle PID resulta num melhor controle do que as técnicas PI ou PD. Na prática, porém, há dificuldade de ajuste dos parâmetros de sintonia. Esta ação é utilizada na indústria em controle de PH, temperatura e outras variáveis analíticas.

A adição da ação derivativa, às ações P e I, torna o controle PID, resolvendo o problema de resposta de uma forma definitiva, uma vez que a variável de processo é trazida para o valor original de uma forma rápida. Mas esta ação de controle tem seu ajuste dificultado, uma vez que é necessário o acerto de três parâmetros. Como a ação derivativa torna o controlador mais difícil de sintonizar, o controle de 3 modos deverá ser utilizado somente em determinados processos que realmente tenham seu desempenho bastante aumentado. A modelagem do controle PID nada mais é que uma união dos 3 parâmetros de controle, podendo ser descrita pela dada equação 3.6.

$$U(s) = (K_p + \frac{K_i}{s} + K_d s)E(s) \quad (3.6)$$

Um controlador bem ajustado nos traz um sinal de resposta sem *overshoot*, com rápido tempo de resposta e livre de erros em regime permanente, como demonstrado na figura 3.5 abaixo:

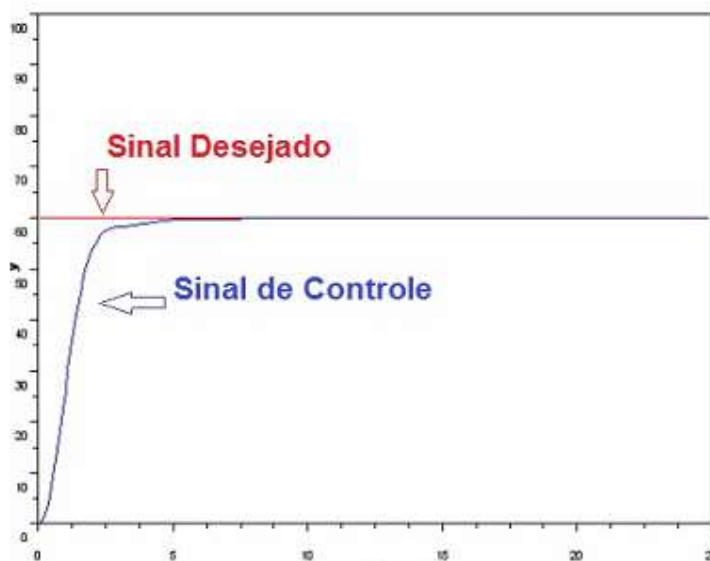


Figura 3.5 Sinal de Resposta de um Controlador PID Sintonizado

Contudo, a simples construção de um controlador PID não garante em nada que seu processo esteja de fato estável e respondendo de forma satisfatória ao sinal de entrada. Para tanto, é de estrita importância que o controlador esteja bem sintonizado, ou seja, seus parâmetros K_p , K_i , K_d , estejam condizentes ao sistema, e que realmente estejam atuando no sistema de forma acertada corrigindo o sinal na saída do sistema de acordo com o valor desejado. Por tal motivo é importante se conhecer os métodos de sintonia do controlador para que possamos fazer um ajuste final, na próxima seção uma discussão mais detalhada sobre o problema da sintonia de controladores é levantada.

3.5 Sintonia de um Controlador PID

Muitas vezes os parâmetros dos controladores PID são estipulados de forma aleatória utilizando a experiência dos operadores da planta controlada, ou simplesmente na tentativa e erro até que se encontre resultados adequados. Este método gera resultados imprecisos mas aceitáveis. Entretanto, para que a sintonia possa ser feita com maior exatidão, estudiosos da área propuseram diversos métodos baseados na ciência matemática do controle, os resultados encontrados utilizando estes métodos acabam por ser bem mais satisfatórios.

Existe uma infinidade de formas para realizar a sintonia do controlador, todos estes métodos se utilizam dos parâmetros inerentes à planta bem como a sua função de transferência para encontrar os valores dos parâmetros K_p , K_i e K_d . Dentre esses métodos focaremos o proposto por Ziegler-Nichols.

Cada método de sintonia nos fornece valores diferentes para os parâmetros do PID, isso por consequência irá alterar o tempo de resposta, a estabilidade, o *overshoot* e o erro em regime permanente do sistema. Cabe ao desenvolvedor analisar e escolher os parâmetros que melhor se respondem as suas necessidades. O trabalho irá propõe uma abordagem empírica para a sintonia do controlador, baseado no método de Ziegler-Nichols. O assunto de sintonia será tratado com mais detalhe no capítulo sobre a sintonia do controlador.

3.5.1 Ziegler-Nichols

Muitas das plantas principalmente aquelas empregadas nos processos industriais podem ser entendidas matematicamente como um relação entre a entrada e saída do sistema. Essas plantas mantem podem ser equacionadas de acordo com o seguinte modelo da equação 3.7:

$$G(s) = Y(s)/U(s) = (K_0 * e^{-sL})/Ts + 1 \quad (3.7)$$

Uma versão quantitativa linearizada deste modelo pode ser obtida por um experimento em Malha Aberta, ou por meio de simulação quando o modelo matemático é conhecido, por meio do procedimento a seguir:

- Mantendo a planta em Malha aberta, force-a até o ponto de operação

$$Y(t) = Y_0$$

para uma entrada,

$$U(t) = U_0$$

- Posteriormente em um tempo,

$$T_0$$

aplique um degrau à entrada do sistema que altere U_0 para U_{inf} .

- Amostre a saída da planta até que esta alcance um novo valor um regime permanente (novo ponto de operação).
- Finalmente calcule o parâmetro K_0 , da seguinte forma (equação 3.8):

$$K_0 = (y_{\infty} - y_0) / (U_{\infty} - U_0) \quad L = t_1 - t_0 \quad T = 2 * t_1 \quad (3.8)$$

Uma vez que tenhamos obtido os valores inerentes ao sistema podemos utilizar as equações propostas por Ziegler-Nichols para estipular os parâmetros do PID. Na tabela (3.1) podemos observar o cálculo para cada um dos parâmetros do PID de acordo com a estratégia escolhida.

Tabela 3.1 – Tabela para Obtenção de Parâmetros Ziegler-Nichols

Tipo de controlador	K_p	T_i	T_d
P	$\frac{T}{L}$	∞	0
PI	$0,9 \frac{T}{L}$	$\frac{L}{0,3}$	0
PID	$1,2 \frac{T}{L}$	$2L$	$0,5L$

4 MICROCONTROLADORES

De acordo com DENARDIN, 2014,

“Um microcontrolador é um sistema computacional completo, no qual estão incluídos uma CPU (Central Processor Unit), memória de dados e programa, um sistema de clock, portas de I/O (Input/Output), além de outros possíveis periféricos, tais como, módulos de temporização e conversores A/D entre outros, integrados em um mesmo componente. As partes integrantes de qualquer computador, e que também estão presentes, em menor escala, nos microcontroladores são: Unidade Central de Processamento (CPU), Sistema de clock para dar sequência às atividades da CPU, Memória para armazenamento de instruções e para manipulação de dados, Entradas para interiorizar na CPU informações do mundo externo, Saídas para exteriorizar informações processadas pela CPU para o mundo externo e Programa (firmware) para definir um objetivo ao sistema”.

Os microcontroladores controlam processos através de periféricos como displays e leds, a partir de um controle lógico realizado a partir de uma varredura realizada nos estados das entradas e saídas. A lógica criada não ocupa definitivamente a memória ROM do microcontrolador, o que dá ao microcontrolador a característica de ser reprogramável. Assim sendo, o desenvolvedor pode gravar a lógica estruturada quantas vezes julgar necessário, de forma que toda vez que o microcontrolador for alimentado o programa interno será executado. Outra característica fundamental destes equipamentos é o seu porte reduzido, isso pelo fato do componente ser constituído de uma única pastilha de silício encapsulada (popularmente chamada de CI ou CHIP).

Os microcontroladores tem grande aplicabilidade em controle de processos devido ao sua capacidade de processamento, proporções reduzidas e robustez. Esses componentes são programados em linguagem de alto nível C, o que torna o desenvolvimento bem mais fácil e dinâmico. Mas para tanto é necessário se conhecer ferramentas fundamentais deste equipamento, como as interrupções, os *Timers* e a geração de pulso modulado. Nos próximos tópicos serão tratados algumas dessas funcionalidades relevantes ao projeto, e serão omitidas tantas outras que não tem aplicabilidade dentro do projeto proposto.

4.1 Interrupções

“Uma interrupção serve para interromper o programa imediatamente. Desta maneira, podem-se tomar atitudes instantâneas. As interrupções são ações tratadas diretamente pelos componentes físicos (hardware), o que as torna muito rápidas e disponíveis em qualquer ponto do sistema. Desta forma, quando uma interrupção acontece, o programa é paralisado, uma função específica (definida pelo programador) é executada, e depois o programa continua a ser executado no mesmo ponto que estava. Elas servem para a solução de muitos problemas complexos” (SOUZA, 2009).

4.2 Portas

”Cada porta de I/O possui dois registradores que controlam suas funções: um registrador PORT e um registrador TRIS. O registrador PORT é utilizado para acesso aos pinos do uC: uma escrita no registrador PORT escreve nos pinos que estiverem configurados como saída, enquanto uma leitura irá ler o nível lógico presente nos pinos I/O. O registrador TRIS é utilizado para configurar cada pino da respectiva porta como entrada ou saída. Assim, cada bit do registrador corresponde a um pino da porta. Se o bit estiver em ‘1’ configura o pino como entrada, e se estiver em ‘0’ como saída” (PEREIRA, 2005).

4.3 Comunicação Serial

”Na comunicação serial, a transmissão de dados é feita dígito por dígito (bit a bit), ou seja, é transmitido um dígito binário de cada vez. A informação 001110010, por exemplo, é transmitida entre dois pontos, e na linha de transmissão colocado apenas um dígito binário de cada vez, começando pelo dígito binário menos significativo. Cada dígito binário permanece na linha de transmissão durante um tempo predefinido, chamado período do dígito binário, para que possa ser lido. A comunicação serial pode ter somente uma linha de transmissão. Esta é uma característica que faz com que ela seja uma forma muito utilizada de

transmissão de dados, principalmente em transmissões de longa distância” (ZANCO, 2005).

4.4 Modulação por Largura de Pulso (PWM)

“O sinal de modulação por largura de pulso (PWM) consiste num sinal de frequência fixa em que é variado o ciclo ativo da onda. É muito utilizado no controle de potência(fontes chaveadas, controles de velocidade, dentre outros), além de servocomandos e outros dispositivos” (PEREIRA, 2007).

”Este é o modo de funcionamento mais poderoso do módulo CCP (Compare, Capture, PWM) e pode ser utilizado em diversas aplicações, como, por exemplo, conversor digital/analógico, fontes chaveadas, dispositivos de prevenção de falta de energia (*nobreaks*), dentre outros. O PWM é um sinal digital com frequência fixa, mas com largura de pulso (ciclo ativo) variável. Pode-se dizer que o ciclo ativo é a parte do ciclo em que o sinal permanece em nível 1” (ZANCO, 2005).

Na figura 4.1 é possível visualizar o formato da onda PWM, e a equação 4.1 descreve a relação entre a onda quadrada do PWM e a tensão contínua relativa a essa onda.

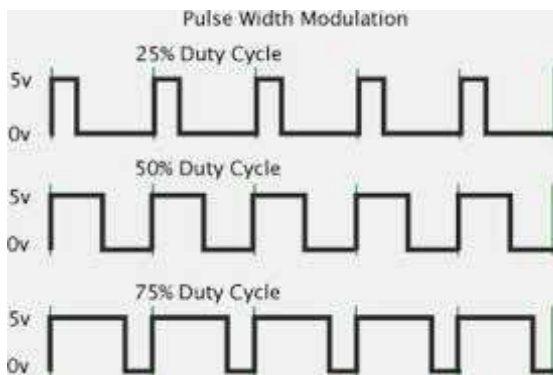


Figura 4.1 Onda PWM

$$V_{cc} = \frac{T_{cicloativo}}{T_{pwm}} * V \quad (4.1)$$

5 MONTAGEM PROPOSTA

O controle de velocidade de motores são utilizados para situações em que se deseja reduzir a velocidade nominal de rotação do motor para faixas de rotações mais baixas de forma a utilizar esse motor para aplicações mais específicas. O controle de velocidade é bastante utilizado em eletrodomésticos, ferramentas, bombas para líquidos, aparelhos para testes mecânicos, secadores de cabelo, ou berbequins.

Para o desenvolvimento deste projeto foi proposta uma montagem onde o circuito de controle projetado emitia um sinal de 0 a 10 Vcc, que alimentava uma entrada analógica do inversor de frequência. Este equipamento foi o responsável pela parte de potência do circuito, acionando o motor trifásico através da aplicação de frequências controladas pelo valor da entrada analógica, de acordo com a parametrização realizada. Alimentando o motor com uma frequência proporcional ao sinal analógico aplicado. Para se fechar a malha foi utilizado um *encoder*, que media a velocidade de rotação do motor. A descrição e funcionamento do motor serão relatadas na sequência do capítulo para que se possa se entender com maior clareza a montagem realizada.

É intuitivo pensar, com razão, que o controle de velocidade do motor pudesse ser realizado apenas com a utilização do inversor de frequência, uma vez que esse possui uma interface para utilização do usuário que permite a estipulação dos valores de frequência aplicados a este motor. Contudo optou-se por utilizar o inversor de frequência apenas para a parte de acionamento de potência do circuito pois o objetivo do trabalho não é tão somente controlar a velocidade, como também se projetar um controlador que fará esse controle com maior eficiência. Permitindo-se projetar parâmetros que otimizem a resposta do sistema, além de se dar uma aplicação da teoria revista acima. Além disso o fato de se realizar esse controle com microcontroladores permite o desenvolvimento do supervisor computacional para a visualização do funcionamento do sistema, atendendo assim a mais um dos objetivos do trabalho, tornando o mais didático possível para a utilização futura dos demais graduandos.

5.1 O Moto de Indução Trifásico

De acordo com Moreno et. al, 1997,

“Os motores elétricos exercem um importante papel na sociedade moderna industrial, estando presentes em diversos setores, tais como: a indústria, o

comércio, a agricultura entre outros. Tal ideia é facilmente confirmada quando se observa que eles utilizam quase 50% de toda energia consumida no mundo. Portanto o aperfeiçoamento e aprimoramento dos motores e acionamentos elétricos tem importância não só do ponto de vista tecnológico- científico, mas também, econômico-financeiro”.

”O motor de indução é o tipo de motor elétrico mais utilizado e difundido, tanto para motorização de sistemas, quanto para processos industriais. Sua principal vantagem é a eliminação do atrito de todos contatos elétricos deslizantes e uma construção bastante simples, o que possibilitou sua construção a um custo ainda mais baixo, sendo que estas máquinas são fabricadas para uma grande variedade de aplicações, desde alguns watts até muitos megawatts” (LEONARD, 1985).

Considerando a alta aplicabilidade do motor de indução trifásico, optou-se por tomar este tipo de motor como o objeto do trabalho. Optou-se por adotar um motor trifásico da WEG, acoplado ao módulo didático da MT03 da Hidro-Eletro, com pinos de fácil conexão e *enconder* já devidamente colocado junto ao eixo de rotação. A ligação realizada foi uma ligação em triângulo, indicada no próprio módulo didático como a ligação ideal para a tensão de alimentação de 220V.

”Mesmo com essas vantagens, os motores de indução não tinham muita importância até alguns anos atrás, quando se levava em consideração aplicações com velocidade variável, pois todas tentativas neste sentido necessitavam de um equipamento adicional, ou então, sofriam grandes perdas de potência. Embora fossem investigados os problemas da eficácia de controlar a velocidade dos motores de indução durante décadas, todas as soluções realizáveis até alguns anos atrás eram muito complicadas e/ou caras. Uma primeira solução foi obtida com relação às técnicas de modelagem, com o propósito de se obter um conjunto de equações dinâmicas mais simples e voltadas para aplicações de controle, mas sua implementação exigia grande esforço computacional, ou os conversores de potência eram inexistentes ou de desempenho insatisfatório” (VAS, 1990).

Nessa vertente é que surgiram os inversores estáticos de frequência, que associados com circuitos microprocessados poderiam contribuir em muito com a realização de um controle mais satisfatório dos motores de indução.

E é justamente com um inversor de frequência que a parte de potência do sistema é realizada. Não seria possível no projeto proposto realizar o acionamento se não por intermédio do inversor. A próxima seção enfocará com mais detalhes o funcionamento desses equipamento e sua aplicação.

5.2 Inversores de Frequência

Inversores de frequência são equipamentos destinados ao controle e variação de velocidade de motores elétricos de indução trifásicos e podem ser utilizados nos mais diversos segmentos industriais. Esses dispositivos são capazes de gerar uma tensão e frequência trifásicas ajustáveis, com a finalidade de controlar a velocidade de um motor de indução trifásico.

Em geral inversores são compostos de uma CPU, IHM, Interfaces e o circuito de potência. É na CPU onde são armazenados os parâmetros e dados do sistema. Além disso a CPU é responsável pela geração dos pulsos de disparo, através da lógica de controle preestabelecida, pelos usuários, para os IGBT's. Na IHM (Interface Homem Máquina é possível visualizar o que está ocorrendo no inversor e parametrizá-lo de acordo com aplicação. Já nas interfaces é que ocorre a captação do sinal vindo de algum circuito externo para controle do inversor. Esses sinais podem ser analógicos ou digitais, no caso em questão o sinal de entrada será analógico, oriundo do controle feito pelo microcontrolador.

Por fim os circuitos de potência são responsáveis pelo princípio de funcionamento dos inversores, que se baseiam em seis diodos retificadores situados no circuito de entrada do inversor. A onda retificada é filtrada por um capacitor. Posteriormente essa tensão contínua aciona um banco de transistores que chaveiam a tensão transformando em um pulso modulado com largura variada, cuja tensão média aplicada no motor tem um comportamento bem próximo ao senoidal. O circuito ilustrado na figura 5.1, demonstra esquematicamente o princípio de funcionamento do inversor e a figura 5.2 mostra o formato da onda na sua saída.

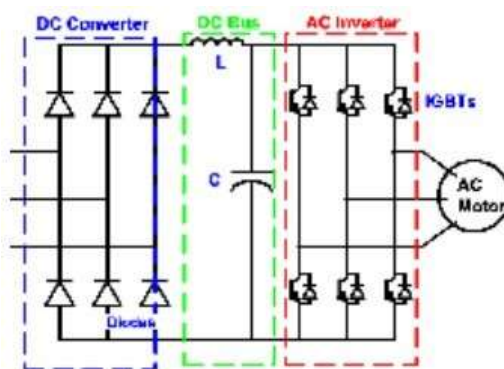


Figura 5.1 Desenho Esquemático Inversor de Frequência

FONTE: ALFA5ELETRICISTA, 2014

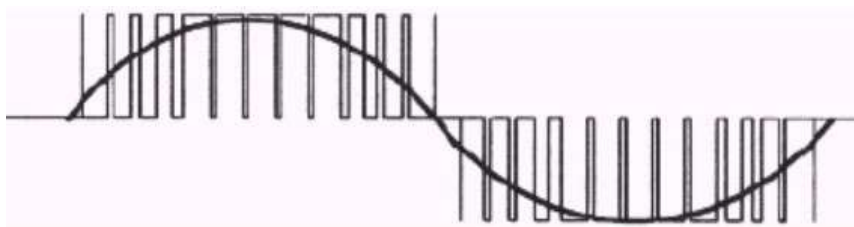


Figura 5.2 Sinal de Saída do Inversor

FONTE: ALFA5ELETRICISTA, 2014

Foi exposto todo o princípio de funcionamento do inversor, porém nada foi citado sobre o esquema de parametrização deste equipamento. O inversor só funciona de acordo com uma certa lógica determinada pelo usuário através da definição dos parâmetros. No próximo tópico será tratado sobre o método de parametrização do inversor e os parâmetros mais importantes utilizados para o controle de velocidade do motor como proposto.

5.2.1 Parametrização do Inversor

“Para que o inversor funcione a contento, não basta instalá-lo corretamente. É preciso “informar” a ele em que condições de trabalho irá operar. Essa tarefa é justamente a parametrização do inversor. Quanto maior o número de recursos que o inversor oferece, tanto maior será o número de parâmetros disponíveis. Existem inversores com tal nível de sofisticação, que o número de parâmetros ultrapassa a marca dos 900” (ALFA5ELETRICISTA, 2014).

Para se conseguir parametrizar o inversor é preciso primeiro entender o que se deseja

realizar com clareza. Dentro dos inúmeros parâmetros que se pode ajustar o usuário pode se confundir e se atrapalhar em sua parametrização. Na montagem realizada foi utilizado o inversor CFW 10 da WEG, que pode ser visto na figura 5.3, portanto basearemos a descrição da parametrização baseados no inversor utilizado.



Figura 5 Erro! Nenhum texto com o estilo especificado foi encontrado no documento..3 Inversor de Frequência

FONTE: MOTORELETRICO, 2012

Como pode-se observar na imagem os botões na tela podem ser usados para parametrizar o inversor, utilizando os botões "up" e "down" para se escolher o parâmetro desejado e o botão "P" para selecioná-lo. Uma vez selecionado o parâmetro deve-se voltar aos botões "up" e "down" para se escolher o valor que se deseja dar ao parâmetro. O significado de cada parâmetro pode ser encontrado no manual do equipamento, com descrição detalhada do funcionamento de cada um. Por motivos lógicos, serão demonstrados na sequência apenas os parâmetros mais relevantes ao projeto.

- Parâmetro 000: Utilizado para tornar acessível os demais parâmetros. Para tanto deve-se inserir valor 5, como espécie de senha, para o esse parâmetro.
- Parâmetro 005: Uma vez acionado o parâmetro 5, será mostrado na IHM o valor de frequência que está sendo entregue ao motor.
- Parâmetro 229: é importante se atentar a esse parâmetro, pois se este for diferente de 0 as teclas da IHM não funcionam e não é possível parametrizar o

inversor.

- Parâmetro 221: Deve ser dado valor 1 para este parâmetro, com o intuito de habilitar a entrada analógica AI1.
- Parâmetro 234: Este parâmetro define os ganhos da entrada analógica segundo a curva de ajuste que é apresentada na figura 5.4.

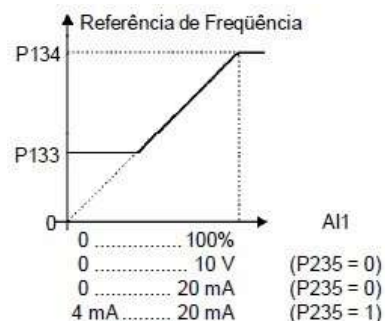


Figura 5.4 Curva de Ganho Analógico

FONTE: ALFA5ELETRICISTA, 2014

- Parâmetros 133 e 134: Como é perceptível pela figura, a definição do ganho no parâmetro 234 depende diretamente dos parâmetros 133 e 134. Esses parâmetros definem os valores mínimo e máximo da frequência de saída (motor) quando o inversor é habilitado. É importante notar que os valores máximos de frequência não devem ultrapassar a velocidade limite de rotação do motor.
- Parâmetro 235: Define o tipo do sinal das entradas analógicas. Para se trabalhar com valores de referência entre 0 e 10 Vcc deve-se definir valor 0 para esse parâmetro.

5.3 Elemento Sensor de Velocidade

Para se fechar a malha do circuito é essencial a utilização de algum elemento sensor que retorne a velocidade de giro do motor, para que o controle possa ser realizado, e para realizar a medição de velocidade podemos utilizar dispositivos eletrônicos chamados *encoders*.

Os *encoders* são sensores ópticos que transmitem um raio de luz infravermelho de

uma extremidade a outra constantemente. Toda vez que o feixe de luz é interrompido o sensor indica nível baixo para o sinal em sua saída. Desse forma a ser acoplado ao motor, juntamente com uma placa circular com metade de superfície translúcida e a outra metade opaca, o *enconder* consegue contar o número de giros do motor em um determinado espaço de tempo. A partir desse princípio de funcionamento obtêm-se um sinal PWM na saída do elemento sensor. Uma vez que determinado o tempo da amostragem dos pulsos vindo do *enconder* é possível saber a velocidade de rotação do motor, em rotações por minuto, fazendo uma proporcionalidade simples.

Assim sendo foi acoplado ao eixo do motor um *encoder* do tipo PHCT203, embutido em uma placa alimentada por uma tensão de 220V que retransmitia a saída um sinal PWM com amplitude máxima de 15V.

5.4 Circuito de Controle

A etapa de controle do sistema foi desenvolvida por meio da programação de um microcontrolador PIC 18F4550 da Microchip.

”Este modelo possui 40 pinos, dos quais 35 podem ser configurados como I/O, e diversos periféricos, tais como memória EEPROM de 256 bytes, um módulo CCP e ECCP, um módulo SPI e I2C, 13 conversores A/D de 10bits de resolução com tempo de aquisição programável, dois comparadores analógicos, uma comunicação EUSART, um TIMER de 8bits (TIMER2) e três de 16bits (TIMER0, TIMER1 e TIMER3), um módulo de detecção de alta/baixa voltagem (HLVD), além de ter um módulo USB 2.0 capaz de operar no modo *low-speed* (1.5Mbps) ou *full-speed* (12Mbps)” (MIYADAIARA, 2009).

O microcontrolador utilizado é bastante robusto, e utilizado para aplicações mais complexas, portanto está superdimensionado ao projeto proposto. Contudo este foi utilizado em virtude de sua disponibilidade para a realização do trabalho. Mas com certeza é necessário se avaliar com bastante cautela qual microcontrolador utilizar, de forma que se opte pelo microcontrolador mais adequado ao projeto, considerando as necessidades de utilização e o custo-benefício.

Para a realização do controle de velocidade foram utilizadas as seguintes portas listadas na sequência:

- Porta 1: Utilizada normalmente com a função de ”*MasterClear*”, utilizada no pro-

jeto apenas como segurança contra a possibilidade de haver algum travamento de software do microcontrolador. Importante notar, que uma vez colocada no sistema deve-se mantê-la acionada constantemente para que o PIC funcione normalmente.

- Portas 13 e 14: Utilizadas para ligar o *crystal* de 20MHz responsável por gerar o "clock" do sistema do PIC.
- Porta 15: A função "Timer1" foi utilizada para captar o sinal de entrada PWM gerado pelo *encoder* e contar o número de pulsos dados durante o intervalo de tempo de estouro do Timer0, interno do sistema, de 1s. Dessa forma foi possível captar as rotações por segundo do sistema que multiplicado por 60 nos fornece as rotações por minuto (rpm) do motor.
- Porta 17: A porta 16 foi utilizada como uma porta CCP (Capture, Compare, PWM). Através dessa porta PWM, pode-se gerar o pulso modulado para controle da velocidade do motor. Ou seja, o sistema responde a variação de velocidade do motor gerando um pulso modulado de maior ou menor largura, aumentando ou diminuindo o giro do motor dessa forma.
- Portas 23,24,25 e 26: Portas utilizadas para alimentação e comunicação USB do microcontrolador com o computador.
- Portas 39 e 40: Utilizadas para gravação ICSP do PIC.

A montagem do circuito foi realizada antes no software Proteus, já simulando as conexões do PIC e todos os circuitos inseridos para o devido controle, o circuito é exposto na figura 5.5.

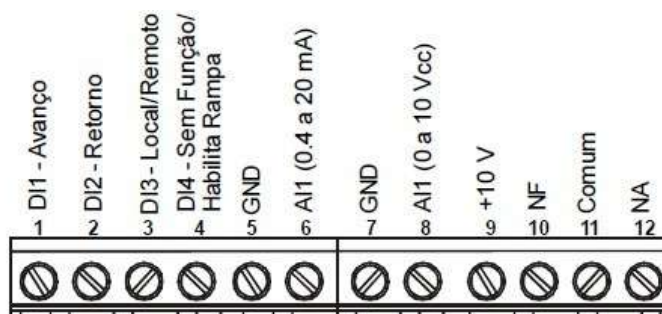


Figura 5.5 Representação do Circuito Desenvolvido no Software Proteus

O microcontrolador, como já foi dito, é mais que suficiente para processar o controlador desenvolvido. Contudo não basta o controlador no circuito de controle para que seja possível realizar a aplicação proposta. Para tanto foram construídos eletrônicos complementares na placa de circuito impresso para que se pudesse tratar o sinal enviado ao inversor de frequência, bem como o sinal recebido do *encoder*. Nos próximos tópicos será exposto, separadamente, a construção e a necessidade de cada um dos circuitos desenvolvidos para o controle de velocidade do motor através do inversor de frequência.

5.4.1 Circuito Ceifador de Onda

De acordo com BOYLESTAD, 1996,

”...Existe um ponto onde a aplicação de uma tensão também negativa resulta em uma mudança brusca na curva característica do diodo. A corrente aumenta a uma taxa muito rápida na direção oposta da região de tensão positiva. O potencial de polarização reverso que resulta desta brusca mudança na curva característica é chamado potencial Zener, e é dado pelo símbolo V_z ... esta mudança brusca na curva característica em qualquer nível é chamada de região Zener, e os diodos que empregam apenas esta porção da curva de uma junção p-n são chamados diodos Zenner... Há uma variedade de circuitos com diodos chamados ceifadores que possuem as características de ”ceifar” uma porção do sinal de entrada, sem distorcer o restante da forma de onda alternada. Dependendo da orientação do diodo, a região positiva ou negativa do sinal de entrada é ”ceifada”.

A onda PWM gerada pelo *encoder* tem tensão de 15 V e possui muito ruído, principalmente devido à interferência do inversor de frequência. Assim sendo, é necessário abaixar a tensão fornecida pelo *encoder*, bem como filtrar um os ruídos de forma a não interferir na leitura dos pulsos pelo Timer 1 na porta 15. Nesse sentido, a onda quadrada que saí do encoder é filtrada por um capacitor colocada em paralelo com uma resistência na saída, constituindo o filtro passa baixa. Foi colocado também um diodo Zenner de 5,6 V, reversamente polarizado, para ceifar a onda quadrada até valores que o PIC possa trabalhar, considerando que este componente trabalha com tensões restritas de na faixa de 0 a 5V. Ou seja, o sinal que entra pela porta 15 do microcontrolador se torna mais livre de ruídos, além de passar a possuir amplitude máxima de 5,6V.

5.4.2 Filtro Passa-Baixa

De acordo com KOERICH, 2014,

”O filtro passa-baixa é constituído por um circuito RC-Série em que a tensão de saída é a do capacitor. Para ondas senoidais de frequências baixas, a reatância capacitiva assume valores altos em comparação com o valor da resistência, dessa maneira a tensão de saída será praticamente igual à tensão de entrada. Para frequências altas, a reatância capacitiva assume valores baixos em comparação com o valor da resistência, atenuando a tensão de saída para um valor praticamente nulo. Dessa maneira, o filtro permite a passagem de sinais de frequências baixas, sendo por isso denominado filtro passa-baixa”.

O filtro passa-baixa foi utilizado na montagem para filtrar o sinal PWM emitido pela porta CCP (porta 17), transformando o pulso modulado em uma tensão contínua da ordem de 5V, que posteriormente é amplificada para servir como referência à entrada analógica do inversor de frequência. Sem o filtro passa-baixa seria inviável o acionamento do motor com o pulso modulado. É importante citar que para se conseguir uma saída mais contínua do filtro passa-baixa se torna necessário calcular o valor da frequência de corte de acordo com a frequência de oscilação do *crystal* do PIC e posteriormente se estipular os valores da resistência e capacitância do filtro de acordo com a frequência de corte. Para tanto, o *crystal* colocado no circuito é de 20MHz, deve-se considerar o valor do *Prescaler* igual a 16, o valor do período igual a 255 e o valor do *postscale* igual a 1. Conhecidos os devidos valores é possível calcular o valor da frequência de corte da onda PWM de acordo com as equações 5.1 e 5.2 e assim dimensionar o valor do capacitor que deve ser inserido no circuito.

$$f_c = \frac{20000000}{(4 * 16 * 255)} = 1225Hz \quad (5.1)$$

$$f_c = \frac{1}{(2 * \pi * R * C)} \quad C = \frac{1}{(2 * \pi * 10 * 1225)} = 13\mu F \quad (5.2)$$

5.4.3 Circuito de Amplificação

De acordo com WENDLING, 2010,

”É muito difícil enumerar a totalidade das aplicações desse fantástico componente denominado Amplificador Operacional. De modo geral, podemos dizer que suas aplicações estão presentes nos sistemas eletrônicos de controle industrial, na instrumentação industrial, na instrumentação médica, nos equipamentos de telecomunicações, nos equipamentos de áudio, nos sistemas de aquisição de dados, etc. O Amplificador Operacional (AOP) é um amplificador multi estágio com entrada diferencial cujas características se aproximam das de um amplificador ideal. As características ideais de um AOP são impedância de entrada infinita, impedância de saída nula, ganho de tensão infinito, resposta de frequência infinita, insensibilidade à temperatura”.

Em virtude do sinal que sai da saída CCP do PIC ter amplitude máxima de 5V, é necessário realizar a amplificação do sinal para que este corresponda à faixa de valores de entrada analógica do inversor que varia de 0 a 10 Vcc. Portanto é necessário construir uma relação de amplificação com ganho 2 para o sinal de entrada do amplificador. Para isso é necessário a utilização de um amplificador operacional LM234 não inversor ou amplificador de ganho constante, para que o valor amplificado não passe a ser negativo. Quando o amplificador é configurado no formato da figura 5.6, obedece a relação de ganho dada pela equação 5.3.

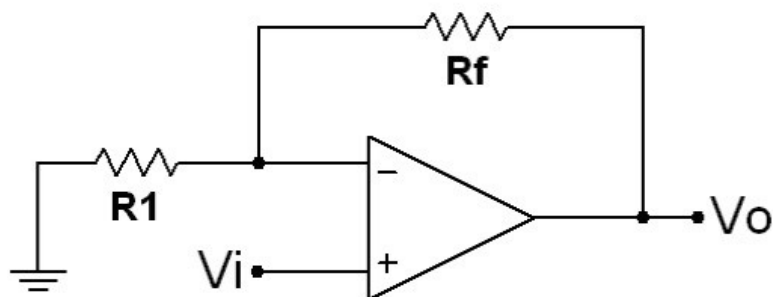


Figura 5.6 Configuração do Amplificador Não-Inversor

FONTE: WENDLING, 2010.

$$V_o = \left(1 + \frac{R_f}{R_1}\right)V_i \quad (5.3)$$

Para efeitos de se ajustar o ganho do amplificador foi colocado um resistor variável entre a saída e a entrada inversora do amplificador, de forma que se pudesse fazer um ajuste mais fino do ganho deixando assim o mais próximo de 2. A saída coletada no pino 1 do amplificador, é enviada como o sinal de referência analógica ao inversor. Para tanto o amplificador deve ser alimentado com as tensão 10Vcc e GND, disponíveis na saída do amplificador, possibilitando que o ganho seja dado dentro da faixa esperada. Na figura 5.7 é possível enxergar as conexões que devem ser realizadas no inversor, para alimentação do amplificador e recepção do sinal analógico. Os pinos 7 e 9 devem fornecer a alimentação do amplificador e o pino 8 recebe a referência.

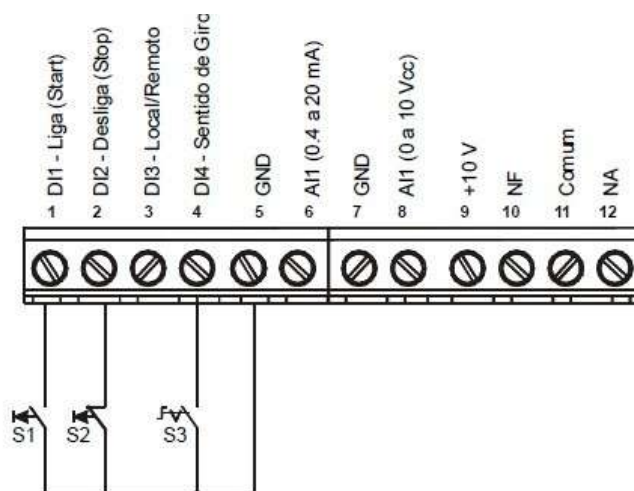


Figura 5.7 Pinos de Ligação do Inversor de Frequência

5.4.4 Comunicação USB

Segundo IBRAHIM, 2002,

”USB é atualmente uma especificação de interface de computador muito popular usado para conectar vários dispositivos periféricos com computadores e microcontroladores. Alguns microcontroladores PIC possuem módulos USB incorporadas. O USB foi projetado para simplificar a comunicação entre periféricos e o computador. Trata-se de uma comunicação assíncrona padronizada, que opera no modo *half-duplex*, com sinal diferencial e codificação NRZI”.

”O barramento é relativamente simples, sendo composto de um terra (GND), um Vcc (5v) e duas linhas de dados D- e D+. Ele é capaz de fornecer tensão de alimentação de 5V aos dispositivos de baixo consumo, suporta altas taxas de comunicação (na versão 2.0 High-Speed pode chegar a 480Mbps), não há necessidade de desligar o PC para conectar/desconectar o dispositivo USB, é *plug and play* e suporta até 127 dispositivos” (MIYADAIARA, 2009).

No circuito construído o USB é utilizado para a alimentação e para interfaceamento com o sistema supervisor. Para perfeito funcionamento do USB foi colocado um capacitor de desacoplamento em cada uma dos pinos de alimentação. Esses capacitores acabaram por se tornarem essenciais ao funcionamento do PIC, pois o funcionamento do inversor de frequência insere um número muito grande de harmônicos na rede, de forma que quando o inversor é ligado ao PIC desliga automaticamente. Os capacitores permitem filtrar o sinal de alimentação e garante que o PIC funciona normalmente.

É importante ressaltar a inserção de um capacitor de 470nF no pino 18 do PIC, conhecido como o Vusb, para que a conexão USB funcione normalmente.

Para comunicação com o sistema supervisor foi utilizado as portas 23 e 24 do PIC. Mas o maior trabalho se encontra na utilização da biblioteca USB do PIC, é necessário realizar se conhecer bem a ideia de transmissão de dados em bytes para que a comunicação se dê com exatidão. Mas o assunto será tratado com mais detalhes no capítulo sobre o desenvolvimento do sistema supervisor.

5.4.5 Placa de Circuito Impresso

Para concretização do projeto foi desenvolvida uma placa de circuito impresso que permitisse unir todos os circuitos eletrônicos citados acima em um só elemento. Essa placa foi desenvolvida no software Proteus, tendo como base o circuito desenvolvido anteriormente, demonstrado na Figura 5.5. O software disponibiliza o recurso de uma geração do desenho da placa a partir dos elementos inseridos no circuito esquemático (ISIS). Para tanto, os componentes foram posicionados na placa de forma a atender os requisitos do circuito, como por exemplo que o *crystal* ficasse próximo ao PIC, que as trilhas não se sobrepusessem, que o USB fosse instalado na borda da placa, dentre outros requisitos fundamentais na construção de circuitos impressos.

É importante ressaltar que alguns componentes colocados no circuito ISIS não possuem

correspondente no circuito de impressão (ARES), então devem ser desenhados a mão, seguindo os dimensionamentos disponíveis nos *datasheets* dos elementos correspondentes. Para este projeto foi necessário desenhar a representação do Trimpot e do botão utilizado para reset do pino ”*Master Clear* (MCLR)”. Abaixo estão listados todos os componentes utilizados para a confecção da placa:

- Resistores (2, 2K Ω , 33K Ω , 10K Ω)
- Capacitores (1nF, 22pF, 470pF, 100nF)
- Diodo Zenner 5,6V
- Amplificador Operacional LM 234
- Trimpot 3386 CW
- Cristal 20MHz
- Diodo 0,7V
- PIC 18F4550
- Botão
- Conectores diversos
- Conector de entrada USB

Uma vez que todos os elementos forem dispostos na placa de forma a atender os requisitos de construção, deve se verificar se não existem trilhas com quinas de 90°, pois estas podem captar interferência para o circuito segundo o princípio das pontas. Além disso deve-se acrescentar a malha de terra ao circuito, quanto maior a malha de terra melhor é para o circuito, pois permite a diminuição de interferências de sinais externos. Deve-se ressaltar que não deve existir grandes vãos livres sem estarem aterrados no circuito impresso pois estes vãos podem provocar uma diferença de potencial entre dois pontos da placa, atrapalhando o funcionamento. Outro fator bastante importante que deve ser citado também é que todos os terras do sistema devem estar conectados, o do *encoder*, do inversor de frequência e da própria placa devem estar ligados a um mesmo ponto.

Uma vez desenhada a placa de maneira correta ela tem o aspecto para impressão mostrado na figura 5.8.

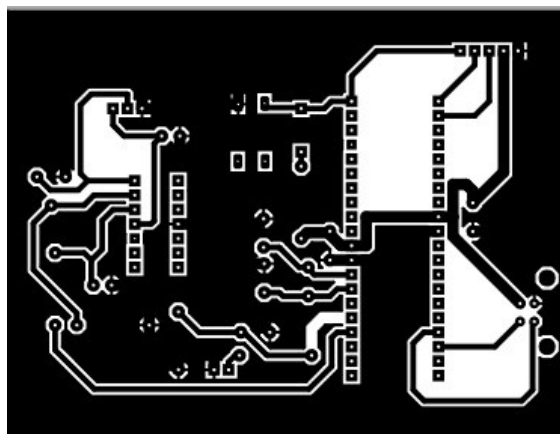


Figura 5.8 Construção da Placa de Circuito Impresso

6 LÓGICA DE CONTROLE E DESENVOLVIMENTO DE SUPERVISÓRIO

No capítulo anterior foram descritos em detalhes todos os procedimentos para a construção do sistema de controle, passando pela escolha do motor, parametrização do inversor e construção da placa de circuito impresso. Nesse capítulo será tratado sobre a etapa de programação do microcontrolador e desenvolvimento de software. Nessa etapa foram empregados os conhecimentos acerca de programação em C e C#, desenvolvendo uma lógica de controle para ser gravada no microcontrolador como também foi desenvolvida uma interface gráfica usada como sistema supervisório.

6.1 Programação do PIC

De acordo com SOUZA, 2009,

” Para que um programa seja escrito e funcione corretamente, basta que as instruções certas sejam colocadas na ordem correta. Esta expressão é totalmente verdadeira, mas se só tivermos isto no código do programa, apesar de ele ser funcional, não será eficiente. Isto porque ele não estará devidamente estruturado e padronizado, dificultando futuras alterações e/ou o entendimento por outros programadores”.

Basicamente foram necessários duas funções primordiais do PIC para se realizar o controle de velocidade requerido, a geração do sinal PWM do pino 17 e a contagem de pulsos do Timer1 a cada estouro do Timer0. A geração do sinal PWM permite o acionamento do motor de acordo com a variação da largura do pulso. Essa funcionalidade está ligada diretamente com o controle de velocidade pois a velocidade varia diretamente proporcional à variação do PWM. Já o timer 1 é utilizado para contar o número de pulsos do sinal PWM gerado pelo *encoder*, permitindo medir a velocidade de rotação do eixo de motor para se fechar a malha de controle. Abaixo serão expostos mais detalhadamente cada uma dessas funcionalidades com os respectivos códigos.

6.1.1 Sinal PWM

Já foi tratado anteriormente sobre ondas PWM e suas aplicações no projeto. Mas para que se possa fazer uso desse recurso do PIC devemos programar a porta 17 como

uma porta de saída e configurara-la para um sinal PWM. Desse forma é possível recorrer ao recurso da modulação de pulso no *void main()*, realizando assim o controle da velocidade do motor. Para tanto é necessário configurar o TRIS da porta como saída, colocando com valor 0 o terceiro bit do TRISC. Na sequência é necessário definir o *Timer2* que é definido para o PWM do PIC e por fim configurar a porta 17 como PWM. Tendo realizado as devidas configurações, passa a ser possível invocar a função PWM. No trecho de código que se segue é possível entender a configuração do PWM como descrita acima:

```
set tris c(0b00000001);
setup timer 2(t2 div by 16,255,1);
setup ccp1(CCP_PWM);
set pwm1 duty(127);
```

Dessa forma a onda coletada no pino 17 será uma onda quadrada com modulação de pulso de acordo com o estipulado na função *set pwm duty()*, podendo esse valor entre 0 e 255.

6.1.2 Contador de Giros

Para conhecer a rotação do motor devemos contar o número de bordas de subida que acontecem no sinal vindo do *encoder* em um determinado espaço de tempo. Para que isso seja possível devemos configurar o *Timer1* na porta 15 para contar o número de pulsos. Isso acontece definindo o timer como um temporizador externo e o *prescaler* como 1, assim como acontece no código mostrado abaixo:

```
setup timer 1(t1 external | t1 div by 1);
```

Após definir o *Timer1* devemos definir o intervalo de tempo para que esses estouros do *timer* (as bordas de subida do PWM) sejam contadas. Para isso definimos o *Timer0* em um segundo.

”O *Timer0* é o melhor auxiliar para a contagem do tempo dentro do PIC. Ele nada mais é que um contador automático de 8 bits para os ciclos de máquina ou pulsos externos. O *Timer0* é armazenado diretamente na RAM do sistema na posição 0x01 e é denominado internamente como TMR” (SOUZA, 2009).

Assim sendo é possível configurar o Timer 0 para contar 1s, de forma que seja possível contar os números de pulsos em 1s e multiplica-los por 60 para se conhecer as rotações por minuto do controlador. É importante lembrar de habilitar as interrupções do Timer0 para que esse necessariamente execute a cada 1s, adquirindo o valor do número de pulsos do Timer1, como se segue no código:

```
setup timer 0 (RTCC INTERNAL | RTCC 8 BIT | RTCC DIV 64);
  enable interrupts(GLOBAL);
  enable interrupts(int timer0); //interrupção do TIMER 0
  set timer0(131)
```

Para adquirir esse valor basta se utilizar a função `get timer1()`, e sucessivamente utilizar a função `set timer1` para zerar a contagem de pulsos. Dessa forma podemos ver no código a seguir a rotina que acontece toda vez que acontece um estouro do Timer0 e a interrupção é chamada. O valor do Timer1 é adquirido, multiplicado por 60 e resetado. Como já foi explicado o valor do Timer1 multiplicado por 60 expressa a velocidade da rotação em rpm.

```
#int timer0
  void trata tmr0 (){
  set timer0(131 + get timer0());
  cont++;
  if(cont > 125) {
  cont = 0;
  velocidade = get timer1() * 60;
  set timer1(0);
  }}
```

Assim fica expressa a ideia do funcionamento do controle pelo microcontrolador, a ideia é que uma vez encontrada a equação discretizada do controlador, a resposta enviada ao PWM seja fruto do tratamento do sinal vindo do contador de giros implementado. Mas a forma de se encontrar a equação do controlador será demonstrada no próximo capítulo.

6.1.3 Fuses

Um ponto importante de se tratar são sobre os *fuses* que devem ser definidos no escopo do programa. Esses *fuses* são responsáveis por determinar o *mode* de funcionamento do microcontrolador e uma vez mal definidos podem prejudicar o funcionamento da aplicação. Esses elementos alteram diretamente na configuração do *clock* do programa, multiplicando e dividindo a frequência proveniente do *crystal* para determinados pontos do circuito. Por vezes, quando mal definidos esses *fuses* podem ocasionar mal funcionamento em elementos como o PIC ou comunicação levando o desenvolvedor a se equivocar com relação a erro. Para a aplicação proposta em que se utiliza um *crystal* de 20HZ foram definidos os seguintes *fuses*:

```
#use delay (clock=20000000)
#fuses HS,NOWDT,NOPROTECT,NOLVP,NODEBUG,USBDIV
PLL5,CPUDIV1,VREGEN,NOWDT,NOPUT,NOBROWNOUT,NOCPD
```

Coloca-se em destaque os *fuses* USBDIV e VREGEN essenciais ao funcionamento do USB. Os *fuses* HS, CPUDIV1 que fazem com que se chegue 20MHz provenientes do *crystal* à CPU.

6.1.4 Gravação do PIC

”De nada adianta elaborarmos e implementarmos um programa se ele não for gravado na memória do componente para que funcione corretamente. Para tal, existem diversas maneiras e ferramentas capazes de efetuar essa gravação” (SOUZA, 2009). Mas seria impossível nos referir a todas elas, por isso nesta seção será tratada sobre a comunicação ICSP, para gravação do PIC. ICSP significa *In Circuit Serial Programming* e oferece uma maneira conveniente de programar PIC Micros sem remover o chip da placa de desenvolvimento ou produção. O ICSP fornece 6 conexões do programador pic ICSP para a sua placa como segue:

- VPP (ou MCLRn) Tensão de programação (geralmente 13V)
- Vcc Alimentação (geralmente 5V)

- GND Ground (0 volts).
- PGD - Data Pino de dados e conexão RB7
- PGC - Clock Pino de clock e conexão RB6
- PGM - LVP *enable* Pino de habilitação LVP e conexão RB3/RB4

”O Vpp se conecta à entrada de reset do microcontrolador PIC o MCLR. Durante a programação ou verificar se este sinal é gerado para a tensão de programação (13,5 V) - VCC + 3,5V. Isso sinaliza para o microcontrolador que a verificação/programação está prestes a começar e para os PICs mais antigos, fornece corrente. O Vcc pode fornecer energia para o seu PIC geralmente usando um regulador de 5V. O GND é a referência à terra (0V). Por fim o PGC e o PGD são os sinais de Dados (PGD) e de *clock* (PGC) responsáveis por transmitir os dados para o PIC. O primeiro dado é enviado, seja alta ou baixa tensão (0/1). Depois de um tempo adequado o *clock* é mudado de nível baixo para alto na borda de subida de *clock* os dados são enviados para o microcontrolador” (PICTRONICS, 2014).

6.2 Sistema Supervisório

Para criação do sistema supervisório foi utilizado a plataforma Visual Studio 2010 da Microsoft. Essa plataforma de desenvolvimento de software permite o desenvolvimento de softwares utilizando diversos tipos de linguagem, C, C++, F#, Visual Basic. Contudo a linguagem escolhida para desenvolvimento desse software foi C#, uma nova vertente de programação em C, voltada para orientação a eventos.

”O Visual Studio é um conjunto de ferramentas para construir aplicações para diversas plataformas, começando de aplicações desktop até aplicações corporativas e aplicações para a internet. Ele inclui três dos compiladores mais populares: Visual Basic, Visual C/C++ e Visual C#. Além dos compiladores, o Visual Studio conta com uma ampla biblioteca de ferramentas de programação. O Visual Studio também possui facilidades tais como editor de texto integrado com o compilador, ajuda interativa e ferramentas de depuração e identificação de erros no código” (MOURA & FERBER, 2009).

Assim sendo o Visual Studio se torna uma ferramenta de excelência no desenvolvi-

mento de Interfaces Computacionais Gráficas, permitindo dessa forma um melhor tratamento das informações, assim como uma melhor visualização. Isso se torna possível através dos mais diversos recursos que a plataforma de desenvolvimento nos proporciona, como botões, *comboBox*, *textBox*, Caixa de Diálogo, entre outras. Essa abordagem gráfica e a relação com banco de dados por meio de linguagem de programação de alto nível, torna o Visual Studio uma ferramenta mais robusta em relação aos pacotes offices, permitindo que o desenvolvedor entregue ao usuário produtos de maior qualidade e confiabilidade.

Sobretudo, como já foi citado, o Visual Studio permite a livre escolha da linguagem de programação. Assim sendo, foi necessário se conhecer um pouco de cada linguagem para que se pudesse tomar a decisão mais acertada de qual linguagem se utilizar. Ao se realizar uma revisão bibliográfica desta, foi possível perceber que a linguagem C#, seria a mais interessante a ser utilizada. O C# é uma linguagem de programação orientada a eventos, oriunda do C++ a qual é uma linguagem orientada a objetos.

”O C# (pronuncia-se ”C sharp”) é uma linguagem de programação criada para o desenvolvimento de uma variedade de aplicações que executam sobre o .NET Framework. C# é uma linguagem simples, poderosa, com tipagem segura e orientada a objetos. As várias inovações no C# permitem o desenvolvimento rápido de aplicações, mantendo a expressividade e a elegância do estilo de linguagens C ... Visual C# é uma implementação da linguagem C# pela Microsoft. Visual Studio oferece suporte a Visual C# com um editor de códigos completo, compilador, modelos de projeto, designers, assistentes de código, um depurador avançado e fácil de usar e outras ferramentas. A biblioteca de classes do .NET Framework fornece acesso a vários serviços do sistema operacional e outras classes úteis e bem estruturadas que aceleram significativamente o ciclo de desenvolvimento” (VISUAL C#, 2014).

Simplificadamente o C# executa um conjunto de códigos todas as vezes que eventos são gerados dentro do sistema. Ou seja, ao se clicar um botão, ou arrastar algum ícone, ou simplesmente alterar o valor de algum elemento é executada uma ação de acordo com as linhas de códigos referentes à aquele evento. Podemos citar ainda como vantagem o fácil esquema de conexão com banco de dados SQL que essa linguagem possui.

6.2.1 Interface Homem-Máquina (IHM)

O sistema supervisorio se baseia em uma tela gráfica dinâmica que facilite a vida do usuário em termos de ação e manutenção no sistema controlado. Para tanto é essencial que se crie uma tela gráfica que represente de forma consistente o sistema que se deseja supervisionar. Quanto mais elementos gráficos um sistema supervisorio possuir mais dinâmica e fácil se torna sua visualização.

O sistema em que se está trabalhando em um sistema bastante simples, cujo o objetivo é apenas controlar a velocidade de um motor, por esse motivo o supervisorio proposto possui poucos e simples elementos mas que representam bem o que se deseja supervisionar e facilitam substancialmente a visualização.

Para criação do supervisorio para controle de velocidade do motor foram utilizados os seguintes elementos relacionados:

- 1 TextBox: Para visualização do Set-Point de Velocidade.
- 1 TrackBar: Para seleção do Set-Point de Velocidade.
- 1 Gauge : Para visualização da velocidade no formato de velocímetro.
- 1XYGraph: Para visualização gráfica da variação da velocidade.
- 2 botoões: Para iniciar e parar a simulação no gráfico.
- Labels: Para disponibilização de informações diversas aos usuários.
- 1 MenuStrip: Para navegação em Menu.
- 1 Tool Strip: Para a verificação da conexão USB.

Os elementos foram dispostos no Form de forma que ficassem mais harmoniosamente posicionados, na intenção de facilitar a utilização dos usuários.

Para se realizar a função via Gauge basta se delegar o valor adquirido do byte de comunicação à variável `aGauge1.Value` como mostra o código a seguir:

```

v1 = Convert.ToInt32(txtSplit[1]) ■ 8;
v1 += Convert.ToInt32(txtSplit[2]);
aGauge1.Value = v1;
aGauge1.CapIdx = 1;
aGauge1.CapText = v1.ToString() + "rpm";

```

Plotar o gráfico de simulação não é tarefa mais difícil do adquirir os valores para o *Gauge*, basta criar um método *Timer1 Tick* para se coletar os valores mostrados no *Gauge* a cada instante de tempo e adicionar os valores ao objeto *XYGraph*. Na sequência ao se apertar o botão da simulação o usuário estará habilitando o método *Timer1* e o gráfico será gerado. O botão para parar a simulação desabilitará o *Timer1*. Essa funcionalidade é mostrada nos códigos a seguir:

```
private void button1 Click(object sender, EventArgs e)
    timer1.Enabled = true;
private void button2 Click(object sender, EventArgs e)
    timer1.Enabled = false;
private void timer1 Tick(object sender, EventArgs e)
    eixo x++;
    eixo y = aGauge1.Value;
    if (eixo x >= 120)
        timer1.Enabled = false;
    xyGraph1.AddValue(0, eixo x, eixo y);
    xyGraph1.DrawAll();
```

6.2.2 Comunicação USB-Serial

Para comunicação do supervisor com o sistema real foi utilizada uma comunicação do tipo USB HID onde os dados são enviados em pacotes de 8 bits e o próprio programador cria o protocolo de comunicação. Esse tipo de comunicação não necessita de instalação de qualquer tipo de driver. Ela se baseia na biblioteca *pic18 usb.h* do firmware e no acréscimo da biblioteca *UsbLibrary.dll* no Visual Studio. Contudo até o momento em que o esse texto foi finalizado a interfaceamento entre a aplicação e o software funcionou somente na simulação gerada no software Proteus. Quando aplicada na sistema real para se coletar os dados da planta física a comunicação falhou prejudicando a obtenção dos resultados. Esse assunto será abordado durante as discussões de resultados, mas devido a essa falha de comunicação será omitido todo código utilizado na tentativa de realizá-la, em virtude do erro não ter sido encontrado até então.

7 PROJETO DO CONTROLADOR

7.1 Método Empírico

De acordo com FRANKLIN, 2013,

”Para se encontrar os parâmetros do motor, o método mais viável e fácil é o método proposto por Ziegler-Nichols. J.G. Ziegler e N.B.Nichols, observaram que as respostas ao degrau de um grande número de sistemas de controle apresentam uma curva de reação do processo, como mostrado na figura 7.1, que pode ser gerada a partir de dados experimentais. A curva em forma de S é característica de muitos sistemas e pode ser aproximada pela resposta ao degrau de uma planta com função de transferência que representa um sistema de primeira ordem com retardo no tempo ou “atraso de transporte” de t_d . Essas constantes podem ser determinadas a partir da resposta do processo ao degrau unitário. Se uma reta tangente é traçada no ponto de inflexão da curva de reação do processo, então a inclinação da reta será $R = A/r$, a interseção da reta tangente com o eixo do tempo identifica o retardo no tempo $L = t_d$ e o valor final da curva fornece o valor de A ”.

Baseado no método proposto por Ziegler e Nichols, pretendia-se realizar a análise da curva de resposta do motor ao degrau, através do supervisor desenvolvido. Dessa forma seria possível encontrar o tempo de atraso e a constante de tempo do sistema nos revelando assim através da equação XX, a função de transferência do sistema bem como os parâmetros do controlador PID que podem ser calculados pela tabela 3.1 exposta no capítulo 3.

Esses parâmetros obtidos permitiriam uma sintonia mais exata do controlador, entretanto por estarem no domínio de Laplace não poderiam ser programados no controle, precisando pra isso que se fizesse uma discretização da controlador.

7.2 Controlador Digital

De acordo com ASTROM & HAGGLUND, 1995,

”Para fazer a implementação da lei de controle em tempo discreto, como por exemplo um sistema de controle PID utilizando um microcontrolador, utilizam-se aproximações numéricas da derivada e da integral. O objetivo é aproveitar as características particulares de cada

uma das ações de um controlador PID a fim de se obter uma melhora significativa do comportamento transitório e em regime permanente do sistema controlado ... A variável de controle é a soma de três termos: termo P, que é proporcional ao erro, o termo I, que é proporcional à integral do erro, e termo D, que é proporcional à derivada do erro. Os parâmetros do controlador são o ganho proporcional K_p , o tempo integral T_i , e o tempo derivativo T_d ... Os requisitos sobre um sistema de controle PID podem incluir muitos fatores, tais como respostas aos sinais de comando, insensibilidade a ruído e variações de processo, e rejeição de distúrbios de carga. O projeto do sistema de controle também envolve aspectos de processos dinâmicos, saturação do atuador, e características de distúrbios ... A ação derivativa não é frequentemente utilizada. É uma observação interessante que muitos controladores industriais tenham somente a ação PI e que em outros controladores a ação derivativa pode ser desligada. Pode-se mostrar que o controle PI é adequado para todos os processos onde as dinâmicas são essencialmente de primeira ordem. É fácil descobrir se esse é o caso, observando-se a resposta ao degrau do processo. Se a resposta ao degrau assemelha-se a de um sistema de primeira ordem, então o controle PI é suficiente. Outra razão é quando o processo é projetado para que sua operação não exija controle fino, mesmo se o processo tem dinâmica de alta ordem, necessita apenas de uma ação integral para eliminar o erro de regime, e de uma resposta transiente adequada, proporcionada pela ação proporcional.”

Baseado nos estudos feitos sobre controladores digitais é possível perceber que é a necessidade da discretização para a implementação do controle dentro do microcontrolador. Essa discretização é realizada pelo método de Tustin, onde a variável s do domínio de Laplace é correlacionada com a variável z discreta e o tempo de amostragem T , de acordo com a equação 7.1.

$$s = \frac{2z - 1}{Tz + 1} \quad (7.1)$$

Logo desenvolvendo as equações a partir da função de transferência obteremos a seguinte equação 7.2 para o controlador PID.

$$Gc(z) = \frac{(Kp + \frac{Kd}{T})z^2 + (TKi - Kp - \frac{2Kd}{T})z + \frac{Kd}{T}}{z(z - 1)} \quad (7.2)$$

Os parâmetros Kp, Ki, Kd da equação 7.2 são encontrados através da tabela de Ziegler-Nichols mostrada no capítulo 3. Essa equação discretizada é inserida no código do controlador para realização do controle. Existe também a possibilidade de se realizar os cálculos no sistema supervisor e apenas se enviar os valores de Kp, Ki e Kd para o microcontrolador, essa opção acaba por se tornar mais viável pois o computador possui mais recursos para realizar os cálculos e retirar os cálculos do PIC ocupa menos sua memória interna.

8 RESULTADOS ALCANÇADOS

O primeiro resultado do trabalho foi a criação da placa de circuito impresso. Depois de projetada no software Proteus, e testada no *proto-board*, a placa foi impressa, corroída e os componentes soldados. As figuras 8.1 e 8.2 mostram a configuração final da placa, com todos os requisitos atendidos, o *crystal* perto do PIC, os capacitores de desacoplamento, o borne de conexão USB na borda da placa, o resistor variável para se ajustar a relação de ganho do AOP, nenhuma trilha se sobrepondo e nem com quinas e o máximo de malha de terra possível.



Figura 8.1 Placa de Circuito Impresso

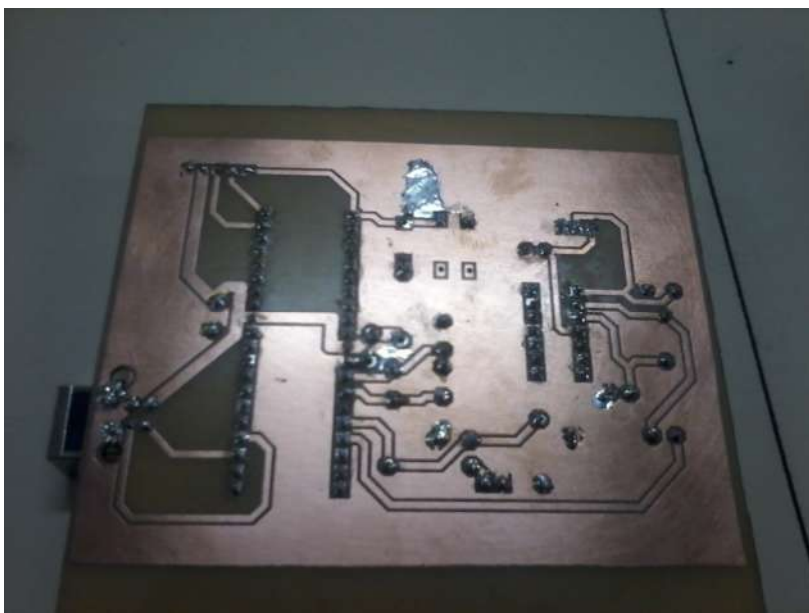


Figura 8.2 Trilhas da Placa de Circuito Impresso

Depois de testada com o multímetro para se averiguar a ausência de curto-circuito a programação do PIC foi gravada através do cabo ICSP. Posteriormente a placa foi ligada ao conjunto inversor-motor e realizados testes. Foram realizados teste de contagem de recepção de pulso do PWM com o osciloscópio, variação do valor do PWM para alteração do velocidade do motor e variação da resistência do trimpot para averiguar o perfeito funcionamento do amplificador. Tanto a gravação como os teste realizados foram bem sucedidos, concluindo com êxito o primeiro objetivo de construção da placa. O desenvolvimento da placa atendeu seu objetivo primordial do acionamento do motor.

O segundo resultado alcançado foi o desenvolvimento da interface gráfica com o Visual Studio, após estudos acerca da comunicação HID e desenvolvimento C#, foi construída uma interface amigável para funcionar como sistema supervisorio para se controlar virtualmente a velocidade do motor. A figura 8.3 mostra o supervisorio desenvolvido com o gráfico plotado durante a simulação. O sistema de comunicação funcionou perfeitamente para efeitos de teste, durante a comunicação com o Proteus todos os dados foram recebidos e enviados com perfeição, atestando o perfeito funcionamento durante a simulação. Contudo a etapa de comunicação com a placa não foi bem sucedida, isso porque não se obteve êxito em se adquirir os dados do sistema real para mostrá-los no Gauge e no Gráfico. Apesar de conseguir “setar” valores para a velocidade PWM do motor, o sentido contrário de comunicação falhou e nenhum dado foi recebido pelo sistema supervisorio. As causas dessa falha até o encerramento do texto não foram encontradas.

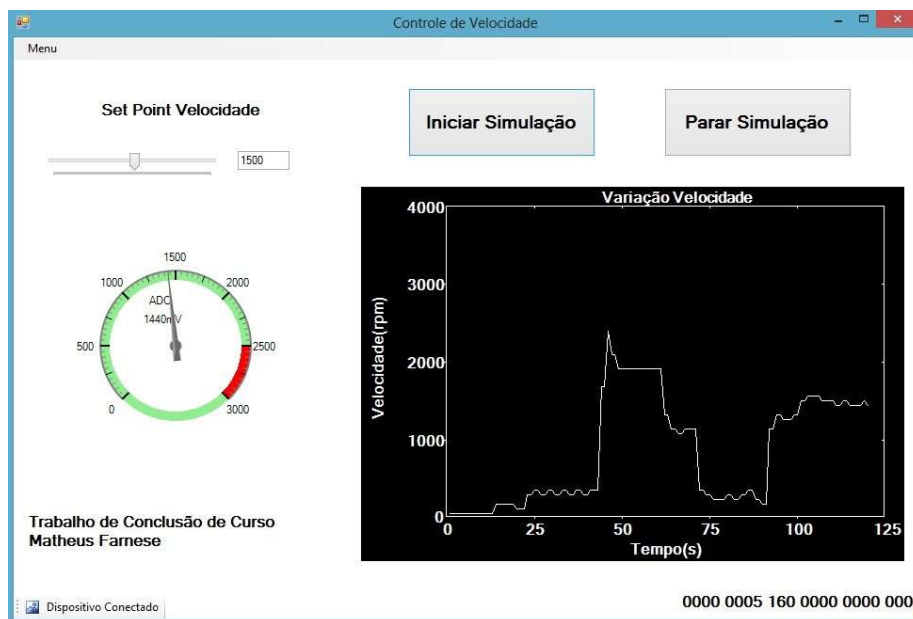


Figura 8.3 Tela do Supervisório Durante Simulação

A falha na comunicação USB prejudicou ainda na obtenção do resultado fundamental do trabalho, que era o projeto do controlador. Isso porque ao não se comunicar não foi possível visualizar a curva da resposta do degrau do sistema, logo não se pode fazer a análise da máxima derivada da curva para se encontrar o tempo de atraso e a constante de tempo do sistema. Conseqüentemente não foi possível se encontrar os parâmetros do PID para o motor estudado, por meio do método de Ziegler-Nichols.

Havia a possibilidade de se realizar o ensaio do motor para se obter os parâmetros do equipamento e encontrar a curva a função de transferência do sistema, mas isso requeria um estudo mais detalhado acerca de modelagem de motor. Devido à circunstâncias especiais não houve tempo hábil para tal.

8.1 Sugestão de Trabalhos Futuros

Sugere-se para o seguimento do projeto um estudo mais detalhado acerca da comunicação USB HID com o intuito de se resolver o problema de comunicação com a placa, de modo a tornar viável a aquisição de dados pelo sistema supervisor.

Sugere-se também que se realize o ensaio do motor estudado, para que uma vez que seja possível se construir o controlador pelo método empírico se possa comparar com o

controle realizado após o ensaio do equipamento.

Por fim fica como sugestão a inserção do *enconder* à placa desenvolvida, uma vez que o *enconder* acoplado no eixo do motor do módulo didático da hidro-eleto é alimentado com 220V e retorna uma tensão de pico do PWM muito alta que precisa ser ceifada, além do fato de estar ligado à mesma rede do inversor provocar muitos ruídos no sinal PWM que é gerado por esse *encoder*.

9 CONCLUSÃO

Ao fim do projeto é apresentado um estudo bastante avançado em controle de velocidade de motor utilizando microcontroladores que pode e deve ser levado em frente, desde que seja dada maior atenção nas falhas que esse projeto apresentou. Apesar dos resultados não terem sido os que foram propostos no início do projeto, acredita-se que houve um resultado satisfatório dentro do que se foi proposto, apesar de estar longe do ideal. Mas esse projeto tem margens pra continuidade para trabalhos futuros.

Como plataforma didática para práticas julga-se que o projeto atendeu seu objetivo, pois permite ao graduando visualizar muito acerca da teoria vista sobre eletrotécnica, eletrônica, programação e sistemas de controle a partir do momento que seja implementado o controlador. Placas de circuito impresso como as confeccionadas tem um grande poder para chamar a atenção de alunos para práticas laboratoriais e deveriam ser tratadas com mais frequência durante o curso.

Uma vez que esse projeto seja aprimorado, passará a ter grande valor didático e com certeza contribuirá muito para aulas práticas do curso de Engenharia de Controle e Automação da Escola de Minas. Por esse motivo apesar de não ter se obtido êxito na construção do controlador, julga-se ao fim do projeto que o objetivo geral foi atingido.

Como relato pessoal, julgo que foi bastante enriquecedor o desenvolvimento do projeto por ter lidado com diversos ramos da automação, me permitindo rever muitos conceitos aprendidos ao longo do curso e aprender diversas coisas novas que ainda não haviam sido estudadas ou aprendidas, como por exemplo a construção de placas de circuito impresso. Apesar de bastante trabalhoso o resultado final valeu a pena todo o trabalho durante o desenvolvimento.

REFERÊNCIAS BIBLIOGRÁFICAS

- ALFA5ELETRICISTA. **Inversor de Frequência, Descrição do Funcionamento.** <<http://www.alfa5eletricista.com.br/index.php/inversor-de-frequencia.html?showall=&limitstart=>>. Acesso em 25/07/2014, 17:13h.
- ASTROM, K.J. ; HAGGLUND, T. **PID Controllers: Theory, Design, and Tuning.** 2ª Edição, Carolina do Norte: Instrument Society of America, 1995.
- BEGA, E.A. **Instrumentação Industrial.** 2ª ed. Editora Interciência, 2000.
- BOYLESTAD, R.; NASHELSKY, L. **Dispositivos Eletrônicos e Teoria de Circuitos.** Editora LTC, 6ª ed., 1996.
- DENARDIN, G. W. **Microcontroladores.** Apostila. Universidade do Estado de Santa Catarina, < <http://pt.scribd.com/doc/226223967/Apostila-Micro-Do-Gustavo-Weber#scribd>>. Acesso em 15/06/2014.
- FRANKLIN, G. F. ; POWELL, J.D. ; EMAMI-NAEINI, A. **Sistemas de Controle para Engenharia.** Editora Bookman, 6ª Edição, 2013.
- IBRAHIM, D. **Microcontroller Based Temperature Monitoring and Control.** 1ª ed. Editora Newnes, 2002.
- KOERICH, A. L. **Filtro Passa-Baixa e Filtro Passa Alta.** Pontifícia Universidade Católica do Paraná. Disponível em <<http://www.ppgia.pucpr.br/~alekoe/CIR/2012-2/Exp07-FiltroPassaAltaPassaBaixa-2012-2.pdf>> Acesso em 03/07/2014.
- LEONARD, W. **Control of Electrical Drives.** Ed. Spring-Verlag. 1985.
- MIYADAIARA, A. N. **Microcontroladores PIC18: Aprenda e Programe em Linguagem.** São Paulo: Ed. Érica, 2009.
- MOTORELETRICO. **Video-aula CFW10.** 2012. Disponível em: <<http://www.motoreletrico.net/index.asp?InCdSecao=3&InCdEditoria=2&InCdMateria=252&V%EDdeo-aula+CFW10>>. Acesso em 25/07/2014.
- MOURA, A.V. ; FERBER, D.F. **O Primeiro Programa em Visual Studio.net.** Unicamp, 2009. Disponível em: <http://www.ic.unicamp.br/~mc102/aulas/p_arnaldo/texto/cap01_introducao_visual_studio.pdf>. Acesso em 27/07/2014.

OGATA, K. **Engenharia de Controle Moderno**. 4^a ed. Editora Prentice Hall (Pearson). 2003.

PALHARES, R. M. **Estabilidade de Sistemas Lineares Realimentados**. 1998. Material de aula. Universidade Federal de Minas Gerais, <http://www.cpdee.ufmg.br/~palhares/aula6_csl.pdf>. Acesso em 19/06/2014.

PEREIRA, F. **Microcontroladores PIC: Técnicas Avançadas**. São Paulo: Ed. Érica, 2005.

PEREIRA, F. **Microcontroladores PIC: Técnicas Avançadas**. São Paulo: Ed. Érica, 7^a Edição, 2007.

PICTRONICS. Detalhes do PIC-ISCP e como usá-lo para os microcontroladores PIC. Disponível em: <http://www.pictronics.com.br/component/content/article/38/69-detalhes-do-icsp-para-os-pics-e-como-usa-lo.html>. Acesso em 28/07/2014, 00:48h.

SOUZA, D. J. **Desbravando o PIC: ampliado e atualizado para PIC 16F628A**. São Paulo: Ed. Érica, 2009.

VISUAL C#. **Visual C#**. Disponível em: <https://msdn.microsoft.com/pt-br/library/kx37x362.aspx>. Acesso em 27/07/2014.

WENDLING, M. **Amplificadores Operacionais**. Versão 2.0, UNESP, 2010.

ZANCO, W.S. **Microcontroladores PIC 16F628A/648A: uma abordagem prática e objetiva**. São Paulo: Ed. Érica, 2005.

Érica, 2009

MORENO, J.; CIPOLLA, M ; PERACULA, J ; BRANCO, P. J. C. **Fuzzy logic based improvements in efficiency optimization of induction motor drives**. IEEE , Fuzzy Conference, 0-7803-3796-4/97 , p. 219-224. 1997.

VAS, P. **Vector Control of AC Machines**. Universidade de Michigan, Clarendon Press, 1990. 332p.