

RODRIGO OLIVEIRA BROSLE

CONTROLADOR PID UTILIZANDO MICRONCONTROLADOR PIC

RODRIGO OLIVEIRA BROSLE

CONTROLADOR PID UTILIZANDO MICROCONTROLADOR PIC

Trabalho de Graduação apresentado ao Conselho de Curso de Graduação em Engenharia Elétrica da Faculdade de Engenharia do Campus de Guaratinguetá, Universidade Estadual Paulista como parte dos requisitos para obtenção do diploma de Graduação em Engenharia Elétrica.

Orientador: Prof. Dr. Márcio Abud Marcelino

Guaratinguetá
2014

B874c	<p data-bbox="347 1451 1321 1637">Brosler, Rodrigo Oliveira Controlador PID utilizando microcontrolador PIC/ Rodrigo Oliveira Brosler – Guaratinguetá, 2014. 89 f : il. Bibliografia: f. 75-77</p> <p data-bbox="347 1671 1321 1778">Trabalho de Graduação em Engenharia Elétrica – Universidade Estadual Paulista, Faculdade de Engenharia de Guaratinguetá, 2014. Orientador: Prof. Dr. Márcio Abud Marcelino</p> <p data-bbox="405 1816 1321 1888">1. Microcontroladores 2. Motores elétricos de corrente contínua I. Título</p> <p data-bbox="1136 1924 1321 1957">CDU 621.381</p>
-------	--

CONTROLADOR PID UTILIZANDO MICROCONTROLADOR PIC


RODRIGO OLIVEIRA BROSLE

ESTE TRABALHO DE GRADUAÇÃO FOI JULGADO ADEQUADO COMO
PARTE DO REQUISITO PARA A OBTENÇÃO DO DIPLOMA DE
"GRADUADO EM ENGENHARIA ELÉTRICA"

APROVADO EM SUA FORMA FINAL PELO CONSELHO DE CURSO DE
GRADUAÇÃO EM ENGENHARIA ELÉTRICA


Prof. Dr. LEONARDO MESQUITA
Coordenador

BANCA EXAMINADORA:


Prof. Dr. MÁRCIO ABUD MARCELINO
Orientador/UNESP-FEG


Prof. Dr. FRANCISCO ANTONIO LOTUFO
UNESP-FEG


Prof. Dr. JOSÉ FELICIANO ADAMI
UNESP-FEG

Abril de 2014

AGRADECIMENTOS

Aos meus pais Rosângela e José, os quais depositaram em mim a confiança necessária e proveram sempre apoio financeiro e moral.

Ao meu orientador, Prof. Dr. Márcio Abud Marcelino, que demonstrou interesse recíproco em me orientar desde o início, trazendo para mim muita confiança e motivação. Desde a escolha do tema até a conclusão, seu interesse e dedicação em orientar foram fundamentais para a conclusão deste trabalho.

A todos os amigos e colegas de turma, que percorreram junto comigo o caminho até aqui. A união, cumplicidade mútua e disposição em ajudar foram essenciais a cada etapa vencida.

A todos os funcionários da Faculdade de Engenharia de Guaratinguetá, os quais proveram um ambiente propício para a minha formação e aprendizado.

Aos professores, especialmente aos mais presentes em minha jornada, os do Departamento de Engenharia Elétrica. Seu empenho em transmitir conhecimento foi e será o principal fator de sucesso dos que já passaram e dos que virão.

BROSLER, R. O. **Controlador PID utilizando microcontrolador PIC**. 2014. 89 f. Trabalho de Graduação (Graduação em Engenharia Elétrica) - Faculdade de Engenharia do Campus de Guaratinguetá, Universidade Estadual Paulista, Guaratinguetá, 2014.

RESUMO

Este trabalho tem como proposta a construção de um controlador PID de baixo custo com microprocessador PIC, e a aplicação em um sistema de posicionamento antes controlado por um circuito integrado dedicado para o controle de posicionamento. Ao aplicar o controle PID em malha fechada as instabilidades no sistema foram minimizadas, trazendo uma resposta mais suave, eliminando vibrações e desgaste mecânico comparado ao sistema com circuito dedicado, cuja ação de controle é muito limitada. O atuador na saída é um motor DC, cuja velocidade foi controlada pela técnica de Modulação por Largura de Pulso (PWM), utilizando uma ponte H para permitir duplo sentido de rotação. O microcontrolador utilizado foi o PIC16F684, que já possui um módulo PWM interno, com entradas analógicas que foram utilizadas como referência e realimentação de posições. O sensor de posição utilizado é um potenciômetro multivoltas acoplado ao eixo do motor por um conjunto de engrenagens. A possibilidade de programação dos coeficientes do PID no microcontrolador, bem como o ajuste da taxa de amostragem, permitem o sistema desenvolvido atingir elevado grau de versatilidade.

PALAVRAS-CHAVE: Controle PID. Microcontrolador. Controle de Velocidade. Motor DC. PWM. Posicionamento.

BROSLER, R. O. **PID controller using PIC microcontroller**. 2014. 89 f. Monograph (Bachelor in Electrical Engineering) - Faculdade de Engenharia do Campus de Guaratinguetá, Universidade Estadual Paulista, Guaratinguetá, 2014.

ABSTRACT

This monograph proposes the implementation of a low cost PID controller utilizing a PIC microcontroller, and its application in a positioning system previously controlled by a dedicated integrated circuit for a positioning system. Applying the closed-loop PID control, the system instability was reduced, and its response was smoother, eliminating vibrations and mechanical wear compared to its response with the dedicated integrated circuit, which has a very limited control action. The actuator of the system is a DC motor, whose speed is controlled by the Pulse Width Modulation (PWM) technique, using a Full-Bridge circuit, allowing the shift of direction of rotation. The utilized microcontroller was the PIC16F684, which has an enhanced PWM module, with its analog converters used as reference and position feedback. The positioning sensor is a multiturn potentiometer coupled to the motor axis by gears. The possibility of programming the PID coefficients in the microcontroller, as well as the adjustment of the sampling rate, allows the implemented system achieving high level of versatility.

KEYWORDS: PID Control. Microcontroller. Speed Control. DC motor. PWM. Positioning.

LISTRA DE ILUSTRAÇÕES

Figura 2.1 - Sistema turbina hidráulica alimentando gerador elétrico	17
Figura 2.2 - Diagrama de blocos de um sistema de controle a malha aberta	18
Figura 2.3 - Diagrama de blocos de um sistema de controle a malha fechada.....	19
Figura 2.4 - Diagrama de blocos de um controlador liga-desliga	21
Figura 2.5 - Diagrama de um controlador on-off com intervalo diferencial	22
Figura 2.6 - Diagrama de blocos de um controlador proporcional.....	23
Figura 2.7 - Diagrama de blocos de um controlador do tipo proporcional e integral	24
Figura 2.8 - Diagrama de blocos de um controlador do tipo proporcional e derivativo	25
Figura 2.9 - Diagrama de blocos de um controlador PID.....	26
Figura 2.10 - Especificações da resposta transitória ao degrau unitário	27
Figura 2.11 - Amostragem e Quantização de uma função contínua no tempo.....	30
Figura 2.12 - Aliasing devido à baixa frequência de amostragem	31
Figura 2.13 - Erro de quantização e saturação de um conversor AD	32
Figura 2.14 - Erros de <i>offset</i> , linearidade e ganho de um conversor AD.....	33
Figura 2.15 - Recuperação de sinal amostrado com (a)retenção de ordem-zero (b)primeira ordem (c)primeira ordem com interpolação	34
Figura 2.16 - Diagrama de blocos do sistema controlado pelo PIC	34
Figura 2.17 - Cálculo da variação de uma função dentro de um intervalo.....	36
Figura 2.18 - Erros na aproximação da derivada pelo método das diferenças finitas	37
Figura 2.19 - Cálculo da área interna de uma figura trapezoidal	37
Figura 2.20 - Aproximação da integral pelo método da integração trapezoidal.....	38
Figura 2.21 - Diagrama de blocos controlador PID a ser implementado no PIC.....	39
Figura 3.1 - Características construtivas de um motor DC de 2 polos	41
Figura 3.2 - Componentes internos de um motor DC de 2 polos	42
Figura 3.3 - Condutor elétrico retilíneo com corrente sob ação de um campo magnético	44
Figura 3.4 - Princípio de funcionamento do motor de corrente contínua de 2 polos	45
Figura 3.5 - Comutação em um motor de corrente contínua de 2 polos.....	45
Figura 3.6 - Circuito equivalente de um motor de corrente contínua.....	46
Figura 4.1 - Sinal de onda quadrada e seus parâmetros.....	48
Figura 4.2 - Variação do ciclo de trabalho (a) duty cycle = 50% (b) 30% (c) 80%	49
Figura 4.3 - Ajuste de velocidade de um motor de corrente contínua pela técnica PWM	49

Figura 4.4 – Causa da dissipação de potência na comutação de tiristores	51
Figura 5.1 - <i>Driver</i> de Ponte Completa (Ponte H).....	52
Figura 6.1 – Diagrama de blocos para seleção da fonte de <i>clock</i>	57
Figura 6.2 – Precisão do oscilador interno de alta frequência.....	57
Figura 6.3 – Função de transferência do conversor AD	58
Figura 6.4 – Diagrama de blocos da seleção canal de entrada conversor AD.....	58
Figura 6.5 – Diagramas de blocos do módulo ECCP configurado como PWM.....	59
Figura 6.6 – Conexão das saídas do PWM ao <i>Driver</i> de Ponte Completa	60
Figura 6.7 – Formas de onda de saída do módulo PWM	61
Figura 7.1 – Fluxograma geral do programa	63
Figura 7.2 – Implementação em linguagem C da regra do trapézio (integral).....	65
Figura 7.3 – Resposta do sistema para $K_p = 0,5 / K_d = 0 / K_i = 0$	67
Figura 7.4 - Resposta do sistema para $K_p = 5 / K_d = 0 / K_i = 0$	68
Figura 7.5 - Resposta do sistema para $K_p = 1,5 / K_d = 0 / K_i = 0$	68
Figura 7.6 - Resposta do sistema para $K_p = 1,5 / K_d = 0 / K_i = 0,5$	69
Figura 7.7 - Resposta do sistema para $K_p = 1,5 / K_d = 0,2 / K_i = 0,5$	69
Figura 7.8 - Resposta do sistema para $K_p = 15 / K_d = 0,2 / K_i = 0,5$	71
Figura 7.9 - Resposta do sistema para $K_p = 10 / K_d = 1, 2 / K_i = 0,5$	71
Figura 7.10 - Resposta do sistema ao degrau para $K_p = 10 / K_d = 0,04 / K_i = 0,5$	72
Figura 7.11 - Resposta do sistema a entrada aleatória para $K_p = 10 / K_d = 0,04 / K_i = 0,5$	73

LISTRA DE ABREVIATURAS E SIGLAS

ABS	- Anti-block Braking System
AD	- Analog to Digital
CI	- Circuito Integrado
DA	- Digital to Analog
DC	- Direct Current
ECCP	- Enhanced Capture/Compare/PWM
I/O	- Input/Output
MOSFET	- Metal Oxide Semiconductor Field Effect Transistor
MV	- Variável Manipulada
PDIP	- Plastic Dual In-Line Package
PD	- Proporcional e Derivativo
PI	- Proporcional e Integral
PID	- Proporcional, Integral e Derivativo
PV	- Variável de Processo
PWM	- Pulse Width Modulation
RAM	- Random Access Memory
ROM	- Read Only Memory

LISTRA DE SÍMBOLOS

K_p	Ganho Proporcional
K_d	Ganho Derivativo
K_i	Ganho Integral
T_i	Tempo Integral / Taxa de Reestabelecimento
T_d	Tempo Derivativo
e	Erro
Φ	Fluxo magnético
B	Campo magnético
E	Campo Elétrico
μ_0	Permeabilidade magnético no vácuo
i	Corrente elétrica
F	Força

SUMÁRIO

1	INTRODUÇÃO	12
1.1	OBJETIVO.....	12
1.2	MOTIVAÇÃO	13
1.3	ORGANIZAÇÃO.....	14
2	CONTROLE PID DISCRETO.....	16
2.1	CONTROLE AUTOMÁTICO.....	16
2.2	CONTROLE A MALHA ABERTA VERSUS MALHA FECHADA	18
2.3	TIPOS DE CONTROLADORES	20
2.3.1	Controladores de duas posições ou liga-desliga (on-off)	20
2.3.2	Controladores proporcionais	22
2.3.3	Controladores do tipo proporcional e integral.....	23
2.3.4	Controladores do tipo proporcional e derivativo.....	24
2.3.5	Controladores do tipo proporcional, integral e derivativo (PID)	26
2.4	CONTROLE DIGITAL	28
2.4.1	Amostragem e quantização	29
2.4.2	Conversão digital para analógica	33
2.5	APROXIMAÇÃO DA INTEGRAL E DERIVADA PARA EQUAÇÕES ALGÉBRICAS	35
2.5.1	Aproximação da derivada pelo método das diferenças finitas (Euler).....	35
2.5.2	Aproximação da integral pela integração trapezoidal	37
2.5.3	Equação algébrica aproximada do controlador PID.....	39
3	MOTOR DE CORRENTE CONTÍNUA	41
3.1	CARACTERÍSTICAS CONSTRUTIVAS DO MOTOR DC	41
3.2	PRINCÍPIO DE FUNCIONAMENTO DO MOTOR DC	43
3.2.1	Circuito equivalente do motor DC	46
4	MODULAÇÃO POR LARGURA DE PULSO (PWM)	48

5	DRIVER DE PONTE COMPLETA (PONTE H).....	52
6	MICROCONTROLADOR PIC	55
7	APLICAÇÃO E RESULTADOS PRÁTICOS	62
7.1	DESENVOLVIMENTO DO PROGRAMA.....	62
7.2	VERIFICAÇÃO DE FUNCIONAMENTO E AJUSTE DOS GANHOS	66
7.2.1	Variação da resposta ao degrau com ganho proporcional.....	67
7.2.2	Variação da resposta ao degrau com ganho integral	68
7.2.3	Variação da resposta ao degrau com ganho derivativo	69
7.2.4	Ajustes dos ganhos do controlador PID (sintonia).....	70
8	CONCLUSÕES.....	74
	REFERÊNCIAS	75
	BIBLIOGRAFIA CONSULTADA	76
	APÊNDICE A – PROGRAMA COMENTADO	78
	APÊNDICE B – DIAGRAMA ESQUEMÁTICO	82
	ANEXO A – CARACTERÍSTICAS OPERACIONAIS IRFZ34N	83
	ANEXO B – CARACTERÍSTICAS OPERACIONAIS IRFZ34NPbF.....	84
	ANEXO C – PRINCIPAIS CARACTERÍSTICAS DO PIC16F684	85
	ANEXO D – AMBIENTES DE PROGRAMAÇÃO, GRAVAÇÃO E TESTE...	88

1 INTRODUÇÃO

1.1 OBJETIVO

Este trabalho tem como objetivo a construção de um Controlador PID de baixo custo utilizando microprocessador PIC. O controle PID é amplamente aplicado em processos industriais que exigem controle de variáveis como temperatura, velocidade, vazão, entre outras grandezas analógicas. O controle PID não exige um conhecimento preciso do modelo matemático do sistema a ser controlado, apesar deste conhecimento ser extremamente auxiliador no momento da sintonia do controlador. Sintonia esta que permite, através da variação dos ganhos proporcional, integral e derivativo do controlador, um ajuste relativamente simples caso ocorram alterações físicas no sistema, que irão implicar em alterações no modelo matemático.

São utilizadas aproximações matemáticas para que as variáveis analógicas do processo e suas derivadas e integrais sejam representadas satisfatoriamente no domínio do tempo discreto ou digital, transformando as equações da integral e da derivada em equações algébricas lineares. Tais aproximações têm como objetivo a redução do processamento realizado por parte do microcontrolador.

Os microcontroladores têm limitações de frequência de trabalho e memória interna (ROM e RAM) disponível. Com um processamento mais enxuto, é possível que um sistema opere com menor frequência e utilize menos memória. O que pode reduzir custos no sentido de se utilizar um microprocessador mais barato, tendo este menor frequência de trabalho e memória interna, com menor consumo de corrente, fator este que reduz custo da fonte de alimentação. Além disso, alguns microprocessadores da família PIC, que foi a utilizada neste trabalho, possuem oscilador interno, que opera numa frequência mais baixa que a máxima, e tem precisão e velocidade satisfatórias para a aplicação em questão. Com o uso deste oscilador interno, também são reduzidos custos de componentes externos ao microcontrolador, como cristais ressonantes, capacitores e espaço de placa.

A aplicação deste controlador foi para um sistema de posicionamento angular em malha fechada, que é composto por uma entrada e uma saída. A entrada analógica é a referência de

posição, neste caso constituída por um potenciômetro. A realimentação é a posição da saída, constituída também por um potenciômetro e acoplada a um motor através de engrenagens.

Optou-se pelo motor de corrente contínua, pois o mesmo apresenta controle de velocidade linear em uma grande faixa com alteração no sentido de rotação relativamente simples. O motor tem o papel de buscar a posição referenciada na entrada. Para isto o mesmo deve ser capaz de inverter seu sentido de rotação e ter sua velocidade controlada pelo controlador PID. Foi utilizado um circuito *Driver* de Ponte Completa, acionado pela técnica de Modulação por Largura de Pulso (PWM).

1.2 MOTIVAÇÃO

Existem circuitos integrados dedicados a realizar o controle de sistemas de posicionamento com as características equivalentes as utilizadas neste trabalho. Como exemplo, a fabricante ON Semiconductors, subdivisão da Motorola Co., possui o circuito integrado (CI) MC33030. Este possui um amplificador operacional utilizado como amplificador de erro, *drive* e freio lógico com memória de direção, *driver* de chave H com capacidade de 1A. Pode ser utilizado em diversas aplicações que necessitam de sensoriamento de posição de um servo motor ou motor DC, em que a posição de saída pode ser convertida em forma de tensão.

Apesar de ter grandes vantagens como, detector de sobrecorrente programável, tempo de atraso de desligamento por sobrecorrente programável, desligamento por sobretensão e ponte H interna com capacidade de até 1A, o MC33030 não possui sistema de partida ou frenagem suave. Este possui uma banda morta de operação (*Dead-Band*) com objetivo de eliminar instabilidade, porém o motor é sempre acionado com tensão plena. Fator que pode ocasionar altas correntes demandadas da fonte de alimentação e do próprio CI durante partidas e frenagens. Além disso, nas partidas e frenagens são realizados movimentos bruscos, os quais dependendo das características do sistema mecânico, inércia da massa e motor, podem provocar instabilidade e tornar o controle ineficiente.

O controle PID proposto neste trabalho tem como objetivo eliminar estes movimentos bruscos que demandam alta corrente no motor e podem provocar instabilidade. Com isto é

possível que um mesmo motor e carga mecânica apresentem resposta mais suave, tornando possível também o uso de uma fonte de alimentação mais enxuta, bem como reduzir a carga aplicada aos dispositivos de potência, reduzindo o custo de implementação e o consumo de energia pelo sistema.

Outras vantagens pretendidas com o controlador proposto neste trabalho são a de flexibilidade na alteração dos tiristores de potência, versatilidade quanto ao ajuste dos ganhos do controlador e da frequência de amostragem e menor número de pinos do CI. Caso seja necessário aumentar a potência fornecida ao motor e no caso dos MOSFETS (*Metal Oxide Semiconductor Field Effect Transistor*) presentes não possuem a capacidade necessária, basta realizar a troca dos dispositivos de potência por outros de maior capacidade. Com o ajuste dos ganhos, frequência de amostragem e a flexibilidade na troca dos dispositivos de potência o circuito se torna muito versátil, abrangendo desde pequenas aplicações como no caso deste trabalho, até grandes estruturas mecânicas. O PIC utilizado possui dois pinos a menos que o CI MC33030, sendo que seis deles não são utilizados neste trabalho, o que permite um aumento das funcionalidades e facilidades posteriormente (expansão do programa).

1.3 ORGANIZAÇÃO

Além do capítulo 1, o trabalho foi descrito da seguinte forma:

O capítulo 2, “Controle PID Discreto”, descreve o controle PID e a forma com o qual este foi implementado no microcontrolador PIC.

O capítulo 3, “Motor de corrente contínua”, apresenta as características construtivas e operacionais do motor DC, bem como os possíveis métodos utilizados para controlá-lo.

No capítulo 4, “Modulação por Largura de Pulso (PWM)”, são apresentadas as principais características deste tipo de modulação aplicada ao controle de velocidade de motores de corrente de contínua, ressaltando fatores e limitações que devem ser observados com relação ao motor e circuito de acionamento.

O capítulo 5, “*Driver* de Ponte Completa”, descreve o funcionamento do circuito conhecido como ponte H e apresenta a construção, a definição de componentes adequados aos fatores de trabalho como frequência, corrente e tensão.

No capítulo 6, “Microcontrolador PIC”, conforme as necessidades do sistema, foi definido o microcontrolador PIC a ser utilizado na implementação do controlador PID. Sua definição foi então justificada e foi apresentado o microcontrolador e de suas características.

O capítulo 7, “Aplicação e Resultados Práticos”, descreve a integração dos componentes descritos nos capítulos anteriores e mostra os resultados obtidos com o controle PID implementado no microprocessador PIC aplicado ao sistema de posicionamento.

No capítulo 8, “Conclusões”, são evidenciadas as conclusões quanto à aplicação deste tipo de controle.

2 CONTROLE PID DISCRETO

2.1 CONTROLE AUTOMÁTICO

No campo da engenharia e ciência há tempo que o Controle Automático tem exercido um papel muito importante, e com o passar do tempo esta importância tem se tornado cada vez maior. No mundo atual procura-se realizar ações de modo mais satisfatório e econômico possível, e o controle mais preditivo e preciso dos processos industriais acarreta em melhoria da qualidade de produção, de eficiência energética e do consumo de recursos naturais.

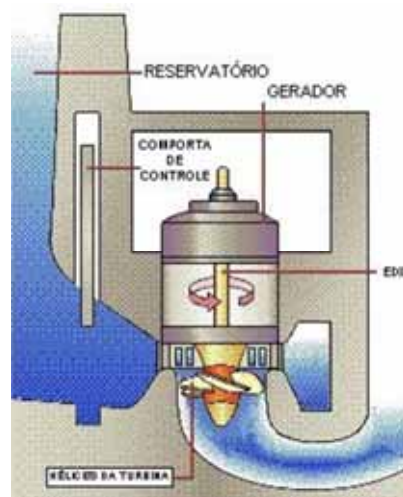
Com a tecnologia cada vez mais presente na vida cotidiana, não se pode restringir a importância da melhoria do controle automático aos processos industriais. O consumo de energia e tempo, bem como a segurança, são fatores fundamentais em qualquer área. O controle automático é responsável pela existência, por exemplo, de máquinas de lavar, que economizam esforço e tempo do cidadão, bem como água do meio ambiente. A constante melhoria do controle automático aplicado em automóveis possibilita motores à combustão consumirem menos combustível fóssil esgotável, e também proporciona aumento da segurança dos passageiros e pedestres com a introdução de sistemas de controle de tração, estabilidade, antitravamento de freios (*Anti-lock Braking System ou ABS*) e bolsas de ar (*Airbag*). Exemplos de aplicações onde o controle automático tem extrema importância são: aviação, membros artificiais robóticos, equipamento bélico, eletrodomésticos e eletrônicos como: impressora, ar-condicionado, etc.

Seja no controle de velocidade de motores, sistemas robóticos, fornos industriais, turbinas, reatores entre outros equipamentos presentes nos processos industriais, sempre o controlador automático está atuando em variáveis físico-químicas como: pressão, temperatura, umidade, viscosidade, vazão, velocidade, deslocamento, etc.

Em sistemas contínuos no tempo, a saída a qualquer instante de tempo é função da entrada e da própria saída. Em qualquer instante de tempo uma variação mínima do sinal na entrada provoca alteração instantânea na saída. Como exemplo a velocidade no eixo turbina hidráulica é uma função contínua no tempo da vazão ou velocidade do líquido que vai de encontro a suas pás. Para o controle da velocidade do eixo desta turbina, basta ter-se controle sobre a vazão ou velocidade da água. Se esta turbina alimenta um gerador de eletricidade que

necessita de velocidade constante, e este gerador está sujeito à variação de carga, então ocorrerão os chamados distúrbios no sistema. A Figura 2.1 mostra um esquema de uma turbina hidráulica com hélices alimentada por água de uma barragem e que por sua vez alimenta um gerador elétrico.

Figura 2.1 - Sistema turbina hidráulica alimentando gerador elétrico



Fonte: (VASCONCELLOS, 2010)

Um distúrbio ou perturbação é caracterizado por um sinal que tende afetar de modo adverso o valor da variável de saída de um sistema. Se um distúrbio for gerado internamente no sistema, ele é dito um distúrbio interno; ao passo que um distúrbio externo é produzido fora do sistema e se comporta como um sinal de entrada no sistema. (OGATA,1998, p. 2)

No exemplo do sistema com turbina hidráulica alimentando gerador, o desgaste dos rolamentos do eixo da turbina pode ser considerado um distúrbio interno, pois é inerente ao sistema, provoca variação na saída se não for compensando, e geralmente é conhecido e previsível. A vazão necessária aplicada às pás da turbina pode ser dita como uma função da carga aplicada ao gerador. Se a carga diminui ou aumenta, a velocidade no eixo da turbina irá se alterar inversamente proporcional à carga exigida do gerador, por conta da variação da força contra eletromotriz aplicada ao eixo da turbina. Um curto-circuito ocorrido na rede de distribuição de energia elétrica pode ser avaliado como distúrbio externo, pois provoca um repentino aumento da carga exigida, o que tende a diminuir a velocidade do eixo. Se um sistema de proteção atuar e desligar totalmente a carga acoplada ao gerador, esta desconexão repentina aumentará a velocidade do eixo da turbina.

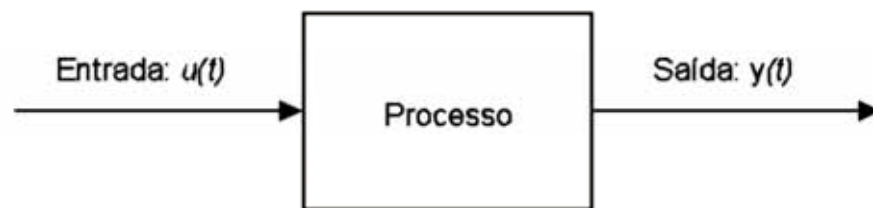
Um controlador automático tem como objetivo manter o valor desejado na saída do processo, eliminando ou reduzindo ao máximo a ação de possíveis distúrbios presentes no

sistema. Uma classificação importante do tipo de controle realizado é se este é realizado em malha aberta ou malha fechada.

2.2 CONTROLE A MALHA ABERTA VERSUS MALHA FECHADA

O controle a malha aberta caracteriza-se pelo fato de o sinal de saída não afetar a ação de controle. Na Figura 2.2, tem-se a representação simplificada de tal sistema de controle. O sistema é dinâmico e representado por um modelo matemático caracterizado por equações diferenciais. A entrada $u(t)$ é chamada de variável manipulada ou variável de controle (MV), e a saída $y(t)$ é a variável controlada ou variável de processo (PV).

Figura 2.2 - Diagrama de blocos de um sistema de controle a malha aberta



Fonte: (MACIEL, 2012)

Um exemplo simples de um sistema de controle a malha aberta é o de um forno à gás. Normalmente existe na válvula de controle uma graduação de temperatura para o usuário. É possível se ter uma previsão da temperatura que estará presente no interior do forno através de um bom conhecimento das características de pressão, vazão e poder calorífico do gás que entrará em combustão, bem como do isolamento térmico do conjunto e faixa de temperatura externa ao qual o forno será submetido. Através deste conhecimento que é feita a graduação para o usuário na válvula. A real temperatura na qual o interior do forno estará não altera a vazão de gás ajustada pela válvula. Se ocorrerem distúrbios como alteração da composição química ou pressão do gás, entupimento dos dutos, deterioração da isolamento térmico do conjunto ou uma simples abertura da porta, a temperatura real no interior do forno estará cada vez mais distante da ajustada tendo como referência a graduação da válvula. De modo automático nada será feito para que isto seja corrigido.

É necessário haver um conhecimento prévio do sistema a ser controlado, seja por meio de um modelo matemático satisfatório ou por teste empírico, para o controle ser eficiente, possuir exatidão e precisão, necessárias em aplicações de sistemas de controle a malha aberta. A inexistência ou baixa influência de distúrbios internos ou externos é também uma exigência para que este tipo de controle seja eficiente. Calibrações e verificações são fundamentais para que o controle continue satisfatório ao longo do tempo.

Em processos onde são exigidas maiores exatidão, precisão, e distúrbios aleatórios tenham grande influência, é indispensável que haja medição das variáveis controladas. Tendo o controlador a medida em tempo real da variável controlada, ajustes da variável controlada podem ser realizados caso ocorram distúrbios no sistema. Este tipo de controle é definido como sistema de controle com retroação ou a malha fechada.

“A expressão controle a malha fechada acarreta sempre o uso de retroação a fim de reduzir o erro do sistema” (OGATA, 1998, p 5).

A Figura 2.3 mostra uma representação simples de um sistema de controle a malha fechada. O controle a malha fechada compara o sinal de retroação, que é o próprio sinal de saída $y(t)$ ou pode ser uma função do sinal de saída, com o sinal de referência $r(t)$ para obter o erro $e(t)$. O erro é a representação de quão distante do valor desejado está da variável de saída. As ações de controle do controlador são tomadas com base no erro, e a saída do controlador $u(t)$ que o processo recebe, é uma função do erro, podendo ser apenas uma proporção, uma derivada, uma integral, ou uma combinação destes.

Figura 2.3 - Diagrama de blocos de um sistema de controle a malha fechada



Fonte: (MACIEL, 2012)

A ação de distúrbios externos ou as variações de parâmetros internos do sistema são total ou praticamente eliminadas em sistemas de controle a malha fechada, o que é uma vantagem em relação ao controle a malha aberta. Porém em um sistema de controle a malha

aberta é muito mais fácil se atingir a estabilidade da variável controlada. Um controlador a malha fechada pode exigir sintonia fina para que erros não sejam corrigidos além do necessário, ocasionando oscilações indesejadas na variável controlada. Como vantagens estão, a melhora da precisão, redução do efeito das não-linearidades e distorções, aumento da largura de faixa na qual o sistema terá resposta satisfatória e menor sensibilidade às variações das características do sistema. A tendência à instabilidade é uma desvantagem, bem como o maior número de componentes necessários para sua implementação em comparação com um sistema de controle a malha aberta.

2.3 TIPOS DE CONTROLADORES

Segundo Ogata (1998) de acordo com a ação de controle, controladores industriais analógicos ou digitais podem ser classificados como controladores de duas posições ou liga-desliga (*on-off*), ou dos tipos: proporcional, proporcional e integral, proporcional e derivativo, proporcional, integral e derivativo.

Um controlador automático compara o valor real da grandeza de saída do processo com a grandeza de referência (valor desejado), determina o desvio e produz um sinal de controle que reduzirá o desvio a zero ou a um valor pequeno. A maneira pela qual o controlador automático produz o sinal de controle é então chamada ação de controle. (OGATA, 1998, p. 177)

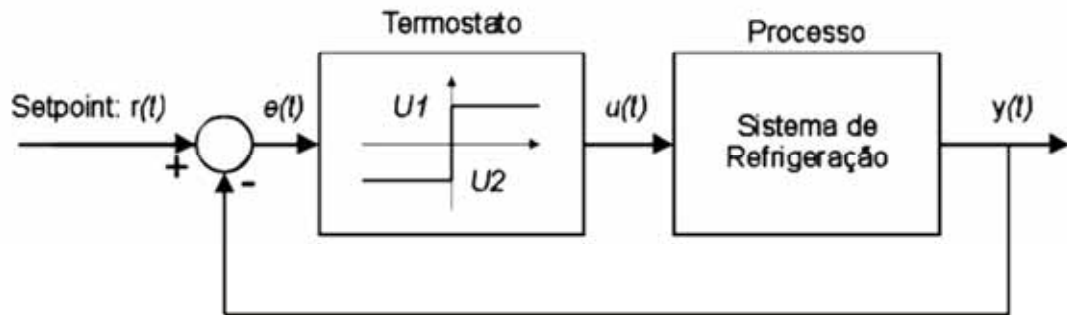
Existem métodos mais modernos de controle, como por lógica difusa (*Fuzzy*) ou redes neurais. Com a evolução da capacidade de processamento versus custo de implementação, estes métodos vêm se tornando cada vez mais atrativos principalmente para processos onde há grande complexidade.

2.3.1 Controladores de duas posições ou liga-desliga (*on-off*)

O controle de duas posições ou do inglês *on-off*, é utilizado principalmente em controladores domésticos devido sua simplicidade e baixo custo. Sua saída possui apenas dois estados, um ligado e outro desligado. Como exemplo, vários sistemas de refrigeração utilizam este tipo de controle, pois pequenas variações em torno da temperatura desejada que possam ocorrer não causarão grande desconforto, no caso de condicionadores de ar, ou perda da

conservação de alimentos ou qualquer outro tipo de material no caso de câmaras de refrigeração.

Figura 2.4 - Diagrama de blocos de um controlador liga-desliga



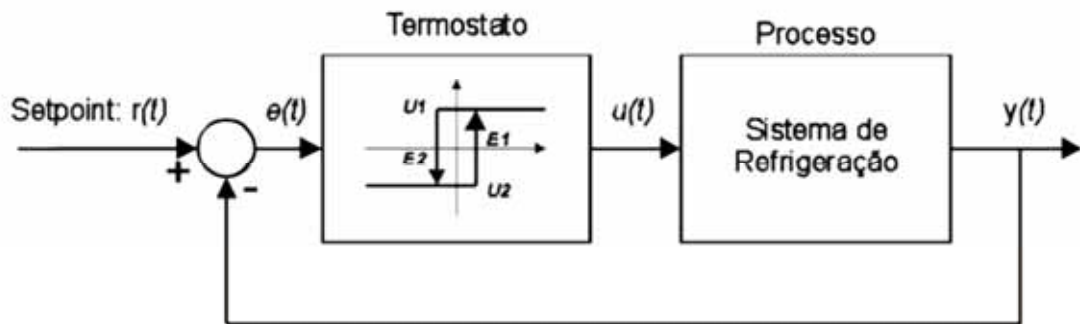
Fonte: (MACIEL, 2012)

Na Figura 2.4 vê-se que o sinal de controle $u(t)$ assume somente dois valores, um se o erro for positivo e outro se o erro for negativo. A equação (2.1) demonstra matematicamente:

$$u(t) = \begin{cases} U1 & \text{se } e(t) > 0 \\ U2 & \text{se } e(t) < 0 \end{cases} \quad (2.1)$$

Percebe-se que não existe definição para $u(t)$ quando o erro é nulo. Dependendo da velocidade com que a variável controlada do processo varia e se esta tende a ficar em torno do valor que gera erro nulo, a comutação entre ligado e desligado pode ocorrer muitas vezes dentro de um pequeno intervalo de tempo. Isto é normalmente indesejável para sistemas eletromecânicos, pois geralmente resulta em maior consumo de energia e diminuição da vida útil dos componentes. Ao se adicionar um intervalo diferencial, comumente referenciado como histerese ou banda morta - do inglês *dead-band* - há a tendência da eliminação deste problema. A Figura 2.5 mostra o diagrama de blocos para um sistema com histerese, e a equação (2.2) demonstra matematicamente.

Figura 2.5 - Diagrama de um controlador on-off com intervalo diferencial



Fonte: (MACIEL, 2012)

$$u(t) = \begin{cases} U1 & \text{se } e1(t) > 0 \\ U2 & \text{se } e2(t) < 0 \end{cases} \quad (2.2)$$

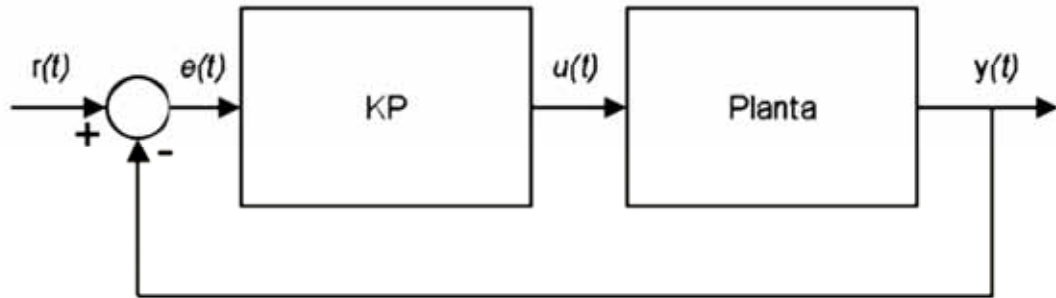
No sistema de refrigeração, evita-se que o acionamento e desligamento ocorram constantemente quando a temperatura alterna seu valor próximo da desejada. O sistema é acionado somente quando a temperatura atinge um valor ligeiramente maior do que o desejado, e é desligado quando a temperatura é ligeiramente menor que a desejada. Assim a frequência de comutação entre os estados ligado e desligado é diminuída e ainda assim a temperatura é mantida dentro de limites desejáveis. Quanto maior a amplitude do intervalo diferencial, menor será a frequência de comutação entre os estados, mas será maior o desvio do valor da variável de saída em torno do valor desejado.

2.3.2 Controladores proporcionais

O controlador proporcional tem a característica de um amplificador linear com ganho ajustável. O ganho é a razão entre o erro $e(t)$ e a saída do controlador $u(t)$. A equação (2.3) demonstra matematicamente sua ação e a Figura 2.6 mostra o diagrama de blocos para tal controlador.

$$u(t) = K_p \cdot e(t) \quad (2.3)$$

Figura 2.6 - Diagrama de blocos de um controlador proporcional



Fonte: (MACIEL, 2012)

O ganho K_p é conhecido como ganho proporcional. É sabido que ao se aumentar o ganho proporcional, é diminuído o erro em regime permanente, ou também chamado regime estacionário ou *offset*, inevitável no controlador proporcional. O erro em regime permanente é definição para a diferença entre o valor da variável de saída real e a referência desejada. Entretanto não se pode aumentar indefinidamente o ganho proporcional, pois o mesmo torna a resposta do sistema mais oscilatória, e pode levar o sistema rapidamente a saturação dos atuadores.

2.3.3 Controladores do tipo proporcional e integral

A ação de controle integral diferencia-se da ação de controle proporcional pelo fato de o valor da saída do controlador $u(t)$ variar conforme uma taxa proporcional ao sinal do erro $e(t)$, e não ser uma razão do próprio erro. Isto faz com que ela atue ao longo do tempo quando existir diferença entre o valor desejado e o valor medido, e não apenas instantaneamente como no caso do controlador proporcional. As equações (2.4) e (2.5) demonstram matematicamente controlador que tem somente a ação de controle integral e seu comportamento.

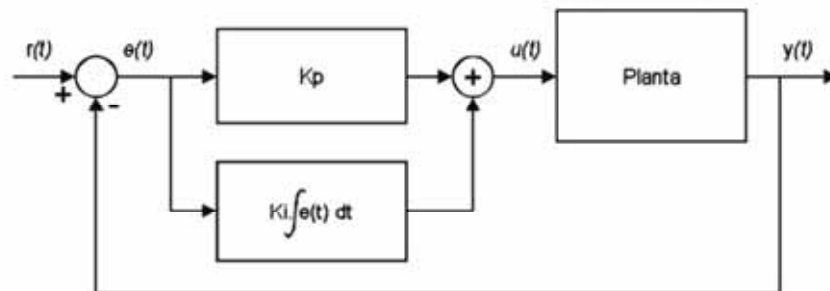
$$\frac{du(t)}{dt} = K_i \cdot e(t) \quad (2.4)$$

$$u(t) = K_i \int_0^t e(t) dt \quad (2.5)$$

K_i é o ganho integral ajustável. Na ação de controle proporcional se o valor do erro for multiplicado por um fator constante, a saída do controlador terá seu valor multiplicado instantaneamente pelo mesmo fator. Já na ação do tipo integral este fator irá multiplicar a velocidade com que o sinal de saída $u(t)$ varia. A ação de controle integral é também denominada controle de reestabelecimento (*reset*).

Isoladamente a ação de controle integral normalmente não representa um método de controle e é associada com outros tipos de ação de controle como a proporcional, a derivativa ou uma associação das três, que caracteriza o controle PID. Um dos problemas apresentados pelo controlador proporcional é a existência do erro em regime estacionário na resposta a uma excitação em degrau. A ação de controle integral em conjunto com a proporcional tem como característica a eliminação de tal erro residual. Um controlador do tipo proporcional e integral (PI) pode ser representado matematicamente conforme a equação (2.6), e seu diagrama de blocos é demonstrado pela Figura 2.7:

Figura 2.7 - Diagrama de blocos de um controlador do tipo proporcional e integral



Fonte: (MACIEL, 2012)

$$u(t) = K_p \cdot e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt \quad (2.6)$$

T_i é o chamado tempo integral, e assim como K_p é ajustável. O inverso do tempo integral é denominado taxa de reestabelecimento, a qual é medida em repetições por minuto e representa o número de vezes por minuto que a parte proporcional da ação de controle é duplicada.

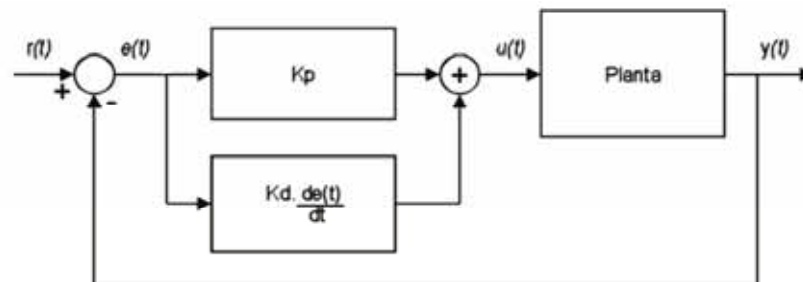
2.3.4 Controladores do tipo proporcional e derivativo

A ação de controle derivativa assim como a integral não é isoladamente um método de controle. Ela tem como característica antecipar o erro atuante, podendo gerar na saída do controlador um valor significativo antes que o sinal do erro se torne um valor grande o suficiente para o controlador proporcional, aumentando assim a sensibilidade do controlador. A equação (2.7) representa matematicamente a resposta de um controlador proporcional e derivativo.

$$u(t) = K_p \cdot e(t) + K_p T_d \frac{de(t)}{dt} \quad (2.7)$$

T_d é uma constante ajustável chamada tempo derivativo. A ação de controle derivativa faz a magnitude da saída do controlador ser proporcional à taxa de variação do erro atuante, e o tempo derivativo representa o intervalo de tempo pelo qual a ação do controle proporcional é antecipada pela ação de controle derivativa. A Figura 2.8 ilustra o diagrama de blocos de um controlador proporcional e derivativo

Figura 2.8 - Diagrama de blocos de um controlador do tipo proporcional e derivativo



Fonte: (MACIEL, 2012)

“O controle derivativo, portanto, antecipa o erro atuante e inicia uma ação corretiva mais cedo, tentando aumentar a estabilidade do sistema” (OGATA, 1998, p. 188)

Quanto ao regime estacionário, pode-se dizer que a ação de controle derivativa não tem influência direta. Porém como é introduzido um amortecimento a resposta do sistema, há a possibilidade de se utilizar um maior valor de ganho proporcional, fato que acarreta na diminuição do erro estacionário. E a introdução da ação de controle derivativa também aumenta a presença de oscilações na saída do controlador.

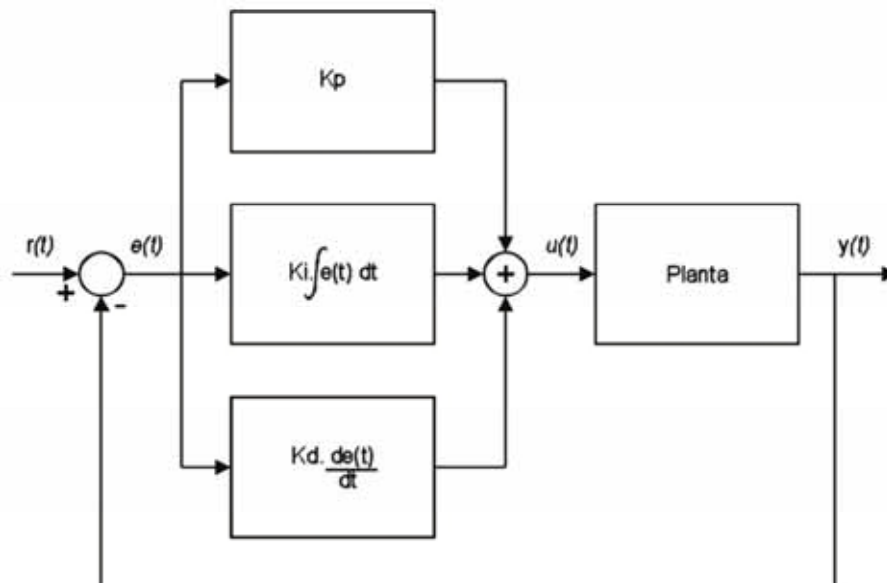
2.3.5 Controladores do tipo proporcional, integral e derivativo (PID)

O controlador proporcional, integral e derivativo, ou simplesmente controlador PID, é frequentemente implementado com o objetivo de unir as vantagens das ações de controle proporcional, integral e derivativa. Havendo possibilidade de ajuste de cada um dos ganhos separadamente, se torna possível a chamada sintonia do controlador PID. Existem variados métodos para se determinar a melhor sintonia, a qual faz com que o sistema tenha a melhor resposta possível em termos de erro estacionário, tempo de subida, estabilidade, tempo de acomodação entre outros fatores tanto no regime permanente quanto transitório.

Pode-se representar um controlador PID ideal através da equação (2.8), e seu diagrama de blocos é mostrado na Figura 2.9:

$$u(t) = K_p \cdot e(t) + K_d \frac{de(t)}{dt} + K_i \int_0^t e(t) dt \quad (2.8)$$

Figura 2.9 - Diagrama de blocos de um controlador PID

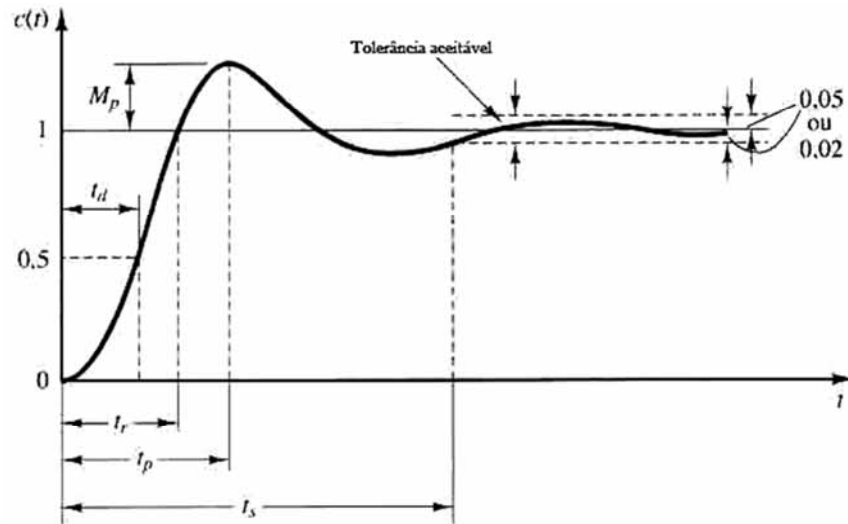


Fonte: (MACIEL, 2012)

Segundo Ogata (1998) os parâmetros geralmente considerados para análise da resposta transitória a excitação por um degrau unitário, mostrados na Figura 2.10, são: tempo de atraso t_d (50% valor final), tempo de subida t_r (10% a 90%, 5% a 95% ou 0% a 100% dependendo das características de amortecimento do sistema), instante de pico t_p (primeiro sobressinal),

máxima sobressinal M_p (valor de pico do *overshoot*) e tempo de acomodação t_s (gasto até se atingir a faixa de tolerância).

Figura 2.10 - Especificações da resposta transitória ao degrau unitário



Fonte (OGATA, 1998)

Segundo Lotufo (2013) analisando isoladamente cada ação de controle, pode-se dizer que o controlador proporcional tem como vantagem, com o aumento de seu ganho K_p , reduzir o tempo de subida e o erro de estado estacionário, embora nunca o elimine. Como desvantagem, principalmente em processos onde a inércia é relativamente grande, o controlador proporcional tende a aumentar o sobre-sinal (*overshoot*). Fato que é muitas vezes indesejado ou até perigoso, dependendo do processo. O controlador integral, com aumento do ganho K_i , tende a diminuir o tempo de subida e como grande vantagem eliminar o erro de estado estacionário. Por outro lado também aumenta o *overshoot* e o tempo de acomodação da resposta do sistema, traduzido como uma piora na resposta transitória. Já o controlador derivativo tem como principais vantagens as diminuições dos valores máximos do *overshoot* e do tempo de acomodação, melhorando a estabilidade e resposta transitória do sistema. Além disso, por introduzir um amortecimento a resposta do sistema, permite que seja utilizado um ganho proporcional maior. Assim reduzindo um pouco mais o erro de estado estacionário. Como o controlador PID é uma soma dos três tipos de ação de controle, não se pode isolar a influência de cada tipo de ação nos parâmetros acima discutidos. Uma alteração no ganho de um dos controladores, ao alterar a resposta do sistema terá influência na saída do outro controlador.

No momento da sintonia do controlador PID, deve se prever os padrões dos sinais que serão utilizados como referência no controlador. Através de análises matemáticas do modelo, caso este seja conhecido, e/ou testes empíricos no local de aplicação deve-se determinar os valores dos ganhos proporcional, integral e derivativo que atendam as especificações de operação e os limites do sistema tanto para regime estacionário quanto transitório. Tais limites e especificações dependem dos fatores que serão levados em consideração no sistema, como desgaste e vida útil dos componentes, consumo de energia, segurança, capacidade e qualidade da produção, limitações físicas dos componentes eletromecânicos aplicados, entre outros.

2.4 CONTROLE DIGITAL

É interessante assinalar que mais da metade dos controladores industriais em uso nos dias atuais utiliza a estratégia de controle PID ou PID modificadas. A maioria dos controladores analógicos é hidráulica, pneumática, elétrica e eletrônica, ou resulta de uma combinação destes tipos. Correntemente, muitos deles são transformados em digitais por intermédio dos microprocessadores (OGATA, 1998, p. 544)

O princípio de atuação de um controlador é o mesmo, esteja este representado por equações diferenciais contínuas ou discretas no tempo. Na representação matemática de sistemas contínuos no tempo é comum se utilizar a Transformada de Laplace. Sua vantagem é de transformar funções comuns, como exponenciais e senoidais, em funções algébricas de uma variável complexa s . Também facilita o cálculo ao obter tanto a resposta transitória quanto a resposta em regime permanente, e permite utilização de técnicas gráficas para determinação do desempenho de um sistema sem a necessidade da resolução de equações diferenciais ordinárias.

Similar a Transformada de Laplace, para representação de sistemas discretos no tempo é comumente utilizada a Transformada Z. Assim como a Transformada de Laplace transforma equações diferenciais lineares invariantes no tempo em equações algébricas em s , a Transformada Z transforma equações a diferença lineares invariantes no tempo em equações algébricas em z , com a diferença que aplica a um sinal contínuo $x(t)$, um sinal amostrado $x(kT)$, ou uma sequência numérica $x(k)$.

2.4.1 Amostragem e quantização

No mundo real, onde o tempo é contínuo, não existem variáveis de tempo discreto. Em uma função contínua, em cada mínima fração de tempo sempre haverá um valor definido, e sua amplitude também possui infinitos possíveis valores a assumir. Os computadores e microcontroladores baseiam seu processamento em ciclos de relógio (ciclo de *clock* ou instrução). Pode-se resumir o ciclo a seguinte sequência: no início do ciclo é realizada a leitura e armazenamento dos valores das entradas digitais, então é feito o cálculo ou processamento dos valores das saídas digitais e estas tem seus valores armazenados em variáveis virtuais, ao fim do ciclo, os valores calculados para as saídas são atualizados nas saídas reais.

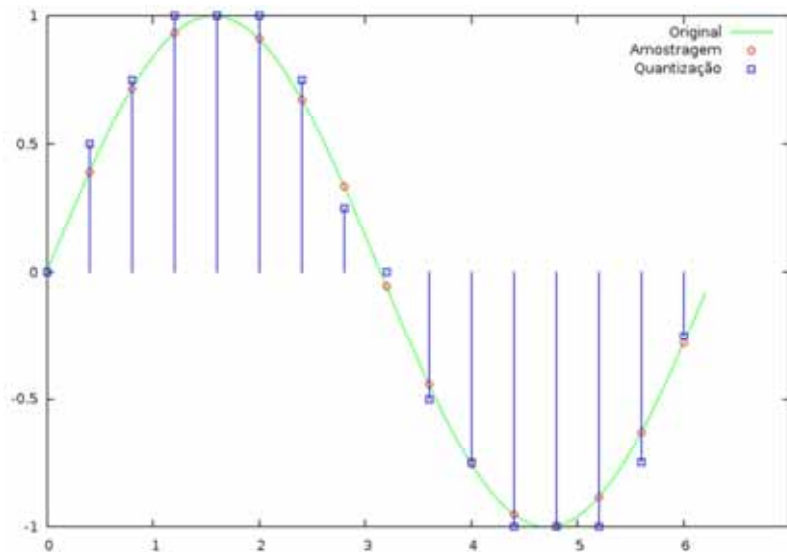
Nos processadores digitais, esta interrupção entre leitura das entradas, processamento e escrita das saídas, impede que seja realizado em tempo real o processamento de funções matemáticas contínuas no tempo. As funções matemáticas devem ser amostradas, ou seja, representadas por funções no tempo discreto, possibilitando assim haver um intervalo de tempo entre as amostras para que ocorra o processamento das equações por parte do processador. Para isto é necessário se representar uma variável contínua no tempo, através de uma sequência numérica digital, como uma função discreta no tempo com valores finitos de amplitude e definidos apenas em tempos específicos. Ao processo de se representar uma variável contínua no tempo por uma função definida no tempo discreto, dá-se o nome de amostragem. Se um sinal amostrado no tempo pode assumir infinitos valores de amplitude, este é chamado de sinal amostrado. Isto é, um trem de pulsos modulado em amplitude.

Além das limitações dadas pelo ciclo de relógio, sistemas digitais possuem limitações de endereçamento e memória. Então para ser possível representar uma função contínua no tempo em um sistema digital, além da amostragem, é necessário se definir para a amplitude do sinal uma faixa máxima de possíveis valores e um passo, ou diferença mínima entre estes valores. Dá-se o nome de quantização ao processo de se representar a amplitude de variáveis contínuas através de um conjunto de valores finitos distintos.

A Figura 2.11 demonstra o processo de quantização e amostragem aplicados a uma função contínua no tempo. A linha contínua é um sinal contínuo em amplitude e no tempo. Em um intervalo de seis segundos, o sinal foi atualizado 15 vezes, o que sugere um período

de amostragem de quatrocentos milissegundos. Os círculos vermelhos sobre a linha contínua representam o sinal amostrado, que é definido em finitos valores no tempo, mas é contínuo em amplitude. Os valores adotados pelo sinal quantizado, representado pelos quadrados azuis coincidentes no tempo com o sinal amostrado, sugerem um passo de 0,25 unidades adotado na quantização.

Figura 2.11 - Amostragem e Quantização de uma função contínua no tempo



Fonte: (HUDLAC, 2009)

Matematicamente a saída do amostrador pode ser representada conforme a equação (2.9):

$$u^*(t) = \sum_{k=0}^{\infty} u(kT)\delta(t - kT) \quad (2.9)$$

T é o período de amostragem. A saída do amostrador é o produto do sinal de entrada $u(t)$ por um trem de impulsos. Tomando a transformada de Laplace da equação (2.9) teremos:

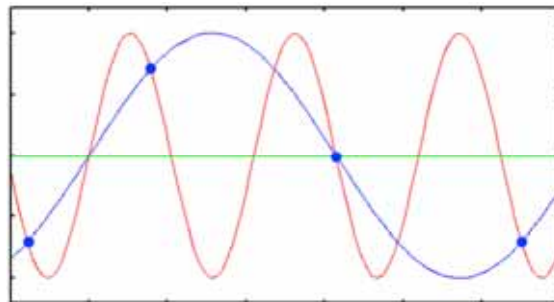
$$L\{u^*(t)\} = \sum_{k=0}^{\infty} u(kT)L\{\delta(t - kT)\} = \sum_{k=0}^{\infty} u(kT)e^{-kTs} \quad (2.10)$$

A transformada Z do sinal amostrado então será:

$$Z\{u(k)\} = \sum_{k=0}^{\infty} u(kT)z^{-k} \quad (2.11)$$

Segundo Ogata (1995) um fator importante a ser considerado é o teorema da amostragem. Tal teorema diz que para que seja possível recuperar um sinal contínuo limitado em frequência, através de uma sequência finita de amostras, é necessário que a frequência de amostragem seja diferente e no mínimo duas vezes maior que a maior frequência presente no sinal amostrado. Não respeitado tal teorema, frequências do sinal amostrado podem ser erroneamente representadas, o que pode provocar um efeito conhecido como *aliasing* no momento da reconstrução do sinal. A Figura 2.12 ilustra o erro na reconstrução de um sinal amostrado com frequência menor que o dobro de sua maior frequência. O sinal original em vermelho é amostrado nos pontos representados pelos círculos azuis. A reconstrução do sinal das amostras é representada pela linha azul.

Figura 2.12 - Aliasing devido à baixa frequência de amostragem



Fonte: (SCIENTIFIC, [201-?])

Quanto maior a frequência de amostragem, maior será a informação obtida sobre a variação de um sinal no tempo e aproxima-se cada vez mais um sinal discreto de um sinal contínuo. Na prática, a velocidade de conversão do conversor analógico para digital (AD) e vice-versa (DA), bem como a velocidade dos barramentos e processamento dos sistemas digitais, são os fatores que limitam a frequência de amostragem. Neste trabalho, foi utilizada uma frequência fixa de amostragem, ou seja, o período entre uma amostra e outra será sempre o mesmo, que é a técnica mais simples a ser implementada. Isto não permite afirmar que é menos precisa ou satisfatória quando aplicada corretamente.

No caso da quantização, em comparação com a amostragem, pode-se dizer que quanto mais restrita a faixa de amplitude de um sinal contínuo e menor o passo entre os possíveis valores de amplitude, melhor será representado o sinal e mais próximo este estará do sinal contínuo em amplitude. A amostragem limita a maior frequência permissível de um sinal. Devido a limitações dos conversores AD, que normalmente operam com tensão e possuem limitados números de bits para representação do valor convertido em código binário, pode-se

dizer que a quantização impõe os limites superior e inferior dos valores que a amplitude de um sinal pode assumir. Valores maiores que o limite superior são representados na forma digital com o seu valor limite. O mesmo ocorre para valores menores que o limite inferior. Quando isto ocorre pode-se dizer que houve saturação dos limites de amplitude do conversor AD. Alguns conversores têm a capacidade de informar o processador quando tal erro de conversão ocorre. De qualquer forma, haverá erro na conversão, o que é indesejado. (OGATA, 1995)

O erro de quantização é outro fator a ser considerado em um sistema digital. Quando o valor da amplitude de uma amostra do sinal contínuo se encontra entre dois valores possíveis a serem assumidos digitalmente, o conversor AD dá a esta amostra o valor possível mais próximo do valor da amostra. Isto implica que um erro na representação ocorrerá, e este terá valor absoluto compreendido entre zero e metade do valor do passo de amplitude. Um menor passo de amplitude, chamado de nível de quantização, diminui este erro e representa mais fielmente o sinal contínuo. Por outro lado exige maior número de bits (maior resolução) tanto do conversor AD quanto das memórias e processador. A Figura 2.13 ilustra o erro de quantização e a saturação no sinal que ocorrem em um conversor AD não ideal.

Figura 2.13 - Erro de quantização e saturação de um conversor AD

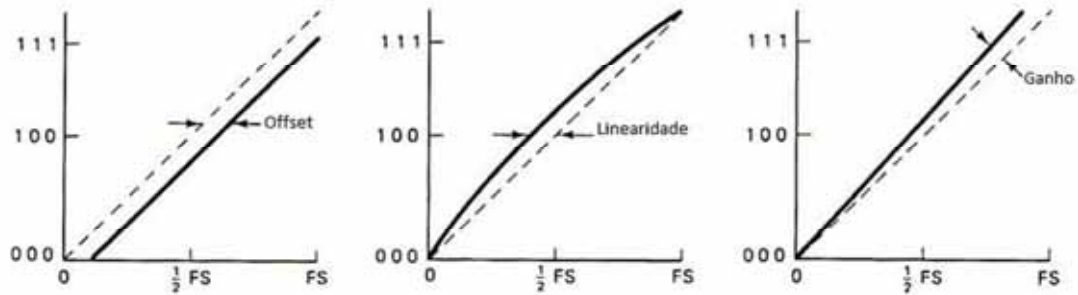


Fonte: (DIRACDELTA, [201-?])

Outros tipos de erros a serem considerados em um conversor AD são os erros de *offset*, linearidade e ganho. As influências na resposta do sistema de tais erros e seus limites devem ser estudadas no momento da escolha do conversor AD, pois estes erros são intrínsecos. Tais erros surgem devido às não linearidades e distorções dos componentes analógicos presentes nos conversores AD (amplificadores operacionais, comparadores, etc). É possível se

compensar os erros de offset e ganho com maior facilidade, pois geralmente são constantes e lineares. A Figura 2.14 demonstra a influência de tais erros na resposta do conversor DA. (OGATA, 1995)

Figura 2.14 - Erros de *offset*, linearidade e ganho de um conversor AD



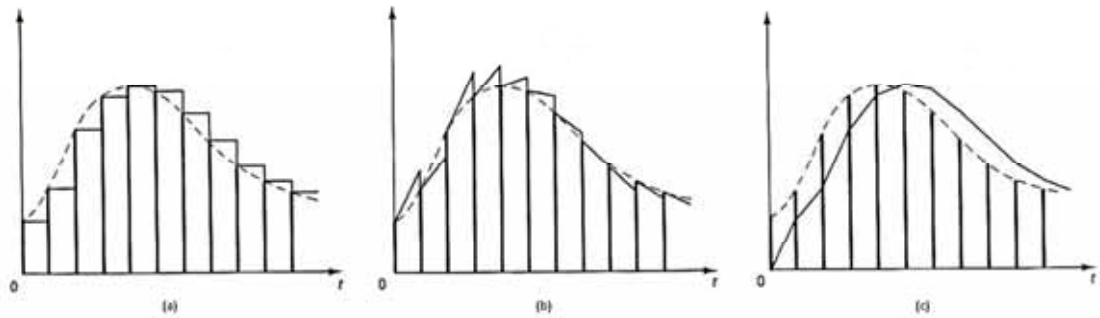
Fonte: (OGATA, 1995)

2.4.2 Conversão digital para analógica

Para recuperação do sinal contínuo ou analógico, através de um sinal digital, são utilizados conversores digital para analógico (DA). Um conversor DA recebe um sinal digital, que pode ser um sinal contínuo amostrado e quantizado, ou como no caso deste trabalho, uma função de saída calculada pelo processador e transforma esta em uma função contínua. Existem diferentes formas de se recuperar um sinal digital. Neste trabalho, foi utilizado o princípio da retenção de ordem zero (*Zero-Order Hold* ou *ZOH*). Esta é a técnica mais simples e consiste em considerar o valor do sinal de saída contínuo constante e igual à última amostra recebida pelo controlador DA até que a próxima amostra seja recebida, quando então o valor é atualizado.

Existem técnicas como a de retenção de primeira ordem, primeira ordem com interpolação entre outras. Estas reproduzem com maior fidelidade um sinal contínuo quando feita sua reconstrução através de um sinal digital. Além da maior complexidade exigida na implementação, tais técnicas de reconstrução têm desvantagens, como erro na predição quando há inversão da derivada do sinal (no caso de primeira ordem), ou atraso de um período de amostragem (primeira ordem com interpolação). A Figura 2.15 mostra a reconstrução de um sinal amostrado através de diferentes tipos de retenção.

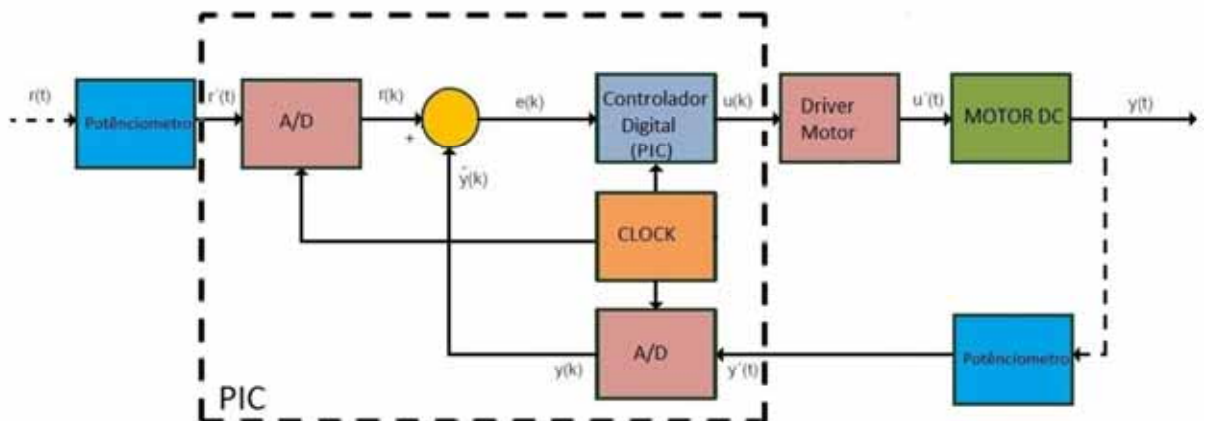
Figura 2.15 - Recuperação de sinal amostrado com (a)retenção de ordem-zero (b)primeira ordem (c)primeira ordem com interpolação



Fonte: (OGATA, 1995)

Neste trabalho não foi utilizado um conversor D/A comum, que converte um sinal digital binário em um sinal de tensão analógico. Como o sistema busca igualar o posicionamento angular de um eixo de entrada com o do eixo de saída, a saída do controlador atua no sistema através do controle da velocidade de um motor de corrente contínua (motor DC). O motor de corrente contínua tem sua velocidade controlada através da técnica de modulação por largura de pulso (PWM). A saída do controlador é na realidade o valor do ciclo de trabalho da onda quadrada aplicada ao circuito eletrônico que aciona o motor (*driver*). Variando-se o ciclo de trabalho, varia-se o valor médio da corrente aplicada ao motor, o que é traduzido como uma variação na velocidade do mesmo. O motor atua movimentando o eixo de saída a fim de corrigir a diferença de posição angular em relação à entrada. O diagrama de blocos do controlador digital implementado é apresentado na Figura 2.16:

Figura 2.16 - Diagrama de blocos do sistema controlado pelo PIC



Fonte: (produção do próprio autor)

2.5 APROXIMAÇÃO DA INTEGRAL E DERIVADA PARA EQUAÇÕES ALGÉBRICAS

Partindo do princípio que microcontroladores operam sempre na forma discreta no tempo, é possível afirmar que, a resolução de equações em tempo real só é possível se estas equações estiverem representadas no tempo discreto. Uma variável do mundo real, representada por uma função contínua no tempo, deve ser amostrada. O período de amostragem desta variável deve ser suficiente para que entre uma amostra e outra sejam realizados todos os cálculos das funções dependentes desta variável. Assim garante-se que as funções de saída do microcontrolador tem o mesmo período de amostragem das entradas. Não há muito sentido em amostrar as entradas com um período menor do que o mínimo necessário para fechamento de um ciclo do microcontrolador (leitura das entradas, processamento, escrita das saídas). (OGATA, 1995)

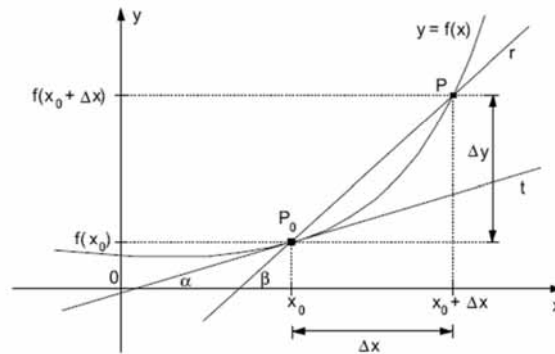
No caso da implementação de um controlador PID, é necessário que estejam disponíveis para o microcontrolador o erro e suas funções integral e derivada. A saída do controlador proporcional é simplesmente a multiplicação do erro, obtido pela diferença entre a entrada e saída amostradas, pelo ganho proporcional. Para que operem de forma digital os controladores integral e derivativo, é necessário que sejam calculadas pelo microcontrolador as funções integral e derivada do erro. A fim de diminuir a carga de processamento do microcontrolador, são utilizadas aproximações que transformam o cálculo da integral e derivada da função erro em equações algébricas envolvendo apenas soma, subtração, multiplicação e divisão. Operações estas simples de serem implementadas em um microcontrolador. (PEREIRA, 2003)

2.5.1 Aproximação da derivada pelo método das diferenças finitas (Euler)

A definição de derivada é a taxa de variação instantânea de uma função. Matematicamente, define-se um intervalo Δx no tempo. A diferença $\Delta f(x)$ entre o final e o começo do intervalo, dividida pelo tamanho do intervalo, representa a taxa de variação da função dentro do intervalo, como mostrado na Figura 2.17 e equação (2.12):

$$\frac{\Delta f}{\Delta x} = \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x} \quad (2.12)$$

Figura 2.17 - Cálculo da variação de uma função dentro de um intervalo



Fonte: (DIRETORIA, 2009)

Ao reduzir-se o intervalo Δx ao máximo, ou seja, quando este tende a zero (intervalo infinitesimal), e calculando-se o limite da razão $\Delta y/\Delta x$, temos a derivada de uma função contínua em um instante particular de tempo x , como mostrado na equação (2.13).

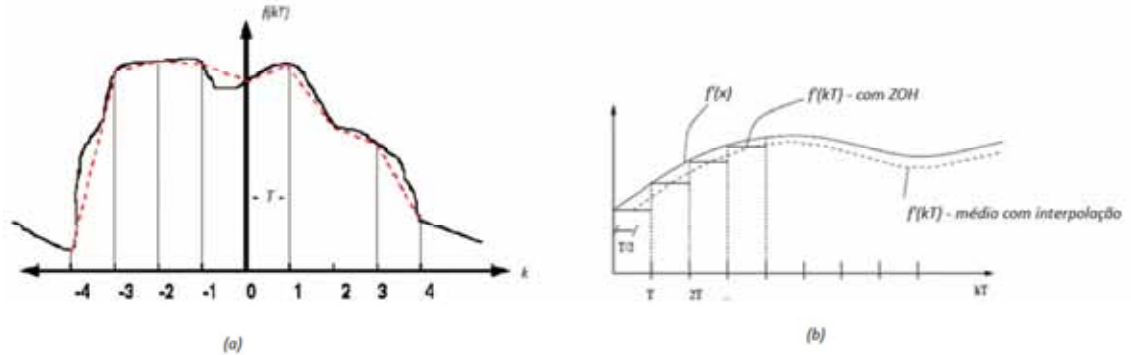
$$f'(x) = \frac{dy}{dx} = \lim_{\Delta x \rightarrow 0} \frac{\Delta y}{\Delta x} \quad (2.13)$$

No caso de sistemas digitais, tem-se o intervalo de tempo limitado pelo período de amostragem T . Partindo da equação (2.12) utiliza-se a aproximação representada pela equação (2.14), para diferenças finitas no tempo, conhecido como método de Euler. (MACIEL, 2012)

$$\frac{de(t)}{dt} \cong \frac{e(kT) - e(kT - T)}{T} \quad (2.14)$$

Como se trata de uma aproximação, obviamente haverá um erro inserido no cálculo da derivada utilizando-se este método. O valor da derivada é considerado constante dentro do intervalo e a função e sua derivada podem variar sem que isto seja percebido, e nem todos valores que a função assume serão considerados no cálculo. Através da Figura 2.18 (a) percebe-se que a derivada (inclinação) das semirretas pontilhadas em vermelho é constante dentro do período T , o que não é real. Além disso, percebe-se na Figura 2.18 (b) que utilizando-se a técnica de retenção de ordem zero, a derivada no ponto (k_0) até o ponto $(k_0 + T)$ leva em consideração a variação entre os pontos $(k_0 - T)$ e (k_0) , o que gera um certo atraso na obtenção da derivada.

Figura 2.18 - Erros na aproximação da derivada pelo método das diferenças finitas



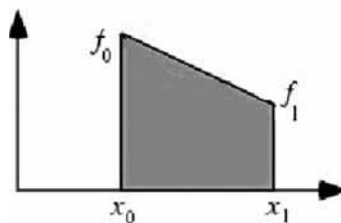
Fonte: (MACIEL, 2012)

Percebe-se que quanto menor o tempo de amostragem T do sistema, mais próximo este estará da situação ideal e menor será o erro ao se realizar este tipo de aproximação.

2.5.2 Aproximação da integral pela integração trapezoidal

A função integral definida caracteriza-se pela representação da área contida abaixo da curva de uma função definida dentro de um intervalo. Definido um intervalo Δx de $[x_0, x_1]$, a integral de $f(x)$ é a representação da área contida abaixo da curva $f(x)$ dentro do intervalo de x_0 a x_1 . Se dentro do intervalo Δx a função tem derivada constante, ou seja, é representada por uma rampa, tem-se a formação de uma figura trapezoidal, conforme a Figura 2.19, e a área sombreada S pode ser calculada conforme equação (2.15).

Figura 2.19 - Cálculo da área interna de uma figura trapezoidal



Fonte: (produção do próprio autor)

$$S = \left(\frac{f(x_0) + f(x_1)}{2} \right) \cdot (x_1 - x_0) = \frac{f(x_0) + f(x_1)}{2} \cdot \Delta x \quad (2.15)$$

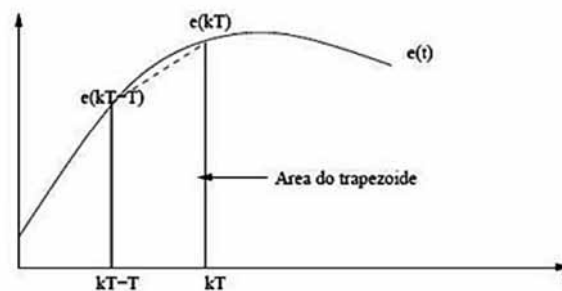
Ao dividir-se o intervalo entre x_1 e x_0 em valores menores de Δx , tem-se o valor da área S definido como uma soma de todas pequenas áreas abaixo da curva contidas no intervalo. Se Δx tender a zero, pode-se utilizar esta definição de área para qualquer tipo de função, não apenas em casos com derivada contínua. Isto é possível, pois se utiliza o valor da função no ponto $f(x_i^*)$. Ao multiplicar o valor da função no ponto por um Δx infinitesimal, tem-se um retângulo de área igual a $f(x_i^*)\Delta x$. Aplicando limite para Δx tendendo a zero nesta somatória tem-se a equação (2.16):

$$\int_{x_0}^{x_1} f(x)dx = \lim_{\Delta x \rightarrow 0} \left[\Delta x \sum_{i=1}^N f(x_i^*) \right] \quad (2.16)$$

A aproximação da integral pela integração trapezoidal consiste em dividir uma função contínua em pequenos intervalos de tempo, com valor igual ao período de amostragem T e considerar que esta varia com taxa constante dentro do período. Unindo assim as definições das equações (2.15) e (2.16), tem-se a somatória de área de pequenos trapézios com base de valor T não infinitesimal representada pela equação (2.17), sendo $e(t)$, o erro na entrada do controlador PID em $(t_1 - t_0)$ coincidente no tempo com $(kT - T)$. A Figura 2.20 representa graficamente tal aproximação:

$$\int_{t_0}^{t_1} e(t)dt \cong \frac{e(kT - T) + e(kT)}{2} T \quad (2.17)$$

Figura 2.20 - Aproximação da integral pelo método da integração trapezoidal



Fonte: (MACIEL, 2012)

A integral em tempo real deve considerar todos os valores acumulados no tempo, portanto se $u_i(kT)$ é o valor da função integral do erro $e(kT)$, pode-se dizer que:

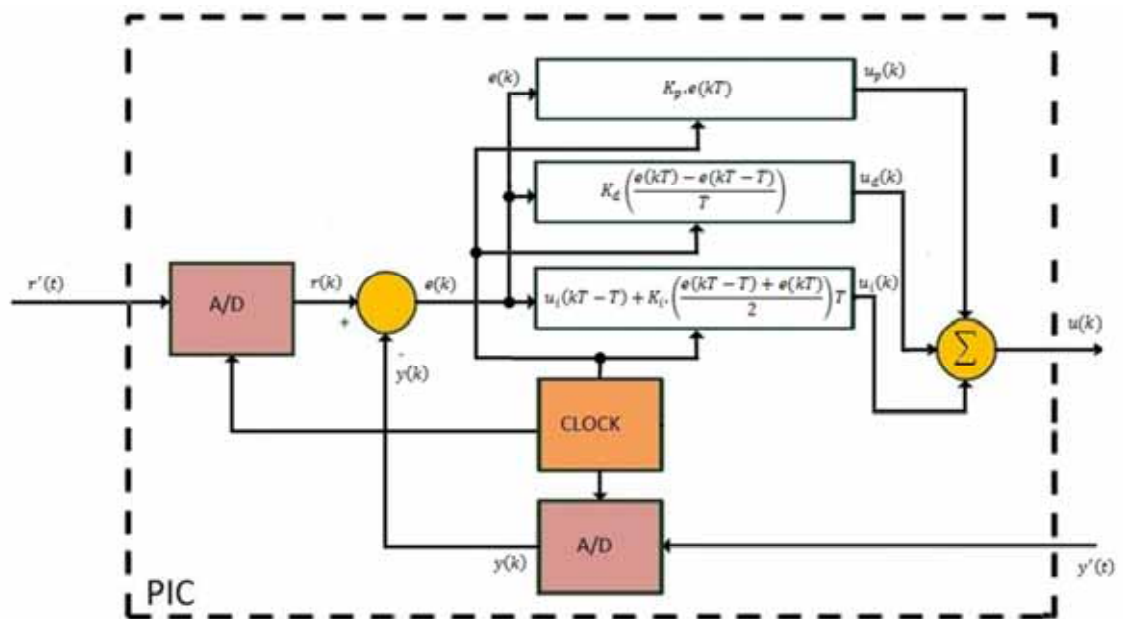
$$u_i(kT) = u_i(kT - T) + \frac{e(kT - T) + e(kT)}{2} T \quad (2.18)$$

Assim como a aproximação da derivada pelo método das diferenças finitas, a aproximação da integral pelo método da integração trapezoidal também insere um erro no cálculo da integral. Erro este, devido à aproximação da função para uma rampa durante o intervalo, e o uso de amostras anteriores ao instante em que se deseja calcular. A diminuição do tempo de amostragem T , faz com que se aproxime o sistema da situação ideal e assim seja diminuído o erro pela integração trapezoidal. (MACIEL, 2012)

2.5.3 Equação algébrica aproximada do controlador PID

Utilizando as aproximações da derivada pelo método das diferenças finitas e da integral pela integração trapezoidal, define-se a equação linear algébrica aproximada do controlador PID a ser implementada no microcontrolador PIC. A Figura 2.21 demonstra um diagrama de blocos do controlador com tal equação, esta descrita por (2.19) e (2.20).

Figura 2.21 - Diagrama de blocos controlador PID a ser implementado no PIC



Fonte: (produção do próprio autor)

$$u(kT) = u_p(kT) + u_d(kT) + u_i(kT) \quad (2.19)$$

$$u(kT) = K_p \cdot e(kT) + \frac{K_d}{T} [e(kT) - e(kT - T)] + u_i(kT - T) + \frac{K_i \cdot T}{2} [e(kT - T) + e(kT)] \quad (2.20)$$

Sendo que $u_p(kT)$ representa a ação de controle proporcional com seu respectivo ganho K_p . Já $u_d(kT)$ caracteriza a ação de controle derivativo com ganho K_d , enquanto $u_i(kT)$ representa a ação de controle integral com ganho K_i . Percebe-se que, o ganho integral K_i não é aplicado à porção acumulada da ação integral $u_i(kT - T)$, pois tal ganho já foi aplicado no momento anterior do cálculo de cada membro relacionado a amostras anteriores contidos na somatória.

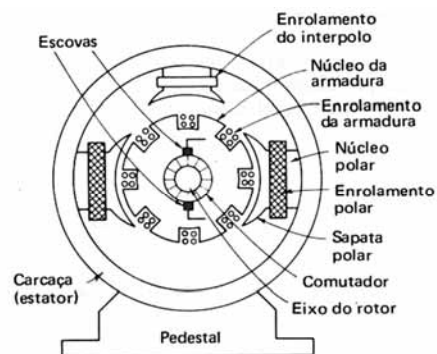
3 MOTOR DE CORRENTE CONTÍNUA

O motor de corrente contínua (motor DC) é amplamente utilizado em servossistemas, devido às suas características operacionais e de controle que proporcionam vantagens como: ampla faixa de variação de velocidade com torque constante, baixo momento de inércia, facilidade no controle da velocidade, flexibilidade e facilidade na inversão do sentido de rotação.

3.1 CARACTERÍSTICAS CONSTRUTIVAS DO MOTOR DC

A Figura 3.1 ilustra uma vista em corte transversal de um motor DC

Figura 3.1 - Características construtivas de um motor DC de 2 polos



Fonte: (TORO, 1999)

Basicamente o motor DC possui uma parte estática, o estator, uma parte rotativa, o rotor, e um sistema de comutação composto por um comutador e escovas. Estes são responsáveis pela inversão da corrente no enrolamento de armadura segundo a fase de rotação, com o objetivo de manter o conjugado sempre no mesmo sentido.

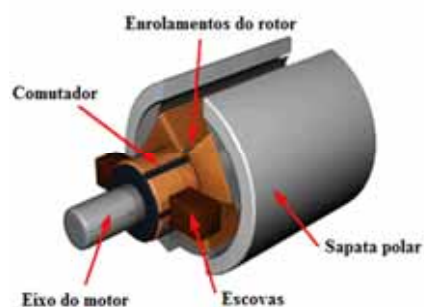
No estator, que é uma estrutura ferromagnética de polos salientes, são enroladas as bobinas de campo, responsáveis pela formação de um campo magnético fixo que irá interagir com o campo de armadura. O estator é composto de uma carcaça metálica que suporta a estrutura da armadura e providencia uma faixa de retorno de fluxo para o circuito magnético criado pelos enrolamentos de campo. Presos à carcaça estão os núcleos polares ou polos,

constituídos de ferro laminado e em torno destes ficam os enrolamentos de campo ou polares. A sapata polar é geralmente mais larga que o núcleo polar, para que o fluxo magnético seja distribuído de forma mais uniforme. Os interpolos e os enrolamentos de interpolo, quando presentes, são compostos de poucas espiras ligadas em série com o circuito de armadura. Sua função é diminuir o efeito indesejado causado pela comutação.

O rotor é um eletroímã montado sobre o eixo do motor. Fato que torna mecanicamente possível a rotação do núcleo de armadura, dos enrolamentos de armadura e do comutador. O núcleo de armadura é composto de camadas laminadas de aço, que juntamente com o tipo de material aplicado visa reduzir as correntes parasitas e diminuir a perda por histerese. Nas ranhuras axiais do núcleo são enrolados os enrolamentos de armadura, constituídos de bobinas isoladas entre si e que irão interagir com o campo dos enrolamentos do estator.

O sistema de comutação é composto pelo comutador e escovas. O comutador é uma superfície cilíndrica com diversas lâminas que são conectadas ao enrolamento do rotor, e é solidário ao eixo do motor. As escovas são ligadas aos terminais de alimentação e exercem pressão sobre o comutador. São geralmente construídas com ligas de carbono, o que as torna condutoras de eletricidade. São responsáveis pelo contato elétrico da parte girante (rotor) com a parte fixa do motor (estator). Devido ao atrito em movimento sofrem desgaste, o que traz a necessidade de troca após previsíveis horas de uso. O material que se desprende das escovas é responsável pela formação de um pó condutor preto, o qual deve ser constantemente retirado do interior do motor a fim de não causar contato elétrico indesejado ou sujar os rolamentos, aumentando os níveis de possíveis fugas elétricas e do atrito mecânico que atua contra o eixo do motor. A Figura 3.2 ilustra mais claramente a disposição física dos elementos internos de um motor DC já descritos.

Figura 3.2 - Componentes internos de um motor DC de 2 polos



Fonte: (ONLINE, [201-?])

O campo magnético fixo no estator pode ser gerado tanto por bobinas ou por ímãs permanentes. Quando a construção do motor DC torna inviável que um estator bobinado forneça a densidade de fluxo necessária ao longo do entreferro da máquina, o ímã permanente, o qual apresenta grande densidade de fluxo, torna-se uma solução atraente. A utilização de ímãs permanentes permite a construção de motores com tamanho e custo reduzido, devido ao fato de a construção se tornar mais simples.

3.2 PRINCÍPIO DE FUNCIONAMENTO DO MOTOR DC

O motor DC é uma máquina que quando energizada por corrente contínua, realiza conversão de energia eletromecânica, ou seja, há troca de energia entre sistema elétrico e mecânico por meio de um campo magnético de acoplamento. As relações entre a tensão de armadura e o torque mecânico desenvolvido no eixo do motor podem ser descritas pela Lei de Faraday e Lei de Ampère. A Lei de Faraday afirma que a força eletromotriz induzida num circuito elétrico fechado é igual à variação do fluxo magnético Φ_B que atravessa a área envolvida do circuito. Matematicamente tem-se a equação (3.1):

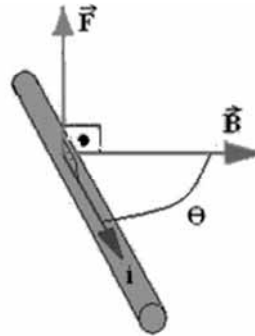
$$\oint_c E \cdot dl = - \frac{d\Phi_B}{dt} \quad (3.1)$$

A Lei de Ampère relaciona a densidade de fluxo magnético B sobre um laço com corrente elétrica i que passa através deste laço. A equação (3.2) demonstra matematicamente tal afirmação:

$$\oint B \cdot dl = \mu_0 \cdot i \quad (3.2)$$

Onde μ_0 é a permeabilidade magnética no vácuo. Em um condutor de tamanho l com corrente i , sendo atravessado por um campo magnético B , como mostrado na Figura 3.3, a força F ortogonal aos vetores campo e corrente, descrita pelas equações (3.3) e (3.4).

Figura 3.3 - Condutor elétrico retilíneo com corrente sob ação de um campo magnético



Fonte: (WATZECK, 2013)

$$\vec{F} = i \cdot \vec{l} \times \vec{B} \quad (3.3)$$

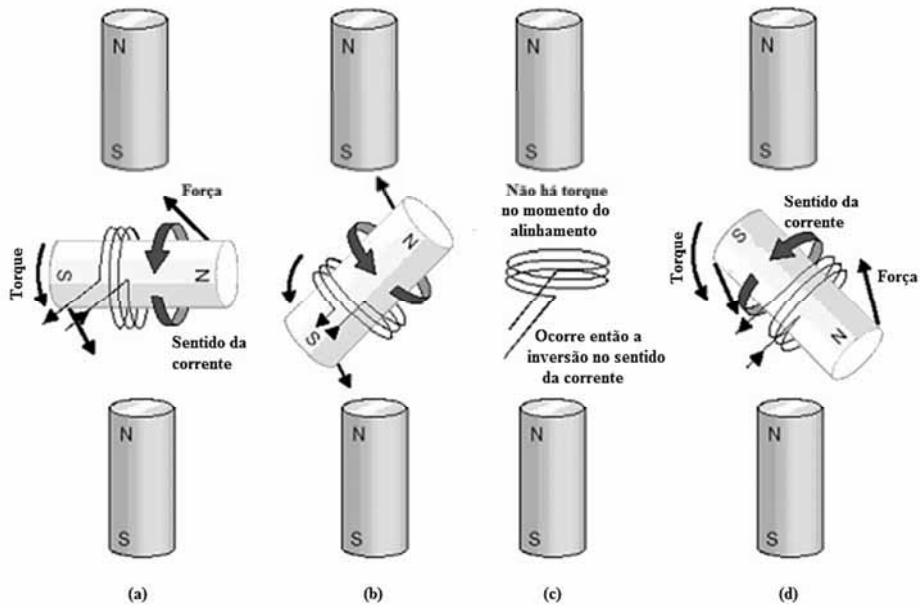
$$|F| = B \cdot i \cdot l \cdot \sin \theta \quad (3.4)$$

Sendo θ o ângulo entre os vetores corrente e campo magnético. Se o campo magnético é perpendicular a corrente ($\theta = 90^\circ$), e a força resultante age sobre uma alavanca com braço r , então o torque T desenvolvido é descrito pela equação (3.5):

$$T = F \cdot r = B \cdot l \cdot r \cdot i \quad (3.5)$$

No motor de corrente contínua, o campo magnético criado pela corrente circulante nas espiras do rotor dará origem um polo magnético norte e um polo magnético sul. Devido à lei da atração e repulsão magnética, os polos do rotor terão a tendência de se alinhar inversamente com os polos magnéticos do estator, originados da corrente nos enrolamentos do estator ou de ímãs permanentes. No momento em que estiverem alinhados não haverá mais torque desenvolvido devido às interações entre os campos magnéticos. Neste instante é que ocorre a comutação (inversão) do sentido da corrente no rotor. Assim os polos magnéticos do rotor serão invertidos e devido à inércia de movimento do rotor, ele continuará se movimentando no mesmo sentido até que os polos magnéticos se alinhem novamente. Quando isto ocorrer é realizada então novamente a inversão no sentido da corrente do rotor. Tal princípio de funcionamento é ilustrado pela Figura 3.4.

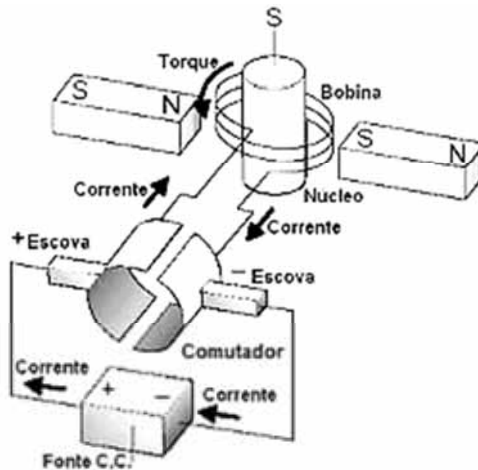
Figura 3.4 - Princípio de funcionamento do motor de corrente contínua de 2 polos



Fonte: (MODELIX, [201-?])

A comutação é o que possibilita a utilização de apenas uma fonte de corrente contínua para alimentação do motor DC. As escovas são fixamente conectadas cada uma a um terminal da fonte de alimentação. O comutador gira solidário ao eixo do motor, fazendo com que as bobinas do rotor, as quais têm seus terminais conectados às lâminas do comutador, tenham invertido o sentido de sua corrente. A Figura 3.5 mostra um motor DC de dois polos e a inversão no sentido da corrente, que ocorre neste caso a cada 180° de rotação.

Figura 3.5 - Comutação em um motor de corrente contínua de 2 polos



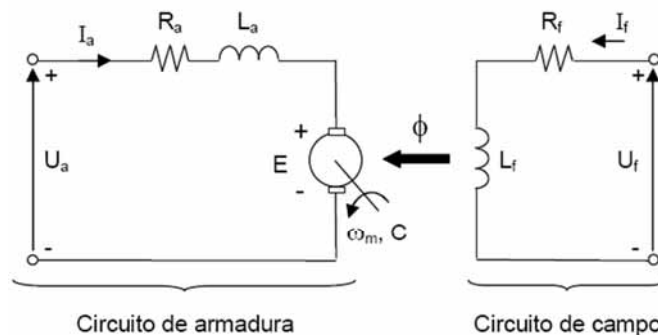
Fonte: (MODELIX, [201-?])

Sabe-se que, devido à indutância das bobinas do rotor, a inversão da corrente e a desmagnetização das bobinas não ocorrem de forma instantânea. O fato de as lâminas do comutador serem curto-circuitadas no momento da comutação ajuda na dissipação de energia acumulada na bobina. Mesmo assim, a abertura instantânea do circuito da bobina faz com que seja induzida uma força eletromotriz que tenta manter a corrente no mesmo sentido. Essa tensão gerada é um fator que limita a aplicação do motor DC em ambientes suscetíveis à explosões, pois causa centelhamento no interior do motor ao atingir altos valores e romper a rigidez dielétrica do ar. Os enrolamentos de interpolo, ou de compensação, são então conectados em série com o rotor e fixados na periferia dos polos do estator. O objetivo é fazer com que por eles circule uma corrente com polaridade oposta ao do enrolamento do rotor, resultando num cancelamento desta reação de armadura.

3.2.1 Circuito equivalente do motor DC

A Figura 3.6 ilustra o circuito equivalente de um motor de corrente contínua.

Figura 3.6 - Circuito equivalente de um motor de corrente contínua



Fonte: (TORO, 1999)

Onde: U_a – tensão de armadura; R_a – resistência de armadura; L_a – indutância de armadura; I_a – corrente de armadura; E – força eletromotriz induzida ou força contra-eletromotriz de armadura; U_f – tensão de campo; R_f – resistência de campo; L_f – indutância de campo; I_f – corrente de campo; Φ – Fluxo magnético;

Aplicando-se a lei de Kirchhoff no circuito de armadura, tem-se a equação (3.6).

$$U_a = R_a \cdot I_a + E + L_a \frac{dI_a}{dt} \quad (3.6)$$

Através da Lei de Faraday relaciona-se a força eletromotriz induzida E com a velocidade n e o fluxo magnético Φ , relação vista na equação (3.7).

$$E = k_1 \cdot \Phi \cdot n \quad (3.7)$$

Onde k_1 é uma constante que depende dos aspectos construtivos do motor. Ao combinar-se as equações (3.6) e (3.7), tem-se então a equação (3.8).

$$n = k_1 \frac{U_a - R_a \cdot I_a - L_a \frac{dI_a}{dt}}{\Phi} \quad (3.8)$$

Ao se considerar o produto da resistência R_a pela corrente de armadura I_a pequeno em relação à tensão de armadura U_a , assim como a influência da indutância L_a , pode-se dizer que a força eletromotriz induzida E é igual à tensão de armadura. Tem-se na equação (3.9) então a seguinte relação de velocidade:

$$n \cong k_1 \frac{U_a}{\Phi} \quad (3.9)$$

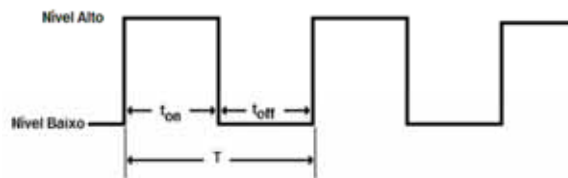
Como k_1 é uma constante intrínseca do motor, percebe-se que é possível se variar a velocidade de um motor DC através da variação da tensão de armadura ou do fluxo magnético de campo. O controle da tensão de armadura é realizado no circuito de armadura, e o controle do fluxo de campo é feito através do controle da corrente ou tensão de campo. A relação entre velocidade e tensão de armadura é linear e diretamente proporcional, e é este tipo de controle que foi aplicado neste trabalho. Diferentes tipos de conexão entre os dois circuitos são possíveis, possibilitando assim diferentes formas de excitação do motor. É possível a conexão dos circuitos em série, paralelo, alguns motores possibilitam uma composição entre série e paralelo, e também é possível excitar os dois circuitos de forma independente. Cada tipo de excitação proporciona relações diferentes entre torque, velocidade e corrente de armadura.

Como o motor DC não é uma máquina ideal, estão presentes perdas relacionadas ao atrito dos rolamentos, fluxos de dispersão, histerese, correntes parasitas, etc. É então importante ressaltar, que a linearidade demonstrada pela equação (3.9) é válida dentro de uma faixa de operação.(OGATA, 1998).

4 MODULAÇÃO POR LARGURA DE PULSO (PWM)

A técnica da modulação por largura de pulso é amplamente aplicada em circuitos eletrônicos como reguladores de tensão (fonte-chaveada), equipamento de telecomunicação, servomotores, e como no caso deste trabalho em circuitos de potência para controle de motores, sejam eles de corrente alternada ou contínua. A modulação por largura de pulso, tradução do nome em inglês *Pulse Width Modulation (PWM)*, consiste basicamente de uma onda quadrada com frequência constante e variação no ciclo de trabalho (*duty cycle*) da onda. A Figura 4.1 ilustra uma onda quadrada e seus parâmetros.

Figura 4.1 - Sinal de onda quadrada e seus parâmetros



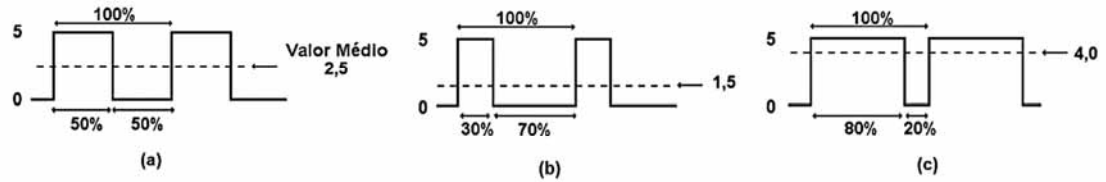
Fonte: (SILVA, 2008)

O período T , inverso da frequência f , é dado pela soma dos tempos em que o sinal permanece em nível alto t_{on} e em nível baixo t_{off} , como na equação (4.1).

$$T = t_{on} + t_{off} = \frac{1}{f} \quad (4.1)$$

Através da equação (4.1), pode-se concluir que se a soma dos tempos em nível alto e em nível baixo permanecer constante, então o período e consequentemente a frequência permanecerão constantes. A razão entre o tempo em nível alto t_{on} pelo período T é chamada de ciclo de trabalho, ou do inglês *duty cycle*, normalmente representado em porcentagem. Ao analisar a área do retângulo formado acima do nível baixo para um período, vê-se que esta razão é a razão do valor da amplitude em nível alto que deve ser somado ao valor em nível baixo para se obter o valor médio. Normalmente em circuitos eletrônicos o valor de nível baixo é zero, então pode-se dizer que a amplitude em nível alto multiplicada pelo ciclo de trabalho é o próprio valor médio, como vemos na Figura 4.2.

Figura 4.2 - Variação do ciclo de trabalho (a) duty cycle = 50% (b) 30% (c) 80%

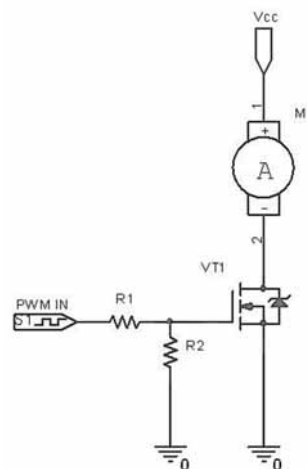


Fonte: (SILVA, 2008)

Se esta onda for aplicada como tensão de armadura em um motor de corrente contínua, através do controle do ciclo de trabalho e conseqüentemente do valor médio, pode-se controlar a velocidade do motor facilmente, pois dentro de uma ampla faixa de operação linear a velocidade é diretamente proporcional à tensão de armadura. Este é o mesmo princípio aplicado em fontes-chaveadas, onde é variado o ciclo de trabalho do sinal aplicado no controle de um tiristor, a fim de manter um valor constante de tensão na saída do regulador mesmo com a variação da carga ou da tensão de entrada do regulador. O valor do ciclo de trabalho também pode ser utilizado para carregar uma informação, como no caso de circuitos de telecomunicações ou de servomotores comumente utilizados em aerodelismo, onde a largura do pulso positivo representa a posição desejada do servomotor.

A Figura 4.3 mostra o ajuste de velocidade através da técnica PWM em um motor de corrente contínua.

Figura 4.3 - Ajuste de velocidade de um motor de corrente contínua pela técnica PWM



Fonte: (GHIRARDELLO, 2008)

O ciclo de trabalho do motor expresso em porcentagem, ou *duty cycle*, é dado por (4.2):

$$d = \frac{t_{on}}{T} \cdot 100\% \quad (4.2)$$

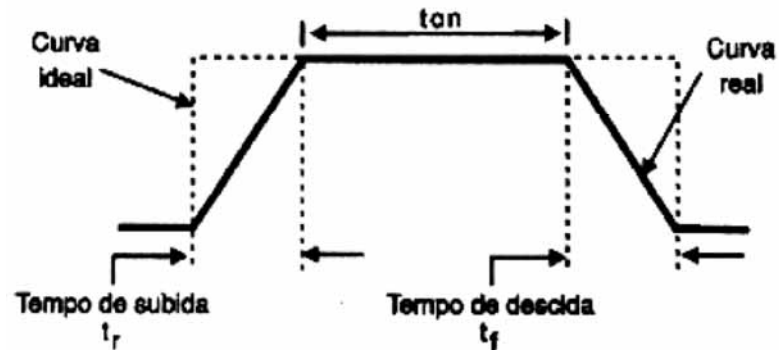
O MOSFET VT1 é responsável pela abertura e fechamento do circuito do motor M1 conforme o sinal PWM aplicado em seu gatilho. No nível alto do sinal há corrente fluindo pelo motor, e no nível baixo não há. Se o ciclo de trabalho aplicado for de 100%, então teremos velocidade $n_{m\acute{a}x}$ no motor. A equação (4.3) mostra a tensão aplicada à armadura do motor U_a e a equação (4.4) a variação da velocidade em relação ao ciclo de trabalho

$$U_a = V_{CC} \cdot \frac{d}{100} = V_{CC} \cdot \frac{t_{on}}{T} \quad (4.3)$$

$$n = n_{m\acute{a}x} \cdot \frac{d}{100} = n_{m\acute{a}x} \cdot \frac{t_{on}}{T} \quad (4.4)$$

É importante ressaltar neste caso, que o motor sendo basicamente uma indutância, a frequente interrupção instantânea do circuito provocará o surgimento de uma tensão reversa a de alimentação de valor elevado aplicada no MOSFET no momento da abertura do circuito. Esta tensão pode danificar o MOSFET, e por esta razão há um diodo antiparalelo conectado a ele. Este diodo, conhecido como volante, tem por finalidade conduzir a corrente reversa que surge devido à reação do motor no momento da abertura do circuito. No momento da escolha do MOSFET, ou outro tiristor de potência a ser utilizado, devem ser respeitadas as características impostas pelo motor como tensão e corrente de alimentação. A frequência utilizada para o chaveamento deve ser determinada de acordo com as características do motor e limitações do MOSFET como, tempo de ligamento e desligamento, máxima frequência de operação. Deve-se atentar para o fato de que por menor que sejam os tempos de comutação do dispositivo de potência, ela não é instantânea. Mesmo que se considere a dissipação de potência no dispositivo nula durante os estados de corte e saturação, no momento da comutação entre estados, existe dissipação. Na comutação de corte para saturação ao passo que a corrente sobe de zero ao valor em regime, a tensão cai do valor de alimentação da carga a próximo de zero. E na comutação da saturação ao corte ocorre o oposto. Durante este tempo de comutação têm-se então que o produto tensão por corrente no elemento de chaveamento não é nulo. A Figura 4.4 ilustra os tempos de subida e descida, em comparação com a curva real de controle aplicada ao dispositivo.

Figura 4.4 – Causa da dissipação de potência na comutação de tiristores



Fonte: (SILVA, 2008)

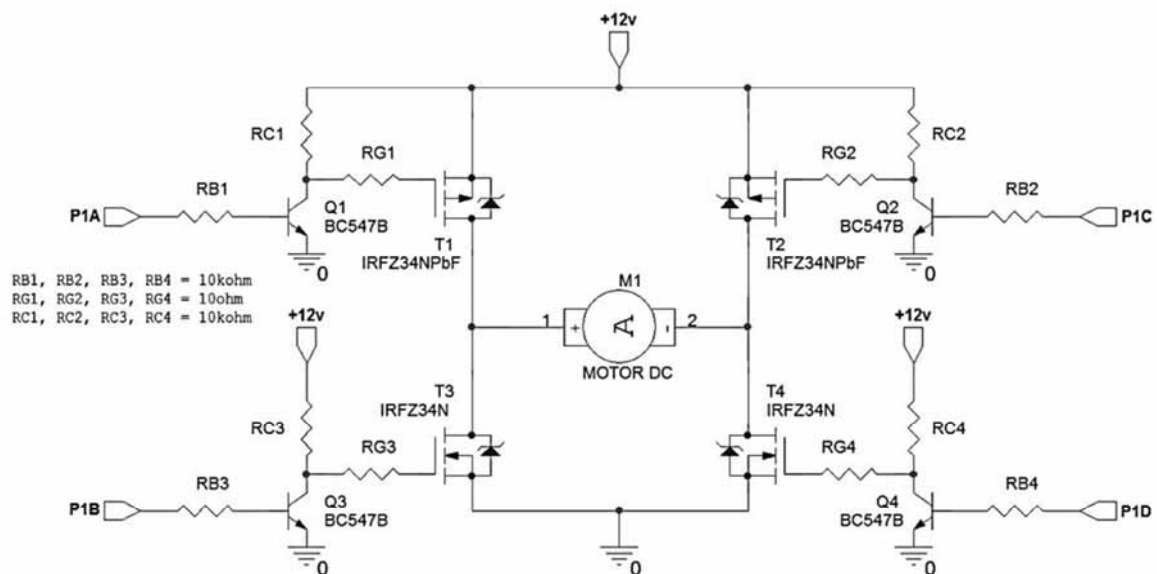
Outro fator a ser considerado na escolha da frequência de chaveamento do motor é a possível ressonância mecânica, que pode causar excessivas vibrações e rápida deterioração do conjunto mecânico. A frequência deve ser tal que considerando os possíveis arranjos de carga, haja garantia da não ocorrência de ressonância. Se a frequência utilizada estiver dentro da faixa audível, podem também surgir ruídos desagradáveis e indesejados provenientes do chaveamento.

5 DRIVER DE PONTE COMPLETA (PONTE H)

O motor de corrente contínua tem como característica a simplicidade na inversão do sentido de rotação de seu eixo. Tal inversão é alcançada ao se inverter a corrente de um dos circuitos de excitação do motor, seja o de campo ou de armadura. Invertido o sentido da corrente de um dos circuitos, é alterado o sentido da força resultante da interação dos campos magnéticos. Como o controle de velocidade neste trabalho está baseado no controle da tensão de armadura, a inversão de sentido será dada com a inversão da corrente no circuito de armadura. Se o motor utilizado for de ímã permanente, onde o campo do estator é constante em módulo e polaridade, este método de controle continua válido.

Muitas aplicações tem sentido de rotação fixo, neste caso sendo necessária a inversão do sentido de rotação para outro sentido que permanecerá fixo, o método mais simples seria a inversão dos terminais de alimentação do circuito de armadura. Em processos onde a mudança de sentido deve ocorrer de forma dinâmica, com o processo em andamento, é necessário que um circuito eletrônico ou eletromecânico realize esta inversão no sentido da corrente de forma automática, atuando através dos sinais de controle. Tal circuito é conhecido como *Driver de Ponte Completa*, do inglês *Full-Bridge*. Este circuito é também conhecido como *Ponte H*, devido ao seu formato o qual se assemelha ao de uma letra H. A Figura 5.1 ilustra o circuito *Driver de Ponte Completa* utilizado neste trabalho.

Figura 5.1 - *Driver de Ponte Completa (Ponte H)*



Fonte: (produção do próprio autor)

O MOSFET foi o elemento de chaveamento escolhido para controle da corrente de armadura do motor de corrente contínua, pois agrega características como baixa resistência de canal, alta frequência de chaveamento e acionamento por tensão (alta impedância de entrada). A baixa resistência de canal implica que quando em estado de condução (saturação), a tensão entre Dreno e Fonte (*Drain* e *Source*) tenha um valor pequeno, ou seja, praticamente não há perda de tensão nos tiristores e o motor recebe tensão de alimentação plena. Diz-se que o MOSFET é acionado por tensão, pois devido suas características construtivas tem alta impedância de entrada na Porta (*Gate*). Tal fato facilita o controle entre seus estados, corte e saturação, e é vantajoso também, pois com corrente de *gate* praticamente inexistente, não há perda por corrente quiescente. Juntas, a baixa resistência de canal, a alta impedância de entrada e a alta frequência de chaveamento, que é traduzida como uma rápida comutação entre os estados de corte e saturação, fazem com que a perda de potência no MOSFET seja muito pequena e o tornam ideal para o chaveamento.

O motor a ser utilizado neste trabalho é de pequena dimensão, e os MOSFETs comercialmente encontrados de menor capacidade de condução de corrente superam a necessidade da carga. Por este motivo foram adotados como MOSFET Canal N o HEXFET da International Rectifier IRFZ34N, e seu equivalente Canal P o IRFZ34NPbF. As características de funcionamento de ambos encontram-se nos Anexos A e B. O transistor NPN BC547 é polarizado de forma a operar também nas regiões de corte e saturação. Sua principal função é a de isolar a saída digital do microcontrolador da porta do MOSFET e utilizar a tensão de 12V que alimenta o motor, maior do que a de 5V digital, como a tensão de *Gate-Source* aplicada, garantindo assim menor resistência de canal em operação. Os MOSFETs utilizados possuem um diodo volante intrínseco, com a função de evitar que o pico de tensão reversa no momento da abertura do circuito de armadura danifique os MOSFETs. Se o elemento de chaveamento não possui tal diodo volante intrínseco, é necessário que este seja adicionado externamente. Como a frequência de chaveamento utilizada é normalmente alta, recomenda-se o uso de diodos rápidos, como o 1N4148.

Os MOSFETs T1 e T4 atuam em conjunto e fazem a corrente de armadura circular em um sentido, e T2 e T3 no sentido oposto, tornando assim possível a inversão no sentido de rotação. O controle deve garantir que T1 e T3, assim como T2 e T4, nunca entrem em condução ao mesmo tempo, principalmente no momento da inversão do sentido de rotação, onde deve ser respeitado o chamado tempo morto (*dead-time*). Isto seria equivalente a um

curto-circuito da alimentação ao terra através dos canais dos MOSFETs, o que pode destruí-los ou danificar a alimentação. A fim de controlar a velocidade do motor, os elementos de chaveamento T3 e T4 são acionados por um sinal de onda quadrada modulada por largura de pulso, ou seja, utilizando a técnica do PWM. Como os elementos de chaveamento trabalham em pares e estão em série, o chaveamento de apenas um componente do par basta. Assim, elimina-se a discordância de tempos de desligamento e ligamento caso o par fosse chaveado em conjunto, e os MOSFETs Canal P (T1 e T2) não têm a necessidade de tempos de comutação tão baixos quanto os de Canal N. O controle das saídas digitais que acionam os elementos de chaveamento é feito pelo microcontrolador e é explanado posteriormente neste trabalho no Capítulo 6.

6 MICROCONTROLADOR PIC

A empresa *Microchip Technologies Inc* é a fabricante dos microcontroladores PIC. Dentro da família PIC existem microcontroladores de 8 até 32 bits. As principais características a serem observadas no momento da escolha de um microcontrolador são: memória de programa (ROM), memória RAM, frequência de operação e quantidade de pinos de entrada e saída digitais (I/O). Além disso, uma grande vantagem dos microcontroladores PIC é que podem agregar aplicações específicas dentro do próprio microcontrolador, o que facilita no momento da implementação do *software* e *hardware*, pois são reduzidos custos com circuitaria externa, barramentos e também de horas de engenharia na elaboração de algoritmos.

Na Figura 2.16 foi ilustrado o diagrama de blocos do sistema a ser controlado pelo microcontrolador PIC. A necessidade de conversores AD, para conversão dos sinais analógicos dos potenciômetros para um sinal digital, fica explícita em tal diagrama. Conversores AD estão disponíveis em vários modelos diferentes de PIC, portanto é interessante que o microcontrolador escolhido possua tal conversor. Com os conversores AD internos ao microcontrolador, elimina-se a necessidade de mais um CI e também a necessidade de barramentos de comunicação entre o conversor AD e o microcontrolador. É facilitada também a maneira de se comandar o início e leitura de uma conversão, pois sem a necessidade de acesso aos barramentos externos, tais comandos são simples instruções internas do microcontrolador.

Deseja-se realizar o controle de velocidade e sentido de rotação do motor através da técnica de modulação por largura de pulso (PWM) aplicada a um *driver* de ponte completa, apresentado na Figura 5.1. Percebe-se que, para isso são necessárias quatro saídas digitais do microcontrolador a fim de controlar o disparo dos transistores. Alguns modelos de PIC possuem a opção de PWM embarcada controlada por *hardware*. Esta é uma opção muito interessante, pois elimina a quantidade de processamento exigida do microcontrolador, pelo fato de que a mudança no valor do ciclo de trabalho e/ou frequência é dada por uma simples escrita nos registradores responsáveis pelo controle do módulo PWM. A geração do sinal PWM fica então por conta de um temporizador independente da unidade lógica, o que significa que o sinal PWM estará sempre ativo deixando o microcontrolador livre para processar outras instruções. Esta é uma grande vantagem já que nesta aplicação deseja-se

eliminar a necessidade de osciladores externos, o que implica em trabalhar com o oscilador interno de menor frequência de operação. A diminuição da carga de processamento é crucial, já que o processamento do controlador PID demanda razoável número de cálculos matemáticos, os quais demandam um considerável número de instruções a serem realizados pelo PIC.

Fisicamente sabe-se que será necessário que o microcontrolador possua no mínimo oito pinos, sendo dois para alimentação, duas entradas analógicas e quatro saídas digitais. Dos microcontroladores PIC de oito bits que possuem quatro canais de saída PWM controlados por *hardware*, existem os modelos PIC16F616 e PIC16F684. Ambos são muito semelhantes, possuindo as características desejadas como oscilador interno até 8MHz, módulo *ECCP* (*Enhanced Capture/Compare/PWM* – ou Captura/Comparação/PWM Embarcado) e oito canais de conversão analógica para digital com resolução de 10bits. Como vantagem o PIC16F616 tem um regulador de tensão interno, podendo ser alimentado com tensão de até 15V, e saída do comparador com *Latch SR* (*Set Reset*), que para esta aplicação não é necessário, já que o módulo *ECCP* será utilizado na função PWM. Por sua vez o PIC16F684 possui a vantagem de ter 256bytes de memória *EEPROM* disponíveis. Ambos satisfazem a quantidade de memória de programa e dados necessária para construção do controlador PID. O encapsulamento utilizado é o 14/PDIP (*14 pin Plastic Dual In-Line Package*), que possibilita a montagem em matriz de contatos, já que se trata de um projeto em desenvolvimento. No mercado local o PIC16F684 foi encontrado com mais facilidade e com menor custo, portanto este foi o modelo utilizado neste trabalho. A pinagem, o diagrama de blocos, principais características elétricas, entre outros fatores relevantes de operação e configuração do PIC16F684 utilizados ao longo deste trabalho, estão presentes no Anexo C.

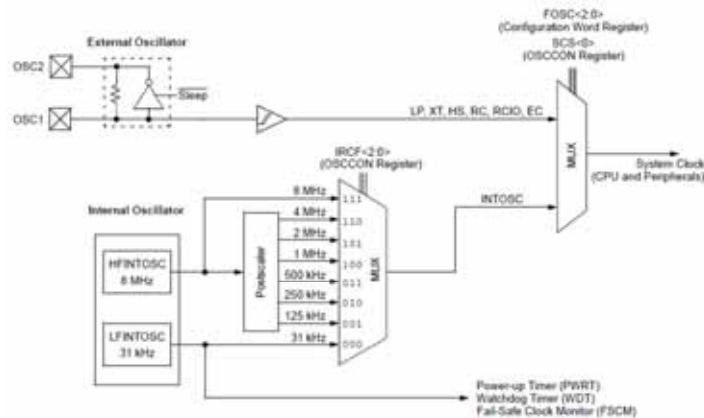
Para desenvolvimento do programa em linguagem C foi utilizado compilador *CCS PIC® C Compiler PCW Version 4.078* da fabricante *CCS (Custom Computer Services, Inc)*. A gravação do programa no microcontrolador PIC foi realizado utilizando-se a ferramenta *PICkit2* em conjunto com o *software MPLAD IDE v8.86* da fabricante *Microchip Technologies Inc*. O ambiente de programação e gravação, o circuito montado em matriz de contatos, bem como o gravador utilizado são demonstrados no Anexo D.

Os principais componentes a serem observados com relação às características de operação (range de tensão, tempo de resposta, entre outros) são o oscilador interno, o conversor AD e o módulo PWM embarcado. A fim de reduzir custos de implementação foi

utilizado o oscilador interno de 8MHz do PIC (HFINTOSC – *High Frequency Internal Oscillator*). Este é controlado pelos registradores: OSSCON (*OSCILLATOR CONTROL REGISTER*), que determina a origem do *clock* a ser utilizado, e pelo registrador OSCTUNE (*OSCILLATOR TUNING REGISTER*), o qual é utilizado para ajuste fino do oscilador interno.

O diagrama de blocos para a seleção da fonte de *clock* é mostrado na Figura 6.1:

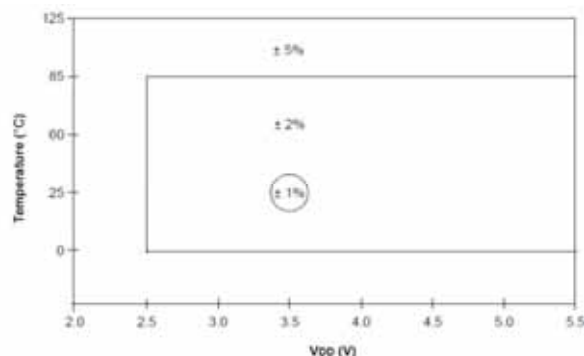
Figura 6.1 – Diagrama de blocos para seleção da fonte de *clock*



Fonte: (MICROCHIP, 2007)

A precisão do oscilador interno de alta frequência é dada pelo gráfico demonstrado na Figura 6.2. Onde V_{DD} é a tensão de alimentação do microcontrolador.

Figura 6.2 – Precisão do oscilador interno de alta frequência

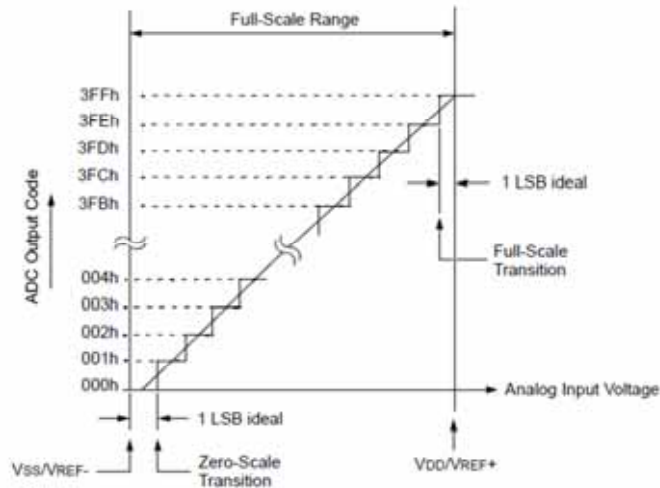


Fonte: (MICROCHIP, 2007)

A tensão de alimentação utilizada neste trabalho para o microcontrolador foi de 5V, Percebe-se então pela Figura 6.2, que tem-se uma precisão de $\pm 2\%$ na frequência do oscilador interno. Se no momento da aplicação for percebida variação permanente, é possível se fazer o ajuste do valor através do registrador OSCTUNE.

O conversor AD interno possui resolução de 10 bits e 8 canais de entrada. Sua função de transferência é apresentada na Figura 6.3.

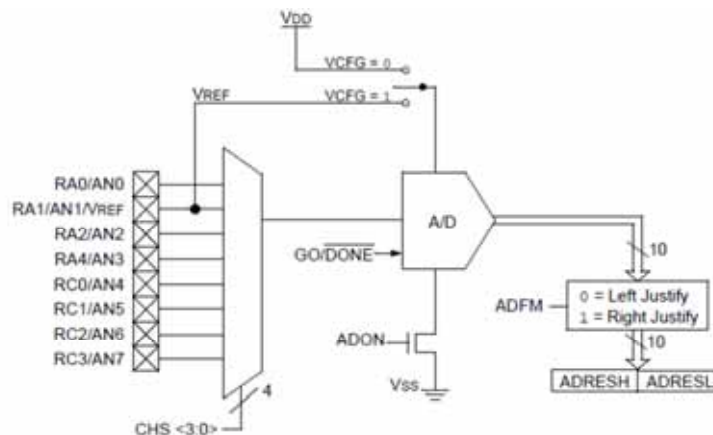
Figura 6.3 – Função de transferência do conversor AD



Fonte: (MICROCHIP, 2007)

A Figura 6.4 apresenta o diagrama de blocos para seleção do canal de entrada.

Figura 6.4 – Diagrama de blocos da seleção canal de entrada conversor AD



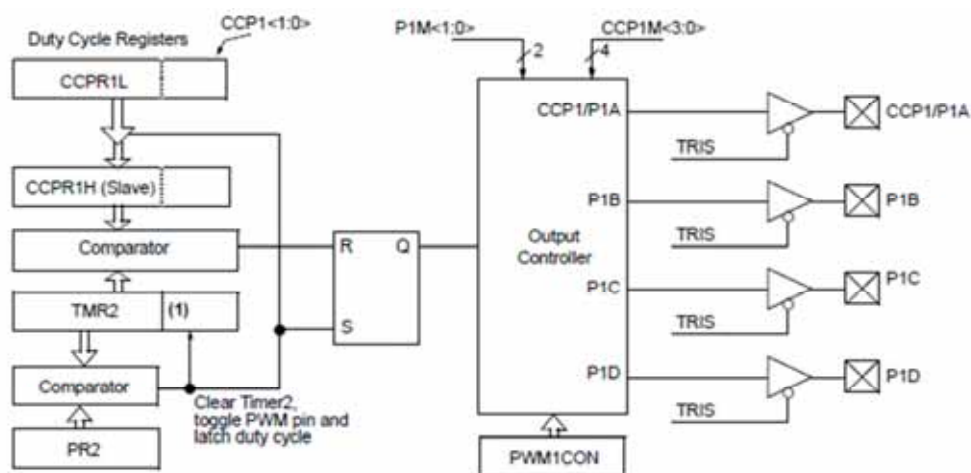
Fonte: (MICROCHIP, 2007)

Os registradores responsáveis pelo controle do conversor AD são o ADCON0 e ADCON1 (*AD CONTROL REGISTER 0/1*). Ambos possuem bits para configuração do canal de entrada, valor do *clock* de conversão AD, tensão de referência *Vref*, e bits de controle para começo e leitura das conversões. Ao fim de cada conversão o resultado é apresentado nos registradores ADRESH e ADRESL (*ADC RESULT REGISTER HIGH/LOW*). Para facilitar

nos cálculos realizados com o valor das conversões, o resultado é justificado a direita, ou seja, o bit menos significativo é o bit 0 (ADFM = 1 no registrador ADCON0). Outro fator a ser observado é que a cada troca de canal de entrada, deve se esperar dois períodos de *clock* do conversor AD para que se inicie outra conversão. Outras características relevantes como tempos mínimos de conversão, modelo interno, entre outros aspectos importantes do PIC16F684 se encontram no Anexo C.

O módulo embarcado PWM (ECCP) utiliza o Temporizador 2 (TIMER2) como base de *clock* e como contador para gerar uma onda quadrada de frequência constante modulada por largura de pulso. O registrador que controla o módulo ECCP é o CCP1CON (*ENHANCED CCP1 CONTROL REGISTER*). Os registradores que controlam o ciclo de trabalho, período e a resolução do PWM são os seguintes: PR2, T2CON, CCPR1L, CCP1CON. O diagrama de blocos do módulo ECCP configurado como PWM é mostrado na Figura 6.5.

Figura 6.5 – Diagramas de blocos do módulo ECCP configurado como PWM

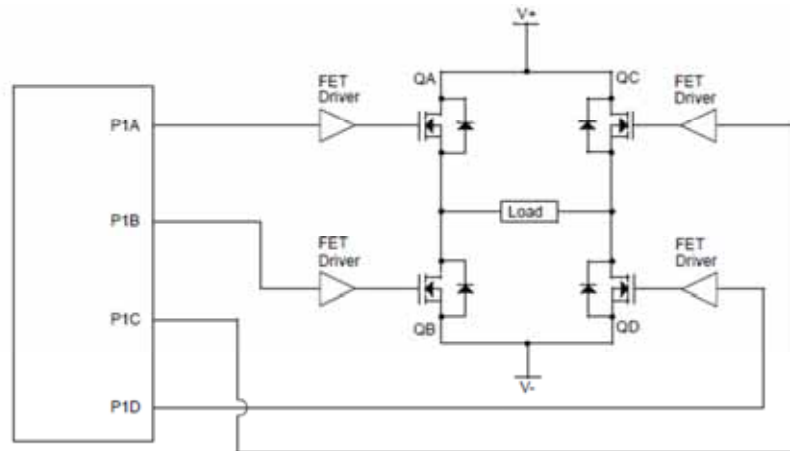


Note 1: The 6-bit timer TMR2 register is concatenated with the 2-bit internal Q clock, or 2 bits of the prescaler to create the 10-bit time base.

Fonte: (MICROCHIP, 2007)

A conexão das saídas no *Driver* de Ponte Completa mostrado na Figura 5.1, é apresentada na Figura 6.6.

Figura 6.6 – Conexão das saídas do PWM ao *Driver* de Ponte Completa

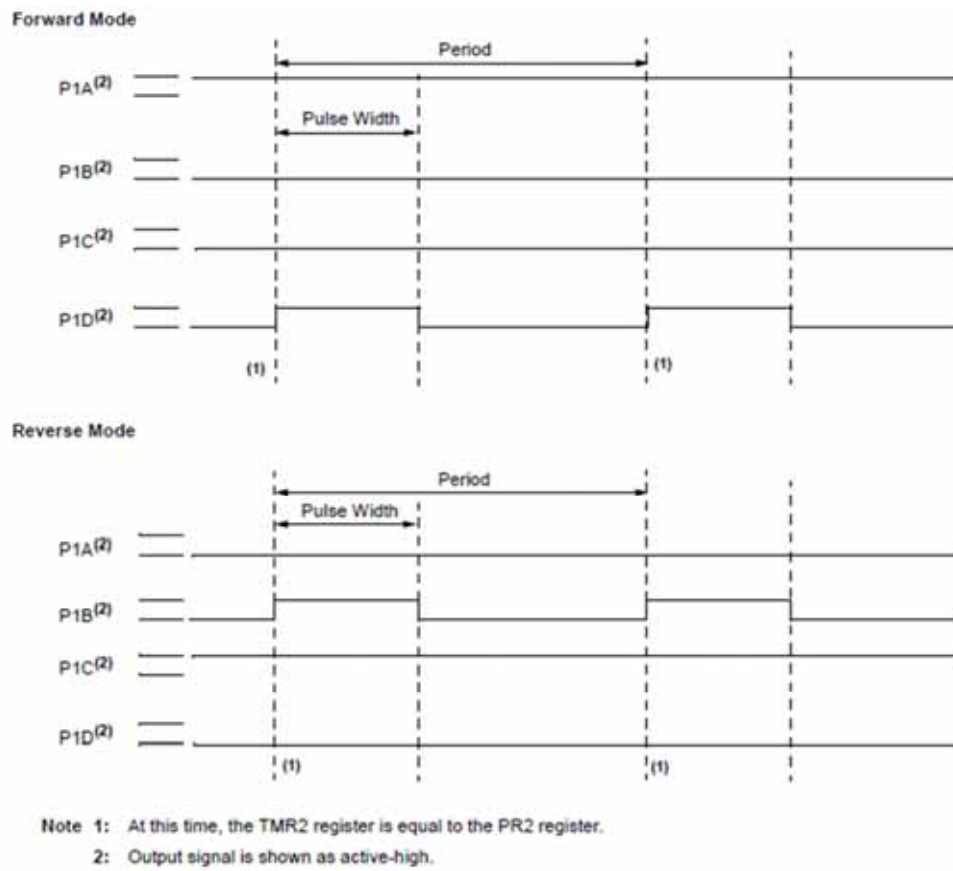


Fonte: (MICROCHIP, 2007)

As características adicionais mais relevantes para construção do controlador estão descritas no Anexo C. Para informações mais detalhadas, como descrição dos registradores, modelos internos, etc, vide manual PIC16F684.

Na Figura 6.7 vê-se que, as saídas P1B e P1D são demonstradas como ativo alto. Entretanto, devido à configuração observada na Figura 5.1, vê-se que na realidade elas devem ser ativo baixo, ou seja, os MOSFETs conectados ao terra conduzem quando P1B e P1D tem valor baixo, devido à inversão dada pelos transistores, que atuam como *buffer*, conectados aos seus respectivos *gates*. Isto garante que o ciclo de trabalho não será o complemento de 100% e que não ocorrerá condução direta pelo canal dos MOSFETs. Este ajuste é feito pelos bits CCP1M1 e CCP1M0 do registrador CCP1CON. O tempo morto (*dead-time*), que deve ser respeitado no momento da inversão do sentido de rotação, é indiretamente respeitado pelo fato do processamento do programa, posteriormente apresentado no Capítulo 7, ser maior que o tempo mínimo necessário. Conclui-se que o tempo que o microcontrolador leva para completar a situação de inversão de sentido é maior que o *dead-time* mínimo a ser respeitado, caso assim não fosse, este deveria ser implementado na programação. A Figura 6.7 ilustra as formas de onda aplicadas ao *Driver* de Ponte Completa com a ressalva dos níveis de ativação anteriormente citados:

Figura 6.7 – Formas de onda de saída do módulo PWM



Fonte: (MICROCHIP, 2007)

7 APLICAÇÃO E RESULTADOS PRÁTICOS

O programa final escrito em linguagem de programação C encontra-se no Apêndice A, e o diagrama esquemático utilizado para verificação do conjunto é apresentado no Apêndice B. O eixo do motor DC é acoplado ao eixo do potenciômetro multivoltas através de um par de engrenagens, tendo a engrenagem solidária ao eixo do motor 15 dentes e a solidária ao eixo do potenciômetro 56 dentes. Cada volta completa do eixo do motor (360°) representa um deslocamento angular de $96^\circ 25' 43''$ no eixo do potenciômetro. Tendo o eixo do potenciômetro uma engrenagem com maior número de dentes (ou maior diâmetro), o ajuste se torna mais sensível, pois um deslocamento angular no eixo do motor causa menor deslocamento no eixo do potenciômetro. Considerando que a carga a ser posicionada é solidária ao eixo do potenciômetro, têm-se a desvantagem de uma menor velocidade angular máxima em relação ao motor. A escolha da relação de deslocamento angular depende dos fatores citados anteriormente, como velocidade angular máxima e sensibilidade de ajuste, bem como da faixa de torque do motor e da carga. Neste trabalho nenhuma carga foi adicionada ao conjunto motor/potenciômetro. Deve-se atentar para o fato que a variação de carga altera as características dinâmicas do conjunto. Portanto a resposta do conjunto será alterada, e para cada variação de carga é necessário reajuste dos ganhos do controlador PID.

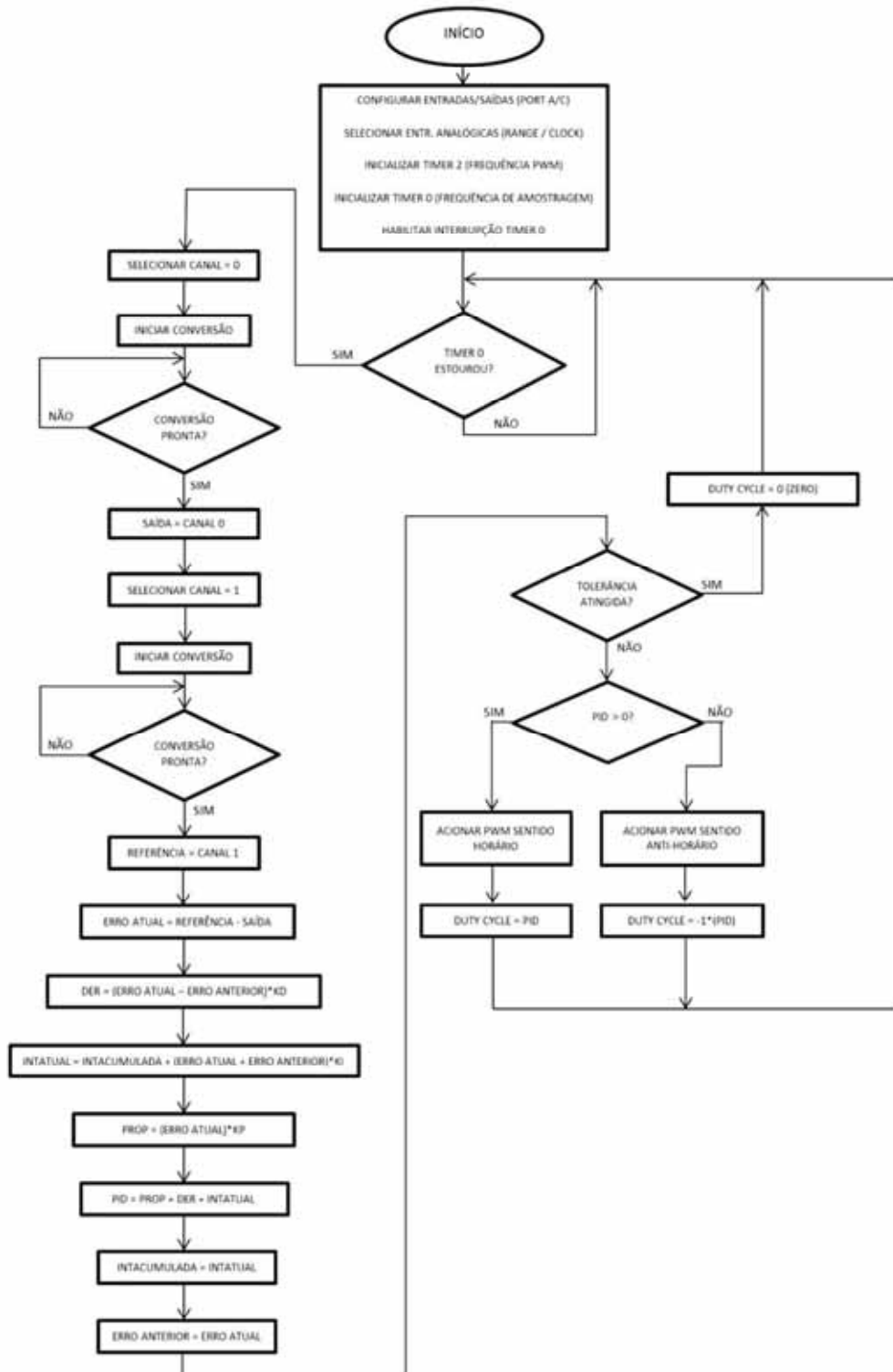
7.1 DESENVOLVIMENTO DO PROGRAMA

Os principais aspectos observados durante o desenvolvimento do programa foram: eficiência do código, tempo mínimo para execução da rotina, limites e restrições das variáveis inteiras e facilidade no ajuste dos ganhos.

A eficiência do código se trata de identificar a ação que deve ser realizada, e realizá-la com sucesso utilizando minimamente os recursos do microcontrolador. Estes recursos podem ser resumidos ao número de instruções executadas e quantidade de memória utilizada. O menor número de instruções torna a execução do programa mais rápida e permite que mais ações sejam realizadas dentro de um mesmo período. Este fator é limitante principalmente no caso da taxa de amostragem do controlador PID. Quanto mais rápido for o cálculo das

equações, maior é a taxa de amostragem possível de ser utilizada. A Figura 7.1 ilustra o fluxograma geral do programa:

Figura 7.1 – Fluxograma geral do programa



Fonte: (produção do próprio autor)

Pode-se observar que o programa consiste na inicialização de alguns registradores, que configuram os pinos de entrada e saída, canais analógicos, frequência e range do conversor AD, taxa de amostragem, etc. Uma vez realizada esta inicialização, o microcontrolador passa a aguardar o estouro do TIMER 0, que representa a taxa de amostragem. Sendo ajustado como uma divisão da frequência do oscilador interno do microcontrolador, para esta aplicação o estouro ocorre a cada 4ms, que representa frequência de amostragem de 250Hz. Esta verificação de estouro é na realidade realizada através de uma interrupção do microcontrolador, a interrupção de TIMER 0, que deve ser habilitada na inicialização do programa. Uma vez habilitada a interrupção, é realizada a leitura dos canais analógicos do microcontrolador, sendo que a cada seleção de canal deve se ter um cuidado de aguardar 2 ciclos de *clock* do microcontrolador para iniciar a conversão. Com o valor das amostras no intervalo disponíveis, é então implementada a equação algébrica aproximada do controlador PID demonstrada pela equação (2.20). Cuidados com a implementação de tais equações devido ao tipo de variável utilizada, bem como outras particularidades do programa, são ressaltadas posteriormente neste trabalho e detalhadas nos comentários do programa apresentado no Apêndice A. Com o valor da saída calculado, é verificado se o sistema já atingiu a tolerância especificada, que corta de vez a atuação do motor (ciclo de trabalho = 0). Caso não tenha atingido tal tolerância, o motor é acionado no sentido correto conforme a saída do controlador. Então o microcontrolador volta a aguardar o estouro do TIMER 0, e assim a rotina se repete.

O compilador utilizado tem como ferramenta, a geração do código *Assembly* gerado correspondente ao código C escrito pelo programador. Utilizando tal ferramenta, foi observado durante o desenvolvimento do programa, qual opção de código em linguagem de programação C gera o código em *Assembly* com menor uso de instruções do microcontrolador. Isto se traduz como uma execução mais rápida do programa e uma menor utilização de memória. Utilizando-se menor quantidade de memória é possível que mais funções sejam utilizadas dentro de um mesmo programa, considerando que a memória do microcontrolador é limitada. O programa apresentado requer, segundo o compilador utilizado, 33% da memória ROM e no pior caso 48% da memória RAM do microcontrolador. Tendo definida a rotina de leitura da entradas, cálculo PID e atualização das saídas, foi verificado que a maior frequência de amostragem possível é de 1kHz.

Os limites e restrições das variáveis inteiras tiveram de ser observados e compensados na programação, pois durante o cálculo da rotina PID a utilização de variáveis do tipo *float* (ponto flutuante de 32 bits) tornou a execução da mesma muito lenta. E pode-se dizer que também seria injustificado uso de tal tipo de variável, já que o conversor AD com resolução de 10 bits fornece valores do tipo *int16*, e o valor do ciclo de trabalho do PWM, que irá atuar no motor, também é um valor inteiro de amplitude máxima igual ao do conversor AD (0 a 1023). A compensação citada é necessária devido ao fato de que com a frequência de amostragem escolhida, o valor da componente integral tornava-se muito grande com as sucessivas somas realizadas a cada iteração. Assim os limites de -32768 a +32767 (variáveis tipo *int16*) eram rapidamente atingidos e a atuação da componente integral era exagerada sobre a saída. Deste modo é realizada uma divisão por mil, dos valores da componente integral. Pelo fato de serem variáveis inteiras foi necessário o cuidado de se acumular o resto de tal divisão, pois como a integral é uma soma contínua, o erro acumulado do resto causa erro no cálculo da integral. Valores pequenos do erro eram considerados zero após a divisão, e assim a principal vantagem do controlador integral, de eliminar o erro de estado estacionário, não era atingida. O trecho do programa que representa a equação (2.18) – cálculo da componente de saída integral ou regra do trapézio - é mostrado na Figura 7.2:

Figura 7.2 – Implementação em linguagem C da regra do trapézio (integral)

```

127:   ilr = (e1*e0);           //Cálc.da comp.int. ui[t]-ui[t-1]-> Ki*T*(e[t-1]+e[t])/2
128:   il = ilr/1000;          //Ajuste para tornar a componente integral menos sensível.
129:   ilr = ilr - (il*1000);   //il é int16, resto da divisão em milésimos é alocado em ilr.
130:   ilr = (ilr*ki) + i0r;    //Milésimos acumulados das amost. ant. (já mult. pelo ganho)
131:                               //somados ao resto da amostra atual multiplicado pelo ganho.
132:   iaux = ilr/1000;        //iaux p/ verificar se os milésimos multiplicados pelo ganho
133:                               //geraram uma ou mais unidades inteiras.
134:   i0r = ilr - (iaux*1000); //Se sim, estas são eliminadas dos milésimos acumulados e
135:                               //somadas ao valor da saída atual.
136:   il = (il*ki) + iaux;     //Soma da parte inteira + (milésimos acumulados/1000).
137:   i0 = i0 + il;           //Soma das comp. int. acumulada + período(atual)
138:                               //ui[t]=ui[t-1]+Ki*T*(e[t-1]+e[t])/2
139:                               //Próx. iter. i0 será o valor acumulado da integral -> ui[t-1]

```

Fonte: (produção do próprio autor)

É interessante observar que o fato de a faixa de operação do conversor AD estar compreendida entre 0 e 5V, auxiliou na normalização dos valores entre o conversor AD, saída do potenciômetro e cálculo do ciclo de trabalho do PWM. O potenciômetro sendo alimentado por uma tensão máxima de 5V, tem a faixa de atuação completamente coincidente com a do conversor AD. Tal conversor tendo resolução de 10 bits, representa a faixa de 0 a 5V com números inteiro de 0 a 1023, que a mesma faixa do ciclo de trabalho do módulo PWM (0% = 0 e 100% = 1023).

Outras duas funções foram adicionadas ao programa, sendo elas a criação de uma faixa de tolerância (*Dead-Band*) e limites do valor de referência. A faixa de tolerância foi introduzida a fim de reduzir atuação desnecessária do controlador sobre a saída quando a mesma já se encontra dentro da faixa de tolerância. Por se tratar de um controlador PID, mesmo depois que o valor de referência é atingido pela saída, ocorrem pequenas oscilações em torno do valor alvo. Com a introdução da tolerância, o motor é praticamente desligado ao entrar dentro desta faixa. Deste modo são eliminadas vibrações desnecessárias, que desgastam o conjunto mecânico, os componentes eletrônicos e aumentavam o consumo de energia. A limitação do valor da referência (entrada) é necessária principalmente na fase de ajuste/sintonia do controlador PID, pois o sensor na saída é um potenciômetro limitado a dez voltas. Se o valor da saída ultrapassar o valor de referência e este valor for igual ou estiver muito próximo dos limites mecânicos do potenciômetro, o fim de curso pode ser atingido abruptamente causando dano ao potenciômetro.

7.2 VERIFICAÇÃO DE FUNCIONAMENTO E AJUSTE DOS GANHOS

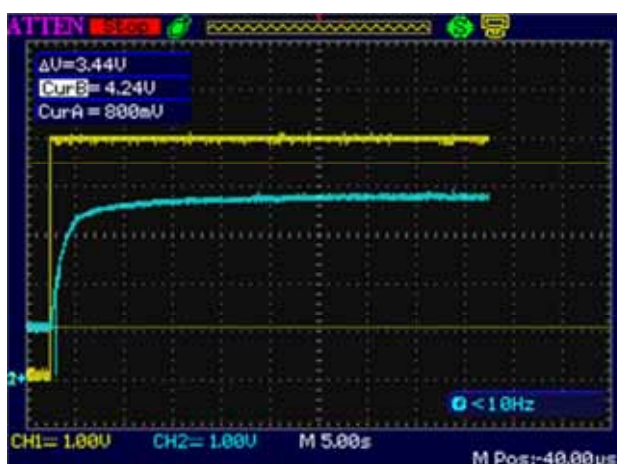
Nas linhas de número 07 a 20 do programa apresentado no Apêndice A são declarados os valores dos ganhos do controlador PID bem como da faixa de tolerância. Para se alterar o valor de um ganho específico, basta alterar o valor declarado e então compilar o programa e gravar no PIC. Deste modo foram realizadas diversas medidas da resposta do sistema a uma excitação por degrau, que é um tipo esperado de sinal de entrada. Este também quando comparado com outros tipos de sinal de mesma amplitude é o que pode causar maior ultrapassagem do valor de referência, pois apresenta a maior derivada.

A fim de ser verificar a resposta do sistema conforme a variação dos ganhos do controlador, diferentes combinações de ganhos foram aplicadas. Os valores da referência e saída do sistema foram mensurados em forma de tensão e registrados por um osciloscópio digital. Nas Figuras de 7.3 a 7.11 as curvas em amarelo representam os sinais de entrada, que por software é limitado em um valor correspondente a 0,8V de limite inferior e 4,24V de limite superior. Tais limites são representados pelas linhas pontilhadas amarelas. Então para o controlador valores de referência de entrada superiores a 4,24V são lidos como 4,24V e valores inferiores a 0,8V são lidos como 0,8V. A curva azul representa o valor de saída.

7.2.1 Variação da resposta ao degrau com ganho proporcional

Para que houvesse maior sensibilidade na sintonia dos ganhos, são realizados ajustes nos mesmo durante a execução do programa, assim têm-se passos do ajuste menor que a unidade. Observando-se os ajustes comentados no programa (Apêndice A), percebe-se que o ganho proporcional declarado deve ser dividido por dois para se obter o ganho proporcional real, bem como o ganho integral. O ganho derivativo deve ser dividido por 25. Aplicando-se ganho proporcional um (real 0,5) temos a resposta do sistema apresentada pela Figura 7.3:

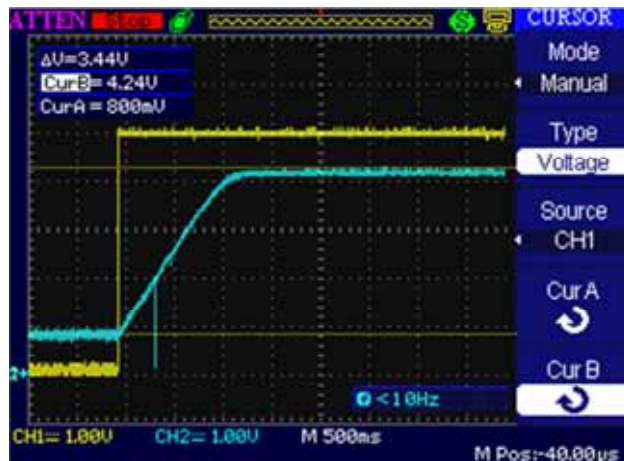
Figura 7.3 – Resposta do sistema para $K_p = 0,5$ / $K_d = 0$ / $K_i = 0$



Fonte: (produção do próprio autor)

Verifica-se que devido a não linearidade do motor DC ao longo de toda faixa de operação, descrita pela equação (3.9), o valor da referência nunca é atingido pela saída. Conforme a erro entre a entrada e saída diminuem, o valor da tensão média aplicada ao motor também diminui, até chegar ao ponto onde esta tensão não é mais suficiente para movimentar o eixo do motor, então o mesmo se mantém parado. Considera-se esta diferença entre o valor desejado e o atingido pelo controlador o erro em estado estacionário. Sabe-se que quanto maior o valor do ganho proporcional, menor será este erro, bem como o tempo de subida. Neste caso verifica-se que o tempo de atraso (50% do valor final) é de aproximadamente dois segundos. Ao se aumentar o valor do ganho proporcional, espera-se que haja menor erro em estado estacionário e também que seja diminuído o tempo de subida e atraso. A Figura 7.4 mostra a resposta do sistema com um ganho proporcional dez vezes maior que o último caso:

Figura 7.4 - Resposta do sistema para $K_p = 5 / K_d = 0 / K_i = 0$



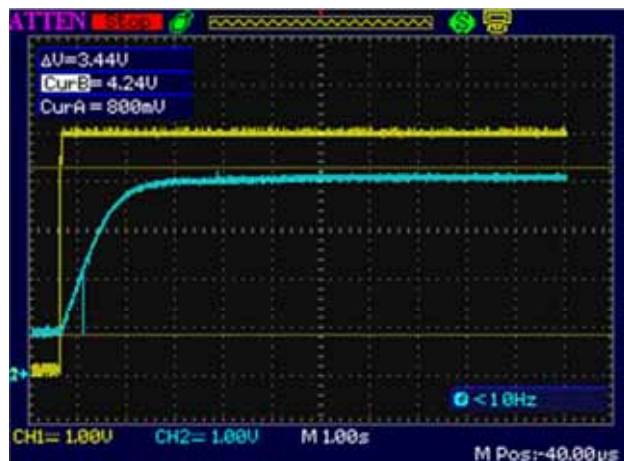
Fonte: (produção do próprio autor)

É possível se observar que o erro em regime permanente foi drasticamente reduzido, traduzido em tensão como uma redução do erro de 0,7V para 0,1V. O tempo de atraso foi reduzido de aproximadamente de dois segundos para 600ms.

7.2.2 Variação da resposta ao degrau com ganho integral

A fim de se observar melhor as variações ocorridas devido aos ganhos integral e derivativo, foi apresentada a resposta do sistema para ganho proporcional igual a 1,5, conforme a Figura 7.5:

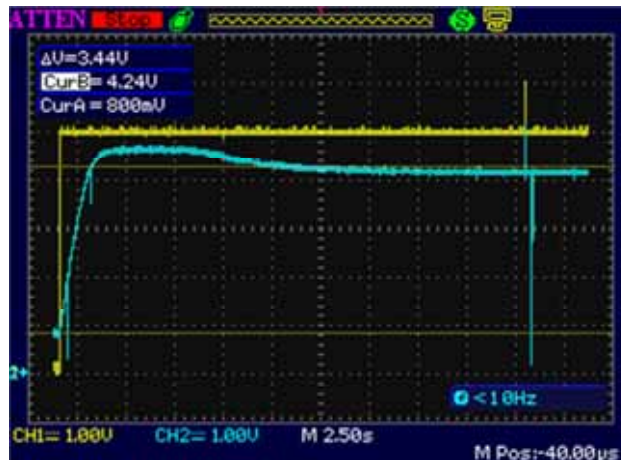
Figura 7.5 - Resposta do sistema para $K_p = 1,5 / K_d = 0 / K_i = 0$



Fonte: (produção do próprio autor)

Adicionando um ganho integral de meia unidade, tem-se a resposta na Figura 7.6:

Figura 7.6 - Resposta do sistema para $K_p = 1,5 / K_d = 0 / K_i = 0,5$



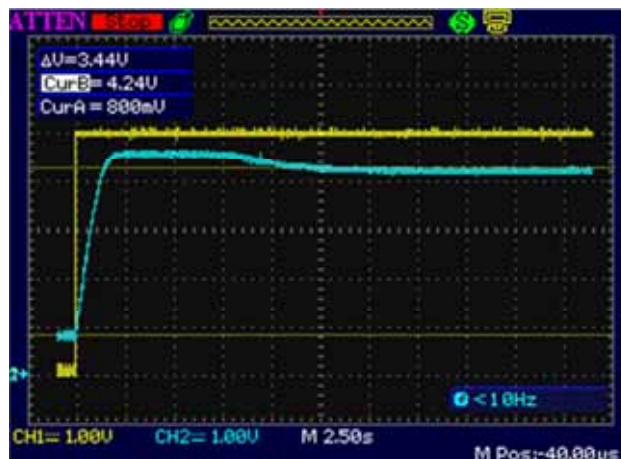
Fonte: (produção do próprio autor)

Considerando-se uma tolerância de 0,2V, tem-se que o tempo de subida foi reduzido de 5,5s para 1,5s. O erro em regime permanente foi eliminado, porém houve ultrapassagem de sinal, que geralmente é indesejada, principalmente em sistemas de posicionamento.

7.2.3 Variação da resposta ao degrau com ganho derivativo

Adicionando ao sistema um ganho derivativo real de 0,2 unidade, que para o programa significa ganho de cinco unidades, obtêm-se a seguinte resposta da Figura 7.7:

Figura 7.7 - Resposta do sistema para $K_p = 1,5 / K_d = 0,2 / K_i = 0,5$



Fonte: (produção do próprio autor)

Analisando-se a variação da resposta do sistema mostrada na Figura 7.7 em comparação com a da Figura 7.6, percebe-se que o tempo de subida foi pouco alterado e que há uma diminuição no valor máximo do sobressinal, o que pode ser traduzido como uma maior suavidade na desaceleração do sistema. Vê-se que o tempo de acomodação também é reduzido, ao passo que a saída do sistema sem a presença da ação de controle derivativo oscila com maior amplitude e durante um maior período de tempo em torno do valor de referência.

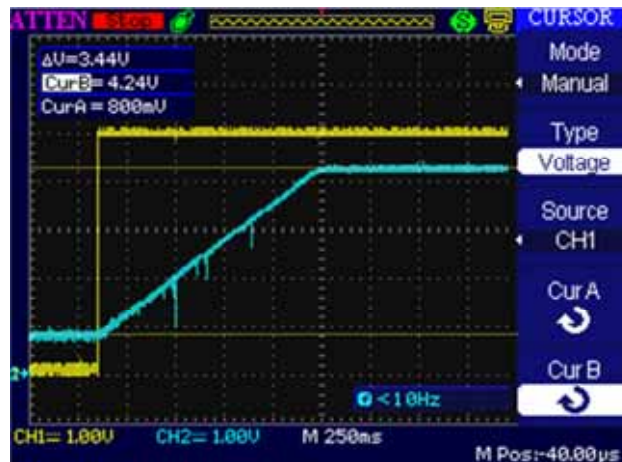
7.2.4 Ajustes dos ganhos do controlador PID (sintonia)

Tendo conhecimento da resposta do sistema para algumas combinações de ganhos do controlador, através de análise e comparação (empiricamente), pode-se ajustar a resposta do sistema buscando as características de tempo de subida, tempo de acomodação, máxima ultrapassagem, etc, desejadas. Como as características do sistema são desconhecidas (coeficiente de atrito dos rolamentos, inércia do motor/carga/sensor, torque, etc) não é possível que seja elaborado um modelo matemático satisfatório para cálculo dos coeficientes dos controladores. Existem métodos empíricos, como o de Ziegler-Nichols, que se baseia na resposta do sistema ao salto (degrau), e a partir desta resposta obtida calcula o ganho proporcional, tempo derivativo e integral para se obter determinada resposta do sistema. (LOTUFO, 2013)

Neste trabalho baseado nas respostas já obtidas anteriormente, pode-se variar os ganhos do controlador, analisar sua resposta e definir a mais satisfatória para cada caso. Analisando a Figura 7.6, vê-se que o ganho integral de meia unidade (0,5) já é satisfatório para eliminar o erro em estado estacionário, e que associado a um ganho proporcional de uma unidade e meia ($K_p = 1,5$) causa grande ultrapassagem de sinal. Ao se aumentar o ganho proporcional, espera-se que o tempo de subida seja diminuído, ou seja, o acúmulo do erro se torna menor. Isto significa também que a integral do erro terá menor valor assim que o valor de referência for atingido pela primeira vez, então espera-se que a ultrapassagem seja menor, já que o sistema não apresenta grande inércia de movimento. Além disso com maior ganho proporcional, no momento em que houver a tendência da ultrapassagem, o controlador reagirá mais abruptamente a este desvio do valor de referência. Ao se aumentar o ganho proporcional

do sistema da Figura 7.7 para 15 unidades, obtém-se a seguinte resposta apresentada na Figura 7.8.

Figura 7.8 - Resposta do sistema para $K_p = 15 / K_d = 0,2 / K_i = 0,5$



Fonte: (produção do próprio autor)

Vê-se um tempo de subida reduzido de 1,75s para 1,15s. Além disso não há mais ultrapassagem do sinal. Esta configuração é precisa quanto ao erro em regime permanente e apresenta resposta bastante rápida. Considerando que em uma situação a rapidez na ação seja menos priorizada que o consumo de energia e deseja-se que o sistema possua desaceleração mais suave, pode-se diminuir o ganho proporcional e aumentar o ganho derivativo. Diminuindo o ganho proporcional para dez unidades e tornando o ganho derivativo seis vezes maior, teremos a resposta apresentada pela Figura 7.9:

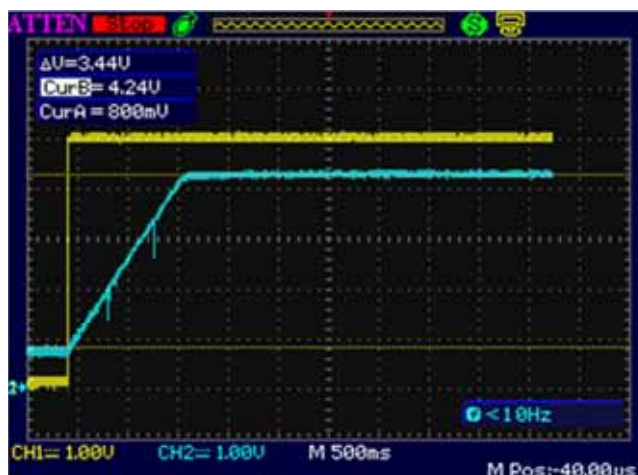
Figura 7.9 - Resposta do sistema para $K_p = 10 / K_d = 1,2 / K_i = 0,5$



Fonte: (produção do próprio autor)

Têm-se que o tempo de subida aumentou para 1,4s. Entretanto percebe-se que a desaceleração do sistema ficou bem mais suave, e ainda não houve ultrapassagem de sinal. Na tela do osciloscópio, bem como nas figuras apresentadas, não é possível se enxergar a oscilação em alta frequência que o sistema permanece após atingir o valor alvo, e por este fato foi introduzida a faixa de tolerância que corta a atuação do motor. Logicamente com maior ganho proporcional, que significa mais sensibilidade e rapidez na correção do erro pelo controlador, esta oscilação será maior. Analogamente o ganho derivativo também aumenta esta oscilação, pois o sistema tende a reagir mais rápido a qualquer variação. Analisando as respostas obtidas pelo osciloscópio digital a observação do sistema em funcionamento a olho nu, foi decidido o uso dos ganhos proporcional com valor de dez unidades ($K_p=10$), ganho derivativo de quatro centésimos de unidade ($K_d=0,04$) e ganho integral de meia unidade ($K_i=0,5$). A resposta do sistema ao degrau é mostrada pela Figura 7.10:

Figura 7.10 - Resposta do sistema ao degrau para $K_p = 10 / K_d = 0,04 / K_i = 0,5$



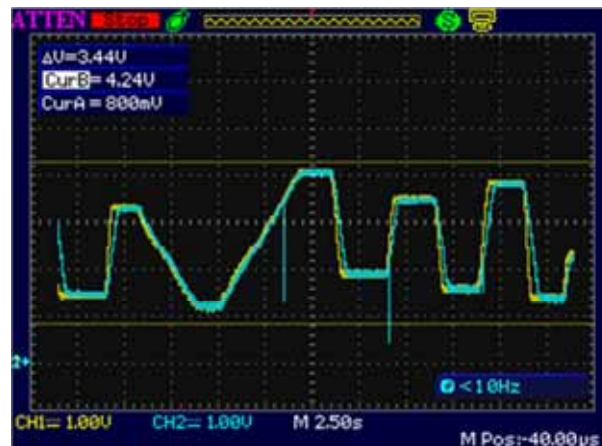
Fonte: (produção do próprio autor)

Vê-se que para tal sintonia, o controlador busca a referência de entrada de maneira veloz, não ocorre ultrapassagem do valor de referência e tampouco oscilação de grande valor em torno do valor de referência após este ter sido atingido pela saída. Com uma faixa de tolerância de $\pm 0,1\%$, poucos instantes após o valor de saída atingir a faixa de tolerância, não há oscilações/vibrações, ou seja, o motor é desligado. Isto evita desgaste excessivo das engrenagens, rolamentos, escovas do motor e dos componentes eletrônicos que o alimentam.

Considerando-se que a saída do sistema deve ser um seguidor da entrada, ou seja, o eixo de saída deve seguir a posição do eixo de entrada, e este sinal de entrada é dinâmico e aleatório (não determinístico), para verificação de tal condição de funcionamento foi aplicada

uma entrada aleatória contínua ao sistema e a resposta equivalente é apresentada na Figura 7.11:

Figura 7.11 - Resposta do sistema a entrada aleatória para $K_p = 10$ / $K_d = 0,04$ / $K_i = 0,5$



Fonte: (produção do próprio autor)

Percebe-se que o sistema responde rapidamente e quando não é limitado pela velocidade máxima do motor, a saída é praticamente uma reprodução fiel da entrada, sem erro em regime e pouco atraso na resposta.

8 CONCLUSÕES

Este trabalho apresentou a proposta da construção de um controlador PID de baixo custo, utilizando o microcontrolador PIC, e a aplicação do mesmo em um sistema de posicionamento antes controlado por circuito integrado dedicado. Tal substituição mostrou-se justificada, tendo o sistema sido testado e apresentado bom desempenho. A versatilidade do sistema provou ser uma grande vantagem, pois possibilita ajuste dos ganhos do controlador, faixa de tolerância, frequência de amostragem e substituição do circuito de potência adequado para qualquer escala de motor, impossíveis com a utilização do circuito dedicado. Além disso, o PIC16F648 não teve todos seus recursos utilizados. Seis pinos de entrada e/ou saída, e 67% da memória de programa não foram utilizados, o que ainda dá a possibilidade de melhoria nas facilidades do controlador, como: display de LCD para entrada dos parâmetros do controlador, comunicação com sensores de sobrecorrente ou sobretensão, entre outros.

As aproximações matemáticas realizadas para a implementação do algoritmo PID se mostraram válidas, tendo em vista que o desempenho do sistema foi verificado e a mudança na resposta do sistema de posicionamento angular foi coerente com a esperada através da variação da intensidade dos diferentes tipos de ação de controle. Através do ajustes dos coeficientes do controlador é possível se obter uma configuração mais próxima da ideal e que atenda os requisitos dos sistema, como faixa de tolerância, tempo de subida, tempo de acomodação, estabilidade, ultrapassagem, os quais conseqüentemente afetam fatores como o consumo, desgaste, geração de ruídos, etc. Um dos principais problemas verificados quando da não utilização do controle PID, era a instabilidade do sistema em torno do valor alvo, o que foi praticamente eliminado com a aplicação do controle PID.

Pode-se concluir que o controle PID é realmente eficiente no controle de processos industriais, e a sua implementação em sistemas digitais, sejam eles de alta capacidade ou embarcados e de baixo custo, como no caso do PIC, é possível e traz resultados satisfatórios quanto a desempenho, custo e versatilidade.

REFERÊNCIAS

LOTUFO, F. A. - **Material didático - Análise e Controle de Sistemas Dinâmicos**, UNESP/Guaratinguetá – Departamento de Engenharia Elétrica, 2013. Disponível em:< <http://www.feg.unesp.br/~falotufu/Disciplinas/Sel3023/>>. Acesso em: 05 mar. 2013.

MACIEL, M.; **Controle PID com aproximação Digital para utilização no PIC**, 2012. Disponível em:< <http://microcontrolado.com/controle-pid-no-pic/>>. Acesso em: 12 mai. 2013.

MICROCHIP Technology Inc. **PIC16F684 Data Sheet 14-Pin, Flash-Based 8-Bit CMOS Microcontrollers with nanoWatt Technology**, 2007. Disponível em: < <http://ww1.microchip.com/downloads/en/DeviceDoc/41202F-print.pdf>> Acesso em: 02 fev. 2013.

OGATA, K. **Discrete-time control systems**. 2nd ed. New Jersey: Prentice-Hall, 1995. 745 p.

OGATA, K. **Engenharia de Controle Moderno**. 3. ed. Rio de Janeiro: Livros Técnicos e Científicos Editora S.A., 1998. 813 p.

PEREIRA, F. **Microcontroladores PIC: programação em C**. 2. ed. São Paulo: Érica, 2003. 358 p.

SILVA, G. B. S. **Controle de Velocidade de Motor de Indução Trifásico usando PIC**. 2008. 83f. Trabalho de Graduação em Engenharia Elétrica, Faculdade de Engenharia do Campus de Guaratinguetá, Universidade Estadual Paulista, Guaratinguetá, 2009.

TORO, V. D. **Fundamentos de Máquinas Elétricas**. 4. ed. Rio de Janeiro: Livros Técnicos e Científicos Editora S.A., 1999. 550 p.

BIBLIOGRAFIA CONSULTADA

DE SOUZA, D. J. **Desbravando o PIC: ampliado e atualizado para PIC16F628A**. 11. ed. São Paulo: Érica, 2007. 268 p.

DIRACDELTA, **Quantization Error**, [201-?]. Disponível em:<http://www.diracdelta.co.uk/science/source/q/u/quantization%20error/source.html#.Uyd33_ldWnZ>. Acesso em: 09 jun. 2013.

DIRETORIA de Hidrografia e Navegação - Marinha do Brasil, **Derivadas e Integrais**, 2009. Disponível em: <<https://www.mar.mil.br/dhn/dhn/ead/pages/matematica/unidade11/explica.html>>. Acesso em: 11 jun. 2013.

GHIRARDELLO, A. **Apostila sobre Modulação PWM**. Disponível em: <http://www.eletronica.org/arq_apostilas/apostila_pwm.pdf> Acesso em: 08 ago. 2013.

HAYES, M. H. **Teoria e problemas de processamento digital de sinais**. 8. ed. Porto Alegre: Bookman, 2006. 466 p.

HUDLAC, **Representação Digital do Som**, 2009. Disponível em: <<http://hudlac.wordpress.com/tag/quantizacao/>>. Acesso em: 05 jun. 2013.

MODELIX Robotics, **Motores**, [201-?]. Disponível em:<http://www.leomar.com.br/modelix/index.php?option=com_content&view=article&id=117:motores&catid=46:data-sheets&Itemid=61&lang=pt>. Acesso em: 08 jul.2013.

ON Semiconductor, **MC33030 Data Sheet - DC Servo Motor Controller/Driver**, 2006. Disponível em:<http://www.onsemi.com/pub_link/Collateral/MC33030-D.PDF>. Acesso em: 02 fev. 2013.

ONLINE Electrical Engineering Study Site, **Permanent Magnet DC Motor or PMDC Motor | Working Principle Construction**, [201?]. Disponível em: <<http://www.electrical4u.com/permanent-magnet-dc-motor-or-pmdc-motor/>>. Acesso em: 01 jul. 2013.

PATANÉ, E. J. **Implementação de Controle de Velocidade em Malha Fechada para Motores de Corrente Contínua Utilizando Sistema de Aquisição de Dados**. 2008, 123 f. Dissertação (Mestrado em Engenharia de Processos Químicos e Bioquímicos). Centro Universitário do Instituto Mauá de Tecnologia, Escola de Engenharia Mauá, São Caetano do Sul, 2008. Disponível em:< <http://www.maua.br/arquivos/dissertacao/h/a1d7a5ba4952ecb686467f78fbe4a1c4?>>. Acesso em: 12 mai. 2013.

RASHID, M. H. **Eletrônica de Potência: Circuitos, Dispositivos e Aplicações**. São Paulo: Makron Books, 1999. 828p.

SCIENTIFIC Volume Imaging, **Aliasing**, [201?]. Disponível em: < <http://www.svi.nl/AliasingArtifacts>>. Acesso em: 07 jun. 2013.

STEWART, J. **Cálculo**. 6. ed. .São Paulo: Cengage Learning, 2009. v. 1. 688p.

VASCONCELLOS, C; BASSO, F.; **Como se Produz Energia Elétrica**, 2010. Disponível em:<http://sfizporquevalenota.blogspot.com.br/2010_10_01_archive.html>. Acesso em: 12 mai. 2013.

WATZECK, JR. **Força magnética em um condutor retilíneo**, 2013. Disponível em:< <http://www.escolassempatria.com/2013/06/forca-magnetica-em-um-condutor-retilineo.html>>. Acesso em: 02 jul. 2013.

APÊNDICE A – PROGRAMA COMENTADO

```

1: #include "G:\pic do 0\main.h"
2: #include <stdlib.h>
3: #include <math.h>
4:
5: // Os ganhos definidos abaixo devem ser números inteiros.
6:
7: #define KPS 20 //Ganho Proporcional Real = KPS/2 - P/ ajustes de 0,5 unidades
8:
9: #define KIS 1 //Ganho Integral Real = KIS/2 - Gan.Inte Real=Ki -> KIS/1000 = Ki.T/2
10: //p/ f=250Hz -> KIS = (Ki.1000)/(2.250) = Ki.2
11:
12: #define KDS 1 //Ganho Derivativo Real = KDS/25 - Gan.Der Real=Kd-> kd = KDS.10 = Kd/T
13: //p/ f=250Hz --> KDS = Kd.25
14:
15: #define PWMorto 35 // Valor mín. do duty cycle do PWM. Sendo 0% = 0 e 100% = 1023.
16: // Quando a saída atinge o valor desejado e está dentro da tolerância.
17: // Abaixo deste valor o motor não se move, mas provoca vibração.
18: // 1 unidade equivale aprox. 0,1% -> Ex. 35 -> duty cycle mínimo 3,5%.
19:
20: #define DeadBand 1 // Módulo do erro onde não haverá resposta, faixa de tolerância.
21: // 1 = unidade do conversor AD, que tem faixa de 0 a 1023 un.
22: // 1 portanto significa (1/1024) = 0,0977% -- 0,1% de tolerância.
23: // Objetivo eliminar vibr./instab. após a saída atingir tolerância.
24:
25: signed int16 C1,C0,e1,e0=0,i0=0,i1,d,di,u,un,ki,kd,kp,ilr,i0r,iaux,elabs;
26: int1 done;
27:
28: /* DESCRIÇÃO DAS VARIÁVEIS DECLARADAS !!!
29:
30: C1 - Valor da posição de saída durante a amostra atual; (C1[t])
31: C0 - Valor da entrada desejada (|.....|); (C0[t])
32: e1 - Erro entre saída e entrada na amostra atual; (e[t]=C1[t]-C0[t])
33: e0 - Erro (|.....|) anterior; (e[t-1]=C1[t-1]-C0[t-1])
34: i0 - Integral do erro acumulada até a amostra anterior; (ui[t-1])
35: i1 - Integral do erro durante amostra atual / aprox. por Área; (ui[t]-ui[t-1])
36: d - Derivada do erro na amostra atual; (ud[t])
37: di - Soma da componente integral+derivativo da saída u[t]; (ud[t]+ui[t])
38: u - Saída do PID que irá controlar o duty cycle do PWM; (u[t]=up[t]+ud[t]+ui[t])
39: un - (|.....|) com sinal invertido; (un[t]=-u[t])
40: ki - Ganho integral - ajustado de KIS dentro da função main(); (Ki)
41: kd - (|.....|) deviativo - (|.....|); (Kd/T)
42: kp - (|.....|) proporcional- (|.....|); (Kp*T/2)
43:
44: ilr - Resto da divisão de i1 por 1000;
45: i0r - Acúmulo do resto de i1 dividido por 1000 dentro das amostras;
46: iaux - Variável auxiliar utilizado no cálculo da integral por aproximação trapezoidal
47: a fim de guardar o resto da divisão do valor de saída integral, já que só se
48: consideram números inteiros.
49:
50: elabs - Módulo do valor do erro (e1);
51: done - bit de controle para averiguar final da conversão AD;
52:
53: */
54:
55: /* Foram utilizadas variáveis inteiras de 16 bits, pois o range de 0x0000 a 0xFFFF
56: (-32768 a 32767) comporta os máximos valores de saída do conversor AD de 10bits -
57: 0x0000 a 0x03FF (0 a 1023), Bem como a resolução do duty cycle do PWM que é de 0 a
58: 1023 (0 a 100%).
59:
60: Após análise dos tempos necessários para conclusão dos cálculos PID a cada amostra,
61: percebeu-se que o uso de variáveis inteiras de 32 bits (signed int32) ou de ponto
62: flutuante 32bits (float) tornam o processamento lento. Com cálculos mais enxutos
63: quanto possível e somados o tempos necessários para aquisição/conversão dos valores de
64: entrada analógica e atualização da saída PWM, constatou-se que era possível realizar
65: este ciclo satisfatória/ dentro de um período de 1000us, que dá ao sistema uma taxa
66: de amostragem MÁXIMA de 1Khz.
67: */
68:
69: #int_timer0 //Interrupção programada por Timer 0, chamada a cada T = 1/Famost.
70: void timer0inter()
71: {
72: set_timer0(0x0B); //Ajuste do timer 0 para estouro cada 4ms - Famoat = 250Hz.
73: set_adc_channel(1); //Seleção do canal 1 para aquisição do valor de entr. ana.
74: delay_us(8); //Tempo de seleção do canal necessário p/ cada troca canal.
75: read_adc(ADC_START_ONLY); //Instrução que dá início a conversão AD.
76: done = adc_done(); //done = 1 quando a conversão AD estiver pronta.
77: while(!done) //Loop para aguardar o término da conversão AD.
78: {
79: done = adc_done();
80: }
81: C1=read_adc(ADC_READ_ONLY); //Estando pronta a conversão AD, o valor é salvo em C1.
82: set_adc_channel(0); //Proced. de conversão AD idêntico para canal analógico 0.
83: delay_us(8);
84: read_adc(ADC_START_ONLY);

```



```

85: done = adc_done();
86: while(!done)
87: {
88:     done = adc_done();
89: }
90: C0 = read_adc(ADC_READ_ONLY); //Sabe-se os valores da ref. e saída na amostra atual.
91:
92: if(C1>869) //Limite máx de referência imposto, para evitar que a saída
93: { //atinga o limite mecânico do potenciômetro, o que pode
94:     C1=869; //ocorrer quando há ultrapassagem do sinal de saída.
95: }
96: if(C1<154) //Limite mín da referência, pelo mesmo motivo do máximo.
97: {
98:     C1=154;
99: }
100:
101: e1 = C1-C0; //Cálculo do erro para amostral atual.
102:
103: if (e1>=0) //Cond. para cálculo do módulo do erro atual -> elabs =|e1|
104: { //Utilizado na rotina para verificação da banda morta.
105:     elabs=e1;
106: }
107: else
108: {
109:     elabs=-e1;
110: }
111:
112: output_low(PIN_A5); //saída A5 para checagem do período de amostr./conversão.
113:
114: if(elabs>DeadBand) //Dentro da tolerância, erro é considerado nulo p/ o controlador.
115: { //Para evitar respota intensa da componente integral e derivada.
116:     e1=e1; //Diminui a instábil. e vibração quando dentro da tolerância.
117: }
118: else
119: {
120:     e1=0;
121: }
122:
123: d = (e1-e0)*kd; //Cálc. da comp. saída der. ud[t] - erro atual menos erro anterior
124: //mult. pelo ganho der.(Famo. fixa, ajuste do ganho dentro da main).
125: //d = Kd*(e[t]-e[t-1])/T
126:
127: i1r = (e1+e0); //Cálc.da comp.int. ui[t]-ui[t-1]-> Ki*T*(e[t-1]+e[t])/2
128: i1 = i1r/1000; //Ajuste para tornar a componente integral menos sensível.
129: i1r = i1r - (i1*1000); //i1 é int16, resto da divisão em milésimos é alocado em i1r.
130: i1r = (i1r*ki) + i0r; //Milésimos acumulados das amost. ant. (já mult. pelo ganho)
131: //somados ao resto da amostra atual multiplicado pelo ganho.
132: iaux = i1r/1000; //iaux p/ verificar se os milésimos multiplicados pelo ganho
133: //geraram uma ou mais unidades inteiras.
134: i0r = i1r - (iaux*1000); //Se sim, estas são eliminadas dos milésimos acumulados e
135: //somadas ao valor da saída atual.
136: i1 = (i1*ki) + iaux; //Soma da parte inteira + (milésimos acumulados/1000).
137: i0 = i0 + i1; //Soma das comp. int. acumulada + período(atual)
138: //ui[t]=ui[t-1]+Ki*T*(e[t-1]+e[t])/2
139: //Próx. iter. i0 será o valor acumulado da integral -> ui[t-1]
140: /*
141: OBS: Sem a divisão por 1000, os valores de (e[t-1]+e[t]) se tornam muito grandes. Estes
142: multiplicados pelo ganho integral com poucas iterações transbordam os limites de
143: +32767/-32768 possíveis para variáveis int16. A fim de não serem desconsiderados os
144: possíveis milésimos gerados pela divisão, a cada iteração é salvo em i0r o valor dos
145: milésimos restantes que não completaram uma unidade inteira. Sem este ajuste, valores
146: importantes acumulados do erro são perdidos, o que causa imprecisão no cálculo contínuo
147: da componente integral de saída, consequentemente imprecisão na saída do controlador.
148:
149: A soma máxima teórica de ei+e0 é (1023+1023) = 2046. O maior valor que os milésimos podem
150: assumir é de 999, que é o maior valor possível a ser multiplicado pelo ganho KIS. Sendo
151: o limite int16 de 32767, o maior ganho KIS possível que garantirá a ausência de erro no
152: cálculo contínuo acumulado é de 32. Mais além disso, com 250 iterações por minuto em um
153: processo lento e com alto ganho, é possível que haja estouro deste limite. Neste caso
154: deve-se avaliar o período de amostragem, pois quanto menor este for, menos iterações
155: serão realizadas por minuto, afastando a possibilidade de estouro destes limites.
156:
157: Este não fator não é tão relevante nas divisões ocorridas nos cálculos das componentes
158: proporcional e derivada, pois ambas atuam instantaneamente e são calculadas a cada
159: amostra, independentemente do valor de saída de cada componente da amostra anterior.
160: */
161:
162:
163: di = i0 + d; //Soma das comp. int. + der. = ui[t] + ud[t];
164: u = (kp*e1)/2; //Divide-se a componente proporcional por 2, antes da soma das 3
165: //para que se tenha um passo de 0,5 unidades no ajuste do ganho.
166: u = u + di; //Soma das 3 comp. PID, u[t] = up[t]+ui[t]+ud[t]; onde up[t]=Kp*e[t]
167:
168: e0 = e1; //Salvando erro atual, que na próx. iteração será erro anterior.

```

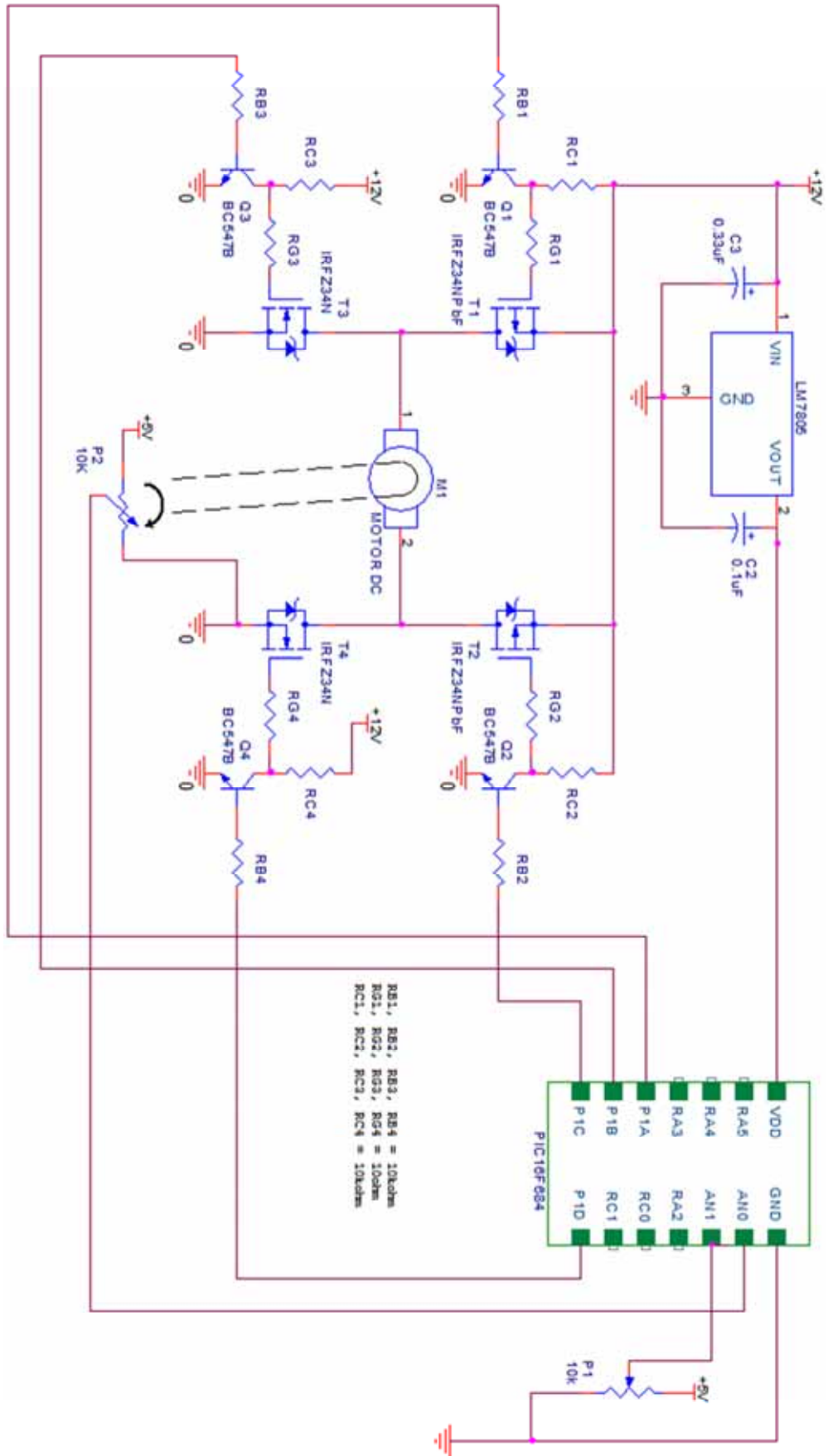
```

169: un = -1*u;          //Inversão de sinal var. da saída. Não há duty cycle negativo.
170:                    //Então inverte-se o sentido de rotação trocando o sinal de u.
171:
172: output_high(PIN_A5); //saída A5 usada para checagem de tempo de amostragem
173:
174:
175: if(u>0)             //PWM ã aceita duty cycle negativo,
176: {                  //feita verificação para escolha do
177:                  //sentido de rotação programado.
178:
179:     if((u<PWMorto) && (elabs<=DeadBand)) //Se a diferença entre referência e saída
180:     { //estiverem dentro da faixa de tolerância
181:         u=0; //definida, e o cálculo da saída gerar
182:         //valor de duty cycle que não movimentaria
183:         //motor, a fim de evitar desgaste
184:         //desnecessário e vibração, o duty cycle
185:         //é considerado nulo.
186:     }
187:     else
188:     {
189:         if(u>1023) //100% de duty cycle corresponde a 1023.
190:         { //Val. de saída maiores não são possíveis
191:             u = 1023; //O duty cycle é truncado em = 1023.
192:         }
193:         setup_ccp1(CCP_PWM|CCP_PWM_FULL_BRIDGE|CCP_PWM_H_L); //Aciona PWM no sentido horár.
194:         //da ponte H configurada para
195:         //os transistores. p/ Vcc sejam
196:         //ativo alto e p/ GND baixo.
197:         set_pwm1_duty(u); //Seta o duty cycle com o valor
198:         //da saída u.
199:     }
200: }
201: else
202: {
203:     if((un<PWMorto) && (elabs<=DeadBand)) //Se o valor da saída resulta negativo,
204:     { //a mesma lógica aplicada acima é então
205:         un=0; //utilizada para o sentido contrário de
206:         //rotação do motor, apenas utilizando o
207:         //valor do módulo de u.
208:     }
209:     else
210:     {
211:         if(un>1023)
212:         {
213:             un = 1023; //duty cycle máx = 1023.
214:         }
215:     }
216:     setup_ccp1(CCP_PWM|CCP_PWM_FULL_BRIDGE_REV|CCP_PWM_H_L); //Aciona PWM no sentido
217:     //anti-horário da ponte H.
218:     set_pwm1_duty(un); //Seta o duty cycle com o
219:     //valor de -u.
220: }
221: clear_interrupt(INT_TIMER0); //Limpa flag interrupção
222: //Timer 0, para que esta
223: //possa ser chamada nova/
224: void main()
225: {
226:     SET_TRIS_C(0x03); //RC2,3,4 e 5 (pinos 8,7,6,5) como saídas e
227:     //RC0 e RC1 (pi. 9,10) entradas.
228:     output_C(0x14); //Estado seguro das saídas PWM desligadas,
229:     //C2/4 alto (Mofets p/ terra), 3 e 5 baixo.
230:     SET_TRIS_A(0x0F); //RA4 e 5 (pi. 3 e 2) como saídas, e de
231:     //RA0 a RA3 (pi. 13,12,11,4) entradas.
232:     setup_adc_ports(sAN0|sAN1); //Seleção de portas analógicas (AN0 e AN1).
233:     setup_adc_ports(VSS_VDD); //Range VSS_VDD para o conversor AD.
234:     setup_adc_ports(0x8000); //Justifica a direita o resultado da
235:     //conversão analógica.
236:     setup_adc(ADC_CLOCK_DIV_16); //Clock ADC 8MHz/16 - TAD = 2us.
237:     setup_timer_2(T2_DIV_BY_16,249,1); //Config. Timer 2 p/ uso do PWM (ECCP),
238:     //clock, frequência e range do duty cycle.
239:     clear_interrupt(INT_TIMER0); //Limpa flag interrupção Timer 0.
240:     setup_timer_0(RTCC_INTERNAL|RTCC_DIV_32); //Incrementa Timer0 a cada 32*(Tyc-Fosc/4)=
241:     // = (8*0,5us), que conta de 0 a 255;
242:     enable_interrupts(INT_TIMER0); //Habilita interrupção Timer 0.
243:     enable_interrupts(0x0B40); // || interrupções de periférico.
244:     enable_interrupts(GLOBAL); // || || globais.
245:
246:     kp = KPS; // kp = KPS - Período de amostragem não altera kp;
247:
248:     ki = KIS; //Dependendo do processo e da taxa de amostragem, é necessário se fazer
249:     //ajustes no ganho ki, antes que este seja aplicado na fórmula final, a
250:     //fim de evitar estouro de variáveis e aumentar a sensibilidade do ajuste
251:     //do ganho (menor passo).
252:
253:     kd = 10*KDS; //Assim como o ganho integral o ganho derivativo deve ser ajustado
254:     //conforme o processo e a taxa de amostragem, a fim de torná-lo sensível

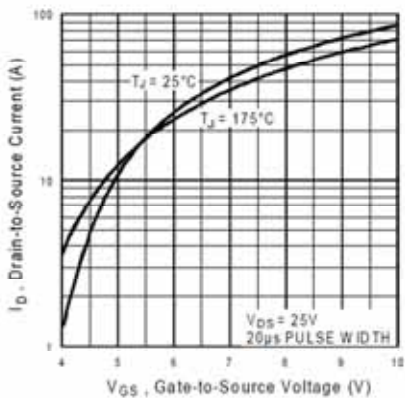
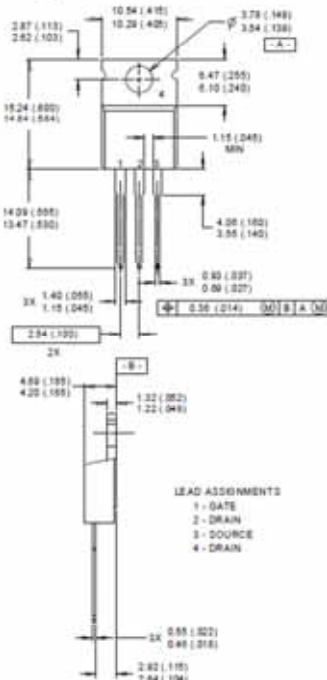
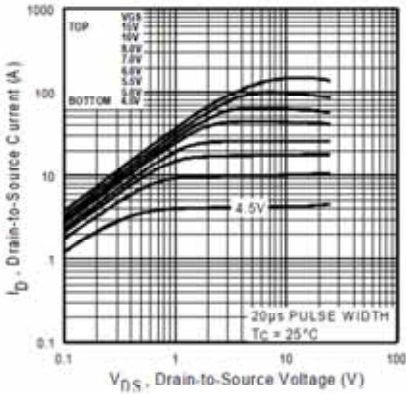
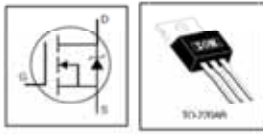
```

```
253:             //ao ajuste.
254:
255: /* Como os ganhos são variáveis int16, o ajuste aqui realizado dentro da função main,
256: deve ser realizado sempre quando é necessário um aumento do ganho declarado no cabeçalho
257: do programa (#define), a fim de evitar multiplicações repetidas a cada iteração. Uma
258: divisão pode torná-lo nulo, quando o resultado for menor que a unidade, ou fazer com que
259: perca exatidão em divisões com resto. Ex: para int16 -> 5/2 = 4/2 = 2. Divisões devem
260: ser realizadas após a multiplicação do ganho pelo erro, dentro da rotina periódica.
261: */
262:
263: while(true) //Após a declaração de pinos de saída/entrada/analógica, acerto de timer,
264: {          //ajuste de ganhos, etc; nada mais é necessário ser pré-processado.
265:           //Inicia-se então a conversão AD, cálculos PID e atualização do duty
266:           //cycle do PWM. Estes devem ser realizados periodicamente, portanto são
267:           //realizados dentro da interrupção do timer0, ajustada pelo valor do clock
268: }          //do timer0 e seu valor de contagem.
```

APÊNDICE B – DIAGRAMA ESQUEMÁTICO



ANEXO A – CARACTERÍSTICAS OPERACIONAIS IRFZ34N



Absolute Maximum Ratings

Parameter	Max.	Units
$I_D @ T_C = 25^\circ\text{C}$	29	A
$I_D @ T_C = 100^\circ\text{C}$	20	
I_{DM}	100	A
$P_D @ T_C = 25^\circ\text{C}$	68	W
	0.45	W/°C
V_{GS}	± 20	V
E_{AS}	65	mJ
I_{AR}	16	A
E_{AR}	6.8	mJ
dv/dt	5.0	V/ns
T_J	-55 to +175	°C
T_{STG}		
	300 (1.6mm from case)	
	10 lbf•in (1.1N•m)	

Thermal Resistance

Parameter	Min.	Typ.	Max.	Units
R_{JC}	—	—	2.2	°C/W
R_{CS}	—	0.50	—	
R_{JA}	—	—	62	

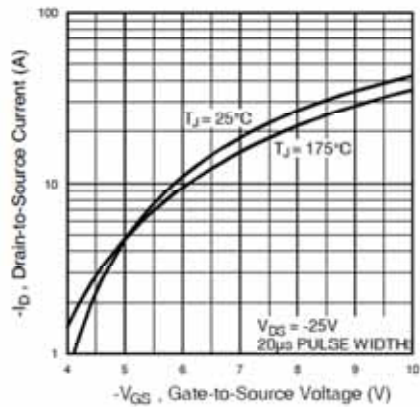
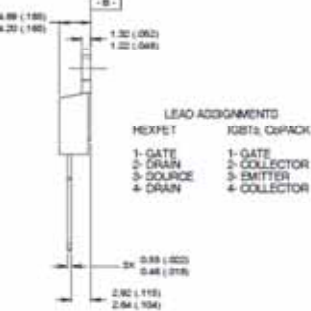
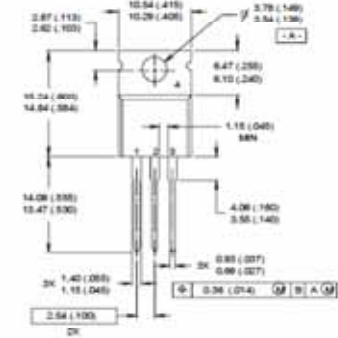
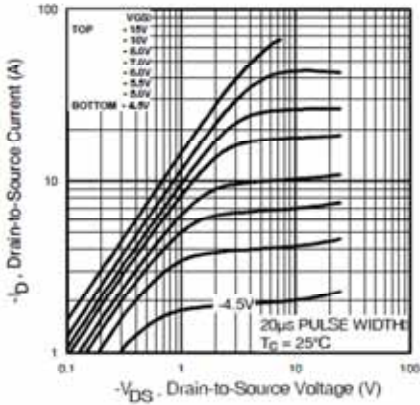
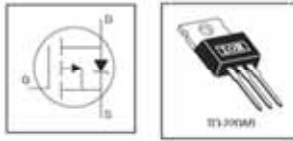
Electrical Characteristics @ $T_J = 25^\circ\text{C}$ (unless otherwise specified)

Parameter	Min.	Typ.	Max.	Units	Conditions
V_{BRDSS}	55	—	—	V	$V_{GS} = 0V, I_D = 250\mu A$
$\Delta V_{BRDSS}/\Delta T_J$	—	0.052	—	V/°C	Reference to 25°C, $I_D = 1mA$
$R_{DS(on)}$	—	—	0.040	Ω	$V_{GS} = 10V, I_D = 16A$
$V_{GS(th)}$	2.0	—	4.0	V	$V_{DS} = V_{GS}, I_D = 250\mu A$
g_{fs}	6.5	—	—	S	$V_{GS} = 25V, I_D = 16A$
I_{DSS}	—	—	25	μA	$V_{DS} = 55V, V_{GS} = 0V$
	—	—	250	μA	$V_{DS} = 44V, V_{GS} = 0V, T_J = 150^\circ\text{C}$
I_{GSS}	—	—	100	nA	$V_{GS} = 20V$
	—	—	-100	nA	$V_{GS} = -20V$
Q_{gs}	—	—	34	nC	$I_D = 16A$
Q_{gs}	—	—	6.8	nC	$V_{DS} = 44V$
Q_{gd}	—	—	14	nC	$V_{GS} = 10V$, See Fig. 6 and 13
$t_{d(on)}$	—	7.0	—	ns	$V_{DD} = 28V$
t_r	—	49	—	ns	$I_D = 16A$
$t_{d(off)}$	—	31	—	ns	$R_G = 18\Omega$
t_f	—	40	—	ns	$R_D = 1.8\Omega$, See Fig. 10
L_D	—	4.5	—	nH	Between lead, 6mm (0.25in.) from package and center of die contact
L_S	—	7.5	—	nH	
C_{iss}	—	700	—	pF	$V_{DS} = 0V$
C_{oss}	—	240	—	pF	$V_{DS} = 25V$
C_{rss}	—	100	—	pF	$f = 1.0MHz$, See Fig. 5

Source-Drain Ratings and Characteristics

Parameter	Min.	Typ.	Max.	Units	Conditions
I_S	—	—	29	A	MOSFET symbol showing the integral reverse p-n junction diode.
I_{SM}	—	—	100		
V_{SD}	—	—	1.8	V	$T_J = 25^\circ\text{C}, I_S = 16A, V_{GS} = 0V$
t_{rr}	—	57	86	ns	$T_J = 25^\circ\text{C}, I_r = 16A$
Q_{rr}	—	130	200	nC	$di/dt = 100A/\mu s$
t_{on}	Intrinsic turn-on time is negligible (turn-on is dominated by L_S+L_D)				

ANEXO B – CARACTERÍSTICAS OPERACIONAIS IRFZ34NPbF



Absolute Maximum Ratings

	Parameter	Max.	Units
A	$I_D @ T_C = 25^\circ C$	-19	Continuous Drain Current, $V_{GS} @ -10V$
	$I_D @ T_C = 100^\circ C$	-14	Continuous Drain Current, $V_{GS} @ -10V$
	I_{DM}	-68	Pulsed Drain Current
W	$P_D @ T_C = 25^\circ C$	68	Power Dissipation
$W/^\circ C$		0.45	Linear Derating Factor
V	V_{GS}	± 20	Gate-to-Source Voltage
mJ	E_{AS}	180	Single Pulse Avalanche Energy
A	I_{AR}	-10	Avalanche Current
mJ	E_{AR}	6.8	Repetitive Avalanche Energy
V/ns	dv/dt	-5.0	Peak Diode Recovery dv/dt
$^\circ C$	T_J	-55 to +175	Operating Junction and Storage Temperature Range
	T_{STG}	300 (1.6mm from case)	Soldering Temperature, for 10 seconds
		10 lb-in (1.1N-m)	Mounting torque, 6-32 or M3 screw

Thermal Resistance

	Parameter	Typ.	Max.	Units
	R_{JC}	—	2.2	Junction-to-Case
	R_{CS}	0.50	—	Case-to-Sink, Flat, Greased Surface
	R_{JA}	—	62	Junction-to-Ambient

Electrical Characteristics @ $T_J = 25^\circ C$ (unless otherwise specified)

Parameter	Min.	Typ.	Max.	Units	Conditions
$V_{(BR)DSS}$	-55	—	—	V	$V_{GS} = 0V, I_D = -250\mu A$
$\Delta V_{(BR)DSS}/\Delta T_J$	—	-0.05	—	V/°C	Reference to 25°C, $I_D = -1mA$
$R_{DS(on)}$	—	—	0.10	Ω	$V_{GS} = -10V, I_D = -10A$
$V_{GS(th)}$	-2.0	—	-4.0	V	$V_{DS} = V_{GS}, I_D = -250\mu A$
g_m	4.2	—	—	S	$V_{DS} = 25V, I_D = -10A$
I_{DSS}	—	—	-25	μA	$V_{DS} = -55V, V_{GS} = 0V$
	—	—	-250	μA	$V_{DS} = -44V, V_{GS} = 0V, T_J = 150^\circ C$
I_{GSS}	—	—	100	nA	$V_{DS} = 20V$
	—	—	-100	nA	$V_{GS} = -20V$
Q_g	—	—	35	nC	$I_D = -10A$
Q_{gs}	—	—	7.9	nC	$V_{DS} = -44V$
Q_{gd}	—	—	16	nC	$V_{DS} = -10V$, See Fig. 6 and 13
$t_{D(on)}$	—	13	—	ns	$V_{DD} = -28V$
t_r	—	55	—	ns	$I_D = -10A$
$t_{D(off)}$	—	30	—	ns	$R_G = 13\Omega$
t_f	—	41	—	ns	$R_D = 2.6\Omega$, See Fig. 10
L_D	—	4.5	—	nH	Between lead, 6mm (0.25in.) from package and center of die contact
L_S	—	7.5	—	nH	
C_{iss}	—	620	—	pF	$V_{DS} = 0V$
C_{oss}	—	280	—	pF	$V_{DS} = -25V$
C_{rss}	—	140	—	pF	$f = 1.0MHz$, See Fig. 5

Source-Drain Ratings and Characteristics

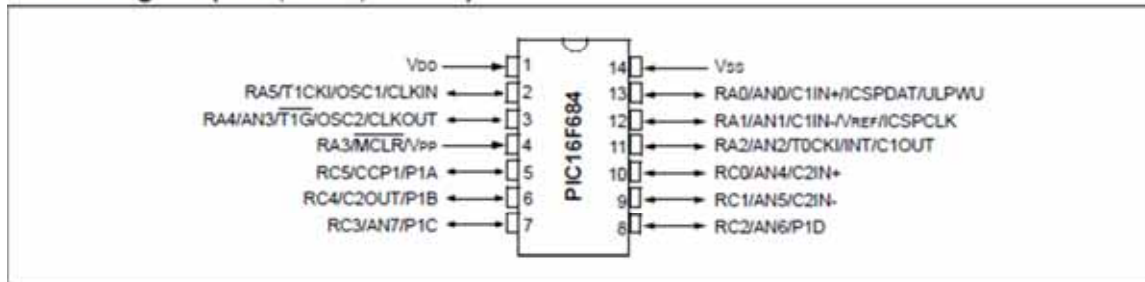
Parameter	Min.	Typ.	Max.	Units	Conditions
I_S	—	—	-19	A	MOSFET symbol showing the integral reverse p-n junction diode.
I_{SM}	—	—	-68	A	
V_{SD}	—	—	-1.6	V	$T_J = 25^\circ C, I_S = -10A, V_{GS} = 0V$
t_{rr}	—	54	82	ns	$T_J = 25^\circ C, I_R = -10A$
Q_{rr}	—	110	160	nC	$dv/dt = -100A/\mu s$
t_{on}	—	Intrinsic turn-on time is negligible (turn-on is dominated by L_S+L_D)			

Fonte: Datasheet IRFZ34NPbF – International Rectifiers ©

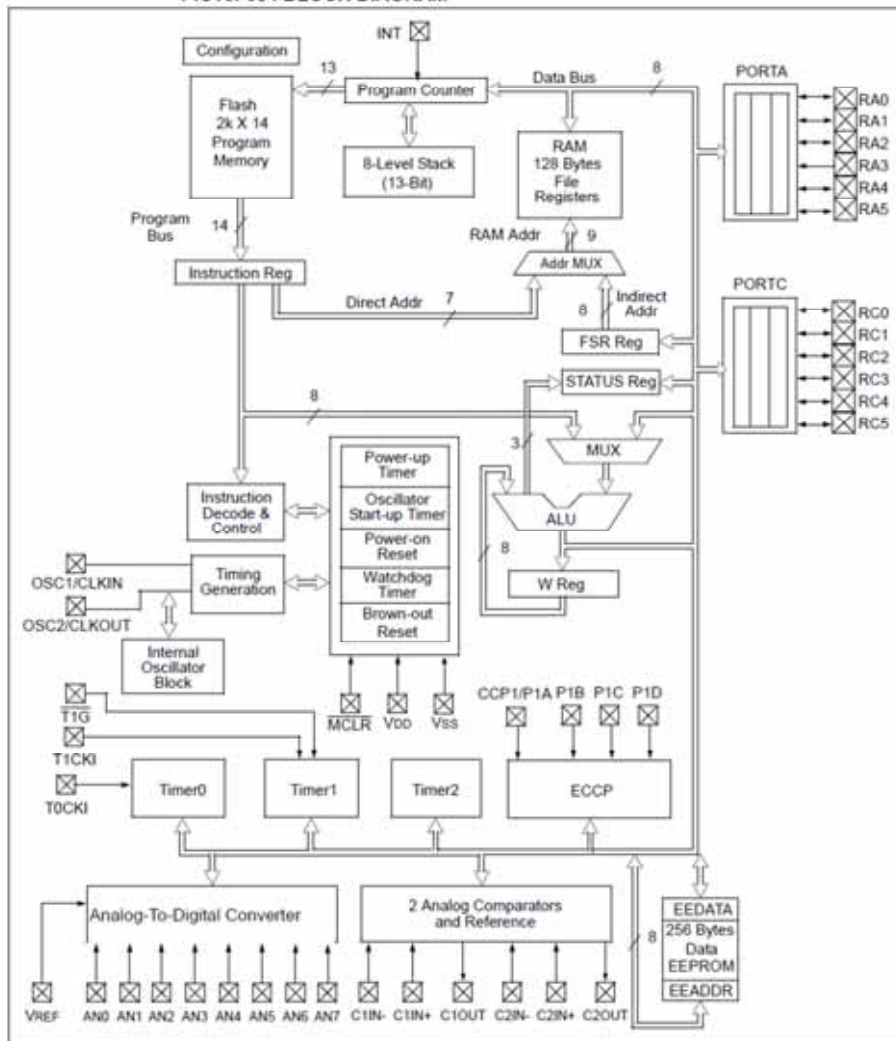
ANEXO C – PRINCIPAIS CARACTERÍSTICAS DO PIC16F684

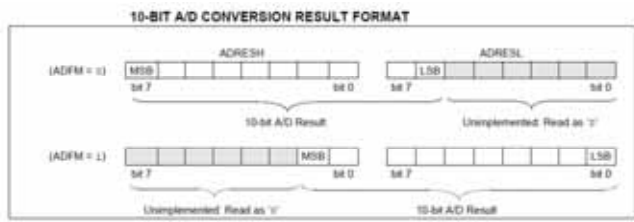
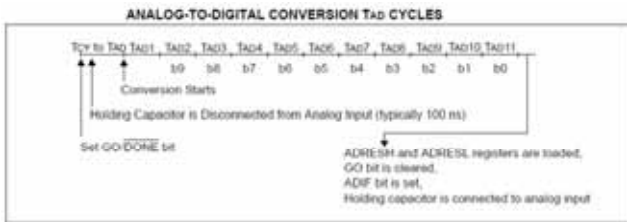
Device	Program Memory	Data Memory		I/O	10-bit A/D (ch)	Comparators	Timers 8/16-bit
	Flash (words)	SRAM (bytes)	EEPROM (bytes)				
PIC16F684	2048	128	256	12	8	2	2/1

14-Pin Diagram (PDIP, SOIC, TSSOP)



PIC16F684 BLOCK DIAGRAM





EQUATION 11-1: PWM PERIOD

$$PWM\ Period = \{(PR2) + 1\} \cdot 4 \cdot TOSC \cdot (TMR2\ Prescale\ Value)$$

EQUATION 11-4: PWM RESOLUTION

$$Resolution = \frac{\log[4(PR2 + 1)]}{\log(2)}\ bits$$

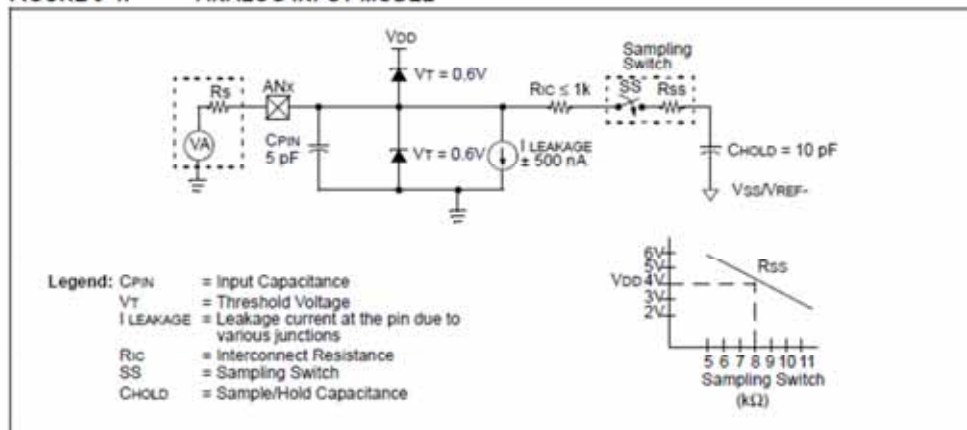
EQUATION 11-2: PULSE WIDTH

$$Pulse\ Width = (CCPR1L:CCP1CON<5:4>) \cdot TOSC \cdot (TMR2\ Prescale\ Value)$$

EQUATION 11-3: DUTY CYCLE RATIO

$$Duty\ Cycle\ Ratio = \frac{(CCPR1L:CCP1CON<5:4>)}{4(PR2 + 1)}$$

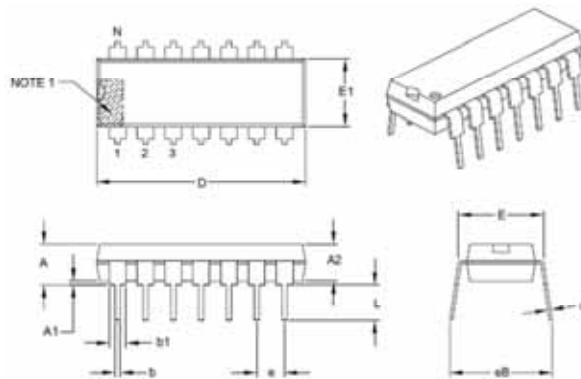
FIGURE 9-4: ANALOG INPUT MODEL



DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise stated)				
			Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended				
Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
D001 D001C D001D	VDD	Supply Voltage	2.0	—	5.5	V	Fosc ≤ 8 MHz: HFINTOSC, EC Fosc ≤ 4 MHz Fosc ≤ 10 MHz Fosc ≤ 20 MHz
D002*	VDR	RAM Data Retention Voltage ⁽¹⁾	1.5	—	—	V	Device in Sleep mode

TABLE 15-9: PIC16F684 A/D CONVERTER (ADC) CHARACTERISTICS

Standard Operating Conditions (unless otherwise stated)							
Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$							
Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
AD01	NR	Resolution	—	—	10 bits	bit	
AD02	E _{IL}	Integral Error	—	—	±1	LSb	V _{REF} = 5.12V
AD03	EDL	Differential Error	—	—	±1	LSb	No missing codes to 10 bits V _{REF} = 5.12V
AD04	E _{OFF}	Offset Error	—	—	±1	LSb	V _{REF} = 5.12V
AD07	E _{GN}	Gain Error	—	—	±1	LSb	V _{REF} = 5.12V
AD06 AD06A	V _{REF}	Reference Voltage ⁽³⁾	2.2 2.7	—	— V _{DD}	V	Absolute minimum to ensure 1 LSb accuracy
AD07	V _{AIN}	Full-Scale Range	V _{SS}	—	V _{REF}	V	
AD08	Z _{AIN}	Recommended Impedance of Analog Voltage Source	—	—	10	kΩ	
AD09*	I _{REF}	V _{REF} Input Current ⁽³⁾	10	—	1000	μA	During V _{AIN} acquisition. Based on differential of V _{HOLD} to V _{AIN} .
			—	—	50	μA	During A/D conversion cycle.



Dimension Limits	Units	INCHES		
		MIN	NCM	MAX
Number of Pins	N	14		
Pitch	e	.100 BSC		
Top to Seating Plane	A	—	—	.210
Molded Package Thickness	A2	.115	.130	.195
Base to Seating Plane	A1	.015	—	—
Shoulder to Shoulder Width	E	.290	.310	.325
Molded Package Width	E1	.240	.250	.290
Overall Length	D	.735	.750	.775
Tip to Seating Plane	L	.115	.130	.150
Lead Thickness	c	.008	.010	.015
Upper Lead Width	b1	.045	.060	.070
Lower Lead Width	b	.014	.018	.022
Overall Row Spacing §	eB	—	—	.430

Fonte: Datasheet PIC16F684 – Microchip Technologies Inc

ANEXO D – AMBIENTES DE PROGRAMAÇÃO, GRAVAÇÃO E TESTE

CCS PIC® C Compiler PCW Version 4.078

```

1  #include "G:\pic do Duman.h"
2  #include "stdio.h"
3  #include "math.h"
4
5  // Os pontos definidos abaixo devem ser números inteiros, o ponto decimal é mais preciso.
6  // Portanto o ponto real aqui será o algoritmo FID e o ponto decimal será o mesmo que foi 100.
7
8  #define FID 1 //fator Proporcional Fp = FID/100
9  #define FID 1 // FID // Integral FID = FID/100
10 #define FID 1 // FID // Integral FID = FID/100
11 #define FID 1000 //FID de amostragem (usado) - FID não utilizado pois valor fixo de amostragem = 1000
12
13 signed long int CI, CO, e1, e0 = 0, us, id = 0, li, d, dl, w, KI, M;
14 int done;
15
16 /* DESCRIÇÃO DAS VARIÁVEIS DECLARADAS !!!
17
18 CI = valor da posição de saída durante da amostra atual;
19 CO = Valor da entrada amostrada .....!!
20 KI = erro entre a posição lida e a amostra amostrada da amostra/amostragem atual
21 M = erro .....!!
22 LI = Integral do erro amostrado até o momento da amostra amostrada;
23 DL = Integral do erro durante amostra atual / erro, por área;
24 KI = Integral do erro na amostra atual;
25 M = soma de componente integral+derivativa de saída w!!!
26 w = saída de controle PID que irá controlar o duty cycle do PWM;
27 M = .....!! SOM SINAIS INTEGRADOS;
28 KI = quando integral - ajustado de FID dentro da função main();
29 M = {} derivativo - .....!! FID;
30 M = {} proporcional - .....!! FID;
31
32 done = bit de controle para tempo de conversão AD das entradas analógicas;
33
34 /* //Utilizaremos variáveis inteiras de 16 bits, pois o range de 0x0000 a 0x7FFF (-32768 a 32767) comporta os cálculos
35 // Valores de saída da conversão AD de 10bits = 0x0000 a 0x3FFF (0 a 16383). Sem isso a resolução do duty cycle do
36 // PWM que é de 0 a 1000 (0 a 100%);
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

Microchip MPLAB IDE v8.86

```

1  //Descrição da função:
2  //Descricao de um período de 100ns, que se os valores das amostras
3  //utilizadas na mesma amostra (para as pontas de Li, Anal,
4  //adquiridas anteriormente para a mesma amostra, ou não de
5  //programa e mesmo sempre sempre sempre, utilizando gabarito 0
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

Programador PICKit2 – Fabricante *Robótica Simples*



Circuito montado em matriz de contatos e bancada de testes:

