



UNIVERSIDADE LUTERANA DO BRASIL
PRÓ-REITORIA DE GRADUAÇÃO
DEPARTAMENTO DE ENGENHARIA ELÉTRICA



Denizard Gomes Rocha

Controle de Equilíbrio
Para Pêndulo Inverso utilizando PID digital

Canoas, Julho de 2009



DENIZARD GOMES ROCHA

**Controle de Equilíbrio
Para Pêndulo Inverso utilizando PID digital**

Trabalho de Conclusão de Curso
apresentado ao Departamento de
Engenharia Elétrica da ULBRA como um
dos requisitos obrigatórios para a obtenção
do grau de Engenheiro Eletricista

Departamento:

Engenharia Elétrica

Professor Orientador:

Msc. Eng. Eletr. Paulo César Cardoso Godoy

Canoas

2009



FOLHA DE APROVAÇÃO

Nome do Autor: Denizard Gomes Rocha

Matrícula: 071011886-6

Título: Controle de Equilíbrio para Pêndulo Inverso utilizando PID digital

Trabalho de Conclusão de Curso apresentado ao Departamento de Engenharia Elétrica da ULBRA como um dos requisitos obrigatórios para a obtenção do grau de Engenheiro Eletricista

Professor Orientador:

Msc. Eng. Eletr. Paulo César Cardoso Godoy

CREA-RS: 116822

Banca Avaliadora:

Dra. Marília Amaral da Silveira

CREA-RS: 50909-D

Conceito Atribuído (A-B-C-D):

MSc. Miriam N. Cáceres Villamayor

Mestre em Engenharia - UFRGS, 1999

CREA: RS067231-D

Conceito Atribuído (A-B-C-D):

Assinaturas:

AUTOR
DENIZARD GOMES ROCHA

ORIENTADOR
MSC. PAULO CÉSAR CARDOSO GODOY

ÁVALIADOR
DRA. MARILIA AMARAL DA SILVEIRA

ÁVALIADOR
MSC. MIRIAM N. CÁCERES VILLAMAYOR

Relatório Aprovado em:

Denizard Gomes Rocha

Universidade Luterana do Brasil



DEDICATÓRIA

Dedico aos meus pais...



AGRADECIMENTOS

A todos que colaboraram direta ou indiretamente na elaboração deste trabalho, o meu reconhecimento.

Aos meus Pais por acreditarem em mim, me apoiando e incentivando.

Ao Professor Valner Brusamarello, por apresentar a proposta de trabalho de conclusão.

Agradeço também ao Professor Paulo Godoy por ter me aceito como orientando após a saída do professor Valner da universidade.

A professora Marília Amaral, pelas dicas preciosas e relevantes ao assunto dedicado a motores descritos neste trabalho.

Aos Professores da Ulbra do curso de engenharia elétrica, pelas aulas ministradas.

Aos colegas da Ulbra pelo apoio, incentivo e ajuda nos trabalhos.



EPÍGRAFE

Veritas vos liberabit...

*Deixar que os fatos sejam fatos naturalmente, sem que sejam forjados para
acontecer....*



RESUMO

Rocha, Denizard Gomes. Controle de Equilíbrio para Pêndulo Inverso utilizando PID digital. Trabalho de Conclusão de Curso em Engenharia Elétrica - Departamento de Engenharia Elétrica. Universidade Luterana do Brasil. Canoas, RS. 2009.

Neste trabalho foi desenvolvido um sistema de controle para um carro com um único eixo, remetendo ao problema clássico do pêndulo inverso. O desenvolvimento do sistema deu-se desde a construção mecânica até o desenvolvimento do hardware e software para o controle do equilíbrio. O pêndulo foi construído sobre um carro de um único eixo, composto por dois motores CC, o que dá ao mesmo a possibilidade de um movimento bidirecional. A rotação da base do pêndulo em relação ao eixo de suporte é medido por um potenciômetro e adquirida por uma placa Analógico/Digital. O acionamento dos motores é efetuado por um circuito de ponte H. Para o controle do pêndulo foi desenvolvido um controlador PID (Proporcional – Integral – Diferencial), implementado por um microcontrolador dsPIC. Antes do ajuste, uma análise aproximada do sistema foi realizada utilizando-se um software simulador. No final deste trabalho, concluiu-se que é possível equilibrar uma haste com auxílio de um controle de motores.

Palavras chave: pêndulo. mecânica. hardware. software. equilíbrio. bidirecional.



ABSTRACT

Rocha, Denizard Gomes. Balance Control for Reverse Pendulum using digital PID. Completion of work of course in Electrical Engineering - Department of Electrical Engineering. Lutheran University of Brazil. Canoas, Rio Grande do Sul. 2009.

In this work its developed a control system for a car with only a single axis, referring to the traditional problem of inverted pendulum. The system's development took place since the mechanical construction to the development of hardware and software to control the balance. The pendulum was built on a car from a single axis, composed of two DC motors, which gives it the possibility of a bi-directional movement. The rotation of the base of the pendulum in relation to the axis of support is measured by an potentiometer and acquired by an Analog / Digital. The power of the engines is performed by a bridge circuit of H. To control the pendulum, its developed a controller PID (Proportional - Integral - Differential), implemented by a microcontroller dsPIC. Before the adjustment, a rough analysis of the system was performed using a software simulator. At the end of this work, concluded that it is possible to balance a rod with the help of a control a car.

Keywords: pendulum. Mech. hardware. software. balance. bidirectional.



LISTA DE ILUSTRAÇÕES

1. Introdução

Figura 1. 1 – Imagem do Pêndulo Inverso.....	15
--	----

2. Referencial Teórico

Figura 2. 1 – Processo a ser controlado.....	18
Figura 2. 2 – Sistema de controle a malha aberta.....	18
Figura 2. 3 – Sistema de controle com retroação a malha fechada.....	18
Figura 2. 4 – Sistema de controle multivariável.....	19
Figura 2. 5 – PID - $K_p=4$; $T_i=1.5$; $T_d=0.1$ (tracejado), 0.4(pontilhado), 2(contínuo).....	20
Figura 2. 6 – Análise de Forças do Pêndulo Inverso com Carro.....	28
Figura 2. 7 – Diagrama de bloco para pêndulo inverso.....	31
Figura 2. 8 – Abrir e fechar da chave - PWM.....	32
Figura 2. 9 – Gráfico do pulso PWM do abrir e fechar de chaves.....	32
Figura 2. 10 – Gráfico do ciclo ativo - PWM.....	33
Figura 2. 11 – Controlando a potencia pelo ciclo ativo.....	33
Figura 2. 12 – Anti-phase PWM.....	34
Figura 2. 13 – Esquema pratico de um PWM.....	35
Figura 2. 14 – Gráfico da variação de calor no pulso.....	36

3. Materiais e Métodos

Figura 3. 1 – Diagrama de Blocos – Sistema de Controle e Pêndulo.....	38
Figura 3. 2 – Modelo do Pêndulo.....	38
Figura 3. 3 – Escala de desempenho.....	39
Figura 3. 4 – Representação de uma ponte H.....	40
Figura 3. 5 – Representação do uso de uma ponte H.....	41
Figura 3. 6 – vista do integrado LM298 (ponte H).....	41
Figura 3. 7 – Gravador <i>MicroICD</i> da <i>Microgenios</i>	42
Figura 3. 8 – esquema básico para gravação dsPIC.....	42
Figura 3. 9 – gráfico de ensaio do motor como gerador.....	44
Figura 3. 10 – tela do software compilador.....	47
Figura 3. 11 – Fluxograma, Software Pêndulo Inverso.....	47

4. Apresentação e Discussão dos Resultados

Figura 4. 1 - Componentes do pendulo inverso.....	49
Figura 4. 2 – Componentes vetoriais do pendulo inverso.....	50
Figura 4. 3 – Primeiro teste de saída PWM para o controle dos motores.....	58
Figura 4. 4 – Segundo teste de saída PWM para o controle dos motores.....	59
Figura 4. 5 – Terceiro teste de saída PWM para o controle dos motores.....	59
Figura 4. 6 – Quarto teste de saída PWM para o controle dos motores.....	60



LISTA DE TABELAS

3. Materiais e Métodos

3.3.4, Descrição dos Testes Realizados

Tabela 3. 1 – Ensaio a vazio como gerador.	44
Tabela 3. 2 – Cálculo das constantes k_1 e k_2	44
Tabela 3. 3 – Ensaio de rotor bloqueado.	45
Tabela 3. 4 – Cálculo de força contra eletromotriz, rotação e torque resistente.	45
Tabela 3. 5 – Cálculo de rendimento.....	46



LISTA DE ABREVIATURAS E SIGLAS

ABNT: Associação Brasileira de Normas Técnicas.

PWM: (*Pulse Width Modulation*) - Modulação de Largura de Pulso.

PID: Controlador Proporcional – Integral – Derivativo

(A/D): Conversores Analógicos para Digital

dsPIC: (*Digital Signal Programmable Integrat Circuit*) - Circuito Integrado Programável de Sinal Digital.

LCD: (Liquid Crystal Display) – Display de Cristal Liquido



LISTA DE SÍMBOLOS

ABNT: Associação Brasileira de Normas Técnicas.

L	comprimento da haste do pêndulo e centro de massa 0,3m
pm	Massa da haste 0,5kg
cm	Massa do carrinho 0,5kg
b	Coefficiente de força viscosa 0,1 Ns/m
g	Aceleração da gravidade 9,8 m/s ²
q	Ângulo do pêndulo radianos
w	Velocidade angular do pêndulo rad/s
I	Momento de inércia do pêndulo 0,06kg/m ²
x	Posição do carrinho metros
v	Velocidade do carrinho m/s
kp	Ganho Proporcional N/rd
ki	Ganho Integral 1/s
kd	Ganho Derivativo s



SUMÁRIO

1. INTRODUÇÃO	14
1.1. OBJETIVOS GERAIS	15
1.2. OBJETIVOS ESPECÍFICOS	15
1.3. METAS	16
1.4. RESULTADOS PREVISTOS	16
2. REFERENCIAL TEÓRICO	17
2.1. CONTROLE AUTOMÁTICO	17
2.2. ALGORITMO PID	19
2.3. CONVERSORES ANALÓGICOS/DIGITAIS	23
2.4. UMA VISÃO GERAL DA LINGUAGEM C	24
2.5. PÊNDULO INVERSO	27
2.6. PWM	31
3. MATERIAIS E MÉTODOS	38
3.1. DESCRIÇÃO GERAL DO SISTEMA	38
3.2. DESCRIÇÃO DOS SISTEMAS MECÂNICOS	39
3.3. DESCRIÇÃO DOS SISTEMAS ELETROELETRÔNICOS.....	39
3.4. DESCRIÇÃO DOS SISTEMAS INFORMÁTICOS E COMPUTACIONAIS.....	46
4. APRESENTAÇÃO E DISCUSSÃO DOS RESULTADOS	49
4.1. MODELO MATEMÁTICO PARA SIMULAÇÃO NO MATLAB	49
4.2. SIMULAÇÕES – MÉTODOS NUMÉRICOS.....	54
4.3. METODOLOGIA DOS TESTES INTEGRADOS NO SISTEMA FÍSICO	57
4.4. ANÁLISES ESTATÍSTICAS	60
4.5. COMPARAÇÕES DOS RESULTADOS DAS SIMULAÇÕES, MODELOS E O SISTEMA REAL	60
5. CONSIDERAÇÕES FINAIS	61
5.1. CONCLUSÕES	61
5.2. SUGESTÕES PARA TRABALHOS FUTUROS	62
REFERÊNCIAS	63
OBRAS CONSULTADAS	64
APÊNDICE A – SOFTWARE DE CONTROLE	65
APÊNDICE B – ESQUEMA ELÉTRICO	67
APÊNDICE C – ESQUEMA DE CONEXÃO DE DISPLAY	68
APÊNDICE D – VISUALIZAÇÃO GERAL DO CIRCUITO	69
APÊNDICE E – PÊNDULO INVERSO	70
ANEXO A – DIAGRAMA DE BLOCOS DO (DSPIC30F4011)	71
ANEXO B – ESQUEMA DE LIGAÇÃO PARA LM298	72
ANEXO C – DIAGRAMA DE BLOCO LM298 (PONTE H)	73
ÍNDICE	74



1. INTRODUÇÃO

A teoria de controles automáticos sempre foi integrada à área da engenharia. Cada vez mais, novos algoritmos e recursos computacionais estão sendo utilizados a fim de realizar os mais diversos tipos de controle. Os sistemas de controle automático são encontrados em abundância em setores da indústria, tais como controle de qualidade e fabricação de produtos, linha de montagem automática, controle de ferramentas, tecnologia espacial e de armamento, sistemas de transporte, sistemas de potência, robôs e muitos outros.

O projeto contemplou o desenvolvimento do software e hardware a realização experimentos práticos com o Pêndulo Inverso, o que introduziu um caráter interdisciplinar ao trabalho, com utilização de componentes elétricos/eletrônicos e mecânicos.

Dentro deste contexto, o presente trabalho procura desenvolver um sistema de controle de um Pêndulo Inverso em uma base móvel de um único eixo utilizando um algoritmo largamente empregado na Indústria e comprovado matematicamente – o PID.

O sistema empregado está baseado no Pêndulo Inverso, qual consiste um dos sistemas clássicos em controle de sistemas, dada a sua complexidade inerente. O sistema é constituído por um carro móvel sobre a qual oscila uma haste. Neste caso tem como um único eixo pretende-se manter a haste na posição vertical. Nesta haste se encontra o sensor que enviara um sinal a placa de controle e assim controlara o torque dos motores para um sentido ou outro para manter o equilíbrio do sistema.



Figura 1. 1 – Imagem do Pêndulo Inverso.

O projeto consiste em demonstrar ao corpo de alunos do curso de engenharia elétrica as possibilidades de criação e de gerenciamento de um projeto. Tendo em vista os conhecimentos adquiridos no curso, demonstrar para novos ingressantes no curso as possibilidades de conhecimentos adquiridos na faculdade de engenharia.

O problema do pêndulo inverso será apresentado em feiras de iniciação a faculdade de engenharia assim despertando mais jovens para os conhecimentos de elétrica e eletrônica.

1.1. Objetivos Gerais

Desenvolver um projeto de software e hardware onde a eficiência do processamento e o tempo de atendimento às solicitações do sistema sejam críticos. Desta forma, ter a oportunidade de colocar em prática todo o conhecimento de sistemas operacionais de tempo real, multiprocessado com sincronização entre processos e controle de hardware.

1.2. Objetivos Específicos

- Estudos de automação e controle
- Adaptação de sensor para medida angular
- Integração de um sistema de aquisição de dados para um microcontrolador
- Implementação de um software para controle de ângulo e posição



1.3. Metas

Ao final do trabalho espera-se ter:

Um hardware juntamente com software controlar e gerenciar um carro de um único eixo mantendo equilíbrio sozinho.

O sistema deverá possuir as seguintes funcionalidades:

- a) Adquirir o sinal do potenciômetro
- b) Mostrar leituras em um display
- c) Manter o equilíbrio

1.4. Resultados Previstos

O sistema construído tende a funcionar de maneira prevista, conseguindo um equilíbrio bastante eficiente do pêndulo. Contudo, o pêndulo não tende a parar numa dada posição, o que é de se esperar já que o sistema deve manter-se em equilíbrio constante.



2. REFERENCIAL TEÓRICO

A engenharia diz respeito ao conhecimento e ao controle de materiais e forças da natureza para o benefício da humanidade. Diz-se que o conhecimento de controle de segmentos, chamados com freqüência de sistemas, tem com finalidade de dotar a sociedade de produtos úteis e econômicos. Os objetivos duplos de conhecimento e controle são complementares, uma vez que o controle efetivo de sistemas requer que os sistemas sejam compreendidos e modelados. Além disso, a engenharia de controle deve considerar muitas vezes o controle de sistemas mal conhecidos, como sistemas de processos químicos. O presente desafio ao engenheiro de controle é criar o modelo e o controle de sistemas modernos, complexos e interligados, como sistemas de controle de tráfego, processos químicos e sistemas robóticos.

2.1. CONTROLE AUTOMÁTICO

A engenharia de controle é baseada nos fundamentos da teoria da retroação, e na análise de sistemas lineares, e integra conceitos da teoria de circuitos e de telecomunicações. Por isso, a engenharia de controle não está limitada a qualquer um dos ramos da engenharia. Por exemplo, freqüentemente um sistema de controle incluiu componentes elétricos, mecânicos e químicos. Além disso, ao se ampliar a compreensão da dinâmica de negócios, de processos sociais e políticos, será ampliada também a capacidade de controlar estes sistemas. Um sistema de controle é uma interconexão de componentes formando uma configuração de sistema que produzirá uma resposta desejada ao sistema. A base para a análise de um sistema é formada pelos fundamentos fornecidos pela teoria dos sistemas lineares, que supõe uma relação de causa e efeito para os

componentes do sistema. Além disso, um componente ou processo a ser controlado pode ser representado por um bloco como mostrado na Figura 2.1.



Figura 2. 1 – *Processo a ser controlado.*

A relação de entrada-saída representa uma relação de causa efeito no processo, que, por sua vez, representa um processamento de um sinal de entrada para fornecer um sinal de saída variável, quase sempre com uma amplificação de potência. Um sistema de controle a malha aberta utiliza um controlador ou atuador de controle para obter a resposta desejada como mostrado na Figura 2.2. Um sistema a malha aberta é um sistema sem rotação.

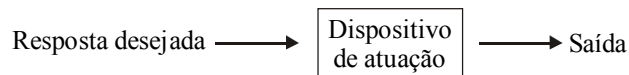


Figura 2. 2 – *Sistema de controle a malha aberta.*

Ao contrário de um sistema de controle a malha aberta, um sistema de controle a malha fechada utiliza uma medida adicional da saída real com a resposta desejada. A medida da saída é chamada de sinal de retroação. Um sistema de controle com retroação a malha fechada simples é mostrado na Figura 2.3.

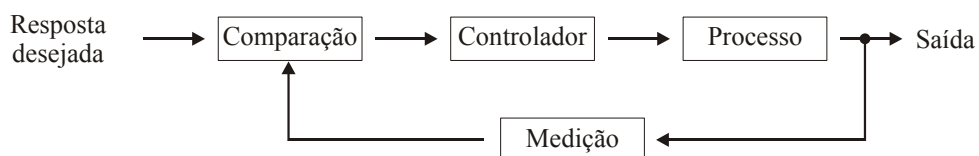


Figura 2. 3 – *Sistema de controle com retroação a malha fechada.*

Um sistema de controle com retroação é um sistema que tende a manter uma relação preestabelecida entre uma variável de sistema e outra, comparando funções destas variáveis e usando a diferença como meio de controle. Um sistema de controle com retroação usa freqüentemente uma função e uma relação preestabelecida entre a saída e a entrada para controlar o processo. Quase sempre a diferença entre a saída do sistema preestabelecida à saída do processo sob controle e a referência de entrada é amplificada e usada para controlar o processo

de modo que a diferença seja continuamente reduzida. O conceito de retroação tem sido o alicerce para a análise e o projeto dos sistemas de controle.

Por causa do aumento de complexidade dos sistemas de controle e do interesse em obter um desempenho ótimo, a importância da engenharia de sistemas de controle cresceu na década passada. Além disso, como os sistemas se tornaram mais complexos, o inter-relacionamento de muitas variáveis controladas deve ser considerado na estrutura de controle. Um diagrama de blocos retratando um sistema de controle multivariável está ilustrado na Figura 2.4.

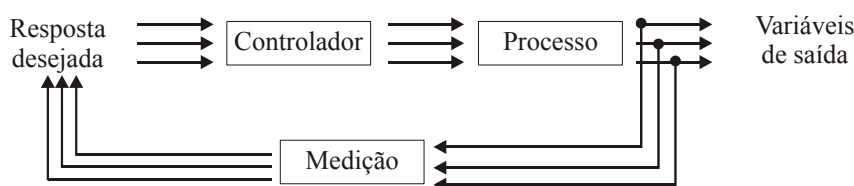


Figura 2. 4 – Sistema de controle multivariável.

Um exemplo comum de sistema de controle a malha aberta é a torradeira elétrica de cozinha. Um exemplo de um sistema de controle de malha fechada é uma pessoa dirigindo um automóvel (supondo que os olhos desta pessoa estejam abertos) a partir da observação da posição do automóvel e fazendo ajustes.

2.2. ALGORITMO PID

A combinação das ações proporcional, integral e derivativa para gerar um só sinal de controle, dá origem ao que chamamos de *controlador proporcional – integral – derivativo* ou simplesmente *PID*. O objetivo é aproveitar as características particulares de cada uma destas ações a fim de se obter uma melhora significativa do comportamento transitório e em regime permanente do sistema controlado. O sinal de controle gerado pelo controlador PID é genericamente dado como:

$$u(t) = K \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right) \quad (2-1)$$

Desta forma tem-se três parâmetros de sintonia no controlador: o ganho proporcional K (ação proporcional), o tempo integral T_i (ação integral) e o tempo derivativo T_d (ação derivativa).

Apesar de termos a disponibilidade das três ações básicas, dependendo da aplicação não será necessário a utilização de uma ou mais destas ações.

2.2.1. O Controlador Proporcional – Integral – Derivativo

A influência da ação derivativa na resposta do sistema pode ser observada na Figura 2.5.

A função de transferência do controlador PID é dada por:

$$G_{pid} = \frac{u(s)}{r(s)} = K \left(1 + \frac{1}{T_i s} + \frac{s p T_d}{s + p} \right) = \frac{K \left(s^2 + \frac{1 + T_d T_i}{T_i} s + \frac{p + T_i p}{T_i} \right)}{s(s + p)} \quad (2-2)$$

É importante ressaltar que as equações apresentadas constituem-se na versão clássica do controlador PID. Outras versões e variações existem, mas a filosofia de funcionamento, a partir da combinação dos efeitos das três ações básicas, é a mesma.

A Figura 2.5 mostra o efeito da ação derivativa considerando-se um controlador PID.

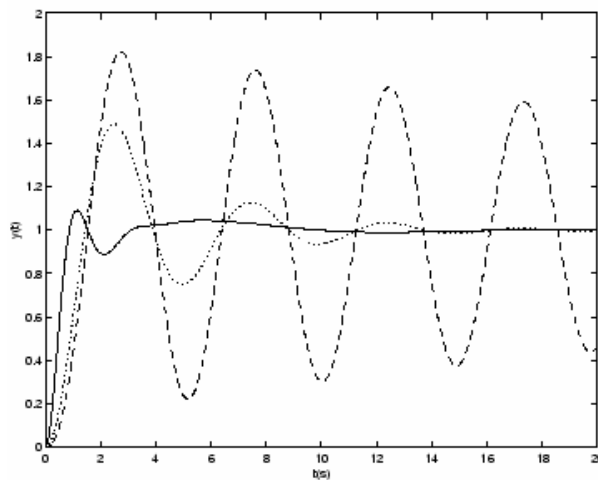


Figura 2. 5 – PID - $K_p=4$; $T_i=1.5$; $T_d=0.1$ (tracejado), 0.4 (pontilhado), 2 (contínuo)

2.2.2. Implementação Digital



A implementação do controlador PID pode ser feita fazendo-se aproximações numéricas das derivadas e da integral que aparecem na lei de controle. Desta forma, é possível descrever cada uma das ações por uma equação de recorrência. As equações de recorrência descrevem as operações matemáticas a serem programadas no microcontrolador ou no microcomputador onde será implementado o PID digital.

Consideremos que o período de amostragem T foi convenientemente escolhido de forma a satisfazer o Teorema da Amostragem. O sinal de controle será atualizado apenas nos instantes de tempo $t_k - kT$, ou simplificadaamente denotaremos a saída do processo, o sinal de referência, o sinal de controle e o erro no instante t_k respectivamente como $y(k)$, $r(k)$, $u(k)$ e $e(k)$.

2.2.3. Ação Proporcional

O termo proporcional discretizado a ser aplicado no instante k é dado por:

$$P(k) = K(r(k-1) - y(k-1)) = K(e(k-1)) \quad (2-3)$$

2.2.4. Ação Integral

A ação integral é dada por:

$$I(t) = \frac{K}{T_i} \int_0^t e(s) ds \quad (2-4)$$

ou equivalentemente por:

$$\frac{dI}{dt} = \frac{K}{T_i} e(k) \quad (2-5)$$

Existem várias maneiras de discretizar, ou seja, aproximar numericamente a equação (2-1). Abaixo são apresentados dois métodos:

- **Backward Differences:**



Neste caso temos a seguinte aproximação para a equação (2-5):

$$\frac{I(k) - I(k-1)}{T} = \frac{K}{T_i} e(k) \quad (2-6)$$

a qual é equivalente a seguinte equação de recorrência:

$$I(k+1) = I(k) + \frac{K}{T_i} e(k) \quad (2-7)$$

- **Aproximação de Tustin**

A aproximação de Tustin, também conhecida como transformação bilinear, nos dá a seguinte equação de recorrência:

$$I(k+1) = I(k) + \frac{KT}{T_i} \frac{e(k+1) + e(k)}{2} \quad (2-8)$$

Observe que a equação (2-8) é equivalente a uma integração numérica utilizando o método dos trapézios.

2.2.5. Ação Derivativa

O termo derivativo pode ser re-escrito da seguinte maneira:

$$\frac{dD}{dt} + Dp = -KT_d \frac{de(t)}{dt} \quad (2-9)$$

Da mesma forma que o termo integral, a equação (2-5) pode ser discretizada segundo vários métodos. Apresentamos abaixo a discretização da ação derivativa utilizando a aproximação por *backward differences* e a aproximação de Tustin.

- **Backward Differences:**

$$\frac{D(k) - D(k-1)}{T} + pD(k) = -KT_d \frac{e(k) - e(k-1)}{T} \quad (2-10)$$

A equação de recorrência é neste caso:

$$D(k) = \frac{1}{1 + pT} D(k-1) - \frac{KT_d T}{1 + pT} (e(k) - e(k-1)) \quad (2-11)$$

- **Aproximação de Tustin:**

Pela aproximação de *Tustin* temos a seguinte equação de recorrência:

$$D(k) = \frac{(pT-2)}{(pT+2)} D(k-1) + \frac{2KT_d}{T(pT+2)} (e(k) - e(k-1)) \quad (2-12)$$

2.2.6. Rotina PID.

A partir do apresentado acima, a rotina para a implementação digital do algoritmo de controle PID, pode ser escrita a partir das equações de recorrência de cada uma das ações. Um exemplo de código básico seria o seguinte:

$$P = K \cdot \text{erro}$$

$$I = I_{\text{ant}} + \{K \cdot T\} \cdot \{2 \cdot T_i\} \cdot (\text{erro} + \text{erro}_{\text{ant}}) / \{2 \cdot T_i\}$$

$$D = (pT-2)/(pT+2) \cdot D_{\text{ant}} + 2 \cdot K \cdot T_d \cdot T \cdot (pT+2) \cdot (\text{erro} - \text{erro}_{\text{ant}})$$

$$\text{sinal_controle} = P + I + D$$

$$\text{erro}_{\text{ant}} = \text{erro}$$

$$D_{\text{ant}} = D$$

2.3. Conversores Analógicos/Digitais

Conversores analógicos para digital (A/D) e digital para analógico (D/A) fornecem uma interface entre o “mundo” analógico e o “mundo” digital computacional, tornando possível uma interface e um controle de sensores e equipamentos via programas de computadores, com uma fácil atualização e ajuste. Entretanto, para isto ser verdade são necessários digitalizadores e controladores rápidos e precisos, compatíveis com o tempo constante dos sistemas controlados.

Sensores inerciais é uma tecnologia madura onde sensores podem alcançar uma resolução de 0.001 partes por milhão (ppm), que está além da resolução e intervalo dinâmico da maioria dos conversores digitais. O intervalo dinâmico necessário é de 0.01°/hora a 400°/hora, que dará uma resolução de 20-bits com uma taxa de saída de 1Hz até 200Hz.

Conceitos Básicos



O projeto de um conversor A/D começa com uma análise dos requisitos do sistema, onde os dois primeiros que precisam ser determinados são:

- Intervalo dinâmico;
- Velocidade da conversão.

Uma característica importante dos sinais analógicos é o ruído no valor analógico que é ou inerente no transdutor, ou no circuito usado para amplificar o sinal analógico. O valor do ruído em um sinal de baixo nível ou a incerteza na medida analógica comparado com o valor de um sinal completo no limite do seu intervalo, determina o intervalo dinâmico de um sinal analógico utilizável. Em outras palavras, nós podemos definir o intervalo dinâmico como a razão entre a maior entrada que pode ser convertida e o menor passo do conversor.

A velocidade de conversão é uma medida de conversões por segundo e está associada com o tempo constante do sistema.

Os outros requisitos que auxiliarão na escolha do conversor A/D são:

- Precisão;
- Características do poder operacional;
- Custo;
- Tamanho;
- Peso;
- Confiabilidade / Complexidade / Sustentabilidade;
- Ambiente operacional.

2.4. Uma Visão Geral da Linguagem C

A Linguagem C foi inventada e implementada primeiramente por *Dennis Ritchie* em um DEC PDP-4 que utilizava o sistema operacional UNIX. C é o resultado de um processo de desenvolvimento que começou com uma linguagem mais antiga, chamada BCPL, que ainda está em uso, em sua forma original, na Europa. BCPL foi desenvolvida por *Martin Richards* e influenciou uma linguagem chamada B, inventada por *Ken Thompson*. Na Década de 1970, B levou ao desenvolvimento de C.

Por muitos anos, de fato, o padrão para C foi a versão fornecida com o sistema operacional UNIX versão 5. Ele é descrito em *The C Programming Language*



Language, de Brian Kernighan e Dennis Ritchie (Englewood Cliffs, N.J.: Prentice Hall, 1978). Com a popularidade dos microcomputadores, um grande número de implementações de C foi criado. Quase que por milagre, os códigos-fontes aceitos por essas implementações eram altamente compatíveis. (Isto é, um programa escrito com um deles podia normalmente ser compilado com sucesso usando-se um outro). Porém, por não existir nenhum padrão, havia discrepância. Para remediar essa situação, o ANSI (*American National Standards Institute*) estabeleceu, no verão de 1983, um comitê para criar um padrão que definiria de uma vez por todas a linguagem C.

C é freqüentemente chamada de linguagem de médio nível para computadores. Isso não significa que C seja menos poderosa, difícil de usar ou menos desenvolvida que uma linguagem de alto nível como BASIC e Pascal, tampouco implica que C seja similar à linguagem *assembly* e seus problemas correlatos aos usuários. C é tratada como uma linguagem de médio nível porque combina elementos de linguagens de alto nível com a funcionalidade da linguagem *assembly*.

Como uma linguagem de médio nível, C permite a manipulação de *bits*, *bytes* e endereços – os elementos básicos com os quais o computador funciona. Um código escrito em C é muito portátil. Portabilidade significa que é possível adaptar um software escrito para um tipo de computador a outro.

Todas as linguagens de programação de alto nível suportam o conceito de tipos de dados. Um tipo de dado define um conjunto de valores que uma variável pode armazenar e o conjunto de operações que pode ser executado com essa variável. Tipos de dados comuns são inteiro, caractere e real. Embora C tenha cinco tipos de dados internos, ela não é uma linguagem rica em tipos de dados como Pascal e Ada. C permite quase todas conversões de tipos. C não efetua nenhuma verificação no tempo de execução, como a validação dos limites das matrizes. Esses tipos de verificações são de responsabilidade do programador.

As versões originais de C não realizam muitos (se é que realizam algum) testes de compatibilidade entre um parâmetro de uma função e o argumento usado para chamar a função. Por exemplo, na versão original de C, você poderia chamar uma função, usando um ponteiro, sem gerar uma mensagem de erro, mesmo que essa função tivesse sido definida, na realidade, como recebendo um argumento em ponto flutuante. No entanto, o padrão ANSI introduziu o conceito de *protótipos de*



funções, que permite que alguns desses erros em potencial sejam mostrados, conforme a intenção do programador.

Outro aspecto importante de C é que ele tem apenas 32 palavras-chaves (27 do padrão de fato estabelecido por *Kernighan* e *Ritchie*, mais cinco adicionais pelo comitê ANSI de padronização), que são os comandos que compõem a linguagem C. As linguagens de alto nível tipicamente têm várias vezes esse número de palavras reservadas. Usando C, um programador pode conseguir aproximadamente a eficiência de código *assembly* combinada com a estrutura de ALGOL ou Modula-2.

O fato de C freqüentemente ser usada em lugar da linguagem *assembly* é o fator mais importante para a sua popularidade entre os programadores. A linguagem *assembly* usa uma representação simbólica do código binário real que o computador executa diretamente. Cada operação em linguagem *assembly* leva a uma tarefa simples a ser executada pelo computador. Embora a linguagem *assembly* dê aos programadores o potencial de realizar tarefas com máxima flexibilidade e eficiência, é notoriamente difícil de trabalhar quando se está desenvolvendo ou depurando um programa. Além disso, como *assembly* não é uma linguagem estruturada, o programa final tende a ser um emaranhado de *jumps*, *calls*, e índices. Essa falta de estrutura torna os programas em linguagem *assembly* difíceis de ler, aperfeiçoar e manter. Talvez mais importante: as rotinas em linguagem *assembly* não são portáveis entre máquinas com unidades centrais de processamento (CPUs) diferentes.

Inicialmente, C era usada na programação de sistemas. Um programa de sistema forma uma porção do sistema operacional do computador ou de seus utilitários de suporte. Por exemplo, os programas que seguem são freqüentemente chamados de programas de sistema:

- Sistemas Operacionais
- Interpretadores
- Editores
- Programas de planilhas eletrônicas
- Compiladores
- Gerenciadores de banco de dados



Em virtude da sua portabilidade e eficiência, à medida que C cresceu em popularidade, muitos programadores começam a usá-la para programa todas as tarefas. Por haver compiladores C para quase todos os computadores, é possível tomar um código escrito para uma máquina, compila-lo e roda-lo em outra com pouca ou nenhuma modificação. Esta portabilidade economiza tempo e dinheiro. Os compiladores C também tendem a produzi um código-objeto muito compacto e rápido.

2.5. *Pêndulo Inverso*

Este tópico tem com objetivo apresentar o projeto de um sistema de controle para um pêndulo Inverso. Primeiro será deduzido um modelo matemático do sistema pêndulo Inverso. Em seguida, será projetado um controlador para manter o pêndulo erguido, em regime estacionário, após os efeitos transitórios (devidos à entrada ou a perturbações) terem se esgotado.

Considere-se o pêndulo Inverso montado num carinho movido a motor, como mostrado na Figura 2.6. Este é um modelo do controle de atitude de um propulsor espacial na decolagem. (O objetivo do sistema de controle de altitude é manter o propulsor numa posição vertical). O pêndulo Inverso é instável na medida em que cairá em qualquer direção a menos que uma força adequada seja aplicada. Considera-se aqui o problema bidimensional, no qual o pêndulo se move somente no plano da página. Admite-se que a massa do pêndulo está concentrada ao final da haste, como mostrado na Figura 2.6. A força de controle de u é aplicada ao carro. Defina-se o ângulo da haste com alinh vertical como θ . (Uma vez que deseja manter o pêndulo Inverso na vertical, o ângulo θ é suposto, ser pequeno). Defina-se também as coordenadas (x,y) do centro de gravidade da massa como (x_G, y_G) . Então,

$$\begin{aligned}x_G &= x + l.\text{sen}\theta \\y_G &= l.\text{cos}\theta\end{aligned}$$

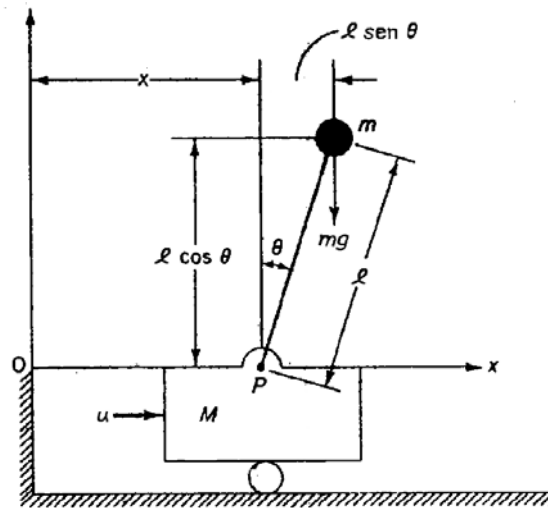


Figura 2. 6 – Análise de Forças do Pêndulo Inverso com Carro.

A aplicação da segunda Lei de Newton na direção x do movimento produz

$$M \frac{d^2 x}{dt^2} + m \frac{d^2 x_G}{dt^2} = u \quad (2-1)$$

ou

$$M \frac{d^2 x}{dt^2} + m \frac{d^2}{dt^2} (x + l \cdot \text{sen } \vartheta) = u \quad (2-2)$$

Notando-se que

$$\frac{d}{dt} \text{sen } \vartheta = (\cos \theta) \dot{\vartheta} \quad (2-3)$$

$$\frac{d^2}{dt^2} \text{sen } \vartheta = -(\text{sen } \theta) \theta^2 + (\cos \theta) \theta \quad (2-4)$$

$$\frac{d}{dt} \cos \theta = -(\text{sen } \vartheta) \theta \quad (2-5)$$

$$\frac{d^2}{dt^2} \cos \theta = -(\cos \theta) \theta^2 - (\text{sen } \theta) \theta \quad (2-6)$$

a Equação (2-2) pode ser escrita como

$$(M + m)x - ml(\text{sen } \theta) \theta^2 + ml(\cos \theta) \theta = u \quad (2-7)$$

A equação de movimento da massa m na direção y não pode ser escrita sem considerar o movimento da massa m na direção x . Por conseguinte, em vez de considerar o movimento da massa m na direção y , considera-se o movimento de

rotação da massa m em torno do ponto P . Aplicando a segunda Lei de Newton ao movimento de rotação, obtém-se:

$$m \frac{d^2 x_G}{dt^2} l \cos \theta - m \frac{d^2 y_G}{dt^2} l \sin \theta = mg l \sin \theta \quad (2-8)$$

ou

$$\left[m \frac{d^2}{dt^2} (x + l \sin \theta) \right] l \cos \theta - \left[m \frac{d^2}{dt^2} (l \cos \theta) \right] l \sin \theta = mg l \sin \theta \quad (2-9)$$

que pode ser simplificada como se segue:

$$m \left[x - l (\sin \theta) (\dot{\theta}^2 + l (\cos \theta) \ddot{\theta}) \right] l \cos \theta - m \left[-l (\cos \theta) (\dot{\theta}^2 - l (\sin \theta) \ddot{\theta}) \right] l \sin \theta = mg l \sin \theta \quad (2-10)$$

Simplificações adicionais resultam em

$$mx \cos \theta + ml \ddot{\theta} = mg \sin \theta \quad (2-11)$$

Claramente, as Equações (2-7) e (2-11) são equações diferenciais não lineares. Como se precisa manter o pêndulo Inverso na vertical, pode-se supor que $\theta(t)$ e $\dot{\theta}(t)$ sejam quantidades pequenas tais que

$$\sin \theta = \theta, \quad \cos \theta = 1 \quad e \quad \theta \dot{\theta} = 0.$$

As Equações (2-7) e (2-8) podem ser linearizadas como se segue:

$$(M + m)x + ml \ddot{\theta} = u \quad (2-12)$$

$$m \ddot{x} + ml \ddot{\theta} = mg \theta \quad (2-13)$$

Essas equações linearizadas são válidas na medida em que θ e $\dot{\theta}$ sejam pequenos. As Equações (2-12) e (2-13) definem um modelo matemático do sistema pêndulo Inverso.

As equações linearizadas do sistema, (2-12) e (2-13) podem ser modificadas para

$$M l \ddot{\theta} = (M + m)g \theta - u \quad (2-14)$$

$$M \ddot{x} = u - mg \theta \quad (2-15)$$

A Equação (2-14) foi obtida eliminando-se \ddot{x} das Equações (2-12) e (2-13). A Equação (2-12) foi obtida eliminando-se $\ddot{\theta}$ das Equações (2-12) e (2-13). Definam-se as variáveis de estado x_1 , x_2 , x_3 e x_4 através de:

$$x_1 = \theta \quad x_2 = \dot{\theta} \quad x_3 = x \quad x_4 = \dot{x}$$



Nota-se que o ângulo θ indica a rotação do pêndulo em torno do ponto P, e x é a localização do carrinho. Considera-se x como a saída do sistema, ou

$$y = x = x_3$$

Por conseguinte, a partir da definição das variáveis de estado e das Equações (2-14) e (2-15), obtém-se:

$$\begin{aligned}x_1 &= x_2 \\x_2 &= \frac{M+m}{Ml} g \cdot x_1 - \frac{1}{Ml} u\end{aligned}\quad (2-16)$$

$$\begin{aligned}x_3 &= x_4 \\x_4 &= -\frac{m}{M} g \cdot x + \frac{1}{M} u\end{aligned}\quad (2-17)$$

Em termos de equações matriciais, tem-se:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{M+m}{Ml} \cdot g & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{m}{M} \cdot g & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \\ \frac{1}{M} \end{bmatrix} u \quad (2-18)$$

$$y = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \quad (2-19)$$

As (2-18) e (2-19) equações fornecem representações no espaço de estados do sistema pêndulo Inverso. (Nota-se que a representação no espaço de estados do sistema não é única. Existe um número infinito de tais representações.).

É desejado manter o sistema pêndulo Inverso tão aprumado quando possível e ainda controlar-se a posição do carrinho, por exemplo, movendo-o sob a forma de impulsos. Para controlar o carrinho, precisa-se construir um servossistema tipo um. O sistema pêndulo Inverso montado no carrinho não possui integrador. Por conseguinte, alimenta-se o sinal de posição y (que indica a posição do carrinho) de volta para a entrada insere-se um integrador no percurso direto de alimentação, como mostrado na Figura 2.7.

Neste exemplo, admitem-se os seguintes valores numéricos para M , m e l :

$$M = 0,5\text{kg} \quad m = 0,007\text{kg} \quad l = 0,5\text{m}$$

Substituindo esses valores, obtém-se:

$$\frac{M+m}{M.l}g = 19,860, \quad \frac{m}{g} = 714,29 \cdot 10^{-6}, \quad \frac{1}{Ml} = 4, \quad \frac{1}{M} = 2$$

As equações (2-18) e (2-19) podem ser escritas como

$$\dot{x} = Ax + Bu$$

$$y = Cx + Du$$

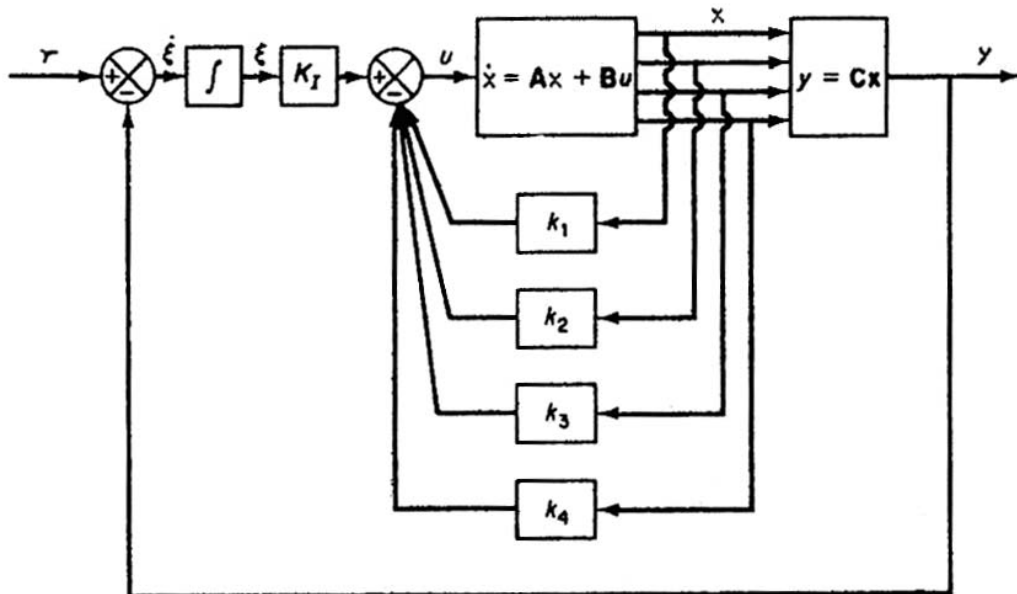


Figura 2. 7 – Diagrama de bloco para pêndulo inverso.

$$\text{onde } A = \begin{bmatrix} 0 & 4 & 0 & 0 \\ 19,860 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 \\ -714,29 \cdot 10^{-6} & 0 & 0 & 0 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 0 \\ -4 \\ 0 \\ 2 \end{bmatrix}, C[0 \ 0 \ 4 \ 0], D=0$$

Referindo à Figura 2.7, as equações para o sistema de controle do Pêndulo Inverso são:

$$x = Ax + Bu \quad (2-20)$$

$$y = Cx \quad (2-21)$$

$$u = -Kx + k_1 \xi \quad (2-22)$$

$$\xi = r - y = r - Cx \quad (2-23)$$

2.6. PWM

PWM é a abreviação de *Pulse Width Modulation* ou Modulação de Largura de Pulso.

Para que se entenda como funciona esta tecnologia no controle de potência, utiliza-se um circuito imaginário formado por um interruptor de ação muito rápida e uma carga que deve ser controlada, de acordo com a figura 2.8.

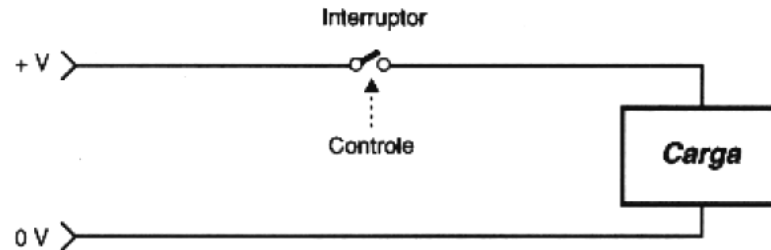


Figura 2. 8 – Abrir e fechar da chave - PWM

Quando o interruptor está aberto não há corrente na carga e a potência aplicada é nula.

No instante em que o interruptor é fechado, a carga recebe a tensão total da fonte e a potência aplicada é máxima.

A idéia é fazer com que a chave seja aberta e fechada rapidamente de modo a ficar 50% do tempo aberta e 50% fechada. Isso significa que, em média, se terá metade do tempo com corrente e metade do tempo sem corrente, veja a figura 2.9.

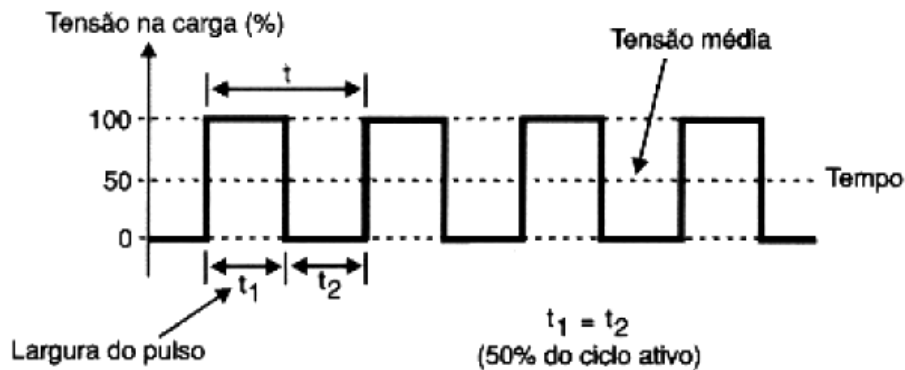


Figura 2. 9 – Gráfico do pulso PWM do abrir e fechar de chaves

A potência média e, portanto, a própria tensão média aplicada à carga é neste caso 50% da tensão de entrada.

O interruptor fechado pode definir uma largura de pulso pelo tempo em que ele fica nesta condição, e um intervalo entre pulsos pelo tempo em que ele fica aberto. Os dois tempos juntos definem o período e, portanto, uma frequência de controle.

A duração de um ciclo completo de operação do interruptor nos define ainda o ciclo ativo, conforme é mostrado na figura 2.10.

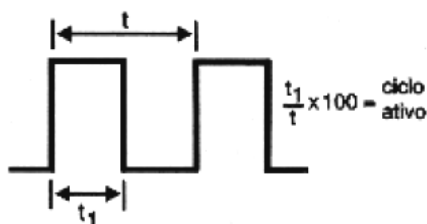


Figura 2. 10 – Gráfico do ciclo ativo - PWM

Variando-se a largura do pulso e também o intervalo de modo a ter ciclos ativos diferentes, é possível controlar a potência média aplicada a uma carga. Assim, quando a largura do pulso varia de zero até o máximo, a potência também varia na mesma proporção, conforme está indicado na figura 2.11.

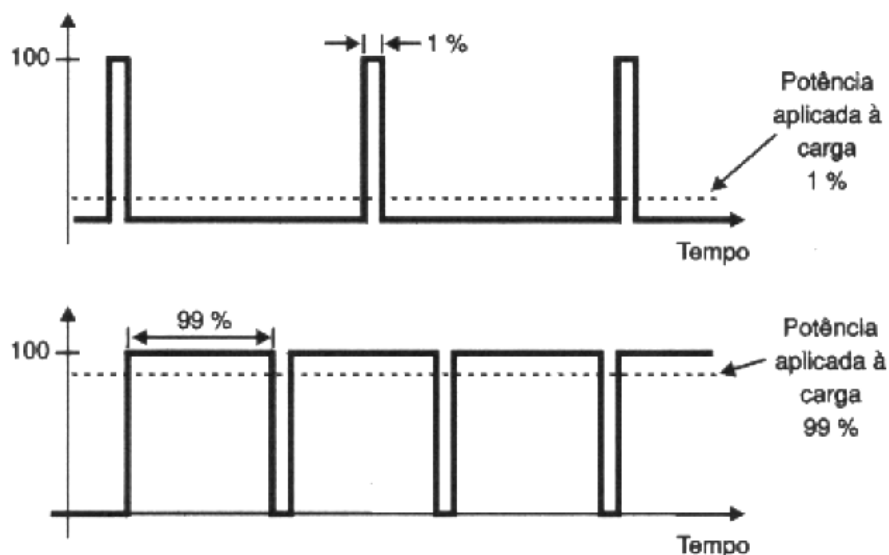


Figura 2. 11 – Controlando a potencia pelo ciclo ativo

Este princípio é usado justamente no controle PWM: modulamos (variemos) a largura do pulso de modo a controlar o ciclo ativo do sinal aplicado a uma carga e, com isso, a potência aplicada a ela.

2.6.1. TIPOS DE PWM

O exemplo de aplicação é o que se denomina de “simples magnitude PWM”, onde o sinal aplicado à carga determina simplesmente a potência que ela deve receber, pela largura do pulso.

No entanto, existe um segundo tipo de controle PWM denominado “*Locked anti-phase PWM*”, que pode incluir na modulação do sinal informações sobre a potência aplicada à carga e o sentido da corrente que deve circular por ela. Este tipo de controle, em especial, é interessante quando se trata de motores elétricos onde o sentido da corrente determina o sentido da rotação ou do torque.

Se os pulsos aplicados à carga não variarem entre 0V e um valor máximo de tensão +V, por exemplo, mas entre uma tensão negativa e uma tensão positiva (-V a +V), o ciclo ativo determina também o sentido de circulação da corrente pela carga.

Se, por exemplo, o ciclo ativo for de 50% conforme mostra à figura 2.12, a tensão média na carga e, portanto a potência será zero.

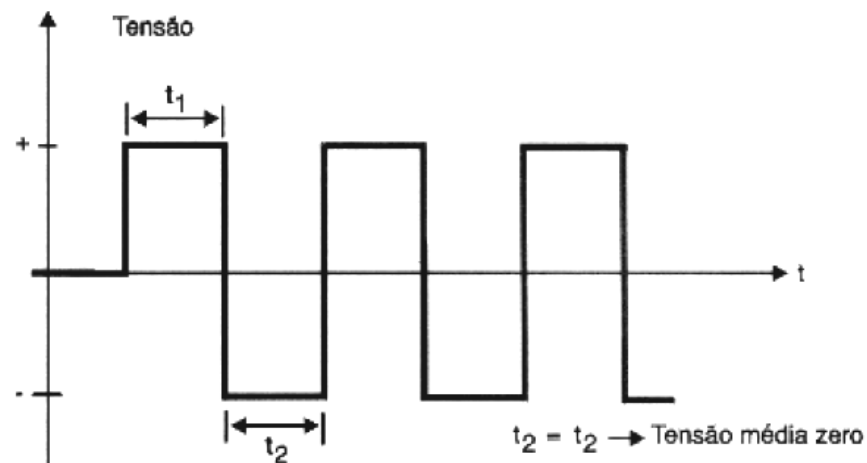


Figura 2. 12 – **Anti-phase PWM**

Agora, variando o ciclo ativo para mais e para menos de 50%, teremos uma predominância dos pulsos positivos ou negativos de modo que a corrente média tende a circular num sentido ou noutro, de acordo com a mesma figura.

Logo, neste tipo de circuito, a corrente na carga variará entre -100% e +100%, conforme o ciclo ativo do sinal aplicado.

Um circuito simples de aplicação para este tipo de controle é fornecido na figura 2.13.

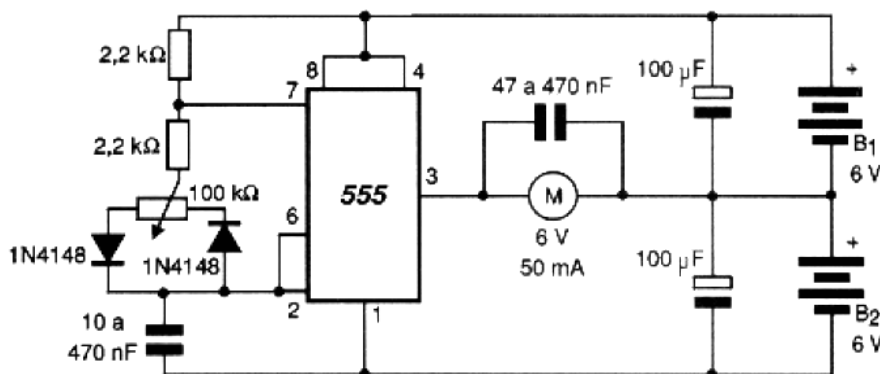


Figura 2. 13 – Esquema pratico de um PWM

Usa-se uma fonte simétrica de +6V/-6V para controlar um pequeno motor de 50mA a partir de um integrado LM555. Uma etapa de potência com transistores pode ser acrescentada a este circuito, para o uso com motores de maior corrente.

O potenciômetro ajusta tanto a largura como os intervalos entre os pulsos de modo que a carga e descarga do capacitor sejam derivadas por diodos diferentes, agindo assim no ciclo ativo do sinal de saída.

Um ponto importante que deve ser observado neste tipo de circuito é que na posição de 50% de ajuste do potenciômetro (potência média nula na carga), na verdade temos uma corrente circulando o tempo todo por ela, o que vai causar dissipação de calor.

Assim sendo, para cargas elevadas, este tipo de controle não é dos mais indicados e não funcionaria.

Mesmo no caso de motores DC é preciso ter muito cuidado na escolha da frequência de operação do circuito para que na condição de parado (0% de potência) ele não se mantenha vibrando na frequência do oscilador. Eventualmente componentes adicionais podem ser previstos em paralelo com o motor como, por exemplo, um capacitor, para evitar este problema.

2.6.2. VANTAGENS DO PWM

Na operação de um controle por PWM existem diversas vantagens a serem consideradas e alguns pontos para os quais se deve ficar atento para não jogar fora estas vantagens.

Na condição de aberto, nenhuma corrente circula pelo dispositivo de controle e, portanto, sua dissipação é nula. Na condição de fechado, teoricamente,

se ele apresenta uma resistência nula, a queda de tensão é nula, e ele não dissipa também nenhuma potência.

Isso significa que, na teoria, os controles PWM não dissipam potência alguma e, portanto, consistem em soluções ideais para este tipo de aplicação.

Na prática, entretanto, isso não ocorre. Em primeiro lugar, os dispositivos usados no controle não são capazes de abrir e fechar o circuito num tempo infinitamente pequeno. Eles precisam de um tempo para mudar de estado e, neste intervalo de tempo, sua resistência sobe de um valor muito pequeno até infinito e vice-versa, numa curva de comutação semelhante a mostrada na figura 2.14.

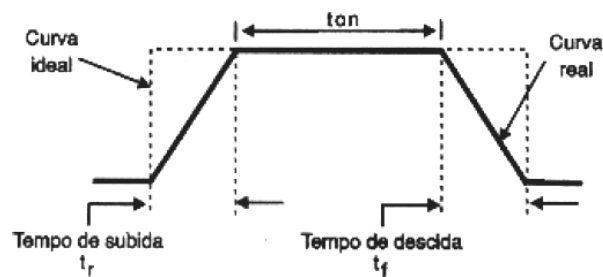


Figura 2. 14 – **Gráfico da variação de calor no pulso**

Neste intervalo de tempo a queda de tensão e a corrente através do dispositivo não são nulas, e uma boa quantidade de calor poderá ser gerada conforme a carga controlada.

Dependendo da frequência de controle e da resposta do dispositivo usado, uma boa quantidade de calor poderá ser gerada neste processo de comutação.

Entretanto, mesmo com este problema, a potência gerada num controle PWM ainda é muito menor do que num circuito de controle linear equivalente. Transistores de comutação rápidos, FETs de potência, e outros componentes de chaveamento podem ser suficientemente rápidos para permitir que projetos de controles de potências elevadas sejam implementados sem a necessidade de grandes dissipadores de calor ou que tenham problemas de perdas de energia por geração de calor que possam ser preocupantes.

O segundo problema que poderá surgir vem justamente do fato de que os transistores de efeito de campo ou bipolares usados em comutação não se comportam como resistências nulas, quando saturados.



Os transistores bipolares podem apresentar uma queda de tensão de até alguns volts quando saturados, o mesmo ocorrendo com os FETs.

Deve-se observar em especial o caso dos FETs de potência que são, às vezes, considerados comutadores perfeitos, com resistências de fração de ohm entre o dreno e a fonte quando saturados ($R_{ds(on)}$) mas na prática não é isso que ocorre.

A baixíssima resistência de um FET de potência quando saturado (resistência entre dreno e fonte no estado *on*) só é válida para uma excitação de comporta feita com uma tensão relativamente alta.

3. MATERIAIS E MÉTODOS

3.1. Descrição Geral do Sistema

O projeto consiste em três partes eletrônicas principais:

- Circuito do microcontrolador dsPIC.
- Circuito da Ponte H
- Motores

Com estas três partes, juntamente com a parte mecânica, obtém-se a montagem de todo o esquema do Pêndulo Inverso.

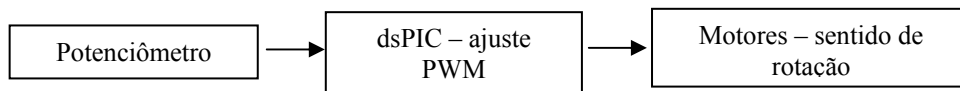


Figura 3. 1 – **Diagrama de Blocos – Sistema de Controle e Pêndulo.**

O sistema foi desenvolvido sobre uma base com dois motores em linha como mostrado na figura 3.2, equipado com uma placa micro processada controlando a rotação dos motores no sentido horário e anti-horário, colocando em equilíbrio a haste e o sistema como um todo. Apêndice E foto do pêndulo construído.

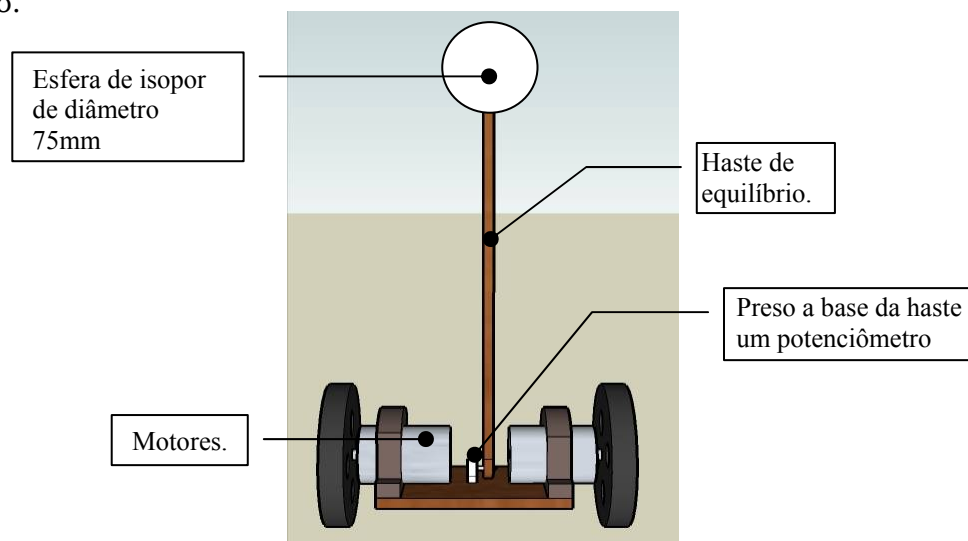


Figura 3. 2 – **Modelo do Pêndulo.**

3.2. Descrição dos Sistemas Mecânicos

As partes mecânicas constituídas no projeto são dois motores CC de 5200 rpm alimentados por uma fonte de tensão de 12V. Presa á estrutura está uma haste, e na base da mesma se encontra um potenciômetro informando para o circuito controlador qual a posição atual da haste, conforme sua situação o controlador fará mudar o sentido de giro dos motores.

3.3. Descrição dos Sistemas Eletroeletrônicos

3.3.1. Hardware

- *dsPic*

Foi montado em uma placa de matriz de contatos o circuito para o microprocessador, no projeto foi utilizado o *dsPIC30f4011*. Este microcontrolador têm em sua interface comandos pré-definidos para o uso de PWM, que podem ser utilizados no controle de motores. Na figura 3.3 é informado a escala de desempenho entre as famílias de microcontroladores.

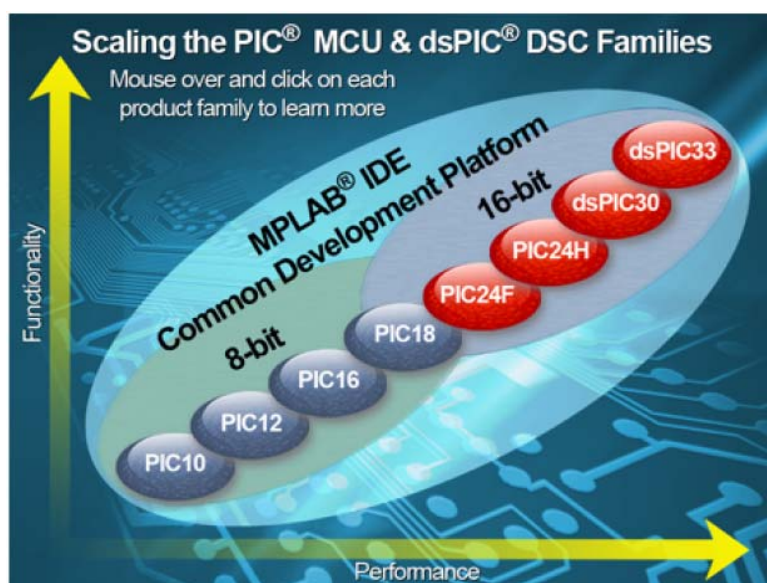


Figura 3. 3 – Escala de desempenho.

Fonte: *Microchip Technology Inc. (www.microchip.com)*

Os microcontroladores PIC estão classificados em famílias, cada qual com uma característica relativa ao seu desempenho e funcionalidade. A família PIC10, de menos recursos, é aplicada a funções de controle *on-off* mais simples e de menor porte, e a família *dsPIC30* e *dsPIC33* são adequadas para processamento e controle envolvendo aquisição, tratamento e processamento veloz de sinais analógicos,

permitindo desenvolvimento de aplicações mais complexas, controles em tempo real de alta velocidade, entre outras.

Este microcontrolador (*dsPIC30f4011*) foi utilizado para se adquirir conhecimento em novas tecnologias no mercado e por ter as características necessárias para o melhor desempenho do projeto.

- **Ponte H**

Apesar de não ser tão conhecida, um dos circuitos mais importantes na elaboração de sistemas automatizados é a ponte H. Trata-se de um circuito utilizado para controlar um motor DC a partir de sinais gerados por um microcontrolador. Devido à disposição dos seus componentes, torna-se extremamente fácil selecionar o sentido da rotação de um motor, apenas invertendo a polaridade sobre seus terminais. Também é importante para a utilização com circuitos digitais, pois como os sinais de saída dos microcontroladores não suportam a corrente necessária e nem possuem a tensão adequada para acionar um motor, é necessária uma unidade de potência que possa alimentá-lo convenientemente.

Uma ponte H básica é composta por quatro chaves mecânicas ou eletrônicas posicionadas formando a letra “H”, sendo que cada uma localiza-se num extremo e o motor é posicionado no meio.

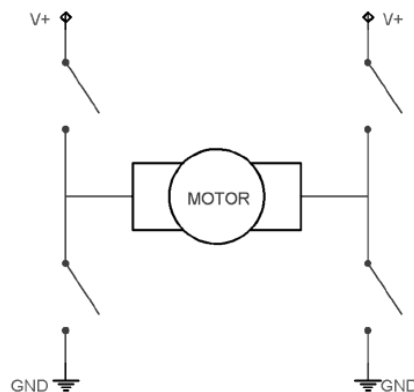


Figura 3. 4 – Representação de uma ponte H.

Para que o motor funcione, basta acionar um par de chaves diagonalmente opostas, o que faz com que a corrente flua do pólo positivo para o negativo atravessando o motor e fazendo-o girar. Para inverter a rotação, desligam-se essas

chaves e aciona-se o outro par de chaves, o que faz com que a corrente siga na direção oposta e, conseqüentemente, o sentido da rotação do motor será alterado.

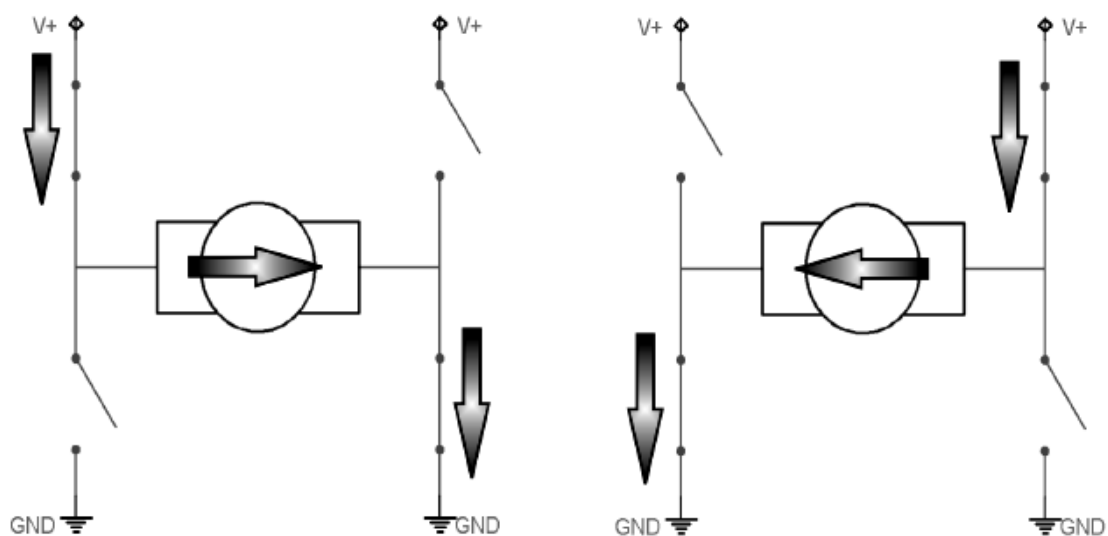


Figura 3. 5 – Representação do uso de uma ponte H.

Caso as duas chaves superiores sejam acionadas, se tem um eficiente mecanismo de freio, o mesmo ocorre se as duas chaves inferiores forem acionadas. Isso porque o motor, como todo componente indutivo, gera uma tensão entre os seus terminais. Quando os terminais do motor são conectados no mesmo pólo, se faz um curto-circuito no motor, e a tensão que gerada pelo motor força-o a girar na direção oposta, fazendo com que ele pare instantaneamente. Mas se todas as chaves forem fechadas, o circuito será desligado e caso o motor estiver girando, ele parará suavemente.

No projeto foi utilizado um integrado *L298*, pode ser visualizado seu formato na figura 3.6. Convenientemente faz o papel de inversão de giro dos motores. Este integrado é de forma bem pratica utilizado sem a necessidade de utilizar componentes discretos para tal fim.

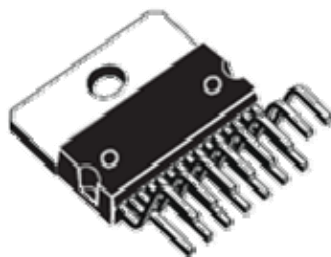


Figura 3. 6 – vista do integrado LM298 (ponte H)

- **Descrição do gravador para microcontrolador**

Para a gravação e programação do microcontrolador foi utilizado o *MicroICD* da *Microgenios*, com este gravador tem a possibilidade de gravar no próprio circuito onde será usado o *dsPIC*. Na figura 3.7 está a visualizado do gravador e na figura 3.8 esta um esquema básico de gravação para o *dsPIC*.



Figura 3. 7 – Gravador *MicroICD* da *Microgenios*

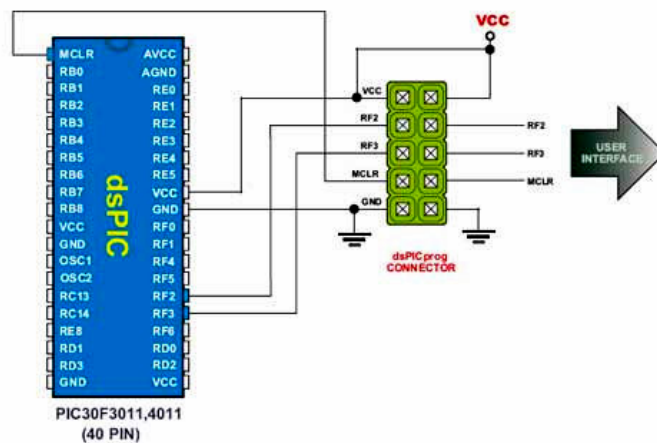


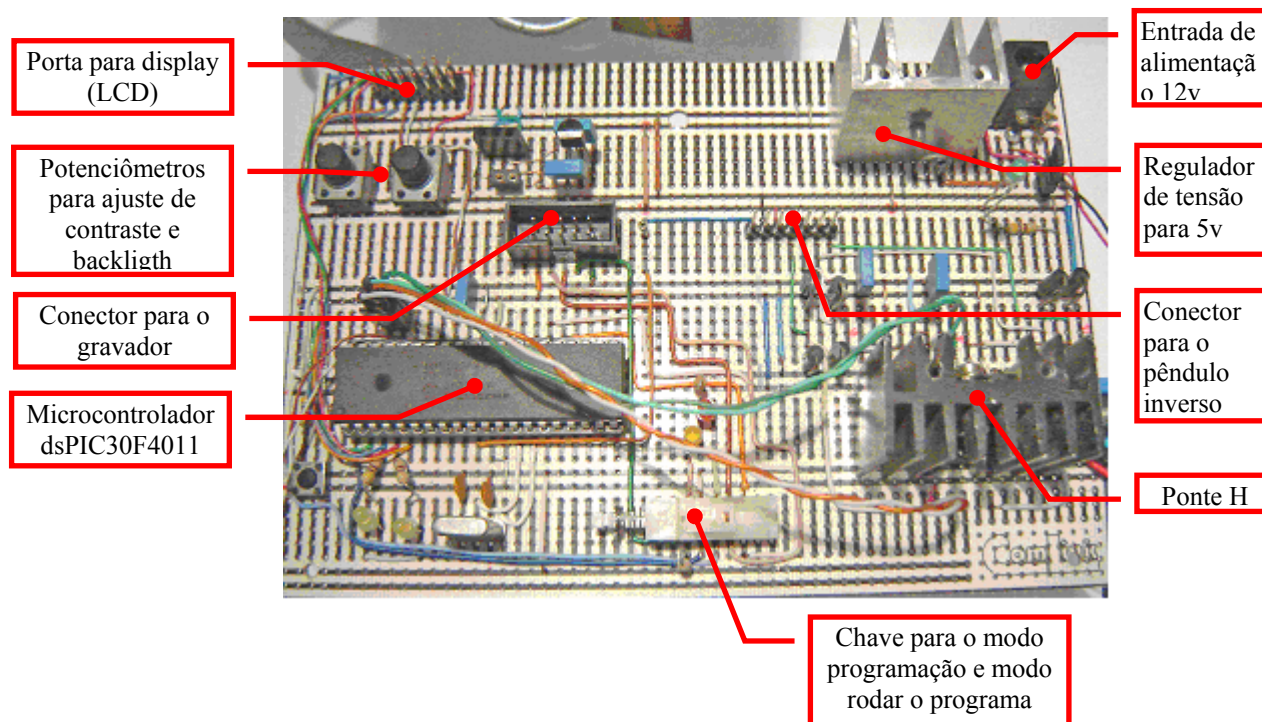
Figura 3. 8 – esquema básico para gravação dsPIC

3.3.2. Descrição

O conjunto do projeto funciona da seguinte forma, é gerado um programa em linguagem de programação C em um software compilador que mais adiante será descrito. Este programa gerado é gravado no microcontrolador, feito teste de execução e analisado o comportamento do pêndulo.

3.3.3. Esquemas Elétricos e PCI

Abaixo, a figura mostra uma imagem da placa controladora do pêndulo inverso com a identificação das suas partes. O esquema elétrico está demonstrado no apêndice B.



3.3.4. Descrição dos Testes Realizados no Hardware

Os testes realizados foram em cima dos motores CC para verificar o rendimento de torque para a devida função de equilibrar o pêndulo inverso. A seguir serão descritos os ensaios e cálculos para fins de constatação a boa resposta dos motores.

Para um bom entendimento será realizada uma breve descrição do cálculo de torque do motor.

- **Ensaio da máquina de corrente contínua a vazio como gerador**

Este ensaio foi realizado no laboratório de automotiva da universidade, onde foi utilizado um torno para acionamento da máquina CC como gerador. Colocando o motor no torno é possível medir a tensão de ensaio a vazio. A seguir são mostrados os valores de tensão para as seguintes rotações do torno.



Tabela 3. 1 – Ensaio a vazio como gerador.

n = Rotação [rpm]	Tensão [V]
50	0,11
100	0,21
200	0,43
400	0,85

Com estes valores de tensão medidas, é possível calcular constantes para o cálculo do torque e do rendimento.

Tabela 3. 2 – Cálculo das constantes k1 e k2

n = Rotação [rpm]	Tensão [V]	K1	k2
50	0,11	0,0021	0,020200435
100	0,21	0,0021	0,020384075
200	0,43	0,0021	0,020384075
400	0,85	0,0021	0,020338165
	Média	0,0021	0,0204

Os valores das constantes k1 e k2 são calculados da seguinte forma:

$$k_1 = \frac{E}{n} \quad k_2 = k_1 \cdot \frac{60}{2 \cdot \pi}$$

Onde E é o valor da *fem* induzida no gerador, e n é o valor de rotação em rotações por minuto.

Para os cálculos seguintes serão usados os valores de k1 e k2 médios.

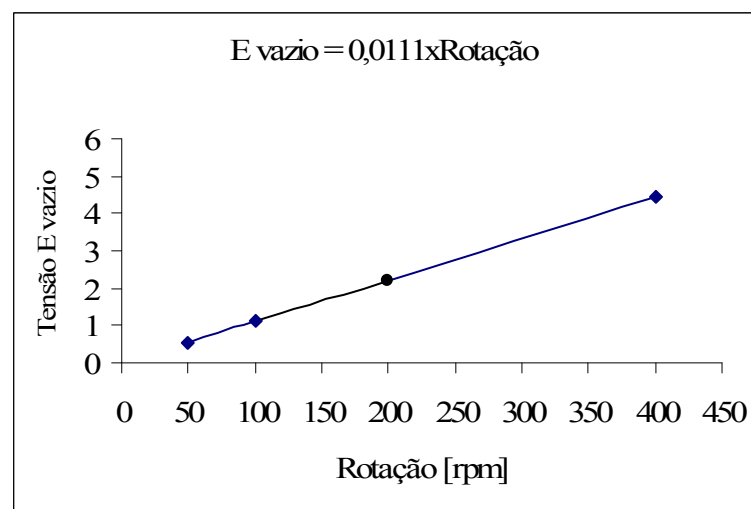


Figura 3. 9 – gráfico de ensaio do motor como gerador.

- **Ensaio de rotor bloqueado**

Neste ensaio foram medidos os valores de tensão e corrente para o rotor bloqueado. Segue a tabela com os valores:

Tabela 3. 3 – Ensaio de rotor bloqueado.

Tensão na Carga [V]	Corrente na Carga [A]	Resistência Total [Ω]
2,6	2,25	1,16

- **Ensaio de motor a vazio**

No ensaio realizado a vazio foram medidos os valores de tensão e corrente. Com estes valores, calculou-se o valor da (E_{ca}) força contra eletromotriz induzida, (n) rotação a vazio e (T) torque resistente. Abaixo estão as fórmulas para calcular essas três grandezas:

$$E_{ca} = V - (R_{total} \cdot I) \quad (3-1)$$

$$n = \frac{E_{ca}}{k_1} \quad (3-2)$$

$$Torque_{resistente} = \frac{(I \cdot E_{ca})}{n \cdot \left(\frac{2 \cdot \pi}{60}\right)} \quad (3-3)$$

Tabela 3. 4 – Cálculo de força contra eletromotriz, rotação e torque resistente.

Corrente [A]	Tensão [V]	Eca [V]	n =Rotação [rpm]	Torque Resistente [Nm]
0,65	12	11,25	5276	0,01323473

- **Cálculo de rendimento**

Depois de realizados os três ensaios foram feitos os cálculos para verificar o rendimento e a assim poder avaliar a eficiência dos motores para o uso no projeto.

A seguir são mostradas as fórmulas usadas para gerar a tabela.

$$Torque_{carga} = fator \cdot k_2 \quad (3-5)$$

$$Torque_{mediovazio} = Torque_{resistente} = \frac{(I \cdot E_{ca})}{n \cdot \left(\frac{2 \cdot \pi}{60}\right)} \quad (3-6)$$

$$n = \left(\frac{V}{k_1}\right) - \left[\left(\frac{R_f}{k_1 \cdot k_2}\right) \cdot (Torque_{carga} + Torque_{mediovazio})\right] \quad (3-7)$$



$$I_{carga} = \frac{V - (n \cdot k_1)}{R_{total}} \quad (3-8)$$

$$P_{ot.mecânica} = Torque_{carga} \cdot \frac{n \cdot \pi}{30} \quad (3-9)$$

$$Rendimento(\%) = 100 \cdot \frac{P_{ot.mecânica}}{V \cdot I_{carga}} \quad (3-10)$$

A tabela a seguir mostra os valores de torque na carga que precisam ser desenvolvidos pelo motor, o número de rotação (n) decai à medida que o torque na carga aumenta e consequentemente a corrente na bobina aumenta.

Tabela 3. 5 – Cálculo de rendimento.

Torque Carga [Nm]	n = Rotação [rpm]	I Carga [A]	Pot. Mecânica [W]	Fator	Rendimento (%)
0,0000	5276	0,65	0,00	0,00	0,00
0,0204	4734	1,65	10,09	1,00	50,98
0,0407	4192	2,65	17,88	2,00	56,21
0,0611	3650	3,65	23,35	3,00	53,30
0,0814	3108	4,65	26,51	4,00	47,50
0,1018	2566	5,65	27,36	5,00	40,35
0,1222	2024	6,65	25,89	6,00	32,45
0,1425	1482	7,65	22,12	7,00	24,10
0,1629	940	8,65	16,04	8,00	15,45
0,1833	398	9,65	7,64	9,00	6,60
0,2036	-144	10,65	-3,07	10,00	-2,40

3.4. Descrição dos Sistemas Informáticos e Computacionais

3.4.1. Software

O programa utilizado para o desenvolvimento do projeto, escolhido entre vários foi o *MikroC_dsPIC*, pois utiliza uma base de bibliotecas definidas para os próprios microcontroladores da família dsPIC.

O software gerado funciona independente sem precisar ser feita alguma configuração. Depois de o programa ser gerado ele é carregado diretamente ao microcontrolador pelo gravador *MicroICD*.

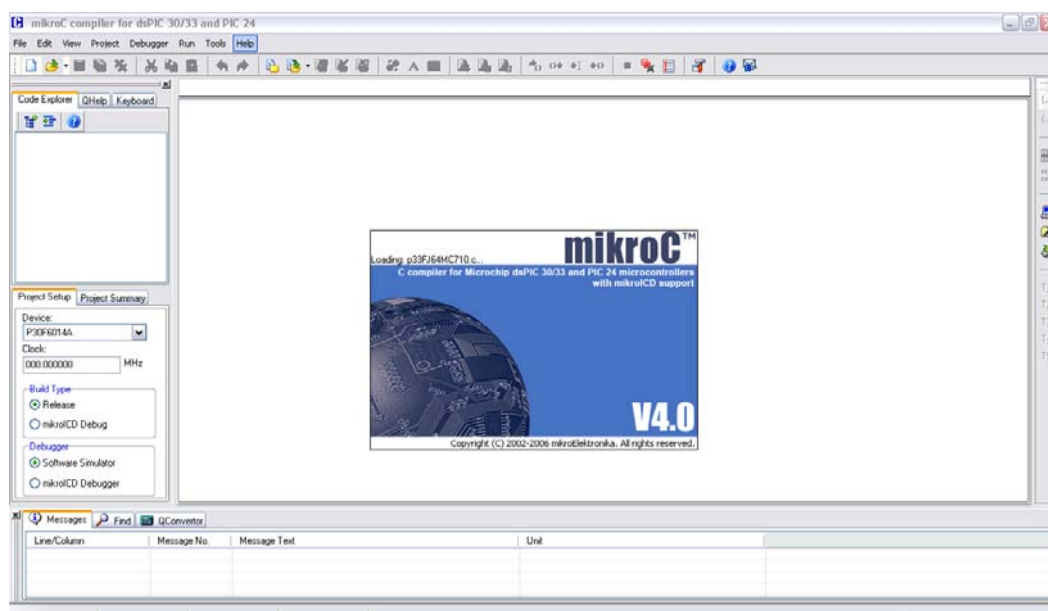


Figura 3. 10 – tela do software compilador.

3.4.2. Interfaces Gráficas e Fluxogramas (UML)

O software de controle do pêndulo inverso foi estruturado da seguinte forma:

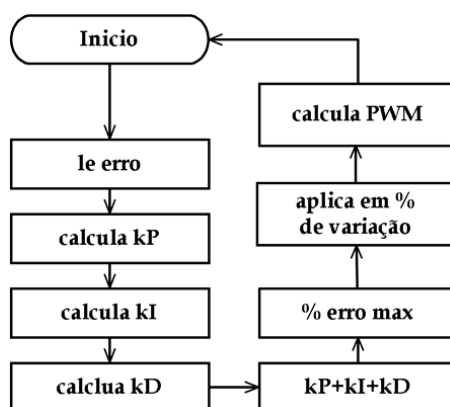


Figura 3. 11 – Fluxograma, Software Pêndulo Inverso.

3.4.3. Descrição das Classes e Funções

O software está dividido nas seguintes funções:

- Main : Estrutura principal do software da placa, nesta função é configurada e realizada a leitura do canal AD, juntamente com a média destes, e realiza a chamada das funções kP, kI, kD; conversão do sinal lido para o cálculo do *dutycycle*.
- duty_50: Esta função foi usada para definir o cálculo do *dutycycle*.



- Pwm_Mc_Init: Esta função inicia a função de PWM do microcontrolador, nela são definidas as seguintes configurações: frequência do PWM em Hz, defini o tipo de saída das portas como complementar ou não, habilita quais portas devem ser usadas e por ultimo define o tipo de *clock*.
- Pwm_Mc_Set_Duty: função muda o PWM razão do *duty*.
- Pid_out: é cálculo da saída do PID.
- Lcd_Custom_Config: configura as portas do microcontrolador para o uso de um LCD (display).

3.4.4. Descrição da Funcionalidade do Software

O software tem a seguinte função de receber a leitura do conversor analógico/digital com este valor é possível fazer o cálculo do ajuste proporcional integral e derivativo. Ao fazer o devido ajuste o programa envia o sinal na saída da porta do microcontrolador, esta porta já tem uma configuração própria em software para o PWM. Em seguida o sinal é enviado ao circuito da ponte H possibilitando o controle dos motores mudando o sentido de rotação horário e anti-horário.

4. APRESENTAÇÃO E DISCUSSÃO DOS RESULTADOS

4.1. Modelo matemático para simulação no MatLab

Neste capítulo será mostrada a modelagem matemática de um carro genérico com características muito próximas ao real, utilizando o software MatLab.

Na figura abaixo vemos o carrinho pêndulo inverso, impulsionado por uma força, F . As equações de movimento dinâmico para o sistema, e linearização sobre o ângulo do pêndulo, $\theta = \text{Pi}$ (em outras palavras, assumir que o pêndulo não passe mais de alguns graus de distância da vertical, escolhida para ser a um ângulo de Pi). Calcular um controle PID para satisfazer todos os requisitos do modelo abaixo indicado.

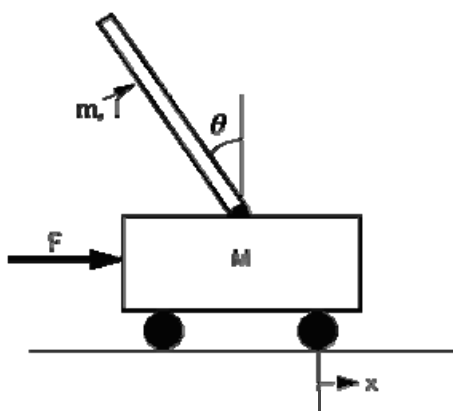


Figura 4. 1 - Componentes do pendulo inverso.

Para este exemplo, vamos assumir que:

M	Massa do carrinho	0.5 kg
m	Massa do pêndulo	0.2 kg
b	Fricção do carrinho	0.1 N/m/sec
l	Comprimento de pêndulo centro de massa	0.3 m
I	Inércia do pêndulo	0.006 kg*m ²
F	Força aplicada ao carrinho	1 N
Theta (θ)	Ângulo vertical do pêndulo	rad

Para o PID, se esta apenas interessada no controle da posição do pêndulo. Portanto, não será levada em conta a posição do carrinho. Para esses pontos, será suposto que o sistema inicia em equilíbrio, e a força F será igual a 1N. O pêndulo deve voltar à sua posição vertical dentro de 5 segundos, e nunca passar mais de 0,05 radianos de distância da linha vertical.

Os requisitos de concepção deste sistema são:

- Resolução de tempo de menos de 5 segundos.
- O ângulo do Pêndulo nunca deve ser superior a 0,05 radianos em relação à vertical.

Assim, para esta seção do pêndulo inverso será tentar controlar tanto o ângulo do pêndulo e da posição do carro. Para tornar o projeto mais próximo ao real será aplicado um passo de entrada para o carrinho. O carrinho deve atingir a sua posição desejada dentro de 5 segundos e têm um tempo de subida em 0,5 segundos. Também terá de ser limitado o ângulo em 20 graus (0,35 radianos), e também deve resolver em menos de 5 segundos.

Os requisitos para a modelagem do pêndulo inverso são os seguintes:

- Tempo de estabilização – t_s (*settling time*) para x e ângulo teta tempo para menos de 5 segundos.
- Tempo de subida – t_r (*rise time*) x para menos de 0,5 segundos.
- Overshoot para teta com menos de 20 graus (0,35 radianos).

Abaixo estão os dois diagramas de corpo livre do sistema.

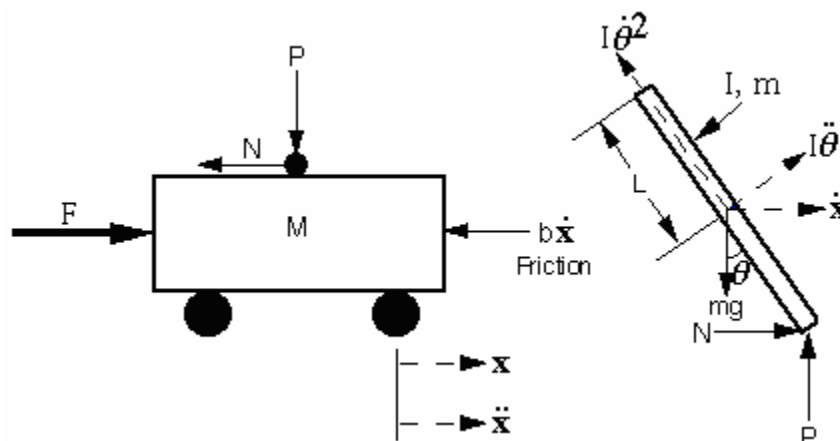


Figura 4. 2 – Componentes vetoriais do pêndulo inverso.



Resumindo as forças do diagrama de corpo livre do carrinho na direção horizontal, obtém-se a seguinte equação de movimento:

$$M\ddot{x} + b\dot{x} + N = F \quad (4-1)$$

Poderiam ser somadas as forças na direção vertical, mas nenhuma informação útil seria adquirida.

Resumindo as forças do diagrama de corpo livre do pêndulo na direção horizontal, pode-se obter uma equação para N:

$$N = m\ddot{x} + ml\ddot{\theta} \cos \theta - ml\dot{\theta}^2 \sin \theta \quad (4-2)$$

Se substituir esta equação na primeira equação, obtém-se a primeira equação do movimento para este sistema:

$$(M + m)\ddot{x} + b\dot{x} + ml\ddot{\theta} \cos \theta - ml\dot{\theta}^2 \sin \theta = F \quad (4-3)$$

Para obter a segunda equação de movimento, a soma das forças perpendiculares ao pêndulo. Resolver o sistema ao longo deste eixo pode-se poupar tempo em um monte de álgebra. Assim deve-se obter a seguinte equação:

Para se livrar do N e P termos da equação acima, soma dos momentos em torno do centróide do pêndulo obter a seguinte equação:

$$P \sin \theta + N \cos \theta - mg \sin \theta = ml\ddot{\theta} + m\ddot{x} \cos \theta \quad (4-4)$$

Combinando estas duas últimas equações, obtém-se a segunda equação dinâmica:

$$(I + ml^2)\ddot{\theta} + mgl \sin \theta = -ml\ddot{x} \cos \theta \quad (4-5)$$

MATLAB só pode trabalhar com funções lineares, este conjunto de equações deve ser linear sobre $\theta = \pi$. Supor que $\theta = \pi + \phi$ (ϕ representa um pequeno ângulo da direção vertical ascendente). Assim, $\cos(\theta) = -1$, $\sin(\theta) = -\phi$, e $(d(\theta) / dt)^2 = 0$. Após a linearização duas equações de movimento geradas (onde u representa a entrada):

$$(I + ml^2)\ddot{\phi} - mgl\phi = ml\ddot{x} \quad (4-6)$$

$$(M + m)\ddot{x} + b\dot{x} - ml\ddot{\phi} = u \quad (4-7)$$



Função Transferência

Para obter a função de transferência do sistema de equações linearizada analiticamente, temos de gerar a primeira transformada de Laplace do sistema de equações. As transformadas de Laplace são:

$$(I + ml^2)\Phi(s)s^2 - mgl\Phi(s) = mlX(s)s^2 \quad (4-8)$$

$$(M + m)X(s)s^2 + bX(s)s - ml\Phi(s)s^2 = U(s) \quad (4-9)$$

Uma vez que será analisada a função em relação ao ângulo Phi, então resolver a primeira equação para X (s),

$$X(s) = \left[\frac{(I + ml^2)}{ml} - \frac{g}{s^2} \right] \Phi(s) \quad (4-10)$$

em seguida, substituir na segunda equação, e re-organizar.

$$(M + m) \left[\frac{(I + ml^2)}{ml} - \frac{g}{s} \right] \Phi(s)s^2 + b \left[\frac{(I + ml^2)}{ml} - \frac{g}{s} \right] \Phi(s)s - ml\Phi(s)s^2 = U(s) \quad (4-11)$$

A função de transferência é:

$$\frac{\Phi(s)}{U(s)} = \frac{\frac{ml}{q}s^2}{s^4 + \frac{b(I + ml^2)}{q}s^3 - \frac{(M + m)mgl}{q}s^2 - \frac{bmgl}{q}s} \quad (4-12)$$

onde,

$$q = [(M + m)(I + ml^2) - (ml)^2] \quad (4-13)$$

A partir da função de transferência acima pode ser visto que há tanto um pólo e um zero na origem. Estes podem ser canceladas e função de transferência torna-se:

$$\frac{\Phi(s)}{U(s)} = \frac{\frac{ml}{q}s}{s^3 + \frac{b(I + ml^2)}{q}s^2 - \frac{(M + m)mgl}{q}s - \frac{bmgl}{q}} \quad (4-14)$$



MATLAB representação a resposta de malha aberta

A função de transferência é encontrada a partir da transformada de Laplace pode ser criado usando MATLAB introduzindo o numerador e o denominador em vetores. Foi criado um arquivo m-file com o seguinte texto para modelar a função de transferência:

```
M = 0.5; m = 0.2; b = 0.1; i = 0.006; g = 9.8; l = 0.3;
q = (M+m)*(i+m*l^2)-(m*l)^2; %simplifies input
num = [m*l/q 0];
den = [1 b*(i+m*l^2)/q -(M+m)*m*g*l/q -b*m*g*l/q];
pend=tf(num,den)
```

Após inserir estes foi feitos os seguintes passos - pressionar F5, salvar o arquivo e pressionar novamente a tecla F5 para compilar o programa, na tela principal do MatLab mostrada a resposta final da função transferência.

Sua saída será:

Função de transferência:

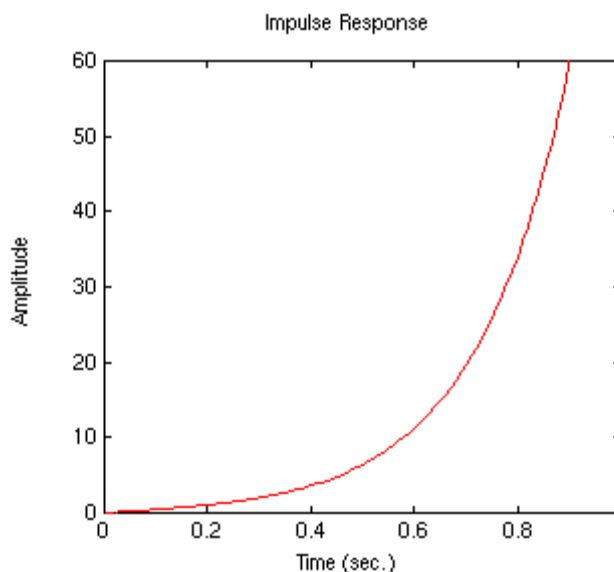
4.545 s

 $s^3 + 0.1818 s^2 - 31.18 s - 4.455$

Para observar a velocidade do sistema de resposta a um impulso aplicada ao carrinho, adicionam-se as seguintes linhas no final do m-file:

```
t=0:0.01:5;
impulse(pend,t)
axis([0 1 0 60])
```

Deve-se obter a seguinte velocidade resposta parcela:





Como se pode ver a partir da parcela, a resposta é totalmente insatisfatória. Não é estável em malha aberta.

Embora a amplitude de saída aumente nos últimos 60 radianos (10 rotações), o modelo é válido apenas para as pequenas θ . O pêndulo vai parar de girar quando se atinge o carrinho ($\theta=90$ graus).

4.2. Simulações – Métodos Numéricos

Solução para o problema utilizando pêndulo inverso, Controle PID.

A função de transferência da planta para este problema é dada abaixo:

$$\frac{\Phi(s)}{U(s)} = \frac{\frac{ml}{q}s}{s^3 + \frac{b(I+ml^2)}{q}s^2 - \frac{(M+m)mgl}{q}s - \frac{bmg l}{q}}$$

onde,

$$q = \left[(M+m)(I+ml^2) - (ml)^2 \right]$$

A concepção critérios (com o pêndulo que recebe um impulso de 1N a partir do carro) são:

- Resolução de tempo de menos de 5 segundos.
- Pêndulo não deve se mover mais do que 0,05 radianos afastados da vertical.

Representação de laço aberto

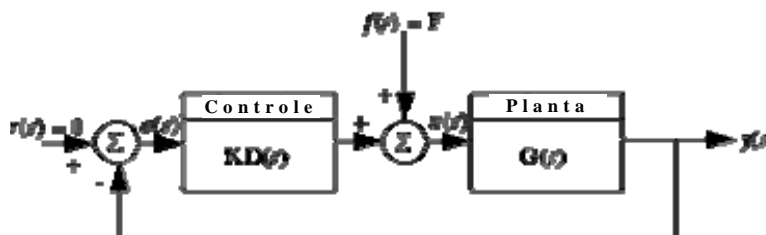
A primeira coisa a fazer quando se utiliza controle PID em MATLAB é encontrar a função de transferência do sistema e verificar para ver se isso faz sentido. A função de transferência encontrados a partir da transformada de Laplace para a saída Phi (o ângulo do pêndulo) pode ser criada utilizando MATLAB, introduzindo o numerador e o denominador em forma de vetores. Criar um m-file (ou um 'M' arquivo localizado no mesmo diretório que MATLAB) e copiar o seguinte texto para modelar a função de transferência:

```
M = .5; m = 0.2; b = 0.1; i = 0.006; g = 9.8; l = 0.3;

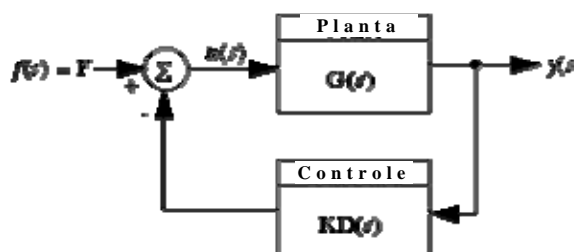
q = (M+m)*(i+m*l^2)-(m*l)^2;
num = [m*l/q 0];
den = [1 b*(i+m*l^2)/q -(M+m)*m*g*l/q -b*m*g*l/q];
pend=tf(num,den);
```

Função de transferência de laço fechado

O controle deste problema é um pouco diferente do que o normal de controle problemas que podem ser utilizadas para. Uma vez que se está a tentar controlar a posição do pêndulo, que deve retornar para a posição vertical após a perturbação inicial, o sinal de referência que se esta monitorando deve ser zero. A força aplicada no carro pode ser adicionada como um impulso. O esquema para este problema deverá ser parecido com o seguinte.



Será mais fácil de determinar a função de transferência apropriada para entrar no MATLAB se primeiro reorganizar o esquema do seguinte modo:



Adicionando o controlador PID

Esta função de transferência de malha fechada pode ser modelado no MATLAB inserindo o seguinte código para o final do seu m-file:

```
Kd = 1; Kp = 1; Ki = 1;  
contr=tf([Kd Kp Ki],[1 0]);  
sys_cl=feedback(pend,contr);
```

Esta função de transferência pressupõe que ambos os controles derivativo e integral, serão necessários, juntamente com controlo proporcional. Isto não tem de ser o caso. Pedisse começar com um controlador PI ou PD, basta substituir o *Contr* de acordo com sua escolha de controlador conforme mostrado abaixo.

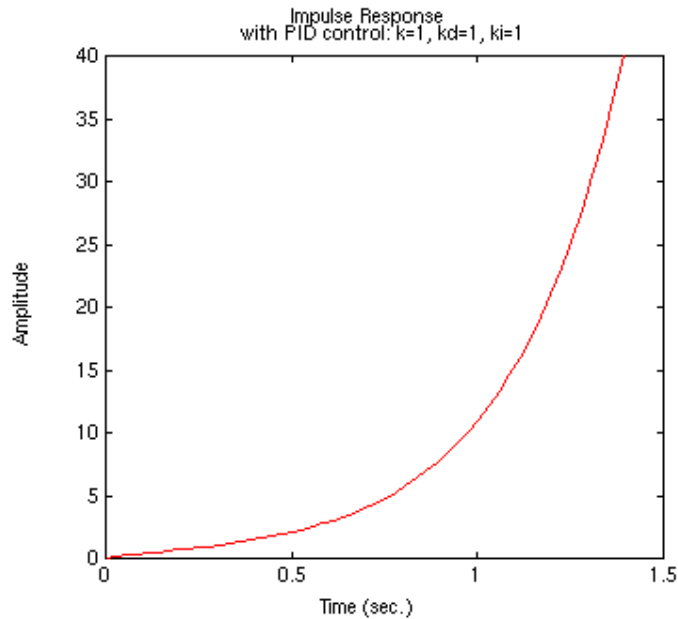
```
contr=tf([Kp Ki],[1 0]); % PI control  
contr=tf([Kd Kp],1); % PD control
```



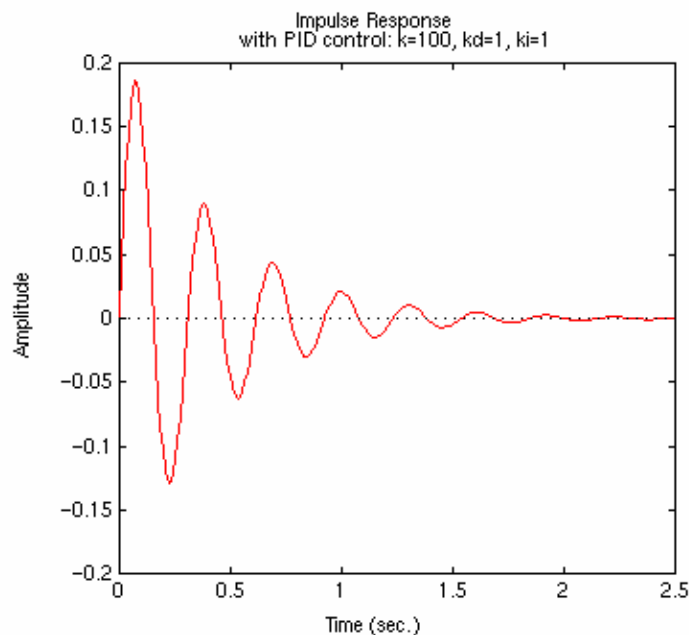
Agora se pode iniciar o controle efetivo deste sistema. Primeiro veremos o que a resposta ao impulso se comporta com os números já calculados. Entrando com o seguinte código para o final do seu m-file:

```
t=0:0.01:5;  
impulse(sys_cl,t)  
axis([0 1.5 0 40])
```

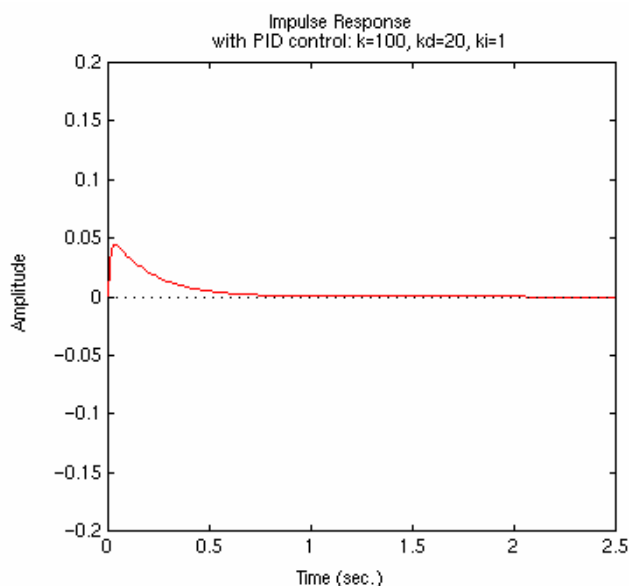
Deve-se obter a seguinte resposta a partir da perturbação do impulso:



Essa resposta ainda não é estável. Aumentando o valor do controle proporcional ao sistema. Aumentando o K_p variável para ver o efeito que tem sobre a resposta. Se definir $K_p = 100$, e definir o eixo ao eixo $([0, 2.5, -0.2, 0.2])$, se obtém a seguinte resposta velocidade:



A resolução de tempo é aceitável em cerca de 2 segundos. Pode-se remover o ganho integral constante para ver que a parcela integral do controle é necessária. A superação é exagerada elevada, pelo que deve ser corrigido. Para diminuir este problema, aumentasse o K_d variável. Com $K_d = 20$, se obtém um resultado satisfatório. Agora é possível ver a seguinte resposta ao impulso:



Como você pode ver, a superação foi reduzida de forma que o pêndulo não se movimenta mais de 0,05 radianos afastado do vertical. Todos os critérios do projeto foram cumpridos, pelo que é necessária mais nenhuma iteração.

4.3. Metodologia dos Testes Integrados no Sistema Físico

Após a conclusão do software foram feitos inúmeros testes somente com o hardware do microcontrolador, assim que cada resposta de comunicação com o dsPIC era aceitável foi elevando o grau de complexidade do software. Concluindo uma etapa de testes de programa integrado ao hardware foram sendo incrementadas as outras partes, como o circuito da ponte H e em seguida juntar os motores no conjunto. Foi criado um software de teste para o acionamento dos motores para verificar se o integrado que faz a inversão de giro dos motores realmente funcionaria com o microcontrolador. Ao se verificar que a ponte H corresponderia de forma desejada passou concepção do software que faria o cálculo do PID e assim ajustaria o sinal do PWM e moveria os motores de um lado para outro conforme o lado que o pêndulo estivesse.

4.3.1. Testes estáticos

A partir do momento que software foi estrutura os primeiros testes foram com a leitura do conversor A/D, como no hardware foi empregado o uso de um display, todas as respostas do conversor eram informadas pelo LCD. Com o potenciômetro ligado a porta do conversor A/D do microcontrolador foi constatado funcionamento de leitura.

Visto que já era possível fazer leituras na porta analógica/digital do dsPIC foram sendo feitos os cálculos dos ganhos k_P , k_I e k_D cada um era visualizado na tela afim de proceder com os testes de software.

4.3.2. Testes dinâmicos

Já com o software respondendo de forma satisfatória foi possível integrar a ponte H ao microcontrolador e fazer o controle dos motores. De forma a ver o sentido de giro a cada inversão da haste ligada ao potenciômetro que por sua vez ligado à porta A/D do microcontrolador.

Sendo assim foi possível analisar a saída do PWM no osciloscópio e ver a cada alteração nos parâmetros de ganho. A seguir serão mostradas imagens do sinal do PWM enviado para a ponte H e a mesma fazendo a inversão de giro dos motores.

Nesta amostra pode ver o sinal do PWM gerado com os seguintes parâmetros $k_p=1$, $k_i=1$ e $k_d=0,01$. A resposta ao pêndulo é muito oscilante pode se perceber com a diferença de tempo no sinal. Com estes parâmetros pode verificar que não é possível o equilíbrio.

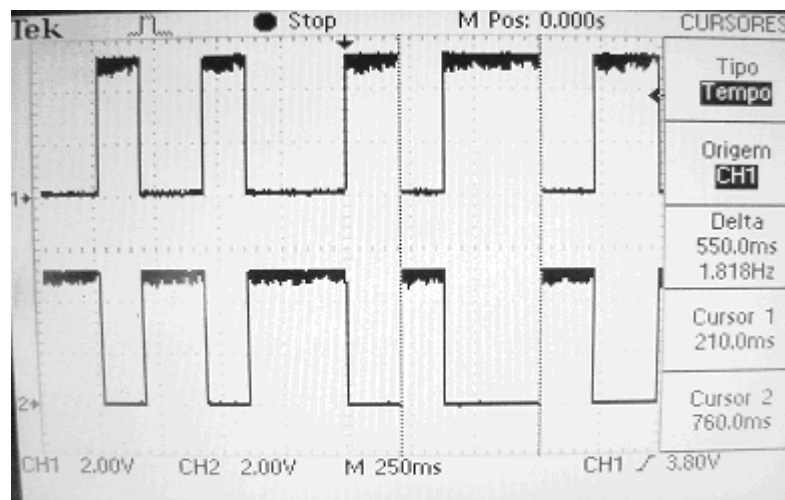


Figura 4. 3 – Primeiro teste de saída PWM para o controle dos motores.

No próximo teste feito foi simulado com novos parâmetros agora com os valores $k_p=1$, $k_i=0,01$ e $k_d=1$. nesta sim pode se visualizar a seqüência continua no sinal do PWM com estes valores de parâmetros o pêndulo chega próximo ao equilíbrio com uma boa resposta.

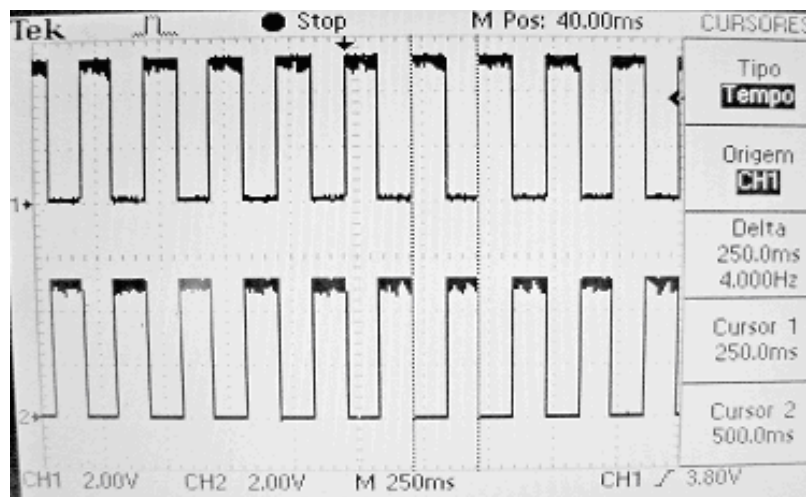


Figura 4. 4 – Segundo teste de saída PWM para o controle dos motores.

Nessa outra imagem pode ser visualizada uma forma de onda do PWM com uma diferença de tempo entre os intervalos. Foi utilizado os parâmetros $k_p=10$, $k_i=1$ e $k_d=1$. Também não teve uma boa resposta para o equilíbrio

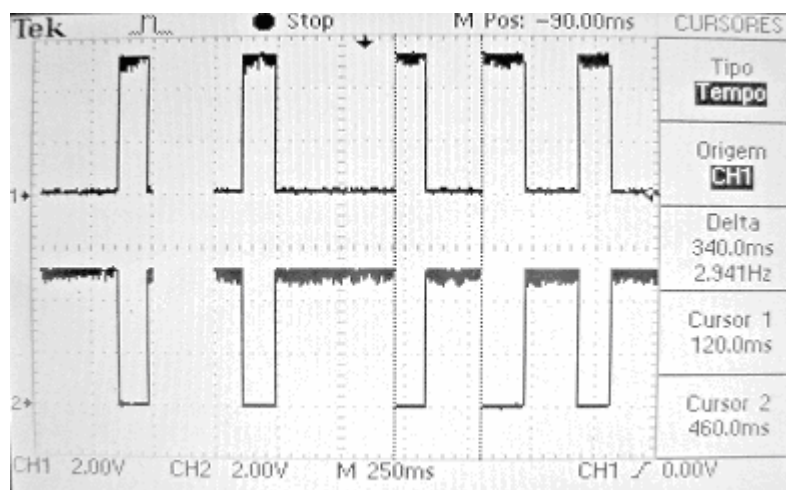


Figura 4. 5 – Terceiro teste de saída PWM para o controle dos motores.

A próxima imagem mostra a melhor resposta para o controle do pêndulo inverso. Com a definição dos parâmetros de $k_p=1$, $k_i=1$ e $k_d=100$. Pode se verificar uma boa resposta no tempo com o sinal do PWM com este retorno é possível ter um equilíbrio mais efetivo do pêndulo.

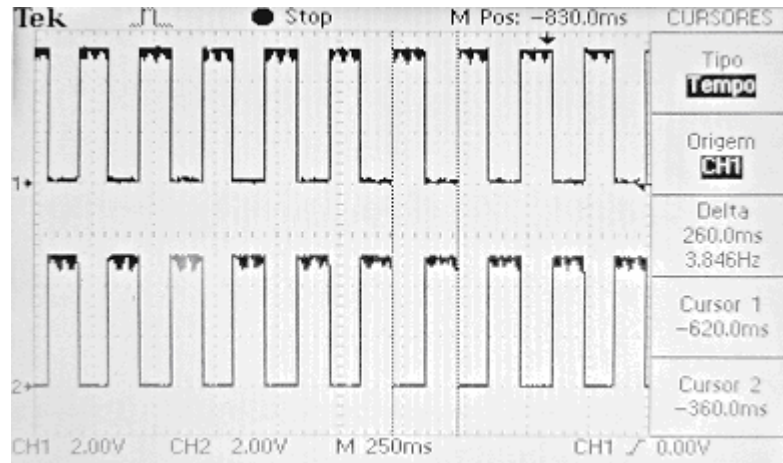


Figura 4. 6 – Quarto teste de saída PWM para o controle dos motores.

4.4. Análises Estatísticas

Com os seguintes dados foi possível verificar o comportamento do algoritmo PID e analisar gráficos de PWM.

4.5. Comparações dos Resultados das Simulações, Modelos e o Sistema Real

Na comparação de resultados do real com o simulado é difícil chegar às simulações ao ponto real, pois existem vários fatores que não foram consideradas nas simulações uma delas é o atrito do pêndulo com o solo e as características físicas de cada componente. Essas características são imprescindíveis para que se possa chegar algo bem próximo do simulado, pois influencia muito na resposta final.

Porem o que se pode avaliar nas simulações criadas é o fato de saber como vai se comportar o pêndulo alterado os parâmetro de ganhos k_p (proporcional), k_i (integral) e k_d (derivativo). Alterando estes parâmetros se avalia o nível de oscilação se cria e qual suposta interferência será criada ao modelo real.



5. CONSIDERAÇÕES FINAIS

O problema inicial encontrado no projeto foi a dificuldade de encontrar motores que pudessem responder ao controle desejado. Foram feitos vários testes em motores CC com a expectativa de chegar a um bom resultado, porém sempre se chegava a motores com repostas bem baixas de torque ou que exigiam uma fonte de alimentação muito alta e com uma corrente muito elevada. Com isso se optou por um modelo menor e consequentemente um controle desejado.

Conforme foi descrito o objetivo era chegar a um bom controle de um pêndulo inverso, neste tempo decorrido foram feitas muitas pesquisas envolvendo o projeto. Na área acadêmica é um projeto bem complexo e de difícil de chegar ao objetivo. Contudo o trabalho foi bem desenvolvido e tanto na área de pesquisa sobre o assunto como na aplicação do controle do pêndulo.

5.1. Conclusões

Este projeto proporcionou a experiência do desenvolvimento de um equipamento completo participando de cada etapa do projeto, desde a concepção inicial até os ensaios finais, vivenciando todas as dificuldades de buscas de informações, obtenção de recursos, relacionamento entre pessoas de áreas diferentes, tendo que atender a qualidade e o cronograma especificados.

O sistema de controle desenvolvido demonstrou estabilidade e foi capaz de controlar o Pêndulo Invertido com sucesso. Foram desenvolvidos sistemas de suporte, teste e apoio que auxiliaram no desenvolvimento do sistema de controle e realizados ensaios de controle com dois sistemas reais, comprovando a eficiência do algoritmo utilizado. Primeiramente utilizou-se um acelerômetro a fim de testar o algoritmo PID desenvolvido, depois transportou-se o algoritmo, já ajustado, para o sistema do Pêndulo Invertido. Os resultados dos ensaios de controle foram apresentados neste trabalho.



Utilizando os resultados obtidos do controlador PID acoplado a sistemas físicos reais, espera-se obter conhecimento relevante para a aplicação de controladores em outros sistemas não lineares similares a este.

5.2. Sugestões para Trabalhos Futuros

- Controle do pêndulo utilizando um giroscópio.
- Utilizar outro tipo de sensor como acelerômetro ou de distancia
- Fazer um outro tipo de controle sem ser o PID.



REFERÊNCIAS

Katsuhiko Ogata - “Engenharia de Controle Moderno” - Prentice/Hall do Brasil, 1982

John OKyere Attia - “Electronics and Circuit Analysis using MATLAB” - RC Press LLC, 1999

Matlab Curriculum Series, Prentice Hall 1998 - Mastering SIMULINK 2”
James B. Dabney; Thomas L. Harman

Duane Hanselman; Bruce Littlefield - “Mastering MATLAB 5 - A comprehensive Tutorial and Reference” - Matlab Curriculum Series, Prentice Hall 1998

Steven W. Smith - “The Scientist and Engineer’s Guide to Digital Signal Processing” - California Technical Publishing, Second Edition 1999



OBRAS CONSULTADAS

Katsuhiko Ogata - "Engenharia de Controle Moderno" - Prentice/Hall do Brasil, 1982

KOSOW, Irving L. - "Máquinas Elétricas e Transformadores"- Editora Globo. 1979.

Pereira, Fábio - "Microcontroladores Pic - Programação em C" – Erica

Zanco, Wagner da Silva - "Microcontroladores PIC - Técnicas de Software e Hardware para Projetos de Circuitos Eletrônicos" - Erica



APÊNDICE A – SOFTWARE DE CONTROLE

```
signed float duty_50;
char txt[16], txt2[16];

const float SetPoint = 565; // Objetivo
const float kP = 5.;
const float kI = 0.01;
const float kD = 0.7;
const float PidMaxError = 1.;
const float PidMaxIntError = 10.;

float ProcessValue = 0;
float i = 0, pid_out;
float LastProcessValue = 565;
float pTerm, iTerm, dTerm, error, IntegralError = 0;

//-----
void main()
{
ADPCFG = 0xFFFF; // programa todas os pinos do portb como I/O de uso geral
Trisb = 0; //configura portb como saida

//configura e inicializa LCD no modo 4 bits
Lcd_Custom_Config(&PORTB, 3,2,1,0, &PORTE, 4,3,5);
Lcd_custom_Cmd(Lcd_CLEAR); // apaga display
Lcd_custom_Cmd(Lcd_CURSOR_OFF); // desliga cursor
// */

TRISBbits.TRISB8 = 1;
Trisb.f8 = 1; //configura portb como saida
TRISEbits.TRISE0=0;
TRISEbits.TRISE1=0;
TRISEbits.TRISE2=0;
TRISEbits.TRISE3=0;

//-----
//delay_ms(500);
duty_50=Pwm_Mc_Init(400, 0x0, 0x11, 16); //Pwm_Mc_Init(5000,1,0xF,0);
Pwm_Mc_Start();
Pwm_Mc_Set_Duty(duty_50, 1);
while(1){
    ProcessValue = Adc_Read(8);
    error = SetPoint - ProcessValue;

    //delay_ms(10);
    WordToStr(ProcessValue, txt2);
    Lcd_Custom_Out(1,5, txt2);
    WordToStr(error, txt2);
    Lcd_Custom_Out(1,12, txt2);
    // */ //Bloco que Imprime valores de AD no LCD
//-----
    /*Calcula o Pterm e o erro limite*/
    if (error > PidMaxError){
        error = PidMaxError;
    }
}
```



```
    }
    else if (error < -PidMaxError){
        error = -PidMaxError;
    }
    pTerm = kP * error;
    /*
    WordToStr(pTerm, txt2);
    Lcd_Custom_Out(2,5, txt2);
    // */
//-----
    /*Calcula o integral erro limite*/
    IntegralError += error;
    if(IntegralError > PidMaxIntError){
        IntegralError = PidMaxIntError;
    }
    else if(IntegralError < -PidMaxIntError){
        IntegralError = -PidMaxIntError;
    }
    iTerm = kI * IntegralError;

//-----
    /*Calcula o Dterm*/
    dTerm = kD * (LastProcessValue - ProcessValue);
    LastProcessValue = ProcessValue;
    pid_out = (pTerm + iTerm + dTerm)*(1000./1.);

    WordToStr(pid_out, txt2);
    Lcd_Custom_Out(2,5, txt2);
    // */ // Bloco que Imprime valores de kd no LCD

    if (pid_out < 0)
    { pid_out = duty_50*pid_out;
      duty_50 = duty_50+pid_out;

      if(duty_50 < 0)
      {duty_50=0;}
    }

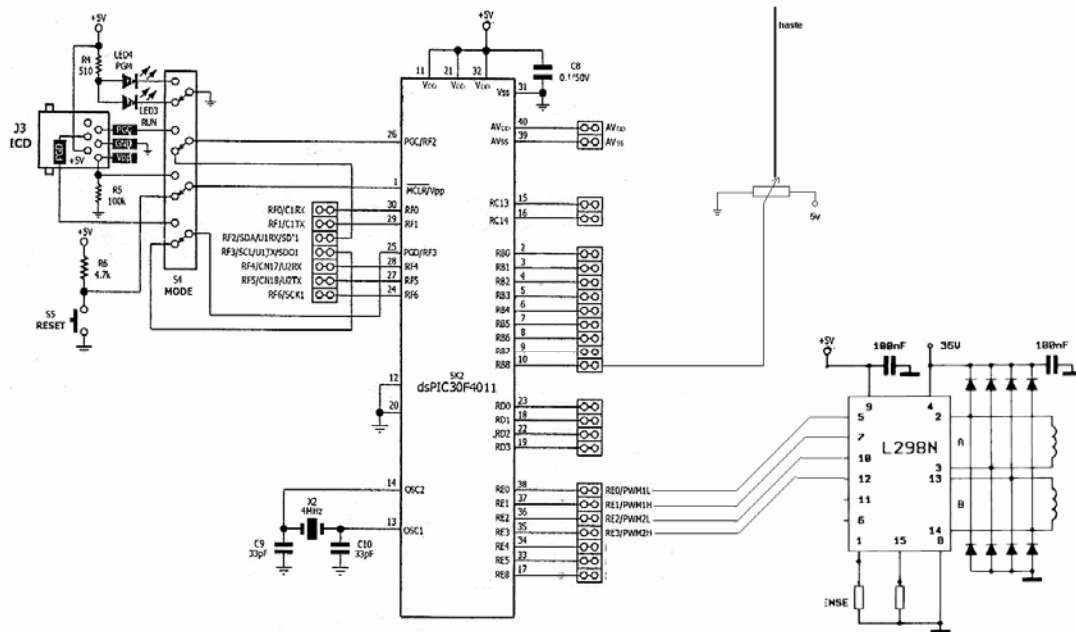
    else if (pid_out > 0)
    { if(duty_50 <= 0)
      {duty_50=100;}

      pid_out = duty_50*pid_out;
      duty_50 = duty_50+pid_out;

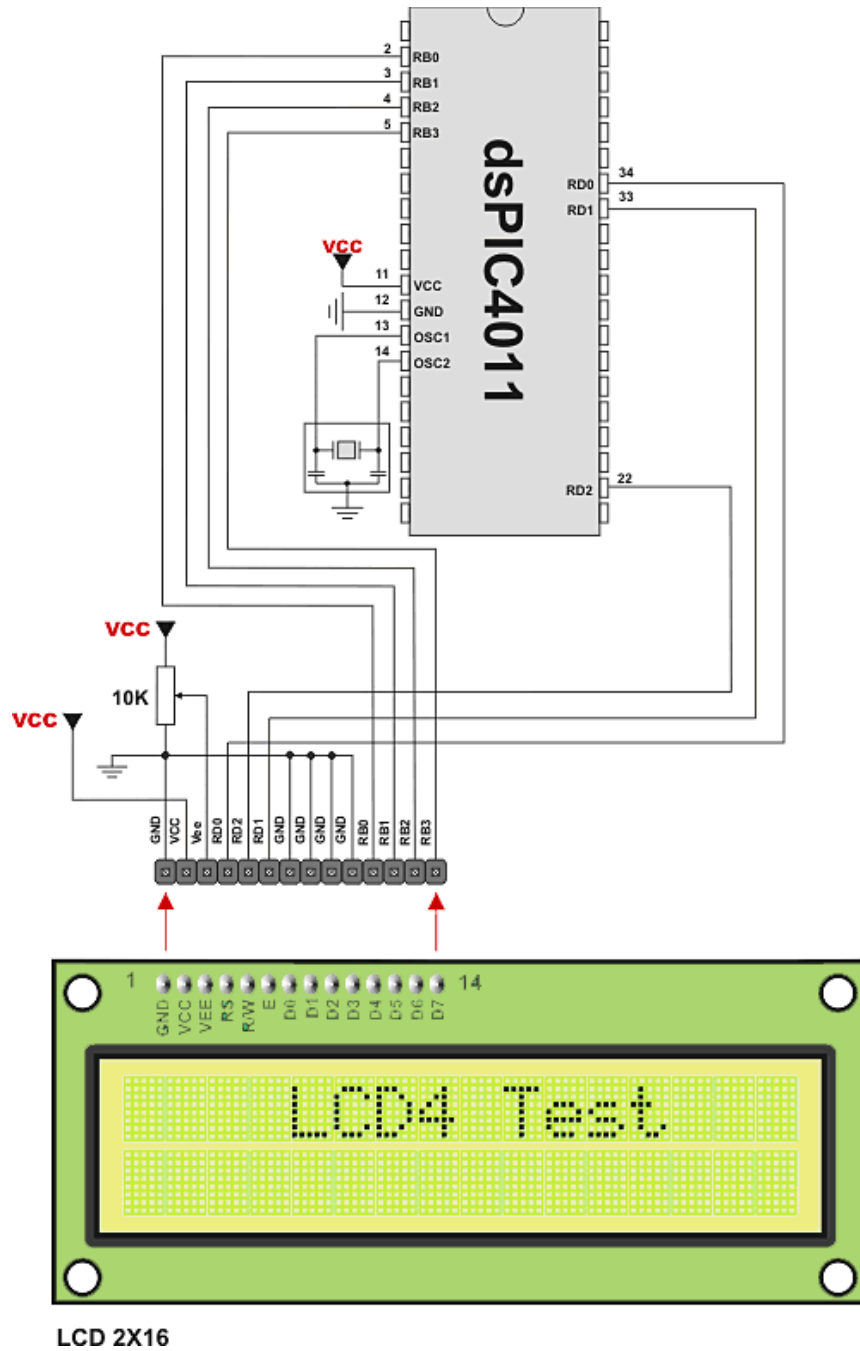
      if(duty_50 > 14998.)
      {duty_50=14998.;}
    }
    Pwm_Mc_Set_Duty(duty_50, 1);
    WordToStr(duty_50, txt2);
    Lcd_Custom_Out(2,12, txt2);
    // */ // Bloco que Imprime valores de saída do PWM no LCD
}
}
```



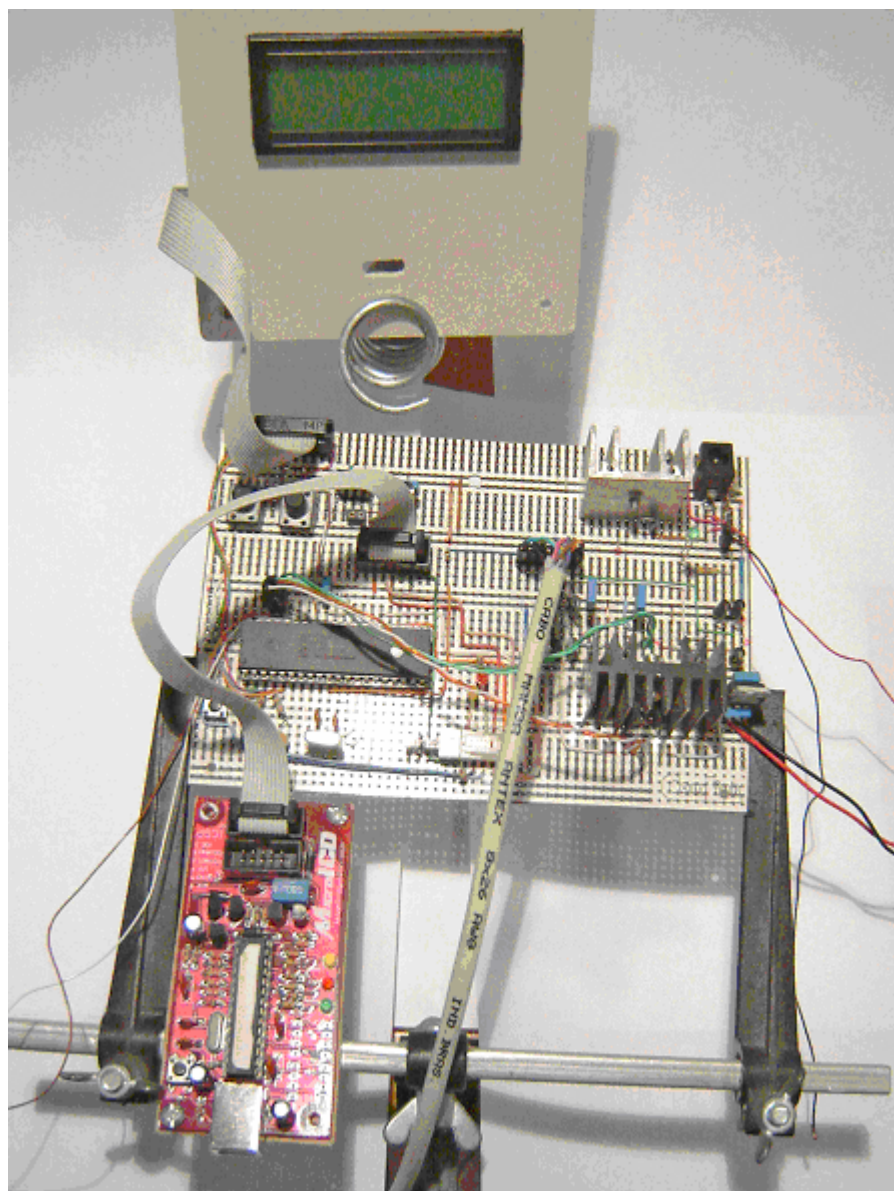
APÊNDICE B – ESQUEMA ELÉTRICO



APÊNDICE C – ESQUEMA DE CONEXÃO DE DISPLAY



APÊNDICE D – VISUALIZAÇÃO GERAL DO CIRCUITO



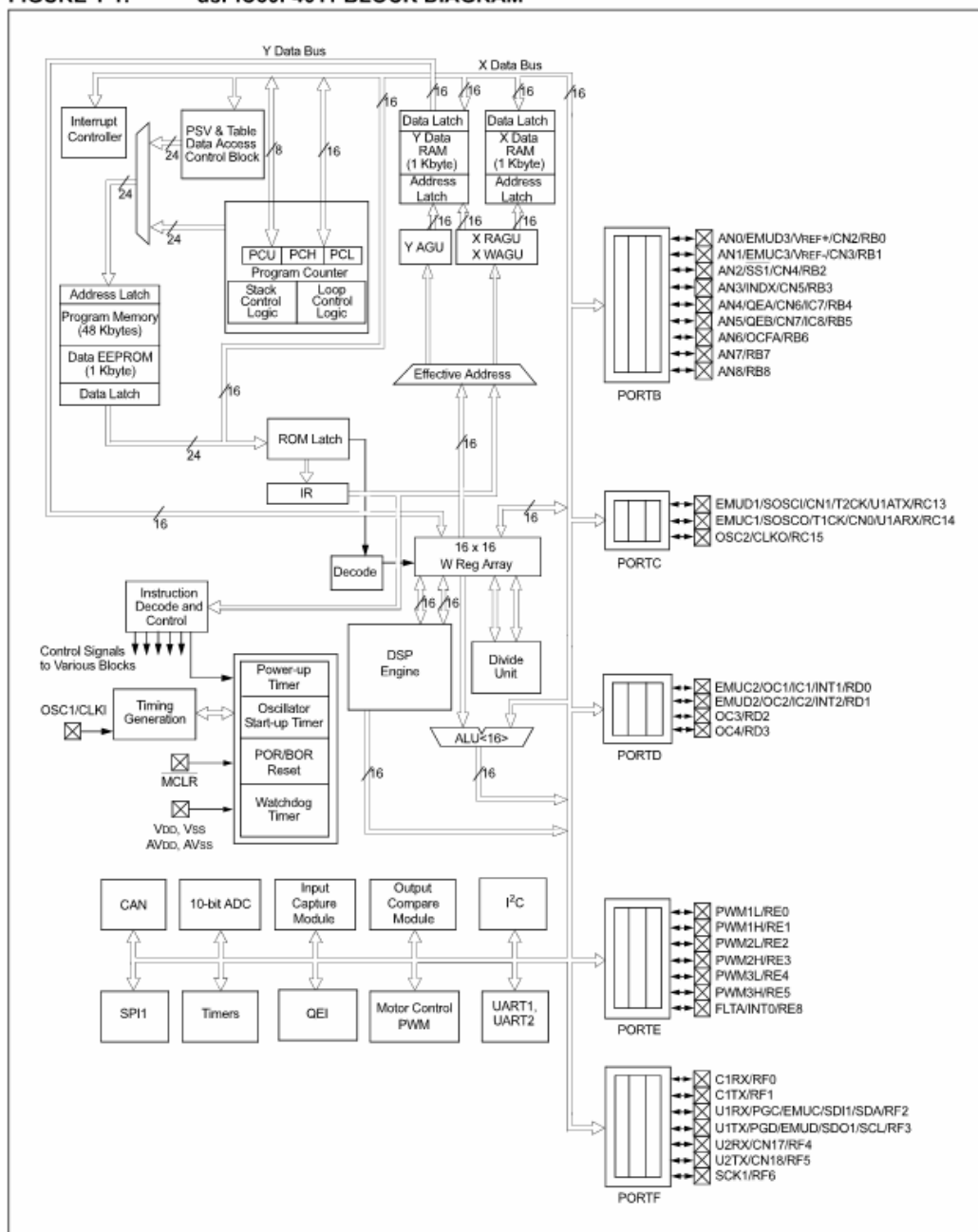
APÊNDICE E – PÊNULO INVERSO



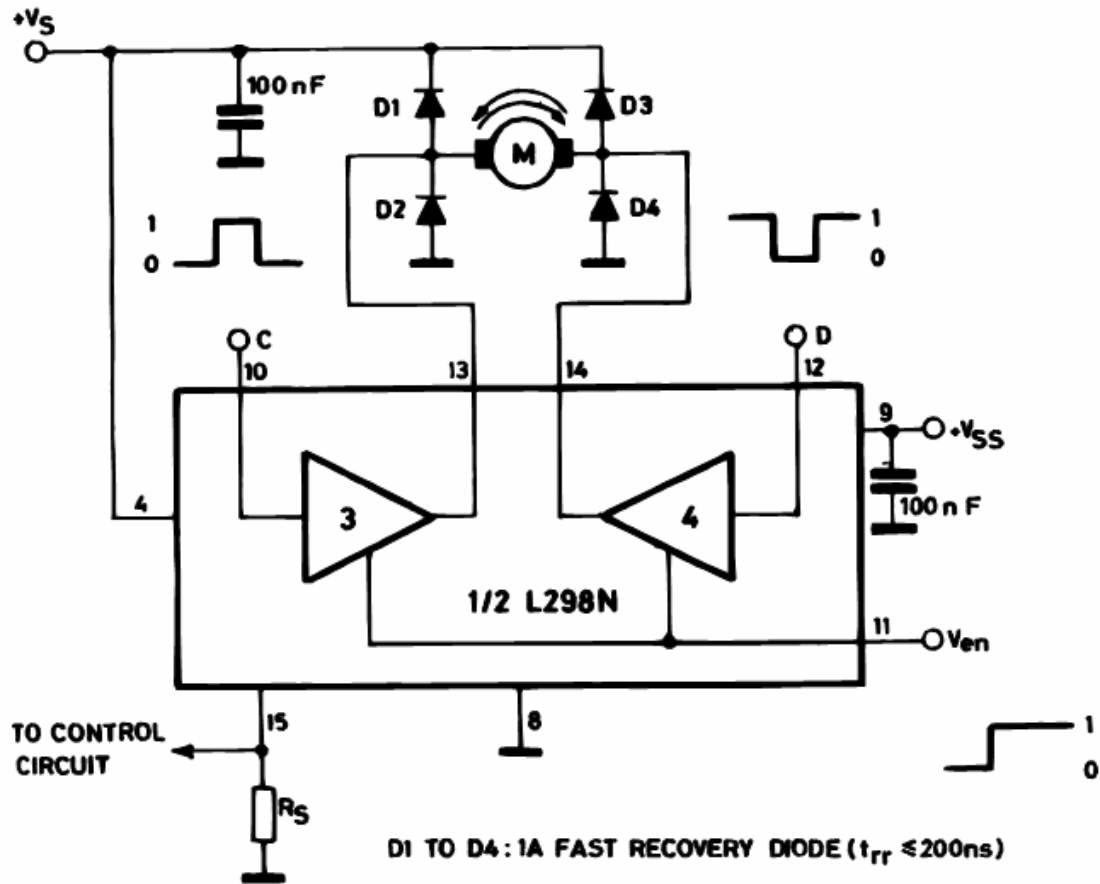


ANEXO A – DIAGRAMA DE BLOCOS DO (DSPIC30F4011)

FIGURE 1-1: dsPIC30F4011 BLOCK DIAGRAM



ANEXO B – ESQUEMA DE LIGAÇÃO PARA LM298



Inputs		Function
$V_{en} = H$	$C = H ; D = L$	Forward
	$C = L ; D = H$	Reverse
	$C = D$	Fast Motor Stop
$V_{en} = L$	$C = X ; D = X$	Free Running Motor Stop

L = Low

H = High

X = Don't care



ÍNDICE

A

Analógicos/Digitais · 23

D

dsPic · 39

M

MatLab · 49

Metas · 16

Metas · 17

O

Objetivos · 15

P

Pêndulo Inverso · 27

PID · 19

Ponte H · 40

PWM · 32