

Relatório de Laboratório de Controle de Sistemas : SEL0328

Aluno:Fabiano Aparecido Marino

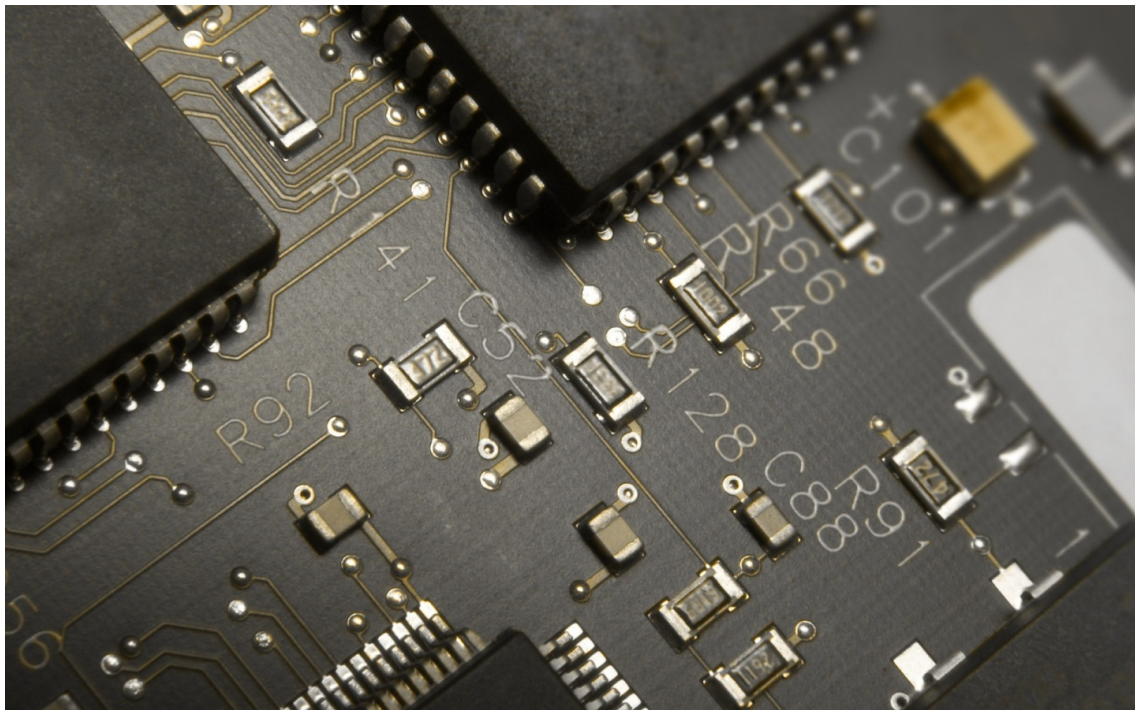
NºUsp:7143980

Pedro Carrascosa

N°Usp:7173480

Professor: Rodrigo Ramos

10 de Fevereiro de 2016



Conteúdo

1	Introdução	4
1.1	Motor de Corrente Contínua	4
1.2	Tacogerador	4
1.3	Optoacoplador	4
1.4	Observação	4
2	Procedimentos Experimentais e Simulação	5
2.1	Função de Transferência do Motor	5
2.2	PWM	7
2.3	Aspecto Prático do PWM	7
3	Conclusão Malha Aberta	7
4	Controle e Ajustes em Malha Fechada	8
4.1	Controladore PI	8
5	Implementação Do Controlador PI	10
6	Verificação dos Critérios de Desempenho em Malha Fechada e Aberta e Comparações.	11
7	Tratamento dos Dados Para a Obteção da Velocidade	13
8	Conclusões	13
9	Apêndice:Simulação no Proteus do Sistema Motor DC	14
10	Revisão de trabalhos feito no Proteus	14
11	Simulações em Malha aberta	15
12	Simulações do Controle em Malha Fechada.	16
13	Simulação PWM no Proteus e Construção da Placa de Controle	17
14	Conclusão Simulação Proteus	19

Lista de Figuras

1	Simulação no Matlab da fução de transfêrencia do motor excitada por um degrau de 12.3 Volts	6
2	Resultado da velocidade do motor segundo uma excitação degrau de 12.3Volts	6
3	Esquema em Diagrama de Blocos do Controle do Motor	8
4	Esquemático do controle PI Simulink	9
5	Esquemático De Malha Fechada no Simulink	9
6	Resultado da Simulação em Malha Fechada Com Controle PI	9
7	Circuito com os Amplificadores Operacionais Atuando Como Porporcional e Integrador, além do PWM	10
8	Resultado da Implementação do Controlador PI	11
9	Resultado da da Simulação do Controlador PI.	11

10	Erro de Regime Permanente.	12
11	Velocidade de Rotação do Motor em RPM	13
12	Simulação Proteus e Matlab e Dados Obidos Experimentalmente [1]	14
13	Simulação Proteus e Matilab e Dados Obidos Experimentalmente	15
14	Erro Com Relação aos Dados Reais da Simulação no Proteus e Matlab em Malha Aberta	16
15	Estrutura Simulada no Proteus com os Componentes Reais do Controlador	16
16	Estrutura Simulada no Proteus e Matlab e Dados Obtidos Experimentalmente	17
17	Circuito PWM	18
18	Simulação do Circuito PWM da Figura Acima.	18
19	Imagem na Forma de PCB Construida no Software Proteus Ares	18
20	Placa Renderizada do Circuito PCB Acima.	19

1 Introdução

O presente relatório tem como objetivo apresentar os resultados obtidos na disciplina de Laboratório de Controle de Sistemas, ministrada pelo professor Rodrigo Ramos. O objetivo das práticas realizadas consiste no controle de velocidade de um motor de corrente contínua, sendo que, para isto, busca-se explorar características do motor e periféricos, tais como função de transferência do motor, ganho do tacogerador, constante do PWM (Pulse Width Modulation) utilizado, dentre outras discutidas no decorrer do relatório. Os experimentos são verificados via simulação, utilizando o software Matlab-Simulink como simulador.

1.1 Motor de Corrente Contínua

A planta utilizada para o controle é um motor de corrente contínua, que consiste em uma máquina elétrica que converte energia elétrica em mecânica, na forma de rotação de um eixo. A construção do motor consiste em dois enrolamentos, um que permanece estático, chamado de estator, e outro que se movimenta chamado armadura. Em ambos há a passagem de corrente, a fim de gerar um campo magnético, porém, devido a disposição desses elementos, o campo gerado cria um torque fazendo com que haja a rotação da armadura, que por sua vez, gira um eixo, completando assim a conversão de energia elétrica em mecânica.

1.2 Tacogerador

Consiste em um elemento que gera um valor de tensão em relação ao valor de velocidade de rotação do eixo do motor. Seu princípio de funcionamento é o mesmo que máquinas geradoras de energia, ou seja, o movimento mecânico faz com que ímãs permanentes girem e devido isso altera-se o fluxo de campo magnético em uma bobina, gerando assim um nível de tensão.

1.3 Optoacoplador

Consiste em dispositivo também utilizado para se medir a velocidade em motores ou elementos girantes, porém seu funcionamento não é magnético, mas sim óptico, sendo que, quanto maior a velocidade, maior o número de pulsos que o optoacoplador emite, havendo uma relação entre a frequência da onda quadrada gerada e velocidade de rotação do motor. Este dispositivo foi utilizado para obter-se o ganho do tacogerador, afinal, sem este, fica impossível utilizá-lo, ou seja, empregá-lo na função de transferência do motor.

1.4 Observação

Por fim, o motor utilizado na prática consiste em um motor de corrente contínua (CC) com um tacogerador e optoacoplador embutido. Sendo o controle realizado de forma analógica, o optoacoplador será usado somente para medir o ganho do tacogerador.

2 Procedimentos Experimentais e Simulação

Primeiramente teve-se como objetivo mensurar a constante de amplificação do tacogerador, para assim poder-se usá-lo como sensor de velocidade. Para isso aplicou-se uma tensão nos terminais do motor e utilizou-se também o optoacoplador para medir a velocidade, uma vez que se conhecia a frequência da onda gerada para uma velocidade de giro do motor, sendo ela 1024 Hertz para π radianos de velocidade. A tabela com os valores inseridos no motor bem como a frequência de rotação lida e a tensão no tacogerador, estão inseridos na tabela (1).

Tabela 1: Valores da tensão aplicada no motor e a referida tensão gerada no tacogerador

Tensão Aplicada (Ta)	Tensão Tacogerador (Vtg)	Frequência do Optoplador
2	2,3	2300
4	5,6	6000
6	9	9700
8	12,6	13500
10	16,1	17500

Velocidade de rotação(Omega)	Constante do Tacogerador(Ktg)
14,11	0,16
36,8	0,15
59,49	0,15
82,79	0,15
107,32	0,15

Uma vez tendo a constante de amplificação do tacogerador fica possível avaliar a velocidade do motor pelo mesmo, pois há uma correlação entre a tensão gerada pelo tacogerador e a velocidade, sendo esta relação a razão entre a velocidade de rotação e a tensão gerada $K_{tg} = \frac{\omega}{Tensaodotacogerador}$ implicando em $\omega = (Tensaodotacogerador) * K_{tg}$.

2.1 Função de Transferência do Motor

O próximo objetivo foi descobrir a função de transferência do motor, sendo que para isto utilizou-se uma função degrau de 12 Volts, que foi aproximada por uma fonte de computador de valor de tensão medido de 12,3 volts. Aplicando-se o degrau verificou-se a tensão de saída no tacogerador por meio de um osciloscópio, que por sua vez, utilizando da função single, permitiu levantar a curva de saída do tacogerador com relação ao degrau de entrada. O osciloscópio traz consigo a tecnologia de emitir uma planilha com os valores da curva, tornando possível extrair os dados e levá-los ao matlab. Por meio da ferramenta IDENT do matlab fica possível descobrir os parametros da função de transferência do sistema de segunda ordem, ou primeira, como for da escolha do programador. Optou-se pelos parametros de um sistema de segunda ordem. Abaixo seguem as figuras da simulação feita no matlab figura (1) bem como a curva levantada por meio do osciloscópio, que se encontra na figura (2).

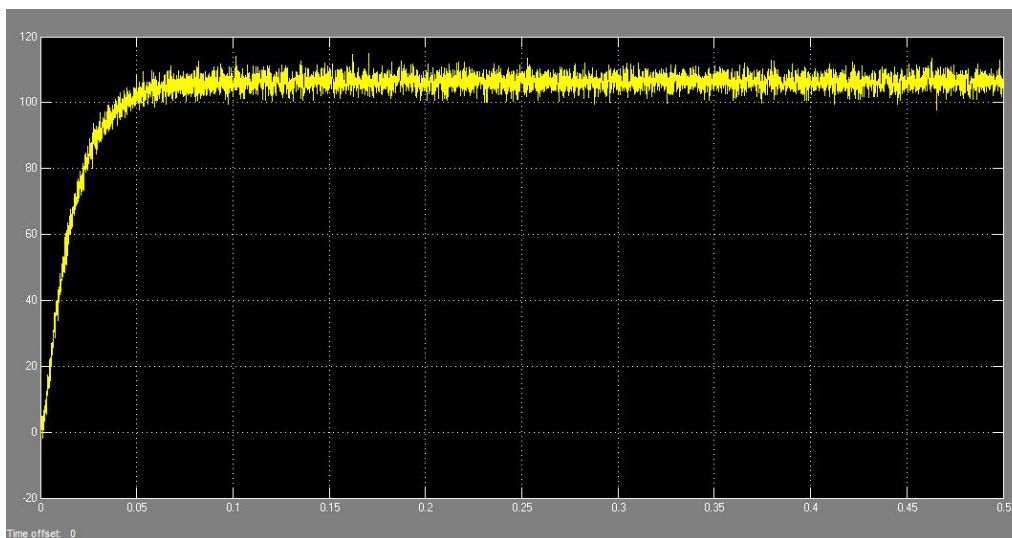


Figura 1: Simulação no Matlab da função de transferência do motor excitada por um degrau de 12.3 Volts

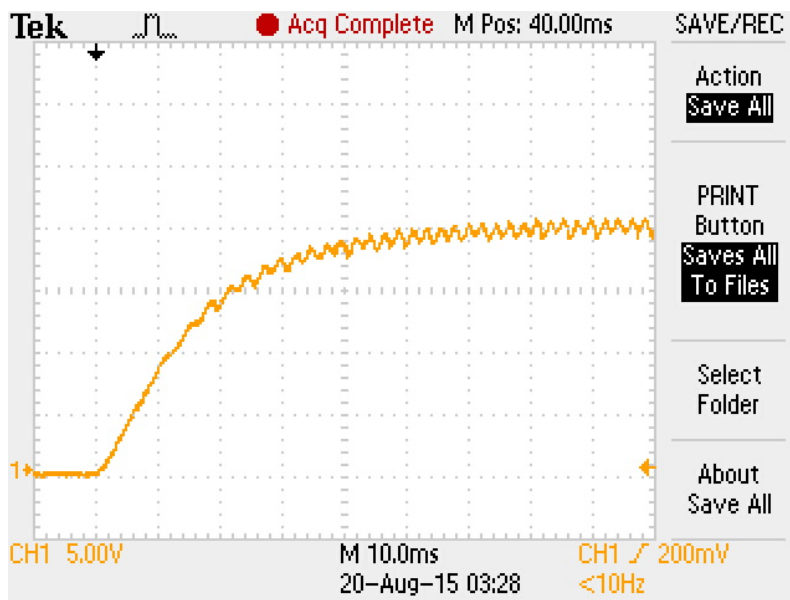


Figura 2: Resultado da velocidade do motor segundo uma excitação degrau de 12.3Volts

Por meio do IDENT do matlab, descobriu-se qual era a função de transferência do motor, e é como se expressa na equação (1) contando com o fato de que deve-se dividir o ganho da função pelo ganho do tacogerador, pois a velocidade foi obtida por meio do tacogerador, havendo, assim, sua influência.

$$G(s)_{motor} = \frac{343878,26}{(s - 482,88)(s - 66,93)} \quad (1)$$

Uma vez definida a função de transferência do motor, cabe agora analisar o dispositivo que atuara fazendo o controle de velocidade do motor, dispositivo cujo nome é PWM (Pulse Width Modulation) que será discutido no próximo tópico.

2.2 PWM

O controle baseado no princípio de modulação por largura de pulso tem base teórica no fato de que a tensão média aplicada por uma onda quadrada é tanto maior quanto maior a largura do pulso em nível alto, e, desse modo, alterando a largura do pulso, pode-se alterar a potência entregue a planta. Abaixo encontra-se a verificação teórica da técnica PWM consistindo em analisar a potência de uma onda quadrada e verificando o que ocorre quando altera-se a largura do pulso.

Lembrando que potência média de uma forma de onda é a integral dentro do período dividido pelo período da mesma, como se segue na equação (2)

$$PotenciaMedia = \frac{1}{T} \int_0^T f(t) dt \quad (2)$$

Como uma função quadrada com largura de pulso variável é definido por y_{max} para $0 \leq t \leq Dt$ e y_{min} para $Dt \leq t \leq T$, tem-se a seguinte expressão após a integração.

$$PotenciaMedia = Dy_{max} + (1 - D)y_{min} \quad (3)$$

Assim observava-se que variando a largura do pulso, que na expressão consiste no valor de D, tem-se uma maior potência para D=100% e 0 de potência para D igual a 0%.

2.3 Aspecto Prático do PWM

O dispositivo PWM também possui uma constante como função de transferência, e para levanta-la foi feita a seguinte experiência prática. Aplicou-se um sinal de PWM no motor e alterou-se a tensão de referência do mesmo, buscando verificar qual era a largura do pulso aplicando quando uma tensão era aplicada no PWM. Os valores verificados seguem na tabela (2).

Tabela 2: Tabela com os valores de tensão aplicado no PWM (V_{in}), tempo em nível lógico alto (T_{on}) Tensão de saída do PWM (V_{out}) constante do PWM (K_{pwm})

V_{in}	T_{on}	V_{out}	K_{pwm}
1,94	18us	2,511	1,294
3,86	32us	4,465	1,156
5,96	52us	7,255	1,217
8,01	68us	9,488	1,184
10	$T_{on}=T$	12	1,2

A constante do PWM é dada pela razão $K_{pwm} = \frac{V_{out}}{V_{in}}$. Seu valor, por meio da tabela, verifica-se que é por volta de 1,2.

Desse modo tem-se a configuração do tacogerador, e sua função de transferência, o mesmo para o motor e para o pwm. Agora resta fechar a malha do sistema e definir os parâmetros de controle, para que assim seja implementado o controle do motor.

3 Conclusão Malha Aberta

A representação esquemática acima ilustra o circuito desejado para que se faça o controle do motor. Uma vez que tenha-se em mãos os parâmetros que definem o motor, o tacogera-

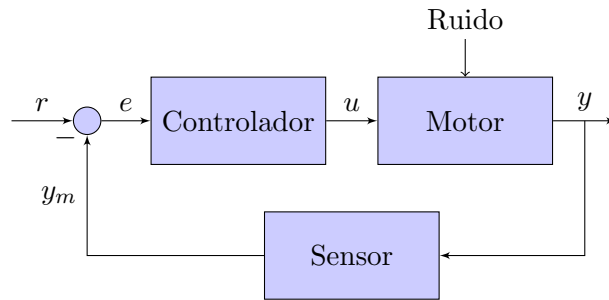


Figura 3: Esquema em Diagrama de Blocos do Controle do Motor

dor, identificado como sensor, bastando agora definir os parâmetros do controlador. Nesta primeira parte do curso fica a conclusão de que o controle de um sistema necessita do conhecimento da função de transferência dos componentes que compõem o sistema, sendo, no presente caso, o motor, a planta, o tacogerador, sendo o sensor, e o dispositivo controlador PWM. Na próxima etapa a busca será por encontrar um controlador via malha fechada do sistema, como ilustrado na figura (3) para assim estabelecer parâmetros de controle que definiram o projeto.

4 Controle e Ajustes em Malha Fechada

Verificou-se que os requisitos de controle em malha fechada, considerando controle em malha aberta, apenas proporcional, não atende os requisitos de controle, sendo eles erro de regime permanente menor que 5%, tempo de acomodação metade do de malha aberta e overshoot de no máximo 4.6%. Assim se faz necessário implementar outros modos de controladores analógicos disponíveis, tais como PI ou PID.

O controle PID não faz jus ao controle do motor, pois, por ter um bloco derivador, este irá elevar a saída do controlador a valores inaceitáveis, afinal há um grande ruído por parte da resposta do motor como pode ser visto na figura (2). Assim se faz a tentativa do controle por meio dos controladores PI, que apenas alteram o sinal de entrada de modo integral e proporcional. O bloco controlador integral zera o erro de regime permanente e o controle proporcional, junto ao integrador, faz com a resposta transiente seja também alterada, bastando ajustar esses parâmetros afim de alcançar os requisitos exigidos de controle.

4.1 Controladore PI

Os modos de controle em malha fechada consistem em realimentar a saída que por sua vez passa por um controlador e é passado para a planta novamente. Um fator crucial, é que o erro é provindo da resposta instantânea da planta em comparação com a entrada é repassado ao controlador e subsequentemente para a planta, mantendo o ciclo até que os requisitos de controle sejam atendidos.

Abaixo, na equação (4) encontram-se as equações que regem o funcionamento do circuito PI, que será utilizado neste projeto. Observe-se que $e(t)$ é o erro que provém entre a comparação da entrada com a saída.

$$u(t) = K_p e(t) + K_i \int_a^t e(t) dt \quad (4)$$

A figura (??) ilustra o esquemático em blocos de um controlador PI.

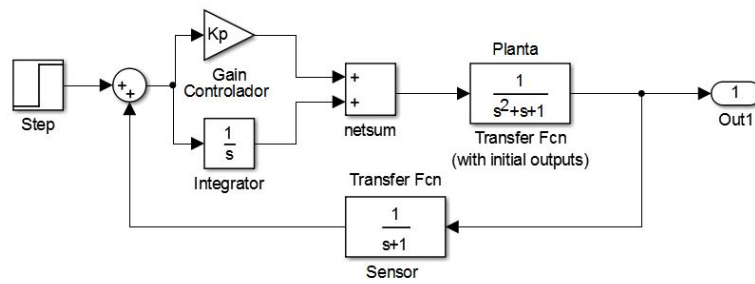


Figura 4: Esquemático do controle PI Simulink

Por meio do Matlab é possível encontrar quais são os valores que devem ser colocados no controlador PI em função dos parâmetros desejáveis de controle, isso feito por meio da ferramenta RLtool. Após obtidos esses valores coloca-se no simulink no referido bloco PI e efetua-se a simulação. Os resultados bem como o bloco esquemático do simulink encontra-se nas figuras (5, 6) logo abaixo.

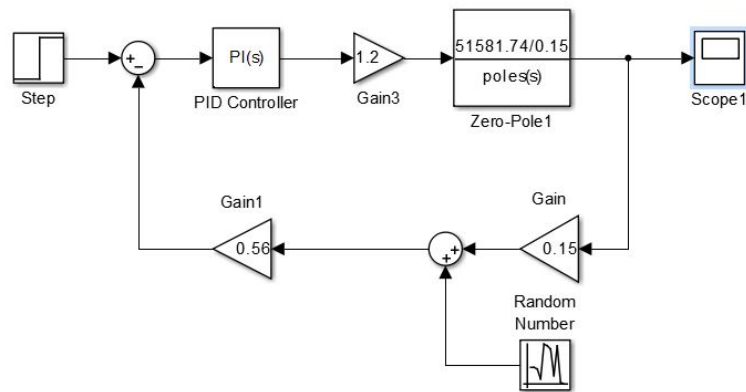


Figura 5: Esquemático De Malha Fechada no Simulink

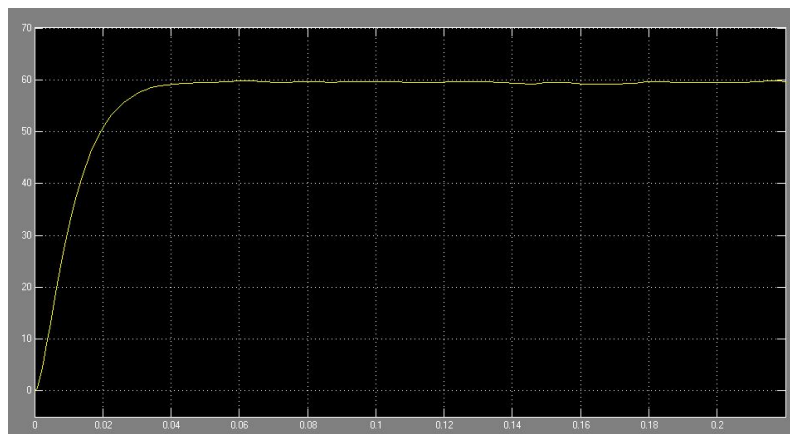


Figura 6: Resultado da Simulação em Malha Fechada Com Controle PI

5 Implementação Do Controlador PI

Partindo dos valores fornecidos para os ganhos K_p e K_i que foram 1,10506 e 82,184 pode-se calcular os valores das resistências e capacitores que irão encontra-se nos amplificadores proporcionais e integradores, bastando que seja satisfeita a razão de ganho em ambos amplificadores. Abaixo, na figura (7) encontra-se a configuração do sistema com os amplificadores operacionais atuando como proporcionais e integradores. Os amplificadores utilizados para efetuar o processo foi o UA741.

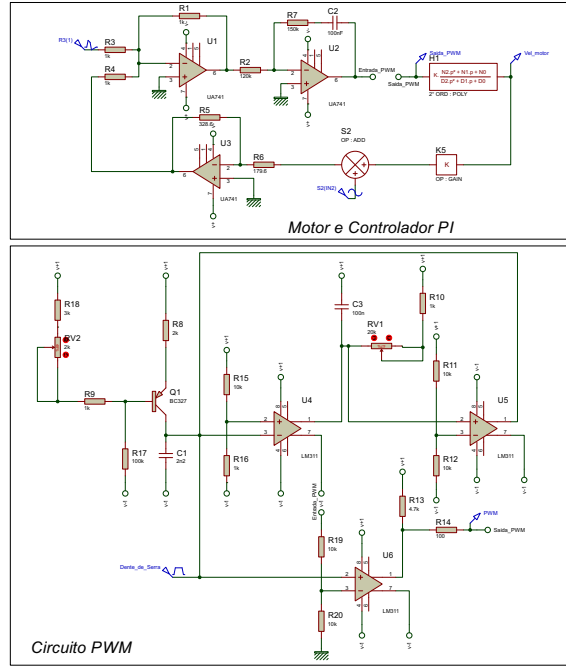


Figura 7: Circuito com os Amplificadores Operacionais Atuando Como Proporcional e Integrador, além do PWM

Assim o controlador PI deve ser implementado como na figura (7) substituindo o bloco de ganho 1.5 pelo dispositivo PWM.

O circuito foi implementado em bancada juntamente com o PWM e a resposta obtida encontra-se na figura (8) logo abaixo. Verifica-se que o resultado se equipara ao simulado, que se encontra na figura (9), valendo todos os procedimentos teóricos utilizados neste experimento.

Como pode-se observar a ponta de prova do osciloscópio estava na saída do tacogerador, o que faz com que a mesma seja dividida pela constante de amplificação do mesmo, que é por volta de 0.15. Assim, sendo esperado que a velocidade do motor em malha fechada tivesse um valor próximo de 60 rad/s para um degrau de 5 Volts foi alcançado, pois, como verifica-se a tensão emitida pelo tacogerador foi de 9.52 Volts, o que, dividido por 0.15 gera um valor de 63,46 rad/s, condizendo com as previsões de simulação, como pode ser visto na figura (6).

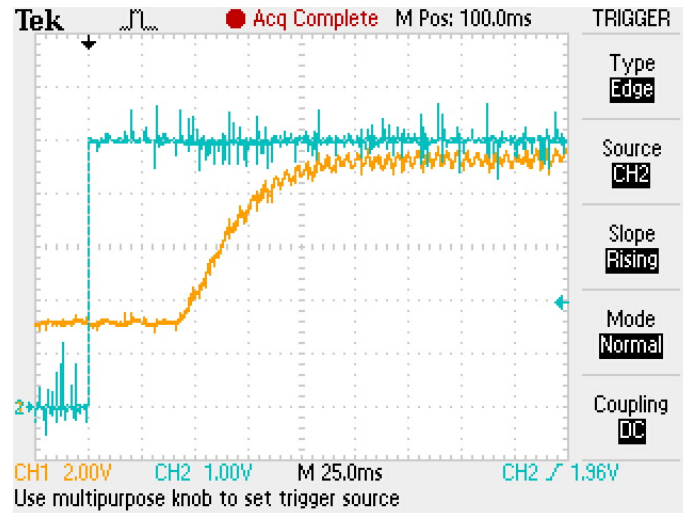


Figura 8: Resultado da Implementação do Controlador PI

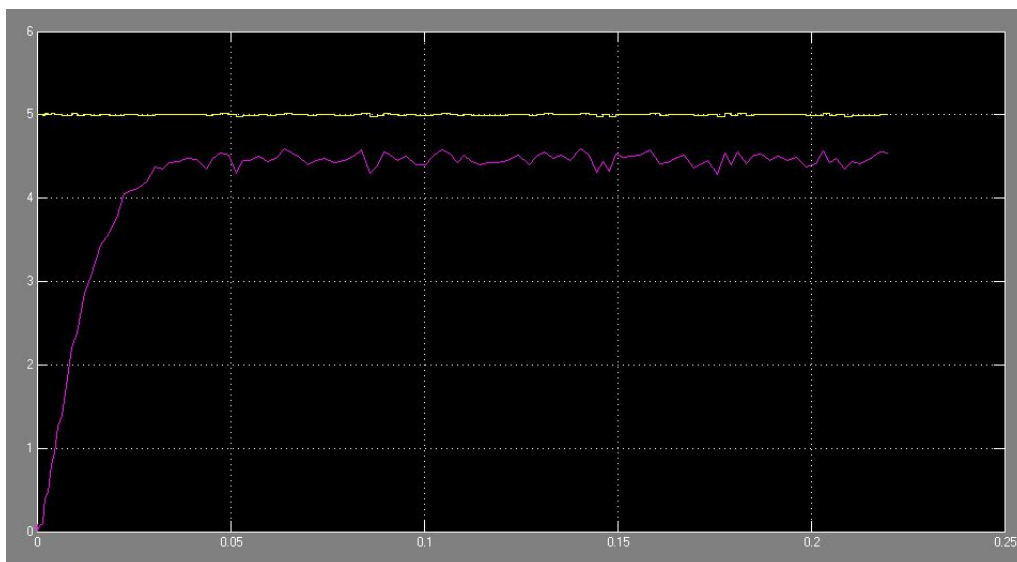


Figura 9: Resultado da da Simulação do Controlador PI.

6 Verificação dos Critérios de Desempenho em Malha Fechada e Aberta e Comparações.

O objetivo de controle era fazer com que o overshoot máximo não ultrapassasse 4.6%, que o erro de regime permanente fosse menor que 5% e o tempo de acomodação fosse metade do de malha aberta. Abaixo encontra-se a explicação para esses termos.

- **Overshoot:** Trata-se do valor que a resposta assume quando dá o primeiro salto de tensão na mesma, excedendo o valor de estado estacionário ou regime permanente. Não houve overshoot em ambas as respostas, malha aberta e fechada, sendo assim não há comparação a fazer, apenas fica o fato que este critério de desempenho foi alcançado, ou ao menos não houve a necessidade de controlá-lo.
- **Erro de Regime Permanente:** Trata-se da diferença que há entre o valor da entrada

e da saída em regime permanente, considerando a malha fechada. Por haver um controlador integrador, o erro é para tender a zero, porém isto limita o tempo de acomodação, havendo então um balanço entre esses dois parâmetros de controle. Como pode ser visto na figura (10) o erro tende a zero com o tempo, e por meio de média aritmética verifica-se que o erro é da ordem de 0.68 sendo 6.8%, quase atendendo ao critério de performance específica, que está na faixa de 5%. Porém, deve-se enfatizar que o sinal de saída possui um grande ruído, o que afeta o sistema no seu estado estacionário, justificando o erro de regime calculado.

- Tempo de acomodação: Trata-se do tempo que sistema leva para deixar de variar a sua resposta devido ao overshoot em um intervalo de amplitude de -5% a +5% do valor de regime. Como não houve overshoot no sistema, pode-se dizer que este fator foi alcançado.

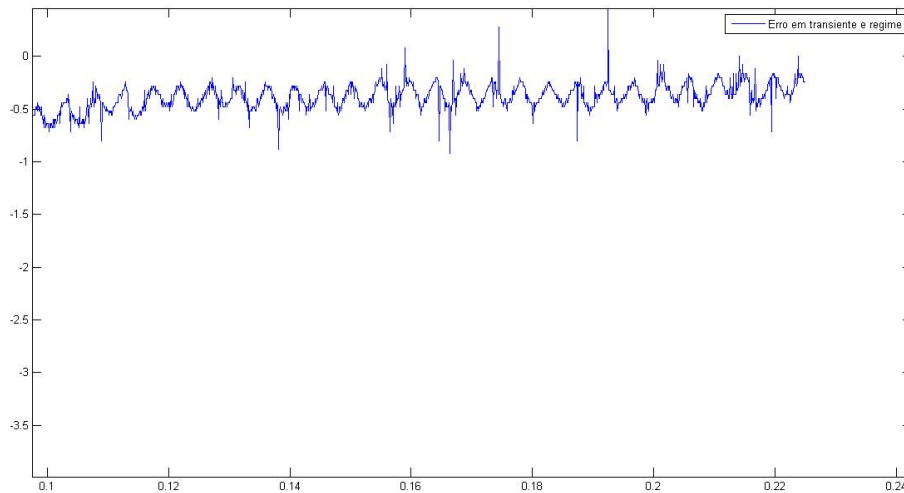


Figura 10: Erro de Regime Permanente.

7 Tratamento dos Dados Para a Obteção da Velocidade

Os dados obtidos experimentalmente presente na figura (8) e na simulação presente na figura (9) representam a tensão gerada na saída do tacogerador, sendo assim não representam a velocidade de rotação do motor, afinal o ganho do tacogerador é 0.15. Assim, há a necessidade de se dividir a resposta por 0.15 para obter a velocidade de rotação, esperando que o resultado seja conforme mostrado na simulação da figura (6).

Abaixo, na figura (11) pode-se observar que a simulação encontra-se em perfeita ligação com os dados apresentados nas figuras que representam a velocidade de rotação do motor.

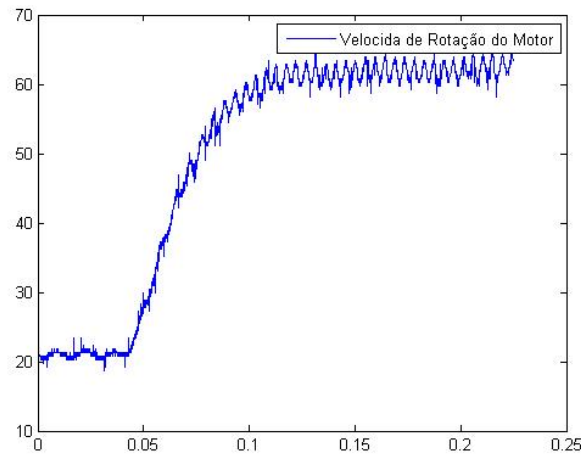


Figura 11: Velocidade de Rotação do Motor em RPM

8 Conclusões

Pode-se ver que as simulações feitas no matlab ficaram com boa precisão com relação aos experimentos, mostrando grande fidelidade do software mesmo com operações complexas, como ajustar os polos em malha fechada por meio do RLTOOL. Houveram falhas mínimas, menos de 1% na verificação do erro de regime permanente com relação ao experimento, porém é aceitável pois há um grande ruído na saída do tacogerador. Outro fato a se observar é que a montagem do circuito na prática deve obedecer a lei de colocar resistências de alto valor para fazer com que os amp-ops funcionem corretamente, pois há problemas com relação à impedância de entrada dos mesmos com baixos valores de resistência. Por fim verificou-se a necessidade de se colocar um capacitor junto à saída do tacogerador com a entrada do circuito PI, pois os ruídos provindo do motor estavam impedindo a captação dos sinais de velocidade.

9 Apêndice: Simulação no Proteus do Sistema Motor DC

O software proteus, oferecido pela empresa Labcenter Electronics, não é gratuito, apesar de haver a versão demo disponível no site da fabricante. Possui diversos componentes tanto na área analógica quanto digital, estando muito deles disponíveis para simulação. Há componentes ideais, que fazem seu princípio sem parâmetros de limitação, e componentes especificados por fabricantes, havendo também a possibilidade de se criar componentes, tanto em esquemático quanto em PCB (Printed Circuit Board) para fins de construção do esquemático produzido. Por enquanto o que se vê de diferencial é a presença de componentes digitais, e isso não se restringe a portas lógicas e flip-flops como encontra-se presente em diversos softwares de simulação, mas também com conversores AD e DA de vários fabricantes e microcontroladores programáveis de diversos fabricantes, indo desde de ARM a outros menos robustos como a famosa família de controladores MCU8051. Display de sete segmentos e de cristal líquido também encontram-se presentes, permitindo a simulação conjunta dos componentes citados. Memórias também encontram-se disponíveis para simulação e esquemático e simulação conjunta com outros periféricos. Em fim, fica a critério do projetista como utilizar os vários componentes presentes no simulador.

Um outro fato que leva à utilização do proteus, é a relação que existe entre suas duas interfaces ISIS e ARES, que permite a construção do projeto esquemático no ISIS apenas transportando para o ARES com as ligações entre componentes já executada, o que é comum em softwares de PCB. O que diferencia o proteus é que o esquemático já possui sua simulação feita, assim o PCB só é executado quando o esquemático estiver funcionando, garantindo de primeira a construção do circuito. A figura (12) ilustra essa vantagem.

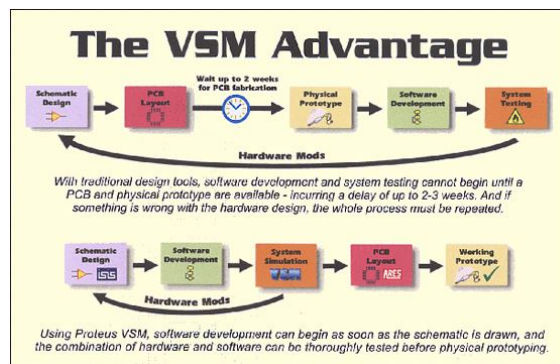


Figura 12: Simulação Proteus e Matlab e Dados Obtidos Experimentalmente [1]

10 Revisão de trabalhos feito no Proteus

Apesar não ser tão extensa quanto a bibliografia de trabalhos que usam o matlab, o proteus tem uma gama de artigos publicados que fazem o seu uso considerável, principalmente quando há a utilização de microcontroladores no sistema em questão, afinal uma grande potencialidade do proteus é a presença de vários microcontroladores e periféricos, sendo que alguns artigos defendem o uso do proteus como ferramenta para o ensino de microcontroladores[2][3]. A presença de vários tipos de motores provém a possibilidade de simulá-los, como pode ser visto no seguinte trabalho[4]. Controle de cargas e sistemas de potência baseado em microcontroladores PIC (Programmable Integrate Circuit) também

encontram-se desenvolvidos[5].Simulação do uso de pesticidas emitidos por meio de circuitos microcontrolados[6].A simulação e construção de display de leds em caracteres de chineses também já foi implementada [7]e outras utilidades utilizando displays [8].

Trabalhos mais impactantes também foram feitos, que utilizam praticamente somente o proteus como software de simulação, como exemplo tem-se a construção de um microcomputador [9], o controle de carga (explicar melhor) [10].A construção de um termostato [11] o controle de temperatura e humidade baseado no proteus [12] avanços na tecnologia de pizeletrico [13], a construção de um voltmetro digital por meio de micrconcontroladore [14], busca de máxima portência utilizando sistemas micrcontrolados [5] e muitos outros trabalhos que utilizam diretamente e indiretamente o proteus como simulador.

11 Simulações em Malha aberta

Primeiramente as simulações visam verificar os resultados dos softwares , proteus e matlab, com relação aos dados reais e a ambos e entre si.Na figura (13) encontra-se os resultados da simulação em malha aberta feita com a função de transferência do motor,bem como a curva obitida experimentalmente e, na figura (14) encontra-se o erro obtido em relação as duas simulações com os dados obtidos experimentalmente.Como pode-se observar o erro é tolerante em ambas as simulação.

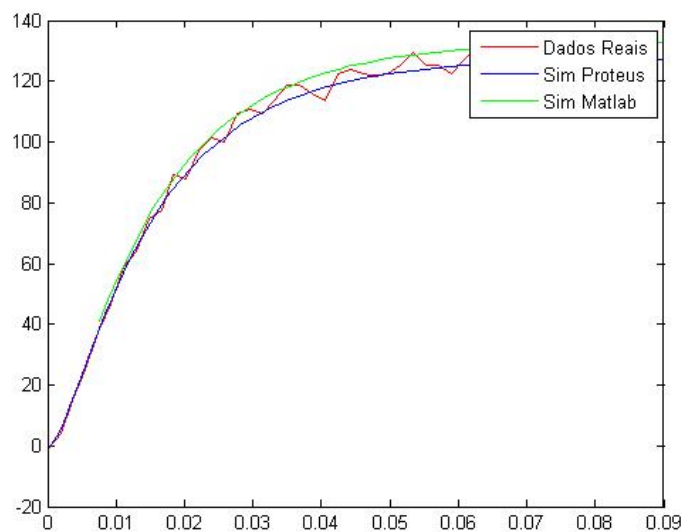


Figura 13: Simulação Proteus e Matilab e Dados Obitdos Experimetalmente

Assim, verifica-se que ambas as simulações obtiveram erros razoáveis com os dados obitidos experimentalmente.Porém, cabe salientar que foi usado no proteus os blocos de função de transferência, que não é seu ponto forte, afinal não há uma grande gama de funções.No que diz respeito a simulação do controle do motor em malha fechada, caberá o uso de componentes reais do Proteus, sendo este seu ponto forte.Isto é conteúdo dos próximos tópicos.

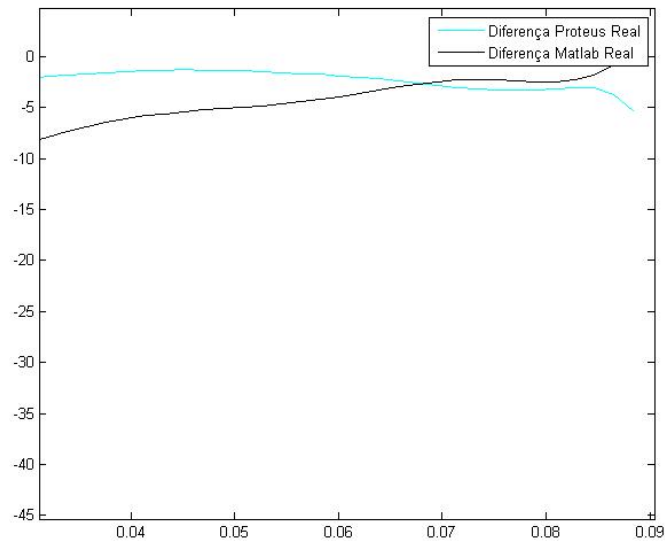


Figura 14: Erro Com Relação aos Dados Reais da Simulação no Proteus e Matlab em Malha Aberta

12 Simulações do Controle em Malha Fechada.

A simulação foi realizada no software Proteus utilizando os componentes e resistências que compõem o controlador, utilizando blocos para representar o PWM e o ganho do tacogerador, além da planta do motor. A figura (15) ilustra a construção feita.

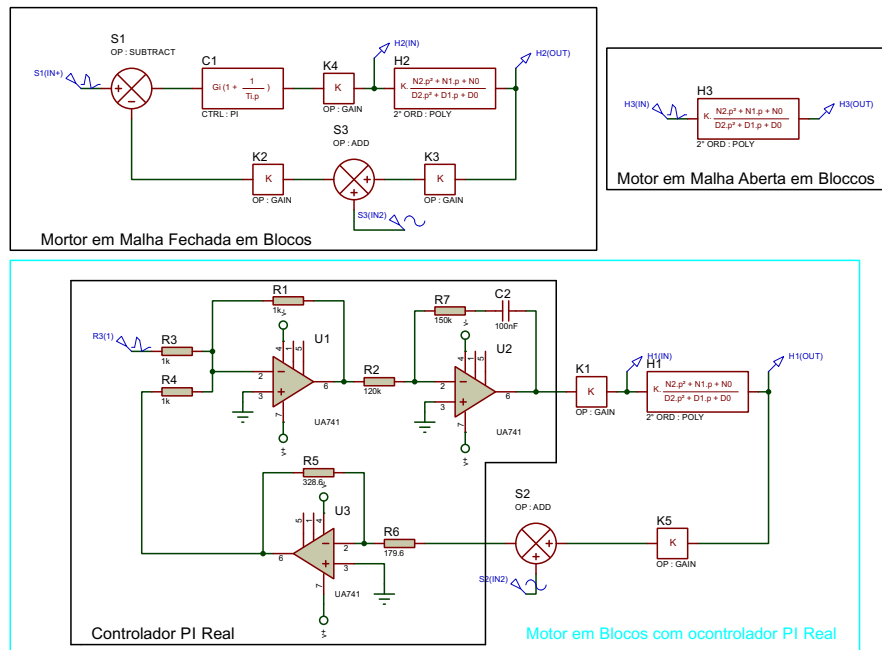


Figura 15: Estrutura Simulada no Proteus com os Componentes Reais do Controlador

Foram feitas as simulações nos softwares Matlab e Proteus e plotadas juntamente com o intuito de verificar a precisão das simulações nos softwares, principalmente no proteus, por ser de menor reputação. A resposta real também foi plotada para verificar a precisão. Tal construção de simulações encontra-se na figura abaixo (16).

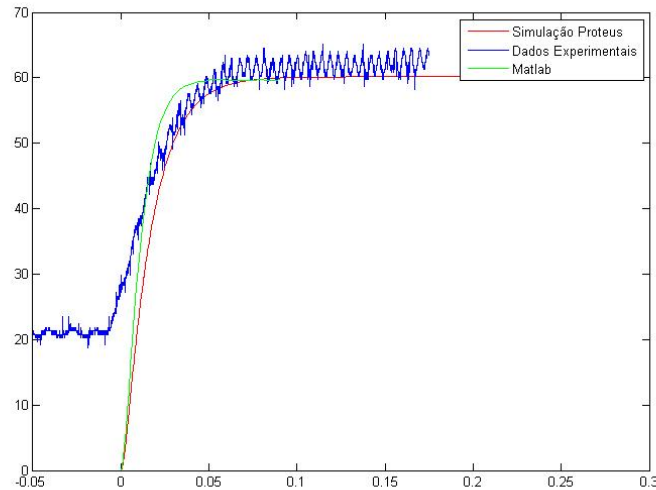


Figura 16: Estrutura Simulada no Proteus e Matlab e Dados Obtidos Experimentalmente

Por meio das comparações feitas entre os simuladores e dados obtidos no laboratório, verifica-se que são confiáveis as simulações no proteus para blocos de controle e amp-ops.

13 Simulação PWM no Proteus e Construção da Placa de Controle

O PWM é uma construção essencial no controle do motor, e seu funcionamento, no quesito de componentes analógicos, se dá por meio da comparação da forma de onda de uma dente de serra com uma tensão de referência. Conforme varia-se a tensão de referência varia-se o duty cycle do PWM. A figura (17) explicita o circuito do PWM utilizado, sendo que a figura (18) expressa a simulação do componente. A simulação foi realizada com uma tensão de entrada no valor de 5,96 Volts, gerando o duty cycle precisamente como mostra a tabela (18) que é de valor 52us.

Como foi dito, o software Proteus possibilita a construção da placa de controle, além de sua simulação. Assim, a construção do projeto se dá quando todas as simulações estarem batendo com os resultados esperados, expresando essa vantagem como explicito na figura (12). O circuito em sua forma de simulação e renderização para o circuito real encontram-se nas figuras (19), (20) abaixo. Com esta placa o controle do motor pode ser feito apenas conectando os elementos expressos na figura da placa real renderizada, sendo que TC é a saída do Tacogerador, +15V o 15 volts e -15V o menos 15 volts, D_S a entrada dente de serra, se não se desejar colocar um componente que gere tal forma de onda na placa como o LM555 que é um dos CI's capaz de gerar essa forma de onda, IN é tensão de referência para o PWMS e a saída do PWM que vai para o terminal do motor.

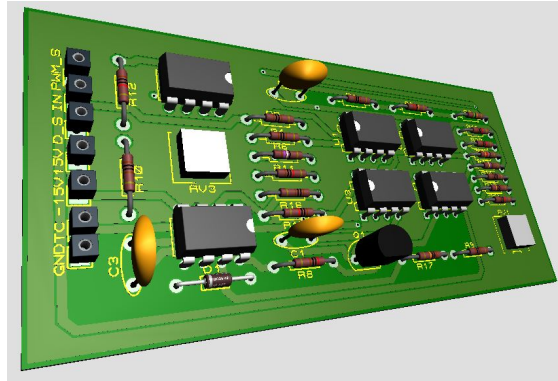


Figura 20: Placa Renderizada do Circuito PCB Acima.

14 Conclusão Simulação Proteus

Como pode ser visto pela comparação das simulações, observa-se que o proteus possui grande fidelidade quanto a simulações, havendo pequena diferença entre a simulação no matlab, dados reais. O fato de poder-se simular e construir o circuito simulado é uma grande vantagem, afinal garante-se que o sistema terá o funcionamento desejado. Assim, conclui-se que tal software é um bom simulador para auxiliar em projetos de controle.

Referências

- [1] linux UBUNTU debian. http://www.labcenter.com/products/vsm/vsm_overview.cfm. Accessed: 2015-09-13.
- [2] Li Wang Bo Su. Application of proteus virtual system modelling (vsm) in teaching of microcontroller. *International Conference on E-Health Networking, Digital Ecosystems and Technologies*, pages 375–378, 2010.
- [3] Dražen Cika and Darko Grundler. Proteus virtual system modelling used for microcontroller education. *MIPRO*, pages 1034–1038, 2010.
- [4] Dr. Dahaman Ishak Mohamad Nasrul Abdul Satar. Application of proteus vsm in modelling brushless dc motor drives. *International Conference on Mechatronics*, 2011.
- [5] Anwar Shabir Ahsan Shahid. Microchip based embedded system design for achievement of high power factor in electrical power systems. *IEEE*.
- [6] Ji Na Liu Wei Zhou Ying-chang Sun Rong-xia, Guo Liang. The simulation design of the ozonizer base on proteus. *The 1st International Conference on Information Science and Engineering*, pages 5265–5267.
- [7] Guo Jinying Ding Yanchuang. Led display screen design and proteus simulation based on single-chip microcomputer. *IEEE*.
- [8] LI Zong-qiang LIU Li-jun. Design and simulation of led clock circuit based on proteus. *International Conference on Computer, Mechatronics, Control and Electronic Engineering*, pages 315–316, 2010.
- [9] W. Chen S. K. Tolpygo J. R. Friedman, V. Patel and J. E. Lukens. The construction of single-chip microcomputer virtual experiment platform based on proteus. *The 5th International Conference on Computer Science and Education*, 2010.
- [10] Ahmad Shukri Bin Fazil Rahman and Abdul Rahim Bin Abdul Raza. Proteus based simulation of a charge controller. *IEEE Conference on Power and Energy*, pages 539–543, 2010.
- [11] Song Kefei HAN Zhenwei. Design of thermostat system based on proteus simulation software. *International Conference on Electronic and Mechanical Engineering and Information Technology*, pages 1901–1904, 2011.
- [12] Pan Jinfeng i Xu Xiumei. The simulation of temperature and humidity control system based on proteus. *International Conference on Mechatronic Science, Electric Engineering and Computer*, pages 1896–1898, 2011.
- [13] Rasli Abd Ghani Noorradiah Ismail. Advance devices using piezoelectric harvesting energy. *IEEE Student Conference on Research and Development*, pages 451–453, 2013.
- [14] Sarvesh SS Rawat Karthik Srinivasam Vikram Puri Jartin Aroram, Gagandeep. Design and develepment of digital voltmeter using differents techniques.