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

DIEGO SANTANA TORGA

DESENVOLVIMENTO DE UMA PLATAFORMA DIDÁTICA PARA PRÁTICAS DE CONTROLE DE VELOCIDADE DE MOTOR DE CORRENTE CONTINUA

MONOGRAFIA DE GRADUAÇÃO EM ENGENHARIA
DE CONTROLE E AUTOMAÇÃO

Ouro Preto, 2016

DIEGO SANTANA TORGA

DESENVOLVIMENTO DE UMA PLATAFORMA DIDÁTICA PARA PRÁTICAS DE CONTROLE DE VELOCIDADE DE MOTOR DE CORRENTE CONTINUA

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

Orientador: Prof. MSc. José Alberto Naves Cocota Júnior

Ouro Preto

Escola de Minas – UFOP

Março/2016

Monografia defendida e aprovada, em 11 de março de 2016, pela comissão avaliadora constituída pelos professores:

Prof. M.Sc. José Alberto Naves Cocota Júnior - Orientador

Prof. M.Sc. João Carlos Vilela de Castro – Professor Convidado

Prof. Dr. Paulo Marcos de Barros Monteiro – Professor Convidado

T682d Torga, Diego Santana.

Desenvolvimento de uma plataforma didática para práticas de controle de velocidade de motor de corrente contínua [manuscrito] / Diego Santana Torga. – 2016.

74f. : il., color., graf., tab.

Orientador: Prof. MSc. José Alberto Naves Cocota Júnior.

Fonte de catalogação: bibem@sisbin.ufop.br

AGRADECIMENTOS

Primeiramente, agradeço a Deus, por me iluminar no processo de transferência e, assim começar a buscar meu sonho, ser um engenheiro de Controle e Automação. Agradeço imensamente aos meus pais, pois foram eles meu alicerce durante toda trajetória. A minha namorada Karina, que esteve ao meu lado desde o início dando total apoio e motivação. A todos meus amigos pelos momentos vividos e aprendizados de vida compartilhados. A Escola de Minas, à Universidade Federal de Ouro Preto e a todos os mestres que souberam me passar os seus conhecimentos de forma clara e objetiva, contribuindo para minha formação pessoal e profissional. Sou grato à REPÚBLICA EXÍLIO juntamente com toda família exilada, no qual tive um grande crescimento pessoal, pelos vários desafios enfrentados e pelos momentos de alegrias jamais esquecidos.

Agradeço ao professor COCOTA pela confiança depositada em mim ao me conceder este projeto iniciado no programa de Pró-Ativa e por me orientar de forma brilhante na conclusão deste trabalho, apesar de todas as dificuldades. Agradeço a todos aqueles que de alguma forma diretamente ou indiretamente me ajudaram para que este trabalho fosse concluído.

RESUMO

Este trabalho apresenta o desenvolvimento de uma plataforma didática para realização de práticas de controle de velocidade de motor de corrente contínua. A bancada é basicamente composta por um motor de corrente contínua, um motor de polos sombreados, que atua como distúrbio (carga) no sistema, um circuito microcontrolado (Arduino UNO) e um circuito para o acionamento dos motores (módulo de potência). Para o controle de velocidade do motor de corrente contínua é utilizado o controlador PID de velocidade discretizado, que se encontra implementado no Arduino. Para interface com o usuário, foi desenvolvido um sistema supervisório no *software* Matlab por meio do *toolbox* GUIDE. O sistema supervisório possibilita ao usuário realizar o experimento da curva de reação ao degrau para determinar os ganhos do controlador PID por meio das sintonias por Ziegler-Nichols e por Cohen-Coon. A bancada pode ser operada em modo manual (malha aberta), tendo como entrada os sinais analógicos de tensão dos potenciômetros referente ao motor de CC e ao motor de polos sombreados, e em modo automático (malha fechada), tendo como referência a velocidade de operação desejada do motor de CC. No modo de operação automático é possível, no sistema supervisório, o acionamento do motor de polos sombreados. As variáveis de interesse (sinal de saída, sinal de controle, erro e referência) são salvas em arquivos com extensão *.mat*, que podem ser carregados por meio do sistema supervisório.

Palavras-chave: Motor de Corrente Contínua, Plataforma Arduino UNO, Controle de Velocidade, MATLAB.

ABSTRACT

This paper presents the development of a didactic platform for speed control practices of direct current motor The bench is basically composed of a direct current motor, a shaded pole motor that acts as disturbance (load) system, a microcontrolled circuit (Arduino UNO) and a circuit for actioning the motors (power module). For the direct current motor speed control is used PID controller discretized speed, which is implemented in the Arduino. For the user interface, is developed a supervisory system in the Matlab *software* through the *toolbox* GUIDE. The supervisory system allows the user to perform the experiment the reaction of the step curve to determine the gains of the PID controller through tunings by Ziegler-Nichols and by Cohen-Coon. The bench can be operated in manual mode (open-loop), taking as input the analog voltage signals of the potentiometers referent the DC motor and the shaded pole motor, and automatic mode (closed loop), taking as the reference speed desired operation of the DC motor. In automatic mode of operation it is possible in the supervisory system, the drive shaded poles motor. The variables of interest (output signal, control signal, error and reference) are saved in files with *.mat* extension, which can be loaded through the supervisory system.

Key words: Direct current motor, Arduino UNO platform, speed control, MATLAB

LISTA DE FIGURAS

LISTA DE TABELAS

Sumário

1 INTRODUÇÃO

Para Dorf e Bishop (2001), a característica mais pertinente na engenharia de controle é a oportunidade de aplicar o conhecimento e o controle em sistemas físicos como máquinas e processos industriais de forma benéfica para a sociedade por meio de projetos de controladores eficazes.

A primeira maneira utilizada pelo homem a fim de controlar um processo é o controle manual, que requer operadores com habilidades e conhecimentos do sistema, e que está presente ainda em muitos processos. Com a demanda no aumento da eficiência operacional das instalações, redução dos custos operacionais, segurança nos processos, conforto nas atividades e qualidade no produto final houve a necessidade do controle automático.

O controle automático significa um papel vital no avanço da engenharia e da ciência, posto que, a utilização de controladores bem sintonizados possibilita que sejam alcançados os resultados desejados nos processos industriais como: controle de velocidade de um motor, temperatura e/ou pressão de um forno, nível de um tanque, fator de potência das instalações, etc., com isso havendo a necessidade dos engenheiros de controle compreender bem esta área da engenharia (OGATA, 2002).

Cursos como Engenharia Elétrica, Engenharia Mecatrônica e Engenharia de Controle e Automação reúnem em suas grades curriculares disciplinas com finalidade de apresentar os conceitos de controle de sistemas como: modelagens matemáticas, malhas de controle, desempenho e estabilidades de sistemas, métodos de controle e projetos de controladores. Esse embasamento teórico adquirido, às vezes não é suficiente para que o aluno compreenda as aplicações reais de controle, tal como os métodos que são empregados a fim de obter os parâmetros do controlador PID e qual a interferência desses parâmetros no controlador. Com isso, há a necessidade de utilizar ferramentas que auxiliem nesta compreensão (KARDEK; COCOTA e FERREIRA, 2015).

A utilização de bancadas didáticas e outros recursos tecnológicos, que auxiliem o processo ensino-aprendizado na teoria de controle, proporcionando experimentos e projetos práticos que complementem a formação de um profissional, estão se tornando comuns no ensino de graduação nas áreas da engenharia (SHUI-CHUN e CHING-CHIH, 2009) (ARAÚJO; FREITAS e SILVA, 2011). Esta prática tornou-se uma indispensável metodologia para motivar os alunos de graduação, uma vez que “[...] o método tradicional – apresentar ao estudante não o problema, mas a solução acabada – priva este estudante de todo a excitação,

corta o impulso criativo e reduz a aventura da humanidade a um monte empoeirado de teoremas" (DORF e BISHOP, 2001). No entanto, a maioria das plataformas experimentais disponíveis e aplicáveis para a graduação ou pós-graduação são muito caras e/ou tem uma arquitetura de *hardware* e *software* fechado. Além disso, o uso exclusivo destas plataformas impede a possibilidade dos alunos de desenvolver uma das características mais importantes de um profissional de engenharia, que é a capacidade de projetar, gerenciar e executar um projeto (COCOTA *et al*., 2013).

Com o programa Pró-Ativa, criado em 1999, uma ação da Pró-Reitoria de Graduação da UFOP (PROGRAD) destinada a contribuir para a melhoria do ensino de graduação que foi desenvolvido o presente trabalho, o qual tem como objetivo desenvolver uma bancada didática para auxiliar no ensino de disciplinas de controle e apoiar projetos que visem o desenvolvimento de metodologias e tecnologias de apoio à aprendizagem.

1.1 Objetivo Geral

Desenvolvimento de uma plataforma didática de baixo custo, para práticas de controle de velocidade de motor de corrente contínua (CC), sendo esta utilizada nas disciplinas de controle, auxiliando no processo de ensino-aprendizagem, visto que o aprendizado teórico associado à prática é de suma importância para a formação de um bom profissional.

1.2 Objetivos Específicos

- Desenvolver uma ferramenta que auxilie o desenvolvimento de práticas nas disciplinas de instrumentação, programação de computadores, acionamentos elétricos, eletrotécnica, eletrônica digital/analógica, sistemas embutidos e teoria de controle.
- Entender a importância de um bom projeto dos sistemas de controle;
- Aplicar por meio da teoria de controle digital o controlador PID de velocidade discretizado no *firmware* do Arduino;
- Compreender os métodos de sintonia de PID de Ziegler-Nichols e Cohen-Coon por meio dos parâmetros encontrados no ensaio da curva de reação ao degrau.

1.3 Justificativa do Trabalho

A razão para desenvolvimento desse trabalho se deve ao fato de plataformas de controles educacionais existentes no mercado, em geral, possuem preços elevados e apresentam arquitetura fechada de *software* e *hardware*, que restringe a possibilidade do aluno de empregar técnicas de controles em sistemas reais.

1.4 Estrutura do Trabalho

O presente trabalho foi dividido em 5 capítulos. O Capítulo 1 apresenta um contexto introdutório sobre o trabalho, apresentando seus principais objetivos e a justificativa. No Capítulo 2 apresenta-se o referencial teórico do projeto, abordando os principais conceitos e aplicações de uso sobre Arduino, sistemas SCADA e teoria de controle, apresentando alguns tipos de controladores e suas particularidades. O Capítulo 3 apresenta a metodologia utilizada na elaboração do presente trabalho, com foco no desenvolvimento e montagem da bancada, abordando seu funcionamento. No capítulo 4 são apresentados os resultados obtidos em diferentes ensaios e no Capítulo 5 é realizada uma breve conclusão frente aos resultados, além de sugestões para trabalhos futuros.

2 REFERENCIAL TEÓRICO

2.1 Introdução ao Arduino

O Arduino é uma placa microcontrolada de código aberto que conta com um ambiente de desenvolvimento integrado (*IDE*). A plataforma Arduino foi desenvolvida com o escopo de dar facilidade para iniciantes que não tenham experiências com desenvolvimento de eletrônica e *software*, e até mesmo para os que já contemplam algum conhecimento e queiram se aprofundar. Com o *hardware* é possível interpretar o mundo externo, os sistemas físicos, por meio da leitura de sensores, chaves, etc., e atuar nas variáveis manipuladas como motores, displays, led's, relés e outros dispositivos que possam ser acionados pelos sinais de saídas. A interface responsável para integrar entradas e saídas é a *IDE*, plataforma de código aberto (*open-source*), que se trata de um ambiente de programação que suporta a linguagem de programação C/C++ (GOMES e TAVARES, 2013).

Existem diversos modelos de plataformas Arduinos: NANO, MEGA, DUE, UNO entre outros; nos quais as principais diferenças entre eles são: a capacidade de processamento, o tamanho do *hardware*, a memória, a quantidade de entradas e saídas.

2.1.1 Arduino UNO

O Arduino UNO (Figura 2.1), *hardware* este utilizado na plataforma, contém dispositivos necessários para apoiar o microcontrolador: cristal de 16Mhz; conector USB; botão reset e o microcontrolador Atmega16U2, responsável pela interface USB para comunicação com o computador e possibilita o *upload* do código binário, gerado após compilação do programa. Basta conectá-lo ao computador pela USB ou por uma fonte externa, que está pronto para utilizar (ARDUINO, 2015).

Figura 2.1 - Arduino UNO

Fonte: Arduino,2015.

2.1.2 Características Básicas

A Tabela 2.1 apresenta as características básicas do Arduino UNO.

Tabela 2.1 - Características Arduino UNO

Microcontrolador	Atmega328P
Tensão Operacional	5V
Tensão de Alimentação (recomendada)	7 - 12V
Tensão de Alimentação (limites)	6 – 20V
Pinos I/O Digitais	14 (dos quais 6 podem ser saídas PWM)
# Pinos de Entrada Analógica	6
Corrente Contínua por Pino I/O	20mA
Corrente Contínua do Pino 3.3V	50mA
Memória Flash	32KB (Atmega328P) dos quais 0.5KB são usados para o *bootloader*
SRAM	2KB
EEPROM	1KB
Frequência do Clock	16MHz

Fonte: Arduino, 2015.

2.1.3 Microcontrolador

O microcontrolador pode ser definido como um “pequeno” componente eletrônico, o qual se encarrega de executar processos lógicos e operações matemáticas por meio de sua Unidade Lógica Aritmética (ULA), que está contida na unidade de processamento central (CPU). Encapsulado em uma única pastilha de silício (conhecido como CI – circuito integrado), este contém todos periféricos necessários para seu funcionamento; unidade de memória, unidade de processamento (CPU), portas de entrada e saída, comunicação serial, *timers*, unidade de temporização, *Watchdog*, PWM’s e conversores analógicos – digitais (SOUZA, 2005).

2.1.4 Alimentação

De acordo com o site do Arduino (2015), a plataforma Arduino UNO pode ser alimentado, pela conexão USB ou fonte externa, sendo esta seleção realizada de forma automática. Para alimentação externa utiliza-se a opção do plug P4 ou o conector de entrada Vin. A plataforma

pode operar com 6 a 20V no caso da alimentação externa, mas tensões proveniente desta alimentação, inferiores a 7V podem fazer com o que o sistema se torne instável devido ao pino de 5V não conseguir suprir seu fornecimento, e para tensões superiores a 12V podem superaquecer o regulador de tensão e danificar a plataforma. Abaixo segue outros pinos de alimentação:

- 5V: A fonte de alimentação utilizada para o microcontrolador e para outros componentes da placa. Pode ser proveniente do pino Vin através de um regulador *on-board* ou ser fornecida pela porta USB;
- 3V3: Alimentação de 3,3 V fornecida pelo circuito integrado FTDI (controlador USB). A corrente máxima é de 50 mA;
- GND (*ground*): Pino terra.

2.1.5 Portas de Entradas e Saídas

Para interface com o mundo externo, o microcontrolador dispõe de portas de E/S (Entrada – Saída), que possibilitam obter leituras das variáveis a serem controladas e atuar nas variáveis de saídas (IBRAHIM, 2006).

Para os 14 pinos de entradas e saídas digitais, utilizamos as funções *digitalRead()* e *digitalWrite()* para leitura e escrita respectivamente. Cada pino pode fornecer ou receber no máximo 40mA para não ocasionar danos permanentes ao microcontrolador. Para os 6 pinos de entradas analógicas (A0,...,A5) utiliza-se a função *analogRead(),* sendo esses com resolução de 10bits e, que convertem o sinal analógico para um valor de 0 a 1023. Para todos esses pinos é utilizado a função *pinMode(pin, estado)*, no qual, *pin* define a porta e estado se este é entrada (*Input*) ou saída (*Output*) na declaração do programa.

Seguem exemplos de alguns pinos com funções especializadas:

- Serial: 0 (RX) e 1 (TX). Usados para receber (RX) e transmitir (TX) dados seriais TTL. Estes pinos são conectados aos pinos correspondentes do circuito integrado FTDI que converte sinais USB-TTL;
- PWM: 3, 5, 6, 9, 10, e 11. Fornecem saída de PWM de 8 bits com a função *analogWrite()*;
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Estes pinos suportam comunicação SPI;
- Led: 13. Existe um LED *onboard* conectado ao pino digital 13;

- AREF. Referência de tensão para entradas analógicas. Usados com a função *analogReference()*;
- Reset: Envia o valor LOW para o pino 1 do microcontrolador, reiniciando-o.

2.1.6 Memórias

O Atmega328p tem 32 KB de memória flash (onde são armazenados os programas), além de 2 KB de SRAM (onde ficam as variáveis criadas no programa) e 1 KB de EEPROM (Electrically-Erasable Programmable Read-Only Memory), que pode ser lida e escrita através da biblioteca EEPROM. É nessa memória que são armazenados os dados que precisam ser permanentes, para segurança de uma eventual queda de energia, e os dados guardados na SRAM, são perdidos na situação supracitada por se tratar de uma memória volátil (ARDUINO, 2015).

2.1.7 Interrupções

Conforme coloca Souza (2005), as interrupções desempenham uma função fundamental, pois são elas que interrompem a execução do programa imediatamente para tomadas de atitudes instantâneas. Esta tratativa é realizada diretamente pelo *hardware* tornando a ação muito mais rápida e disponível em qualquer ponto do sistema. Ocorrida uma interrupção à execução do programa é interrompida e direcionada para a função definida pelo programador, a qual será executada, e após seu término o programa volta de onde este foi desviado. A Figura 2.2 apresenta de forma intuitiva o que ocorre quando é gerada uma interrupção.

Figura 2.2 - Interrupções
Fonte:MIYADAIRA,2009

2.1.7.1 Interrupções Externas

Interrupções externas são interrupções que são provocadas por sinais externos ligados as portas específicas do microcontrolador. Desta forma é possível executar funções necessárias quanto ao projeto em desenvolvimento na ocorrência de um sinal que necessite de uma ação imediata (SOUZA, 2005).

Os pinos 2 e 3 são as portas utilizadas para estas interrupções, sendo configuradas pela função *attachInterrupt(interrupt, function, mode).*

- *Interrupt* – Pino que será verificado a interrupção - *int* 0 pino 2 e *int* 1 pino 3;
- *Function* – Função que será executada quando houver a interrupção;
- *Mode* – Existem 4 modos de detecção de interrupção.
    - *Rising* – Executa quando o pino vai de baixo para alto;
    - *Filling* – Executa quando o pino vai de alto para baixo;
    - *Change* – Executa sempre que os pinos mudam de estados;
    - *Low* – Executa quando o pino é baixo.

2.1.7.2 Interrupções Por Timers

Timer ou ainda contador, são recursos que os microcontroladores utilizam para executar contagem de tempos. Estas interrupções ocorrem sempre que estes *timers* estouram sua contagem ou atingem valores especificados pelo programador por meio de configurações realizadas nos registradores dos respectivos *timers*. A contagem de tempo pode ser realizada pelo incremento interno do *clock* da máquina como também por um sinal externo. Desta forma se torna muito importante, pois, conseguimos definir intervalos de tempos o qual será executada a função necessária (SOUZA, 2005).

Os *timers* presentes no Arduino UNO são:

- *Timer0 – Timer* com resolução de 8bits responsável pelas funções *delay(), millis()* e *micros()*;
- *Timer1 – Timer* com resolução de 16bits;
- *Timer2 – Timer* com resolução de 8bits.

Seguem alguns registradores importantes para estas configurações:

- *TCCRxA* e *TCCRxB* – (*Timer/Counter Control Register);*
- *OCRxA - (Output Compare Register);*

- *TIMSKx – (Timer/Counter Interrupt Mask Register).*
  Onde "x" é o número do *timer* em questão.

Para interrupções geradas por meio do *clock* interno,(com frequência de 16MHz), e sabendo a resolução do *timer*, podemos obter o valor máximo que este pode conter, e partir disso, definir o intervalo que será gerada a interrupção. Esta é acionada pela rotina de serviço de interrupção (*Interrup Service Routine*, ISR). Sua chamada é realizada pela função *ISR(TIMERx_COMPA_vector)*, onde "x" é o *timer* em questão.

A Equação 2.1 possibilita determinar o limite de contagem do *timer*, e a Equação 2.2 possibilita determinar o tempo para as interrupções.

$$\text{Contagem} = (2^{n} - 1) \tag{2.1}$$

onde n – é o número de bits do *timer*

$$\text{Tempo} = \left(\frac{1}{16\text{MHz}} + \text{Contagem}\right) \tag{2.2}$$

Em muitas situações este tempo é muito pequeno e se faz necessário o uso de *prescaler* (divisor), o que proporciona o aumento do período do *timer* fazendo a divisão do sinal de *clock* em potências de 2. Para aumentar mais ainda a precisão deste período, pode-se utilizar o modo de operação *CTC (Clear Timer on Compare Match)* que não executa a *ISR* em caso de *overflow*, mas sim na comparação de um valor calculado pelo programador que assume o registrador *OCRxA (Output Compare Register)*, onde "x" é o *timer* em questão. Este cálculo define o intervalo de tempo da interrupção. Para as configurações do *prescaler* e ativação do módulo *CTC*, devem ser analisados quais os bits devem estar em 1 ou 0 conforme o *datasheet* (CARLOS; MORAES e RODRIGO).

A Equação 2.3 apresenta qual valor o registrador OCRxA deve receber para executar a interrupção de acordo com o tempo definido pelo programador.

$$OCRxA = \left(\frac{16\text{MHz}}{\text{Prescaler}} \text{x tempo}_{\text{escolhido}}\right) - 1 \tag{2.3}$$

As limitações deste tempo estão de acordo com a capacidade de contagem do *timer*. Logo, para que esta interrupção ocorra de forma correta o tempo escolhido deve ser um valor menor ou igual que a contagem dada pela Equação 2.1.

2.1.8 PWM

PWM é uma técnica que consiste em variar a tensão entregue as cargas, aplicando uma onda quadrada e mantendo uma frequência fixa e variando somente o ciclo ativo desta onda. Esta técnica permite uma variação entre 0 e 100% da tensão média entregue, conforme apresentado na Figura 2.3 e na Equação 2.4 (PEREIRA, 2005).

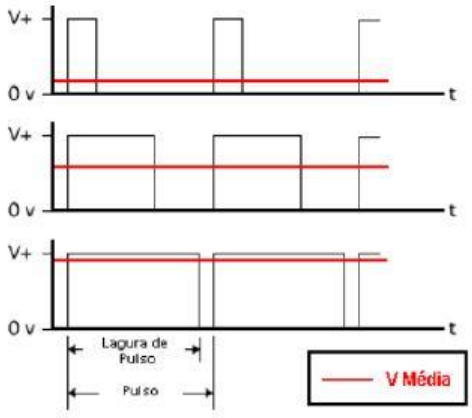


Figura 2.3 - Sinal de PWM

Fonte: PEREIRA, 2005

$$Vmédia = \frac{1}{T}\int_0^T v(t)dt \tag{2.4}$$

Onde:

$Vmédia$: valor da tensão média em CC;

T: período do sinal;

$v(t)$: valor instantâneo da tensão.

Os pinos 3, 5, 6, 9, 10 e 11 são os que permitem o uso deste recurso. A frequência padrão utilizada é 490hz, podendo esta ser alterada de acordo com as necessidades do projeto. Estas alterações são realizadas configurando os respectivos *timers* por meio de seus registradores:

- Pinos 5 e 6 – *Timer0* – (nestes pinos devem tomar cuidado, pois como o *Timer0* é responsável pelas funções de tempo, descritas na subseção 2.1.7.2, estas podem ser afetadas);
- Pinos 9 e 10 – *Timer1*;
- Pinos 3 e 11 – *Timer2*.

A seguir é apresentada a linha de código utilizada no *firmware* da Arduino, que tem a função alterar a frequência de PWM de 490hz para 3906hz, que é utilizada na bancada.

TCCR2B = TCCR2B & 0b11111000 | 0x02;

Esta alteração da frequência é necessária para o funcionamento adequado do motor CC, que não opera de maneira desejada em baixas frequências do PWM.

2.1.9 Comunicação Serial

A comunicação serial existente entre o Arduino e o computador é realizada no padrão UART TTL (5V) e está disponível nos pinos digitais 0 (RX) e 1 (TX). O CI responsável por promover esta comunicação pela porta USB é o Atmega16U2, o qual cria uma porta COM virtual no computador. O *IDE (Integrated Development Environment)* do Arduino inclui um monitor serial (*Serial Monitor),* onde é possível visualizar dados simples de textos que estão sendo enviados e/ou recebidos (ARDUINO, 2015).

2.1.10 IDE

O IDE é o ambiente onde são desenvolvidos os programas que serão compilados e enviados para o microcontrolador do *hardware* do Arduino. Conforme apresentado na Figura 2.4 a interface gráfica construída em Java é fácil e simples de utilizar e é compatível com os sistemas operacionais Windows, Mac OS X e Linux. Para desenvolvimento do programa é necessário utilizar a função *Setup()* que é responsável pela inicialização das variáveis, modo dos pinos entre outras necessárias e a função *Loop()* que é responsável pela execução do algoritmo de forma repetitiva (ARDUINO, 2015).

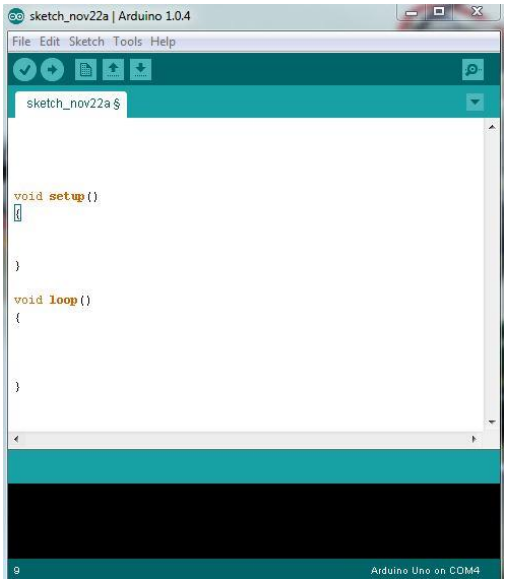


Figura 2.4 - IDE Arduino

2.2 Transistores

Transistor é um dispositivo semicondutor formado por três camadas: PNP ou NPN dependendo de sua construção física. Independente desta construção, as camadas da extremidade são chamadas de emissor e coletor e a camada central de base. O resultado entre as junções dessas camadas (P-N-P ou N-P-N), chamada "camada de depleção" é que para ocorrer fluxo de corrente é necessária uma diferença de potencial maior que 0,7V entre as camadas para o material de silício, utilizado na construção do componente, e na temperatura de 25°C (MALVINO, 1997).

Com o transistor é possível controlar a corrente entre emissor e coletor de acordo com a corrente aplicada na base. Pequenas variações de corrente na base provocam grandes variações de correntes entre emissor e coletor (KARDEK e RODRIGUES, 2015).

Na Figura 2.5 encontra-se ilustrada a simbologia do transistor NPN com os respectivos sentidos das correntes de coletor (I_C), base (I_B) e emissor (I_E), modelo utilizado no acionamento dos motores da bancada deste trabalho.

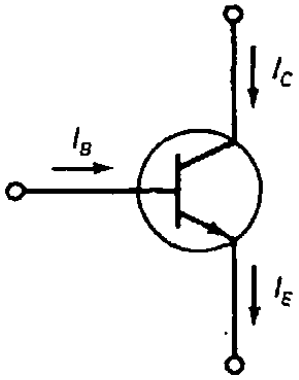


Figura 2.5 - Símbolo Transistor NPN

O transistor NPN apresenta algumas regiões de operação:

- Saturação – Funciona como uma chave fechada, corrente de base alta;
- Ativa – Funciona como um amplificador de sinal, de acordo com a corrente de base. Muito utilizado para sinais fracos, como de rádio e TV;
- Corte – Funciona como uma chave aberta, corrente de base muito baixa próxima de zero ou nula.

Na Figura 2.6 tem-se ilustrado o circuito EC (emissor comum) utilizado como base no circuito eletrônico de acionamento dos motores, no qual Vcc é a fonte externa, Rc a carga do motor CC ou motor de polos sombreados e Vbb sinal do Arduino. Para a carga Rc o módulo

de potência contará com um diodo de roda livre, em função da carga indutiva presente nos motores, conforme será discutido na seção 3.2.2.

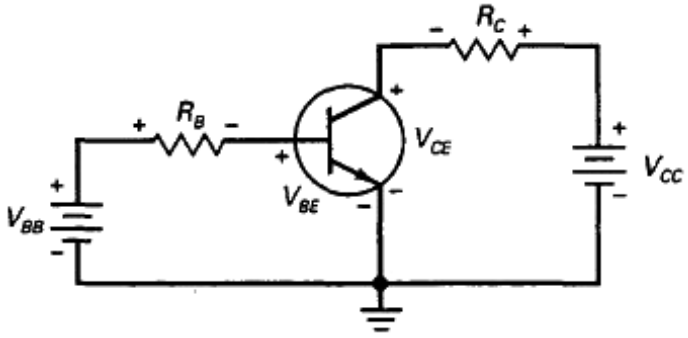


Figura 2.6 - Transistor Ligação Emissor Comum

Utilizando a lei das correntes de Kirchhoff na Figura 2.5 temos:

$$I_E = I_C + I_B \tag{2.5}$$

$$B_{CC} = \frac{I_C}{I_B} \tag{2.6}$$

onde B_{CC} - representa o ganho de corrente do transistor.

De forma análoga, a partir da Figura 2.6 temos que I_B é dado por:

$$I_B = \frac{V_{BB} - V_{BE}}{R_B} \tag{2.7}$$

2.3 Sensores

Sensores são dispositivos capazes de relacionar uma grandeza física (velocidade, força, pressão, deslocamento, etc.) por meio de um sinal elétrico, podendo este ser um sinal digital ou analógico. Para leituras de sensores analógicos em um sistema microprocessado é necessária a conversão analógico-digital do sinal conforme apresentado na seção 2.6.5, já para sensores digitais, os mesmos podem ser ligados diretamente a estes sistemas, desde que atendam suas especificações (IBRAHIM, 2006).

Para Ibrahim (2006), os sensores são dispositivos fundamentais nos sistemas de controle em malha fechada, pois, por meio da leitura da variável medida e sua comparação com o valor de referência se obtém o erro, este utilizado no controlador.

2.3.1 Sensor de Posição - *Encoder*

O *encoder* é um transdutor que converte um movimento angular ou linear em uma série de pulsos digitais elétricos. Estes pulsos gerados podem ser usados para determinar velocidade, taxa de aceleração, distância, rotação, posição ou direção.

O *encoder* é basicamente formado por uma fonte de luz infravermelha, um fotosensor e um disco com secções opacas e transparentes. Estes sensores podem ser do tipo absoluto ou incremental. Na Figura 2.7 temos o exemplo de um *encoder* de deslocamento angular incremental, no qual o deslocamento angular é determinado pela contagem dos pulsos de luz que atravessa as seções transparentes e alcança o receptor de sinal luminoso (fotosensor), incrementando ou decrementando a contagem de acordo com o sentido de rotação.

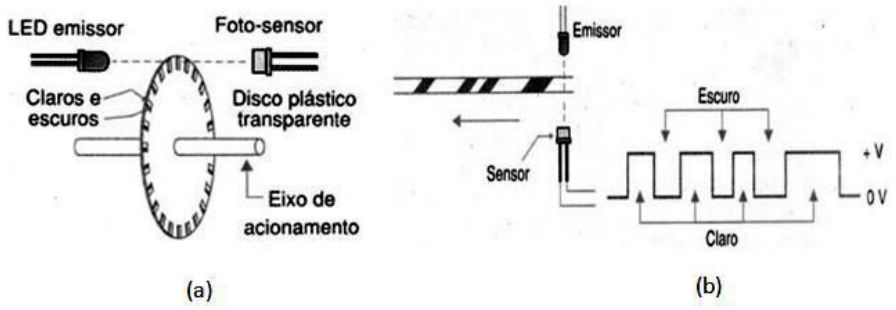


Figura 2.7 - (a) Esquemático do Encoder, (b) Sinal de saída

Fonte: www.newtoncbraga.com.br

A partir da medida da posição, pode-se determinar a velocidade ao derivar a posição no tempo. Esta diferenciação pode ser realizada por meio do *hardware* (amplificadores operacionais, por exemplo) ou pelo sistema microprocessado. Neste trabalho a velocidade é determinada pela derivada da posição pelo sistema microprocessado.

O encoder utilizado é apresentado na seção 2.4.1, por se tratar de um conjunto motor-redução-*encoder*.

2.4 Motores Elétricos

Motores elétricos é um dispositivo com capacidade de converter energia elétrica em energia mecânica. Esta energia final é utilizada para o acionamento de diversos tipos de equipamentos e máquinas e tem grande aplicação industrial, como no transporte de cargas, processamento de materiais, transporte de fluídos, dentre outros. A utilização em grande escala de motores elétricos se deve também à sua relação com o tipo de energia utilizada (energia elétrica), considerada uma energia limpa e de baixo custo. Estima-se que cerca de 40% da energia

elétrica consumida no país é destinada ao acionamento de motores elétricos em geral (FILIPPO, 2000).

2.4.1 Motor de Corrente Contínua (CC)

Motores CC caracterizam-se por sua versatilidade. Por possuir grande facilidade de controle, estes são muito utilizados em aplicações que se desejam amplo controle de velocidades ou de um controle preciso da saída do motor. Com a tecnologia de estado sólido em ascensão, muitas das aplicações estão substituindo os motores CC por motores CA. Entretanto, devido a sua versatilidade e seu sistema de acionamento simplificado, assegura-se o uso continuado de motores CC em uma ampla variedade de aplicações (FITZGERALD; CHARLES e UMANS, 2006).

Neste trabalho foi utilizado o motor CC modelo "JGA25-371 DC" (Figura 2.8), que trabalha com tensão nominal de 12V. Este motor possui um *encoder* acoplado em seu eixo que geram dois sinais defasados de 90º, conhecido como *encoder* em quadratura, sendo possível interpretar o sentido de rotação, com resolução de 334 pulsos por revolução completa. Este motor também contempla um redutor na saída de seu eixo com relação de redução de 1:21.3. Na Tabela 2.2 temos apresentados dados referentes ao motor supracitado.

Tabela 2.2 - Característica Motor CC

Tensão		Sem Carga (Motor de Pólos Sombreados 0Vcc)		Com Carga (Motor de Pólos Sombreados 33.9Vcc)	
Faixa	Nominal	RPM	Corrente	RPM	Corrente
6 – 24V	11.6Vcc	192	48mA	192	63mA

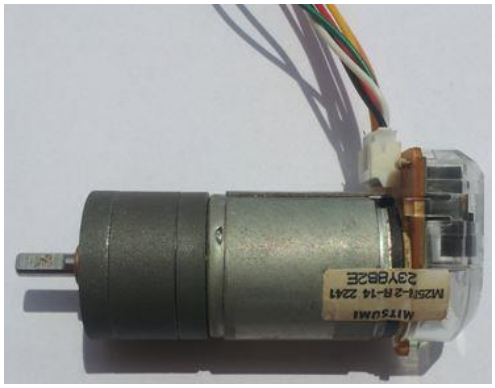


Figura 2.8 - Motor CC

Segue informações importantes quanto aos cabos do motor CC deste trabalho:

- Laranja – Alimentação Motor CC;
- Amarelo – Alimentação Motor CC;
- Vermelho – Alimentação dos *Encoders* (5Vcc);
- Preto – GND do *Encoder*;
- Branco – Sinal de Saída do *Encoder*;
- Verde – Sinal de Saída do *Encoder* (defasado em 90º com relação ao anterior).

OBS: A ligação dos cabos Laranja/Amarelo nos bornes da placa de acionamento, polo positivo (+) e polo negativo (-), irão simplesmente determinar o sentido de rotação do motor.

2.4.2 Motor de Indução de Polos Sombreados

Motores de indução de polos sombreados têm usualmente polos salientes com uma porção de cada polo envolvida por uma espira de cobre em curto-circuito chamada *bobina de arraste*. O resultado das correntes induzidas nesta bobina é a produção de um campo girante que se move no sentido da porção não sombreada para a porção sombreada do polo, produzindo o torque que fará o motor partir e atingir sua velocidade nominal. Estes motores produzem baixo conjugado de partida e estão disponíveis com potências nominais de até cerca de 50W e podem ser encontrados em uma ampla variedades de equipamentos incluindo refrigeradores, ventiladores, bombas, máquinas de lavar e secadores (FITZGERALD; CHARLES e UMANS, 2006). Na Figura 2.9 temos o motor de polos sombreados utilizado neste trabalho, que tem como objetivo ser um distúrbio para o sistema.

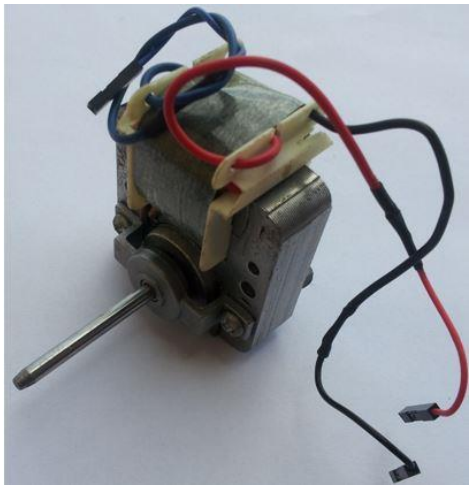


Figura 2.9 - Motor de Polos Sombreados

Segue informações importantes quanto aos cabos do motor de polos sombreados deste trabalho:

- Vermelho + Preto – Resistência Ôhmica de 140ohms;
- Preto + Azul - Resistência Ôhmica de 157ohms;
- Vermelho + Azul - Resistência Ôhmica de 17ohms.

OBS: Não se deve ligar o cabo vermelho/azul devido a sua baixa resistência e risco de danificar circuito de potência.

2.5 Sistema Supervisório

São sistemas capazes de processar informações do processo e torná-la disponível para o operador ou qualquer outro usuário para interesses específicos. Conhecido como sistemas SCADA (*Supervisory Control And Data Acquisition*) permitem monitoramento do processo em tempo real, assessorando tomadas de decisões automáticas ou por iniciativa do operador do processo (DE SOUZA, 2005).

Sistemas supervisório desempenham três atividades básicas:

- Supervisão – Monitoramento do processo por meio de gráficos de tendência de variáveis analógicas e/ou digitais;
- Operação – Substitui as funções da mesa de controle, otimizando o processo de ligar e desligar equipamentos ou ainda mudar o modo de operação dos equipamentos de controle;
- Controle – Responsável por definir os *set-points* do mecanismo de controle dinamicamente. O algoritmo de controle é executado em unidades de processamento separado chamado CLP (Controlador Lógico Programável). Algoritmos estes também executados em microcontroladores.

Para elaboração do sistema supervisório utiliza-se o *Toolbox* GUIDE do Matlab_R2013A. Este *Toolbox* contempla vários objetos, botões, telas, caixa de textos, caixa de seleção, gráficos, painéis, *slider* dentre outros, os quais atendem todos os recursos propostos para a supervisão, operação e controle da bancada.

2.6 Sistema de Controle Clássico

Segundo Ogata (2002), um sistema de controle tem por objetivo alcançar resultados desejados para o sistema (este não limitado a algo físico), a partir de uma combinação de componentes que atuam em conjunto. Um sistema pode ser definido como um bloco que recebe entradas e

produz saídas com respostas a estas entradas. Para estes tipos de sistemas que representam uma relação entrada-saída, existem dois sistemas de controle conhecidos:

- Sistemas de controle em malha aberta – são sistemas o qual a saída (variável medida) não exerce nenhuma influência na ação de controle. Não existe retroalimentação do sinal para comparação com a entrada, conforme apresentado na Figura 2.10.

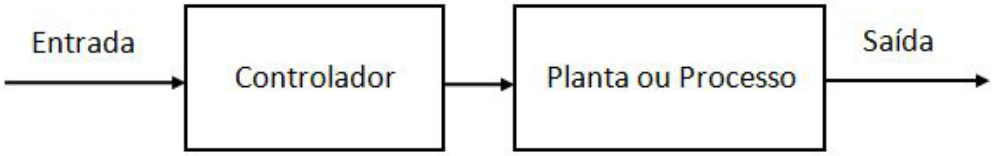


Figura 2.10 - Controle Malha Aberta

- Sistemas de controle em malha fechada – são sistemas o qual a saída (variável medida) exerce um efeito direto na ação de controle. São sistemas retroalimentados que tem o objetivo de obter um erro entre entrada e saída, e este alimentar o controlador a fim de reduzir este erro atuando na variável de manipulada, conforme apresentado na Figura 2.11.

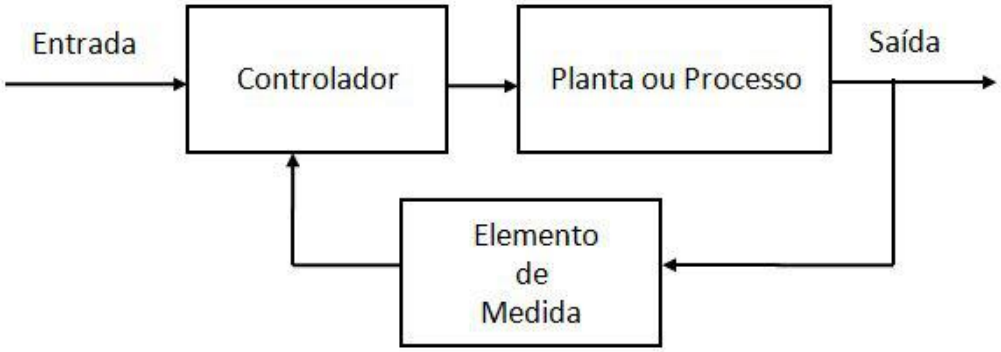


Figura 2.11 - Controle Malha Fechada

Dado que os objetivos, de acordo com Ogata (2002), sejam obter resultados desejados para os sistemas, necessitamos compreender alguns tipos de controladores, estes adequados para cada aplicação. Os controladores automáticos industriais clássicos podem ser classificados de acordo com a ação de controle.

2.6.1 Controlador Proporcional (P)

O controlador proporcional atua na variável manipulada proporcionalmente ao erro e(t), este sendo o resultado da diferença entre o valor desejado (*set-point*) e a grandeza medida. Kp é

conhecido como o ganho do controlador (CAMPOS e TEIXEIRA, 2006). A ação de controle u(t) para o controlador é expressa por:

$$u(t) = K_P\, e(t) \tag{2.8}$$

Portanto, para o ganho Kp muito elevado pode fazer com que o sistema fique instável e para ganhos muito baixos este poderá ficar distante do valor desejado, já que este controlador não consegue eliminar o erro em regime permanente (IBRAHIM, 2006).

2.6.2 Controlador Proporcional Integral (PI)

Para o controlador PI a saída atuante na variável manipulada se dá em função do erro e da integral do erro (LOURENÇO, 1996), conforme apresentado na Equação 2.9.

$$u(t) = K_P\left(e(t) + \frac{1}{T_i}\int_0^t e(t)dt\right) \tag{2.9}$$

Aplicando transformada de Laplace para ação de controle acima temos a equação expressa no domínio da variável complexa:

$$U(s) = K_P\left(1 + \frac{1}{T_i s}\right)E(s) \tag{2.10}$$

Em que o termo integral (Ti) é o tempo necessário para que ação de controle integral seja atualizada (Figura 2.12). Note que a ação de controle proporcional possui uma resposta em frequência limitada para todas as frequências.

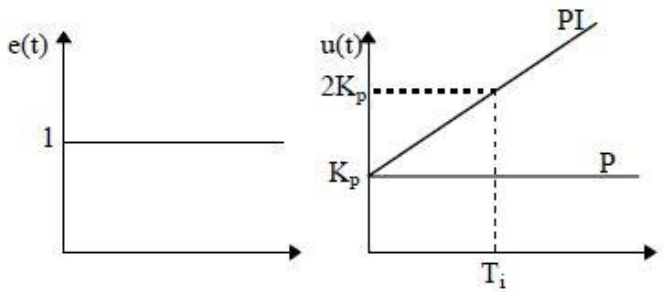


Figura 2.12 - Ação de Controle Proporcional e Integral

Fonte: LOURENÇO, 1996.

Este controlador tem como característica eliminar o erro em regime permanente, pois se adiciona um polo na origem da função de transferência do controlador (LOURENÇO, 1996).

Em Ibrahim (2006) temos que a ação integral pode ser ajustada pelo ganho Ki, para conduzir o erro a zero no tempo necessário. Vale ressaltar que ganhos (Ki) elevados podem resultar em

oscilações e ganhos baixos podem levar a respostas lentas do sistema. O ganho Ki é expresso na Equação 2.11.

$$K_i = \frac{K_P}{T_i} \tag{2.11}$$

2.6.3 Controlador Proporcional Derivativo (PD)

Para o controlador PD a saída atuante na variável manipulada se dá em função do erro e da derivada do erro (LOURENÇO, 1996), conforme expresso na Equação 2.12.

$$u(t) = K_P \left(e(t) + T_d \frac{de(t)}{dt} \right) \tag{2.12}$$

Aplicando transformada de Laplace para ação de controle anterior temos no domínio da variável complexa:

$$U(s) = K_P (1 + T_d s) E(s) \tag{2.13}$$

Em que o termo derivativo (T_d) é o período de tempo antecipado pela ação derivativa relativamente à ação proporcional, sendo expresso em unidades de tempo, conforme apresentado na Figura 2.13.

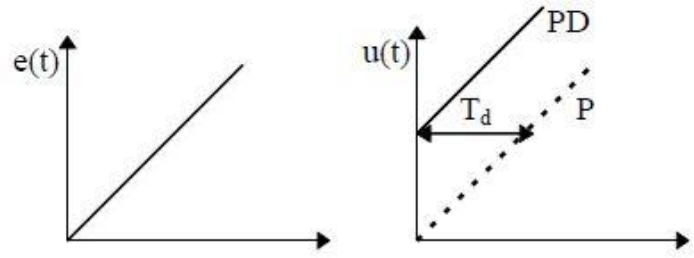


Figura 2.13 - Ação de Controle Derivativo
Fonte: LOURENÇO, 1996

Segundo Lourenço (1996) a utilização da ação de controle derivativa tem por consequência melhoria da estabilidade do sistema, reduzindo o *overshoot* e melhorando a resposta transitória.

Todavia, a vantagem de ser antecipatória ao erro, a ação de controle derivativa tem a desvantagem de amplificar os sinais de ruídos e causar um efeito de saturação no atuador (OGATA, 2002). O ganho derivativo (K_d) é expresso na Equação 2.14.

$$K_d = K_P T_d \tag{2.14}$$

2.6.4 Controlador Proporcional Integral Derivativo (PID)

Conhecido como controlador de três termos ou controlador PID (DORF e BISHOP, 2001) a saída atuante na variável manipulada se dá em função do erro, integral do erro e a derivada do erro (LOURENÇO, 1996), conforme apresentado na Equação 2.15.

$$u(t) = K_P\left(e(t) + \frac{1}{T_i}\int_0^t e(t)dt + T_d\frac{de(t)}{dt}\right) \tag{2.15}$$

Aplicando transformada de Laplace para ação de controle acima temos:

$$U(s) = K_P\left(1 + \frac{1}{T_i s} + T_d s\right)E(s) \tag{2.16}$$

Na Figura 2.14 temos a representação em diagrama de bolcos do controlador PID.

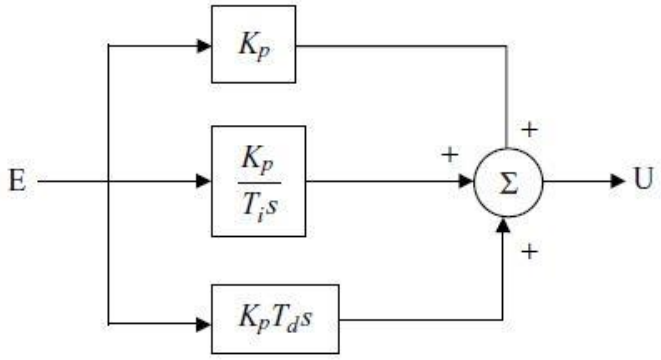


Figura 2.14 - Controlador PID

Fonte: IBRAHIN, 2006

2.6.5 Controle Digital

Visto que o computador utiliza dados amostrados em intervalos de tempos, e estes são expressos de maneira discreta, podemos analisar a estabilidade e a resposta transitória de um sistema no domínio de Z. O uso de computadores digitais como dispositivos controladores cresceu-se fortemente durante as duas últimas décadas (DORF e BISHOP, 2001), em função da disponibilidade de dados digitais que podem ser compartilhados em uma rede digital.

O computador digital recebe e trata os sinais digitais convertidos pelos conversores analógico-digitais e após processar as entradas de acordo com os objetivos específicos converte os sinais de saída para analógicos por meio dos conversores digitais-analógicos atuando nas variáveis manipuladas. Na Figura 2.15 temos a representação do diagrama de blocos de um sistema de controle digital.

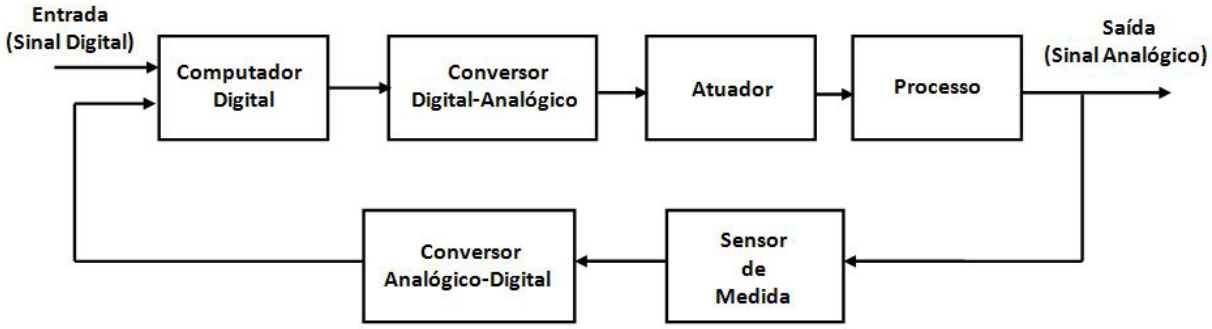


Figura 2.15 - Diagrama de Blocos de um Sistema de Controle Digital

Diversas são as vantagens da utilização do controle digital incluindo o aumento na sensibilidade de medição; sensibilidade reduzida ao ruído dos sinais; capacidade de reconfigurar facilmente via *software* o algoritmo de controle; menor consumo requerido pelos sensores e atuadores digitais, devido à melhora na sensibilidade que resulta em níveis mais baixos de energias, etc. (DORF e BISHOP, 2001).

2.6.6 Implementação do Controlador Discretizado

De acordo com Ibrahim (2006), para implementação do controlador PID contínuo, Equação 2.15, em um computador digital ou em um microcontrolador é necessário representar o tempo contínuo no tempo discreto, já que os elementos controladores são sistemas microprocessados. Existem vários métodos de fazer esta conversão. Para implementação deste trabalho é utilizada aproximação pelo método de Euler para a derivada, Equação 2.17, e a aproximação trapezoidal para a integral, Equação 2.18.

$$\frac{de(t)}{dt} \approx \frac{e(Kh) - e(Kh - h)}{h} \tag{2.17}$$

$$\int_0^t e(t)dt \approx \sum_{K=1}^{n} he(Kh) \tag{2.18}$$

Onde:

h: é o tempo ou intervalo de amostragem (ou período de amostragem);

e(Kh) : é o valor do erro no instante atual;

e(Kh – h): é o valor do erro no instante de amostragem anterior ao atual.

Assim, a Equação 2.15 passa a ser expressa por:

$$u(Kh) = K_P\left[e(Kh) + T_d\frac{e(Kh) - e(Kh-h)}{h} + \frac{h}{T_i}\sum_{K=1}^{n} he(Kh)\right] + u(0) \qquad (2.19)$$

onde $u(0)$: é o último sinal de controle.

O PID dado na Equação 2.19, agora está sob uma forma adequada que pode ser implementado em um computador digital. Esta forma de controlador PID é conhecida como controlador PID posicional.

A forma discreta do controlador PID também pode ser derivada da transformada Z da Equação 2.16 (Equação 2.20).

$$U(z) = K_P\left[1 + \frac{h}{T_i(1-z^{-1})} + T_d\frac{(1-z^{-1})}{h}\right]E(z) \qquad (2.20)$$

Expandindo Equação 2.20 temos:

$$\begin{aligned} u(Kh) = u(Kh-h) + K_P[e(Kh) - e(Kh-h)] + \frac{K_P h}{T_i}e(Kh) \\ + \frac{K_P T_d}{h}[e(Kh) - 2e(Kh-h) - e(Kh-2h)] \end{aligned} \qquad (2.21)$$

O controlador expresso pela Equação 2.21 é conhecido como o controlador PID de velocidade. Neste caso, a ação de controle atual usa o valor do controle anterior como referência. Se existe um grande erro, a resposta do controlador PID de velocidade pode ser lenta, especialmente se o tempo de integração (T_i) é elevado.

2.6.7 Métodos de Sintonia do Controlador pela Curva de Reação

Quando um controlador PID é usado num sistema, é importante ajustar o controlador a fim de obter a resposta desejada. Sintonizar o controlador PID envolve a seleção de valores para os parâmetros K_P, T_i e T_d do controlador. Existem diversos métodos para estes ajustes que vão desde as primeiras técnicas desenvolvidas por Ziegler-Nichols, Cohen-Coon e até ajustes automáticos mais recentes para os controladores (IBRAHIM, 2006).

Para sintonia do controlador PID no presente trabalho utiliza-se o Método da Curva de Reação, o qual a partir de um degrau dado ao sistema em malha aberta é possível obter os parâmetros (atraso (L); Constante de tempo (T) e Ganho direto do sistema (K_0)) para sintonia de Ziegler-Nichols e Cohen-Coon.

Muitas plantas, particularmente as decorrentes nas indústrias de processo podem ser satisfatoriamente descritas como um sistema de primeira ordem (GOODWIN; GRAEBE e SALGADO, 2000) conforme apresentado na Equação 2.22.

$$\mathrm{Gc(s)} = \frac{\mathrm{K_0 e^{-sL}}}{Ts + 1} \qquad sendo \quad T > 0 \tag{2.22}$$

Onde:

L: é o atraso do sistema;

T: é a constante de tempo;

$\mathrm{K_0}$: é o ganho direto do sistema.

Para Goodwin, Graebe e Salgado (2000), a versão quantitativa linearizado deste modelo pode ser obtida a partir da experiência em malha aberta utilizando o seguinte procedimento:

1. Coloque a planta no ponto de operação normal (em regime permanente) em malha aberta, no qual a saída inicial é dada por y_0 e para uma entrada inicial constante dada por u_0.
2. Após um intervalo de tempo, aplica-se uma mudança na entrada da planta (degrau), este deve estar em uma faixa de 10 a 20% de seu fundo de escala.
3. Registra-se a saída da planta até que se instale o novo ponto de operação (y_∞).

Com este procedimento, os sistemas descritos pela Equação 2.22 irão apresentar uma curva de resposta similar à apresentada na Figura 2.16, que é conhecida como a Curva de Reação do Processo, na qual *m.s.t* significa a reta tangente no ponto de derivada máxima da curva.

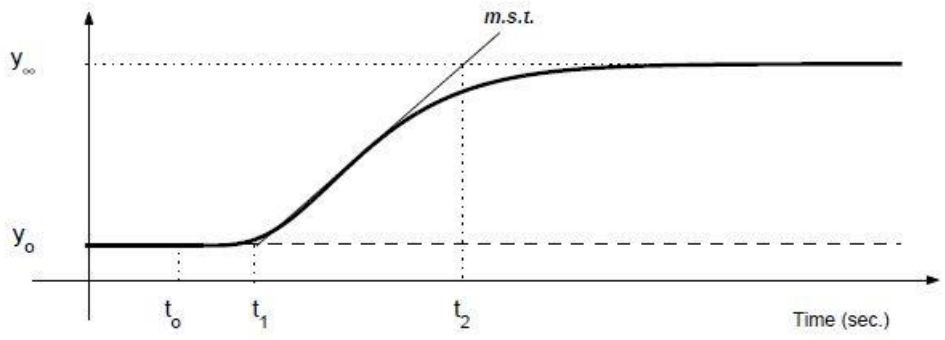


Figura 2.16 - Curva de Reação de um Sistema de 1º Ordem

Fonte: GOODWIN, GRAEBE e SALGADO,2000.

O ganho direto do sistema (K_0) pode ser obtido por:

$$K_0 = \frac{y_\infty - y_0}{u_f - u_0}; \quad \text{onde: } L = t_1 - t_0;\ T = t_2 - t_1 \tag{2.23}$$

A sintonia empírica proposta por Ziegler-Nichols, tem como objetivo conseguir um amortecimento na resposta do sistema, na razão de 4:1 do primeiro para o segundo pico da resposta (GOODWIN; GRAEBE e SALGADO, 2000). A partir dos parâmetros encontrados na Equação 2.22, Ziegler-Nichols propôs encontrar os parâmetros do controlador de acordo com as relações proposta na Tabela 2.3.

Tabela 2.3 - Sintonia de Ziegler Nichols

Tipo de Controlador	Kp	Ti	Td
P	$\frac{T}{K_0 L}$		
PI	$\frac{0.9T}{K_0 L}$	$3L$	
PID	$\frac{1.2T}{K_0 L}$	$2L$	$0.5L$

Fonte: GOODWIN, GRAEBE e SALGADO,2000

Entretanto, esta sintonia revela uma elevada sensibilidade do desempenho para diferentes valores da razão entre atraso (L) e a constante de tempo (T) (Equação 2.24). Na Figura 2.17 temos representado a saída de um controlador projetado por Ziegler-Nichols em função de diferentes razões entre o atraso e a constante de tempo do sistema. Note que para atrasos pequenos em relação a elevadas constantes de tempo, o sistema passa apresentar uma maior oscilação e sua saída, e a medida que a constante de tempo fica reduzida em relação ao atraso, o sistema fica sub amortecido.

$$x = \frac{L}{T} \tag{2.24}$$

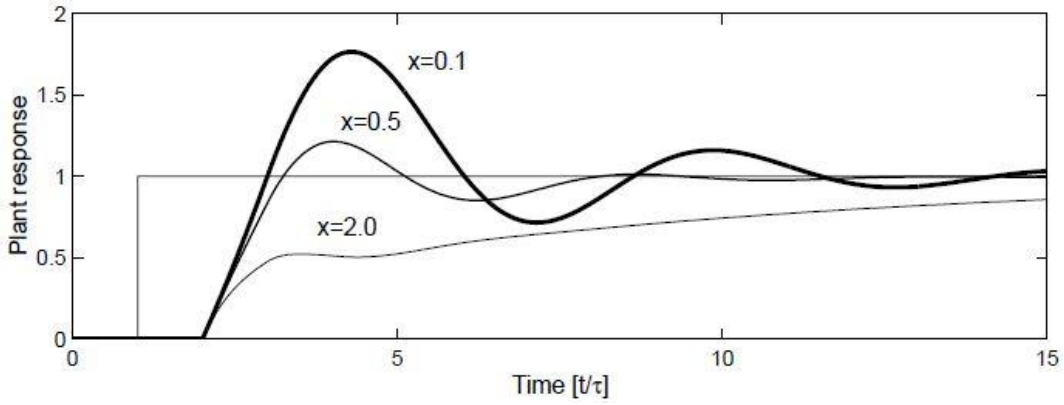


Figura 2.17 - Sensibilidade do Sistema para Sintonia Ziegler-Nichols

Fonte: GOODWIN, GRAEBE e SALGADO,2000

De acordo com Goodwin, Graebe e Salgado (2000), para melhorar esta sintonia Cohen-Coon avançou nos estudos para encontrar as configurações que levam uma menor dependência do controlador em relação a razão do atraso e a constante de tempo. A proposta de Cohen-Coon de sintonia está apresentada na tabela 2.4.

Tabela 2.4 - Sintonia de Cohen Coon

Tipo de Controlador	Kp	Ti	Td
P	$\frac{T}{K_0 L}(1+\frac{L}{3T})$		
PI	$\frac{T}{K_0 L}(0.9+\frac{L}{12T})$	$\frac{L(30T+3L)}{9T+20L}$	
PID	$\frac{T}{K_0 L}(\frac{4}{3}+\frac{L}{4T})$	$\frac{L(32T+6L)}{13T+8L}$	$\frac{4LT}{11T+2L}$

Fonte: GOODWIN, GRAEBE e SALGADO,2000

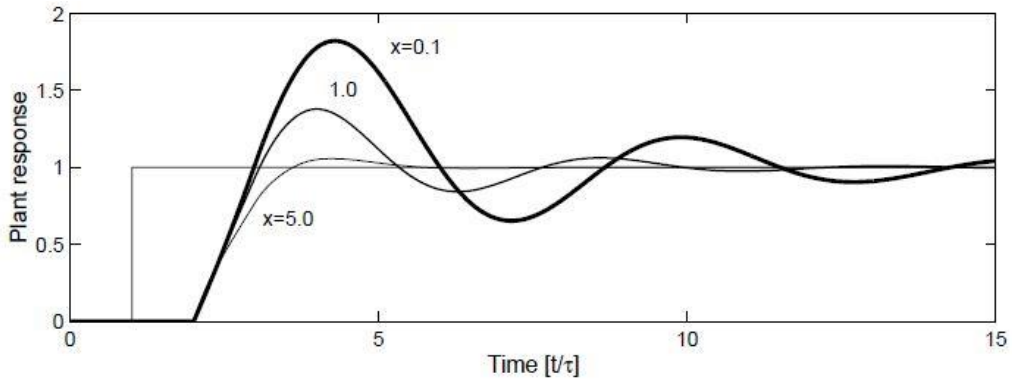


Figura 2.18 - Sensibilidade do Sistema para Sintonia Cohen-Coon

Fonte: GOODWIN, GRAEBE e SALGADO,2000

Observe que conforme apresentado na Figura 2.18 o sistema opera um comportamento mais homogênio para diferentes valores da razão entre atraso (L) e constante de tempo (T).

3 DESENVOLVIMENTO

3.1 Projeto Estrutural da Bancada

Para condicionamento dos motores e do circuito de acionamento, foi utilizado o *software* SOLIDWORKS, para desenvolvimento das peças que compõem o conjunto da bancada, conforme apresentado no ANEXO. Como material foram utilizadas chapas de acrílico de 4, 6 e 10 mm de espessura, que foram cortadas a laser.

3.2 Projeto do *Hardware*

Para desenvolvimento do circuito do *hardware* foi utilizado o *software* PROTEUS/ISIS (versão 8.0 Professional) como ambiente de desenvolvimento e simulação do circuito eletrônico, em razão de possuir uma extensa biblioteca de componentes e *packages*, permitindo ao próprio desenvolvedor criar e configurar novos itens da biblioteca satisfazendo a necessidade do projeto. O *software* PROTEUS/ARES possibilita visualizar o projeto em 3D, que auxilia o usuário analisar a disposição dos componentes no *hardware*.

Dado a importância de reduzir o espaço ocupado pelo circuito na bancada, o *hardware* foi projetado em forma de um *shield*, o qual se condiciona sobre o Arduino. Esta conexão é realizada por barras de pinos de 40mm, os quais satisfazem a conexão entre o circuito e o Arduino e possibilita a expansão de *shields* por sobreposição.

3.2.1 Alimentação da Bancada

Dado que alimentação do Arduino, proveniente da porta USB do computador, não é suficiente para fornecer a tensão e corrente para o motor de polos sombreados e para motor CC, foi utilizada uma fonte externa de 36Vcc/1A, conforme apresentada na Figura 3.1.

Figura 3.1 - Fonte externa bancada, modelo S-36-36

Para conexão da fonte externa ao *hardware*, é utilizado o plug Jack P4 (Macho - saída da fonte e Fêmea – presente no *hardware*). Com o objetivo de ter um ponto comum entre o *hardware* desenvolvido o Arduino e a fonte (mesmo referencial – 0V) é utilizado uma barra de pinos, o qual está ligada ao plug P4 no ponto GND, conforme apresenta a Figura 3.2.

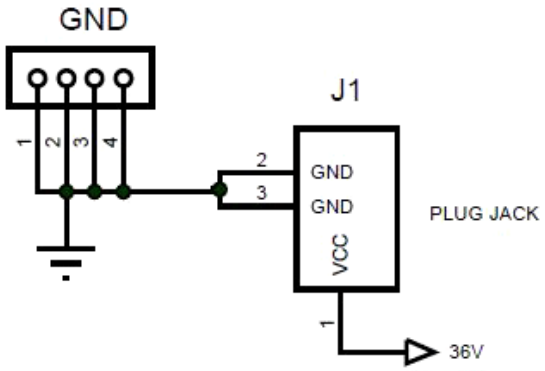


Figura 3.2 - Conexão da Fonte ao *Hardware*

A fonte externa utilizada é de uso exclusivo dos motores. Para as demais necessidades do uso de energia (*led's*, *encoder*, potenciometros e botão), é utilizado a fonte de 5V proveniente do computador que alimenta o Arduino UNO. A Figura 3.3 apresenta a barra de pinos do arduino, que alimenta o *hardware* em 5V e liga o ponto comum entre computador, fonte externa e Arduino (pinos 2 e 3).

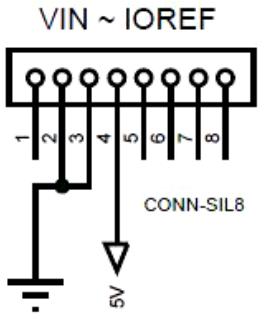


Figura 3.3 - Alimentação Arduino ao *Hardware*

3.2.2 Acionamento dos Motores

A alimentação dos motores é realizada de forma paralela, o qual o motor de polos sombreados está ligado diretamente à saída do plug P4 (36V) e o motor CC está ligado na saída do regulador de tensão LM7812. A necessidade deste regulador se deve ao fato do motor CC trabalhar com tensão nominal de 12V, reduzindo portanto a tensão de 36V para 12V neste regulador. Para identificação de funcionamento do regulador, é utilizado um *led* como sinalização na saída do mesmo (resistor de 680Ω usado para limitar a corrente no *led*). Na

Figura 3.4 temos apresentado o diagrama esquemático de ligação do regulador de tensão, o qual foram utilizados como padrão os capacitores 330nF e 100nF.

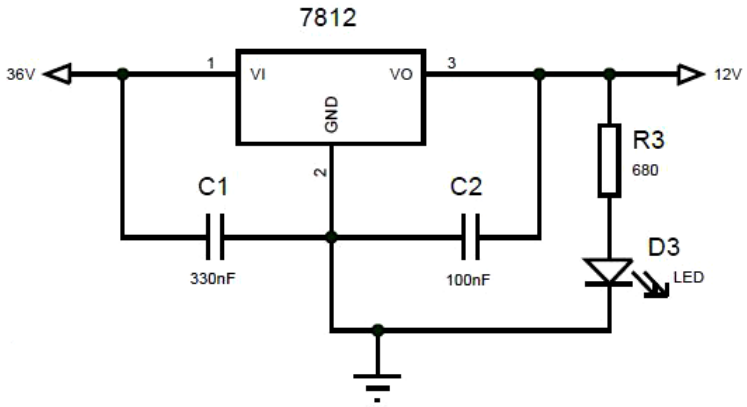


Figura 3.4 - Ligação Regulador de Tensão LM7812

Em função do aquecimento do regulador, foi utilizado um dissipador de alumínio com objetivo de dissipar este calor e com isso aumentar a vida útil do regulador.

Para o acionamento dos motores com o Arduino, foram utilizados os transistores TIP122, já que as saídas do Arduino não são capazes de fornecer corrente suficiente para os motores. Esta identificação de corrente foi obtida por meio de experimento, os quais foram realizados testes com o motor CC livre (sem carga) e com a máxima carga (motor de polos sombreados com a tensão de 33.9Vcc em seus terminais) e obtivemos uma corrente no intervalo de 42mA a 65mA para operação sem carga, e a corrente 185mA para máxima carga. A partir disso o TIP122 foi escolhido, pois possibilita o acionamento de cargas em 12 Vcc com corrente de até 5A, operando com folga sem o risco de queima do componente.

Visto que o método de acionamento utilizado é por largura de pulsos (PWM), a região de operação do transistor utilizada foi à saturação e corte. A corrente de base para que ocorra a saturação nos respectivos transistores é apresentada na Equação 2.25 para o motor CC e Equação 2.26 para o motor de polos sombreados, que definem quais são os respectivos resistores de base para as correntes. Equação 2.7 foi utilizada como base para a determinação destes resistores.

$$R_{\_Base1} = \frac{5V - 0.7V}{650\mu A} = 6.615K\Omega \tag{2.25}$$

$$R_{\_Base2} = \frac{5V - 0.7V}{0.2\mu A} = 21.5K\Omega \tag{2.26}$$

Na Equação 2.25 e 2.26 foram utilizados ganhos de 100 e 900 respectivamente, de acordo com análise do *datasheet* do transistor.

Para garantir a saturação no acionamento, foram utilizados $R_{\_Base1} = 5{,}6K\Omega$ e $R_{\_Base2} = 10K\Omega$, que correspondem aos resistores comerciais disponíveis no laboratório e mais próximos dos calculados. Caso fossem utilizados resistores de maior valor não ocorreria a saturação dos transistores.

Para segurança do TIP122 no momento de interrupção de potência entregue a carga (motor) e manter a corrente contínua na armadura, foram utilizados diodos do modelo (1N4007) ligados em paralelo com os motores, conhecido como diodo de roda livre. Este circula a corrente produzida pela indutância do motor até que esta caia a zero, assim evitando danos ao semicondutor.

Na Figura 3.5 temos o circuito eletrônico (módulo de potência) de acionamento de cada motor.

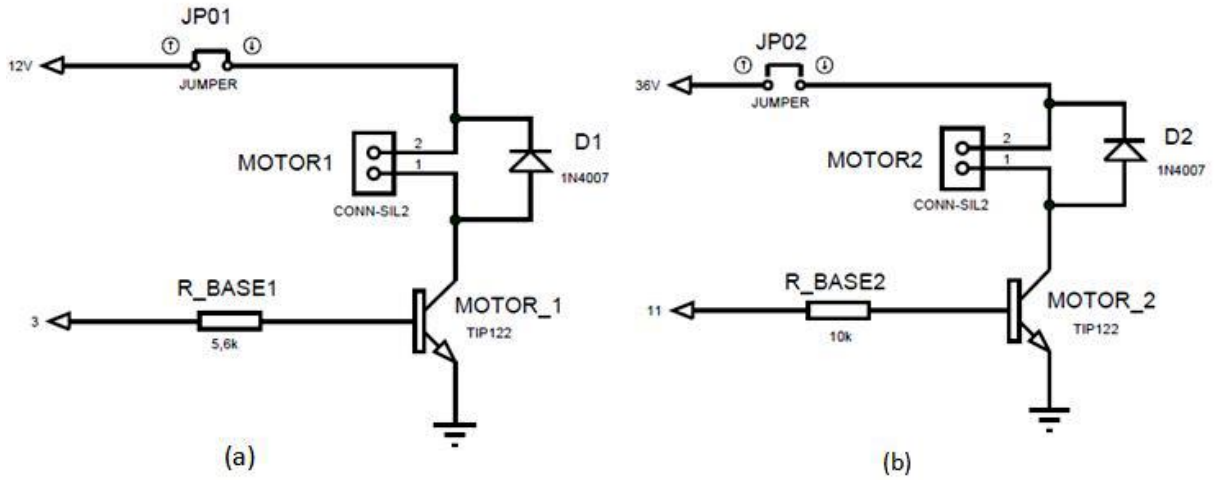


Figura 3.5 - Acionamento Motores, (a) Motor CC, (b) Motor de Polos Sombreados

No desenvolvimento do acionamento dos motores foram deixados dois pontos (JP01 e JP02) conforme apresentado na Figura 3.5. Estes pontos são para futuros trabalhos que possibilitem a estimação da velocidade para o motor CC e o distúrbio no sistema (freio-motor de polos sombreados) por meio da leitura da corrente por técnicas *sensorless*.

O motor de polos sombreados é utilizado como distúrbio/freio no movimento de rotação do eixo do motor CC. A operação normal do motor de polos sombreados é em tensão alternada em 127Vac ou 240Vac. Para o trabalho como freio, está sendo manipulado com tensão contínua de 36Vcc, o qual não ocorre variação do fluxo nas bobinas e com isso criação dos polos norte e sul fixo, atuando como uma resistência ao movimento do eixo.

O acoplamento mecânico utilizado entre os motores para transferência de torque e rotação é um acoplamento elástico em alumínio com diâmetro externo de 18mm, comprimento de 25mm e furo de 4mm.

3.2.3 Modo de Operação

A bancada apresenta dois modos de operação: manual e automático

- Manual – manipulada somente pelos potenciômetros (Figura 3.6).

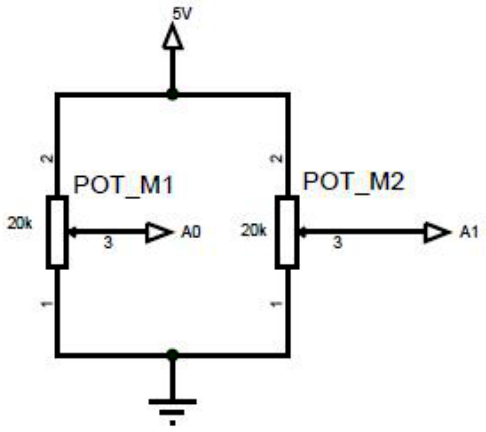


Figura 3.6 - Operação Modo Manual

Nesta operação o Arduino recebe os valores de tensão advindo dos potenciômetros, por meio de entradas analógicas com resolução de 10bits, que são convertidas em sinais PWM's com resolução de 8bits nas saídas do Arduino.

- Automático – operação somente pelo supervisório.

O status o qual o usuário deseja operar a bancada pode ser definido de forma física, botão presente no *hardware* ou por meio do *software*, supervisório. Na Figura 3.7 temos a representação do circuito do botão presente no *hardware*.

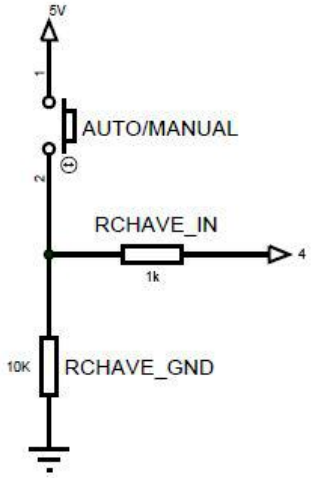


Figura 3.7 - Circuito Botão

Para facilitar a identificação do modo de operação da bancada (manual / automático) pelo usuário, foi disposto no circuito um par de *led's* para sinalizar o modo de operação.

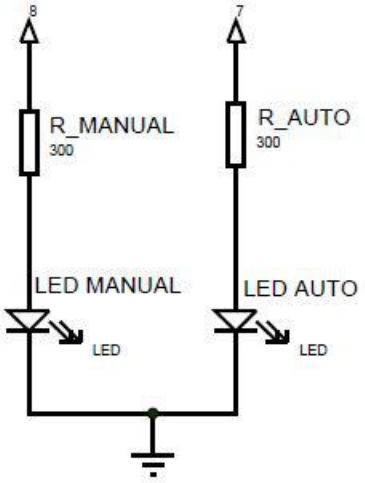


Figura 3.8 - Circuito Led's

3.3 Placa de Circuito Impresso

Finalizado e simulado o circuito eletrônico no PROTEUS/ISIS, a próxima etapa foi o desenvolvimento da placa de circuito impresso utilizando o mesmo *software* no ambiente ARES.

Para atender a ideia do circuito como *shield*, as distâncias entre as barras de pinos foram mensuradas para possibilitar a conexão do circuito sobre o Arduino. Com a dimensão da placa definida, deu-se prosseguimento na distribuição espacial dos componentes, realização das trilhas e malha de terra, e em paralelo com este desenvolvimento a utilização do ambiente 3D

Visualizer. A Figura 3.9 apresenta o resultado do projeto do circuito impresso (a) e a visão 3D da placa (b).

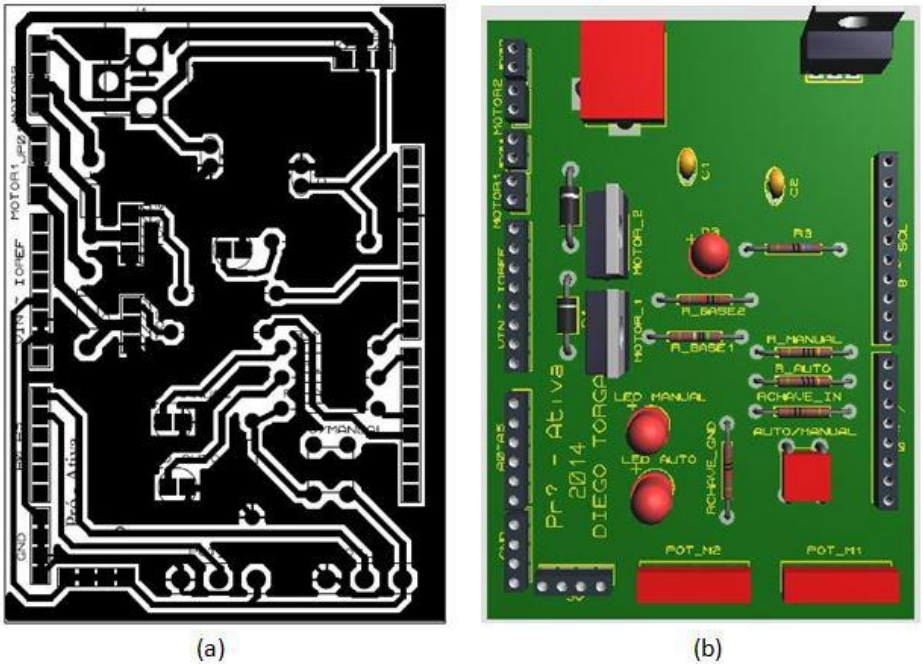


Figura 3.9 - (a) Circuito ARES, (b) Visualização 3D

3.4 Descrições das Conexões Elétricas

Na Figura 3.10 temos a vista superior do circuito desenvolvido (*shield*) com a identificação de sua pinagem. Nas Tabelas 3.1 e 3.2 temos a descrição das conexões entre o *shield* e o Arduino, bem como das conexões entre o *shield* e os atuadores e sensores presentes na bancada.

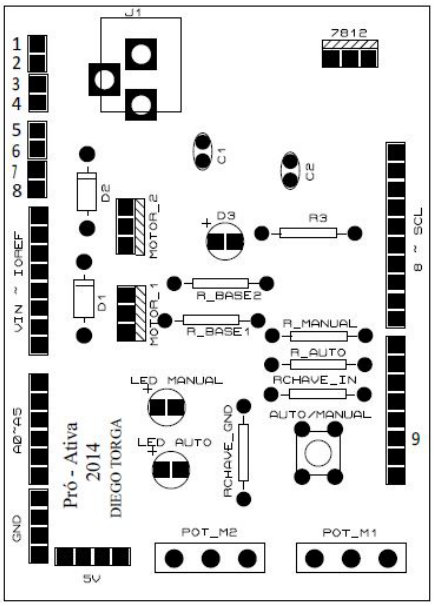


Figura 3.10 - Vista Superior do *Shield*

Tabela 3.1 - Diagrama de Conexão do Arduino e *Shield*

Arduino e *Shield*	
Pinos do Arduino	Função
A0	Leitura - Potenciômetro Motor CC (POT_M1)
A1	Leitura - Potenciômetro Motor de Polos Sombreados (POT_M2)
3	Sinal do PWM para o Motor CC (R_BASE1)
4	Leitura – Botão (RCHAVE_IN)
7	Acionamento do *Led* Manual – Verde (R_MANUAL)
8	Acionamento do *Led* Automático – Vermelho (R_AUTO)
11	Sinal do PWM para o Motor de Polos Sombreados (R_BASE2)
5V	Alimentação dos circuitos de 5V e da Barra de Pinos de 5V para demais necessidades.
GND	Aterramento da placa e da barra de pinos GND para demais necessidades

Tabela 3.2 - Diagrama *Hardware* / Bancada

Shield e Bancada	
Pinos do *shield* (Figura 3.10)	Função
1 e 2	Ponto disponível para a leitura de corrente do Motor de Polos Sombreado
3 e 4	Ponto de Ligação do Motor de Polos Sombreados (fios preto/vermelho)
5 e 6	Ponto disponível para a leitura de corrente do Motor CC
7 e 8	Ponto de Ligação do Motor CC (fios amarelo/laranja)
9	Ligação do sinal do *encoder* para leitura de posição angular (fio verde ou branco)

Barra de Pinos 5V	Ponto de Alimentação do *encoder* (fio vermelho)
Barra de Pinos GND	Ponto de terra do *encoder* (fio preto)

3.5 Telas - Supervisório

Dado o objetivo de realizações de práticas de controle de velocidade da respectiva bancada, foi desenvolvido um passo a passo para adquirir os parâmetros de controle (K_p, T_i e T_d,) por meio da curva de reação com as sintonias de Ziegler-Nichols e Cohen-Coon. Uma vez obtidos os parâmetros do controlador, os mesmo são discretizados com base na Equação 2.21.

3.5.1 Tela - Inicial

A Tela Inicial (Figura 3.11) apresenta as etapas a serem seguidas, o qual para cada sub tela é descrito o que será apresentado e a sequência a ser realizada. Para entrar nas sub telas basta clicar em seus respectivos botões. Na barra de menu localizada na parte superior da Tela Inicial contemplam algumas funções adicionais:

- Resultados
    - Resposta ao Degrau – Apresenta a resposta ao degrau adquirida na respectiva sub tela;
    - Gráfico_01 – Apresenta o gráfico 1 salvo na Tela Supervisório Bancada, apresentando a velocidade1, erro1 e tensão1;
    - Gráfico_02 – Apresenta o gráfico 2 salvo na Tela Supervisório Bancada, apresentando a velocidade2, erro2 e tensão2;
    - Gráfico_03 – Apresenta o gráfico 3 salvo na Tela Supervisório Bancada, apresentando a velocidade3, erro3 e tensão3.

  Obs – Estas informações satisfazem a necessidade de avaliação dos resultados sem que seja necessário realizar novamente todo processo. Os gráficos 1, 2 e 3, são salvos a critério do usuário da bancada, e de forma cíclica (gráfico 4 assume o gráfico 1, gráfico 5 assume o gráfico 2, e assim por diante, salvando somente os últimos 3 gráficos de interesse do executante). O gráfico da Resposta ao Degrau é armazenado de forma automática na execução desse processo.
- A Bancada
    - Sobre – Apresenta um breve contexto de como surgiu o projeto e seu desenvolvimento, contextualizando a conquista de menção honrosa durante

encontro de saberes da UFOP em 2014, e bem como informações sobre o autor.

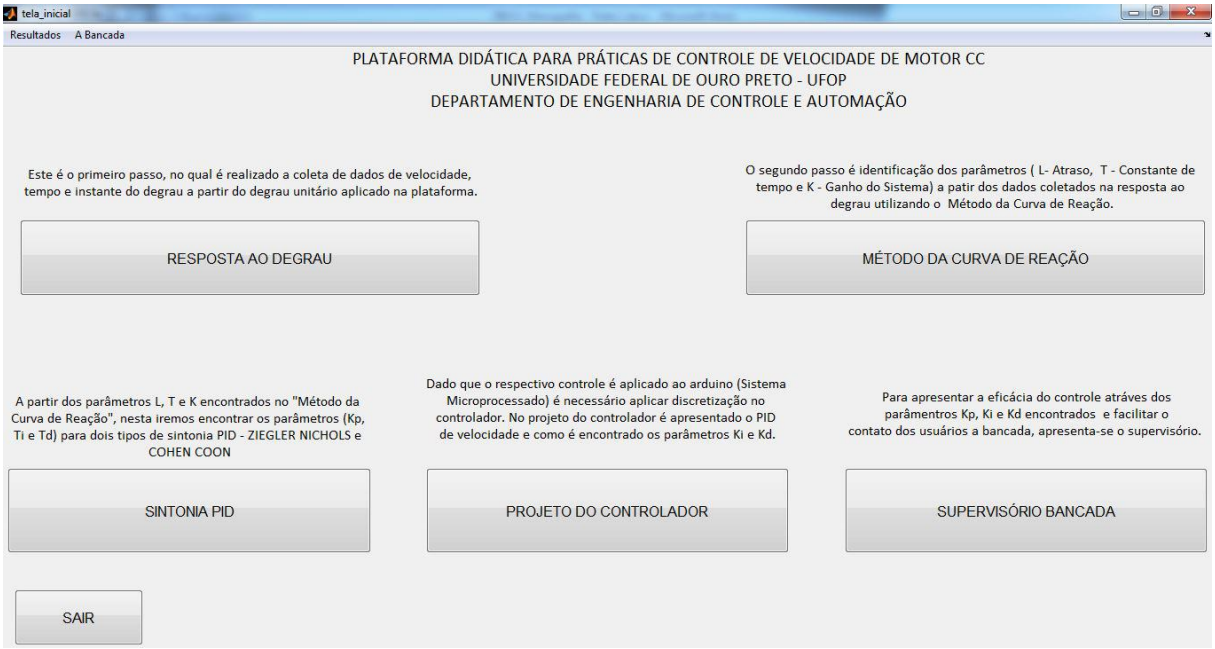


Figura 3.11 - Tela Inicial

3.5.2 Tela - Resposta ao Degrau

A primeira etapa a ser realizada é o ensaio para a amostragem do sinal de saída (velocidade) em função de um sinal na entrada em forma de degrau. Para conexão do supervisório a bancada é utilizada a comunicação serial entre Arduino e computador, na qual o Matlab (ambiente de desenvolvimento do supervisório) somente envia e recebe dados, e o controle é realizado pelo Arduino.

A tela do ensaio de resposta do sistema ao degrau (Figura 3.13) é descrita a seguir:

- Escolha da porta COM – Porta virtual criada pelo computador para comunicação com Arduino;
- Botão Conectar – Uma vez definida a porta de comunicação (COM), este inicia a conexão entre a bancada e o supervisório. Caso ocorra a escolha da porta COM errada é apresentado uma mensagem de erro para o usuário. Nesta etapa de conexão é apresentado três modos de status:
    - OFF_LINE – Sem conexão com a bancada. Sinalizada em vermelho;
    - INICIALIZANDO – Efetuando conexão entre supervisório e Arduino. Sinalizado em azul;

- ON_LINE – Comunicação estabelecida com sucesso. Sinalizado em Verde. Nesta situação ocorre a mudança do status do botão (Conectar para Desconectar e vice-versa caso ocorra à interrupção da conexão) e liberação do botão “INICIAR COLETA DE DADOS”, que é responsável pelo início de execução do ensaio de resposta ao degrau.

- Parâmetros do Degrau
  - Tensão inicial – Define qual a tensão inicial será aplicada ao motor CC para que se estabeleça o primeiro ponto de operação em regime permanente antes de aplicar o degrau na entrada;

    Dado que o motor é acionado pelo método de PWM, é necessário converter o valor de tensão dado pelo usuário para o PWM correspondente. O PWM do Arduino é de 8bits, o que oferece uma resolução de 256 valores (0 – 255). Para isso, com o auxilio do osciloscópio (marca Tektronix, modelo TDS 1001B) foi identificado o comportamento do motor (tensão na armadura) em função da variação do sinal de PWM pelo Arduino (Figura 3.12).

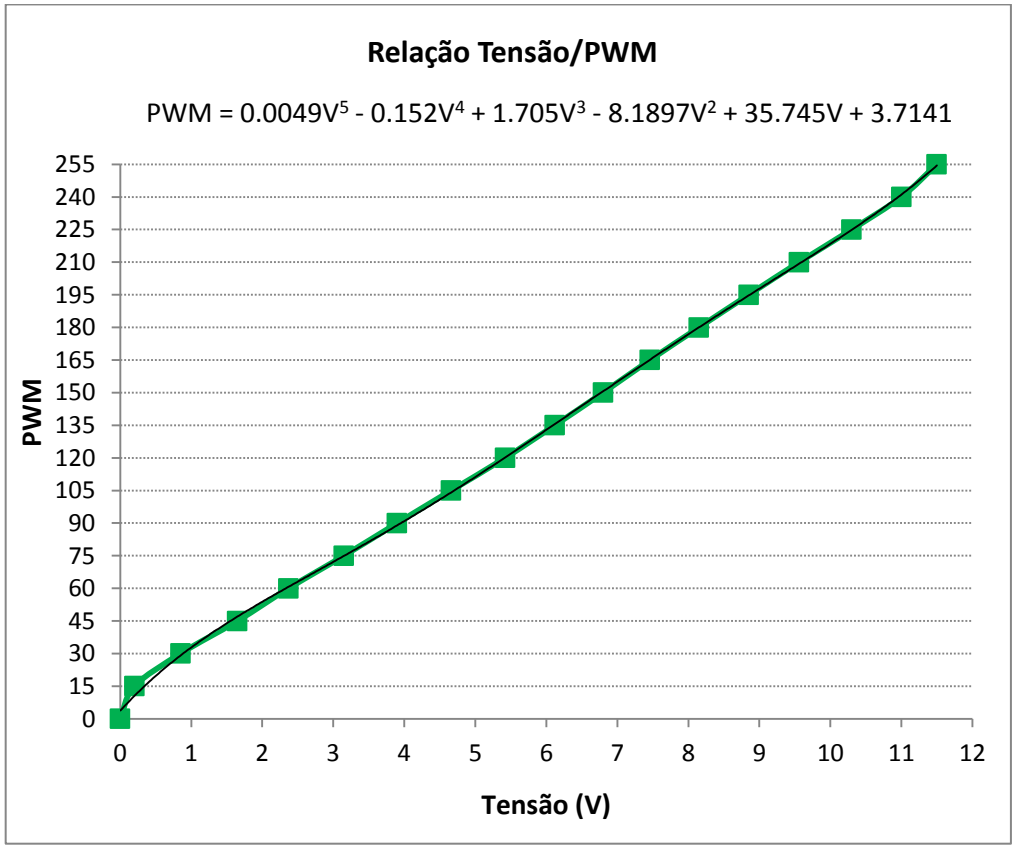


Figura 3.12 - Relação Tensão / PWM

A partir dos dados coletados (Figura 3.12) e com auxilio do MS Excel (versão 2007) foi determinado uma Equação (2.22) de 5º grau que representa a relação entre a variação do sinal de PWM e a tensão na armadura do motor CC.

$$PWM = 0.0049V^5 - 0.152V^4 + 1.705V^3 - 8.1897V^2 + 35.745V + 3.7141 \quad (2.22)$$

- Valor do degrau – Define qual a porcentagem do degrau a ser aplicado no fundo de escala (12V) na entrada da bancada após 5s de operação. O tempo de todo ensaio da resposta ao degrau é de 10s.

  Obs: O valor de tensão inicial e final está limitado entre 0 a 12V respectivamente, pois 12V é a tensão nominal de operação do motor. Para porcentagem do degrau é limitado entre 0 a 20%, pois de acordo com Goodwin, Graebe e Salgado (2000) 20% é o valor máximo recomendado para obtenção dos parâmetros de controle pelo método da curva de reação. Para valores fora destes limites, ao clicar no botão "INICIAR COLETA DE DADOS" é apresentada uma mensagem, a qual exibe os valores de limites corretos para cada parâmetro do degrau. Enquanto esses parâmetros não atenderem os limites estabelecidos o ensaio não será habilitado ao usuário.

- Tempo de Amostragem - É o tempo o qual o Arduino amostra as variáveis e envia as informações para o Matlab (velocidade, tempo e instante do degrau). O usuário pode escolher entre 8ms e 10ms. O período de 8ms foi o tempo mínimo alcançado pelo processamento do computador utilizado (Processador Intel i5, 2,53GHz, 4GB de RAM, sistema operacional de 64 bits e Windows 7 Home Premiun).

  A seguir é a apresentada as configurações dos registradores do *Timer1* conforme definido na Equação 2.3 para os respectivos tempos de amostragem.

  // INICILALIZANDO OS REGISTRADORES COM ZERO

  TCCR1A = 0;

  TCCR1B = 0;

  TCNT1 = 0;//Local onde é armazenado o valor real do temporizador.

  OCR1A = **Resultado da Equação 2.3;**

  // ATIVANDO MODO DE OPERAÇÃO CTC

  TCCR1B = (1 << WGM12);

  // ESCOLHENDO PRESCALER DE 1024

  TCCR1B = (1 << CS12);

  TCCR1B = (1 << CS10);

  // ATIVANDO O MODO CTC PARA TIMER1

  TIMSK1 = (1 << OCIE1A).

  Obs: Estas configurações são realizadas na inicialização do Arduino, dentro do *Void Setup ()*.

- Botão Iniciar Coleta de Dados – Envia para o Arduino as configurações realizadas pelo usuário e inicia o ensaio de resposta ao degrau, cujos dados amostrados são exibidos durante o experimento. Após finalização do processo é apresentado uma caixa de mensagem informando que a coleta de dados foi realizada com sucesso. A seguir ocorre a interrupção da conexão de forma automática.
- Botão Voltar – Volta a Tela Inicial.

Figura 3.13 - Tela Resposta ao Degrau

É importante que o usuário só inicie a comunicação com o Arduino após preenchimento dos parâmetros do degrau e escolha do tempo de amostragem, e inicialize a coleta de dados após estabelecida comunicação. Este procedimento é necessário para que o tempo computado pelo Arduino durante o período de comunicação não seja elevado e gere erros de arredondamento na escala do tempo referente ao ponto de inflexão da curva de reação ao degrau.

3.5.3 Tela - Método da Curva de Reação

Realizado o ensaio da resposta ao degrau, a próxima etapa é encontrar os parâmetros K_0, T e L, a partir dos dados amostrados. Antes de analisar a curva de reação do degrau, o usuário necessita suavizar os dados amostrados, por meio de uma média móvel entre 1, 3, 5 ou 7 amostras. O operador seleciona a opção desejada e clica em "OBTER DADOS". Esta suavização é efetuada pela função *smooth* do Matlab.

Na execução do botão, após definido a suavização, são apresentados três gráficos que são utilizados para determinar o ponto de inflexão da curva de reação onde será traçada a tangente (*m.s.t*) figura 2.16:

- Gráficos da resposta ao degrau – Apresentam dois gráficos (1º - Resposta ao Degrau com os dados brutos e 2º - Resposta ao Degrau considerando a suavização escolhida);
  Obs: Os parâmetros K_0, T e L tendem a sofrer pequenas alterações de acordo com a suavização escolhida.

- Gráfico do Cálculo do Ponto de Inflexão – Apresenta qual é o ponto de inflexão máximo da reposta ao degrau. Determinado por meio da derivada do vetor de velocidade. É com base neste ponto que é traçado a reta tangente (*m.s.t)* conforme figura 2.16.

A partir destes gráficos o usuário consegue obter as seguintes informações:

- Tempo o qual ocorreu o ponto de inflexão máximo;
- O valor da derivada no ponto de inflexão máximo;
- Velocidade do motor CC no instante de tempo que ocorreu o ponto de inflexão máximo.

Preenchido as respectivas informações, e utilizando recursos matemáticos do Matlab, é determinada a reta tangente ao clicar no botão "GRÁFICO CURVA DE REAÇÃO", conforme ilustra a Figura 3.14. A partir disso, é computado de forma automática os parâmetros do atraso (L), constante de tempo (T) e ganho direto do sistema (K_0).

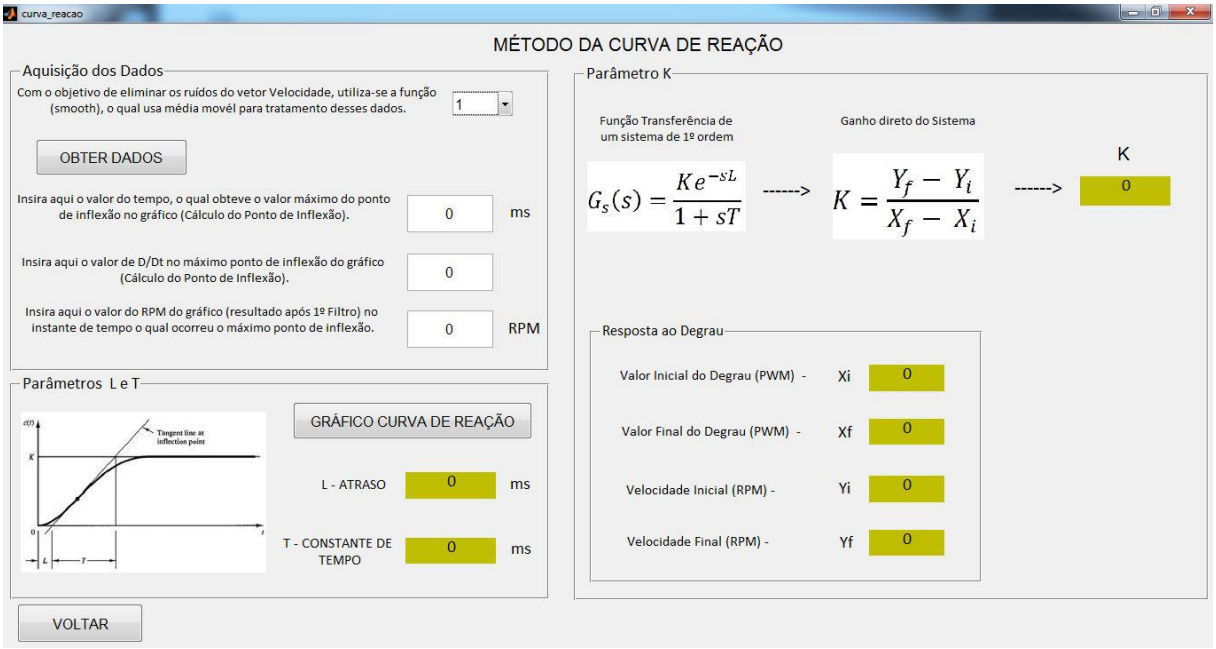


Figura 3.14 - Tela do Método da Curva de Reação

3.5.4 Tela - Sintonia PID

A tela de sintonia PID (Figura 3.15) computa de forma automática os parâmetros do controlador (K_p, T_i e T_d) por meio da sintonia proposta por Ziegler-Nichols e Cohen-Coon, a critério do usuário, a partir das constantes L, T, e K_0 que foram determinadas na etapa anterior.

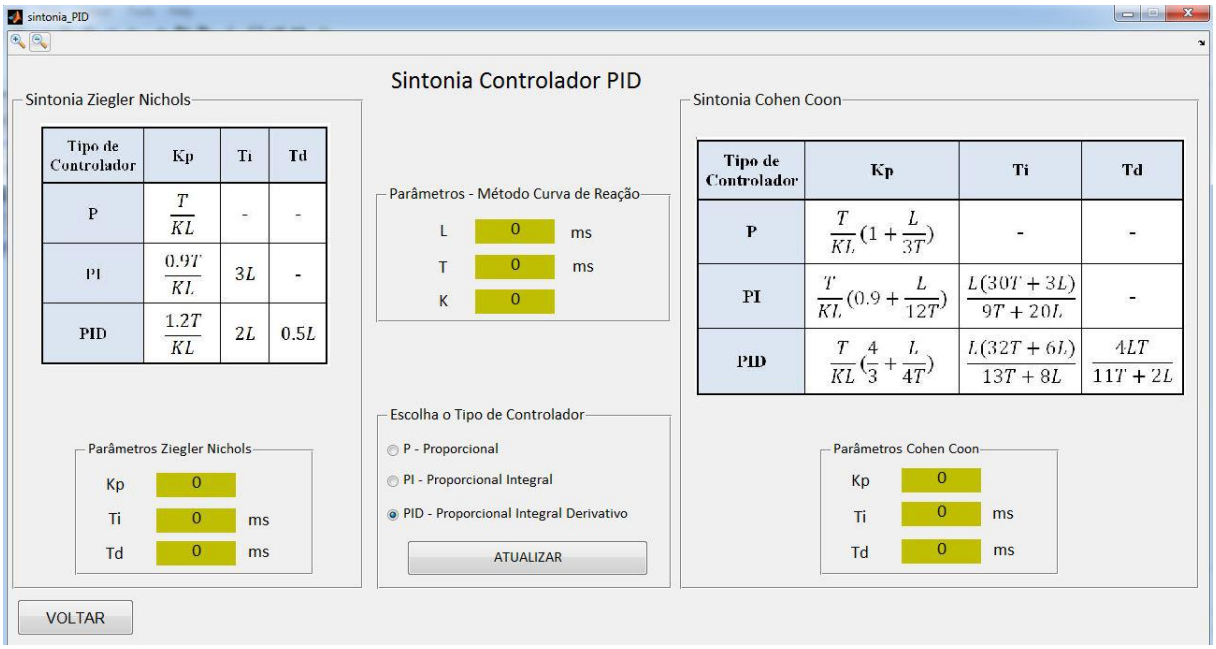


Figura 3.15 - Sintonia Controlador PID

3.5.5 Tela - Projeto do Controlador

Dado que o controle a ser aplicado está sendo executado por um sistema microprocessado e que os parâmetros (K_p, T_i e T_d) determinados no passo anterior estão no domínio do tempo, é disponibilizado ao usuário a tela Projeto Controlador Digital (Figura 3.16) que permite a conversão destes parâmetros nos ganhos do controlador digital (K_p, K_i e K_d).

O tempo de amostragem do controlador é de 5ms.

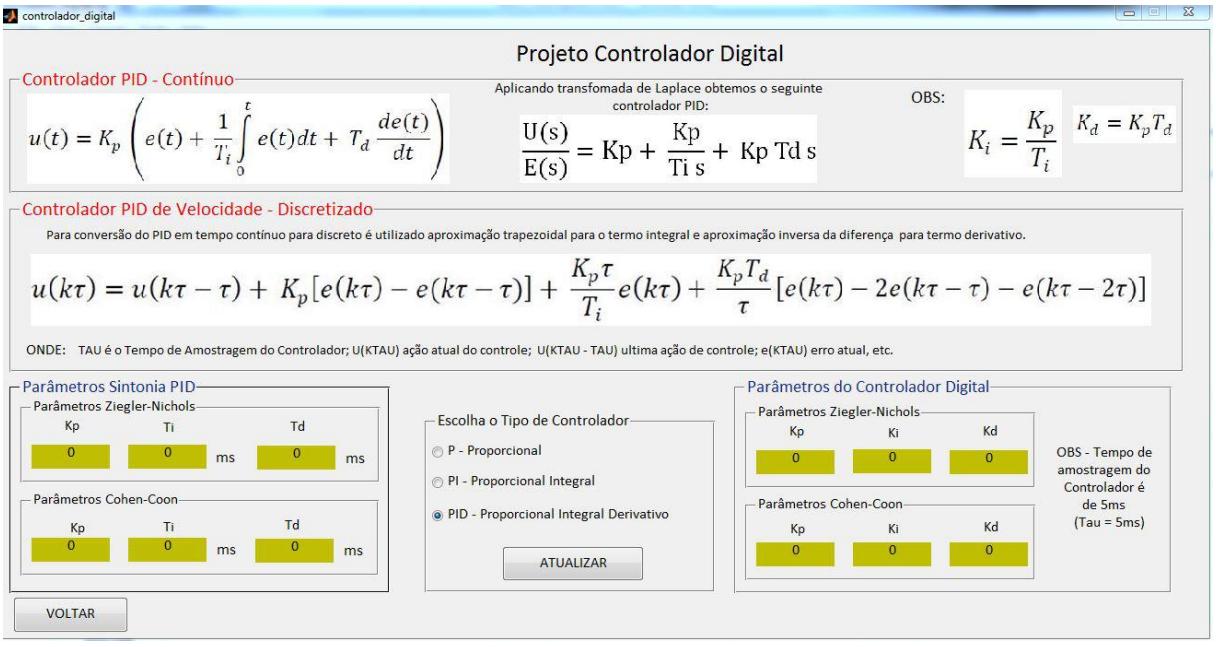


Figura 3.16 - Tela Projeto do Controlador

3.5.6 Tela – Supervisório Bancada

A tela Supervisório Bancada (Figura 3.18) permite ao usuário escolher entre o modo de operação manual ou automático; definir a porcentagem do distúrbio no sistema (freio – motor de polos sombreados); visualizar e salvar as variáveis de interesse (velocidade, erro e tensão) durante a execução do ensaio.

A tela do supervisório é descrita a seguir:

- Escolha da Porta COM e botão Conectar - idem apresentado na tela Resposta ao Degrau;
- Modo Bancada – botão que define qual o modo de operação da bancada (manual / automática). Bancada sempre inicia no modo manual;
- RPM – exibe a velocidade do motor CC, em rotações por minuto;
- Controlador PID
    - K_p, K_i e K_d – Ganhos de controle aplicados no PID de velocidade. Esses podem ser atualizados de forma automática, pela escolha da sintonia e tipo do controlador por meio do botão "Atualizar" ou podem ser inseridos pelo usuário nas respectivas caixas de textos;
    - *Set_Point* – Valor de velocidade do motor CC desejado pelo usuário da bancada;

      Obs – Na mudança do modo de operação da bancada de manual para automática o *Set_Point* assume o valor de velocidade atual do motor CC e na mudança de operação de automática para manual o motor CC assume o valor de referência do seu potenciômetro.
    - Enviar – Botão que envia para o Arduino as configurações do controlador definido pelo usuário.
- Carga – Define qual a intensidade do distúrbio (motor de polos sombreados) de 0 a 100% sobre o motor CC. Dado que o acionamento do motor de polos sombreados é efetuado pelo método de PWM, a porcentagem é convertida em uma faixa de PWM (0 a 255 decimal);
- Gráfico da Velocidade – Apresenta a velocidade do motor CC e do *Set_Point* quando no modo automático;
- Gráfico do Erro – Apresenta o erro da velocidade em relação ao *Set_Point* quando está no modo automático;

- Gráfico da Tensão – Apresenta a tensão aplicada na armadura do motor CC. Conforme apresentado na (seção 3.5.2) é necessário relacionar tensão com PWM. Convertido o sinal de PWM de saída, o sistema supervisório plota a tensão estimada com base no polinômio de 3º grau (Equação 2.23) que foi aproximado com base nos resultados experimentais presente na Figura 3.17.

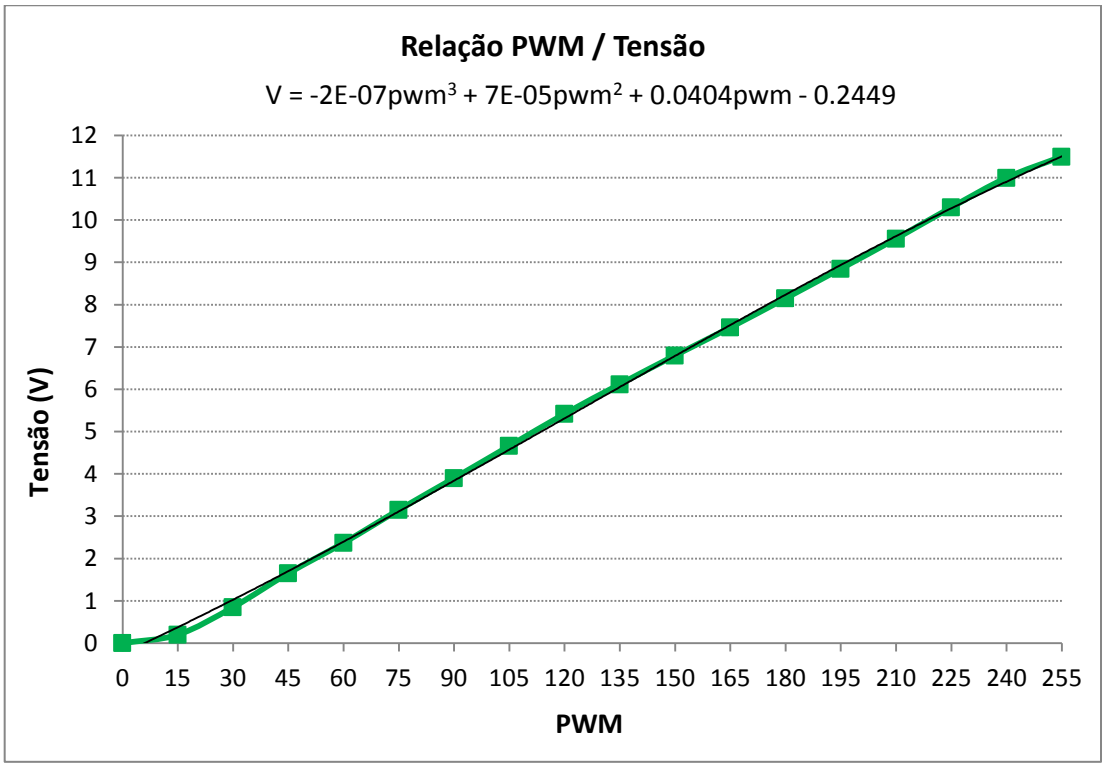


Figura 3.17 - Relação PWM / Tensão

$$V = -2E^{-7}pwm^3 + 7E^{-5}pwm^2 + 0.0404pwm - 0.2449 \quad (2.23)$$

Obs – esta equação é implementada no algoritmo do Arduino.

- Salvar Dados
    - Botão Salvar – Salva os dados que estão sendo visualizados nos respectivos gráficos (velocidade, *Set_Point*, erro, tensão e tempo);
    - Análise dos Resultados – Apresenta os gráficos dos dados coletados no botão salvar.
- Botão Voltar – Volta a Tela Inicial.

Obs – Todo controle da Bancada só é disponível após conexão com o supervisório.

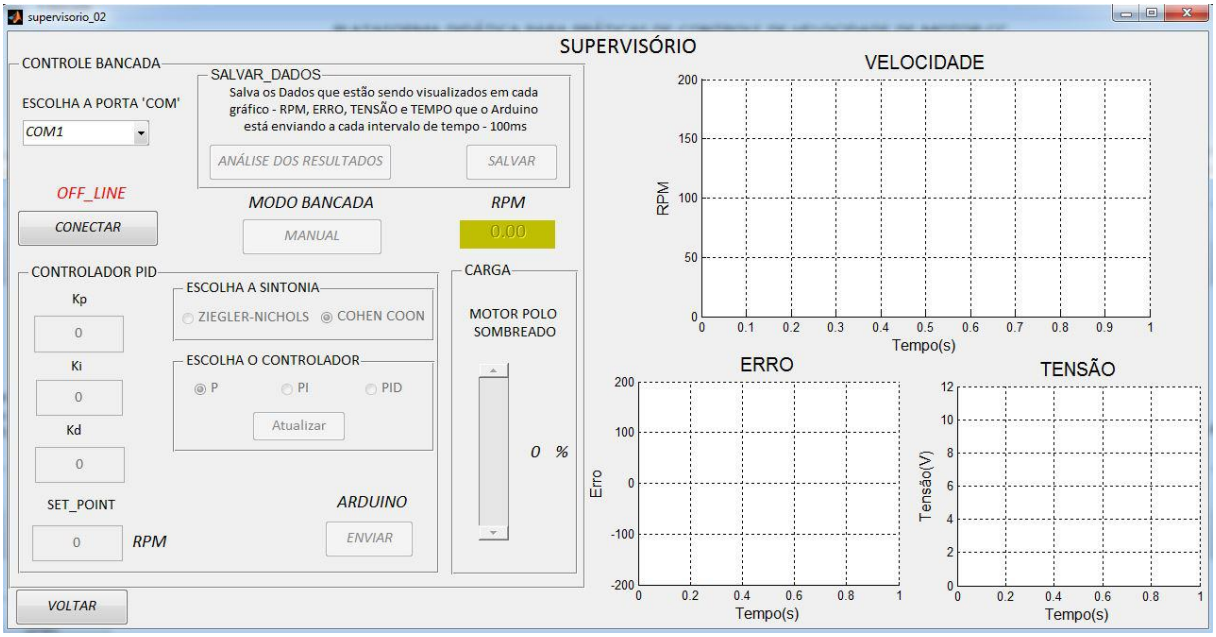


Figura 3.18 - Tela Supervisório Bancada

3.6 Arquitetura do *Software*

Para entendimento do funcionamento do sistema Supervisório, a Figura 3.19 demonstra a arquitetura do software com a plataforma Arduino.

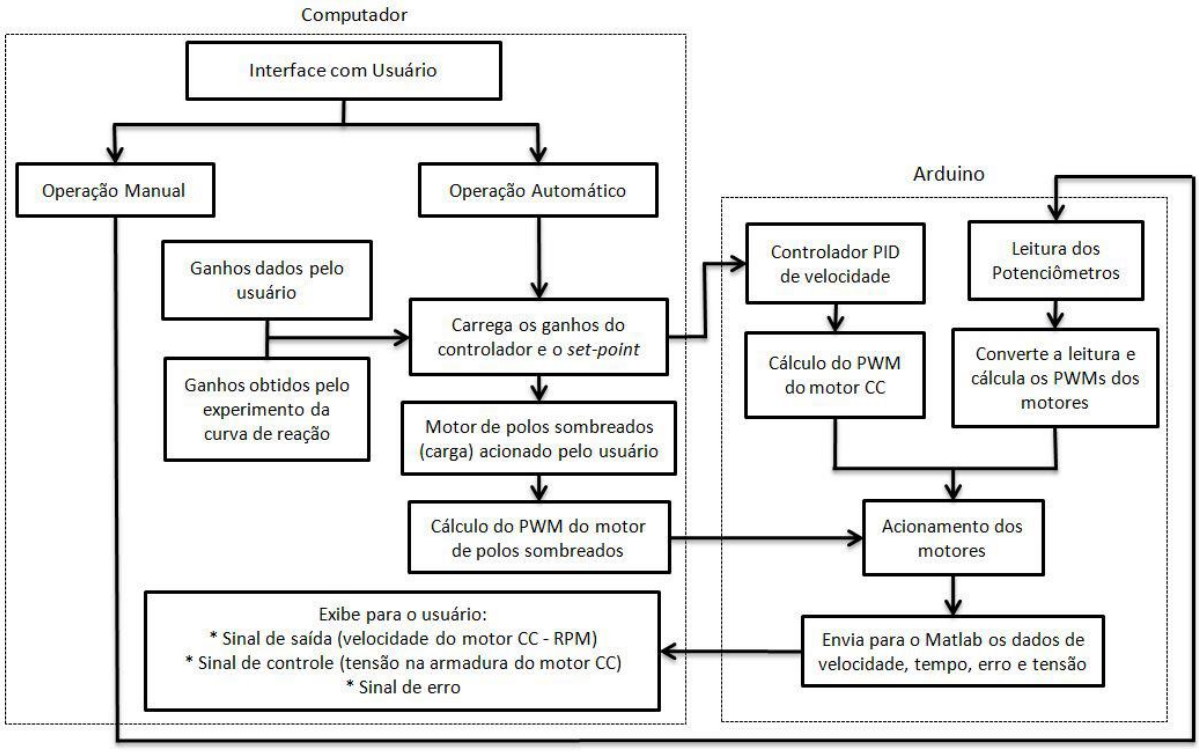


Figura 3.19 – Arquitetura do Software

Na Figura 3.20 temos o fluxograma do controlador PID de velocidade do motor CC, implementado no *firmware* do Arduino.

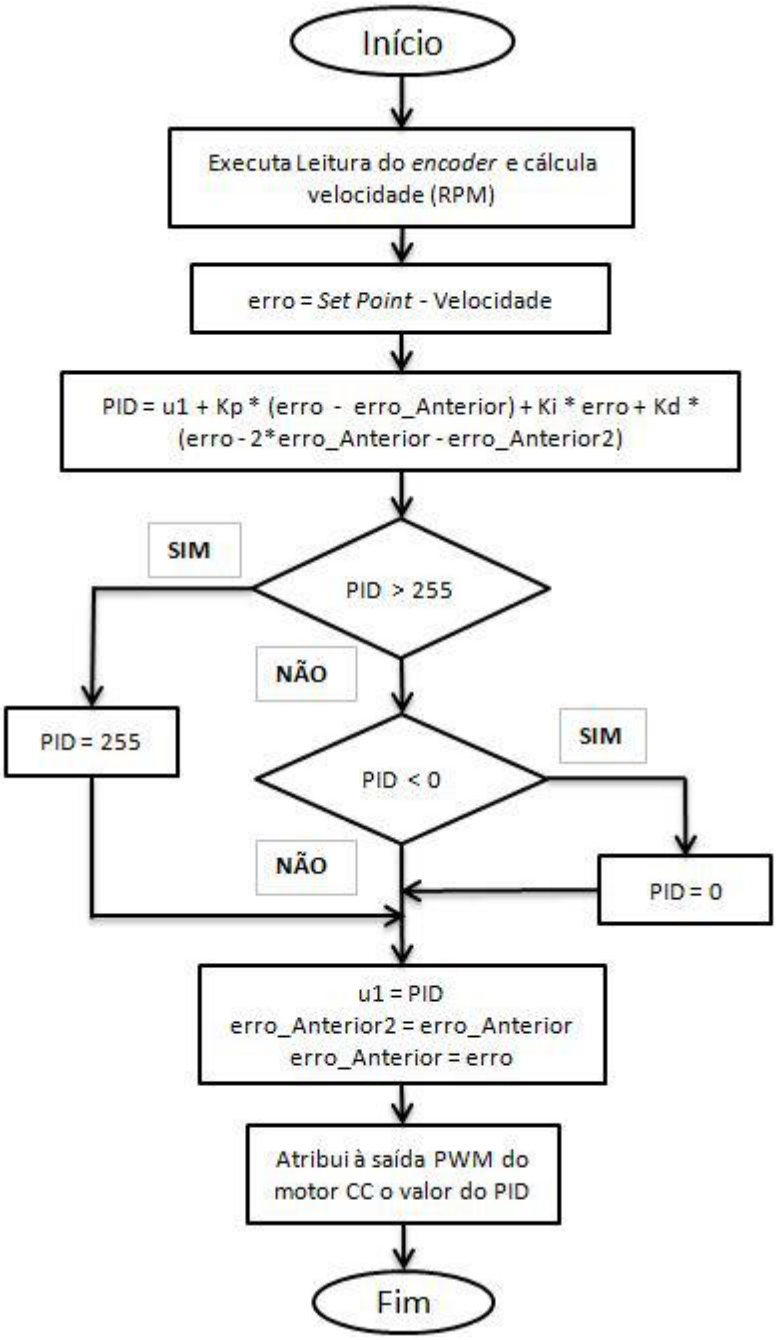


Figura 3.20 - Fluxograma do Controlador PID de Velocidade

4 RESULTADOS

4.1 A Bancada Didática

As peças em acrílico foram cortadas a laser em empresa especializada localizada em Belo Horizonte. A colagem das peças em acrílico que compõe a bancada foi efetuada no laboratório de Máquinas Elétricas da Escola de Minas.

Para o projeto do *hardware* foram realizadas as seguintes etapas:

- Dimensionamento da placa de circuito impresso – placa de fibra de vidro;
- Transferência térmica do circuito para placa – prensa térmica;
- Corrosão da placa – utilização do percloreto de ferro (FeCL3);
- Perfuração da placa – furadeira de bancada;
- Verificação das trilhas – multímetro;
- Soldagem dos componentes.

Na Figura 4.1 temos a bancada didática que foi desenvolvida, na qual se verifica o condicionamento e conexão dos motores, a disposição dos circuitos e dos demais componentes da bancada.

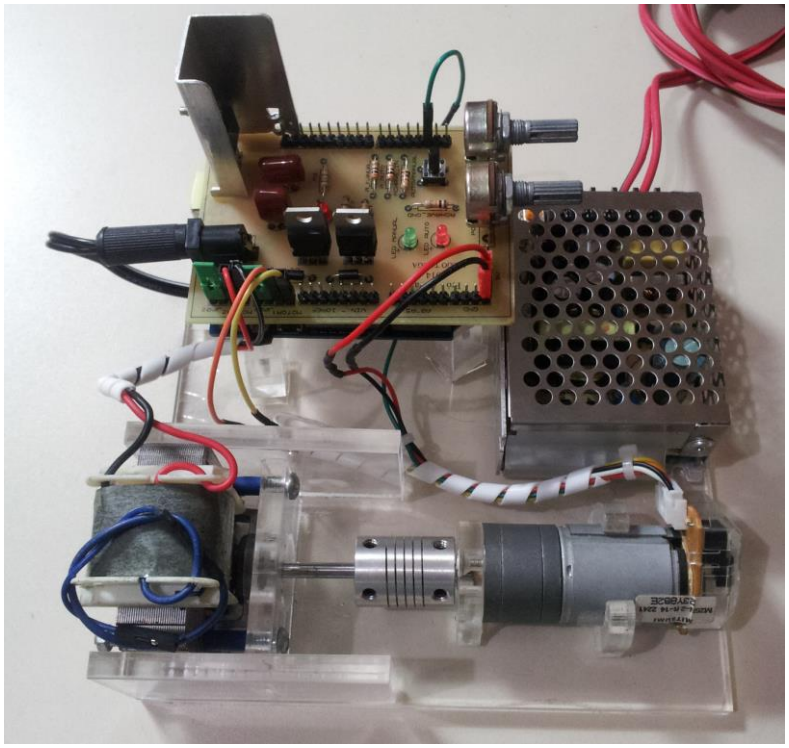


Figura 4.1 - A Bancada Didática

4.2 Projeto do Controlador

Com base no passo a passo do supervisório, são apresentados ao usuário os resultados de cada etapa que compõe o método da curva de reação, os resultados das sintonias Ziegler-Nichols e Cohen-Coon e o desempenho de cada sintonia aplicada ao controlador PI de velocidade.

4.2.1 Resposta ao Degrau

Para realização do ensaio de resposta ao degrau da bancada é empregado o procedimento descrito em Goodwin, Graebe e Salgado (2000), conforme apresentado na subseção 2.6.7.

Para ilustrar o ensaio de reposta ao degrau, vamos utilizar os seguintes parâmetros:

- Tensão inicial na armadura do motor CC – 7Vcc;
- Porcentagem do degrau – 20% em relação ao fundo de escala (12Vcc);
- Intervalo de amostragem – 8ms.

A partir destes parâmetros obtivemos a resposta como apresenta na Figura 4.2. Nesta figura temos a velocidade no eixo do motor CC, antes da redução, para o primeiro intervalo em regime permanente, dada uma tensão de 7Vcc no motor, e um segundo patamar em regime permanente, associado a uma entrada constante de 9.4Vcc na armadura do motor CC.

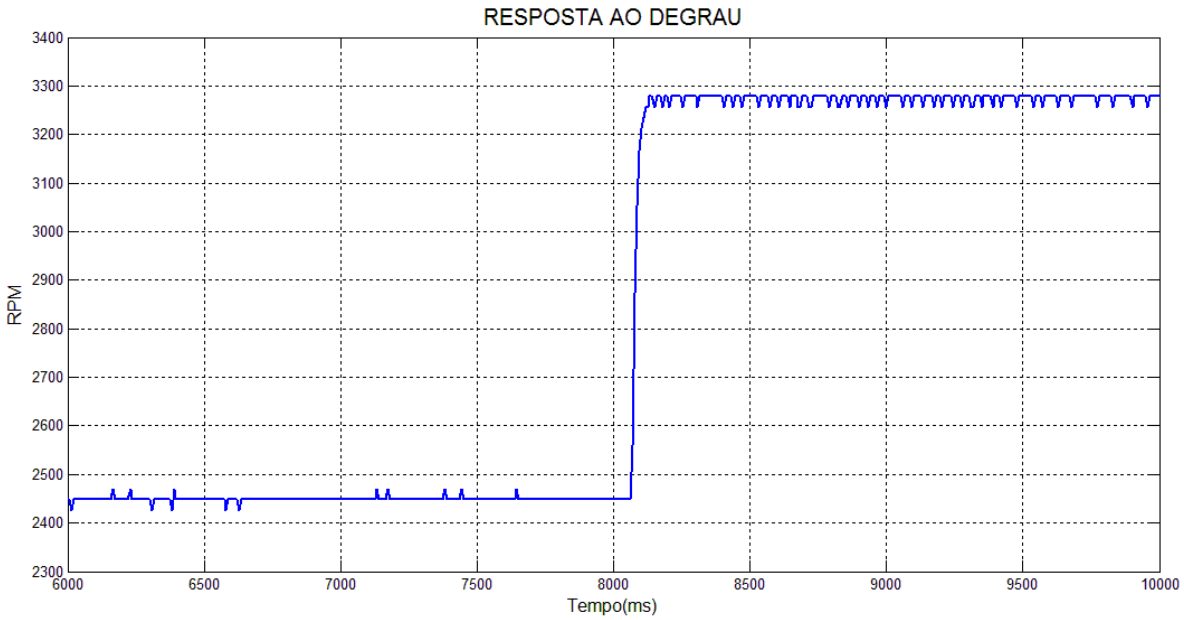


Figura 4.2 - Resposta ao Degrau

4.2.2 Método da Curva de Reação

Utilizando a suavização de dados igual a 1 para os dados coletados na resposta ao degrau, a Figura 4.3 apresenta o cálculo da derivada para cada ponto amostrado da saída (velocidade)

no ensaio de resposta ao degrau. Na Figura 4.4 temos estes dados brutos e suavizados (quando utilizado um valor maior que 1 para a função *smooth*) da curva de reação.

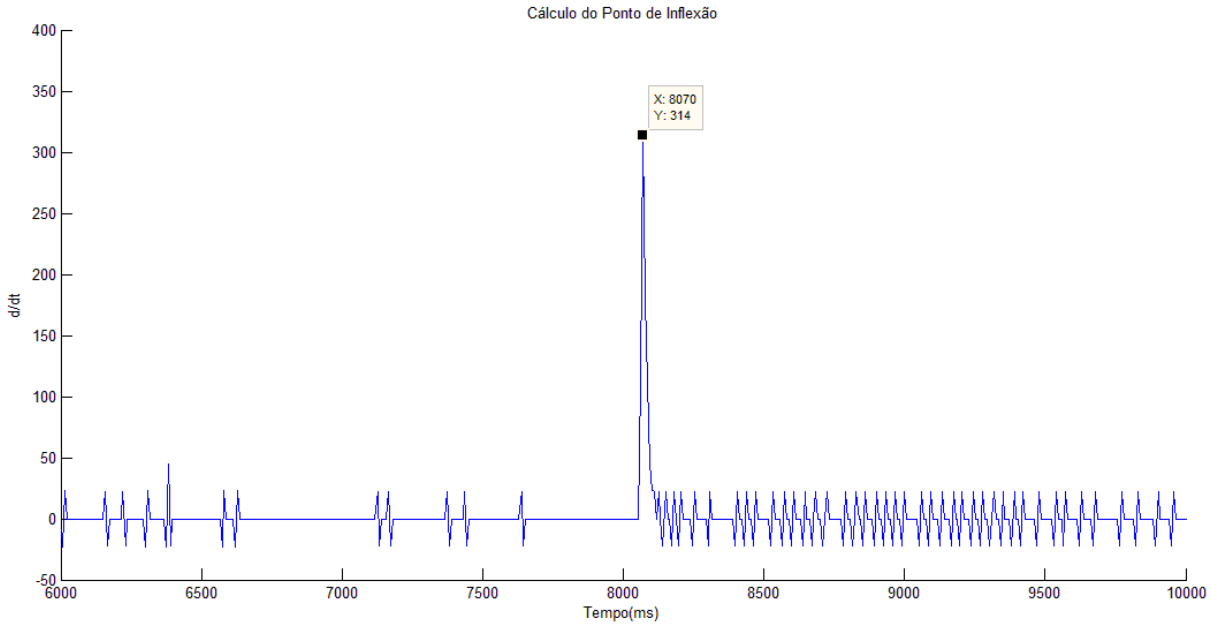


Figura 4.3 - Cálculo do Ponto de Inflexão

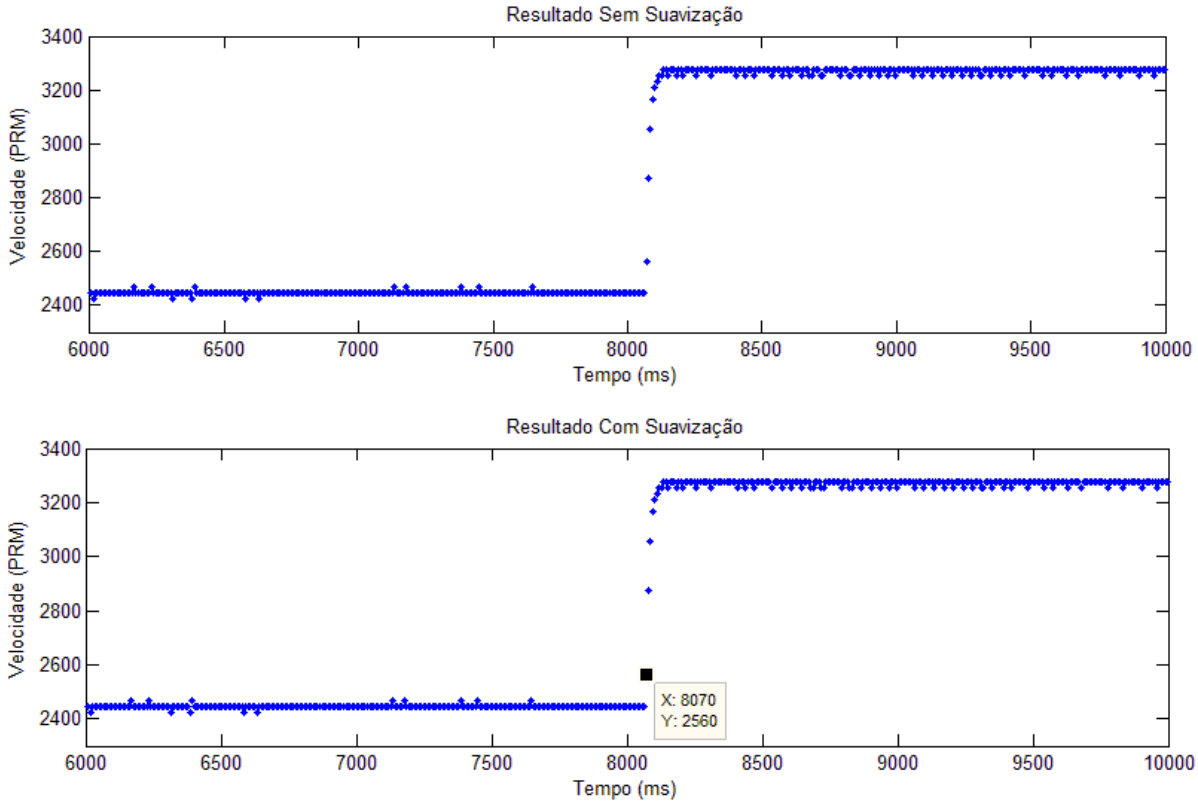


Figura 4.4 - Resposta ao Degrau Tratada

Conforme apresentado na subseção 3.5.3, a partir dos gráficos das figuras 4.3 e 4.4, é extraído as seguintes informações:

- Tempo do ponto de inflexão = 8070 ms (Figura 4.3);
- Valor da derivada no ponto de inflexão = 314 (Figura 4.3);
- Velocidade no gráfico com suavização = 2560 RPM (Figura 4.4).

Na Figura 4.5 temos o gráfico da curva de reação com a identificação dos intervalos de tempo associados ao atraso (L) e a constante de tempo do sistema (T).

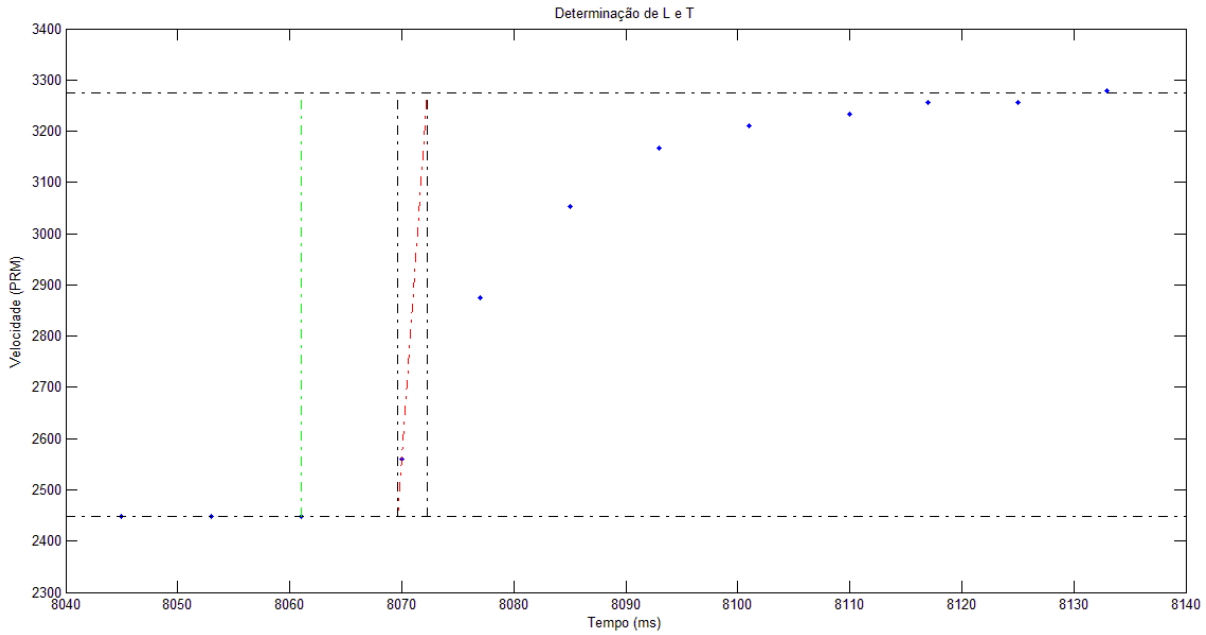


Figura 4.5 - Determinação de L e T

Na Figura 4.6 temos a exibição das constantes K_0, L e T que foram determinados empiricamente pelo ensaio de curva de reação a um degrau na entrada.

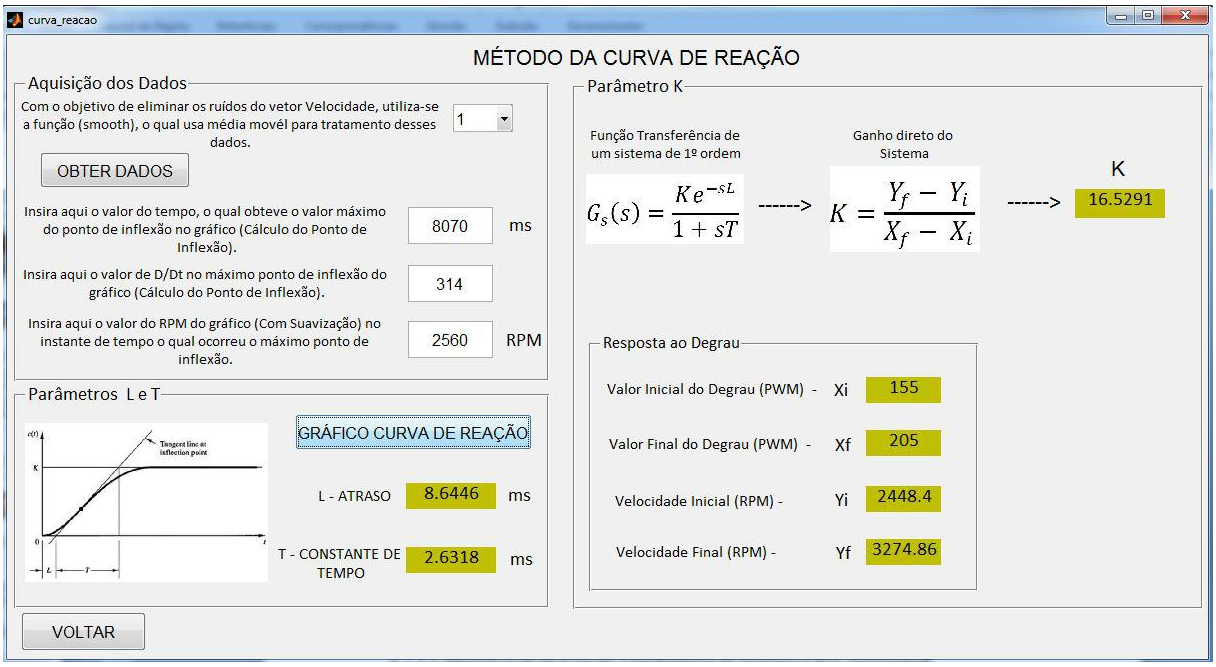


Figura 4.6 - Resultado Tela Método da Curva de Reação

K = 16.5291, L = 8.6446ms e T = 2.6318ms que são valores determinados no ensaio anterior.

4.2.3 Identificação da Função Transferência do Sistema e do Controlador

De acordo com Goodwin, Graebe e Salgado (2000) a função transferência do sistema em malha aberta foi obtida a partir da Equação 2.22, com base nos parâmetros K, L e T determinados no ensaio anterior.

$$G_S(S) = \frac{16.5291e^{-8.6446ms\ S}}{2.6318ms\ S + 1} \tag{2.24}$$

Os ganhos do controlador PI foram determinados pelos métodos de sintonia por Ziegler-Nichols e Cohen-Coon, conforme apresentado a seguir.

4.2.4 Ziegler Nichols – Determinação dos Ganhos do Controlador PI

O ganho K_p e o tempo integrativo T_i do controlador presente na Tela Sintonia Controlador PID (Figura 4.9) foram determinados a partir da Tabela 2.3.

$$K_p = 0.9\ \frac{T}{KL} = 0.9\ \frac{2.6318ms}{16.5291\ x\ 8.6446ms} = 0.016577 \tag{2.25}$$

$$T_i = 3\ L = 3\ x\ 8.6446ms = 25.9338ms \tag{2.26}$$

Por meio da Equação 2.10 determinou-se a função de transferência do controlador.

$$G_C(S) = 0.016577\left(1 + \frac{1}{25.9338ms\ S}\right) = \frac{(0.429ms\ S + 0.016577\ )}{25.9338ms\ S} \qquad (2.27)$$

Na Figura 4.7 temos o diagrama de blocos do sistema em malha fechada com o controlador PI.

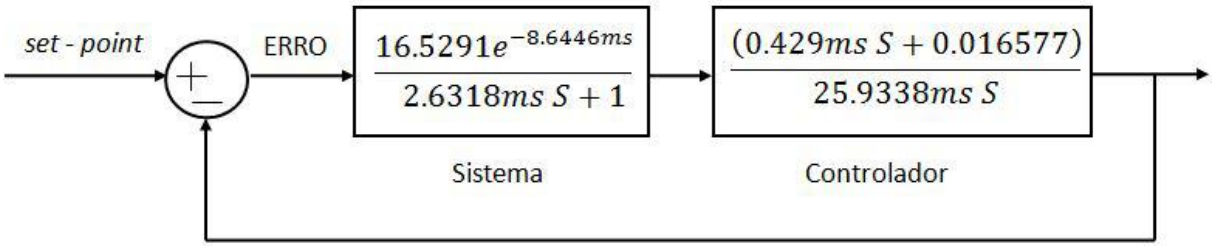


Figura 4.7 - Diagrama de Blocos Segundo Sintonia Ziegler-Nichols

4.2.5 Cohen Coon – Determinação dos Ganhos do Controlador PI

O ganho K_p e o tempo integrativo T_i do controlador exibidos na Tela Sintonia Controlador PID (Figura 4.9) foram determinados a partir da Tabela 2.4.

$$K_P = \frac{T}{KL}\left(0.9 + \frac{L}{12T}\right) = \frac{2.6318ms}{16.5291\ x\ 8.6446ms}\left(0.9 + \frac{8.6446ms}{12\ x\ 2.6318ms}\right) = 0.021618 \qquad (2.28)$$

$$T_i = \frac{L(30T + 3L)}{9T + 20L} = \frac{8.6446ms(30\ x2.6318ms + 3\ x\ 8.6446ms\ )}{9\ x\ 2.6318ms + 20\ x\ 8.6446ms} = 4.6125ms \qquad (2.29)$$

Por meio da Equação 2.10 determinou-se a função de transferência do controlador.

$$G_C(S) = 0.021618\left(1 + \frac{1}{4.6125ms\ S}\right) = \frac{(0.099ms\ S + 0.021618)}{4.6125ms\ S} \qquad (2.30)$$

Na Figura 4.8 temos o diagrama de blocos do sistema em malha fechada com o controlador PI.

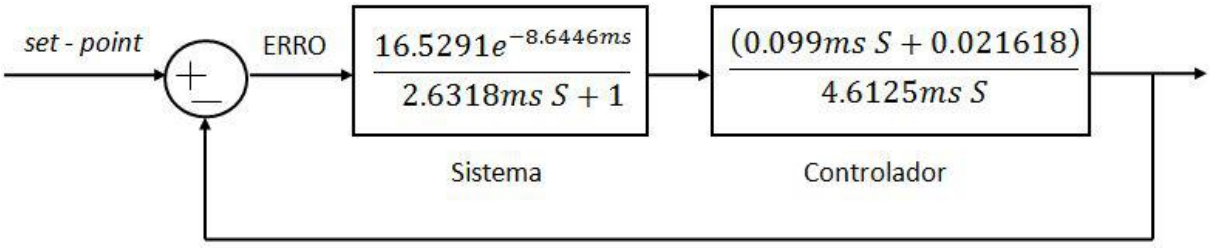


Figura 4.8 - Diagrama de Blocos Segundo Sintonia Cohen-Coon

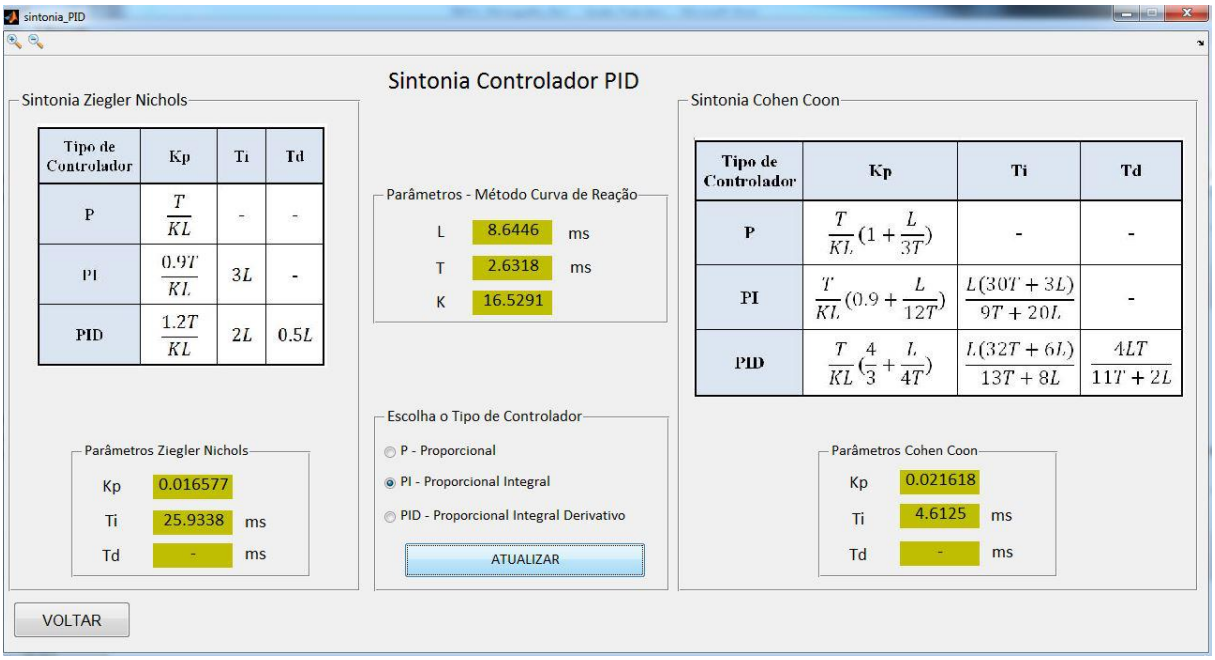


Figura 4.9 - Tela Sintonia Controlador PID Contínuo

4.2.6 Discretização do Controlador PI

Para implementação do controlador PI de velocidade foi utilizado a Equação 2.21. A ação de controle executada pelo Arduino ocorre a cada 5ms, sendo este o intervalo de amostragem utilizado no cálculo do ganho K_i.

Os ganhos do controlador PI por Ziegler-Nichols discretizado ($K_p = 0.016577$ e $K_i = 0.003196$) foram determinados pelo cálculo da Equação 2.31. Na Figura 4.10 temos a interface do usuário que exibe estes resultados.

$$u(k\ x\ 5) = u(k\ x\ 5-5) + 0.016577\ [e(k\ x\ 5) - e(k\ x\ 5-5)] + \frac{0.016577\ \ x\ 5ms}{25.9338ms}e(k\ x\ 5) \quad (2.31)$$

$$u(k\,x\,5) = u(k\,x\,5-5) + 0.016577\,[e(k\,x\,5) - e(k\,x\,5-5)] \\ +0.003196e(k\,x\,5) \tag{2.32}$$

De modo similar, realiza-se o cálculo da discretização do controlador PI segundo sintonia de Cohen-Coon, obtendo $K_p = 0.021618$e $K_i = 0.023434$.

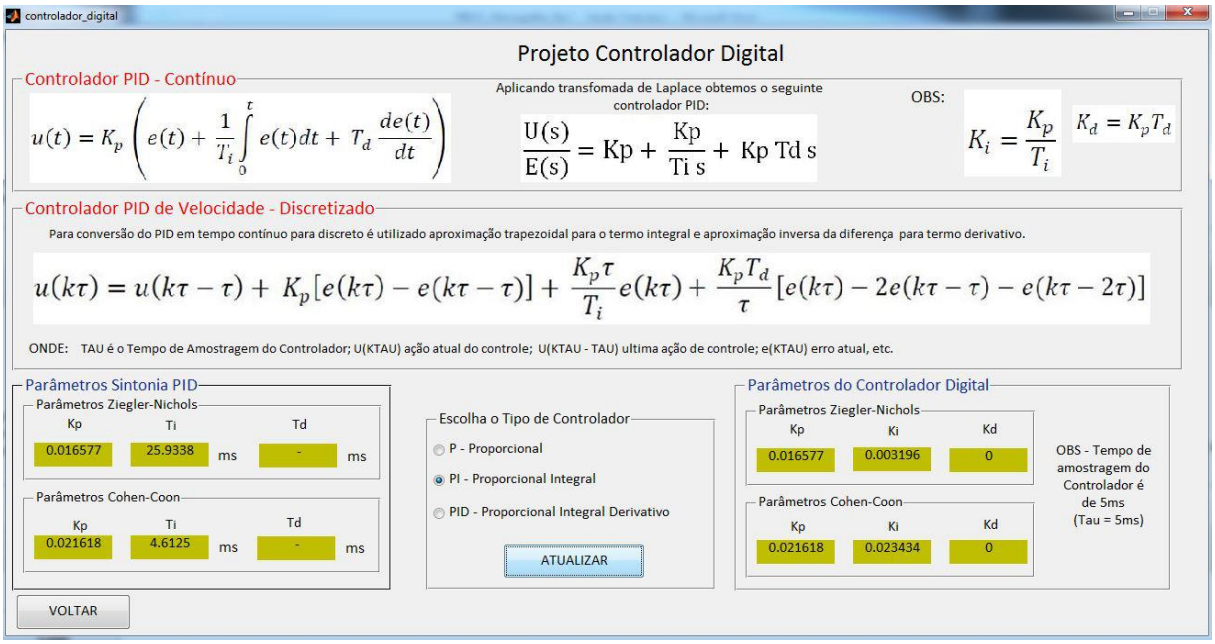


Figura 4.10 - Tela Projeto Controlador Digital

4.3 Resposta do Controlador sem Alteração de Carga no Motor

Dado o objetivo de verificar o desempenho do controle de velocidade do motor CC, a partir das sintonias Ziegler-Nichols e Cohen-Coon, foram realizados ensaios para diferentes pontos de operação (velocidade do motor).

Ao iniciar a operação da bancada, foi colocado o motor CC em uma velocidade de 100RPM no modo manual (potenciômetro). Em seguida, foi alterado o status de operação da bancada para modo automático, no qual o controlador assume sua função e o valor da referência (*set-point*) é atribuído ao valor atual da velocidade. Por fim, conforme apresentado na Figura 4.11 foi alterado a referência de velocidade na seguinte sequência 140RPM, 90RPM, 115RPM e 100RPM, aguardando sempre o sistema entrar em regime antes da alteração da referência do sistema. Na mesma figura temos os gráficos do sinal de entrada da bancada (tensão na armadura do motor) e sinal do erro em RPM.

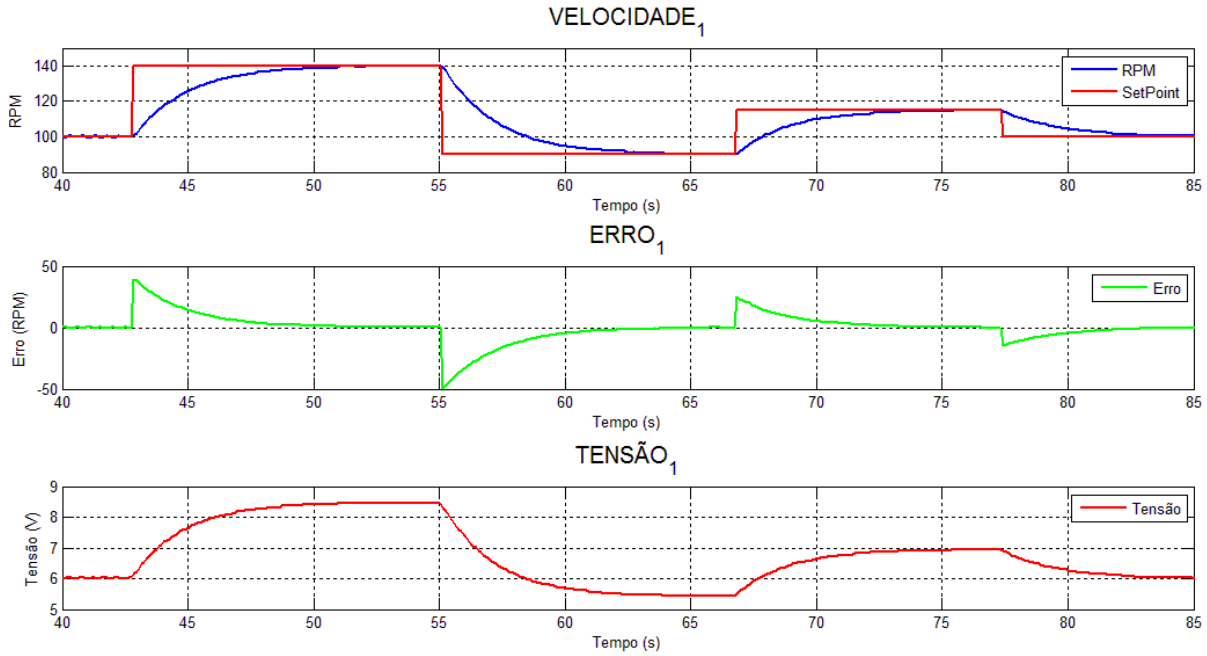


Figura 4.11 - Resposta do Controlador PI para Sintonia Ziegler-Nichols

Na Figura 4.12 temos o mesmo ensaio com a resposta do sistema com o controlador sintonizado pelo método de Cohen-Coon.

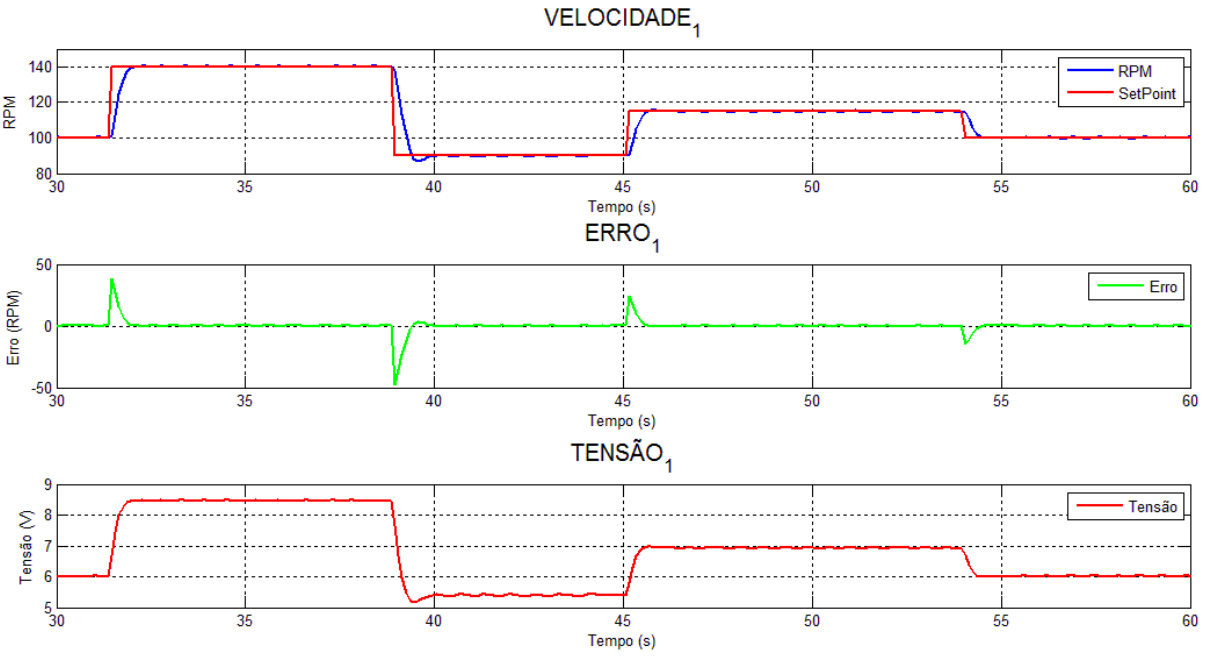


Figura 4.12 - Resposta do Controlador PI para Sintonia Cohen-Coon

Analisando os experimentos em malha fechada, é possível observar que para sintonia de Cohen-Coon a reposta foi mais rápida. Isto é justificado em Ibrahim (2006), que para o controlador de velocidade da Equação 2.21 aplicado a bancada, se existir um grande erro, a resposta do controlador pode ser lenta, especialmente se o tempo de integração (T_i) é grande.

Neste caso temos para Ziegler-Nichols $T_i = 25.9338ms$ e para Cohen-Coon $T_i = 4.6125ms$, representando uma razão de aproximadamente 5.62 vezes mais para a sintonia Ziegler-Nichols.

4.4 Resposta do Controlador com Alteração de Carga no Motor

Para demonstrar o efeito do freio no sistema, a Figura 4.13 apresenta três intervalos que estão numerados e em realce. No primeiro intervalo (1), com atuação de 100% do freio (motor de polos sombreados em 36Vcc) o sistema está em malha aberta, portanto não consegue manter a velocidade de operação quando sujeito ao distúrbio de carga. No instante de 37s a carga é desacoplada do sistema. Já no instante de tempo próximo de 41s ocorre a passagem para operação em modo automático, considerando o controlador PI sintonizado pelo método de Cohen-Coon. Em 47.5s, ocorre a mudança de referência de 100 para 115RPM. Note que o controlador compensa o erro em função da alteração do sinal de referência. No segundo intervalo (2), ocorre a atuação de 100% do freio, que resulta em um erro, o qual é corrigido pela ação do controlador, restabelecendo a velocidade em 115RPM. No terceiro intervalo (3), ocorre a retirada da carga, que resulta em um novo erro, que novamente é compensado pela ação do controlador.

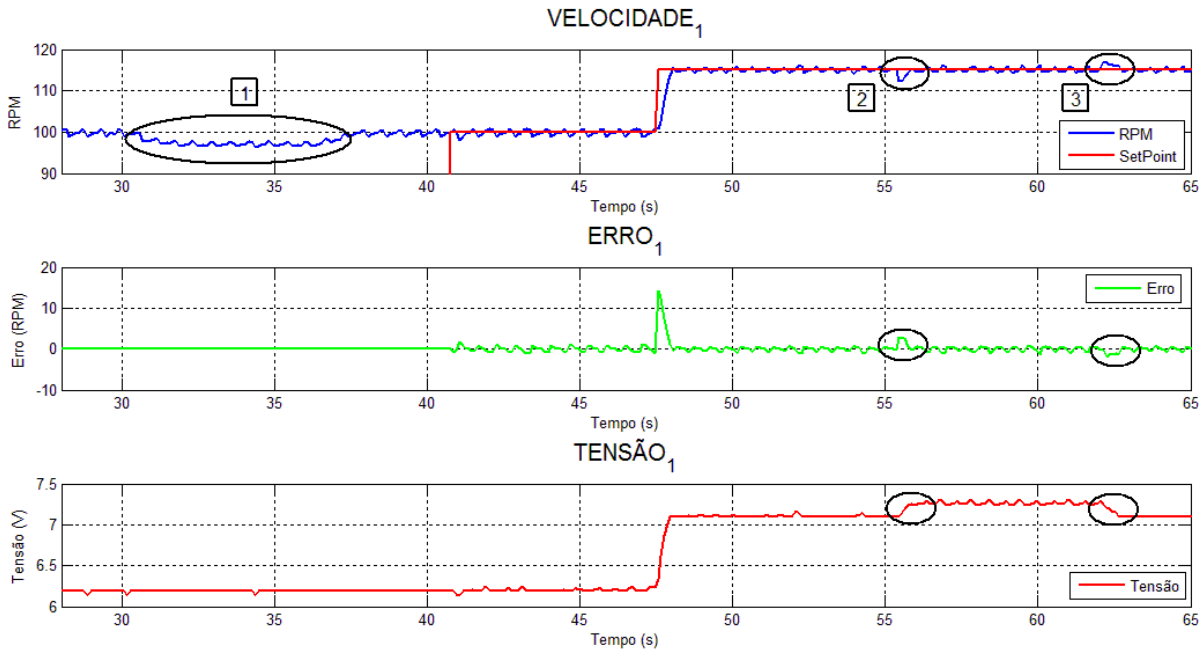


Figura 4.13 - Resposta do Controlador com Alteração de Carga no Motor

5 CONCLUSÃO

A plataforma possibilita ao discente a experiência com o ensaio da curva de reação ao degrau e a compreensão dos parâmetros relacionados ao ganho direto do sistema (K_0), atraso (L) e constante de tempo do sistema (T) característicos em sistema de primeira ordem.

As sintonias de Ziegler-Nichols e Cohen-Coon obtidas por meio do método da curva de reação se mostraram satisfatórias. Como esperado, o desempenho do controlador sintonizado por Cohen-Coon apresentou pouca influência da razão do atraso em relação a constante de tempo (L/T). O controlador PID de velocidade discretizado implementado no *firmware* do Arduino foi eficaz em diversos pontos de operação, apresentando desempenho satisfatório quando o sistema está sujeito a alteração de carga.

A interface com usuário (GUI) desenvolvida no *software* Matlab apresenta fácil manipulação e possibilita análise dos resultados obtidos pelo usuário, podendo os dados experimentais serem carregados no *software* Matlab para posterior análise.

Dados os resultados de controle, a plataforma apresenta uma ferramenta útil para consolidação das disciplinas de controle, uma vez que os discentes podem aplicar em um sistema real a teoria ministrada em sala de aula e entender a importância da sintonia de um controlador PID.

Como sugestões para trabalhos futuros sugere disponibilizar a bancada para acesso remoto, e implementar um sistema de controle de velocidade por meio de *sensorless*.

REFERENCIAL TEÓRICO

ARAÚJO, I.; FREITAS, V.; SILVA, O. **Desenvolvimento e aplicação de motores de corrente contínua virtuais aplicados nas aulas laboratoriais de controle de sistemas**. XXXIX congresso brasileiro de educação de engenharia. Blumenau: [s.n.]. 2011.

ARDUINO. **Arduino uno**, 2015. Disponivel em: <https://www.arduino.cc/en/Main/ArduinoBoardUno>. Acesso em: 20 nov. 2015.

CAMPOS, M. C. M.; TEIXEIRA, G. C. H. **Controles Típicos de Equipamentos e Processos Industriais**. 1ª. ed. São Paulo: Edgar Blucher, 2006. ISBN ISBN 85-212-0398-5.

CARLOS, T. E.; MORAES, C. H. V. D.; RODRIGO, A. M. A. M. **Busduino - interface de baixo custo para protocolo industrial**. ITAJUBÁ. 8pg.

COCOTA, J. A. N. J.; ABRÃO, D. C.; LOPES, A. G.; MEDEIROS, M. R. O.; LOPES, E. S. S. **Development of Tangible Experiments for Motivating Undergraduate Students**. IEEE Global Engineering Education Conference (EDUCON). Berlin: [s.n.]. 13 - 15 Março 2013. p. 497-506.

DATASHEET ATMEGA328P, Atmel , 2009. Disponível em <http://www.atmel.com/images/doc8161.pdf>. Acesso em 12 Ago. 2014.

DATASHEET LM7812, Fairchild , 2014. Disponível em <https://www.fairchildsemi.com/datasheets/LM/LM7812.pdf>. Acesso em 05 Jul. 2015.

DATASHEET TIP122, Fairchild , 2014. Disponível em <https://www.fairchildsemi.com/datasheets/TI/TIP120.pdf>. Acesso em 15 Jul. 2015.

DE SOUZA, B. R. **Uma Arquitetura para Sistemas Supervisórios Industriais e sua Aplicação em Processo de Elevação Artificial de Petróleo**. Natal, Fevereiro 2005.

DORF, R. C.; BISHOP, R. H. **Sistemas de Controle Modernos**. 8ª. ed. RIO DE JANEIRO: LTC, 2001. ISBN ISBN 0-201-30864-9.

FILIPPO, F. **Motor de Indução**. São Paulo: Érica, 2000.

FITZGERALD, E. . A.; CHARLES, K.; UMANS, D. . S. **Máquinas Elétricas**. 6ª. ed. São Paulo: [s.n.], 2006.

GOMES, B.; TAVARES, A. L. **Uma solução com Arduino para controlar e monitorar processos industriais.** abr. 2013. 4. Disponivel em: <https://www.arduino.cc/en/Guide/Introduction>. Acesso em: 08 Setembro 2015.

GOODWIN, C.; GRAEBE, F.; SALGADO, E. **Control System Design**. Valparaíso: [s.n.], 2000.

IBRAHIM, D. **Microcontroller Based Applied Digital Control**. [S.l.]: Jonh Wiley & Sons,LTD, 2006. ISBN ISBN:0-470-86335-8.

KARDEK, R. S. A.; COCOTA, A. N. J.; FERREIRA, V. M. D. **Development of an educational tool for Control Engineering**. OURO PRETO, Março 2015.

KARDEK, R. S. A.; RODRIGUES, C. L. C. **Eletrônica de Potência e Acionamentos Elétricos**. Ouro Preto: [s.n.], 2015.

LOURENÇO, J. **Sintonia De Controladores P.I.D**. Janeiro 1996.

MALVINO, P. A. **Eletrônica**. 4ª. ed. [S.l.]: [s.n.], v. 1, 1997.

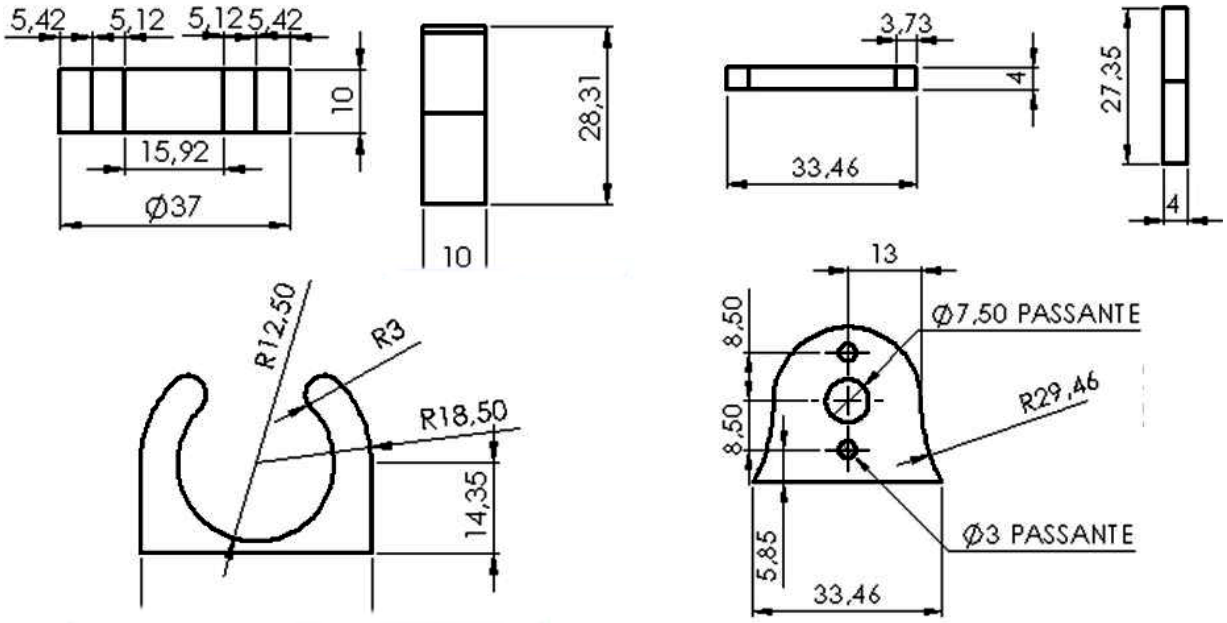
OGATA, K. **Engenharia de Controle Moderno**. 4ª. ed. [S.l.]: Prentice Hall, 2002.

PEREIRA, F. **Microcontroladores PIC: Programação em C.** 1ª Edição. ed. São Paulo: Érica, 2005.

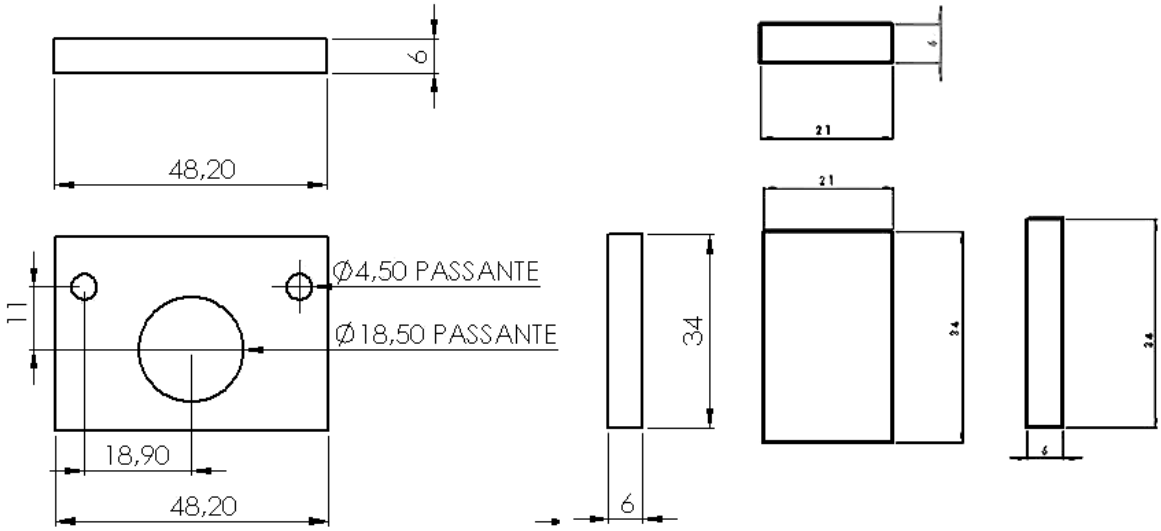
SHUI-CHUN, L.; CHING-CHIH, T. **Development of a Self-Balancing Human Transportation Vehicle for the Teaching of Feedback Control**. IEEE Transactions on Education, v. v.52, p. p. 157-168, 2009.

SOUZA, J. D. **Desbravando o PIC Ampliado e Atualizado para PIC16F628A**. 8ª Edição. ed. São Paulo: Érica, 2005.

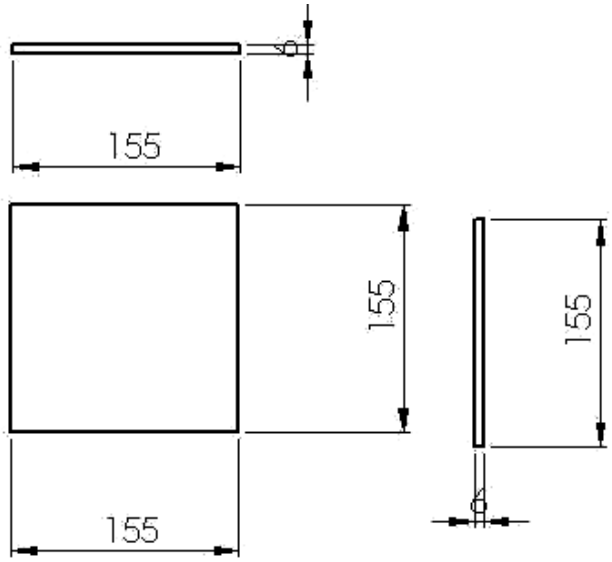
ANEXO



Peças de Suporte do Motor CC



Peças de Suporte do Motor de Polos Sombreados



Peça da Base da Bancada

Obs: as cotas estão em mm.