



**CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS  
UNIDADE ARAXÁ**

**MATHEUS MELO BORGES**

**SISTEMA DIDÁTICO PARA APLICAÇÃO DE TÉCNICAS DE  
CONTROLE**

**ARAXÁ/MG**

**2016**

**MATHEUS MELO BORGES**

**SISTEMA DIDÁTICO PARA APLICAÇÃO DE TÉCNICAS DE  
CONTROLE**

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia de Automação Industrial, do Centro Federal de Educação Tecnológica de Minas Gerais - CEFET/MG, como requisito parcial para obtenção do grau de Bacharel em Engenharia de Automação Industrial.

Orientadora: Prof.<sup>a</sup> Dra. Aline Fernanda Bianco  
Coorientador: Prof. Me. Luis Paulo Fagundes

**ARAXÁ/MG**

**2016**



**Serviço Público Federal**  
**MINISTÉRIO DA EDUCAÇÃO**  
**CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS**  
**COORDENAÇÃO DO CURSO DE ENGENHARIA DE AUTOMAÇÃO INDUSTRIAL/ ARAXÁ**

**TRABALHO DE CONCLUSÃO DE CURSO – TCC – ATA DE DEFESA**

ATA DA DEFESA DE TRABALHO DE CONCLUSÃO DE CURSO DE ENGENHARIA DE AUTOMAÇÃO INDUSTRIAL do aluno *Matheus Melo Borges*

Às 17h do dia 11 de julho de 2016, reuniu-se, no Centro Federal de Educação Tecnológica de Minas Gerais – CEFET-MG/ Unidade Araxá, a Comissão Examinadora de Trabalho de Conclusão de Curso para julgar, em exame final, o trabalho intitulado **Sistema Didático para Aplicação de Técnicas de Controle** como parte dos requisitos para a obtenção do Título de Engenheiro de Automação Industrial. Abrindo a sessão, a Presidente da Comissão, Profa. Dra. Aline Fernanda Bianco, após dar a conhecer aos presentes o teor das Normas Regulamentares do Trabalho Final, passou a palavra ao candidato para apresentação de seu trabalho. Seguiu-se a arguição pelos examinadores, com a respectiva defesa do candidato. Logo após, a Comissão se reuniu, sem a presença do candidato e do público, para julgamento e expedição do resultado final.

Após a reunião da Banca Avaliadora, o candidato foi considerado: *aprovado* Com nota final de: *90,25* / 100.

O resultado final foi comunicado publicamente ao candidato pela Presidente da Comissão. O aluno abaixo-assinado declara que o trabalho ora identificado é da sua autoria material e intelectual, excetuando-se eventuais elementos, tais como passagens de texto, citações, figuras e datas, desde que as mesmas identifiquem claramente a fonte original, explicitando as autorizações obtidas dos respectivos autores, quando necessárias. Declara ainda, neste âmbito, não estar a violar direitos de terceiros.

Matheus Melo Borges  
*Nome do aluno*

*Matheus Melo Borges*  
*assinatura*

Araxá, 11 de julho de 2016.  
*local e data*

Nada mais havendo a tratar, a Presidente encerrou os trabalhos e lavrou a presente ATA, que será assinada por todos os membros participantes da Comissão Examinadora.

Araxá, 11 de Julho de 2016.

*Aline F. Bianco*  
Prof. Dra. Aline Fernanda Bianco (Orientadora)

*Luis Paulo Fagundes*  
Prof. Me. Luis Paulo Fagundes

*Francisco de Assis Cipresso*  
Prof. Esp. Francisco de Assis Cipresso

*Carlos Renato Noll*  
Prof. Carlos Renato Noll

**DEDICO ESTE TRABALHO**

*Aos meus pais,  
que sempre me incentivaram nos momentos mais difíceis.*

## **AGRADECIMENTOS**

Aos meus pais.

À minha orientadora e ao meu coorientador que me auxiliaram durante o desenvolvimento deste trabalho.

Aos meus amigos de escola que sempre estiveram ao meu lado, tornando essa caminhada mais prazerosa.

## RESUMO

Neste trabalho de conclusão de curso foi desenvolvido um sistema didático que futuramente poderá servir de apoio para as disciplinas que abordam a Teoria de Controle nos laboratórios, bem como nas aulas teóricas do CEFET-MG Campus IV Araxá. O sistema permitirá que se utilize técnicas de controle e sintonia de controladores, fortemente difundidos na literatura, para fazer o controle de uma planta. Foi feito um estudo para se obter o modelo matemático de um circuito elétrico e, após isso, este circuito elétrico desempenhou o papel da planta neste sistema. Por meio de um microcontrolador Arduino foi possível fazer a aquisição do sinal de saída da planta e a aplicação das ações de controle foram geradas pelo software MATLAB, que foi utilizado como ferramenta de interface e controlador.

**Palavras-chave:** Sistema didático. Controle clássico. Arduino. Sintonia de controladores.

## **ABSTRACT**

In this project was developed a didactic system that may be support for the disciplines of control theory in the laboratory, and the CEFET-MG Campus IV Araxá classroom as well. The system allows using control techniques and controller tuning, based in literature, to control the plant. A study was made to obtain the mathematical model of an electric circuit, after that, the electric circuit make the role of the plant in this system. Through an Arduino microcontroller was possible to make an acquisition of the output signal of the plant and the control actions were generated by MATLAB software, which was used as an interface tool and controller.

**Keywords:** Didactic system. Classic control. MATLAB. Arduino. Controller tuning.

## LISTA DE ILUSTRAÇÕES

Figura 1.1 - Regulador de esferas de Watt. ....	1
Figura 2.1 - Descrição de um sistema de controle.....	5
Figura 2.2 - Gráfico da função impulso.....	6
Figura 2.3 - Gráfico da função degrau.....	6
Figura 2.4 - Gráfico da função rampa. ....	7
Figura 2.5 - Gráfico da função parábola.....	7
Figura 2.6 - Gráfico da função senóide. ....	7
Figura 2.7 - Processo de troca de calor.....	8
Figura 2.8 - Processo de troca de calor em malha aberta. ....	9
Figura 2.9 - Diagrama de blocos de um sistema em malha aberta. ....	10
Figura 2.10 - Processo de troca de calor utilizando controle manual.....	11
Figura 2.11 - Diagrama de blocos de um sistema em malha fechada. ....	11
Figura 2.12 - Processo de troca de calor utilizando controle automático.....	12
Figura 2.13 - a. Sistema linear; b. Sistema não-linear.....	13
Figura 2.14 - Sistema linear invariante no tempo com uma entrada e uma saída. ....	14
Figura 2.15 - Sistema de malha fechada.....	15
Figura 2.16 - Curva de resposta ao degrau unitário.....	19
Figura 2.17 - Parâmetros de desempenho de um sistema de segunda ordem.....	19
Figura 2.18 - Diagrama de blocos de um controlador PID.....	23
Figura 2.19 - Resposta ao degrau unitário de uma planta. ....	24
Figura 2.20 - Curva da resposta ao degrau com aspecto de “S”.....	24
Figura 2.21 - Oscilação com período $P_{cr}$ . ....	26
Figura 2.22 - Representação de Duty-Cycle para PWM. ....	29
Figura 3.1 - Representação do circuito elétrico da planta. ....	30
Figura 3.2 - Considerações iniciais para o nó 1.....	30

Figura 3.3 - Conteúdo do arquivo do support package.....	33
Figura 3.4 - Instalação do script. ....	34
Figura 3.5 - Configuração do tipo do Arduino e porta. ....	34
Figura 3.6 - Comando do MATLAB para iniciar a conexão.....	35
Figura 3.7 - Mensagem de conexão bem sucedida. ....	35
Figura 3.8 - Representação do circuito elétrico após alteração. ....	36
Figura 3.9 - Trilhas do circuito gerado em software. ....	36
Figura 3.10 - Representação gráfica da placa confeccionada.....	37
Figura 3.11 - Posição dos componentes na placa. ....	37
Figura 3.12 - Interface gráfica desenvolvida em MATLAB. ....	38
Figura 3.13 - Montagem física do sistema didático. ....	40
Figura 4.1 - Resposta do sistema, ensaio em malha aberta com entrada em degrau unitário. .	42
Figura 4.2 - Desempenho da planta. ....	44
Figura 4.3 - Ação de controle proporcional com ganho ( $K_p$ ) em 1.....	45
Figura 4.4 - Ação de controle proporcional com ganho ( $K_p$ ) em 5.....	45
Figura 4.5 - Ação de controle proporcional com ganho ( $K_p$ ) em 10.....	45
Figura 4.6 - Ação de controle proporcional-integral com ganho ( $K_i$ ) em 0,2.....	46
Figura 4.7 - Ação de controle proporcional-integral com ganho ( $K_i$ ) em 0,4.....	47
Figura 4.8 - Ação de controle proporcional-integral com ganho ( $K_i$ ) em 1.....	47
Figura 4.9 - Ação de controle proporcional-derivativo com ganho ( $K_d$ ) em 1. ....	48
Figura 4.10 - Ação de controle proporcional-derivativo com ganho ( $K_d$ ) em 10. ....	48
Figura 4.11 - Ação de controle proporcional-derivativo com ganho ( $K_d$ ) em 20. ....	48
Figura 4.12 - Ação PID. ....	49
Figura 4.13 - Controle PID pela sintonia de Ziegler-Nichols, primeiro método.....	50
Figura 4.14 - Resposta do sistema para o ganho ( $K_p$ ) crítico. ....	51
Figura 4.15 - Controle PID pela sintonia de Ziegler-Nichols, segundo método. ....	51

## **LISTA DE TABELAS E QUADROS**

Tabela 2.1 - Valores de $K_p$ , $T_i$ e $T_d$ baseado na primeira regra de Ziegler Nichols.....	25
Tabela 2.2 - Valores de $K_p$ , $T_i$ e $T_d$ baseado na segunda regra de Ziegler Nichols. ....	26
Tabela 3.1 - Teste de desempenho do loop infinito do algoritmo do MATLAB. ....	39

## LISTA DE ABREVIATURAS E SIGLAS

A/D – Analógico para digital

GND – *Ground*

GUIDE – *Graphical user interface development environment*

IDE – *Integrated development environment*

LED – *Light emitting diode*

MIMO – *Multiple input, multiple output*

MV – *Manipulated variable*

PID – Proporcional-Integral-Derivativo

PV – *Process variable*

PWM – *Pulse width modulation*

SISO – *Single input, single output*

SP – *Set Point*

# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>1</b>
1.1	Uma breve história da automação	1
1.2	Apresentação	2
1.3	Estrutura do trabalho	4
<b>2</b>	<b>REVISÃO BIBLIOGRAFICA</b>	<b>5</b>
2.1	Definições prévias	5
2.1.1	Sistemas de controle	5
2.1.2	Planta	8
2.1.3	Variáveis de processo (PV) e Variável manipulada (MV)	8
2.1.4	<i>Set point</i> (SP)	9
2.1.5	Sistemas em malha aberta	9
2.1.6	Sistemas em malha fechada	10
2.1.7	Distúrbio	11
2.1.8	Controlador automático	12
2.2	Controle clássico	12
2.2.1	Conceito	12
2.2.2	Sistemas dinâmicos	13
2.2.3	Função de transferência de sistemas lineares	14
2.2.4	Análise da resposta em um sistema de segunda ordem	16
2.2.5	Estabilidade	21
2.2.6	Controlador PID	21
2.3	Conversão A/D	27
2.4	Modulação por largura de pulso	28
<b>3</b>	<b>METODOLOGIA</b>	<b>30</b>
3.1	Modelamento do circuito elétrico	30
3.2	Comunicação arduino e matlab	33
3.3	Confecção de placa de circuito impresso	35
3.4	Interface gráfica	38
3.4.1	Determinando tempo constante de aquisição	38
3.5	Montagem	40
<b>4</b>	<b>RESULTADOS E DISCUSSÃO</b>	<b>41</b>

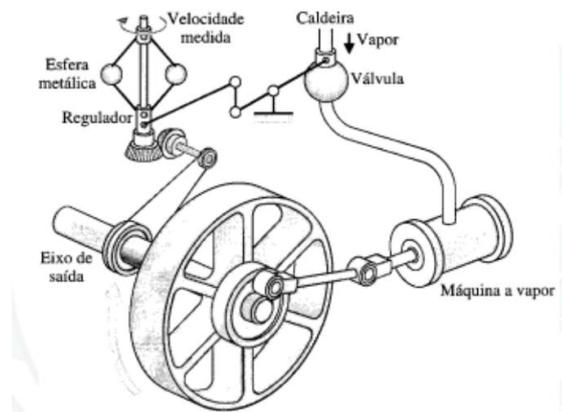
4.1	Função de transferência .....	41
4.2	Características do sistema.....	41
4.2.1	Coeficiente de amortecimento .....	41
4.2.2	Estabilidade .....	42
4.3	Desempenho do sistema .....	43
4.4	Aplicação de controle clássico .....	44
4.4.1	Ação de controle proporcional .....	44
4.4.2	Ação de controle proporcional-integral.....	46
4.4.3	Ação de controle proporcional-derivativo.....	47
4.4.4	Ação de controle proporcional-integral-derivativo .....	49
4.5	Sintonia Ziegler-Nichols .....	49
4.5.1	Primeiro método .....	49
4.5.2	Segundo método .....	50
<b>5</b>	<b>CONCLUSÕES.....</b>	<b>53</b>
	<b>REFERÊNCIAS .....</b>	<b>53</b>
	<b>ANEXO.....</b>	<b>56</b>

# 1 INTRODUÇÃO

A proposta do presente trabalho é realizar a montagem de um sistema didático de controle que se utiliza de um circuito elétrico como sistema ou planta, de um microcontrolador para aquisição de dados do sistema e de um *software* para executar cálculos de controle. O usuário terá uma interface gráfica para operar, visualizar gráficos e gerar comandos a fim de aplicar técnicas de controle clássico, através de um controlador PID (proporcional-integral-derivativo).

## 1.1 Uma breve história da automação

As primeiras aplicações que se tem conhecimento de sistemas de controle com o uso de retroação surgiu na Grécia no período de 300 a 1 a.C. Conhecido como relógio d'água de Ktesibios, este equipamento operava por gotejamento de água para um reservatório medidor, e o nível de água do reservatório podia ser utilizado para informar o tempo decorrido (NISE, p.4, 2011). No início do século XVII, antes da primeira Revolução Industrial, foram introduzidos os primeiros tipos de máquinas energéticas complexas, ou seja, que transformavam um tipo de energia em outras. Papin, inventou a válvula da panela de pressão e apresentou a primeira máquina a vapor, munida de pistão (AGUIRRE et al., 2007, p. 29).



**Figura 1.1 - Regulador de esferas de Watt.**

**Fonte: DORF (p.3, 2001).**

Com aplicação industrial, é aceito como o primeiro controlador automático com retroação o regulador de esferas de James Watt, desenvolvido em 1769, era utilizado para controlar a velocidade das máquinas a vapor. O mecanismo mensurava a velocidade do eixo de saída e utilizava a movimentação das esferas para controlar a entrada de vapor na máquina. Conforme aumentava a rotação do eixo de saída, os pesos em forma de esferas se elevavam e com esse

movimento realizavam a desacoplagem do eixo, conseqüentemente, fechando a válvula, conforme figura 1.1 (DORF, p.4, 2001).

A teoria dos sistemas de controle como é conhecida hoje, começou a ser introduzida na segunda metade do século XIX. James Clerk Maxwell publicou um estudo do critério de estabilidade para um sistema de terceira ordem, em 1868. Já em 1874, Edward John Routh acrescentou à obra de Willian Kingdon Clifford o critério de estabilidade para os sistemas de quinta ordem. Alexandr Michailovich Lyapunov também fez sua contribuição para a formulação e desenvolvimento das teorias e práticas de estabilidade do sistema de controle atualmente (NISE, p.4-5, 2011).

Sistemas de controle industrial das décadas de 40 e 50 utilizavam, em muitos casos, controladores PID no controle de temperatura, pressão, nível etc. Ziegler e Nichols, no início da década de 40 produziram regras de ajuste de controladores PID, conhecido por método de Ziegler-Nichols (OGATA, p.2, 2010). Durante as décadas de 60 e 70 houve uma grande evolução em controle de sistemas através da introdução da tecnologia eletrônica, provocado pelo desenvolvimento de circuitos de controle eletrônicos e chaveamentos sem contato físico. Ainda no final da década de 60, a utilização dos circuitos integrados permitiu a elaboração dos minicomputadores, sua aplicação em ambientes industriais geraram profundas mudanças (MIYAGI, p.3, 1996).

Atualmente os sistemas de controle automático estão amplamente difundidos nas indústrias controlando seus processos, garantindo maior produção e padronização das técnicas. Encontra-se também sistemas de controle auxiliando na orientação, navegação e controle de mísseis e veículos espaciais, assim como aviões e navios. No exemplo de naves espaciais, seu sistema é dotado de inúmeros sistemas de controle assistidos por computador com base de tempo compartilhada. Sem estes recursos, seria impossível manter ou ajustar a nave em órbita terrestre ou manter a vida humana em seu interior (NISE, p.5, 2011).

## **1.2 Apresentação**

A utilização de Sistemas de Controle, que podem ser entendidos como um conjunto de dispositivos que comandam o comportamento de outros dispositivos, é muito abrangente e, por isso, o desenvolvimento de novos métodos e aplicações ainda torna-se necessário. Tencionando contribuir com tal desenvolvimento, neste projeto foram aplicadas técnicas de

controle e sintonia de controladores para analisar o comportamento dinâmico de sistemas sujeitos a tais procedimentos.

Construiu-se um sistema didático que, caso provar-se eficaz, poderá servir como uma plataforma de testes e demonstração de algumas das técnicas de controle que poderão ser simuladas através dele. O sistema didático proposto, entende-se como uma ferramenta que auxilia na compreensão de algum estudo, o qual possui a finalidade de instruir, neste caso, por meio de testes que será possível executar. Este sistema didático é denominado como objeto de aprendizagem, de acordo com BRAGA, et al. (p. 22-25, 2014) podem ser considerados objetos de aprendizagem quaisquer recursos como imagens, vídeos, softwares e simulações que sejam utilizados para o aprendizado. O mesmo é constituído de um microcontrolador Arduino Mega que realiza aquisição dos dados de saída, ou seja, a leitura do valor medido da tensão de saída de uma planta que, neste caso, é representado por uma combinação de um circuito elétrico; e, por último, de um computador que executa o *software* MATLAB que, por sua vez, aplica as ações de controle para tentar controlar esse sistema físico e, até mesmo, gera gráficos em tempo real.

Alguns artigos e pesquisas foram produzidos em relação à utilização de plataformas didáticas utilizando Arduino e MATLAB, dentre eles pode-se citar os estudos de AL-BUSAIDI (2012), que desenvolveu um ambiente didático para controlar um robô bípede baseado em Arduino e MATLAB. Esse trabalho aborda uma alternativa de baixo custo para a produção de um robô bípede. O MATLAB foi usado para o controle e a visualização do processo em questão, e o Arduino teve como uma de suas funções aplicar o controle final nos servos atuadores e adquirir de dados dos sensores que estão instalados na base do “pé” do robô. SILVA e GOMES (2007) desenvolveram um ambiente digital de simulação para projeto de compensadores por lugar das raízes, utilizando também a ferramenta computacional MATLAB com uma interface gráfica amigável que, segundo os autores, o trabalho complementa tópicos de projeto de controladores industriais muitas vezes abordados pela faculdade de forma não totalmente satisfatória.

Um ponto positivo da abordagem proposta é que, ao término desta pesquisa, todo o conhecimento adquirido e a montagem futuramente efetuada poderá ser utilizada para a implementação de uma planta de sistema didático nos laboratórios do Centro Federal de Educação Tecnológica de Minas Gerais – CEFET-MG – Araxá. Sendo assim, este trabalho produzirá mais um equipamento que pode servir de apoio didático tanto para as aulas da disciplina de Sistemas de Controle de Processos Contínuos, como para quem desenvolve pesquisas que demandam tais instrumentos. Nessa medida, essa proposta de estudo garante a

aplicação de todo o conhecimento adquirido ao longo do curso de Engenharia de Automação Industrial de modo prático, que ajudará na compreensão das técnicas de controle utilizadas.

Diante do exposto, aventa-se a seguinte hipótese de pesquisa: se implementado um sistema didático, que permita a aplicação de técnicas de controle e sintonia de controladores de forma eficaz, então se pode chegar à conclusão de que é possível implementar um sistema didático que possa ser uma ferramenta eficaz no aprendizado das teorias aplicadas ao controle em automação, auxiliando assim, o entendimento sobre as técnicas de controle e sintonia de controladores que são empregadas nas mais diversas aplicações num ambiente industrial.

### **1.3 Estrutura do trabalho**

Capítulo 2 – Revisão Bibliográfica – Este capítulo expõe, de forma sucinta, um estudo da teoria de controle relacionada com o tema proposto. É realizada a introdução de conceitos prévios de controle e a abordagem dos conceitos mais importantes referentes ao controle clássico, com enfoque em estabilidade de sistemas, função de transferência, controlador PID e regras de sintonia de Ziegler-Nichols.

Capítulo 3 – Metodologia – Este capítulo demonstra as técnicas e métodos empregados no decorrer do desenvolvimento do trabalho, apresentando o modelamento matemático do sistema proposto, a elaboração da interface gráfica e a confecção do circuito elétrico.

Capítulo 4 – Resultados e discussões – Neste capítulo são apresentados os resultados de diversos testes de aplicação de controle e comparados com o resultado esperado, em referência às bibliografias.

Capítulo 5 – Conclusão – Por fim, este capítulo faz uma crítica dos resultados obtidos a partir de testes realizados, demonstrados no Capítulo 4, e sugere propostas de desenvolvimentos de trabalhos futuros, baseado no que foi exposto neste trabalho de conclusão de curso.

## 2 REVISÃO BIBLIOGRAFICA

### 2.1 Definições prévias

#### 2.1.1 Sistemas de controle

É um sistema composto de subsistemas e processos (plantas) a fim de controlar as saídas dos processos. Um sistema de controle apresenta uma saída ou resposta em função de uma entrada ou estímulo (NISE, p.2, 2011).



**Figura 2.1 - Descrição de um sistema de controle.**

**Fonte: Própria, baseado em NISE (p.2, 2011).**

Como observado na figura 2.1, um sistema de controle fornece uma saída para uma dada entrada. A entrada pode ser entendida como uma resposta desejada e a saída como uma resposta real (NISE, p.2, 2011).

Ainda conforme (NISE, p.2, 2011):

[...] Utilizando os sistemas de controle, podemos movimentar grandes equipamentos com uma precisão que, de outra forma, não seria possível. Podemos apontar enormes antenas em direção às maiores distâncias do universo para capturar sinais de rádio de baixa intensidade; o que, manualmente, seria impossível.

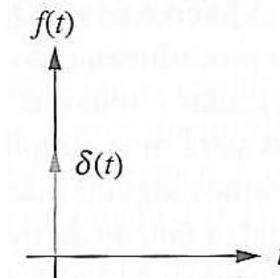
E isso deixa claro a vantagem de se fazer uso de sistemas de controle e sistemas automatizados em meio industrial, pois há inúmeras tarefas que não podem ser realizadas sem o auxílio destes recursos.

##### 2.1.1.1 Tipos de entradas

Sinais de entrada são utilizados para analisar o comportamento do sistema e, comumente, utiliza-se sinais menos complexos como entrada em impulso, degrau, rampa, parábola, senóides etc.

O impulso é um tipo de entrada que possui valor infinito em  $t = 0$ , e zero para  $t > 0$ . Esta entrada fornece uma energia inicial para o sistema, de forma que a resposta relacionada àquele impulso seja a resposta transiente do sistema. A função impulso é amplamente utilizada para deduzir a modelagem do sistema (NISE, p.17, 2011).

$$\delta(t) = \begin{cases} \infty, & t = 0 \\ 0, & \text{caso contrário} \end{cases} \quad 2.1$$

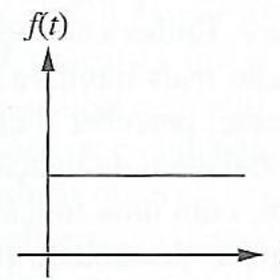


**Figura 2.2 - Gráfico da função impulso.**

**Fonte: NISE (p.16, 2011).**

A função degrau é caracterizada por um comando constante, o comando de entrada em degrau possui, normalmente a mesma forma da saída. A entrada em degrau é amplamente utilizada, pois, as respostas transiente e em regime estacionário são observadas com facilidade (NISE, p.17, 2011).

$$u(t) = \begin{cases} 1, & t > 0 \\ 0, & t < 0 \end{cases} \quad 2.2$$

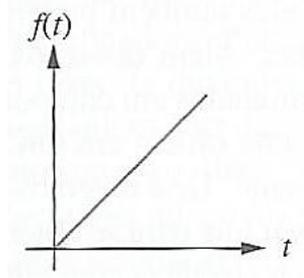


**Figura 2.3 - Gráfico da função degrau.**

**Fonte: NISE (p.16, 2011).**

A entrada da função rampa apresenta um aspecto de um comando linearmente crescente. São utilizadas para analisar o erro do regime estacionário dos sistemas (NISE, p.17, 2011).

$$tu(t) = \begin{cases} t, & t \geq 0 \\ 0, & \text{caso contrário} \end{cases} \quad 2.3$$

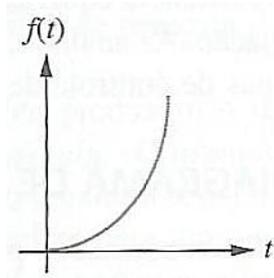


**Figura 2.4 - Gráfico da função rampa.**

**Fonte: NISE (p.16, 2011).**

A entrada da função parábola apresenta a mesma característica da entrada rampa e são utilizadas também para analisar o erro do regime estacionário dos sistemas (NISE, p.17, 2011).

$$\frac{1}{2}t^2u(t) = \begin{cases} \frac{1}{2}t^2, & t \geq 0 \\ 0, & \text{caso contrário} \end{cases} \quad 2.4$$

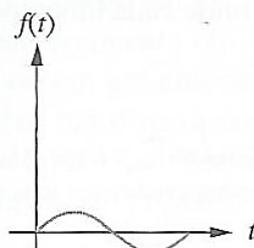


**Figura 2.5 - Gráfico da função parábola.**

**Fonte: NISE (p.16, 2011).**

A entrada da função senóide é utilizadas para testar sistemas físicos com finalidade de obter a sua função de transferência (NISE, p.17, 2011).

$$u(t) = \begin{cases} \sin \omega t, & t \geq 0 \\ 0, & \text{caso contrário} \end{cases} \quad 2.5$$



**Figura 2.6 - Gráfico da função senóide.**

**Fonte: NISE (p.16, 2011).**

### 2.1.2 Planta

Um sistema físico a ser controlado é constituído por um equipamento ou um conjunto de componentes de um equipamento que opere de forma integrada, com o objetivo de realizar determinada operação (OGATA, p.2, 2010).

Um exemplo de planta ou processo que pode ser citado, muito comum, é o trocador de calor. Sua função é aquecer um fluido qualquer utilizando vapor. Para que o trocador de calor opere satisfatoriamente, deve-se considerar uma série de fatores, que são: Vazão e temperatura do fluido que será aquecido, vazão do gás que fornecerá calor ao fluido, capacidade calorífica de ambos os fluidos, perdas térmicas etc. (BEGA, p.1, 2006). O processo do trocador de calor pode ser visto na figura 2.7.

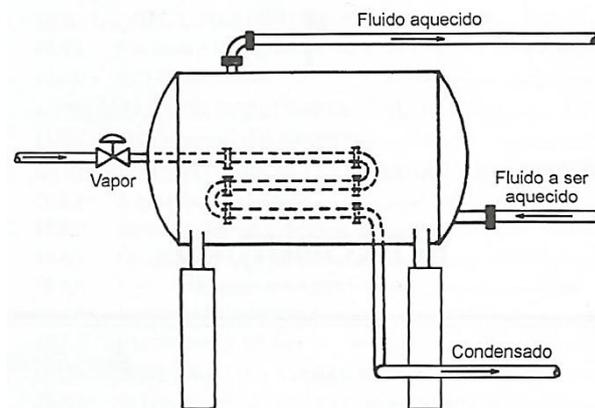


Figura 2.7 - Processo de troca de calor.

Fonte: BEGA (p.2, 2006).

### 2.1.3 Variáveis de processo (PV) e Variável manipulada (MV)

Processo pode ser entendido como toda operação a ser controlada. Já as variáveis de processo são todas aquelas grandezas físicas que fazem parte do processo. Estas variáveis podem ser: nível, temperatura, pressão, vazão, pH, tensão, corrente elétrica etc. O processo pode ser controlado medindo estas grandezas físicas e aplicando-se ações para alterá-las conforme desejado. Variáveis de processo também podem ser chamadas de variáveis controladas.

Segundo (OGATA, p.2, 2010):

[...] Normalmente, a variável controlada é a saída do sistema.

A variável manipulada é uma grandeza constantemente modificada pelo controlador, visando afetar o valor da variável controlada. (OGATA, p.2, 2010).

Ou seja, o controlador atua diretamente aplicando ações de controle na variável manipulada de modo que a variável de processo alcance e a mantenha no valor desejado.

#### 2.1.4 *Set point* (SP)

O *set point*, ou valor desejado, é um valor previamente estipulado e que se torna um referencial para que o sistema tente, através de ações de controle, alcançar e manter ao longo do tempo.

#### 2.1.5 Sistemas em malha aberta

Sistemas de controle em malha aberta são sistemas que o sinal de saída não exerce ação de controle no sistema, ou seja, não há realimentação. Isso quer dizer que o sinal de saída não é usado para ser comparado com a entrada, assim sendo, a precisão desse sistema depende de uma calibração (OGATA, p.7, 2010).

De acordo com (OGATA, p.7, 2010), podemos citar um exemplo prático de sistema em malha aberta que:

[...] é o da máquina de lavar roupas. As operações de colocar de molho, lavar e enxaguar em uma lavadora são executadas em uma sequência em função do tempo. A lavadora não mede o sinal de saída, isto é, não verifica se as roupas estão bem lavadas.

Seguindo o exemplo do trocador de calor citado acima, pode-se estender os conceitos de sistema em malha aberta aplicados nele. No caso da figura 2.8, tem-se a informação da temperatura do fluido na saída do trocador, que no caso é a variável controlada, não será utilizada como referência para ajustar as variáveis de entrada.

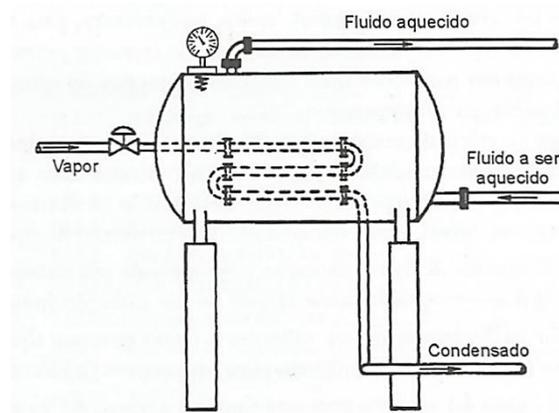
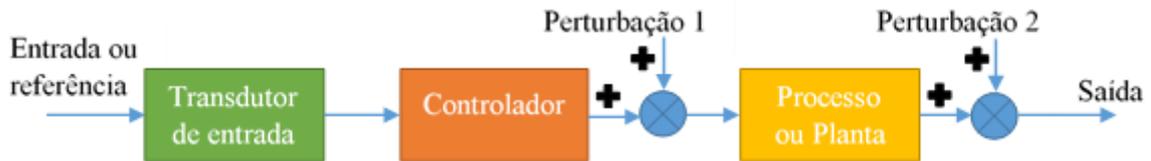


Figura 2.8 - Processo de troca de calor em malha aberta.

Fonte: BEGA (p.2, 2006).

Se nesta configuração de sistema houver perturbações, ou distúrbios, conforme a figura 2.9 representa, nenhuma ação vai ser tomada para que se corrija este problema. Qualquer ruído adicionado tanto na Perturbação 1 quanto na Perturbação 2, serão adicionados ao processo (NISE, p.7, 2011).



**Figura 2.9 - Diagrama de blocos de um sistema em malha aberta.**

Fonte: Própria, baseado em NISE (p.7, 2011).

### 2.1.6 Sistemas em malha fechada

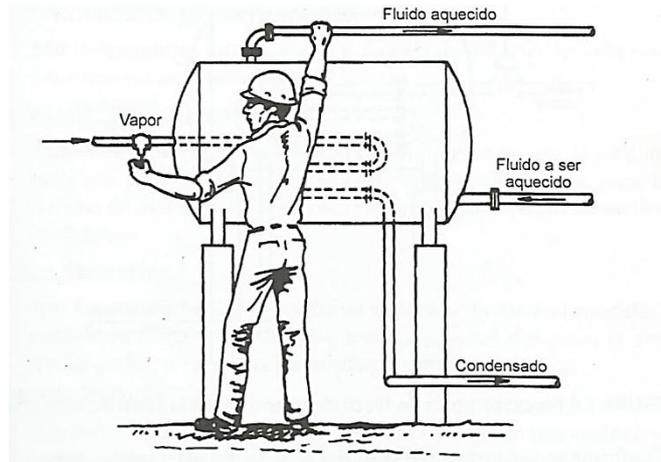
Em um sistema em malha fechada o sinal de erro realimenta o controlador, para minimizar o erro e ajustar o valor de saída de acordo com o valor desejado. Nesse tipo de sistema, são utilizados sensores, que medem a resposta da saída e, em seguida, convertem o sinal para a forma utilizada pelo controlador (NISE, p.8, 2011).

O erro, basicamente, é a diferença do valor desejado (referência) com o valor real em que o sistema se encontra, medido na saída do sistema. Esse erro é tratado pelo controlador que fará a ação de controle para atenuá-lo.

Conforme (BEGA, p.3, 2006):

[...] O controle manual em malha fechada pode ser realizado por um operador humano (controle manual) ou mediante a utilização de instrumentação (controle automático).

No exemplo mostrado na figura 2.10, o operador medirá a temperatura do fluido aquecido (variável controlada) e irá atuar diretamente alterando a vazão de vapor (variável manipulada) que entra no sistema. Ou seja, o operador estará medindo a temperatura através do tato e por meio de comparação a um valor em que a temperatura deveria estar. Daí, ele aplica correções até que a temperatura do fluido aquecido se encontre no valor desejado. Neste caso, seu tato é o sensor, e seu cérebro é o controlador.

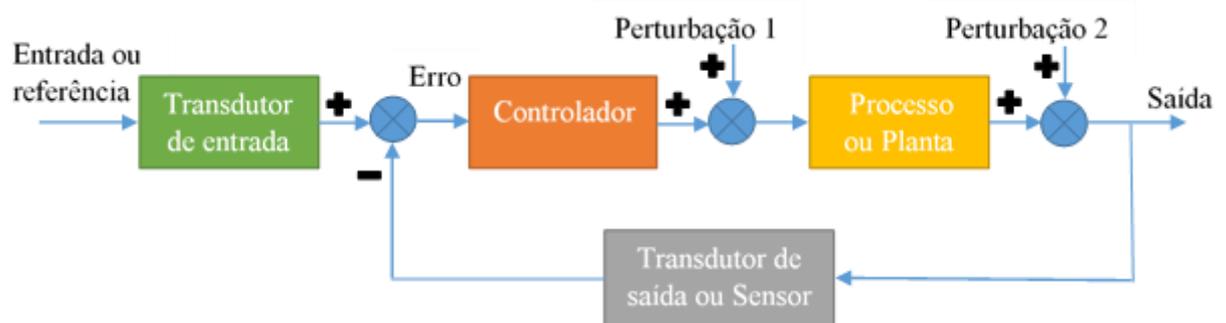


**Figura 2.10 - Processo de troca de calor utilizando controle manual.**

Fonte: BEGA (p.3, 2006).

Concluindo este assunto como mencionado em (NISE, p.8, 2011):

[...] Assim, os sistemas de malha fechada possuem a vantagem óbvia de apresentar uma acurácia superior à dos sistemas de malha aberta. Eles são menos sensíveis a ruídos, perturbações e alterações do ambiente.



**Figura 2.11 - Diagrama de blocos de um sistema em malha fechada.**

Fonte: Própria, baseado em NISE (p.7, 2011).

### 2.1.7 Distúrbio

Distúrbio é um sinal, uma perturbação que afeta diretamente o valor de saída e tende levar o sistema à instabilidade. Podendo ser de natureza interna, quando gerado dentro do sistema, e externa, quando gerado fora do sistema. Quando o distúrbio for de natureza externa, ele se comporta como um sinal de entrada do sistema (OGATA, p.3, 2010).

### 2.1.8 Controlador automático

Controlador Automático é um equipamento que realiza tomada de decisões em um sistema. Ele trabalha comparando o valor real da resposta da planta com a entrada de referência (valor desejado). Posteriormente, determina o desvio gerado pela diferença desses valores de entrada e saída e produz o sinal de controle que irá reduzir ou levar a zero esse desvio (OGATA, p.18, 2010). Um microcontrolador, como o da família PIC da empresa MICROCHIP, ou mesmo um Arduino, por exemplo, podem ser utilizados para realizar operações de controle de sistemas, quando previamente programados para tal.

Na figura 2.12, continuando o exemplo do trocador de calor, temos um sistema em malha fechada com controle automático, a diferença agora é que a medição da temperatura é feita pelo transmissor de temperatura (TT), a comparação do sinal medido e do valor de *set point* é realizado no controlador de temperatura (TRC), e o mecanismo que atua operando a válvula que aumenta ou diminui a vazão de vapor é a válvula de controle (TY) (BEGA, p.4, 2006):

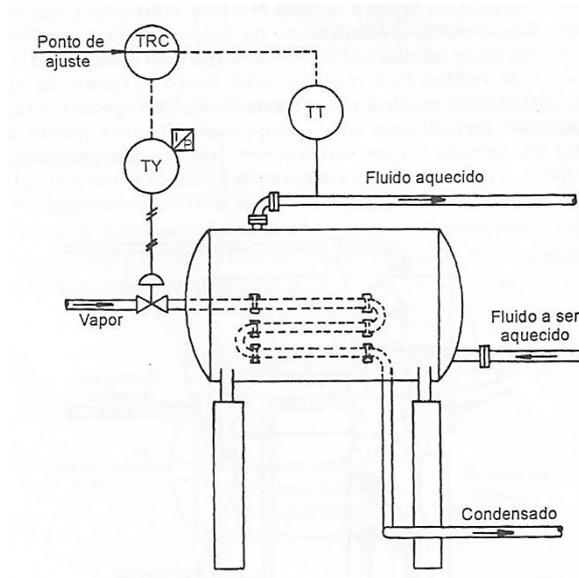


Figura 2.12 - Processo de troca de calor utilizando controle automático.

Fonte: BEGA (p.4, 2006).

## 2.2 Controle clássico

### 2.2.1 Conceito

O controle clássico é um método de controle que trata da resposta em frequência. Esse controle é baseado em combinações de métodos analíticos, transformadas de Laplace etc. Esse controle é utilizado comumente em plantas lineares, invariantes no tempo, com uma entrada e uma saída, do inglês, SISO (*single- input, single-output*). Para sistemas de múltiplas

entradas e saídas, do inglês, MIMO (*multiple-input, multiple-output*) torna-se uma tarefa bastante trabalhosa controlar o sistema pelo fato de ser muito mais difícil utilizar os conhecimentos intuitivos em um modelo complexo (AGUIRRE et al., p.86, 2007).

## 2.2.2 Sistemas dinâmicos

O conceito de sistemas dinâmicos, definido por MONTEIRO (p.41, 2006), são sistemas que possuem elementos variantes no tempo. Em MAYA (p.4, 2014), o autor mostra que os sistemas dinâmicos são sistemas que possuem memória e o seu comportamento depende de suas condições iniciais, além de serem opostos aos sistemas estáticos, que respondem instantaneamente à excitação aplicada.

### 2.2.2.1 Características dos sistemas dinâmicos

#### a. Linearidade

Sistemas lineares, são sistemas que satisfazem as condições do princípio da homogeneidade e da superposição (MAYA, p.4, 2014).

A propriedade inerente de homogeneidade refere a resposta do sistema à multiplicação da entrada por um escalar. Sendo assim, para uma entrada  $u(t)$  tem-se uma saída  $y(t)$ , se a mesma entrada agora for multiplicada pelo escalar  $A$ ,  $Au(t)$ , para sistemas lineares ter-se-á a saída  $Ay(t)$ . Isto é, a multiplicação da entrada por um escalar fornece a saída multiplicada pelo mesmo escalar (NISE, p.71, 2011).

A propriedade da superposição descreve que a soma das entradas de um sistema é igual a soma das saídas. De forma análoga, pode-se exemplificar que dado uma saída  $y_1(t)$  para uma respectiva entrada  $u_1(t)$  e uma saída  $y_2(t)$  para uma respectiva entrada  $u_2(t)$ , então para um sinal de entrada  $u_1(t) + u_2(t)$  deve-se obter a saída  $y_1(t) + y_2(t)$  (NISE, p.71, 2011).

A linearidade do sistema pode ser vista na representação da figura 2.13.

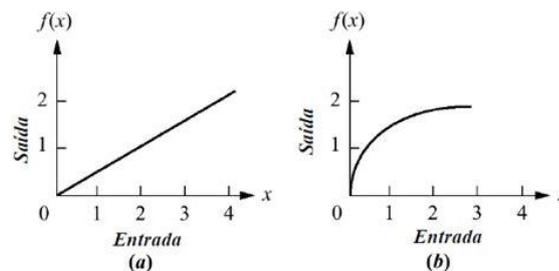


Figura 2.13 - a. Sistema linear; b. Sistema não-linear.

Fonte: NISE (p. 72, 2011).

b. Variância e invariância

Sistemas lineares invariantes no tempo são sistemas dinâmicos com parâmetros invariantes no tempo, os quais podem ser descritos por equações diferenciais lineares invariantes no tempo, ou seja, coeficientes constantes. Sistemas caracterizados por equações diferenciais com coeficientes variantes no tempo, por sua vez, são denominados sistemas lineares variantes no tempo (OGATA, p.12, 2010).

c. Quantidade de entradas e saídas

Uma outra característica que define um sistema é a quantidade de entrada e saída que possui, podendo ser caracterizado em monovariável ou multivariável. Em sistemas que possuem uma entrada e uma saída são sistemas monovariáveis, ou SISO. Sistemas que possuem duas ou mais entradas ou duas ou mais saídas são sistemas multivariáveis, ou MIMO (MAYA, p.2 e4, 2014).

### 2.2.3 Função de transferência de sistemas lineares

Segundo OGATA (p.12, 2010), a função de transferência caracteriza as relações de entrada e saída de componentes ou de sistemas.

Considerando um sistema linear, invariante no tempo, de apenas uma entrada e uma saída, com condições iniciais nulas conforme representação da figura 2.14 tem-se:



**Figura 2.14 - Sistema linear invariante no tempo com uma entrada e uma saída.**

Fonte: Própria, baseado em MAYA (p.45, 2014).

$$\begin{aligned} q_n y(t)^{(n)} + q_{n-1} y(t)^{(n-1)} + \dots + q_2 \ddot{y}(t) + q_1 \dot{y}(t) + q_0 y(t) \\ = p_m u(t)^{(m)} + p_{m-1} u(t)^{(m-1)} + \dots + p_1 \dot{u}(t) + p_0 u(t) \end{aligned} \quad 2.6$$

Onde:

$$y(t)^{(n)} = \frac{d^n}{dt^n} y(t), \quad y(t)^{(n-1)} = \frac{d^{n-1}}{dt^{n-1}} y(t), \dots, \quad \dot{y}(t) = \frac{d}{dt} y(t) \quad 2.7$$

$$u(t)^{(m)} = \frac{d^m}{dt^m} u(t), \quad u(t)^{(m-1)} = \frac{d^{m-1}}{dt^{m-1}} u(t), \dots, \quad \dot{u}(t) = \frac{d}{dt} u(t) \quad 2.8$$

Os coeficientes  $q$  e  $p$  são constantes, variando de zero à  $n$  e  $m$ , respectivamente. Sendo que  $n > m$  e o valor de  $n$  indica a ordem da equação, e conseqüentemente, a ordem do sistema.

Aplicando-se a transformada de Laplace na equação diferencial ordinária, equação 2.6, com condições iniciais nulas, obtém-se a equação 2.9:

$$\begin{aligned} (q_n s^n + q_{n-1} s^{n-1} + \dots + q_2 s^2 + q_1 s + q_0) Y(s) \\ = (p_m s^m + p_{m-1} s^{m-1} + \dots + p_2 s^2 + p_1 s + p_0) U(s) \end{aligned} \quad 2.9$$

De forma análoga, com as equações 2.7 e 2.8, tem-se as equações 2.10 e 2.11, respectivamente:

$$Y(s) = \mathcal{L}[y(t)], \quad sY(s) = \mathcal{L}[\dot{y}(t)], \quad s^2 Y(s) = \mathcal{L}[\ddot{y}(t)], \quad \dots \quad 2.10$$

$$U(s) = \mathcal{L}[u(t)], \quad sU(s) = \mathcal{L}[\dot{u}(t)], \quad s^2 u(s) = \mathcal{L}[\ddot{u}(t)], \quad \dots \quad 2.11$$

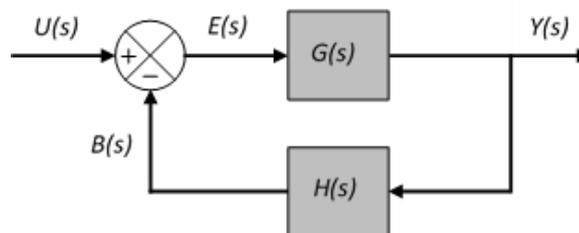
De acordo com MAYA (p.46, 2014):

“Denomina-se função de transferência  $G(s)$  de um sistema linear, de parâmetros concentrados, invariantes no tempo e de entrada e saída únicas a relação entre as transformadas de Laplace da variável de saída e da variável de entrada, supondo as condições iniciais nulas.”

Desta forma obtém-se a seguinte equação:

$$G(s) = \frac{\mathcal{L}[\text{saída}]}{\mathcal{L}[\text{entrada}]} = \frac{Y(s)}{U(s)} \quad 2.12$$

Em um sistema em malha fechada como mostra a figura 2.15, a saída  $Y(s)$  e a entrada  $U(s)$  se relaciona da seguinte forma:



**Figura 2.15 - Sistema de malha fechada.**

**Fonte: Própria, baseado em OGATA (p.16, 2010).**

$$Y(s) = G(s)E(s) \quad 2.13$$

Onde  $E(s)$  representa o erro atuante, e este erro pode ser expresso por:

$$E(s) = U(s) - B(s) \quad 2.14$$

Ou:

$$E(s) = U(s) - H(s)Y(s) \quad 2.15$$

Substituindo  $E(s)$ , encontrado na equação 2.15, na equação 2.13, tem-se:

$$Y(s) = G(s)[U(s) - H(s)Y(s)] \quad 2.16$$

Simplificando, encontra-se:

$$\frac{Y(s)}{U(s)} = \frac{G(s)}{1 + G(s)H(s)} \quad 2.17$$

Deve-se enfatizar que a aplicação da função de transferência se limita a sistemas de equações diferenciais lineares invariantes no tempo. A função de transferência é uma característica intrínseca do sistema, que independe da natureza da função de entrada ou de excitação. Caso a função de um sistema não for conhecida, ela pode ser determinada experimentalmente por meio da utilização de entradas conhecidas e da análise das respectivas respostas do sistema (OGATA, p.13, 2010).

#### 2.2.4 Análise da resposta em um sistema de segunda ordem

De acordo com DORF (p.35 e 183, 2001), considerando um sistema de segunda ordem, com retroação a malha fechada, sua saída pode ser demonstrada como:

$$Y(s) = \frac{G(s)}{1 + G(s)} U(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} U(s) \quad 2.18$$

Para uma entrada degrau unitário, temos:

$$Y(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad 2.19$$

Onde  $\zeta$  é a relação de amortecimento, e  $\omega_n$  é a frequência natural do sistema. As raízes desta equação característica são:

$$s_1, s_2 = -\zeta\omega_n \pm j\omega_n\sqrt{1 - \zeta^2} \quad 2.20$$

Analisando a equação 2.20, constata-se que quando  $\zeta > 1$ , as raízes são reais; quando  $\zeta < 1$ , as raízes são complexas conjugadas; quando  $\zeta = 1$ , as raízes são iguais e reais.

### 2.2.4.1 Sistema subamortecido

Em OGATA (p.151, 2010) mostra-se que num sistema subamortecido o coeficiente de amortecimento  $\zeta$  assume valores maiores que zero e menores que um, ou seja,  $0 < \zeta < 1$ . Sendo assim, a relação da função de transferência pode ser escrita como:

$$\frac{Y(s)}{U(s)} = \frac{\omega_n^2}{(s + \zeta\omega_n + j\omega_d)(s + \zeta\omega_n - j\omega_d)} \quad 2.21$$

Onde  $\omega_d = \omega_n\sqrt{1 - \zeta^2}$ . Esta frequência  $\omega_d$  é a frequência natural amortecida.

Para um degrau unitário, e aplicando-se a transformada de Laplace inversa, obtém-se a equação da saída do sistema, da seguinte forma:

$$Y(s) = \frac{1}{s} - \frac{s + \zeta\omega_n}{(s + \zeta\omega_n)^2 + \omega_d^2} - \frac{\zeta\omega_n}{(s + \zeta\omega_n)^2 + \omega_d^2} \quad 2.22$$

Em consulta da tabela de transformada de Laplace encontrada em OGATA (p.781, 2010) demonstra-se que:

$$\mathcal{L}^{-1}\left[\frac{s + \zeta\omega_n}{(s + \zeta\omega_n)^2 + \omega_d^2}\right] = e^{-\zeta\omega_n t} \cos \omega_d t \quad 2.23$$

$$\mathcal{L}^{-1}\left[\frac{\omega_d}{(s + \zeta\omega_n)^2 + \omega_d^2}\right] = e^{-\zeta\omega_n t} \sen \omega_d t \quad 2.24$$

De forma análoga, aplicando-se a transformada inversa de Laplace na equação de saída, tem-se:

$$\begin{aligned} \mathcal{L}^{-1}[Y(s)] = y(t) &= 1 - e^{-\zeta\omega_n t} \left( \cos \omega_d t + \frac{\zeta}{\sqrt{1 - \zeta^2}} \sen \omega_d t \right) \\ &= 1 - \frac{e^{-\zeta\omega_n t}}{\sqrt{1 - \zeta^2}} \sen \left( \omega_d t + \tan^{-1} \frac{\sqrt{1 - \zeta^2}}{\zeta} \right), \text{ para } t \geq 0 \end{aligned} \quad 2.25$$

Pode-se perceber que a resposta apresenta uma oscilação senoidal amortecida. Em regime permanente,  $t$  tendendo ao infinito ( $t \rightarrow \infty$ ), não existe erro entre a entrada e a saída (OGATA, p.152, 2010).

### 2.2.4.2 Sistema sem amortecimento

Para que o sistema não tenha amortecimento, o coeficiente de amortecimento  $\zeta$  tem que assumir um valor igual a zero. Sendo assim, a resposta do sistema não vai ser amortecida e as oscilações permanecerão constantes (OGATA, p.152, 2010).

Com amortecimento nulo, a resposta do sistema  $y(t)$ , tem-se a seguinte expressão substituindo  $\zeta = 0$  na equação:

$$y(t) = 1 - \cos \omega_n t, \text{ para } t \geq 0 \quad 2.26$$

### 2.2.4.3 Sistema criticamente amortecido

Quando os polos de uma função de transferência forem iguais, o sistema é denominado criticamente amortecido. Para que os polos de uma função sejam iguais, o valor de  $\zeta$  é igual a um. Como descrito em OGATA (p.152, 2010) a equação de saída do sistema, utilizando-se uma entrada em degrau unitário,  $U(s) = 1/s$ , é:

$$Y(s) = \frac{\omega_n^2}{(s + \omega_n)^2 s} \quad 2.27$$

Sua transformada inversa de Laplace, pode ser apresentada como:

$$y(t) = 1 - e^{-\omega_n t} (1 + \omega_n t), \text{ para } t \geq 0 \quad 2.28$$

### 2.2.4.4 Sistema superamortecido

No caso de  $\zeta > 1$ , tem-se que os polos de  $Y(s)/U(s)$  são reais, negativos e desiguais. Em uma entrada de degrau unitário, a saída  $Y(s)$  pode ser representada como (OGATA, p.152, 2010):

$$Y(s) = \frac{\omega_n^2}{(s + \zeta\omega_n + \omega_n\sqrt{\zeta^2 - 1})(s + \zeta\omega_n - \omega_n\sqrt{\zeta^2 - 1})s} \quad 2.29$$

Sua transformada inversa de Laplace é:

$$y(t) = 1 + \frac{1}{2\sqrt{\zeta^2 - 1}(\zeta + \sqrt{\zeta^2 - 1})} e^{-(\zeta + \sqrt{\zeta^2 - 1})\omega_n t} - \frac{1}{2\sqrt{\zeta^2 - 1}(\zeta - \sqrt{\zeta^2 - 1})} e^{-(\zeta - \sqrt{\zeta^2 - 1})\omega_n t} \quad 2.30$$

Simplificando, obtém-se:

$$y(t) = 1 + \frac{\omega_n}{2\sqrt{\zeta^2 - 1}} \left( \frac{e^{-s_1 t}}{s_1} - \frac{e^{-s_2 t}}{s_2} \right), \text{ para } t \geq 0 \quad 2.31$$

Onde  $s_1 = (\zeta + \sqrt{\zeta^2 - 1})\omega_n$  e  $s_2 = (\zeta - \sqrt{\zeta^2 - 1})\omega_n$ . Sendo os dois termos exponenciais decrescentes.

Na figura 2.16 tem-se que a variação da resposta do sinal de saída  $y(t)$  ao longo do tempo, em função de diferentes valores de coeficientes de amortecimento  $\zeta$ .

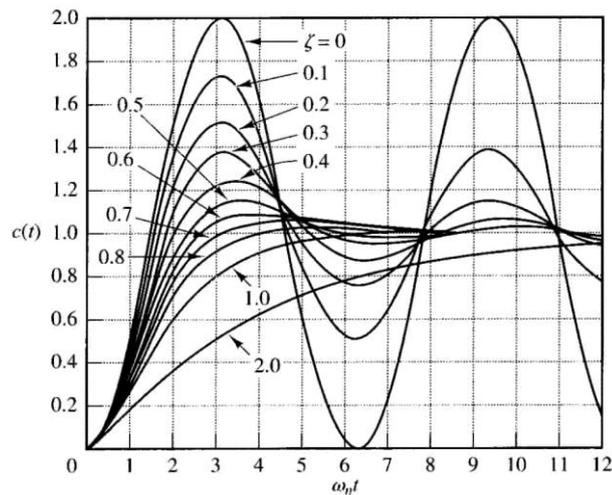


Figura 2.16 - Curva de resposta ao degrau unitário.

Fonte: OGATA (p.154, 2010).

#### 2.2.4.5 Desempenho de um sistema

De acordo com AGUIRRE (p.48, 2007) os índices de desempenho de um sistema são utilizados, normalmente, para determinar a qualidade da resposta. Em OGATA (p.154, 2010), o autor define que na prática, a resposta transitória de um sistema apresenta oscilações amortecidas antes de atingir o regime permanente. De acordo com a característica da curva da resposta no tempo, são obtidos os índices de desempenho, conforme figura 2.17.

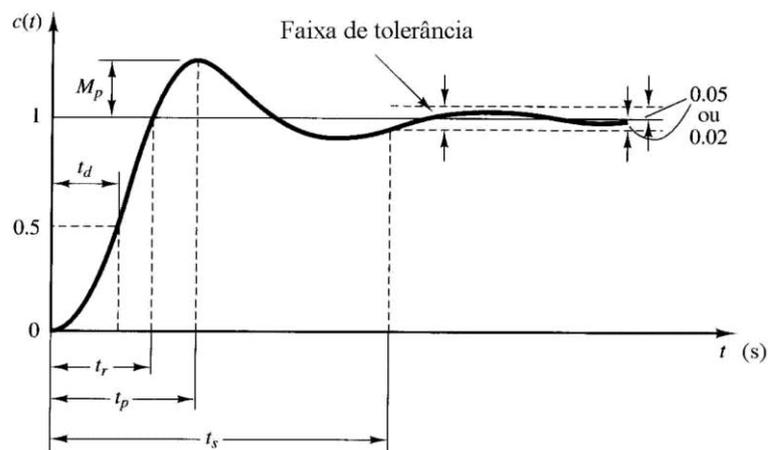


Figura 2.17 - Parâmetros de desempenho de um sistema de segunda ordem.

Fonte: OGATA (p.155, 2010).

Os índices que são extraídos da resposta ao degrau podem ser descritos como tempo de atraso  $t_d$ , tempo de subida  $t_r$ , tempo de pico  $t_p$ , sobressinal  $M_p$ , tempo de acomodação  $t_s$ .

a. Tempo de atraso ( $t_d$ )

Tempo de atraso entende-se como o tempo levado para que a resposta alcance a metade do valor final, valor unitário (OGATA p.154, 2010).

b. Tempo de subida ( $t_r$ )

Conforme AGUIRRE (p.50, 2007), o parâmetro do tempo de subida está relacionado com a velocidade do sistema, ou seja, sistemas rápidos possuem baixos valores de  $t_r$ . O tempo de subida pode ser encontrado, como visto em OGATA (p.154, 2010), como sendo o tempo requerido que a resposta passe de 10 a 90%, de 5 a 95% ou de 0 a 100% do valor final. Em sistemas de segunda ordem considera-se normalmente valores de 0 a 100% do valor final.

c. Tempo de pico ( $t_p$ )

O tempo de pico pode ser compreendido como o tempo transcorrido até atingir o primeiro pico de sinal, valor máximo (NISE, p.140, 2011).

d. Sobressinal ( $M_p$ )

O valor de sobressinal é comumente representado em porcentagem, e pode ser compreendido como o valor máximo de pico medido a partir do valor do degrau unitário (OGATA p.154, 2010).

Em AGUIRRE (p.49, 2007), o autor expressa que:

“[...] Este valor, em muitos casos, está associado a questões de segurança, tais como tensão máxima que um circuito pode suportar, máxima deformação que uma estrutura pode suportar sem que haja ruptura do material, etc.”

Conforme OGATA (p.154, 2010), destaca-se que o valor máximo de sobressinal indica diretamente a estabilidade relativa do sistema, o qual pode ser expresso como:

$$M_p = \frac{y(t_p) - y(\infty)}{y(\infty)} \times 100\% \quad 2.32$$

e. Tempo de acomodação ( $t_s$ )

Tempo necessário para que o sistema atinja o valor de regime em torno do valor do degrau unitário. Existe um valor de tolerância de até 5% acima ou abaixo do valor de regime, para que se considere uma variação aceita (AGUIRRE p.49, 2007).

### 2.2.5 Estabilidade

A estabilidade de um sistema é a especificação mais importante. E de acordo com NISE (p.236, 2011), Um sistema linear invariante no tempo é considerado estável se a resposta natural do sistema tende a zero quando o tempo tende ao infinito.

Com a utilização de realimentação se torna possível estabilizar processos instáveis, ajustando parâmetros com finalidade de adaptar o desempenho em regime transitório. Para processos que já são estáveis em malha aberta, utiliza-se também a realimentação para ajustar o desempenho de modo alcançar especificações de um dado projeto (DORF, p.230, 2001).

### 2.2.6 Controlador PID

A maioria dos controladores industriais em uso atualmente se utilizam de controles do tipo PID ou PID modificado. A utilidade do controle PID está na sua vasta aplicabilidade à maioria dos sistemas de controle, principalmente quando o modelo matemático da planta é desconhecido e os métodos analíticos não podem ser utilizados (OGATA, p.557, 2010).

#### 2.2.6.1 Ação de controle proporcional:

O controle proporcional é aquele que atua em proporção ao erro. Contudo, quando mal regulado, pode gerar um erro residual (estacionário). Num controlador que possui ação de controle proporcional a relação entre, a saída do controlador e o sinal do erro atuante é (OGATA, p.20, 2010):

$$u(t) = K_P e(t) \quad 2.33$$

Onde:

- $u(t)$  = saída do controlador;
- $e(t)$  = erro atuante;
- $K_P$  = ganho proporcional.

Aplicando-se a transformada de Laplace na equação 2.33, obtém-se:

$$\frac{U(s)}{E(s)} = K_P \quad 2.34$$

Qualquer que seja o mecanismo real e o tipo de energia utilizada na operação, o controlador proporcional é essencialmente um amplificador de ganho ajustável (OGATA, p.53, 2010).

### 2.2.6.2 Ação de controle integral:

O controle integral elimina o erro estacionário do sistema em resposta ao degrau de entrada, o que é um grande aperfeiçoamento do controle proporcional puro, que não impede o erro residual (OGATA, p.197, 2010).

Para o controlador com ação integral, temos uma variação proporcional do valor de saída em relação ao erro atuante. Ou seja (OGATA, p.21, 2010):

$$\frac{du(t)}{dt} = K_i e(t) \quad 2.35$$

Ou:

$$u(t) = K_i \int_0^t e(t) dt \quad 2.36$$

Onde:

- $K_i$  = constante ajustável.

Assim, a função de transferência do controlador integral é, aplicando a transformada de Laplace na equação 2.35, tem-se:

$$\frac{U(s)}{E(s)} = \frac{K_i}{s} \quad 2.37$$

### 2.2.6.3 Ação de controle proporcional-integral:

Como visto em OGATA (p.21, 2010) a função do controle proporcional-integral é:

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt \quad 2.38$$

De forma análoga, a função de transferência para este tipo de controlador é:

$$\frac{U(s)}{E(s)} = K_p \left( 1 + \frac{1}{T_i s} \right) \quad 2.39$$

Onde:

- $T_i$  = tempo integrativo.

### 2.2.6.4 Ação de controle proporcional-derivativo:

Quando acrescentado a um controlador proporcional, a ação de um controle derivativo permite que o sistema obtenha um controlador de alta sensibilidade. A vantagem de se utilizar a ação de controle derivativo é a rápida resposta a uma taxa de variação do erro atuante.

Portanto, esse tipo de controle prevê o erro atuante e inicia uma ação corretiva antecipada antes que o erro se torne muito elevado (OGATA, p.21, 2010).

A ação deste tipo de controle se comporta de acordo com a equação abaixo:

$$u(t) = K_p e(t) + K_p T_d \frac{de(t)}{dt} \quad 2.40$$

Sua função de transferência é:

$$\frac{U(s)}{E(s)} = K_p(1 + T_d s) \quad 2.41$$

Onde:

- $T_d$  = tempo derivativo.

### 2.2.6.5 Ação de controle proporcional-integral-derivativo:

A combinação das ações de controle proporcional, controle integral e de controle derivativo denomina-se ação de controle proporcional-integral-derivativo (PID). Com essa combinação, a vantagem individual de cada uma das três ações de controle é contemplada (OGATA, p.21, 2010).

A equação das três ações, em um controlador, é dada por:

$$u(t) = K_p \cdot e(t) + \frac{K_p}{T_i} \cdot \int_0^t e(t) dt + K_p \cdot T_d \frac{de(t)}{dt} \quad 2.42$$

E a função de transferência de um controle PID é:

$$\frac{U(s)}{E(s)} = K_p \left( 1 + \frac{1}{T_i s} + T_d s \right) \quad 2.43$$

O diagrama de blocos de um controlador proporcional-integral-derivativo é apresentado na figura 2.18:

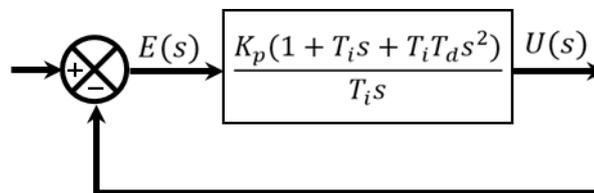


Figura 2.18 - Diagrama de blocos de um controlador PID.

Fonte: Própria, baseada em OGATA (p.22, 2010).

### 2.2.6.6 Sintonia de um controlador PID: Método Ziegler-Nichols

Em sintonia de controladores, quando é viável obter o modelo matemático da planta, se torna possível aplicar diversas técnicas de projeto para determinar os parâmetros do controlador.

Mas, se a planta tiver uma complexidade muito elevada, de modo que não possa obter facilmente seu modelo matemático, não será possível aplicar a mesma abordagem analítica que é utilizada quando se tem o modelo matemático da planta. Desta forma, deve-se recorrer a métodos experimentais de sintonia de controladores PID (OGATA, p.522, 2010).

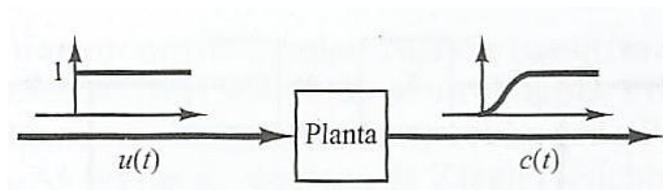
De acordo com OGATA (p.522, 2010):

[...] Ziegler e Nichols propuseram regras para a determinação de valores do ganho proporcional  $K_p$ , do tempo integral  $T_i$  e do tempo derivativo  $T_d$  baseadas na característica da resposta temporal de uma dada planta.

Existem dois métodos de sintonia Ziegler-Nichols:

a. Primeiro Método

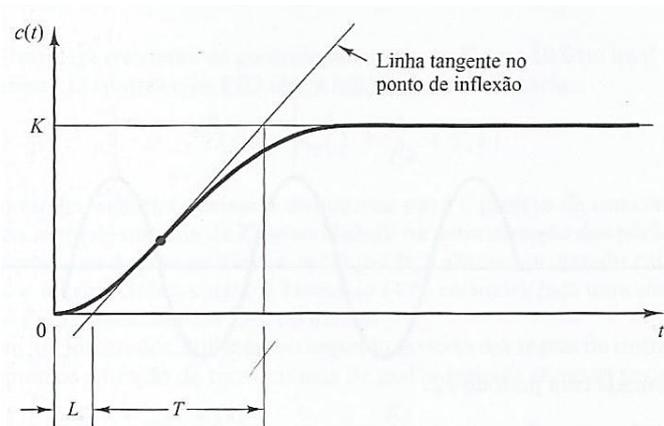
Neste primeiro método, deseja-se obter experimentalmente a resposta da planta em função de uma entrada em degrau unitário, de acordo com a figura 2.19:



**Figura 2.19 - Resposta ao degrau unitário de uma planta.**

**Fonte: OGATA (p.523, 2010).**

Esse método tem eficácia apenas se a curva no gráfico da resposta da planta apresenta um aspecto de um “S”, como é mostrado na figura 2.20. A curva característica da resposta ao degrau pode ser gerado através de uma simulação dinâmica da planta ou gerada experimentalmente (OGATA, p.523, 2010).



**Figura 2.20 - Curva da resposta ao degrau com aspecto de “S”.**

**Fonte: OGATA (p.523, 2010).**

De acordo com a curva da figura 2.2 apresentada, pode-se destacar duas constantes que serão utilizadas para determinar os valores de  $K_p$ ,  $T_i$  e  $T_d$ . Sendo  $L$  o atraso e  $T$  o tempo.

Segundo (OGATA, p.523, 2010):

[...] O atraso e a constante de tempo são determinados desenhando-se uma linha tangente no ponto de inflexão da curva com o formato em S e determinando-se a intersecção da linha tangente com o eixo dos tempos e a linha  $c(t) = K$ .

A função de transferência obtida é uma aproximação por um sistema de primeira ordem com um atraso de transporte, sendo:

$$\frac{C(s)}{U(s)} = \frac{K e^{-Ls}}{T_s + 1} \quad 2.44$$

De acordo com o primeiro método, os valores de  $K_p$ ,  $T_i$  e  $T_d$  assumirão os valores da fórmula da tabela 2.1:

**Tabela 2.1 - Valores de  $K_p$ ,  $T_i$  e  $T_d$  baseado na primeira regra de Ziegler Nichols**

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

**Fonte: OGATA (p.524, 2010).**

Um controlador PID sintonizado por este método Ziegler-Nichols de sintonia, em analogia com a equação 2.43, é representado pela expressão:

$$G_c(s) = 1,2 \frac{T}{L} \left( 1 + \frac{1}{2Ls} + 0,5Ls \right) = 0,6T \frac{\left( s + \frac{1}{L} \right)^2}{s} \quad 2.45$$

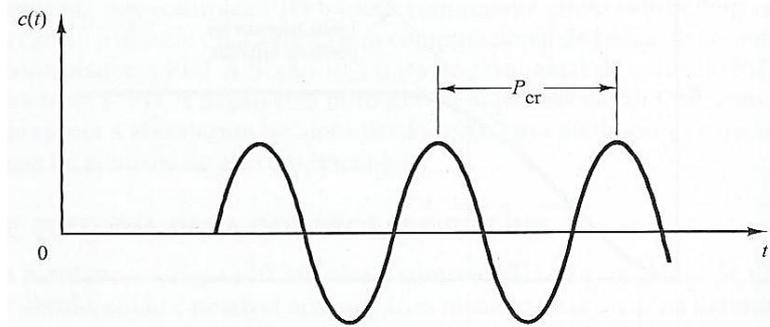
#### b. Segundo Método

No segundo método, é utilizado somente um ganho proporcional. Os valores dos ganhos integrativo e derivativo são definidos como:  $T_i \rightarrow \infty$  e  $T_d = 0$ .

O valor do ganho proporcional,  $K_p$ , será aumentado de 0 a um valor crítico, denominado  $K_{cr}$ , até que a saída do sistema apresente uma oscilação sustentada. Se a resposta do sistema não

apresentar uma oscilação quando o valor crítico do ganho proporcional for alcançado, este método não é recomendado.

O valor do ganho crítico  $K_{cr}$  e seu respectivo período de oscilação  $P_{cr}$  são extraídos experimentalmente conforme a figura 2.21:



**Figura 2.21 - Oscilação com período  $P_{cr}$ .**

**Fonte: OGATA (p.524, 2010).**

De acordo com o segundo método, os valores de  $K_p$ ,  $T_i$  e  $T_d$  assumirão os valores da fórmula da tabela 2.2:

**Tabela 2.2 - Valores de  $K_p$ ,  $T_i$  e  $T_d$  baseado na segunda regra de Ziegler Nichols.**

Tipo de controlador	$K_p$	$T_i$	$T_d$
P	$0,5K_{cr}$	$\infty$	0
PI	$0,45K_{cr}$	$\frac{1}{1,2}P_{cr}$	0
PID	$0,6K_{cr}$	$0,5P_{cr}$	$0,125P_{cr}$

**Fonte: OGATA (p.525, 2010).**

Um controlador PID sintonizado por este método Ziegler-Nichols de sintonia, em analogia com a equação 2.43, é representado pela expressão:

$$G_c(s) = 0,6K_{cr} \left( 1 + \frac{1}{0,5P_{cr}s} + 0,125P_{cr}s \right) G_c(s) = 0,075K_{cr}P_{cr} \frac{\left( s + \frac{4}{P_{cr}} \right)^2}{s} \quad 2.46$$

### 2.3 Conversão A/D

O conversor A/D é um dispositivo eletrônico, inserido dentro do microcontrolador (neste caso, o Arduino), que faz uma representação digital de uma grandeza analógica previamente medida. Esse sinal analógico se encontra na forma de faixa de tensão ou de corrente elétrica. O sinal digital é representado por dois estados de tensão, nível baixo e nível alto (0 ou 1).

Ou seja, como o conversor A/D disponível é de 10 bits, pode-se ter uma conversão com um range de 10 bits, que representa 1024 diferentes valores entre uma faixa de tensão de 0V a 5V que o Arduino esteja fazendo a aquisição, por exemplo. Estes 1024 valores são representados entre 0 (0000000000) e 1023 (1111111111), sendo 0V representado como 0 e 5V representado com 1023.

Para converter um sinal analógico para um valor digital correspondente àquela leitura, o conversor executa o seguinte cálculo:

$$K = \frac{ref_+ - ref_-}{2^n - 1} \quad 2.47$$

$$l_D \approx \left\| \frac{l_A - ref_-}{K} \right\| \quad 2.48$$

Onde:

- $K$  = Resolução mínima analógica da leitura;
- $ref_+$  = Limite superior da leitura analógica;
- $ref_-$  = Limite inferior da leitura analógica;
- $n$  = Número de bits da conversão;
- $l_D$  = Leitura convertida em digital;
- $l_A$  = Leitura analógica;
- $\|x\|$  = Número inteiro mais próximo à  $x$ .

Como o valor de entrada analógico máximo ( $ref_+$ ) é de 5V, o valor analógico mínimo ( $ref_-$ ) é de 0V e o número de bits da conversão é de 10 bits. Tem-se a seguinte resolução mínima para a conversão, utilizando a equação 2.47:

$$K = \frac{5V - 0V}{2^{10} - 1} \approx 4,89mV \quad 2.49$$

Este resultado indica que para cada incremento de 4,89mV, aproximadamente, de tensão na entrada analógica do Arduino, tem-se um acréscimo de 1 bit na variável digital. Ou seja, 0V significa 0 em digital, 4,89mV significa 1 em digital, 9,78mV significa 2 em digital, até que se atinja 5V, como foi apresentado anteriormente, possui um valor de 1023.

Com o valor da resolução mínima já computada, necessita-se apenas do valor de leitura da entrada analógica que o Arduino está coletando. Se a tensão medida, por exemplo, esteja apresentado um valor de 3,2V, a conversão será dada conforme o cálculo abaixo, de acordo com a equação 2.48, tem-se que:

$$l_D \approx \left\| \frac{3,2V - 0V}{4,89mV} \right\| \approx \lceil 654,72 \rceil \approx (655)_{10} = (1010001111)_2 \quad 2.50$$

O resultado da conversão implica que numa escala de 0 a 1023, a tensão de 3,2V apresenta um valor aproximado de 655 em base decimal, e o valor 1010001111 em base binária.

A conversão A/D é uma etapa fundamental que será aplicada no sistema didático proposto neste trabalho, pois todos os valores gerados pela saída (resposta) da planta, coletado pelo Arduino, são valores em níveis de tensão, ou seja, valores analógicos. E para que possa ser feito o controle via *software* ou até mesmo pelo próprio Arduino, que não será o caso, necessita-se que os números estejam na forma digital. Apenas desta forma será possível fazer a leitura dos valores de tensão de saída da planta.

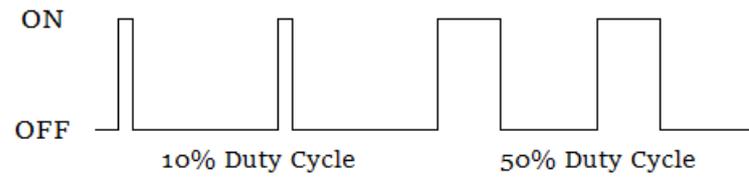
## 2.4 Modulação por largura de pulso

A modulação por largura de pulso ou PWM, do inglês "*Pulse Width Modulation*", é uma técnica bastante empregada que utiliza-se de chaveamento para controlar por exemplo velocidade de motores, intensidade luminosa de LED's, controle de servo motores, dentre outras aplicações. Este controle é realizado através da largura do pulso de uma onda quadrada. O conceito é que, se uma fonte de tensão é ligada constantemente à uma carga, a fonte irá gerar 100% de sua tensão. Caso o circuito seja aberto, ou seja, a chave seja aberta, essa tensão aplicada na carga é nula. Quando é controlado esse tempo de chaveamento, pode-se obter quaisquer níveis de tensão média entre zero e a tensão máxima da fonte de energia. Ou seja, se uma fonte estiver 50% do tempo com corrente e 50% do tempo sem corrente, num período de tempo  $t$ , contando que os intervalos de liga e desliga sejam igualmente espaçados, pode-se considerar que a tensão média para esse caso é de 50%.

O parâmetro do PWM, chamado de *duty-cycle*, expressa esta razão entre o tempo de funcionamento total e o tempo de pulso. Sendo assim, segue que:

$$DutyCycle = 100\% \frac{\text{largura do pulso}}{\text{período}} \quad 2.51$$

O *duty-cycle* é expresso em porcentagem, figura 2.22, já a largura de pulso e período são unidades de tempo, comumente representados em ms (milissegundos) ou  $\mu$ s (microsegundos) pois as frequências de chaveamento utilizadas são relativamente altas.



**Figura 2.22 - Representação de Duty-Cycle para PWM.**

**Fonte: Própria.**

Para o caso do microcontrolador utilizado no presente trabalho, o Arduino, possui pinos de saída digital aptos para operar em PWM, basta configurar via programação. Sua frequência de chaveamento padrão é de 500 Hz, este valor foi conferido em osciloscópio e dado como verdade. Desta forma pode-se ter quaisquer valores de tensão média entre 0 e 5 volts.

### 3 METODOLOGIA

#### 3.1 Modelamento do circuito elétrico

Com finalidade de se obter a função de transferência do sistema elétrico representado pela figura 3.1. Calculou-se o quociente da tensão de saída do sistema pela tensão de entrada do sistema, de acordo com o que foi demonstrado em (OGATA, 2010, p.64) para modelagem de sistemas elétricos.

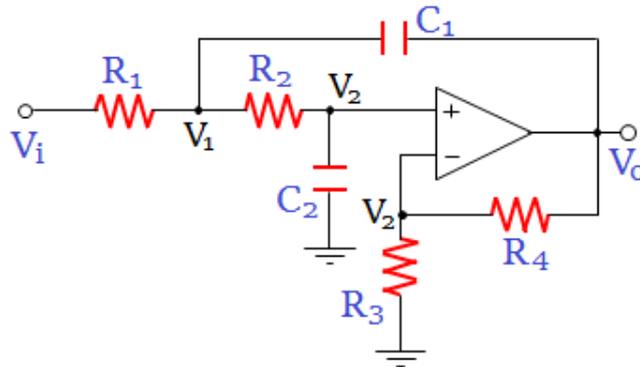


Figura 3.1 - Representação do circuito elétrico da planta.

Fonte: Própria.

Para efeito de cálculo, serão utilizados os parâmetros de um amplificador operacional ideal, conforme exposto em (GRUITER, 1988, p.5) que afirma que o amplificador responde à diferença de tensão aplicada em suas entradas. Ainda o autor ressalta que deve ser infinita a impedância de entrada e zero a impedância de saída do amplificador. Com isso, percebe-se que as tensões nos polos de entrada do amplificador operacional ideal tendem sempre a serem iguais, e sua corrente de entrada, zero.

Considera-se também, que todas as correntes envolvidas no circuito, saindo do nó 1, conforme apresentado na figura 3.2.

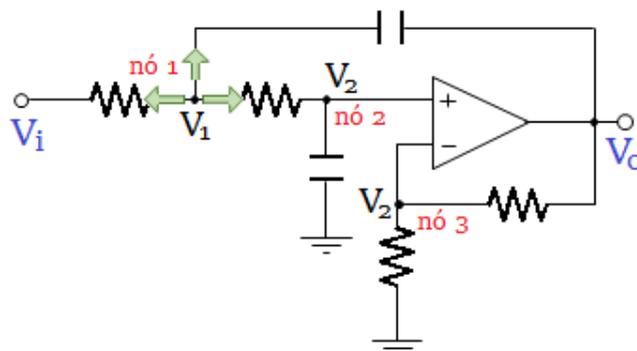


Figura 3.2 - Considerações iniciais para o nó 1.

Fonte: Própria.

Com isso, empregando a Lei das correntes de Kirchhoff, que estabelece que a soma de todas as correntes, que saem e que entram, em um nó deve ser zero (HALLIDAY, 2009, p.176).

Tem-se:

$$\frac{V_1 - V_i}{R_1} + \frac{V_1 - V_2}{R_2} + \frac{V_1 - V_o}{Z_{C1}} = 0 \quad 3.1$$

Sendo que:

- $V_i$  = Tensão de entrada;
- $V_o$  = Tensão de saída;
- $V_1$  = Tensão do nó 1;
- $V_2$  = Tensão do nó 2 ou 3;
- $Z_{C1}$  = Impedância do Capacitor 1;
- $Z_{C2}$  = Impedância do Capacitor 2.

Dando prosseguimento, é necessário expressar as tensões dos nós do circuito em função das tensões de entrada e saída do sistema, para depois substituí-las numa única expressão. Desta forma obtém-se:

$$V_2 = V_1 \frac{Z_{C2}}{Z_{C2} + R_2} \quad 3.2$$

$$V_2 = V_o \frac{R_3}{R_3 + R_4} \quad 3.3$$

$$V_1 = V_o \frac{R_3}{R_3 + R_4} \cdot \frac{Z_{C2}}{Z_{C2} + R_2} \quad 3.4$$

No tópico que se refere à impedâncias complexas, abordado em (OGATA, 2010, p.66), explicita-se a preferência do autor em escrever diretamente a transformada de Laplace das equações para capacitores e indutores, para não ter a necessidade de escrevê-las em equações diferenciais. Raciocínio esse, somente válido se considerarmos a hipótese de que as condições iniciais são nulas.

Procedendo de forma análoga, tem-se a transformada de Laplace para os capacitores 1 e 2.

$$Z_{C1} = \frac{1}{s \cdot C_1} \quad 3.5$$

$$Z_{C2} = \frac{1}{s \cdot C_2} \quad 3.6$$

Substituindo a equação 3.6 na expressão 3.4, segue que:

$$V_1 = V_o \frac{R_3}{R_3 + R_4} \cdot (1 + s \cdot R_2 \cdot C_2) \quad 3.7$$

As equações 3.3 e 3.7 substituídas na equação obtida pela Lei das correntes de Kirchhoff, equação 3.1, origina:

$$\begin{aligned} & \left[ \frac{1}{R_1} \left( V_o \frac{R_3}{R_3 + R_4} \cdot (1 + s \cdot R_2 \cdot C_2) \right) - \frac{V_i}{R_1} \right] \\ & + \left[ \frac{1}{R_2} \left( V_o \frac{R_3}{R_3 + R_4} \cdot (1 + s \cdot R_2 \cdot C_2) \right) - \left( V_o \frac{R_3}{R_3 + R_4} \cdot \frac{1}{R_2} \right) \right] \\ & + \left[ s \cdot C_1 \left( V_o \frac{R_3}{R_3 + R_4} \cdot (1 + s \cdot R_2 \cdot C_2) \right) - s \cdot C_1 \cdot V_o \right] = 0 \end{aligned} \quad 3.8$$

Isolando o único termo com o valor de tensão de entrada,  $V_i$ :

$$\begin{aligned} V_o \frac{R_3}{R_3 + R_4} \left[ \frac{1}{R_1} + \frac{s \cdot R_2 \cdot C_2}{R_1} + \frac{1}{R_2} + \frac{s \cdot R_2 \cdot C_2}{R_2} - \frac{1}{R_2} + s \cdot C_1 + s^2 \cdot R_2 \cdot C_1 \cdot C_2 \right. \\ \left. - s \cdot C_1 \left( \frac{R_3 + R_4}{R_3} \right) \right] = \frac{V_i}{R_1} \end{aligned} \quad 3.9$$

A equação foi manipulada para estar na forma característica de uma função de transferência. Ou seja, a tensão de saída dividida pela tensão de entrada do sistema, com estes termos isolados num lado da equação.

$$\frac{V_o}{V_i} = \frac{1}{R_1} \cdot \frac{\frac{R_3 + R_4}{R_3}}{\frac{1}{R_1} + \frac{s \cdot R_2 \cdot C_2}{R_1} + \frac{1}{R_2} + \frac{s \cdot R_2 \cdot C_2}{R_2} - \frac{1}{R_2} + s \cdot C_1 + s^2 \cdot R_2 \cdot C_1 \cdot C_2 - s \cdot C_1 \left( \frac{R_3 + R_4}{R_3} \right)} \quad 3.10$$

De acordo com (GRUITER, 1988, p.208), o circuito elétrico adotado para análise deste projeto é um filtro passa-baixas de 2ª ordem tipo Butterworth. Em referência ao que foi adotado pelo autor, a razão dos resistores  $R_3$  e  $R_4$ , representados na figura 3.1, determinam o ganho de tensão do sistema. Este ganho, representado pela simbologia  $K$ , se comporta:

$$K = \frac{R_3 + R_4}{R_3} = 1 + \frac{R_4}{R_3} \quad 3.11$$

Substituindo  $K$  da equação 3.11 na equação 3.10:

$$\frac{V_o}{V_i} = K \cdot \frac{1}{1 + s \cdot R_2 \cdot C_2 + s \cdot R_1 \cdot C_2 + s \cdot R_1 \cdot C_1 + s^2 \cdot R_1 \cdot R_2 \cdot C_1 \cdot C_2 - s \cdot R_1 \cdot C_1 \cdot K} \quad 3.12$$

Simplificando os termos, obtém-se a função de transferência no domínio da frequência:

$$\frac{V_o}{V_i} = K \cdot \frac{1}{s^2(R_1 \cdot R_2 \cdot C_1 \cdot C_2) + s(C_2(R_1 + R_2) - R_1 \cdot C_1(K - 1)) + 1} \quad 3.13$$

### 3.2 Comunicação arduino e matlab

A comunicação entre o *software* MATLAB e o *hardware* Arduino é feita através de uma comunicação serial via cabo USB, que interliga o Arduino ao computador. Para possibilitar esta comunicação foi utilizada um *support package*, um programa suporte, desenvolvido pela MathWorks® (empresa desenvolvedora de MATLAB) que facilita esta conexão.

O primeiro teste de comunicação foi realizado na plataforma Windows, utilizando-se a versão R2016a do MATLAB e a versão 1.6.8 do ArduinoIDE (*software* de desenvolvimento de códigos do Arduino).

O primeiro passo foi baixar o *support package* disponibilizado na página da MathWorks®, no endereço ([www.mathworks.com/hardware-support/arduino-matlab.html](http://www.mathworks.com/hardware-support/arduino-matlab.html)). Após a conclusão do *download* deve ser feita a instalação do arquivo. Para isto foi realizado a descompactação do arquivo recebido para extrair a pasta com o nome “ArduinoIO”, esta pasta pode ser alocada em qualquer caminho no computador. O conteúdo do arquivo baixado é apresentado conforme figura 3.3.

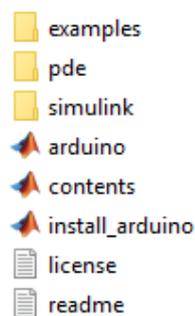
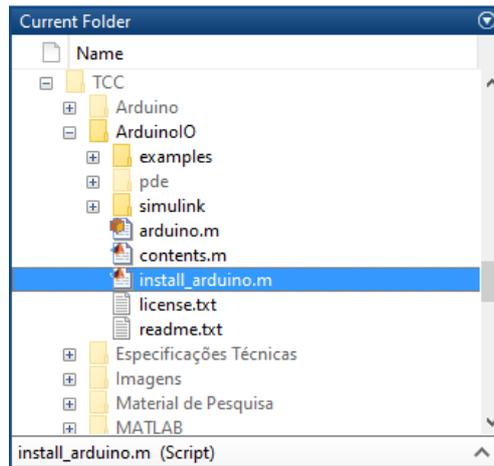


Figura 3.3 - Conteúdo do arquivo do support package.

Fonte: Própria.

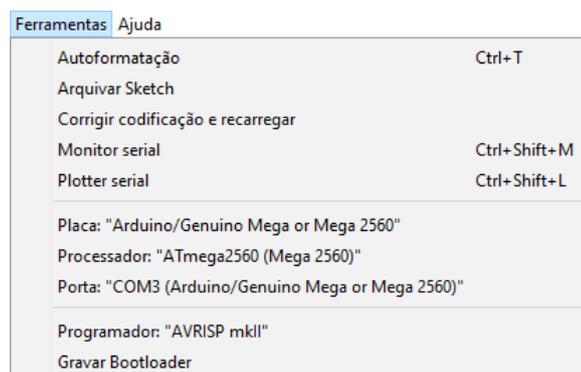
Com o MATLAB aberto, foi executado o arquivo “install\_arduino.m” localizado no caminho da pasta antes escolhido. Este comando deve ser realizado utilizando o MATLAB, conforme figura 3.4.



**Figura 3.4 - Instalação do script.**

**Fonte: Própria.**

Após esta etapa, o IDE do Arduino deve ser aberto a fim de compilar um outro programa contido no arquivo baixado. Para isso foi transferido o arquivo “adio.pde” localizado na pasta “ArduinoIO\pde” para a memória do Arduino. O arquivo em questão tem por finalidade configurar o *hardware* para realizar a leitura e escrita nas portas analógicas e digitais pelo MATLAB, lembrando que o Arduino deve estar devidamente conectado ao computador e, pelo IDE, devidamente atribuído o tipo do *hardware* utilizado e a porta serial em que o computador designou ao Arduino. No projeto foi utilizado o Arduino Mega 2560, a configuração do dispositivo pelo IDE pode ser apresentada na figura 3.5.



**Figura 3.5 - Configuração do tipo do Arduino e porta.**

**Fonte: Própria.**

Uma vez que esta configuração do Arduino foi realizada, não há necessidade de refazê-la, tendo em vista que o arquivo “adio.pde” ficará registrado na memória do dispositivo, salvo o caso em que por ventura se grave outro programa sobrescrevendo o anterior ou limpe a memória do Arduino. Nestes casos, deverá ser feita novamente sua configuração.

Para inicializar a comunicação, no MATLAB, foi atribuído o seguinte comando, conforme apresentado na figura 3.6:

```
% Conenctando ao Arduino:
a = arduino('COM3');
```

**Figura 3.6 - Comando do MATLAB para iniciar a conexão.**

**Fonte: Própria.**

Ao executar o código, uma mensagem do MATLAB foi apresentada, indicando que a conexão com o Arduino ocorreu com sucesso.

```
Basic Analog and Digital I/O (adio.pde) sketch detected !
Arduino successfully connected !
```

**Figura 3.7 - Mensagem de conexão bem sucedida.**

**Fonte: Própria.**

A partir do momento em que o programa retorna a mensagem, conforme figura 3.7, pode ser dado comandos, pelo MATLAB, de leitura e escrita de portas analógicas e digitais da placa utilizada no trabalho.

A vantagem deste *support package* é que ela permite dar comandos diretamente no Arduino utilizando-se o MATLAB, assim, torna-se mais simples a interface destes dispositivos. O módulo utilizado foi o de comando I/O das portas da placa do Arduino, o necessário para a execução do projeto. Este *support package* também contém suporte para controle de servomotores e leitura de *encoders*, mas para isso, o arquivo a ser instalado no Arduino deve ser outro, específico para cada caso.

### 3.3 Confeção de placa de circuito impresso

Quando se definiu os valores de resistência e capacitância dos componentes presentes no circuito, confeccionou-se uma placa de circuito impresso com intuito de agilizar futuros testes e ensaios, tendo em vista que uma vez que a placa estivesse já montada, não se perderia tempo em montá-la novamente, apenas interligá-la ao Arduino.

O método de confecção utilizado foi o método de transferência térmica. Para início de tal procedimento, necessita-se do desenho das trilhas do circuito. Por meio do Proteus 8.1 (*software* de simulação e projeto de circuitos eletrônicos), da empresa Labcenter Electronics, foi gerado um desenho das trilhas do circuito. Este desenho é uma representação do circuito, esboçado na simulação. Houve a necessidade de adicionar um diodo e um resistor a mais na saída do circuito para impedir que a saída do amplificador operacional gere uma tensão

negativa, podendo danificar o Arduino, figura 3.8. Com isso, na simulação pelo Proteus e na placa de circuito impresso constam estes componentes a mais. O desenho das trilhas do circuito pode ser demonstrado na figura 3.9.

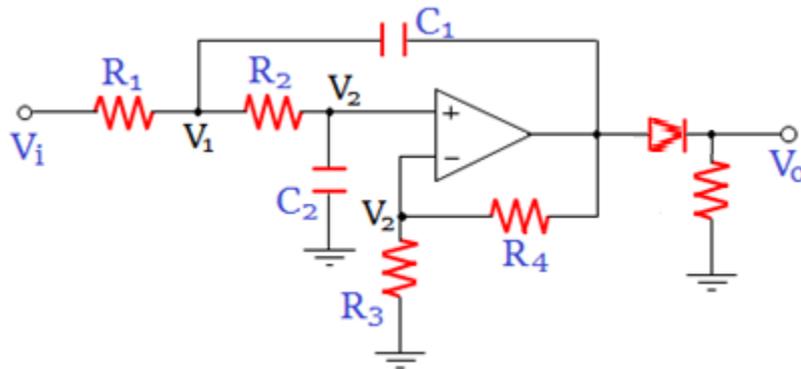


Figura 3.8 - Representação do circuito elétrico após alteração.

Fonte: Própria.

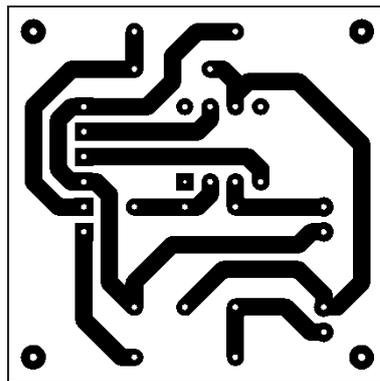


Figura 3.9 - Trilhas do circuito gerado em software.

Fonte: Própria.

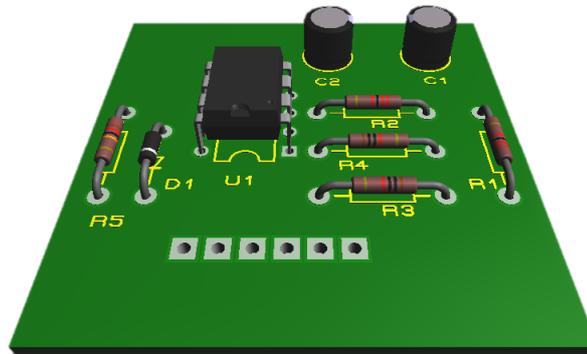
A próxima etapa foi a impressão do desenho num papel fotográfico utilizando uma impressora tipo *tonner*. O método de transferência térmica se dá por transferir o desenho do papel para a placa do circuito por meio de aplicação de calor, e para isso, a impressão deve seguir estes requisitos. Um aspecto que se deve atentar é a escala em que foi feita a impressão, porque a distância entre os pinos de um amplificador operacional, por exemplo, devem estar compatíveis, entre o real e o impresso.

Seguindo esse procedimento, fez-se a transferência térmica do desenho para a placa de fenolite cobreada, aplicando-se calor numa temperatura constante de 200° C durante 120 segundos e se utilizando de uma prensa térmica. Após o término da transferência deve-se analisar se todo o desenho foi passado para a placa, sem falhas. Quando o desenho das trilhas foi devidamente transferido, realizou-se a corrosão da camada cobreada da placa com a utilização de uma solução de percloroeto de ferro e água. Após toda a camada de cobre

destacar-se da placa, foi utilizada uma esponja de aço para remover a tinta da impressora que protegia o desenho do circuito. Para proteger as trilhas de uma eventual oxidação, foi aplicado nitrato de prata na parte de cobre do circuito, a fim de dar um trato para garantir maior vida útil da placa de circuito impresso.

Para finalizar a montagem da placa, foi efetuado a furação nos locais indicados, o correto posicionamento dos componentes e, a soldagem, para a fixação e interligação deles por meio das trilhas. O resultado final está ilustrado conforme figura 3.10.

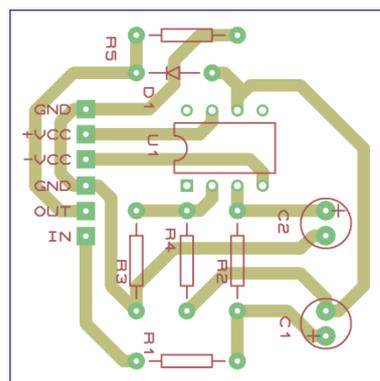
O amplificador operacional que foi utilizado é um amplificador simples LM741.



**Figura 3.10 - Representação gráfica da placa confeccionada.**

**Fonte: Própria.**

Uma aspecto que deve ser considerado importante é a disposição dos componentes e a ordenação dos terminais de entrada, saída, terra, +vcc e -vcc. Estes terminais estão especificados de acordo com a figura 3.11.



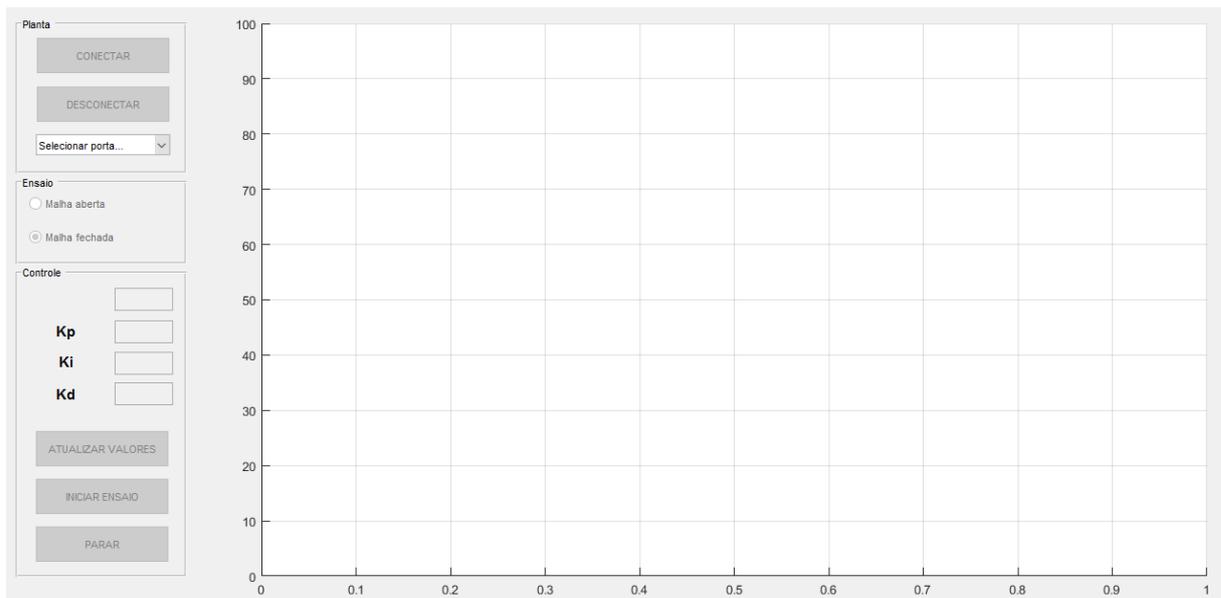
**Figura 3.11 - Posição dos componentes na placa.**

**Fonte: Própria.**

### 3.4 Interface gráfica

Desenvolveu-se uma interface gráfica para simplificar a operação do usuário com o algoritmo de controle, sendo essa interface para o controle da tensão de saída da planta, conforme figura 3.12.

O desenvolvimento deste ambiente utilizou do recurso GUIDE (do inglês, *graphical user interface development environment*), que é um ambiente de desenvolvimento de interface gráfica ao usuário disponibilizado pelo MATLAB.



**Figura 3.12 - Interface gráfica desenvolvida em MATLAB.**

**Fonte: Própria.**

Nesta interface é possível: selecionar a porta lógica de computador cujo Arduino esteja utilizando, conectá-lo e desconectá-lo; configurar o tipo de ensaio sendo malha aberta ou malha fechada; parametrizar ganhos de ação de controle (caso ensaio em malha fechada), e dar comando para iniciar ensaio e parar, sendo que, assim que o ensaio é iniciado, atualiza-se o gráfico em tempo real.

#### 3.4.1 Determinando tempo constante de aquisição

Um problema encontrado na utilização do *software* MATLAB como interface ao usuário é o recurso do gráfico em tempo real, porque a cada aquisição de dado, o sistema deve atualizar os valores mensurados. Apesar do tempo de cada iteração que compreende a leitura, cálculo e ações de controle, e disponibilização do gráfico serem relativamente rápidos, o tempo gasto para tal é variável pois leva em consideração, principalmente, o processamento do computador. Se estender este pensamento, considera-se que o tempo de cada iteração é

relativo para cada computador, assim sendo, há uma necessidade em padronizar este tempo gasto.

Com base em análises feitas para medir o tempo decorrido para realizar a aquisição dos dados do Arduino, realizar cálculos de controle, exibir o gráfico com as variáveis e implementar uma ação de controle, chegou-se num tempo de 50 ms (milissegundos), ou seja, 20 iterações por segundo. Este tempo foi suficiente para realizar tais operações.

Para padronizar o tempo decorrido, empregou-se uma lógica que utiliza do comando “tic”, iniciando um contador em segundos, no final do laço de repetição foi utilizado o comando “toc”, que fornece também em segundos, a janela de tempo de quando iniciou-se a contagem até a execução do “toc”. Nas iterações subsequentes, o tempo decorrido é reinicializado ao comando “tic”. Desta forma, nota-se uma melhoria considerável na aquisição de dados, levando para um período uniforme. O código empregado, foi:

```
while 1           % início da iteração (loop infinito)
    tic;          % início da contagem de tempo
    ...          % aquisição, ação de controle, exibição gráfico...
    t = toc;      %
    while t < 0.05 % comparação de tempo decorrido
        t = toc;  % atualização do tempo decorrido
    end          %
end             % fim da iteração
```

Uma análise do tempo de iteração (base de tempo em 50 ms) foi feita em dois ensaios, com 600 iterações cada. Os valores de tempo foram extraídos no próprio MATLAB e pode-se perceber que a média de tempo de cada iteração do laço de repetição permaneceu num valor muito próximo ao desejado e, o desvio padrão apresentou pouca variação em relação à média, demonstrando constância de intervalos entre as aquisições de dados. Os valores da análise realizada são apresentados pela tabela 3.1:

**Tabela 3.1 - Teste de desempenho do loop infinito do algoritmo do MATLAB.**

	<b>Ensaio 1</b>	<b>Ensaio 2</b>
<b>Média (s)</b>	0,050010	0,050011
<b>Desvio padrão (<math>\pm s</math>)</b>	0,000026	0,000019

**Fonte: Própria.**

### 3.5 Montagem

O sistema didático conta com a utilização de um computador, um Arduino, o circuito elétrico (planta), um cabo USB tipo A para tipo B, fios de cobre jumper para ligação e o *software* MATLAB devidamente instalado.

De forma exposta na figura 3.13, o Arduino se conectará ao computador pelo cabo USB e toda troca de dados entre eles será realizada através do cabo. A entrada da sinal da planta, será a saída PWM do Arduino, representado pela linha vermelha, a saída da planta conectará com a entrada analógica do Arduino, representado pela linha preta. Os fios de proteção “terra” ligados à planta, representados pelos fios de cor verde, têm por objetivo conectar um no Arduino, o outro à fonte de tensão, desta forma garante-se um “terra” comum para todo o sistema. Deve-se salientar que os dois conectores “terra” (GND), estão interligados nas trilhas do circuito, de forma exposta na figura 3.10.

Nas entradas +vcc e -vcc, por conveniência, adotou-se o uso de +5V e -5V.

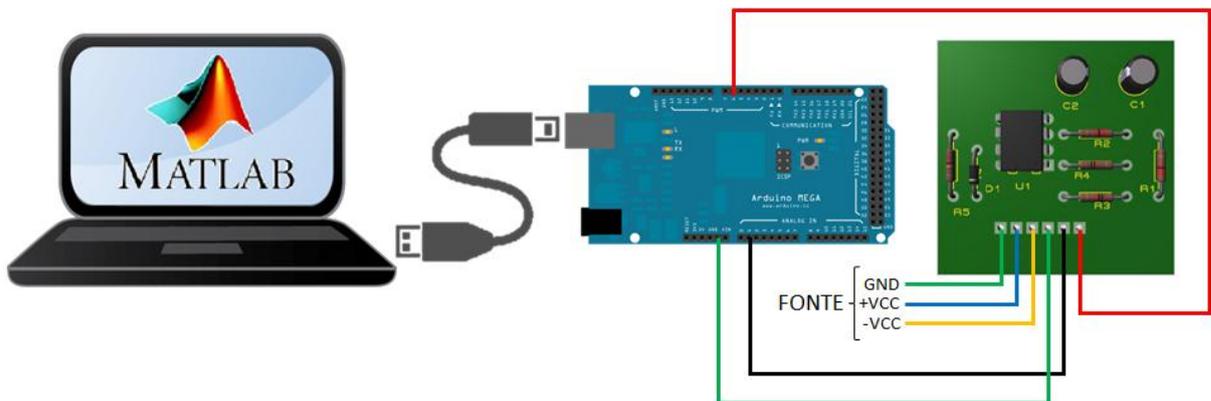


Figura 3.13 - Montagem física do sistema didático.

Fonte: Própria.

## 4 RESULTADOS E DISCUSSÃO

Neste capítulo serão apresentados os resultados dos testes realizados com a planta interligada ao Arduino, e os gráficos gerados pelo MATLAB. Uma comparação das respostas obtidas com os conceitos observados na teoria é feita, a fim de validar os resultados obtidos experimentalmente.

Como pode ser visto na figura 3.1 que representa o esquema elétrico da planta utilizada, tem-se no sistema, uma entrada de tensão  $V_i$  e uma saída de tensão  $V_o$ . Portanto, denomina-se que o sistema utilizado é monovariável ou SISO, isto é, de uma entrada e uma saída.

### 4.1 Função de transferência

Na capítulo 3, foi obtida a função de transferência da planta em função das resistências ( $R_1$ ,  $R_2$ ,  $R_3$  e  $R_4$ ) e das capacitâncias ( $C_1$  e  $C_2$ ), apresentado pela equação 3.13. Adotou-se, arbitrariamente, valores para os componentes eletrônicos presentes no sistema, sendo resistências de 22 k $\Omega$  para os componentes  $R_1$  e  $R_2$ , resistências de 10 k $\Omega$  para os componentes  $R_3$  e  $R_4$ , capacitâncias de 47  $\mu$ F para os componentes  $C_1$  e  $C_2$ .

Desta forma, pode-se expressar a função de transferência em valores numéricos, ou seja:

$$\frac{V_o}{V_i} = 2 \cdot \frac{1}{1,069156s^2 + 1,034s + 1} \quad 4.1$$

Dividindo os termos da equação 4.1 por 1,069156, tem-se a função de transferência de segunda ordem do circuito elétrico:

$$\frac{V_o}{V_i} = \frac{1,8706344069528}{s^2 + 0,96711798839458s + 0,93531720347639} = G(s) \quad 4.2$$

### 4.2 Características do sistema

Para extrair mais informações sobre o sistema, realizou-se um estudo sobre sua função de transferência e conseqüentemente, ensaios com a planta para verificar a correspondência da prática com a teoria.

#### 4.2.1 Coeficiente de amortecimento

Para determinar o coeficiente de amortecimento  $\zeta$ , considera-se uma resposta do sistema à entrada degrau unitário, equação 2,19. Logo, o coeficiente de amortecimento pode ser expresso como:

$$2\zeta\omega_n s = 0,96711798839458s \quad 4.3$$

$$\omega_n^2 = 0,93531720347639 \quad 4.4$$

$$\zeta = \frac{0,96711798839458}{2\sqrt{0,93531720347639}} = 0,5 \quad 4.5$$

O valor do coeficiente de amortecimento  $\zeta$  foi igual a 0,5 e, como visto anteriormente, quando este valor se limita entre  $0 < \zeta < 1$  pode-se dizer que o sistema apresenta uma resposta subamortecida para uma dada entrada degrau unitário.

Foi efetuado um ensaio em malha aberta com uma entrada degrau unitário e o resultado pode ser visto conforme figura 4.1. Este gráfico obtido está de acordo com a resposta esperada do sistema, e ainda se comporta de forma análoga com a resposta apresentada na figura 2.16 para um sistema com  $\zeta = 0,5$ .

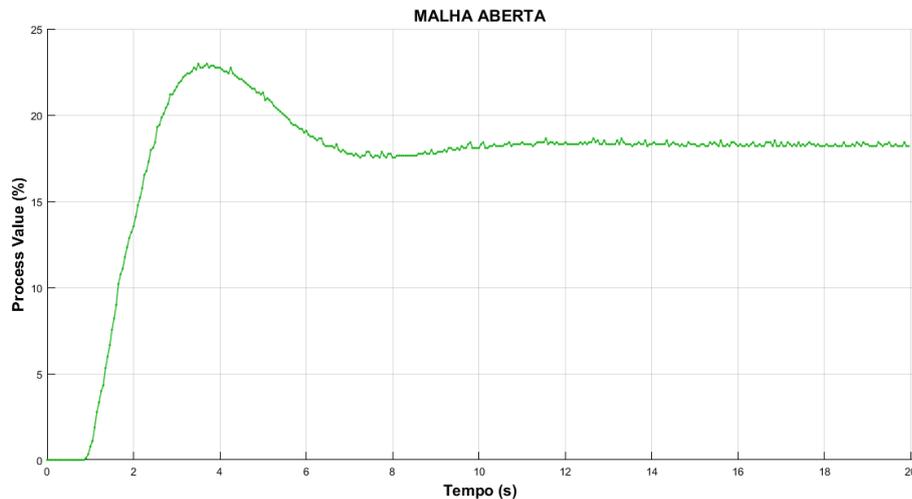


Figura 4.1 - Resposta do sistema, ensaio em malha aberta com entrada em degrau unitário.

Fonte: Própria.

#### 4.2.2 Estabilidade

Sobre a estabilidade do sistema, pode-se afirmar que este é estável pois a resposta natural do sistema tendeu à zero conforme o decorrer do tempo. Isto pode ser observado conforme o gráfico exposto na figura 4.1.

Analisando a equação para os polos do sistema, apresentada no capítulo 2, equação 2.20, tem-se:

$$\begin{aligned} s_1 &= -\zeta\omega_n + j\omega_n\sqrt{1-\zeta^2} = -0,5 \cdot 0,967118 + j0,967118\sqrt{1-0,5^2} \\ &= -0,484 + j0,838 \end{aligned} \quad 4.6$$

$$\begin{aligned} s_2 &= -\zeta\omega_n - j\omega_n\sqrt{1-\zeta^2} = -0,5 \cdot 0,967118 - j0,967118\sqrt{1-0,5^2} \\ &= -0,484 - j0,838 \end{aligned} \quad 4.7$$

Também verifica-se a estabilidade a partir do cálculo dos polos que são:  $s_1 = -0,484 + j0,838$  e  $s_2 = -0,484 - j0,838$ , ou seja, estão alocados no semiplano esquerdo do plano complexo.

### 4.3 Desempenho do sistema

Com finalidade de se obter o desempenho do sistema, utilizou-se a mesma curva de saída da planta para uma entrada degrau, veja figura 4.1. Após alguns pontos da curva serem destacados, em analogia à figura 2.17, originou-se a figura 4.2 que representa o desempenho real do sistema, permitindo que os índices desse desempenho possam observados.

O tempo de atraso ( $t_d$ ), entendido como o tempo decorrido para que se alcance a metade do valor final, é de aproximadamente 1,6 segundos. Já o tempo de subida ( $t_r$ ), que neste exemplo foi compreendido como o tempo requerido entre 0 e 100% do valor final, de acordo com o gráfico foi de 2,5 segundos. O tempo de pico ( $t_p$ ) conforme a figura 4.2 foi de 3,6 segundos, esse tempo de pico considera-se como o tempo que se leva para que a resposta atinja o primeiro pico de sinal. O último tempo analisado, o tempo de acomodação ( $t_s$ ), foi observado um intervalo de aproximadamente 10,3 segundos para este índice.

Para determinar o sobressinal ( $M_p$ ) do sistema considera-se o valor de pico máximo, normalmente ocorrido no instante ( $t_p$ ), em relação ao valor do degrau. Este valor de sobressinal pode ser expresso em porcentagem, de acordo com a equação 2.32, que é:

$$M_p = \frac{23\% - 18,2\%}{18,2\%} \times 100\% = 26,37\% \quad 4.8$$

Com este resultado encontrado pode-se dizer que para uma entrada degrau unitário, a resposta do sistema apresentou um sobressinal de 26,37% acima do valor do degrau, ou seja, um *overshoot* de 26,37% em relação ao valor de degrau.

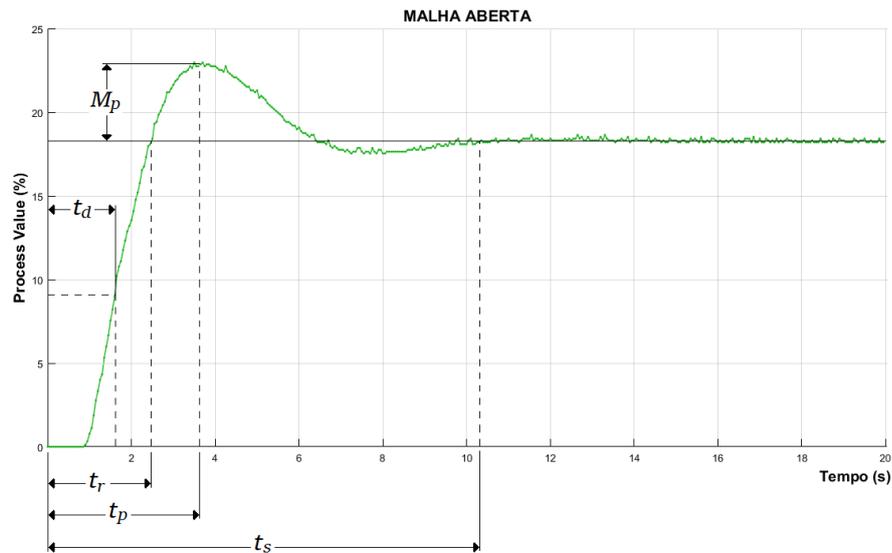


Figura 4.2 - Desempenho da planta.

Fonte: Própria.

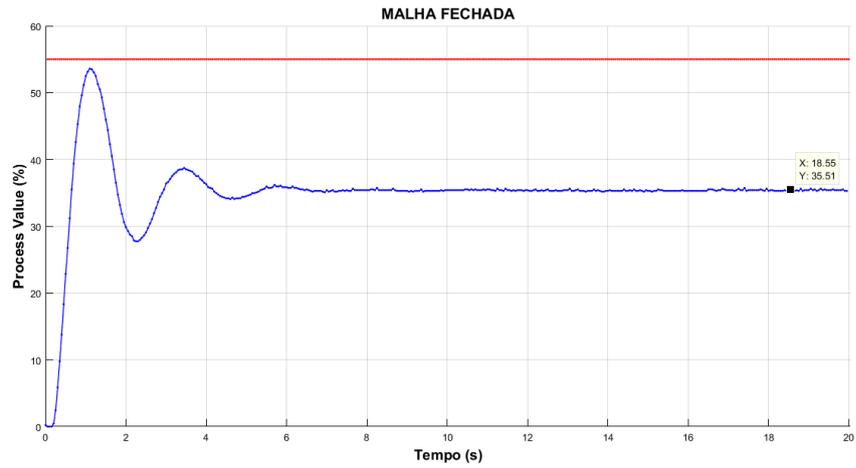
#### 4.4 Aplicação da Teoria de controle clássico

Ensaios com realimentação, ou seja, em malha fechada, foram realizados a fim de demonstrar o efeito de cada ação de controle do controle PID. Para as figuras apresentadas na seção 4.4, as respostas do sistema (PV) estarão representadas no gráfico pelo traçado de cor azul, e o *set point* de cor vermelho.

##### 4.4.1 Ação de controle proporcional

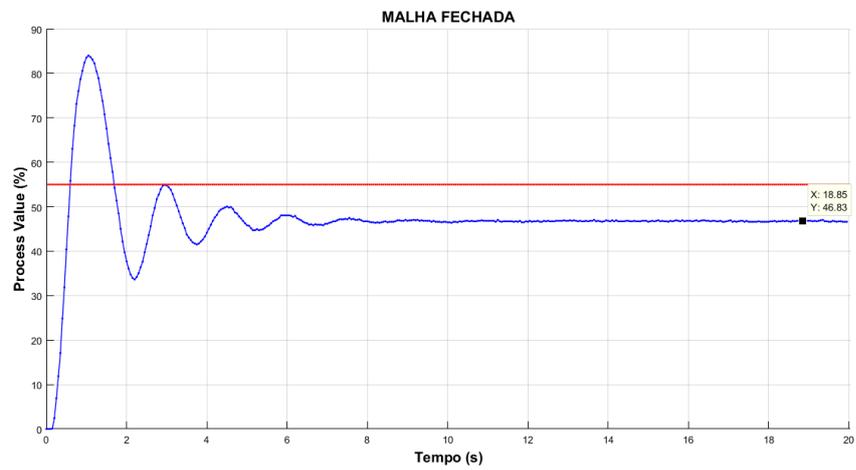
Para analisar o efeito da ação proporcional num controle, foram efetuados testes em que somente utilizou-se do ganho proporcional como ação de controle, os ganhos integral e derivativo, neste caso, foram considerados zero. Executou-se um teste com ganho igual a 1, conforme figura 4.3, outro teste com ganho igual a 5, conforme figura 4.4, e o último teste considerou-se o ganho proporcional igual a 10, representado pela figura 4.5. Através destes resultados pode-se afirmar que as respostas obtidas condizem com o esperado e com o que foi observado no estudo das bibliografias.

De acordo com as respostas obtidas, representadas pelas figuras 4.3, 4.4 e 4.5, verificou-se que nenhum valor de ganho proporcional foi satisfatório para controlar o sistema pois, a saída da planta não alcançou o valor de *set point* gerando ainda um erro residual. Nota-se também que a resposta apresenta um maior *overshoot*, sobressinal, e maior oscilação conforme o ganho proporcional é gradualmente aumentado.



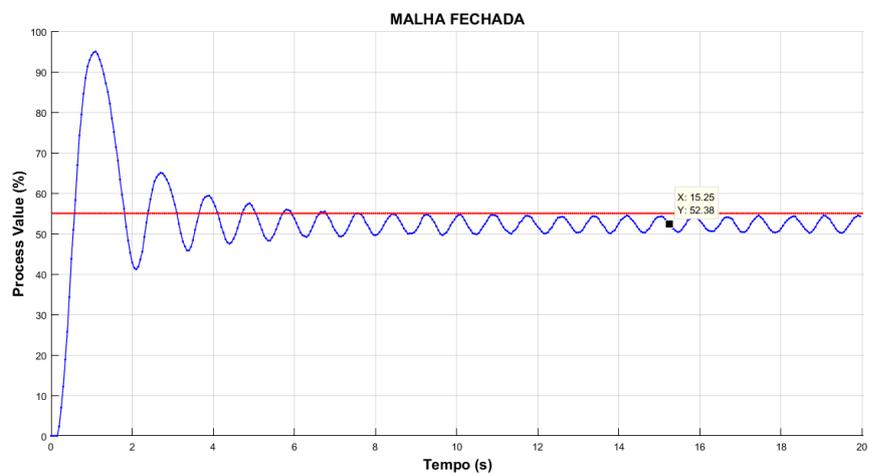
**Figura 4.3 - Ação de controle proporcional com ganho ( $K_p$ ) em 1.**

**Fonte: Própria.**



**Figura 4.4 - Ação de controle proporcional com ganho ( $K_p$ ) em 5.**

**Fonte: Própria.**



**Figura 4.5 - Ação de controle proporcional com ganho ( $K_p$ ) em 10.**

**Fonte: Própria.**

#### 4.4.2 Ação de controle proporcional-integral

Com finalidade de demonstrar como a ação integral influencia num controle proporcional-integral, o valor de ganho proporcional foi fixado em 1, e o valor de ganho integral foi variado. O ganho derivativo manteve-se zero para estes testes.

Foram executados testes com o valor de ganho integral igual a 0,2, conforme figura 4.6, outro teste com ganho igual a 0,4, de acordo com a figura 4.7, e um último teste com o valor do ganho integral igual a 1, segundo figura 4.8. O resultado destas três curvas correspondem com a resposta esperada e de acordo com a teoria de controle.

Utilizando-se o controle proporcional-integral obteve-se êxito para controlar o sistema, uma vez que o valor de *set point* foi devidamente atingido, mesmo que com diferentes tempos para tal, ou seja, o erro residual foi eliminado pelo efeito a ação integral. Com valores de  $K_i$  menores, como apresentados pela figura 4.6, a resposta obteve um pico de sobressinal pequeno e poucas oscilações até a acomodação do sinal, o tempo para que o erro possa ser considerado “zero” foi relativamente grande, entre 30 e 35 segundos. O ganho  $K_i$  utilizado na figura 4.7 teve o melhor resultado entre os ensaios pois a saída atingiu o *set point* consideravelmente mais rápido que os demais testes, em torno de 20 segundos, mas apresentou um sobressinal acentuado e moderada oscilação em torno do valor desejado. Com  $K_i$  alto, figura 4.8, pode-se perceber elevado sobressinal e elevada oscilação, o erro neste caso foi eliminado em aproximadamente 35 segundos.

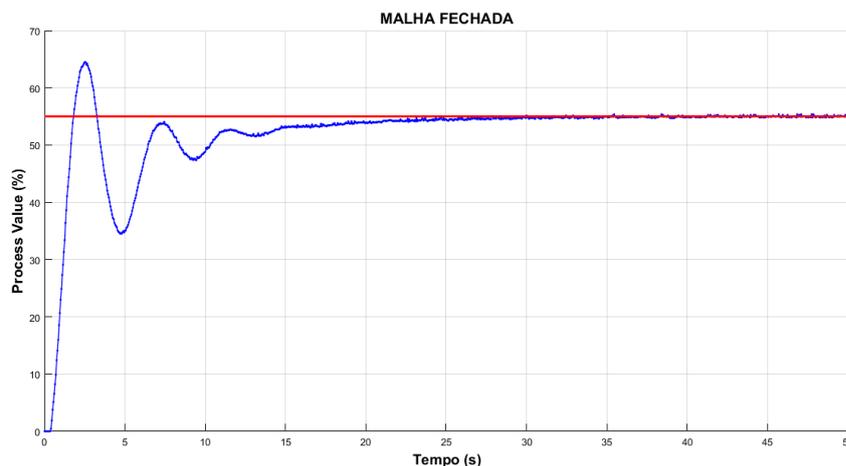
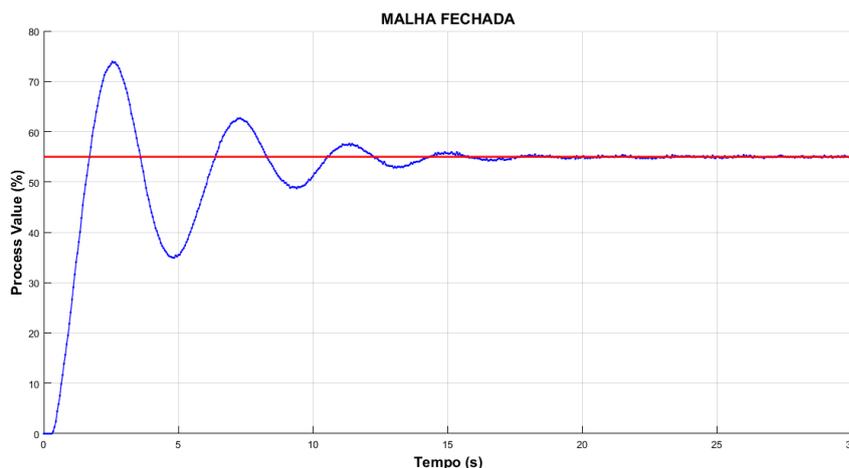


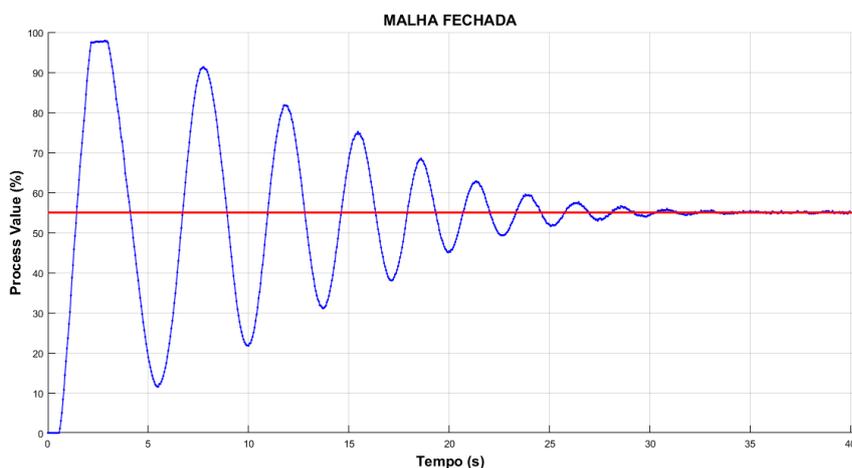
Figura 4.6 - Ação de controle proporcional-integral com ganho ( $K_i$ ) em 0,2.

Fonte: Própria.



**Figura 4.7 - Ação de controle proporcional-integral com ganho ( $K_i$ ) em 0,4.**

**Fonte: Própria.**

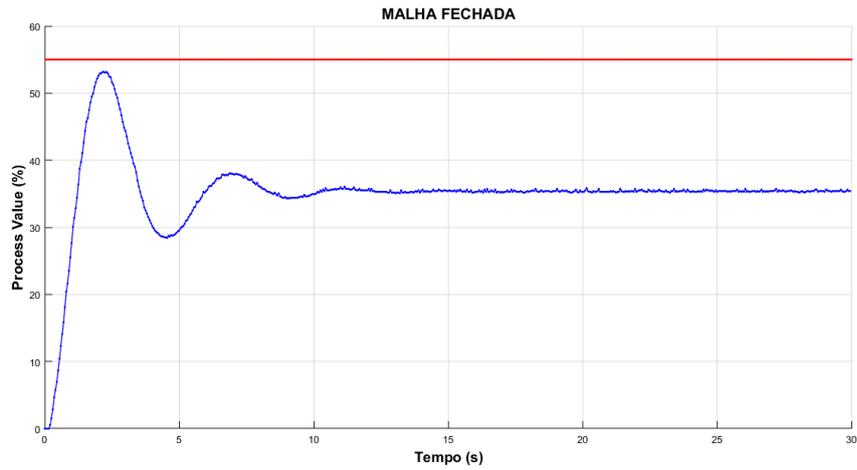


**Figura 4.8 - Ação de controle proporcional-integral com ganho ( $K_i$ ) em 1.**

**Fonte: Própria.**

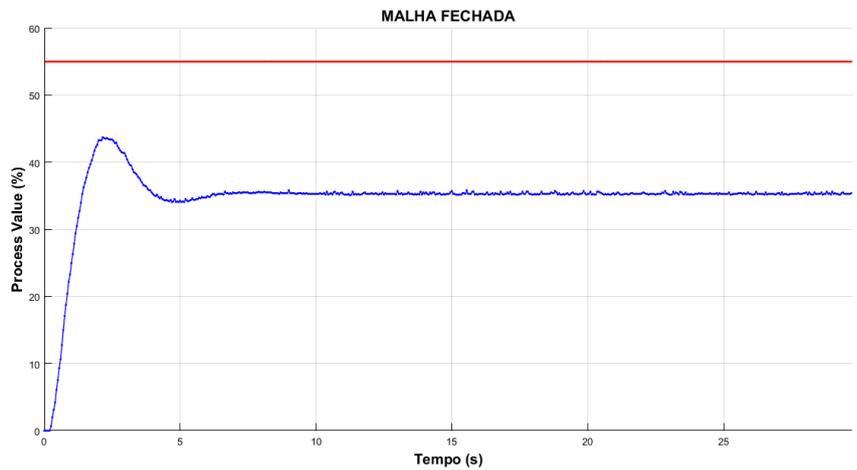
#### **4.4.3 Ação de controle proporcional-derivativo**

Para constatar como a ação derivativa atua em um controle proporcional-derivativo, o valor de ganho proporcional foi fixado em 1, e o valor de ganho derivativo foi alterado. O ganho integral manteve-se zero para estes testes. Os ensaios executados foram com o valor de  $K_p$  igual a 1, conforme figura 4.9, outro teste com ganho igual a 10, de acordo com a figura 4.10, e um último teste com o valor do ganho derivativo em 20, segundo figura 4.11. A saída da planta se comportou de acordo com a resposta esperada e em conformidade com a teoria de controle. Nota-se que com o aumento do ganho derivativo não foi necessário para alcançar o valor de *set point*, o ganho derivativo é uma ação que visa alterar a sensibilidade da ação de controle e atua em resposta com uma taxa de variação do erro.



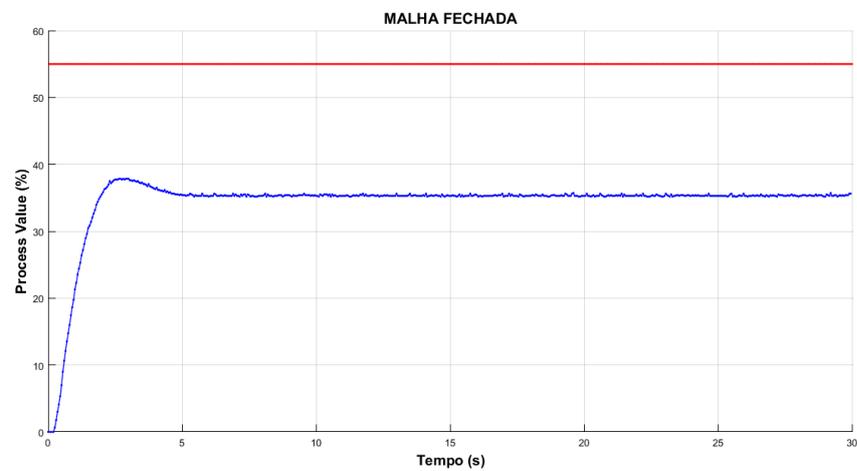
**Figura 4.9 - Ação de controle proporcional-derivativo com ganho ( $K_d$ ) em 1.**

**Fonte: Própria.**



**Figura 4.10 - Ação de controle proporcional-derivativo com ganho ( $K_d$ ) em 10.**

**Fonte: Própria.**



**Figura 4.11 - Ação de controle proporcional-derivativo com ganho ( $K_d$ ) em 20.**

**Fonte: Própria.**

#### 4.4.4 Ação de controle proporcional-integral-derivativo

Quando aplicou-se o controle proporcional-integral-derivativo, ou PID, pode-se perceber que as características e vantagens inerentes de cada uma das ações de controle se faz presente. Para o ensaio realizado com ação PID foi adotado o valor de  $K_p$  igual a 1, para o valor do ganho integral  $K_i$ , optou-se pelo valor de 0,4, pois esse valor apresentou melhor performance entre todos os testes de ganho integral, e por último o ganho  $K_d$ , de forma análoga à análise feita com o ganho integral de 0,4, o ganho derivativo obteve melhor resposta quando utilizado igual a 20.

Adotando estes valores pode-se perceber que a resposta apresentou pouca oscilação, em função do alto ganho derivativo, e uma rápida eliminação do erro por conta do ajuste ideal do ganho integral. Assim, a saída do sistema obteve o melhor desempenho constatado levando um tempo entre 10 a 15 segundos para alcançar o *set point*. O gráfico deste ensaio pode ser visto conforme figura 4.12.

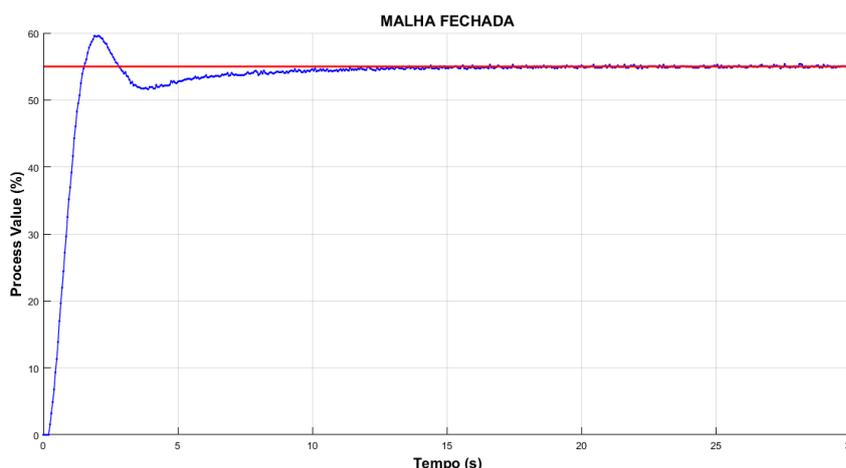


Figura 4.12 - Ação PID.

Fonte: Própria.

#### 4.5 Sintonia Ziegler-Nichols

Executou-se o método de sintonia de controladores Ziegler-Nichols, para extrair os valores dos ganhos e compará-los com o que já foi visto e discutido anteriormente.

##### 4.5.1 Primeiro método

O primeiro método consiste em observar a resposta do sistema para uma entrada tipo função em degrau. Para isso utilizou-se a figura 4.1 como base para obter os valores de atraso ( $L$ ) e de tempo ( $T$ ), conforme explícito na seção 2.2.6.6. O atraso foi de 0,833 segundos, de acordo

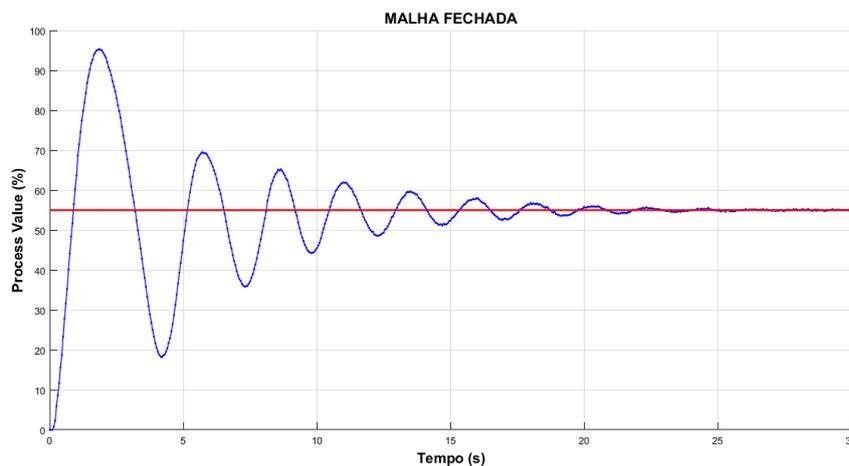
com o observado, já o tempo ( $T$ ) observa-se um período de 1,566 segundos. Com estes valores pode-se calcular o ajuste do controlador PID pela tabela 2.1, sendo:

$$K_p = 1,2T = 1,2 \cdot 1,566 = 2,255 \quad 4.9$$

$$K_i = \frac{K_p}{T_i} = \frac{K_p}{2L} = \frac{2,255}{2 \cdot 0,833} = 1,353 \quad 4.10$$

$$K_d = K_p T_d = 2,255 \frac{T}{2} = \frac{2,255 \cdot 0,833}{2} = 0,939 \quad 4.11$$

Empregando estes valores no ensaio, obtém-se a figura 4.13. A resposta para este primeiro método de sintonia foi satisfatória, uma vez que o tempo para se alcançar o *set point* foi relativamente curto. No caso de não possuir informação sobre a função de transferência do sistema este método se mostrou eficaz, embora não tenha sido a melhor resposta encontrada. Observou-se que o primeiro método pode direcionar para um caminho certo de forma mais eficaz.

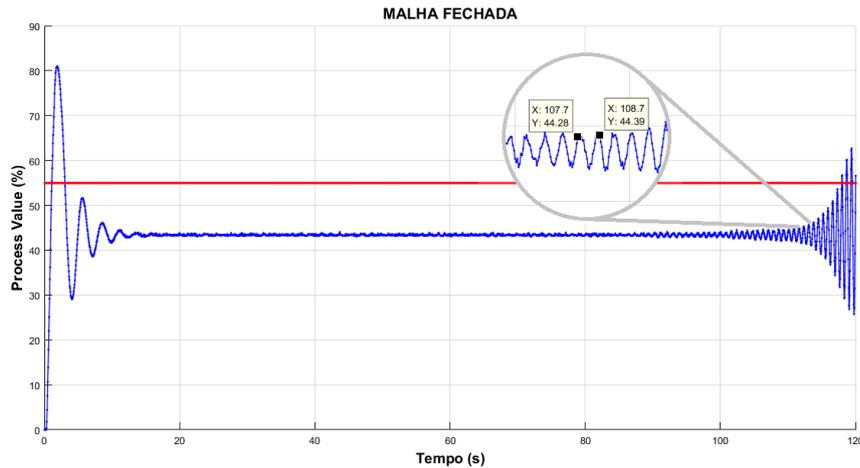


**Figura 4.13 - Controle PID pela sintonia de Ziegler-Nichols, primeiro método.**

**Fonte: Própria.**

#### 4.5.2 Segundo método

Para executar o segundo método, realizou-se um ensaio em malha fechada apenas com ganho proporcional com intuito de encontrar o ganho proporcional crítico  $K_{cr}$  e seu respectivo período de oscilação  $P_{cr}$ . De acordo com o teste realizado, segundo figura 4.14, o ganho crítico encontrado foi igual a 2, seguido com um período de oscilação de 1 segundo.



**Figura 4.14 - Resposta do sistema para o ganho ( $K_p$ ) crítico.**

**Fonte: Própria.**

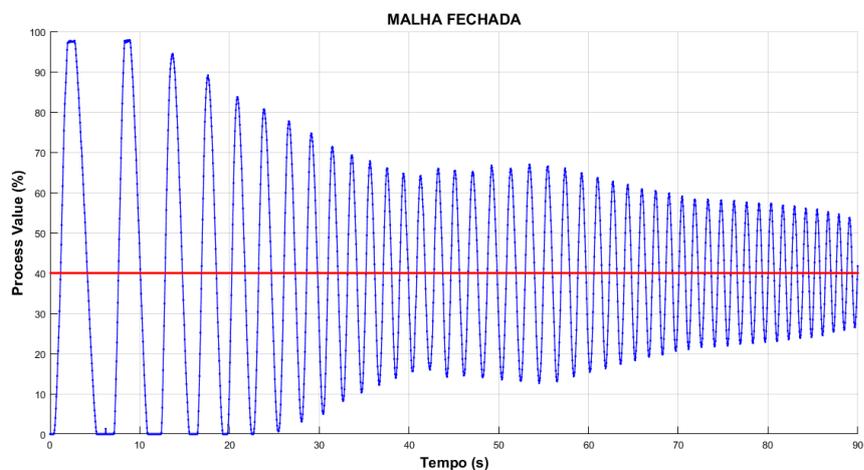
Assim, os valores dos ganhos  $K_p$ ,  $K_i$  e  $K_d$  são expressos na forma proposta do segundo método de sintonia, tabela 2.2, sendo:

$$K_p = 0,6K_{cr} = 1,2 \quad 4.12$$

$$K_i = \frac{K_p}{T_i} = \frac{K_p}{0,5P_{cr}} = \frac{1,2}{0,5} = 2,4 \quad 4.13$$

$$K_d = K_p T_d = K_p 0,125P_{cr} = 1,2 \cdot 0,125 \cdot 1 = 0,15 \quad 4.14$$

A resposta do sistema utilizando o segundo método de Ziegler-Nichols, figura 4.15, não demonstrou-se adequada pois observou-se uma oscilação da resposta do sistema e um sobressinal muito elevados, além de não controlar o sistema. Este efeito deve-se ao fato que no momento da obtenção do ganho crítico, o período de oscilação  $P_{cr}$  apresentou-se baixo, com isso o ganho integral  $K_i$  obtido pela fórmula foi alto.



**Figura 4.15 - Controle PID pela sintonia de Ziegler-Nichols, segundo método.**

**Fonte: Própria.**

Após a conclusão dos testes, observa-se que a meta de estar em correspondência com os resultados previstos, isto é, com aquilo visto na teoria foi alcançada. Através dos ensaios realizados, pode-se comprovar de forma experimental o comportamento das ações de controle proporcional, integral e derivativa.

## 5 CONCLUSÕES

Este trabalho propôs a montagem de um sistema didático para aplicação de técnicas de controle, utilizando um circuito elétrico como sistema ou planta, um microcontrolador para aquisição de dados do sistema e um *software* para realizar a execução dos cálculos de controle e gerar a interface gráfica.

Primeiramente, foram executados testes de modo a garantir que o microcontrolador se comunicasse com o *software* de forma eficiente. Optou-se por utilizar um *support package* criado especialmente para esta interface entre o Arduino e o MATLAB, o que garantiu a eficiência do procedimento.

Com o esquema elétrico da planta previamente definido, realizou-se seu modelamento matemático, objetivando extrair a função de transferência do sistema. Em seguida, executou-se a montagem do sistema físico, que se consistiu da confecção de uma placa de circuito impresso.

Por fim, realizou-se a montagem do sistema didático propriamente dito, e programou-se as técnicas de controle previstas, dadas pelo controlador PID. Uma interface gráfica amigável foi implementada a fim de se realizar ensaios de controle do sistema proposto, disponibilizando gráficos das variáveis de controle em tempo real, e ações de controle comandadas também via interface.

O resultado dos testes realizados foram abordados no capítulo 4. Pode-se perceber que diferentes combinações de ações de controle (proporcional, integral, derivativa) foram utilizadas a fim de demonstrar correspondência dos gráficos obtidos com as respostas esperadas, discutidas no capítulo 2, onde é introduzido conceitos gerais de controle e conceitos de teoria de controle clássico. As respostas obtidas através dos testes com a planta tiveram um comportamento bastante satisfatório, ou seja, apresentou-se dentro do previsto pela literatura. Conclui-se, então, que o sistema pode ser empregado para ensaios didáticos uma vez que comprova os resultados teóricos e com isso, torna-se uma ferramenta para o apoio ao aprendizado. Dentre os métodos de sintonia previstos para utilização, que eram o primeiro e o segundo método de Ziegler Nichols para sintonia de controladores, apenas o primeiro método de sintonia teve um desempenho satisfatório.

Apesar da interface gráfica com o usuário criada apresentar-se eficiente, assemelhando-se com um supervisor simples, a mesma demonstrou ser em diversas ocasiões lenta. Isso ocorre devido ao fato do MATLAB apresentar gráfico em tempo real o que sobrecarrega o

processador do computador. Com isso, foram realizados alguns ajustes para que este problema fosse minimizado, escolhendo-se um tempo de amostragem para cada aquisição de dados.

O desenvolvimento deste sistema didático, seguindo o objetivo inicial do trabalho, permite que possam ser desenvolvidos muitos outros projetos futuros no Campus Araxá do CEFET-MG.

Propostas futuras seriam: a implementação de novos métodos de controle bem como novos métodos de sintonia de controladores não explorados, realizando a coleta da tensão aplicada nos capacitores do circuito. Também poderão ser futuramente feitas aplicações de técnicas de controle moderno no modelo espaço de estados. Outra proposta seria uma reformulação da interface gráfica com usuário, explorando um outro *software* para ser levado em comparação com a interface utilizada no presente trabalho. Por fim, a possibilidade da utilização de uma outra combinação de circuito elétrico instável em malha aberta, para real aplicação de técnicas de controle que levem o sistema à estabilidade em malha fechada.

## REFERÊNCIAS

- AGUIRRE, L. A. et. al. **Enciclopédia de Automática Controle & Automação**. 1. ed. São Paulo: Blücher, 2007.
- AL-BUSAIDI, A. M. **Development of an Educational Environment for Online Control of a Biped Robot using MATLAB and Arduino**. 2012. Department of Mechanical and Industrial Engineering, College of Engineering, Sultan Qaboos University, Oman.
- BEGA, E. A. et. al. **Instrumentação Industrial**. 2. ed. Rio de Janeiro: Interciência, 2006.
- BRAGA (org), J. C. **Objetos de aprendizagem, volume 1: introdução e fundamentos**. Santo André: UFABC, 2014.
- DORF, R.C, BISHOP, R.H. **Sistemas de controle modernos**. 8. ed. Rio de Janeiro: LTC, 2001.
- MAYA, P.A. **Controle Essencial**. 2.ed. São Paulo: Pearson Education do Brasil, 2014.
- GRUITER, A. F. de. **Amplificadores Operacionais: Fundamentos e aplicações**. São Paulo: McGraw-Hill, 1988.
- HALLIDAY, D. **Fundamentos de física, volume 3: Eletromagnetismo**. 8 ed. Rio de Janeiro: LTC, 2009.
- MIYAGI, P. E. **Controle programável: fundamentos do controle de sistemas a eventos discretos**. 1. ed. São Paulo: Blücher, 1996.
- MONTEIRO, L. H. A. **Sistemas Dinâmicos**. 2. ed. São Paulo: Livraria da Física, 2006.
- NISE, N. S. **Engenharia de sistemas de controle**. 5. ed. Rio de Janeiro: LTC, 2011.
- OGATA, K. **Engenharia de Controle Moderno**. 5. ed. São Paulo: Prentice Hall, 2010.
- SILVA, I. M., GOMES, F. J. **Ferramenta interativa para aprendizado em controle: projeto de compensadores utilizando lugs das raízes**. 2007. Universidade Federal de Juiz de Fora, Faculdade de Engenharia e Arquitetura, Juiz de Fora – MG.

## ANEXO – CÓDIGO FONTE

Este anexo apresenta o código fonte do algoritmo da interface gráfica implementada no *software* MATLAB.

```
function varargout = tela_ensaio_malha_fechada(varargin)

gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @tela_ensaio_malha_fechada_OpeningFcn,
                  ...
                  'gui_OutputFcn',  @tela_ensaio_malha_fechada_OutputFcn,
                  ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function tela_ensaio_malha_fechada_OpeningFcn(hObject, eventdata, handles,
varargin)
handles.output = hObject;

guidata(hObject, handles);

function varargout = tela_ensaio_malha_fechada_OutputFcn(hObject,
eventdata, handles)

varargout{1} = handles.output;

function SP_value_Callback(hObject, eventdata, handles)

function SP_value_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Kp_value_Callback(hObject, eventdata, handles)

function Kp_value_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Ki_value_Callback(hObject, eventdata, handles)
```

```

function Ki_value_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Kd_value_Callback(hObject, eventdata, handles)

function Kd_value_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function atualizar_valores_Callback(hObject, eventdata, handles)
    global SP;
    global Kp;
    global Ki;
    global Kd;
    SP = str2double(char(get(handles.SP_value,'String')));
    Kp = str2double(get(handles.Kp_value,'String'));
    Ki = str2double(char(get(handles.Ki_value,'String')));
    Kd = str2double(char(get(handles.Kd_value,'String')));

function iniciar_Callback(hObject, eventdata, handles)
    set(handles.iniciar,'enable','off')
    set(handles.parar,'enable','on')
    set(handles.desconectar_planta,'enable','off')
    set(handles.malha_fechada,'enable','off')
    set(handles.malha_aberta,'enable','off')
    global parar;
    parar = false;
    global a;

    switch get(get(handles.tipo_ensaio,'SelectedObject'),'Tag')

    case 'malha_fechada'
        global SP;
        global Kp;
        global Ki;
        global Kd;

        timeInterval = 0.05;
        time = 0;
        seg = 0;

        PV = 0;
        ErroAnt = 0;
        Erro = 0;
        dif_Erro = 0;
        Erro_acum = 0;
        P = 0;
        I = 0;
        D = 0;

        count = 1;
        curva = handles.Grafico1;
        grafico = plot(curva,seg,PV,'Marker','.', 'LineWidth',1,'Color',[0 0
1]);
        set(handles.Grafico1,'Color',[1 1 1],'Gridcolor',[0.5 0.5 0.5])

```

```

hold on;
grid on;

while 1
    tic;
    leitura = a.analogRead(1);

    PV(count) = leitura*0.11097;
    seg(count) = time;

    time = time+timeInterval;
    count = count+1;

    xlim(curva, [min(0) max(time)]); %Ajuste do eixo x no tempo
    ylim(curva, [min(0) max(100)]);
    set(grafico, 'YData', PV, 'XData', seg);

    Erro = SP - (leitura*0.11097);
    dif_Erro = Erro - ErroAnt;

    P = Kp*Erro;
    I = I+Ki*((Erro+ErroAnt)*timeInterval/2);
    D = Kd*dif_Erro;
    PID = P+I+D;

    ErroAnt = Erro;

    if PID > 100
        PID = 100;
    end
    if PID < 0
        PID = 0;
    end

    AcaoPWM = PID*2.55;
    x = round(AcaoPWM);

    a.analogWrite(6, x);

    pause(0.00001);
    t = toc;

    while t < 0.05
        t = toc;
    end

    if parar == true
        parar = false;
        break;
    end
    toc;
end
case 'malha_aberta'
timeInterval = 0.05;
time = 0;
seg = 0;
PV = 0;
count = 1;
curva = handles.Grafico1;

```

```

    grafico = plot(curva,seg,PV, 'Marker', '.', 'LineWidth',1, 'Color', [0 0.6
0]);
    set(handles.Grafico1, 'Color', [1 1 1], 'Gridcolor', [0.5 0.5 0.5])
    hold on;
    grid on;

    while 1
        tic;
        leitura = a.analogRead(1);

        PV(count) = leitura*0.11097;
        seg(count) = time;

        time = time+timeInterval;
        count = count+1;

        xlim(curva, [min(0) max(time)]); %Ajuste do eixo x no tempo
        ylim(curva, [min(0) max(100)]);
        set(grafico, 'YData', PV, 'XData', seg);

        pause(0.00001);
        t = toc;

        while t < 0.05
            t = toc;
        end

        if parar == true
            parar = false;
            break;
        end
        toc;
    end
end

function parar_Callback(hObject, eventdata, handles)
global parar;
set(handles.iniciar, 'enable', 'on')
set(handles.parar, 'enable', 'off')
set(handles.desconectar_planta, 'enable', 'on')
set(handles.malha_fechada, 'enable', 'on')
set(handles.malha_aberta, 'enable', 'on')
cla %limpa gráfico
set(handles.Grafico1, 'Color', [1 1 1])
parar = true;

function conectar_planta_Callback(hObject, eventdata, handles)
global a;
global Porta;
delete(a);
set(handles.conectar_planta, 'String', 'CONECTANDO...', 'BackgroundColor', [1,0
.2,0.2])
set(handles.desconectar_planta, 'enable', 'off')
a = arduino(Porta);
set(handles.conectar_planta, 'String', 'CONECTADO!', 'BackgroundColor', [0.4,1,
0.4])
set(handles.desconectar_planta, 'enable', 'on')
set(handles.conectar_planta, 'enable', 'off')
set(handles.malha_fechada, 'enable', 'on')
set(handles.malha_aberta, 'enable', 'on')

```

```

set(handles.atualizar_valores,'enable','on')
set(handles.iniciar,'enable','on')

function desconectar_planta_Callback(hObject, eventdata, handles)
global a;
global SP;
global Kp;
global Ki;
global Kd;
delete(a);
set(handles.conectar_planta,'String','CONECTAR','BackgroundColor',[0.94,0.94,0.94])
set(handles.conectar_planta,'enable','on')
set(handles.SP_value,'enable','off')
set(handles.Kp_value,'enable','off')
set(handles.Ki_value,'enable','off')
set(handles.Kd_value,'enable','off')
set(handles.atualizar_valores,'enable','off')
set(handles.iniciar,'enable','off')
set(handles.desconectar_planta,'enable','off')
SP = 0;
Kp = 0;
Ki = 0;
Kd = 0;
set(handles.SP_value,'String',' ')
set(handles.Kp_value,'String',' ')
set(handles.Ki_value,'String',' ')
set(handles.Kd_value,'String',' ')
cla %limpa gráfico
set(handles.Grafico1,'Color',[0 0 0])
grid off;
set(handles.malha_fechada,'enable','off')
set(handles.malha_aberta,'enable','off')

function selecionar_porta_Callback(hObject, eventdata, handles)
global Porta;
contents = get(handles.selecionar_porta,'String');
Porta = contents{get(handles.selecionar_porta,'Value')};
set(handles.conectar_planta,'enable','on')

function selecionar_porta_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function malha_fechada_Callback(hObject, eventdata, handles)
    set(handles.sp_mv,'string','SP (%)')
    set(handles.SP_value,'enable','on')
    set(handles.Kp_value,'enable','on')
    set(handles.Ki_value,'enable','on')
    set(handles.Kd_value,'enable','on')

function malha_aberta_Callback(hObject, eventdata, handles)
    set(handles.sp_mv,'string','MV (%)')
    set(handles.SP_value,'enable','on')
    set(handles.Kp_value,'enable','off')
    set(handles.Ki_value,'enable','off')
    set(handles.Kd_value,'enable','off')

```