

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA  
DEPARTAMENTO ACADÊMICO DE MECÂNICA  
CURSO SUPERIOR DE TECNOLOGIA EM MECATRÔNICA INDUSTRIAL

CLEVERTON BAZAN BERTANHA  
GUILHERME LUIZ SZALBOT

**DESENVOLVIMENTO DE UM SISTEMA DE  
CONTROLE TIPO BOLA-BARRA**

TRABALHO DE CONCLUSÃO DE CURSO

CURITIBA  
2016

CLEVERTON BAZAN BERTANHA  
GUILHERME LUIZ SZALBOT

**DESENVOLVIMENTO DE UM SISTEMA DE  
CONTROLE TIPO BOLA-BARRA**

Trabalho de Conclusão de Curso de Graduação, apresentado ao Curso Superior de Tecnologia em Mecatrônica Industrial, dos Departamentos Acadêmicos de Eletrônica e Mecânica, da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para obtenção do título de Tecnólogo.

Orientador: Prof. M.Sc. Anderson Levati Amoroso.

CURITIBA  
2016

## **TERMO DE APROVAÇÃO**

CLEVERTON BAZAN BERTANHA  
GUILHERME LUIZ SZALBOT

### **DESENVOLVIMENTO DE UM SISTEMA DE CONTROLE TIPO BOLA-BARRA**

Este trabalho de conclusão de curso foi apresentado no dia 06 de novembro de 2016, como requisito parcial para obtenção do título de Tecnólogo em Mecatrônica Industrial, outorgado pela Universidade Tecnológica Federal do Paraná. Os alunos foram arguidos pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

---

Prof. Dr. Milton Luiz Polli  
Coordenador de Curso  
Departamento Acadêmico de Mecânica

---

Prof. M.Sc. Sérgio Moribe  
Responsável pela Atividade de Trabalho de Conclusão de Curso  
Departamento Acadêmico de Eletrônica

#### **BANCA EXAMINADORA**

---

Prof. Ubiradir Mendes Pinto  
UTFPR

---

Prof. Dr. Valmir de Oliveira  
UTFPR

---

Prof. M.Sc. Anderson Levati Amoroso  
Orientador – UTFPR

“A Folha de Aprovação assinada encontra-se na Coordenação do Curso”

## RESUMO

BERTANHA, Cleverton Bazan; SZALBOT, Guilherme Luiz. **Desenvolvimento de um sistema de controle tipo bola-barra**. 2016. 66 f. Trabalho de Conclusão de Curso (Curso Superior de Tecnologia em Mecatrônica Industrial) – Departamentos Acadêmicos de Eletrônica e Mecânica, Universidade Tecnológica Federal do Paraná. Curitiba, 2016.

Este trabalho tem por objetivo o desenvolvimento de um dispositivo para sistema de controle do tipo bola-barra podendo ser utilizado nas atividades práticas de controle de processos. O projeto consiste em um dispositivo capaz de equilibrar uma bola sobre uma barra usando o controle PID. O centro da barra é fixado no eixo do servomotor, o que permite alterar a inclinação da barra, fazendo a esfera rolar livremente sobre a mesma. A função do controlador é manter a bola equilibrada em qualquer ponto sobre a barra dentro de uma faixa definida. O dispositivo possui um sensor ultrassônico que realiza a leitura da posição da bola e envia essa informação ao microcontrolador.

**Palavras-chave:** Sistema bola-barra. Controle PID. Microcontrolador.

## ABSTRACT

BERTANHA, Cleverton Bazan; SZALBOT, Guilherme Luiz. **Development of a ball and beam control system**. 2016. 66 f. Trabalho de Conclusão de Curso (Curso Superior de Tecnologia em Mecatrônica Industrial) – Departamentos Acadêmicos de Eletrônica e Mecânica, Universidade Tecnológica Federal do Paraná. Curitiba, 2016.

This work aims to develop a ball-and-beam control system that can be used in practical process control activities. The system consists of an electromechanical device able to control the position of a small ball on a bar using PID control. The center of the bar is fixed to the servomotor shaft, which allows to change the inclination of the bar, causing the ball to roll freely on it. The control function is to keep the ball balanced at any point on the bar within a defined range. The device also has an ultrasonic sensor that is responsible for reads the ball position and sends this information to the microcontroller.

**Keywords:** Ball and beam system. PID Control. Microcontroller.

## LISTA DE FIGURAS

Figura 1 - Sistema bola-barra.....	14
Figura 2 - Sistema de controle em malha aberta.....	16
Figura 3 - Sistema de controle em malha fechada .....	17
Figura 4 - Sistema de controle de malha fechada com perturbações .....	17
Figura 5 - Gráfico da ação proporcional .....	20
Figura 6 - Gráfico da ação integral .....	21
Figura 7 - Gráfico da ação proporcional e integral .....	22
Figura 8 - Componentes internos do servomotor .....	24
Figura 9 - Reflexão de ondas sonoras do sensor.....	27
Figura 10 - Microcontroladores.....	28
Figura 11 - Arduino Uno .....	30
Figura 12 - Base principal.....	32
Figura 13 - Barra .....	33
Figura 14 - Suporte do motor .....	33
Figura 15 - Montagem do motor .....	34
Figura 16 - Suporte da barra .....	35
Figura 17 - Suporte do sensor.....	35
Figura 18 - Montagem do conjunto mecânico .....	36
Figura 19 - Diagrama em blocos .....	37
Figura 20 - Arduino Mega.....	38
Figura 21 - Placa LCD <i>Keypad Shield</i> .....	39
Figura 22 - Associação de resistores do módulo LCD .....	39
Figura 23 - Sensor ultrassônico HC-SR04 .....	40
Figura 24 - Diagrama de funcionamento do sensor ultrassônico .....	40
Figura 25 - Servomotor TowerPro MG995 .....	41
Figura 26 - Fluxograma do processo.....	43
Figura 27 - Inicialização do <i>firmware</i> .....	44
Figura 28 - Rotina do <i>setup</i> .....	45
Figura 29 - Sistema de calibração.....	46
Figura 30 - Detecção dos botões do módulo LCD.....	47
Figura 31 - Seleção dos parâmetros .....	48
Figura 32 - Normalização do valor lido pelo sensor .....	49
Figura 33 - Rotina para parar a interrupção .....	49
Figura 34 - Função que para a interrupção .....	50
Figura 35 - Primeira barra utilizada .....	51
Figura 36 - Primeiro protótipo do sistema .....	52
Figura 37 - Segundo protótipo do sistema .....	53
Figura 38 - Função para ler a posição da esfera.....	54
Figura 39 - Primeira rotina de controle .....	56
Figura 40 - Cálculo do erro do controlador.....	56
Figura 41 - Cálculo do sinal de controle do sistema.....	57

## LISTA DE SIGLAS E ACRÔNIMOS

CNC	Comando Numérico Computadorizado
EEPROM	<i>Electrically Erasable Programmable Read-Only Memory</i>
EVA	Etil Vinil Acetato
GPS	<i>Global Positioning System</i>
I/O	<i>Input/Output</i>
IDE	<i>Integrated Development Enviroment</i>
IDI	<i>Interaction Design Institute</i>
LCD	<i>Liquid Crystal Display</i>
LED	<i>Light Emmiting Diode</i>
PD	Proporcional-Derivativo
PI	Proporcional-Integral
PID	Proporcional-Integral-Derivativo
PWM	<i>Pulse Width Modulation</i>
RC	Rádio-Controle
USB	<i>Universal Serial Bus</i>

# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>8</b>
1.1	PROBLEMA	9
1.2	OBJETIVOS	10
1.2.1	Objetivo geral	10
1.2.2	Objetivos específicos	10
1.3	JUSTIFICATIVA	10
1.4	PROCEDIMENTOS METODOLÓGICOS	11
1.5	EMBASAMENTO TEÓRICO	11
1.6	ESTRUTURA DO TRABALHO	12
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>13</b>
2.1	SISTEMA BOLA-BARRA	13
2.2	SISTEMAS DE CONTROLE	15
2.2.1	Classificação dos sistemas de controle	16
2.3	AÇÕES DE CONTROLE	18
2.3.1	Controlador proporcional (P)	19
2.3.2	Controlador proporcional-integral (PI)	20
2.3.3	Controle proporcional-derivativo (PD)	22
2.4	SERVOMOTOR TIPO RÁDIO-CONTROLE	23
2.5	SENSORES	25
2.5.1	Sensor ultrassônico	26
2.6	MICROCONTROLADORES	27
2.7	A PLATAFORMA ARDUINO	29
<b>3</b>	<b>METODOLOGIA E DESENVOLVIMENTO</b>	<b>31</b>
3.1	DESENVOLVIMENTO MECÂNICO	31
3.1.1	Base retangular	32
3.1.2	Barra	32
3.1.3	Suporte do motor	33
3.1.4	Suporte da barra	34
3.1.5	Suporte do sensor	35
3.1.6	Montagem completa do sistema mecânico	36
3.2	CIRCUITO ELETRÔNICO	36
3.2.1	Controlador	37
3.2.2	Interface com o usuário	38
3.2.3	Aplicação do sensor ultrassônico	40
3.2.4	Atuador	41
3.2.5	Utilização das entradas e saídas do microcontrolador	41
3.3	CONTROLE	42
<b>4</b>	<b>APRESENTAÇÃO E ANÁLISE DOS RESULTADOS</b>	<b>51</b>
4.1	ENSAIOS DO CONJUNTO MECÂNICO	51
4.2	TESTES DO CIRCUITO ELETRÔNICO E DE CONTROLE	53
<b>5</b>	<b>CONSIDERAÇÕES FINAIS</b>	<b>58</b>
	REFERÊNCIAS	60
	APÊNDICE A – DESENHO 2D: SUPORTE DO MOTOR	63
	APÊNDICE B – DESENHO 2D: SUPORTE DA BARRA	64
	APÊNDICE C – DESENHO 2D: BASE	65



# 1 INTRODUÇÃO

O controle automático apresenta um papel muito importante para a engenharia e para a ciência. Ele está presente em veículos espaciais, em processos de guiamento de mísseis, sistemas robóticos e similares. "O controle automático tornou-se uma parte importante e integrante dos processos industriais e de manufatura modernos" (OGATA, 1998, p. 1), afinal esta tecnologia tem sido empregada para melhorar a produtividade dos processos industriais através do aumento da qualidade dos produtos, da confiabilidade e do nível de segurança empregados no sistema, da diminuição da necessidade de retrabalho dos produtos, além de não necessitar mais de um operador para realizar atividades manuais e repetitivas (CAMPOS, TEIXEIRA, 2006, p. 3).

Os primeiros trabalhos significativos de controle automático foram o de James Watt, no século XVIII, com seu controle centrífugo utilizado no controle de velocidade de uma máquina a vapor. Em seguida, no ano de 1922, Minorsky trabalhou em controladores automáticos para pilotar navios mostrando que era possível determinar a estabilidade do sistema usando equações diferenciais. Em 1932, Nyquist desenvolveu um procedimento usado para determinar a estabilidade em sistemas de malha fechada. Logo depois, em 1934, Hazen introduziu o termo servomecanismos para designar sistemas de controle de posição (JÚNIOR, 2013).

Vários conceitos e definições a respeito do tema foram propostos no decorrer dos anos. Para Nise (2012), o sistema de controle consiste em subsistemas, processos ou plantas construídos com o objetivo de se obter uma saída desejada com um desempenho adequado, dada uma entrada especificada. Esses sistemas podem ser usados em diversos processos dentro da indústria, podendo ser empregados nos setores automobilísticos, aéreos e espaciais, na medição de pressão temperatura e vazão, entre outros. O controle de sistemas modernos e complexos é um desafio destinado para a engenharia de controle. Sua aplicação se estende a sistemas de controle de tráfego, processos químicos, robótica e automação industrial. "O maior desafio para a engenharia de controle está na oportunidade de controlar máquinas, processos industriais e econômicos para benefício da sociedade" (DORF; BISHOP, 2013, p. 1).

Sistema bola-barra é um sistema de controle de equilíbrio onde uma esfera deve permanecer em equilíbrio sobre uma barra com guias laterais. Na barra, existe um ponto fixo ligado a um eixo de rotação acionado por um servomotor. Tal controle deve ser realizado automaticamente após pré-determinar uma posição para a bola. O desafio deste princípio está em como controlar o sistema, visto que a bola é deslocada com a inclinação da barra, ou então, a mesma é colocada em uma posição que não seja a pré-determinada (distúrbios). Para o controle deste sistema podem ser utilizados as seguintes técnicas: Controle Proporcional, Integral e Derivativo (PID), Controle Neural, Lógica Fuzzy, entre outros.

O desenvolvimento do dispositivo tem como objetivo a sua utilização para fins didáticos proporcionando o aprendizado prático em laboratórios ou em universidades já que na maioria das vezes, os sistemas de controle instáveis reais são de grande porte ou perigosos e não podem estar presentes nesses ambientes. O sistema bola-barra possui as mesmas características de um sistema instável. Além disso, as universidades, na maioria das vezes, não contam com mecanismos capazes de representar a aplicação prática de sistemas de controle.

## 1.1 PROBLEMA

Os sistemas de controle reais quando classificados como instáveis e perigosos ou por possuírem um custo elevado não são encontrados facilmente em laboratórios universitários. Entretanto, sistemas mais simples podem substituir esses processos complexos e de alto risco simulando suas propriedades e seus efeitos. O sistema de controle do tipo bola-barra representa muito bem os processos instáveis pois possui as mesmas características destes processos. Entretanto, os laboratórios do Departamento Acadêmico de Eletrônica da Universidade Tecnológica Federal do Paraná, no campus Curitiba, não apresentam nenhum conjunto didático para estudo deste tipo de sistema. O desafio deste trabalho está em desenvolver um sistema do tipo bola-barra com fins didáticos para auxiliar no ensino das disciplinas que envolvem sintonia de controle.

## 1.2 OBJETIVOS

### 1.2.1 Objetivo geral

Desenvolver um sistema de controle didático baseado em um mecanismo tipo bola-barra.

### 1.2.2 Objetivos específicos

- a) Arquitetar a estrutura mecânica;
- b) Elaborar o circuito eletrônico;
- c) Desenvolver o *firmware* de operação do sistema;
- d) Projetar o sistema de controle do mecanismo;
- e) Desenvolver a interface com o usuário em um microcontrolador.

## 1.3 JUSTIFICATIVA

Geralmente, os sistemas instáveis reais são perigosos e não podem ser estudados em laboratórios que não estejam adequados para este propósito. O sistema bola-barra foi desenvolvido para resolver este problema porque é um mecanismo simples, seguro e que tem as mesmas características de um sistema instável, visto que “sistema estável é um sistema dinâmico com uma resposta limitada para uma entrada limitada” (DORF; BISHOP, 2013, p. 296). Ou seja, um sistema é considerado instável se a saída diverge sem limites a partir do estado de equilíbrio quando o sistema é submetido a uma condição inicial.

A ideia principal é disponibilizar aos alunos uma abordagem prática para auxiliar no aprendizado de sistemas de controle, visto que o sistema bola-barra pode simular sistemas de controle instáveis reais. As atividades práticas facilitam o

entendimento da disciplina aplicando os conteúdos teóricos lecionados dentro das salas de aula em um dispositivo físico.

#### 1.4 PROCEDIMENTOS METODOLÓGICOS

Para o desenvolvimento do sistema de controle bola-barra foram definidas algumas etapas a fim de se alcançar o resultado final deste projeto. O primeiro passo é pesquisar a respeito dos métodos de controle existentes, o funcionamento do sistema bola-barra, conceitos de programação de microcontroladores e o tipo de sensor que melhor se ajusta a aplicação que será designada. Este estudo servirá para adquirir o embasamento teórico para a realização deste projeto.

A segunda etapa será a aquisição dos elementos mecânicos e dos componentes eletrônicos para o trabalho. Nesta etapa a equipe definiu que os itens comprados deverão ser comerciais, para facilitar eventuais necessidades de trocas dos componentes empregados no protótipo. Logo que a equipe tenha obtido estes itens começará a construção da parte mecânica e eletrônica do sistema, com a integração do sistema na sequência.

A partir da construção do sistema físico do projeto será implementado o *firmware* de controle no microcontrolador. Concluída esta etapa, será realizada a sintonia do controle através dos resultados obtidos e dos parâmetros calculados, com a finalização desta etapa ocorrendo somente quando o sistema estiver equilibrando a bola na posição determinada na barra.

#### 1.5 EMBASAMENTO TEÓRICO

Para a produção deste trabalho serão realizadas diversas pesquisas sobre o sistema bola-barra, sistemas de controle, ações de controle, microcontroladores e, em especial, o Arduino, sensores e servomotores. No que diz respeito ao sistema bola-barra, as principais referências serão Wang (2007) e Wellstead (2000). Em relação aos assuntos pertinentes aos sistemas e ações de controle os principais

referenciais teóricos serão Ogata (1998), Dorf e Bishop (2013) e Campos e Teixeira (2006).

Quanto ao tema envolvendo servomotores serão consultados Medeiros (2011) e Mantovani (2013). Já para o assunto de sensores serão analisados as publicações de Thomazini e Albuquerque (2007) e Bega et al (2011). No que tange aos microcontroladores, as principais referências serão Souza (2010), Martins (2005) e Denardin (2011). Para compreender os conceitos do Arduino serão utilizados como referencial teórico Evans, Noble e Hochenbaum (2013) e McRoberts (2011).

## 1.6 ESTRUTURA DO TRABALHO

A estrutura deste trabalho está apresentada abaixo:

**Capítulo 1 - Introdução:** serão apresentados o tema, o problema, os objetivos deste trabalho, a justificativa, os procedimentos metodológicos, as referências que servem para o embasamento teórico e a estrutura geral do trabalho.

**Capítulo 2 - Fundamentação Teórica:** serão abordados os conceitos do sistema bola-barras, sistemas de controle, ações de controle, servomotores, sensores, microcontroladores e Arduino.

**Capítulo 3 - Metodologia e Desenvolvimento:** neste item serão descritos o desenvolvimento da estrutura mecânica, do circuito eletrônico e do sistema de controle do projeto.

**Capítulo 4 - Apresentação e Análise de Resultados:** neste capítulo serão discutidos as análises feitas em cima do sistema bola-barras e os resultados obtidos durante o desenvolvimento do projeto.

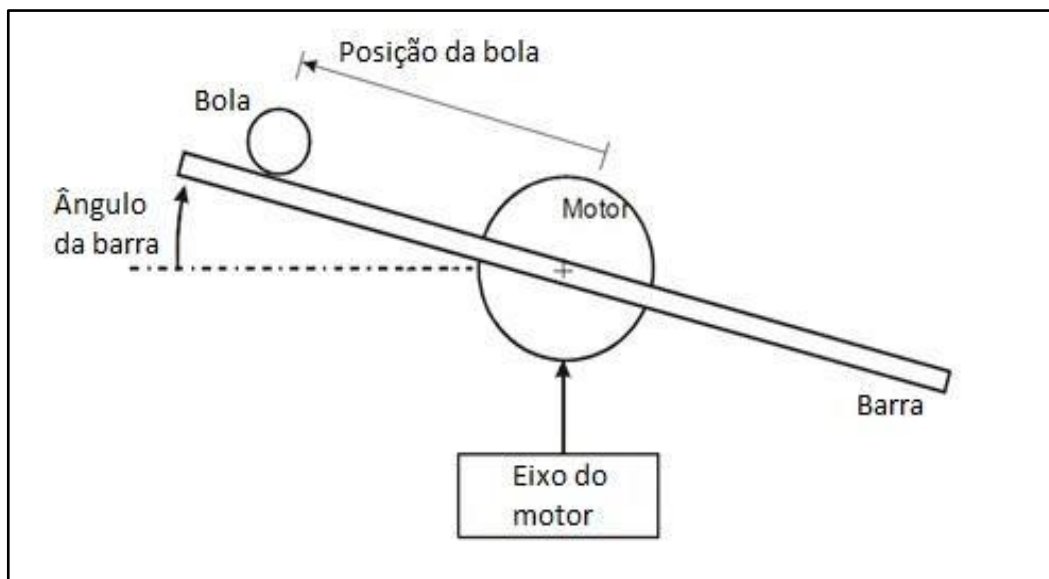
**Capítulo 5 - Considerações Finais:** nesta seção será analisado se os objetivos da pesquisa foram atingidos e o meio pelo qual esses resultados foram obtidos. Também serão apresentados eventuais problemas que surgiram durante o desenvolvimento do trabalho, bem como eventuais melhorias que podem ser adicionadas neste sistema a partir desta pesquisa.

## 2 FUNDAMENTAÇÃO TEÓRICA

Para o desenvolvimento deste projeto foi necessário a compreensão dos conceitos que envolvem o sistema bola-barra; os tipos e definições dos sistemas de controle existentes; as características que cada ação de controle possui funcionando separadas ou integradas; e os conceitos e funcionalidades referentes a servomotores, sensores ultrassônicos e ao Arduino. Todos esses temas serão abordados na sequência deste capítulo.

### 2.1 SISTEMA BOLA-BARRA

O sistema bola-barra pode ser encontrado em muitos laboratórios universitários por ser um dos modelos mais importantes e populares no ensino da engenharia de sistemas de controle (WELLSTEAD, 2000). Este modelo apresenta características que o tornam uma ferramenta educacional efetiva para o ensino dos princípios de realimentação de controle (CHOI; EVANKO; DORSETT, 2008). O principal objetivo deste modelo é controlar a bola em uma determinada posição na barra através da posição angular do motor e, também, rejeitar os distúrbios sofridos como, por exemplo, o empurrão de um dedo na esfera (SCALABRIN et al, 2014). Este conjunto possui dois graus de liberdade: um é a bola rolando para cima e para baixo na barra e o outro é a haste girando em torno do eixo central do motor (WANG, 2007), como pode ser verificado na Figura 1.



**Figura 1 - Sistema bola-barras**  
**Fonte: Adaptado de Wellstead (2000).**

A posição da bola na barra pode ser medida através de um sensor (WELLSTEAD, 2000). Esse sensor se comunica com o sistema através de um sinal de controle de realimentação cuja função é informar a posição da bola na barra (WANG, 2007).

A tarefa que este sistema deve executar é controlar automaticamente a posição da esfera na haste alterando o ângulo da barra, não sendo esta atividade de fácil execução, pois a bola não permanece em um único local na barra, mas move-se com uma aceleração proporcional ao ângulo da haste. Este sistema em malha aberta é considerado instável porque o sinal de saída – a posição da bola – aumenta ilimitadamente para uma entrada fixa, que é o ângulo da barra (WELLSTEAD, 2000).

O desenvolvimento do sistema bola-barras auxiliou e continua a ajudar o estudo dos sistemas instáveis, os quais devem ser levados a laboratório para análise, pois estudar os seus modelos reais é perigoso. Os sistemas instáveis estão presentes em muitos processos industriais e tecnológicos modernos, dentre os quais se podem citar as indústrias de processos químicos através do processo de controle de reações exotérmicas, na geração de energia e no controle horizontal de um foguete ou aeronave (WELLSTEAD, 2000). Para Wang (2007), um exemplo de sistema instável é "a estabilização horizontal de uma aeronave durante o pouso".

## 2.2 SISTEMAS DE CONTROLE

O controle automático está presente em diversos segmentos da engenharia e da ciência. Ele está envolvido nos sistemas robóticos e de guiamentos de mísseis, além de ser parte integrante nos processos industriais e de manufatura modernos como, por exemplo, no comando de máquinas-ferramentas e no projeto de caminhões e automóveis. O controle automático também está presente em operações de controle de pressão, temperatura, vazão, entre outros processos (OGATA, 1998, p. 1).

Um sistema de controle pode ser explicado como "uma interconexão de componentes formando uma configuração de sistema que proporcionará uma resposta desejada do sistema" (DORF; BISHOP, 2013, p. 2). A teoria de sistemas lineares fornece toda a base para o estudo de um sistema de controle, já que ela propõe uma relação entre a entrada e a saída deste modelo, ou seja, é a relação entre a causa e o efeito do processo, é o processo do sinal de entrada para emitir um sinal de saída (DORF; BISHOP, 2013, p. 2).

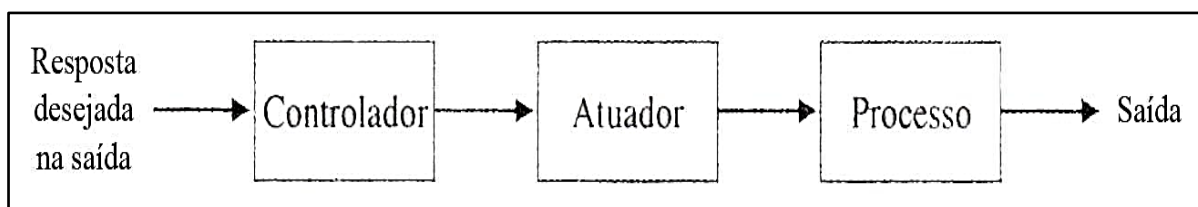
Os termos básicos de sistemas de controle serão definidos abaixo para a melhor compreensão deste tema no decorrer da pesquisa, conforme Ogata (1998, p. 2):

- Variável controlada: é a grandeza medida e controlada, normalmente é o valor de saída do sistema;
- Variável manipulada: é a grandeza modificada pelo controlador que afeta o valor da variável controlada;
- Sistemas a controlar (planta): é um equipamento, ou parte dele, que desempenha uma função e será controlado;
- Processo: é toda operação controlada;
- Sistemas: são componentes combinados entre si que realizam certo objetivo;
- Distúrbios: é um sinal que pode interferir negativamente na variável de saída do sistema. Pode ser classificado como distúrbio interno ou externo.



### 2.2.1 Classificação dos sistemas de controle

Os sistemas de controle podem ser classificados como sistema de controle de malha aberta e de malha fechada (OGATA, 1998, p. 5). O sistema de controle de malha aberta utiliza um controlador e um atuador para controlar o processo, sem necessitar do sinal de realimentação (BISHOP; DORF, 2013, p. 2). Neste tipo de sistema o sinal de saída não é medido e, conseqüentemente, o seu valor não é comparado com o sinal de entrada (OGATA, 1998, p. 5), como pode ser visualizado na Figura 2.

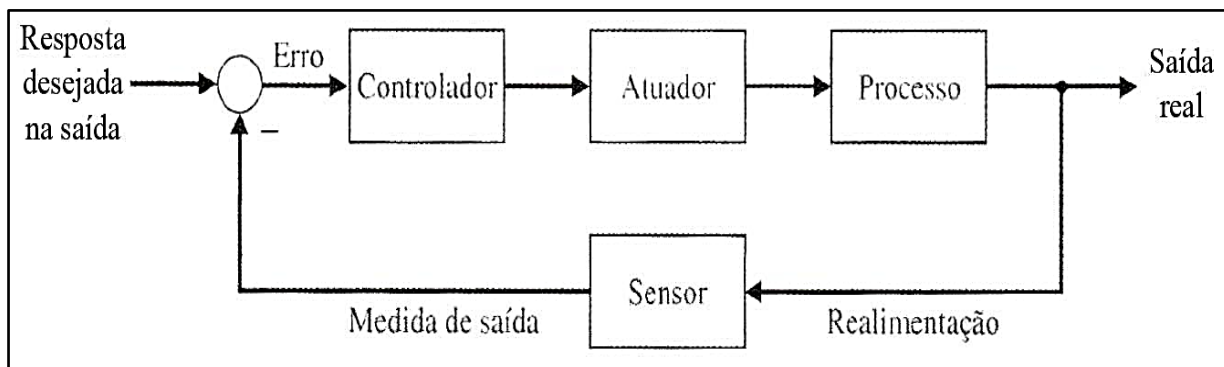


**Figura 2 - Sistema de controle em malha aberta**  
Fonte: Adaptado de Dorf e Bishop (2013, p. 2).

Como não existe a realimentação, a resposta do sinal de saída gerado está ligada diretamente ao sinal de entrada (DORF; BISHOP, 2013, p. 180). O controle de malha aberta só é utilizado na prática porque é simples e barato e quando a relação entre os sinais de saída e de entrada do processo são conhecidos, pois este tipo de controle não funciona corretamente na presença de distúrbios, sejam eles internos ou externos (OGATA, 1998, p. 5).

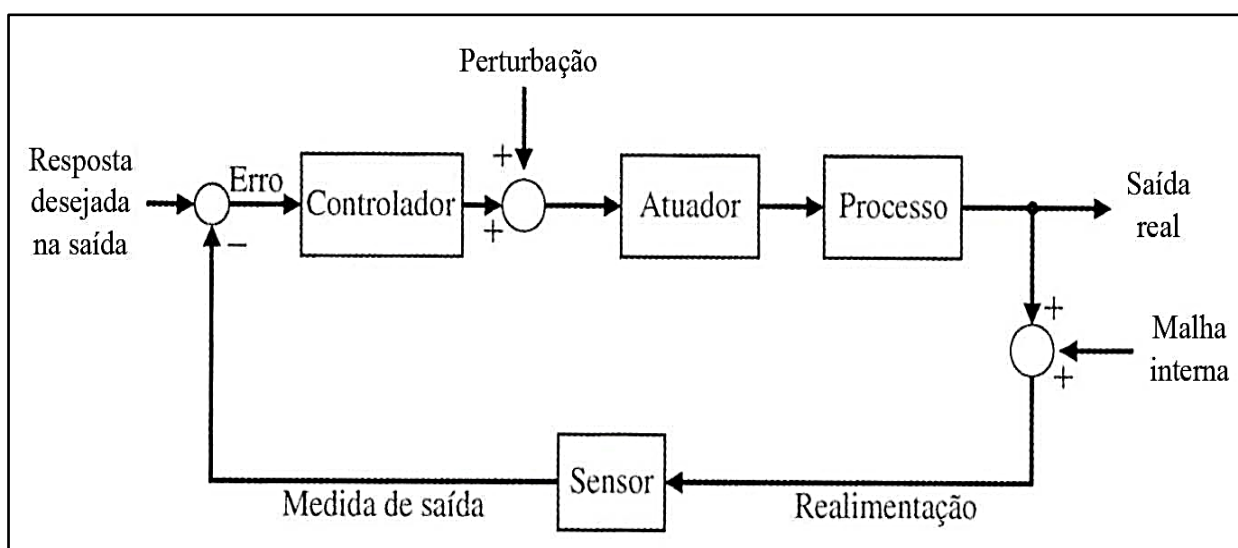
A ideia principal de um sistema de controle de malha fechada é retransmitir o valor do sinal de saída do sistema para o controlador utilizado para comandar o sistema (FRANKLIN; POWELL; EMAMI-NAEINI, 2013, p. 1). O sistema de controle com malha fechada simples utiliza uma relação pré-determinada entre o valor de saída medido e o valor da entrada de referência para comandar o processo. Usualmente, essa diferença obtida é usada para controlar todo o processo, de tal forma que possa ser reduzida sucessivamente essa diferença (DORF; BISHOP, 2013, p. 2). Essa diferença é o sinal de erro, obtido através da diferença entre o sinal de entrada e o sinal de realimentação, que incita o controlador para que este amenize o erro e traga o valor do sinal de saída para o valor pretendido (OGATA, 1998, p. 5). Resumidamente, "um sistema em malha fechada usa uma medida do

sinal de saída e uma comparação com a saída desejada para gerar um sinal de erro que é usado pelo controlador para ajustar o atuador" (DORF; BISHOP, 2013, p. 180). A Figura 3 exemplifica um sistema de controle de malha fechada.



**Figura 3 - Sistema de controle em malha fechada**  
 Fonte: Adaptado de Dorf e Bishop (2013, p. 2).

O sistema de controle em malha fechada apresenta algumas vantagens em relação ao controle de malha aberta. Uma dessas vantagens é a capacidade do sistema de rejeitar distúrbios externos e de atenuar o ruído de medida, que são perturbações presentes em aplicações reais (DORF; BISHOP, 2013, p. 3). Estes parâmetros podem ser visualizados no diagrama em blocos da Figura 4.



**Figura 4 - Sistema de controle de malha fechada com perturbações**  
 Fonte: Adaptado de Dorf e Bishop (2013, p. 3).

Além disso, o controle de malha fechada possui menor sensibilidade a variações nos parâmetros do processo, reduz o erro em regime permanente do

sistema e facilita o controle e ajuste da resposta transitória do sistema (DORF; BISHOP, 2013, p. 181). Porém, os sistemas de controle em malha fechada apresentam problemas quando se refere à questão da estabilidade. Este tipo de controle tende a querer corrigir os erros além do necessário, ocasionando eventualmente oscilações crescentes ou constantes de amplitude com o tempo (OGATA, 1998, p. 6).

Em resumo, pode-se dizer que o uso do controle de malha aberta é indicado quando as referências de entrada do sistema já são conhecidas antecipadamente e quando não há perturbações. Já o controle de malha fechada é adequado quando há algum distúrbio presente ou existam mudanças não previstas anteriormente em parâmetros do sistema (OGATA, 1998, p. 5).

### 2.3 AÇÕES DE CONTROLE

O tipo de controlador mais utilizado na indústria é o proporcional, integral e derivativo (PID). Este tipo de controle destaca-se entre os demais porque o ajuste dos seus parâmetros a fim de se obter um resultado satisfatório do sistema é realizado de forma simples, além do fato deste controlador estar presente em vários equipamentos de controle na indústria (CAMPOS, TEIXEIRA, 2006, p. 23).

O funcionamento do controlador PID ocorre basicamente da seguinte forma: inicialmente há o cálculo do erro entre a variável controlada do processo e o *setpoint* e, após este cálculo, é gerado um sinal de controle para tentar eliminar este erro. O desvio que há entre a variável controlada e o *setpoint* é utilizado de forma distinta nos módulos proporcional (P), integral (I) e derivativo (D). Na prática os controladores mais utilizados são: proporcional (P); proporcional e integral (PI); proporcional e derivativo (PD); e proporcional, integral e derivativo (PID) (CAMPOS, TEIXEIRA, 2006, p. 23).

### 2.3.1 Controlador proporcional (P)

No controlador proporcional a saída gerada é proporcional ao erro e o ganho deste controlador é representado através do fator multiplicativo  $K_p$ . Na equação (1) é possível visualizar o algoritmo do controlador proporcional (CAMPOS, TEIXEIRA, 2006, p. 24).

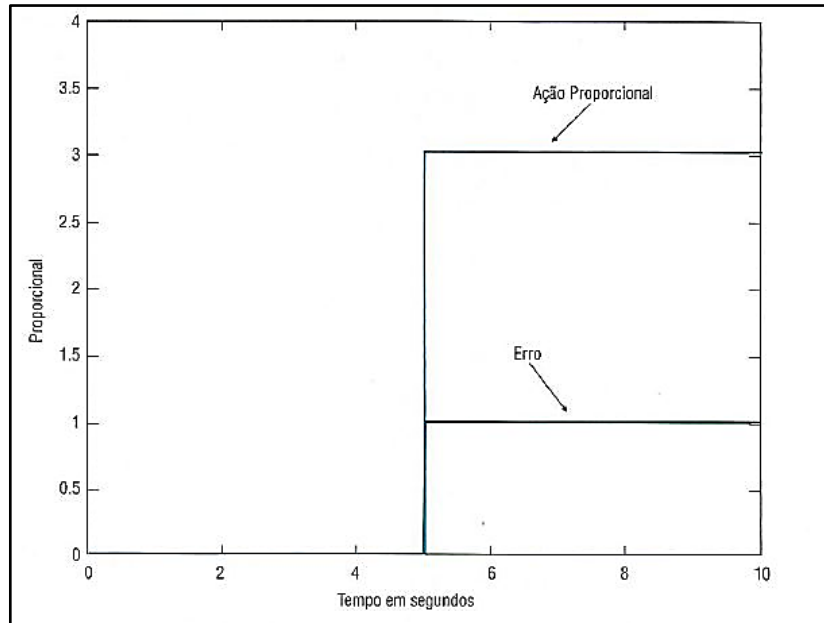
$$u(t) = K_p \cdot e(t) + u_0 \quad (1)$$

Onde:

- $u(t)$ : saída do controlador;
- $K_p$ : ganho proporcional do controlador;
- $e(t)$ : erro calculado;
- $u_0$ : valor inicial do controlador.

Nos controladores industriais há dois tipos de ação: direta ou reversa. A ação direta pode ser definida segundo Campos e Teixeira (2006, p. 24) como aquela que "quando a variável do processo (PV) aumenta, a saída do controlador também aumenta". Já para o caso da ação reversa, a saída do controlador irá diminuir quando a variável do processo aumentar. A escolha da ação do controlador deve ser correta a fim de garantir que o controlador funcione adequadamente. Caso a ação escolhida seja errada, o sistema pode-se tornar instável e não será possível a atuação do controlador em automático (CAMPOS, TEIXEIRA, 2006, p. 25).

A Figura 5 exemplifica a ação do controlador proporcional. Pode-se observar que para um erro em degrau, a ação do controlador também será em degrau. Isto ocorre porque a ação proporcional tem a mesma forma do erro, pois ela é multiplicada apenas pelo ganho do controlador – conforme visto na equação (1). Desta forma, se o erro não variar a saída do controlador também não variará, permitindo assim o erro em regime permanente.



**Figura 5 - Gráfico da ação proporcional**  
**Fonte: Campos e Teixeira (2006, p. 25).**

### 2.3.2 Controlador proporcional-integral (PI)

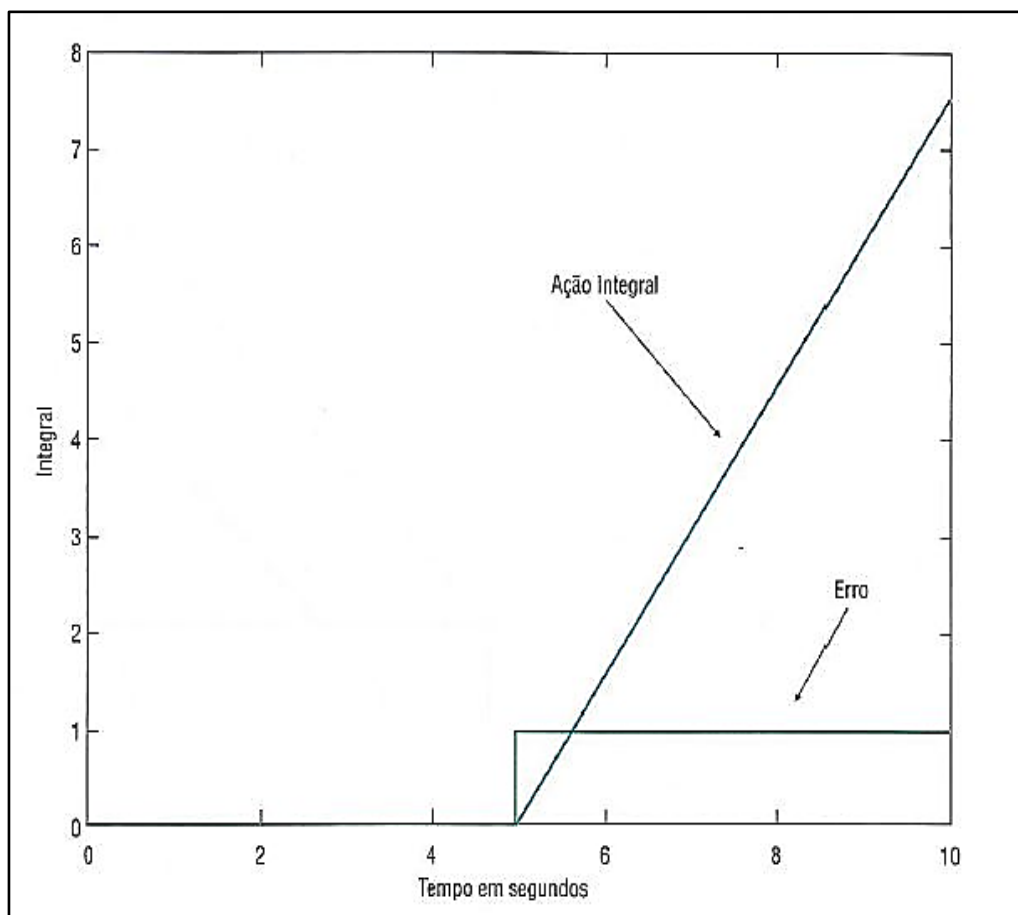
A saída do controlador proporcional e integral gerada é proporcional ao erro – ação proporcional – e à integral do erro, que nada mais é que a ação integral (CAMPOS, TEIXEIRA, 2006, p. 26). Este tipo de controlador pode ser representado pela equação (2) que é de um PI paralelo clássico.

$$u(t) = K_p \cdot e(t) + K_p \cdot \frac{1}{T_i} \cdot \int e(t)dt + u_0 \quad (2)$$

Onde:

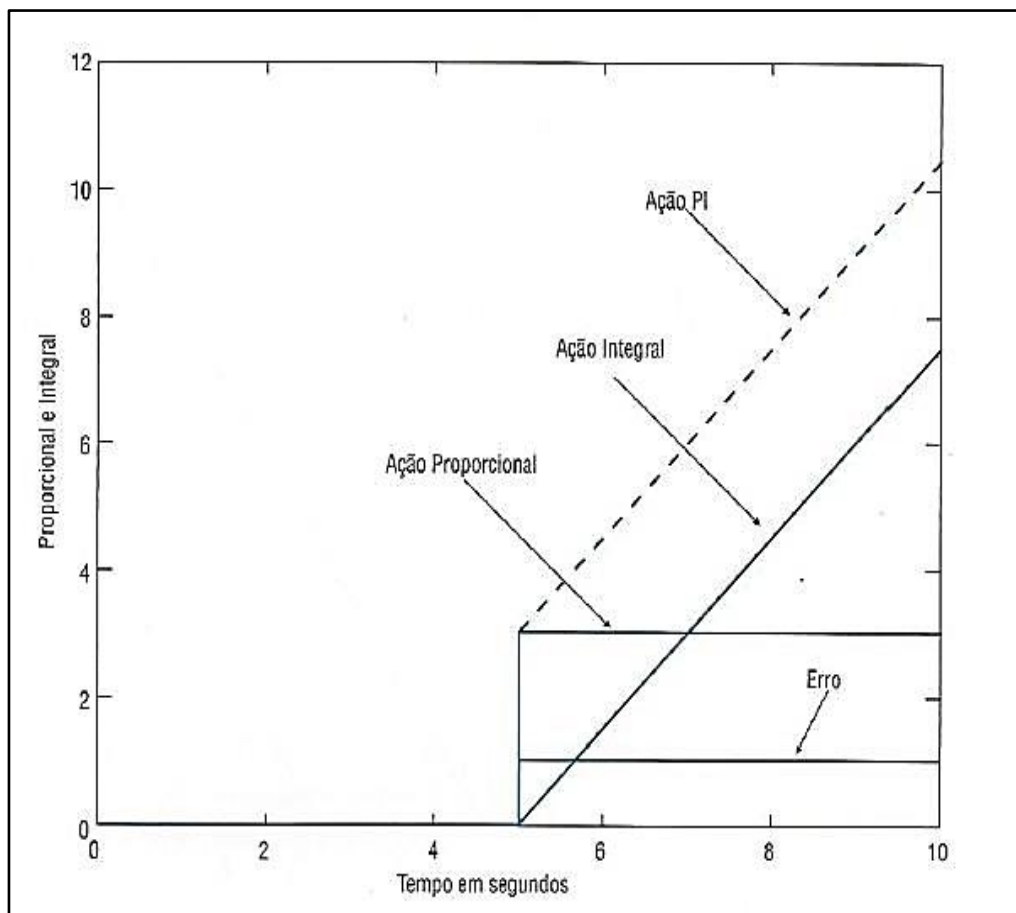
- $u(t)$ : saída do controlador;
- $K_p$ : ganho proporcional do controlador;
- $e(t)$ : erro calculado;
- $T_i$ : tempo integral;
- $1/T_i$ : ganho integral do controlador ou número de repetições por segundo;
- $u_0$ : valor inicial do controlador.

Na Figura 6 pode-se visualizar o que ocorre com a ação integral quando há um erro em degrau. Esta ação será responsável por gerar uma rampa no sinal do controlador, ou seja, ela realizou a "integral" do degrau para chegar até o sinal de rampa. A ação integral funcionará aumentando ou diminuindo o sinal do controlador enquanto o erro existir (CAMPOS, TEIXEIRA, 2006, p. 26).



**Figura 6 - Gráfico da ação integral**  
Fonte: Campos e Teixeira (2006, p. 27).

Já a Figura 7 mostra a ação de controle proporcional e integral. Pode-se observar que a ação proporcional atua no exato momento em que o erro é calculado, contudo, a saída continua sendo alterada devido à ação integral estar agindo enquanto houver o erro. Desta forma, o controle proporcional e integral não aceita erro em regime permanente entre a variável controlada e o *setpoint* do sistema, pois pode haver a saturação do sinal de saída do controlador (CAMPOS, TEIXEIRA, 2006, p. 26).



**Figura 7 - Gráfico da ação proporcional e integral**  
 Fonte: Campos e Teixeira (2006, p. 28).

### 2.3.3 Controle proporcional-derivativo (PD)

Na prática só a ação de controle derivativa não é utilizada, por isso, ela é implementada juntamente com a ação proporcional. Esta ação de controle conjunta permite que o sistema reaja antecipadamente gerando uma ação corretiva mais cedo e é benéfico no regime transitório. Porém, o controle PD aumenta os sinais de ruído e não corrige o erro em estado estacionário (ARAÚJO, 2007). O controlador proporcional e derivativo também está habilitado para evitar as oscilações que ocorrem em processos lentos (CAMPOS, TEIXEIRA, 2006, p. 30).

A equação do controle proporcional e derivativo pode ser visualizada na equação (3) (CAMPOS, TEIXEIRA, 2006, p. 30).

$$u(t) = K_p \cdot e(t) + K_p \cdot T_d \cdot \frac{de(t)}{dt} + u_0 \quad (3)$$

Onde:

- $u(t)$ : saída do controlador;
- $K_p$ : ganho proporcional do controlador;
- $e(t)$ : erro calculado;
- $T_d$ : tempo derivativo;
- $u_0$ : valor inicial do controlador.

## 2.4 SERVOMOTOR TIPO RÁDIO-CONTROLE

Servomotor é um circuito fechado, um servo mecanismo que utiliza realimentação de posição final, permitindo o controle preciso da posição angular medida, como também, a velocidade do eixo de saída pelo envio de um sinal na entrada. (MEDEIROS, 2011). Os servomotores são utilizados quando se necessitam de controle de velocidade, alta precisão de posicionamento, dimensões reduzidas, entre outros fatores (WEG, 2003). Uma das características mais marcantes do servomotor é a capacidade de manter uma posição mesmo quando sofre uma força em direção contrária ao movimento de seu “braço”. Ao tentar alterar a posição do servomotor, verifica-se uma resistência feita pelo motor, a qual é denominada de torque (MANTOVANI, 2013). Este tipo de atuador pode ser encontrado em sistemas de automação, na robótica e em auto e aeromodelismo, onde o posicionamento preciso do servomotor é essencial para o funcionamento destes sistemas (EPUSP, 2014).

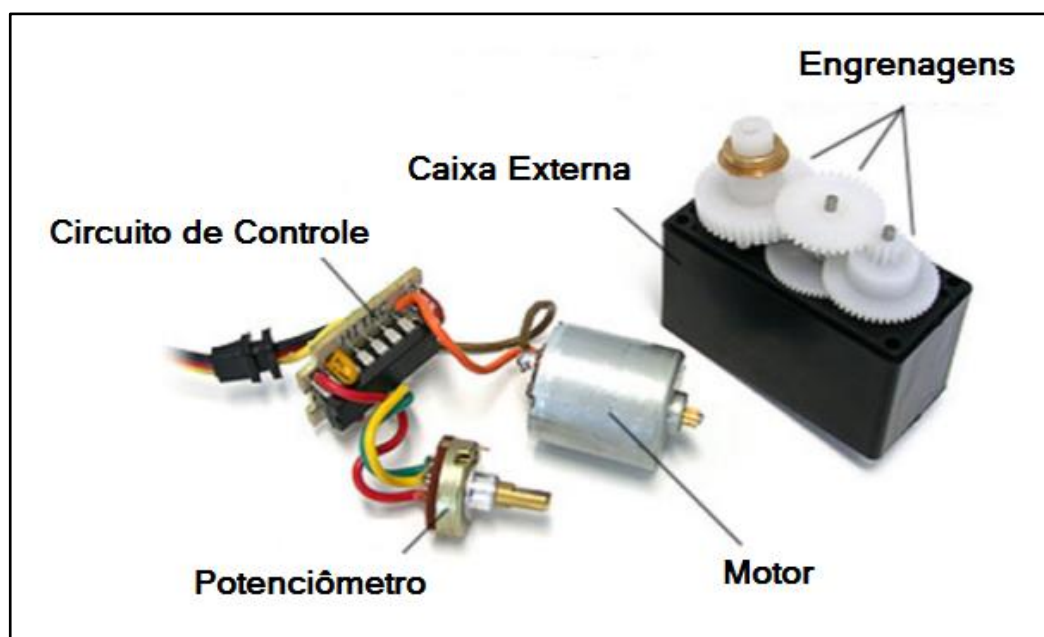
Segundo Mantovani (2013) os servomotores do tipo rádio-controle (RC) podem ser classificados em *standard*, mini e micro. As diferenças entre esses modelos estão no seu peso e tamanho. Os servos *standard* são os maiores e robustos, pesando até 35 g. O modelo mini é menor que o *standard* e pesa entre 20 e 28 g. Já o modelo micro é o menor dos três tipos e pesa de 6 até 20 g. Apesar de



suas pequenas dimensões, os três modelos de servomotores RC apresentam, proporcionalmente, alto torque.

De acordo com Mantovani (2013) os servomotores RC são compostos pelos componentes abaixo e podem ser observados na Figura 8.

- Circuito de controle: é o responsável por monitorar a posição do potenciômetro a acionar o motor para alcançar a posição pré-determinada;
- Potenciômetro: deve monitorar a posição do eixo de saída do servo;
- Motor: responsável por movimentar as engrenagens e o eixo principal do servomotor;
- Engrenagens: realizam a redução da rotação do motor aumentam o torque e movimentam o potenciômetro junto com o eixo.



**Figura 8 - Componentes internos do servomotor**  
Fonte: Adaptado de Mantovani (2013).

O servomotor tipo RC deve ser alimentado com tensão de 5 V e o controle da posição angular do mesmo ocorre com sinais PWM. Este sinal é monitorado em períodos de 20 ms (frequência de 50 Hz) e os pulsos possuem largura de 1 até 2 ms. Quando a largura do pulso é de 1 ms, o servo está na posição 0°, já na posição de 90° a largura do pulso é igual a 1,5 ms e enquanto a largura durar 2 ms a posição

do servo é correspondente a  $180^\circ$ . Desta forma, conforme há variação na largura do pulso, a posição do motor será alterada, variando de 0 até  $180^\circ$  (EPUSP, 2014).

## 2.5 SENSORES

Os sensores podem ser definidos como "dispositivos sensíveis a alguma forma de energia do ambiente que pode ser luminosa, térmica, cinética, relacionando informações sobre uma grandeza que precisa ser medida" (THOMAZINI, ALBUQUERQUE, 2007, p. 17). Essas grandezas medidas podem ser pressão, temperatura, velocidade, posição, etc. A principal função dos sensores é analisar e obter os valores das variáveis físicas do ambiente no qual estão empregados, sejam eles industriais, comerciais, domésticos ou automobilísticos. Porém, após obter esses valores, nem sempre o sensor possui as características necessárias para ser utilizado num sistema de controle, sendo necessária a aplicação de um circuito de interface para que esse sinal produzido possa ser manipulado e lido pelo controlador (THOMAZINI, ALBUQUERQUE, 2007, p. 17).

Basicamente, os sensores podem ser classificados em analógicos e digitais. As saídas dos sensores analógicos possuem a capacidade de assumir qualquer valor ao longo do tempo, desde que esses valores estejam dentro da faixa de operação do sensor. Algumas variáveis que podem ser medidas são: pressão, temperatura, umidade, vazão, distância. Esses valores são calculados por elementos sensíveis com circuitos eletrônicos não digitais (THOMAZINI, ALBUQUERQUE, 2007, p. 18). Em compensação, os sensores digitais apresentam apenas dois valores de saída ao longo do tempo: zero ou um. Este tipo de sensor é utilizado frequentemente em sistemas de detecção de passagem de objetos (THOMAZINI, ALBUQUERQUE, 2007, p. 18).

Algumas características devem ser observadas antes de empregar o sensor em determinado sistema como, por exemplo, o tipo de saída do sensor (digital ou analógica), facilidade de manutenção, custo, calibração, dimensões, faixa de trabalho e vida útil, entre outros (THOMAZINI, ALBUQUERQUE, 2007, p. 23).

### 2.5.1 Sensor ultrassônico

O sensor ultrassônico utiliza as ondas sonoras para calcular a distância do objeto até o sensor. Essas ondas possuem frequência de oscilação maior do que as audíveis pelos seres humanos, ou seja, as ondas ultrassônicas operam numa frequência acima de 20 kHz. O cálculo da distância percorrida pela onda é baseado na velocidade de propagação da onda no ar, onde a mesma alcança uma velocidade de 340 m/s. Caso o meio de propagação não seja o ar, a velocidade deverá ser compensada, levando em conta o meio em que a onda está se propagando (BEGA et al., 2011, p. 177).

A operação dos sensores ultrassônicos tem por base o princípio da lei da ótica física, segundo Bega et al. (2011, p. 177): "o ângulo de incidência da onda é igual ao seu ângulo de reflexão". Devido a esta característica, os objetos medidos podem apresentar qualquer superfície refletiva e podem ser redondos. Outros fatores que influem no funcionamento deste tipo de sensor são: ângulo de superfície e a rugosidade do objeto, mudança da temperatura ou da umidade do ambiente (THOMAZINI, ALBUQUERQUE, 2007, p. 38).

A Figura 9 mostra o funcionamento de um sensor ultrassônico. Ele é baseado no princípio da reflexão das ondas sonoras. Quando a onda ultrassônica encontra uma barreira ou um objeto, ela é refletida e retorna ao sensor, processo conhecido como eco. A referência para a posição do objeto ou da barreira é baseado no tempo decorrido entre o instante em que o sinal é emitido e o instante que o sinal refletido é recebido pelo sensor (THOMAZINI, ALBUQUERQUE, 2007, p. 144).

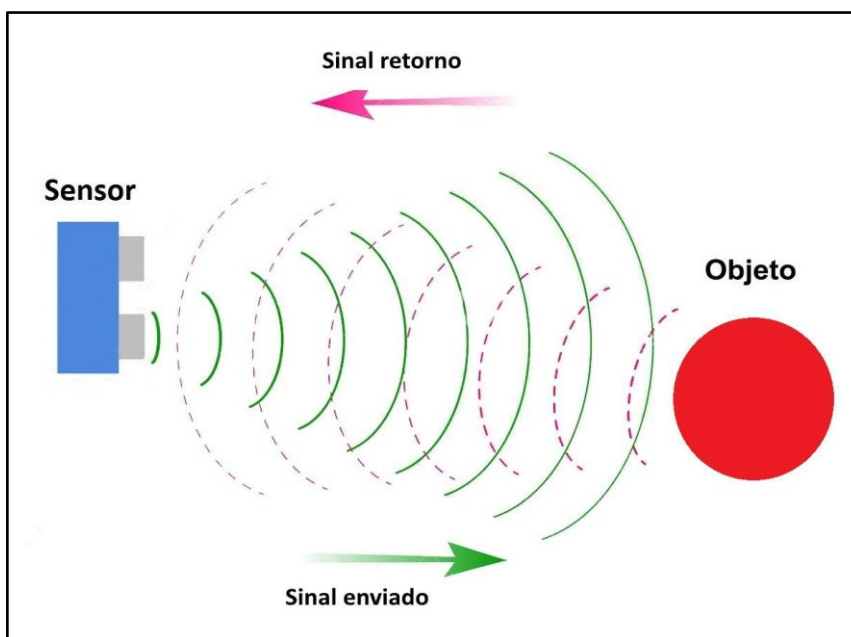


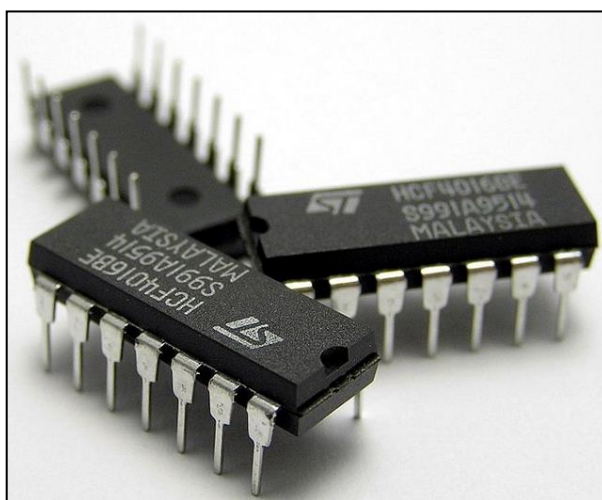
Figura 9 - Reflexão de ondas sonoras do sensor  
Fonte: Adaptado FilipeFlop (2016a).

## 2.6 MICROCONTROLADORES

Os microcontroladores são componentes eletrônicos que podem ser programados para executar tarefas no controle de processos lógicos, que se baseiam nas ações lógicas que devem ser executadas dependendo do estado dos periféricos de entrada e/ou saída, periféricos estes que podem ser LED (*Light Emitting Diode*), *displays* de segmentos, sensores, relês e muitos outros. O microcontrolador é um componente eletrônico programável, sua lógica de operação é estruturada na forma de um programa e gravada dentro do componente e, depois de gravada, cada vez que o microcontrolador for alimentado o programa que está inserido internamente nele será executado. Os microcontroladores possuem um único e pequeno encapsulamento, onde se encontra todos os componentes necessários para controlar um processo, ou seja, é provido, internamente, de memória de programa e dados EEPROM (*Electrically Erasable Programmable Read-Only Memory*), portas de entrada e saída paralelas I/O (*Input/Output*), temporizadores, contadores, interface de comunicação serial, geradores de sinais PWM (*Pulse Width Modulation*), conversores analógicos e digitais (SOUZA, 2010).

Os diversos tipos de microcontroladores existentes no mercado são diferenciados pela quantidade de memória interna de armazenamento de dados, instruções de programas, velocidade de processamento, quantidade de portas configuráveis de entrada e saída, bem como quantidades e tipos de periféricos acoplados, a arquitetura do microcontrolador e finalmente o número de instruções existentes internamente ao microprocessador (MARTINS, 2005). Os microcontroladores também podem ser classificados em famílias, dependendo da aplicação a que se destinam. A partir da aplicação que a família de microcontroladores se destina, um conjunto de periféricos específicos é escolhido e integrado a um determinado microprocessador (DENARDIN, 2011). A popularização dos microcontroladores se deve ao seu baixo custo, baixo consumo de energia, sofisticação e personalização. São utilizados como soluções em projetos onde se buscam um melhor custo benefício e empregados nos mais diversos equipamentos.

Os microcontroladores estão presentes nas mais diversas áreas, dentre as quais citamos a automação industrial, comercial e predial, área automobilística, setor agrícola, produtos manufaturados, eletrodomésticos, telecomunicações, máquinas de escritório e residenciais, controles remotos, brinquedos, sistemas de supervisão, entre outros. No mercado existe uma grande quantidade de microcontroladores tais como: 8051, 8096, 68HC705, 68HC11 e os PICs. A Figura 10 mostra alguns tipos de microcontroladores existentes no mercado.

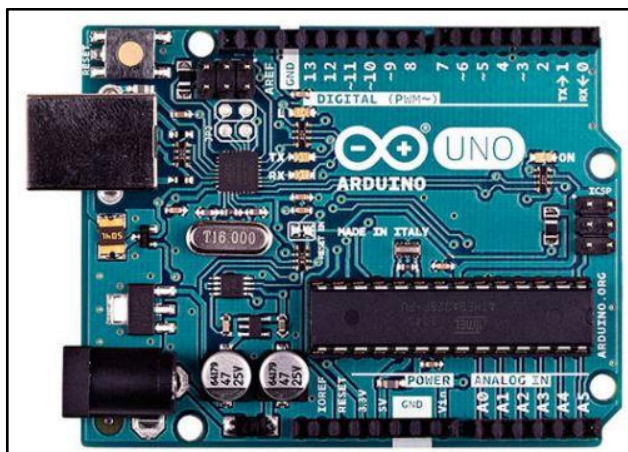


**Figura 10 - Microcontroladores**  
Fonte: Eletrônica Progressiva (2014).

## 2.7 A PLATAFORMA ARDUINO

O Arduino foi desenvolvido inicialmente para ser utilizado por *designers* em projetos de computação, porém devido a sua popularidade o Arduino tem sido empregado por fabricantes e desenvolvedores que estão empenhados na construção dos seus próprios projetos (EVANS, NOBLE, HOCHENBAUM, 2013, p. 24). A ideia do Arduino foi concebida em 2005, no IDI (*Interaction Design Institute*), situado na cidade de Ivrea, na Itália. O grande mentor deste projeto foi o professor Massimo Banzi juntamente com David Cuartielles. Os dois buscavam uma forma de baratear e facilitar o ensino de tecnologia aos estudantes de *design*, pois os materiais que existiam eram difíceis de usar e também eram caros. As duas principais metas estipuladas por Banzi e Cuartielles eram que esta nova plataforma fosse barata e que qualquer pessoa pudesse utilizá-la. Em virtude da implementação destas características no Arduino, a popularidade desta placa cresceu rapidamente, alcançando até mesmo o público de outras áreas (EVANS, NOBLE, HOCHENBAUM, 2013, p. 25).

Os componentes básicos de uma placa Arduino são um microprocessador Atmel AVR, um oscilador para gerar o *clock* da placa e um regulador linear 5 volts. Dependendo da versão do Arduino há também uma saída USB, possibilitando a conexão do mesmo a um computador para fazer o *upload* ou recuperar os dados contidos na placa (MCROBERTS, 2011, p. 24). A placa possui também pinos digitais que podem ser codificados como saída ou entrada e os mesmos podem propiciar saídas de modulação por largura de pulso (PWM) caso sejam programados para tal. Há na placa vários protocolos de comunicação e um botão de *reset* (EVANS, NOBLE, HOCHENBAUM, 2013, p. 26). A Figura 11 demonstra o leiaute de um Arduino, no caso a versão Uno.



**Figura 11 - Arduino Uno**  
**Fonte: Arduino (2016a).**

Por possuir *hardware* e *software* de fonte aberta, é possível que os códigos e até o projeto do Arduino sejam utilizados por qualquer pessoa e com qualquer propósito. A partir desta característica é que surgiram as placas-clone e outras placas baseadas no Arduino (MCROBERTS, 2011, p. 24). Muitas dessas placas desenvolvidas têm a característica de *shields*, que são placas especializadas que possibilitam aumentar a funcionalidade básica do Arduino (EVANS, NOBLE, HOCHENBAUM, 2013, p. 26). Os *shields* apresentam em seus circuitos diversos componentes como *displays* de LCD, receptores GPS, sensores, teclados, entre outros, implementando assim mais funcionalidades ao Arduino (MCROBERTS, 2011, p. 24).

A programação desta placa é possível através da IDE (*Integrated Development Enviroment*) do Arduino, que é um *software* no qual o usuário desenvolve o seu código baseado na linguagem C. Essa interface permite que o usuário escreva *sketches*, que nada mais são que programas para o Arduino, e faça o *upload* dos mesmos para a memória *flash* do microcontrolador através de uma porta USB, fazendo com que o Arduino interaja com qualquer periférico que esteja conectado a ele (MCROBERTS, 2011, p. 24).

### 3 METODOLOGIA E DESENVOLVIMENTO

O projeto do sistema de controle bola-barra foi desenvolvido para permitir o estudo de sistemas de controle instáveis, onde o usuário, após calibrar o sistema, pode definir o *setpoint* do mesmo, alterar os parâmetros do controlador - ganhos proporcional, integral e derivativo - e, desta forma, observar a resposta do sistema aos parâmetros definidos.

Este sistema é composto basicamente de três partes: mecânica, circuito eletrônico e controle. Todos esses conjuntos atuam de forma integrada para que o sistema funcione corretamente. A descrição destas partes pode ser observada abaixo:

- Mecânica: é constituído por uma base; uma barra; um suporte para o motor, um para a barra e outro para o sensor; além de componentes de fixação;
- Circuito eletrônico: é formado pelo microcontrolador Arduino Mega 2560; uma placa contendo um *display* de LCD e mais 6 chaves tácteis chamado de LCD *Keypad Shield*; um sensor ultrassônico e um servomotor;
- Controle: é integrado pelo *firmware* de operação do sistema que roda no microcontrolador. Foi desenvolvido analisando os requisitos do projeto e com a integração da mecânica e da eletrônica.

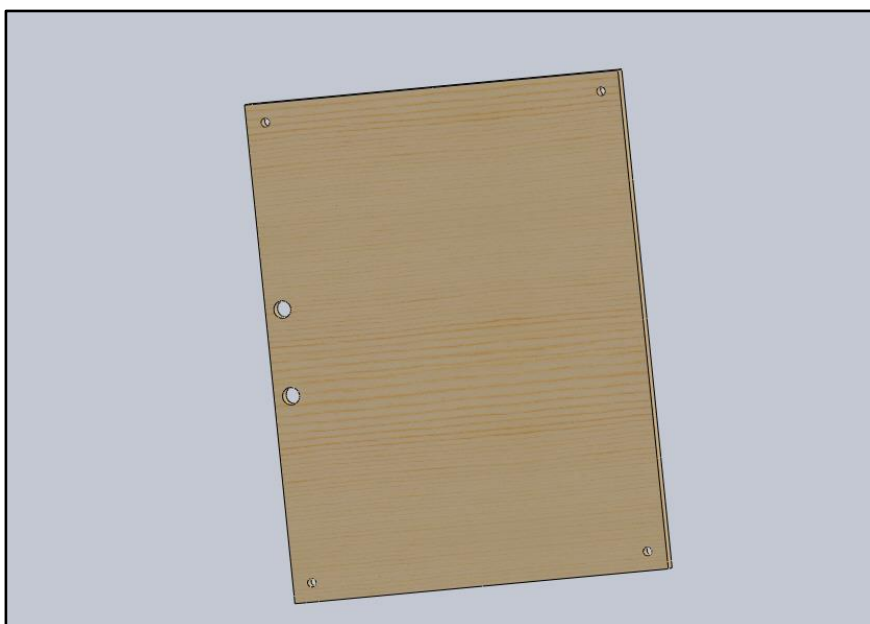
#### 3.1 DESENVOLVIMENTO MECÂNICO

O sistema de controle bola-barra é composto, na sua totalidade, por uma base, uma barra, uma bola, servomotor, um sensor ultrassônico, seguido por suportes e elementos de fixação para agrupar todos os componentes. Os materiais foram adquiridos de modo a se obter baixo custo de produção. O projeto mecânico foi desenvolvido com o auxílio do *software Solid Works 2013*, sendo possível realizar os desenhos 2D para confecção das peças: base, suporte da barra e suporte do motor e um desenho 3D para simular a montagem das peças.



### 3.1.1 Base retangular

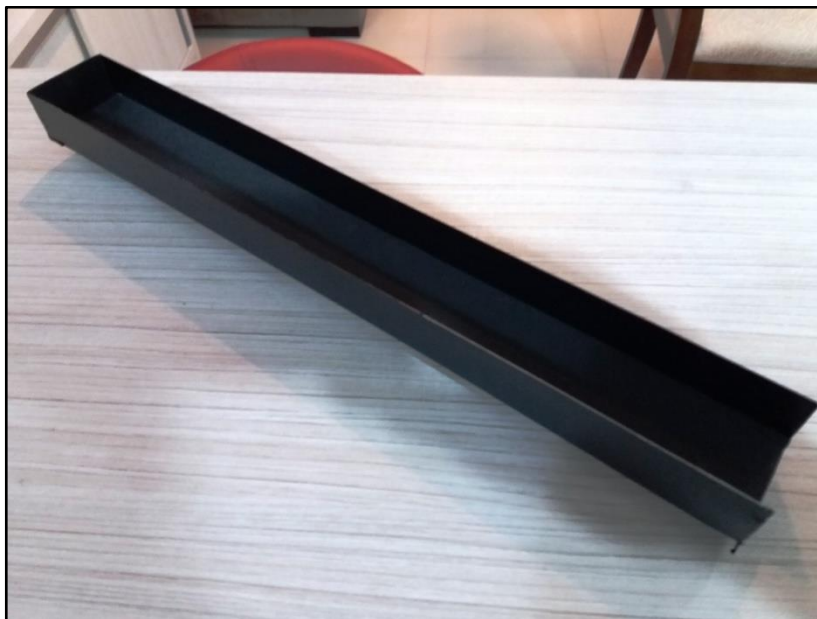
O primeiro componente a ser trabalhado foi a base principal apresentada em forma retangular, como mostra a Figura 12. Ela foi confeccionada em madeira MDF devido à sua rigidez e sua leveza. Sua função é a de suportar toda estrutura com componentes mecânicos, as placas eletrônicas, microcontrolador e *display*.



**Figura 12 - Base principal**  
**Fonte: Autoria própria.**

### 3.1.2 Barra

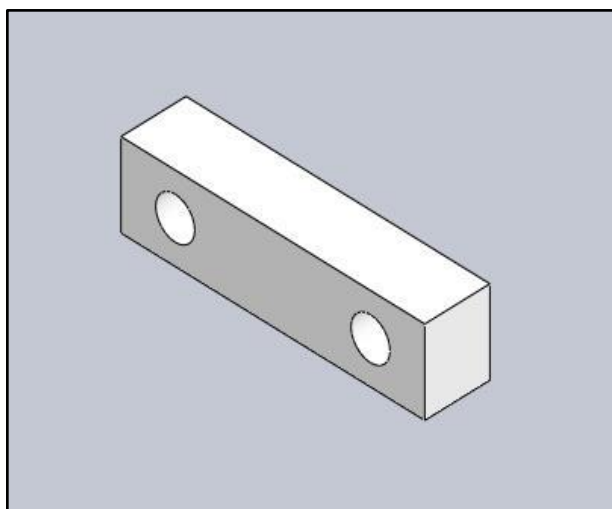
O próximo item a ser trabalhado foi a barra. Ela foi adaptada a partir de um perfil guia de aço galvanizado de 45 mm de largura e altura por 400 mm de comprimento. Este material foi escolhido pois além de ser leve e proporcionar o controle angular da barra pelo servomotor, este perfil facilitava a fixação do sensor na haste. Ela foi revestida com papel *contact* preto. A montagem final da barra pode ser observada na Figura 13.



**Figura 13 - Barra**  
**Fonte: Autoria própria.**

### 3.1.3 Suporte do motor

O suporte do motor foi confeccionado em náilon com o auxílio de fresadora convencional e furadeira de bancada. A Figura 14 mostra o desenho do suporte do motor.



**Figura 14 - Suporte do motor**  
**Fonte: Autoria própria.**

A principal função desta peça é fixar o motor na base e evitar que tenha folga excessiva no mesmo. O suporte foi ligado à base para apoiar o servomotor. Este, por sua vez, foi fixado ao suporte por parafusos juntamente com uma presilha que também foi fixada à base, podendo ser visto na Figura 15.

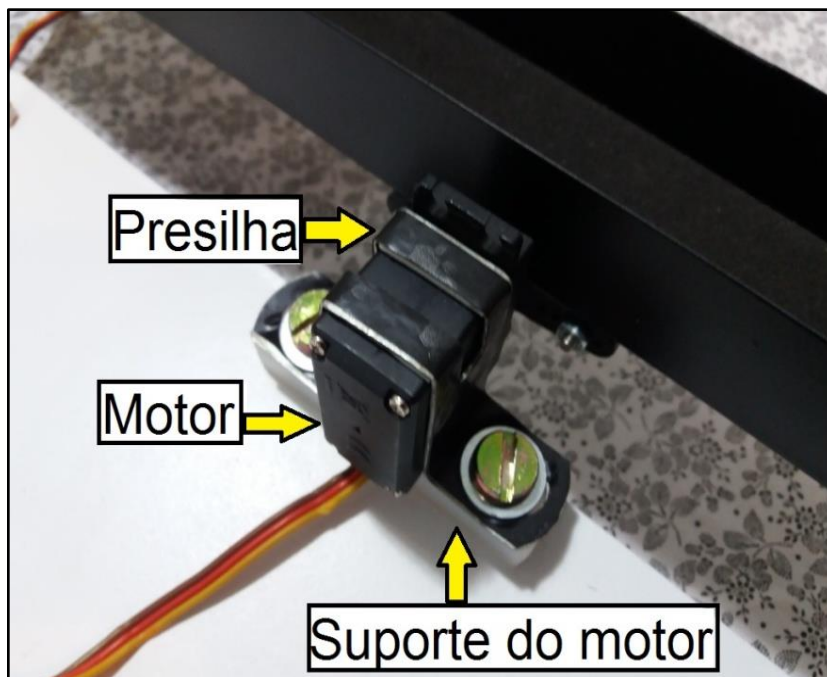
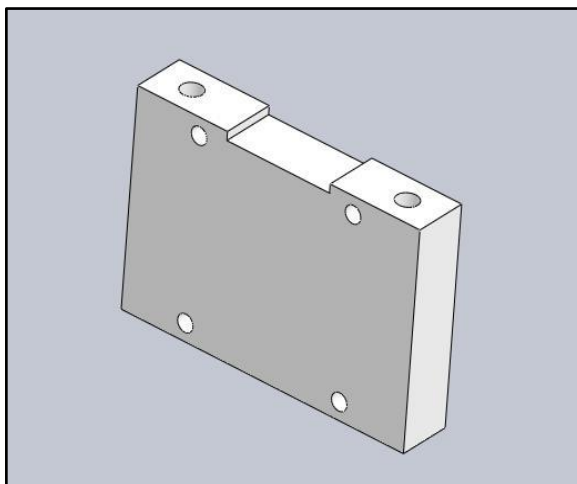


Figura 15 - Montagem do motor  
Fonte: Autoria própria.

#### 3.1.4 Suporte da barra

O suporte da barra também foi fabricado em náilon com o auxílio de fresadora convencional e furadeira de bancada, segundo Figura 16. O suporte foi ligado à barra por meio de parafusos e porcas de fixação. A montagem desta peça no suporte acoplado ao eixo do servomotor também foi realizada com porcas e parafusos possibilitando movimentos angulares do motor.



**Figura 16 - Suporte da barra**  
**Fonte: Autoria própria.**

### 3.1.5 Suporte do sensor

A ideia inicial era produzir este suporte do sensor do sensor. Porém, como existem alguns modelos disponíveis à venda na internet, de baixo custo e que são específicos para o tipo de sensor ultrassônico adotado, pois possuem as dimensões correspondentes ao sensor, foi decidido pela compra do mesmo. O suporte foi adaptado e fixado em uma das extremidades da barra através de parafusos e porcas fornecendo encaixe preciso para o sensor ultrassônico, consonante com a Figura 17.



**Figura 17 - Suporte do sensor**  
**Fonte: Autoria própria.**

### 3.1.6 Montagem completa do sistema mecânico

A montagem não exigiu muita complexidade devido ao dispositivo possuir poucas peças, porém foi necessário ajustar o sistema para que o mesmo possua boa precisão para, conseqüentemente, obter um melhor desempenho, pois qualquer interferência poderia comprometer o equilíbrio. A montagem completa pode ser observada na Figura 18.

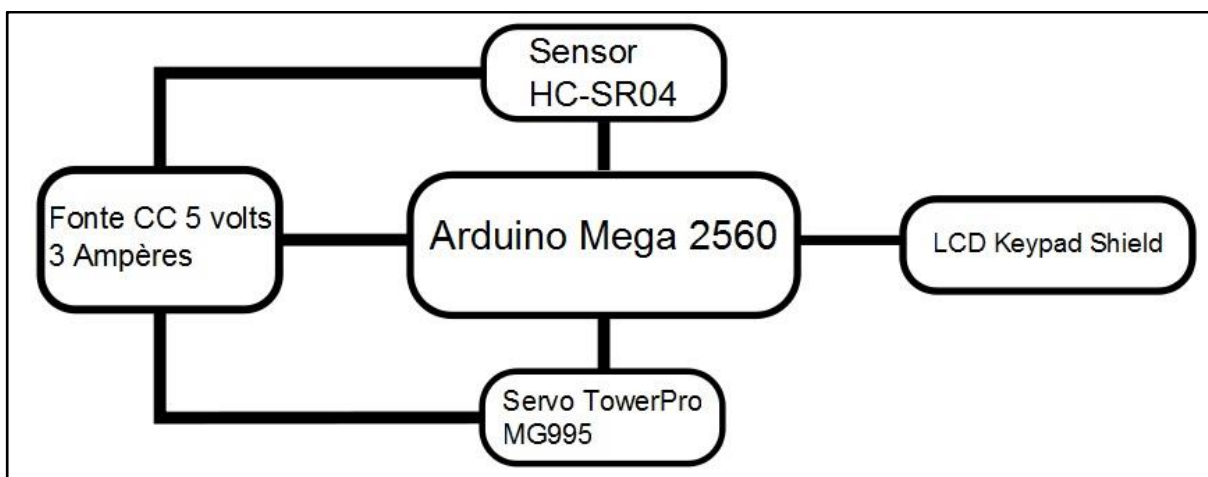


**Figura 18 - Montagem do conjunto mecânico**  
**Fonte: Autoria própria.**

## 3.2 CIRCUITO ELETRÔNICO

Para o desenvolvimento deste sistema foram empregados alguns componentes eletrônicos que podem ser visualizados no diagrama em blocos representado na Figura 19 e que são classificados na seqüência:

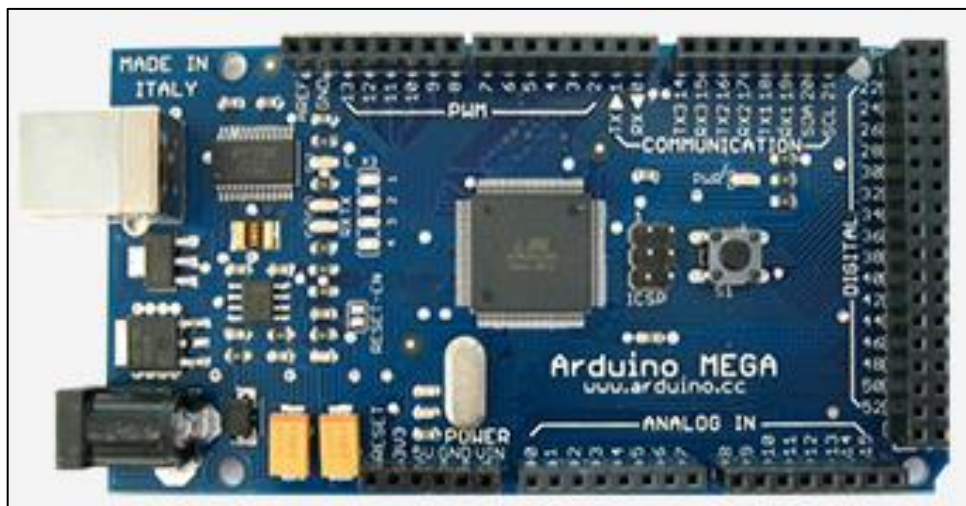
- Controlador: Arduino Mega 2560;
- Interface com usuário: LCD *Keypad Shield*;
- Sensor: Sensor ultrassônico HC-SR04;
- Atuador: Servomotor TowerPro MG995.



**Figura 19 - Diagrama em blocos**  
Fonte: Autoria própria.

### 3.2.1 Controlador

O controlador utilizado neste sistema é o Arduino Mega 2560. Esta placa possui um processador Atmega 2560, contêm 54 pinos digitais que podem funcionar como entrada ou saída, há 16 pinos de entrada analógica e mais 14 portas que podem gerar sinais PWM. Esta placa também possui um botão de *reset*, conexão USB para o carregamento dos programas e suporta entrada de tensão de corrente contínua de 6 a 20 V, sendo mais indicado a faixa de 7 a 12 V. A Figura 20 mostra a placa do Arduino Mega 2560 utilizada neste projeto.



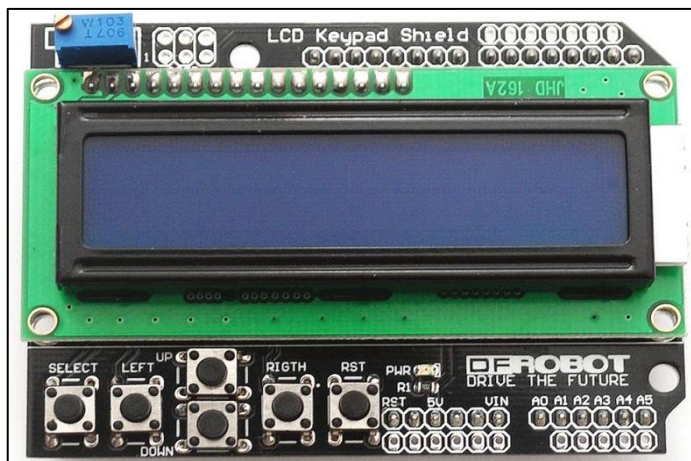
**Figura 20 - Arduino Mega**  
**Fonte: Arduino (2016b).**

Todas as etapas do processo do sistema bola-barras são controladas pelo Arduino. Basicamente este controlador recebe a informação da posição da bola na barra através do sensor ultrassônico, mede a diferença com o *setpoint* inserido pelo usuário, realiza um cálculo levando em conta os parâmetros do controlador PID e, conseqüentemente, atua no servomotor para que a bola esteja na posição desejada.

### 3.2.2 Interface com o usuário

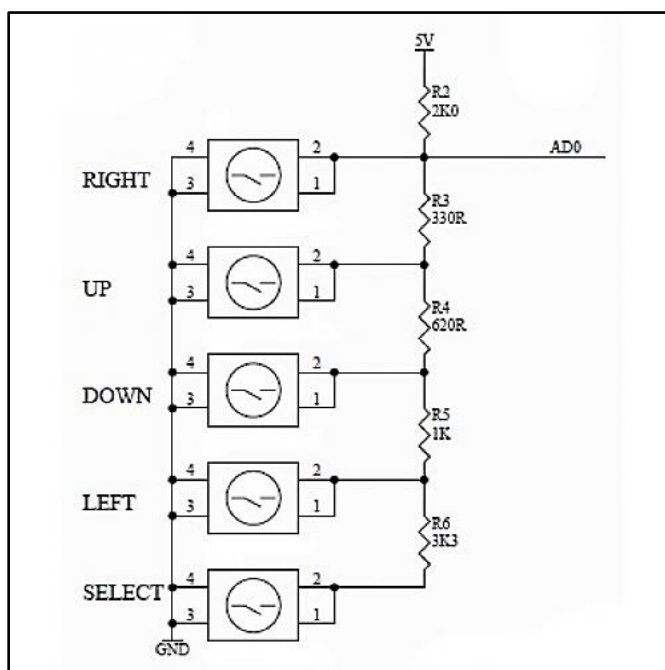
A interação do sistema com o usuário é realizada através da placa LCD *Keypad Shield*. Esta placa é composta por um *display* de LCD, seis chaves tácteis e mais um potenciômetro para ajustar o brilho da tela do LCD, como mostra a Figura 21.





**Figura 21 - Placa LCD Keypad Shield**  
**Fonte: Natanael (2012).**

Este *shield* possui barra de pinos macho para facilitar a conexão com a placa do Arduino. Das 6 chaves tácteis que a placa contém, só é possível definir a funcionalidade de cinco delas, pois uma é utilizada exclusivamente como *reset* e está identificada no módulo como RST. Para realizar a leitura das outras cinco chaves, que servem para aumentar ou diminuir os parâmetros do controlador, é utilizada uma associação de resistores que está ligada no pino analógico A0 do Arduino, conforme mostra a Figura 22. Essa associação economiza a utilização de diversos pinos da placa do controlador.



**Figura 22 - Associação de resistores do módulo LCD**  
**Fonte: Arduino e Cia (2013).**



### 3.2.3 Aplicação do sensor ultrassônico

Para realizar a verificação da posição da bola na barra utiliza-se o sensor ultrassônico HC-SR04, identificado na Figura 23. Este sensor possui 4 pinos: dois para a alimentação 5 V e dois pinos que são identificados com *trigger* e *echo* e que servem para o cálculo da distância do objeto. O funcionamento básico deste sensor ocorre da seguinte forma: o controlador emite um pulso alto de 5 V por pelo menos 10 microsegundos no pino *trigger*. Após este tempo, o módulo enviará automaticamente um ciclo de 8 sinais ultrassônicos de 40 kHz e esperará esse sinal retornar. Se o sensor detectar o sinal de retorno, o pino *echo* será colocado em nível alto e o tempo dele ativado será proporcional a distância do objeto, conforme mostra a Figura 24. O cálculo realizado para medir a distância em centímetros está expresso na equação (4):

$$\text{Distância} = \text{largura do pulso echo} / 58 \quad (4)$$



Figura 23 - Sensor ultrassônico HC-SR04  
Fonte: FilipeFlop (2016a).

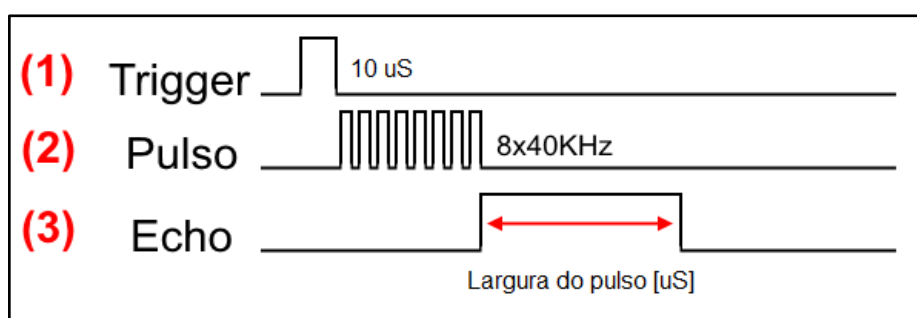


Figura 24 - Diagrama de funcionamento do sensor ultrassônico  
Fonte: FilipeFlop (2016a).

### 3.2.4 Atuador

A principal função do servomotor TowerPro Mg995 é realizar o controle angular da bola na barra, representado na Figura 25. Este servomotor foi escolhido pois apresenta alto torque, engrenagens metálicas que oferecem maior robustez ao projeto, acopladores de plástico que facilitam a fixação da barra no mesmo, além da facilidade para controlar a sua rotação em comparação a motores de corrente contínua, pois não é necessário nenhum *driver* adicional e todo o controle de posição ocorre através de sinais PWM enviados pelo Arduino para o servomotor.



**Figura 25 - Servomotor TowerPro MG995**  
Fonte: FilipeFlop (2016b).

### 3.2.5 Utilização das entradas e saídas do microcontrolador

No Quadro 1 pode-se observar quantos pinos do Arduino Mega 2560 são usados e para qual finalidade eles são utilizados.

Item	Descrição	Endereço	Utilização
1	LCD Keypad Shield	4	Transferência de dados para o <i>display</i>
2	LCD Keypad Shield	5	Transferência de dados para o <i>display</i>
3	LCD Keypad Shield	6	Transferência de dados para o <i>display</i>
4	LCD Keypad Shield	7	Transferência de dados para o <i>display</i>
5	LCD Keypad Shield	8	Pino RS do <i>display</i>
6	LCD Keypad Shield	9	Pino <i>enable</i> do <i>display</i>
7	LCD Keypad Shield	10	Ativa o <i>backlight</i> do <i>display</i>
8	LCD Keypad Shield	A0	Entrada analógica para ler os botões do módulo
9	Sensor ultrassônico	2	Controlar o <i>trigger</i> do sensor
10	Sensor ultrassônico	3	Controlar o <i>echo</i> do sensor
11	Servomotor	12	Controlar a rotação do motor

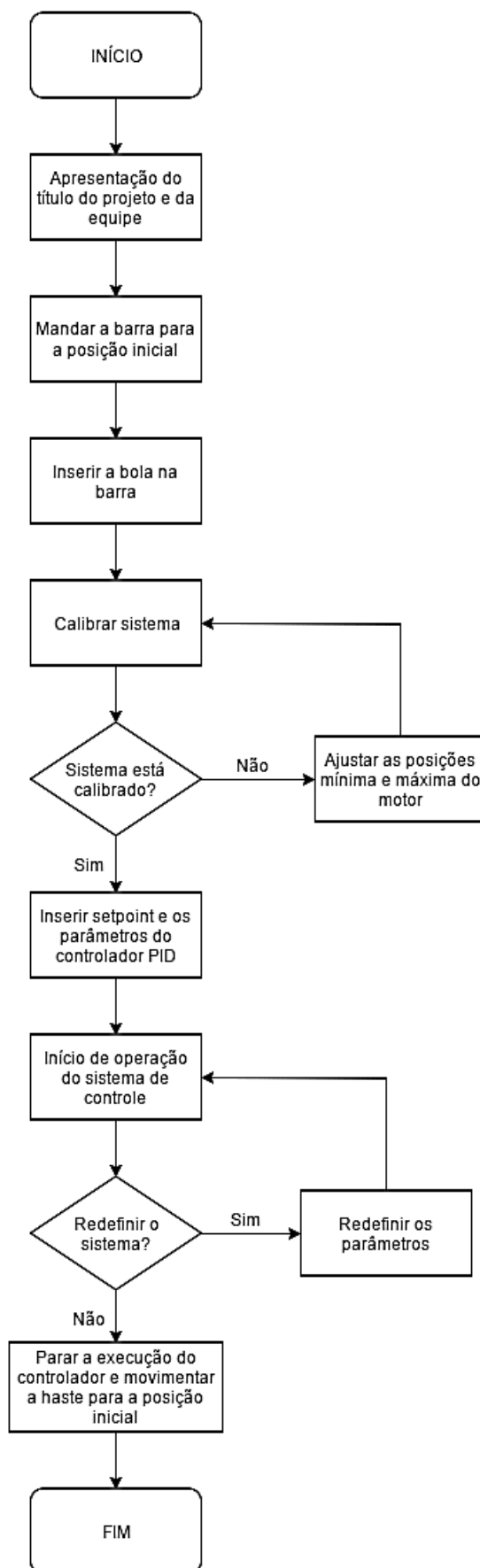
**Quadro 1 – Entrada e saídas do Arduino**

Fonte: Autoria própria.

### 3.3 CONTROLE

O sistema bola-barras é todo controlado pelo Arduino. Este microcontrolador tem a missão de adquirir os parâmetros inseridos pelo usuário, adquirir o sinal enviado pelo sensor ultrassônico e controlar o servomotor, além de realizar todo o intertravamento dos dispositivos que estão integrados.

Toda a programação do microcontrolador foi realizada através da IDE do próprio Arduino. O *firmware* de operação do sistema consiste em fazer uma breve apresentação do projeto, posicionar a barra paralela à base que é a sua posição inicial, calibrar o sistema com as posições máximas e mínimas tanto para o motor quanto para a esfera, obter os parâmetros do controlador inseridos pelo teclado e, na sequência, iniciar o processo de estabilização da bola na barra conforme o *setpoint* inserido pelo usuário. Todo o processo pode ser visualizado através do fluxograma na Figura 26.



**Figura 26 - Fluxograma do processo**  
Fonte: Autoria própria.

Para realizar o controle de todo o sistema foram desenvolvidas algumas rotinas a serem rodadas no microcontrolador e que, na sequência, foram agrupadas para formar o *firmware* final de operação. Não houve preocupação com a economia de pinos ou de memória do microcontrolador, pois o Arduino Mega fornece memória suficiente para controlar este projeto.

Num primeiro momento foram definidas as bibliotecas a serem utilizadas no programa. Devido aos componentes empregados no projeto foram usadas quatro bibliotecas: uma para o *display* de LCD, outra para a utilização do servomotor, uma terceira para acessar a memória EEPROM do Arduino e a última para utilizar o Timer 1 do microcontrolador. Nessa etapa foi realizado o mapeamento dos pinos que serão utilizados pelo *display* LCD, a definição dos valores analógicos atribuídos aos botões do módulo LCD *Keypad Shield* lidos pela porta A0 do Arduino, além da definição dos valores dos atrasos (*delays*) em milisegundos, conforme mostra a Figura 27.

```
#include <LiquidCrystal.h> //Biblioteca do display LCD
#include <Servo.h> //Biblioteca do servomotor
#include <EEPROM.h> //Biblioteca para armazenar o valor da posição do motor na memória EEPROM do Arduino
#include <TimerOne.h> //Biblioteca para ativar o Timer 1

// Definição das variáveis do sistema

#define DELAY1 3000
#define DELAY2 2000
#define DELAY3 100
#define DELAY4 50
#define PINO_SERVO 12 //Pino utilizado para controlar a rotação do servomotor
#define PINO_TRG 2 //Pino utilizado para o trigger do sensor
#define PINO_ECHO 3 //Pino utilizado para o echo do sensor
#define SELECT 800 //Valor analógico definido para o botão SELECT
#define RIGHT 95 //Valor analógico definido para o botão RIGHT
#define LEFT 600 //Valor analógico definido para o botão LEFT
#define UP 100 //Valor analógico definido para o botão UP
#define DOWN 400 //Valor analógico definido para o botão DOWN
#define CHAVE1 630
#define CHAVE2 1000
#define MENOR_DISTANCIA 3
#define MAIOR_DISTANCIA 38
#define POS_ORIGEM 87

LiquidCrystal lcd (8, 9, 4, 5, 6, 7); //Mapa dos pinos para o display LCD
Servo motor; //Definindo o nome da variável para o servomotor
```

**Figura 27 - Inicialização do *firmware***  
**Fonte: Autoria própria.**

Para este programa foram definidas algumas variáveis globais que atuarão tanto na função de selecionar os parâmetros de controle quanto na interrupção para

o cálculo do controle. Também foram determinadas as funções que funcionarão apenas uma vez dentro da rotina *setup* do Arduino, com exceção da rotina de interrupção que será chamada a cada 50 ms. As funções que irão funcionar só uma vez são: apresentação, verificação de posição, calibração do sistema e determinação dos parâmetros do controlador. A Figura 28 mostra as variáveis globais do sistema e a chamada das funções dentro da rotina *setup*.

```
//Variáveis globais do sistema

double pos, distancia_minima, distancia_maxima, pos_45negativo, pos_45positivo, pos_esfera, controle, setpoint, sp;
double Kp = 0.01;
double Ki = 0;
double Kd = 0;
double erro1 = 0;
double controle_prev = 0;
int memoria = 0;

void setup()
{
  lcd.begin(16, 2); //Inicia as 16 colunas e as 2 linhas do LCD
  apresentacao(); //Faz uma introdução do título do trabalho, dos alunos responsáveis e do professor orientador
  verificar_pos(); //O motor vai para a posição inicial
  calibracao(); //Faz a calibração do sistema
  sp = distancia_minima; //Atribui a variável SP a distância mínima captada pelo sensor
  parametros(); //Usuário escolhe os valores de setpoint, Kp, Ki e Kd
  Timer1.initialize(50000); //A interrupção roda a cada 50 ms
  Timer1.attachInterrupt(interruptao); //Função a ser chamada a cada interrupção
}
```

**Figura 28 - Rotina do *setup***

Fonte: Autoria própria.

As funções apresentação e verificação de posição do motor são de baixa complexidade. A primeira função apenas escreve dados no *display* de LCD, ou seja, vai informar o título do trabalho, os alunos responsáveis e o professor orientador do projeto. Já a função de verificação apenas manda o motor ir para a posição inicial, ou seja, o motor fica paralelo a base de madeira.

Já a rotina de calibração tem uma função muito importante dentro do contexto de todo o sistema. É ela que determina os ângulos mínimo e máximo de operação do sistema e também as distâncias mínima e máxima que a bola irá alcançar na haste. A Figura 29 exemplifica a operação desta função.

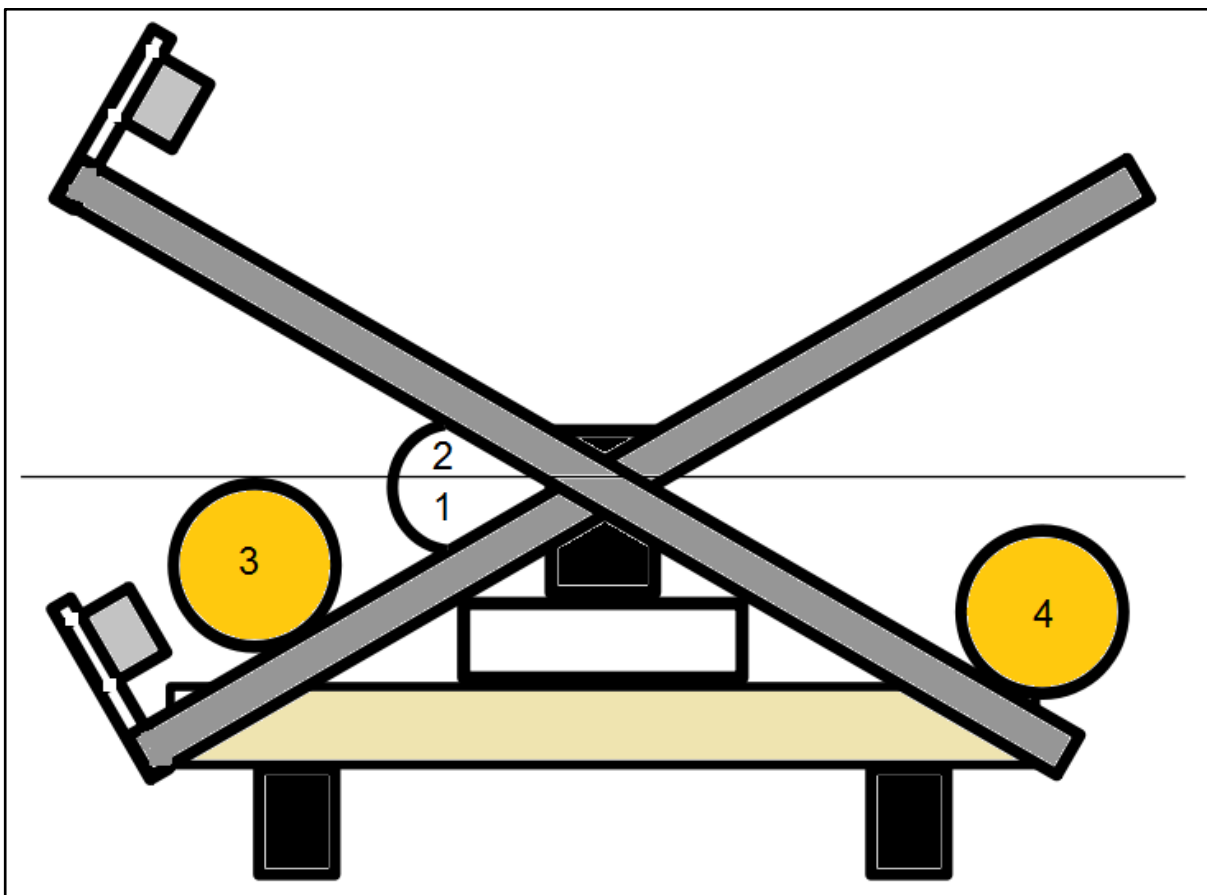


Figura 29 - Sistema de calibração  
Fonte: Autoria própria.

Onde:

- 1: ângulo mínimo de operação;
- 2: ângulo máximo de operação;
- 3: distância mínima da bola na barra;
- 4: distância máxima da bola na barra.

Após o início da rotina de calibração a barra desloca-se automaticamente em  $30^\circ$  negativos em relação a horizontal, indicado pelo número 1 na Figura 29. Na primeira etapa é solicitado que o usuário coloque a bola na barra para que seja possível determinar a distância mínima da esfera, conforme indicado pelo número 3. Neste momento é possível que o usuário altere este ângulo mínimo pressionando o botão *up* para fazer a barra subir ou apertando *down* para fazer a haste descer ainda mais. Depois da barra estar na posição desejada, o usuário deve pressionar o botão *select* para que a posição mínima do motor e a distância mínima da bola sejam

gravadas pelo microcontrolador. Na sequência a haste sobe automaticamente até a posição de 30 graus positivos em relação a base, representado pelo número 2. Na segunda etapa mede-se a distância máxima que a bola está na haste, simbolizado pelo número 4. Pode-se alterar o ângulo da barra através do teclado da mesma forma como citado na primeira etapa. Ao finalizar essa etapa pressionando o botão *select*, o microcontrolador irá armazenar a posição máxima do motor e a distância máxima da bola. Uma parte desta função de calibração está representada na Figura 30.

```

while (OK == false) { //enquanto o botão select não é pressionado

  botao = analogRead(0);
  delay(DELAY3);

  if (botao < RIGHT) {}

  else if (botao < UP) { //se o botão UP for pressionado a barra irá subir
    pos++;
    motor.write(pos);
  }

  else if (botao < DOWN) { //se o botão DOWN for pressionado a barra irá descer
    pos--;
    motor.write(pos);
  }

  else if (botao < LEFT) {}

  else if (botao < SELECT) { //Salva a posição do motor e da bola e também sai do loop
    OK = !OK;
    pos_45negativo = motor.read();
    distancia_minima = lerPosicao();
  }

}

```

**Figura 30 - Detecção dos botões do módulo LCD**  
**Fonte: Autoria própria.**

Enquanto o botão *select* não é pressionado o programa fica rodando num *loop* infinito. Durante a execução deste laço é lido o pino analógico A0 para ver se alguma tecla foi pressionada. Se o botão *up* for pressionado, a variável relacionada a posição do motor será incrementada e o servomotor irá se mover um grau positivo, caso o botão *down* seja pressionado, a variável de posição será decrementada e o motor irá se deslocar em um grau negativo. O programa só sairá deste *loop* se o



botão *select* for pressionado, pois a variável “OK” passará da condição *false* para *true* e será atribuído a duas variáveis globais a posição da bolinha e também a posição do motor.

Depois de calibrar o sistema será a vez da função que possibilita ao usuário inserir os parâmetros do controlador. Nesta etapa serão definidos o valor do *setpoint*, dos ganhos do controlador (Kp), integral (Ki) e derivativo (Kd). Esses quatro valores são armazenados em variáveis globais porque são utilizados nesta função e durante a interrupção. Na Figura 31 poderá ser observada a seleção do valor do *setpoint* do sistema.

```

while (select == false)
{
    botao = analogRead(0);
    delay(DELAY3);
    lcd.clear();
    lcd.print("Setpoint:");
    lcd.setCursor(11, 0);
    lcd.print(sp);
    if (botao < RIGHT) {}
    else if (botao < UP) {
        sp++;
        if (sp > distancia_maxima) {
            sp = distancia_maxima;
        }
    }
    else if (botao < DOWN) {
        sp--;
        if (sp < distancia_minima) {
            sp = distancia_minima;
        }
    }
    else if (botao < LEFT) {}
    else if (botao < SELECT) {
        select = !select;
    }
}

```

**Figura 31 - Seleção dos parâmetros**  
**Fonte: Autoria própria.**

Nesta rotina também há um *loop* infinito enquanto o botão *select* não for pressionado. O valor inicial do *setpoint* e o máximo são iguais as distâncias mínima e máxima da bola, respectivamente, medidas durante a rotina de calibração, pois não pode-se permitir que seja inserido um valor menor ou maior do que os medidos durante esta rotina. Para aumentar o valor do *setpoint* deve-se pressionar o botão *up*

e para diminuir pode-se apertar o botão *down*. Enquanto o botão *select* não for pressionado é mostrado no *display* de LCD o valor atual do *setpoint* escolhido. Esse laço *while* só será encerrado quando o botão *select* for pressionado para a variável “OK” mudar de estado e, na sequência, ser solicitado para o usuário inserir o valor do ganho integral. Para todos os parâmetros do controlador há um valor mínimo e máximo estabelecidos na rotina do *firmware*.

Para ser realizado o cálculo do controle dentro da função de interrupção é necessário normalizar duas variáveis: o *setpoint* e a posição atual da esfera. O processo de normalização consiste em converter essas variáveis numa escala que vai de -1 a +1. Já para a variável do sinal de controle, que irá atuar diretamente na posição do motor, o processo de normalização é invertido: pega-se o valor normalizado e converte-o para uma escala entre a posição mínima e máxima do motor. A Figura 32 mostra o programa para envio o envio da posição lida da esfera para a função que normaliza esse valor e retornando-o para a variável “*sppmm*”.

```
sppmm = calc_esfera(pos_esfera, distancia_minima, distancia_maxima, -1, 1);
}

//Função calc_esfera normaliza o valor da posição atual da esfera entre -1 e +1
float calc_esfera(float x, float in_min, float in_max, float out_min, float out_max)
{
    return (x - in_min) * (out_max - out_min) / (in_max - in_min) + out_min;
}
```

**Figura 32 - Normalização do valor lido pelo sensor**  
Fonte: Autoria própria.

Após a execução do *setup* no Arduino será iniciado a rotina *loop* dentro do microcontrolador. Como essa função roda infinitamente no controlador, foi adicionada uma rotina dentro deste *loop* que fica aguardando o usuário pressionar o botão *select* para desligar a interrupção, conforme mostra a Figura 33.

```
void loop()
{
    delay(DELAY4);
    parar_int();
}
```

**Figura 33 - Rotina para parar a interrupção**  
Fonte: Autoria própria.

A função "parar\_int", indicada na Figura 34, consiste em, basicamente, pausar a atuação da interrupção no sistema e para isso acontecer o botão *select* deve ser pressionado. Foi criada uma rotina que enquanto a variável "confirmar" for falsa, fica rodando um *loop* que lê a cada 100 ms a entrada analógica A0. Se o botão *select* for apertado a interrupção é pausada, a função para alterar os parâmetros é chamada e após a finalização dessa rotina a interrupção é novamente ativada e, desta forma, a variável "confirmar" torna-se verdadeira, encerrando o *loop*.

```
void parar_int()
{
    int botao;
    boolean confirmar = false;
    if (confirmar == false)
    {
        botao = analogRead(0);
        delay(DELAY3);

        if (botao > CHAVE1 && botao < CHAVE2)
        {
            Timer1.stop();
            parametros();
            delay(DELAY3);
            Timer1.attachInterrupt(interruptcao);
        }
        confirmar = true;
    }
}
```

**Figura 34 - Função que para a interrupção**  
**Fonte: Autoria própria.**

## 4 APRESENTAÇÃO E ANÁLISE DOS RESULTADOS

Este capítulo abordará os resultados provenientes dos ensaios realizados junta ou separadamente com a parte mecânica, eletrônica e de controle.

### 4.1 ENSAIOS DO CONJUNTO MECÂNICO

Os primeiros testes feitos com o sistema bola-barra envolveram a parte mecânica. Estes ensaios foram realizados para verificar se o conjunto apresentava folgas, pois se este tipo de problema surgisse a estabilidade da bola na barra e a precisão do sistema estariam comprometidas. Porém nenhuma folga foi constatada durante a realização dos ensaios nos suportes e, desta forma, nenhuma das peças usinadas sofreu alterações.

Contudo, durante os ensaios de estabilização da esfera na haste observou-se que o sistema não alcançaria um ponto de equilíbrio por dois motivos específicos: a barra possuía as laterais muito baixas e a fixação do suporte do sensor deixava o sensor muito alto, dificultando a leitura correta da posição da esfera. A primeira barra utilizada durante o desenvolvimento do projeto era uma adaptação de uma tampa de canaleta para fiação elétrica. Esse material foi escolhido pois era leve, oferecia a possibilidade de fixação do suporte do sensor e era de fácil de processamento. Entretanto, as abas laterais desta haste eram muito pequenas e durante o deslizamento da bola pela barra a canaleta não conseguia conduzir a esfera corretamente, ocasionando a queda da bola durante o processo de controle da posição. A Figura 35 mostra a primeira barra utilizada no projeto.



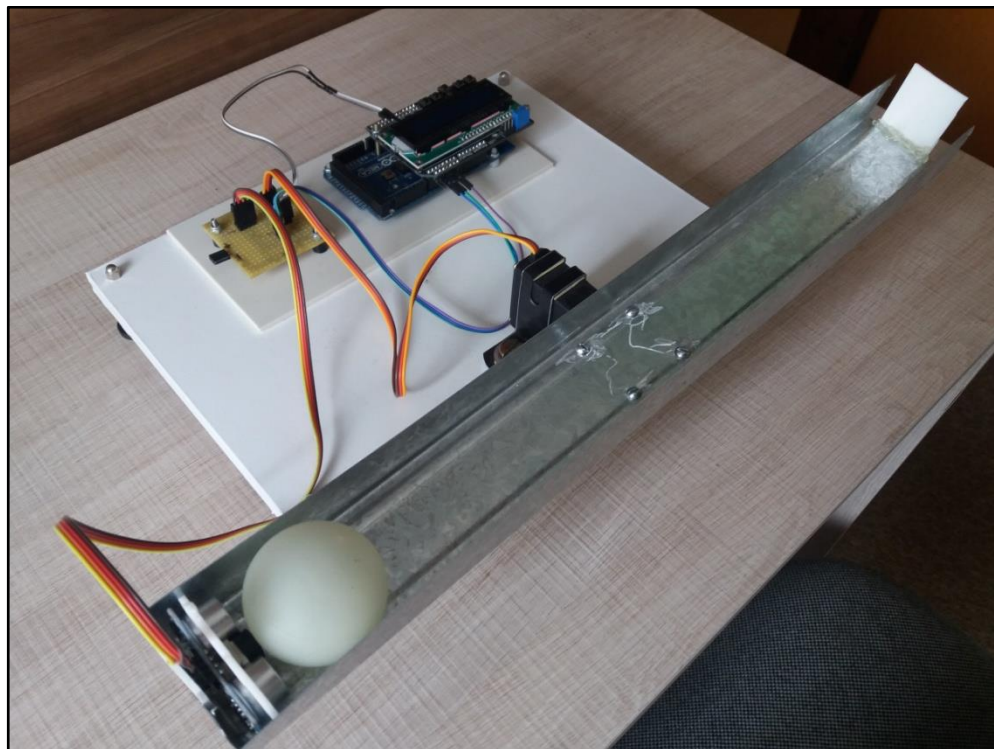
**Figura 35 - Primeira barra utilizada**  
**Fonte: Autoria própria.**

O problema de leitura do sensor se devia à fixação da base do suporte do sensor na parte superior da canaleta. Como ela era muito baixa, o sensor ficava posicionado acima do centro da bola. Desta forma, conforme a bola se afastava do sensor, o valor medido da distância da esfera era completamente afetada. A Figura 36 apresenta o primeiro protótipo do sistema bola-barras com a tampa da canaleta adaptada de haste.



**Figura 36 - Primeiro protótipo do sistema**  
**Fonte: Autoria própria.**

A solução encontrada para resolver estes problemas foi a substituição da tampa da canaleta por um perfil guia de aço galvanizado. Este perfil, além de ser leve, apresenta maior robustez em relação a haste antiga, permite uma fixação diferente do suporte do sensor ultrassônico, conforme mostrado na Figura 17, é de fácil processamento (furar, dobrar ou cortar) e possui as abas laterais maiores, aproximadamente 45 mm, as quais ajudam a conduzir a bola durante o processo de estabilização da esfera na haste. A Figura 37 mostra o segundo protótipo do sistema com a nova barra.



**Figura 37 - Segundo protótipo do sistema**  
**Fonte: Autoria própria.**

As últimas alterações realizadas na estrutura da barra foram a inserção de E.V.A. na superfície onde a bola desliza e a criação de um batente no final da barra. A primeira mudança citada foi para aumentar o atrito entre a bola e a barra, pois durante os testes verificou-se que havia uma certa dificuldade para estabilizar a esfera na posição pretendida. Notou-se que após a colagem do E.V.A. na barra houve uma melhora significativa no processo de controle. Já a criação do batente na barra serviu para não deixar a bola cair durante o funcionamento do projeto e também para auxiliar no processo de calibração do sistema, de acordo com a Figura 13.

#### 4.2 TESTES DO CIRCUITO ELETRÔNICO E DE CONTROLE

Durante os ensaios realizados com todo o sistema integrado - mecânica, eletrônica e controle - houve duas mudanças vitais para o projeto: um relacionado a leitura da posição da bola na barra e o outro ligado aos parâmetros do controlador. A leitura da bola na barra persistiu com alguns erros mesmo com a mudança do



material da haste e da posição do suporte do sensor, mostrando que havia problemas com o sensor ou com a programação.

Foi estabelecida uma rotina simples para verificar a leitura realizada pelo sensor. Durante a execução deste programa pode-se observar que sensor apresentava uma leitura correta quando o objeto medido era plano, porém quando o objeto era uma esfera o sensor revelava dificuldades para ler a posição da bola. Para amenizar este problema foi retirado do programa a biblioteca do sensor ultrassônico e foi adicionada uma função, exibida na Figura 38, para ler a posição da esfera.

```
double lerPosicao()
{
    delay(500);

    const int trig = PINO_TRG;
    const int echo = PINO_ECHO;
    long duration, cm;

    pinMode(trig, OUTPUT);
    digitalWrite(trig, LOW);
    delayMicroseconds(2);
    digitalWrite(trig, HIGH);
    delayMicroseconds(10);
    digitalWrite(trig, LOW);
    pinMode(trig, INPUT);

    duration = pulseIn(echo, HIGH);

    cm = duration / (29 * 2);

    return cm;
}
```

**Figura 38 - Função para ler a posição da esfera**  
**Fonte: Autoria própria.**

Esta função começa com atraso de 500 ms para que não haja o risco das ondas emitidas pelo ultrassom entrarem em conflito quando essa função for chamada. Na sequência o pino *trigger* do sensor é classificado como saída e colocado em nível baixo. Após um atraso de 2 microssegundos o *trigger* vai receber um pulso alto durante 10 microssegundos e, na sequência, ele será colocado em nível baixo e será classificado como entrada. A partir desse momento o módulo irá emitir as ondas e o pino *echo*, que está ativado como entrada, irá aguardar o sinal

retornar em nível alto. O tempo necessário para as ondas retornarem será utilizado para calcular a distância da bola, como pode ser visualizado na variável “cm”. Essa variável será retornada toda vez que a função de leitura do sensor for chamada no programa.

Outra mudança importante durante o teste do *firmware* foi a alteração dos parâmetros do controlador. No início os parâmetros inseridos pelo usuário eram:

- SP: *setpoint*;
- Kc: ganho do controlador;
- Ti: tempo integral;
- Td: tempo derivativo.

O programa pegava esses valores inseridos pelo usuário e aplicava na rotina de interrupção para o cálculo do ganho integral (Ki), derivativo (Kd) e proporcional (K). Com esses valores obtidos, o controlador calculava o erro e, na sequência, aplicava todos esses parâmetros numa expressão para determinar o valor do sinal de controle e, desta forma, o valor normalizado do controle era transformado num valor entre a posição máxima e mínima do servomotor. Depois de realizar o controle no motor, os valores do controle e do erro eram atualizados para a próxima rotina de interrupção. Tais cálculos podem ser visualizados na Figura 39.

Um dos problemas deste algoritmo era a falta do cálculo dos controladores proporcional, integral e derivativo separados. Outra questão percebida também foi que para a ação integral ser anulada no sistema, o valor do tempo de integral deveria ser muito alto, o que tornava inviável a inserção destes valores através do teclado do módulo LCD. Para resolver este problema foi criada uma nova rotina na qual o usuário insere automaticamente os valores dos ganhos proporcional, integral e derivativo, conforme citado na subseção 3.3.



```

Ts = 0.05;
Ki = Kc * Ts / Ti; //Ganho integral

K = Kc - Ki / 2; //Ganho proporcional

Kd = Kc * Td / Ts; //Ganho derivativo

pos_esfera = lerPosicao();
sppmm = calc_esfera(pos_esfera, distancia_minima, distancia_maxima, -1, 1);

erro = setpoint - sppmm; //cálculo do erro

controle = controle1 + (K + Ki + Kd) * erro - (K + 2 * Kd) * erro1 + Kd * erro2; //cálculo do controle

if (controle > 1) {
    controle = 1;
}
if (controle < -1) {
    controle = -1;
}
pos = calc_posicao(controle, -1, 1, pos_45negativo, pos_45positivo);

motor.write(pos);
controle1 = controle;
erro2 = erro1;
erro1 = erro;

```

**Figura 39 - Primeira rotina de controle**  
**Fonte: Autoria própria.**

Neste novo algoritmo os cálculos dos controles proporcional, integral e derivativo são feitos separadamente e são inseridos dentro da rotina de interrupção, tais como na rotina anterior. Após a declaração das variáveis dessa nova função é realizada a leitura da posição da bola na barra, este valor medido é normalizado e, posteriormente, é calculado o erro do sistema, que pode ser representado pela diferença entre o *setpoint* e o valor medido pelo sensor. Tais expressões podem ser notadas na Figura 40.

```

void interrupcao()
{
    double cont_proporcional, cont_integral, cont_derivativo, Ts, erro, sppmm;

    Ts = 0.05;

    pos_esfera = lerPosicao(); //Leitura da posição da bola

    sppmm = calc_esfera(pos_esfera, distancia_minima, distancia_maxima, -1, 1); //Normalização do valor lido

    erro = setpoint - sppmm; //Cálculo do erro

```

**Figura 40 - Cálculo do erro do controlador**  
**Fonte: Autoria própria.**

Logo após a operação envolvendo o erro ser realizada, serão feitos os cálculos dos controladores. O primeiro será o controlador proporcional, seguido do

integral e por último o derivativo. Depois de obter esses três valores, eles serão somados para resultar no valor do sinal de controle final. Caso esse valor calculado ultrapasse o limite mínimo ou máximo imposto, ele será apontado como -1 ou +1 respectivamente. O passo seguinte é desnormalizar o valor do controle para ele atuar no servomotor, conforme mostra a Figura 41, que apresenta os cálculos explicados neste parágrafo.

```
cont_proporcional = Kp * erro; //Controlador proporcional

cont_integral = Ki * (erro + erro1) * Ts / 2 + controle_prev; //Controlador integral

cont_derivativo = Kd * (erro - erro1) / Ts; //Controlador derivativo

controle = cont_proporcional + cont_integral + cont_derivativo; //Sinal de controle final

if (controle > 1) {
    controle = 1;
}

if (controle < -1) {
    controle = -1;
}

pos = calc_posicao(controle, -1, 1, pos_45negativo, pos_45positivo); //Desnormalização do sinal de controle

controle_prev = cont_integral; //Atualização dos valores do controlador

erro1 = erro; //Atualização do erro

motor.write(pos); //Atuação do sinal de controle no motor

}
```

**Figura 41 - Cálculo do sinal de controle do sistema**

**Fonte: Autoria própria.**

Após todas essas mudanças realizadas no *firmware* de operação e na estrutura mecânica do conjunto, o sistema de controle bola-barra desenvolvido apresentou uma melhora significativa na leitura da posição da esfera na haste e, desta forma, foi possível controlar a posição da bola.

## 5 CONSIDERAÇÕES FINAIS

Este projeto surgiu após algumas análises que evidenciaram que os laboratórios do Departamento Acadêmico de Eletrônica da Universidade Tecnológica Federal do Paraná, câmpus Curitiba, não apresentavam nenhum dispositivo de controle do tipo bola-barra. Este modelo possibilita o estudo de sistemas reais instáveis, os quais possuem custos elevados de aquisição, além de serem perigosos e de grande porte.

A primeira fase do trabalho foi a pesquisa sobre os sistemas bola-barra. Durante a busca sobre este assunto, verificou-se alguns modelos existentes no mercado, além de outros dispositivos criados com fins acadêmicos. Após essa pesquisa realizada definiu-se que o sistema deveria ser de baixo custo e os componentes utilizados deveriam ser de fácil aquisição. Para a estrutura mecânica foi mandado produzir um suporte para a barra e um para o motor, já os demais itens são comerciais. Para o circuito eletrônico foi usado uma placa do Arduino Mega 2560, pois além de ser adquirido facilmente, é de simples programação. Todos os outros componentes eletrônicos deste projeto são compatíveis com o Arduino e são fáceis de manusear ou programar.

Durante a execução deste trabalho surgiram alguns problemas que exigiram tomadas de decisões rápidas por parte da equipe. A primeira delas é referente à mudança da barra, devido ao primeiro modelo utilizado apresentar problemas em relação ao equilíbrio da esfera sobre a mesma, além de dificultar a fixação do suporte do sensor e a leitura da distância da bola. A solução adotada para este problema foi a substituição da barra por uma guia linear de aço galvanizado. Outro problema encontrado foi em relação ao sensor ultrassônico. Apesar de ser viável economicamente, este sensor apresenta dificuldades em medir a distância em objetos com a superfície curva, como a esfera utilizada no projeto. Para resolver esta dificuldade foi necessário verificar profundamente o funcionamento do componente e adaptar a rotina de leitura do sensor dentro do *firmware* de operação.

Este sistema pode ser aprimorado substituindo o sensor ultrassônico por outro sensor com maior precisão como, por exemplo, uma fita resistiva. Outra melhora a ser realizada é a integração entre o Arduino e o *software* MATLAB, para

analisar as respostas do controlador durante o funcionamento do projeto. Apesar de não possuir essas características, este trabalho alcançou seu objetivo final, ou seja, a criação de um sistema de controle do tipo bola-barra para fins didáticos.

## REFERÊNCIAS

- ARAÚJO, F. M. U. **Sistemas de Controle**. 2007. Disponível em: <<http://www.dca.ufrn.br/~meneghet/FTP/Controle/scv20071.pdf>>. Acesso em: 23 mar. 2016.
- ARDUINO. **Arduino UNO**. 2016a. Disponível em: <<http://www.arduino.org/products/boards/arduino-uno>> Acesso em: 12 set. 2016.
- ARDUINO. **Arduino MEGA 2560**. 2016b. Disponível em: <<http://www.arduino.org/products/boards/arduino-mega-2560>>. Acesso em: 12 set. 2016.
- ARDUINO E CIA. **Shield LCD 16x2 com Keypad**. 2013. Disponível em: <<http://www.arduinoecia.com.br/2013/08/arduino-shield-lcd-16x2-com-keypad.html>>. Acesso em: 10 ago. 2016.
- BEGA, E. A. et al. **Instrumentação Industrial**. 3. ed. Rio de Janeiro: Interciência, 2011.
- CAMPOS, M. C. M. M.; TEIXEIRA, H. C. G. **Controles típicos de equipamentos e processos industriais**. 1. ed. São Paulo: Edgard Blucher, 2006.
- CHOI, C; EVANKO, D.; DORSETT, A. **A ball-on-beam system with an embedded controller**. 2008. Disponível em: <<http://www.asee.org/documents/sections/northeast/2008/A-Ball-on-Beam-System-with-an-Embedded-Controller.pdf>>. Acesso em: 15 ago. 2015.
- DENARDIN, G. W. **Apostila Microcontroladores**. 2011. Disponível em: <[http://www.pessoal.utfpr.edu.br/gustavo/apostila\\_micro.pdf](http://www.pessoal.utfpr.edu.br/gustavo/apostila_micro.pdf)> Acesso em 10 mar. 2014.
- DORF, R. C.; BISHOP, R. H. **Sistemas de controle modernos**. 12. ed. Rio de Janeiro: LTC, 2013.
- ELETRÔNICA PROGRESSIVA. **Microcontroladores – o que são, para que servem e onde são usados**. 2014. Disponível em: <<http://www.eletronicaprogressiva.net/2014/08/Microcontroladores-O-que-sao-Para-que-servem-Onde-sao-usados.html>>. Acesso em: 07 set. 2016.

EPUSP. **Controle de um servo motor**. 2014. Disponível em: <[http://www2.pcs.usp.br/~labdig/pdf/files\\_2014/control-servo-semester.pdf](http://www2.pcs.usp.br/~labdig/pdf/files_2014/control-servo-semester.pdf)>. Acesso em: 18 jul. 2016.

EVANS, M.; NOBLE, J.; HOCHENBAUM, J. **Arduino em ação**. 1. Ed. São Paulo: Novatec, 2013.

FILIFELOP. **Sensor de Distância Ultrassônico HC-SR04**. 2016a. Disponível em: <<http://www.filifelep.com/pd-6b8a2-sensor-de-distancia-ultrassonico-hc-sr04.html>>. Acesso em: 07 set. 2016.

FILIFELOP. **Servo TowerPro MG995 Metálico**. 2016b. Disponível em: <<http://www.filifelep.com/pd-1469f0-servo-towerpro-mg995-metalico.html>>. Acesso em: 08 set. 2016.

FRANKLIN, G. F.; POWELL, J. D.; EMAMI-NAEINI, A. **Sistemas de controle para engenharia**. 6. ed. Porto Alegre: Bookman, 2013.

JÚNIOR, R. C. **Malha Aberta x Malha Fechada 2**. 2013. Disponível em: <[http://www.corradi.junior.nom.br/malha\\_abertaxfechada\\_2013\\_2.pdf](http://www.corradi.junior.nom.br/malha_abertaxfechada_2013_2.pdf)>. Acesso em: 10 out. 2015.

MANTOVANI, S. C. A. **Servo Motor**. 2013. Disponível em: <<http://www.feis.unesp.br/Home/departamentos/engenhariaeletrica/aula-4---servo-motor-13-03-2013-final.pdf>>. Acesso em: 17 jul. 2016.

MARTINS, N. A. **Sistemas Microcontrolados: uma abordagem com o microcontrolador PIC16F84**. São Paulo: Novatec, 2005.

MCROBERTS, M. **Arduino Básico**. 1. Ed. São Paulo: Novatec, 2011.

MEDEIROS, R. L. P. **Desenvolvimento e aplicação de motores de corrente contínuas virtuais aplicadas nas aulas laboratoriais de controle de sistemas**, Universidade Federal do Pará, Faculdade de Engenharia Elétrica, Belém. 2011.

NATANAEL. **Usando o LCD Keypad Shield com a biblioteca LiquidCrystal.h**. 2012. Disponível em: <<https://blogdonatanael.blogspot.com.br/2012/12/usando-o-lcd-keypad-shield-com.html>>. Acesso em: 08 ago. 2016.

NISE, N. S. **Engenharia de Sistemas de Controle**. 6. Ed. Rio de Janeiro: LTC, 2012.

OGATA, K. **Engenharia de controle moderno**. 3. ed. Rio de Janeiro: Prentice-Hall, 1998.

SCALABRIN, R. F. et al. **Projeto de controladores robustos  $H_2$  e  $H_\infty$  por realimentação de estados através da abordagem por LIMS**. In: CONGRESSO BRASILEIRO DE EDUCAÇÃO EM ENGENHARIA, 2014, Juiz de Fora. Engenharia: múltiplos saberes e atuações. Manaus: Universidade do Estado do Amazonas, 2014. Disponível em <<http://www.abenge.org.br/cobenge-2014/Artigos/129781.pdf>>. Acesso em: 08 ago. 2015.

SOUZA, D. J. **Desbravando o PIC**. 12. ed. São Paulo: Editora Érica Ltda, 2010.

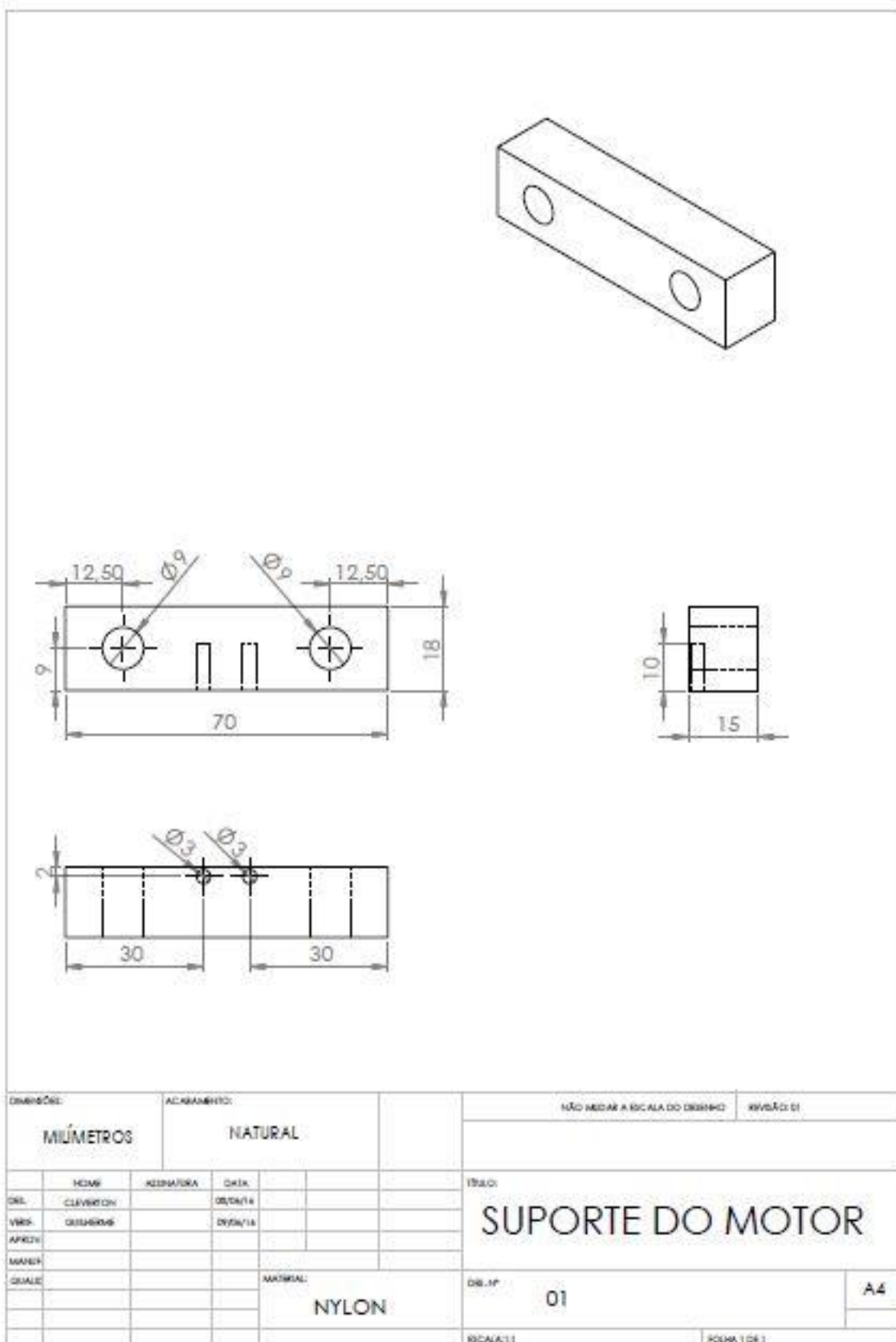
THOMAZINI, D.; ALBUQUERQUE, P. U. B. **Sensores industriais: fundamentos e aplicações**. 3. ed. São Paulo: Érica, 2007.

WANG, W. **Control of a Ball and Beam System**. 2007. Disponível em <[http://data.mecheng.adelaide.edu.au/robotics/projects/2007/BallBeam/Wei\\_Final\\_Thesis.pdf](http://data.mecheng.adelaide.edu.au/robotics/projects/2007/BallBeam/Wei_Final_Thesis.pdf)>. Acesso em: 15 ago. 2015.

WEG. **Variação de velocidade**. Jaraguá do Sul: Weg, 2003.

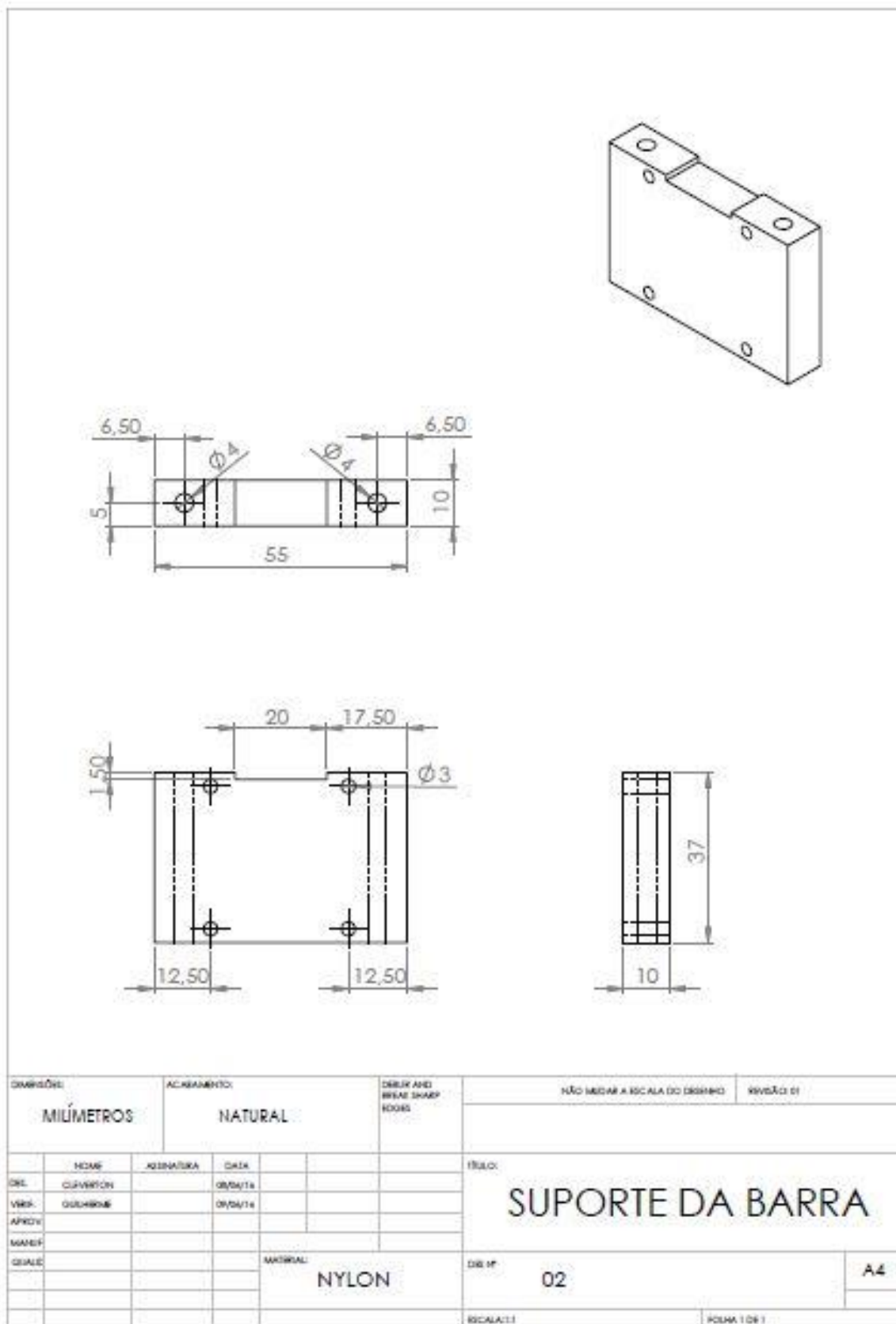
WELLSTEAD, P. **Ball and beam 1: Basics**. 2000. Disponível em <<http://www.control-systems-principles.co.uk/whitepapers/ball-and-beam1.pdf>>. Acesso em: 08 ago. 2015.

## APÊNDICE A – DESENHO 2D: SUPORTE DO MOTOR





## APÊNDICE B – DESENHO 2D: SUPORTE DA BARRA



## APÊNDICE C – DESENHO 2D: BASE

