



UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO E AUTOMAÇÃO
CURSO DE ENGENHARIA MECATRÔNICA

PROJETO E CONSTRUÇÃO DE UM SISTEMA PLACA-BOLA PARA FINS DIDÁTICOS

Yasmin Honório de Medeiros

Orientador: Prof. Dr. Fábio Meneghetti Ugulino de Araújo

Natal/RN
Junho de 2016

PROJETO E CONSTRUÇÃO DE UM SISTEMA PLACA-BOLA PARA FINS DIDÁTICOS

Yasmin Honório de Medeiros

Orientador: Prof. Dr. Fábio Meneghetti Ugulino de Araújo

Monografia apresentada à Banca Examinadora do Trabalho de Conclusão do Curso de Engenharia Mecatrônica, em cumprimento às exigências legais como requisito parcial à obtenção do título de Engenheiro Mecatrônico.

Natal/RN
Junho de 2016

Agradecimentos

Ao meu pai, Ivanilson Medeiros, pelo apoio e ajuda durante a execução desse trabalho.

Ao meu orientador, Fábio Meneghetti Ulugino de Araújo, e ao meu supervisor, Alcemy Gabriel Vitor Severino, pelo suporte técnico e orientação.

À Ícaro Bezerra Queiroz de Araújo, pela solução de dúvidas na utilização de softwares.

Ao professor Márcio Valério de Araújo, pelo guia na fabricação das peças utilizadas.

À Válber César Cavalcanti Roza, pela ajuda na concepção da ideia deste projeto.

À Leonardo Silva Souza, pelo apoio e opiniões durante a fase de testes.

Aos meus colegas de graduação, pelas opiniões, revisões e socorros nas horas inusitadas.

Resumo

Com o avanço da tecnologia, faz-se necessário ter cada vez mais precisão na realização de processos. Este projeto tem como intenção viabilizar a implementação de diferentes estratégias de controle que serão aplicadas em um controle de posição, que é uma área de variadas aplicações no âmbito da robótica, através da fabricação de uma planta com fins didáticos. Para isso, será utilizado um sistema placa-bola, acionada por servos motores, onde a posição da bola é obtida por captura de imagens e a posição desejada determinada por uma *interface* de controle. Para isso, a proposta é produzir uma mesa quadrada com centro estável e angulação variável, que possa ser movida através do uso de servomotores de alta precisão. Os testes foram realizados de forma separada, dividindo em acionamento dos motores, processamento de imagem e simulação do sistema de controle. A partir do funcionamento de cada parte isolada, espera-se que o mesmo aconteça de forma integrada, ou seja, que haja comunicação do controlador com a leitura e envio de dados para os motores, do controle da planta com a posição adquirida por processamento de imagens e dos blocos do sistema com os dos servomotores. O trabalho foi condizente com o proposto ao mostrar como deve ser feita a comunicação entre o computador e os motores, bem como em capturar a posição real do objeto com frequência. Assim, é possível testar diversas estratégias de controle.

Palavras-chave: Processamento de Imagem; Controle de Posição; Ball and Plate; Automação.

Abstract

With the advance of technology, the precision of processes parameters has become more important. This document was made with the intention of creating an opportunity to study different types of controllers by making a position control of a ball on a surface, which is an area of many applications on the robotics field, through the manufacturing of an asset with didactic purposes. To do that, it was needed a table as base, servomotors for movement, a camera to acquire images, an interface for the user to determine the desired position and a ball as the main object. That way, it will be possible to create a control system on a computer through softwares where the voltage for the motors should be the output. For that, the proposition is to produce a square table with a stable center and variable angles that can be moved with the utilization of high-precision servomotors on two axes. The tests were first made separately, divided in motor operation, image processing and simulation of the control system. When all the pieces are working well isolated, they should be connected in a way that is possible to approve the behavior of the project. This means that the motors control must be connected to their input/output signal, the plant control must be connected to the position defined by image processing and those blocks must be connected with each other. This project was efficient on showing how the communication between a computer, the motors and the table should be made and on capturing the object position in real time. With that being accomplished, it is possible to test distinct controllers and apply these techniques in others systems with some adaptations.

Keywords: Image Processing; Position Control; Ball and Plate; Automation.

Sumário

Lista de Figuras	iii
Lista de Tabelas	v
1 Introdução	1
2 Fundamentação Teórica	3
2.1 Processos de Fabricação	3
2.1.1 Furação	4
2.1.2 Serragem	6
2.1.3 Torneamento	6
2.1.4 Polimento	6
2.2 Junta Homocinética	7
2.3 Processamento Digital de Imagens	8
2.3.1 Identificação de Regiões	10
2.3.2 Operações em Imagens	10
2.3.3 Histograma	11
2.3.4 Limiarização	11
2.3.5 Identificação de Objetos	11
2.4 Modelagem Matemática e Controle	12
2.4.1 Modelagem Física	13
2.4.2 Controle	14
2.5 Softwares Utilizados	19
2.5.1 Matlab	19
2.5.2 Simulink	21
3 Metodologia	23
3.1 Manufatura	23
3.1.1 Material e Peças	24
3.1.2 Processo de Fabricação	25

3.2	Controle do Sistema	28
3.3	Processamento de Imagem	30
4	Resultados e Discussões	31
4.1	Calibração do PDI	31
4.2	Simulação do Controlador	32
4.3	Teste de Movimento dos Motores	34
4.4	Teste de Identificação de Objeto	37
5	Conclusão	39
	Referências bibliográficas	39
A	Código para Identificação de Posição do Objeto	42

Lista de Figuras

2.1	Parâmetros de usinagem	4
2.2	Esquemático de um processo de furação	5
2.3	Furadeira de bancada com avanço manual	5
2.4	Torno mecânico	7
2.5	Funcionamento de uma junta homocinética	8
2.6	Processos de manipulação de imagens	9
2.7	Diagrama de blocos de um dos eixos	13
2.8	Controle em malha aberta	14
2.9	Controle em cascada	16
2.10	Diagrama de blocos para o sistema real	17
2.11	Esquema de blocos para controle em cascata	18
2.12	Esquema de blocos para controle derivativo real	18
2.13	Interface do Matlab	21
2.14	Interface de blocos do Simulink	22
3.1	Modelo do projeto mecânico	23
3.2	Junta homocinética com hastes	25
3.3	Motor SRV02 da Quanser	26
3.4	Encaixe das juntas homocinéticas na mesa	26
3.5	Base final com motores	27
3.6	Modelo do projeto em vista isométrica	27
3.7	Sistema mecânico final	28
3.8	Diagrama de blocos para motores e planta	29
3.9	Diagrama de blocos completo	30
4.1	Imagem processada	31
4.2	Teste do sistema de controle	32
4.3	Novo diagrama de blocos para o controle	32
4.4	Teste 1 do controle PID	33
4.5	Teste 2 do controle PID	34

4.6	Programa do Simulink para teste do sistema mecânico	35
4.7	Varição de ângulo dos motores durante teste	35
4.8	Varição de tensão enviada para o motor A	36
4.9	Varição de tensão enviada para o motor B	36
4.10	Movimento do objeto no eixo X	37
4.11	Movimento do objeto no eixo Y	37
4.12	Movimento relativo do objeto	38

Lista de Tabelas

3.1	Peças Utilizadas	24
3.2	Orçamento	25

Capítulo 1 - Introdução

O conceito de sistemas é amplo e pode se estender a diversas áreas da ciência. Contudo, para a engenharia, eles são definidos por Morton (1959) como uma integração composta de dispositivos, estruturas especializadas e sub-funções, onde os métodos tendem a otimizar a função geral do sistema de acordo com a ponderação dos objetivos e para utilizar a maior capacidade de suas partes. Isso implica que um sistema físico é considerado um conjunto de entidades materiais que estão vinculadas e possuem processos individuais com um objetivo em comum.

Essas partes podem incluir tudo que é necessário para produzir resultados relevantes para o sistema, como *hardwares*, *softwares*, documentos, instalações, etc. Elas têm como objetivos influenciar propriedades, qualidades, características, funções, comportamento ou performance do sistema como um todo. Contudo, com a implementação de um controle das variáveis, o comportamento do sistema deve ter mais chances de ser como esperado.

Conseguir controlar uma planta consiste em ter sucesso na atuação sobre a mesma de forma a obter respostas de acordo com objetivos previamente estabelecidos. Isso pode ser incluído em todos os tipos de modelos, diagramas, simulações, equações e etc. Modelar o conjunto é essencial para a implementação de um controle, pois ajuda a entender o funcionamento do mesmo. Para tornar a modelagem matemática de sistemas eletromecânicos possível é necessária uma simplificação, por isso deve ser feita a separação dos processos. Ainda assim, podem aparecer problemas relacionados à eficácia do controle para situações extremas.

Antes da década de 30, a modelagem consistia em equações diferenciais, até que Minorsky propôs a primeira estrutura de um controlador em 1922. Em 1940, controladores PID já eram comercializados (principalmente com o surgimento do método Ziegler-Nichols em 1942). Até a década de 50, sistemas SISO conseguiram ser interpretados e o controle clássico surgiu. Na década de 70, os primeiros computadores comerciais começaram a ser utilizados como controladores em sistemas de controle de grande porte. Nessa época já existia controle ótimo, moderno, com variáveis de estado e multivariável. Para a redução de custos, o Controlador Lógico Programável foi implementado, sendo

um computador projetado para trabalhar no âmbito industrial. Desde então, o controle não-linear, adaptativo e preditivo são os que têm sido mais explorados.

Um dos problemas que começaram a aparecer com o desenvolvimento da robótica foi a detecção de posição. A tentativa de criar sistemas autônomos passou a incluir um controle de deslocamento de forma a deixar precisa seus movimentos e medições. Com isso, diversos sistemas mais simples foram utilizados para testar os tipos de controle com essa finalidade. Atualmente, o controle de posição tem aplicações variadas, desde protótipos caseiros, trabalhos de fins acadêmicos à sistemas complexos.

Em busca de uma precisão maior nos movimentos, motores de passo foram introduzidos a essa área. De acordo com WhatIs (2005), em tradução livre, "motor de passo é um tipo especial de motor elétrico que move em incrementos, ou passos, ao invés de girar suavemente como um motor convencional faz. O tamanho do incremento é medido em ângulo e pode variar com sua aplicação". Sendo assim, quanto menor o passo, mais preciso o controle realizado por ele será.

A proposta do projeto é produzir uma mesa com centro estável e angulação variável, através do uso de servomotores de alta precisão em dois lados perpendiculares entre si, comumente conhecido como *Ball and Plate*. Dessa forma, espera-se que seja possível movimentar uma esfera qualquer colocada em cima do plano e realizar um controle de posição sob ela. O propósito da criação dessa iniciativa é que o trabalho seja utilizado para fins didáticos, para a implementação e teste de variadas estratégias de controle. Ele busca unir as áreas de modelagem, controle, fabricação e processamento de imagens, de forma a gerar um mecanismo que pode ser reconfigurado e utilizado para simulações práticas.

Este documento tem como objetivo dar a base teórica para a execução do projeto, bem como explicar detalhadamente tudo o que foi realizado para o seu funcionamento. Ele será dividido entre a parte mecânica, de processamento de imagens e o controle, separadamente, para que depois seja realizada a conexão entre elas. Por fim, esse arquivo mostrará alguns dos testes feitos e dará uma resposta para a implementação desta aplicação.

Capítulo 2 - Fundamentação Teórica

Este capítulo tem como objetivo fundamentar os conteúdos necessários para o desenvolvimento do projeto como um todo. Ele será separado entre os processos utilizados para a fabricação dos componentes mecânicos, a teoria por trás de processamento de imagens juntamente com as funções essenciais para sua aplicação e a implementação de controle básico de variável através da modelagem matemática do sistema. Com a conclusão dessa seção, deve ser possível compreender e colocar em prática todos os métodos descritos na metodologia do processo.

2.1 Processos de Fabricação

Durante a construção do projeto foi necessário a utilização de diversos processos de fabricação. Segundo Losekann (2001), "a fabricação pode ser definida como a arte e a ciência de transformar os materiais em produtos finais utilizáveis e - num contexto de economia de mercado - rentáveis. O processo global de fabricação é uma série de interações complexas entre materiais, máquinas, pessoas e energia, começando com a criação de peças individuais que irão finalmente constituir, através de operações de montagem, um produto final (...) Sempre existe uma interação entre o material da peça e o processo de fabricação, na qual um exerce restrições sobre o outro".

Fabricar envolve desde procedimentos de união à retirada de matéria, contudo neste trabalho apenas processos de usinagem foram necessários. Mais adiante, poderá ser vista uma relação e descrição dos principais métodos aplicados nesse projeto, de forma a dar base teórica suficiente para o entendimento da realização da produção como um todo.

O ato de usinar pode ser explicado como um procedimento que realiza a remoção de material, com o objetivo de atingir uma forma específica. A usinagem produz peças com bom acabamento superficial e geometrias relativamente complexas, combinada a um limite de tolerância dimensional útil para elevadas especificações. As desvantagens da usinagem podem incluir desperdício de material e incapacidade de melhorar problemas provenientes de uma fundição.

Os principais parâmetros a serem considerados em um processo de usinagem são:

profundidade de corte, ângulo de direção da ferramenta, tamanho do avanço e velocidade de corte. Os movimentos mais importantes a serem identificados são o de avanço e o de corte, onde a retirada efetiva do material consistirá na combinação de movimentos. A Figura 2.1 ilustra o conceito dessas variáveis.

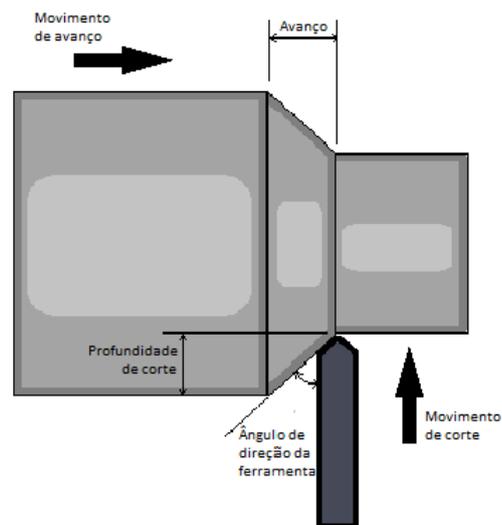


Figura 2.1: Parâmetros de usinagem

Para o auxílio durante a usinagem, existem máquinas adaptadas que podem efetuar a função com melhor eficiência, como tornos e fresadoras. Estes instrumentos são mais utilizados quando o processo está para ser feito em uma matéria de maior rigidez.

2.1.1 Furação

A furação é um tipo de processo de usinagem, sendo especificamente destinado à obtenção de furos. Para a sua execução, é necessário o auxílio de uma ferramenta multicortante, comumente chamada de broca. O procedimento é realizado movimentando a peça ou a ferramenta de corte por uma trajetória contínua coincidindo com a estrutura da peça. O processo de furação de uma peça pode ser realizado de forma manual ou automática, de modo que o método manual possui baixo grau de precisão.

A forma da concepção dos furos pode ser classificada em quatro tipos:

- Furação em cheio - realiza a abertura de um furo do início ao fim;
- Escareamento - alargamento de um furo já existente;
- Furação escalonada - obtém um furo com mais de um diâmetro;
- Furação de centros - produz um furo de centro visando uma operação posterior.

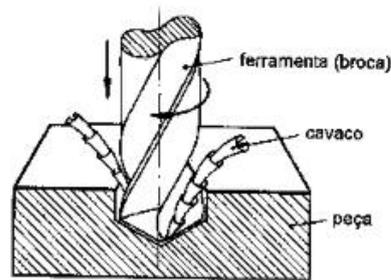


Figura 2.2: Esquemático de um processo de furação

Fonte: Ebah. Acesso em: Maio de 2016.

É recomendável utilizar mecanismos de apoio e medição ao realizar um furo. O uso de furadeiras é uma maneira comum de garantir a eficiência do procedimento. Elas variam de acordo com a aplicação e podem ser, por exemplo, manuais (portáteis porém com baixa qualidade de confecção) ou de bancada. Nesse projeto, ambos os tipos foram usados, sendo que a furadeira de bancada realizou furos de movimentação mais simples porém com melhor acabamento. Ela também pode possuir avanço manual, como é o caso da escolhida para a execução desse projeto.

A Figura 2.3 mostra o modelo da furadeira de bancada utilizada para realizar parte dos furos, encontrada no Laboratório de Robótica presente no Departamento de Computação e Automação da UFRN. Para este modelo a altura de bandeja e a profundidade do furo são controlados manualmente pelo operador.



Figura 2.3: Furadeira de bancada com avanço manual

Fonte: Rufflav. Acesso em: Maio de 2016.

2.1.2 Serragem

O corte realizado com a utilização de serras é conhecido como serragem. Para sua execução, a ferramenta de corte deve girar e/ou se deslocar, de forma a retirar material com ajuda da inércia do movimento constante. O processo pode ser retilíneo ou circular, onde no segundo a ferramenta gira em torno de seu próprio eixo.

A serragem também pode ser manual ou ter seu instrumento acoplado a uma máquina. De acordo com Sousa (2008), o serrote manual é uma ferramenta portátil utilizada para o corte de pequenas peças e operações que necessitam da versatilidade que ele fornece. Esse tipo de método foi utilizado no projeto pela simplicidade de aplicação, principalmente uso em materiais curtos.

2.1.3 Torneamento

Torneamento foi definido por Souza (2011) como um processo mecânico de usinagem destinado à obtenção de superfícies de revolução com auxílio de uma ou mais ferramentas monocortantes, de forma que a peça gira em torno do eixo principal de rotação da máquina e a ferramenta se desloca simultaneamente segundo uma trajetória coplanar com o referido eixo.

Nesse método de usinagem a peça possui movimento de rotação e a ferramenta possui movimentos de avanço e penetração. Ele é utilizado para a modificação de materiais de revolução e pode ser classificado quanto à forma da trajetória, sendo ela retilínea ou curvilínea.

O torno mecânico é o principal equipamento usado para torneiar. Ele funciona girando o objeto e o mesmo é geralmente fixado por meio de três castanhas dispostas de 120 graus entre si e apertadas simultaneamente. A Figura 2.4 mostra o torno utilizado para os procedimentos dentro deste projeto.

2.1.4 Polimento

No final de um processo de usinagem, deseja-se obter uma peça com superfície que apresente características apropriadas, a fim de garantir segurança, confiabilidade e durabilidade ao componente. O acabamento não está diretamente relacionado à elementos de textura ou padrões, porém um bom acabamento implica em um baixo valor de rugosidade. Ela é importante porque influencia em diversas qualidades dos elementos, como a qualidade de deslizamento, resistência ao desgaste, transferência de calor, vedação e aderência.



Figura 2.4: Torno mecânico

Cada processo de fabricação gera um determinado tipo de acabamento, mas todas apresentam irregularidades. Na usinagem, os parâmetros de corte exercem uma grande influência na integridade da superfície do componente. Um aumento do avanço tende a aumentar a rugosidade e, em baixas velocidades, o mau acabamento também pode ser causado por arestas postizas de corte, uma vez que pedaços do material aderem à peça. O cavaco é uma das causas de cortes de um usuário ao ter contato com a peça pós usinagem.

Para dar um bom caráter às peças, é recomendado que elas passem por um procedimento de finalização de acabamento superficial. Isso implica em tratar as superfícies para que elas tenham uma aparência estética melhor.

2.2 Junta Homocinética

De acordo com RUVILLE (2015), a tarefa principal das juntas homocinéticas é transmitir sem fricção uma força de rotação. Juntas fixas implica que elas não permitem movimentos axiais, de forma a terem o suporte do pivô fixo, enquanto as roláveis tornam possível um movimento angular simultâneo ao movimento axial.

O modelo de junta homocinética utilizado como base para a adaptação nesse projeto foi o que utiliza elementos esféricos, que é um dos tipos de juntas roláveis. Essa classificação é frequentemente encontrada em juntas de carros no sistema de transmissão e possui esferas para ajudar no deslizamento multidirecional das articulações. Na Figura 2.5, é possível ver uma ilustração da junta rolável com esferas em funcionamento.

Para esse trabalho, foi feita uma junta homocinética adaptada, onde um dos eixos tem

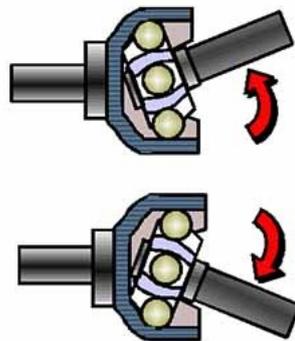


Figura 2.5: Funcionamento de uma junta homocinética

Fonte: BestCars (2015). Acesso em: Maio de 2016.

em sua extremidade uma esfera central, de modo que ela estará em contato com alguns pontos das paredes do envoltório, mas não com ele inteiro. A intenção era simplificar a fabricação da junta, visto que não havia necessidade de alta precisão, pois os movimentos seriam curtos e o peso sobre ela seria facilmente suportado.

2.3 Processamento Digital de Imagens

O processamento de imagens tem aplicações e utilidade em diversas áreas de conhecimento, como raios x, biometrias, rotulação de produtos, leitura de satélites, restauração de imagens, exploração geológica, inspeção de produtos, etc. Ainda assim, com sua difusão, ele é muitas vezes usado para substituir métodos, de maneira a ser mais preciso ou extrair mais informações. Neste projeto, ele será utilizado como localizador de objeto para rastrear a posição a cada instante da esfera.

De acordo com Gonzales e Woods (1992), uma imagem pode ser separada numa função de dois parâmetros, correspondente a coordenadas espaciais e o valor da função representa a intensidade ou nível de cinza da figura naquele ponto. Imagem digital é definida quando os valores das coordenadas e da intensidade são finitos. No campo de processamento digital, cada localização e valor podem ser identificado pelo nome *pixel*.

Marques e Vieira (1999) afirmam que o grande desafio permanece sendo o de fazer com que os sistemas de visão artificial trabalhem em diferentes condições de luminosidade, contraste, posicionamento relativo dos objetos em uma cena sem perder a capacidade de interpretar a cena, de forma análoga a nossa capacidade de reconhecer um amigo ou parente com relativa facilidade, independentemente de ele estar usando óculos ou não, ter deixado crescer a barba ou estar no carro ao lado do nosso em uma esquina num final

de tarde, onde não dispomos de outra imagem senão a vista de perfil e onde as condições de luminosidade são bastante inferiores às que obteríamos ao meio-dia.

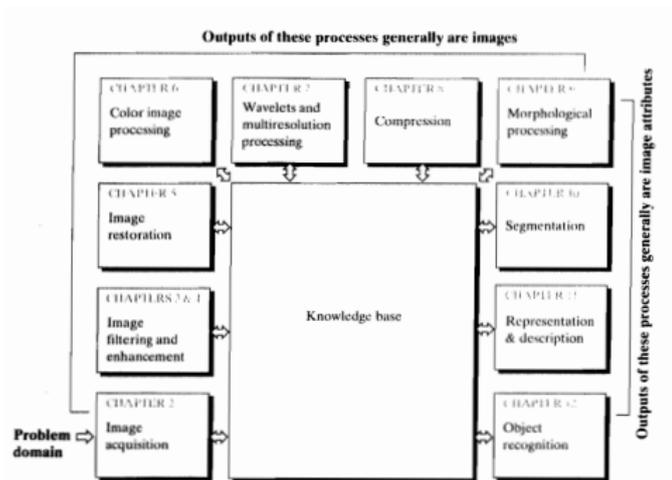


Figura 2.6: Processos de manipulação de imagens

Fonte: Gonzales e Woods (1992). Acesso em: Maio de 2016.

A Figura 2.6 mostra os diferentes tipos de manipulação que o processamento de imagens abrange. É possível identificar que o caso utilizado aqui envolve aquisição de imagens como entrada de processo e resulta em reconhecimento de objetos. Outras aplicações são consequências desses fins mostrados na figura.

De acordo com Gonzales e Woods (1992), entende-se que, para implementar um método de processamento de imagens, os componentes básicos fundamentais são:

- *Hardware* especializado - realiza operações lógicas em figuras;
- Computador - para executar as principais operações;
- *Software* de processamento - contém pacotes específicos para a aplicação;
- Armazenamento - essencial, pois imagens e vídeos ocupam grande espaço de memória;
- *Display* - monitores para inspeção do processo;
- Dispositivos de *hardcopy* - aparelhos de captura de imagens;
- Rede de comunicação - necessária para transmissão de dados.

A representação de imagens consiste em descrevê-la (de forma discreta) por um vetor de no mínimo três elementos, sendo dois deles sua posição no plano e o terceiro a razão entre sua altura e largura (intensidade média). Nessa etapa, deve ser possível extrair as informações com precisão, de forma a distinguir dígitos similares. Para a identificação de

imagens coloridas, utiliza-se três planos de componentes de cor, como, por exemplo, os de vermelhos, azuis e verdes. A combinação desses elementos corresponde à cor em cada ponto.

De acordo com Marques e Vieira (1999), para obter uma imagem digital de qualidade semelhante a de uma imagem de televisão P&B, são necessários 512×512 *pixels* e 128 níveis de cinza. Em geral, 64 níveis de cinza são considerados suficientes para o olho humano. Apesar disto, a maioria dos sistemas de visão artificial utiliza imagens com 256 níveis de cinza. Os *softwares* de processamento de imagens em geral confirmam isso, pois identificam a intensidade do *pixel* por um valor entre 0 e 255.

2.3.1 Identificação de Regiões

O primeiro conceito necessário para entender como separa regiões de uma imagem é o de conectividade. Diversos critérios podem ser aplicados para identificar se dois *pixels* estão ou não conectados. O comum é adotar uma regra para classificar a posição e adicionar um limiar de proximidade de tons. Sendo assim, entende-se que dois *pixels* estão conectados se eles são vizinhos e possuem intensidade próxima. A determinação da vizinhança pode ser de 4 (vizinhos em direções horizontais ou verticais) ou 8 (inclui *pixels* a 1 de distância todas as direções).

2.3.2 Operações em Imagens

A seguir estão descritas as principais operações algébricas e os efeitos que elas têm em matrizes de imagens:

- Adição - cada *pixel* da imagem resultante terá o seu valor de intensidade definido pela soma dos tons desse mesmo ponto nas outras imagens. A adição de valores positivos resulta num tom mais claro.
- Subtração - os *pixels* do resultado terão como valor a diferença das intensidades dos pontos de suas entradas. Esse valor pode ser positivo ou negativo.
- Multiplicação - como a operação normalmente faz, o valor do ponto resultante será a multiplicação escalar dos valores de intensidade dos *pixels* de entrada.
- Divisão - esse cálculo dará como resultado a razão dos valores de entrada, para cada *pixel* das imagens.
- Convolução - faz-se esse método de maneira discreta, utilizando máscaras em forma de matrizes, espelhando-a e movendo-a ponto a ponto pela imagem a ser implementada e fazendo uma multiplicação escalar em cada passo, em que a soma das

multiplicações dos pares de pontos darão o resultado da intensidade naquele *pixel*.

2.3.3 Histograma

De acordo com Marques e Vieira (1999), o histograma de uma imagem é simplesmente um conjunto de números indicando o percentual de *pixels* naquela imagem que apresentam um determinado nível de cinza. Através da visualização do histograma de uma imagem obtemos uma indicação de sua qualidade quanto ao nível de contraste e quanto ao seu brilho médio (se a imagem é predominantemente clara ou escura).

Essas análises também podem ser utilizadas em imagens coloridas, em que, para cada plano de decomposição, os valores de ocorrência de intensidade são calculados. Computacionalmente falando, os histogramas são implementados em gráficos de barras inicializados em valor nulo e acrescentado a cada par de ponto e intensidade percorrido pela imagem.

2.3.4 Limiarização

Limiarização, também conhecido como *thresholding*, foi definido por Devi (2006) como um processo de conversão de uma imagem em escalas de cinza para uma binária utilizando um limiar como referência. Esse processo consiste em separar a figura em duas regiões, sendo uma o fundo e a outra o objeto, através da transformação do tom, em que os tons abaixo do limiar serão redefinidos com valor mínimo e os acima possuirão novo valor máximo na nova imagem.

Os algoritmos de implementação podem ser separados em globais e adaptativos, onde o primeiro terá um limiar fixo para toda a imagem e o segundo varia o limiar durante o percurso através de alguma condição. Muitos métodos utilizam o histograma para fazer a binarização ou, pelo menos, o cálculo do limiar.

2.3.5 Identificação de Objetos

O reconhecimento de objetos envolve processos de extração de características de imagens. Jain, Kasturi e Schunck (1995) definem esse conceito como um sistema que consegue encontrar objetos do mundo real com uma imagem dele, usando modelos previamente conhecidos. Os maiores problemas nessa área geralmente consistem na implementação matemática dos modelos e na rotulação de objetos. Jain, Kasturi e Schunck (1995) também menciona que a rotulação necessita de reconhecimento parcial e o reconhecimento completo precisa da rotulação.

A detecção de características em imagens ajuda a formar a hipótese de objeto e elas variam de acordo com a implementação e aplicação do criador, que deve levar em conta o tipo de objeto e plano de fundo. Usa-se a segmentação de imagens para separar os objetos do fundo da imagem, de forma a facilitar o reconhecimento de objetos, já que o processo de segmentação costuma ser mais simples.

Para esse projeto, a reconhecimento da forma e das características do objeto não são relevantes, já que só existe um item e os procedimentos serão feitos *pixel a pixel*. Sendo assim, o reconhecimento de objeto consistirá apenas nos passos relativos à identificação do mesmo. Logo, envolverá a rotulação e segmentação da imagem, mas não necessitará de um modelo de objeto nem processos mais complexos.

De acordo com Achary e Ray (2005), em imagens binárias, um objeto é considerado um conjunto de *pixels* de valor 1 (branco) conectados. As principais características de objetos binários são área, perímetro e afins, como pode ser verificado através das variáveis de retorno da função de reconhecimento de objeto presentes no Matlab® (seção 2.5.1).

As principais interferências na utilização de reconhecimento de objetos são definidas por Bouman (2012) como: variação do ponto de vista, iluminação, oclusão, escala, deformação, diferença no mesmo tipo de objeto e plano de fundo. Neste projeto, o ponto de vista e a escala não irão mudar drasticamente, apesar do movimento de inclinação da mesa, pois a câmera está fixa e não terá problemas de oclusão, deformação nem diferentes tipos de objeto, além da esfera utilizada, que foi a mesma em todos os testes realizados.

2.4 Modelagem Matemática e Controle

De acordo com Bassanezi (2002), "a modelagem matemática consiste na arte de transformar problemas da realidade em problemas matemáticos e resolvê-los interpretando suas soluções na linguagem do mundo real". Ele também afirma que ela pode ser utilizada tanto como um método científico de pesquisa como uma estratégia de ensino. Entende-se que, então, a modelagem é a tentativa de explicar um sistema real, sendo muito utilizada para formular maneiras de agir sobre ele.

Um modelo bem estruturado deve conter resultados parciais baseados nas leis da física, que depois podem ser corrigidos. Deve-se levar em conta que o que se usa são apenas aproximações da realidade. É possível perceber que a disposição de dados e seus ajustes têm uma grande importância na eficiência e no entendimento da representação, tornando a solução de problemas mais fácil.

Modelos de sistemas dinâmicos implicam em simulações de diferentes estágios da realidade. Eles envolvem variáveis dependentes e independentes, onde geralmente esta

segunda é o tempo e as outras variam em função dela. A análise das relações entre as variáveis do sistema é essencial para a obtenção de um controle.

De acordo com Dorf e Bishop (2001), os passos para executar um projeto de engenharia de controle e atender necessidades do sistema consistem em estabelecer a finalidade do projeto, selecionar um controlador e ajustar os parâmetros. Eles complementam dando a nota de que os controladores quase sempre consistem em amplificadores somadores. O bom desempenho do controle desejado está diretamente associado ao uso de sensores e atuadores como dispositivos de contato com o sistema.

2.4.1 Modelagem Física

O uso de equações diferenciais é o mais comum para sistemas dinâmicos. Elas podem ser obtidas utilizando as leis físicas do processo, seja ele elétrico, mecânico, hidráulico ou termodinâmico. Este trabalho realiza a modelagem matemática e o controle de posição para um sistema placa-bola. Como o mesmo sistema de atuadores será responsável pelo movimento de cada eixo do plano, a estratégia de controle será idêntica para cada direção. Com base nos processos de modelagem citados em Instructor... (2011) ilustrados pela Figura 2.7, é possível representar o modelo matemático a seguir. Para esse projeto, foi assumido que o servomotor responsável pelo movimento em um eixo não gera interferências no outro.

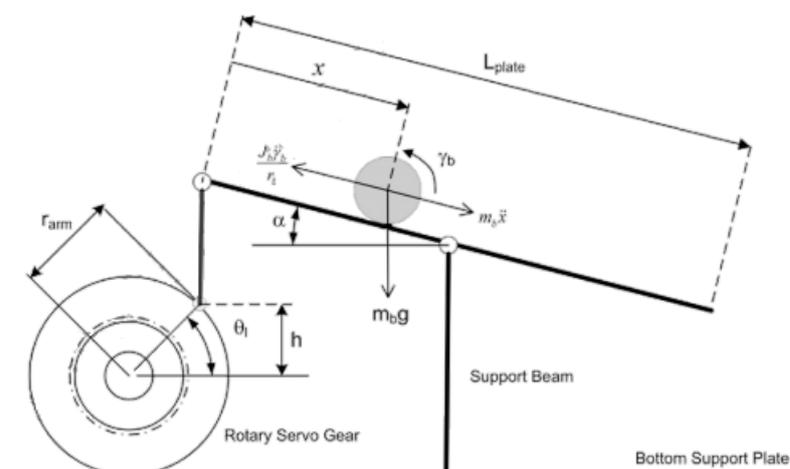


Figura 2.7: Diagrama de blocos de um dos eixos

Dispensando a fricção do objeto com a superfície, a equação da dinâmica responsável por esse sistema, será definida pela soma das forças do sistema, descritas abaixo:

$$\sum F = m_b \frac{d^2x}{dt^2} = F_{x,t}(t) - F_{x,r}(t) \quad (2.1)$$

Onde $F_{x,t}$ representa a inércia do objeto e $F_{x,r}$ é a força da gravidade. Juntas, elas devem ser equivalentes a proporção da massa com a aceleração dinâmica do sistema. Separadamente, elas podem ser calculadas pelas fórmulas a seguir.

$$F_{x,t} = m_b g \text{sen}\alpha(t) \quad (2.2)$$

$$F_{x,r} = \frac{d^2x}{dt^2} J_b \frac{1}{r_b^2} \quad (2.3)$$

Substituindo-se as equações 2.2 e 2.3 na 2.1, constata-se que é possível obter uma relação entre a posição x e todos os parâmetros do sistema através de:

$$\frac{d^2x}{dt^2} = \frac{m_b g \text{sen}\alpha(t) r_b^2}{m_b r_b^2 + J_b} \quad (2.4)$$

Sabendo que $\text{sen}\theta = h/r_{arm}$ e $\text{sen}\alpha = h/L$, é possível definir que $\text{sen}\alpha = r_{arm} \text{sen}\theta/L$. É possível considerar o sistema como linear, assumindo que $\text{sen}\theta$ é equivalente a θ , já que a variação é pequena. Substituindo essas variáveis, tem-se a seguinte simplificação:

$$\frac{d^2x}{dt^2} = \frac{m_b g r_{arm} r_b^2 \theta}{L(m_b r_b^2 + J_b)} \quad (2.5)$$

2.4.2 Controle

Para o projeto do controle, é necessário ter a modelagem do sistema e implementar o controlador. Os dados em um sistema a ser controlado são lidados em espectros de frequência, por isso são utilizadas transformadas. A seguir podem ser vistos os tipos de ações e configurações que um controle pode assumir, com foco no que foi essencial para esse trabalho. A Figura 2.8 mostra um sistema de controle em malha aberta.



Figura 2.8: Controle em malha aberta

O controle é dividido em dois tipos: controle em malha aberta e controle em ma-

lha fechada. O de malha aberta realiza é mais simples, onde a saída depende apenas da entrada. Um sistema de controle em malha fechada é aquele que utiliza a saída do controlador como entrada realimentada, geralmente negativa, para poder ser comparada com o *setpoint* desejado.

Ações de Controle

O controlador é geralmente colocado em série com a planta, de forma a produzir uma ação de controle na variável desejada. As principais ações isoladas são: de duas posições, proporcional, derivativa e integral.

A ação de duas posições é utilizada quando o elemento atuante possui apenas duas posições fixas, como 0 e 1. Dessa forma, o controle só irá prover dois valores de saídas, que serão escolhidos ao comparar o sinal de entrada com um limiar. Assim, sistemas binários podem optar por um controlador mais fácil de aplicar, como esse.

Uma ação de controle do tipo proporcional implica em multiplicar o sinal de entrada por uma constante. Dessa forma, a ação será uma amplificação do sinal de entrada. A Equação 2.6 dá a aplicação de um controle proporcional.

$$X(s) = K_p E(s) \quad (2.6)$$

Para implementar uma ação de controle do tipo derivativa, utiliza-se um ganho para amplificar a derivada da variável introduzida. Ela consegue ajudar na estabilidade do sistema, pois diminui o impacto da entrada, mas na prática só é aplicada em conjunto com uma ação proporcional. A Equação 2.7 mostra a modelagem de um controlador proporcional-derivativo.

$$X(s) = (K_p + K_d s) E(s) \quad (2.7)$$

Como uma função de transferência com grau do numerador maior que o do denominador é inviável na prática, pois implicaria em um sistema não causal, um controlador PD deve ser implementado usando a Equação 2.8.

$$X(s) = \frac{(K_p + K_d a)s + K_p a}{s + a} E(s) \quad (2.8)$$

A ação integral corresponde a uma taxa de variação na saída proporcional à entrada. Assim, o sinal de saída crescerá ou decrescerá com uma proporção dada pela variável de ingresso. Com ela, é possível melhorar o erro do estado estacionário do sistema. Geralmente, o controlador com esse tipo de ação apenas adiciona ela à proporcional, como

visto na Equação 2.9.

$$X(s) = \left(K_p + \frac{K_i}{s}\right)E(s) \quad (2.9)$$

Há também a possibilidade de juntar as três ações em um só controlador, conhecido como controle PID. Assim, o sinal pode agir tanto no regime transitório quanto no estacionário.

Controle em Cascata

De acordo com Ribeiro (2005), o controle cascata permite um controlador primário regular um secundário, melhorando a velocidade de resposta e reduzindo os distúrbios causados pela malha secundária. Uma malha desse tipo de controle tem dois controladores com realimentação negativa, onde a saída do controlador primário estabelece o ponto de referência do secundário. A Figura 2.9 ilustra o uso de controle em cascata.

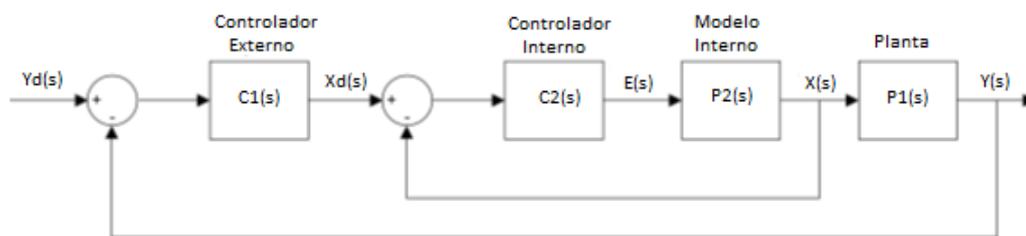


Figura 2.9: Controle em cascata

Esse tipo de controle usa variáveis auxiliares para detectar rapidamente o efeito das perturbações, assim corrigindo mais rápido o erro causado por elas. A saída do controlador externo estabelece o valor que o parâmetro interno deve ter e realiza um controle sobre ele, de forma que a saída do controlador secundário irá ser aplicada no modelo final do sistema.

É afirmado por Smith e Corripio (2008) que, ao se projetar estratégias de controle em cascata, a consideração mais importante é que a malha interna (variável secundária) deve responder mais rapidamente às variações no distúrbio e na variável manipulada do que a malha externa (variável primária). Assim, essa condição deve ser levada em conta no projeto do controlador para esse trabalho.

Controle Proporcional-Velocidade

Um controlador proporcional-velocidade (PV) é um método utilizado em sistemas mecânicos para controle de posição. De acordo com Zanin, Santos e Junior (2007), um controlador PV possui duas condições corretivas, um ganho proporcional à velocidade do objeto e um ganho proporcional ao erro de posição. A Equação 2.10 mostra a representação da ação desse tipo de controle.

$$X(s) = K_p(E_r(s) - E(s)) - sK_v E(s) \quad (2.10)$$

Implementação

Com as fórmulas obtidas nas seções anteriores, é possível encontrar uma relação entre a variação de ângulo do motor e o movimento do objeto, tornando viável a implementação de um controlador. A Figura 2.10 mostra um bloco que representa o sistema e suas variáveis de entrada e saída, sendo a tensão enviada para o motor e a posição do objeto, respectivamente. Neste trabalho, a posição de entrada será a variação de posição que se deseja realizar, o quanto o objeto deve se mover para a posição desejada, e a saída será a variação de ângulo complementar que o motor deve realizar.



Figura 2.10: Diagrama de blocos para o sistema real

Para o cálculo da constante que multiplica o sistema, sabe-se que a variável J_b é dada por $2m_b r_b^2/5$. Assim, com simplificações, a Equação 2.5 pode ser reescrita como:

$$\frac{d^2x}{dt^2} = \frac{5gr_{arm}\theta}{7L} \quad (2.11)$$

O diagrama de blocos completo do controle do sistema pode ser visto na Figura 2.11. Eles foram adaptados do Instructor... (2011) fornecido pela empresa Quanser juntamente com os servomotores. Ele mostra que o sistema também realizará o controle do ângulo relativo aos motores, fazendo assim um controle em cascata. A diferença para este projeto é que não será usado o último bloco para estimar a posição real, mas sim um processamento de imagens em tempo real.

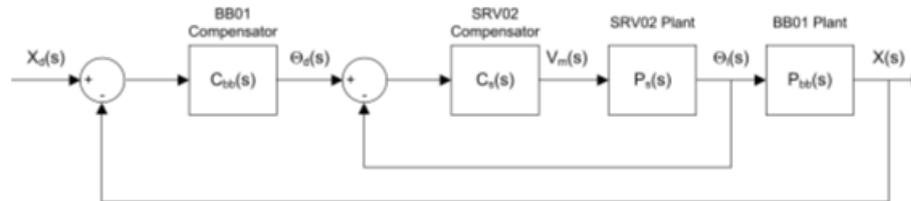


Figura 2.11: Esquema de blocos para controle em cascata

Com os valores do erro de regime, tempo máximo de pico e percentagem de *overshoot*, é possível determinar os parâmetros da equação do sistema padrão mostrada por 2.14 através das definições 2.12 e 2.13.

$$t_p = \frac{\pi}{\omega_n \sqrt{1 - \zeta^2}} \quad (2.12)$$

$$PO = 100 \exp \frac{-\pi \zeta}{\sqrt{1 - \zeta^2}} \quad (2.13)$$

$$H(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (2.14)$$

No esquemático dado pela Quanser, está sendo utilizado um controle PD no loop externo e um PV (proporcional e de velocidade) no loop interno. Para o PV, a função de transferência foi calculada para um $t_p = 0,15s$ e $PO = 5\%$, achando:

$$H(s) = \frac{835}{s^2 + 40 * s + 835} \quad (2.15)$$

O diagrama de blocos do controlador interno está mostrado na Figura 2.12.

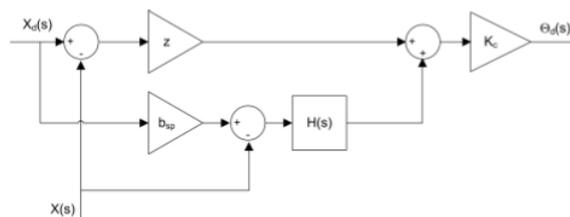


Figura 2.12: Esquema de blocos para controle derivativo real

2.5 Softwares Utilizados

O projeto aqui definido utiliza os *softwares* Matlab® e Simulink® para a implementação computadorizada do controle de motores e processamento de imagem. A corporação MathWorks (1994) é responsável por eles e seu website possui explicações sobre suas funções, definições e também tutoriais. Os cursos disponíveis no site são apresentados em pequenos módulos em forma de tópicos e também possuem assistência *online*. As informações aqui definidas foram adaptadas de seu site.

A empresa MathWorks é líder em desenvolvimento de *softwares* de computação matemática para a área tecnológica. O Matlab® usa linguagem de alto nível e ambiente interativo, envolvendo funções multidisciplinares, como processamento de imagens, comunicação, sistemas de controle e finança computacional. O Simulink® é um ambiente de diagrama de blocos para simulações e *design* baseado em modelos, podendo também utilizar geração automática de códigos e verificação de sistemas embarcados.

2.5.1 Matlab

O Matlab® foi utilizado tanto para a parte de processamento de imagem quanto para a implementação do controle. Para este trabalho, o diretório do sistema de controle foi adaptado diretamente do tutorial e material fornecido pela Quanser®, a empresa responsável pelos motores de passo. Portanto, o código de configuração dos blocos foi apenas setado com os valores das variáveis do sistema.

Enquanto isso, para o processamento de imagens, foi necessária uma pesquisa abrangente sobre as funções disponíveis nessa área. A seguir, encontra-se uma lista resumida das consideradas principais para um desenvolvimento nessa modalidade, com exemplo de aplicação e definição. Elas são utilizadas através dos diretórios *Image Processing Toolbox* e *Image Acquisition Toolbox*, acessíveis através de *toolbox/images/images*.

- Definir Matriz - Variável-matriz = [número-a11 número-a12 número-a13 ... número-a1n; número-a21 número-a22 ... número-a2n; ...número-amn]. Cria uma nova matriz com dimensões mxn;
- Ler imagem - variável = imread('nome-da-imagem.tipo'). Recebe uma imagem da mesma pasta do documento e salva em uma variável de matriz;
- Receber vídeo - var = VideoReader('vídeo.tipo'); nova-variável = read(var). A primeira linha "lê"o vídeo e a segunda salva ele numa variável do programa;
- Aquisição por webcam - var = videoinput('winvideo', numero-do-dispositivo, 'tipo-de-cores'). Inicializa um objeto de vídeo de aquisição contínua através do adaptado

- "winvideo" para *webcam* e armazena em uma variável;
- Gerenciamento de aquisição - *imaqtool*. Abre uma ferramenta capaz de mostrar os dispositivos de captura de vídeo conectados ao *software*;
 - Abrir janela - *figure*. Abre uma nova janela onde podem ser mostradas variáveis do programa;
 - Visualizar imagem - *imshow(variável)*. Faz o comando de mostrar uma variável do tipo imagem;
 - Visualizar vídeo - *imshow(variável)*. Mostra uma variável do tipo vídeo;
 - Salvar em nova imagem - *imwrite(variável, 'nome-do-arquivo.tipo', tipo)*. Cria um novo arquivo de imagem com a variável de matriz definida no código;
 - Obter tamanho da imagem - $[linhas, colunas, planos] = \text{size}(variável)$. Retorna um vetor com os valores para as três características da imagem: número de linhas, de colunas e de planos;
 - Atributos da imagem - *whos variável*. Mostra as características da imagem, como nome, tamanho, número de *bytes*, classe e atributos;
 - Negativo de uma imagem - $nova-var = classe-da-var(-1*(double(var)-255))$. Salva o negativo de uma variável do tipo imagem em uma nova variável;
 - Aumentar brilho - $nova-var = var + constante$. A função de soma em imagens adiciona valores que implicarão em deixar os tons mais claros;
 - Aumentar contraste - $limiar = constante$; $nv-1 = double(var)/limiar$; $nv-2 = nv-1 .* nv-1$; $nv-3 = classe-da-var(nv-2*limiar)$. Essa sequência de comandos consegue dar mais contraste à imagem sem precisar modificar o histograma, normalizando os tons dela e depois exponenciando os mesmos;
 - Binarização - $limiar = constante$; $d = x \geq limiar$. Define um limiar e binariza a imagem, onde o valor 1 (branco) será dado àqueles pontos com valor maior ou menor que o limiar e o resto terá valor 0 (preto);
 - Filtragem - $máscara = matriz-quadrada-ímpar-de-soma-zero$; $nova-variável = conv2(variável, máscara)$. Realiza a convolução de uma matriz de imagem com uma menor responsável por uma máscara específica para o filtro desejado;
 - Obter plano de cor - $nv = variável(:, :, i)$. Armazena um dos planos da imagem colorida, onde *i* é o número do plano, que pode, por exemplo retornar as componentes vermelhas (1), verdes (2) ou azuis (3);
 - Identificação de objetos - $nova-var = regionprops(var, 'dado-desejado')$. Retorna características do objeto encontrado na imagem, esses dados podem ser: 'Centroid' (calcula o centro geométrico), 'Area', 'BoundingBox' (retorna um retângulo ao redor da imagem), 'FilledImage' (retorna a imagem binária sem buracos), entre

outros.

- Conversão de tipos - `nova-var = dither(variável)`, transforma a imagem em preto e branco. `nova-var = im2bw(variável, ponto-de-corte)`, binariza a imagem em 0 e 1 com um limiar de corte. `nova-var = rgb2gray(variável)`, transforma uma imagem de planos vermelhos, verdes e azuis em um plano só de tons de cinza;
- Corte de imagem - `imcrop(var, [xinicial yinicial largura altura])`. Corta a imagem em um novo retângulo com dimensões definidas pelo usuário;
- Obter histograma - `histograma = imhist(variável)`. Armazena o histograma da imagem com 256 cores;
- Equalização de histograma - `histeq(histograma)`. Equaliza o histograma já calculado de uma imagem;
- Mostrar histograma - `imhist(histograma)`. Exibe o histograma calculado previamente de uma imagem.

A Figura 2.13 mostra a *interface* básica do Matlab® versão 2012b. Ela é usada para chamada de funções e execução de códigos que não estão salvos. Através dela, é possível abrir arquivos salvos em `.m` e rodá-los como *scripts* atuais. No lado direito da tela, existe um espaço que mostra as variáveis do sistema e seus detalhes.

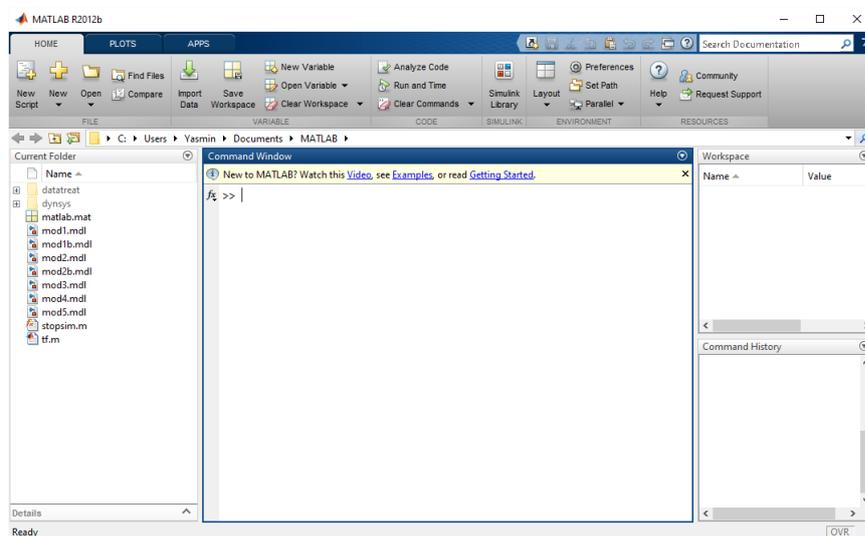


Figura 2.13: Interface do Matlab

2.5.2 Simulink

Simulink é um ambiente de diagramas de blocos que pode ser usado para desenhar, simular, implementar e testar comunicações, controles, processamento de sinais e etc. Ele

também possui editor gráfico de sistemas dinâmicos, biblioteca customizável de blocos e resolução de modelos. O tutorial do site dá uma introdução ao uso do *software* para a modelagem de sistemas dinâmicos.

A biblioteca do Simulink envolve blocos dos tipos mostrados na Figura 2.14, com alto nível de variedade e aplicações, exemplificado na lista do canto esquerdo. Os arquivos disponíveis para entrada nesse *software* são escritos como documentos .mdl.

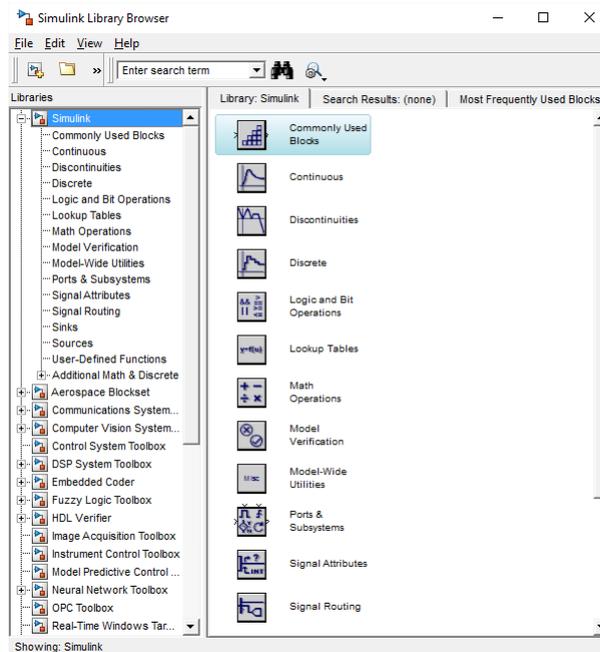


Figura 2.14: Interface de blocos do Simulink

Capítulo 3 - Metodologia

Este capítulo irá tratar da organização dos métodos utilizados para o trabalho, bem como a definição de parâmetros e implementação dos processos vistos no capítulo anterior. Ele consiste na descrição dos passos de fabricação, adaptação do modelo matemático para o sistema em questão, cálculo da implementação de controladores e descrição dos códigos.

3.1 Manufatura

Os processos e passos descritos nessa seção foram todos feitos com base nas informações dadas pela subseção 2.1, incluindo as ferramentas. Para atender limites de peso, atrito e tamanho, as propriedades dos materiais a serem utilizados foram levados em conta ao escolhê-los. Em virtude disso e de uma análise de preços de mercado, decidiu-se que a maior parte da base mecânica seria feita com MDF de tamanhos variados e que os instrumentos que necessitassem de mais robustez seriam feitos com materiais metálicos.

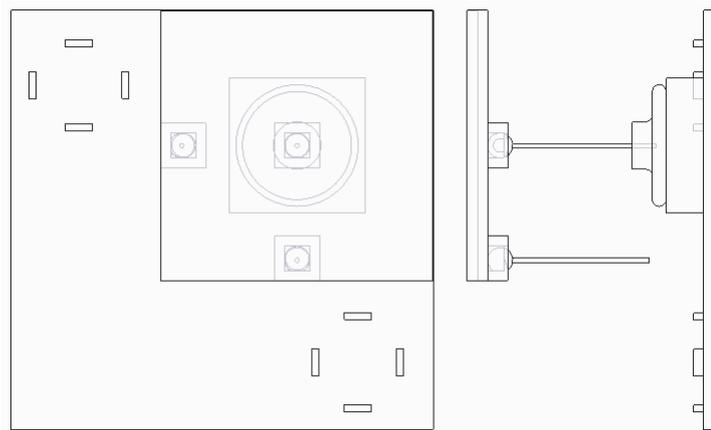


Figura 3.1: Modelo do projeto mecânico

Na Figura 3.1 é possível verificar o modelo utilizado como base para a fabricação das peças do sistema. O processo, as peças e os materiais serão detalhados nas seções a seguir.

Peça	Dimensões	Quantidade
Esfera	37mm de diâmetro	3
Caixa de MDF	57x57x38,82mm	3
Haste da Junta	6,5mm de diâmetro 20cm de comprimento	3
Haste do Motor	12,5x2x0,5cm	2
Mesa	40x40x1,5cm	1
Base Metálica da Mesa	20cm de diâmetro 5cm de altura	1
Apoio MDF da Mesa	20x20x5,5cm	1
Base Total	62x62x1,5cm	1
Encaixe do Motor	1x4x1,5cm	8
Servomotor	12x12x18cm	2
Base MDF da Câmera	10x10x1,5cm	1
Haste em L da Câmera	1,20x0,4m	1
Câmera	-	1

Tabela 3.1: Peças Utilizadas

3.1.1 Material e Peças

Para ajudar na visualização e no entendimento dos procedimentos de fabricação, encontra-se na Tabela 3.1 a lista de peças e elementos utilizados. Assim, é possível separar os processos de manufatura pelos produtos que serão utilizados. Os componentes comprados foram os servomotores e a câmera, incluindo acessórios, e os outros itens foram manufaturados especificamente para o projeto, em diferentes laboratórios da Universidade Federal do Rio Grande do Norte.

Tendo em vista o número de peças e a utilização de cada uma, foi possível escolher os materiais mais específicos e os tamanhos estimados para eles. Além disso, com o desenrolar do processo, peças adicionais foram compradas, como porcas, aruelas, parafusos e até brocas. Para dar um valor do material a um projeto como este, calculou-se o preço de cada elemento.

Na Tabela 3.2 é possível ver uma estimativa do orçamento total para a produção desse trabalho, sem levar em conta os motores. Os servomotores foram desconsiderados do cálculo por não precisarem ter essa especificação exata, já que eles foram comprados da Quanser e possuem qualidade, precisão e complexidade muito além do necessário.

Material	Quantidade	Preço Estimado
MDF e mão de obra	-	150,00
Parafusos 26cm	3	14,00
Barra chata 3/4x1/5" de inox	2	20,00
Flanela branca e cola	1	11,00
Parafusos pequenos, porcas e arruelas	60+	5,00
Esferas de resina	3	9,00
Bola de borracha	1	3,00
Base branca de aço (sucata)	1	20,00
Cantoneiras de Alumínio	3	8,00
Câmera USB	1	60,00
Total		300,00

Tabela 3.2: Orçamento

3.1.2 Processo de Fabricação

A mesa que serve de plano de referência matemática foi comprada já com as dimensões desejadas, de madeira, e em sua superfície foi colado um tecido do tipo flanela de mesmas dimensões. A adição do tecido serve para ajudar a aumentar o atrito e dar mais tempo ao processamento do sistema para efetuar o movimento desejado sem sofrer mudanças bruscas da posição do objeto.



Figura 3.2: Junta homocinética com hastes

Para utilizar juntas homocinéticas, foram feitos furos de 6mm nas esferas de resina com um torno mecânico. Depois, as hastes cilíndricas foram encaixadas nelas, de forma que a profundidade seja no mínimo a metade do diâmetro das esferas. Em seguida, as hastes foram serradas para terem exatamente 20cm de distância do encaixe.

Um furo de 3mm foi feito sobre essas hastes com centro a 1cm de distância do fim delas, para poder formar um encaixe em L com o motor. Esse conjunto pode ser visto na Figura 3.2.



Figura 3.3: Motor SRV02 da Quanser

O encaixe do sistema no motor foi feito com precaução, pois o servomotor da Quanser já vem com suportes prontos. Dessa forma, foi necessário apenas fazer os furos exatos e de forma alinhada. Na segunda haste, furada com os 3mm de encaixe com o apoio da junta, tiveram dois furos de 4,5mm e um de 6mm no centro deles, a 12,5cm de distância do encaixe em L. O motor SRV02 da Quanser está mostrado na figura 3.3. É possível ver os encaixes da haste que vem com ele.



Figura 3.4: Encaixe das juntas homocinéticas na mesa

Esse conjunto representará o eixo de movimento da junta homocinética. O encaixe dele foi feito com MDF de diversos tamanhos, de maneira a formar uma caixa que foi parafusada na mesa. Para a junta estar completa, furos de 3cm foram feitos nas tampas e as esferas com haste foram encaixadas antes do suporte ser parafusado.

Na parte inferior da mesa foram encaixados três conjuntos desse tipo, um no centro e os outros no meio de laterais perpendiculares entre si. Isso pode ser visto na Figura 3.4, onde a haste central será responsável por absorver o maior impacto do peso e as laterais terão mais liberdade, sendo encarregadas do movimento em cada eixo.

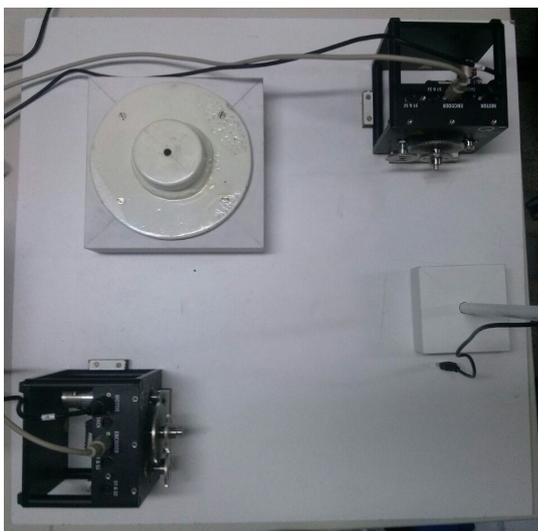


Figura 3.5: Base final com motores

Para ajudar na sustentação da mesa, foi utilizada uma base de mesmo material e dimensões maiores, onde pôde ser colocado encaixe para os motores e uma base mais pesada para sustentar o eixo central do plano. Também foi adicionado, temporariamente, uma haste em L para suportar a câmera, de forma que ela fique virada para o centro da mesa. O conjunto formado pela base e seus acessórios é exemplificado na Figura 3.5.

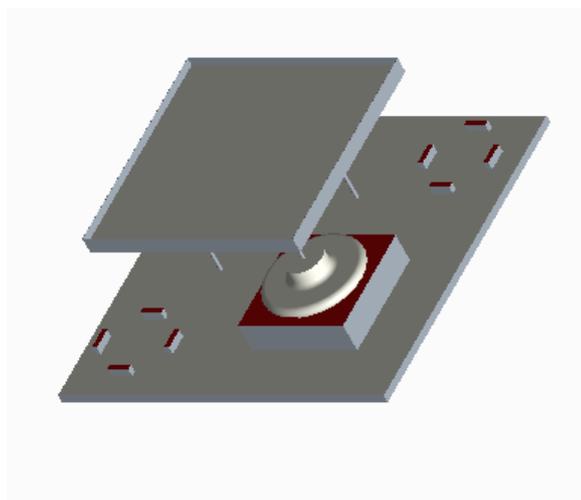


Figura 3.6: Modelo do projeto em vista isométrica

A Figura 3.6 mostra o modelo inicial do projeto em vista isométrica para comparação com o resultado. O sistema real completo pode ser visto na Figura 3.7. Como objeto de referência para o processo, foi escolhida uma bola de borracha, de forma a realizar o controle de posição sobre ela.

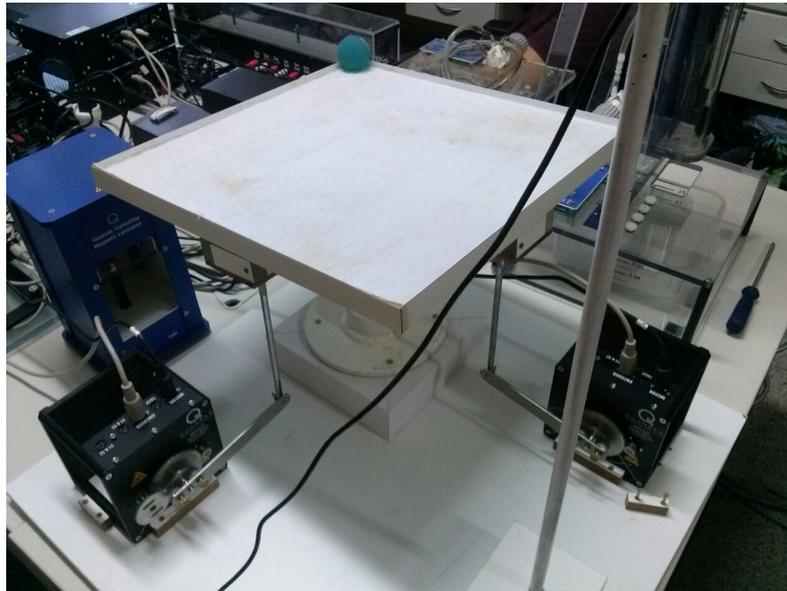


Figura 3.7: Sistema mecânico final

3.2 Controle do Sistema

Para esta etapa, foi calculado e adaptado a modelagem do sistema, com base nas equações vistas no capítulo passado. A partir dos diagramas e definições das funções de transferências para o controle total em cascata, foi estimado um controlador proporcional-derivativo específico.

Como visto no capítulo anterior, um dos eixos do sistema pode ser descrito pela Equação 2.10. Para este projeto, os seguintes parâmetros foram utilizados:

$$L = 40cm$$

$$g = 9,89m/s^2$$

$$r_{arm} = 12cm$$

Substituindo as constantes na equação da função de transferência 3.3 para o sistema em malha aberta, obtém-se a relação mostrada na equação abaixo.

$$H(s) = \frac{2,22}{s^2} \quad (3.1)$$

Para melhorar o erro de regime transitório do sistema, decidiu-se implementar um controlador PD. Para o cálculo dos ganhos e das variáveis de controle, foi estabelecido que o tempo de estabilização deve ser de 1s e o overshoot deve ter 0,05 do valor de referência. Assim, encontra-se $K_p = 0,207$ e $K_d = 0,0768$ e a função de transferência correspondente dada pela equação 3.2, definindo $a = 10$.

$$H(s) = \frac{0,975s + 2,07}{s + 10} \quad (3.2)$$

A função de transferência de um sistema é no plano de frequência como a fração da saída pela entrada. Utilizando Transformada de Laplace na Equação 2.5, a função de transferência dessa planta pode ser encontrada por:

$$H(s) = \frac{1}{s^2} \frac{m_b g r_{arm} r_b^2 \theta}{L(m_b r_b^2 + J_b)} \quad (3.3)$$

A modelagem do sistema de motores foi projetada pela Quanser em série com a planta, como mostra a Figura 3.8. Assim, a função de transferência total desse sistema é dada por 3.4, com $\tau = 0,0248$ e $k = 1,53$.

$$H(s) = \frac{k}{s^3(\tau s + 1)} \frac{m_b g r_{arm} r_b^2 \theta}{L(m_b r_b^2 + J_b)} \quad (3.4)$$

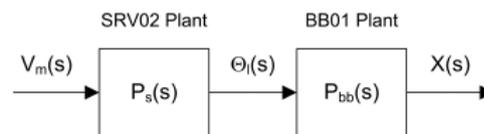


Figura 3.8: Diagrama de blocos para motores e planta

A Figura 3.9 mostra o diagrama de blocos do sistema completo no Simulink. Com essa configuração, o sistema de controle está pronto para ser testado. Para a sua implementação, a saída do controle de ângulo deve ser a alimentação dos motores e a realimentação negativa desse controlador deve receber os valores reais da posição. Essa conexão com os motores pode ser feita através de blocos específicos para materiais da Quanser. Além disso, o controlador da malha externa deve entrar como referência na malha interna, como já mostrado, e sua entrada realimentada deve vir diretamente do processo de captura de posição.

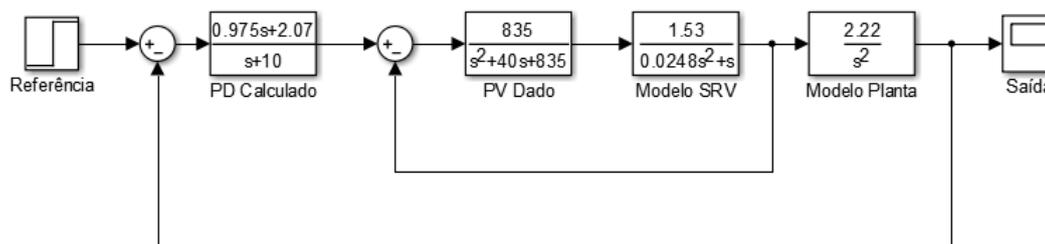


Figura 3.9: Diagrama de blocos completo

3.3 Processamento de Imagem

Os processos e funções citados nessa seção foram definidos previamente neste documento. O bloco de cálculo de posição foi feito no Matlab e se encontra em anexo. Ele é responsável pela captura de vídeo contínua, reconhecimento do objeto e retorno de sua posição.

Para realizar a implementação, o ponto inicial foi definir uma variável responsável pela captura de vídeo. Para cada *frame* de vídeo, deve ser calculada a posição do objeto. Então, em um *loop* infinito, um *snapshot* do vídeo é feito e realizado o processo descrito a seguir.

Em cada passo, são inicializadas as constantes de posição para cada eixo e um contador em valores nulos. Cada imagem capturada deve ser transformada para escalas de cinza. Em seguida, define-se um limiar e é realizado o processo de binarização. Como a imagem é retangular, é necessário utilizar um processo de corte, para que ela se ajuste às dimensões do plano e não sofra interferência de peças auxiliares.

Com a imagem em preto e branco, calcula-se o centro do objeto ao percorrer a matriz em busca de pixels de valor 1 (branco) que foram definidos como parte do objeto no momento de limiarização. Para cada *pixel* certo, soma-se sua localização às constantes de posição e adiciona-se um no contador. Dessa forma, ao final do processo, é possível encontrar o centro geométrico ao dividir as posições achadas para cada eixo pelo número de pontos encontrados.

Por fim, deve-se apenas converter as unidades. O processamento inteiro foi feito com *pixels* e dependente da resolução da imagem. Ela foi implementada no final do código e ainda deve haver uma conversão de centímetros (saída da equação) para metros, já que os blocos de controle do sistema trabalham com esta unidade.

Capítulo 4 - Resultados e Discussões

4.1 Calibração do PDI

Alguns testes de calibração foram necessários, para configurar o corte da imagem a e a conversão de pixels para centímetros. Para uma imagem 320x240 (eixo x por eixo y), o corte foi feito em 43x5, com largura e altura de 225 *pixels*. O resultado desse processo foi enfim impresso na tela, como mostra a Figura 4.1.

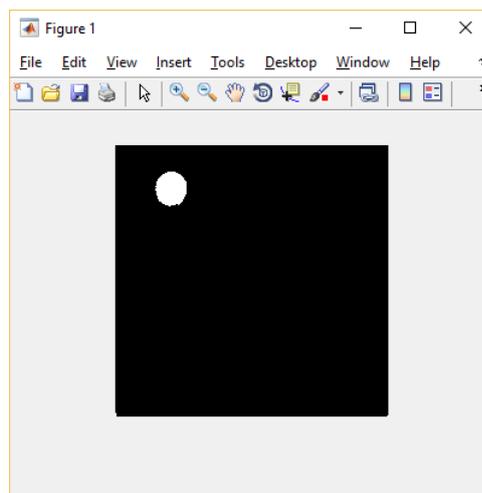


Figura 4.1: Imagem processada

Com as constantes de corte, é possível encontrar a relação dada pela equação:

$$P_r = \frac{P_p}{5,625} \quad (4.1)$$

Porém, com alguns testes adicionais, foram feitas adaptações. Por fim, foi possível notar que a Equação 4.2 mostrou resultados mais precisos para o que era visto no sistema real. Ainda assim, é necessário verificar com maior precisão essa medição.

$$P_r = \frac{P_p}{5} - 2 \quad (4.2)$$

4.2 Simulação do Controlador

Para o teste de simulação do sistema de controle, foi usado o diagrama mostrado na Figura 3.9 e implementado no Simulink. O controlador PD foi projetado sem considerar o ganho da malha dos motores, por conta do alto ganho do controlador PV, e os resultados mostram que esses valores não foram suficientes pois o sinal diverge.

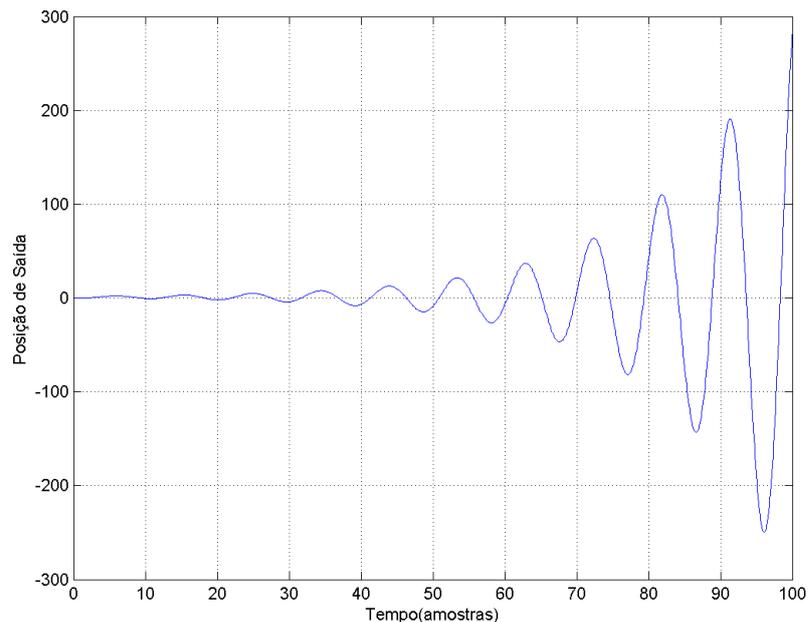


Figura 4.2: Teste do sistema de controle

A Figura 4.2 mostra o comportamento do sistema simulado. Apesar de estar melhor do que sem o controlador, o sinal diverge, não conseguindo fazer o proposto. Isso implica que o ajuste de parâmetros não foi adequado para o sistema.

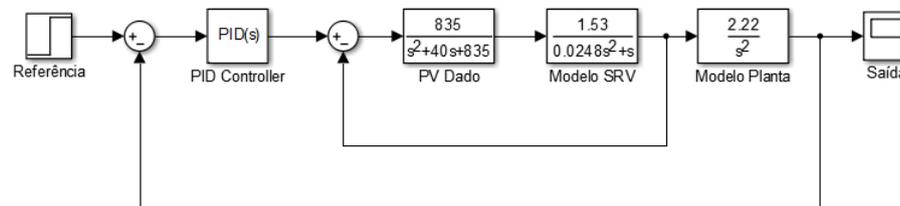


Figura 4.3: Novo diagrama de blocos para o controle

Para resolver esse problema, o bloco de controle PD calculado foi substituído por um bloco PID no Simulink. A partir daí, foi possível adequar os ganhos de controle através do artifício "tune" presente no bloco PID, que recalcula as constantes de acordo com o sistema e o tempo de estabilização escolhido pelo usuário. Esses ganhos foram aproximadamente $K_p = 0,068$, $K_d = 1,326$ e $K_i = 0,00077$ (ou seja, desprezível) para um tempo de estabilização de 1s. A Figura 4.3 mostra o diagrama com os novos blocos.

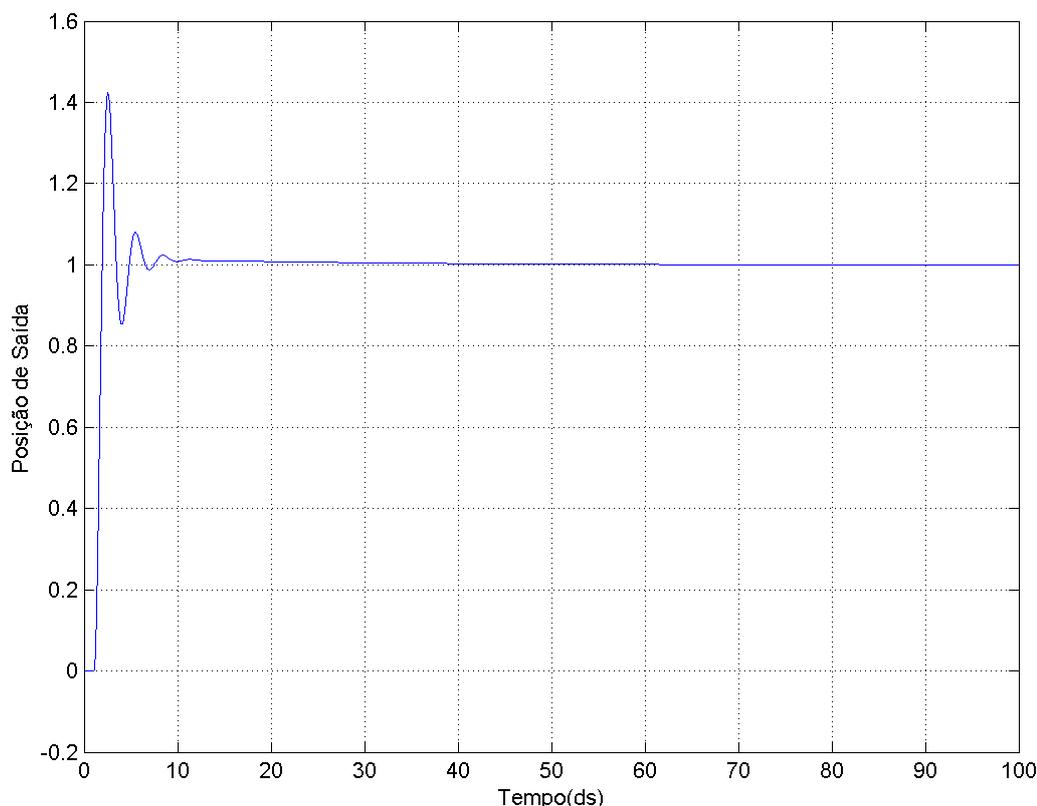


Figura 4.4: Teste 1 do controle PID

As Figuras 4.4 e 4.5 mostram os resultados das simulações para uma referência variando 1 e 20 posições, respectivamente.

É possível notar que os resultados foram satisfatórios, pois, apesar do *overshoot* de aproximadamente 40% para grandes variações de posição, o sistema consegue chegar ao *setpoint* desejado dentro do tempo estabelecido. Se o tempo de estabilização fosse maior, o sistema teria menos mudanças bruscas. Fica a critério do projetista escolher o que se encaixa melhor em seu trabalho.

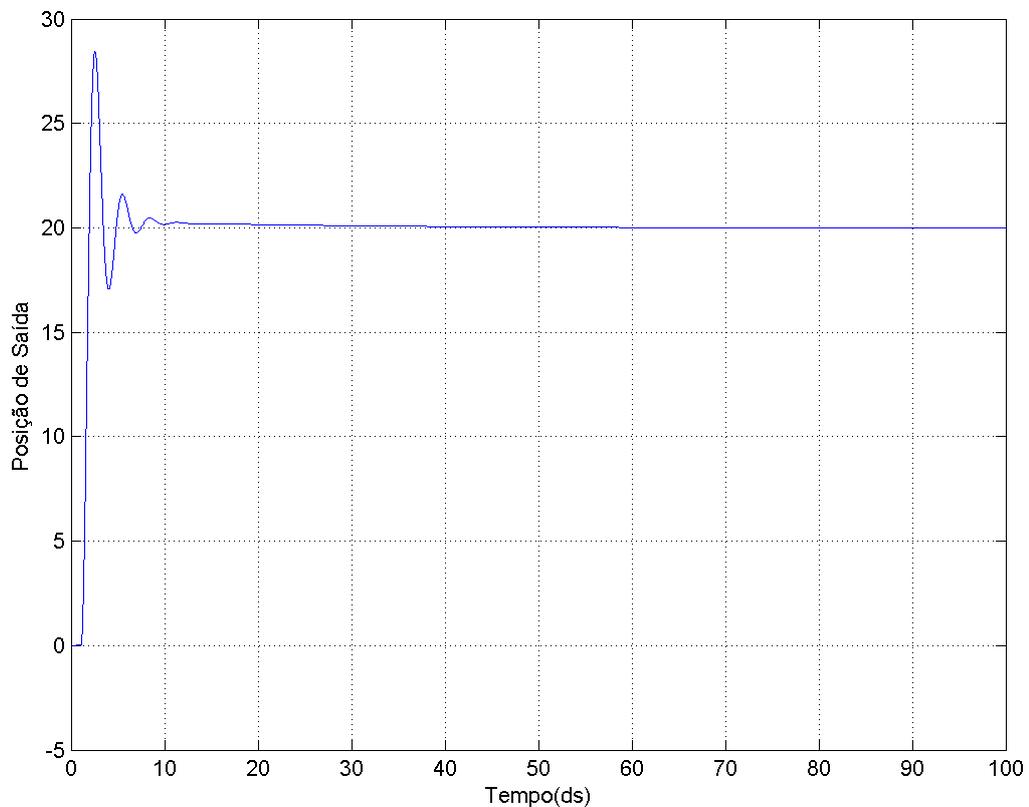


Figura 4.5: Teste 2 do controle PID

4.3 Teste de Movimento dos Motores

Para os testes de movimentação dos motores, usou-se o Simulink, de forma que os blocos do programa podem ser vistos na Figura 4.6. Foram dados sinais de referências aos motores, de forma que eles variassem aproximadamente 25 graus. Para isso, um gerador de pulso foi colocado junto com uma constante, garantindo que os motores variem para ângulos negativos.

Um atraso de 25% do período total foi usado no sinal de referência do servomotor A. Isso permitiu que os motores fossem testados em quatro configurações, combinando ângulos positivos e negativos para cada um deles. Um bloco de saturação antes da escrita de sinal nos motores garantiu que nenhuma tensão além do permitido pelas especificações fosse colocada.

O programa compara o *setpoint* com a leitura de ângulo dos motores e envia a diferença com ganho para eles, de forma a garantir a direção correta de movimento para cada um. Para a parte da direção, primeiramente foram realizados os testes sem conectar com

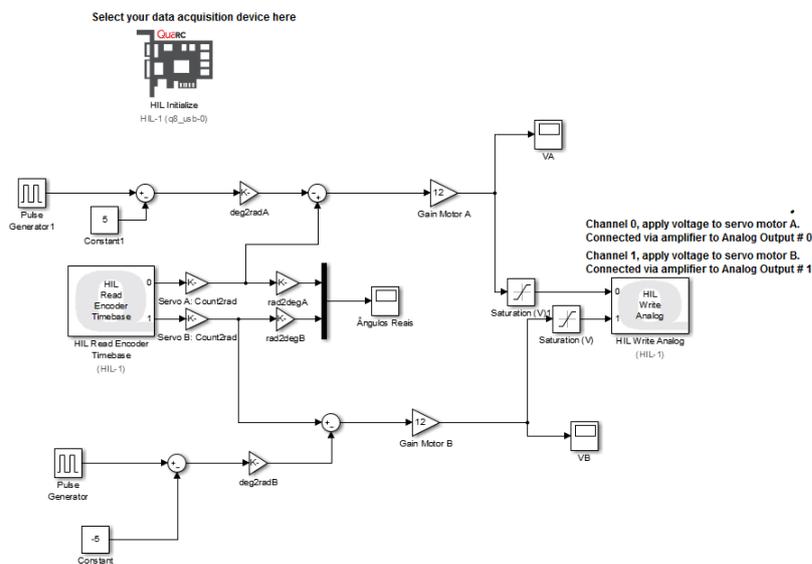


Figura 4.6: Programa do Simulink para teste do sistema mecânico

a mesa, pois o servomotor denominado como B realiza movimentos na direção oposta ao motor A para esse sistema, por conta das suas localizações ao plano. Assim, os *setpoints* e leituras de ângulo para B foram de sinal negativo.

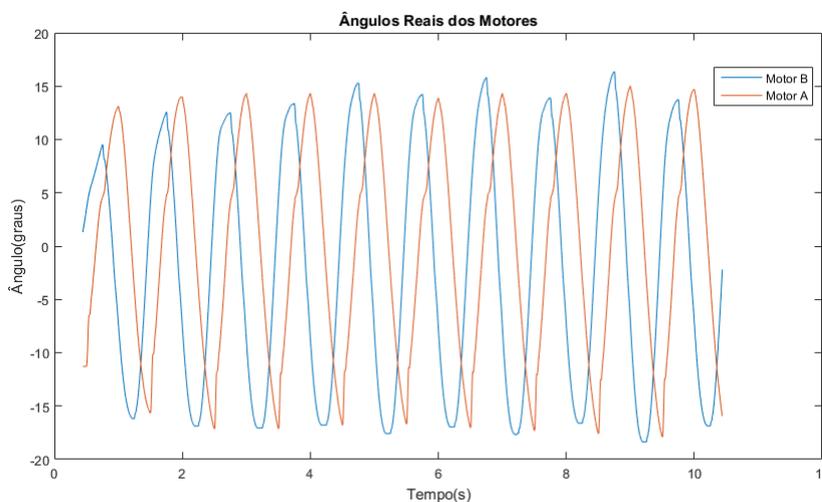


Figura 4.7: Variação de ângulo dos motores durante teste

As Figuras 4.7, 4.8 e 4.9 mostram os resultados dos gráficos dos sinais de movimento para os ângulos dos motores, tensão enviada ao motor A e tensão enviada ao motor B, respectivamente. É possível notar que os motores conseguiram fazer o movimento desejado, com poucas alterações.

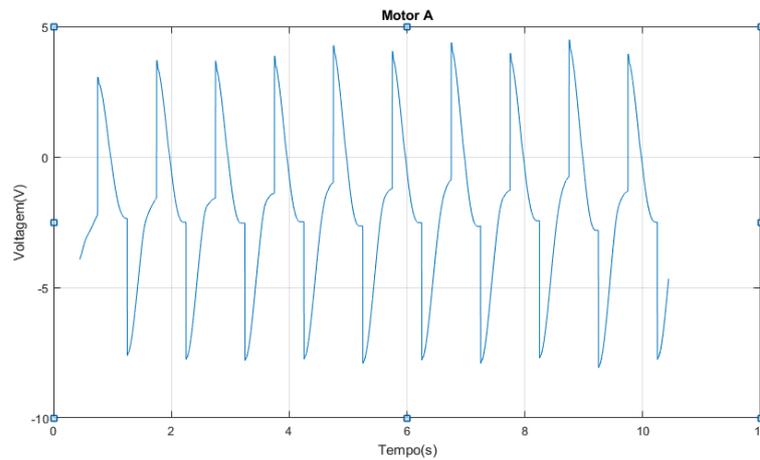


Figura 4.8: Variação de tensão enviada para o motor A

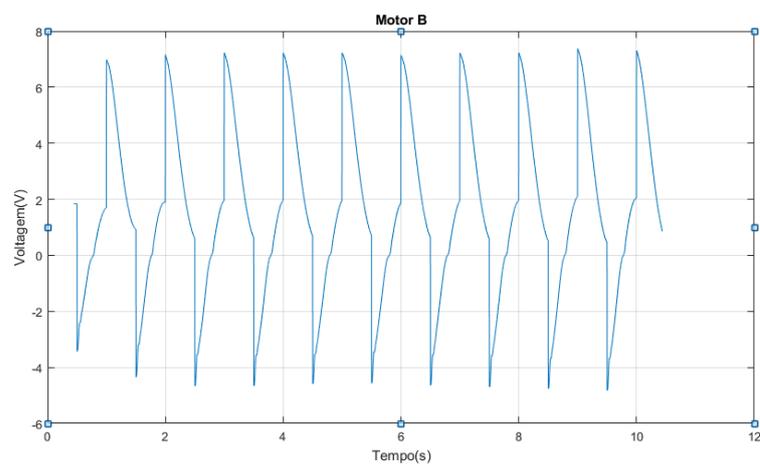


Figura 4.9: Variação de tensão enviada para o motor B

Com a análise do gráfico, é possível notar que os motores começaram variando menos do que o devido e depois de um tempo estabilizaram o movimento. Isso pode ser atribuído à diferença do atrito estático para o dinâmico, pois em repouso os motores sofreriam mais resistência. Com a implementação do controlador PV nos motores, espera-se que essa dificuldade seja ultrapassada. Ainda assim, os motores realizaram o movimento dentro dos limites de amplitude, garantindo que os resultados fossem satisfatórios.

4.4 Teste de Identificação de Objeto

Para o teste de captura de posição através de *webcam*, foi utilizado o código descrito na seção anterior, com a modificação que permitiu o armazenamento das posições medidas em vetores. Assim, ao final do teste, foi possível plotar o movimento em cada eixo em relação ao tempo e o desenho do movimento relativo entre eles.

Como o teste foi realizado sem a conexão com os motores, o movimento da mesa foi simulado manualmente, movendo-a de maneira a fazer o objeto ir para os quatro cantos do plano e por fim o centro. O resultado do movimento em cada eixo pode ser visto nas Figuras 4.10 e 4.11 , para x e para y, respectivamente.

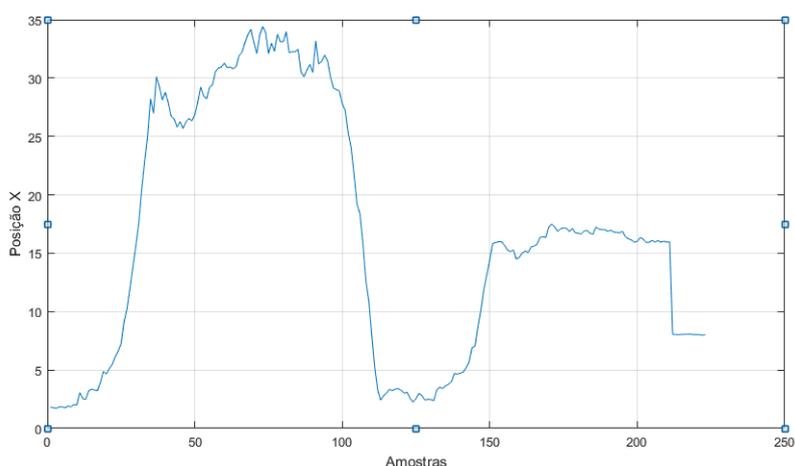


Figura 4.10: Movimento do objeto no eixo X

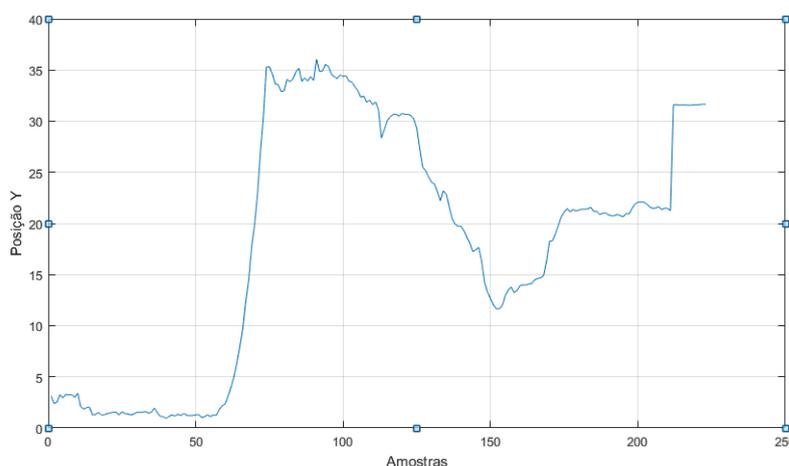


Figura 4.11: Movimento do objeto no eixo Y

Além disso, a posição foi rastreada ao plotar o x pelo y, de forma que é possível verificar o movimento real feito, confirmando o sucesso do teste. Este resultado se encontra na Figura 4.12.

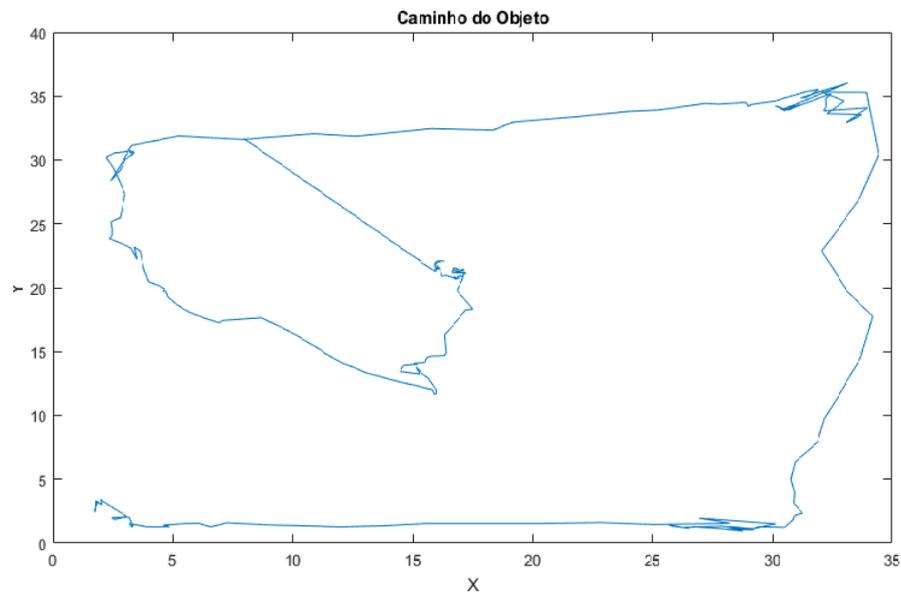


Figura 4.12: Movimento relativo do objeto

A figura anterior conseguiu representar a trajetória do objeto observada na planta real. Assim, o teste foi considerado satisfatório, podendo assumir que o código está pronto para utilização.

Capítulo 5 - Conclusão

Os testes e simulações realizados durante este trabalho tiveram resultados adequados, uma vez que foram calibradas as configurações do projeto. Os motores conseguiram movimentar o sistema sem esforço depois que o atrito estático foi vencido e devem ter um desempenho ainda melhor caso o controlador PV seja colocado. O processamento de imagens conseguiu obter a posição do objeto rapidamente com precisão aceitável. O projeto de controlador simulado apresentou uma resposta satisfatória ao sistema para as condições testadas, podendo ser melhorado com mais testes e ajustes.

Para um estudo mais aprofundado do processamento de imagens, deve-se verificar o atraso entre a captura de delas e o retorno de posição, como também realizar testes de medições de forma mais precisa para confirmar a veracidade dos cálculos. A modelagem do controle foi testada de forma independente em apenas um eixo e para testar os dois deve-se verificar a influência que o movimento em um eixo possui no outro.

Infelizmente, não foi possível executar o projeto como um todo, devido à incompatibilidade entre as configurações necessárias para a leitura/escrita dos motores e a entrada de vídeo da câmera utilizada. Caso fosse utilizada uma câmera da Quanser, seria possível substituir o bloco de leitura de vídeo por um específico para essa marca, o que pode resolver o problema. Dessa maneira, a captura feita entraria no bloco de processamento de imagens já utilizado.

Para trabalhos futuros, pode ser feita a junção dos blocos de controle utilizados com o processamento de imagem e a leitura/escrita dos servomotores. A fabricação poderia ser melhorada com a manufatura de juntas homocinéticas de melhor precisão e a utilização de ferramentas de maior qualidade e eficiência. Recomenda-se que seja feita uma haste em U invertido externa ao sistema para dar maior sustentação para a câmera. Com a união dos elementos do sistema, será possível aplicar e analisar o funcionamento de variados tipos de controladores e configurações.

Referências Bibliográficas

- ACHARY, T.; RAY, A. K. *Image Processing- Principles and Applications*. [S.l.]: Wiley Sons, Inc., 2005. page.1212
- ARAÚJO, F. M. U. de. *Sistemas de controle*. Universidade Federal do Rio Grande do Norte. 2013.
- BASSANEZI, R. C. *Ensino-Aprendizagem com Modelagem Matemática*. [S.l.]: Editora Contexto, 2002. page.1212
- BESTCARS. Semiárvores: trocar ambas ou apenas de um lado. Disponível em <http://bestcars.uol.com.br/bc/mais/cons-tecnico/semiarvores-trocar-ambas-ou- apenas-de-um-lado/>. 2015. page.88
- BOUMAN, K. *Image processing technology and applications: Object recognition and detection in natural images*. MIT, 2012. page.1212
- DEVI, H. K. A. Thresholding: A pixel-level image processing methodology preprocessing technique for an ocr system for the brahmi script. <http://doi.org/10.5334/aa.06113>, 2006. page.1111
- DORF, R. C.; BISHOP, R. H. *Sistemas de Controle Moderno*. [S.l.]: LTC Editora, 2001. Tradução feita por Bernado Severo da Silva Filho. page.1313
- EBAH. *Processos de fabricação - furação*. Disponível em <http://www.ebah.com.br/content/ABAAAABiIAI/furacao>. 2012.
- FERREIRA, A. Metodologia para implantação de laboratórios remotos via internet na Área de automação da manufatura. *Congresso Brasileiro de Engenharia de Fabricação*, 2003.
- GONZALES, R. C.; WOODS, R. E. *Digital Image Processing*. [S.l.]: Pearson Education, 1992. page.88, page.99
- INSTRUCTOR Workbook - Ball and Beam Experiment for Matlab/Simulink Users. [S.l.], 2011. page.1313, page.1717
- JAIN, R.; KASTURI, R.; SCHUNCK, B. G. *Machine Vision*. [S.l.]: McGraw-Hill, Inc., 1995. page.1111
- LOSEKANN, D. E. C. R. *Processos de Fabricação I: Unidade 1*. [S.l.: s.n.], 2001. page.33

- MARQUES, O.; VIEIRA, H. *Processamento Digital de Imagens*. [S.l.]: Rio de Janeiro: Brasport, 1999. page.88, page.1010, page.1111
- MATHWORKS, I. T. Mathworks website. Disponível em <http://www.mathworks.com/>. 1994. page.1919
- MORTON, J. A. *Integration of Systems Engineering with Component Development*. [S.l.: s.n.], 1959. page.11
- OGATA, K. *Modern Control Engineering*. [S.l.]: Prentice/Hall, Inc., 1970.
- RIBEIRO, M. A. *Controle de Processo - 8a edição*. [S.l.]: Tek Treinamento Consultoria, 2005. page.1616
- RUFLAV. Furadeira de bancada schulz fsb16p-220v - 1/2cv - mandril: 16mm (5/8"). Disponível em http://www.ruflav.com.br/produto_furadeira-de-bancada-schulz-fsb16p-220v---1-2cv---mandril:-16mm-5-8_354.
- RUVILLE. *Manual Técnico de Juntas Homocinéticas*. [S.l.], 2015. Disponível em https://www.ruville.com/fileadmin/user_upload/redaktion/pdfs/Technikbroschueren/ESP/CV_Joints_SP_d.pdf. page.77
- SMITH, C. A.; CORRIPIO, A. B. *Princípios e Prática do Controle Automático do Processo*. [S.l.]: LTC, 2008. page.1616
- SOUSA, C. A. B. de. *Serralharia - UP 0043 - Corte por Serragem*. [S.l.]: CENFIM - Centro de Formação Profissional da Indústria Metalúrgica e Metalomecânica, 2008. page.66
- SOUZA, D. A. J. de. *Processos de Fabricação por Usinagem: Parte I*. [S.l.]: UFRGS, 2011. page.66
- WHATIS. *What Is Stepper Mottor*. 2005. page.22
- ZANIN, I. A. A.; SANTOS, R. B.; JUNIOR, V. L. Controle-pv de posição de um carro de movimento linear. *XIV Congresso Nacional de Estudantes de Engenharia Mecânica*, 2007. Universidade Federal de Uberlândia. page.1717

Apêndice A - Código para Identificação de Posição do Objeto

```
vid = videoinput('winvideo', 1, 'YUY2_320x240');
set(vid, 'ReturnedColorSpace', 'RGB');
set(vid, 'FramesPerTrigger', Inf);
set(vid, 'ReturnedColorspace', 'rgb');
5 vid.FrameGrabInterval = 1;
start(vid)
for k=1:50000
    centroids(1) = 0;
    centroids(2) = 0;
10 cont = 0;
    I=getsnapshot(vid);
    In = rgb2gray(I);
    thresh = max(max(In))*0.6;
    In = In<thresh;
15 In2 = imcrop(In, [43 4 225 225]);
    for i=1:225
        for j=1:225
            if In2(i,j)==1
                cont=cont+1;
20 centroids(1) = centroids(1) + i;
                centroids(2) = centroids(2) + j;
            end
        end
    end
    ponto_x = centroids(1)/cont;
25 ponto_y = centroids(2)/cont;
    x(k) = ponto_x/5 - 2
    y(k) = ponto_y/5 - 2
    figure(1)
30 imshow(In2);
end
```
