

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL  
ESCOLA POLITÉCNICA

**SISTEMA DE CONTROLE REGULATÓRIO PARA RASTREAMENTO  
DE OBJETOS UTILIZANDO CÂMERA PAN-TILT**

Porto Alegre, 24 de junho de 2020.

**Autor: Lucas Mattos da Silva**

Pontifícia Universidade Católica do Rio Grande do Sul

Curso de Engenharia de Controle e Automação

Av. Ipiranga, 6681 - Prédio 30 - CEP: 90619-900 - Porto Alegre - RS - Brasil

Email: lucas.mattos@edu.pucrs.br

**Orientador: Prof. Rafael da Silveira Castro**

Pontifícia Universidade Católica do Rio Grande do Sul

Av. Ipiranga, 6681 - Prédio 30 - Bloco A - Sala 226 - CEP: 90619-900 - Porto Alegre - RS - Brasil

Email: rafael.castro@pucrs.br

**RESUMO**

Este trabalho apresenta a aplicação de um controle regulatório em uma câmera pan-tilt para que o objeto rastreado seja posicionado no centro da imagem. Cada eixo da plataforma pan-tilt é tratado como um sistema independente, onde o sinal de entrada do sistema é dado pelo deslocamento do objeto rastreado com relação ao centro da imagem da câmera e o atuador do sistema é o motor que gira este eixo. A identificação do sistema é feita através do método dos Mínimos Quadrados e o projeto do controlador em malha fechada é feito através da alocação de polos pelo Lugar Geométrico das Raízes.

**Palavras-chave:** Câmera Pan-Tilt; LGR; Mínimos Quadrados; Rastreamento de objetos; Controle digital.

**1 INTRODUÇÃO**

A visão computacional é uma potente ferramenta para sensoriamento, fornecendo diversas informações sobre o meio físico sem a necessidade de proximidade ou contato do sensor. Aplicando visão computacional a um sistema robótico temos o que é chamado de controle servo visual (KIKUCHI, 2007) e este pode ter diversas aplicações para pesquisas de controle. Este trabalho utiliza uma câmera em uma plataforma pan-tilt, que permite a rotação da câmera em dois graus de liberdade. Algoritmos de filtragem de cores e detecção de bordas permitem determinar a posição de um objeto na imagem capturada pela câmera. Utilizando a posição do objeto na imagem e o comando dos motores que movimentam a plataforma, cria-se a possibilidade de um sistema de rastreamento regulatório.

## **1.1 Motivação**

O processamento de imagens é uma ferramenta poderosa e está cada vez mais acessível, com bibliotecas e softwares que buscam simplificar o desenvolvimento de aplicações que fazem uso desta tecnologia. Assim podemos substituir sensores como os inerciais e os de ultrassom por uma câmera comum na hora de desenvolver um projeto de controle.

Com o processamento de imagens e uma câmera pan-tilt temos a oportunidade de pôr em prática diversos conhecimentos adquiridos no curso de engenharia de controle e automação. Assim desenvolvendo um projeto que mostra de maneira didática, simples e interessante a aplicação prática das ferramentas apresentadas ao longo do curso.

## **1.2 Objetivo**

Os objetivos deste trabalho são:

- a) Desenvolver e construir o protótipo de uma câmera pan-tilt com ferramentas de processamento de imagem e controle dos servo-motores;
- b) Identificar os sistemas através de ensaios;
- c) Projetar sistemas de controle que respondem de maneira satisfatória a distúrbios de posição em regime regulatório;

## **1.3 Delimitações do Trabalho**

Este trabalho concentrou-se nas partes relacionadas a área de controle de sistemas digitais lineares. Assim não está no escopo deste trabalho ferramentas avançadas de detecção no processamento de imagens, as não-linearidades do sistema e os aspectos econômicos do projeto.

# **2 REVISÃO BIBLIOGRÁFICA**

## **2.1 Trabalhos Prévios baseados em servo-visão para plataformas pan-tilt**

Existem diversos trabalhos relacionados a servo-visão em plataformas pan-tilt na literatura pois este sistema é uma ferramenta poderosa na robótica atual. Em suas diversas aplicações como no auxílio na movimentação de robôs humanoides (FUSCO et al., 2014) e para efeitos e técnicas de fotografia (MENDONÇA, 2014), encontramos diferentes metodologias referentes a estratégia para detecção do objeto, estrutura da plataforma pan-tilt, obtenção de um modelo estimado para o sistema e estratégia para projeto do controlador.

### **2.1.1 Detecção do Objeto**

Trabalhando com o processamento de imagens coloridas temos a cor como um fator muito influente, que facilita a extração de um objeto em uma cena (Gonzalez and Woods, 2010, p. 16). Na literatura encontramos diversas ferramentas que permitem a realização de tarefas relacionadas a detecção de objetos em uma imagem.

Uma maneira de realizar a detecção é através de softwares com este tipo de processamento para aplicações específicas, como o digiCamControl para fotografia (MENDONÇA, 2014). Estes normalmente necessitam de alguma câmera específica ou são softwares pagos.

Existem também softwares com pacotes de ferramentas para processamento de imagem em tempo real, como o MATLAB utilizado por Castro (2013). Este tipo de software costuma possuir diversas ferramentas matemáticas para trabalhar com os dados obtidos da imagem.

Uma alternativa que está ganhando cada vez mais atenção é a utilização de bibliotecas de visão computacional de código aberto. Destas bibliotecas destaca-se a OpenCV que é uma biblioteca multiplataforma livre para uso acadêmico e comercial, utilizada por Fusco et al. (2014), Ridel (2016) e Silva et al. (2014). Normalmente estas bibliotecas necessitam menos poder computacional, porém possuem menos ferramentas matemáticas para trabalhar com os dados obtidos.

### **2.1.2 Plataforma pan-tilt**

A geometria da plataforma pan-tilt não possui muitas variações dentro da literatura, visto que seu nome já descreve quais eixos possuem liberdade de rotação.

Sua estrutura mecânica é escolhida na literatura de acordo com a câmera utilizada. Para o trabalho de Mendonça (2014), utiliza-se uma estrutura de barras de alumínio pois este utiliza uma câmera DSLR em seu projeto Já no trabalho desenvolvido por Fusco et al. (2014) uma estrutura menor e menos robusta, pois este utiliza uma webcam comercial.

O motor escolhido para este tipo de sistema são, na maioria da literatura, servo-motores pois estes possuem baixo custo, controle de posição simples e a performance necessária para estas aplicações. Os modelos escolhidos variam de acordo com a estrutura mecânica da plataforma e peso da câmera. Para sua plataforma capaz de movimentar uma câmera DSLR, Mendonça (2014) utiliza motores do modelo HS-430BH, que possui em suas especificações torque de até 5 kg.cm. Já Kikuchi (2007) utiliza uma plataforma menor para uma webcam comercial, e por isso optou por motores do modelo CS-61, com torque de até 3 kg.cm. Azevedo (2017) assume na sua modelagem do sistema que este utiliza motores de corrente contínua controlados pela tensão de armadura, porém não foi construído um protótipo para que seja feita a comparação com a utilização de servo-motores.

Mendonça (2014) utiliza a plataforma Arduino para gerar os sinais PWM necessários para posicionamento dos motores. Esta plataforma traz vantagens como baixo custo e alta disponibilidade no mercado, já que esta é muito utilizada para prototipagem. Devido a necessidade de controlar muitos servo-motores além daqueles utilizados na plataforma pan-tilt, Fusco et al. (2014) utilizam a placa MRC-3024. Esta apresenta como principal vantagem a possibilidade de controlar até 16 motores simultaneamente, porém necessita de outra plataforma de processamento que envie os comandos de movimentação.

### **2.1.3 Obtenção de modelo**

Na literatura são encontradas diversas estratégias para obtenção do modelo matemático do sistema. Segundo Ljung (1999), modelos matemáticos possuem dois caminhos para o seu desenvolvimento. Um desses caminhos consiste em separar o sistema em subsistemas com propriedades conhecidas e que possuem a relação entre as suas variáveis determinadas através de trabalhos empíricos anteriores. O outro caminho se trata da obtenção dos modelos matemáticos baseados na experimentação, com a gravação e análise de dados de entrada e saída do sistema. Esse caminho é conhecido como identificação de sistemas.

Mendonça (2014) utiliza a identificação de sistemas, apontando a câmera para um objeto fixo e registrando a diferença, em pixels, do centro do objeto para o centro da imagem da câmera enquanto são aplicados sinais de entrada do tipo degrau no sistema. Após a coleta dos dados, estes são tratados utilizando o método de minimização dos erros de predição para estimar os modelos para o cada eixo na forma de função de transferência. O modelo identificado atendeu as necessidades de maneira satisfatória para o projeto do sistema de controlador.

Azevedo (2017) e Kikuchi (2007) modelam o sistema considerando a ótica da câmera e a matriz referente a movimentação dos motores. Utilizando a matriz Jacobiana da plataforma robótica e um modelo de cinemática diferencial (AZEVEDO, 2017) ou o princípio do fluxo ótico (KIKUCHI, 2007), foi estimado um modelo na forma de espaço de estados. Comparando com os modelos de função de transferência obtidos por Mendonça (2014), estes modelos representam de maneira mais detalhada e precisa a dinâmica do sistema, porém possui maior complexidade para sua construção.

#### **2.1.4 Estratégia de controle**

Para manter o objeto rastreado no centro da imagem da câmera, um sistema de controle regulatório deve ser projetado. Este sistema de controle deve apresentar erro nulo em regime permanente a distúrbios de posição. Na literatura a estratégia para o desenvolvimento do controlador do sistema varia de acordo com o método utilizado para criar o modelo do sistema.

Mendonça (2014) compara o método de Ziegler-Nichols para sintonia de controladores do tipo Proporcional-Integral com o método de alocação de polos e zeros utilizando o Lugar Geométrico das Raízes (LGR). Sua conclusão é de que a sintonia dos controladores pelo método Ziegler-Nichols apresenta uma resposta lenta, porém menos oscilatória e com menos sobressinal quando comparada com os controladores projetados utilizando o método de alocação de polos e zeros pelo LGR. Assim o método Ziegler-Nichols é mais apropriado para objetos que se movem lentamente ou aplicações onde a câmera deve ser mover de maneira mais lenta e suave. O método de alocação de polos e zeros pelo LGR é mais apropriado para sistemas que exigem uma resposta rápida.

Silva et al. (2014) apresenta um controlador tipo Proporcional-Integral-Derivativo (PID) com a sintonização feita de maneira experimental. Este utiliza instruções simples sobre quando aumentar ou diminuir cada parâmetro do controlador PID de acordo com a resposta do sistema, e chegou aos parâmetros finais através da realização de ensaios e ajustes no controlador.

Azevedo (2017) utiliza a realimentação de estados como estratégia de controle. Este projeto tinha como requisito a estabilidade e convergência do sistema de controle em malha fechada, assim seu foco é na análise destes.

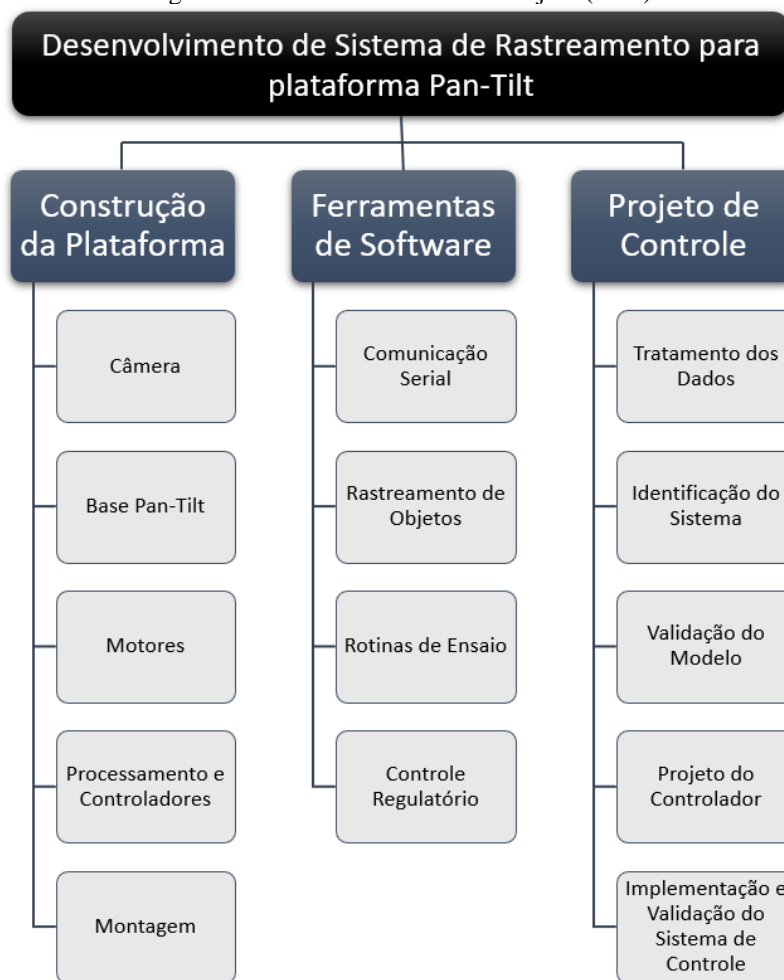
Através do critério de minimização linear quadrática, Kikuchi (2007) projetou dois controladores diferentes para o modelo, um Proporcional-Integral (PI) e um proporcional com estimação das perturbações externas utilizando um filtro de Kalman (LQG). O desempenho do controlador LQG apresenta uma resposta mais suave e menos oscilatória que o PI para este modelo.

### **3 METODOLOGIA**

O projeto de controle regulatório para uma plataforma pan-tilt exige o desenvolvimento de ferramentas de para processamento de imagem, movimentação da plataforma e etapas relacionadas ao projeto do sistema de controle em malha fechada.

A Estrutura Analítica do Projeto (EAP) apresentada na Figura 1 exhibe como o desenvolvimento deste projeto está estruturado.

Figura 1 - Estrutura Analítica do Projeto (EAP).



Fonte: O autor.

### 3.1 Construção da Plataforma

Este trabalho utiliza componentes de hardware e mecânica disponíveis no mercado para sua implementação. O principal critério para escolha de peças deve ser compatibilidade com a proposta do projeto e com os outros componentes.

#### 3.1.1 Câmera

Este é o principal componente do projeto, que funciona como sensor do sistema. Para escolha da câmera as características mais relevantes são a resolução, a taxa de quadros e a compatibilidade com o sistema operacional utilizado.

A câmera escolhida para o projeto é uma webcam da fabricante Logitech, modelo C270, exibida na Figura 2. Esta apresenta características que atendem as necessidades do projeto, conforme a Tabela 1. Seu principal diferencial está no formato e tamanho, que facilitam a montagem em bases pan-tilt que não são específicas para esse modelo de câmera.

Figura 2 - Webcam Logitech C270.



Fonte: LOGITECH

Tabela 1 - Informações da Webcam Logitech C270

<b>Modelo</b>	Logitech C270
<b>Resolução Máxima</b>	720p
<b>Taxa de Quadros</b>	30 quadros por segundo
<b>Compatibilidade</b>	Windows 10 ou posterior, Windows 8, Windows 7, macOS, Chrome OS, Android
<b>Tipo de Conexão</b>	USB

Fonte: LOGITECH.

### 3.1.2 Base Pan-Tilt

A base deve ser escolhida de acordo com opções do mercado que acomodem a câmera que é utilizada no projeto. A base fabricada por FILAFILL atende a necessidade do projeto. Esta não possui um nome de modelo específico, sendo nomeado pelo fabricante como *Support camera PAN-TILT*. A Figura 3 corresponde a uma foto desta base.

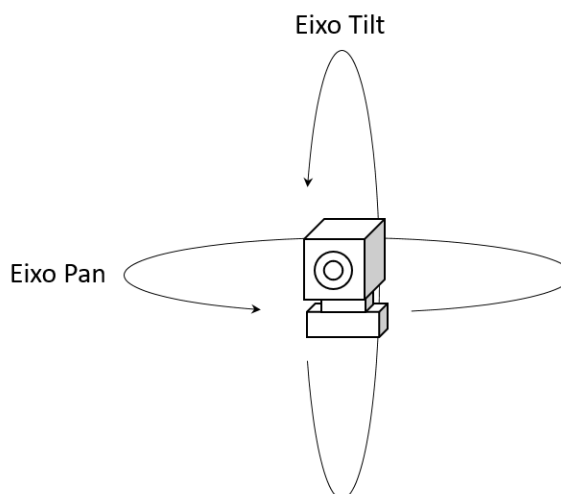
Figura 3 - Base Pan-Tilt (motores não inclusos)



Fonte: FILAFILL.

Os eixos pan-tilt são convencionados a partir da câmera como referência conforme a Figura 4.

Figura 4 - Eixos pan e tilt com a câmera como referência.



Fonte: O autor.

### 3.1.3 Servo-Motores

Estes são os componentes responsáveis pela movimentação dos eixos pan-tilt na plataforma. Estes necessitam possuir dimensionamento adequando para a base utilizada, precisão adequada para a aplicação do projeto e devem ser facilmente controlados.

Para este projeto utiliza-se servo-motores devido a sua facilidade de controle utilizando a plataforma Arduino, seu tamanho compacto e seu mecânico interno de controle de posição que garante a precisão necessária para o projeto.

De acordo com as recomendações do fabricante da base utilizada neste projeto, o protótipo utiliza servo-motores modelo MG90S, da fabricante TowerPro. As características deste modelo são apresentadas na Tabela 2.

Tabela 2 - Informações do servo-motor TowerPro MG90S.

<b>Modelo</b>	MG90S
<b>Peso</b>	13.4g
<b>Dimensões</b>	22.8×12.2×28.5mm
<b>Torque de imobilização</b>	1.8kg/cm (4.8V); 2.2kg/cm (6.6V)
<b>Velocidade de operação</b>	0.10sec/60degree (4.8V); 0.08sec/60degree (6.0V)
<b>Tensão de operação</b>	4.8V
<b>Largura de Banda Morta</b>	1us

Fonte: TOWERPRO.

### 3.1.4 Processamento e Controladores

Para execução dos softwares de processamento de imagem e ações do sistema de controle é utilizado um computador desktop com sistema operacional Windows 10. Este foi escolhido pois apresenta capacidade de processamento suficiente para executar as todas as ações necessárias para a taxa de amostragem do sistema. A Tabela 3 apresenta a configuração deste computador. Para controle dos motores foi utilizada uma placa da plataforma Arduino,

que recebe a informação de posição dos motores via comunicação serial. A Tabela 4 apresenta as configurações da placa Arduino utilizada.

Tabela 3 - Informações do computador desktop.

<b>Versão do Sistema Operacional</b>	Windows 10 Education 64 bits
<b>Processador</b>	Intel Core i5-3470 @3,20GHz
<b>Memória Ram</b>	8,00GB DDR3
<b>Placa de Vídeo Dedicada</b>	NVIDIA GeForce GTX 750 Ti
<b>Fonte de Alimentação</b>	EVGA 600W 80 Plus Bronze
<b>Armazenamento</b>	Sandisk SSD 240GB

Fonte: O autor.

Tabela 4 - Informações da plataforma Arduino.

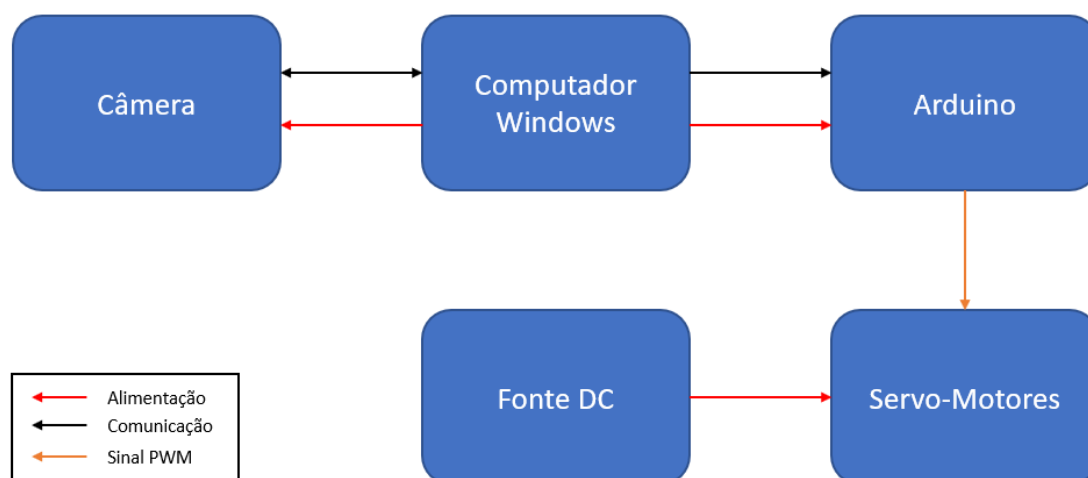
<b>Modelo</b>	ARDUINO UNO REV3
<b>Microcontrolador</b>	ATmega328P
<b>Velocidade de Clock</b>	16MHz
<b>Pinos PWM digital</b>	6

Fonte: ARDUINO.

### 3.1.5 Montagem

A etapa de montagem do protótipo apresenta duas atividades distintas. Uma é a integração mecânica dos componentes, onde os motores e a câmera devem ser montados na base pan-tilt. A outra etapa consiste nas conexões necessárias para a comunicação e alimentação do sistema. O diagrama da Figura 5 apresenta as conexões de comunicação do sistema e alimentação dos componentes.

Figura 5 - Diagrama do Blocos do Protótipo.



Fonte: O autor.



### 3.2 Ferramentas de software

Para o funcionamento do sistema diversas aplicações de software devem ser implementadas. A plataforma Arduino deve ser capaz de receber dados via serial e interpretar esses dados, posicionando corretamente os motores. O computador Windows deve ter a capacidade de processar a imagem, realizar as etapas referentes ao controle em malha fechada e enviar para o Arduino via serial. Também serão implementados no computador Windows algoritmos para realização de ensaios automatizados, assim facilitando a coleta de dados. Para o desenvolvimento no computador Windows a linguagem de programação Python foi escolhida devido a sua estrutura simples, velocidade de execução, disponibilidade de bibliotecas para processamento de imagem e capacidades multiplataforma.

#### 3.2.1 Comunicação Serial

O projeto apresenta a necessidade de um algoritmo capaz de prover a comunicação entre o computador Windows e a plataforma Arduino, via serial, para que o posicionamento dos servo-motores seja alterado pelo algoritmo de controle. No Windows, a biblioteca *pySerial* fornece ferramentas para enviar as mensagens. Assim, no algoritmo é formada uma mensagem no formato  $X[1]Y[2]f$ , onde  $[1]$  e  $[2]$  são os valores em microssegundos da largura de pulso correspondente ao ângulo desejado para os eixos pan e tilt, respectivamente. O caractere  $f$  indica o final da mensagem. A conversão do ângulo para largura de pulso é feita pelo mesmo algoritmo seguindo a Eq. (1).

$$w = \left( 1000 + a \frac{1000}{180} \right) \quad (1)$$

$w$  - Largura de Pulso [microssegundos]

$a$  - Ângulo desejado [Graus]

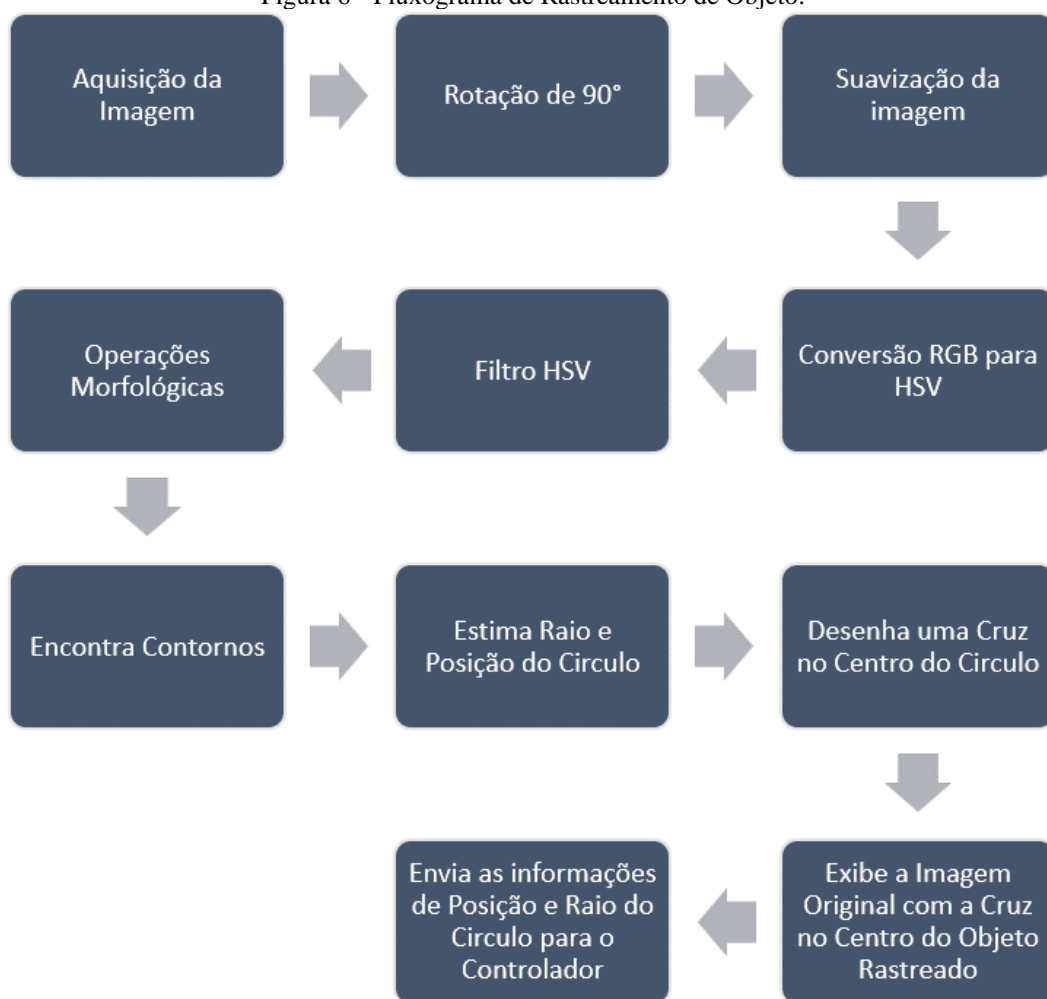
A plataforma Arduino recebe a mensagem via serial através da função *Serial.read* e, através da função *servo.writeMicroseconds* da biblioteca *servo.h* modifica a posição dos motores de acordo com a mensagem recebida.

#### 3.2.2 Rastreamento de Objetos

O algoritmo de rastreamento de objetos pode ser resumido no fluxograma da Figura 6.

Para o processamento de imagens, a biblioteca *OpenCV* é utilizada. Esta é uma biblioteca em código aberto e disponível para diversas plataformas.

Figura 6 - Fluxograma de Rastreamento de Objeto.



Fonte: O autor.

O algoritmo começa com a aquisição da imagem da câmera. É necessário rotacionar a imagem em 90° devido a maneira como a câmera foi posicionada na plataforma pan-tilt.

A suavização da imagem então é feita utilizando a função *GaussianBlur*, que aplica o desfoque da imagem utilizando uma função gaussiana, assim reduzindo os níveis de ruído e detalhamento da imagem.

Na próxima etapa do algoritmo, a imagem é convertida do formato Red-Green-Blue (RGB) para o formato Hue-Saturation-Value (HSV) através da função *cvtColor*. Isso é feito pois o filtro de cor utilizado em formato HSV é mais fácil de ser ajustado.

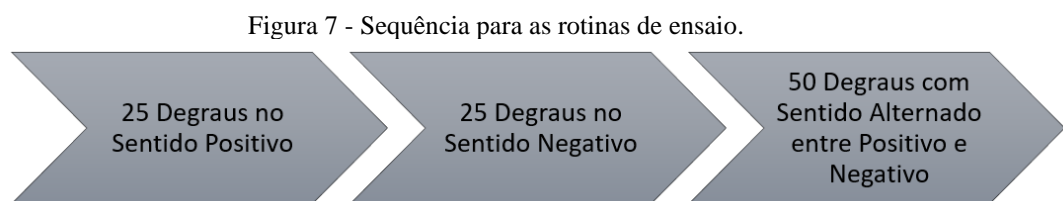
Após a conversão aplica-se um filtro previamente estabelecido, assim criando uma imagem onde os pontos fora dos parâmetros HSV do filtro possuem cor preta e os pontos dentro dos parâmetros HSV do filtro possuem cor branca. São aplicadas as operações morfológicas *erode* e *dilate*, que consistem em séries de convoluções na imagem para remoção de ruídos.

Após a aplicação dos filtros, são encontrados os contornos da imagem resultante. Isso é feito através da função *findContours*. Um teste é feito para determinar se algum contorno foi encontrado na imagem e, em caso positivo, o maior dos contornos encontrados é utilizado para estimar o raio e a posição do círculo referente ao objeto rastreado.

O algoritmo então exibe a imagem original com uma cruz desenhada no centro do objeto rastreado e retorna as informações de posição e raio do objeto para serem utilizadas pelo controlador. A informação de posição refere-se a diferença entre a posição do objeto e o centro da imagem. Assim o centro da imagem corresponde a posição zero e o deslocamento do centro gera valores inteiros, positivos ou negativos de acordo com a convenção, que correspondem a diferença de pixels do centro do objeto rastreado para o centro da imagem.

### 3.2.3 Rotinas de Ensaio

Para auxiliar na coleta de dados o projeto possui uma rotina que realiza ensaios referente a resposta ao degrau do sistema em malha aberta. Essa rotina varia o sentido do sinal de entrada conforme é apresentado na Figura 7.



Fonte: O autor.

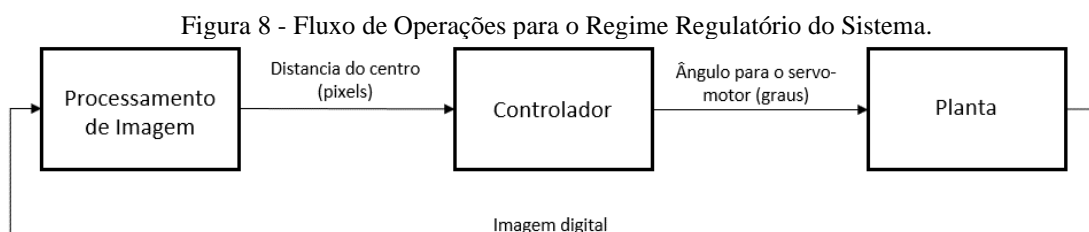
Essa sequência tem o objetivo incluir nos ensaios os dados referentes às variações do sistema, principalmente com relação a folga presente nos servo-motores.

Durante as execuções desta rotina de ensaios o tempo de amostragem de cada coleta de dados é medido e caso este ultrapasse 10% do tempo de amostragem nominal o ensaio é invalidado e repetido.

As sequências são aplicadas da mesma maneira para o eixo pan e para o eixo tilt.

### 3.2.4 Controle Regulatório

Para o regime regulatório do sistema de controle, um algoritmo executa o rastreamento do objeto e, com os valores de posição retornados por este, aplica o valor na entrada do controlador, que processa os dados e envia os comandos de posicionamento dos motores para o Arduino via serial. O valor de sinal de referência para este algoritmo é zero para o eixo pan e para o eixo tilt, assim este regula o sistema para que a posição do objeto rastreado seja zero, ou seja, o centro da imagem. A Figura 8 apresenta o fluxo de operações para o regime de controle regulatório.



Fonte: O autor.

### 3.3 Projeto de Controle

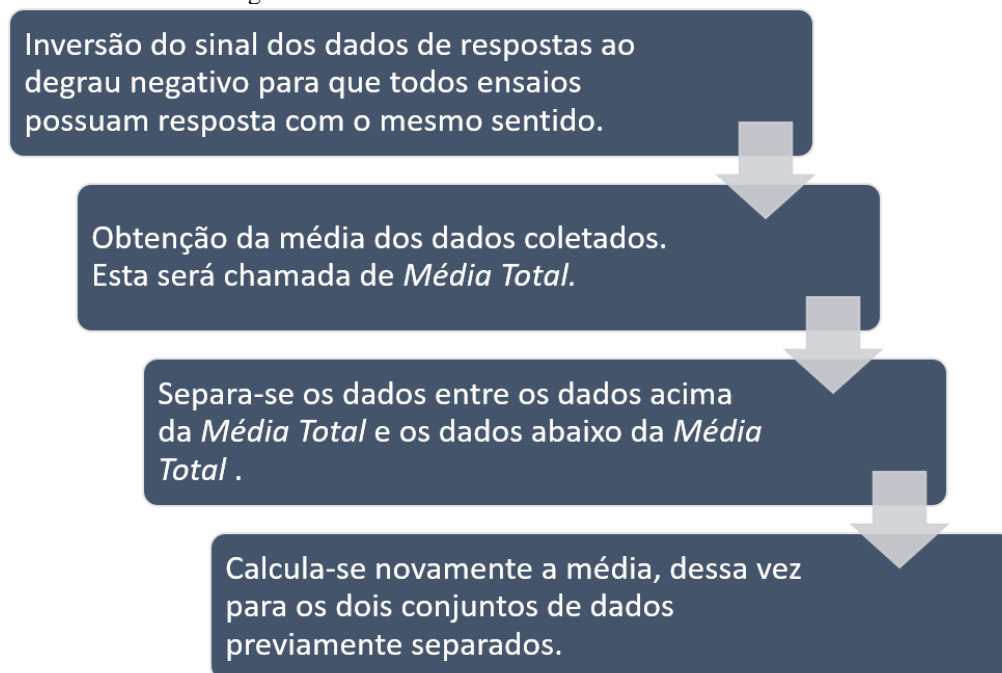
Esta seção apresenta as metodologias utilizadas para a construção do sistema de controle baseando-se principalmente nas ferramentas aprendidas durante o curso.

#### 3.3.1 Tratamento de dados

Para a construção do modelo do sistema, coletam-se dados referentes a resposta ao degrau do sistema em malha aberta, seguindo a rotina apresentada no tópico 3.2.3 deste documento. O degrau aplicado é de 10 graus. Este valor foi definido experimentalmente de acordo com o campo de visão da câmera.

O ensaio é executado com a distância entre o objeto rastreado e a câmera de 0,3, 0,6 e 1,2 metro. Isso resulta em um volume de dados de trezentos ensaios para cada eixo. Os dados obtidos são tratados conforme a Figura 9 para que os modelos obtidos representem as variações do sistema satisfatoriamente.

Figura 9 - Fluxo de Tratamento dos Dados Coletados.



Fonte: O autor.

O objetivo do tratamento de dados é que se objetiva um modelo médio do comportamento do sistema, um modelo médio para quando o sistema não apresenta folga nos seus atuadores e um modelo para quando o sistema apresenta folga nos seus atuadores. Isso é necessário pois uma característica dos servo-motores convencionais é a presença de folga, principalmente quando se inverte o sentido de rotação destes.

Por conta do tamanho do objeto rastreado e do ambiente onde realizam-se os ensaios, define-se a distância de 0,6 metro entre o objeto rastreado e a câmera como distância nominal de apresentação.

Observa-se através dos ensaios que a distância entre o objeto rastreado e a câmera tem pouca influência na resposta do sistema. Assim, para garantir o desempenho do controlador, utiliza-se as respostas obtidas com a atuação do atuador sem folga para a maior distância, a

atuação do atuador com folga para a menor distância e a média de todas as respostas para a distância nominal de apresentação.

Para garantir o bom desempenho durante as apresentações, considera-se também as respostas de maior e menor ganho absoluto durante os ensaios na distância nominal de apresentação.

A Tabela 5 descreve todas as respostas que devem ser utilizadas para construção dos modelos que representam o sistema.

Tabela 5 - Respostas em malha aberta utilizadas para construção dos modelos

<b>Resposta</b>	<b>Descrição</b>
Média Alta	Média dos ensaios sem folga no atuador, realizados com distância entre o objeto rastreado e a câmera de 1,2 metros.
Média Baixa	Média dos ensaios sem folga no atuador, realizados com distância entre o objeto rastreado e a câmera de 0,3 metro.
Média Total	Média dos ensaios com e sem folga no atuador, realizados com distância entre o objeto rastreado e a câmera de 0,6 metro.
Máxima	Resposta com maior ganho absoluto obtida nos ensaios realizados com distância entre o objeto rastreado e a câmera de 0,6 metro.
Mínima	Resposta com menor ganho absoluto obtida nos ensaios realizados com distância entre o objeto rastreado e a câmera de 0,6 metro.

Fonte: O autor.

### 3.3.2 Identificação do Sistema

Para obtenção do modelo utiliza-se a identificação do sistema determinando os parâmetros de um modelo auto regressivo com entrada exógena (ARX) através do método de Mínimos Quadrados.

O modelo ARX pode ser escrito de forma compacta conforme a Eq. (2).

$$A(q)y(n) = B(q)u(n) + e(n) \quad (2)$$

Onde temos  $A(q)$  e  $B(q)$  são polinômios no operador  $q$ , que é um operador de atraso.

Temos  $u(n)$  que é o sinal de entrada,  $y(n)$  que é o sinal de saída e  $e(n)$  que é uma variável aleatória branca. A variável aleatória branca será desconsiderada para esta construção pois esta não é conhecida.

Supondo que se esteja trabalhando com uma planta de segunda ordem, assume-se que os polinômios  $A(q)$  e  $B(q)$  são conforme Eq. (3) e Eq. (4).

$$A(q) = 1 - a_1q^{-1} - a_2q^{-2} \quad (3)$$

$$B(q) = b_1 + b_2q^{-1} + b_3q^{-2} \quad (4)$$

Assim podemos reescrever a Eq. (2) como proposto na Eq. (5):

$$y(n) = a_1 y(n-1) + a_2 y(n-2) + b_1 u(n) + b_2 u(n-1) + b_3 u(n-2) \quad (5)$$

A Eq. (5) pode ser descrita na seguinte forma matricial:

$$y(n) = [y(n-1) \quad y(n-2) \quad u(n) \quad u(n-1) \quad u(n-2)] \begin{bmatrix} a_1 \\ a_2 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} \quad (6)$$

Para uma série de  $N$  medições, a Eq. (6) pode ser representada de forma aumentada como:

$$\begin{bmatrix} y(2) \\ y(3) \\ y(4) \\ \vdots \\ y(N-1) \end{bmatrix} = \begin{bmatrix} y(1) & y(0) & u(2) & u(1) & u(0) \\ y(2) & y(1) & u(3) & u(2) & u(1) \\ y(3) & y(2) & u(4) & u(3) & u(2) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ y(N-1) & y(N-2) & u(N) & u(N-1) & u(N-2) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} \quad (7)$$

A relação matricial da Eq. (7) pode ser escrita de forma compacta como:

$$Y = \Psi \theta \quad (8)$$

O método dos Mínimos Quadrados corresponde a utilizar a solução da pseudo-inversa na Eq. (8) para estimar os parâmetros da matriz  $\theta$ , conforme a Eq. (9):

$$\theta = (\Psi^T \Psi)^{-1} \Psi^T Y \quad (9)$$

Aplicando a transformada  $Z$  na Eq. (5) temos a função de transferência que representa esta planta:

$$G(z) = \frac{Y(z)}{U(z)} = \frac{b_3 z^2 + b_2 z + b_1}{z^2 - a_1 z - a_2} \quad (10)$$

Com a substituição dos parâmetros estimados na Eq. (9) na função de transferência da Eq. (10) apresenta-se modelo estimado da planta, assim concluindo a modelagem desta.

O software Matlab possui a ferramenta *ident* que é capaz de realizar este método.

### 3.3.3 Validação do Modelo

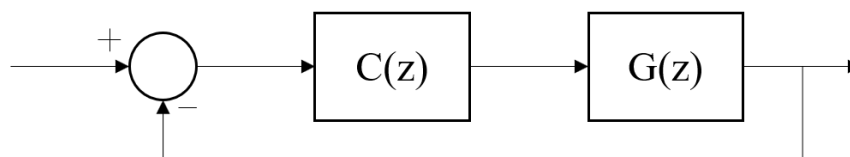
A validação do modelo consiste em verificar se o modelo obtido na identificação do sistema corresponde satisfatoriamente ao modelo real. Assim para este trabalho aplica-se um sinal do tipo degrau em malha aberta na planta simulada, que consiste na função de transferência obtida na identificação do sistema.

A resposta da planta simulada deve ser comparada a resposta da planta real para um degrau do mesmo valor. Se a comparação mostrar que o modelo obtido atende o projeto de maneira satisfatória conclui-se que o modelo está validado.

### 3.3.4 Projeto do Controlador

Considerando o sistema de controle em malha fechada da Figura 10, onde  $G(z)$  corresponde a planta e  $C(z)$  ao controlador projetado, o projeto por Lugar Geométrico das Raízes (LGR) corresponde em mostrar a contribuição de cada polo ou zero do sistema em malha aberta nas localizações dos polos em malha fechada para todos os valores de ganho do controlador  $C(z)$  (BEZERRA, 2007).

Figura 10 - Sistema de Controle em Malha Fechada Genérico.



Fonte: O autor.

A ferramenta *rltool* do software *MATLAB* permite que os polos e zeros do controlador sejam posicionados livremente no plano complexo, exibindo linhas que representam os caminhos dos polos do sistema em malha fechada de acordo com a variação do ganho do controlador. Esta ferramenta também exibe, em tempo real, as respostas do sistema em malha fechada para diversos tipos de sinais, permitindo que o ganho do controlador e a posição de seus polos e zeros sejam ajustados de acordo com os requisitos do projeto.

Como o foco deste projeto é o controle regulatório para demonstração, os requisitos foram determinados de maneira experimental. Assim definiu-se que sistema deve se comportar de modo a rejeitar distúrbios de posição com tempo de acomodação máximo de 0,5 segundo e com o mínimo de sobressinal possível. Este deve apresentar também erro nulo ao sinal de referência do tipo degrau.

### 3.3.5 Implementação e Validação do Sistema de Controle

O sistema de controle é implementado de modo que os valores resultantes de uma referência subtraída dos valores de posição retornados pelo algoritmo de rastreamento de objetos, do tópico 3.2.2 deste documento, resultem no sinal de entrada do controlador. Este calcula o sinal de controle que corresponde a quantos graus o motor deve se deslocar e o envia para o Arduino. Para execução em regime regulatório o valor da referência é igual a zero pois desta forma, em regime permanente, o objeto rastreado fica no centro da imagem captada pela câmera.

A validação do sistema de controle é feita de duas maneiras. Primeiramente avalia-se o desempenho do controlador realizando ensaios de resposta ao degrau e comparando-os com os resultados esperados. Após a validação da resposta ao degrau, o sistema é configurado para regime regulatório e avalia-se sua resposta conforme a movimentação do objeto rastreado, para assim verificar seu desempenho na resposta a distúrbios de posição.

## 4 APRESENTAÇÃO E DISCUÇÃO DOS RESULTADOS

Esta seção apresenta os resultados de cada etapa da metodologia deste trabalho.

### 4.1 Construção da Plataforma

A plataforma foi construída utilizando apenas peças e componentes disponíveis no mercado. Com relação a ferramentas, apenas uma chave de fenda cruzada M2 foi necessária para a construção do protótipo.

#### 4.1.1 Câmera

A câmera utilizada, webcam Logitech C270, apresentou imagem com nitidez e resolução suficientes para a aplicação, conforme a Figura 11.

Figura 11 - Imagem capturada pela webcam Logitech C270.



Fonte: O autor.

#### 4.1.2 Base Pan-Tilt

A base pan-tilt da fabricante FilaFill foi de fácil montagem e compatível com o restante dos componentes do protótipo.

#### 4.1.3 Servo-Motores

Os servo-motores MG90S, da TowerPro, atendem a necessidade do projeto sem a necessidade de qualquer alteração ou ajuste.

Durante os ensaios, o consumo máximo de corrente foi de 0,2 ampere por servo-motor. Assim os dois servo-motores são alimentados por uma fonte de corrente contínua de 5 Volts e capacidade de 2 amperes, que garante a potência necessária para o funcionamento.



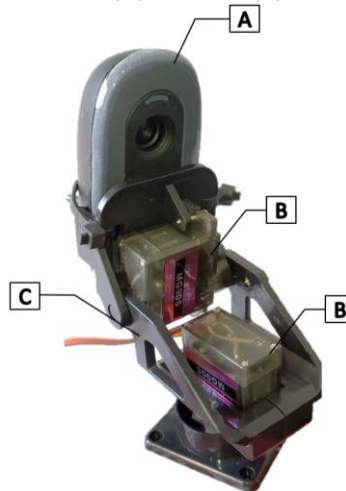
#### 4.1.4 Processamento e Controladores

Durante a realização dos ensaios o computador Windows e o Arduino não apresentaram problemas de desempenho. O tempo de amostragem de 40 milissegundos é respeitado pelo sistema desde que as duas plataformas não executem tarefas secundárias.

#### 4.1.5 Montagem

A montagem é dividida em duas etapas. A primeira trata da integração da câmera e dos servo-motores com a base pan-tilt. Esta é apresentada na Figura 12.

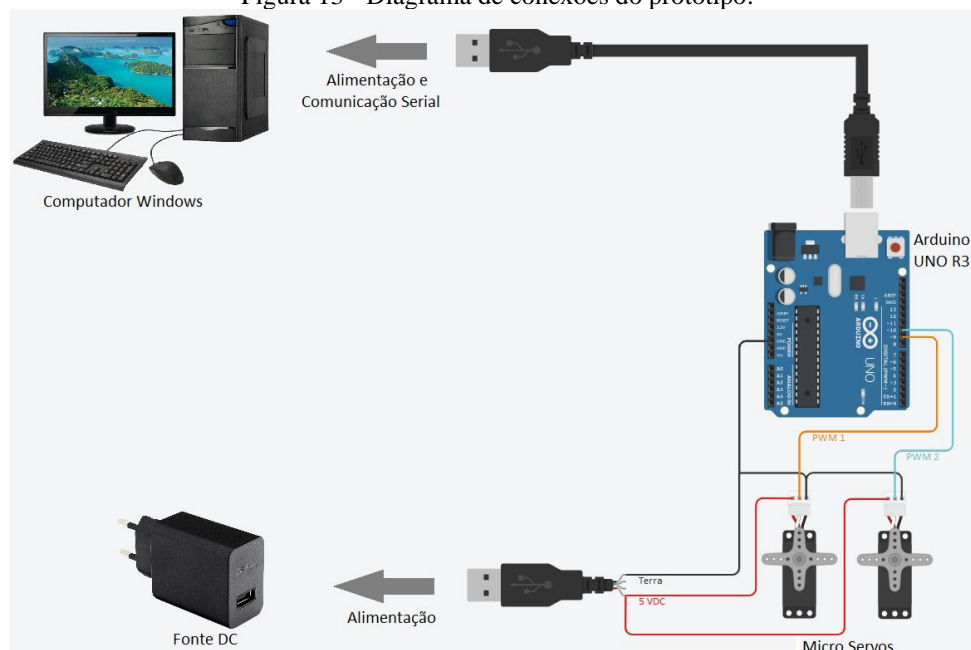
Figura 12 - Apresentação Câmera Pan-Tilt. (A) Câmera. (B) Servo-motores. (C) Base pan-tilt.



Fonte: O autor.

A segunda etapa trata da conexão de alimentação elétrica e de comunicação dos componentes do protótipo. O diagrama da Figura 13 ilustra como essas conexões são feitas.

Figura 13 - Diagrama de conexões do protótipo.



Fonte: O autor.

## 4.2 Ferramentas de Software

As ferramentas de software para o computador Windows foram desenvolvidas utilizando a IDE Python chamada PyCharm, da desenvolvedora JetBrains. A estrutura das ferramentas de software é composta por três objetos e um código principal de execução. O primeiro objeto corresponde a comunicação serial com o Arduino. O segundo objeto é responsável por todos os algoritmos relacionados ao processamento de imagens. O terceiro objeto corresponde a todas as funções relacionadas ao projeto de controle, tanto na aplicação deste em malha fechada como nas rotinas de ensaio utilizadas no projeto. O código para o Arduino foi feito na IDE da própria plataforma. Este conta com a capacidade de comunicação serial e movimentação dos servo-motores. Foi determinado para o sistema completo uma taxa de amostragem nominal de 40 milissegundos, assim todas as rotinas respeitam essa taxa de amostragem.

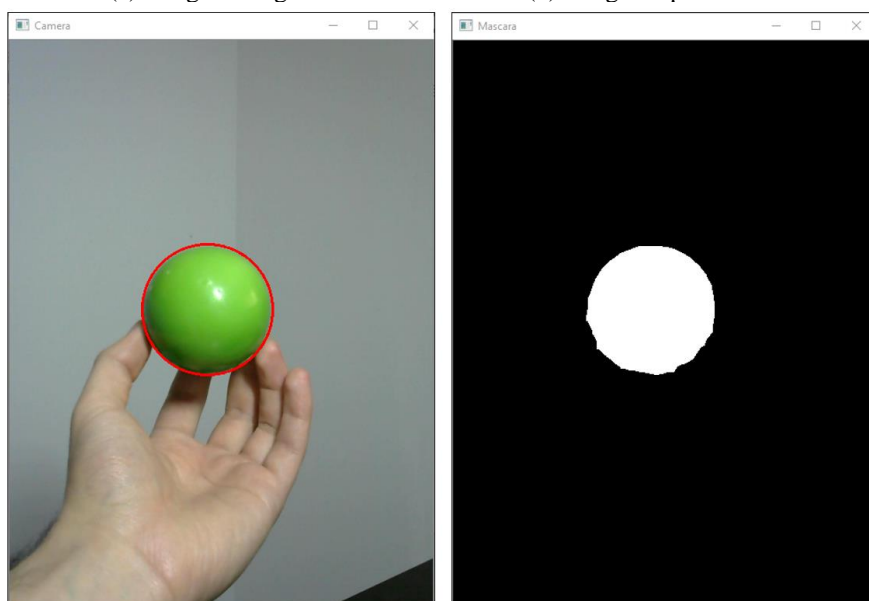
### 4.2.1 Comunicação Serial

A comunicação serial entre o computador Windows e o Arduino é feita via conexão USB, onde este utiliza a porta serial COM disponível no sistema operacional Windows e a porta serial zero do Arduino. A comunicação é feita com velocidade de transmissão de 9600 bits por segundo.

### 4.2.2 Rastreamento de Objetos

O filtro de cores apresenta resultados satisfatórios, detectando com precisão os objetos e apresentando pouco ruído. A Figura 14 apresenta uma comparação entre a imagem capturada pela câmera e a imagem filtrada, assim como o círculo com posição e raio estimados pelo filtro na imagem capturada.

Figura 14 - Desempenho do filtro de cores para detecção de objetos.  
(a) Imagem Original. (b) Imagem após filtro.

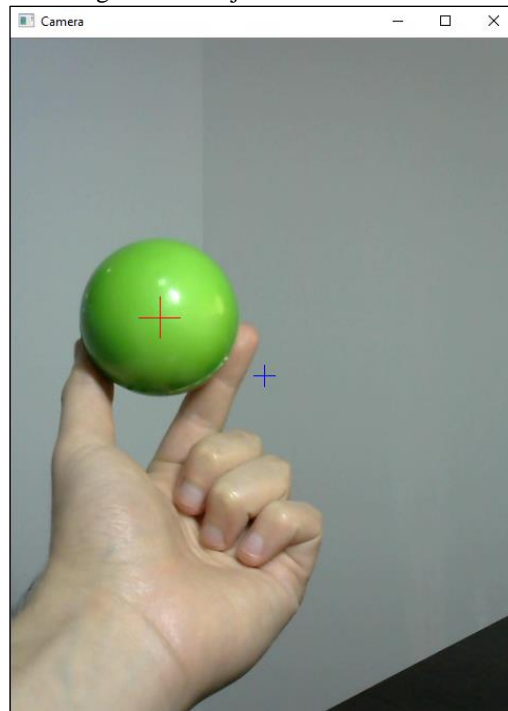


Fonte: O autor.

O rastreamento de objeto retorna a distância dos eixos vertical e horizontal para a origem da imagem e o raio estimado do objeto rastreado. Este também desenha sobre a imagem original

uma cruz vermelha no centro do objeto rastreado e uma cruz azul no centro da imagem, como demonstrado na Figura 15.

Figura 15 - Objeto sendo rastreado.



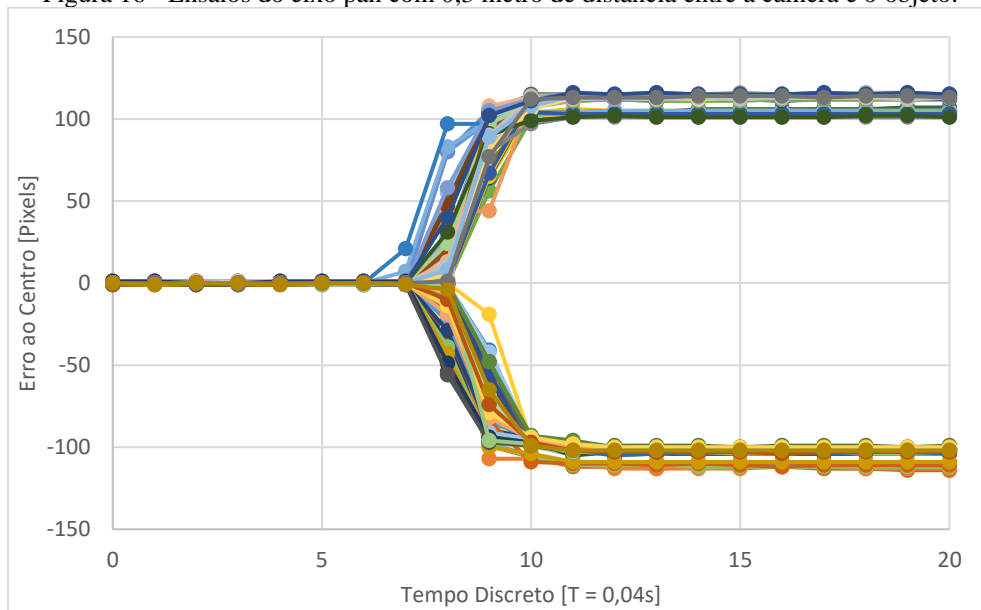
Fonte: O autor.

#### 4.2.3 Rotinas de Ensaio

As rotinas de ensaio ocorrem de acordo com o especificado, realizando 25 ensaios com sinal de entrada do tipo degrau positivo, 25 ensaios com sinal de entrada do tipo degrau negativo e 50 ensaios com sinal de entrada do tipo degrau alternando o sentido a cada ensaio. Os ensaios em que alguma das amostras não atende ao período de amostragem nominal são descartados. Assim as rotinas resultam em 100 ensaios para cada eixo, garantindo o tempo de amostragem.

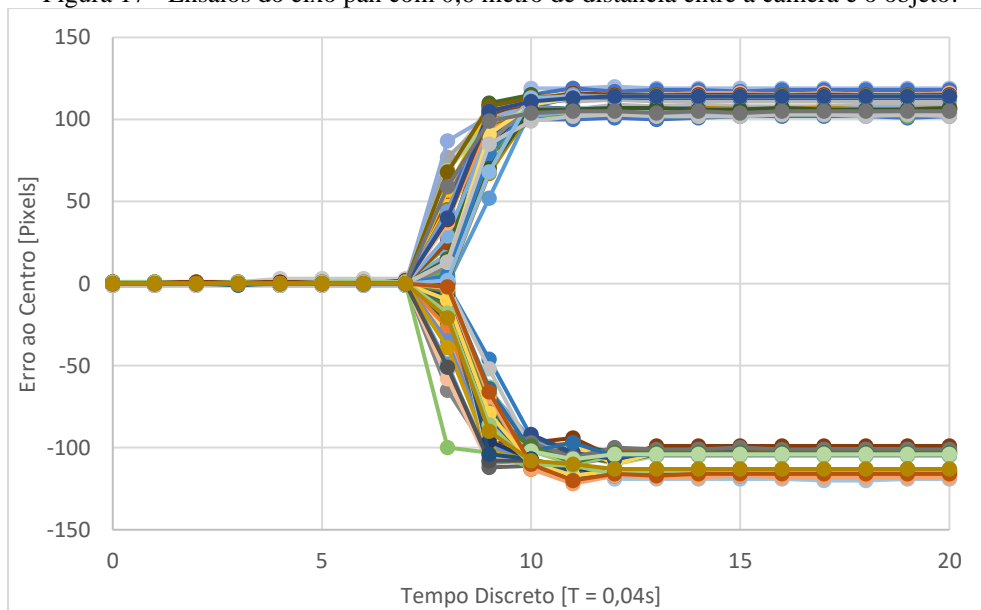
As Figuras 16, 17 e 18 apresentam, em gráfico, os dados coletados nos ensaios do eixo pan com distância entre objeto rastreado e câmera de 0,3 metro, 0,6 metro e 1,2 metro respectivamente. No gráfico, o eixo das abscissas representa o tempo discreto e o eixo das ordenadas o erro ao centro da imagem em pixels.

Figura 16 - Ensaio do eixo pan com 0,3 metro de distância entre a câmera e o objeto.



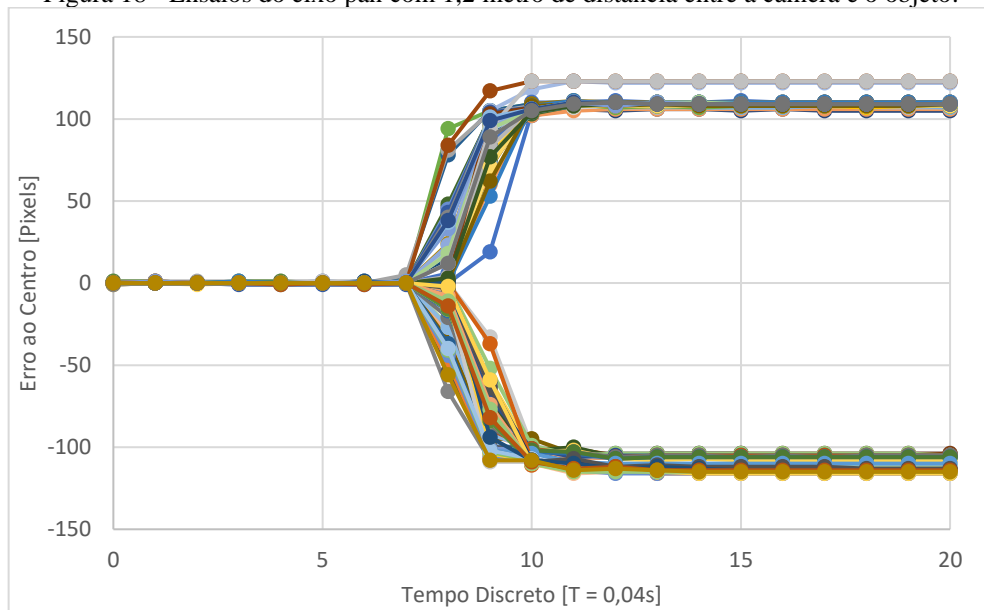
Fonte: O autor.

Figura 17 - Ensaio do eixo pan com 0,6 metro de distância entre a câmera e o objeto.



Fonte: O autor.

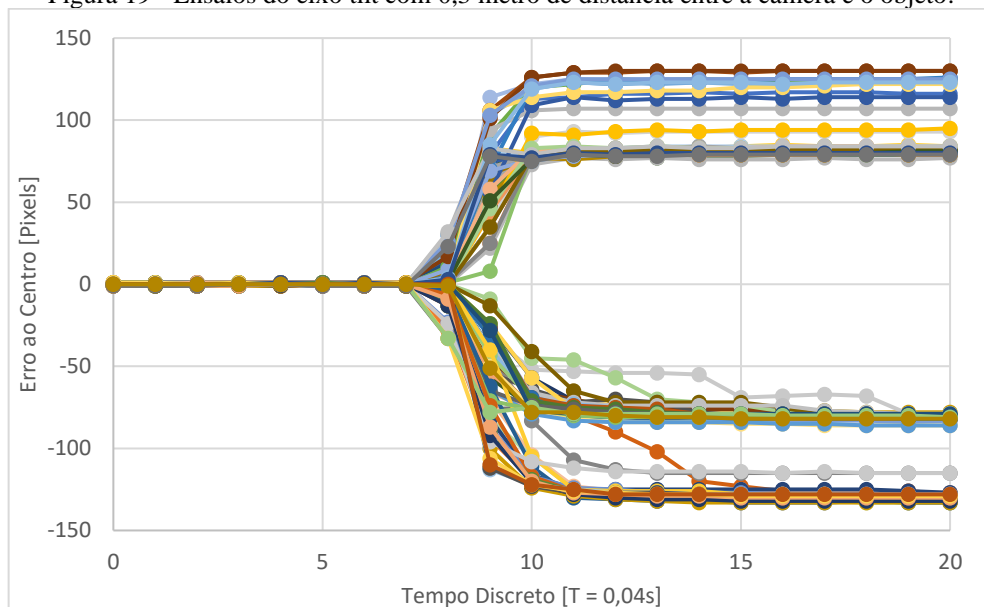
Figura 18 - Ensaios do eixo pan com 1,2 metro de distância entre a câmera e o objeto.



Fonte: O autor.

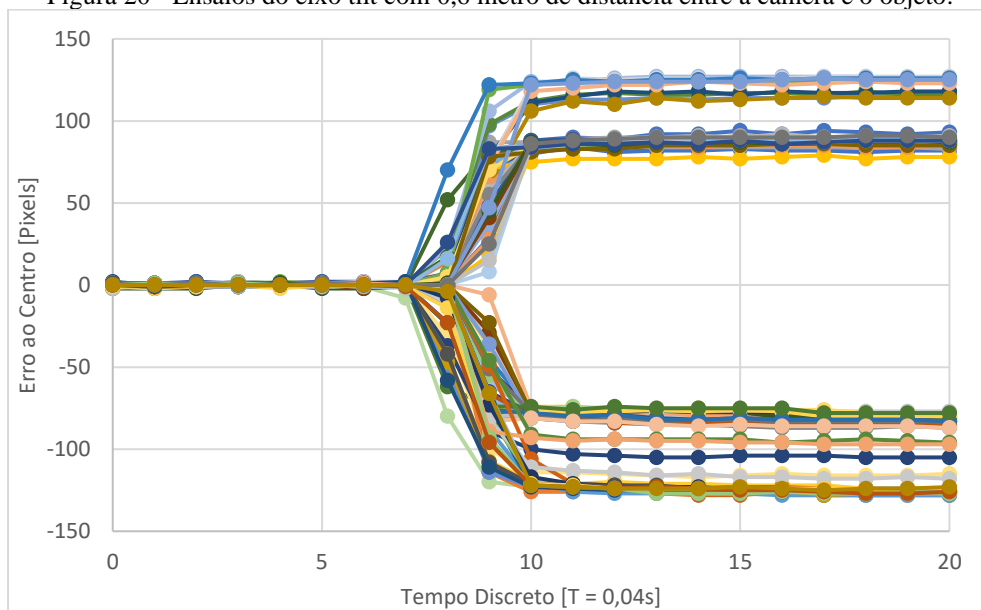
As Figuras 19, 20 e 21 apresentam, em gráfico, os dados coletados nos ensaios do eixo tilt com distância entre objeto rastreado e câmera de 0,3 metro, 0,6 metro e 1,2 metro respectivamente. No gráfico, o eixo das abscissas representa o tempo discreto e o eixo das ordenadas o erro ao centro da imagem em pixels.

Figura 19 - Ensaios do eixo tilt com 0,3 metro de distância entre a câmera e o objeto.



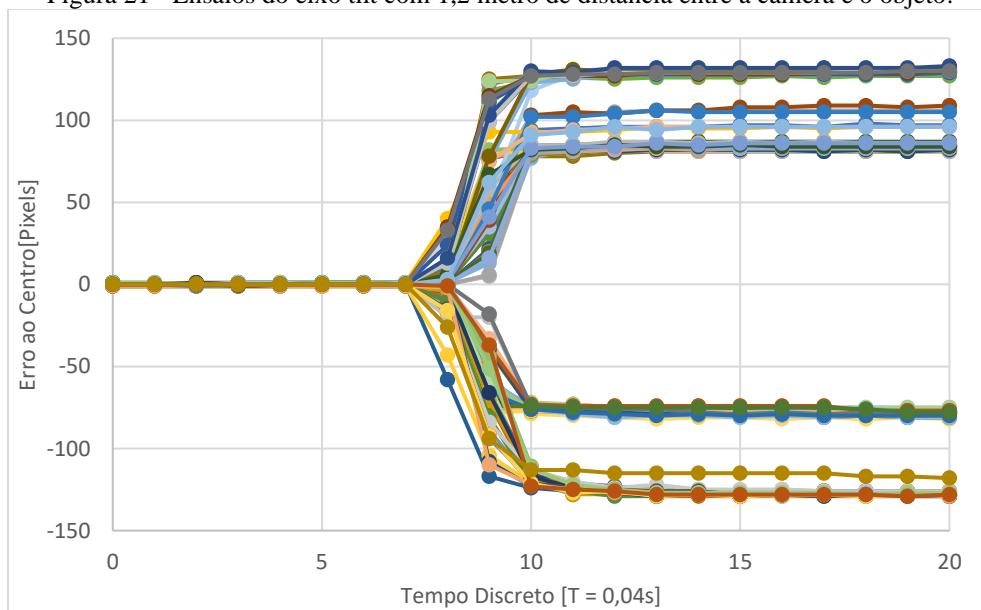
Fonte: O autor.

Figura 20 - Ensaio do eixo tilt com 0,6 metro de distância entre a câmera e o objeto.



Fonte: O autor.

Figura 21 - Ensaio do eixo tilt com 1,2 metro de distância entre a câmera e o objeto.



Fonte: O autor.

#### 4.2.4 Controle Regulatório

O algoritmo de controle regulatório apresenta resultado satisfatório. Este para de enviar comandos para os servo-motores caso não encontre o objeto. Uma tecla para zerar o erro acumulado pelo sistema de controle foi adicionada para correção de casos onde este é mascarado por erros no filtro de cores como quando, por exemplo, o ambiente apresenta a mesma cor do objeto rastreado.

### 4.3 Projeto de Controle

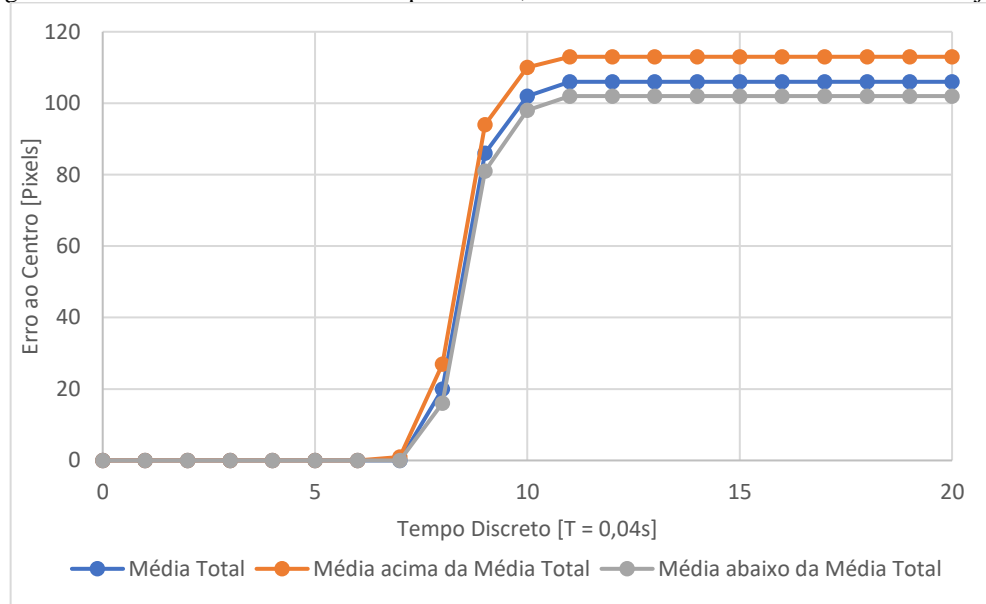
Esta seção apresenta os resultados as etapas do projeto de controle desde o tratamento dos dados dos ensaios em malha aberta até a implementação e validação do sistema de controle

em malha fechada. Foi determinado para o sistema a taxa de amostragem nominal de 40 milissegundos. Essa taxa foi determinada a partir de características dos componentes do protótipo.

#### 4.3.1 Tratamento de Dados

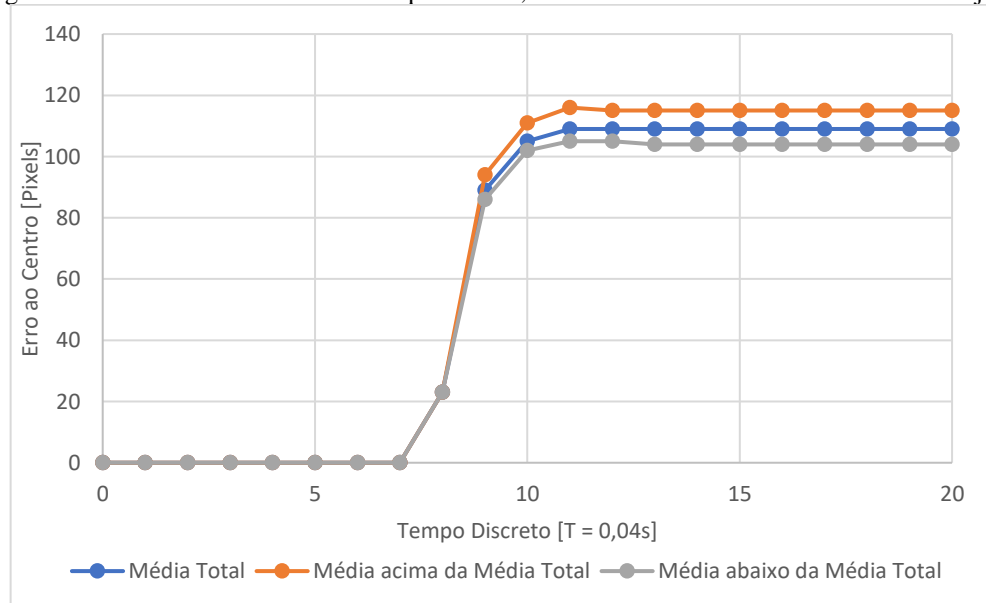
Os resultados obtidos após o tratamento dos dados para os ensaios no eixo pan estão representados nas Figuras 22, 23 e 24 para os ensaios com distância entre objeto rastreado e câmera de 0,3 metro, 0,6 metro e 1,2 metro respectivamente. No gráfico, o eixo das abscissas representa o tempo discreto e o eixo das ordenadas o erro ao centro da imagem em pixels.

Figura 22 - Médias dos ensaios do eixo pan com 0,3 metro de distância entre a câmera e o objeto.



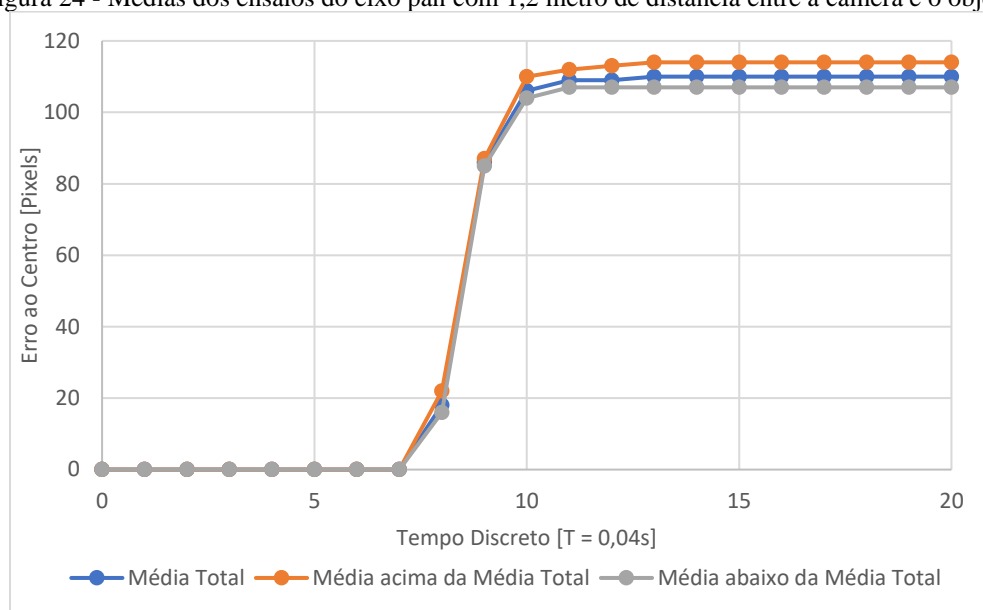
Fonte: O autor.

Figura 23 - Médias dos ensaios do eixo pan com 0,6 metro de distância entre a câmera e o objeto.



Fonte: O autor.

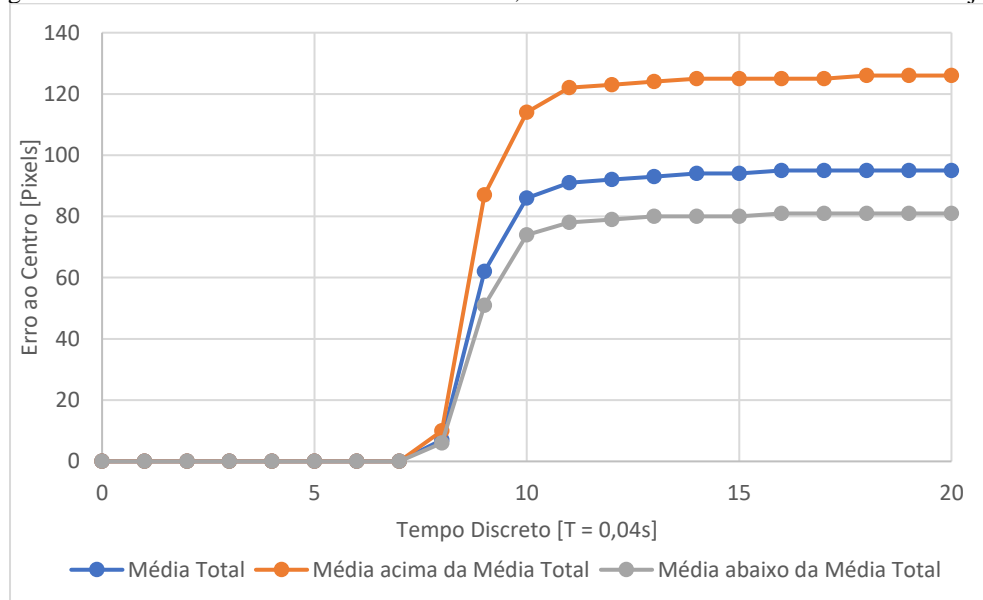
Figura 24 - Médias dos ensaios do eixo pan com 1,2 metro de distância entre a câmera e o objeto.



Fonte: O autor.

Os resultados obtidos após o tratamento dos dados para os ensaios no eixo tilt estão representados nas Figuras 25, 26 e 27 para os ensaios com distância entre objeto rastreado e câmera de 0,3 metro, 0,6 metro e 1,2 metro respectivamente. No gráfico, o eixo das abscissas representa o tempo discreto e o eixo das ordenadas o erro ao centro da imagem em pixels.

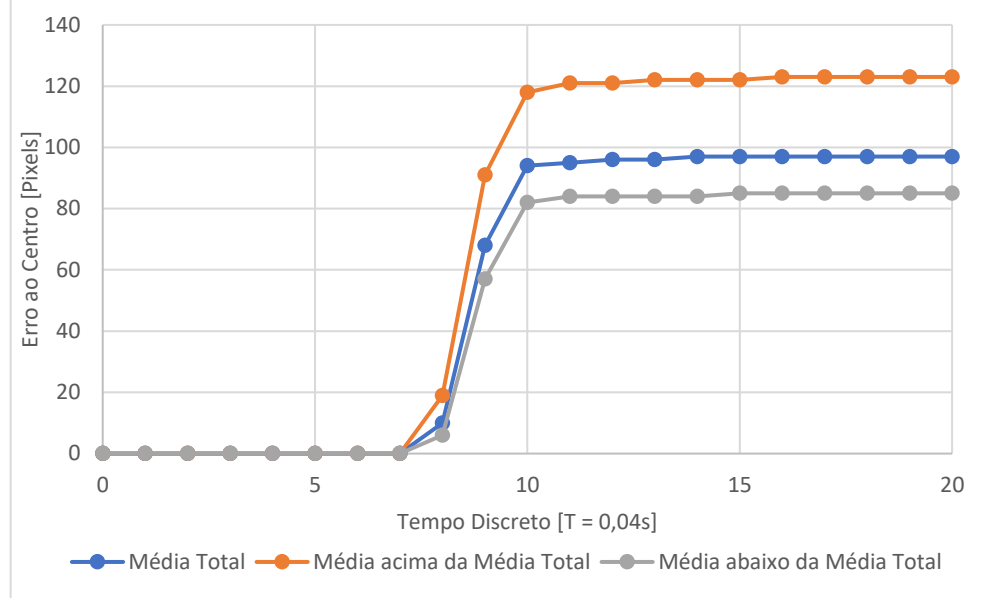
Figura 25 - Médias dos ensaios do eixo tilt com 0,3 metro de distância entre a câmera e o objeto.



Fonte: O autor.

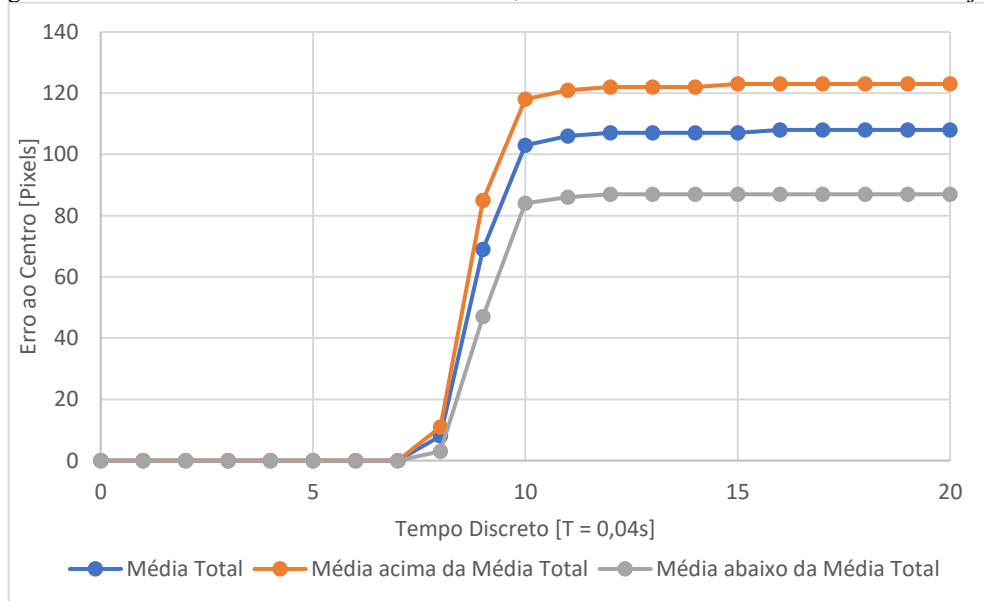


Figura 26 - Médias dos ensaios do eixo tilt com 0,6 metro de distância entre a câmera e o objeto.



Fonte: O autor.

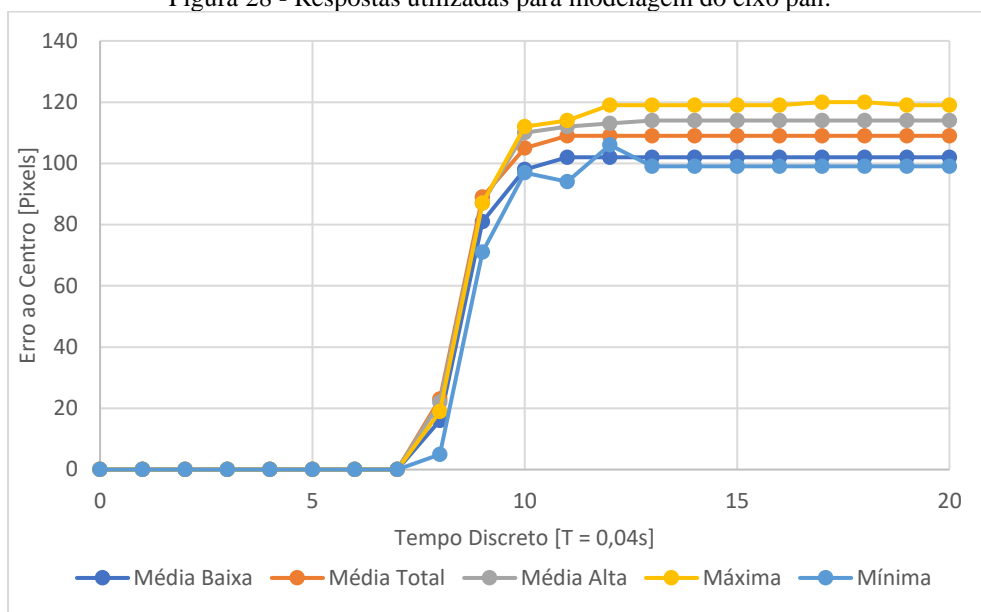
Figura 27 - Médias dos ensaios do eixo tilt com 1,2 metro de distância entre a câmera e o objeto.



Fonte: O autor.

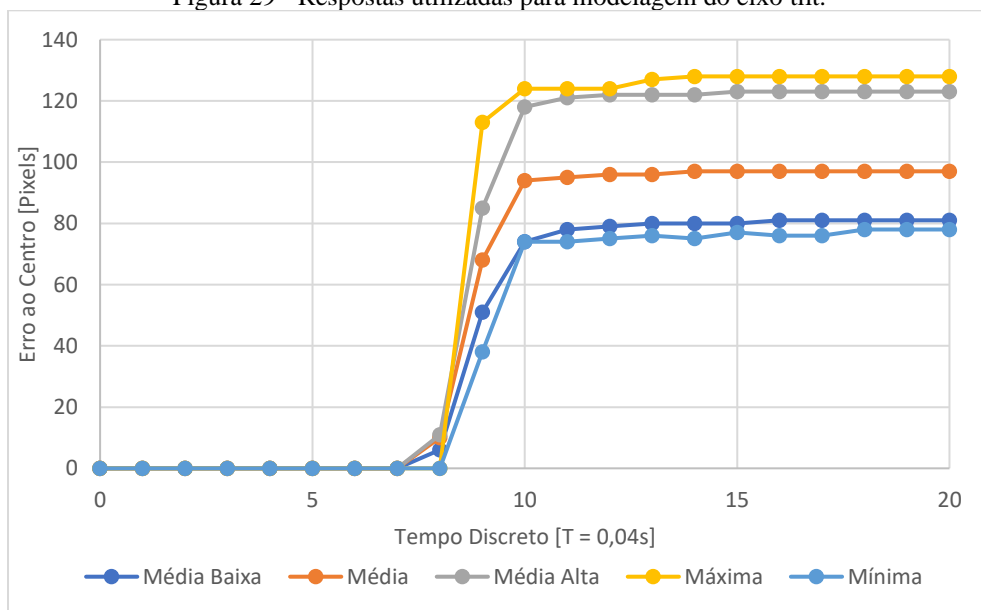
Assim, seguindo a Tabela 5 do tópico 3.3.1 deste documento, as respostas utilizadas para a modelagem do sistema do eixo pan estão representadas na Figura 28, e para o eixo tilt estão representadas na figura 29. No gráfico, o eixo das abscissas representa o tempo discreto e o eixo das ordenadas o erro ao centro da imagem em pixels. Pode-se notar que o eixo tilt apresenta uma não-linearidade maior que o eixo pan.

Figura 28 - Respostas utilizadas para modelagem do eixo pan.



Fonte: O autor.

Figura 29 - Respostas utilizadas para modelagem do eixo tilt.



Fonte: O autor.

#### 4.3.2 Identificação do Sistema

Os modelos foram obtidos utilizando a ferramenta *Ident* do software Matlab. Esta utiliza as repostas fornecidas juntamente ao sinal de entrada dos ensaios para realizar uma aproximação pelo modelo ARX utilizando o método dos Mínimos Quadrados.

A quantidade de polos e zeros de cada modelo foi definida experimentalmente.

As Tabelas 6 e 7 apresentam as funções de transferência resultantes da identificação dos sistemas dos eixos pan e tilt, respectivamente.

Tabela 6 - Funções de transferência identificadas para o sistema do eixo pan.

<b>Resposta Utilizada</b>	<b>Função de Transferência (Eixo Pan)</b>
<b>Média Baixa</b>	$\frac{3,48 \cdot 10^{-6} z^2 + 1,6 z + 6,079}{z^3 - 0,2635 z^2 + 0,007111 z + 0,009379}$
<b>Média Alta</b>	$\frac{0,0003065 z^2 + 2,198 z + 5,651}{z^3 - 0,3892 z^2 + 0,1132 z - 0,03574}$
<b>Média Total</b>	$\frac{-1,216 \cdot 10^{-6} z^2 + 2,3 z + 6,046}{z^3 - 0,2409 z^2 - 0,004502 z + 0,01131}$
<b>Máxima</b>	$\frac{0,001266 z^2 + 1,892 z + 6,116}{z^3 - 0,3787 z^2 + 0,0893 z - 0,04015}$
<b>Mínima</b>	$\frac{0,003993 z^3 + 0,4917 z^2 + 6,77 z + 4,861}{z^4 + 0,3475 z^3 - 0,08834 z^2 - 0,1231 z + 0,08894}$

Fonte: O autor.

Tabela 7 - Funções de transferência identificadas para o sistema do eixo tilt.

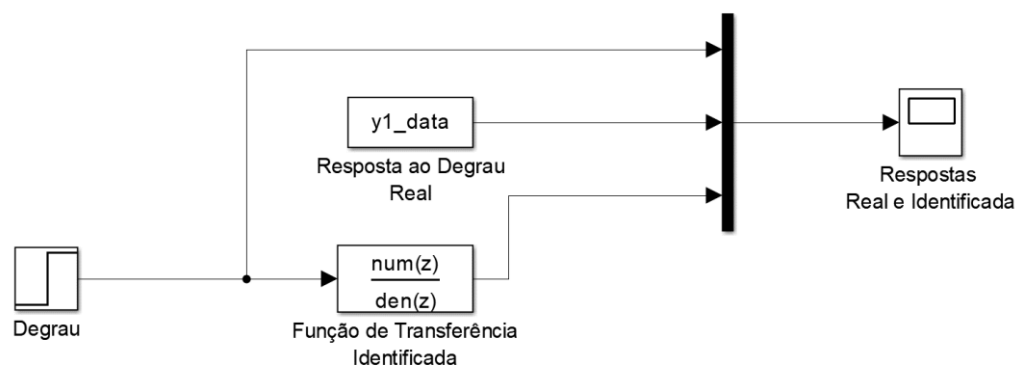
<b>Resposta Utilizada</b>	<b>Função de Transferência (Eixo Tilt)</b>
<b>Média Baixa</b>	$\frac{-0,0001849 z^2 + 0,601 z + 4,177}{z^3 - 0,5356 z^2 + 0,1912 z - 0,06649}$
<b>Média Alta</b>	$\frac{0,0003743 z^2 + 1,097 z + 6,9}{z^3 - 0,4647 z^2 + 0,1618 z - 0,04697}$
<b>Média Total</b>	$\frac{0,001112 z^2 + 0,9936 z + 5,355}{z^3 - 0,469 z^2 + 0,182 z - 0,05863}$
<b>Máxima</b>	$\frac{-0,0002423 z^2 + 0,002128 z + 11,28}{z^3 - 0,1 z^2 + 0,01552 z - 0,03223}$
<b>Mínima</b>	$\frac{3,58 \cdot 10^{-5} z^3 + 0,0002902 z^2 + 3,8 z + 3,669}{z^4 + 0,02132 z^3 - 0,0228 z^2 - 0,001471 z - 0,02928}$

Fonte: O autor.

#### 4.3.3 Validação do Modelo

A validação acontece na ferramenta *Simulink* do software Matlab. Nesta, um degrau é aplicado a função de transferência obtida para o sistema e sua resposta é comparada com os dados reais. A montagem desta aplicação no Simulink está apresentada na Figura 30.

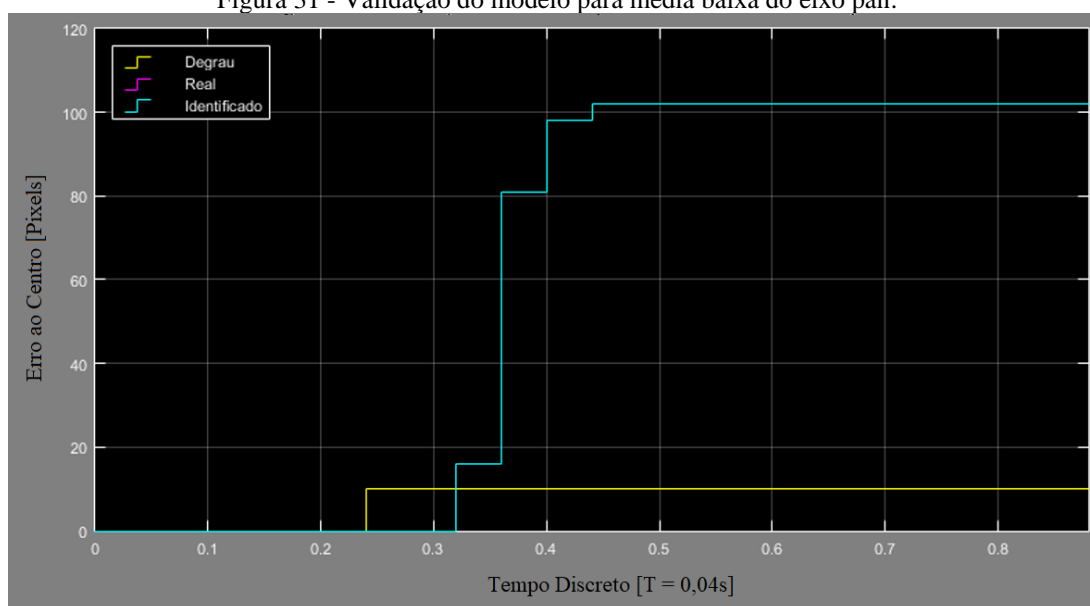
Figura 30 - Aplicação no Simulink para validação do modelo.



Fonte: O autor.

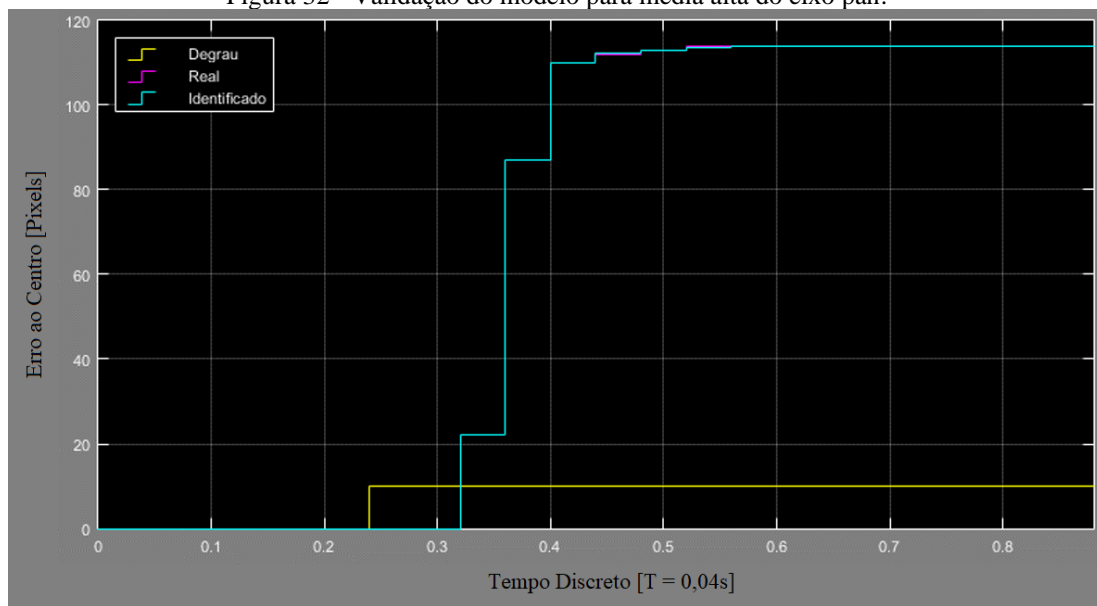
As Figuras 31, 32, 33,34 e 35 representam a validação dos modelos para o eixo pan. No gráfico, o eixo das abscissas representa o tempo discreto e o eixo das ordenadas o erro ao centro da imagem em pixels.

Figura 31 - Validação do modelo para média baixa do eixo pan.



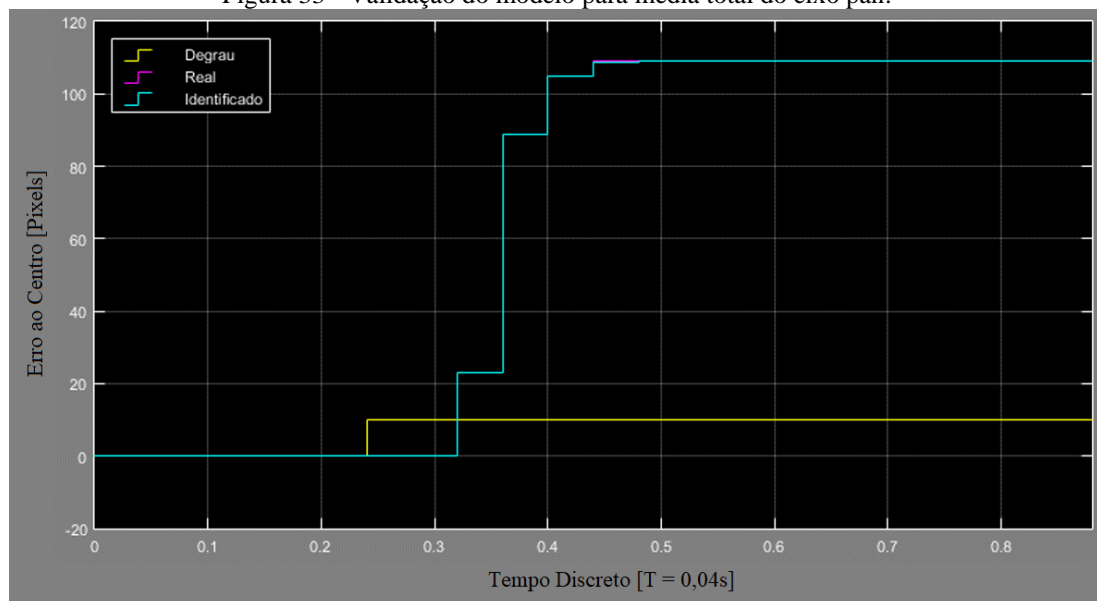
Fonte: O autor.

Figura 32 - Validação do modelo para média alta do eixo pan.



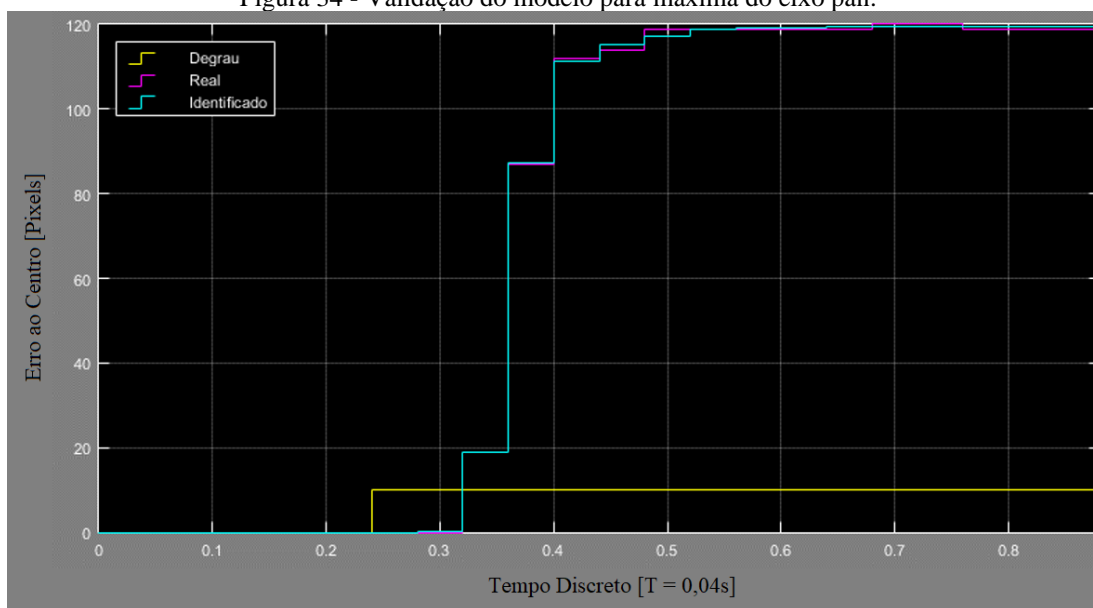
Fonte: O autor.

Figura 33 - Validação do modelo para média total do eixo pan.



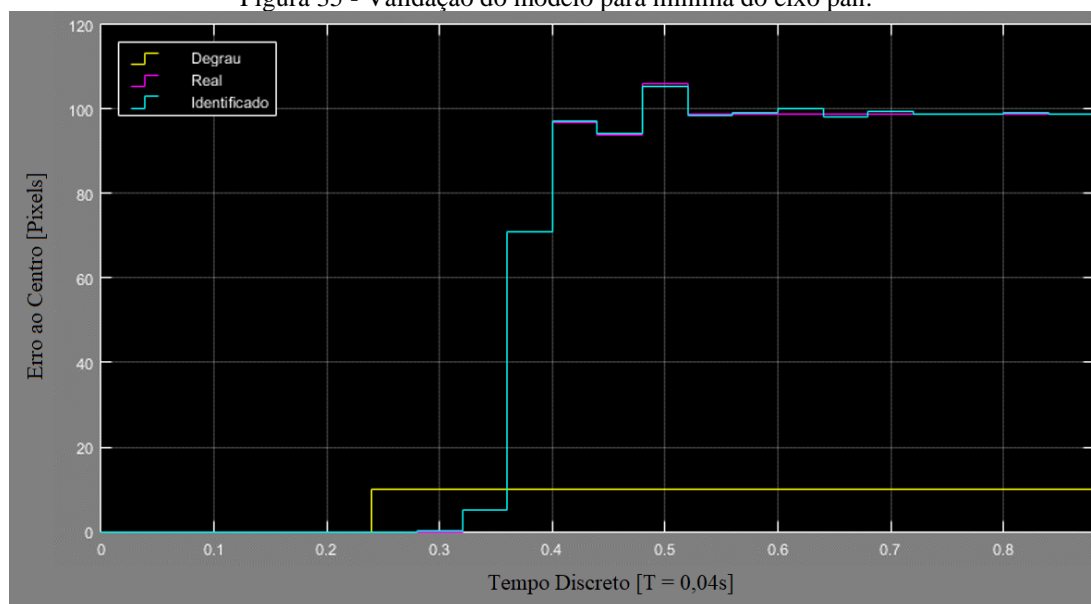
Fonte: O autor.

Figura 34 - Validação do modelo para máxima do eixo pan.



Fonte: O autor.

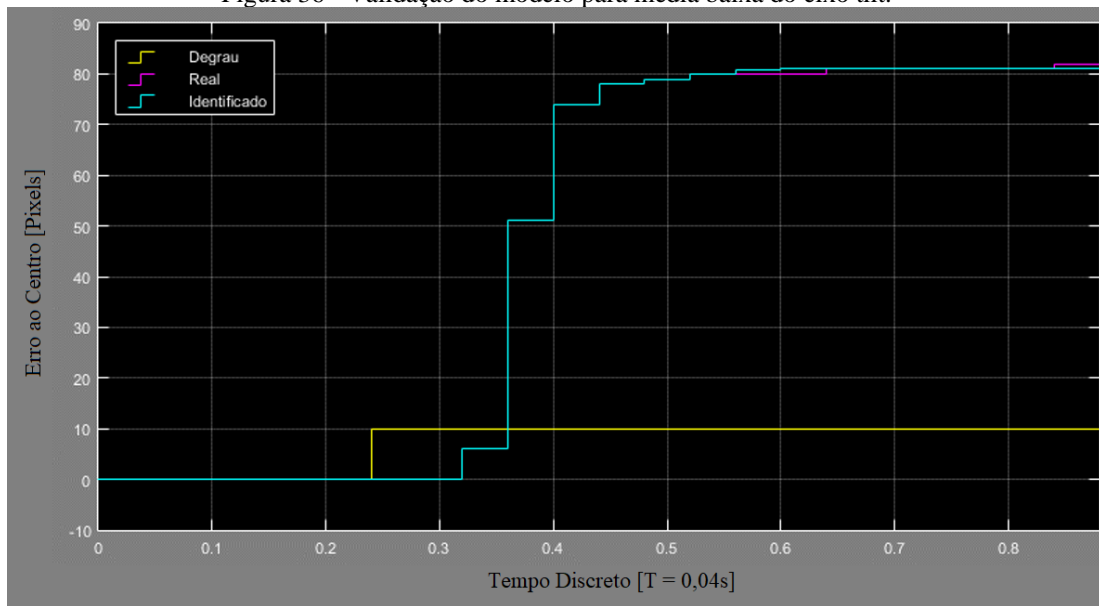
Figura 35 - Validação do modelo para mínima do eixo pan.



Fonte: O autor.

As Figuras 36, 37, 38, 39 e 40 representam a validação dos modelos para o eixo tilt. No gráfico, o eixo das abscissas representa o tempo discreto e o eixo das ordenadas o erro ao centro da imagem em pixels.

Figura 36 - Validação do modelo para média baixa do eixo tilt.



Fonte: O autor.

Figura 37 - Validação do modelo para média alta do eixo tilt.

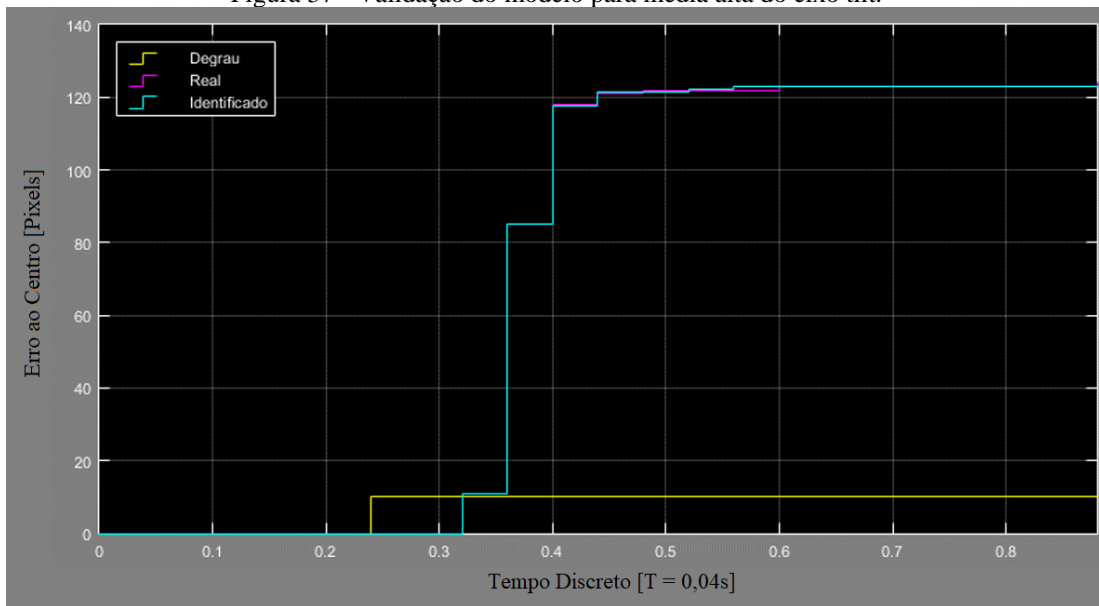
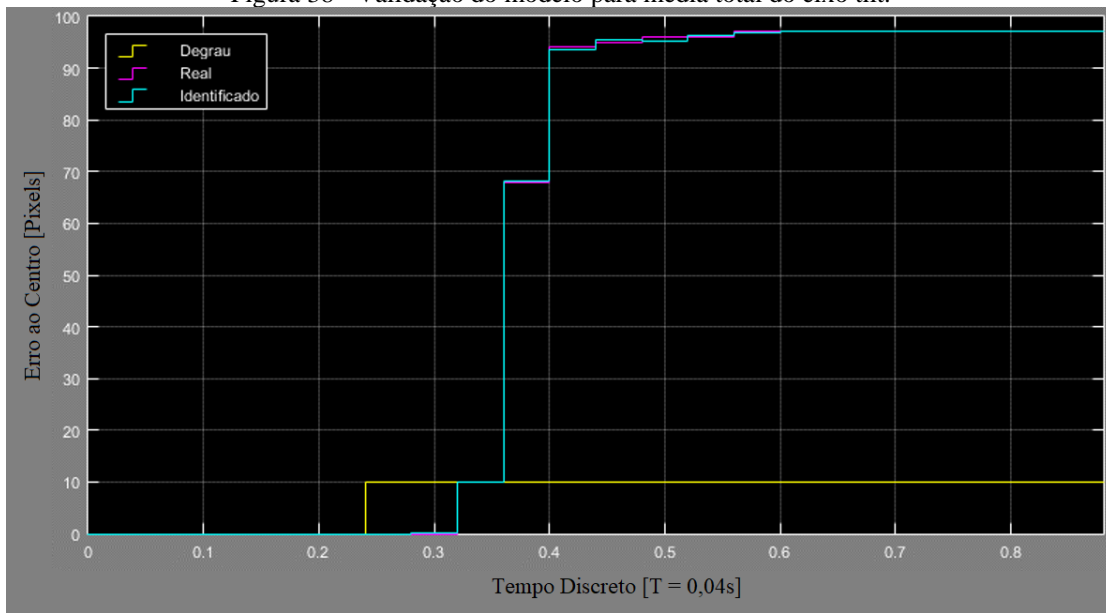
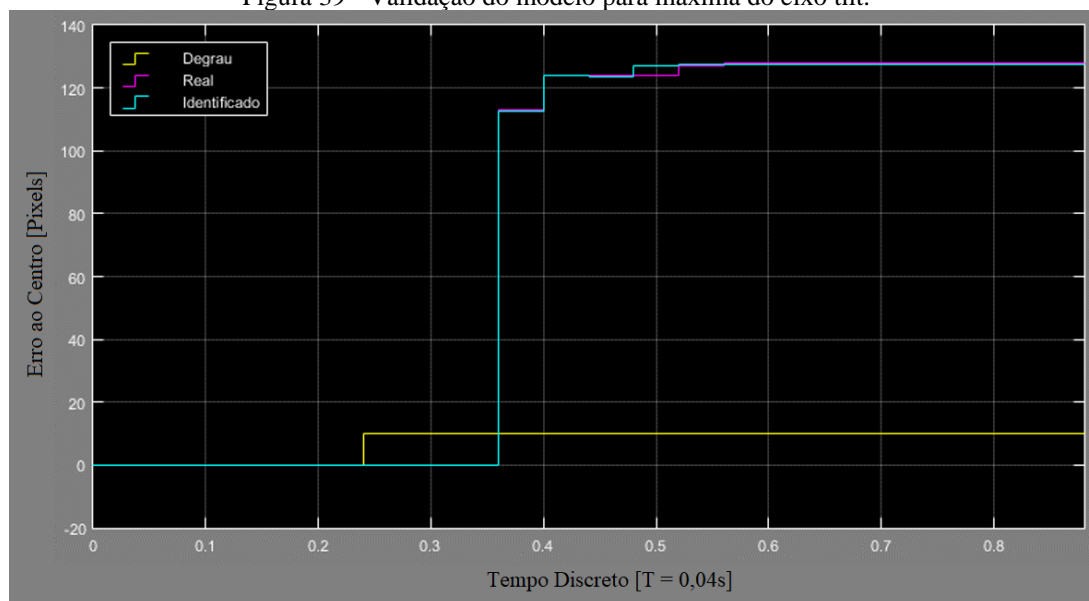


Figura 38 - Validação do modelo para média total do eixo tilt.



Fonte: O autor.

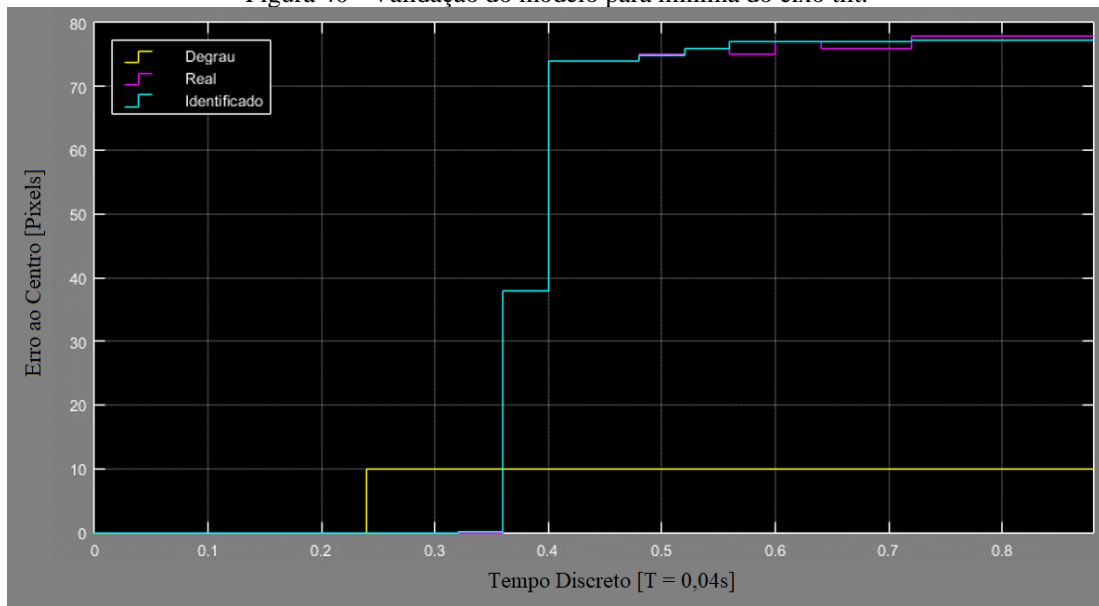
Figura 39 - Validação do modelo para máxima do eixo tilt.



Fonte: O autor.



Figura 40 - Validação do modelo para mínima do eixo tilt.



Fonte: O autor.

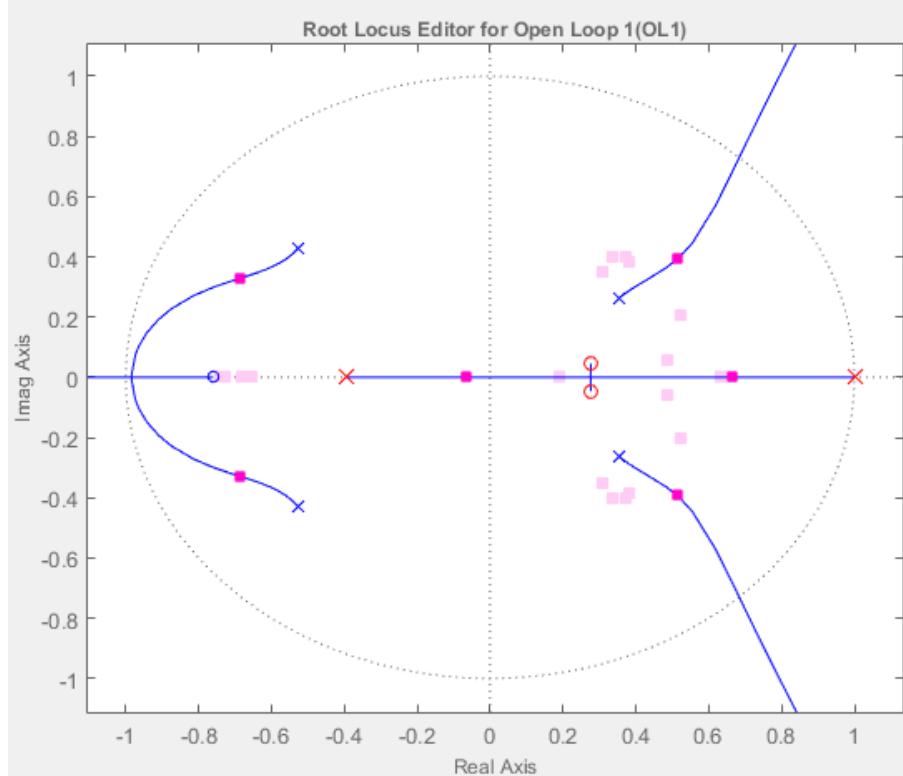
#### 4.3.4 Projeto do Controlador

O controlador foi projetado utilizando a ferramenta *rltool* do software Matlab. Essa ferramenta possui uma interface gráfica que permite a adição, movimentação e remoção de polos e zeros do controlador no Lugar Geométrico das Raízes, mostrando em tempo real a resposta do sistema em malha fechada para cada ganho deste mesmo controlador.

Para este projeto, os cinco modelos identificados (Média Total, Média Alta, Média Baixa, Máxima e Mínima) foram agrupados de modo que a ferramenta *rltool* mostre simultaneamente a resposta dos cinco modelos para o controlador projetado. Isso é feito no Matlab utilizando a função *stack*. Na resposta ao degrau apresentada pelo *rltool* foi utilizada a configuração *bounds*, que mostra uma linha mais grossa na resposta do sistema, representando a variação entre os modelos agrupados para esta malha de controle.

A Figura 41 representa o Lugar Geométrico das Raízes do sistema projetado para o eixo pan, mostrando apenas a área onde se encontram os polos me malha fechada do sistema.

Figura 41 - Polos no Lugar Geométrico das Raízes do sistema projetado para o eixo pan.



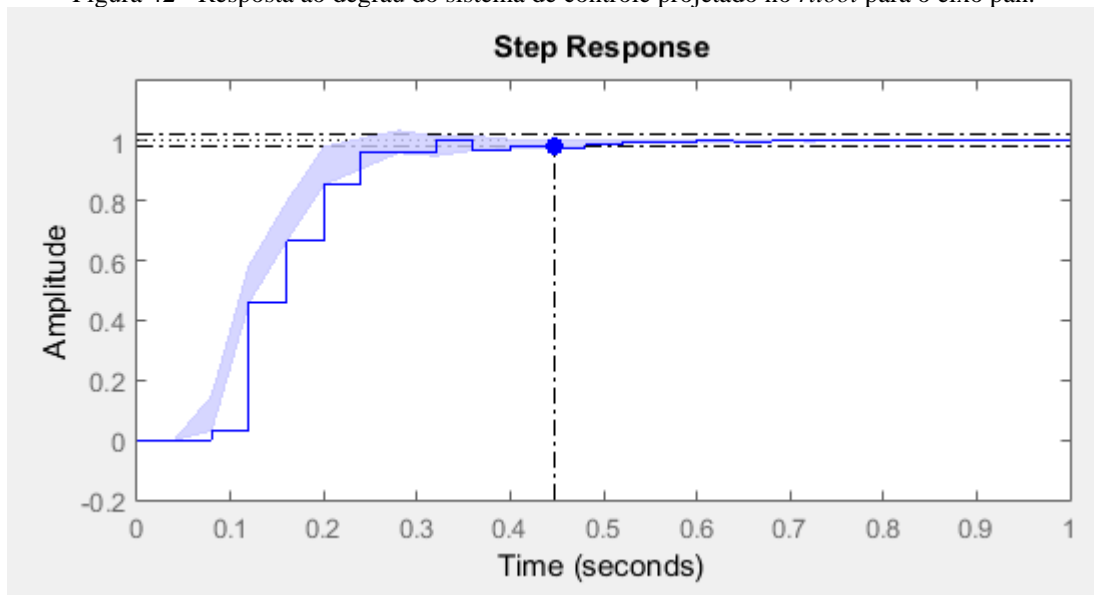
Fonte: O autor.

No projeto, um polo integrador foi adicionado para garantir o erro de posição nulo. Um par de zero complexos foi adicionado para estabilizar o sistema, pois somente com o integrador o sistema seria instável. Com isso, mais um polo real foi adicionado para que o controlador seja causal. A posição do polo real e do par de zeros complexos adicionados assim como o ganho do controlador foram ajustados manualmente de acordo com a resposta do sistema em malha fechada.

Os pontos sombreados no LGR representam a variabilidade do sistema obtida através do conjunto de respostas agrupados na função *Stack*. Estes garantem que o sistema, em todos os cenários, possui todos polos em malha fechada dentro do disco unitário. Isso garante a estabilidade robusta do sistema de controle desenvolvido.

A resposta ao degrau no *rltool* apresenta tempo de acomodação inferior a 0,5 segundo e baixo sobressinal. Essa resposta é apresentada na Figura 42.

Figura 42 - Resposta ao degrau do sistema de controle projetado no *rltool* para o eixo pan.



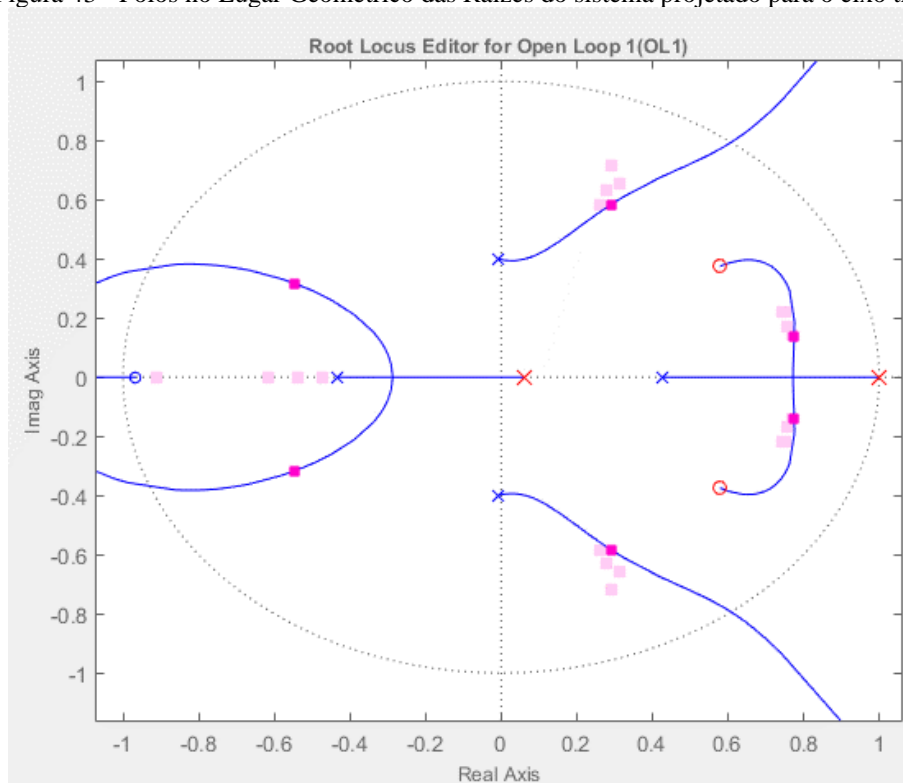
Fonte: O autor.

Assim, para o eixo pan, o controlador projetado possui a função de transferência apresentada na Eq. (11):

$$C(z) = \frac{0,06455 z^2 - 0,03577 z + 0,005091}{z^2 - 0,6037 z - 0,3963} \quad (11)$$

Para o eixo tilt, a Figura 43 representa o Lugar Geométrico das Raízes do sistema projetado, mostrando apenas a área onde se encontram os polos me malha fechada do sistema.

Figura 43 - Polos no Lugar Geométrico das Raízes do sistema projetado para o eixo tilt.



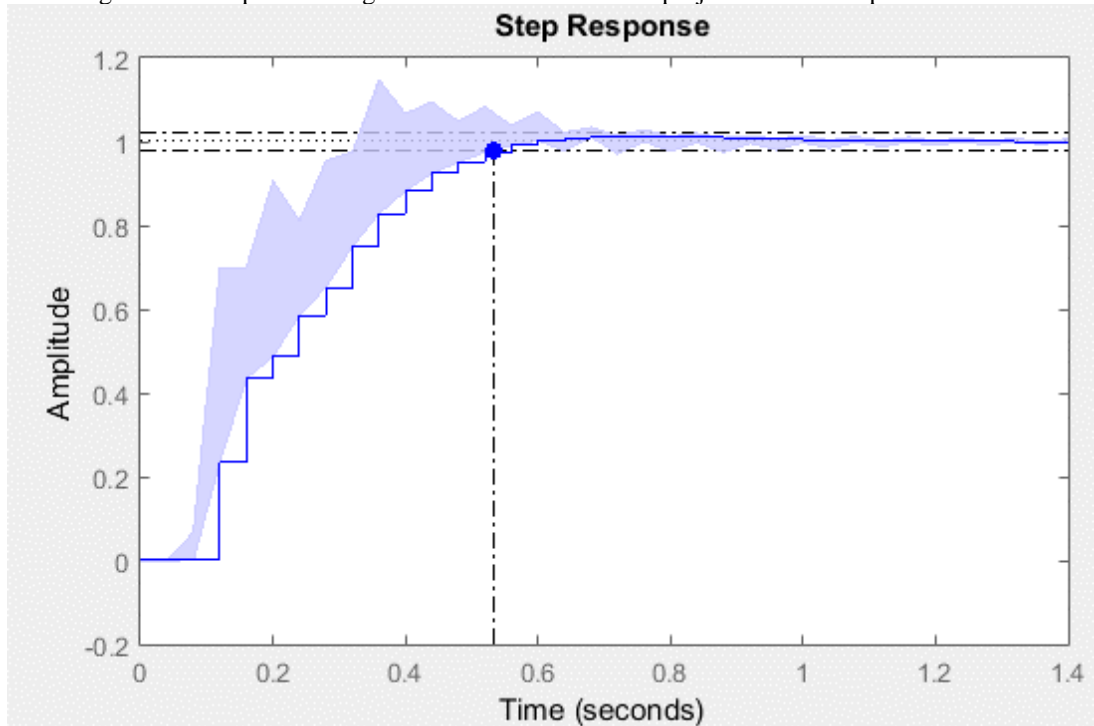
Fonte: O autor.

Este também possui um polo integrador, um polo real e um par de zeros complexos pelos mesmos motivos do eixo pan. Nota-se que os zeros do controlador projetado para o eixo tilt estão mais afastados do eixo real que no controlador do eixo pan, isso foi necessário pois o sistema em malha aberta apresenta um comportamento mais oscilatório que no caso anterior.

Nota-se também a presença de pontos sombreados mais próximos do limite do disco unitário. O motivo disto é que o sistema do eixo tilt apresenta maior variabilidade, apresentando maior variação nas repostas em malha fechada para os diferentes modelos obtidos. Para garantir a estabilidade robusta deste sistema, optou-se por aceitar um tempo de acomodação maior e mais oscilação na resposta ao degrau.

A resposta ao degrau no *rltool* apresenta tempo de acomodação inferior a 0,6 segundo e baixo sobressinal, porém para alguns modelos a resposta apresentou um sobressinal mais significativo. Isso acontece porque o eixo tilt apresenta mais folga no seu motor e assim a não-linearidade do sistema se torna mais influente na hora de projetar o controlador. Na prática o impacto não será tão visível porque, na média, as respostas apresentam um nível de sobressinal aceitável. Essa resposta é apresentada na Figura 44.

Figura 44 - Resposta ao degrau do sistema de controle projetado no *rltool* para o eixo tilt.



Fonte: O autor.

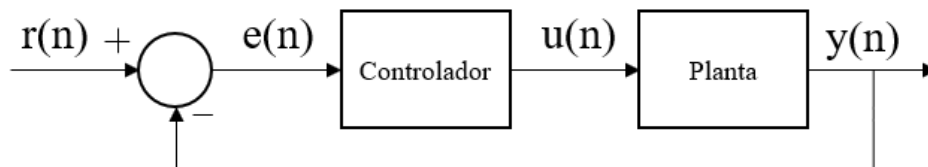
Para o eixo tilt, o controlador projetado é apresentado na Eq. (12):

$$C(z) = \frac{0,06183 z^2 - 0,07155 z + 0,02935}{z^2 - 1,061 z + 0,06056} \quad (12)$$

#### 4.3.5 Implementação e Validação do Sistema de Controle

Para implementação do controle em malha fechada, é realizada a transformada inversa das funções de transferência dos controladores. A Figura 45 apresenta as variáveis do sistema de controle em malha fechada no domínio no tempo discreto.

Figura 45 - Sistema de controle em malha fechada no domínio do tempo discreto.



Fonte: O autor.

Para o eixo pan, a transformada inversa da Eq. (11) é apresentada na Eq. (13):

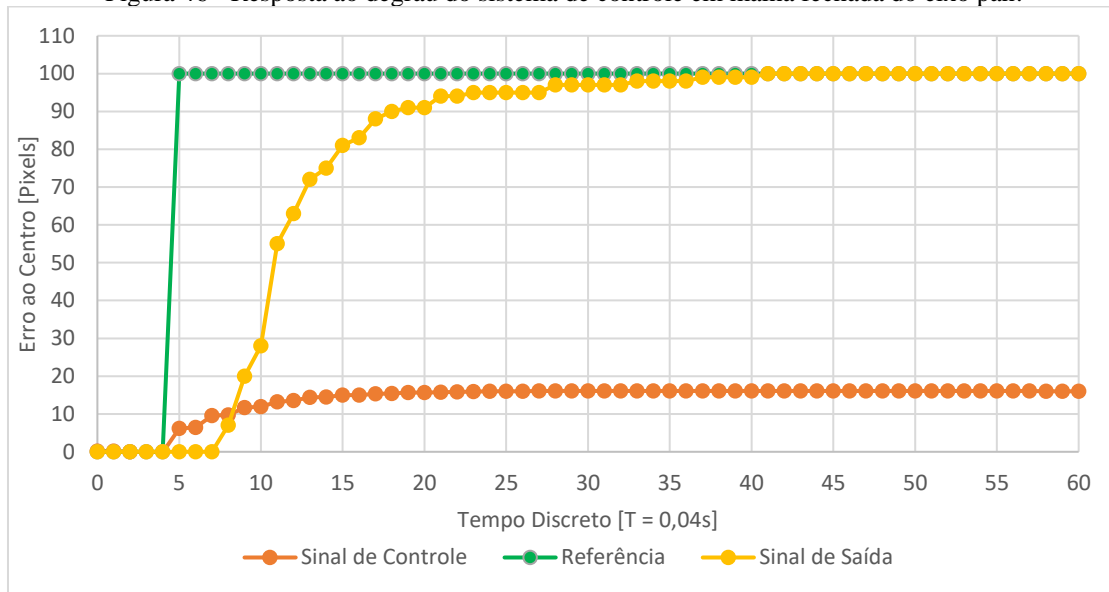
$$u(n) = 0,0645574 e(n) - 0,03577e(n-1) + 0,005091e(n-2) + 0,6037u(n-1) + 0,3963u(n-2) \quad (13)$$

Para o eixo tilt, a transformada inversa da Eq. (12) é apresentada na Eq. (14):

$$u(n) = 0,06183 e(n) - 0,07155e(n - 1) + 0,02935e(n - 2) + 1,061u(n - 1) - 0,06056u(n - 2) \quad (14)$$

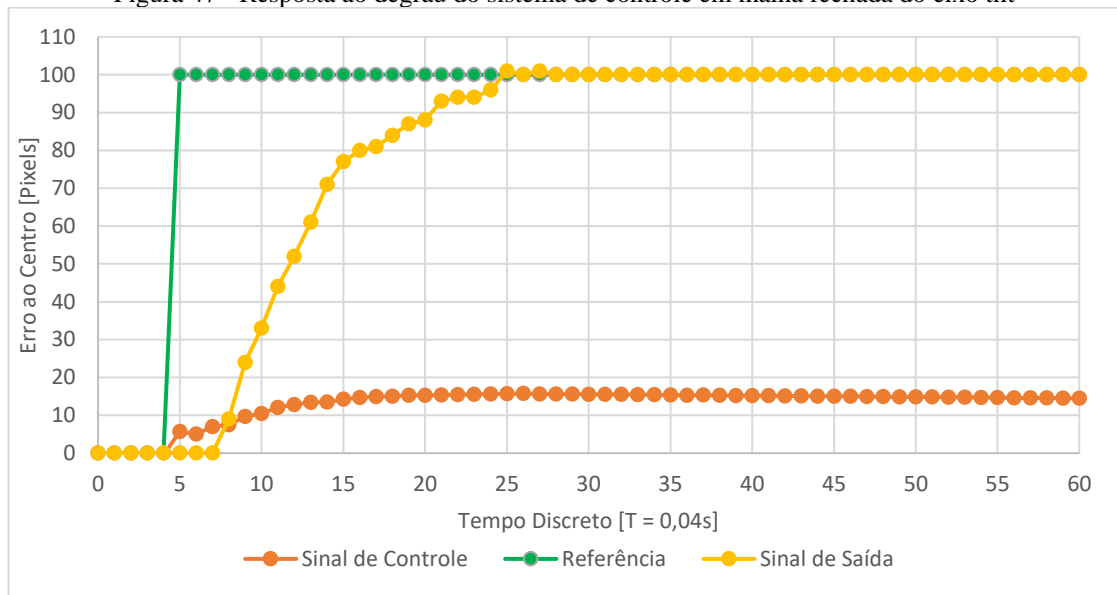
Assim as respostas ao degrau para os sistemas de controle em malha fechada dos eixos pan e til estão representadas nas Figuras 46 e 47, respectivamente. O degrau aplicado é de 100 unidades para melhor visualização dos resultados. No gráfico, o eixo das abscissas representa o tempo discreto e o eixo das ordenadas o erro ao centro da imagem em pixels.

Figura 46 - Resposta ao degrau do sistema de controle em malha fechada do eixo pan.



Fonte: O autor.

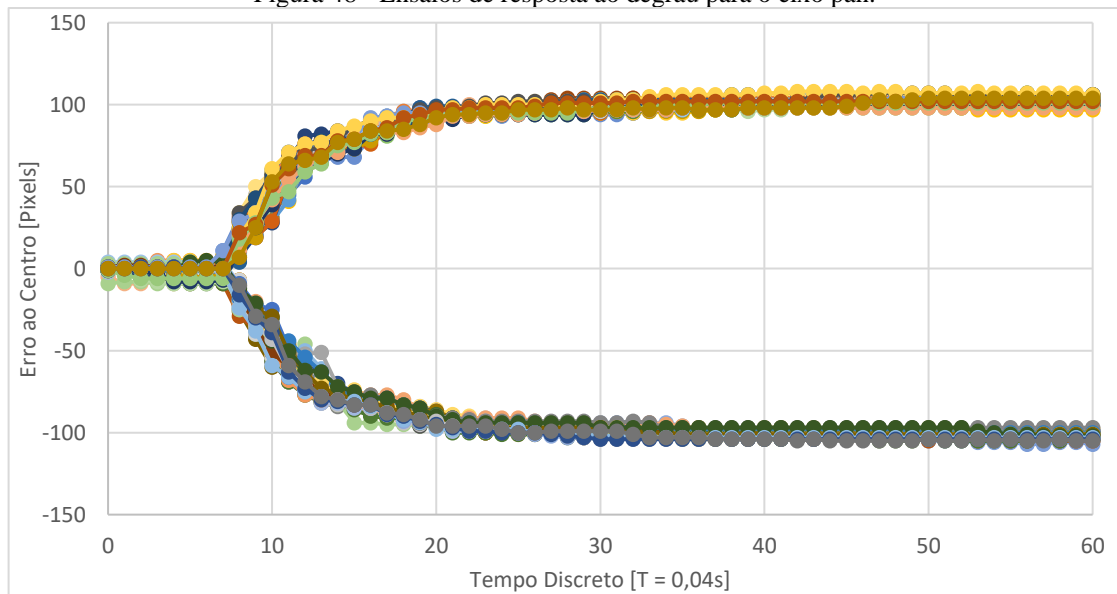
Figura 47 - Resposta ao degrau do sistema de controle em malha fechada do eixo tilt



Fonte: O autor.

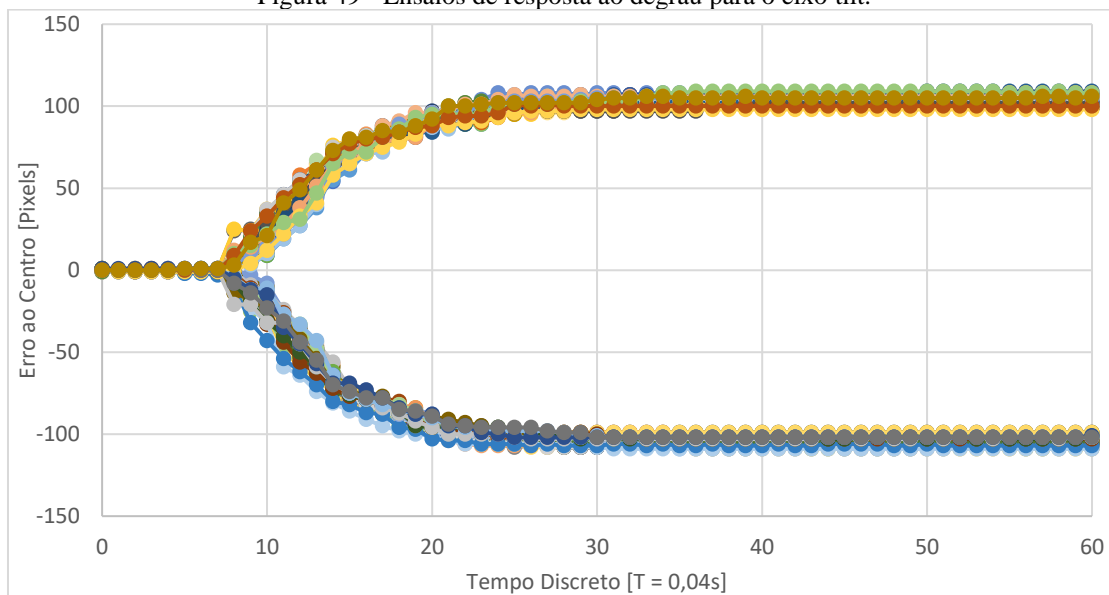
Para avaliar a robustez do sistema de controle desenvolvido, foram realizados ensaios de resposta ao degrau do sistema em malha fechada, de acordo com a sequência no tópico 3.2.3 deste documento. Os resultados estão apresentados na Figura 48 para o eixo pan e na Figura 49 para o eixo tilt.

Figura 48 - Ensaios de resposta ao degrau para o eixo pan.



Fonte: O autor.

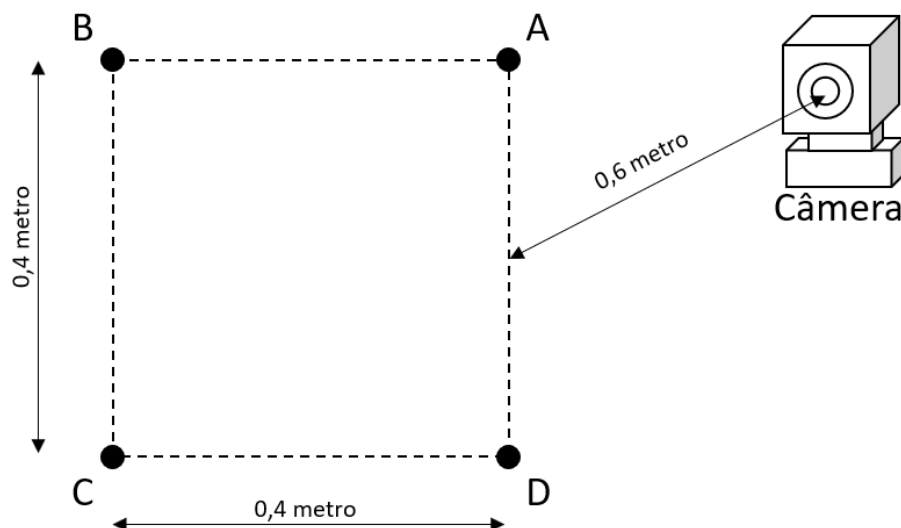
Figura 49 - Ensaios de resposta ao degrau para o eixo tilt.



Fonte: O autor.

Para avaliar o desempenho do controlador em regime regulatório, o objeto rastreado foi posicionado a 0,6 metro da câmera e o objeto realizou uma trajetória quadrada paralela a câmera com retas de aproximadamente 0,4 metro de comprimento. A Figura 50 ilustra essa trajetória.

Figura 50 - Trajetória do ensaio em regime regulatório

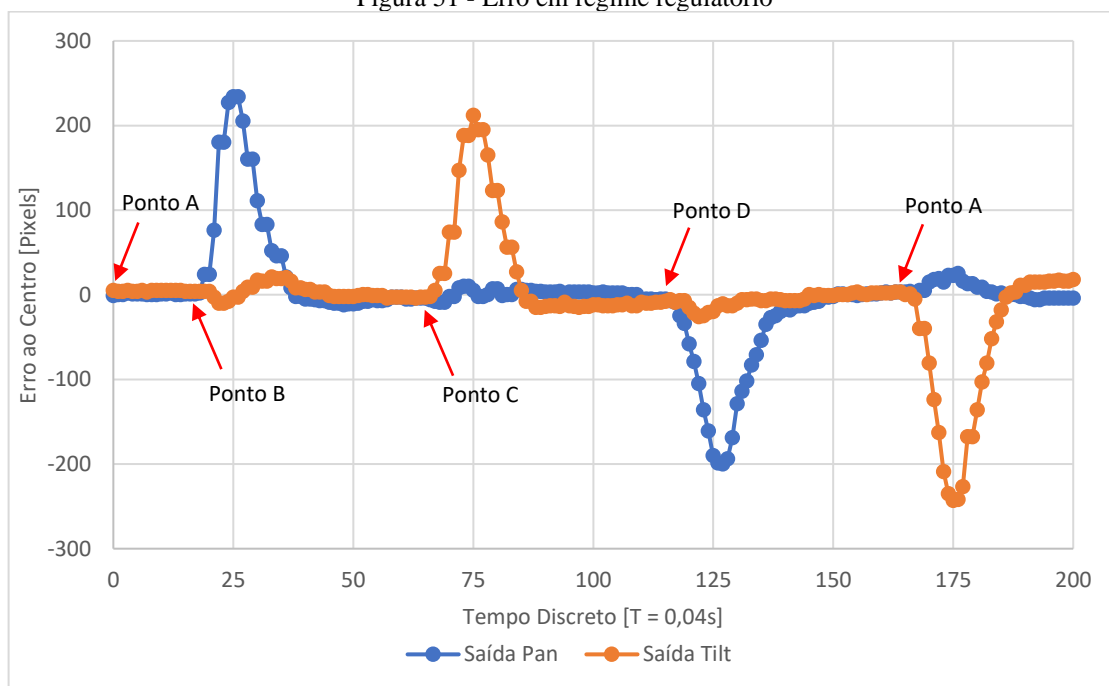


Fonte: O autor.

O erro com relação a origem para este ensaio é apresentado na Figura 51, que também indica o instante no tempo discreto em que o objeto rastreado altera sua posição para cada ponto. No gráfico, o eixo das abscissas representa o tempo discreto e o eixo das ordenadas o erro ao centro da imagem em pixels.



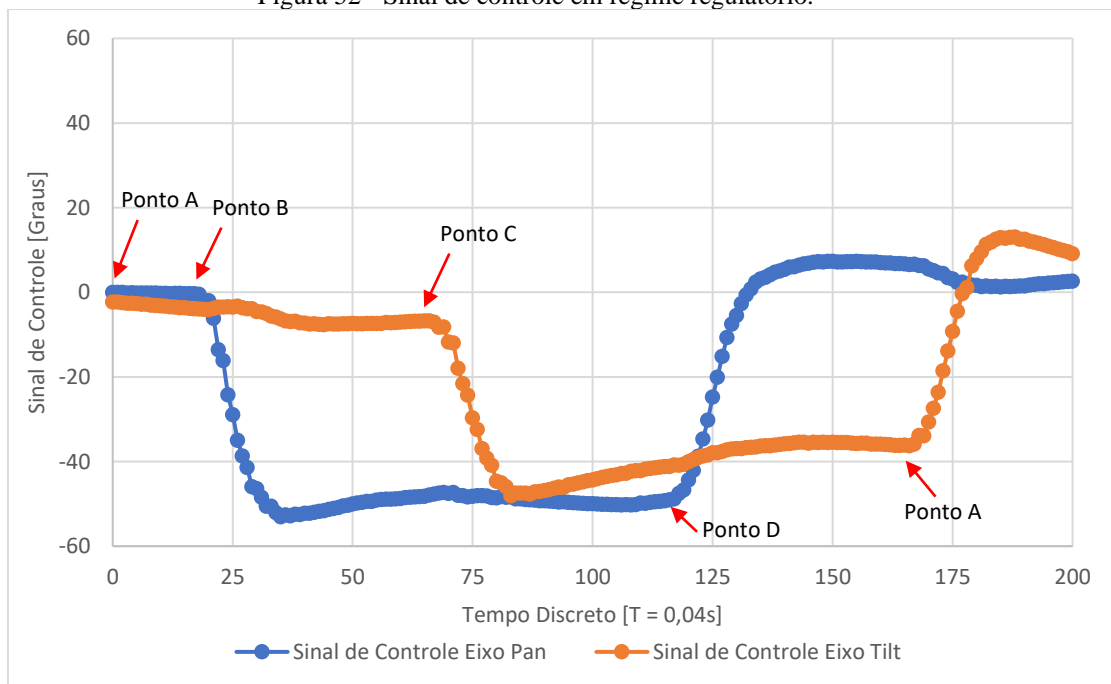
Figura 51 - Erro em regime regulatório



Fonte: O autor.

A Figura 52 apresenta o sinal de controle durante este ensaio. No gráfico, o eixo das abscissas representa o tempo discreto e o eixo das ordenadas representa a posição de cada atuador, em graus.

Figura 52 - Sinal de controle em regime regulatório.



Fonte: O autor.

Demonstrações do funcionamento deste projeto podem ser visualizados no vídeo disponível no endereço: <https://youtu.be/1qF-k6oR0NE>.

## 5 CONCLUSÃO

Este trabalho teve como objetivo o desenvolvimento de um sistema de controle em malha fechada para uma câmera pan-tilt. Para isso foi necessário o desenvolvimento do software de rastreamento de objetos, a aplicação de métodos de identificação de sistema para obtenção de modelos matemáticos e o projeto do controlador utilizado. Todos estes conteúdos são abordados durante o curso de Engenharia de Controle e Automação, o que tornou o projeto muito interessante pois possibilitou ao aluno pôr em prática diversas habilidades adquiridas e exercitadas durante a graduação.

O sistema desenvolvido neste trabalho apresentou desempenho satisfatório e robusto para rejeição de distúrbios de posição. As ferramentas de software desenvolvidas atendem as necessidades da aplicação e são de uso intuitivo e multiplataforma.

Todos os arquivos referentes a este trabalho, incluindo as ferramentas de software desenvolvidas, podem ser consultados através do endereço <https://github.com/Lucas-puc/TCC>.

Para trabalhos futuros sugere-se a implementação do sistema de modo embarcado, onde o processamento de imagem e o comando dos servo-motores são feitos pela mesma plataforma, pois isso tornaria o sistema mais rápido assim permitindo a aplicação de um sistema de controle com tempo de acomodação menor. Sugere-se também a implementação de um sistema com ação integral dupla para rejeitar distúrbios de velocidade e de estratégias *anti-windup* para melhorar o desempenho durante a saturação dos atuadores.

## 6 REFERÊNCIAS BIBLIOGRÁFICAS

AGUIRRE, LUIS ANTONIO. Introdução À Identificação de Sistemas, 2. ed. Editora UFMG, 2007.z

AZEVEDO, DANIELA DE ARAUJO. Controle de orientação de uma câmera pan-tilt por servovisão baseada em imagem com realimentação de saída. 2017. Dissertação (Mestrado em Engenharia Elétrica) - Centro de Educação Tecnológica Celso Suckow da Fonseca, Rio de Janeiro, 2017.

FILAFILL. Disponível em <[https://filafill.com/en/electronics/268-cammera-pan-tilt-support.html?search\\_query=pan+tilt&results=43](https://filafill.com/en/electronics/268-cammera-pan-tilt-support.html?search_query=pan+tilt&results=43)>. Acesso em junho de 2020.

FUSCO, R. J.; COELHO, A. D.; ASSIM, W. O. Desenvolvimento de um sistema de visão para robô humanoide. **Anais do 6º Seminário Mauá de Iniciação Científica**. São Paulo: Instituto Mauá de Tecnologia, 2014.

GONZALEZ, R. C.; WOODS, R. E. Processamento digital de imagens. 3. ed. São Paulo: Pearson Prentice Hall, 2010.

KIKUCHI, DAVI YOSHINOBU. Sistema de controle servo visual de uma câmera pan-tilt com rastreamento de uma região de referência. 2007. Dissertação (Mestrado em Engenharia de Controle e Automação Mecânica) - Escola Politécnica, Universidade de São Paulo, São Paulo, 2007.

LJUNG, LENNART. System Identification: Theory for the User. 2. ed. Prentice Hall, 1999.

LOGITECH. Disponível em: <<https://www.logitech.com/pt-br/product/hd-webcam-c270>>. Acesso em: 10 de junho de 2020.

MENDONÇA, LUÍS PAULO CARVALHO DE. Sistema automático de rastreamento dinâmico de alvos para câmera. 2014. Monografia (Graduação em Engenharia de Controle e Automação) - Escola de Engenharia, Universidade Federal de Minas Gerais, Belo Horizonte, 2014.

RIDEL, DANIELA ALVES. Detecção e rastreamento de obstáculos em ambientes urbanos utilizando visão estéreo. 2016. Dissertação (Mestrado em Ciências de Computação e Matemática Computacional) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Paulo, 2016.

SILVA, ANDRÉ BRAZ DA; SANTOS, LEONARDO CAETANO DOS. Controle de dispositivos robóticos usando visão computacional. Trabalho de conclusão de curso (Tecnologia em Análise e desenvolvimento de Sistemas) - Instituto Federal de São Paulo, Campus Caraguatatuba, 2014.

TOWERPRO. Disponível em: <<http://www.towerpro.com.tw/product/mg90s-3/>>. Acesso em: 10 de junho de 2020.