SISTEMA DE RASTREAMENTO DE OBJETOS ATRAVÉS DO CONTROLE SERVO VISUAL DE UMA CÂMERA COM ESTRUTURA PAN-TILT

Douglas Hansen, Rodrigo Trentini
Instituto Federal de Santa Catarina
Câmpus Jaraguá do Sul – Rau – Curso de Bacharelado em Engenharia Elétrica
e-mail: douglas.h5@aluno.ifsc.edu.br, rodrigo.trentini@ifsc.edu.br
Trabalho de Conclusão de Curso – 11/02/2022

***Resumo* – O controle servo visual e a visão computacional são áreas de estudo que vem ganhando espaço em aplicações como sistemas robóticos e de segurança. Desse modo, o problema de rastrear automaticamente um objeto em movimento se tornou um tópico de interesse, sendo alvo de diversos estudos. Diante disto, este trabalho tem como objetivo o desenvolvimento de um sistema de rastreamento de objetos através de uma câmera e uma estrutura Pan-Tilt. Para isso, é realizada a modelagem do sistema através das relações geométricas entre projeção da câmera e objeto rastreado e determinada a dinâmica dos atuadores da estrutura, bem como realiza-se o projeto de controladores PD com ganho estático e escalonamento de ganhos através do método do Lugar Geométrico das Raízes. Os resultados experimentais mostram um nível de eficiência satisfatório dos sistemas estabelecidos através dos métodos adotados.**

***Palavras-chave* – Controle servo visual, visão computacional, Pan-Tilt, Lugar Geométrico das Raízes, escalonamento de ganhos**

OBJECT TRACKING SYSTEM THROUGH VISUAL SERVOING OF A CAMERA WITH PAN-TILT STRUCTURE

***Abstract* – The visual servoing and computer vision are study areas that have been gaining ground in applications such as robotics and security systems. Thus, the problem of automatically tracking a moving object has become a topic of interest and the subject of several studies. Therefore, this paper presents the development of an object tracking system through a camera and a Pan-Tilt structure. For this purpose, the system is modeled through the geometric relationships between camera projection and tracked object and the dynamics of the structure's actuators are also determined, as well as the design of PD controllers with static gain and gain scheduling through the Root Locus method. The experimental results show a satisfactory level of efficiency of the established systems through the adopted methods**

***Keywords* – Visual servoing, computer vision, Pan-Tilt, Root Locus method, gain scheduling**

I. INTRODUÇÃO

Com o desenvolvimento da tecnologia, câmeras e microfones foram introduzidos como meios de aquisição de dados, podendo ser encontrados em equipamentos utilizados no cotidiano da grande maioria das pessoas, como computadores e dispositivos móveis. Esses dispositivos permitem o uso e processamento dos dados obtidos de maneira significativa, principalmente através de imagens digitais.

Uma das vantagens da utilização de câmeras em dispositivos automatizados é a possibilidade da obtenção de diversas informações, as quais são provenientes das imagens capturadas de forma simultânea e podendo ser realizada com o uso de apenas um único componente sensor [1]. Junto da rápida evolução da capacidade e velocidade de processamento de microcontroladores surgiram formas de trabalho inovadoras e assim dando origem a novos campos de estudo e pesquisa.

A visão computacional compreende uma das áreas de atuação da inteligência artificial, podendo ser definida como um campo científico onde se extrai informações de imagens digitais [1], ou ainda como um conjunto de técnicas computacionais destinadas a explicitar as propriedades geométricas e dinâmicas do mundo tridimensional a partir de imagens digitais [2].

Originando-se dessa, surgiu o termo *controle servo visual*, o qual envolve o uso de uma ou mais câmeras e um sistema de visão computacional para controlar a posição de um sistema robótico [3], sendo resultado da evolução tecnológica e junção de diversas áreas como processamento de imagens em alta velocidade, cinemática, dinâmica, teoria de controle e computação em tempo real [4].

Em geral, os primeiros sistemas visuais trabalhavam em malha aberta, onde primeiramente observa-se a posição do objeto, e posteriormente movimenta-se o atuador robótico, sendo conhecido como "*look-then-move*" [3]. Assim, estes se tornam extremamente dependentes da precisão do sensor visual e do manipulador robótico. Como alternativa buscando o aumento dessa precisão, adota-se uma malha de realimentação do sensoriamento visual [5].

Desta maneira, o controle servo visual nada mais é do que um sistema de controle em malha fechada, com os dados obtidos pelo sensor em tempo real e repassados ao controlador através do laço de realimentação [5].

Existem diversas áreas e aplicações que fazem uso de visão

computacional e controle servo visual, como em [6] onde realiza-se a inspeção de placas de circuito impresso baseando-se em características relacionadas ao contorno, posicionamento e histograma dos componentes. Em [7] apresenta-se uma inspeção da qualidade de impressão em rótulos utilizando segmentação de contornos. Controle de veículos aéreos não tripulados, como Menezes [8] traz alternativas para melhoria do desempenho da servovisão clássica, utilizando e realizando a comparação entre os controladores clássicos, de ganho variável e filtro *fuzzy* para um veículo aéreo multirrotor. Bezerra [9] realiza um comparativo via simulação entre os métodos de controle *Image Based Visual Servoing* (IBVS) e *Position Based Visual Servoing* (PBVS) para um veículo aéreo quadrirrotor.

Sistemas robóticos com dois eixos de movimentação e uma câmera são alvos iniciais de estudos sobre controle servo visual. Kikuchi [5] apresenta uma modelagem do sistema por meio de espaço de estados e utiliza rastreamento por uma região de referência para fazer com que a câmera siga um objeto. Já Ruangpayoongsak [10] apresenta um protótipo de baixo custo com mesmo objetivo e Ross [11] utiliza a estrutura em um robô planetário. Ambos realizam a modelagem através da cinemática e utilizando as notações de Denavit-Hartenberg, aplicando controladores LQR [10] e *feedforward* [11]. Allen e Oh [12] e Tsai *et al* [13] extraem a dinâmica do sistema através das relações geométricas do modelo de projeção da câmera junto com a dinâmica dos servomotores posicionados nas juntas do sistema robótico. Estes estudos mostram a amplitude de aplicações que sistemas servo visuais com dois eixos podem apresentar, bem como diferentes formas de abordagem.

Tendo como base os trabalhos citados, este documento apresenta o desenvolvimento de um sistema com dois graus de liberdade capaz de rastrear a posição de um objeto através da aquisição de dados visuais de uma câmera. Um sistema de controle servo visual projetado, responsável por movimentar a estrutura mecânica, conhecida como *Pan-Tilt*, fazendo com que o objeto de interesse fique centralizado na imagem capturada. Realiza-se a comparação entre os resultados práticos obtidos com ganho estático e também com o escalonamento de ganhos.

Este trabalho divide-se em: Seção II para a fundamentação teórica, onde são abordados conceitos e técnicas sobre sistemas de controle servo visuais e demais temas relacionados ao desenvolvimento, Seção III tratando dos Materiais e Métodos, Seção IV para a modelagem do sistema, Seção V para o projeto dos controladores, Seção VI para os resultados obtidos e por fim Seção VII para a Conclusão.

II. FUNDAMENTAÇÃO TEÓRICA

Nesta seção são abordados tópicos fundamentais para entendimento dos métodos utilizados no desenvolvimento do protótipo, como características sobre processamento de imagens digitais e definições do sistema de controle utilizado, bem como sobre os atuadores e estrutura mecânica do sistema robótico.

A. Processamento de imagens digitais

Uma imagem digital é uma representação bidimensional de uma cena tridimensional. A mesma é expressa por uma matriz de M linhas e N colunas em função de duas coordenadas discretas, usualmente tratadas como *(i,j)* ou *(x,y)*, que representa o plano da imagem [7]. Essa matriz contém um número finito de elementos, chamados pixels, os quais apresentam localização e valor de intensidade específicos.

Entende-se como processamento de imagens práticas como aprimoramento da qualidade da imagem, a restauração através da eliminação de efeitos conhecidos que causam sua degradação e também extração de características da imagem, como localização de elementos específicos através de contornos e texturas [2]. Para a visão computacional, os níveis de complexidade do processamento de imagens dependem muito da aplicação final, geralmente se limitando à extração de características e filtros para aumento da qualidade.

A literatura mostra principalmente dois métodos de extração de características para a identificação de objetos, sendo a percepção de cores e a detecção de contornos. A identificação pode ser realizada no sistema de cores HSV (*Hue Saturation Value*). Esse sistema é muito similar ao HSL (*Hue Saturation Luminance*), o qual é frequentemente utilizado na visão computacional pois assim como o HSV, permite diferenciar cores facilmente em quantidades de matiz e saturação [7].

O HSV é separado em três escalas conforme seu próprio nome: *Hue* (Matiz), a qual caracteriza o comprimento da onda dominante da cor; *Saturation* (Saturação), a qual caracteriza a pureza da cor e *Value* (Valor), o qual caracteriza o brilho e intensidade da cor, onde a cor preta possui brilho zero.

Esse entendimento é essencial para a detecção de objetos por cores, pois é necessário determinar a faixa de valores que o mesmo se encontra e assim identificá-lo na imagem digital capturada. A Fig. 1 apresenta o cilindro HSV, a fim de auxiliar na visualização deste sistema de cores.

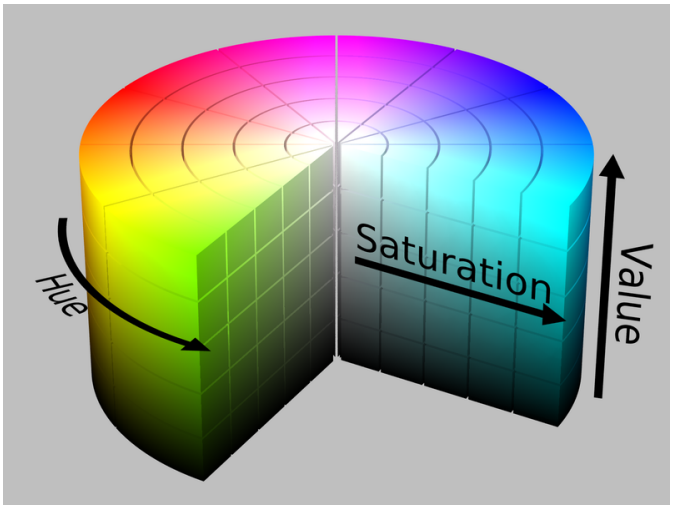

Fig. 1. Cilindro representativo do sistema de cores HSV. Retirado de [14].

O processo de filtragem se refere à formação de uma nova imagem, onde os valores dos novos pixels são transformações baseadas nos pixels da imagem original [1] e tem objetivo de eliminar ou reduzir ruídos da mesma.

Os ruídos mais comuns são o Gaussiano e o impulsivo [7]. O ruído Gaussiano pode ser modelado como variações aleatórias ao redor de um pixel, porém dentro de um desvio padrão fixo. Muitas vezes, quando não se sabe a origem exata do ruído, o mesmo é atribuído como sendo Gaussiano [2]. Já o ruído impulsivo é gerado pelo processo de aquisição da imagem, alterando de forma aleatória os pixels, tornando seus valores muito diferentes dos originais e muitas vezes afetando pixels vizinhos [2].

Dada a existência desses ruídos, utilizam-se métodos de filtragem para realizar a suavização e desfoque das imagens através da convolução com o uso de filtros passa-baixas [15], onde os principais tipos de filtros são: média de pixels (*averaging*), desfoque Gaussiano, desfoque mediano e filtragem bilateral.

B. Características de sistemas servo visuais

Um sistema de controle servo visual em tempo real opera em duas etapas [13]: a primeira consiste na aquisição dos dados visuais e identificação do objeto de interesse, através de uma câmera e utilizando artifícios de visão computacional; a segunda etapa compreende a unidade servo controlada, sendo responsável por realizar os movimentos através dos atuadores (servomotores) a fim de realizar os movimentos programados a partir das imagens obtidas.

Além disso, pode-se classificar o mesmo em relação a posição da câmera no sistema robótico em duas configurações, conforme representado na Fig. 2, onde as setas indicam a direção do movimento realizado e o visto pela câmera.

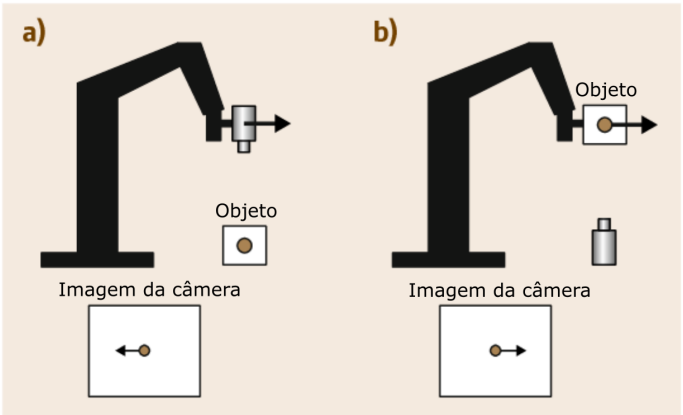

Fig. 2. (a) Configuração *Eye-in-hand*. (b) Configuração *Eye-to-hand*. Adaptado de [16].

A configuração *Eye-in-hand* consiste em posicionar a câmera junto ao ponto final de atuação do robô, onde a mesma está sujeita às variações de movimento junto do braço robótico. Essa configuração resulta em um sistema capaz de fornecer informações de posição relativas ao ponto final de atuação do sistema robótico, conhecido como *end-effector*, diretamente no espaço de trabalho do mesmo [3]. Esta configuração também é chamada de *end-point closed-loop control*.

Já a *Eye-to-hand* consiste no posicionamento da câmera de maneira estacionária, em um ponto fixo no qual a mesma observará o *end-effector* e também o objeto de interesse, sendo necessário o reconhecimento de ambos no algoritmo de identificação visual [3]. Além disso, o controle pode ser monocular quando possui somente uma câmera, ou binocular estéreo quando se utiliza duas. Neste último o sistema é capaz de extrair informações tridimensionais [17].

Existem duas principais técnicas de controle com relação ao cálculo do erro: *Image Based Visual Servoing* (IBVS) e também a *Position Based Visual Servoing* (PBVS). Ambas não utilizam informações de posição das juntas do sistema robótico [3], pois de acordo com a taxonomia estabelecida por Weiss [18], um dos precursores em estudos envolvendo controle servo visual, caso sejam utilizados sensores para realimentação dessa informação, o sistema de controle deixa de ser considerado servo visual.

Na estrutura IBVS, o controle da posição das juntas é realizado diretamente através de características extraídas da imagem [3]. Assim, o erro é calculado através da função tarefa cinemática sendo a diferença entre valores da imagem atual e os desejados, a qual transforma os pontos do sub espaço de características para o espaço de posicionamento do robô, fazendo com que o mesmo se posicione de forma a obter erro nulo [17] e não envolvendo nenhum método de estimação da posição do objeto de interesse [3]. A Fig. 3 apresenta o diagrama de blocos da estrutura IBVS conforme estabelecido por Weiss [18], onde f_d indica a posição desejada do braço robótico.

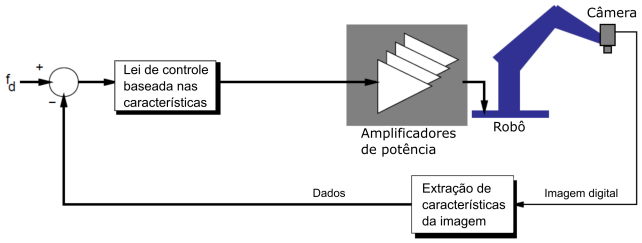

Fig. 3. Estrutura IBVS (*Image Based Visual Servoing*). Adaptado de [3].

Já o controle PBVS, consiste em obter parâmetros tridimensionais estimados através de medições da imagem [16], os quais são utilizados em conjunto com o modelo geométrico do alvo, a fim de determinar a posição do mesmo em relação à câmera [3]. Também é conhecido como *servo visão 3D*.

A Fig. 4 apresenta o diagrama de blocos da estrutura conforme estabelecido por Weiss [18], onde a malha de realimentação possui uma estimação de posição além da extração de características da imagem. Assim, o controlador trabalha sobre posições em um sistema cartesiano [4], recebendo um valor de referência f_d para sua posição desejada.

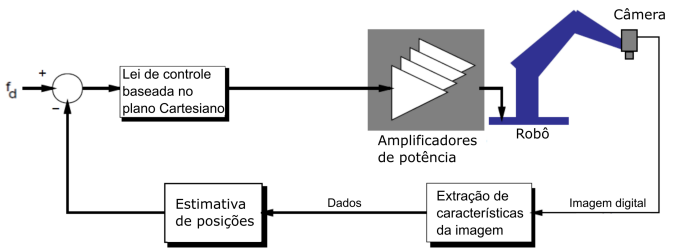

Fig. 4. Estrutura PBVS (*Position Based Visual Servoing*). Adaptado de [3].

C. Estrutura e atuadores do sistema robótico

O sistema de movimentação *Pan-Tilt* compreende um atuador robótico com dois graus de liberdade, comandado por servo motores e que trabalha sobre seu próprio eixo de rotação, possibilitando a movimentação angular da câmera acoplada ao mesmo, realizando os movimentos de inclinação (*tilt*) e rotação (*pan*) [5].

Levando em conta os vastos estudos sobre teorias para modelagem de manipuladores robóticos, essa estrutura pode ser considerada um manipulador com dois graus de liberdade. Assim é possível estimar um modelo através de análises cinemáticas ou outras técnicas existentes na literatura [11]. A Fig. 5 apresenta um esquema da estrutura com seus respectivos eixos de rotação.

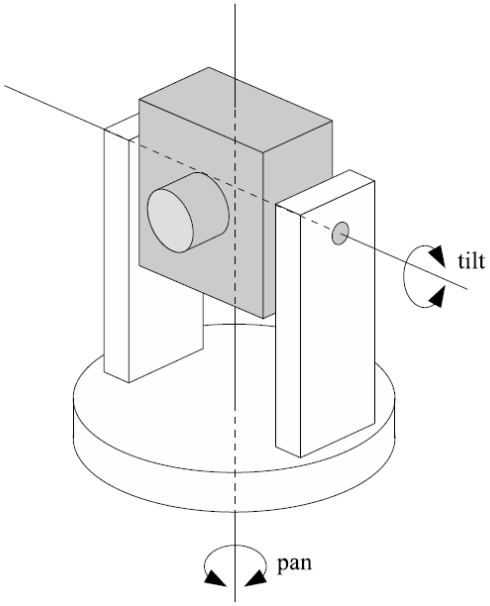

Fig. 5. Estrutura de movimentação *Pan-tilt*. Retirado de [5].

Para atuação da mesma, são utilizados servo motores em ambos os eixos. Corke [3] realizou diversos trabalhos comprovando que servo motores controlados em modo de velocidade apresentam uma resposta satisfatória quando submetidos às teorias de controle servo visual nesta estrutura.

Os servo motores de rotação contínua, também conhecidos como *servos* 360°, possuem um eixo que gira continuamente no qual se controla a velocidade e sentido de rotação do mesmo [19]. Já os servo motores controlados por posição, apresentam precisão porém possuem um ângulo máximo de movimentação. Assim, os servos de rotação contínua permitem que o movimento de rotação (*pan*) seja realizado sem limitantes angulares [19], dando mais versatilidade ao sistema.

Os servos de rotação contínua são controlados através da modulação por largura de pulso (*Pulse Width Modulation - PWM*), onde a velocidade rotacional e sua direção são determinados pela duração do pulso de nível lógico alto [20].

Este sinal PWM comumente trabalha numa frequência de 50 Hz, ou seja, um período de 20 milissegundos [21]. Com um tempo de nível lógico alto de cerca de 1,5 milissegundos o eixo do motor fica imóvel, sendo esse ponto conhecido como *rest point* [22] e a maioria dos modelos de servo motores apresenta uma regulagem deste ponto. Pulsos com larguras superiores à 1,5 milissegundos resultam numa rotação no sentido anti-horário, com a velocidade aumentando conforme a largura do pulso aumenta [22]. De maneira análoga, larguras de pulso abaixo do *rest point* resultam na rotação de sentido horário, com a velocidade aumentando conforme a largura de pulso diminui. A Fig. 6 apresenta a relação entre a largura do pulso e o comportamento do servo motor.

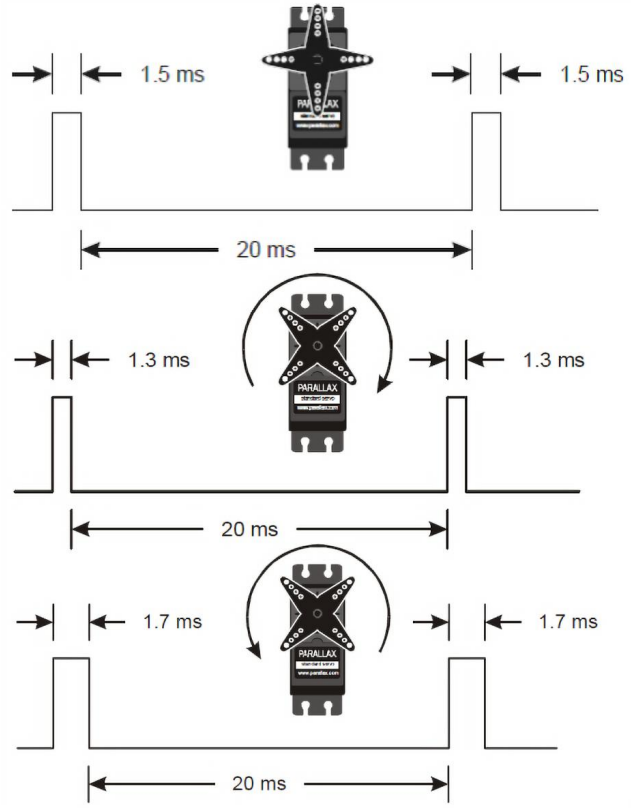

Fig. 6. Sinal de controle PWM para servos motores de rotação contínua. Adaptado de [20].

D. Controladores no domínio discreto

Com sua rápida evolução, os microprocessadores ganharam espaço no controle de processos em tempo real, substituindo uma série de componentes que são utilizados para o controle analógico, tornando o sistema mais versátil e preciso [23].

Na grande maioria das aplicações, o sistema ou processo a ser controlado e o atuador são sistemas analógicos. Assim, se faz necessária uma conversão digital para analógica através de um DAC (*Digital-to-Analog Converter*) e também o processo inverso, analógico para digital através de um ADC (*Analog-to-Digital Converter*) [24] para que ocorra a comunicação do processo com o microcontrolador. A Fig. 7 apresenta o diagrama básico e simplifcado para um sistema de controle digital.

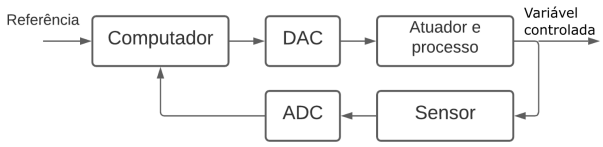

Fig. 7. Configuração de um sistema digital. Adaptado de [24].

O DAC é responsável por enviar as informações do microcontrolador para o processo, enquanto que o ADC repassa as informações obtidas para o microprocessador através do sensor responsável pela medição da variável controlada.

A conversão desses sinais ocorre repetitivamente em instantes de um tempo T, também conhecido como período de amostragem [25]. Essa é uma variável de grande importância quando se trabalhando com sistemas digitais e segundo o Teorema da Amostragem de Nyquist, a frequência de amostragem (sendo essa o inverso do período) deve ser no mínimo duas vezes a frequência do sistema, para garantir a reconstrução de um sinal contínuo a partir de uma sequência amostrada, sem perdas. No entanto, para fins de aplicação em sistemas de controle costumam-se empregar regras práticas e o período de amostragem é tratado como uma variável de sintonia quando há tal. Assim, buscando garantir estabilidade e estando dentro das limitações físicas do mesmo, deve-se utilizar um valor de dez vezes o valor da frequência do sistema [23].

A maioria das análises de sistemas e projeto de controladores é realizada no domínio da frequência complexa de Laplace [24], onde estes ficam em função da variável complexa s e buscando facilitar a aplicação de controladores digitais utiliza-se a Transformada Z [23]. A mesma é uma ferramenta importante que simplifica a solução de problemas em tempo discreto convertendo equações diferenciais em equações a diferenças.

Após análise e projeto dos controladores no domínio Z, aplica-se alguma das técnicas de aproximação de diferenciação para se implementar o controlador, como por exemplo o Método de Euler, também conhecido como *forward difference*, o método diferencial ou *backward difference* e também o método de Tustin [24]. Cada um com seu nível de complexidade e fidelidade ao sistema no tempo contínuo.

III. MATERIAIS E MÉTODOS

Nesta seção são apresentados os procedimentos para aquisição dos dados e processamento da imagem obtida, bem como componentes e acionamentos utilizados para o desenvolvimento do protótipo.

A. Aquisição e processamento das imagens

Para construção do dispositivo foi utilizada uma *webcam* Logitech HD C920, sendo o componente sensor e responsável pelo envio das informações visuais para o computador. Esses dados são recebidos e processados através da linguagem de programação Python 3.8 e do uso da biblioteca para visão computacional *OpenCV* 4.5.

Foi desenvolvido um algoritmo, conforme Anexo 1, para identificação do objeto por meio da criação de uma máscara de cor, a qual consiste em uma nova imagem digital somente com as cores preta e branca, conforme apresentado na Fig. 8. Para isso, foi necessária a determinação dos limites superiores e inferiores da intensidade de cor no espaço HSV que compreendem o objeto a ser identificado, a qual foi realizada de maneira experimental, variando-se estes valores até que a máscara evidencie somente o objeto de interesse.

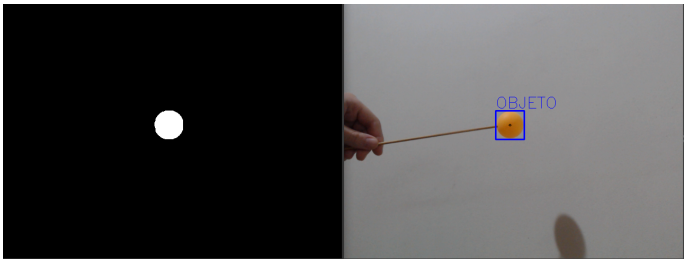

Fig. 8. Máscara de cor e imagem original.

Nessa nova imagem, aplica-se a técnica de detecção de contornos, a qual corresponde à identificação dos pixels brancos que possuem pixels vizinhos da cor preta. Com isso, obtém-se as coordenadas dos extremos do objeto e também de seu centro.

Em seguida, utiliza-se um filtro mediano, buscando suavizar as bordas do objeto a ser identificado e diminuir ruídos posteriores na determinação de seu centro. Com essa posição encontrada, pode-se calcular o erro em relação ao centro da imagem conforme,

$$erro = \frac{Dim}{2} - centro_{\text{obj}} \tag{1}$$

Onde *Dim* corresponde à dimensão ou resolução da imagem, ou seja, largura e altura da imagem digital capturada, em pixels, sendo de 640 x 480. O $centro_{\text{obj}}$ é a posição central do objeto identificado e o *erro* é a distância entre o centro do objeto e da imagem nos eixos horizontal e vertical. Dessa maneira, os erros máximos que podem ser obtidos são de $\pm$ 320 pixels para o eixo horizontal e $\pm$ 240 pixels para o eixo vertical.

B. Acionamento dos servo motores e montagem do protótipo

Obtidos os valores de erro conforme a seção anterior, estes são enviados através de comunicação Serial para a placa Arduino Mega, responsável pelo acionamento e controle dos servo motores de rotação contínua SM-S4306R.

Estes servo motores, permitem uma tensão de alimentação de 4,8 até 6,0 V, onde torque e velocidade são proporcionais à tensão [26]. Além disso, seu consumo de corrente atinge aproximadamente 1 A, excedendo o limite máximo de corrente da placa Arduino Mega e diante dessa limitação, utiliza-se uma fonte externa de 5 Vcc, 2,3 A.

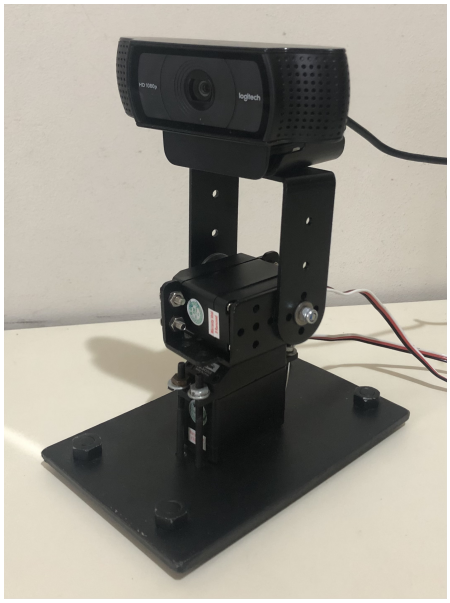

Fig. 9. Protótipo final desenvolvido.

O protótipo final pode ser observado na Fig. 9, no qual sua estrutura mecânica foi desenvolvida com chapas e acoplamentos de engrenagem de aço, garantindo robustez e boa fixação entre os eixos, bem como a base do protótipo proporciona a sustentação necessária para realização dos movimentos.

Os sinais *PWM* enviados pela placa Arduino para acionamento e controle da velocidade dos motores tem sua largura de pulso variando de 1,50 milissegundos, a qual mantém o motor parado, até cerca de 1,86 milissegundos, atingindo a máxima velocidade de rotação no sentido anti-horário, e até aproximadamente 1,14 milissegundos, na qual se atinge a máxima velocidade no sentido horário.

Na Fig. 10 está representado o esquema de ligação utilizado para acionamento dos servomotores, onde a placa Arduino é alimentada pela porta serial conectada ao computador, por onde também são recebidos os valores de erro.

C. Determinação da distância focal da câmera

A distância focal é a relação de distância entre a lente da câmera, o plano de foco e a imagem. Neste projeto, a grandeza dessa relação é dada em pixels pois relaciona a distância real,

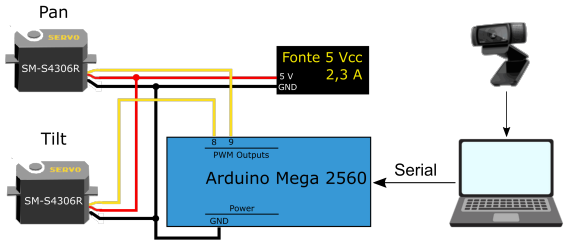

Fig. 10. Esquema de ligação do circuito.

dada em milímetros, e a distância projetada na imagem digital, em pixels. A determinação deste parâmetro da câmera é necessário pois é utilizado posteriormente na modelagem da dinâmica do sistema.

O valor da distância focal foi estimado através de experimentos práticos baseados no modelo da Fig. 11, onde são utilizados dois corpos de teste com comprimentos diferentes, sendo L_1 de 233 mm e L_2 de 115 mm.

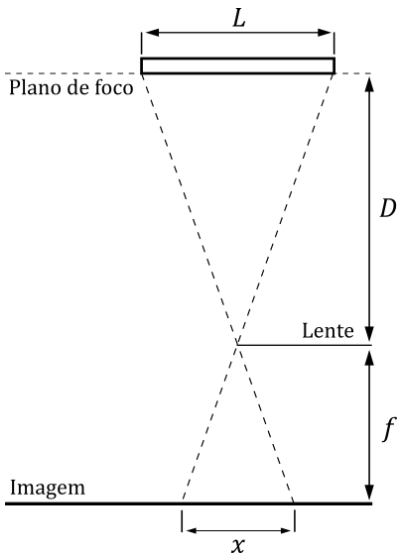

Fig. 11. Modelo do experimento realizado para determinação da distância focal.

Observa-se na Fig. 11 que L e x são paralelos e os ângulos formados pelas linhas tracejadas de projeção do objeto são opostos pelo vértice, ou seja, são triângulos congruentes.

Dessa forma, pode-se utilizar da semelhança de triângulos e determinar a distância focal através de (2),

$$f = \frac{x}{L} \cdot D \tag{2}$$

Onde x corresponde à largura do objeto em pixels, L à largura do objeto em milímetros e D corresponde à distância entre o plano de foco, onde encontra-se o corpo de teste, e a lente da câmera. Buscando a obtenção de diferentes amostras para o cálculo da distância focal, realizaram-se medições com

diferentes distâncias D. Os valores obtidos estão dispostos na Tabela I.

TABELA I
Valores de distância focal obtidos através do experimento.

D (mm)	x_1 (pixels)	x_2 (pixels)	f_1 (pixels)	f_2 (pixels)
300	487	242	627,04	631,30
400	368	185	631,76	643,48
500	297	147	637,34	639,13
600	250	126	643,78	657,39

Realizando o cálculo da média dos oito valores obtidos, tem-se uma distância focal de aproximadamente 638,90 pixels, apresentando um desvio padrão de cerca de 9,55 pixels, o que corresponde a 1,49%. Considerando a precisão dos instrumentos de medição utilizados no experimento, admite-se como um resultado satisfatório e desta maneira, adota-se o valor encontrado na determinação da dinâmica do sistema.

IV. MODELAGEM DO SISTEMA

Esta seção apresenta a modelagem do sistema combinando o modelo de projeção da câmera, extraído através de características dadas pelas relações geométricas, enquanto que a dinâmica dos servomotores é determinada experimentalmente.

A. Dinâmica de projeção da câmera

Para identificação do sistema, se faz necessária a determinação da função de transferência que relaciona o deslocamento do objeto e o ângulo da câmera. O sistema utilizado apresenta os movimentos de *Pan* e *Tilt* que podem ser modelados individualmente, resultando em funções de transferência idênticas para o método de modelagem adotado.

A obtenção da dinâmica de projeção da câmera é realizada através das relações geométricas do sistema, conforme estabelecido na Fig. 12. Este modelo considera o deslocamento do objeto perpendicularmente ao eixo Z_O, onde em um primeiro instante o objeto se localiza no ponto alvo A, e em instante seguinte no ponto alvo B.

Dessa forma, através das relações entre o eixo Z_O e os pontos alvo e também assumindo movimentações em pequenos ângulos, pode-se determinar o ângulo θ_b referente ao deslocamento do objeto conforme,

$$\theta_b = \theta_t - \theta_c \approx \frac{x_t}{Z+L} - \frac{x_c}{Z+L} \tag{3}$$

Onde x_t é o deslocamento atual do objeto em relação ao eixo Z_O, x_c é o deslocamento em instante anterior, Z a distância da lente da câmera até o plano de foco e L é a distância entre o centro de rotação da câmera e a extremidade da lente.

Ainda adotando a aproximação de pequenos ângulos e utilizando o modelo *pin-hole* para projeção da câmera, o qual assume que todos os raios luminosos do objeto se concentram em um único ponto focal no plano da imagem [27], obtém-se,

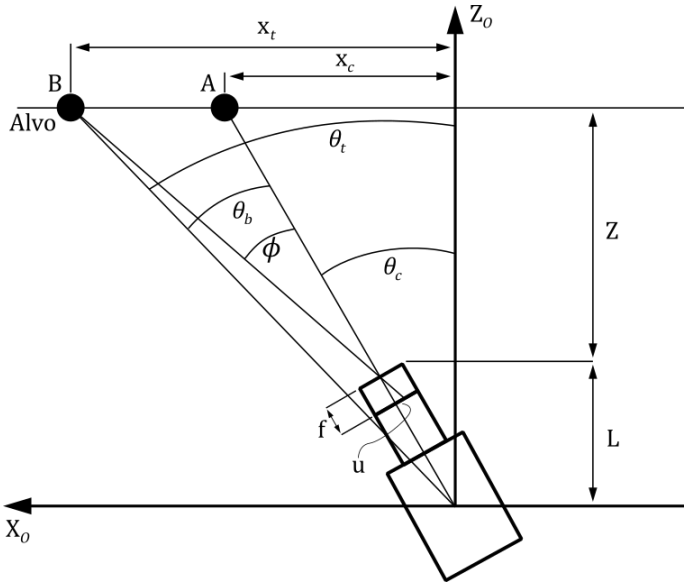

Fig. 12. Modelo de relações geométricas para o modelo de projeção da câmera. Adaptado de [12] e [13].

$$\phi \approx \frac{x_t - x_c}{Z} \approx \frac{u}{f} \tag{4}$$

onde ϕ corresponde ao ângulo atual entre o alvo e o centro da lente e u é a distância entre a projeção do objeto e o centro da imagem digital, dada em pixels. Manipulando a Equação (3) e isolando o deslocamento do alvo, tem-se,

$$x_t - x_c = \theta_b(Z+L) \tag{5}$$

Utilizando as Equações (4) e (5), pode-se estabelecer uma relação entre o ângulo θ_b e a distância deslocada no plano da imagem, a qual corresponde à função de transferência de realimentação $H(s)$ conforme,

$$u = \frac{f}{Z}\theta_b(Z+L) \rightarrow \frac{u}{\theta_b} = \frac{f(Z+L)}{Z} \tag{6}$$

Através da Fig. 12 obtém-se a relação entre o deslocamento linear do alvo e o deslocamento angular, expressa por,

$$\frac{\theta_t}{x_t} = \frac{1}{Z+L} \tag{7}$$

As equações expressas em (6) e (7) são empregadas posteriormente junto da dinâmica do servo motor na concepção do sistema em malha fechada.

B. Dinâmica dos servo motores

O experimento para identificação da dinâmica de velocidade dos servo motores foi realizado acoplando um potênciometro ao eixo do servo motor, utilizando um divisor de tensão. Onde através de testes verificou-se que para uma volta do servomotor, ocorre a variação de aproximadamente 50 *bits* na leitura da entrada analógica. Assim, cada variação aproximada de 7,2 *bits*

corresponde à 1 grau de deslocamento angular do servomotor. Diante disso, foram aplicadas diferentes larguras de pulso no servomotor, resultando em diferentes velocidades. As curvas obtidas podem ser observadas na Fig. 13.

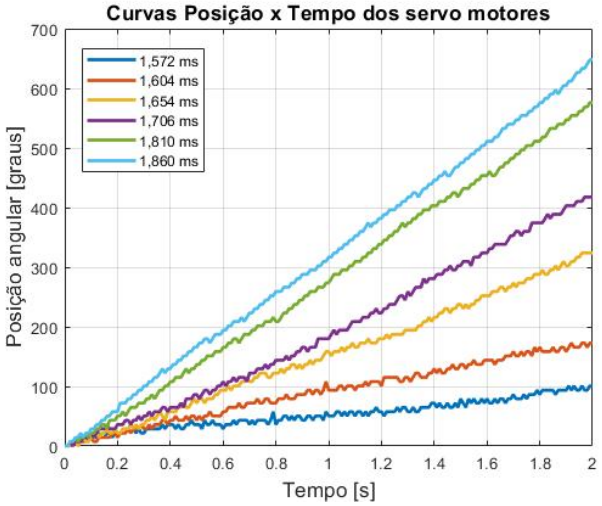

Fig. 13. Curvas de posição angular no tempo para os diferentes sinais *PWM* aplicados ao servomotor.

Percebe-se a existência de pequenas oscilações nas curvas, devido a não linearidade da tensão no potênciometro, além de pequenos ruídos provenientes da resolução do conversor analógico-digital do Arduino e também do atrito existente no acoplamento utilizado entre motor e potênciometro. Diante disso, sabendo que a aquisição dos dados foi realizada com um período de amostragem de 10 milissegundos, aplica-se a Transformada Rápida de Fourier (FFT) nos sinais, obtendo resultados homogêneos para a densidade espectral, conforme apresentado na Fig. 14.

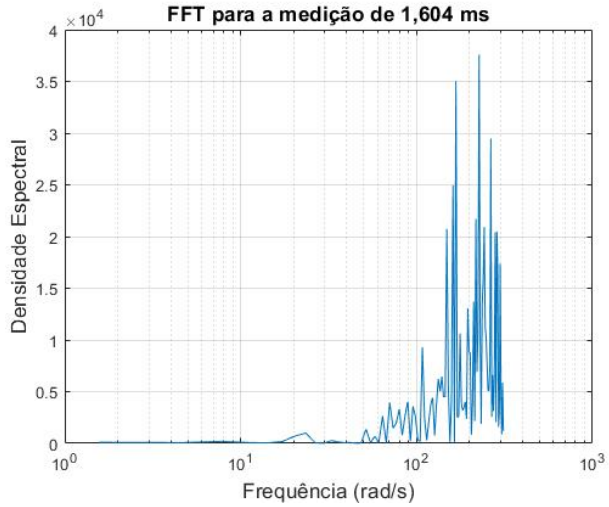

Fig. 14. Densidade espectral do sinal de largura de pulso de 1,604 ms obtida através da FFT.

Na Fig. 14 está a densidade espectral para a largura de pulso aplicada de 1,804 milissegundos, na qual percebe-se que as principais frequências estão acima de 10 rad/s. Com auxílio do *software* Matlab, aplicou-se um filtro passa-altas com frequência de corte de 5 rad/s e em seguida utilizou-se o Estimador Paramétrico dos Mínimos Quadrados Recursivo como técnica de identificação para determinar as funções de transferência no domínio discreto para as respostas obtidas, sendo posteriormente convertidas para o domínio contínuo, também através do Matlab. Para essa técnica de identificação, adotou-se o modelo dinâmico *ARMAX* (Autoregressivo com Média Móvel e entradas exógenas), onde o polinômio $C(z)$ foi identificado e omitido pois não é posteriormente utilizado no decorrer do trabalho.

Com objetivo de verificar a semelhança entre as dinâmicas encontradas, aplicou-se uma entrada de degrau unitário para cada uma, resultando nas respostas apresentadas na Fig. 15.

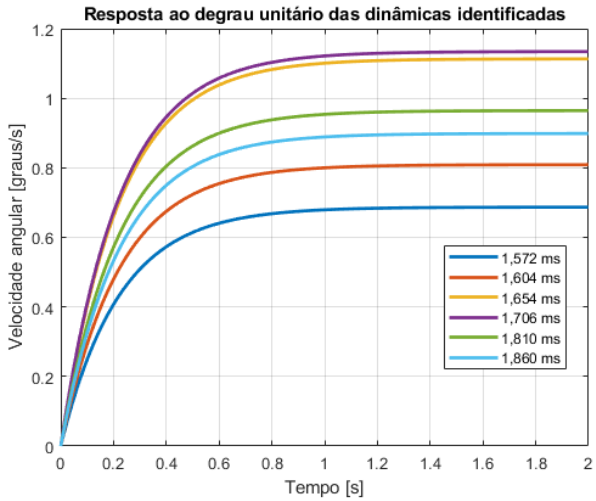

Fig. 15. Resposta ao degrau para os modelos identificados.

As respostas obtidas apresentam diferentes ganhos estáticos e constantes de tempo da planta, indicando uma não-linearidade da dinâmica de velocidade do servo motor. Diante disso, decide-se por encontrar uma planta principal sendo dada pela média dos ganhos estáticos e constantes de tempo das plantas estimadas, e outras duas plantas intermediárias entre a média e as plantas identificadas com caractéristicas mais distantes. Dessa maneira, a função de transferência principal do servo motor $M(s)$, correspondendo a velocidade angular $\dot{\theta}_c$ em função da largura de pulso aplicada PW, apresenta um ganho estático de 0,933 e uma constante de tempo de 0,267 segundos, e é dada por,

$$\frac{\dot{\theta}_c}{PW} = M(s) = \frac{3,50}{s+3,75} \tag{8}$$

Já as funções de transferência intermediárias determinadas são dadas por (9) e (10),

$$M_1(s) = \frac{3,08}{s+2,98} \tag{9}$$

$$M_2(s) = \frac{4,43}{s+5,46} \tag{10}$$

C. Diagrama final do sistema

Obtidas as funções de transferência para projeção da câmera e as dinâmicas dos servo motores, o diagrama do sistema de malha fechada é definido conforme a Fig. 18.

O mesmo é composto pela malha de realimentação, a qual apresenta a função de transferência $H(s)$ que relaciona o erro em radianos e pixels, obtida conforme (6). A malha direta é formada pelo controlador digital e pela planta $P(s)$ do servo motor, a qual corresponde à integral da dinâmica de velocidade $M(s)$, resultando na dinâmica de posição do motor, sendo dada por (11) para a planta principal,

$$\frac{\theta_c}{PW} = P(s) = \frac{3,5}{s(s+3,75)} \tag{11}$$

Já o deslocamento do objeto é visto como uma perturbação externa no sistema, onde sua dinâmica é a relação descrita conforme a Equação (7).

V. PROJETO E DISCRETIZAÇÃO DOS CONTROLADORES

O projeto dos controladores é realizado pelo método do Lugar Geométrico das Raízes (LGR), o qual consiste em analisar a posição dos polos da malha direta da planta e escolher a posição dos polos de malha fechada de acordo com a resposta desejada.

Sabendo que a planta $P(s)$ apresenta um integrador natural, já apresentando a caraterística de rastreamento de referência e rejeição à perturbações externas, um controlador PD é suficiente para que haja uma resposta satisfatória do sistema. Dessa forma, decide-se por utilizar um controlador com ganho estático, projetado a partir da planta principal $P(s)$ e devido a não-linearidade da planta, utiliza-se um controle adapativo com escalonamento de ganhos, conhecido como *Gain Scheduling*, com seu princípio de funcionamento ilustrado na Fig. 16.

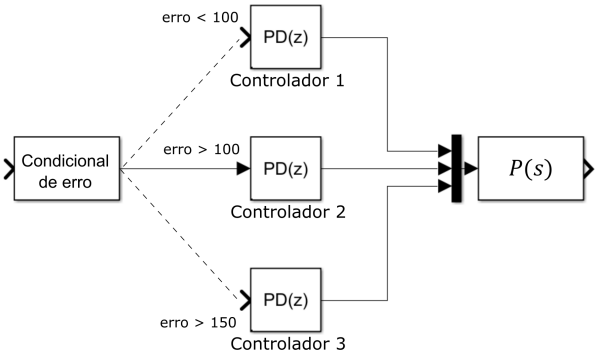

Fig. 16. Diagrama do funcionamento do controle com escalonamento de ganhos.

Essa técnica consiste em utilizar diferentes ganhos para diferentes pontos de operação da planta, que neste caso são baseados no valor do erro medido.

O processo para determinação dos ganhos dos controladores é análogo para as três plantas determinadas anteriormente. Dessa maneira, levando em conta a planta principal $P(s)$, proveniente de $M(s)$, a malha direta do sistema é dada por,

$$P^*(s) = \frac{C(s)}{K_p} P(s)H(s) \tag{12}$$

Para a análise desta, um controlador PD sem filtro derivativo é dado por,

$$C(s) = K_p(sT_d + 1) \tag{13}$$

onde $\frac{1}{T_d}$ representa a posição do zero a ser inserido no sistema pelo controlador. Sabendo que a posição dos polos da planta estão localizados em 0 e -3,75 rad/s, opta-se por posicionar o zero em 10 rad/s, obtendo o valor de $T_d = 0,1$ segundos. Deste modo, é possível verificar o Lugar das Raízes da malha direta, conforme a Fig. 17.

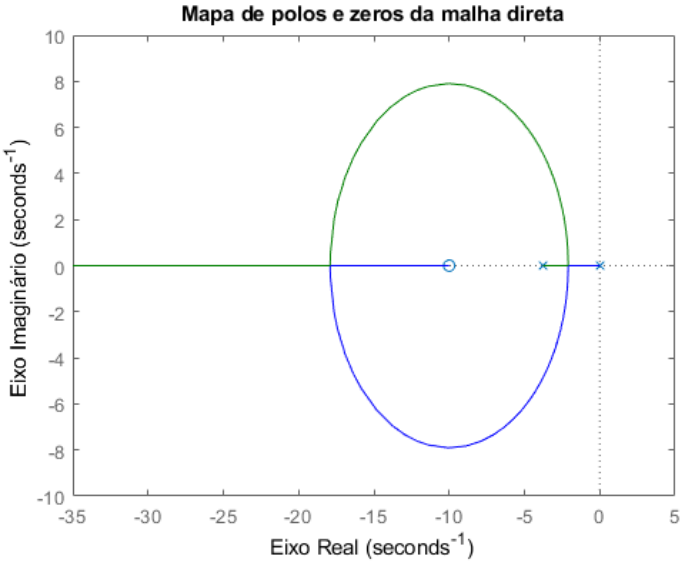

Fig. 17. Lugar das Raízes da malha direta.

Sabendo que é possível posicionar os polos de malha fechada sobre o eixo real, o que sugere que a resposta do sistema se aproximará de uma de primeira ordem, através do Matlab e junto de testes práticos realizados define-se como ganho do controlador $K_p = 0,4$, o qual irá posicionar os polos de malha fechada em -30 e -13,1 rad/s.

De maneira análoga, a planta proveniente de $M_1(s)$ apresenta polos em 0 e -2,98 rad/s. Mantendo o valor de $T_d = 0,1$ segundos e utilizando $K_p = 0,67$, os polos de malha fechada do sistema com o segundo controlador localizam-se em -49,6 e -11,8 rad/s. Já para a planta proveniente de $M_2(s)$, mantendo o mesmo valor

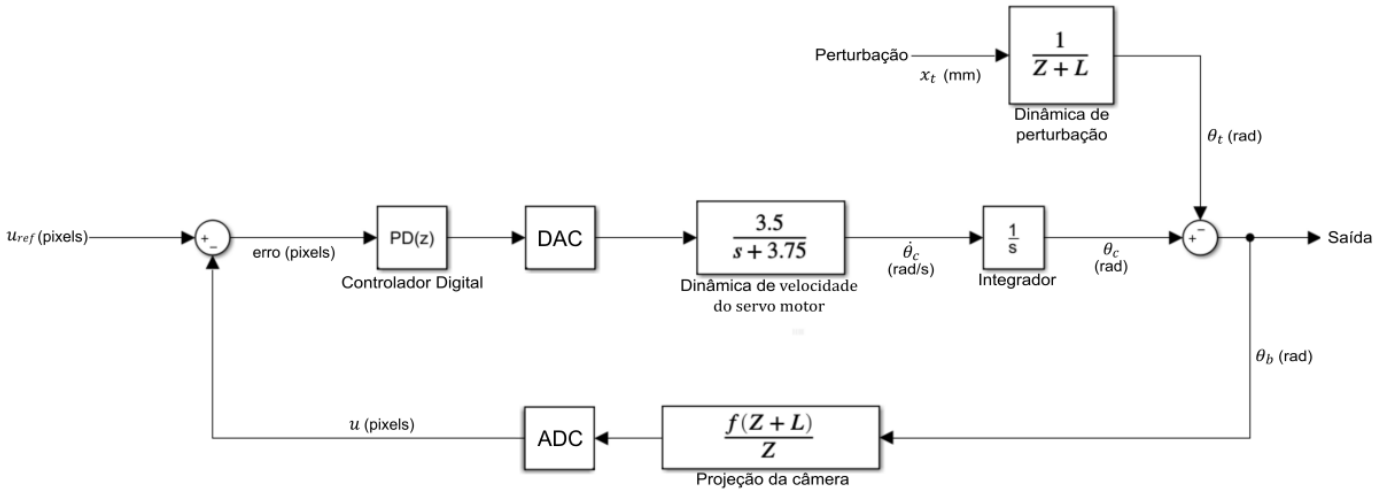

Fig. 18. Diagrama de blocos do sistema.

de T_d e utilizando $K_p = 0,90$, os polos de malha fechada se localizam em -120 e -10,4 e sendo este o terceiro controlador. O critério de escolha dos valores do ganho K_p para os controladores baseados nas plantas intermediárias foi manter o polo dominante da malha fechada em frequências próximas entre os sistemas, além de serem validados de maneira iterativa em testes práticos.

Para a discretização e implementação dos controladores digitais é necessária a determinação do período de amostragem, a qual é realizada a partir dos polos de malha fechada do sistema. Utilizando o Teorema da Amostragem de Nyquist e sabendo que o polo mais rápido da planta principal encontra-se em 30 rad/s, a frequência de banda do sistema é dada por,

$$f_b = \frac{\omega_b}{2\pi} = \frac{30}{2\pi} = 4,7746\ Hz \tag{14}$$

A determinação da frequência de amostragem f_s utiliza uma constante K, a qual deve estar entre 5 e 10 para uma atenuação de pouco mais de 10 vezes do sinal. Utilizando um valor de $K = 6$, tem-se,

$$f_s = 7 \cdot K \cdot f_b = 200,5352\ Hz \tag{15}$$

Assim, calcula-se o período de amostragem T sendo o inverso da frequência conforme,

$$T = \frac{1}{f_s} \approx 4,986\ ms \tag{16}$$

Com o período de amostragem estabelecido, aplica-se o método de discretização *backward Euler*, no qual o sinal de controle u_C é dado por,

$$u_{C[k]} = \frac{1}{T}[K_p \cdot (T_d + T) \cdot e_{[k]} - K_p \cdot T_d \cdot e_{[k-1]}] \tag{17}$$

O mesmo é implementado no algoritmo de controle, o qual pode ser visualizado no Anexo 2 onde são executados indivudalmente para cada um dos eixos de movimentação.

VI. RESULTADOS

Com objetivo de verificar o rastreamento do objeto e comportamento do sistema, visto que o valor de referência u_{ref} é de 0 pixels, foram aplicadas perturbações dos tipos degrau e rampa, com diferentes amplitudes em ambos os eixos de rotação, utilizando controladores com ganho estático e também com a aplicação do escalonamento de ganhos.

Primeiramente, realizou-se a simulação utilizando a planta principal e os ganhos definidos para o controlador PD com ganho estático, obtendo os resultados da Fig. 19.

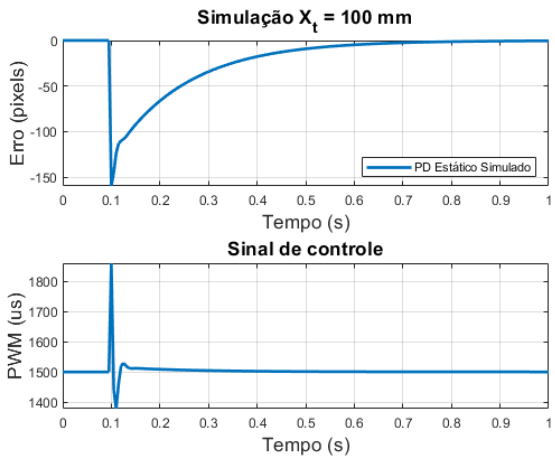

Fig. 19. Resultado de simulação para uma perturbação degrau de 100 mm de deslocamento.

Para essa resposta, aplicou-se uma perturbação do tipo degrau referente à um deslocamento de 100 mm do objeto, o que implica em um erro de aproximadamente 150 pixels. Com isso, o tempo de acomodação foi de cerca de 0,5 segundos, onde o sinal de controle atingiu o seu máximo de 1860 microssegundos, valor o qual foi saturado pois corresponde à velocidade máxima atingida pelos servomotores utilizados na prática.

Em seguida, verificado o comportamento do sistema modelado, realizaram-se testes práticos no protótipo. Para uma perturbação do tipo degrau com um deslocamento de 100 mm do objeto, os resultados obtidos para valores de erro estão dispostos na Fig. 20 para o eixo X (*Pan*) e Fig. 21 para o eixo Y (*Tilt*).

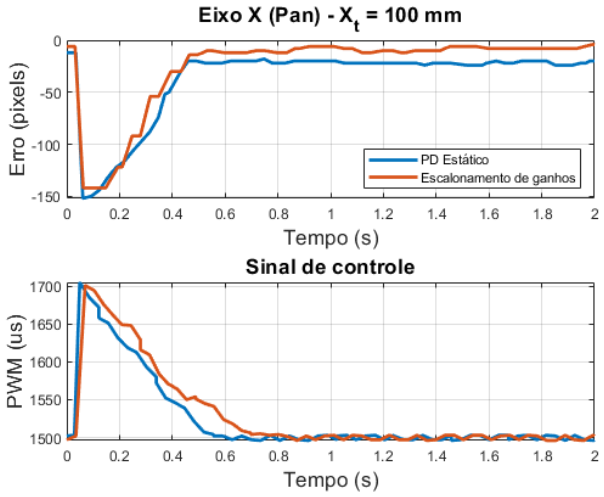

Fig. 20. Respostas dos sistemas para perturbação do tipo degrau com 100 mm de deslocamento no eixo X.

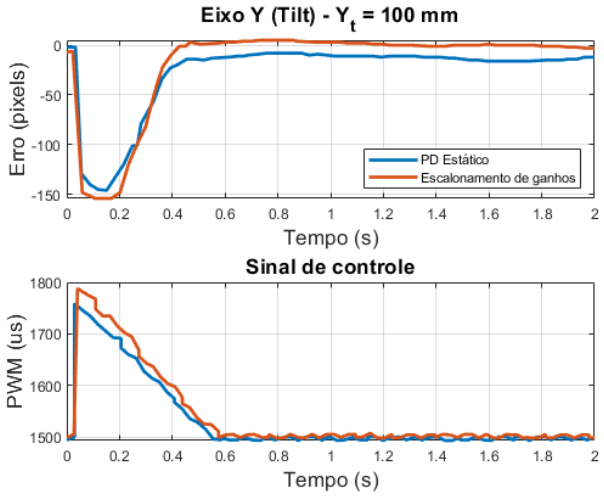

Fig. 21. Respostas dos sistemas para perturbação do tipo degrau com 100 mm de deslocamento no eixo Y.

As respostas obtidas para o eixo X apresentam um tempo de acomodação de aproximadamente 0,5 segundos e ausência de sobressinal, apresentando característica superamortecida para ambos os controles utilizados. O ganho estático apresenta um erro de regime de cerca de 20 pixels, enquanto que o escalonamento de ganhos apresenta um erro de regime de 10 pixels. Além disso, o máximo sinal de controle atingido foi de aproximadamente 1700 microssegundos.

Já a resposta obtida para o eixo Y, apresenta um tempo de acomodação de 0,45 segundos e erros próximos de 11 e 2 pixels, utilizando o ganho estático e o ganho escalonado, respectivamente. O sinal de controle foi um pouco superior com o escalonamento de ganhos, atingindo um máximo de 1788 microssegundos enquanto que o ganho estático atingiu o pico de 1759 microssegundos.

Aplicando uma perturbação do tipo degrau com deslocamento de 150 mm, obtiveram-se os resultados conforme a Fig. 22 para o eixo X. Ambos os controles apresentaram um tempo de acomodação próximo de 0,58 segundos, porém o erro de regime para o ganho escalonado foi de 15 pixels, enquanto que com o ganho estático esse valor foi de 30 pixels. O sinal de controle também se mostrou semelhante, sendo levemente superior no ganho escalonado, com seu pico em 1777 microssegundos enquanto que o ganho estático atingiu 1764 microssegundos.

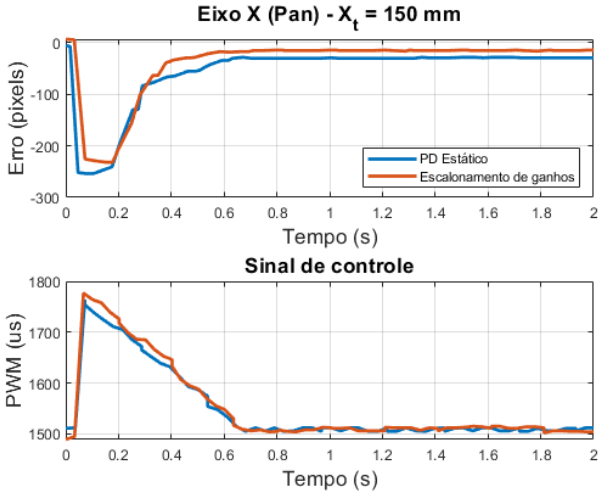

Fig. 22. Respostas dos sistemas para perturbação do tipo degrau com 150 mm de deslocamento no eixo X.

Já para o movimento de *Tilt* do eixo Y com a perturbação degrau de 150 mm, obteve-se o resultado conforme a Fig. 23. A resposta para o controle com ganho estático apresentou um erro de regime de 25 pixels, com um tempo de acomodação de 0,4 segundos. Já o escalonamento de ganhos apresentou um tempo de acomodação semelhante com um erro menor, sendo de aproximadamente 10 pixels. Percebe-se que o sinal de controle para o ganho escalonado foi superior durante o período de atuação, com um valor máximo de 1786 microssegundos, enquanto que para o ganho estático esse valor foi de 1745 microssegundos.

Em seguida, aplicou-se uma perturbação em rampa com

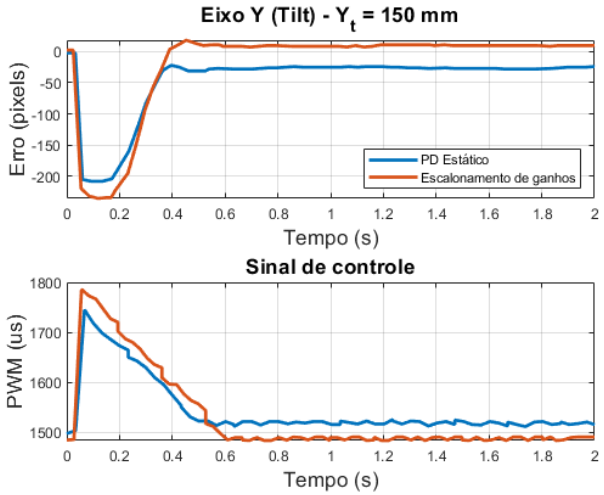

Fig. 23. Respostas dos sistemas para perturbação do tipo degrau com 150 mm de deslocamento no eixo Y.

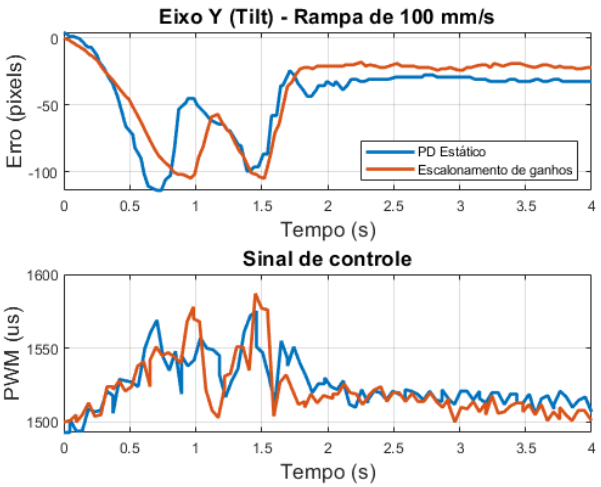

Fig. 25. Respostas dos sistemas para perturbação do tipo rampa com velocidade de deslocamento de 100 mm/s no eixo Y.

velocidade do objeto de 100 mm/s durante 2 segundos, resultando em um deslocamento total de 200 mm.

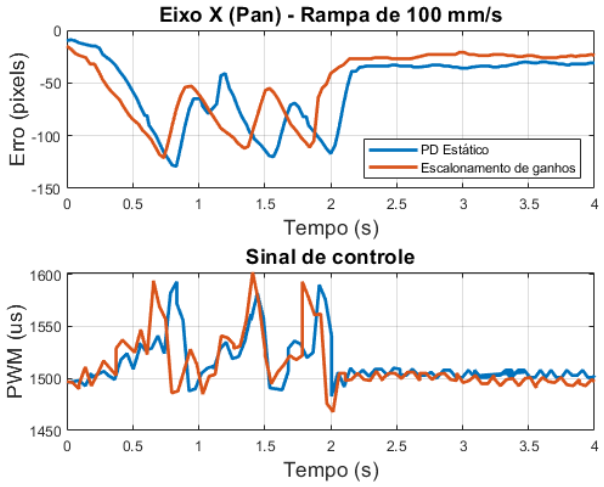

Fig. 24. Respostas dos sistemas para perturbação do tipo rampa com velocidade de deslocamento de 100 mm/s no eixo X.

As respostas obtidas no eixo X, dispostas na Fig. 24 apresentam dinâmicas oscilatórias, com um erro máximo de aproximadamente 120 pixels para ambos os controladores. Os valores para o erro de regime ficaram próximos, sendo cerca 24 pixels para o escalonamento de ganhos e 34 pixels para ganho estático. Observando o sinal de controle, percebe-se que sua oscilação ocorre de acordo com o valor de erro, onde os picos apresentam valores de 1602 microssegundos com o ganho escalonado e 1593 com o ganho estático.

Já as respostas obtidas para o eixo Y, dispostas na Fig. 25 apresentaram uma leve semelhança em sua dinâmica oscilatória e tendo como diferença o erro de regime atingido, sendo de aproximadamente 30 pixels para o ganho estático e 21 pixels para o escalonamento de ganhos. Já para o sinal de controle, os maiores valores alcançados foram de 1587 microssegundos no escalonamento de ganhos e 1575 microssegundos para o ganho estático.

Por último, aplicou-se uma perturbação em rampa com velocidade do objeto de 200 mm/s durante 1 segundo, resultando em um deslocamento total de 200 mm, no qual obtiveram-se as respostas apresentadas nas Figs. 26 e 27.

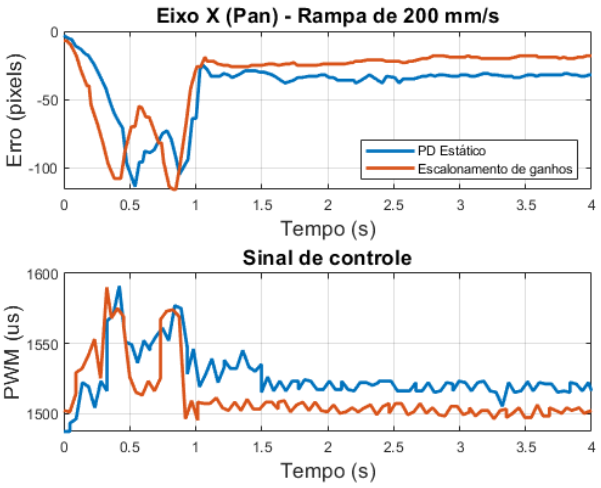

Fig. 26. Respostas dos sistemas para perturbação do tipo rampa com velocidade de deslocamento de 200 mm/s no eixo X.

As respostas obtidas para o eixo X conforme a Fig. 26, mostram que ambos os sistemas apresentaram oscilação semelhante com um erro de até 115 pixels e atingindo o valor de regime em tempos aproximados. Porém, o erro de regime para o ganho estático foi maior, sendo cerca de 32 pixels, enquanto que o ganho escalonado atingiu um erro próximo de 18 pixels. Enquanto isso, os máximos sinais de controle para foram de

1590 microssegundos para ambos os sistemas, porém nota-se a diferença quando o erro atinge o seu valor de regime. O valor do sinal de controle para o ganho estático se mantém em torno de 1520 microssegundos, enquanto que o ganho escalonado oscila próximo de 1500 microssegundos. Essa diferença ocorre devido a zona morta existente, onde o sinal de controle do ganho estático não é suficiente para movimentar o servomotor.

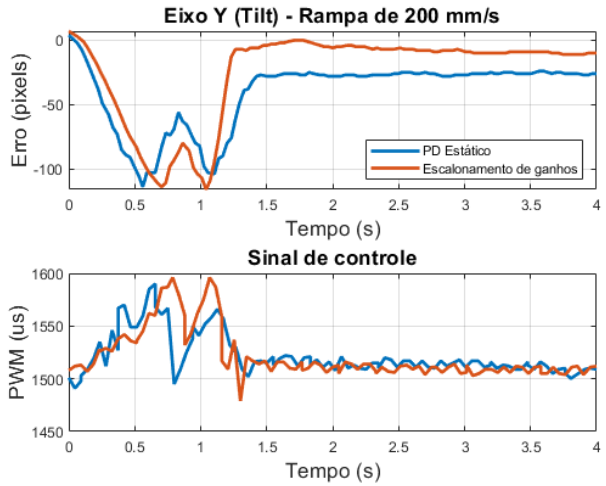

Fig. 27. Respostas dos sistemas para perturbação do tipo rampa com velocidade de deslocamento de 200 mm/s no eixo Y.

Os resultados obtidos para o eixo Y, conforme dispostos na Fig. 27, mostram que o controle com ganho estático apresentou oscilações com maiores amplitudes e atingiu o valor de regime em aproximadamente 1,45 segundos, sendo um erro de cerca de 26 pixels. Já o escalonamento de ganhos atingiu o regime em 1,30 segundos e com um valor de erro próximo de 8 pixels, aprensentando um sinal de controle mais elevado com dois picos em 1596 microssegundos, enquanto que o ganho estático atinge um pico de 1590 microssegundos e um segundo pico de 1566 microssegundos.

Os resultados aqui apresentados foram obtidos através de testes realizados com movimentos manuais, assim existindo uma falta de uma repetibilidade dos resultados obtidos, devido aos possíveis erros humanos no posicionamento dos degraus, variação da velocidade e deslocamento nas rampas realizadas. Entretanto, comparando o resultado entre a simulação e os degraus de 100 mm, percebe-se a proximidade do valor de erro para a perturbação. A maior diferença notada é do sinal de controle, visto que na simulação não existe a zona morta originada do atrito da estrutura.

Contudo, analisando os resultados práticos obtidos, observa-se que o controle com escalonamento de ganhos apresentou dinâmicas muito semelhantes às obtidas com o uso do ganho estático, porém apresentando um erro de regime inferior e mais próximo da referência. Percebe-se também que durante a atuação do controle e deslocamento do objeto, o controle com escalonamento de ganhos apresenta um valor superior ao ganho estático, o que resulta no menor erro de regime pois a velocidade angular do servomotor é maior, permanecendo menos tempo na zona morta de atuação do servomotor, onde o torque é insuficiente para movimentar o eixo.

VII. CONCLUSÃO

O desenvolvimento de um sistema de controle servo visual para rastreamento de objetos pode ser realizado de diferentes maneiras e com diferentes abordagens. Neste trabalho, construiu-se uma estrutura robótica com dois graus de liberdade e uma *webcam* montada na configuração *Eye-in-hand*. Nesta, a aquisição e processamento de imagens foi realizada através de uma *webcam* conectada ao computador, por meio da biblioteca de visão computacional *OpenCV* em linguagem *Python*. Já a execução do algoritmo de controle e comando dos servo motores foi realizada através de uma placa Arduino Mega, recebendo através de comunicação Serial os valores de erro obtidos pelo algoritmo em *Python*.

Utilizou-se da modelagem matemática do sistema através das relações geométricas entre câmera e objeto rastreado, no qual os eixos foram tratados individualmente, bem como foi determinada de maneira experimental a dinâmica de funcionamento dos servo motores utilizados para atuação dos eixos do sistema robótico.

Os eixos de movimentação *Pan* e *Tilt*, os quais correspondem aos eixos X e Y, respectivamente, são atuados por servo motores controlados no modo de velocidade. Estes eixos foram individualizados aplicando-se controladores PD digitais independentes para ambos, os quais foram projetados através do método para sintonia de controladores chamado de Lugar Geométrico das Raízes (LGR).

Diante da não linearidade da dinâmica do sistema, foram realizados testes práticos empregando os controladores projetados com ganho estático e com escalonamento de ganhos, onde o chaveamento dos diferentes ganhos foi realizado conforme três faixas distintas de erro. Ambos os sistemas de controle foram capazes de atingir o rastreamento do objeto, contudo, a utilização do escalonamento de ganhos se mostrou mais eficiente ao obterem-se respostas com menor erro em seu valor de regime, conforme os resultados apresentados.

Como dificuldades encontradas, pode-se mencionar o método de aquisição de dados e acionamento dos atuadores. Visto que o processo de aquisição da imagem, cálculo do erro e envio deste para o Arduino através da porta Serial apresentou um período de amostragem muito inconstante e superior ao calculado, variando de 20 a 40 milissegundos. Isso pode ter afetado negativamente o controle digital, uma vez que um período de amostragem em acordo com o sistema é fundamental seu bom funcionamento.

Além disso, em alguns dos experimentos realizados, notou-se a variação do erro após o mesmo atingir um valor próximo do seu valor de referência, isso ocorre devido ao processo de focalização realizado pela câmera, o qual gera pequenos ruídos na imagem, porém não apresentando interferência significativa

no sistema. Dessa maneira, com objetivo de eliminar ou reduzir estes ruídos, pode-se desabilitar o ajuste automático das configurações da câmera, como o foco automático e parametrizar seu valor de forma manual.

A inexistência de um equipamento adequado para realização dos testes de uma forma que garanta a repetibilidade das perturbações dificultou a aquisição dos resultados, uma vez que não se pode garantir com exatidão se as posições e velocidades foram idênticas nos testes realizados com ambos os métodos de controle.

Conforme os resultados obtidos, nota-se que para uma mesma perturbação aplicada, as respostas dos eixos *Pan* e *Tilt* apresentaram pequenas diferenças. Isso ocorre devido a influência do peso da câmera sobre o eixo *Tilt*, já que a modelagem utilizada não leva em consideração características mecânicas. Outra característica mecânica que também influenciou no funcionamento foi o atrito entre estrutura e eixo dos servo motores. A existência desse atrito resultou no surgimento de uma zona morta para velocidades baixas, em que o torque não é suficiente para movimentar o eixo em questão, impedindo o sistema de alcançar o erro nulo em regime permanente.

Como sugestões para trabalhos futuros, pode se utilizar um sistema embarcado em que a aquisição e processamento da imagem, bem como acionamento e controle dos atuadores seja realizado em um único microcontrolador, garantindo um período de amostragem mais preciso.

Visto a influência de características mecânicas, como peso da estrutura e atrito nas juntas, podem ser utilizadas modelagens matemáticas que incluam essas características, tratando ambos os eixos como um único sistema. Projetando assim controladores mais robustos e também considerando as não linearidades do sistema.

REFERÊNCIAS

[1] R. Krishna, "Computer vision: Foundations and applications", *Reference Book*, vol. 213, 2017.

[2] E. Trucco, A. Verri, *Introductory techniques for 3-D computer vision*, vol. 201, Prentice Hall Englewood Cliffs, 1998.

[3] P. I. Corke, *et al.*, *Visual Control of Robots: high-performance visual servoing*, Research Studies Press Taunton, UK, 1996.

[4] S. Hutchinson, G. D. Hager, P. I. Corke, "A tutorial on visual servo control", *IEEE transactions on robotics and automation*, vol. 12, no. 5, pp. 651–670, 1996.

[5] D. Y. Kikuchi, *Sistema de controle servo visual de uma câmera pan-tilt com rastreamento de uma região de referência.*, Tese de Doutorado, Universidade de São Paulo, 2007.

[6] A. R. d. Mello, *et al.*, "Sistema de inspeção visual de placas de circuito impresso para linhas de produção em pequenas séries em um contexto multiagentes", , 2015.

[7] K. Dawson-Howe, *A practical introduction to computer vision with opencv*, John Wiley & Sons, 2014.

[8] R. P. B. d. Menezes, "Controle servo visual de veículos aéreos multirrotores", , 2013.

[9] F. Bezerra Filho, M. Jácome, E. L. Villareal, "Aplicação do Controle Servo Visual em um Quadrirrotor", *Proceeding Series of the Brazilian Society of Computational and Applied Mathematics*, vol. 4, no. 1, 2016.

[10] N. Ruangpayoongsak, "Mobile robot positioning by using low-cost visual tracking system", *in MATEC Web of Conferences*, vol. 95, p. 08006, EDP Sciences, 2017.

[11] J. Ross, *Application of Feedforward Control to Pan-Tilt Cameras on Planetary Rovers*, Tese de Doutorado, Carleton University, 2014.

[12] P. Y. Oh, K. Allen, "Visual servoing by partitioning degrees of freedom", *IEEE Transactions on Robotics and Automation*, vol. 17, no. 1, pp. 1–17, 2001.

[13] M.-C. Tsai, K. Chen, M.-Y. Cheng, K.-C. Lin, "Implementation of a real-time moving object tracking system using visual servoing", *Robotica*, vol. 21, no. 6, pp. 615–625, 2003.

[14] J. E. Gang, *Color composition*, Rutgers The State University of New Jersey-New Brunswick, 2016.

[15] Open Source Computer Vision Documentation, "Open CV: Smoothing images", Online; Acesso em 28 de Setembro de 2021, 2021, URL: `https://docs.opencv.org/4.5.2/d4/d13/tutorial_p_y_filtering.html`.

[16] F. Chaumette, S. Hutchinson, "Visual Servoing 24 . Visual Servoing and Visual Tracking", , 2008.

[17] D. M. de Souza Lima[1], T. N. B. da Silva[1], O. R. Pinheiro, "CONTROLE SERVO VISUAL APLICADO A MANIPULADORES ROBÓTICOS", .

[18] A. C. Sanderson, L. E. Weiss, "Adaptive visual servo control of robots", *in Robot vision*, pp. 107–116, Springer, 1983.

[19] SparkFun Electronics, "Continuous Rotation Servo Trigger Hookup Guide", Online; Acesso em 30 de Setembro de 2021, 2016, URL: `https://learn.sparkfun.com/tutorials/continuous-rotation-servo-trigger-hookup-guide/all`.

[20] Parallax, "Parallax Continuous Rotation Servo", Online; Acesso em 01 de Outubro de 2021, 2011, URL: `https://docs.rs-online.com/870b/0900766b8123f8a1.pdf`.

[21] N. Pinckney, "Pulse-width modulation for microcontroller servo control", *IEEE Potentials*, vol. 25, no. 1, pp. 27–29, 2006, doi:10.1109/MP.2006.1635026.

[22] Pololu Robotics Electronics, "Spring RC SM-S4303R Continuous Rotation Servo", Online; Acesso em 01 de Outubro de 2021, 2021, URL:

https://www.tme.eu/Document/004105658202ef89808bba102ca4bf3f/POLOLU-1248.pdf.

[23] R. C. Monzani, "Controladores Analógicos e Digitais", , 2006.

[24] M. S. Fadali, A. Visioli, *Digital control engineering: analysis and design*, Academic Press, 2013.

[25] G. F. Franklin, J. D. Powell, M. L. Workman, *et al.*, *Digital control of dynamic systems*, vol. 3, Addison-wesley Reading, MA, 1998.

[26] Spring Model Electronics, "43R Servo (360° Rotation) Specification", Online; Acesso em 05 de Dezembro de 2021, 2021, URL: https://datasheetspdf.com/pdf/897796/SPRINGRC/SM-S4306R/1.

[27] J. A. Fayman, O. Sudarsky, E. Rivlin, "Zoom tracking", *in Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No. 98CH36146)*, vol. 4, pp. 2783–2788, IEEE, 1998.

ANEXO 1 - ALGORITMO DE AQUISIÇÃO E PROCESSAMENTO DA IMAGEM

Python

```python
# Importa bibliotecas

import cv2
import numpy as np
import serial
import datetime

# Configura Comunicacao Serial

porta = 'COM5'
baudRate = 115200

ligarArduino = True

if ligarArduino:
    SerialArduino = serial.Serial(porta, baudRate)

# Captura de video da webcam
webcam = cv2.VideoCapture(1)

# Loop principal
while 1:

    # Leitura do video obtido pela
    # webcam em imagens
    _, imageFrame = webcam.read()

    # Identifica dimensoes da imagem e posicao central
    alturaImagem, larguraImagem = imageFrame.shape[:2]

    centroX = int(larguraImagem / 2)
    centroY = int(alturaImagem / 2)
    centroImagem = (centroX, centroY)

    # Converte as imagens obtidas no espaco RGB para espaco HSV
    hsvFrame = cv2.cvtColor(imageFrame, cv2.COLOR_BGR2HSV)

    # Ranges da cor a ser detectada
    obj_lower = np.array([0, 92, 177], np.uint8)
    obj_upper = np.array([30, 248, 255], np.uint8)

    # Criacao da mascara de cor e aplicacao de filtragem mediana
    obj_mask_nof = cv2.inRange(hsvFrame, obj_lower, obj_upper)
    obj_mask = cv2.medianBlur(obj_mask_nof, 5)

    # Transformacao morfologica para detectar uma unica cor
    kernal = np.ones((5, 5), "uint8")

    obj_mask = cv2.dilate(obj_mask, kernal)
    res_obj = cv2.bitwise_and(imageFrame, imageFrame,
                              mask=obj_mask)

    # Deteccao de contornos
    contours, hierarchy = cv2.findContours(obj_mask,
                                           cv2.RETR_TREE,
                                           cv2.CHAIN_APPROX_SIMPLE)
    for pic, contour in enumerate(contours):
        area = cv2.contourArea(contour)
        if area > 320:
            x, y, w, h = cv2.boundingRect(contour)  # Posicao eixo x, Posicao eixo y, Largura, Altura
            imageFrame = cv2.rectangle(imageFrame, (x, y),
                                       (x + w, y + h),
                                       (255, 0, 0), 2)
            centroObjX = x + int(w / 2)
```

```
            centroObjY = y + int(h / 2)
            centroObj = (centroObjX, centroObjY)
            cv2.circle(imageFrame, centroObj, 2, 0, 0, 255)  # cv2.circle(Imagem a desenhar, coordenadas, raio, cor)

            cv2.putText(imageFrame, "OBJETO", (x, y - 5),
                        cv2.FONT_HERSHEY_SIMPLEX,
                        1.0, (255, 0, 0))

            # Calculo do erro
            erroX = centroX - centroObjX
            erroY = centroY - centroObjY

            # Envia os valores de erro para o Arduino
            if ligarArduino:
                SerialArduino.write(('X' + str(erroX) + 'Y' + str(erroY)).encode())

    # Visualiza imagem e encerramento do programa
    cv2.imshow("Sistema de Rastreamento de Objetos", imageFrame)
    cv2.imshow("Mascara", obj_mask)
    if cv2.waitKey(10) & 0xFF == ord('q'):
        SerialArduino.close()
        webcam.release()
        cv2.destroyAllWindows()
        break
```

ANEXO 2 - ALGORITMO DE ACIONAMENTO E CONTROLE DOS SERVO MOTORES

C++

```
// Bibliotecas utilizadas
#include <Servo.h>
#include <TimerThree.h>

// Constantes do Controlador PD sem filtro derivativo
volatile double KpX = 0.4;
volatile double KpY = 0.4;
volatile double Td = 0.1;

// Declaracao da constante de tempo de amostragem
volatile double ts = 0.005;

// Declaracao das variaveis do sinal de controle
volatile double uX;
volatile double uY;

// Declaracao das variaveis de erro
volatile double erroX;
volatile double erroX_ant = 0;

volatile double erroY;
volatile double erroY_ant = 0;

// Inicializacao dos servos, sem movimento
volatile double servoXSpeed = 1485;
volatile double servoYSpeed = 1500;

Servo servoX; // Servo Horizontal
Servo servoY; // Servo Horizontal

// Inicializacao do algoritmo
void setup() {
  Serial.begin(115200);                    // Habilita comunicacao Serial
  servoX.attach(9);                        // Pino para o PWM do servo horizontal
  servoY.attach(12);                       // Pino para o PWM do servo vertical
```

```
  servoX.writeMicroseconds(servoXSpeed);   // Ordena o servoX ficar parado
  servoY.writeMicroseconds(servoYSpeed);   // Ordena o servoY ficar parado

  Timer3.initialize(ts*1000000);           // Tempo de amostragem da interrupcao
  Timer3.attachInterrupt(controlePD);      // Nomeia a interrupcao de tempo
}

// Recebimento dos valores de erro
void loop() {

      if(Serial.available() > 0)
  {
    if(Serial.read() == 'X')
    {
      erroX = Serial.parseInt();
      if(Serial.read() == 'Y')
      {
        erroY = Serial.parseInt();
      }
    }
  }
}

// Interrupcao de tempo / Algoritmo de controle
void controlePD(){

  // Escalonamento de ganhos
  if(erroX > 100 || erroX < -100) KpX = 0.67;
  else if (erroX > 150 || erroX < -150) KpX = 0.9;
  else KpX = 0.4;

  if(erroY > 100 || erroY < -100) KpY = 0.67;
  else if (erroY > 150 || erroY < -150) KpY = 0.9;
  else KpY = 0.4;

  // Calculo dos sinais de controle
  uX  = 1/ts*(KpX*(Td+ts)*erroX - KpX*Td*erroX_ant);
  uY  = 1/ts*(KpY*(Td+ts)*erroY - KpY*Td*erroY_ant);

  // Envio do sinal PWM
  servoXSpeed = 1485 + uX;
  servoYSpeed = 1500 - uY;

  servoX.writeMicroseconds(servoXSpeed);
  servoY.writeMicroseconds(servoYSpeed);

  // Atualizacao das variaveis utilizadas para o controle
  erroX_ant = erroX;
  erroY_ant = erroY;
  }
```

DOUGLAS HANSEN

SISTEMA DE RASTREAMENTO DE OBJETOS ATRAVÉS DO CONTROLE SERVO VISUAL DE UMA CÂMERA PAN-TILT

Este trabalho foi julgado adequado para obtenção do título em Bacharel em Engenharia Elétrica, pelo Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina, e aprovado na sua forma final pela comissão avaliadora abaixo indicada.

Jaraguá do Sul, 11 de Fevereiro de 2022.

_____
Prof. Dr. Rodrigo Trentini Preuss
Orientador
IFSC – Campus Jaraguá do Sul – Rau

_____
Prof. Dr. Antonio da Silva Silveira
UFPA – Universidade Federal do Pará

_____
Prof. MSc Edilson Hipólito da Silva
IFSC – Campus Jaraguá do Sul – Rau

Comprovante de Assinatura Eletrônica

contraktor

Datas e horários baseados no fuso horário (GMT -3:00) em Brasília, Brasil
Sincronizado com o NTP.br e Observatório Nacional (ON)
Certificado de assinatura gerado em 16/02/2022 às 15:08:11 (GMT -3:00)

Termo de Aprovação - não assinado.pdf

ID única do documento: #1c1794fd-2b9c-45b8-ae54-6c4a6e450150

Hash do documento original (SHA256): 592df5c6706d1140af09417f9a8d7d2e3896204b0552278d3440045549db2572

Este Log é exclusivo ao documento número #1c1794fd-2b9c-45b8-ae54-6c4a6e450150 e deve ser considerado parte do mesmo, com os efeitos prescritos nos Termos de Uso.

Assinaturas (3)

- Rodrigo Trentini Preuss (Participante)
  Assinou em 16/02/2022 às 15:09:23 (GMT -3:00)
- Edilson Hipolito da Silva (Participante)
  Assinou em 17/02/2022 às 07:17:10 (GMT -3:00)
- Antonio da Silva Silveira (Participante)
  Assinou em 16/02/2022 às 15:37:59 (GMT -3:00)

Histórico completo

Data e hora	Evento
16/02/2022 às 15:08:15 (GMT -3:00)	Rodrigo Trentini Preuss solicitou as assinaturas.
16/02/2022 às 15:09:23 (GMT -3:00)	Rodrigo Trentini Preuss (Autenticação: e-mail rodrigo.trentini@ifsc.edu.br; IP: 190.102.49.168) assinou. Autenticidade deste documento poderá ser verificada em https://verificador.contraktor.com.br. Assinatura com validade jurídica conforme MP 2.200-2/01, Art. 10o, §2.

Data e hora	Evento
16/02/2022 às 15:37:59 (GMT -3:00)	Antonio da Silva Silveira (Autenticação: e-mail asilveira@ufpa.br; IP: 191.178.140.251) assinou. Autenticidade deste documento poderá ser verificada em https://verificador.contraktor.com.br. Assinatura com validade jurídica conforme MP 2.200-2/01, Art. 10o, §2.
17/02/2022 às 07:17:11 (GMT -3:00)	Documento assinado por todos os participantes.
17/02/2022 às 07:17:10 (GMT -3:00)	Edilson Hipolito da Silva (Autenticação: e-mail edilson.hipolito@ifsc.edu.br; IP: 179.223.196.202) assinou. Autenticidade deste documento poderá ser verificada em https://verificador.contraktor.com.br. Assinatura com validade jurídica conforme MP 2.200-2/01, Art. 10o, §2.