

Universidade Federal de Minas Gerais  
Escola de Engenharia  
Curso de Graduação em Engenharia de Controle e Automação

**Sistema Automático de Rastreamento  
Dinâmico de Alvos para Câmeras**

Luís Paulo Carvalho de Mendonça

Orientador: Prof. Guilherme Augusto Silva Pereira  
Supervisor: Luiz Fernando Etrusco Moreira

Belo Horizonte, Novembro de 2014

Universidade Federal de Minas Gerais  
Escola de Engenharia  
Curso de Graduação em Engenharia de Controle e Automação

## **Monografia**

# **Sistema Automático de Rastreamento Dinâmico de Alvos para Câmeras**

Monografia submetida à banca examinadora para avaliação curricular da disciplina Projeto Final de Curso II, como requisito para obtenção do título de bacharel em Engenharia de Controle e Automação.

Belo Horizonte, Novembro de 2014

# Agradecimentos

A todos os meus professores que me mostraram os caminhos do conhecimento que iria trilhar até me formar como engenheiro de controle e automação. Especialmente àqueles que souberam ensinar para a vida, e não somente para as provas.

Ao professor Guilherme Pereira e ao engenheiro Luiz Fernando Moreira, que toparam fazer parte do projeto. Sem supervisão e orientação, nem sempre a criatividade consegue distinguir no divertido e interessante o que é útil e viável.

Aos meus colegas de curso, que compartilharam comigo nos últimos anos os desafios, as alegrias e os aprendizados da graduação.

Aos que se dispuseram a escutar minhas exposições sobre cada detalhe do projeto para depois trazerem suas considerações e críticas construtivas, indispensáveis para o sucesso do trabalho.

Aos meus pais, demais familiares e amigos, que me apoiaram tanto nesses anos e me ajudaram a recordar a cada instante o que de mais valor tem a vida.

# Resumo

Este trabalho apresenta o desenvolvimento de um sistema automático de rastreamento de alvos móveis para câmeras DSLR. O sistema funciona através do controle de uma estrutura motorizada pan-tilt em malha fechada, com realimentação de informações de rastreamento visual. São detalhadas as etapas de projeto e construção de cada uma das partes do equipamento: a base motorizada, com suporte para a câmera e eixos pan e tilt direcionados por servomotores; o sistema embarcado, desenvolvido em plataforma Arduino, que aciona os servos a partir de informações recebidas do PC; e o software de controle, responsável pelo rastreamento visual do alvo nas imagens recebidas da câmera, por meio de filtro HSL, e pelo controle em malha fechada da base de forma a manter o objeto rastreado no centro da imagem. Para o projeto de controle, o processo foi modelado e controladores PI foram sintonizados com duas estratégias diferentes: o método empírico de Ziegler-Nichols para resposta ao degrau, e o ajuste baseado na análise do lugar das raízes. Testes de rastreamento foram realizados, com diferentes tipos de movimento do alvo, para comparar o desempenho dos controladores. O controlador baseado no lugar das raízes demonstrou mais facilidade para rastrear objetos rápidos, mas apresentou oscilação em estado estacionário. Concluiu-se que o sistema satisfaz aplicações gerais de rastreamento para velocidades angulares do alvo em relação à câmera de até  $0,3 \text{ rad/s}$ .

# Abstract

This work presents the development of an automatic moving target tracking system for DSLR cameras. The system works through the closed-loop control of a motorized pan-tilt structure, with visual tracking information feedback. The steps in the planning and construction of each part of the equipment are detailed: the motorized mount, which supports the camera and has the pan and tilt axes moved by servomotors; the embedded system, developed with the Arduino platform, which commands the servos based on information received from the PC; and the control software, responsible for the visual tracking of the target in the images received from the camera, through HSL filtering, and the closed-loop control of the mount so that the tracked object is kept in the center of the image. For the control design, a process model is identified and PI controllers were tuned by using two different strategies: the Ziegler-Nichols empirical method for step response, and the tuning based on the analysis of the root locus. Tracking tests were performed, with different target movements, so that the performance of the controllers could be compared. The one based on root locus performed better with fast tracked objects, but showed some oscillation on steady state. It was concluded that the system complies with general tracking applications for angular velocities of the target in relation to the camera up to 0,3 rad/s.

# Sumário

|  |             |
|--|-------------|
| <b>Agradecimentos</b>                                      | <b>i</b>    |
| <b>Resumo</b>  | <b>ii</b>   |
| <b>Abstract</b>  | <b>iii</b>  |
| <b>Sumário</b>   | <b>v</b>    |
| <b>Lista de Figuras</b>                                    | <b>vii</b>  |
| <b>Lista de Tabelas</b>                                    | <b>viii</b> |
| <b>1 Introdução</b>  | <b>1</b>    |
| 1.1 Motivação . . . . .                                    | 1           |
| 1.2 Objetivos . . . . .                                    | 3           |
| 1.3 O SARDAC . . . . .                                     | 3           |
| 1.3.1 Base motorizada . . . . .                            | 4           |
| 1.3.2 Sistema embarcado para acionamento da base . . . . . | 4           |
| 1.3.3 Software de controle . . . . .                       | 5           |
| 1.4 Relevância Técnica . . . . .                           | 6           |
| 1.5 Estrutura do Trabalho . . . . .                        | 7           |
| <b>2 Revisão Bibliográfica</b>                             | <b>9</b>    |
| 2.1 Sistema Pan-Tilt . . . . .                             | 9           |
| 2.1.1 GigaPan . . . . .                                    | 10          |
| 2.1.2 Swivl . . . . .                                      | 11          |
| 2.1.3 SFH-30 . . . . .                                     | 13          |
| 2.1.4 MPT2100-DS . . . . .                                 | 15          |
| 2.2 Controle Servo Visual . . . . .                        | 17          |
| 2.2.1 Aplicações práticas . . . . .                        | 17          |
| 2.2.2 Rastreamento visual . . . . .                        | 18          |
| 2.2.3 Técnicas de controle . . . . .                       | 19          |

|          |   |           |
|----------|---|-----------|
| 2.3      | Contribuições para o SARDAC . . . . .                 | 20        |
| 2.3.1    | Hardware . . . . .                                    | 20        |
| 2.3.2    | Software . . . . .                                    | 21        |
| <b>3</b> | <b>Desenvolvimento do Projeto</b>                     | <b>23</b> |
| 3.1      | Base Motorizada . . . . .                             | 23        |
| 3.1.1    | Projeto Mecânico e Construção . . . . .               | 23        |
| 3.1.2    | Especificação dos Requisitos de Acionamento . . . . . | 25        |
| 3.2      | Sistema Embarcado . . . . .                           | 29        |
| 3.3      | Software de Controle . . . . .                        | 31        |
| 3.3.1    | Implementação do Software . . . . .                   | 31        |
| 3.3.2    | Modelagem do Processo . . . . .                       | 33        |
| 3.3.3    | Projeto de Controle . . . . .                         | 37        |
| <b>4</b> | <b>Resultados</b>                                     | <b>44</b> |
| 4.1      | Rastreamento Visual . . . . .                         | 44        |
| 4.2      | Controle em Malha Fechada . . . . .                   | 45        |
| 4.2.1    | Simulador de Alvos . . . . .                          | 45        |
| 4.2.2    | Análise de Problemas . . . . .                        | 47        |
| 4.2.3    | Desempenho dos Controladores . . . . .                | 49        |
| <b>5</b> | <b>Conclusões</b>                                     | <b>58</b> |
| 5.1      | Desenvolvimento do SARDAC . . . . .                   | 58        |
| 5.2      | Trabalhos Futuros . . . . .                           | 60        |
|          | <b>Referências Bibliográficas</b>                     | <b>64</b> |

# Lista de Figuras

|      |   |    |
|------|---|----|
| 1.1  | Constituição geral do SARDAC. . . . .                                   | 2  |
| 1.2  | Base motorizada. . . . .  | 4  |
| 1.3  | Sistema embarcado Arduino . . . . .                                     | 5  |
| 1.4  | Tela de Live View do digiCamControl . . . . .                           | 6  |
| 2.1  | GigaPan EPIC Pro + tripé + câmera [10]. . . . .                         | 10 |
| 2.2  | Aplicações do Swivl [23]. . . . .                                       | 12 |
| 2.3  | Aplicações do SFH-30 [3]. . . . .                                       | 14 |
| 2.4  | Aplicações do MPT2100-DS [21]. . . . .                                  | 16 |
| 3.1  | Modelagem da base motorizada no SketchUp. . . . .                       | 24 |
| 3.2  | Peças da base motorizada. . . . .                                       | 25 |
| 3.3  | Situação hipotética para análise de requisitos do SARDAC. . . . .       | 26 |
| 3.4  | Conversão de distância focal em ângulo de visão. . . . .                | 26 |
| 3.5  | $\theta_A - \theta_S$ para aceleração máxima . . . . .                  | 28 |
| 3.6  | Modelagem da base motorizada no SolidWorks. . . . .                     | 29 |
| 3.7  | Vista superior da placa Arduino Uno [1] . . . . .                       | 30 |
| 3.8  | Módulo SARDAC. . . . .  | 32 |
| 3.9  | Dados de identificação no eixo Pan . . . . .                            | 35 |
| 3.10 | Resposta dos modelos de Pan para dados de identificação . . . . .       | 35 |
| 3.11 | Dados de identificação no eixo Tilt . . . . .                           | 36 |
| 3.12 | Resposta dos modelos de Tilt para dados de identificação . . . . .      | 36 |
| 3.13 | Representação da malha fechada de controle do SARDAC. . . . .           | 37 |
| 3.14 | Malha fechada com controlador PID [19]. . . . .                         | 38 |
| 3.15 | Método de Ziegler-Nichols para resposta ao degrau [19]. . . . .         | 38 |
| 3.16 | Lugar das raízes da malha de Pan . . . . .                              | 41 |
| 3.17 | Respostas ao degrau no distúrbio da saída no eixo x da imagem . . . . . | 41 |
| 3.18 | Lugar das raízes da malha de Tilt . . . . .                             | 42 |
| 3.19 | Respostas ao degrau no distúrbio da saída no eixo y da imagem . . . . . | 42 |
| 3.20 | Relação entre distância focal e deslocamento do alvo na imagem. . . . . | 43 |

|      |  |    |
|------|--|----|
| 4.1  | Ajuste manual do filtro HSL para rastreamento visual. . . . .            | 46 |
| 4.2  | Interface do programa simulador de alvos. . . . .                        | 47 |
| 4.3  | Histograma dos períodos de amostragem. . . . .                           | 48 |
| 4.4  | Tipo de deslocamento do alvo usado em cada teste. . . . .                | 50 |
| 4.5  | Ilustração do teste com trajetória em degraus. . . . .                   | 51 |
| 4.6  | Comparação de respostas ao distúrbio em degrau na malha de pan. . . . .  | 52 |
| 4.7  | Comparação de respostas ao distúrbio em degrau na malha de tilt. . . . . | 53 |
| 4.8  | Ilustração do teste com trajetória em rampas. . . . .                    | 55 |
| 4.9  | Comparação de respostas ao distúrbio em rampa na malha de pan. . . . .   | 56 |
| 4.10 | Comparação de respostas ao distúrbio em rampa na malha de tilt. . . . .  | 56 |
| 4.11 | Rastreamento em máxima velocidade. . . . .                               | 57 |

# Lista de Tabelas

|     |   |    |
|-----|---|----|
| 2.1 | Especificações do GigaPan EPIC Pro. . . . .                 | 11 |
| 2.2 | Especificações do SFH-30. . . . .                           | 14 |
| 2.3 | Especificações do MTP2100-DS. . . . .                       | 15 |
| 3.1 | Regra de Ziegler-Nichols para sintonia de PID [19]. . . . . | 38 |
| 5.1 | Custos envolvidos na construção do SARDAC. . . . .          | 59 |

# Capítulo 1

## Introdução

No atual cenário de avanços tecnológicos, as áreas de visão computacional e controle servo visual têm tido relativo destaque em aplicações práticas. Realidade aumentada; mosaicagem de imagens de satélites; utilização de câmeras no controle automático de veículos, são alguns exemplos dos recursos viabilizados por essas tecnologias. Inserido nessa realidade, este trabalho teve como proposta o desenvolvimento de um Sistema Automático de Rastreamento Dinâmico de Alvos para Câmeras (SARDAC).

O SARDAC, mostrado na Figura 1.1, foi concebido como uma solução completa, composto de *hardware* e *software*, que possibilita o rastreamento de alvos móveis por uma câmera, mantendo-os no campo de visão do equipamento e permitindo registrá-los em imagens fotográficas ou vídeo. O produto foi desenvolvido como uma plataforma básica de *tracking*, com uma grande variedade de aplicações. Porém, futuramente ele pode ser customizado para atender melhor a aplicações específicas, em que o alvo e o objetivo da captura das imagens sejam definidos.

### 1.1 Motivação

A escolha do tema para o Projeto de Final de Curso foi influenciada por um interesse especial por Fotografia. A posse de uma câmera DSLR Nikon D5100 e outros equipamentos fotográficos também poderia baratear significativamente um projeto que vinculasse essa área com tecnologias de Controle e Automação.

As primeiras análises da aplicação prática do SARDAC envolveram sua utilização no campo da Fotografia, possibilitando a realização de diversos efeitos e técnicas:

- Fotos de elementos com rápida ou constante movimentação;

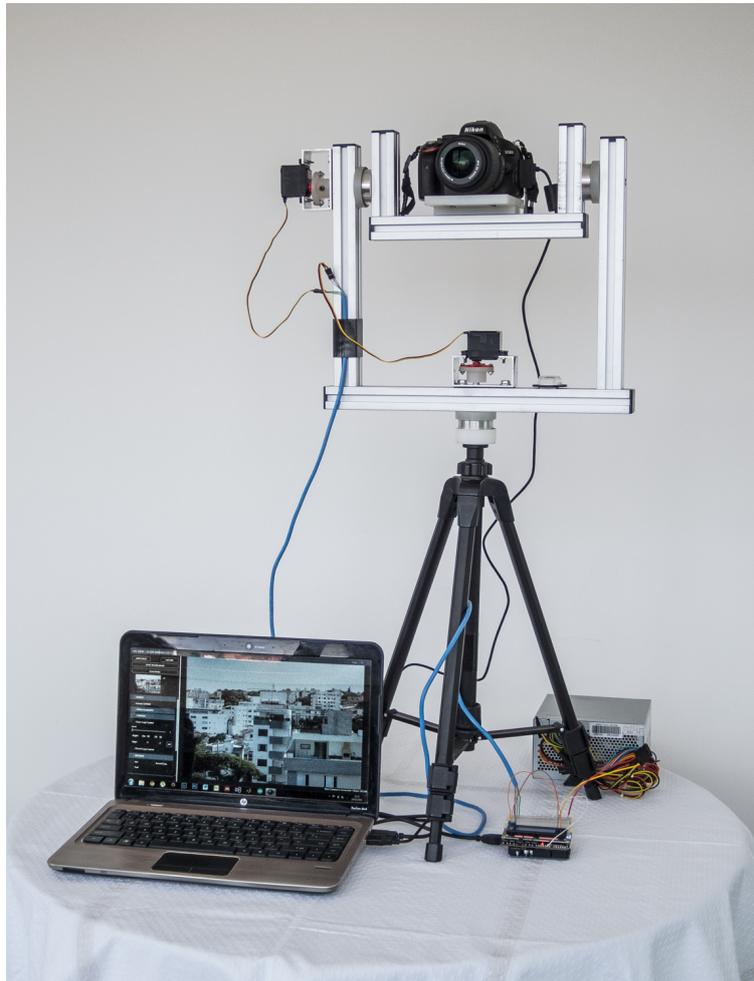


Figura 1.1: Constituição geral do SARDAC.

- *Panning*: foto de longa exposição em que a câmera acompanha o movimento do objeto, de forma que este aparece nítido e o fundo borrado;
- *Timelapse* (vídeo formado de uma sequência de fotos) com elemento fixo na imagem, como a lua, por exemplo.

Ainda que a motivação inicial do projeto tenha pouca relação direta com o meio industrial e o mercado de engenharia, podem ser identificadas diversas aplicações de maior interesse comercial, algumas das quais serão citadas na Seção 1.3.

## 1.2 Objetivos

O trabalho realizado tem como objetivo geral **o projeto e desenvolvimento de um protótipo funcional de um sistema pan-tilt motorizado com controle em malha fechada, por realimentação visual, do posicionamento da câmera, a fim de rastrear alvos móveis para captura de imagens e vídeos.**

Foram definidos também os seguintes objetivos específicos:

- Atingir um bom desempenho de controle para movimentos relativamente rápidos do alvo;
- Oferecer uma interface simples com o usuário para ajuste de parâmetros e configurações do sistema;
- Possuir uma estrutura mecânica resistente e durável.
- Permitir a adição de recursos de *hardware* e *software* posteriormente, ampliando as possibilidades de aplicação do sistema.
- Gerar uma documentação que possibilite a reprodução do projeto por outros, além de oferecer o software como *open-source*.

A próxima seção apresenta os elementos de *hardware* e *software* dos quais o sistema é composto.

## 1.3 O SARDAC

Além da câmera propriamente dita, o sistema é constituído por uma base motorizada para a câmera com dois graus de liberdade, um sistema embarcado para acionamento da base, e um *software* rodando em um PC para comunicação e controle remoto da câmera, captura e processamento de imagens, e controle da base motorizada de acordo com os critérios estabelecidos.

Após a definição dos objetivos principais do produto, foram definidos alguns requisitos qualitativos de projeto para cada parte do SARDAC, analisando suas características desejáveis e a viabilidade de cada aspecto. Essas primeiras especificações gerais nortearam o início do projeto, definindo um foco nas pesquisas bibliográficas e auxiliando no levantamento de materiais e custos. A seguir, cada um dos elementos do SARDAC será apresentado, juntamente com seus requisitos básicos. Ao longo do trabalho serão agregadas especificações mais detalhadas e quantitativas, definidas a partir da análise de produtos semelhantes, e do projeto detalhado de cada uma das partes que compõe o sistema.

### 1.3.1 Base motorizada

A base motorizada, melhor visualizada na Figura 1.2, é o mecanismo de movimentação da câmera. Ela possui dois graus de liberdade rotacionais, cada um movimentado por um servomotor. Cada grau de liberdade oferece uma movimentação da câmera referente a uma das dimensões da imagem capturada (2D).



Figura 1.2: Base motorizada.

#### Requisitos Gerais:

- Estrutura leve, resistente e de fácil manuseio.
- Dois graus de liberdade, um para tilt e outro para pan.
- Acionamento que dispense controle de potência.
- Ajuste de posicionamento da câmera, a fim de eliminar os torques estáticos devidos à gravidade no eixo de tilt e de reduzir os momentos de inércia.
- Fixação em tripé ou superfície estável.
- Condução interna de fios.

### 1.3.2 Sistema embarcado para acionamento da base

Para que os motores da base possam ser controlados via software pelo computador, é necessário um sistema embarcado para acionamento dos mesmos a partir de informações recebidas do PC. Este sistema foi desenvolvido com um

Arduino Uno, um microcontrolador de placa única desenvolvido como *hardware open-source* [1]. Foi ainda utilizado um *shield* (placa que se encaixa no Arduino para estender suas funcionalidades) LCD Keypad para controle manual da base. O sistema é mostrado na Figura 1.3, ainda em fase de teste, separado da base motorizada.

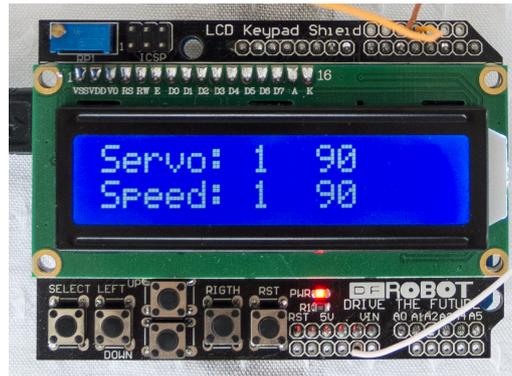


Figura 1.3: Sistema embarcado Arduino

#### Requisitos Gerais:

- Utilização de plataformas eletrônicas pré-fabricadas.
- Possibilidade de operação em malha aberta.
- Fixação juntamente à base motorizada.
- Taxa de processamento que não prejudique o desempenho de controle do sistema.

#### 1.3.3 Software de controle

As funções de controle e processamento de imagens do SARDAC foram desenvolvidas como extensão do *digiCamControl* [6], um *software open-source* para Windows que oferece diversas funcionalidades de comunicação e controle remoto para câmeras DSLR (*Digital Single-Lens Reflex*). Na Figura 1.4 é mostrada a tela de Live View do *software*, que apresenta em tempo real a visão da câmera recebida via USB, e permite diversos ajustes remotos e a captura de fotos e vídeo.

Foi implementado na tela de Live View um módulo próprio do SARDAC. Esse módulo inclui, basicamente, o processamento de imagens e o controle do sistema. A primeira função processa cada quadro recebido da câmera a fim de identificar o alvo e rastrear seu centroide. A segunda função, em caso de

controle em malha fechada, calcula periodicamente a distância do centroide para o ponto desejado na imagem, e a partir disso envia para o Arduino um comando de reposicionamento dos servos. Foram implementadas duas malhas de controle independentes, uma para cada dimensão da imagem.

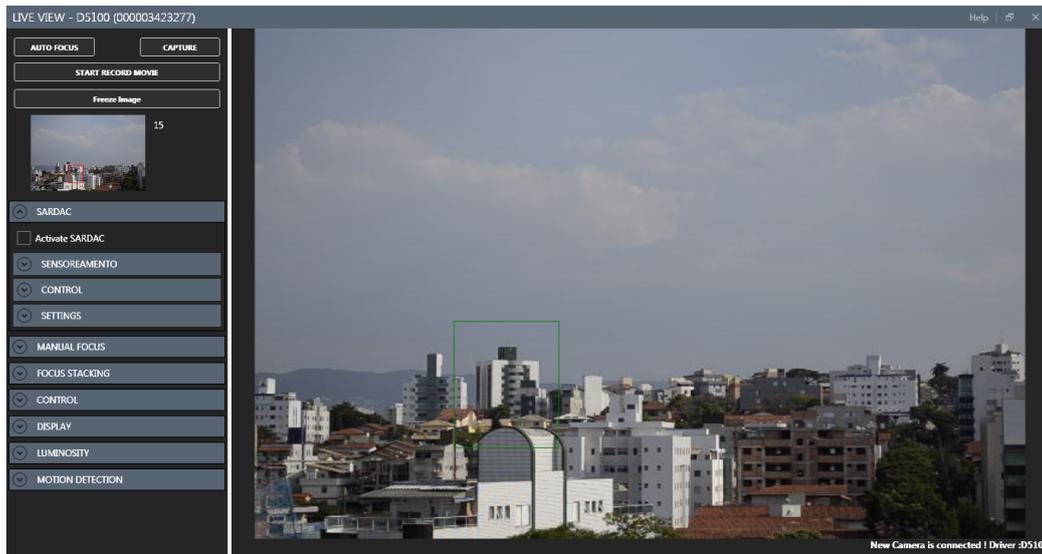


Figura 1.4: Tela de Live View do digiCamControl

### Requisitos Gerais:

- Aproveitamento de *software open-source*, se possível. Se necessário, funções de comunicação com a câmera poderiam ser programadas em mais baixo nível com SDK (*Software Development Kit*) da Nikon.
- Utilização dos recursos de visão computacional da câmera para rastreamento, se possível.
- Possibilidade de controle em malha aberta ou fechada.

## 1.4 Relevância Técnica

Podemos identificar diversas aplicações comerciais e industriais para o SARDAC, dentre elas:

- Câmeras de segurança. Ex.: rastreamento de pessoas e outros elementos móveis num ambiente, possibilitando cobrir uma maior área de vigilância.

- Câmeras embarcadas em VANTs ou outros veículos. Ex.: monitoramento de alvos em solo por VANTs militares; Cobertura de grandes eventos utilizando VANTs.
- Monitoramento de peças/equipamentos móveis em processos produtivos. Ex.: rastreamento de produtos em esteiras; acompanhamento de máquina móvel.
- Identificação de “corpos estranhos” em ambiente industrial. Ex.: rastreamento de produtos fora do padrão; identificação de pessoas em local não autorizado.

Como será visto no Capítulo 2, já se encontram soluções comerciais para controle de câmera utilizando sistema pan-tilt. Por serem equipamentos especializados e com poucas opções no mercado, a maioria tem um preço alto, variando entre US\$1.000,00 e US\$10.000,00, dependendo do nível de especialização, recursos adicionais e nicho de mercado almejado. Aqueles produtos mais baratos que isso possuem baixo desempenho ou recursos limitados. Além disso, a maior parte dos produtos pesquisados apresenta somente a opção de controle em malha aberta, com a utilização de um *joystick* ou outros controles manuais.

O trabalho desenvolvido pretende, portanto, oferecer uma opção de sistema pan-tilt com um controle de posicionamento em malha fechada. Também espera-se obter um baixo custo em relação aos preços de mercado, utilizando uma estrutura eletro-mecânica mais simples, lançando mão de plataformas *open-source* de *hardware* e *software*, definindo o rastreamento de alvos como a aplicação específica do equipamento, e dispensando recursos que não sejam relevantes para tal aplicação.

## 1.5 Estrutura do Trabalho

Neste capítulo foi apresentada uma síntese do trabalho desenvolvido e cada um de seus componentes. Vimos também os objetivos do projeto que nortearam a realização do trabalho e algumas especificações gerais inicialmente definidas.

No Capítulo 2 será apresentada uma revisão bibliográfica afim com este projeto, e como as referências puderam servir de parâmetro no desenvolvimento e testes do SARDAC. Também serão identificados alguns dos produtos atualmente disponíveis no mercado que possuem alguma semelhança com o trabalho realizado.

O desenvolvimento do projeto é documentado no Capítulo 3. O trabalho é dividido em etapas, correspondentes aos elementos do sistema indicados na Seção 1.3, e são apresentadas a metodologia e a teoria utilizadas no planejamento e construção de cada uma delas.

O Capítulo 4 compreende os testes e validações, seus resultados e uma discussão dos mesmos. Tais resultados permitiram avaliar e comparar o desempenho de diferentes métodos de controle, o devido funcionamento dos componentes eletroeletrônicos e mecânicos do sistema, a comunicação entre equipamentos, a configuração de parâmetros e funcionalidades através do computador, etc.

O Capítulo 5 conclui o trabalho, analisando o cumprimento dos objetivos propostos inicialmente, identificando as dificuldades surgidas ao longo do processo e sugerindo focos de melhoria para futuros trabalhos. O posicionamento do SARDAC frente aos outros produtos disponíveis no mercado também é avaliada.

# Capítulo 2

## Revisão Bibliográfica

Esse capítulo apresenta um levantamento de algumas das soluções de *hardware* e *software* já desenvolvidas nas áreas de sistemas pan-tilt e controle servo visual. A análise de cada solução permitiu definir critérios mais concretos para o desenvolvimento do SARDAC e boa parte das especificações e requisitos de cada uma das partes do equipamento. A Seção 2.1 foca no *hardware*, apresentando alguns sistemas pan-tilt e um estudo da relevância de suas características no contexto do SARDAC e da viabilidade de implementação das mesmas. Já a Seção 2.2 foca no *software*, analisando trabalhos acadêmicos para identificar algumas aplicações práticas de controle servo visual e as principais técnicas utilizadas para rastreamento visual e para o controle de robôs a partir do processamento de imagens. A Seção 2.3 sintetiza os elementos de *hardware* e *software* apontados nas seções anteriores que foram escolhidos para integrar o SARDAC, além de relacionar as características que, apesar de não terem feito parte do projeto, podem agregar valor ao produto em implementações futuras.

### 2.1 Sistema Pan-Tilt

Para auxiliar no projeto da estrutura e acionamento do SARDAC, foram analisados produtos comercializados que pudessem servir como referência para o trabalho. A seguir serão apresentados quatro produtos e suas possíveis contribuições para o desenvolvimento do SARDAC. Eles foram escolhidos por apresentarem propostas distintas, seja em relação ao formato e construção, à utilização ou à estratégia de controle, de forma que a análise do conjunto pudesse gerar diversas opções para a linha inicial do projeto.



Figura 2.1: GigaPan EPIC Pro + tripé + câmera [10].

### 2.1.1 GigaPan

A empresa GigaPan oferece soluções automáticas para captura e mosaicagem de imagens, a fim de obter panoramas com resoluções da ordem de gigapixels. O produto consiste numa base robótica pan-tilt com suporte para câmeras, e num *software* que une as fotografias capturadas para formar uma imagem panorâmica de alta resolução. O site do produto [10] também oferece uma galeria própria para os usuários compartilharem suas imagens, que podem ser visualizadas de forma interativa. Existem três modelos diferentes do GigaPan, cada um voltado para uma categoria diferente de câmera. O modelo utilizado como referência foi o GigaPan EPIC Pro, mostrado na Figura 2.1, que possui uma estrutura mais robusta e suporta câmeras DSLR. Tal modelo é vendido por US\$995,00.

O GigaPan é inicialmente calibrado para a câmera e distância focal utilizados, a partir do deslocamento angular necessário para que uma referência estática (ponto ou linha, e.g. linha do horizonte) percorra toda a imagem. O usuário, em seguida, aciona manualmente os eixos de rotação até os pontos extremos da imagem final, e o equipamento inicia então a série de capturas,

posicionando automaticamente a câmera nos devidos quadrantes e comandando os disparos através de um cabo. A base do equipamento pode ser encaixada no topo de um tripé ou outro suporte por meio de uma rosca. Há também um ajuste de posicionamento da câmera em relação ao eixo de rotação de tilt, eliminando os torques estáticos no motor e alinhando o obturador com o eixo. Algumas especificações do produto são apresentadas na Tabela 2.1.

|                    |                               |
|--------------------|-------------------------------|
| Massa com bateria  | 3,3Kg                         |
| Carga máxima       | 4,5Kg                         |
| Faixa de movimento | pan: 360°, tilt: -65° / + 90° |
| Precisão           | pan: 0,12°, tilt: 0,04°       |
| Dimensões          | 271 × 302 × 150mm             |
| Alimentação        | Bateria 7,2V                  |

**Tabela 2.1: Especificações do GigaPan EPIC Pro.**

Os pontos fortes e fracos do GigaPan EPIC Pro são os seguintes:

#### Pontos Positivos

- Capacidade de carga relativamente alta.
- *Design* simétrico que oferece uma maior estabilidade à estrutura.
- Permite ajuste de posicionamento da câmera em relação ao eixo de tilt.
- A operação a partir de bateria permite a utilização em locais isolados.
- Pode ser fixado em tripé.

#### Pontos Negativos

- Tem uma aplicação bem específica, não é possível adaptar o equipamento para *tracking* ou outros fins.
- Possui muitas partes de fabricação própria, como peças de plástico injetadas.
- Possui dinâmica lenta, já que na aplicação de mosaicagem o foco é na precisão do posicionamento e não na velocidade.

### 2.1.2 Swivl

O Swivl é produzido pela empresa homônima [23], e, dos quatro produtos revisados, é o mais portátil e acessível. O equipamento foi desenvolvido com foco na captura de vídeo a partir de *smartphones* e *tablets*, apesar do



Figura 2.2: Aplicações do Swivl [23].

novo modelo possuir uma maior capacidade de carga, possibilitando até o acoplamento de câmeras DSLR. Algumas de suas aplicações são ilustradas na Figura 2.2, e dentre elas podemos citar a gravação de vídeo-aulas e realização de videoconferências. É vendido por aproximadamente US\$200,00.

O sistema pan-tilt do Swivl é encapsulado em um corpo compacto em forma de disco. O principal recurso do equipamento é o rastreamento do usuário durante a captura de vídeo, mas isso não é realizado através de processamento de imagem: o usuário precisa carregar um pequeno transmissor cujo sinal é rastreado pela base. A restrição do movimento é de 360° para pan e 30° para tilt, e a distância entre a base e o usuário deve ser de no máximo 10m. O transmissor também é responsável por captar a voz do usuário e possui alguns controles para o vídeo, desde que o dispositivo de captura tenha o aplicativo proprietário instalado. No caso em que uma câmera fotográfica for acoplada ao Swivl, o dispositivo iOS ou Android pode ser usado como central de controle da base e da câmera. O equipamento pode ser alimentado através de baterias ou na rede elétrica.

Em análise da relevância do Swivl para o desenvolvimento do SARDAC, foram feitas as seguintes observações:

**Pontos Positivos**

- O recurso de *tracking* a partir da detecção de um transmissor oferece uma abordagem mais simples que a visão computacional.
- Controle remoto da base a partir de dispositivos móveis, como *smartphones* e *tablets*. É um recurso interessante para controle de posicionamento da câmera em malha aberta.
- Possibilidade de operar pela rede elétrica ou através de baterias. Isso permite que o sistema seja utilizado em ambientes remotos.
- É compacto e tem baixo custo.

**Pontos Negativos**

- Pelas dimensões do Swivl e os vídeos comerciais do produto, supõe-se que a dinâmica do sistema seja lenta.
- A estrutura do equipamento não possibilita o ajuste de posicionamento da câmera.
- A necessidade de fixar o transmissor ao alvo restringe bastante a aplicação do produto, não sendo possível a utilização para rastreamento de alvos variáveis (e.g. câmera de segurança), distantes (e.g. aviões), perigosos (e.g. animais selvagens), etc.

**2.1.3 SFH-30**

Produzido pela Mark Roberts Motion Control [3], o SFH-30 é o mais avançado dos quatro sistemas pan-tilt avaliados neste trabalho. Ele foi desenvolvido para ser usado com câmeras fotográficas DSLR e filmadoras profissionais, como se pode ver na Figura 2.3. Algumas de suas aplicações são: *live action*, eventos, esportes, animação em *stop motion* e fotografia *time lapse*. Ele é vendido por aproximadamente US\$9.000,00 em sua versão mais simples.

O SFH-30 pode ser controlado através de diversas interfaces (RS232, RS422, Ethernet, Wireless) e dispositivos (*pan bars*, console com *joystick*, *handwheels*, iPhone com aplicativo proprietário, PC com *software* proprietário). O equipamento pode funcionar com a base fixada em superfície embaixo ou em cima da câmera. A Tabela 2.2 mostra outras especificações do produto disponibilizados pelo fabricante.

A grande precisão do SFH-30 e controlabilidade oferecida pelo software proprietário permitem um detalhado planejamento do movimento e repetibilidade do mesmo. O equipamento é bastante customizável com itens opcionais: tipo do motor (servo ou motor de passo), *encoders* e outros sensores embarcados, eixos de acionamento opcionais (translação vertical e horizontal

|                   |                      |
|-------------------|----------------------|
| Massa             | 5Kg                  |
| Carga máxima      | 13Kg                 |
| Velocidade máxima | 100°/s               |
| Aceleração máxima | 100°/s <sup>2</sup>  |
| Dimensões         | 261 × 263 × 135mm    |
| Alimentação       | 24V CC / 110-240V AC |
| Exatidão          | < 0,01°              |
| Resolução         | < 0,0001°            |

Tabela 2.2: Especificações do SFH-30.



Figura 2.3: Aplicações do SFH-30 [3].

em trilho, acionamento de zoom e foco). Apesar de ser um produto relativamente avançado, o SFH-30 não oferece nenhuma solução de *tracking* em malha fechada. A seguir são destacadas algumas observações relevantes sobre os pontos fortes e fracos do equipamento:

#### Pontos Positivos

- O SFH-30 possui um *design* robusto e um alto desempenho e precisão.
- O produto é uma solução bastante completa e se adapta a diversos nichos de mercado.
- Permite a utilização de *joystick* e diversos outros controles manuais para controle do sistema em malha aberta.

**Pontos Negativos**

- Possui alto custo e peso excessivo.
  - Diversos recursos que oferece não são relevantes para o SAR-DAC ou apresentariam grande
- dificuldade de implementação, por exemplo as diversas interfaces de comunicação e os eixos adicionais de acionamento.
- Não possui controle automático de posicionamento em malha fechada.

**2.1.4 MPT2100-DS**

O sistema pan-tilt MPT2100-DS é fabricado pela ServoCity [21], e se apresenta como uma solução intermediária em comparação com os dois produtos anteriores. Não possui tantos recursos quanto o SFH-30, mas é mais robusto que o Swivl, suportando uma maior carga. Ele é vendido no site do fabricante por aproximadamente US\$1.400,00.

Como pode ser observado na Figura 2.4, uma das características do equipamento é seu *design* simples e rústico. Ele é construído em alumínio, com rolamentos dos eixos em aço inoxidável. As peças ligadas ao eixo de tilt possuem ajustes de balanço da câmera, e os motores são ligados às engrenagens dos eixos por correias dentadas de kevlar. A Tabela 2.3 mostra algumas especificações do produto.

|              |                |
|--------------|----------------|
| Massa        | 5Kg            |
| Carga máxima | 18Kg           |
| Velocidade   | 0,5°/s - 20°/s |
| Alimentação  | 12V CC         |

**Tabela 2.3: Especificações do MTP2100-DS.**

O MPT2100-DS é controlado em malha aberta a partir de uma central de controle com *joystick* ligado à base através de um cabo Ethernet CAT6. Porém, é possível conectar os servos em um receptor de sinal de rádio para serem operados por controle remoto. Pode-se também montar o sistema com uma peça adicional, adicionando um grau de liberdade para movimento de *roll*. Os servos também podem ser trocados por outros que possibilitam uma maior velocidade em detrimento do torque. Apesar do sistema pan-tilt oferecer apenas o controle em malha aberta, outros produtos da ServoCity possuem o controle em malha fechada com referência externa de posição. Seguem algumas observações a respeito do produto:



Figura 2.4: Aplicações do MPT2100-DS [21].

#### Pontos Positivos

- É montado a partir de perfis de alumínio pré-fabricados. O *design* favorece uma construção simples e rápida, reduzindo a necessidade de peças usinadas. Os cortes no perfil são úteis para o encaixe de uma peça na outra e montagem dos motores e engrenagens.
- Supõe-se que o uso de correias dentadas para ligar os motores às engrenagens dos eixos ameniza problemas de histerese e desencaixe em função da falta de precisão no posicionamento dos motores.
- O *design* simétrico do sistema oferece maior equilíbrio em movimentos de *panning*, principalmente se ele estiver montado sobre tripés.

#### Pontos Negativos

- Sua estrutura é rústica e formada por um número maior de peças.
- Não possui controle automático de posicionamento em malha fechada.

## 2.2 Controle Servo Visual

Desde o início da década de 80, são realizadas pesquisas na área de controle servo visual (conceito sugerido pela primeira vez em 1979 [12]), ou seja, o acionamento de estruturas robóticas em malha fechada usando a captura e processamento de imagens para fornecer informações de realimentação. A aplicabilidade dessa área de pesquisa é bastante diversa, e as tecnologias desenvolvidas ao longo desses anos podem ser encontradas na indústria, no mercado de vigilância e segurança, no ramo militar, em produções de fotografia e vídeo, e no próprio meio científico e acadêmico.

Uma aplicação de controle servo visual pode ser dividida, basicamente, em duas tarefas: a captura de imagens e o devido processamento para extrair dela as informações desejadas, e a interpretação das informações por um algoritmo de controle. A seguir é feito um breve estudo das estratégias utilizadas para cada uma dessas tarefas, e também são apresentadas algumas aplicações específicas de pesquisas na área.

### 2.2.1 Aplicações práticas

Foram citadas algumas áreas em que tecnologias de controle servo visual podem ser aplicadas. Encontrou-se na bibliografia aplicações bem específicas, que ajudam a ampliar as possibilidades de uso do SARDAC. Em [4], por exemplo, os autores buscam solucionar o problema de instabilidade na aquisição de imagens em observações submarinas. O uso de suportes motorizados para operar câmeras em ambientes aquáticos pode resultar em movimentos indesejados de baixa frequência causados pelas correntes marinhas. Estratégias de controle servo visual são utilizadas para estabilizar a imagem a partir do acionamento do suporte pan-tilt em malha fechada.

Em [8], o foco é no estudo do comportamento de insetos voadores. Câmeras pan-tilt foram controladas para seguirem o inseto em *close-up* ao longo de sua trajetória de voo. As imagens foram processadas a fim de se calcular a trajetória do animal em coordenadas 3D do mundo, além de definir sua orientação ao se aproximar de alimentos.

Os autores de [2] desenvolveram um mecanismo para fornecer informações visuais em sistemas de direção assistida para veículos. Com o objetivo de monitorar o veículo à frente, um sistema de controle para uma câmera pan-tilt-zoom foi implementado de forma a manter estável em escala e posição a visão da traseira do veículo.

Um sistema multifuncional de rastreamento visual em tempo real é apresentado em [26], sendo testado para monitoramento de tráfego veicular, detecção de criaturas marinhas e detecção e rastreamento de invasores. Em

[15], o trabalho desenvolvido é vinculado à análise do movimento humano. Um conjunto de câmeras pan-tilt são controladas para rastrear uma pessoa vestindo uma roupa com marcadores pontuais, de forma a reproduzir automaticamente seus movimentos em ambiente virtual 3D.

### 2.2.2 Rastreamento visual

Em [16], os autores explicam que os métodos de rastreamento em imagens podem ser separadas em dois enfoques que são fundamentalmente diferentes:

- **Rastreamento baseado em reconhecimento:** O objeto é reconhecido em uma série de imagens a partir de critérios pré-definidos, e sua posição é extraída. Pode ser realizado em três dimensões, e tanto a translação quanto a rotação do objeto podem ser estimados. Possui as desvantagens de apenas rastrear objetos reconhecíveis, e possuir algoritmos de maior custo de execução.
- **Rastreamento baseado em movimento:** O objeto alvo é rastreado a partir de seu movimento em relação ao plano de fundo, independentemente de seu tamanho ou forma. Técnicas baseadas em movimento ainda podem ser divididas em métodos de fluxo óptico ou de energia de movimento, ambos tratados em [16].

O rastreamento por reconhecimento apresenta uma maior diversidade de técnicas sugeridas por vários autores. Alguns dos critérios identificados na bibliografia para reconhecimento do alvo são os seguintes:

- **Cor:** Em [27] são realizadas detecções de faces a partir de padrões pré-definidos de cor de pele mapeados no espaço RGB.
- **Fundo constante:** [11] propõe o mapeamento prévio do ambiente, correspondente ao plano de fundo da câmera. A identificação do alvo se dá então pela subtração entre a imagem recebida e imagem armazenada para o ambiente.
- **Contornos:** Sabendo-se o formato do objeto, utilizam-se funções paramétricas para buscar o alvo correspondente na imagem. Em [5] a parametrização de retas é usada para detectar linhas de estradas para a navegação de veículos robóticos aéreos.
- **Região de referência:** Em uma imagem inicial é definida uma região que contém o alvo a ser rastreado, e o valor de seus pixels é armazenado. Nas imagens seguintes é realizada uma varredura de janelas até

encontrar aquela que mais se assemelha à região de referência. Esse é o método utilizado em [14], e o autor aponta que normalmente a função de correlação utilizada para indicar a semelhança entre janela e região de referência é a soma dos quadrados das diferenças (SSD). O algoritmo de varredura busca minimizar essa função. Em [25] os autores também utilizam a técnica, mas propõe um novo algoritmo de busca hierárquica para minimizar o número de janelas utilizadas na varredura.

### 2.2.3 Técnicas de controle

Uma vez que as informações de posição do alvo na imagem foram obtidas, precisam ser tratadas por um algoritmo de controle para que ele possa comandar o acionamento do robô a fim de cumprir certos requisitos. Em [14] são apresentadas duas classificações distintas para sistemas de controle servo visual no contexto de câmeras pan-tilt:

- **Estrutura de controle hierárquica ou não-hierárquica:** Na estrutura hierárquica, a malha de controle possui dois laços de realimentação: um externo realimentado pela posição do alvo na imagem, e um interno para controle de posição dos motores do robô. Numa estrutura não-hierárquica não há malha de controle interna. Como a taxa de amostragem necessária para controle de posição de motores é muito maior que as taxas comumente utilizadas em aplicações de controle servo visual, e como a maioria dos robôs apresentam controladores próprios de posicionamento, a estrutura hierárquica é mais utilizada.
- **Controle baseado em posição ou em imagem:** Para o controle baseado em posição, as informações visuais são usadas em conjunto com modelos do alvo e da câmera para gerar a posição do alvo no espaço 3D, e essa informação será passada ao controlador. No controle baseado em imagem, as próprias informações de rastreamento do alvo na imagem em 2D são fornecidas ao controlador. O posicionamento de manipuladores robóticos com vários graus de liberdade normalmente requer a informação do trajeto do alvo nas três dimensões. Para uma câmera pan-tilt, com apenas dois graus de liberdade, o controle baseado em imagem é suficiente, além de apresentar menor custo computacional.

O método mais simples para fechar a malha de controle servo visual é a utilização de um modelo da câmera para calcular as velocidades de rotação pan e tilt a partir da posição do alvo na imagem. Essa e outras estratégias *look-and-move*, como chama o autor em [14], são mais simples mas têm a desvantagem de considerar apenas a última posição conhecida do alvo para o

controle. Quando o alvo se encontra em movimento, essa informação sempre estará defasada. Para resolver este problema, em [27] um modelo cinemático do alvo é usado para prever sua posição futura. Em [20] a posição do alvo é estimada através de um Filtro de Kalman. Diversas opções de predição são apresentadas em [13] para rastreamento em aplicações de controle servo visual, incluindo o Filtro de Kalman para predição um passo à frente e N passos à frente.

Em [20] é usado um modelo linear de movimento projetado sobre a imagem baseado no fluxo óptico, incluindo deslocamentos do alvo, da câmera, e ruídos. Para o controle são testados controladores proporcional e integral (PI), adaptativo e linear quadrático gaussiano (LQG). O autor em [14] segue a mesma metodologia, comparando um controlador PI e um proporcional com estimação de perturbações externas (deslocamento do alvo). No primeiro caso, o critério de determinação do controlador é linear quadrático (LQ), e no segundo caso a estimação é feita com um Filtro de Kalman, resultando num controlador LQG.

## 2.3 Contribuições para o SARDAC

O estudo de sistemas pan-tilt comercializados e de projetos de pesquisa voltados para controle servo visual permitiu definir os detalhes e requisitos específicos do projeto, complementando a especificação inicial apresentada no Capítulo 1. A seguir são apresentados os elementos extraídos da revisão bibliográfica para aplicação no desenvolvimento do SARDAC. Eles estão divididos pela natureza da aplicação, hardware ou software, e pela prioridade do desenvolvimento, ou seja, se a característica é importante para o cumprimento do objetivo principal do SARDAC e fez parte do desenvolvimento inicial, ou se ela, apesar de viável, cumpre objetivos secundários e pode ficar para aplicações futuras para agregar novas funcionalidades ao produto.

### 2.3.1 Hardware

#### Aplicações iniciais

Foram identificadas nos produtos pesquisados as seguintes características com relevância para o desenvolvimento do *hardware* SARDAC:

- Construção da estrutura da base utilizando perfis de alumínio pré-fabricados. Isso permite um peso reduzido e facilidade na montagem.

- A base motorizada será projetada com design simétrico, assim como o GigaPan ou o MPT2100-DS, para aumentar a estabilidade durante a movimentação e reduzir o estresse nos eixo de rotação.
- A estrutura mecânica será dimensionada para cargas menores que o MPT2100-DS e o SFH-30, possibilitando uma estrutura mais leve, uma dinâmica mais rápida do sistema, e a utilização de servomotores menores e mais baratos.

### Aplicações futuras

O SARDAC é um produto que oferece diversas possibilidades de desenvolvimentos futuros. Dos recursos de *hardware* encontrados na pesquisa, alguns deles merecem uma atenção nesse sentido:

- Adição de recursos para controle em malha aberta através de *joystick*, *smartphone*, botões ligados ao Arduino, etc.
- Adequação do sistema para que os servos possam operar também com alimentação de baterias.

### 2.3.2 Software

A análise das diversas aplicações de controle servo visual encontradas na literatura permitiu identificar técnicas de rastreamento visual e controle adequadas para o SARDAC. Buscou-se aquelas técnicas que não fossem demasiadamente complexas, mas que pudessem ser compatíveis com o *software open-source* a ser utilizado e com a base pan-tilt a ser construída, de forma que a capacidade físico-motora do sistema não ficasse subutilizada. A seguir são apontadas as técnicas de controle servo visual relevantes para o SARDAC, tanto de imediato quanto para implementações futuras.

### Aplicações iniciais

- Pela simplicidade e eficácia para aplicações diversas em ambiente controlado, o rastreamento do alvo na imagem do SARDAC será baseado em reconhecimento, usando o critério de cor do alvo.
- Como serão utilizados servomotores com controle interno de posição, a estratégia de controle a ser utilizada é hierárquica.
- O mecanismo pan-tilt do SARDAC é simples e o controle poderá ser baseado em imagem.

- Será projetado um controle PI.

#### **Aplicações futuras**

- Outras estratégias de rastreamento visual poderão ser implementadas para ampliar as possibilidades de atuação do SARDAC.
- Implementação de controle adaptativo para melhor adequação a alvos diferentes, com movimentos diversos.
- Utilização de Filtro de Kalman ou outras estratégias de predição para compensar o atraso na captura de imagens.

O próximo capítulo apresenta o desenvolvimento de cada parte do SARDAC, descrevendo a teoria, as ferramentas e os métodos utilizados para garantir a implementação das características definidas até o momento.

# Capítulo 3

## Desenvolvimento do Projeto

Este capítulo compreende todas as etapas de desenvolvimento do SARDAC. Para cada um dos componentes principais do sistema são apresentados os passos percorridos para definir seus requisitos, projetá-los e construí-los.

O primeiro item desenvolvido foi a **base motorizada**, etapa que envolveu projetos mecânicos e análises físicas do processo. Em seguida, foi adicionado o **sistema embarcado**, utilizando uma plataforma Arduino como ponte de acionamento entre o PC e os servo-motores. Na última etapa, foi desenvolvido o **controle** do sistema, envolvendo praticamente apenas software, mas se apresentando como a fase mais complexa.

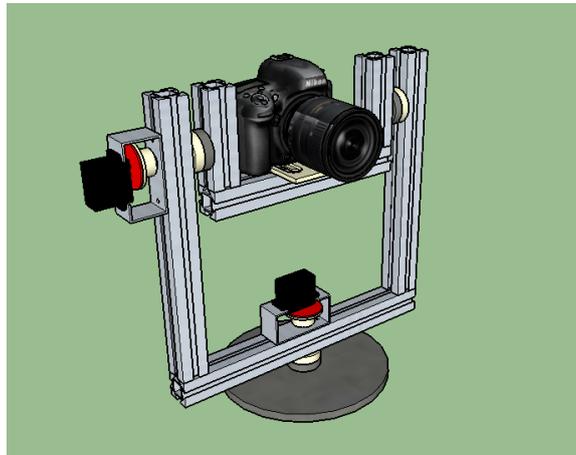
### 3.1 Base Motorizada

No SARDAC, a base motorizada constitui o elemento físico do sistema. A estrutura da base e os servomotores em conjunto com o corpo da câmera e a lente compõe o processo a ser controlado, motivo pelo qual este foi o primeiro elemento do projeto a ser desenvolvido.

A seguir são sistematizadas as etapas de desenvolvimento da base motorizada, organizadas de acordo com a estrutura cronológica básica do projeto.

#### 3.1.1 Projeto Mecânico e Construção

Decidiu-se utilizar perfis modulares de alumínio para a construção da estrutura principal da base. Além de leves e resistentes, eles podem ser montados facilmente através de conectores que requerem pouca usinagem. Os perfis e acessórios (conectores, tampas, parafusos e porcas) foram fornecidos pela EngLegho [24] ao custo de R\$230,00. Foram utilizados perfis quadrados  $30 \times 30$ mm.



**Figura 3.1:** Modelagem da base motorizada no SketchUp.

Um modelo 3D da base, visto na Figura 3.1 foi feito no *software* SketchUp [22]. O modelo, já com as devidas medidas, foi utilizado como referência para usinar e montar as peças. Os trabalhos de usinagem foram realizados na oficina mecânica do Colégio Técnico da UFMG (Coltec). A Figura 3.2 mostra todas as peças utilizadas na construção da base.

Foram usinados em nylon as buchas de suporte dos eixos (2), as peças de suporte da câmera (4), os acoplamentos dos servomotores nos eixos (5), e o suporte inferior (6) para fixação do conjunto num tripé fotográfico. Em alumínio foram usinados os eixos de rotação (3) e os suportes para fixação dos servos na estrutura (7). O nylon foi escolhido para usinagem das primeiras peças principalmente em função das buchas dos eixos (2), já que é um material resistente e oferece baixo coeficiente de atrito. Por sua vez, o alumínio foi escolhido para as demais peças por ser um metal leve e fácil de usinar. O ajuste de posicionamento dos eixos de tilt (3) e do suporte da câmera (4) no “U” interno da estrutura permite alinhar o centro de massa da câmera com o eixo de rotação, reduzindo os momentos de inércia e eliminando o torque estático no motor de tilt.

O resultado final é mostrado na Figura 1.2. Futuramente os fios e cabos podem correr internamente à estrutura, já que os perfis são ocos. Pode-se ver que os servomotores foram acoplados diretamente aos eixos de rotação, dispensando o uso de transmissão e facilitando a construção. A próxima seção detalha os requisitos de acionamento que orientaram a compra dos servos e possibilitaram esse posicionamento dos mesmos.

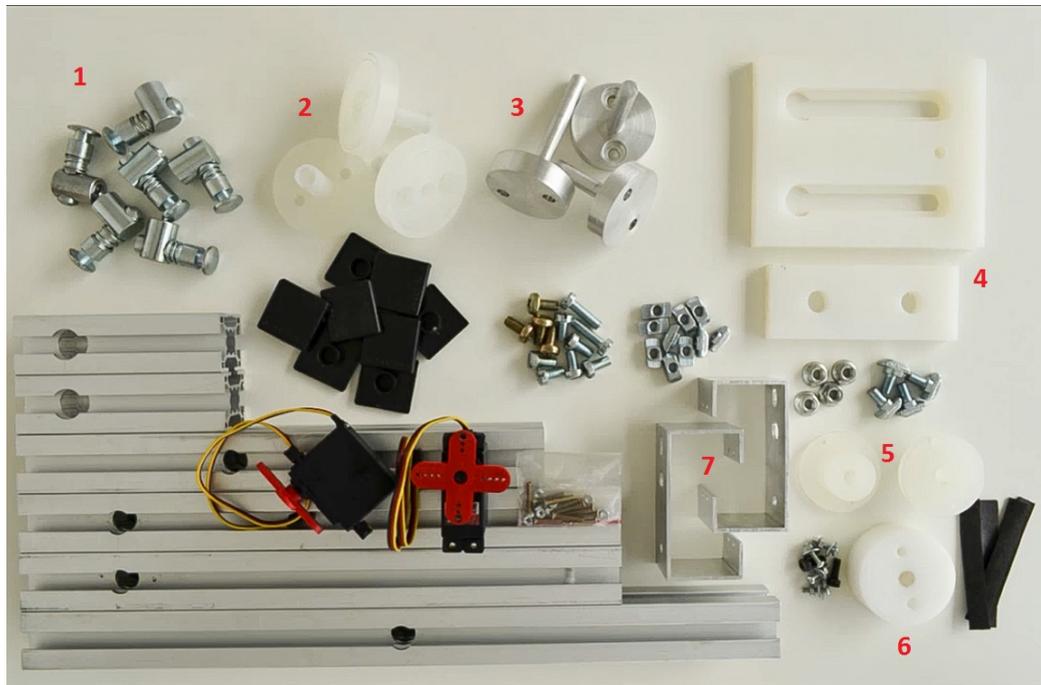
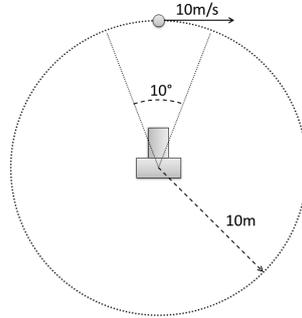


Figura 3.2: Peças da base motorizada.

### 3.1.2 Especificação dos Requisitos de Acionamento

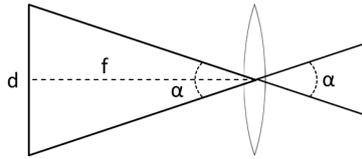
Para definir os requisitos de potência e velocidade dos servomotores, foi criada uma situação hipotética para a aplicação do SARDAC em sua capacidade máxima: o rastreamento de um objeto com velocidade linear de 10m/s, passando a 10m da câmera, com direção de movimento ortogonal ao eixo de pan da base e na mesma altura da câmera. Foi considerada uma distância focal de 200mm (máximo alcance da Nikkor 55–200mm, maior lente disponível). Assim, para rastrear o objeto, o SARDAC precisa controlar somente o eixo de pan. Tal situação está ilustrada na Figura 3.3. O eixo de pan foi escolhido pois em torno dele o momento de inércia do equipamento é maior, e para o uso de dois servos iguais o dimensionamento deve ser feito considerando o eixo de maior requisito. Para pequenos ângulos no movimento da câmera, o movimento linear do objeto pode ser aproximado para um arco de circunferência centrada na câmera, e portanto a trajetória do alvo foi considerada como uma circunferência.

Na situação ilustrada, o SARDAC inicia o rastreamento do objeto no momento em que este passa pelo centro da imagem. O ângulo de visão foi encontrado com auxílio da Equação (3.1), em que  $\alpha$  é o ângulo de visão,  $d$  é o tamanho do sensor na direção medida, e  $f$  é a distância focal da lente. A



**Figura 3.3:** Situação hipotética para análise de requisitos do SARDAC.

equação pode ser deduzida a partir da Figura 3.4.



**Figura 3.4:** Conversão de distância focal em ângulo de visão.

$$\frac{d/2}{f} = \tan \frac{\alpha}{2}$$

$$\alpha = 2 \arctan \frac{d}{2f} \quad (3.1)$$

Considerando um sensor padrão com 35mm na dimensão horizontal (direção do movimento) e a distância focal máxima da lente (200mm), temos um ângulo de aproximadamente 10°.

Deseja-se encontrar o torque a ser aplicado no eixo de pan necessário para que o sistema, partindo do estado de repouso, acelere rápido o suficiente para reposicionar o alvo no centro da imagem sem que ele chegue a sair do campo de visão da câmera. Ou seja, a diferença entre a distância angular percorrida pelo alvo,  $\theta_A$ , e a distância percorrida pelo SARDAC,  $\theta_S$ , não pode ultrapassar 5°. Considerando a aplicação de um torque constante, pode-se encontrar a aceleração necessária  $\alpha$  proporcional ao torque. Considerando estados iniciais nulos para a câmera, podemos seguir o seguinte raciocínio:

$$\theta_A = \omega_A t$$

$$\theta_S = 1/2\alpha t^2$$

$$\theta_A - \theta_S < \theta_{max}$$

$$\omega_A t - 1/2\alpha t^2 < \theta_{max}$$

$$1/2\alpha t^2 - \omega_A t + \theta_{max} > 0 \quad (3.2)$$

Para que a desigualdade acima seja verdadeira para qualquer instante, o ponto mínimo da parábola deve ter valor positivo. O tempo no ponto mínimo é o seguinte:

$$\frac{d}{dt}(1/2\alpha t^2 - \omega_A t + \theta_{max}) = 0$$

$$\alpha t - \omega_A = 0$$

$$t = \omega_A/\alpha$$

Substituindo na Equação (3.2):

$$1/2\alpha \frac{\omega_A^2}{\alpha^2} - \omega_A \frac{\omega_A}{\alpha} + \theta_{max} > 0$$

Após algumas manipulações, chegamos em:

$$\alpha > \frac{\omega_A^2}{2\theta_{max}} \quad (3.3)$$

A Equação (3.3) nos mostra a aceleração angular necessária para cumprir o requisito de movimento da câmera. Substituindo as incógnitas:

$$\omega_A = \frac{V_A}{r} = \frac{10\text{m/s}}{10\text{m}} = 1\text{rad/s}$$

$$\theta_{max} = 5^\circ \cong 0,0873\text{rad}$$

$$\alpha > 5,73\text{rad/s}^2$$

Utilizando esse valor de aceleração para a câmera, pode-se ver na Figura 3.5 o gráfico de  $\theta_A - \theta_S$ . Confirma-se que o alvo permanece na tela ( $\theta_A - \theta_S < 0,0873$ ), e ainda pode-se ver que o objeto seria reposicionado no centro da

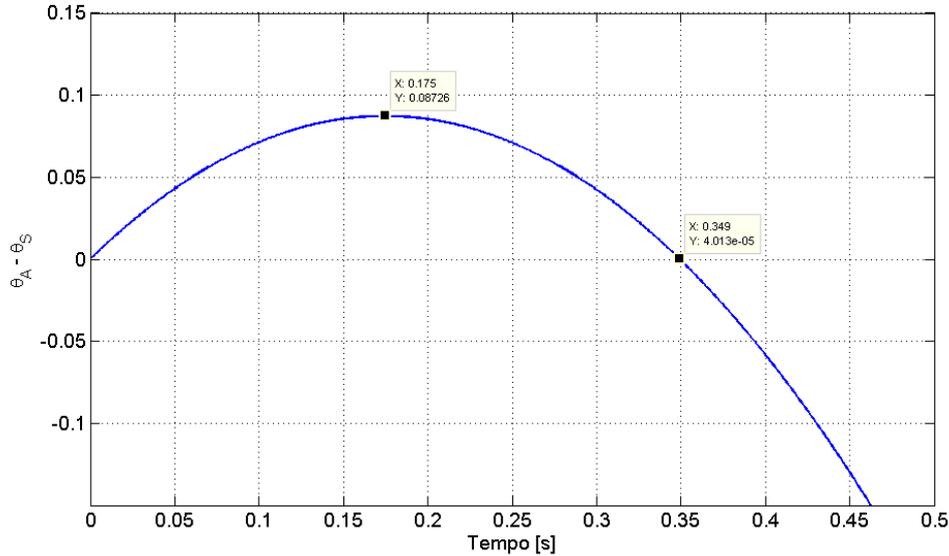
Figura 3.5:  $\theta_A - \theta_S$  para aceleração máxima

imagem em um tempo mínimo de 0,35s em situação ideal, apesar de um grande *overshoot* em seguida pela necessidade de desaceleração.

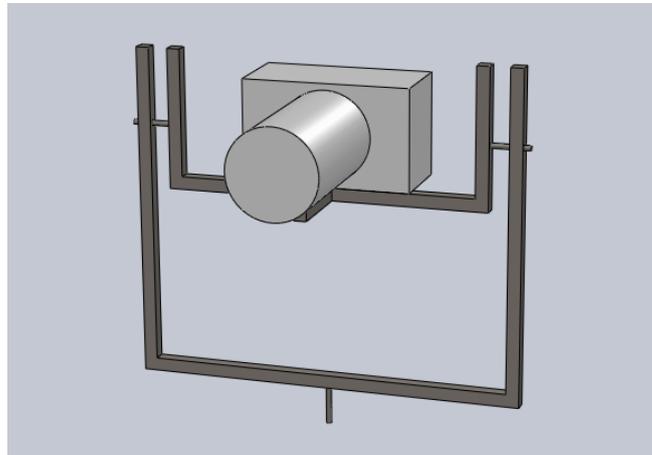
Tendo obtido a aceleração necessária, para cálculo do torque seria preciso encontrar o momento de inércia do sistema em torno do eixo de pan. Para isso foi utilizado o software de modelagem e simulação mecânica SolidWorks. A câmera foi modelada por um bloco retangular e um cilindro, representando o corpo e a lente, respectivamente. A cada um foram atribuídas medidas e densidades aproximadas. Para isso, cada um dos elementos foi medido para obtenção dos volumes, e as massas foram obtidas no site do fabricante [17]. Temos então os seguintes cálculos das densidades:

$$\text{Corpo: } d = m/V = 0,560\text{Kg}/(0,13\text{m} \cdot 0,08\text{m} \cdot 0,06\text{m}) \cong 900\text{Kg}/\text{m}^3$$

$$\text{Lente: } d = m/(\pi r^2 h) = 0,335\text{Kg}/(\pi \cdot (0,035\text{m})^2 \cdot 0,120\text{m}) \cong 730\text{Kg}/\text{m}^3$$

A estrutura da base foi modelada a partir de barras com densidade linear de 0,8Kg/m, semelhante à dos perfis de alumínio. O modelo desenvolvido pode ser visto na Figura 3.6.

Utilizando as ferramentas de propriedades de massa do SolidWorks, foram encontrados os momentos de inércia do sistema. O maior deles se encontra no eixo vertical  $\mathbf{z}$ , aquele em que é realizado o movimento de pan. Ou seja, é esse o movimento que exigirá mais dos motores. O valor encontrado é  $I_z = 0,0169\text{Kg m}^2$ . Temos então o torque máximo necessário:  $\tau_z = I_z \alpha = 0,0169 \cdot 5,73 \cong 0,0968 \text{ Nm}$ . A velocidade máxima durante o rastreamento é



**Figura 3.6:** Modelagem da base motorizada no SolidWorks.

de  $\omega_{max} = \alpha t_{max} = 5,73 \cdot 0,35 \cong 2,0$  rad/s, atingida no instante  $t_{max}$  em que o alvo é reposicionado no centro da imagem. Mas na realidade, a velocidade máxima dos servos deve ser apenas maior que a velocidade do alvo para garantir que ele volte à posição de referência.

É possível concluir então que os servos devem oferecer um torque mínimo de 0,0968 Nm e uma velocidade de operação acima de 1 rad/s. Os motores comprados são do modelo HS-430BH da Hitec ao custo de R\$80,00 a unidade. São servos de alta tensão, e podem ser operados com alimentação de 6,0V ou 7,4V. De acordo com o fabricante, quando operado a 7,4V, o modelo oferece um torque de 0,5 Nm e velocidade máxima de 7,5 rad/s, cumprindo facilmente os requisitos. Observa-se que isso não garante a reprodução da situação em um caso real de rastreamento, já que a análise realizada simula uma atuação ideal do sistema de controle, em que não há, por exemplo, atrasos no acionamento e no sensoramento. Pode-se justificar também com essa análise que a utilização de uma transmissão, seja ela com engrenagens ou correia, não é necessária. Porém, ao não utilizar uma transmissão, a limitação de giro dos motores, que possuem uma extensão de apenas 180°, também se aplica aos movimentos de pan e tilt.

## 3.2 Sistema Embarcado

O sistema embarcado tem a função de acionar os servo-motores a partir de comandos enviados pelo PC, e foi desenvolvido com o uso de uma placa Arduino Uno, mostrada na Figura 3.7.

O Arduino possui a vantagem de ser oferecido como uma plataforma

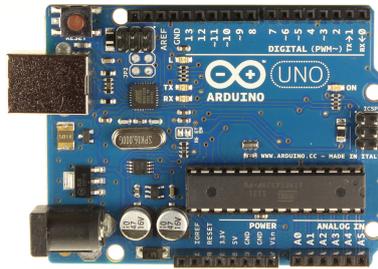


Figura 3.7: Vista superior da placa Arduino Uno [1]

*open-source* de microcontrolador programável, de forma que há diversas bibliotecas cooperativas de funções e aplicações. A programação do Arduino é feita em uma IDE (*Integrated Development Environment*) própria para Windows usando linguagem C/C++, e as rotinas, depois de compiladas, são carregadas por USB na memória flash do microcontrolador. Há uma biblioteca, utilizada no SARDAC, específica para o acionamento de servos, fazendo uso das saídas digitais da placa que são compatíveis com sinal PWM (*Pulse-Width Modulation*, padrão do sinal de controle dos servos). O Arduino é alimentado somente via USB, e suas saídas de tensão de 5V são capazes de alimentar pequenos servos, de baixa potência. Como os servos usados no SARDAC são de maior potência e faixa de tensão acima do normal, foi utilizada uma fonte de computador para alimentá-los. O uso das conexões da fonte de 5V e 12V resultou numa diferença de potencial nominal de 7V, e 6,8V real. Futuramente pode ser utilizada uma fonte menor dedicada para o equipamento.

Segue um algoritmo simplificado do código implementado no Arduino. Após uma rotina inicial, esse código é executado indefinidamente em *loop* enquanto o Arduino estiver ligado.

#### Operação local

```
Lê o estado do teclado do shield
Botão 1 apertado -> incrementa a posição do servo selecionado
Botão 2 apertado -> decrementa a posição do servo selecionado
Botão 3 apertado -> incrementa de 1 grau/loop a velocidade
Botão 4 apertado -> decrementa de 1 grau/loop a velocidade
Botão 5 apertado -> alterna o servo selecionado
```

#### Operação remota

```
Recebe string pelo USB
Separa string em ângulos dos servos 1 e 2
Converte ângulos em larguras de pulso correspondentes
Envia PWM para os servos
Escreve ângulos atualizados na tela LCD
```

## 3.3 Software de Controle

O principal papel do *software* de controle é se comunicar com a câmera e com o Arduino e oferecer mecanismos de controle do SARDAC em malha aberta e em malha fechada. Foi utilizado o digiCamControl [6], um *software open-source* desenvolvido em .NET para comunicação e controle remoto de câmeras DSLR conectadas via USB. Através do programa é possível ajustar muitas das configurações da câmera (abertura, tempo de exposição, ISO, foco, ajuste de branco, qualidade de imagem, etc.), tirar fotos e gravar vídeos, e ver as imagens transferidas para o PC. As funcionalidades do SARDAC foram programadas a partir do código fonte do digiCamControl v1.0.749.0 no Visual Studio 2013. A seguir são descritos os passos de desenvolvimento do software e do sistema de controle em malha fechada.

### 3.3.1 Implementação do Software

Todo o módulo do SARDAC foi desenvolvido de forma a integrar a tela de *Live View* do digiCamControl, na qual a imagem capturada pelo sensor da câmera é visualizada em tempo real, como se estivesse sendo mostrada na própria tela da câmera. A janela foi mostrada na Figura 1.4, mas a Figura 3.8 apresenta os detalhes do módulo do SARDAC.

Uma das vantagens de utilizar o digiCamControl é que o programa faz uso do AForge.NET [7], um *framework* de C# *open-source* composto de bibliotecas para diversas aplicações: processamento de vídeo e imagem, visão computacional, robótica, redes neurais, lógica fuzzy, entre outras. As bibliotecas de processamento de imagens e visão computacional facilitaram bastante a implementação do módulo do SARDAC.

Ampliando a definição no capítulo anterior de rastreamento visual do alvo através de cor, foi utilizado um filtro HSL para a separação do objeto e o plano de fundo. O usuário pode selecionar uma combinação qualquer dos três componentes: *hue* (matiz), *saturation* (saturação) e *luminance* (luminância). Para cada componente usado, o usuário ajusta seu valor e *threshold* para que correspondam às características do objeto a ser rastreado. O objetivo é que, em toda a imagem, apenas o alvo seja definido pelos valores ajustados. Após o filtro HSL, a imagem passa por um processo de binarização (necessário para a detecção de *blobs*), para o qual o usuário define o *threshold*. Ao iniciar o rastreamento, a imagem em preto e branco é usada como parâmetro em uma função que detecta *blobs*, o maior *blob* é selecionado, e seu centroide é calculado. Um retângulo é desenhado na imagem original em torno do maior *blob*, ou seja, o alvo, e as coordenadas de seu centroide são armazenadas na memória para serem usadas no controle em malha fechada.

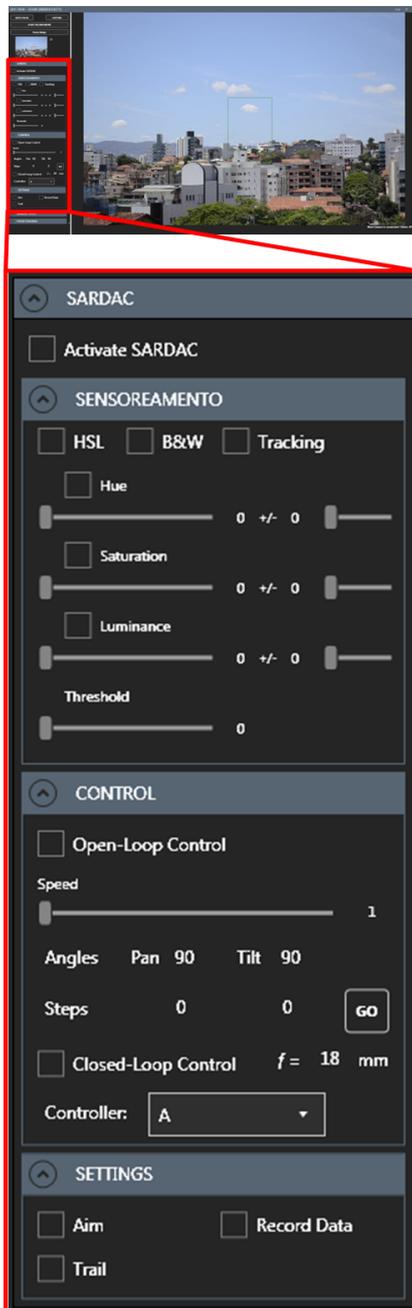


Figura 3.8: Módulo SARDAC.

(em pixels).

As rotinas de rastreamento visual e de controle em malha fechada são executadas por um temporizador de 50ms. Esse intervalo já estava configurado para outras rotinas de processamento do digiCamControl na tela de *Live View*, pois a câmera não consegue oferecer uma taxa de atualização da

Na aba de controle, o usuário pode escolher se controla o SARDAC em malha aberta ou fechada. Em malha aberta, é ajustada a velocidade (ângulo, em décimos de grau, de incremento ou decremento a cada iteração), e o usuário pode controlar a base motorizada pelas setas direcionais do teclado: as setas verticais controlam o movimento de tilt, e as horizontais o movimento de pan. Enquanto uma das teclas estiver pressionada, um temporizador de 50ms controla a execução da rotina. Também é possível indicar diretamente nos campos de pan e tilt os ângulos desejados. Em malha fechada, pode-se selecionar o tipo de controlador a ser utilizado dentre aqueles previamente programados nas linhas de código. Antes de ligar o controle, o usuário deve informar a distância focal da lente, pois é um parâmetro importante no cálculo do ganho do controlador, como será mostrado à frente. O módulo do SARDAC ainda possui uma aba para algumas configurações: é possível ativar uma mira em formato de + no centro da imagem, um rastro das posições anteriores do centroide do alvo, ou a gravação de dados. Este último recurso grava em um arquivo de texto a cada iteração de controle os seguintes dados: hora atual (com resolução de ms), sinal de controle para cada servo (em décimos de grau), posição do centroide do alvo na imagem nos eixos  $x$  e  $y$  (em pixels), e o erro de posicionamento em cada eixo

imagem muito acima de 20fps. Os quadros recebidos da câmera via USB e usados ao longo de todo o processo de rastreamento possuem uma resolução de  $640 \times 426$  pixels. Essa resolução é maior que aquela utilizada em boa parte da literatura, mas a eficiência das funções do AForge.NET e a capacidade de processamento dos computadores atuais permitem a devida execução do controle com o período de amostragem definido, dispensando a necessidade de redimensionamento das imagens. A posição de referência é fixa no centro da imagem ( $x = 320$ ,  $y = 213$ ), definida em linha de código. A cada iteração do controlador, o sinal de controle, em graus, para os dois servomotores, é convertido numa *string* e enviado para a porta serial em que o Arduino está conectado. Também foi implementado um saturador lógico, que verifica, antes de enviar o sinal, se os valores estão entre os limites de  $0^\circ$  e  $180^\circ$ . Quanto à posição inicial do sistema, foi definido o valor de  $90^\circ$  tanto para pan quanto para tilt.

### 3.3.2 Modelagem do Processo

Para projetar o controlador, primeiramente foi necessário modelar o processo. Como foi definido no capítulo anterior que o controle seria baseado em imagem, isso significa encontrar as equações que relacionam o sinal de controle para os servos, em graus, com o deslocamento do alvo na imagem, em pixels. Para a identificação dos modelos nos eixos de pan e tilt, um objeto estático foi rastreado e colocado em malha aberta no centro da imagem. Com a coleta de dados ativada, foram gerados pequenos degraus no sinal de comando para os servos em torno da posição inicial, e os deslocamentos do alvo na imagem foram registrados juntamente com os demais dados necessários para a identificação.

O primeiro requisito para a modelagem foi calcular o período de amostragem, ou seja, verificar se os temporizadores estavam executando as rotinas de controle com intervalos de 50ms como configurado no código do programa. Foi encontrado um período médio entre amostras de 62ms. Essa diferença para o valor configurado será discutida no próximo capítulo. É mostrado a seguir a determinação dos modelos para pan e tilt, feitas com a lente ajustada a uma distância focal de 24mm.

#### Pan

A Figura 3.9 mostra os dados de identificação do modelo de pan. As marcações nos gráficos mostram um atraso na resposta do sistema aos sinais de controle. Analisando este e outros gráficos de resposta do sistema, verificou-se um atraso máximo de  $4T$  (quatro períodos de amostragem). Também pode

ser observado na curva de deslocamento no eixo  $x$  da imagem uma histerese significativa, discutida no próximo capítulo.

A identificação dos modelos foi realizada através da interface *ident* do MatLab, que facilita o pré-processamento dos dados e utiliza a função *tfest()* para obter os modelos. Por sua vez, *tfest()* usa o método de minimização de erros de predição (PEM) para estimar os coeficientes das funções de transferência a partir de dados temporais [9]. Foram estimados dois modelos discretos: um de primeira ordem, com um polo, mais atraso e outro de segunda ordem, com dois polos e um zero, mais atraso. O atraso de  $4T$  foi passado ao estimador como parâmetro. Na Figura 3.10 é comparada a resposta aos dados de identificação entre os modelos e o processo real. Foi escolhido para o projeto do controlador de pan o modelo de segunda ordem mais atraso, com erro quadrático médio 12% menor que o modelo de primeira ordem (diferença minimizada pelo erro comum aos dois devido à histerese). A Equação (3.4) apresenta a função de transferência do modelo estimado. Vê-se que, no tempo discreto, atrasos múltiplos do período de amostragem convertem-se em polos em 0.

$$G_p = z^{-4} \cdot \frac{-0,1541z^{-1}}{1 - 1,244z^{-1} + 0,4045z^{-2}} \quad (3.4)$$

### Tilt

A Figura 3.11 mostra a resposta do sistema no eixo tilt para a mesma variação do sinal de controle. Novamente um atraso máximo de  $4T$  é identificado. Observa-se que não há histerese perceptível, e a dinâmica do sistema em tilt é mais rápida, mas subamortecida. Os picos de *overshoot* são consequência da malha de controle de posição interna dos servos.

Utilizando os mesmos recursos do MatLab empregados no modelo de pan, também foram estimados para tilt um modelo de primeira ordem, com um polo, mais atraso e um de segunda ordem, com dois polos e um zero, mais atraso. Suas respostas aos dados de identificação são comparadas na Figura 3.12. O modelo de segunda ordem foi novamente escolhido principalmente por conseguir reproduzir a resposta subamortecida do sistema. Seu erro quadrático médio na janela de identificação é 73% menor que o erro do modelo de primeira ordem. A Equação (3.5) apresenta a função de transferência do modelo estimado para tilt.

$$G_t = z^{-4} \cdot \frac{1,231z^{-1}}{1 - 0,03801z^{-1} + 0,1463z^{-2}} \quad (3.5)$$

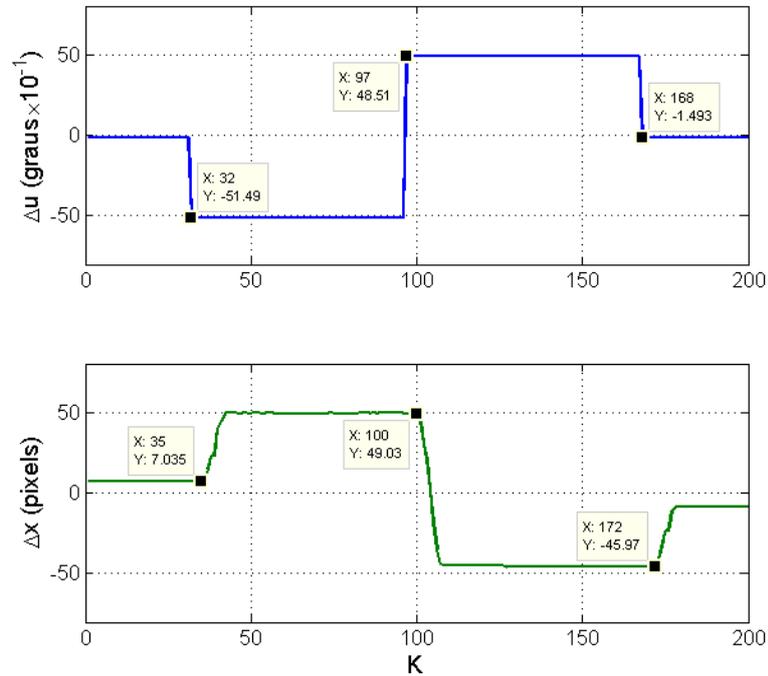


Figura 3.9: Dados de identificação no eixo Pan

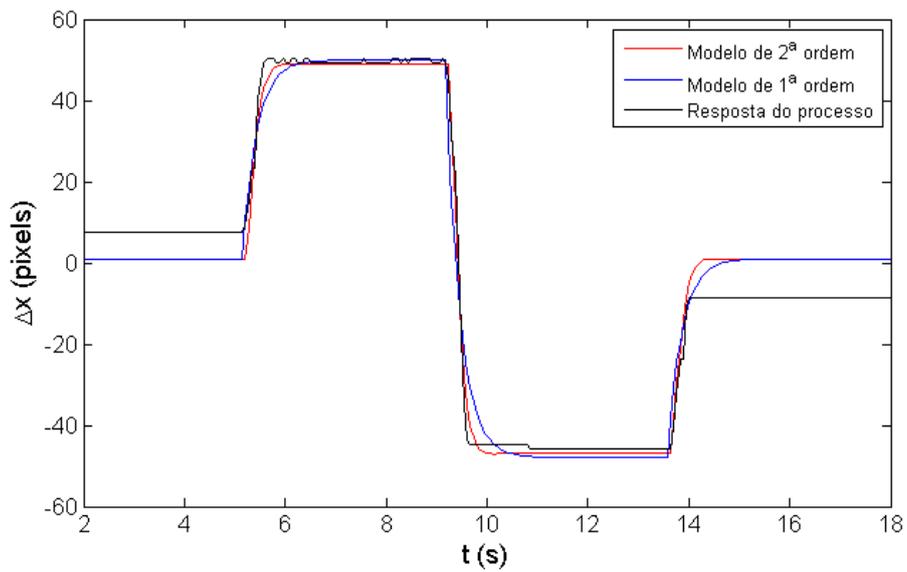


Figura 3.10: Resposta dos modelos de Pan para dados de identificação

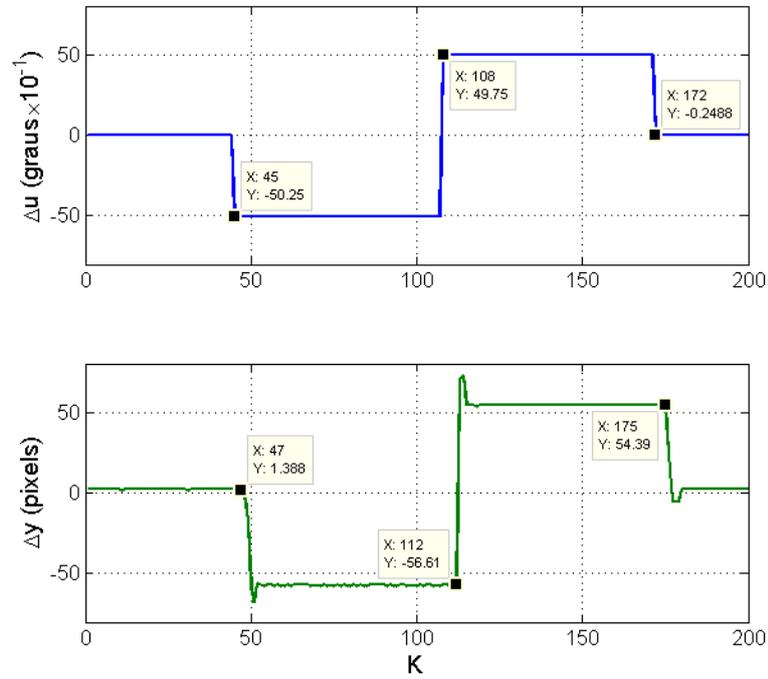


Figura 3.11: Dados de identificação no eixo Tilt

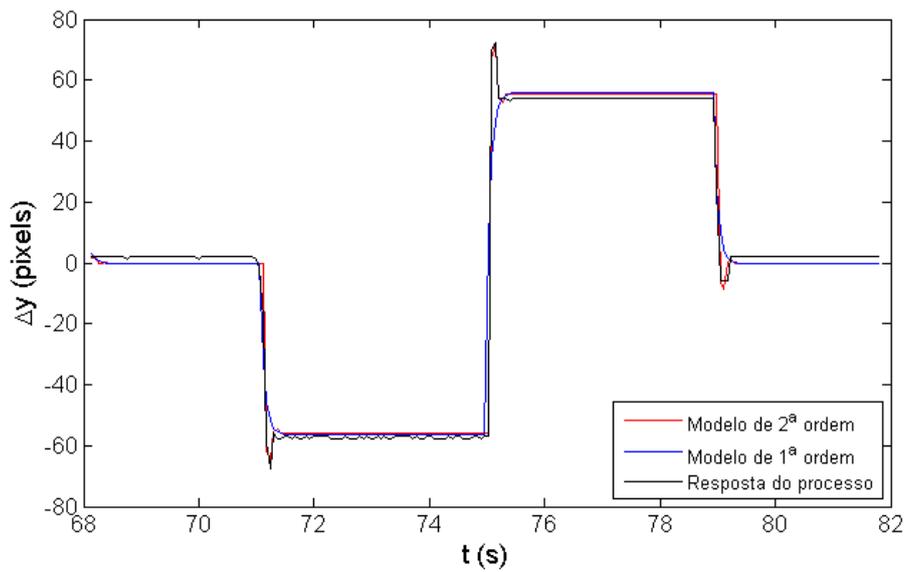


Figura 3.12: Resposta dos modelos de Tilt para dados de identificação

### 3.3.3 Projeto de Controle

Obtidos os modelos dinâmicos do sistema para pan e tilt, foi realizado o projeto do controlador para cada uma das malhas. Como definido no capítulo anterior, foram projetados controladores PI. Como os processos são do tipo 0 (não possuem integradores), o controle PI permite erro nulo em estado estacionário para entradas em degrau na referência e no distúrbio da saída. A Figura 3.13 mostra o controlador PI inserido numa representação das malhas de controle do SARDAC em pan ou tilt. O bloco G representa todo o processo modelado em caixa preta anteriormente, e o movimento do alvo é interpretado como um distúrbio na saída.

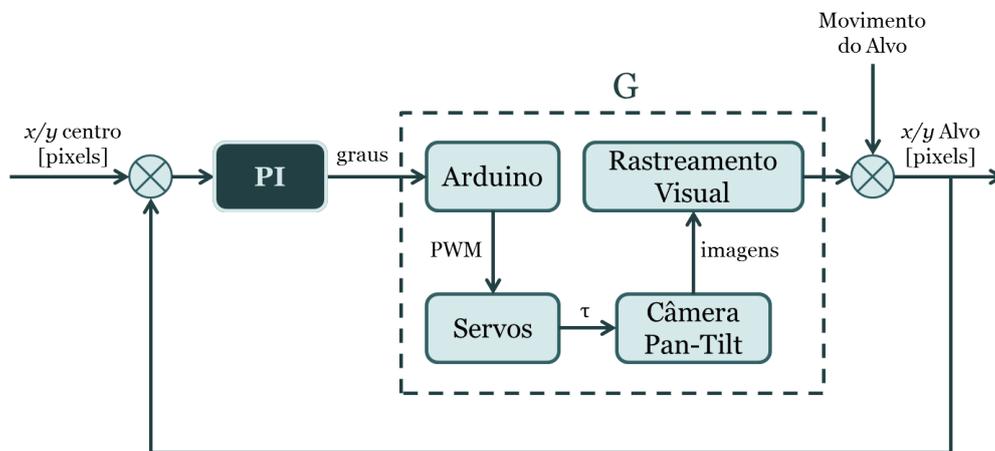


Figura 3.13: Representação da malha fechada de controle do SARDAC.

Os parâmetros dos controladores foram definidos a partir de dois métodos: o primeiro é um método empírico de Ziegler-Nichols, e o segundo é análise do lugar das raízes. O algoritmo que implementa os controladores no *software* utiliza a saturação do sinal de controle como técnica *anti-windup*. A seguir são apresentados os projetos usando cada um dos métodos. Como a identificação dos modelos do processo foi feita a partir de dados coletados com uma distância focal fixa em 24mm, em seguida será mostrado como parametrizar os ganhos dos controladores em função da distância focal.

#### Método de Ziegler-Nichols

O autor apresenta em [19] algumas das regras empíricas sugeridas por Ziegler e Nichols para sintonia de controladores PID. O método utilizado neste trabalho consiste na análise da resposta ao degrau do processo para determinar os valores  $K_p$  e  $T_i$  do controlador mostrado na Figura 3.14. Como o controlador desejado é um PI, o valor de  $T_d$  é zero.

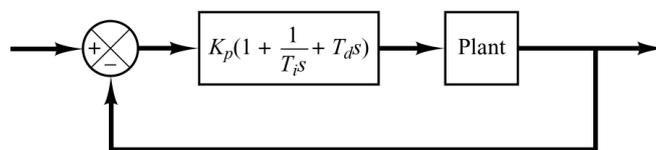


Figura 3.14: Malha fechada com controlador PID [19].

O método requer que a resposta do processo a um degrau na entrada tenha um formato em “S”. A partir de uma linha tangente ao ponto de inflexão da curva, são identificados o atraso  $L$  e a constante de tempo  $T$ , assim como mostrado na Figura 3.15. A partir das constantes encontradas, são definidos os valores do PID de acordo com as regras da Tabela 3.1.

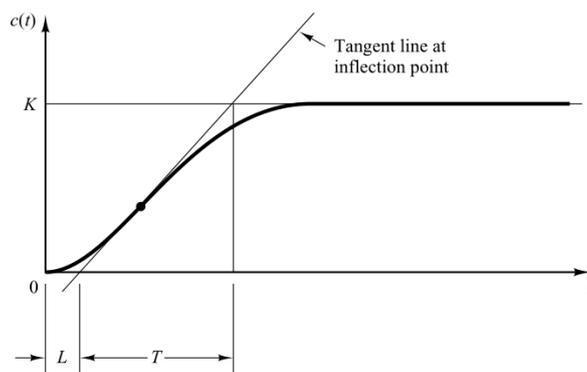


Figura 3.15: Método de Ziegler-Nichols para resposta ao degrau [19].

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

Tabela 3.1: Regra de Ziegler-Nichols para sintonia de PID [19].

Foram analisadas as respostas ao degrau do modelo de segunda ordem mais atraso para pan, e do modelo de primeira ordem mais atraso para tilt. Para uma análise no tempo contínuo, as respostas discretas passaram por uma interpolação. O modelo de segunda ordem para tilt não foi usado neste

método pois não tem resposta em forma de “S”. As constantes das respostas de pan e tilt encontradas foram as seguintes:  $L_p = 0,2587$ ,  $T_p = 0,3109$ ,  $L_t = 0,2481$ ,  $T_t = 0,0402$ . A partir de tais constantes, a regra de Ziegler-Nichols da Tabela 3.1 para controlador PI fornece os seguintes valores:  $K_p^{pan} = 1,0816$ ,  $T_i^{pan} = 0,8623$ ,  $K_p^{tilt} = 0,4761$ ,  $T_i^{tilt} = 0,6366$ . Esses valores correspondem a um controlador PI no tempo contínuo. Em [18] é deduzida a forma discreta recursiva, ou de velocidade, do PID, mostrada na Equação (3.6), em que  $e(k)$  é o erro na amostragem  $k$ ,  $u(k)$  é a saída do controlador,  $T$  é o período de amostragem, e  $K_p$ ,  $T_i$  e  $T_d$  são valores do PID contínuo:

$$u(k) = u(k-1) + K[e(k) - e(k-1)] + K_i e(k) + K_d[e(k) - 2e(k-1) + e(k-2)] \quad (3.6)$$

$$K = K_p - \frac{K_i}{2}; \quad K_i = \frac{KT}{T_i}; \quad K_d = \frac{KT_d}{T}$$

Como o controlador final é um PI e  $T_d = 0$ , aplicando a transformada Z temos a Equação (3.7):

$$\frac{U(z)}{E(z)} = \frac{(K + K_i) - Kz^{-1}}{1 - z^{-1}} \quad (3.7)$$

Calculando  $K$  e  $K_i$  a partir dos valores para tempo contínuo e substituindo na Equação (3.7), temos os controladores em tempo discreto  $C_p^{zn}$  e  $C_t^{zn}$ , para pan e tilt respectivamente, indicados na Equação (3.8) e na Equação (3.9).

$$C_p^{zn}(z) = \frac{-1,1205(1 - 0,9306z^{-1})}{1 - z^{-1}} \quad (3.8)$$

$$C_t^{zn}(z) = \frac{0,49926(1 - 0,9071z^{-1})}{1 - z^{-1}} \quad (3.9)$$

### Método do lugar das raízes

Para sintonia dos controladores PI a partir do lugar das raízes, foi utilizada a interface *sisotool* do MatLab, que oferece diversas configurações e ferramentas de sintonia, além do lugar das raízes, e gráficos de análise para visualização de respostas do sistema em malha fechada.

Para ambas as malhas, de pan e tilt, foram estabelecidos os seguintes critérios de desempenho, definidos de forma arbitrária:

- Tempo de acomodação menor que 2s (faixa de acomodação de  $\pm 2\%$  do valor final)

- *Overshoot* na resposta do sistema menor que 10%

A sintonia dos controladores  $C_p^{lr}$  e  $C_t^{lr}$ , de pan e tilt respectivamente, foi feita com base no lugar das raízes das malhas e nos gráficos de resposta da malha fechada ao degrau. É importante ressaltar que respostas a variações nas referências não são muito relevantes, já que estas permanecem constantes. Portanto, foi analisado como a saída do sistema e do controlador se comportam quando há um distúrbio na saída, ou seja, quando o alvo se move na imagem. Os valores de *overshoot* e tempo de acomodação foram observados nesses gráficos para sintonia do PI. Isso significa que os controladores seriam eficientes para rastreamento do alvo.

A Figura 3.16 mostra o lugar das raízes da malha de pan com  $C_p^{lr}$  ajustado. Os requisitos de *overshoot* e tempo de acomodação foram especificados, e no gráfico é indicada em branco a região que os atende para o caso de um sistema na forma canônica de segunda ordem (recurso do MatLab para visualização dos requisitos de projeto). Como o sistema em análise é de ordem maior, devido aos polos introduzidos pelo atraso, a região delimitada apenas oferece um ponto inicial para a sintonia, uma vez que não corresponde exatamente à resposta real.

As respostas do processo e do controlador de pan para degrau no distúrbio de saída, com  $C_p^{lr}$  ajustado, são mostradas na Figura 3.17. O controlador de pan sintonizado possui a função de transferência reproduzida na Equação (3.10):

$$C_p^{lr}(z) = \frac{-0,46(1 - 0,75z^{-1})}{1 - z^{-1}} \quad (3.10)$$

O mesmo processo de sintonia foi realizado para o controlador PI da malha de tilt. A Figura 3.18 mostra o lugar das raízes da malha com  $C_t^{lr}$  ajustado. Já as respostas do processo e do controlador de tilt para degrau no distúrbio de saída, com  $C_t^{lr}$  ajustado, são mostradas na Figura 3.17. O controlador de tilt sintonizado possui a seguinte função de transferência:

$$C_t^{lr}(z) = \frac{0,25(1 - 0,46z^{-1})}{1 - z^{-1}} \quad (3.11)$$

Percebe-se que, enquanto o sinal da saída de  $C_p^{lr}$  corresponde ao sinal do distúrbio,  $C_t^{lr}$  apresenta uma saída com sinal contrário. Isso ocorre devido às coordenadas que se utiliza normalmente para localização dos pixels na imagem: a origem está no canto superior esquerdo, o eixo x cresce para a direita (mesmo sentido em que o ângulo de pan aumenta) e o eixo y cresce para baixo (sentido contrário ao crescimento do ângulo de tilt).

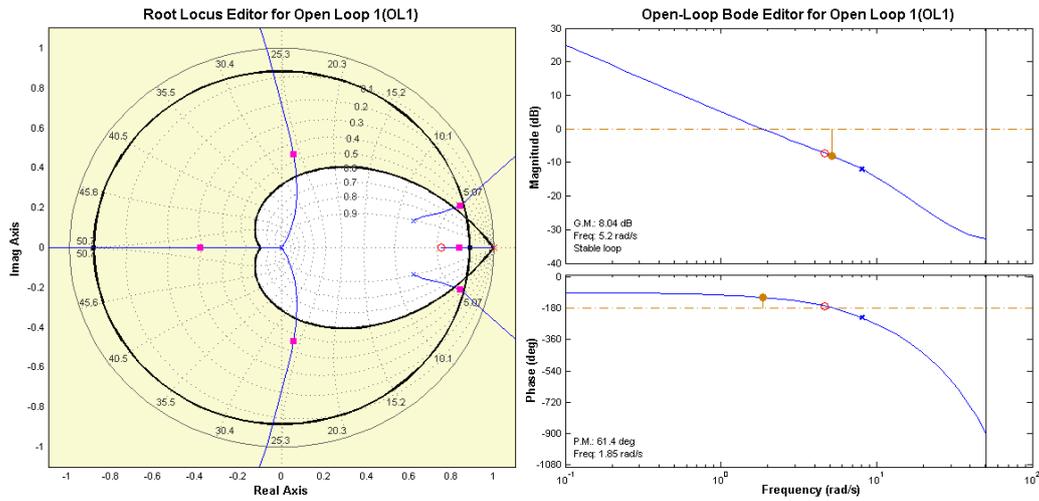


Figura 3.16: Lugar das raízes da malha de Pan

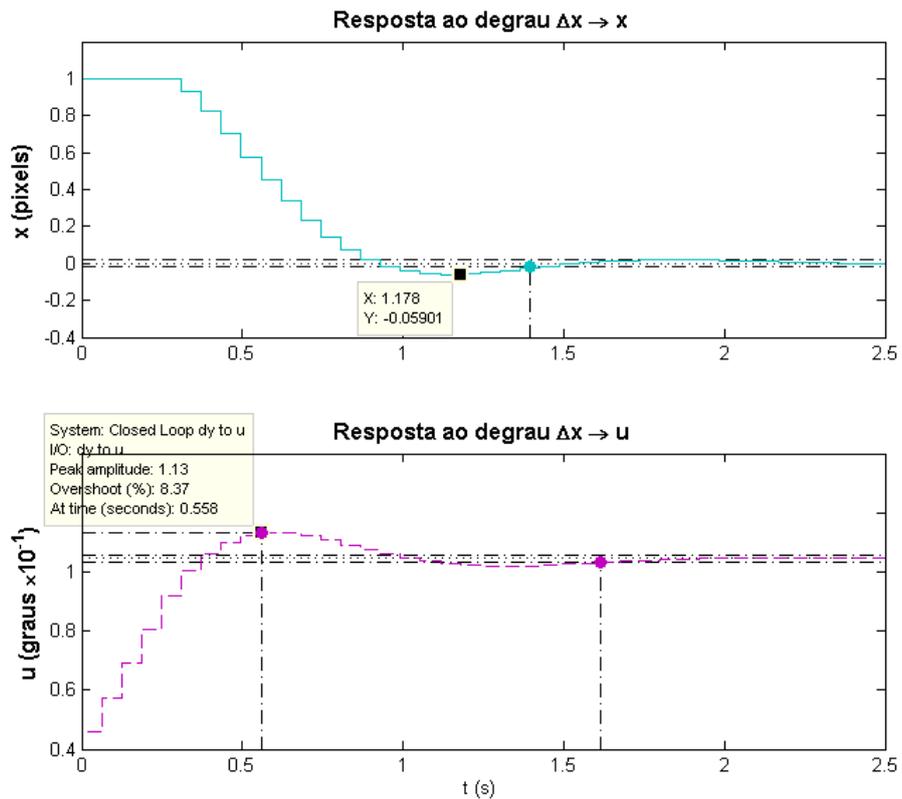


Figura 3.17: Respostas ao degrau no distúrbio da saída no eixo x da imagem

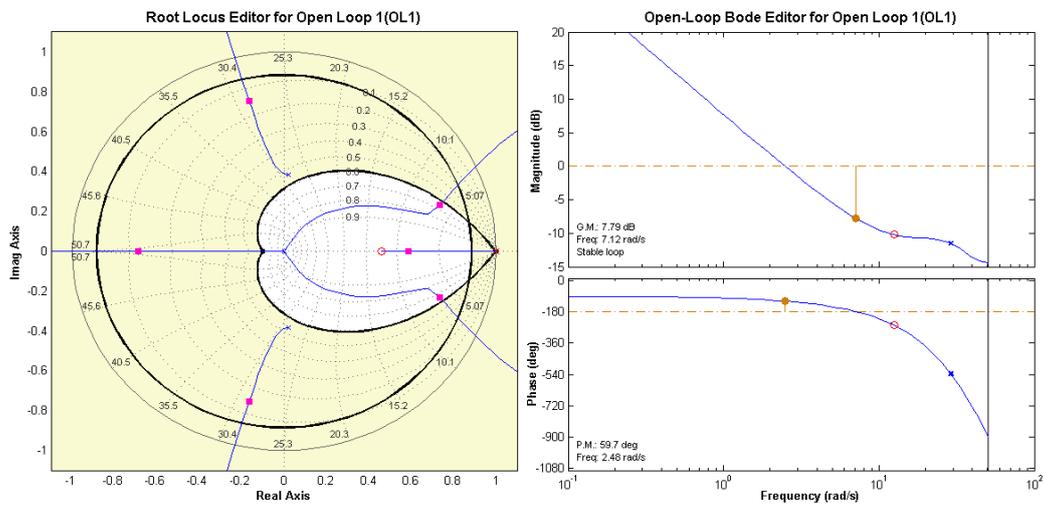


Figura 3.18: Lugar das raízes da malha de Tilt

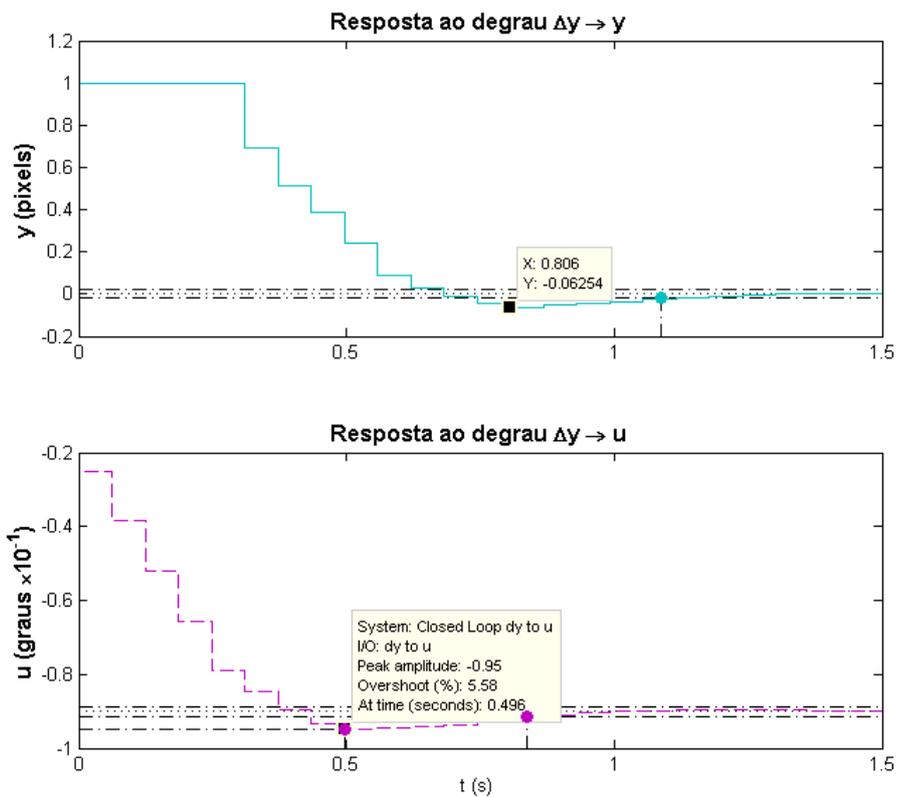


Figura 3.19: Respostas ao degrau no distúrbio da saída no eixo  $y$  e da imagem

### Parametrização em função da distância focal

Para permitir um controle parametrizado em função da distância focal da lente, investigou-se a relação entre o deslocamento angular da câmera  $\Delta\alpha$ , o deslocamento resultante do centroide do alvo na imagem  $\Delta d$  e a distância focal  $f$ . A Figura 3.20 auxilia nessa análise.

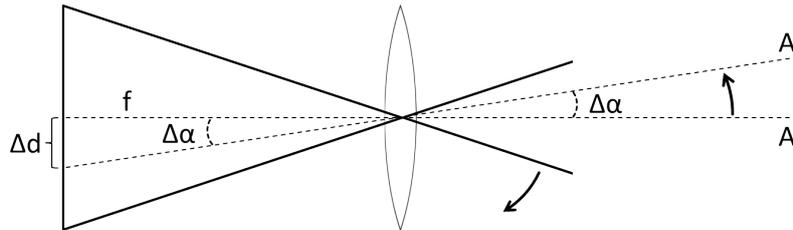


Figura 3.20: Relação entre distância focal e deslocamento do alvo na imagem.

Quando a câmera rotaciona em um dos eixos um deslocamento angular  $\Delta\alpha$ , a direção do centroide (no mundo) do alvo no referencial da câmera passa de  $A$  para  $A'$ , ocasionando em seu deslocamento na imagem de  $\Delta d$ . Temos então a seguinte relação:

$$\Delta d = \tan(\Delta\alpha)f \quad (3.12)$$

Pela Equação (3.12) percebe-se que se for ajustado uma nova distância focal  $f' = \gamma f$ , para que o deslocamento na imagem  $\Delta d$  permaneça constante, é necessário um deslocamento angular  $\Delta\alpha'$  de maneira que:

$$\tan(\Delta\alpha') = \frac{1}{\gamma} \tan(\Delta\alpha)$$

Como o ganho do controlador no eixo de interesse é diretamente proporcional a  $\Delta\alpha$ , a cada vez que uma nova distância focal for dada, o programa deve atualizar os ganhos dos controladores de acordo com a Equação (3.13), em que  $K$  é o ganho do PI sintonizado para uma distância focal  $f = 24\text{mm}$ ,  $\gamma = f'/24$  e  $K'$  é o novo ganho. Isso garante que os critérios de desempenho da malha fechada sejam cumpridos para qualquer distância focal, desde que o usuário forneça o valor correto.

$$K' = \tan^{-1}\left[\frac{1}{\gamma} \tan(K)\right] \quad (3.13)$$

No próximo capítulo é mostrado o SARDAC em funcionamento. Testes e validações são feitos a fim de verificar o cumprimento dos requisitos. São discutidos os resultados, comparados os métodos de controle, e apontadas as dificuldades encontradas.

# Capítulo 4

## Resultados

Neste capítulo avalia-se o funcionamento do SARDAC. Alguns problemas detectados ainda na fase de desenvolvimento do projeto foram investigados, e novos testes e validações foram realizados para verificar o cumprimento dos requisitos. As dificuldades encontradas durante os testes são relatadas, e os métodos diferentes de controle do sistema tiveram seu desempenho comparado.

A análise dos resultados foi dividida nos dois componentes principais de uma aplicação de controle servo visual: o rastreamento visual e o controle de acionamento do pan-tilt baseado na imagem.

### 4.1 Rastreamento Visual

A Figura 4.1 mostra parte do procedimento descrito no capítulo anterior para ajustes manuais dos filtros para rastreamento visual de alvos. Ao longo das experimentações, identificou-se as seguintes situações em que cada componente do filtro HSL pode ser útil:

- **Matiz (H)**: quando a cor do alvo é diferente do plano de fundo.
- **Saturação (S)**: quando o alvo é mais saturado que o plano de fundo.
- **Luminância (L)**: quando o alvo reflete ou emite muita luz, ou absorve muita luz, em relação ao plano de fundo.

Ao usar uma combinação dos componentes do filtro (HS, HL, SL ou HSL), os alvos foram rastreados com sucesso em todos os ambientes testados: ambiente externo com iluminação natural e fundo detalhado, ambiente interno com pouca iluminação, etc. Foram observadas as seguintes vantagens da técnica de rastreamento visual escolhida:

- É versátil, se adaptando a diversos ambientes, iluminações e características do alvo.
- Possui baixo custo computacional e fácil implementação.
- Os ajustes manuais são intuitivos (lidam com conceitos comuns de brilho e cor).

Por outro lado, o rastreamento do alvo pela caracterização no espaço HSL também gera dificuldades e limitações para o sistema. São elas:

- É necessário que o alvo se mantenha na imagem enquanto o ajuste manual dos filtros é feito. Isso pode ser um problema para objetos em constante movimento.
- Alvos com características de cor e brilho semelhantes ao fundo são difíceis de rastrear, mesmo em movimento. Isso também vale para objetos com superfície complexa (multicolor, incidência irregular de luz, padrões alternados, etc.)
- Durante o controle pan-tilt, se surgir ao fundo algo com características próximas às do objeto, o foco no alvo pode se perder.
- Um rastreamento ótimo do alvo depende da habilidade do usuário para ajustar o filtro HSL.

## 4.2 Controle em Malha Fechada

Nesta seção faz-se uma análise dos dados coletados durante os testes em malha fechada. O objetivo é tentar responder às seguintes perguntas: qual a influência dos problemas apontados anteriormente no controle do SARDAC? É possível contorná-los? Como criar um ambiente de testes que garanta uma coerência dos dados e repetibilidade dos experimentos? Quais dos controladores têm melhor desempenho? Eles cumpriram os requisitos de projeto?

### 4.2.1 Simulador de Alvos

Para garantir uma repetibilidade nos testes de rastreamento e facilitar a variação das características do objeto rastreado (cor e tipo de movimento), foi desenvolvido em .NET um simulador de alvos. Sua interface simples pode ser vista na Figura 4.2.

O simulador de alvos oferece as seguintes funcionalidades:

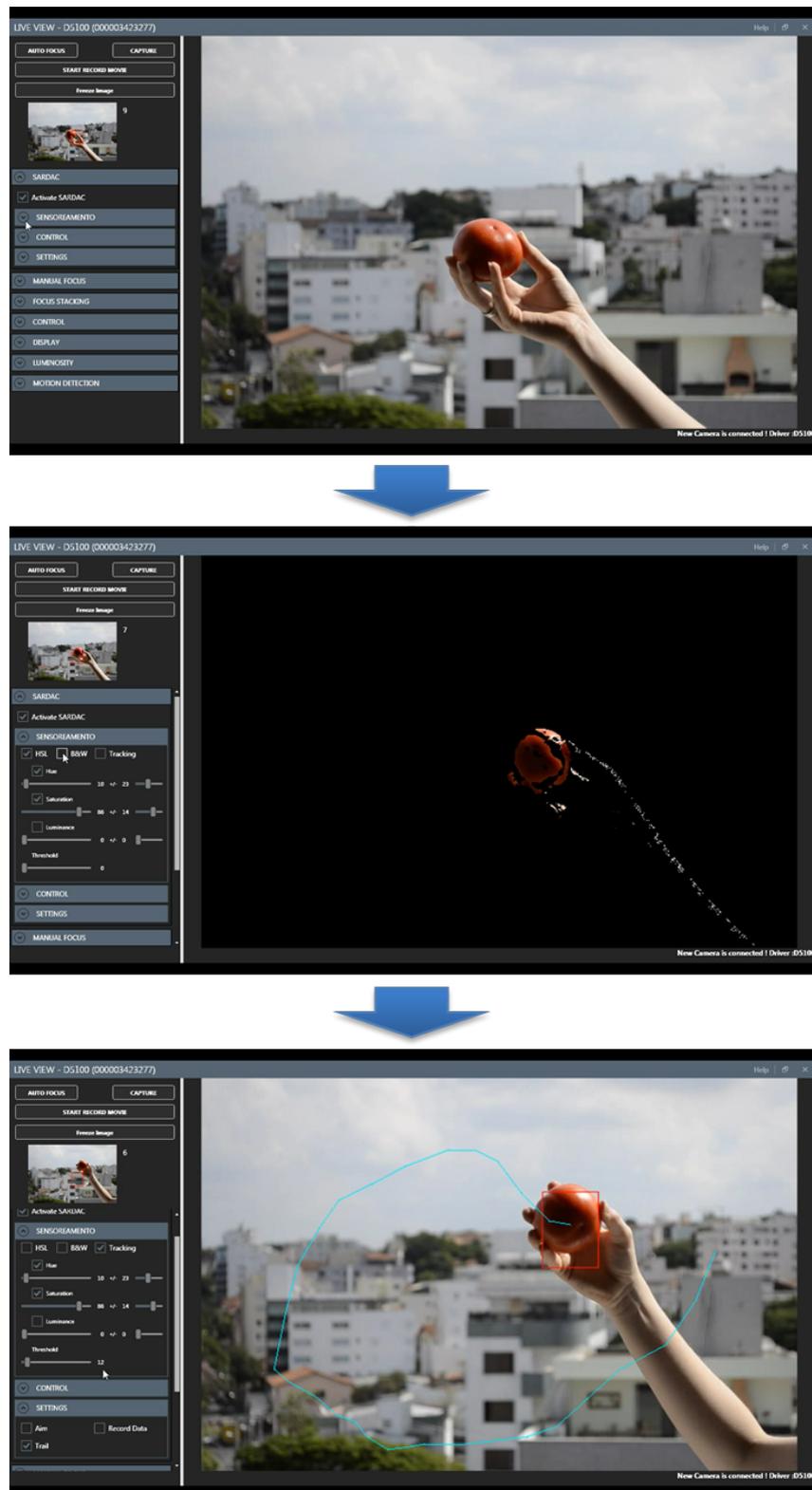


Figura 4.1: Ajuste manual do filtro HSL para rastreamento visual.

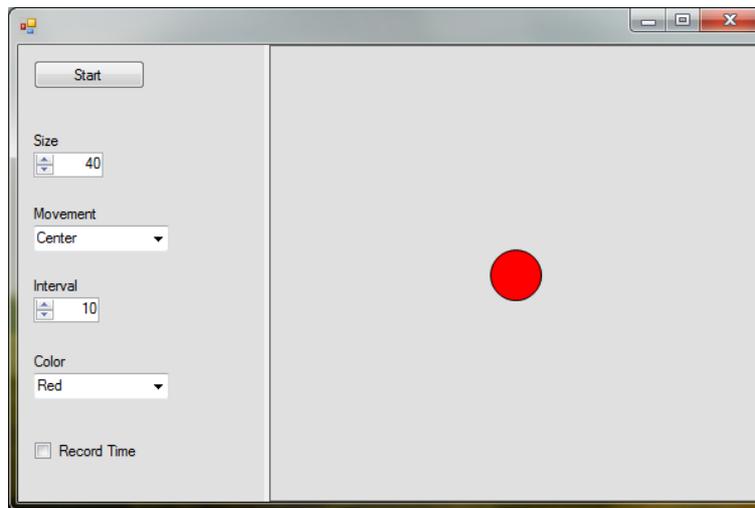


Figura 4.2: Interface do programa simulador de alvos.

- Ajuste do tamanho do alvo (diâmetro do círculo).
- Escolha do tipo de movimento do alvo nos eixos  $x$  e  $y$ : degraus (o círculo salta de um vértice a outro da tela no sentido horário), rampas (seguindo as arestas da tela formando um quadrado) e senoidal (movimento circular na tela). O movimento é iniciado ou interrompido com um botão próprio.
- Velocidade do movimento (intervalos em ms entre cada iteração do movimento).
- Seleção da cor do círculo.
- Registro em arquivo de texto das posições do alvo em função do tempo.

### 4.2.2 Análise de Problemas

Antes de avaliar o desempenho dos controladores projetados, foram investigados alguns problemas detectados no capítulo anterior. Foi analisado se eles poderiam interferir no controle em malha fechada, e neste caso, se algo poderia ser feito para resolvê-los. Foram identificados três principais problemas: o período de amostragem diferente do definido, o atraso na resposta do sistema e a histerese no movimento rotacional de pan.

### Período de Amostragem

Como exposto no Capítulo 3, o temporizador que controla a execução das rotinas de captura de imagens e controle do SARDAC foi configurado com intervalo de 50ms. No entanto, o período de amostragem observado nos dados coletados variava em torno de 62ms. Isso ocorre pois o programa roda em Windows 7, sistema operacional que não oferece recursos nativos de tempo real, e a execução precisa dos temporizadores não pode ser garantida. Resta saber se a variação do período de amostragem em torno de 62ms é significativa. Para isso, em uma das coletas de dados, os intervalos entre amostras foram calculados e usados para gerar um histograma, mostrado na Figura 4.3.

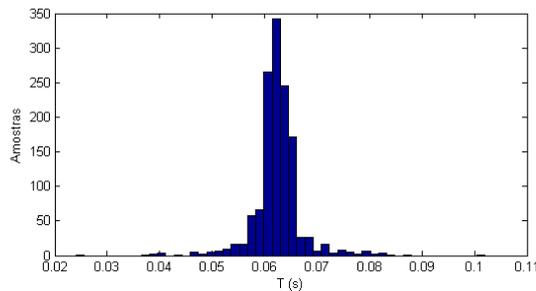


Figura 4.3: Histograma dos períodos de amostragem.

Podemos ver que os períodos de amostragem não variam muito em torno da média de 62ms. O desvio padrão dos dados é de 4,6ms, e 95% da massa de dados se encontra entre 52ms e 72ms. Definida a média de  $T = 62\text{ms}$  para a modelagem do sistema e projeto dos controladores, variações de 10ms no período de amostragem não devem ter impacto significativo no controle em malha fechada, visto que tais valores são menores que os atrasos e as constantes de tempo do processo em uma ordem de grandeza.

### Atraso do Sistema

Como já relatado, observou-se um atraso na resposta do sistema de até  $4T$ , ou seja, para um alvo estático rastreado, seu deslocamento na imagem só era percebido  $4T$  – ou menos – após o envio de um degrau para o acionamento pan-tilt. Para avaliar se o atraso ocorria no acionamento dos servomotores ou na captura das imagens, e se algo poderia ser feito a respeito, dois experimentos foram feitos:

1. O simulador de alvos foi colocado em modo de deslocamentos em degrau. O registro de posições em arquivo de texto permitiria detectar o

instante em que o alvo passa de um ponto para o ponto seguinte, no extremo oposto da tela. Enquanto isso, o *software* de controle também está em funcionamento, rastreando o objeto e registrando suas posições na imagem em outro arquivo de texto. A execução de ambos os programas na mesma máquina garante o uso do mesmo relógio e a possibilidade de comparar os instantes em que o alvo se move no mundo e na imagem.

2. Uma segunda câmera foi usada para filmar duas telas: em uma estava o simulador de alvos em modo degrau, e na outra estava o *software* de controle na tela de *Live View*. Sabendo a taxa de quadros por segundo da filmagem desta segunda câmera, seria possível visualizar o atraso e convertê-lo de quadros para segundos.

Ambos os experimentos indicaram um atraso máximo na captura das imagens de aproximadamente 250ms, ou  $4T$ . Isso permitiu concluir que a porção significativa do atraso ocorria na captura dos quadros da câmera. Acredita-se que ele seja inerente à comunicação via USB com o equipamento.

### Histerese no Movimento de Pan

No capítulo anterior foi apontada uma histerese significativa no movimento de pan. Averiguou-se se o acoplamento dos servos nos eixos de rotação possuíam folgas ou movimentos livres, mas todas as peças da base motorizada estavam devidamente fixadas. Concluiu-se que a histerese ocorre internamente nos servomotores. Como todo o peso da estrutura é sustentado pelas peças do eixo de pan, é nele que ocorre o maior atrito estático. Isso explica o fato da histerese ser perceptível somente neste eixo. Para reforçar essa hipótese, ao deslocar a base manualmente em torno de um dos eixos pode-se ouvir o motor do servo sendo acionado para retorná-lo à posição de referência. Caso o deslocamento seja muito pequeno, o servomotor, em funcionamento, não consegue mover a estrutura. Isso também leva a concluir que, caso o controlador interno dos servos possua um integrador, ele não consegue integrar pequenos erros.

### 4.2.3 Desempenho dos Controladores

Para análise do desempenho dos controladores projetados no Capítulo 3, eles foram colocados à prova em dois testes. No primeiro, foi gerada uma trajetória do alvo com deslocamentos em degrau. No segundo, a trajetória apresentou deslocamentos em rampa. Os movimentos do alvo no mundo resultam em deslocamentos na imagem, considerados como distúrbios na saída

das malhas fechadas de pan e tilt. Desta forma, os testes avaliam a resposta das malhas fechadas para dois tipos de distúrbio na saída, mantendo constante a referência de posição do alvo na imagem. A Figura 4.4 representa o deslocamento horizontal ou vertical do alvo em cada um dos testes, compreendendo dois ciclos de movimento no simulador. Para pequenos ângulos, os deslocamentos angulares do objeto em relação à câmera em pan e tilt também se aproximam de degraus e rampas. Devido à inviabilidade de uso da lente 55-200mm, a ser justificada adiante, os testes foram realizados com uma Nikkor 18-55mm.

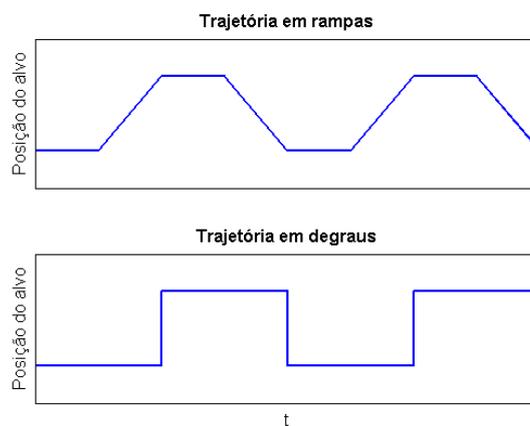


Figura 4.4: Tipo de deslocamento do alvo usado em cada teste.

### Trajetoária em Degraus

Para este teste foi utilizado o simulador de alvos no modo degrau. A Figura 4.5 ilustra o experimento. Uma pequena distância focal, de 24mm, foi ajustada, de modo que o alvo não saísse da imagem ao saltar de um vértice para o próximo. O distúrbio resultante, também em degrau, permitiu verificar o cumprimento dos requisitos definidos no capítulo anterior para o projeto do controlador: *overshoot* menor que 10% e tempo de acomodação menor que 2s.

A Figura 4.6 mostra as respostas da malha de pan ao distúrbio em degrau para cada controlador. Para facilitar a comparação, as janelas de tempo e os eixos verticais estão na mesma escala para ambos. O teste foi realizado ao longo de dois ciclos de movimento do alvo no simulador. O sinal de controle  $u$ , em décimos de grau, se aproxima dos deslocamentos angulares em degraus, na tentativa de acompanhar o movimento do alvo. Mas para o controlador projetado pelo método de Ziegler-Nichols,  $C_p^{zn}$ , pode-se perceber uma maior

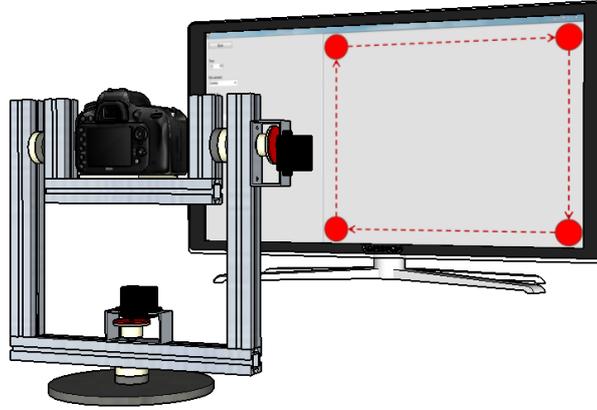


Figura 4.5: Ilustração do teste com trajetória em degraus.

oscilação. Por sua vez, o controlador projetado com auxílio do lugar das raízes,  $C_p^{lr}$ , por possuir um ganho integrativo maior, é mais prejudicado pela histerese em pan, oferecendo grande oscilação em estado estacionário. O gráfico da posição do centroide do alvo no eixo  $x$  da imagem também permite verificar essa oscilação. Apesar disso, verifica-se que  $C_p^{lr}$  resulta numa resposta mais rápida. O valor de referência no centro da imagem ( $x = 320$  pixels) está marcado com a linha pontilhada. Os últimos gráficos mostram melhor as características das respostas ao comparar o melhor desempenho de cada controlador para apenas um dos degraus. Marcadores foram adicionados para cálculo do *overshoot*,  $M_o$ , *undershoot*,  $M_u$ , e tempo de acomodação,  $t_a$ . Foi necessário definir empiricamente o instante de acomodação, devido às oscilações em estado estacionário causadas pela histerese.

$$M_o^{zn} = 100\% \cdot (320 - 262)/(556 - 320) \cong 25\%$$

$$M_o^{rl} = 100\% \cdot (409 - 320)/(320 - 62) \cong 34\%$$

$$M_u^{zn} = 100\% \cdot (439 - 320)/(556 - 320) \cong 50\%$$

$$t_a^{zn} = (1338 - 1283)T = 3,41s$$

$$t_a^{rl} = (2078 - 2035)T = 2,66s$$

Apesar de oferecer menor *overshoot*,  $C_p^{zn}$  apresenta grande *undershoot* e um tempo de acomodação 28% maior que  $C_p^{lr}$ . Nenhum dos controladores cumpriu os requisitos de  $M_o < 10\%$  e  $t_a < 2s$ .

A Figura 4.7 apresenta os dados dos mesmos testes com trajetória em degraus, mas para a malha de tilt. O menor atrito neste eixo de movimento resulta em menor histerese, e os degraus dos sinais de controle estão melhor

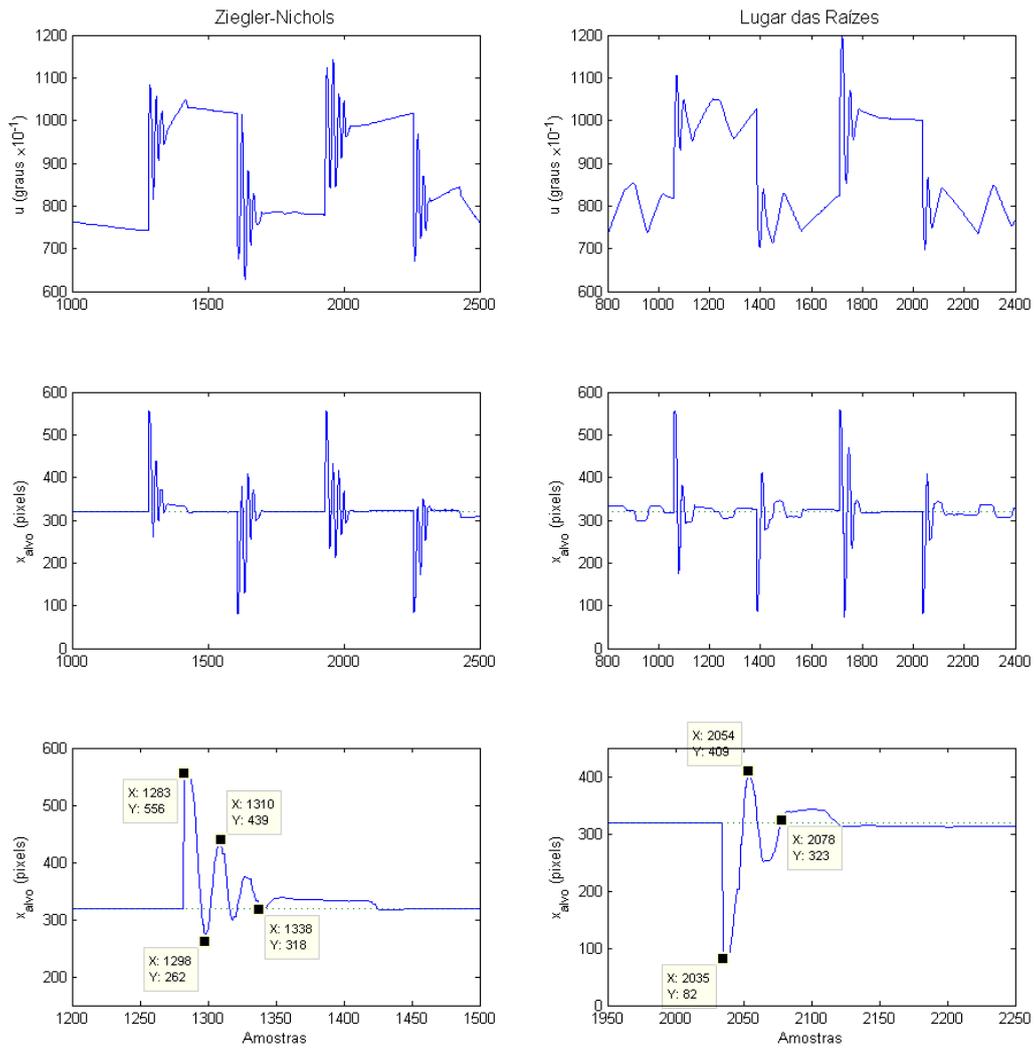


Figura 4.6: Comparação de respostas ao distúrbio em degrau na malha de pan.

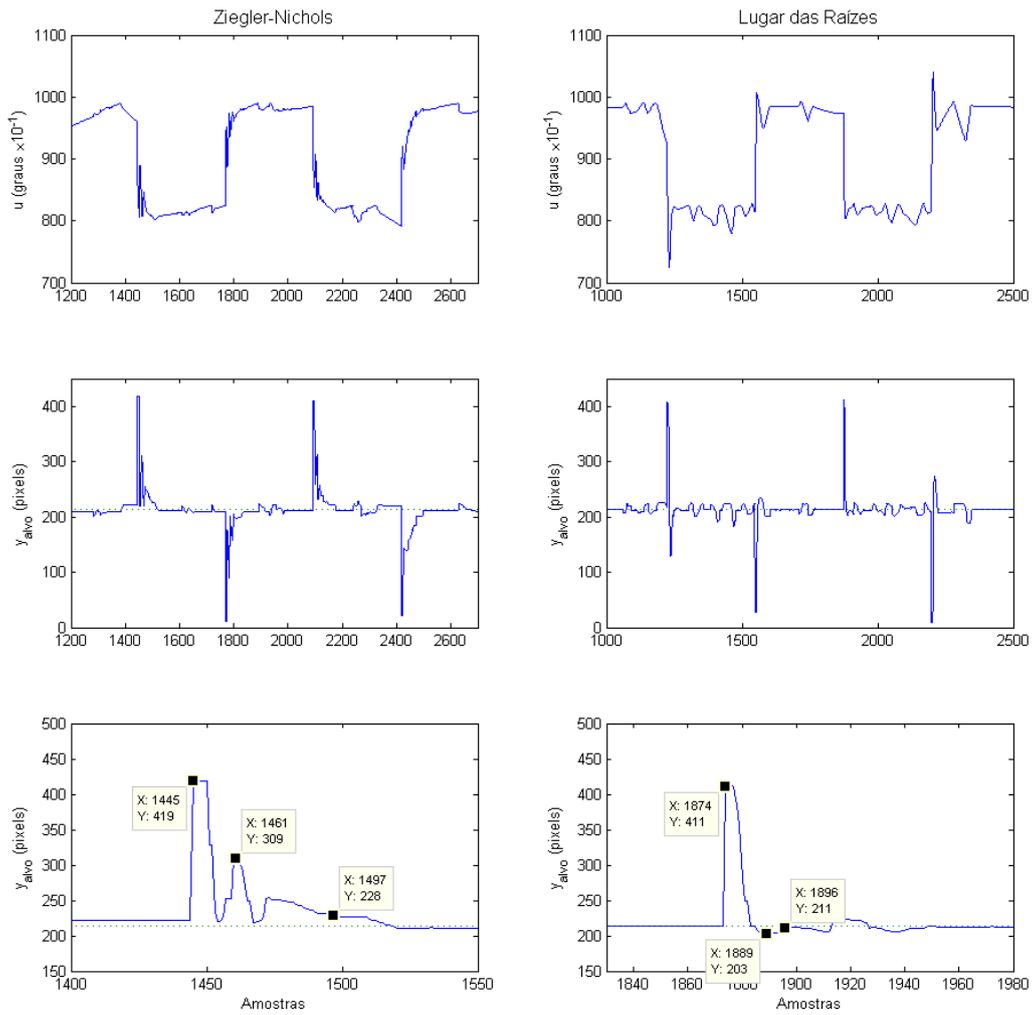


Figura 4.7: Comparação de respostas ao distúrbio em degrau na malha de tilt.

definidos, indicando uma maior facilidade para rastrear o alvo em tilt. Ainda assim, a oscilação em estado estacionário para  $C_t^{lr}$  é significativa. A referência no centro da imagem ( $y = 213$  pixels) está demarcada com a linha pontilhada. Em geral, as mesmas observações qualitativas feitas para os controladores em pan podem ser feitas para tilt:  $C_p^{lr}$  possui menor  $t_a$  e maior  $M_o$ ,  $C_p^{zn}$  é mais estável em estado estacionário mas apresenta grande  $M_u$ . Os seguintes valores comparativos foram calculados para a malha de tilt a partir do melhor desempenho de cada controlador:

$$\begin{aligned} M_o^{rl} &= 100\% \cdot (213 - 203)/(411 - 213) \cong 5\% \\ M_u^{zn} &= 100\% \cdot (309 - 213)/(419 - 213) \cong 46\% \\ t_a^{zn} &= (1497 - 1445)T = 3,22s \\ t_a^{rl} &= (1896 - 1874)T = 1,36s \end{aligned}$$

Observa-se que para a malha de tilt os requisitos de  $M_o < 10\%$  e  $t_a < 2s$  foram atendidos.

### Trajetoória em Rampas

Outro teste foi realizado, utilizando o simulador de alvos em modo rampa, para verificar o desempenho dos controladores no rastreamento de alvos em movimento contínuo. O experimento é ilustrado na Figura 4.8. Neste teste foi ajustada uma distância focal de 55mm. A Figura 4.9 mostra as respostas na malha de pan, e a Figura 4.10 na malha de tilt. Novamente percebe-se a maior dificuldade para acompanhar o alvo na primeira malha. Verifica-se o erro em estado estacionário não nulo para distúrbios em rampa, assim como previsto. Neste quesito, na malha de tilt,  $C_t^{lr}$  leva vantagem sobre  $C_t^{zn}$ , com um erro menor.

Para este teste, foi ajustada uma velocidade do alvo no simulador de 0,09m/s. A velocidade foi determinada medindo-se, na tela em que o simulador foi projetado, um trecho linear do trajeto e o tempo necessário para percorrê-lo. Como a tela estava a uma distância de 1,2m da câmera, temos uma velocidade angular máxima do SARDAC  $\omega = 0,09/1,2 = 0,075\text{rad/s}$ . Esse valor ainda está bem distante daquele definido como meta no Capítulo 3, de 1rad/s. Porém, como pode ser observado na Figura 4.9, essa velocidade já está próxima da instabilidade para  $C_p^{zn}$  na malha de pan.

Buscando-se quantificar o desempenho de cada controlador nas condições deste teste em rampa, os erros de cada um foram medidos dentro das janelas de amostragem mostradas nas Figuras 4.9 e 4.10 através do índice de desempenho IAE (integral do erro absoluto), computado em tempo discreto

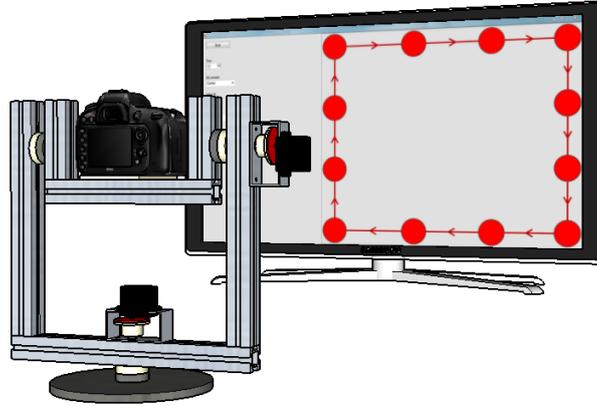


Figura 4.8: Ilustração do teste com trajetória em rampas.

por  $\sum_k |e(k)|$ . Para a malha de pan,  $IAE_p^{zn} = 32600$  e  $IAE_p^{lr} = 29241$ . Na malha de tilt,  $IAE_t^{zn} = 30729$  e  $IAE_t^{lr} = 15406$ . O índice de desempenho usado penaliza pequenos e grandes erros com o mesmo peso. Ainda assim,  $C_p^{lr}$  e  $C_t^{lr}$  levaram vantagem nas malhas, mesmo resultando em oscilação em estado estacionário.

A fim de determinar o desempenho máximo de  $C_p^{lr}$  e  $C_t^{lr}$ , a velocidade do alvo no simulador foi aumentada, mantendo a distância focal de 55mm, até que se atingisse instabilidade nas malhas, particularmente na de pan. A velocidade máxima do alvo que ainda possibilitava o rastreamento é de 0,44m/s a 1,2m de distância, ou velocidade angular máxima de 0,36rad/s, aproximadamente 1/3 da meta. E neste caso, ainda que o SARDAC não tenha perdido o alvo de vista, o rastreamento se mostrou inutilizável para qualquer aplicação prática do sistema, como se pode ver na Figura 4.11. Nos gráficos de localização do alvo na imagem, todos os picos e vales planos indicam que o alvo saiu temporariamente do campo de visão da câmera.

No capítulo anterior foi definido o propósito de utilizar a lente 55-200mm para rastreamento, ajustando a distância focal até os 200mm máximos. Testes foram feitos também com a referida lente, mas o ângulo de visão para  $f = 200\text{mm}$  é tão fechado, que as oscilações em estado estacionário no eixo de pan decorrentes da histerese já eram suficientes para deslocar o alvo de um lado ao outro da imagem. Concluiu-se que o uso da lente 55-200mm fica inviável até que esse problema seja resolvido.

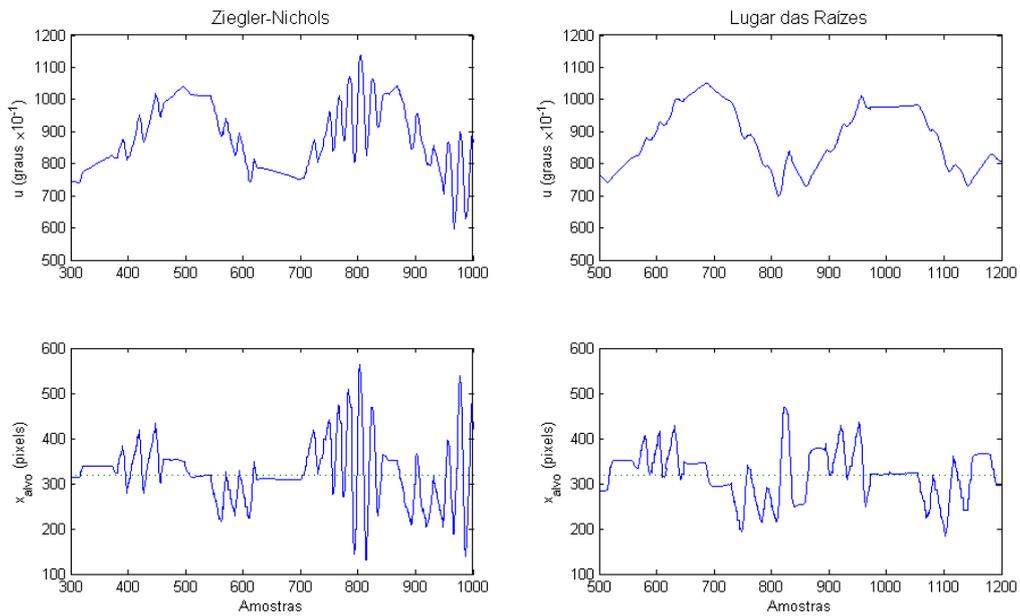


Figura 4.9: Comparação de respostas ao distúrbio em rampa na malha de pan.

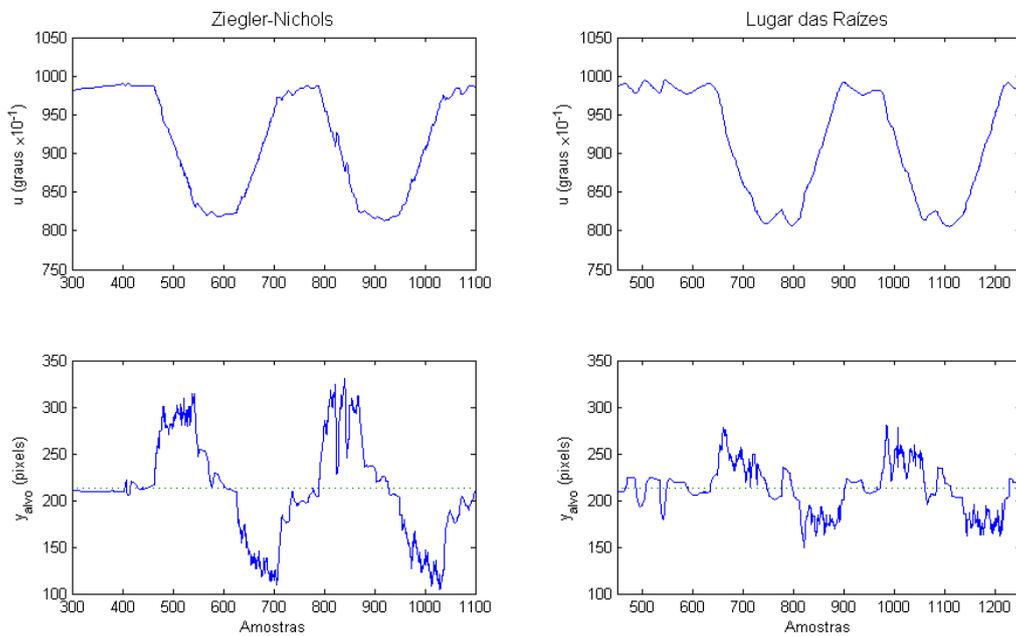


Figura 4.10: Comparação de respostas ao distúrbio em rampa na malha de tilt.

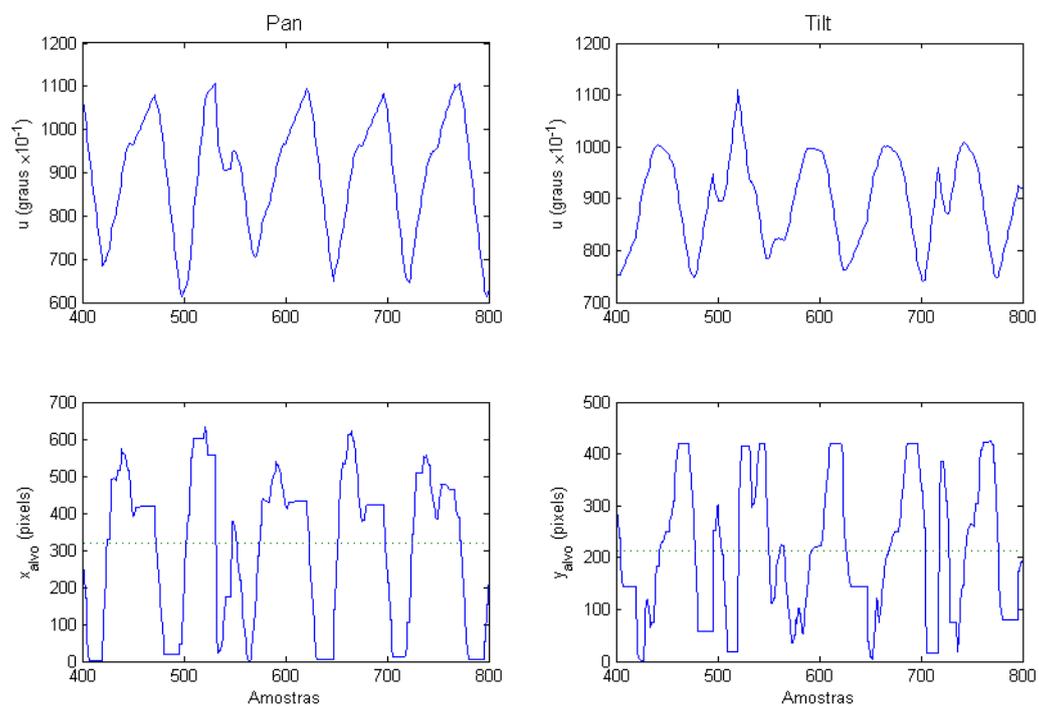


Figura 4.11: Rastreamento em máxima velocidade.

# Capítulo 5

## Conclusões

Este trabalho desenvolveu o SARDAC, um sistema de rastreamento de alvos móveis que opera automaticamente uma base pan-tilt através de técnicas de controle servo visual. A concepção do produto incluiu tanto os componentes de *hardware* quanto de *software*.

### 5.1 Desenvolvimento do SARDAC

A seguir são apresentadas as conclusões referentes a cada uma das etapas de desenvolvimento do projeto: base motorizada, sistema embarcado, software de controle, projeto dos controladores.

#### **Base Motorizada**

A base pan-tilt foi construída a partir de materiais simples, e a usinagem e montagem manual das peças pode ter ocasionado uma falta de precisão nos movimentos. A utilização de perfis modulares de alumínio facilitou a montagem e ofereceu uma boa rigidez e capacidade de carga à estrutura, mas sem deixá-la muito pesada. O maior atrito estático no eixo de pan causou uma histerese significativa no servo que opera este movimento, comprometendo o rastreamento em altas velocidades ou com distâncias focais muito grandes. Apesar desses pequenos problemas estruturais, o custo do SARDAC ficou bem abaixo do preço das bases pan-tilt analisadas, vendidas no exterior, como mostra a Tabela [5.1](#).

#### **Sistema Embarcado**

Um sistema embarcado foi implementado na plataforma Arduino para receber do PC sinais de controle de posicionamento dos eixos e acionar os

|   |                  |
|---|------------------|
| Perfis de alumínio e acessórios de montagem | R\$230,00        |
| Materiais para usinagem                     | R\$80,00         |
| Tripé                                       | R\$100,00        |
| Arduino + <i>shield</i>                     | R\$80,00         |
| Servomotores                                | R\$160,00        |
| <b>Total</b>                                | <b>R\$650,00</b> |

Tabela 5.1: Custos envolvidos na construção do SARDAC.

servomotores de forma correspondente. Essa é uma tarefa simples, que o microcontrolador consegue executar numa velocidade suficiente para não introduzir atrasos nas malhas de controle. É interessante observar que a plataforma Arduino vem diversificando cada vez mais seus de módulos de extensão e funcionalidades de *hardware* e *software*, podendo oferecer futuramente outros recursos úteis para o SARDAC.

### Software de Controle

Todo o *software* de controle do sistema foi desenvolvido a partir do *digicamcontrol*, um programa *open-source* para comunicação e controle remoto de câmeras DSLR. Para o módulo de rastreamento visual foi utilizada a estratégia de mapeamento do alvo no espaço de cores HSL, através do ajuste manual de cada componente do filtro. Essa estratégia se mostrou versátil pela capacidade de adaptação a alvos de características diferentes. Porém, a eficácia é baixa no rastreamento de objetos com superfície complexa, e o ajuste manual a torna inviável para alvos que saem constantemente do campo de visão da câmera. Nestes casos, métodos mais avançados de rastreamento seriam mais adequados.

### Projeto dos Controladores

Para o projeto de controle, o processo foi modelado a partir de dados coletados ao se induzir deslocamentos do alvo na imagem através de movimentos nos eixos pan e tilt. A partir dos modelos obtidos, controladores PI para cada malha foram projetados por meio de dois métodos: regras de sintonia de Ziegler-Nichols para resposta ao degrau, resultando nos controladores  $C_p^{zn}$  e  $C_t^{zn}$ , e alocação de polos e zeros a partir da análise dos lugares das raízes, resultando em  $C_p^{lr}$  e  $C_t^{lr}$ . Em testes de desempenho das malhas de pan e tilt, os controladores *zn* apresentaram menos oscilação em estado estacionário e menor *overshoot*, mas ao custo de maior tempo de acomodação e grande *undershoot*. Por sua vez, os controladores *lr* resultaram em rápida acomoda-

ção e maior facilidade para rastrear alvos em movimento, mas amplificando os efeitos de histerese nos eixos e gerando maior oscilação em estado estacionário. Apenas  $C_t^{lr}$  cumpriu os requisitos de projeto, mas espera-se que  $C_p^{lr}$  também os cumpra se o problema de histerese em pan for minimizado. Conclui-se que os controladores  $zn$  são adequados para aplicações em que o alvo se move lentamente e movimentos mais suaves da câmera são necessários em estado estacionário. Já os controladores  $lr$  são úteis para rastreamento de objetos rápidos. Para aplicações gerais de rastreamento, o controle implementado para o SARDAC é suficiente, em especial para o uso de lentes com menor distância focal. Alvos com velocidades angulares em relação à câmera de até 0,3rad/s podem ser rastreados.

## 5.2 Trabalhos Futuros

O enfoque amplo do trabalho – que envolveu projeto eletro-mecânico, usinagem, visão computacional, modelagem de sistemas, sintonia de controladores, dentre outros tópicos – não permitiu que se aprofundasse em certos aspectos do SARDAC. Isso faz dele uma boa plataforma para pesquisas e trabalhos futuros. Modificações eletro-mecânicas e a elaboração de sistemas de controle mais complexos podem ser feitos para otimizar o desempenho do produto. Além disso, muitas são as possibilidades de ampliação de recursos e funcionalidades. A seguir são sugeridos alguns tópicos para futuras implementações.

### Base Motorizada e Sistema Embarcado

Uma primeira melhoria a ser feita na base deve buscar minimizar a histerese no eixo de pan. A lubrificação do eixo, o uso de rolamentos, ou a troca por um servo que ofereça maior torque para pequenos erros de posição, são possíveis caminhos para contornar o problema. Toda a fixação pode ser internalizada na estrutura, e um suporte pode ser construído para fixar na base o sistema embarcado, que ainda não é “embarcado” na realidade. Na falta de alternativa em mãos, foi usada uma fonte de computador para alimentar os servos. Uma fonte dedicada pode ser adquirida, fornecendo os exatos 7,4V de tensão máxima, e provavelmente melhorando um pouco o desempenho dos motores.

Módulos de extensão para Arduino podem ampliar as funcionalidades do sistema embarcado. Um *shield* de potência pode ser usado para alimentar os servomotores, e módulos wi-fi podem possibilitar uma comunicação sem fio com o PC.

### Software de Controle

Foram apontadas desvantagens no uso do filtro HSL com ajustes manuais para o rastreamento visual. Outros métodos de rastreamento podem ser implementados futuramente, como por região de referência e detecção de fluxo óptico. O próprio AForge Framework usado no *digiCamControl* oferece muitas ferramentas neste sentido. Algumas mudanças na interface do *software* também podem facilitar o uso e agregar funcionalidades: uma sugestão é o uso de cliques na imagem para definir a referência de posição do alvo, ou para mapear o pixel clicado no espaço HSL e utilizar os valores no rastreamento visual.

O controle das malhas pan e tilt talvez seja o elemento que mais ofereça possibilidades de trabalhos futuros. Observou-se na literatura uma gama muito grande de técnicas utilizadas em aplicações de controle servo visual. Os controladores PI desenvolvidos para o SARDAC estão dentre as opções mais simples. Realimentação de estados, controle adaptativo, estimação de estados com Filtro de Kalman, controle preditivo para minimizar o impacto do atraso. Estas são algumas das possibilidades para controle do SARDAC.

# Referências Bibliográficas

- [1] Arduino. Arduino Uno. <http://arduino.cc/>, 2013. Acesso: 15/11/2013.
- [2] X. Clady, François Collange, Frederic Jurie, and Philippe Martinet. Object tracking with a pan-tilt-zoom camera: application to car driving assistance. *IEEE International Conference on Robotics and Automation*, 2:1653–1658, 2001.
- [3] Mark Roberts Motion Control. SFH-30. <http://www.mrmoco.com/products/heads/SFH-30/>, 2013. Acesso: 10/11/2013.
- [4] Armel Crétual and François Chaumette. Dynamic stabilization of a pan and tilt camera for submarine image visualization. *Computer Vision and Image Understanding*, 79(1):47–65, 2000.
- [5] Geraldo Figueiredo da Silveira Filho. *Controle servo visual de veiculos robóticos aéreos*. PhD thesis, Universidade Estadual de Campinas, 2002.
- [6] digiCamControl. <http://digidcamcontrol.com/>, 2014. Acesso: 25/07/2014.
- [7] AForge.NET Framework. <http://www.aforgenet.com/>, 2014. Acesso: 01/11/2014.
- [8] S.N. Fry, M. Bichsel, P. Müller, and D. Robert. Tracking of flying insects using pan-tilt cameras. *Journal of Neuroscience Methods*, 101(1):59–67, 2000.
- [9] MathWorks. Transfer function estimation. <http://www.mathworks.com/help/ident/ref/tfest.html>, 2014. Acesso: 02/11/2014.
- [10] GigaPan. <http://www.gigapan.com/>, 2013. Acesso: 30/09/2013.
- [11] Ismail Haritaoglu, David Harwood, and Larry S. Davis. Active outdoor surveillance. In *Proceedings of the International Conference on Image Analysis and Processing*, pages 1096–1099. IEEE, 1999.
- [12] J. Hill and W. Park. Real time control of a robot with a mobile camera. In *Proceedings of the 9th International Symposium on Industrial Robots*, pages 233–246, 1979.

- [13] Alison E. Hunt and Arthur C. Sanderson. *Vision-based predictive robotic tracking of a moving target*. Carnegie-Mellon University, the Robotics Institute, 1982.
- [14] Davi Yoshinobu Kikuchi. *Sistema de controle servo visual de uma câmera pan-tilt com rastreamento de uma região de referência*. PhD thesis, Universidade de São Paulo, 2007.
- [15] Kazutaka Kurihara, Shin Hoshino, Katsu Yamane, and Yoshihiko Nakamura. Optical motion capture system with pan-tilt camera tracking and realtime data processing. *IEEE International Conference on Robotics and Automation*, 2:1241–1248, 2002.
- [16] Don Murray and Anup Basu. Motion tracking with an active camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):449–459, 1994.
- [17] Nikon. <http://www.nikon.com/>, 2013. Acesso: 04/12/2013.
- [18] Katsuhiko Ogata. *Discrete-time control systems*. Prentice-Hall, Inc., 2nd edition, 1995.
- [19] Katsuhiko Ogata. *Modern Control Engineering*. Prentice-Hall, Inc., 5th edition, 2009.
- [20] Nikolaos P. Papanikolopoulos, Pradeep K. Khosla, and Takeo Kanade. Visual tracking of a moving target by a camera mounted on a robot: a combination of control and vision. *IEEE Transactions on Robotics and Automation*, 9(1):14–35, 1993.
- [21] ServoCity. MPT2100-DS Pan & Tilt System. [http://www.servocity.com/html/mpt2100-ds\\_pan\\_\\_\\_tilt\\_system.html/](http://www.servocity.com/html/mpt2100-ds_pan___tilt_system.html/), 2013. Acesso: 11/11/2013.
- [22] SketchUp. <http://www.sketchup.com/>, 2014. Acesso: 01/11/2014.
- [23] Swivl. <http://www.newswivl.com/>, 2013. Acesso: 10/11/2013.
- [24] Eng Legho Systems. <http://www.engelegho.com.br/>, 2014. Acesso: 01/11/2014.
- [25] Mi-Ching Tsai, K.Y. Chen, M.Y. Cheng, and Ku-Chin Lin. Implementation of a real-time moving object tracking system using visual servoing. *Robotica*, 21(06):615–625, 2003.
- [26] Chun-Kai Wang, Ming-Yang Cheng, Chien-Hsien Liao, Cheng-Chien Li, Chia-Yang Sun, and Mi-Ching Tsai. Design and implementation of a multi-purpose real-time pan-tilt visual tracking system. In *Proceedings of the IEEE International Conference on Control Applications*, volume 2, pages 1079–1084. IEEE, 2004.

- [27] Gang Xu and Takeo Sugimoto. Rits eye: a software-based system for real-time face detection and tracking using pan-tilt-zoom controllable camera. In *Proceedings of the 14th International Conference on Pattern Recognition*, volume 2, pages 1194–1197. IEEE, 1998.