

TRABALHO DE GRADUAÇÃO

**SISTEMA INTEGRADO DE DETECÇÃO E
ACOMPANHAMENTO DE FACES PARA
CÂMERA EM MONTAGEM PAN-TILT**

Murillo Ribeiro de Souza

Brasília, dezembro de 2008



**ENGENHARIA
MECATRÔNICA**
UNIVERSIDADE DE BRASÍLIA

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia

TRABALHO DE GRADUAÇÃO

**SISTEMA INTEGRADO DE DETECÇÃO E
ACOMPANHAMENTO DE FACES PARA
CÂMERA EM MONTAGEM PAN-TILT**

Murillo Ribeiro de Souza

*Relatório submetido ao Departamento de Engenharia
Mecatrônica como requisito parcial para obtenção
do grau de Engenheiro de Controle e Automação*

Banca Examinadora

Prof. Geovany Araújo Borges, ENE/UnB

Orientador

Prof. Pedro de Azevedo Berger, CIC/UnB

Eng. Edson Mintsu Hung, ENE/UnB

Agradecimentos

Agradeço primeiramente aos meus pais, Valdicésio e Lucília, pelo apoio nesta minha jornada pela faculdade, pois confiaram em mim desde o princípio. Também agradeço ao professor Geovany, que deu total suporte ao desenvolvimento do meu trabalho e em minha formação profissional. Agradeço à SAGEM ORGA pela parceria e o apoio no desenvolvimento do meu projeto. Ao pessoal do GPDS, que cedeu espaço para trabalhar e desenvolver minha pesquisa. E um forte abraços aos amigos, que estiveram comigo na faculdade pelos últimos 5 anos.

Murillo Ribeiro de Souza

RESUMO

O rastreamento de objetos em seqüências de imagens é um tema de pesquisa aplicável em diversos campos. Isto é possível devido à existência de excelentes algoritmos de rastreamento e bons computadores digitais. A necessidade do uso do rastreamento em seqüências de imagens existe devido às incertezas na extração de características de faces. Neste projeto, a filtragem estocástica - filtro de Kalman - é usada como uma ferramenta para detecção de características das faces. Estas características serão utilizadas como parâmetros a serem aplicados na estratégia de controle - controle servo-visual. O sinal de controle será enviado para dois motores que serão responsáveis por movimentar o sistema, que contem as *webcams*. Estas câmeras são responsáveis por enviar imagens, que contém faces a serem detectadas. O objetivo do projeto é desenvolver um sistema robusto que controle uma estrutura com câmeras com montagem *pan-tilt* com informações adquiridas do filtro de Kalman, para ser utilizado como base em um sistema de rastreamento de faces para biometria visual 3-D.

ABSTRACT

Object tracking in a sequence of images is a research theme that is applicable on many fields. This is possible due to the excellent tracking algorithms and the lowcost of powerful digital computers. The need of using the tracking in a sequence of images exists because of uncertain in the features extraction of the faces. In this project, the stochastic filtering - Kalman's filter - is used as a tool to the faces features detection. These features will be used as parameters to the employed control strategy - servo-visual control. The control signals are sent to two motors responsible for moving the system, which contains webcams. These cameras are responsible to send images that contain the faces to be detected. The main objective of this project is to develop a robust system that controls a pan-tilt mounted camera structure with information acquired from the Kalman filter, which will be used as the base for a 3-D visual biometry face tracking system.

SUMÁRIO

1	INTRODUÇÃO	1
1.1	RASTREAMENTO	2
1.2	MOTIVAÇÃO	2
1.3	OBJETIVOS DO PROJETO	2
1.4	APRESENTAÇÃO DO TRABALHO	5
2	FUNDAMENTAÇÃO TEÓRICA	6
2.1	FILTRAGEM ESTOCÁSTICA	6
2.1.1	FUNDAMENTAÇÃO	6
2.1.2	FILTRO DE KALMAN	11
2.2	DESCRITORES DE REGIÃO	12
2.2.1	MÉDIA DE REGIÃO	13
2.2.2	COVARIÂNCIA DE REGIÃO	13
2.2.3	APLICAÇÃO DOS DESCRITORES	15
2.3	CONTROLE	16
2.3.1	CONTROLE SERVO-VISUAL	16
3	DESENVOLVIMENTO	18
3.1	INTRODUÇÃO	18
3.2	APLICAÇÃO DO RASTREAMENTO DE FACES	20
3.2.1	RESTRICÇÕES APLICADAS À PREDIÇÃO	22
3.3	DETERMINAÇÃO DE FACES SIMILARES	22
3.4	MONTAGEM FÍSICA DO SISTEMA	23
3.4.1	MONTAGEM DA PARTE MECÂNICA	25
3.4.2	MONTAGEM DA PARTE ELETRÔNICA	25
3.5	ESTRATÉGIAS DE CONTROLE	27
3.5.1	CONTROLE PI DIGITAL <i>anti-windup</i>	27
3.5.2	CONTROLE SERVO-VISUAL ADAPTATIVO	29
4	RESULTADOS EXPERIMENTAIS	32
4.1	INTRODUÇÃO	32
4.2	AVALIAÇÃO DE PARÂMETROS DA FILTRAGEM ESTOCÁSTICA	32
4.3	AVALIAÇÃO DAS ESTRATÉGIAS DE CONTROLE	36
5	CONCLUSÕES	40

5.1	TRABALHOS FUTUROS.....	40
	REFERÊNCIAS BIBLIOGRÁFICAS	42
	ANEXOS.....	43
I	DESCRIÇÃO DO CONTEÚDO DO DVD	44
I.1	PASTAS DO DVD	44
II	DESCRIÇÃO DO FIRMWARE.....	45
II.1	FIRMWARE DESENVOLVIDO PARA O ATMEL(R) ATMEGA8(R)	45

LISTA DE FIGURAS

1.1	Componentes de um sistema de rastreamento típico	3
1.2	Exemplo de aplicação de rastreamento em vigilância e controle do trânsito	3
1.3	Visão geral da montagem e funcionamento do sistema de detecção e acompanhamento de faces.....	4
2.1	Sistema típico do problema da filtragem estocástica	7
2.2	Diagrama do filtro de Kalman.....	11
2.3	Seqüência de imagens utilizadas para teste dos descritores e das medidas de similaridade.....	15
3.1	Fluxografo geral do software de rastreamento e controle	19
3.2	Fluxografo do módulo de rastreamento de faces.....	24
3.3	Montagem do sistema <i>pan-tilt</i> com três câmeras	25
3.4	Sistema de <i>webcam</i> com montagem <i>pan-tilt</i>	26
3.5	Montagem da placa de controle, com as saídas PWM para os servo-motores.....	26
3.6	Estratégia geral de controle do sistema com câmeras com montagem <i>pan-tilt</i>	28
3.7	Modelagem do sistema de servo-controle segundo os eixos X e Z	30
3.8	Gráfico da relação entre distância do face na imagem e o raio da face	31
4.1	Gráfico da evolução da posição X para $r=0.05$ e $q=0.005$	33
4.2	Gráfico da evolução da posição Y para $r=0.05$ e $q=0.005$	34
4.3	Gráfico da evolução do raio R para $r=0.05$ e $q=0.005$	34
4.4	Gráfico da evolução da posição X para $r=0.05$ e $q=0.1$	35
4.5	Gráfico da evolução da posição Y para $r=0.05$ e $q=0.1$	35
4.6	Gráfico da evolução do raio R para $r=0.05$ e $q=0.1$	36
4.7	Gráfico da evolução erro em X para controle CPIA	37
4.8	Gráfico da evolução erro em Y para controle CPIA	38
4.9	Gráfico da evolução erro em X para CSVA	38
4.10	Gráfico da evolução erro em Y para CSVA	39
II.1	Código do firmware - parte1	46
II.2	Código do firmware - parte2	47
II.3	Código do firmware - parte3	48

LISTA DE TABELAS

3.1	Tabela de valores de limiares para informações rastreadas.....	22
3.2	Tabela de valores de ganhos do controle PI <i>anti-windup</i>	29
3.3	Tabela de relação entre distância e o raio da face.....	31
4.1	Valores utilizados para os parâmetros r e q do filtro de Kalman	33

LISTA DE SÍMBOLOS

Índices e valores

k	índice de tempo discreto
f	distância focal

Matrizes e vetores

F	matriz de transformação do processo dinâmico em tempo discreto
H	matriz de observação do filtro de Kalman
I_k	matriz identidade $k \times k$
Q	matriz de covariância da incerteza do processo
R	matriz de covariância da incerteza do ruído da medição
P	matriz de covariância da incerteza das variáveis de estado
z_k	vetor de observação do instante k
y_k	vetor de saída do sistema no instante k
w_k	vetor de ruído de medição, gaussiano
r_k	vetor de ruído de medição, gaussiano
L	matriz de interação
\dot{V}	vetor de velocidades instantâneas
$\dot{\theta}$	vetor de velocidades angulares instantâneas

Sobrescritos

\cdot	variação temporal
$-$	valor médio
\hat{x}_k	estimativa de X no instante k
$\hat{x}_{k/k-1}$	predição de X no instante $k+1$
A^T	transposta da matriz A
A^{-1}	inversa da matriz A

Símbolos Gregos

α	razão entre pixels
Δ	variação entre duas grandezas similares
λ	constante de tempo

Siglas

$N(\mu, \sigma^2)$	Distribuição normal com média μ e variância σ^2
FDP	função densidade de probabilidade
PDI	processamento digital de imagens
CR	controle realimentado
TLC	teorema do limite central
VFO	vetor de faces observadas
VFR	vetor de faces rastreadas
$CPIA$	controle PI <i>anti-windup</i>
$CSVA$	controle servo-visual adaptativo

Capítulo 1

Introdução

Nos últimos anos, houve um grande número de avanços no campo de rastreamento de contornos em seqüência de imagens. Isto ocorreu devido a fatores como a melhoria do poder de processamento dos computadores e nos algoritmos de rastreamento.

A proliferação e a sofisticação dos sistemas de vigilância, tanto civis como militares, geraram bastante interesse em algoritmos capazes de detectar grandes quantidades de objetos usando dados de diversas fontes (sensores). A quantidade de informações enviadas pelos sensores modernos dificulta no processamento de sinais, devido a intensidade de cálculos. Os avanços em hardware e o desenvolvimento de algoritmos mais eficientes nos últimos anos permitiram o aumento do processamento de sinais. Isto levou ao aumento de dados disponíveis à detecção e associação de dados.

Na década de 60, foi reconhecido que havia incerteza associada a objetos rastreados, em adição a sua inacurácia, que é modelada, normalmente, com a adição de um ruído. Esta incerteza adicional esta relacionada com a origem do processo de medição: a medição que será usada no rastreamento pode não ser originada no alvo desejado. Esta situação pode acontecer em um sistema de vigilância quando um radar, sonar ou câmera está operando em um ambiente ruidoso ou com falsos alarmes. Isto também ocorre quando sensores são programados para observar dois ou mais alvos, mas não consegue associá-los às trajetórias. Uma situação similar ocorre quando, em um problema de trajetória, existem vários objetos, mas não se sabe a quantidade e algumas medidas de sensores podem ser espúrias.

A aplicação de algoritmos de estimação simples, usando a medida de objetos em ambientes em que medidas espúrias ocorrem, pode levar a um desempenho inferior do sistema. Isto porque tais algoritmos não levam em consideração que a medida advinda de sensores podem vir de outras fontes que não sejam o alvo em questão [1].

1.1 Rastreamento

O *rastreamento* é o processamento de medidas obtidas de um objeto com objetivo de manter uma estimativa de seu estado atual, que tipicamente consiste de:

1. Componentes cinemáticos (posição, velocidade, aceleração, etc.).
2. Outros componentes (sinal irradiado, características espectrais, informações de *feature*, etc.).
3. Constantes (Coeficientes, velocidade de propagação, etc.).

As *medições* são observações (com ruídos) relacionadas ao estado do objeto. Como exemplo, a medida de posição de faces em imagens (centro e raio da face) e a medida de amplitude de sinais. Este ruído na observação possui várias fontes, podendo ser interferência de outros objetos do sistema ou do próprio objeto de medição.

As medições de interesse em aplicações com múltiplos alvos não são, geralmente, um vetor de pontos, mas, de fato, as saídas de sub-sistemas de processamento de sinais e detecção, como mostrado na Figura 1.1.

1.2 Motivação

Atualmente, os sistemas de segurança tem-se evoluído de modo rápido e constante. Muitos produtos para controle de acesso, vigilância, autenticação de documentos e pessoas estão sendo desenvolvidos e colocados no mercado. As formas são várias: detectores de metais, portas com fechaduras eletrônicas, necessitando-se de leitura de cartão, leitura biométrica (digitais, íris, face, etc.) ou ambas. Na área de vigilância, câmeras com capacidade de rastrear contornos (silhueta do corpo ou face) podem acompanhar movimento de intrusos, facilitando o processo de identificação dos mesmos. Em vigilância no trânsito, pode-se utilizar câmeras para determinar intensidade do tráfego de veículos por uma dada via, como mostrado na Figura 1.2.

Observando-se esta evolução neste ramo, pesquisas que melhorem as técnicas de rastreamento de pessoas e objetos ajudam a aumentar a qualidade das informações recebidas, tornando-se mais rápidas as tomadas de decisões. Neste sentido, a empresa SAGEM ORGA apoia o desenvolvimento dessas novas técnicas.

A detecção de faces pode ser utilizada em câmeras que se movam, acompanhando objetos de interesse. Além disso, pode ser utilizada para extrair características da face, utilizando tais informações para autenticação de usuários de qualquer sistema.

1.3 Objetivos do projeto

O trabalho aqui descrito trata de um sistema em tempo real que rastreia N faces em seqüência de imagens enviadas por uma câmera, determinando seus centros, raios e profundidades e enviando

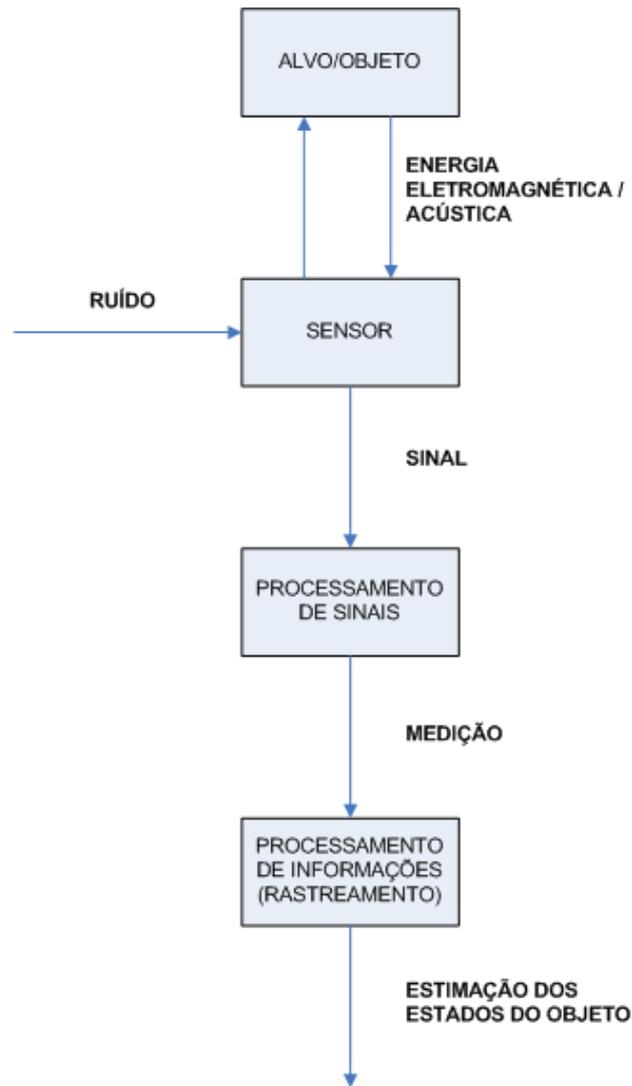


Figura 1.1: Componentes de um sistema de rastreamento típico

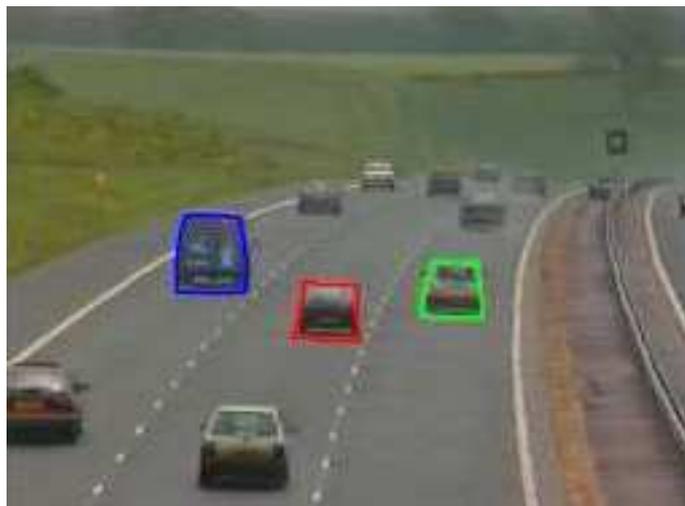


Figura 1.2: Exemplo de aplicação de rastreamento em vigilância e controle do trânsito

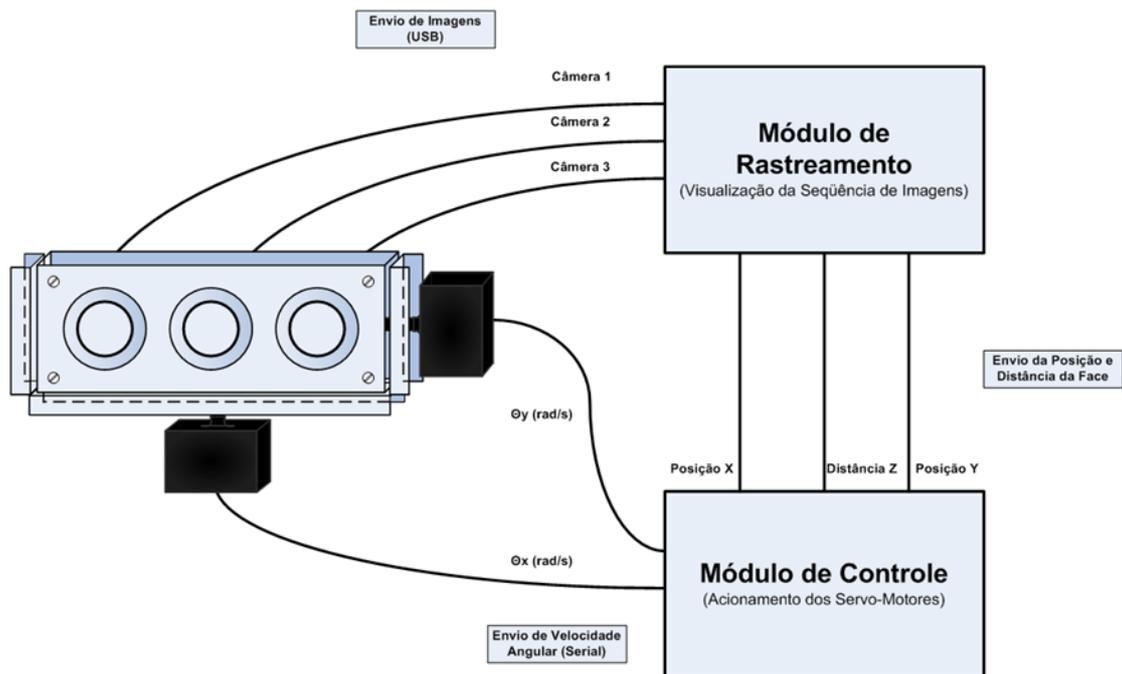


Figura 1.3: Visão geral da montagem e funcionamento do sistema de detecção e acompanhamento de faces

estas informações para um controlador que atual sobre motores, fazendo com que sistema centre a face da primeira pessoa a aparecer na imagem. O sistema explora as vantagens da utilização do filtro de Kalman e controle servo-visual adaptativo.

O sistema utiliza a filtragem estocástica para melhorar a qualidade da informação advindo do sistema de sistema de detecção de faces. O sistema de controle atua para manter centrada a face desejada.

Ao final, foram colhidos dados para mostrar como está a dinâmica do sistema de rastreamento e controle. São retiradas as devidas conclusões sobre os resultados obtidos.

O objetivo final deste projeto é criar uma base para a segunda fase do projeto (em conjunto com a SAGEM ORGA) que será o desenvolvimento de um sistema de rastreamento de faces para biometria visual 3-D.

O projeto pode ser entendido como uma malha fechada, que contém os motores como atuadores, as câmeras e o módulo de detecção como sistemas de medição e o módulo de controle como o controle do sistema. A Figura mostra uma visão geral do sistema.

O projeto do sistema foi completo, isto é, partiu-se de uma idéia inicial do sistema (de um sistema que fizesse detecção de faces e pude-se acompanhá-la) e todos os módulos (funções) e a montagem física (mecânica e eletrônica) foram desenvolvidas durante a execução do projeto.

1.4 Apresentação do trabalho

O trabalho é dividido em quatro partes. No capítulo 2, é feita uma revisão bibliográfica sobre filtragem estocástica, descritores de regiões e controle servo-visual adaptativo. Em seguida, o capítulo 3 descreve como foram formuladas as questões envolvidas na revisão bibliográfica. Resultados experimentais são discutidos no capítulo 4, em que dados da dinâmica do sistema são apresentados, seguidos das conclusões no capítulo 5. Os anexos contêm material complementar, que são o firmware da parte de acionamento dos motores e conteúdo do DVD que acompanha este trabalho.

Capítulo 2

Fundamentação Teórica

Uma revisão sobre a filtragem estocástica (filtro de Kalman). Além disso, uma pequena passagem por descritores de região e finalmente uma descrição sobre a principal técnica de controle utilizada.

2.1 Filtragem Estocástica

A utilização de filtragem estocástica traz maior qualidade ao processo de detecção de faces no sistema, pois é uma técnica que permite rastreamento de múltiplos alvos em sistemas em que as medições contêm incertezas, assim como apresentado na seção *Rastreamento* do capítulo 1. Neste trabalho, o rastreamento será feito com a utilização do filtro de Kalman. A seção 2.1.1 contém uma pequena teoria envolvida no desenvolvimento do filtro e na seção 2.1.2, a solução geral do problema do filtro de Kalman.

2.1.1 Fundamentação

A filtragem pode ser entendida em dois sentidos. Imagine uma quantidade que se deseja $s(\cdot)$, que não se pode medir. Suponha que também há uma segunda medição, $z(\cdot)$, ruidosa, que é relacionada com $s(\cdot)$ e pode ser medida. O primeiro conceito de filtragem, mais genérico, é que a filtragem é a recuperação de $s(\cdot)$ a partir de $z(\cdot)$. A medida ruidosa é utilizada para determinar características internas do sistema. A segunda definição é que a filtragem é utilizada para distinguir características a partir de dois tipos de informações relacionadas: a medição e a predição [2].

Modelo de tempo discreto de GAUSS-MARKOV

O sistema para o qual o filtro será desenvolvido é o sistema que segue o modelo de tempo discreto de GAUSS-MARKOV e possui a forma de um sistema linear, dimensional finito que servirá como o modelo do sinal desejado apresentada na Figura 2.1. Tem-se, deste sistema, a seguinte formulação

$$x_{k+1} = F_k x_k + G_k w_k \quad (2.1)$$

$$z_k = y_k + v_k = H_k^T x_k + v_k \quad (2.2)$$

Nesta representação, tem-se que x_k é o estado do sistema no instante de tempo kT . O valor

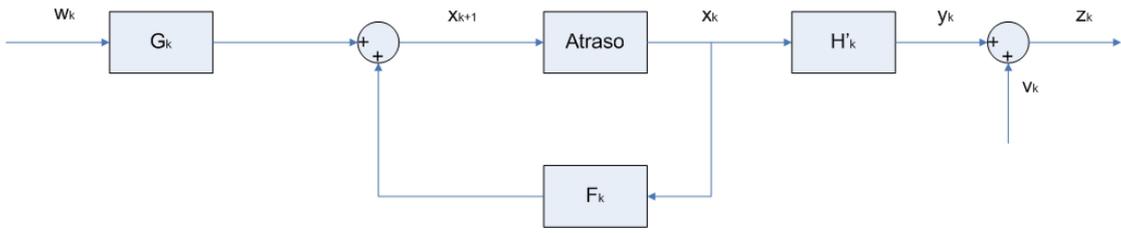


Figura 2.1: Sistema típico do problema da filtragem estocástica

de w_k é o ruído do processo. As matrizes F_k e G_k representam as matrizes de transformação do sistema e do ruído, respectivamente. Estas matrizes contém as informações sobre a dinâmica do sistema e do ruído. A matriz H_k representa a matriz de observação, que transforma o vetor de estados na saída do sistema. Geralmente a saída do sistema é dada apenas por

$$y_k = H_k^T x_k \quad (2.3)$$

Contudo, foi adicionado um ruído v_k , relacionado ao efeito de variáveis externas ao processo de medição. Assim, surge a medida do processo z_k . A entrada do sistema w_k também é ruidosa. Objetivo será aplicar a filtragem sobre um sistema discreto como o apresentado. Isto significa utilizar as medidas z_0, z_1, \dots, z_k para extrair o valor de x_k . As variáveis v_k e w_k são ruídos brancos Gaussianos, de média nula e independentes. Sendo v_k um ruído branco, a densidade disjunta de probabilidade é produto das densidades individuais. A função densidade de probabilidade de uma variável gaussiana é totalmente descrita por sua média e variância, como mostrado nas equações abaixo.

$$m_v = E\{v\} \quad (2.4)$$

$$R = E\{(v - m_v)(v - m_v)^T\} \quad (2.5)$$

Quando v possui dimensão n e R é não singular, a função densidade de probabilidade (FDP) desta distribuição é dada por

$$P_v = \frac{1}{(2\pi)^{n/2} |R|^{1/2}} \exp\left\{-\frac{1}{2}(v - m_v)^T R^{-1}(v - m_v)\right\} \quad (2.6)$$

Uma vez que v_k é um ruído branco, tem média 0 e variância dada por $R = E\{v_k v_k^T\}$.

Com essas características, tem-se que, para dados v_k e v_l , com k diferente de l , tem-se

$$E\{v_k v_l^T\} = E\{v_k\} E\{v_l^T\} = 0 \quad (2.7)$$

A primeira igualdade é válida devido ao fato do ruído do sistema ser branco. A segunda porquê o sinal é de média nula. Pode-se ver que $v \sim N(0, \sigma_v^2)$. Conseqüentemente, se é sabido que $E\{v_k v_k^T\} = R_k$, tem-se que

$$E\{v_k v_l^T\} = R_k \lambda_{kl} \quad (2.8)$$

Em que λ_{kl} é o delta de Kronecker, dado por

$$\lambda_{kl} = \begin{cases} 1, & k = l \\ 0, & k \neq l \end{cases} \quad (2.9)$$

Estes mesmos resultados são obtidos e válidos para a variável $\{w_k\}$.

Estado inicial

O estado inicial do sistema ocorre em $k=0$, em que o vetor x_0 é um vetor qualquer. Tem-se que x_0 deve ser uma variável randômica, em que

$$\bar{x}_0 = E\{x_0\} \quad (2.10)$$

$$P_0 = E\{(x_0 - \bar{x}_0)(x_0 - \bar{x}_0)^T\} \quad (2.11)$$

A variável x_0 é independente de v_k e w_k . Assumir que o modelo gaussiano é valido para descrever o processo permite a aplicação do *teorema do limite central*, em que processos não gaussianos são aproximados por uma gaussiana [3]. Também o problema de filtragem são mais simples de se resolverem quando as variáveis são assumidas como sendo gaussianas.

Propriedades gaussianas e de MARKOV do sistema

A variável x_k possui três importantes características. A primeira é que x_k é uma variável gaussiana randômica. Isto ocorre devido às características de $\{w_k\}$ e do estado inicial $\{x_0\}$, tornando x_k gaussiana para qualquer k arbitrário. Pode-se ver que

$$x_k = \phi_{k,0} x_0 + \sum_{l=0}^{k-1} \phi_{k,l-1} G_l w_l \quad (2.12)$$

$$\phi_{k,l} = F_{k-1} F_{k-2} \dots F_l \quad (2.13)$$

Observa-se que x_k é uma combinação linear de x_0 e de w_0, w_1, \dots, w_l .

A segunda propriedade é que x_k é um processo gaussiano randômico. Pode-se demonstrar que para $k_i (i = 1, 2, \dots, m)$, o conjunto de variáveis x_k é gaussiano. A terceira propriedade é que x_k é um processo de MARKOV. Isto significa que para $x_{k_i} (k_1 < k_2 < \dots < k_m < k)$, a densidade de probabilidade de x_k dada por x_{k_1}, \dots, x_{k_m} é, simplesmente

$$p(x_k | x_{k_1}, x_{k_2}, \dots, x_{k_m}) = p(x_k | x_{k_m}) \quad (2.14)$$

Essa característica se deve a w_k e à casualidade do sistema descrita pela equação 2.1. Pode-se ver, então que

$$x_k = \phi_{k,k_m} x_{k_m} + \sum_{l=k_m}^{k-1} \phi_{k,l-1} G_l w_l \quad (2.15)$$

O conhecimento dos valores de $x_{k_1}, x_{k_2}, \dots, x_{k_m}$ não é necessário, uma vez que somente x_{k_m} é relevante para determinar x_k . Quanto ao processo de medição z_k , também é uma gaussiana. Isto é, de fato, verdadeiro pelas mesmas razões de x_k . As variáveis x_k e z_k são conjuntamente gaussianas. Contudo, z_k não é MARKOV. Isto porque y_k não é um processo branco. O significado desta afirmação é que y_{k-2} e y_{k-1} contém mais informações sobre y_k do que apenas y_{k-1} .

Propagação de médias de variâncias

As variáveis x_k e z_k são conjuntamente gaussianas. Assim, dependem exclusivamente de suas médias e variâncias. Tem-se, então, como demonstrado na equação 2.15, o valor de $\{x_k\}$, com $E\{x_k\} = \phi_{k,0} x_0$ e $E\{x_{k+1}\} = F_k E\{x_k\}$. Já o valor de $\{z_k\}$ foi obtido na equação 2.2, com $E\{z_k\} = H_k^T E\{x_k\}$. Em relação às suas covariâncias, temos a seguinte formulação

$$P_{x_k, l} = \{(x_k - \bar{x}_k)(x_l - \bar{x}_l)^T\} \quad (2.16)$$

Substituindo o resultado obtido em 2.15 em 2.16, tem-se

$$P_{x_k, l} = E\left\{[\phi_{k,0}(x_0 - \bar{x}_0) + \sum_{m=0}^{k-1} \phi_{k,m-1} G_m w_m][\phi_{l,0}(x_0 - \bar{x}_0) + \sum_{n=0}^{l-1} \phi_{l,n-1} G_n w_n]^T\right\} \quad (2.17)$$

$$P_{x_k, l} = \phi_{k,l} \left\{ \phi_{l,0} P_0 \phi_{l,0}^T + \sum_{m=0}^{l-1} \phi_{l,m+1} G_m Q_m G_m^T \phi_{l,m+1}^T \right\} \quad (2.18)$$

Para o caso em que $k=l$, tem-se

$$P_{x_k, k} = P_k = E\{(x_k - \bar{x}_k)(x_k - \bar{x}_k)^T\} \quad (2.19)$$

Culminando no resultado abaixo

$$P_{x_{k+1}} = F_k P_k F_k^T + G_k Q_k G_k^T \quad (2.20)$$

Em que P_k é a matriz de covariância dos estados do sistema e Q_k é a matriz de covariância da incerteza do processo.

A covariância para o processo z_k é dada por

$$\text{cov}(z_k, z_l) = E\{(z_k - \bar{z}_k)(z_l - \bar{z}_l)^T\} \quad (2.21)$$

$$\text{cov}(z_k, z_l) = E\{H_k^T (x_k - \bar{x}_k) x_l - \bar{x}_l)^T H_l\} + E\{H_k^T (x_k - \bar{x}_k) v_l^T\} + E\{v_k (x_l - \bar{x}_l)^T H_l\} + E\{v_k v_l^T\} \quad (2.22)$$

Como a variável x_k é independente de v_l , a covariância do processo em questão é dada por

$$\text{cov}(z_k, z_l) = H_k \phi_{k,l} P_l H_l + R_k \delta_{k,l} = H_k^T P_k \phi_{k,l} H_l + R_k \delta_{k,l} \quad (2.23)$$

Critério de estimação

A idéia da estimação é determinar informações de uma variável a partir de outra. Assim, o objetivo é obter informações de x_k a partir de z_k .

Para isso, deve-se entender o conceito da probabilidade condicional. Assim, para duas variáveis aleatórias X e Y

$$P_{X|Y}(X|Y) = \frac{P_{X,Y}(X,Y)}{P_Y(Y)} \quad (2.24)$$

Com $P_Y(Y) \neq 0$.

Estimativa da menor variância de X dado $Y=y$

A função $P_{X|Y}(X|Y)$ é uma função que gera como saída um número ou um vetor. Deve-se encontrar um valor de x que maximize $P_{x|y}(X|Y)$, obtendo-se uma estimativa. Denote-se por \hat{x} a estimativa tomada pelo vetor X . Temos que \hat{x} não é igual a x . Para medir este erro, tem-se

$$E\{\|x - \hat{x}\|^2 | Y = y\} \quad (2.25)$$

Em que $\|a^2\| = a'a$.

Tem-se que \hat{x} é um número fixo. Contudo, o conhecimento de y não faz com que x deixe de ser uma variável aleatória. Definimos como estimativa de menor variância de \hat{x} como aquela que

$$E\{\|x - \hat{x}\|^2 | Y = y\} \leq E\{\|x - z\|^2 | Y = y\} \quad (2.26)$$

Em que z seja qualquer vetor, determinando de algum modo a partir de y .

O valor de \hat{x} é único e possui outros nomes, como estimativa de menor média quadrática. A principal propriedade do estimador de variância mínima está descrita no seguinte teorema: *sejam X e Y dois vetores com distribuição conjunta aleatória e que a medida de Y seja o valor dado por y . Seja \hat{x} a estimativa de menor variância de X como definida acima. Então \hat{x} é definido unicamente como sendo a média condicional de X dado $Y=y$, ou seja,*

$$\hat{x} = E\{X|Y = y\} = \int_{-\infty}^{+\infty} x P_{X|Y}(X|Y) dx \quad (2.27)$$

Propriedade do estimador de variância mínima

Este estimador de mínima variância é dada por

$$E\{\|X - \hat{X}(Y)\|^2\} \leq E\{\|X - Z(Y)\|^2\} \quad (2.28)$$

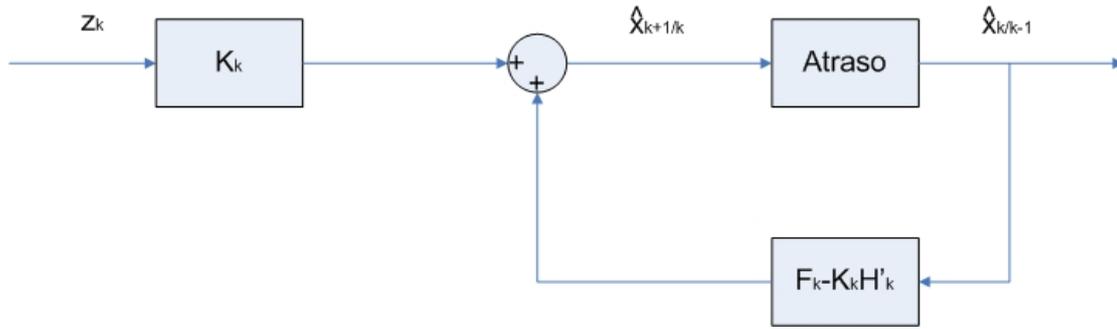


Figura 2.2: Diagrama do filtro de Kalman

Desta formulação, surge um importante teorema, que será a base para o desenvolvimento do filtro de Kalman e é descrito por: *Dados X e Y , dois vetores aleatórios conjuntos. O estimador de menor variância \hat{X} de X , dado Y é*

$$\hat{X} = E\{X|Y\} \quad (2.29)$$

2.1.2 Filtro de Kalman

O problema do filtro de Kalman

Para o sistema da Figura 2.1, definido para $K \geq 0$, suponha que $\{v_k\}$ e $\{w_k\}$ são independentes, com média nula, ruídos brancos gaussianos com

$$E[v_k v_k^T] = R_k \delta_{kl} \quad (2.30)$$

$$E[w_k w_k^T] = Q_k \delta_{kl} \quad (2.31)$$

Suponha também que o estado inicial x_0 é gaussiano com média \bar{x}_0 e covariância P_0 , independente de $\{v_k\}$ e $\{w_k\}$. Deve-se determinar as estimativas

$$\hat{x}_{k/k-1} = E[x_k | Z_{k-1}] \quad (2.32)$$

$$\hat{x}_{k/k} = E[x_k | Z_k] \quad (2.33)$$

E o erro associado às matrizes de covariância $P_{k/k-1}$ e $P_{k/k}$.

A solução para o problema do filtro de Kalman

O filtro de Kalman possui a forma mostrada na Figura 2.2. Trata-se de um sistema que recebe a saída de valores de estados do processo descrito pela Figura 2.1 e de posse de parâmetros, como os valores das matrizes F_k e H_k , por exemplo, estima o valor dos estados de x_k .

O sistema proposto para o filtro é modelado da seguinte maneira

$$\hat{x}_{k+1/k} = (F_k - K_k H_k^T) \hat{x}_{k/k-1} + K_k z_k \quad (2.34)$$

O ganho da matriz K_k é dado pelo da matriz de erros de covariâncias por

$$K_k = F_k P_{k/k-1} H_k [H_k^T P_{k/k-1} H_k + R_k]^{-1} \quad (2.35)$$

Assumindo que a inversa existe, e a matriz de erro de covariância é dada recursivamente pela equação de Riccati

$$P_{k+1/k} = F_k [P_{k/k-1} + P_{k/k-1} H_k [H_k^T P_{k/k-1} H_k + R_k]^{-1} H_k^T P_{k/k-1}] F_k^T + G_k Q_k G_k^T \quad (2.36)$$

Esta equação é inicializada por P_0 . Pode-se obter, então

$$\hat{x}_{k/k} = \hat{x}_{k/k-1} + P_{k/k-1} H_k [H_k^T P_{k/k-1} H_k + R_k]^{-1} (z_k - H_k^T \hat{x}_{k/k-1}) \quad (2.37)$$

$$P_{k/k} = P_{k/k-1} - P_{k/k-1} H_k [H_k^T P_{k/k-1} H_k + R_k]^{-1} H_k^T P_{k/k-1} \quad (2.38)$$

Podemos derivar também, dessas equações a seguinte relação

$$\hat{x}_{k+1/k} = F_{k/k} \hat{x}_{k/k} \quad (2.39)$$

$$\sum_{k+1/k} = F_k \sum_{k/k} F_k^T + G_k Q_k G_k^T \quad (2.40)$$

Para a utilização do filtro de kalman, passa-se por duas etapas. A primeira é a de predição do estado e da covariância do sistema e é descrita pelas equações 2.39 e 2.40. A outra etapa é a de correção dos valores utilizando-se a medição e é descrito pelas equações 2.37 e 2.38.

2.2 Descritores de região

A seleção de características é um dos passos mais importantes em problemas de detecção e de classificação. Boas características devem ser robustas e fáceis de computar e algoritmos eficientes são necessários para tarefas como reconhecimento e rastreamento.

O conjunto de valores de imagens tais como cores, gradientes e respostas a filtros são as respostas mais simples para escolha de características de imagens. Nesta linha, será proposto o primeiro descritor de regiões, que é calculado relacionado-se a média das componentes RGB dos pixels de uma região selecionada.

Contudo, estas características não são robustas na existência de variação de luminosidade e movimentação de objetos na imagem. Para superar estas limitações, será proposto um segundo descritor de regiões, que é obtido relacionando-se os valores de covariâncias dos pixels da região de interesse.

Como será demonstrado em seguida, o primeiro descritor é bastante suscetível à luminosidade, enquanto o segundo, não [4].

2.2.1 Média de região

O cálculo da média RGB é uma operação relativamente simples, pois envolve apenas a soma dos valores das componentes RGB de cada pixel, dividido pelo número de *pixels* da região. Denote-se por R uma região contida em uma imagem I (três dimensões de cor), cuja largura seja w e altura h . Denote-se um vetor \bar{V} , tal que: $\bar{V} = (M_R \ M_G \ M_B)$, que contém a média de cada componente de cor. As equações 2.41, 2.42 e 2.43 mostram o procedimento para cálculo das médias.

$$M_R = \frac{1}{N} \sum_{j=0}^{h-1} \sum_{i=0}^{w-1} R_{ij} \quad (2.41)$$

$$M_G = \frac{1}{N} \sum_{j=0}^{h-1} \sum_{i=0}^{w-1} G_{ij} \quad (2.42)$$

$$M_B = \frac{1}{N} \sum_{j=0}^{h-1} \sum_{i=0}^{w-1} B_{ij} \quad (2.43)$$

Em que R_{ij} , G_{ij} e B_{ij} são os valores das componentes de cor em um dado pixel na posição (i, j) . Os valores de i e j estão limitados pela largura w e a altura h da imagem. A valor de N , número de pixels, é dado pela multiplicação dos valores w e h .

Os descritores são utilizados para cálculo de similaridade entre duas regiões. No caso do descritor calculado a partir das médias das cores dos pixels, a diferença entre duas regiões (R_1 e R_2) é dada pela diferença de distância (D) entre as médias RGB de cada região e pode ser obtido a partir da equação 2.44.

$$D = \sqrt{(M_{R1} - M_{R2})^2 + (M_{G1} - M_{G2})^2 + (M_{B1} - M_{B2})^2} \quad (2.44)$$

2.2.2 Covariância de região

Como formulado no início da seção 2.2.1, imagine-se uma região R em uma imagem I . Esta região pode ser representada em termos de sua covariância P pela seguinte relação.

$$P = \frac{1}{N-1} \left(\sum_{j=0}^{h-1} \sum_{i=0}^{w-1} (V_{ij} - \bar{V})(V_{ij} - \bar{V})^T \right) \quad (2.45)$$

Em que N é igual ao produto h e w . A matriz P é 3×3 . O vetor V representa os valores RGB de cada pixel ij da região R .

Diferentemente do cálculo de distância do descritor calculado pela média, o cálculo de distância pela covariância não está no espaço Euclidiano. Para isso, deve-se usar o cálculo de similaridade entre duas matrizes de covariância, que é dado por

$$D_c = \sqrt{\sum_{i=1}^n \ln^2(\lambda_i(\overline{P}_1, \overline{P}_2))} \quad (2.46)$$

Em que $\{\lambda_i(\overline{P}_1, \overline{P}_2)\}_{i=1\dots n}$ representam os autovalores generalizados de P_1 e P_2 , computados por

$$\lambda_i \overline{P}_1 x_i - \overline{P}_2 x_i = 0 \quad (2.47)$$

E x_i os autovetores generalizados.

Autovalores e autovetores generalizados

O descritor de regiões proposto nesta seção lida com o problema dos autovalores generalizados, necessário para o cálculo de medida de similaridade, apresentado na subseção anterior. O problema é descrito pela equação 2.48.

$$Ax = \lambda Bx \quad (2.48)$$

Em que A e B são matrizes, diferentes da matriz identidade. Do problema da medida de similaridade, tem-se que as matrizes A e B podem ser os valores das matrizes de covariância de duas regiões similares (P_1 e P_2) em uma seqüência de imagens.

Na maioria dos problemas, A e B não são singulares e pode-se lidar com o sistema da seguinte maneira.

$$(B^{-1}A)x = \lambda x \quad (2.49)$$

Muitas vezes, A e B são simétricas e B é positiva definida. Entenda-se por simétrica, uma matriz M , quando $m_{ij} = m_{ji}$. E positiva definida quando $vMv > 0$ para todos os vetores. A matriz $(B^{-1}A)$ não é simétrica, mas pode-se recorrer ao problema do autovalor simétrico utilizando a decomposição de Cholesky.

Para se entender a decomposição de Cholesky, imagine uma matriz quadrada M , simétrica e positiva definida. Então ela tem uma decomposição triangular mais eficiente. Quando é possível utilizar, o algoritmo de Cholesky é duas vezes mais rápido para resolver equações lineares do que algoritmos alternativos. Ao invés de procurar arbitrariamente por matrizes triangulares LU, a decomposição de Cholesky constrói uma matriz L tal que a sua transposta L^T serve como a parte superior da decomposição triangular. Pode-se então fazer

$$LL^T = M \quad (2.50)$$

Esta fatoração é chamada também da *raiz quadrada de M*. Os elementos das matriz L são dados da seguinte maneira.

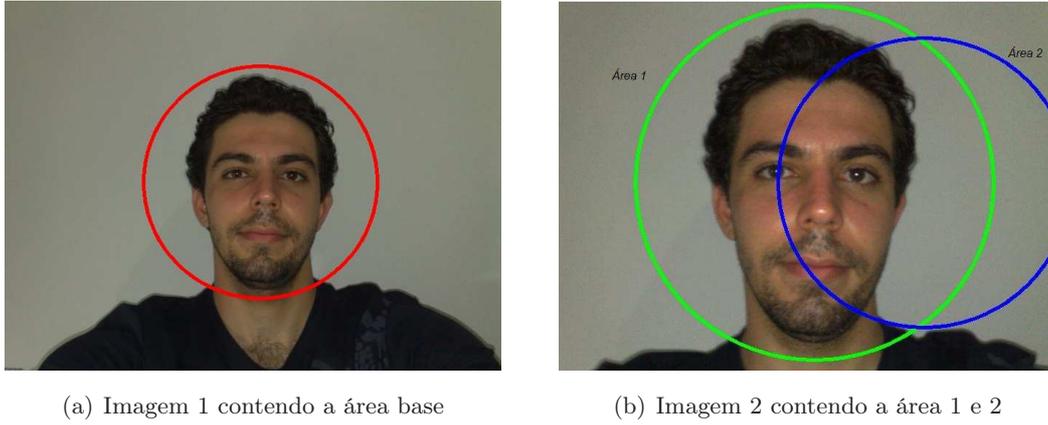


Figura 2.3: Seqüência de imagens utilizadas para teste dos descritores e das medidas de similaridade

$$L_{ii} = (a_{ii} - \sum_{k=1}^{i-1} L_{ik}^2)^{1/2} \quad (2.51)$$

$$L_{ji} = \frac{1}{L_{ii}} (a_{ij} - \sum_{k=1}^{i-1} L_{ik} L_{jk})^{1/2} \quad (2.52)$$

Para $j = i + 1, i + 2, \dots, N$.

Voltando ao problema dos autovalores genéricos, da decomposição de Cholesky, tem-se que $B = LL^T$. Multiplicando L^{-1} na equação 2.48, tem-se

$$C(L^T x) = \lambda(L^T x) \quad (2.53)$$

Em que,

$$C = L^{-1} A (L^{-1})^T \quad (2.54)$$

A matriz C é simétrica e possui os mesmos autovalores do problema original (2.48). [5]

2.2.3 Aplicação dos descritores

Esta seção tem por objetivo mostrar alguns resultados obtidos utilizando-se os descritores propostos nas seções 2.2.1 e 2.2.2. O procedimento utilizado basea-se no cálculo de similaridade entre os descritores de três regiões. Uma região, que é a região base, encontra-se na imagem 1 e é em relação a ela que serão calculadas medidas de similaridade com as áreas 1 e 2, presentes na imagem 2. A Figura 2.3 indica a localização de cada área na seqüência de imagens. A determinação destas áreas foi feita de modo manual, mas de maneira a selecionar, na imagem 2 um região similar e outra não similar. Observando-se esta figura, vê-se que a mesma face aparece nas duas imagens, mas em posições e com iluminações diferentes. Para se testar as medidas de similaridade, determinou-se a área 1, que cobre toda a face e a área 2, que cobre a parcialmente a face.

Calculando-se os descritores de cada região, temos os seguintes resultados para as médias

$$MED_B = \begin{pmatrix} 44 \\ 60 \\ 49 \end{pmatrix} \quad (2.55)$$

$$MED_1 = \begin{pmatrix} 57 \\ 80 \\ 58 \end{pmatrix} \quad (2.56)$$

$$MED_2 = \begin{pmatrix} 132 \\ 146 \\ 132 \end{pmatrix} \quad (2.57)$$

Em que MED_B , MED_1 e MED_2 correspondem à média RGB da área base, área 1 e área 2, respectivamente. O cálculo de medida de similaridade resultou nos valores $Dm_{B-1} = 25,5$ e $Dm_{B-2} = 148,42$, respectivos às medidas de similaridade entre a área base e as áreas 1 e 2.

Calculando-se os descritores de cada região, temos os seguintes resultados para as covariâncias

$$COVB = \begin{pmatrix} 1552,81 & 1611,16 & 1505,32 \\ 1611,16 & 1933,14 & 1909,96 \\ 1505,32 & 1909,96 & 1949,78 \end{pmatrix} \quad (2.58)$$

$$COV1 = \begin{pmatrix} 1710,98 & 1641,77 & 1508,49 \\ 1641,77 & 1912,77 & 1882,62 \\ 1508,49 & 1882,62 & 1924,39 \end{pmatrix} \quad (2.59)$$

$$COV2 = \begin{pmatrix} 1220,71 & 1293,41 & 1245,80 \\ 1293,41 & 1662,21 & 1701,60 \\ 1245,80 & 1701,60 & 1795,93 \end{pmatrix} \quad (2.60)$$

O cálculo de medida de similaridade resultou nos valores $D_{C-B1} = 11,86$ e $D_{C-B2} = 20,68$.

Como era esperado, as duas técnicas de medida de similaridade indicaram que a face na área 1 é mais similar à área base do que a área 2.

2.3 Controle

2.3.1 Controle servo-visual

Controle servo-visual se refere ao uso de dados de visão computacional para controlar movimentação de robôs, ou no caso deste trabalho, controlar a movimentação de servo-motores. Os dados de visão devem vir de uma câmera montada diretamente sobre o robô ou em um manipulador, em que o movimento do robô induz a câmera a se mover. As informações também podem vir de uma câmera de montagem fixa, observando o movimento do robô.

O controle servo-visual se utiliza de técnicas de processamento de imagem, visão computacional e teoria de controle.

Definição do problema

O objetivo da estratégia de controle é minimizar o erro de referência do sistema, tal que

$$e(t) = s^* - s(m(t), a) \quad (2.61)$$

Em que $m(t)$ são as medidas retiradas da imagem (coordenadas do ponto de interesse, por exemplo). O valor a representa o conjunto de parâmetros intrínsecos da câmera.

Estes conjuntos de valores são responsáveis por gerar o vetor de características da imagem, $s(m(t), a)$, que no caso deste trabalho representam informações do centro da face principal, que será explicado no próximo capítulo. O vetor s^* contém os valores desejados, ou seja, a referência de posição da face principal nas imagens vindas da *webcam*. Assim sendo, o controle do sistema tem por objetivo centrar a face principal na imagem, de tal forma que o sistema foque e acompanhe a referência desta face.

Uma vez que s esteja selecionado, o desenvolvimento do esquema de controle passa a ser a próxima tarefa. Geralmente, a melhor abordagem é o controle da velocidade. Para fazer tal controle, é necessário saber a relação entre a variação no tempo de s e a velocidade da câmera. Denote-se a velocidade da câmera como sendo $V_c = (v_c, \omega_c)$, em que v_c em a velocidade linear instantânea e ω_c a velocidade angular instantânea. A relação entre \dot{s} e V_c é dada por

$$\dot{s} = L_s V_c \quad (2.62)$$

Em que L_s é a matriz de interação.

A matriz de interação é uma matriz que relaciona a velocidade de movimentação da câmera (esta movimentação pode ser linear ou angular) e a variação da posição de um dado objeto na imagem. Assim, a matriz de interação relaciona a movimentação do objeto no espaço e sua movimentação na plano da imagem.

Utilizando-se as informações contidas nas equações 2.61 e 2.62, tem-se:

$$\dot{e} = L_e V_c \quad (2.63)$$

Em que $L_e = L_s$.

Considerando V_c como sendo o valor que será enviado ao servo-motores, isto é, o sinal de controle, para uma queda exponencial do erro, tem-se:

$$V_c = -\lambda L_e^{-1} e \quad (2.64)$$

Esta formulação será utiliza na estratégia de controle, sendo que a definição da matriz L_e será apresentada no capítulo 3 [6].

Capítulo 3

Desenvolvimento

Apresentação das restrições e formulação utilizadas no desenvolvimento do trabalho, além da montagem física do sistema.

3.1 Introdução

Nos capítulos anteriores, foi apresentada parte da teoria utilizada no desenvolvimento deste trabalho. Neste capítulo, será apresentada como as técnicas mostradas foram utilizadas, além de maiores detalhes sobre a montagem e programação do sistema.

A Figura 1.3, apresentada no Capítulo 1 contém informações sobre a montagem geral do sistema. Tem-se uma parte mecânica, que corresponde à montagem das câmeras com dois graus de liberdade (montagem pan-tilt). A parte de acionamento dos motores, denominada parte eletrônica e finalmente o software responsáveis pelos cálculos do filtro de Kalman, descritores de região e do controle.

Simplificadamente, o software se divide em duas *threads* separadas. A principal contém a inicialização de variáveis e a parte responsável pelo rastreamento da imagem vinda da *webcam*. Esta thread também é responsável por enviar dados da realimentação da malha de controle. E a thread secundária, chamada controle, é a responsável pelo controle de velocidade do sistema montado. A Figura 3.1 mostra um fluxograma geral do sistema.

Este software desenvolvido para o projeto foi estruturado em C, utilizando-se o software Visual C++ 2005 [7] e as APIs OpenCV v1.1 [8], VideoInput v1.1.1 e GMATRIX [9]. A API OpenCV é responsável por mostrar imagens na tela, bem como acessar informações de pixels, além de fazer o trabalho de se encontrar faces nas imagens vindas da *webcam*. A VideoInput é responsável por permitir que o sistema possa receber imagens de mais de uma câmera. Esta característica é importante para a segunda fase deste projeto, que será o desenvolvimento de rastreamento de faces para biometria 3-D. A API GMATRIX será responsável pelos cálculos matriciais em todos os módulos: rastreamento, descritores de região e controle.

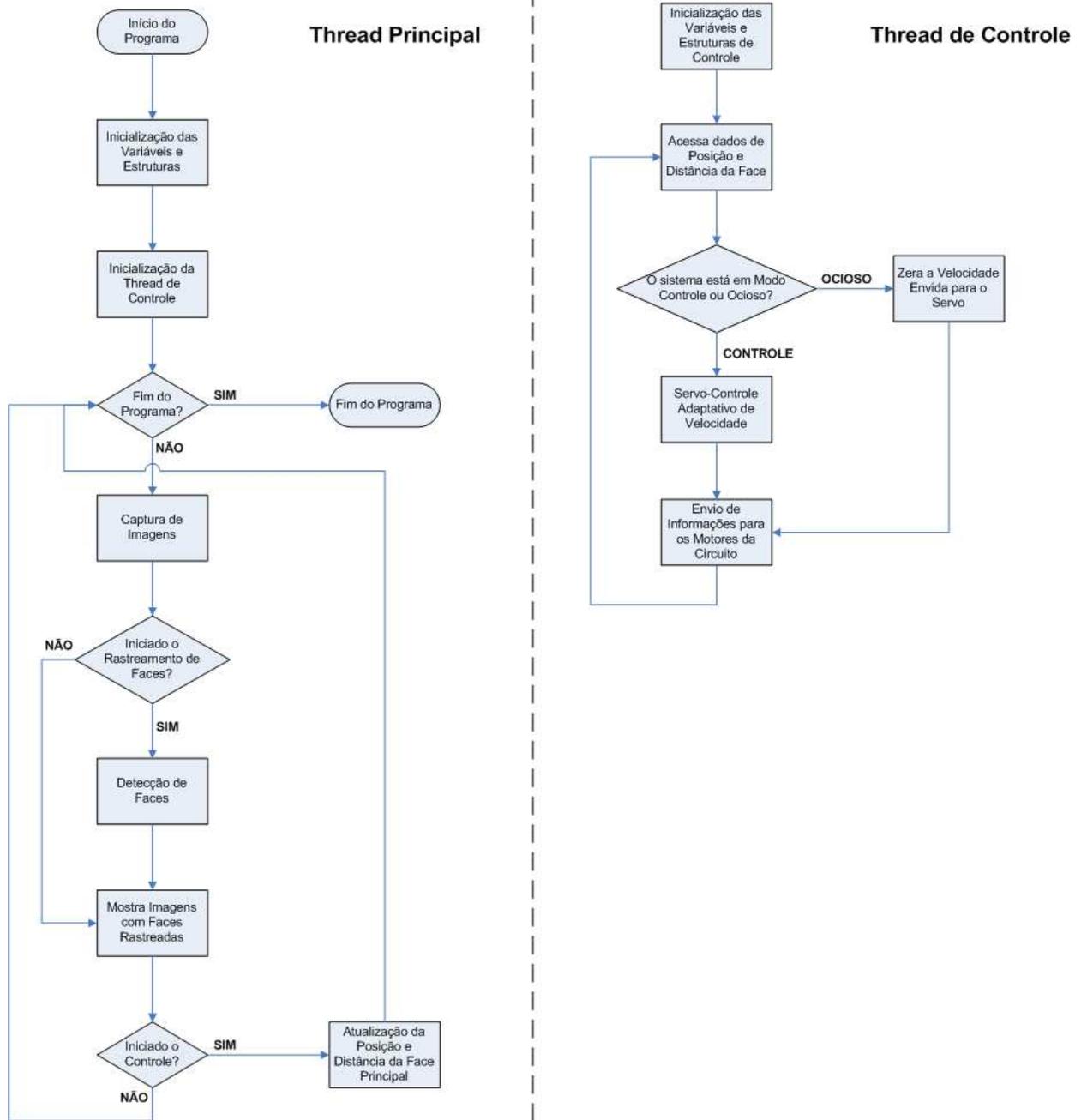


Figura 3.1: Fluxografo geral do software de rastreamento e controle

3.2 Aplicação do rastreamento de faces

Como já citado, a API OpenCV foi utilizada para processar as imagens vindas da *webcam*. A partir de um arquivo de configurações, o sistema aplica a função chamada *cvHaarDetectObjects*, responsável por extrair os dados das faces presentes no frame vindo da câmera. Estes dados são o valor do centro da face (x, y) e raio do círculo que representa o contorno do rosto. Para a modelagem do sistema, usar-se-á o modelo de 1a. ordem. Neste modelo, considera-se que tanto o centro da face quanto o raio se movimentam com velocidade constante

$$x = x_0 + \dot{x}T \quad (3.1)$$

$$y = y_0 + \dot{y}T \quad (3.2)$$

$$r = r_0 + \dot{r}T \quad (3.3)$$

Em que T representa o período entre rastreamento de faces entre duas imagens seqüenciais. Os valores \dot{x} , \dot{y} e \dot{r} representam as velocidades das coordenadas x e y e do raio da face, respectivamente.

O vetor de estados, que é alimentado com essas informações, deverá conter os valores de posições e velocidades que representam cada face, como mostrado abaixo.

$$\hat{x}_k = \begin{pmatrix} x_k \\ \dot{x}_k \\ y_k \\ \dot{y}_k \\ r_k \\ \dot{r}_k \end{pmatrix} \quad (3.4)$$

Os valores de velocidade não são medidos diretamente, mas estimados através do processo de correção do filtro de Kalman. A escolha do modelo de 1a. ordem faz com que o valor de w_k seja zero e equação 2.1 seja reduzida à forma

$$x_{k+1} = F_k x_k \quad (3.5)$$

Neste caso, a matriz F_k , da transformação do processo dinâmico, possui seguinte forma

$$F_k = \begin{pmatrix} 1 & T & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & T & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & T \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (3.6)$$

A matriz Q_k , que representa a covariância do incerteza do processo é dada por

$$Q_k = \begin{pmatrix} \frac{1}{3}T^3 & \frac{1}{2}T^2 & 0 & 0 & 0 & 0 \\ \frac{1}{2}T^2 & T & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{3}T^3 & \frac{1}{2}T^2 & 0 & 0 \\ 0 & 0 & \frac{1}{2}T^2 & T & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{3}T^3 & \frac{1}{2}T^2 \\ 0 & 0 & 0 & 0 & \frac{1}{2}T^2 & T \end{pmatrix} \quad (3.7)$$

Outro aspecto importante a se considerar é que na equação $P_{k+1/k} = F_k P_{k/k} F_k^T + G_k Q_k G_k^T$ (predição do filtro de Kalman), a matriz G_k será modelada como a matriz identidade I_6 multiplicado por um parâmetro q configurável. Este parâmetro está diretamente ligado à calibração do filtro de Kalman. Um valor menor de q indica que o processo é incerto e a predição possuirá maior influência no valor do centro da face e seu raio. Um valor maior, contudo, indica que o processo de medição será mais influente na detecção da face.

Para a etapa de correção, os valores medidos da função de encontra faces na imagem vinda da câmera, contidos no vetor Z (equação 3.8), são a posição (x, y) e o raio das faces apenas. As velocidades dessas componentes não são observadas. Sua forma é mostrada na equação abaixo

$$z = \begin{pmatrix} \hat{x}_k \\ \hat{y}_k \\ \hat{r}_k \end{pmatrix} \quad (3.8)$$

A matriz H_k , responsável pela adequação dos valores de medida aos estados do sistema do sistema possuirá a forma mostrada em equação 3.9

$$H_k = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \quad (3.9)$$

Ainda na etapa da correção, a matriz R_k , da covariância do processo de medição, será também a identidade I_3 , multiplicada por um parâmetro r configurável no sistema. Este valor representa a incertaza do processo de medição. Se o valor de r for aumentado, a medição dos estados do sistema são consideradas mais incertas, levando a maior influência da predição dos estados. Um valor menor, torna a observação mais influente. O parâmetro r e q são valores que calibram a dinâmica do filtro de Kalman

As condições iniciais do sistema são da seguinte maneira. A matriz de covariância das incertezas das variáveis de estado P_0 é igual a I_6 . Já o vetor de estados estimados \hat{x} inicia com os valores dos estado X , Y e R iguais aos observados naquele instante, sendo que \dot{X} , \dot{Y} e \dot{R} iniciam com o valor 0.

Tabela 3.1: Tabela de valores de limiares para informações rastreadas

Limiar	Valor
<i>MIN_COORD_X</i>	0
<i>MAX_COORD_X</i>	320
<i>MIN_COORD_Y</i>	0
<i>MAX_COORD_Y</i>	240
<i>MIN_RADIUS</i>	5
<i>MAX_RADIUS</i>	120

3.2.1 Restrições aplicadas à predição

A etapa da predição é realizada antes da etapa de correção. O que acontece com filtro de Kalman é que a partir do momento em que não são observadas faces compatíveis da imagem (tema abordado na seção 3.3), apenas a etapa de predição é realizada. Isto pode gerar alguns problemas se a velocidade de variação dos parâmetros estiver com módulo relativamente alto. Após algumas etapas de predição, o sistema pode levar a, por exemplo, posições de face fora da imagem, isto é, ultrapassam os limites, o que pode gerar erros de processamento. O mesmo pode ocorrer para o raio. Se o valor de variação do raio estiver menor que zero e com módulo com valor alto, pode-se obter um valor de raio negativo, o que gera um erro de processamento. Para resolver tal problema, foi necessária a aplicação de certos limiares máximos e mínimos para os valores de centro da face e raio. A tabela 3.1 mostra os valores destes limiares. Estes valores são adimensionais e representam a quantidade de pixels. Os limiares da posição do centro da face estão relacionados à dimensão da imagem enviada pela *webcam*. A dimensão é 320×240 . Quanto ao limiar do raio, um raio menor que 5 não é de fácil visualização e algo maior que 120 já estaria fora da imagem enviada.

3.3 Determinação de faces similares

Como descrito na seção 2.2 do capítulo anterior, os descritores de região são valores que descrevem características de cada região selecionada em imagens vindas da câmera. E estes valores são comparados de alguma de duas formas com valores de outras regiões para se definir a medida de similaridade. Para tanto, foi apresentado dois descritores e duas medidas de similaridade extraídas a partir dos valores daqueles. Para melhorar o acerto das medidas, foi acrescentada também a informação de distância entre as faces. Isto foi feito calculando-se além da distância via descritores, a distância entre o centro das faces e adicionando esta informação à primeira calculada.

No projeto, a medida de similaridade baseada na covariância é a principal medida. Contudo, os dois descritores são utilizados, pois o algoritmo de Cholesky, apresentado em 2.2.2 e utilizado no cálculo do descritor de região baseado na covariância, pode apresentar erros, dependendo dos valores das matrizes envolvidas (uma divisão por zero, por exemplo). Assim, usa-se o outro descritor que, apesar de apresentar resultados mais fracos, ainda é um bom descritor e cobre possíveis falhas do primeiro.

Para se determinar se uma imagem saída da etapa de predição possui observação válida, aplicam-se os descritores e as medidas de similaridades. Primeiramente, deve-se definir os dois vetores, o vetor faces rastreadas (VFR) e o vetor de faces observadas (VFO). O VFO é um vetor conseguido a partir da função da API OpenCV apresentada anteriormente. O VFR é o vetor que contém dados das faces rastreadas. Os passos podem ser descritos da seguinte forma

1. Percorre-se o VFO.
2. Verifica-se se a posição em questão possui similaridade a outra face do VFR.
3. Se sim, passa-se à próxima face do VFO. Caso contrário, aplica-se o descritor de região baseado na covariância.
4. Verifica se houve problema. Se não, observa-se se a medida de similaridade esta abaixo de um limiar definido por *LIM_COV*.
5. Se houve problema, aplica-se o descritor baseado na média e verifica se o resultado esta abaixo do limiar *LIM_MED*
6. Se a medida estiver abaixo dos limiares, passa-se as próximas etapas do filtro de Kalman. Se não, passa-se à próxima face do VFO

Os valores de *LIM_COV* e *LIM_MED* são respectivamente 55 e 65, valores estes obtidos experimentalmente. A obtenção destes valores foi feita observando-se o resultado advindo dos descritores de região eo quanto estava-se errando ou acertando na medida de similaridade. Um valor menor, faz com que faces similares, de fato, não sejam consideradas similares. Já um valor maior, aumenta a taxa de erro na medida de similaridade, podendo-se duas regiões diferentes serem consideradas similares.

Caso alguma face do VFR não encontre similar no VFO por cinco vezes consecutivas (valor definido pela variável *SCORE*), então ela é retirada do VFR. Nestas cinco vezes sem observação, acontece apenas a etapa de predição do filtro de Kalman.

A Figura 3.2 mostra como sistema de detecção de faces foi estruturado. Este fluxografo indica os passos mostrados nesta seção e também as etapas de aplicação do filtro de Kalman.

3.4 Montagem física do sistema

O sistema de rastreamento de faces foi montado utilizando-se câmeras na configuração pan-tilt. Isto permite ao sistema dois graus de liberdade. A estrutura contendo as *webcams* pode se mover em torno do eixo X e Y, permitindo que exista controle de posição, colocando-se a face principal (primeira posição não-nula do vetor de faces) no centro das imagens. Existe um pequeno hardware montado sob a estrutura da câmera, que possui um microcontrolador ATMEL(R) ATMEGA8(R) [10], responsável por receber informações da porta serial do computador, processar e enviar para o motor. Na subseção 3.4.1, serão descritos alguns detalhes da montagem do sistema. No subseção 3.4.2, descrito o hardware e o firmware presente no microcontrolador.

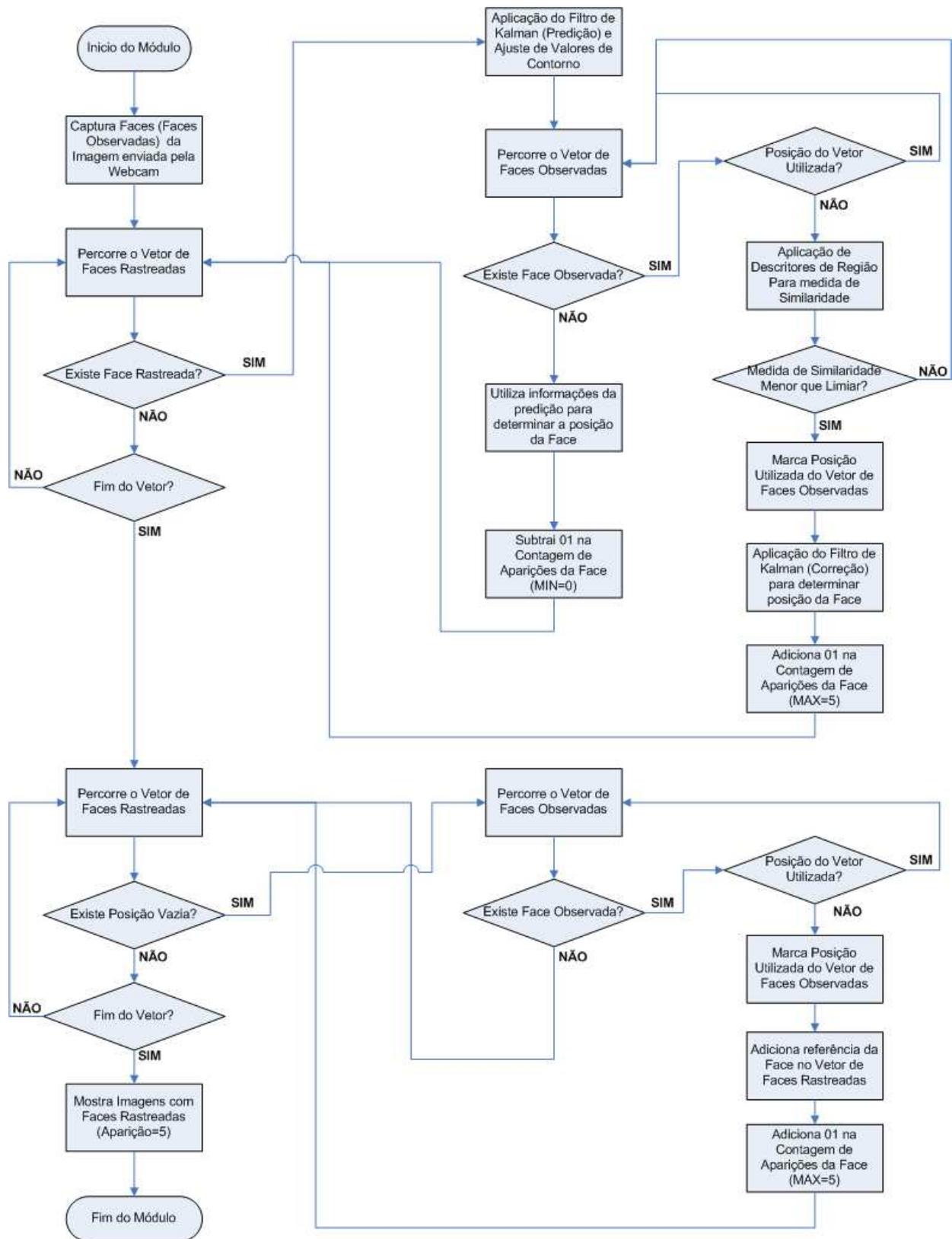


Figura 3.2: Fluxografo do módulo de rastreamento de faces

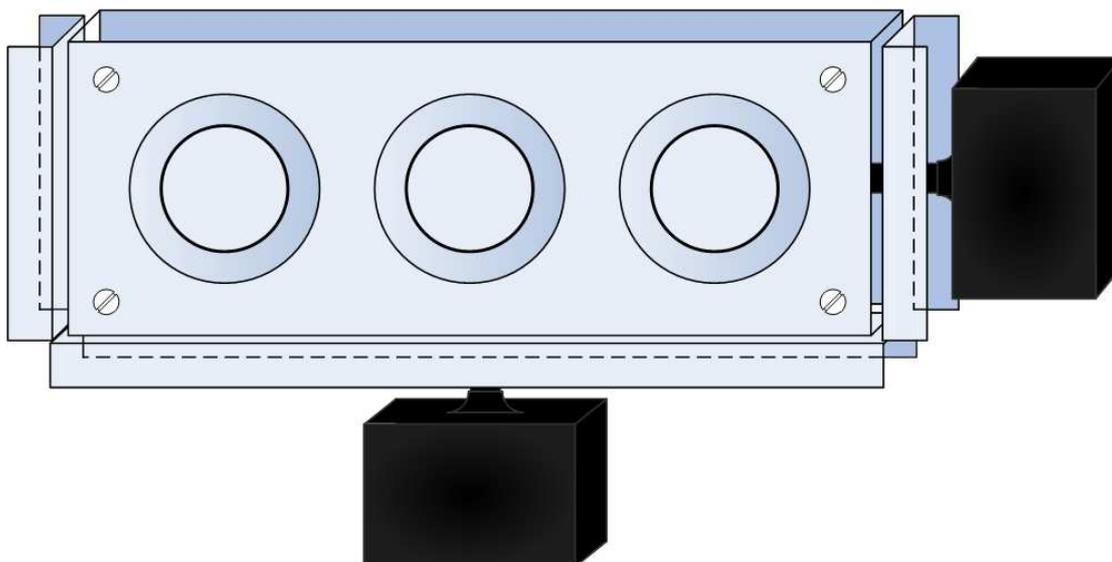


Figura 3.3: Montagem do sistema *pan-tilt* com três câmeras

3.4.1 Montagem da parte mecânica

Para a montagem mecânica, foram utilizadas três *webcams*. Apesar de, neste projeto, apenas uma câmera fornecer as imagens, esta montagem será utilizada para outras finalidades, que farão parte da próxima etapa deste projeto, descrita no capítulo 1. A Figura 3.3 mostra o esquema de montagem *pan-tilt* do sistema.

O sistema foi montado sobre o motor responsável pelo movimento de rotação sobre o eixo Y. O movimento deste motor gera o movimento de translação na imagem no eixo X. Isto é, a rotação deste motor desloca um dado ponto, projetado no plano da imagem, em relação ao eixo X. O segundo está fixada na estrutura, no eixo de rotação X. Seu movimento gera translação no eixo Y. Esta forma de montagem permite que se tenha dois graus de liberdade. Isto será fundamental para a estratégia de controle desejada, pois permite a independência dos eixos e controle individual de cada motor.

A montagem desta estrutura é de grande facilidade, devido a sua simplicidade. Contudo, esta montagem sobre o eixo do motor pode gerar alguma vibração no sistema, quando este estiver se movimentando. Apesar disto, esta dificuldade não afeta o funcionamento do sistema, nem desestabiliza o sistema.

A Figura 3.4 é uma fotografia retirada do sistema real, montado em laboratório. Na caixa preta, abaixo do sistema, se encontra a placa de controle dos motores.

3.4.2 Montagem da parte eletrônica

A parte eletrônica montada tem o objetivo de promover o acionamento para a parte mecânica. A Figura 3.5 mostra a estrutura da montagem.

A estrutura foi montada sobre uma placa perfurada, aonde foram soldados os principais componentes. O sistema provê comunicação serial (RS-232) a 9600 bauds, montado sob a supervisão

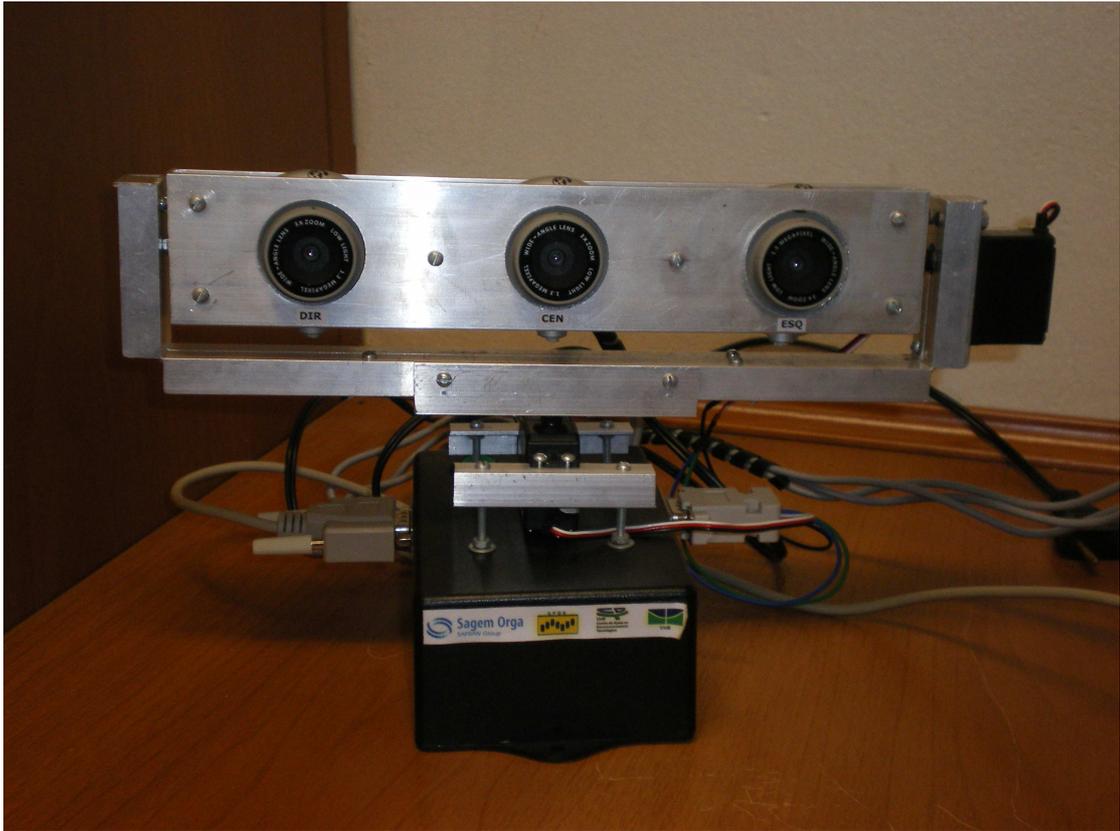


Figura 3.4: Sistema de *webcam* com montagem *pan-tilt*

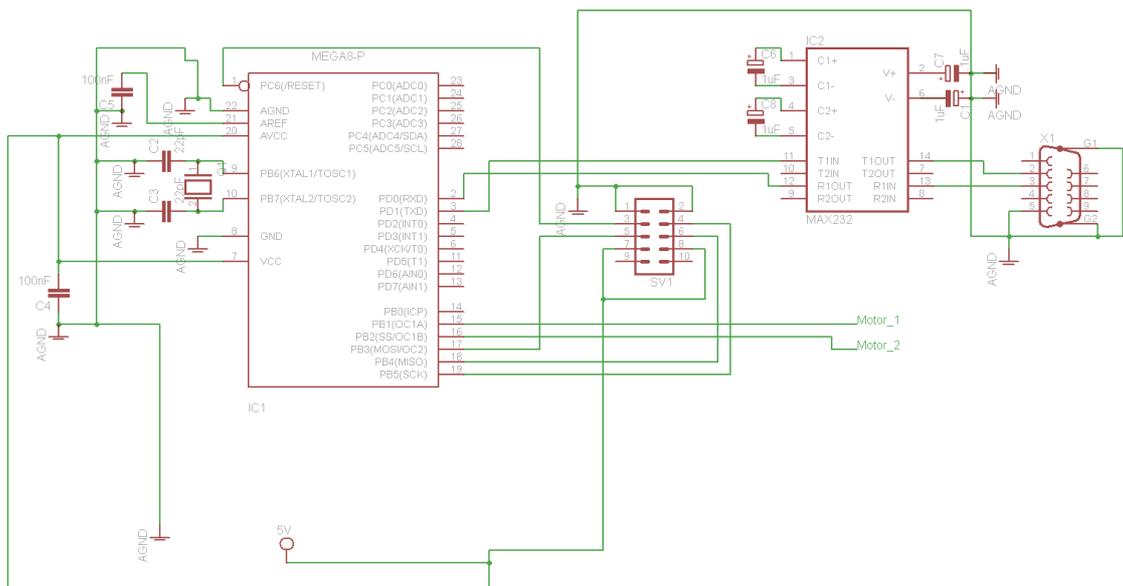


Figura 3.5: Montagem da placa de controle, com as saídas PWM para os servo-motores

do microcontrolador ATMEL(R) ATMEGA8(R). Os motores utilizados já possuem acionamento embutido em suas estruturas. Sendo assim, a placa é responsável apenas por enviar o sinal PWM para os motores.

Os motores possuem a seguinte característica: o PWM a 0.0V leva a posição de 0 graus. O PWM a 5.0V leva os motores a 180 graus. Contudo, as estratégias de controle aplicadas são para a velocidade do motor. Para resolver esta limitação, a controle de velocidade foi feito via firmware. Basicamente o sistema recebe uma referência de velocidade. Esta velocidade é multiplicada pelo intervalo de tempo da interrupção configurada, que é de 0,222s. A resultado é a variação de ângulo, $\Delta\theta$, que será adicionada à posição atual do motor. A posição final do motor é transformada em sinal de PWM que é enviada pela porta de saída do sinal. Este procedimento é utilizado no acionamento dos dois motores.

A mensagem é enviada pelo computador para a serial seguindo um pequeno protocolo. O primeiro bit é um sinal que indica para qual motor será enviado o sinal. Os bits seguintes indicam a velocidade do motor. O bit de parada é representado pelo caracter '0'.

Os códigos do firmware estão presentes no Anexo II, comentados.

3.5 Estratégias de controle

No desenvolvimento deste trabalho, duas estratégias de controle foram utilizadas. A primeira foi o controle PI digital com *anti-windup*. A outra foi o controle servo-visual adaptativo. Para cada uma destas estratégias, foram coletados dados que serão mostrados e analisados no Capítulo 4.

O controle neste projeto será responsável por centrar a face principal do vetor de faces rastreadas na imagem da *webcam*. Os sistemas a serem controlados pelos dois motores podem ser representados pelos diagramas de bloco da Figura 3.6.

Estes diagramas simplificados mostram o controle inserido na dinâmica do sistema. Também é possível ver como o filtro de Kalman trabalha no canal de realimentação do sistema. O ruído r mostrado é utilizado para indicar qualquer interferência externa no sistema.

É importante relembrar que o sistema é amostrado, sendo que os controladores também o serão.

3.5.1 Controle PI digital *anti-windup*

O controle PI, no domínio da frequência, pode ser descrito de acordo com a formulação a seguir [11].

$$C_{PI}(s) = \frac{W(s)}{E(s)} = K\left(1 + \frac{1}{K_i s}\right) \quad (3.10)$$

A forma digitalizada deste controlador pode ser descrita pelas seguintes equações [12]

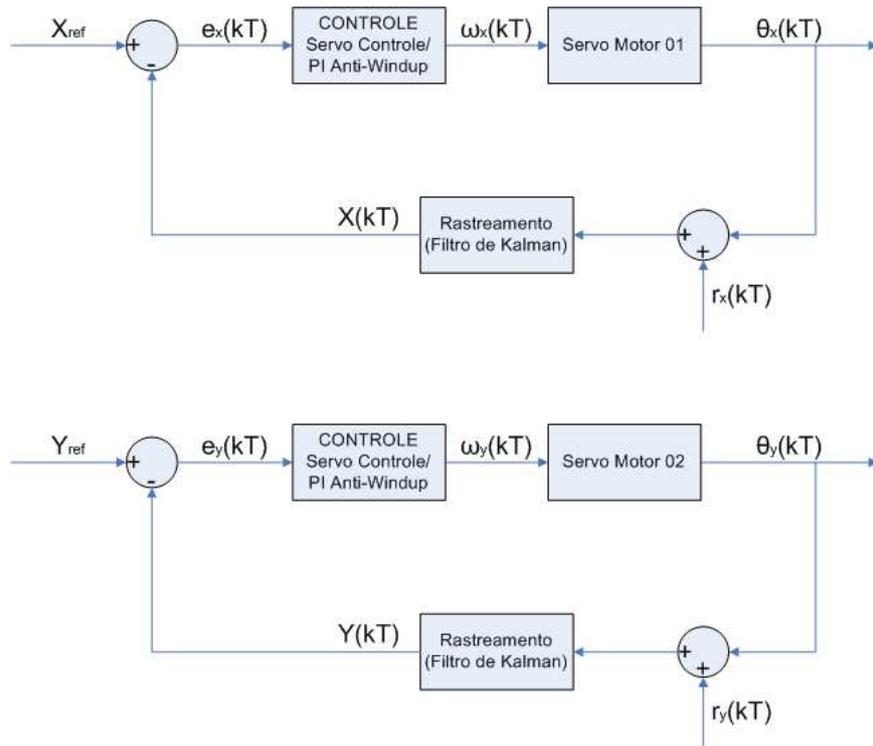


Figura 3.6: Estratégia geral de controle do sistema com câmeras com montagem *pan-tilt*

$$e_i(k) = e_i(k-1) + T * e(k) \quad (3.11)$$

$$w(k) = K_p * e(k) + K_i * e_i(k) \quad (3.12)$$

A entrada do controlador $e(k)$ representa a diferença entre a posição da imagem (X, Y) e a referência de posição (centro da imagem). A saída $w(k)$ representa informação de velocidade angular enviada para o motor.

O controle PI *anti-windup* é utilizado para se evitar que o erro em relação à referência seja utilizado no canal integral do controlador, quando saturar o sistema. O *anti-windup* é implementado da seguinte maneira.

1. Se $U(k)$ for maior que MAX_SPEED ou menor que MIN_SPEED , seleciona-se *flag* para 0.
2. Na próxima passagem, se *flag* igual a 0, então $e_i(k) = e_i(k-1)$.

Os valores dos ganhos do sistema foram obtidos de maneira experimental, pois para esta estratégia de controle, o sistema não foi modelado. Na Tabela 3.2, encontram-se esses valores de ganho.

A saída para o sistema é dada em radianos por segundo.

Tabela 3.2: Tabela de valores de ganhos do controle PI *anti-windup*

Ganho	Valor
K_{iX}	0,0000001
K_{pX}	0,0005
K_{iY}	0,0000001
K_{pY}	0,0005

3.5.2 Controle servo-visual adaptativo

Esta aplicação do controle é baseada na teoria apresentada no Seção 2.3.1 do Capítulo 2. No projeto, será utilizado o controle servo-visual clássico [6]. Esta estratégia prevê que um ponto 3-D com coordenadas $P = (X, Y, Z)$, projeta na imagem um ponto $p = (x, y)$, tal que

$$\begin{cases} x = X/Z = (u - c_u)/f\alpha \\ y = Y/Z = (v - c_v)/f \end{cases} \quad (3.13)$$

Em que $m = (u, v)$ representa as coordenadas do ponto na imagem em pixels e $a = (c_u, c_v, f, \alpha)$ são parâmetros intrínsecos: c_u e c_v são a referência, f a distância focal e α a razão das dimensões dos pixels. Os valores de f e α foram obtidos experimentalmente via calibração da câmera e possuem os valores de 321,1 e 1, respectivamente.

Para a aplicação da equação 2.64, será necessário antes definir a matriz de iteração L_e . Neste caso, deve-se modelar o sistema, a partir da equação $\dot{s} = L_s V_c$. A figura 3.7 mostra como está relacionada a movimentação de objetos na imagem em relação à variação do ângulo da câmera. Desta Figura, pode-se extrair as seguintes relações

$$\theta_{max} = \arctg\left(\frac{X}{Z}\right) \quad (3.14)$$

$$\theta_{min} = \arctg\left(\frac{X'}{Z}\right) \quad (3.15)$$

Como se deseja um erro nulo de referência, o eixo Z deve estar exatamente sobre o eixo pontilhado. Esta restrição leva a $X' = 0$ e, conseqüentemente a $\theta_{min} = 0$. Derivando-se a equação 3.14 em relação ao tempo, tem-se

$$\dot{\theta}_x = \dot{\theta}_{max} - \dot{\theta}_{min} = \dot{\theta}_{max} = -\frac{\dot{X}}{1 + \frac{X}{Z}} \quad (3.16)$$

É importante indicar que a movimentação do objeto na imagem em profundidade (eixo Z) não afeta a variação do ângulo. Utilizando-se a equação 3.13 e $\alpha = 1$, podemos derivar a seguinte relação

$$\dot{u} = -\frac{f}{Z}\left(1 + \frac{X^2}{Z^2}\right)\dot{\theta}_x \quad (3.17)$$

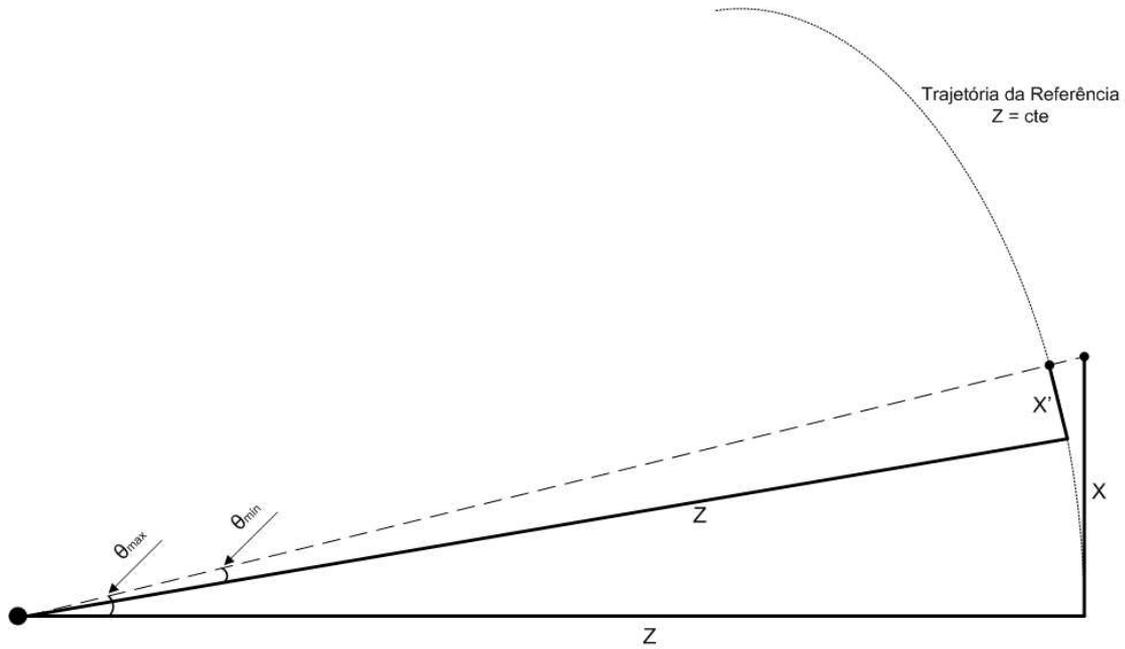


Figura 3.7: Modelagem do sistema de servo-controle segundo os eixos X e Z

Para o eixo Y, teremos exatamente a mesma formulação, em que

$$\dot{v} = -\frac{f}{Z}\left(1 + \frac{Y^2}{Z^2}\right)\dot{\theta}_y \quad (3.18)$$

As equações 3.17 e 3.18 correspondem a seguinte relação $\dot{m} = L_m \dot{\theta}_c$, em que

$$L_m = \begin{pmatrix} -\frac{f}{Z}\left(1 + \frac{X^2}{Z^2}\right) & 0 \\ 0 & -\frac{f}{Z}\left(1 + \frac{Y^2}{Z^2}\right) \end{pmatrix} \quad (3.19)$$

O sistema consegue prover diretamente todos os valores para alimentar a matriz L_m , exceto a distância Z do objeto da imagem. Para estimar o valor de Z, utilizou-se a informação do raio da face, obtida pelo filtro de Kalman. Para tanto, foram feitas medições de raio em pixels e distâncias da câmera. A Tabela 3.3 relaciona valores médios de raios e distâncias da face da câmera.

A gráfico mostrado na Figura 3.8 indica a relação entre as duas medidas e é feita uma regressão exponencial que levou à relação

$$\hat{Z} = 71,941r^{-1,274} \quad (3.20)$$

Teste preliminares desta relação foram realizados e foi obtido um bom resultado. Apesar de variação do raio, devido ao rastreamento pelo filtro de Kalman, a média da distância se manteve próximo ao valor real, que foi medido com fita métrica. Desta forma, a equação 2.64 fica da seguinte maneira.

$$V_c = -\lambda \hat{L}_e^{-1} e \quad (3.21)$$

Tabela 3.3: Tabela de relação entre distância e o raio da face

Distância da Câmera (m)	Raio (pixels)
0,5	51,78
0,6	42,23
0,7	36,65
0,8	33,55
1,0	29,19
1,1	25,47
1,2	24,25
1,5	22,62

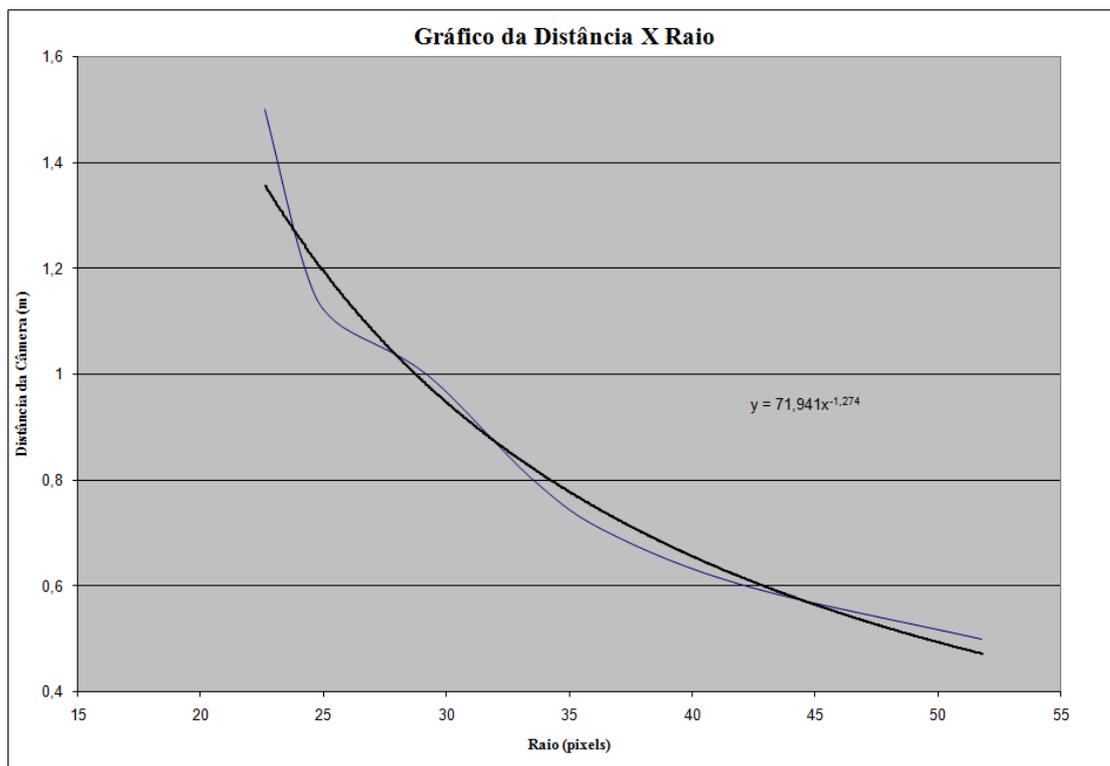


Figura 3.8: Gráfico da relação entre distância do face na imagem e o raio da face

Capítulo 4

Resultados Experimentais

Apresentação das avaliações de resultados de rastreamento e controle do projeto, identificando quais as melhores estratégias.

4.1 Introdução

O propósito da avaliação de resultados é testar o funcionamento geral dos itens descritos neste documento. Entre este itens, estão dois principais: o rastreamento promovido pela filtragem estocástica e a dinâmica do erro controlado pelo módulo de controle. Para cada item, foram coletados dados para valores de parâmetros dos sistemas. Para o caso do filtro de Kalman, foram testados dois conjuntos de valores para os parâmetros \mathbf{r} e \mathbf{q} . Para o controle, buscou-se resultados para as duas estratégias de controle propostas: controle PI anti-windup e controle servo-visual adaptativo.

Como citado anteriormente, toda a implementação destes sistemas foi feita em programa na linguagem C, para computador pessoal e não se encontra anexada neste relatório. O firmware e o hardware estão apenas responsáveis por acionar os motores da estrutura. A máquina utilizada foi um computador com processador Intel Core 2 Duo 2.2 GHz, com memória RAM de 1.99 GHz e sistema operacional MS Windows XP SP2.

4.2 Avaliação de parâmetros da filtragem estocástica

Nesta avaliação, foram colhidos dados do filtro de Kalman, que neste caso são os valores de predição, medição e correção das coordenadas X , Y e o raio R que representam a face principal. O procedimento para captação dos valores é descrito a seguir. Criaram-se rotinas para receber e gravar as informações citadas acima em um arquivo de texto. Além disso, também foi gravado o tempo acumulado de execução do experimento. O formato dos dados foi tal que pudessem ser exportados para tabelas do Microsoft(R) Excel(R) e posteriormente serem feitos gráficos. Em relação à referência, posicionei minha face em frente a *webcam* e movimentei minha face de forma aleatória. O controle de posição não foi utilizado neste experimento, ficando a câmera parada. A duração de cada um não foi rigidamente definida, mas foi deixado tempo suficiente para que a dinâmica geral do sistema pudesse ser analisada.

Tabela 4.1: Valores utilizados para os parâmetros r e q do filtro de Kalman

Medição	R	Q
1	0,05	0,1
2	0,05	0,005

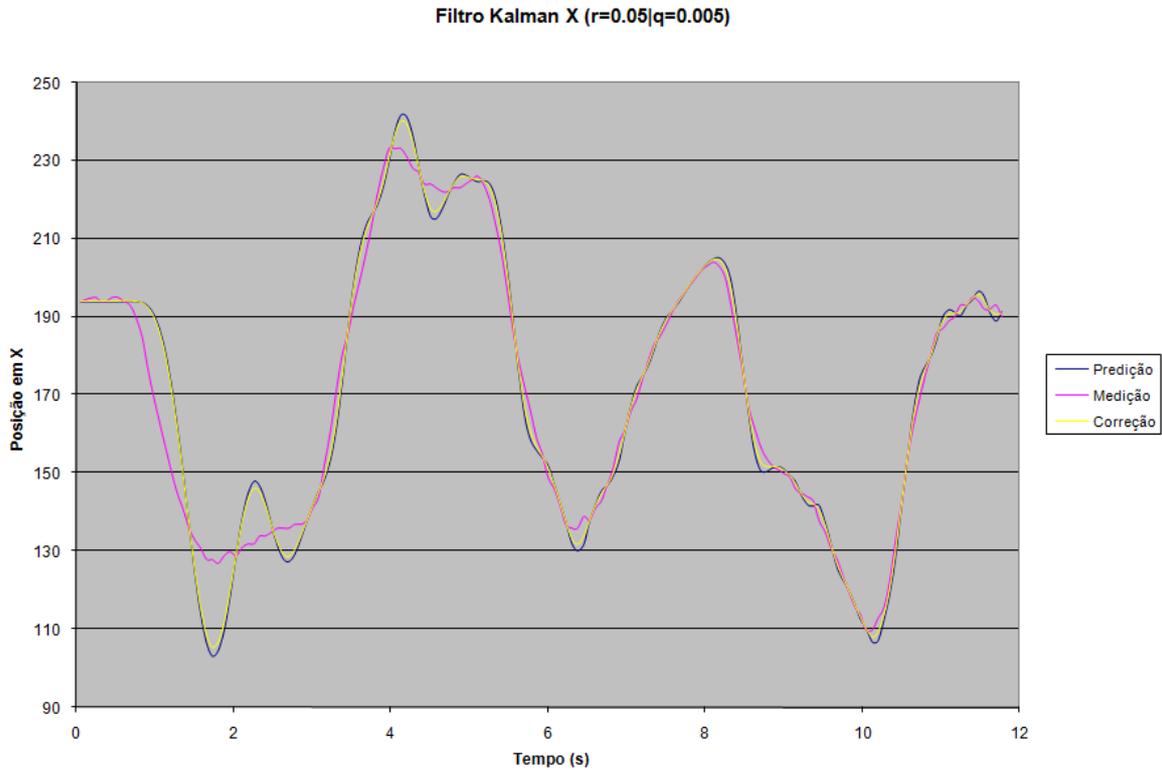


Figura 4.1: Gráfico da evolução da posição X para $r=0.05$ e $q=0.005$

A Tabela 4.1 mostra os valores utilizados para os parâmetros r e q do filtro de Kalman. As Figuras 4.1, 4.2, 4.3, 4.4, 4.5 e 4.6 mostram os resultados obtidos para cada escolha de parâmetros.

As Figuras 4.1, 4.2 e 4.3 mostram uma resposta do filtro quando os parâmetros r e q são 0,05 e 0,005, respectivamente. A escolha desses valores, feita de modo empírico, indica ao filtro de Kalman que o processo de medição trás grande incerteza em seu valor. Por isso, a correção do sistema tende a acompanhar mais o sinal de predição do que o medição. Isto pode ser visto, por exemplo, em momentos em que os valores de X , Y e R recebem um degrau.

As Figuras 4.4, 4.5 e 4.6 mostram uma resposta do filtro quando os parâmetros r e q são 0,05 e 0,1, respectivamente. A escolha desses valores, também de forma empírica, indica ao filtro de Kalman que a medição é mais confiável. Observa-se destas figuras, que o sinal de correção está mais próximo da medição, sofrendo menor influência da predição.

Os dois resultados apresentam duas abordagem diferentes: a medição mais ou menos confiável. Contudo, observações da medição indicam que seu valor é confiável. Além disso, o resultado obtido pelo primeiro experimento demonstra também uma característica ruim desta escolha de

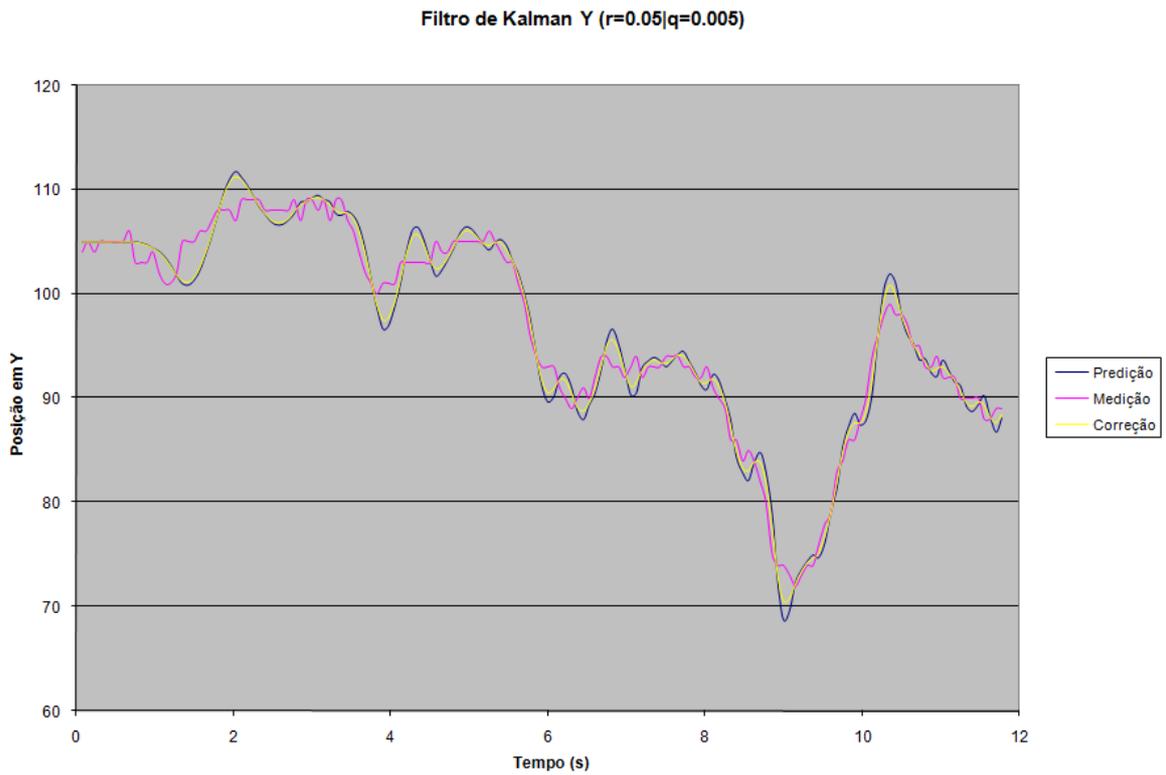


Figura 4.2: Gráfico da evolução da posição Y para $r=0.05$ e $q=0.005$

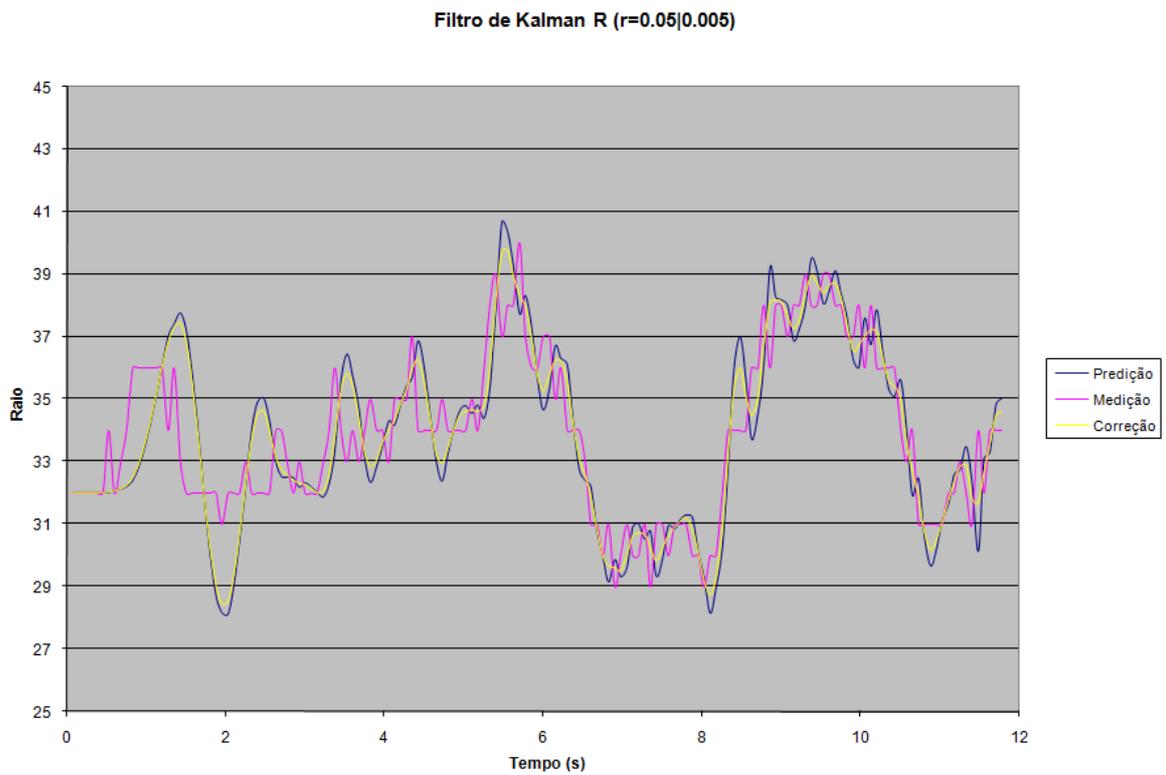


Figura 4.3: Gráfico da evolução do raio R para $r=0.05$ e $q=0.005$

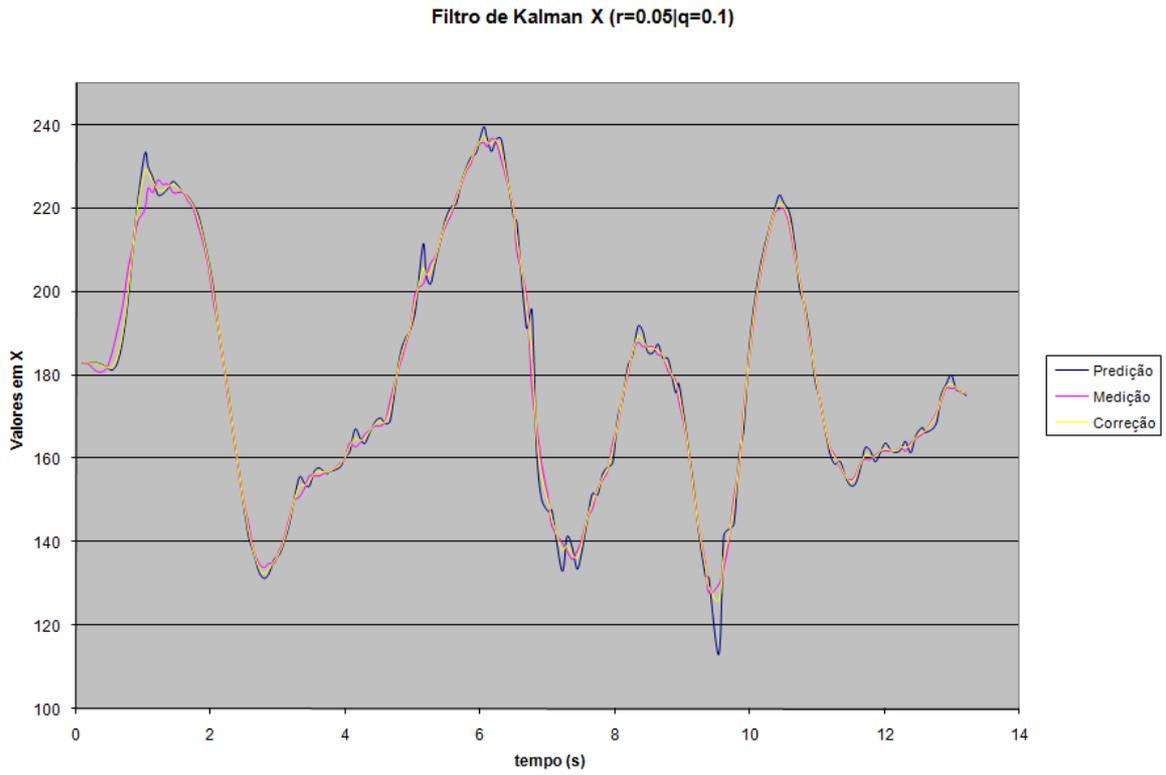


Figura 4.4: Gráfico da evolução da posição X para $r=0.05$ e $q=0.1$

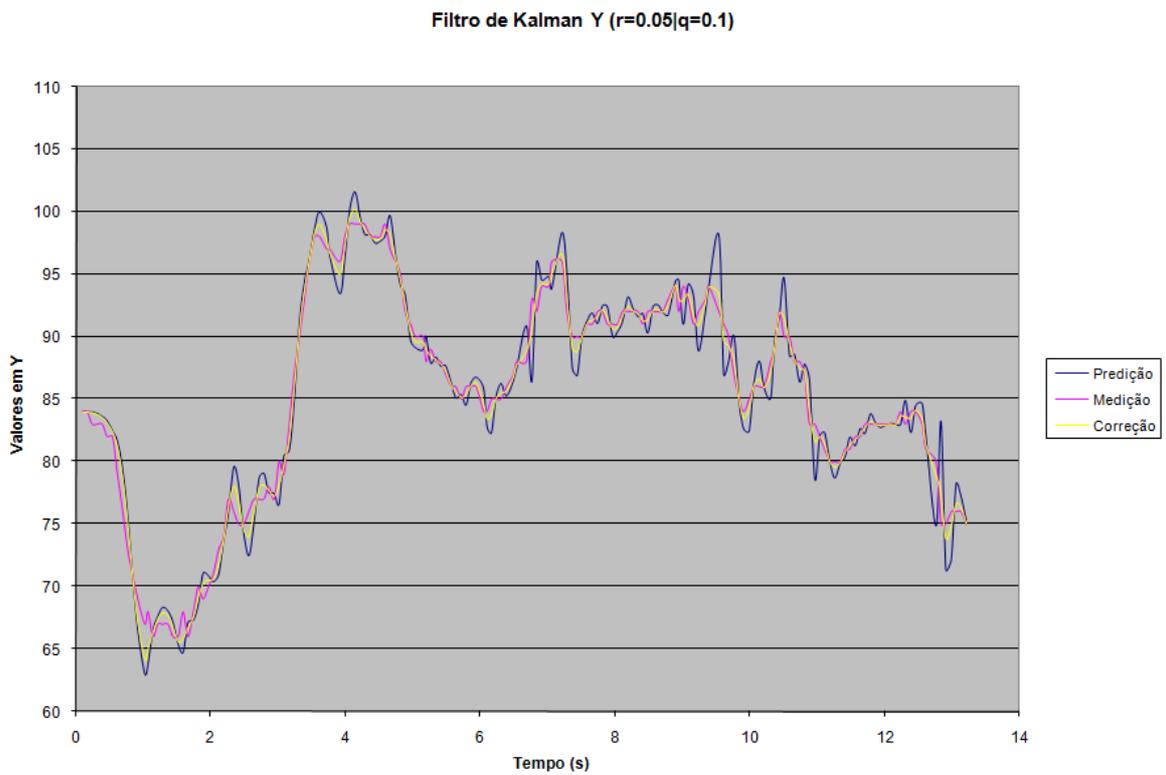


Figura 4.5: Gráfico da evolução da posição Y para $r=0.05$ e $q=0.1$

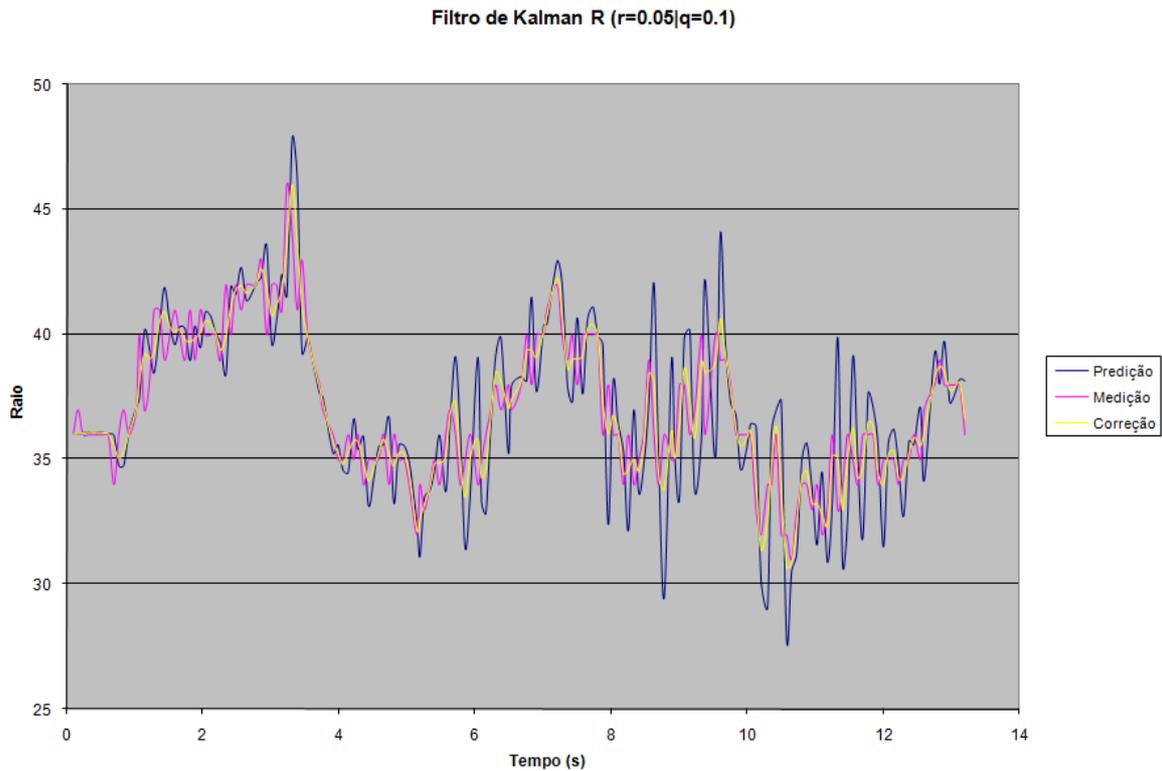


Figura 4.6: Gráfico da evolução do raio R para $r=0.05$ e $q=0.1$

valores para r e q : o centro da face fica circulando a referência central até atingí-la. Isto pode ser visto nos gráficos ao se observarem os sinais de medição (lilás) e o de correção (amarelo), em que este apresenta grande sobre-passo em relação ao primeiro. Este efeito, no eixo X e Y , gera o efeito descrito anteriormente, de um raio circulando a face na imagem. Para o controle, este tipo de resultado não é interessante. Estes sobre-passos fazem com que o sistema acione os motores a todo instante para garantir erro nulo de referência (tema de controle será abordado da próxima seção), fazendo com que a estrutura balance bastante até atingí-la. Apesar de o sistema não perder a estabilidade, a utilização de $r = 0,05$ e $q = 0,1$ gera um sinal melhor para o controle do que quando se usa o outro conjunto de valores, sendo os primeiro valores utilizados para as medições e análise do módulo de controle.

4.3 Avaliação das estratégias de controle

Esta avaliação mostra os resultado obtidos das duas estratégias de controle apresentadas (controle PI anti-windup - CPIA - e controle servo-visual adaptativo - CSVA) para os valores dos parâmetros do filtro de Kalman sendo $r = 0,05$ e $q = 0,1$. O procedimento para coleta de dados foi semelhante ao utilizado na seção 4.2. Mas o conteúdo foi diferente. Neste caso, foi gravado no arquivo texto valores de erro em relação à referência. Novamente, posicionei minha face em frente à *webcam* e me movimente para que o controle pudesse agir. Os gráficos criados também foram gerados pelo Microsoft(R) Excel(R).

Dinâmica do controle PI anti-windup no eixo X
($r=0.05|q=0.1|K_p=0.0005|K_i=0.0000001$)

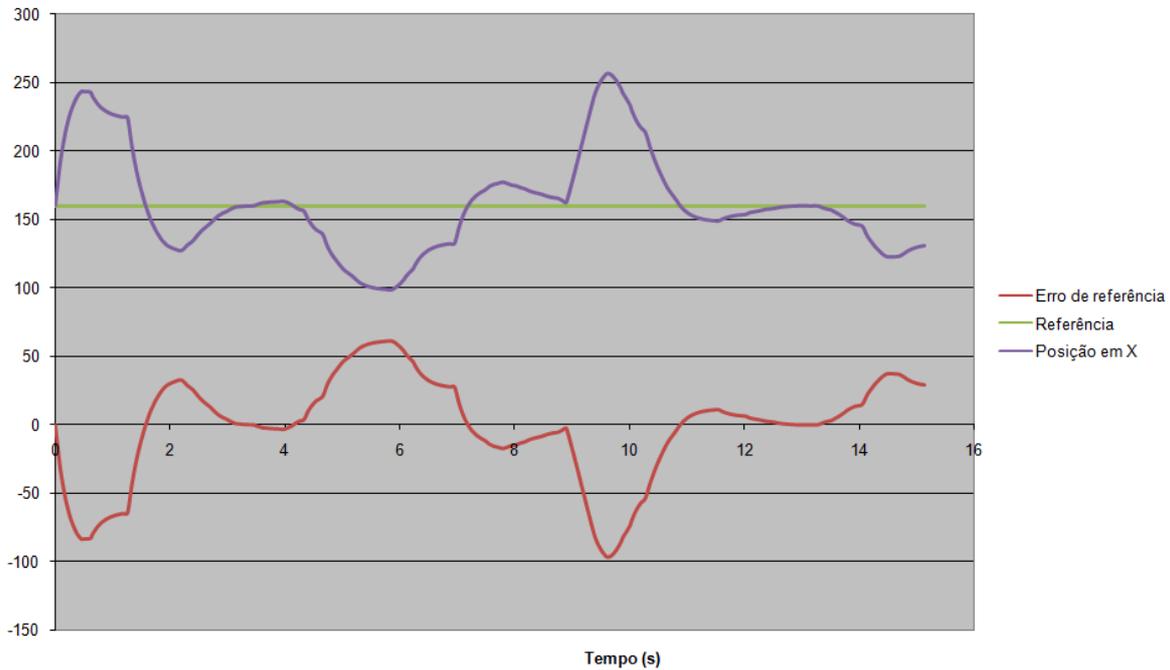


Figura 4.7: Gráfico da evolução erro em X para controle CPIA

Nesta avaliação, vale ressaltar que uma característica que vem da medição (e, conseqüentemente, do rastreamento) do sistema e que dificulta a aplicação de testes de controle é a movimentação do centro da face (captado pela aplicação) independente do movimento real. Este característica se deve principalmente as mudanças de luminosidade captadas pela imagem, o que afeta na determinação do centro e raio da face.

As Figuras 4.7, 4.8, 4.9 e 4.10 indicam os resultados obtidos para avaliação das estratégias de controle.

As Figuras 4.7 e 4.8 mostram a dinâmica do erro do sistema (em X e Y) para controle PI. A amplitude do sinal em X é maior do em Y . Isto porque movimento entre direita e esquerda (eixo X) podem ser feitos com mais facilidade do que movimento de cima para baixo (eixo Y). Observando-se os resultados obtidos, vê-se que o sistema tende ao erro nulo, mas apresenta sobre-passo. Isto pode ser visto entre 0 e 4 segundos no gráfico do eixo X e entre 5 e 9 segundos no gráfico do eixo X . Este sobre-passo não é interessante, devido à face detectada estar sempre em movimento. Para se evitar o sobre-passo, uma estratégia é deixar o sistema mais lento, ajustando-se os ganhos K_{pX} e K_{pY} . A penalidade desta estratégia é que o sistema vai atingir a referência em um tempo maior.

As Figuras 4.9 e 4.10 mostram a dinâmica do erro do sistema (em X e Y) para CSVA. A principal característica observada é a queda exponencial do erro. Esta avaliação foi feita com o rosto a um distância de 1,0 metro. Isto para se ter uma idéia da queda do erro exponencial. Contudo, independente da distância do sistema, a queda exponencial do erro deve ser igual, com constante de tempo λ , como proposto no capítulo 2, seção 2.3. Outra característica importante é

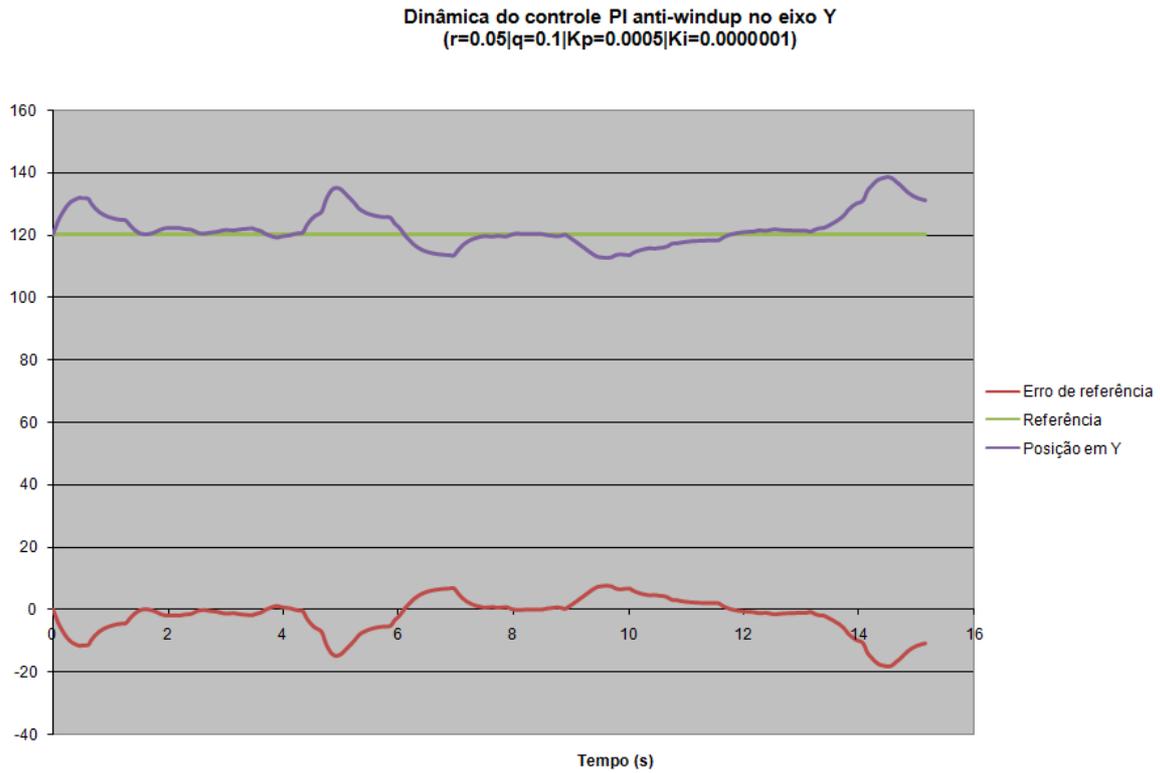


Figura 4.8: Gráfico da evolução erro em Y para controle CPIA

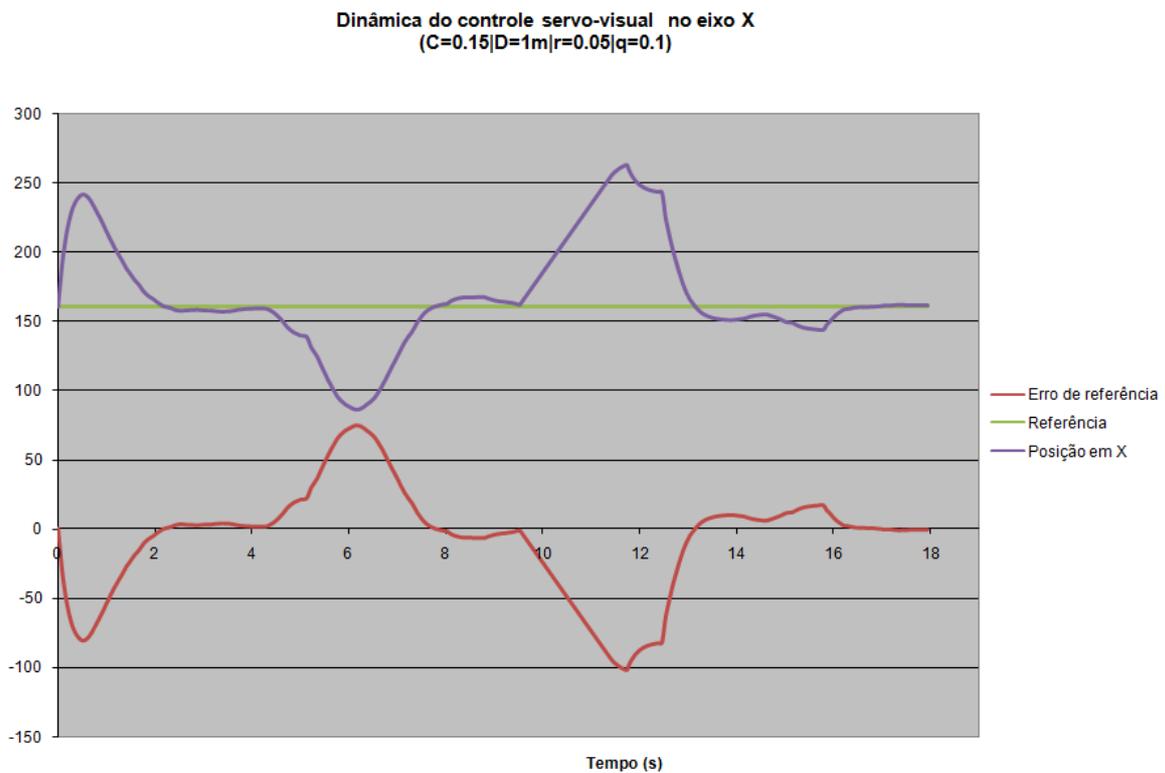


Figura 4.9: Gráfico da evolução erro em X para CSVA

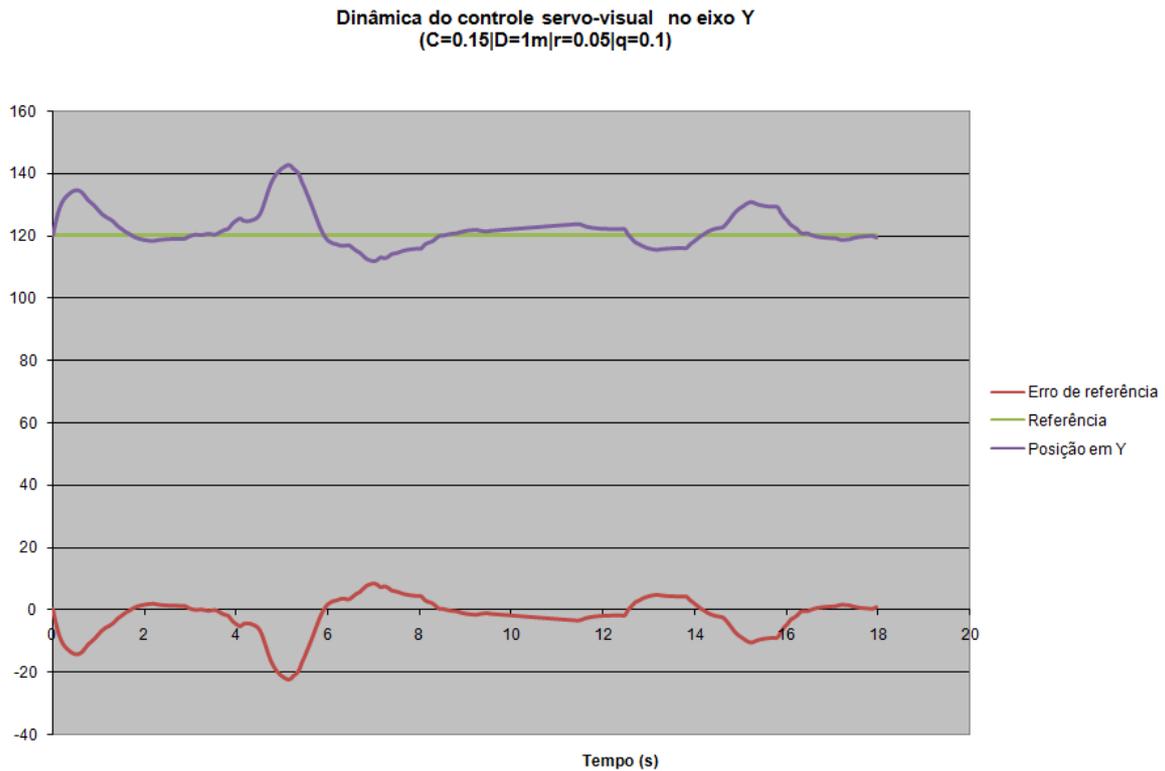


Figura 4.10: Gráfico da evolução erro em Y para CSVA

a não existência de sobre-passo. A queda do erro é bastante suave, evitando-se que a estrutura do sistema (parte mecânica) vibre, podendo-se perder a referência da face rastreada.

Observando os resultados obtidos, pode-se aferir que a estratégia de controle servo-visual adaptativo apresenta um melhor resultado para o controle de velocidade do sistema.

Em termos de processamento, o controle servo-visual adaptativo e o controle PI anti-windup apresentam baixo consumo computacional, pois envolve cálculos relativamente simples.

Assim, o CSVA é a estratégia de controle foi a técnica de controle utilizada para o problema deste relatório.

Capítulo 5

Conclusões

O rastreamento de faces em seqüências de imagens é um campo relativamente novo e sua aplicação foi possível através da melhora em duas vertentes: a primeira no campo do processamento (advento dos computadores digitais de maior capacidade de processamento). No outro, melhoria dos algoritmos de rastreamento, em que, nesse ponto, se insere o filtro de Kalman.

O objetivo final deste trabalho foi explorar as vantagens da filtragem estocástica (filtro de Kalman) no rastreamento de faces. No caso deste trabalho, a face foi descrita por sua posição X e Y na imagem e pelo raio. Cada face rastreada é independente uma da outra sendo que o filtragem estocástica era aplicada separadamente para cada um. Nesta vertente, construiu-se uma estrutura que seria contralada com informações vindas do rastreamento (câmera com montagem pan-tilt).

Os resultados obtidos na seção 4 indicam que a estratégia de controle que melhor apresentou resultados foi o CSVA e que o rastreamento de faces também mostrou um excelente resultado. Isto gerou um sistema que consegue acompanhar uma referência de face, apesar das restrições da parte mecânica.

O ponto mais importante deste trabalho é a possibilidade de se utilizar os códigos desenvolvidos para diversas outras áreas. Por exemplo, o rastreamento de faces pode ser utilizado para determinar a posição de uma dada face e desta informação extrair informações de características da mesma. E o objeto rastreado pode ser qualquer outro, desde que se possa modelar o objeto por um círculo. O sistema completo (incluse montagem física) pode ser utilizado para visão de um rôbo, por sua montagem com dois graus de liberdade. Para tanto, deve-se definir que tipos de informações devem ser rastreadas e melhorar a parte mecânica (para se evitar que a estrutura montada oscile com seu próprio funcionamento).

5.1 Trabalhos futuros

A filtragem estocástica é um ferramenta que pode ser utilizada em diversas vertentes. O presente trabalho demonstrou como a referida ferramenta pode ser explorada para rastreamento de faces. Nesta vertente, as propostas de trabalhos futuras são várias. A principal seria a utilização do conhecimento atual para o desenvolvimento de sistemas que façam, além do rastreamento, a

extração de características da face presente na imagem, utilizando as três câmeras montadas. Esta nova proposta exigirá o estudo de novos tópicos na área de visão computacional e processamento de imagens e será também uma nova parceria entre a SAGEM-ORGA e o grupo GPDS/CDT.

Outra possível aplicação é utilizar a totalidade do projeto e associá-la a algum tipo de rôbo. Devido à robustez do sistema, este sistema seria independente do movimento do corpo do rôbo, mantendo-se a estabilidade do sistema.

Uma outra aplicação, bastante viável, seria a utilização do sistema em segurança. Como o sistema responde a faces na tela, centrando-as, um sistema de segurança poderia detectar um intruso e focar as câmeras na pessoa.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] BAR-SHALOM, Y.; LI, X.-R. *Estimation and Tracking: Principles, Techniques and Software*. [S.l.]: Norwood: Artech House, 1987.
- [2] ANDERSON, B. D. O.; MOORE, J. B. *Optimal Filtering*. [S.l.]: Dover Books on Engineering, 2005.
- [3] MAGALHÃES, M. N.; LIMA, A. C. P. de. *Noções de Probabilidade e Estatística*. [S.l.]: EdUSP, 2004.
- [4] TUZEL, O.; PORIKLI, F.; MEER, P. Region covariance: A fast descriptor for detection and classification. *ECCV*, v. 02, p. 589–600, 2006.
- [5] PRESS, W. H. et al. *Numerical Recipes in C: The Art of Scientific Computing*. Second. [S.l.]: Cambridge University Press, 1992.
- [6] CHAUMATTE, F.; HUTCHINSON, S. Visual servo control part i: Basic approaches. *IEEE Robotics and Automation Magazine*, v. 06, p. 82–90, 2006.
- [7] IPL. *Introdução ao Visual Studio 2005: Linguagem C*. [S.l.]: IPL, 2005.
- [8] INTEL. *Open Source Computer Vision Library: Reference Manual*. [S.l.]: Intel, 2001.
- [9] BORGES, G. A. *GMATRIX: Uma biblioteca matricial para C/C++*. [S.l.]: UnB, 2005.
- [10] ATMEL. *8-bit AVR(R) with 8K Bytes In-System Programmable Flash: ATmega8*. [S.l.]: Atmel, 2003.
- [11] OGATA, K. *Engenharia de Controle Moderno*. [S.l.]: PHB, 1993.
- [12] FRANKLIN, G. F.; POWELL, J. D. *Digital Control of Dynamic Systems*. [S.l.]: Addison Wesley Longman, 1998.

ANEXOS

I. DESCRIÇÃO DO CONTEÚDO DO DVD

I.1 Pastas do DVD

O DVD que acompanha este trabalho contém material complementar ao seu conteúdo. Seu objetivo principal é servir de auxílio a futuros projetos que venham a ser desenvolvidos tanto como continuação deste projeto, como para aplicação em outros. Procurou-se nele adicionar informações importantes sobre o desenvolvimento e aplicação do projeto.

- Relatório - onde encontra-se o relatório final de curso em PDF;
- Codificação - onde encontra-se o código usado nas simulações descritas, em linguagem C, bem como executáveis e DLL's necessárias;
- Apresentação - as transparências e vídeos apresentados durante a defesa;
- Documentação - onde encontra-se a documentação dos códigos, em páginas HTML, feitas com o software doxygen;
- Material - contém artigos e livros em formato digital, utilizados para desenvolvimento do trabalho;
- Imagens - contém imagens do sistema, utilizadas na confecção do presente trabalho.

A pasta Codificação possui os executáveis utilizados para se extrair os dados utilizados no capítulo 4.

II. DESCRIÇÃO DO FIRMWARE

Este anexo apresenta características do firmware presente no microcontrolador ATMEL(R) ATMEGA8(R) utilizado. Somente serão apresentados este códigos, pois aqueles utilizado para desenvolvimento do software são extensos. Os códigos apresentados estão comentados e algumas explicações são dadas também em forma de comentário. O desenvolvimento deste firmware foi feito pelo aluno de graduação Bruno Amuí, que também participa do projeto, como bolsista de iniciação científica.

II.1 Firmware desenvolvido para o ATMEL(R) ATMEGA8(R)

```

//-----

/*Inclúdes necessários para funcionameto de funções do atmega8*/
#include <stdio.h>
#include <stdlib.h>
#include <avr/delay.h>
#include <avr/io.h>
#include <string.h>
#include <inttypes.h>
#include <avr/sleep.h>
#include <avr/interrupt.h>
#include <avr/signal.h>

#define TRUE 1
#define FALSE 0

/*declaração de variáveis*/
int anguloA = 2700, anguloB= 2700 ;
float VelA = 1.0, VelB = 1.0, vA=0.0,vB=0.0,OCR1Afloat=2700.0,OCR1Bfloat=2700.0;

SIGNAL (SIG_OUTPUT_COMPARE2)
{
    /*aplicando a velocidade obtida nos motores via sinal PWM*/
    if( (OCR1Afloat + vA) > 700.0 && (OCR1Afloat + vA) < 4700.0 ) OCR1Afloat+=vA;
    if( (OCR1Bfloat + vB) > 700.0 && (OCR1Bfloat + vB) < 4700.0 ) OCR1Bfloat+=vB;
    OCR1A=(int)OCR1Afloat;
    OCR1B=(int)OCR1Bfloat;
}

void ioinit (void)
{
    /* TIMER2 é de 8 bits */
    // Considera-se relógio de 16MHz.
    /* Interrupções */
    TIFR |= _BV(OCF2);
    TCCR2 |= _BV(WGM21)|_BV(CS22)|_BV(CS20);
    TCNT2 = 0;
    TIMSK |= _BV(OCIE2);
    OCR2 = 156;
    sei ();
}

int USART_Receive( void )
{
    /* espera dados a serem recebidos */
    while ( !(UCSRA & (1<<RXC)) );

    /* recebe e retorna dados do buffer*/
    return UDR;
}

int USART_Transmit( char data )
{
    /*Espera buffer ficar vazio para transmitir*/
    while ( !( UCSRA & (1<<UDRE)) );

    /* Coloca dados no Buffer para serem enviados */
    UDR = data;
    return 0;
}

void USART_Init(void)
{
    /*zerando o conteúdo do registrador de dados da serial*/
}

```

Figura II.1: Código do firmware - parte1

```

UDR=0;
//Seta baud rate de 9600 para fosc= 16MHz
UBRRH = 0;
UBRRL = 103;
//Habilita transmissão e recepção serial
UCSRB = _BV(TXEN)| _BV(RXEN);
//Setar tamanho palavra
//URSEL = 1 -> seleciona acesso para UCSRC (0 is UBRRH) reading and writing
//UMSEL = 0 -> Asynchronous Operation
//UPM1 e UPM0 = 00 -> sem paridade
//USBS = 1 -> seleciona 2 bits de parada a ser inserido pelo transmissor
//UCSZ 2 1 0 = 011 -> escolhe tamanho de caractere de 8 bits p/ dado
UCSRC = _BV(URSEL) | _BV(UCSZ1) | _BV(UCSZ0);
/*registrando as funções de leitura e escrita de um byte pela serial*/
fdevopen(USART_Transmit, USART_Receive);
}

void delay(void)
{
    unsigned int n;
    // Espera 500 ms
    for(n=1000;n>0;--n)
        _delay_ms(16.0); // de acordo com o manual, o máximo tempo é 262.14 ms / (F_CPU em
    MHz).
}

void PWM_Iniciar (void){
    /*inicialização do PWM*/
    DDRB=0xFF;
    ICR1=40000;
    OCR1A=2700;
    OCR1B=2700;
    TCCR1A=0xA2;
    TCCR1B=0x1A;
    return 0;
}

int main(void) {

    int i;
    void USART_Init(void);
    int USART_Transmit( char data );
    int USART_Receive( void );
    char ch;
    char *y,*r;
    float vParam = 0.0 , pParam = 0.0;

    USART_Init();
    delay();
    printf ("Liguei");

    PWM_Iniciar();
    printf("Inicio PWM \n");
    ioinit();
    printf("Inicio TIMER \n");
    for(;;)
    {
        /*recebe o bit que indica qual motor será utilizado*/
        scanf ("%c", &ch);

        //servoA -> velocidade em X
        if (ch=='@')
        {
            char recebe = '\0';
            char mensagem[10];
            controleVelocidade = 1;
            i=0;
            while (recebe!='|')
            {

```

Figura II.2: Código do firmware - parte2

```

        scanf("%c",&recebe);
        mensagem[i]=recebe;
        i++;
        mensagem[i] = '\0';
    }
    recebe='a';
    i--;
    mensagem[i]='\0';
    VelA = (float) strtod(mensagem, NULL);
    cli();
    vA = (VelA*0.2222222);
    sei();
}
//servoB -> velocidade em Y
if (ch=='$')
{
    char recebe = '\0';
    char mensagem[10];
    controleVelocidade = 1;
    i=0;
    while (recebe!='|')
    {
        scanf("%c",&recebe);
        mensagem[i]=recebe;
        i++;
        mensagem[i] = '\0';
    }
    recebe='a';
    i--;
    mensagem[i]='\0';
    VelB = (float) strtod(mensagem, NULL);
    cli();
    vB = (VelB*0.2222222);
    sei();
}
}
}
//-----

```

Figura II.3: Código do firmware - parte3