

TRABALHO DE GRADUAÇÃO

**SISTEMA PARA CALIBRAÇÃO DE ROBÔ PARALELO
BASEADO EM COMPENSAÇÃO ESTATÍSTICA DE ERROS
COM AVALIAÇÃO EXPERIMENTAL**

Luiz Vitor da Silva Vieira Reis

Brasília, Dezembro de 2018



**ENGENHARIA
MECATRÔNICA**
UNIVERSIDADE DE BRASÍLIA

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia
Curso de Graduação em Engenharia de Controle e Automação

TRABALHO DE GRADUAÇÃO

SISTEMA PARA CALIBRAÇÃO DE ROBÔ PARALELO
BASEADO EM COMPENSAÇÃO ESTATÍSTICA DE ERROS
COM AVALIAÇÃO EXPERIMENTAL

Luiz Vitor da Silva Vieira Reis

*Relatório submetido como requisito parcial de obtenção
do grau de Engenheiro de Controle e Automação*

Banca Examinadora

Prof. Dr. José Maurício S. T. Motta, _____
ENM/UnB
Orientador

Prof. Dr. Guilherme Caribé de Carvalho, _____
ENM/UnB
Examinador interno

Prof. Dr. Calos Humberto Llanos Quintero, _____
ENM/UnB
Examinador interno

Brasília, Dezembro de 2018

FICHA CATALOGRÁFICA

REIS, LUIZ VITOR DA SILVA VIEIRA

Sistema para calibração de robô paralelo baseado em compensação estatística de erros com avaliação experimental

[Distrito Federal] 2018.

xiv, 59p., 297 mm (FT/UnB, Engenheiro, Controle e Automação, 2015). Trabalho de Graduação – Universidade de Brasília. Faculdade de Tecnologia.

1. Calibração de robôs

2. Calibração de câmeras

3. Similaridade de erros

4. Modelagem estatística

I. Mecatrônica/FT/UnB

II. Título (Série)

REFERÊNCIA BIBLIOGRÁFICA

REIS, LUIZ VITOR DA SILVA VIEIRA, (2018). Sistema para calibração de robô paralelo baseado em compensação estatística de erros com avaliação experimental. Trabalho de Graduação em Engenharia de Controle e Automação, Publicação FT.TG-*n*°17, Faculdade de Tecnologia, Universidade de Brasília, Brasília, DF, 59p.

CESSÃO DE DIREITOS

AUTOR: Luiz Vitor da Silva Vieira Reis

TÍTULO DO TRABALHO DE GRADUAÇÃO: Sistema para calibração de robô paralelo baseado em compensação estatística de erros com avaliação experimental.

GRAU: Engenheiro de Controle e Automação

ANO: 2018

É concedida à Universidade de Brasília permissão para reproduzir cópias deste Trabalho de Graduação e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte desse Trabalho de Graduação pode ser reproduzida sem autorização por escrito do autor.

Luiz Vitor da Silva Vieira Reis

Campus Darcy Ribeiro, Faculdade de Tecnologia, Universidade de Brasília.

70910-900 Brasília – DF – Brasil.

Dedicatória

Dedico esse trabalho à minha mãe.

Luiz Vitor da Silva Vieira Reis

Agradecimentos

Agradeço a todos aqueles que contribuíram para o desenvolvimento desse trabalho.

Luiz Vitor da Silva Vieira Reis

RESUMO

Robôs paralelos ou máquinas cinemáticas paralelas são mecanismos de cadeia cinemática fechada, em que o efetuador terminal é ligado a base do robô por duas ou mais cadeias cinemáticas independentes. Os robôs paralelos são considerados altamente precisos devido ao não acúmulo dos erros de junta, diferentemente dos robôs seriais, sendo muito utilizados em tarefas que necessitam de alta precisão no posicionamento do efetuador terminal. Em contrapartida, tais robôs podem apresentar perda de precisão devido a erros de fabricação, folga nos elementos de transmissão de movimento, etc. Portanto, é necessário aplicar um método de calibração que permita corrigir os erros de posicionamento causados por essas imperfeições. O objetivo desse trabalho é desenvolver um sistema de calibração do modelo cinemático de um robô industrial de configuração paralela, para melhora da exatidão do posicionamento, com base em um método de compensação estatística de erros por modelagem estatística de amostras com similaridade de erros. Nesse trabalho foi utilizado um sistema de medição baseado em visão computacional. O sistema de medição consiste em uma câmera montada no robô, de modo que se possa medir o posicionamento do efetuador terminal em relação a um sistema de coordenadas global fixado em uma placa de calibração, o modelo RAC de calibração de câmeras foi implementado para resolver esse problema. O sistema foi avaliado experimentalmente com um robô paralelo ABB IRB 360, e os resultados estão contidos nesse relatório. Os menores erros encontrados após a calibração da câmera foram de 0,564 mm, 0,687 mm e 0,899 mm nos eixos x, y e z do sistema de coordenadas global.

Palavras Chave: Calibração de robôs, robôs paralelos, similaridade de erros, modelagem estatística, calibração de câmera.

ABSTRACT

Parallel robots or parallel kinematic machines are closed kinematic chain mechanisms, wherein the terminal effector is connected to the base of the robot by two or more independent kinematic chains. Parallel robots are considered highly accurate due to non-accumulation of joint errors, unlike serial robots, and are widely used in tasks that require high precision in the end effector positioning. In contrast, such robots may suffer accuracy loss due to manufacturing errors, slack in motion transmission elements, etc. Therefore, it is necessary to apply a calibration method that allows to correct the positioning errors caused by these imperfections. The goal of this work is to develop a kinematic model calibration system of an industrial robot with parallel configuration, to improve the positioning accuracy, with statistical compensation based model by statistical modelling of samples with error similarity. In this work a vision-based measurement system was used. The measurement system consists of a camera mounted on the robot, so that the end effector positioning relative to a global coordinate system can be measured, the RAC camera calibration model has been implemented to solve this problem. The system was evaluated experimentally with a parallel robot ABB IRB 360, and the results are contained in this report. The smallest errors found after camera calibration were 0.564 mm, 0.687 mm and 0.899 mm on the x, y, and z axes of the global coordinate system.

Keywords: Robot calibration, parallel robots, error similarity, statistical modelling, camera calibration.

SUMÁRIO

1	INTRODUÇÃO	1
1.1	CONTEXTUALIZAÇÃO	1
1.2	DEFINIÇÃO DO PROBLEMA	2
1.3	OBJETIVOS DO PROJETO	2
1.4	APRESENTAÇÃO DO MANUSCRITO	2
2	FUNDAMENTOS TEÓRICOS	4
2.1	INTRODUÇÃO	4
2.2	CALIBRAÇÃO DE ROBÔS	4
2.2.1	DESCRIÇÃO DO ROBÔ UTILIZADO	6
2.2.2	MODELAGEM CINEMÁTICA DIRETA	10
2.2.3	MODELAGEM CINEMÁTICA INVERSA	15
2.3	VISÃO COMPUTACIONAL	16
2.3.1	CÂMERAS	17
2.3.2	MODELO DE CÂMERA SEM DISTORÇÃO	18
2.3.3	MODELO DE CÂMERA COM DISTORÇÃO	21
2.3.4	O MODELO RAC DE CALIBRAÇÃO DE CÂMERAS	23
2.3.5	DESCRIÇÃO DA CÂMERA ELENTE UTILIZADAS	24
2.4	ANÁLISE ESTATÍSTICA BASEADA EM SIMILARIDADE DE ERROS	26
2.4.1	ANÁLISE QUALITATIVA DA SIMILARIDADE DE ERROS DE POSICIONAMENTO	26
2.4.2	ANÁLISE QUANTITATIVA DA SIMILARIDADE DE ERROS DE POSICIONAMENTO	27
2.4.3	IDENTIFICAÇÃO E COMPENSAÇÃO DE ERROS UTILIZANDO SIMILARIDADE DE ERROS	28
3	DESENVOLVIMENTO EXPERIMENTAL	32
3.1	INTRODUÇÃO	32
3.2	IMPLEMENTAÇÃO DO MODELO RAC DE CALIBRAÇÃO DE CÂMERA	32
3.2.1	TRANSFORMAÇÃO DO SISTEMA DA CÂMERA PARA O SISTEMA DE COORDENADAS GLOBAIS	36
3.2.2	CALIBRAÇÃO DA RAZÃO μ	37
3.2.3	CALIBRAÇÃO DO CENTRO DA IMAGEM	38
4	RESULTADOS	39

4.1	INTRODUÇÃO	39
4.2	EXPERIMENTO E RESULTADO DA CALIBRAÇÃO DA CÂMERA	39
4.2.1	PARÂMETROS INTRÍNSECOS OBTIDOS	43
4.2.2	PARÂMETROS EXTRÍNSECOS OBTIDOS E AVALIAÇÃO DA PRECISÃO DA CÂMERA	44
4.3	EXPERIMENTO E RESULTADO DO ESTIMADOR ESTATÍSTICO	46
5	CONCLUSÕES	49
	REFERÊNCIAS BIBLIOGRÁFICAS	50
	APÊNDICES	52
I	IMPLEMENTAÇÃO DO MODELO RAC NO MATLAB	53

LISTA DE FIGURAS

2.1	Diagrama de blocos do sistema de calibração de um robô [1] (adaptado).....	5
2.2	Robô ABB IRB 360 <i>flexpicker</i> [2].....	6
2.3	Eixos de movimentação do robô [2].....	7
2.4	Representação geométrica da base do robô.	8
2.5	Representação geométrica da plataforma móvel do robô.	9
2.6	Representação geométrica completa do robô [3].	10
2.7	Representação geométrica completa do robô [3].	12
2.8	Exemplos de aplicações de visão computacional [4]......	17
2.9	Geometria de uma câmera com projeção perspectiva e com distorção radial da lente [5].	19
2.10	Efeitos das distorções radial e tangencial.	21
2.11	Câmera <i>Basler</i> utilizada no projeto.....	25
2.12	Curva típica de um semivariograma [6].	27
3.1	Placa de calibração com pontos coplanares utilizada no trabalho.....	33
3.2	Padrão da placa de calibração.	33
4.1	Configuração hand-mounted.	40
4.2	Seleção de região da imagem.....	42
4.3	Pontos de calibração extraídos da imagem.	42
4.4	FlexPendant usado para movimentar o robô.	44
4.5	Erros de posicionamento no eixo X robô.....	47
4.6	Erros de posicionamento no eixo Y robô.....	47
4.7	Erros de posicionamento no eixo Z robô.	48

LISTA DE TABELAS

2.1	Especificações câmera <i>Basler acA2500-14um</i>	25
2.2	Especificações lente <i>Edmund Optics 4.5mm/F1.95</i>	25
4.1	Exatidão e repetitividade de trajetórias do robô [2].	39
4.2	Calibração do centro da imagem e da razão μ	43
4.3	Calibração da distância focal e do coeficiente de distorção radial.	44
4.4	Deslocamentos dados pelo controlador do robô, pelo sistema de visão e erro entre eles, todos os dados estão em milímetros	45
4.5	Média e desvio padrão dos erros.	46

LISTA DE SÍMBOLOS

Símbolos

θ_1	Junta de rotação ao redor do eixo 1 do robô	[graus]
θ_2	Junta de rotação ao redor do eixo 2 do robô	[graus]
θ_3	Junta de rotação ao redor do eixo 3 do robô	[graus]
B_1, B_2, B_3	Posições das juntas 1, 2 e 3 da base do robô, respectivamente.	
E_1, E_2, E_3	Posições dos lados da plataforma móvel do robô.	
A_1, A_2, A_3	Parâmetro geométrico descrito na figura 2.6	[mm]
$\{X_B, Y_B, Z_B\}$	Sistema de coordenadas da base do robô como mostrado na figura 2.6	[mm]
$\{X_E, Y_E, Z_E\}$	Sistema de coordenadas do TCP do robô como mostrado na figura 2.6	[mm]
u_B	Parâmetro geométrico descrito na figura 2.4	[mm]
v_B	Parâmetro geométrico descrito na figura 2.4	[mm]
l_B	Comprimento do lado do triângulo que representa a base do robô	[mm]
u_E	Parâmetro geométrico descrito na figura 2.5	[mm]
v_E	Parâmetro geométrico descrito na figura 2.5	[mm]
l_E	Comprimento do lado do triângulo que representa a plataforma móvel robô	[mm]
l	Parâmetro geométrico descrito na figura 2.6	[mm]
L	Parâmetro geométrico descrito na figura 2.6	[mm]
R	Matriz de rotação 3x3 que alinha o sistema de coordenadas do objeto com o sistema de coordenadas da câmera. Possui coordenadas: $r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8$ e r_9 .	
T	Vetor de translação de coordenadas T_x, T_y e T_z .	
T_x	Distância da origem do sistema de coordenadas da câmera para a origem do sistema de coordenadas do objeto na direção x do sistema de coordenadas da câmera.	[mm]
T_y	Distância da origem do sistema de coordenadas da câmera para a origem do sistema de coordenadas do objeto na direção y do sistema de coordenadas da câmera.	[mm]

T_z	Distância da origem do sistema de coordenadas da câmera para a origem do sistema de coordenadas do objeto na direção z do sistema de coordenadas da câmera.	[mm]
f	Distância focal da lente da câmera.	
fx	Constante igual a $f \cdot s_u$.	
fy	Constante igual a $f \cdot s_v$.	
(C_x, C_y)	Coordenadas do centro da imagem.	[pixeis]
s_u	Fator de escala horizontal	[1/mm]
s_v	Fator de escala vertical	[1/mm]
μ	Razão dos fatores de escala.	
k	Coefficiente de distorção radial da lente.	
$\{xw, yw, zw\}$	Sistema de coordenadas global do sistema de visão.	[mm]
$\{x, y, z\}$	Sistema de coordenadas da câmera.	[mm]
(u, v)	Coordenadas da imagem	[mm]
$D_u()$	Operador para o componente horizontal das distorções da lente na direção u	[unidades físicas].
$D_v()$	Operador para o componente vertical das distorções da lente na direção v	[unidades físicas].
(X, Y)	Coordenadas da imagem	[pixeis]
$D_X()$	Operador para o componente horizontal das distorções da lente na direção X	[pixeis]
$D_Y()$	Operador para o componente vertical das distorções da lente na direção Y	[pixeis]
(X_f, Y_f)	Coordenadas da imagem no computador	[pixeis]
O	Origem do sistema de coordenadas da câmera.	
O_i	Origem do sistema de coordenadas da imagem.	
P_u	Ponto sem distorção no modelo <i>pinhole</i> .	
P_d	Ponto com distorção no modelo <i>pinhole</i> .	
h	Distância entre as amostras do sistema de similaridade de erros.	
$\gamma(h)$	Função semivariograma.	
$N(h)$	Número de pares de pontos separados por uma distância h .	
$e(P_i)$	Erro de posicionamento na amostra P_i .	
Co	Semivariância de duas amostras em um mesmo ponto.	
C	Covariância de duas amostras no mesmo ponto.	
λ	Multiplicador de lagrange.	
\mathbf{C}	Matriz de covariância das amostras.	
\mathbf{c}	Vetor de covariância das amostras.	
e_x	Erro medido no ponto P_i na direção x .	[mm]
e_y	Erro medido no ponto P_i na direção y .	[mm]

e_z	Erro medido no ponto P_i na direção z.	[mm]
\hat{e}_x	Erro estimado no ponto P_i na direção x.	[mm]
\hat{e}_y	Erro estimado no ponto P_i na direção y.	[mm]
\hat{e}_z	Erro estimado no ponto P_i na direção z.	[mm]
$w_x^{(i)}$	Peso da interpolação em na direção x.	
$w_y^{(i)}$	Peso da interpolação em na direção y.	
$w_z^{(i)}$	Peso da interpolação em na direção z.	

Subscritos

$Cov(h)$	Covariância entre duas amostras separada por uma distância h .
----------	--

Sobrescritos

$\hat{}$	Valor estimado para a variável
—	Segmento de reta

Siglas

DOF	Degrees Of Freedom - <i>Graus de Liberdade</i>
RAC	Radial Alignment Constraint - <i>Restrição de Alinhamento Radial</i>
TCP	Tool Center Point - <i>Ponto Central da Ferramenta</i>

Capítulo 1

Introdução

1.1 Contextualização

A programação de um robô pode ser definida como o processo que irá indicar a sequência de ações que um robô deverá cumprir durante a realização de uma tarefa [7]. As duas principais formas de programação de um robô são a programação *online*, e a programação *offline* [8]. No caso da programação *offline* o robô pode ser programado a distância via linhas de comando que serão executadas pelo controlador do manipulador, no momento da programação não é necessário que se faça o uso do robô. Já no caso da programação *online* o robô é movimentado para as posições que se deseja e as suas posições são salvas pelo controlador que poderá executá-las quando necessário [8].

A programação *offline* é mais vantajosa pois não necessita que um operador fique em contato com o robô durante a sua programação, evita a parada da linha de produção em que o robô está inserido, reduz o tempo de *setup*, entre outros. Em uma linha de produção que possui entre 30 e 40 robôs, o tempo de programação utilizando a programação *online* pode chegar a 400 horas [9].

Do ponto de vista prático existem limitações que podem impossibilitar o uso da programação *offline* em algumas aplicações. Esses erros podem ocorrer devido a vários fatores. Em relação ao robô as principais fontes de erro são as imprecisões no modelo geométrico do robô usado pelo controlador. Em relação ao ambiente, é difícil na prática se obter a posição precisa de objetos com relação ao um determinado sistema de coordenadas. Em relação ao controlador, sua imprecisão numérica pode afetar na precisão do posicionamento do robô [10].

Para que se possa contornar os problemas citados no paragrafo anterior e possibilitar o uso da programação *offline*, é necessário que se faça o uso de um modelo de calibração que irá corrigir os erros de posicionamento do robô. A grande maioria dos algoritmos de calibração de robôs existentes baseiam-se na identificação e compensação dos parâmetros geométricos reais do robô, mas essa abordagem envolve desenvolvimento matemático complexo que influencia no custo computacional do sistema de calibração. Dessa forma, é interessante que se possa utilizar um modelo de calibração que não necessite da obtenção dos parâmetros geométricos do robô.

Nesse trabalho é desenvolvido um modelo de calibração baseado em análise de erros com avaliação experimental, utilizando um sistema de medição com base em visão computacional pela sua simplicidade e facilidade de instalação nos manipuladores.

1.2 Definição do problema

Robôs paralelos ou máquinas cinemáticas paralelas são mecanismos de cadeia cinemática fechada, em que o efetuador terminal é ligado a base do robô por duas ou mais cadeias cinemáticas independentes. Os robôs paralelos são considerados altamente precisos devido ao não acumulo dos erros de junta, diferentemente dos robôs seriais, sendo muito utilizados em tarefas que necessitam de alta precisão no posicionamento do efetuador terminal. Em contrapartida, tais robôs podem apresentar perda de precisão devido a erros de fabricação, folga nos elementos de transmissão de movimento, carga de manuseio do robô, etc. Portanto, é necessário aplicar um método de calibração de forma a compensar tais imperfeições que afetam o correto posicionamento do manipulador.

1.3 Objetivos do projeto

O principal objetivo desse trabalho é desenvolver um sistema de calibração de robôs com um sistema de medição baseado em visão computacional, e implementar um modelo estatístico com base em similaridade de erro que seja capaz de emular a cinemática direta complexa dos robôs paralelos, a partir da obtenção de parâmetros de equações de regressão não-linear que relacionem o erro de posicionamento do elemento terminal com cada uma das variáveis de juntas dos atuadores do robô.

A fim de alcançar tais objetivos foram seguidos os seguintes passos:

- Implementar e analisar a precisão de um sistema de medição baseado em visão computacional capaz de medir posições do manipulador.
- Implementar um programa computacional com auxílio do software MATLAB, que relaciona o posicionamento do robô dado pelo controlador com o posicionamento dado pelo sistema de visão computacional por regressão não-linear.
- Desenvolver um modelo de calibração do posicionamento de um robô baseado em análise estatística de similaridade de erros.
- Testar o sistema de calibração desenvolvido com o robô ABB-IRB-360.

1.4 Apresentação do manuscrito

Esse trabalho está dividido em cinco capítulos que permitem um entendimento sequencial e completo das ferramentas utilizadas para o seu desenvolvimento.

No primeiro capítulo é feita a contextualização do problema, a definição do problema e os objetivos que se deseja alcançar com a realização do trabalho.

O capítulo 2 é voltado para a revisão bibliográfica de todas as ferramentas teóricas necessárias para a implementação do trabalho. A revisão bibliográfica foi dividida em duas partes principais, sendo a primeira sobre robótica que inclui uma visão geral sobre calibração de robôs e a descrição do robô utilizado. Na segunda parte é feita a revisão bibliográfica de visão computacional sobre: modelo de câmera utilizado, descrição da câmera utilizada e calibração de câmeras. Tais passos serão necessários para entender como se obtém os parâmetros extrínsecos e intrínsecos da câmera e como o sistema de visão é utilizado para medição de posicionamentos do robô.

No capítulo 3 é feito o desenvolvimento do trabalho e também a explicação dos procedimentos experimentais utilizados, com base na fundamentação teórica anteriormente desenvolvida.

No capítulo 4 são mostrados os resultados finais do sistema de calibração.

E por fim, no capítulo 5 são mostradas as conclusões e discussões a respeito dos resultados finais obtidos.

Capítulo 2

Fundamentos Teóricos

2.1 Introdução

Este capítulo faz um breve resumo sobre o problema da calibração de robôs industriais e uma descrição do robô ABB IRB 360 utilizado no projeto. No campo de visão computacional são introduzidas informações como: modelo de câmera, calibração de câmeras e uma descrição completa sobre o desenvolvimento de algoritmos de visão computacional que irão dar base para o entendimento das medições de posicionamento do robô feitas com esse sistema. Por fim é feita a descrição do modelo estatístico com base em similaridade de erros que servirá de base para a calibração dos erros de posicionamento do robô.

2.2 Calibração de Robôs

Os robôs industriais são utilizados atualmente nas mais diversas tarefas como soldagem, montagem de produtos, pintura, movimentação de peças, entre muitas outras. Sendo assim, o uso dessas máquinas se torna cada vez mais necessário para aumentar a quantidade e a qualidade dos produtos produzidos. Boa parte dessas tarefas exigem precisão do posicionamento do robô.

Existem dois tipos principais de programação de um robô industrial: a programação online, em que o programador necessita ter contato direto com o robô para programá-lo, e a programação offline, em que o programador pode programar os movimentos do robô sem a necessidade de contato direto. A programação offline é mais vantajosa pois evita paradas longas na linha de produção para programar robôs. Mas, existem dificuldades que tornam a programação offline difícil de ser praticada, uma dessas dificuldades é a falta de precisão no posicionamento estático dos robôs, e é aí que a calibração do robô desempenha um papel importante [1].

A calibração de um robô busca melhorar a precisão de posicionamento do robô, por meio de modificação de seu software. O processo de calibração de um robô consiste em quatro passos: modelagem, medição, identificação e implementação. A figura 2.1 mostra como é feita a calibração de um robô na prática. No processo de modelagem busca-se encontrar a modelagem cinemática direta

do robô, através da qual se pode relacionar suas variáveis de junta com os erros de posicionamento. O sistema de medição é o responsável por encontrar os erros de posicionamento do TCP (Tool Center Point) do robô. A partir de um sistema computacional pode-se fazer o processo de identificação dos parâmetros necessários para a última fase da calibração. Na fase de implementação os parâmetros cinemáticos identificados são alterados na unidade de controle do robô, concluindo-se o processo de calibração.

A calibração de um robô pode ser dividida em geométrica e não-geométrica, na calibração geométrica busca-se encontrar parâmetros que influenciam no posicionamento estático do robô, e na calibração não-geométrica busca-se encontrar os parâmetros que influem no posicionamento dinâmico do robô. Mas, os erros geométricos são os principais fatores que provocam erros de posicionamentos nos robôs [1]. Dessa forma, o foco desse trabalho será a correção dos erros de posicionamento estático do robô.

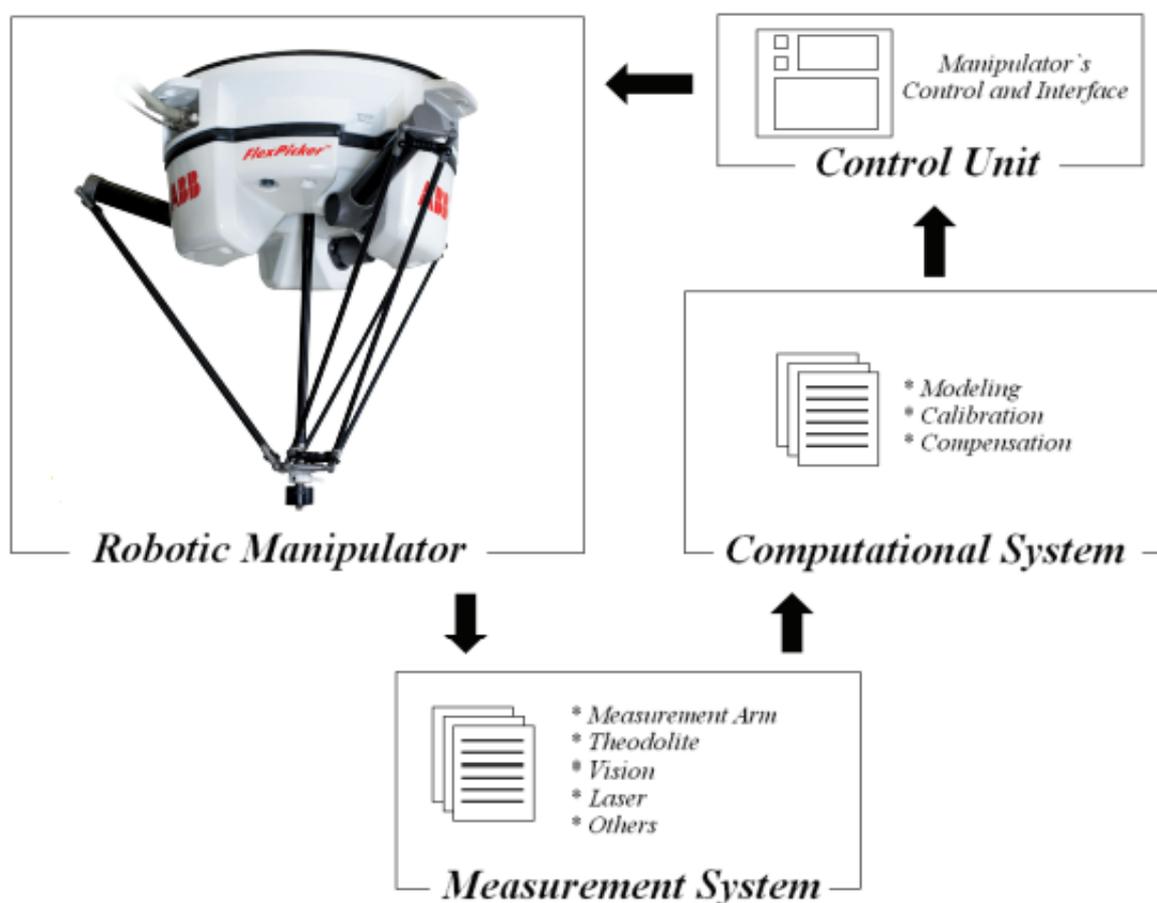


Figura 2.1: Diagrama de blocos do sistema de calibração de um robô [1] (adaptado)

Para esse trabalho, no processo de modelagem foram desenvolvidas equações que relacionam as variáveis de junta do robô com o erro de posição do efetuador terminal. Para isso é necessário

utilizar um sistema de medição capaz de medir os posicionamentos da plataforma móvel do robô. Com isso, pode-se relacionar o erro do sistema de medição com as variáveis de junta do robô. No processo de identificação deve-se encontrar um conjunto de parâmetros, referentes aos erros de posição e orientação do efetuador terminal.

As equações de cinemática inversa do robô, mostram que a posição final do efetuador terminal e as variáveis de junta são relacionadas diretamente. Portanto os erros de posição do robô irão apresentar semelhança de erros, que está relacionada com os desvios das variáveis de junta, sendo assim, é utilizada a aplicação da análise estatística de similaridade de erros para modelar as amostras de posicionamento do robô. A partir daí pode ser feito o processo de calibração do robô IRB 360.

2.2.1 Descrição do Robô Utilizado

O robô utilizado nesse projeto é o **ABB IRB 360 flexpicker**, como o mostrado na figura 2.2. Tal manipulador foi baseado na família de robôs IRB 140, foi projetado para indústrias que possuem grande necessidade de flexibilização, como tarefas de montagem e tarefas *pick and place*. O IRB 360 pode ser utilizado também, em tarefas de exigem altas velocidades, pois ele pode realizar até 200 tarefas *pick* por minuto [2].



Figura 2.2: Robô ABB IRB 360 *flexpicker* [2]

As características do robô como: capacidade de carga, alcance, número de eixos, entre outras, depende da especificação do robô *flexpicker*, pois o mesmo possui diferentes modelos. O modelo utilizado no projeto é o **IRB 360-8/1130**, que possui capacidade de manuseio de cargas de até 8 kg, diâmetro da base de 1130 mm e 4 eixos de movimentação.

O IRB 360 possui 3 juntas rotacionais ligadas à base do robô que realizam movimentos ao redor dos eixos 1, 2 e 3, mostrados na figura 2.3, e outra junta de rotação ligada ao efetuador terminal, que realiza movimento ao redor do eixo 4. As variáveis de junta de rotação serão chamadas de θ_1 , θ_2 e θ_3 , e realizam movimentos ao redor dos eixos 1, 2 e 3 respectivamente. Os Graus de liberdade de um robô, é o número de variáveis independentes que são necessárias para se definir a posição de um corpo no espaço em qualquer instante. Dessa forma, o IRB 360 possui 4 graus de liberdade, sendo 3 para movimentos de translação em X, Y e Z, e um quarto para rotação simples em torno do eixo fixo Z.

Esse robô é composto por três braços idênticos dispostos em paralelo entre a base fixa superior e a plataforma móvel inferior (elemento terminal). As juntas de revolução superiores são acionadas via atuadores rotacionais fixados na base. Em seu modelo o ângulo nulo é definido na posição em que o elo está no plano horizontal, ou seja, quando $[\theta_1 \ \theta_2 \ \theta_3]^T = [0^\circ \ 0^\circ \ 0^\circ]$.

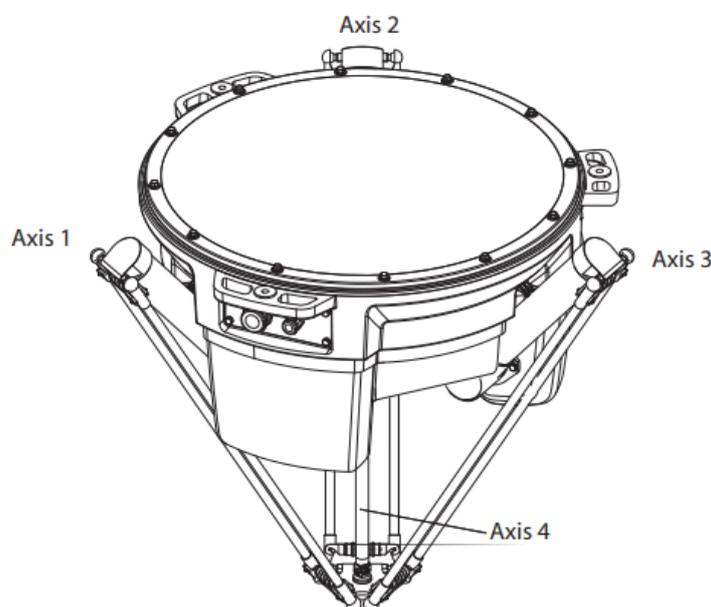


Figura 2.3: Eixos de movimentação do robô [2]

Após a breve introdução feita sobre as características do IRB 360, será feita uma análise geométrica do manipulador, necessária para o desenvolvimento das modelagens cinemáticas direta e inversa do robô.

A modelagem geométrica feita posteriormente, foi baseada no modelo descrito em [3] com algumas alterações para ser adaptada ao robô ABB IRB 360. Duas alterações principais foram feitas, a primeira os eixos Z_B e Z_E , colocados na base e na plataforma móvel foram direcionados para baixo, pois dessa forma os deslocamentos em Z do robô serão positivos, na segunda alteração, os ângulos de junta θ_2 e θ_3 foram trocados para se adaptar ao modelo do robô utilizado nesse trabalho.

A figura 2.4 mostra uma representação geométrica da base do robô, modelada como um triângulo equilátero. Nessa figura os pontos B_1 , B_2 e B_3 representam as posições das juntas da base do

robô, e l_B representa o comprimento do lado do triângulo. Um sistema de coordenadas cartesiano (X_B, Y_B, Z_B) foi colocado no triângulo, e outros dois valores, u_B e v_B , também foram colocados, para auxiliar no cálculo de algumas transformações [3].

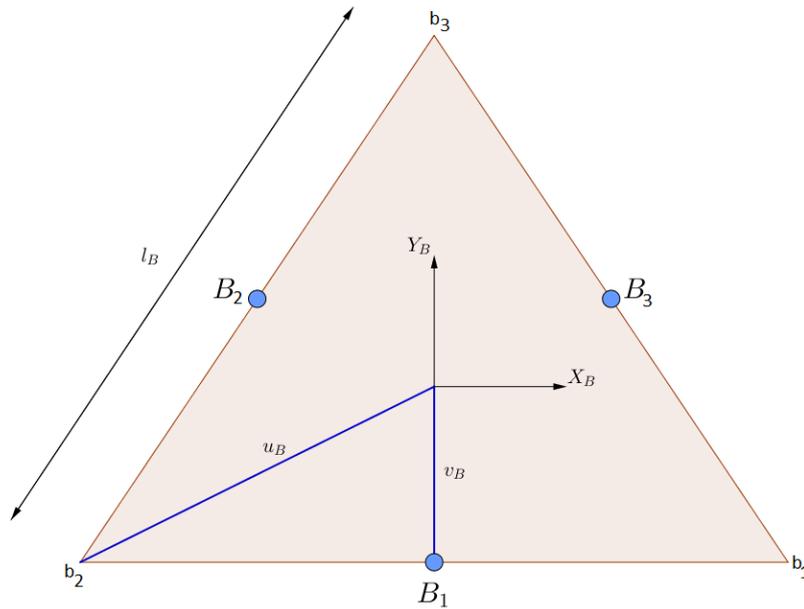


Figura 2.4: Representação geométrica da base do robô.

Dessa forma, seja ${}^B B_i$ uma transformação linear da origem do sistema de coordenadas (X_B, Y_B, Z_B) para o ponto B_i , e considerando a representação da figura 2.4 pode-se desenvolver as seguintes transformações, que servirão para o cálculo de cinemática do robô:

$${}^B B_1 = \begin{bmatrix} 0 \\ -v_B \\ 0 \end{bmatrix} \quad (2.1)$$

$${}^B B_2 = \begin{bmatrix} -\frac{\sqrt{3}}{2}v_B \\ \frac{1}{2}v_B \\ 0 \end{bmatrix} \quad (2.2)$$

$${}^B B_3 = \begin{bmatrix} \frac{\sqrt{3}}{2}v_B \\ \frac{1}{2}v_B \\ 0 \end{bmatrix} \quad (2.3)$$

Os vértices do triângulo equilátero da base fixa são:

$${}^B b_1 = \begin{bmatrix} \frac{l_B}{2} \\ -v_B \\ 0 \end{bmatrix} \quad (2.4)$$

$${}^B b_2 = \begin{bmatrix} -\frac{l_B}{2} \\ -v_B \\ 0 \end{bmatrix} \quad (2.5)$$

$${}^B b_3 = \begin{bmatrix} 0 \\ u_B \\ 0 \end{bmatrix} \quad (2.6)$$

em que:

$$v_B = \frac{\sqrt{3}}{6}l_B \quad u_B = \frac{\sqrt{3}}{3}l_B \quad v_E = \frac{\sqrt{3}}{6}l_E \quad u_E = \frac{\sqrt{3}}{3}l_E$$

Uma análise geométrica idêntica foi feita para a plataforma móvel fixada ao efetuador terminal, a figura 2.5 mostra tal representação, que também possui um sistema de coordenadas (X_E, Y_E, Z_E) , nesse caso l_E representa o comprimento da lateral do triângulo, E_1 , E_2 e E_3 representam os vértices do triângulo, e também foram colocados valores, u_E e v_E , para auxiliar no cálculo das transformações.

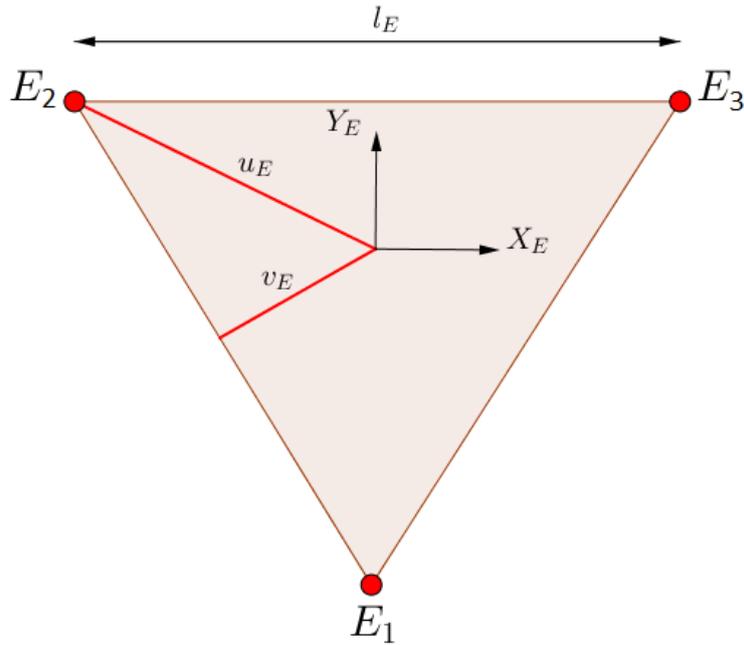


Figura 2.5: Representação geométrica da plataforma móvel do robô.

Seguindo a mesma notação utilizada para as transformações anteriores, encontramos que:

$${}^E E_1 = \begin{bmatrix} 0 \\ -u_E \\ 0 \end{bmatrix} \quad (2.7)$$

$${}^E E_2 = \begin{bmatrix} -\frac{l_E}{2} \\ v_E \\ 0 \end{bmatrix} \quad (2.8)$$

$${}^E E_3 = \begin{bmatrix} \frac{l_E}{2} \\ v_E \\ 0 \end{bmatrix} \quad (2.9)$$

2.2.2 Modelagem Cinemática Direta

A cinemática direta de um robô, busca encontrar a posição do efetuador terminal, em função de suas variáveis de junta. A cinemática direta de um robô paralelo é mais complexa que a cinemática inversa, pois requer a solução de múltiplas equações algébricas não lineares, e várias soluções válidas podem ser encontradas.

A figura 2.6 mostra uma representação geométrica da estrutura completa do robô paralelo. Primeiramente serão encontradas as transformações que ligam a origem da base do robô até o efetuador terminal, para cada um dos braços do manipulador, tais transformações serão utilizadas no cálculo da cinemática direta do robô.

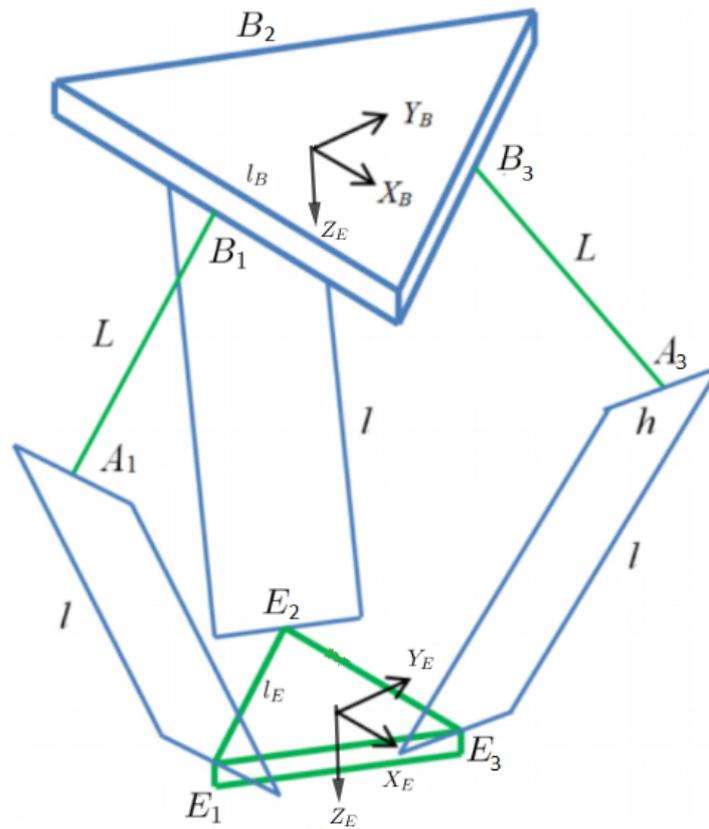


Figura 2.6: Representação geométrica completa do robô [3].

As transformações que levam $[B_1 \ B_2 \ B_3]^T$ até os pontos $[A_1 \ A_2 \ A_3]^T$ são dadas por:

$${}^B L_1 = \begin{bmatrix} 0 \\ -L \cos \theta_1 \\ L \sin \theta_1 \end{bmatrix} \quad (2.10)$$

$${}^B L_2 = \begin{bmatrix} -\frac{\sqrt{3}}{2}L \cos \theta_2 \\ \frac{1}{2}L \cos \theta_2 \\ L \sin \theta_2 \end{bmatrix} \quad (2.11)$$

$${}^B L_3 = \begin{bmatrix} \frac{\sqrt{3}}{2}L \cos \theta_3 \\ \frac{1}{2}L \cos \theta_3 \\ L \sin \theta_3 \end{bmatrix} \quad (2.12)$$

E as transformações que levam a origem do sistema da base, até os pontos A_i , são dadas por:

$${}^B A_i = {}^B B_i + {}^B L_i, i = 1, 2, 3 \quad (2.13)$$

Como os valores de ${}^B B_i$ já foram dados anteriormente pelas equações 2.1, 2.2, 2.3, então:

$${}^B A_1 = \begin{bmatrix} 0 \\ -v_B - L \cos \theta_1 \\ L \sin \theta_1 \end{bmatrix} \quad (2.14)$$

$${}^B A_2 = \begin{bmatrix} -\frac{\sqrt{3}}{2}(v_B + L \cos \theta_2) \\ \frac{1}{2}(v_B + L \cos \theta_2) \\ L \sin \theta_2 \end{bmatrix} \quad (2.15)$$

$${}^B A_3 = \begin{bmatrix} \frac{\sqrt{3}}{2}(v_B + L \cos \theta_3) \\ \frac{1}{2}(v_B + L \cos \theta_3) \\ L \sin \theta_3 \end{bmatrix} \quad (2.16)$$

Graças ao movimento apenas de translação da plataforma móvel, existe uma solução analítica direta para o qual um conjunto de solução correto é facilmente escolhido. Conhecendo-se as variáveis de junta $\Theta = [\theta_1 \ \theta_2 \ \theta_3]^T$ podemos calcular ${}^B A_i$, $i = 1, 2, 3$, utilizando as equações 2.14, 2.15 e 2.16. Considerando agora o novo diagrama mostrado na figura 2.7, define-se três centros de esferas virtuais ${}^B A_i v = {}^B A_i - {}^E E_i$, $i = 1, 2, 3$.

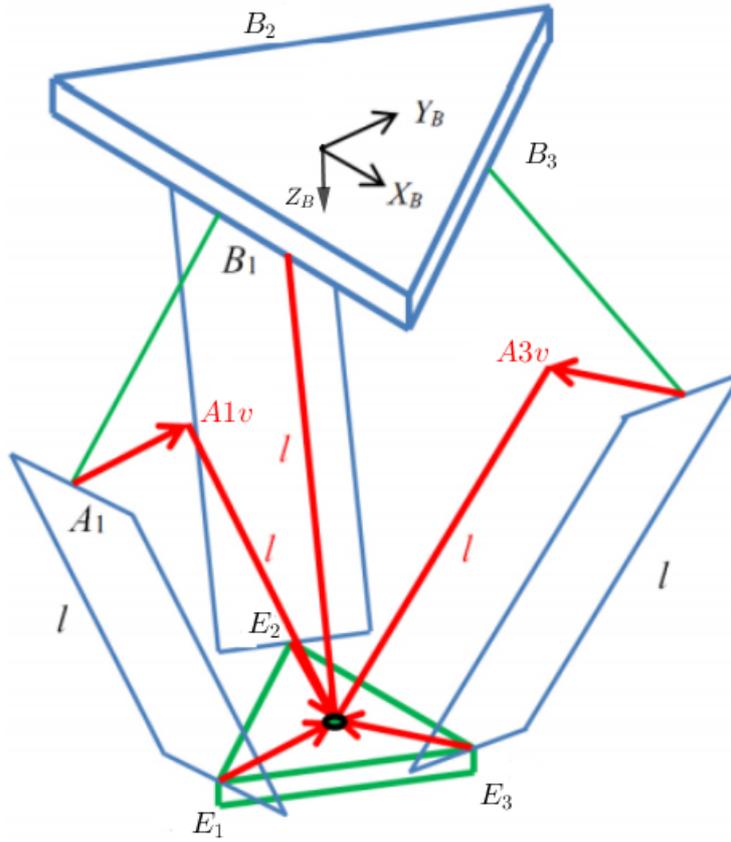


Figura 2.7: Representação geométrica completa do robô [3].

Os valores de ${}^B A_i v$, $i = 1, 2, 3$ são dados por:

$${}^B A_1 v = \begin{bmatrix} 0 \\ -v_B - L \cos \theta_1 + u_E \\ L \sin \theta_1 \end{bmatrix} \quad (2.17)$$

$${}^B A_2 v = \begin{bmatrix} -\frac{\sqrt{3}}{2}(v_B + L \cos \theta_2) - \frac{l_e}{2} \\ \frac{1}{2}(v_B + L \cos \theta_2) - v_e \\ L \sin \theta_2 \end{bmatrix} \quad (2.18)$$

$${}^B A_3 v = \begin{bmatrix} \frac{\sqrt{3}}{2}(v_B + L \cos \theta_3) + \frac{l_e}{2} \\ \frac{1}{2}(v_B + L \cos \theta_3) - v_e \\ L \sin \theta_3 \end{bmatrix} \quad (2.19)$$

A solução para a cinemática direta do robô será o ponto de interseção entre as três esferas, portanto, o ponto ${}^B E_E$ será a interseção entre os pontos:

$$([{}^B A_1 v], l) \quad (2.20)$$

$$([{}^B A_2 v], l) \quad (2.21)$$

$$([{}^B A_2 v], l) \quad (2.22)$$

em que $([{}^B A_i v], l)$ representa uma esfera centrada em ${}^B A_i v$, de raio l .

Agora será descrito o método analítico para resolver o problema da interseção das esferas, desenvolvido por [3]. Primeiramente, é suposto que existem três esferas (c_1, r_1) , (c_2, r_2) e (c_3, r_3) , em que (c_i, r_i) representa uma esfera centrada em $c_i = [x_i \ y_i \ z_i]^T$ de raio r_i . As coordenadas c_1 , c_2 e c_3 devem possuir um mesmo referencial, sendo que nesse caso o referencial será a origem da base do robô.

As equações das três esferas são dadas por:

$$(x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2 = r_1^2 \quad (2.23)$$

$$(x - x_2)^2 + (y - y_2)^2 + (z - z_2)^2 = r_2^2 \quad (2.24)$$

$$(x - x_3)^2 + (y - y_3)^2 + (z - z_3)^2 = r_3^2 \quad (2.25)$$

A solução $[x \ y \ z]^T$ das equações 2.23, 2.24 e 2.25, será o ponto de interseção das esferas, ou seja, a solução para a cinemática direta. Portanto, deve-se encontrar tais valores, isso pode ser feito da seguinte forma, seja:

$$a_{11}x + a_{12}y + a_{13}z = b_1 \quad (2.26)$$

$$a_{21}x + a_{22}y + a_{23}z = b_2 \quad (2.27)$$

em que:

$$a_{11} = 2(x_3 - x_1); \quad (2.28)$$

$$a_{12} = 2(y_3 - y_1); \quad (2.29)$$

$$a_{13} = 2(z_3 - z_1); \quad (2.30)$$

$$a_{21} = 2(x_3 - x_2); \quad (2.31)$$

$$a_{22} = 2(y_3 - y_2); \quad (2.32)$$

$$a_{23} = 2(z_3 - z_2); \quad (2.33)$$

e:

$$b_1 = r_1^2 - r_3^2 - x_1^2 - y_1^2 - z_1^2 + x_3^2 + y_3^2 + z_3^2 \quad (2.34)$$

$$b_2 = r_2^2 - r_3^2 - x_2^2 - y_2^2 - z_2^2 + x_3^2 + y_3^2 + z_3^2 \quad (2.35)$$

encontrando os valores de z nas equações 2.26 e 2.27:

$$z = \frac{b_1}{a_{13}} - \frac{a_{11}}{a_{13}}x - \frac{a_{12}}{a_{13}}y \quad (2.36)$$

$$z = \frac{b_2}{a_{23}} - \frac{a_{21}}{a_{23}}x - \frac{a_{22}}{a_{23}}y \quad (2.37)$$

subtraindo as equações 2.36 e 2.37:

$$x = f(y) = a_4y + a_5 \quad (2.38)$$

substituindo 2.38 em 2.37:

$$z = f(y) = a_6y + a_7 \quad (2.39)$$

em que:

$$a_1 = \frac{a_{11}}{a_{13}} - \frac{a_{21}}{a_{23}} \quad (2.40)$$

$$a_2 = \frac{a_{12}}{a_{13}} - \frac{a_{22}}{a_{23}} \quad (2.41)$$

$$a_3 = \frac{b_2}{a_{23}} - \frac{b_1}{a_{13}} \quad (2.42)$$

$$a_4 = -\frac{a_2}{a_1} \quad (2.43)$$

$$a_5 = -\frac{a_3}{a_1} \quad (2.44)$$

$$a_6 = \frac{-a_{21}a_4 - a_{22}}{a_{23}} \quad (2.45)$$

$$a_7 = \frac{b_2 - a_{21}a_5}{a_{23}} \quad (2.46)$$

substituindo 2.38 e 2.39 em 2.23, 2.24 e 2.25, encontra-se a seguinte equação quadrática em y :

$$ay^2 + by + c = 0 \quad (2.47)$$

em que:

$$a = a_4^2 + 1 + a_6^2; \quad (2.48)$$

$$b = 2a_4(a_5 - x_1) - 2y_1 + 2a_6(a_7 - z_1) \quad (2.49)$$

$$c = a_5(a_5 - 2x_1) + a_7(a_7 - 2z_1) + x_1^2 + y_1^2 + z_1^2 - r_1^2 \quad (2.50)$$

tem-se duas soluções:

$$y_{\pm} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \quad (2.51)$$

por fim, substituem-se os valores de y encontrados em 2.51 nas equações 2.38 e 2.39:

$$x_{\pm} = a_4 y_{\pm} + a_5 \quad (2.52)$$

$$z_{\pm} = a_6 y_{\pm} + a_7 \quad (2.53)$$

as duas soluções para a cinemática direta do robô serão:

$$[x_+ \ y_+ \ z_+]^T \quad (2.54)$$

$$[x_- \ y_- \ z_-]^T \quad (2.55)$$

Dessa forma, caso os vetores ${}^B A_i v$, $i = 1, 2, 3$ e o comprimento l dos elos do robô sejam conhecidos, pode-se determinar a posição final do efetuador terminal em função de suas variáveis de junta.

2.2.3 Modelagem Cinemática Inversa

O problema a ser solucionado pela cinemática inversa é encontrar os valores das variáveis de junta do robô $\Theta = [\theta_1 \ \theta_2 \ \theta_3]^T$, dada uma determinada posição do efetuador terminal ${}^B \mathbf{E}_E = [x \ y \ z]^T$.

Nesse caso, o problema da cinemática inversa pode ser resolvido independentemente para cada uma das três pernas do robô. Geometricamente, a solução da cinemática inversa para cada perna é a interseção entre um círculo (de raio L e centrado nos pontos das articulações situados em ${}^B \mathbf{B}_i$) e uma esfera (de raio l , centrada nos vértices da plataforma móvel, nos pontos ${}^E \mathbf{E}_i$).

A solução para esse problema será feita de forma analítica, utilizando as expressões encontradas anteriormente. As três equações escalares independentes de cinemática inversa são da forma [3]:

$$E_i \cos \theta_i + F_i \sin \theta_i + G_i = 0, \quad i = 1, 2, 3 \quad (2.56)$$

em que:

$$\begin{aligned} E_1 &= 2L(y + a), \quad F_1 = 2zL, \quad G_1 = x^2 + y^2 + z^2 + a^2 + L^2 + 2ya - l^2; \quad E_2 = -L(\sqrt{3}(x + b) + y + c), \\ F_2 &= 2zL, \quad G_2 = x^2 + y^2 + z^2 + b^2 + c^2 + L^2 + 2(xb + yc) - l^2; \quad E_3 = L(\sqrt{3}(x - b) - y - c), \\ F_3 &= 2zL, \quad G_3 = x^2 + y^2 + z^2 + b^2 + c^2 + L^2 + 2(-xb + yc) - l^2; \end{aligned}$$

em que:

$$a = v_B - u_E, \quad b = \frac{l_E}{2} - \frac{\sqrt{3}}{2}v_B, \quad c = v_E - \frac{1}{2}v_B$$

Para resolver as equações far-se-á uma substituição trigonométrica. Seja:

$$t_i = tg \frac{\theta_i}{2} \quad (2.57)$$

então

$$\cos\theta_i = \frac{1 - t_i^2}{1 + t_i^2} \quad (2.58)$$

e

$$\sen\theta_i = \frac{2t_i}{1 + t_i^2} \quad (2.59)$$

Substituindo as equações 2.58 e 2.59 na equação 2.56, encontra-se a seguinte expressão simplificada:

$$(G_i - E_i)t_i^2 + (2F_i)t_i + (G_i + E_i) = 0 \quad (2.60)$$

As soluções da equação 2.60 são:

$$t_i = \frac{-F_i \pm \sqrt{E_i^2 + F_i^2 - G_i^2}}{G_i - E_i} \quad (2.61)$$

por fim, utilizando as expressões 2.57 e 2.61 encontra-se o desejado:

$$\theta_i = 2\arctg(t_i) \quad (2.62)$$

Duas soluções resultam das duas raízes encontradas na equação 2.60. Ambas estão corretas, pois existem duas soluções válidas - as articulações da direita e esquerda. Isto produz duas soluções da cinemática inversa para cada perna do robô, para um total de oito possíveis soluções válidas. Geralmente a única solução com todos os joelhos dobrados para fora será escolhida.

2.3 Visão Computacional

A visão computacional pode ser vista como o problema inverso da formação de uma imagem, enquanto que o processo de formação de uma imagem se preocupa em como os objetos dão origem às imagens, a visão computacional utiliza imagens para recuperar descrições de objetos no espaço [11]. Atualmente a visão computacional é usada nas mais diversas aplicações, tais como: construção de trajetórias, reconstrução de objetos 3D, reconhecimento de movimentos, reconhecimento facial, etc. A figura 2.8 mostra algumas dessas aplicações.

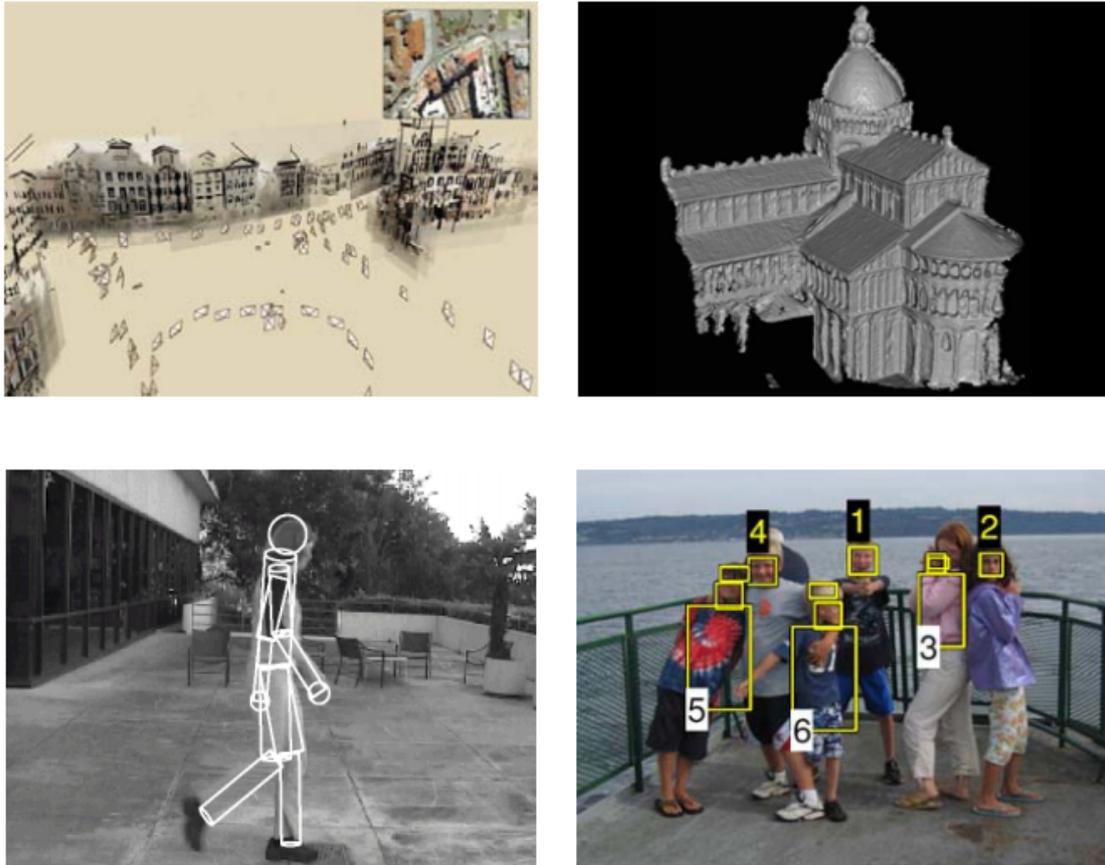


Figura 2.8: Exemplos de aplicações de visão computacional [4].

Em robótica industrial a visão computacional proporciona uma série de aplicações. Em sua grande maioria a combinação de hardware e software oferece ao robô orientação para que o mesmo possa se movimentar e executar suas funções, baseado em captura e processamento de imagens do ambiente. Com o uso da visão por computador é possível aumentar a qualidade de inspeção de produtos, aumentar a produtividade, reduzir o tempo de setup de máquinas, aumentar a flexibilidade da produção, entre outros.

Nesse trabalho a visão computacional é usada como um sistema de visão robótica que irá auxiliar nas medições de posicionamento e deslocamento do efetuador terminal do robô ABB IRB 360, com o uso do modelo de calibração RAC de câmeras.

2.3.1 Câmeras

Uma câmera é o dispositivo responsável pelo mapeamento entre o mundo 3D (espaço do objeto) e uma imagem 2D [12]. Para relacionar os pontos no espaço relativos a um sistema de coordenadas global com pontos no plano de uma imagem, é necessário que se desenvolva um modelo matemático que, a partir de matrizes de transformação, realiza o mapeamento dos pontos no plano da imagem. O sistema de coordenadas de um objeto no espaço será chamado de sistema de coordenadas global, agora em diante.

O processo de calibração de uma câmera busca encontrar os parâmetros de uma câmera, que podem ser divididos em *extrínsecos* e *intrínsecos*. Os parâmetros extrínsecos são os responsáveis por definir a localização e orientação do sistema de coordenadas da câmera em relação ao sistema de coordenadas global. Dessa forma, os parâmetros extrínsecos de uma câmera são:

- **Matriz de rotação \mathbf{R} :** é uma matriz ortogonal de rotação, que alinha o sistema de coordenadas da câmera com o sistema de coordenadas global, que pode ser representada por:

$$\mathbf{R} = \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix} \quad (2.63)$$

- **Vetor de translação \mathbf{T} :** é uma matriz coluna que translada o sistema de coordenadas global, de forma que fique na mesma posição do sistema de coordenadas da câmera, que pode ser representado por:

$$\mathbf{T} = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \quad (2.64)$$

em que T_x , T_y e T_z representam translações nos eixos do sistema de coordenadas global (xw, yw, zw).

Já os parâmetros intrínsecos descrevem as características de uma câmera, e são necessários para relacionar as coordenadas em pixels de uma imagem com as coordenadas correspondentes no sistema de referência da câmera. Os parâmetros intrínsecos de uma câmera são:

- **Distância focal da lente f :** Esse valor geralmente se refere à distância focal efetiva do sistema de lente física.
- **Centro da imagem (C_x, C_y) :** caso seja uma lente perfeita sem distorções, o centro da imagem pode ser considerado como o próprio centro do sensor CCD em pixels. Mas na prática isso não ocorre, pois quando as lentes possuem distorções o centro da imagem costuma variar em relação ao centro do sensor CCD.
- **Fator de escala μ :** O fator de escala para câmeras de matriz discreta é o fator de conversão entre um elemento da imagem do computador e um elemento do sensor da câmera [13].
- **Coefficiente de distorção da lente k :** é o coeficiente que "mede" o grau de distorção radial de uma lente real, e serve para calcular a relação entre as coordenadas distorcidas e não distorcidas de uma imagem.

2.3.2 Modelo de Câmera Sem Distorção

Como dito anteriormente, o modelo matemático do sistema de visão busca relacionar as coordenadas da imagem de um ponto no espaço, com as coordenadas desse mesmo ponto em relação ao

sistema de coordenadas global. O primeiro modelo aqui apresentado não possui distorções na lente, tal modelo é conhecido como câmera perspectiva ou modelo *pinhole*. A figura 2.9 mostra a geometria de uma câmera, em que $\{x_w, y_w, z_w\}$ representa o sistema de coordenadas global, $\{x, y, z\}$ o sistema de coordenadas da câmera e $\{X, Y\}$ representa o sistema de coordenadas da imagem medido em pixels.

A figura 2.9 também mostra a origem do sistema de coordenadas da câmera O , e a origem do sistema de coordenadas da imagem O_i . As duas origens estão localizadas sobre o mesmo eixo z , chamado de *eixo óptico*. A distância focal f , é, por definição, a distância entre essas duas origens.

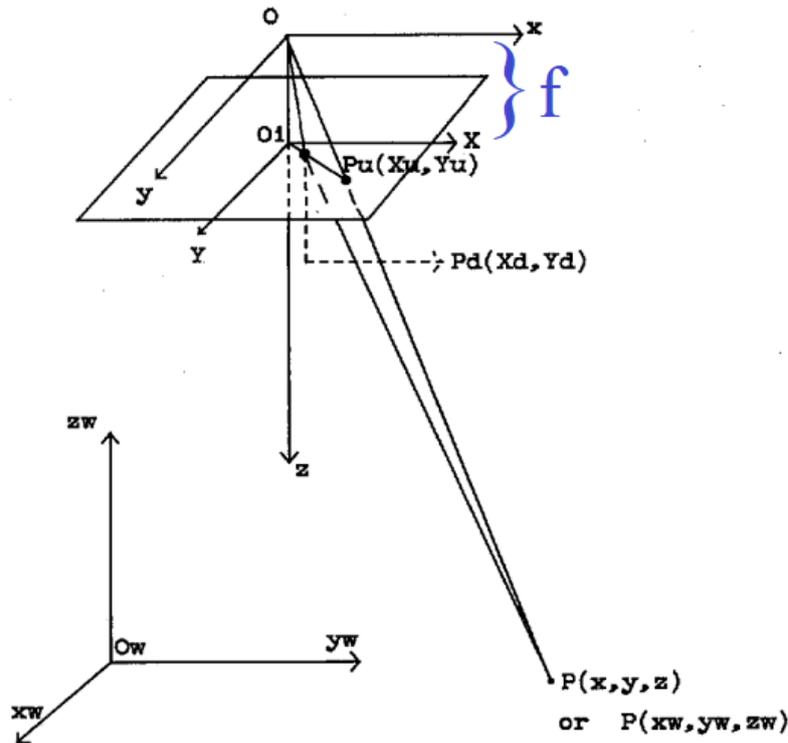


Figura 2.9: Geometria de uma câmera com projeção perspectiva e com distorção radial da lente [5].

Dessa forma, a transformação entre o sistema de coordenadas global $\{x_w, y_w, z_w\}$ e o sistema da câmera é dada por:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \mathbf{R} \cdot \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} + \mathbf{T} \quad (2.65)$$

em que \mathbf{R} e \mathbf{T} representam a matriz de rotação e o vetor de translação citados na seção anterior.

Considerando que o modelo *pinhole* não leva em consideração as distorções da lente (ponto P_u), então a transformação das coordenadas da câmera $\{x, y, z\}$ para as coordenadas da imagem são dadas pelas seguintes equações de perspectiva:

$$u = f \cdot \frac{x}{z} \quad (2.66)$$

$$v = f \cdot \frac{y}{z} \quad (2.67)$$

As coordenadas da imagem (X, Y) se relacionam com as coordenadas (u, v) , geralmente dadas em milímetros, da seguinte forma:

$$X = s_u \cdot u \quad (2.68)$$

$$Y = s_v \cdot v \quad (2.69)$$

Os fatores de escala s_u e s_v são os responsáveis por converter as coordenadas da câmera em milímetros para as coordenadas da imagem (X, Y) em pixels. Dessa forma, os fatores de escala s_u e s_v e a distância focal f são os parâmetros intrínsecos da câmera sem distorção, e \mathbf{R} e \mathbf{T} são os seus parâmetros extrínsecos.

Combinando-se as equações 2.66 e 2.68 e também as equações 2.67 e 2.69, obtêm-se:

$$X = s_u \cdot u = f \cdot s_u \cdot \frac{x}{z} = f x \cdot \frac{x}{z} \quad (2.70)$$

$$Y = s_v \cdot v = f \cdot s_v \cdot \frac{y}{z} = f y \cdot \frac{y}{z} \quad (2.71)$$

A partir da equação 2.65, obtêm-se as equações que relacionam as coordenadas globais com as coordenadas da imagem, para o modelo da câmera sem distorção [13]:

$$X = f x \cdot \frac{r1.xw + r2.yw + r3.zw + Tx}{r7.xw + r8.yw + r9.zw + Tz} \quad (2.72)$$

$$Y = f y \cdot \frac{r4.xw + r5.yw + r6.zw + Ty}{r7.xw + r8.yw + r9.zw + Tz} \quad (2.73)$$

As coordenadas (X, Y) são relativas ao canto da imagem, mas na prática elas devem ser relativas ao centro da imagem, como mostra a figura 2.9. Sendo assim a seguinte transformação deve ser feita:

$$X = X_f - C_x \quad (2.74)$$

$$Y = Y_f - C_y \quad (2.75)$$

em que (X_f, Y_f) é um ponto qualquer na imagem, e (C_x, C_y) é o centro da imagem.

2.3.3 Modelo de Câmera Com Distorção

Embora o modelo *pinhole* apresente um mapeamento entre o mundo 3D e uma imagem, o mesmo não leva em consideração as deficiências existentes em lentes reais. Tais defeitos podem causar alterações no mapeamento dos pontos, como mostra a figura 2.9. Nessa figura, o ponto $P_u = (X_u, Y_u)$ representa o mapeamento do ponto $P(x_w, y_w, z_w)$ na imagem sem distorção, e o ponto $P_d = (X_d, Y_d)$ representa esse mesmo ponto com distorção. Dessa forma, para um modelo mais preciso é necessário que se leve em consideração as deformidades existentes em lentes reais.

Os principais tipos de distorções existentes em lentes reais são: distorção radial e distorção tangencial, como mostra a figura 2.10. A distorção radial ocorre principalmente pela curvatura radial dos elementos da lente, fazendo com que os pontos na imagem se desloquem para dentro ou para fora, dependendo do sinal do coeficiente de distorção radial. Já a distorção tangencial, existe quando a lente e o plano da imagem não são perfeitamente paralelos.

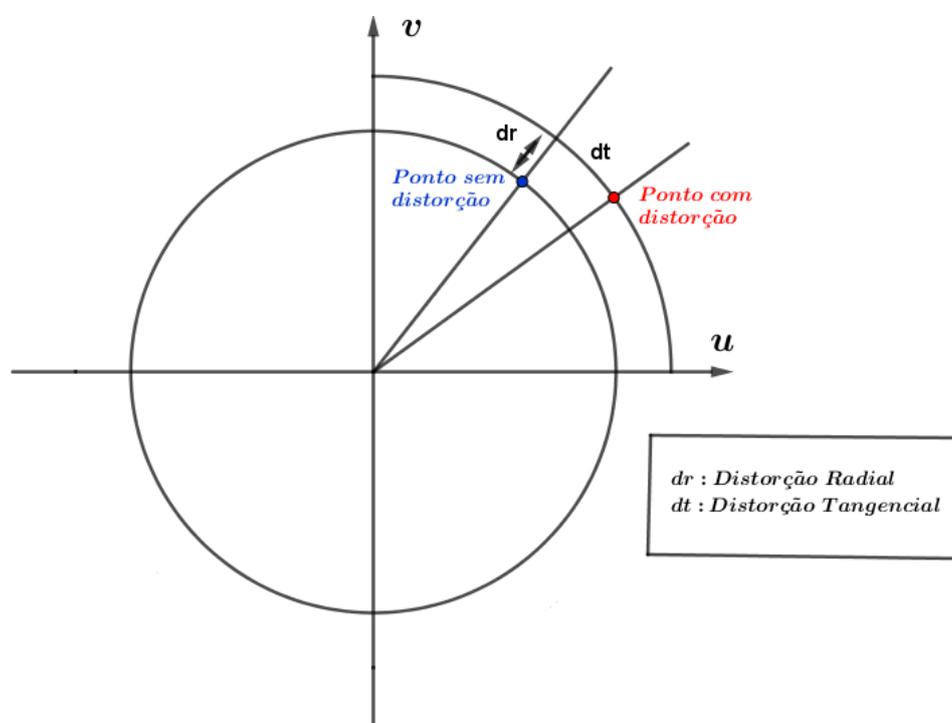


Figura 2.10: Efeitos das distorções radial e tangencial.

Para introduzir os efeitos das distorções das lentes, deve-se levar em consideração os erros posicionais causados pelas deformidades. Sendo assim, as seguintes alterações podem ser feitas [13], [14]:

$$u' = u + D_u(u, v) \quad (2.76)$$

$$v' = v + D_v(u, v) \quad (2.77)$$

em que u e v , definidos em 2.66 e 2.67, são as coordenadas livres de distorções, e u' e v' são esses mesmos pontos com distorções.

O modelo de câmera que será utilizado nesse trabalho é conhecido como modelo RAC. Nesse modelo só é considerada a distorção radial da lente da câmera, pois é a distorção dominante. Dessa forma, tendo em conta apenas a distorção radial [14]:

$$D_u(u, v) = k.u.(u^2 + v^2) \quad (2.78)$$

$$D_v(u, v) = k.v.(u^2 + v^2) \quad (2.79)$$

em que k representa o coeficiente de distorção radial da lente. Considerando que as coordenadas exatas (u, v) não podem ser obtidas de observações reais contaminadas por ruído, então os argumentos das equações descritas acima podem ser reescritos da seguinte forma [14]:

$$D_u(u, v) = -D'_u(u', v') \quad (2.80)$$

$$D_v(u, v) = -D'_v(u', v') \quad (2.81)$$

em que (u', v') são as coordenadas observáveis com distorção, e D'_u e D'_v possuem a forma das equações 2.76 e 2.77 com coeficientes redefinidos [13]. Reescrevendo as equações 2.76 e 2.77:

$$u = u' + D_u(u', v') \quad (2.82)$$

$$v = v' + D_v(u', v') \quad (2.83)$$

Com o auxílio das equações 2.70 até 2.73, pode-se obter o modelo matemático para lentes com distorção:

$$X + D_X(X, Y) = fx \cdot \frac{r1.xw + r2.yw + r3.zw + Tx}{r7.xw + r8.yw + r9.zw + Tz} \quad (2.84)$$

$$Y + D_Y(X, Y) = fy \cdot \frac{r4.xw + r5.yw + r6.zw + Ty}{r7.xw + r8.yw + r9.zw + Tz} \quad (2.85)$$

em que:

$$D_X(X, Y) = s_u \cdot D_u\left(\frac{X}{s_u}, \frac{Y}{s_v}\right) \quad (2.86)$$

$$D_Y(X, Y) = s_v \cdot D_v\left(\frac{X}{s_u}, \frac{Y}{s_v}\right) \quad (2.87)$$

2.3.4 O Modelo RAC de Calibração de Câmeras

Atualmente existem diferentes métodos de calibração de câmeras, o modelo RAC (Radial Alignment constraint - Restrição de Alinhamento Radial), descrito em [5], se encaixa no método de "dois passos", pois primeiramente utiliza uma solução direta para encontrar a maioria dos parâmetros de calibração e em seguida utiliza algum método iterativo para encontrar os parâmetros remanescentes. Tal modelo é amplamente utilizado, pois possui baixo custo computacional e precisão eficiente.

Nesse modelo, as coordenadas do centro da câmera (C_x, C_y) , e a razão dos fatores de escala (μ) :

$$\mu = \frac{fy}{fx} = \frac{s_v}{s_u} \quad (2.88)$$

devem ser conhecidos a priori. Os métodos de estimação desses fatores serão discutidos mais a frente.

O problema a ser resolvido por esse modelo, é encontrar os parâmetros extrínsecos \mathbf{R} e \mathbf{T} , os parâmetros fx e fy descritos nas equações 2.70 e 2.71, e o coeficiente de distorção radial k . Para isso, utiliza-se um conjunto de pontos coplanares cujas coordenadas são conhecidas no sistema de coordenadas global, e cujas coordenadas da imagem do computador são medidas [13].

Como nesse modelo é contabilizado apenas a distorção radial, das equações 2.78, 2.79, 2.86 e 2.87 obtêm-se:

$$D_X(X, Y) = k.X. \left(\frac{X^2}{s_u^2} + \frac{Y^2}{s_v^2} \right) \quad (2.89)$$

$$D_Y(X, Y) = k.Y. \left(\frac{X^2}{s_u^2} + \frac{Y^2}{s_v^2} \right) \quad (2.90)$$

seja:

$$r^2 = \frac{X^2}{s_u^2} + \frac{Y^2}{s_v^2} \quad (2.91)$$

ou, da equação 2.88:

$$r^2 = \mu.X^2 + Y^2 \quad (2.92)$$

então:

$$D_Y(X, Y) = k.Y.r^2 \quad (2.93)$$

$$D_Y(X, Y) = k.Y.r^2 \quad (2.94)$$

Portanto, das equações 2.84 e 2.85 obtêm-se:

$$X.(1 + k.r^2) = fx. \frac{r1.xw + r2.yw + r3.zw + Tx}{r7.xw + r8.yw + r9.zw + Tz} \quad (2.95)$$

$$Y.(1 + k.r^2) = fy. \frac{r4.xw + r5.yw + r6.zw + Ty}{r7.xw + r8.yw + r9.zw + Tz} \quad (2.96)$$

Sempre que as distorções diferentes da distorção radial, forem nulas, uma equação de restrição de alinhamento radial é mantida [13]. A seguinte relação é mantida:

$$\frac{X}{Y} = \mu^{-1}. \frac{r1.xw + r2.yw + r3.zw + Tx}{r4.xw + r5.yw + r6.zw + Ty} \quad (2.97)$$

que pode ser obtida dividindo-se a equação 2.95 pela equação 2.96.

Por fim, caso seja feita a seguinte aproximação:

$$(1 + k.r^2) \cong \left(\frac{1}{1 - k.r^2} \right), \text{ quando } k.r^2 \ll 1 \quad (2.98)$$

as equações 2.95 e 2.96, podem ser escritas do seguinte modo:

$$\frac{X}{1 - k.r^2} \cong fx. \frac{r1.xw + r2.yw + r3.zw + Tx}{r7.xw + r8.yw + r9.zw + Tz} \quad (2.99)$$

$$\frac{Y}{1 - k.r^2} \cong fy. \frac{r4.xw + r5.yw + r6.zw + Ty}{r7.xw + r8.yw + r9.zw + Tz} \quad (2.100)$$

Tal alteração é usada para linearizar o modelo RAC, e para que se possa encontrar uma solução linear de mínimos quadrados para o problema de calibração da câmera quando a distorção radial é considerada [13].

2.3.5 Descrição da Câmera e Lente Utilizadas

A câmera utilizada nesse projeto é como a mostrada na figura 2.11. Tal câmera é fabricada pela *Basler* e possui modelo **acA2500-14um**. Possui interface USB 3.0 sensor CCD *ON semiconductor MT9P031* mono CMOS, que oferece 14 frames por segundo com resolução de 5 MP. Já a lente utilizada é do fabricante *Edmund Optics* e possui distância focal fixa de 4.5 mm.



Figura 2.11: Câmera *Basler* utilizada no projeto.

A tabela 2.1 mostra as especificações gerais da câmera.

Resolução (H x V pixels)	2592 x 1944
Sensor da câmera	ON semiconductor MT9P031, CMOS
Tamanho do sensor	5.7 mm x 4.3 mm
Tamanho efetivo na diagonal	7.2 mm
Tamanho do pixel	2.2 μm x 2.2 μm
mono/color	mono
Interface	USB 3.0, taxa máxima: 5 Gbit/s
Alimentação	5 VDC, via USB
Peso	<80g
Software	Basler pylon Camera Software Suite 4.

Tabela 2.1: Especificações câmera *Basler acA2500-14um*.

A tabela 2.2 mostra as especificações gerais da lente da câmera.

Distância focal	4.5 mm
Tipo	lente de distância focal fixa
Abertura	f/2 - f/11
Número de elementos (grupos)	7(6)
Diâmetro externo	40 mm
Comprimento	37.5 mm
Peso	69 g

Tabela 2.2: Especificações lente *Edmund Optics 4.5mm/F1.95*.

2.4 Análise estatística baseada em similaridade de erros

Essa seção tem como objetivo apresentar os conceitos de análise de dados espaciais em estatística, os semivariogramas e a krigagem. Esses conceitos serão utilizados na análise estatística de similaridade de erros de posicionamentos do robô.

2.4.1 Análise Qualitativa da Similaridade de Erros de Posicionamento

No campo de estudo da agronomia de precisão, verifica-se que as propriedades do solo são dependentes dos locais onde os dados foram coletados, ou seja, dependem de suas posições espaciais. Por exemplo, caso a umidade do solo em um determinado ponto seja de 30% , em pontos próximos do ponto anteriormente medido a umidade tende a assumir um valor também próximo de 30%. Mas, caso essa medida seja feita a uma distância longa daquele ponto a umidade tenderá a assumir um valor diferente de 30%.

Existem inúmeros exemplos em diversos campos de estudo em que um determinado fator de interesse depende da distribuição espacial de suas amostras. Nesse contexto surgiu a geoestatística que utiliza o conceito de variáveis regionalizadas (variáveis que são dependentes da distribuição espacial) na avaliação da variabilidade espacial. A geoestatística não se limita apenas em obter um modelo de dependência espacial, pretende também estimar valores de pontos nos locais onde não foram coletadas amostras.

A função semivariância mede o grau de dependência espacial entre duas amostras. O seu valor depende da distância entre as amostras, pontos próximos possuem semivariância menor e pontos mais distantes possuem semivariância maior. Com os valores da semivariância, pode-se plotar o gráfico chamado semivariograma que é um medidor do grau de semelhança entre pontos vizinhos.

A partir desses conceitos apresentados, pode-se perceber que tal teoria pode ser aplicada ao objetivo desse trabalho. A cinemática direta do manipulador ABB-IRB-360 mostra que a posição do efetuador terminal depende apenas das variações dos ângulos de junta do robô, já que seus parâmetros cinemáticos são constantes. Portanto, os erros de posicionamento irão também depender dos valores das variáveis de junta do robô. Dessa forma, configurações de juntas semelhantes irão apresentar posições do TCP próximas e conseqüentemente erros de posicionamento próximos. Podem-se tratar então os erros de posicionamento como uma variável regionalizada [6][15].

Com o uso dos conceitos apresentados será possível fazer estimativa de erro de posicionamento para o robô em pontos onde não foram coletados dados, por conta da dependência espacial das variáveis de junta do manipulador [6].

A estimativa citada no paragrafo anterior comumente é feita através do uso da interpolação por krigagem [16]. A krigagem consiste em ponderar os vizinhos mais próximos do ponto a ser estimado obedecendo os critérios de não tendenciosidade, que significa que em média a diferença entre valores estimados e observados para o mesmo ponto deve ser nula e ter mínima variância, ou seja, que os estimadores possuam menor variância dentre todos os estimadores não tendenciosos [16].

Portanto, a krigagem pode ser aplicada para encontrar valores de erros para pontos não amostrados por meio da interpolação dos dados de outras amostras.

2.4.2 Análise Quantitativa da Similaridade de Erros de Posicionamento

A similaridade ilustrada anteriormente pode ser modelada quantitativamente para os erros de posição do robô a partir do uso da função semivariograma. A semivariância dos erros de posição do robô é a metade da variância do incremento entre dois erros de posição no espaço de trabalho. A partir desses valores de semivariâncias pode-se plotar um gráfico chamado semivariograma, que relaciona as semivariâncias entre pontos em função da distância h entre eles: [6][17]:

$$\gamma(h) = \frac{1}{2 \cdot N(h)} \sum_{i=1}^{N(h)} [e(P_i) - e(P_i + h)]^2 \quad (2.101)$$

em que $\gamma(h)$ chamado de semivariograma experimental, $N(h)$ é o número de pares de amostras que possuem distância constante igual a h , e $e(P_i)$ representa o erro de posição do ponto P_i . A curva do gráfico semivariograma pode ser obtida calculando-se vários valores de semivariâncias relacionados a diferentes distâncias h . Um gráfico semivariograma típico é mostrado na figura 2.12, em que C_o e C são a semivariância e a covariância de duas observações de erros de posição em um mesmo ponto, respectivamente [6]. O valor a corresponde à distância em que $\gamma(h)$ vale $(C_o + C)$.

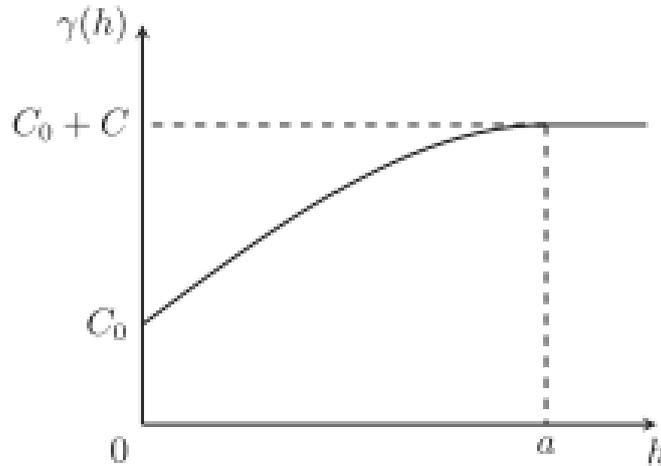


Figura 2.12: Curva típica de um semivariograma [6].

Existe uma relação entre a semivariância e a covariância [18], dada por:

$$\gamma(h) = Cov(o) - Cov(h) = \sigma^2 - Cov(h) \quad (2.102)$$

em que $Cov(h)$ é a covariância entre os erros de duas posições separadas por uma distância h , e σ^2 é a covariância entre os erros aleatórios de posicionamento em uma mesma posição do efetuador terminal, que é igual ao parâmetro C da figura 2.12.

A curva do semivariograma mostra algumas propriedades da similaridade de erros de posição. Quando a distância de dois pontos é pequena, a variância do incremento entre seus erros de posição é também pequena, isso implica que os erros de posições próximas irão apresentar maior similaridade que posições distantes. Pode-se ver que quanto maior for a distância h escolhida maior será a variância, e conseqüentemente maior será a independência entre os erros de posição. A partir de valores $h > a$ a semivariância se torna constante, isso significa que a similaridade de erros só existe para uma determinada faixa de valores de h [6][17].

O ajuste da curva do gráfico é baseado nos seguintes modelos de experiência: *modelo esférico*, *modelo circular*, *modelo Gaussiano* e *modelo linear* [6][18]. A forma geral do *modelo esférico* é dada por:

$$\gamma(h) = \begin{cases} C_0 + C \left(\frac{3h}{2a} - \frac{h^3}{2a^3} \right), & 0 \leq h \leq a \\ C_0 + C, & h > a \end{cases} \quad (2.103)$$

No caso do *modelo circular*, a forma geral será:

$$\gamma(h) = \begin{cases} C_0 + C [1 - \exp(-\frac{h}{a})], & 0 \leq h \leq 3a \\ C_0 + C, & h > 3a \end{cases} \quad (2.104)$$

Para o *modelo Gaussiano* será:

$$\gamma(h) = \begin{cases} C_0 + C [1 - \exp(-\frac{h^2}{a^2})], & 0 \leq h \leq \sqrt{3}a \\ C_0 + C, & h > \sqrt{3}a \end{cases} \quad (2.105)$$

Por fim, a forma geral do *modelo linear* é:

$$\gamma(h) = \begin{cases} C_0 + C \left(\frac{h}{a} \right), & 0 \leq h \leq a \\ C_0 + C, & h > a \end{cases} \quad (2.106)$$

Na prática, a curva do semivariograma pode ser obtida selecionando-se o melhor modelo de ajuste, de acordo com a distribuição dos dados amostrados [6].

2.4.3 Identificação e Compensação de Erros Utilizando Similaridade de Erros

2.4.3.1 Descrição do Algoritmo

A principal ideia da compensação de erros por similaridade de erros é utilizar a similaridade de erros de posição do TCP e das amostras para estimar o erro da posição do TCP, e então modificar o controle do robô para aplicar as devidas compensações [6][15]. Se os erros de posição de n amostras forem medidos no espaço de trabalho do robô, o erro de posição do TCP em um ponto P_o que precisa ser compensado, pode ser escrito como uma combinação linear das amostras:

$$\hat{e}_x(P_0) = \sum_{i=1}^n w_x^{(i)} e_x(P_i) \quad (2.107)$$

$$\hat{e}_y(P_0) = \sum_{i=1}^n w_y^{(i)} e_y(P_i) \quad (2.108)$$

$$\hat{e}_z(P_0) = \sum_{i=1}^n w_z^{(i)} e_z(P_i) \quad (2.109)$$

em que $\hat{e}_x(P_0)$, $\hat{e}_y(P_0)$ e $\hat{e}_z(P_0)$ são os erros de posição estimado para o ponto P_0 nos eixos X_B , Y_B e Z_B respectivamente, $w_x^{(i)}$, $w_y^{(i)}$ e $w_z^{(i)}$ são os pesos de interpolação, e $e_x(P_i)$, $e_y(P_i)$ e $e_z(P_i)$ são os erros das amostras P_i ($i = 1, 2, 3, \dots, n$) ao longo dos eixos X_B , Y_B e Z_B respectivamente, tal que $\{X_B, Y_B, Z_B\}$ representa o sistema de coordenadas da base do robô.

Portanto, o objetivo desse modelo de identificação de erros é calcular os pesos de interpolação entre a posição do TCP e cada uma das amostras. Para otimizar o valor estimado dos erros de posição do TCP, os pesos da interpolação devem garantir que a estimativa seja imparcial (não tendenciosa) enquanto a variância da estimativa é minimizada [6].

2.4.3.2 Identificação de Erros

Como um método que fornece a melhor predição linear não tendenciosa dos valores, o método de krigagem ordinária é usado para estimar os erros de posição do robô nesse trabalho, os diferentes tipos de krigagem existentes podem ser encontradas em [16]. Para garantir que a estimativa não seja tendenciosa, o valor esperado do erro de previsão entre o erro de posição estimado e o medido do ponto do TCP deve ser zero, ou seja, o valor esperado do erro de posicionamento estimado deve ser igual ao do erro de posicionamento real [6].

Por exemplo, o erro de posição do TCP ao longo do eixo x é dado por:

$$E[\hat{e}_x(P_0)] - E[e_x(P_0)] = m \quad (2.110)$$

em que m é a esperança matemática (valor esperado) do erro de posição. De acordo com a teoria de análise de dados espaciais, se os erros de posição forem considerados como variáveis aleatórias estacionárias, então:

$$E[\hat{e}_x(P_0)] = \sum_{i=1}^n w_x^{(i)} E[e_x(P_i)] = m \sum_{i=1}^n w_x^{(i)} = m \quad (2.111)$$

deve-se levar em consideração a seguinte restrição:

$$\sum_{i=1}^n w_x^{(i)} = \mathbf{w}_x^T \mathbf{I} = 1 \quad (2.112)$$

em que \mathbf{w}_x é um vetor $n \times 1$ com todos os pesos, e \mathbf{I} é um vetor de dimensão $n \times 1$ composto por números 1. Sob a condição de imparcialidade dos dados, a variância da estimativa do erro de

posição no eixo x_R é dado por:

$$\begin{aligned}
\sigma_x^2 &= E \left[(\hat{e}_x(P_o) - e_x(P_o))^2 \right] \\
&= E \left[\hat{e}_x^2(P_o) \right] + E \left[e_x^2(P_o) \right] - 2Cov \left[\hat{e}_x(P_o), e_x(P_o) \right] \\
&= E \left[\left(\sum_{i=1}^n w_x^{(i)} e_x(P_i) \right)^2 \right] + \sigma^2 - 2Cov \left[\sum_{i=1}^n w_x^{(i)} e_x(P_i), e_x(P_o) \right] \\
&= \sum_{i=1}^n \sum_{j=1}^n w_x^{(i)} w_x^{(j)} Cov \left[e_x(P_i), e_x(P_j) \right] + \sigma^2 - 2 \sum_{i=1}^n w_x^{(i)} Cov \left[e_x(P_i), e_x(P_o) \right] \\
&= \mathbf{w}_x^T \mathbf{C} \mathbf{w}_x + \sigma^2 - 2\mathbf{w}_x^T \mathbf{c}
\end{aligned} \tag{2.113}$$

em que \mathbf{C} é uma matriz de covariâncias $n \times n$, nesse caso C_{ij} é a covariância dos erros de posição das amostras P_i e P_j . σ^2 é a variância dos erros de posição, \mathbf{c} é um vetor de covariâncias $n \times 1$, em que o elemento c_i é a covariância dos erros de posição entre a posição do TCP P_0 e a amostra P_i . Uma vez que a curva do semivariograma é ajustada, os valores de σ^2 , C_{ij} e c_i podem ser calculados pela equação 2.102.

A estimativa pode ser transformada em um problema de otimização com restrições, após construção das equações 2.112 e 2.113. Minimizar $(\mathbf{w}_x^T \mathbf{C} \mathbf{w}_x + \sigma^2 - 2\mathbf{w}_x^T \mathbf{c})$, com:

$$\mathbf{w}_x^T \mathbf{I} = 1 \tag{2.114}$$

Utilizando o método do multiplicador de Lagrange, a função de Lagrange da equação 2.114 é:

$$L(\mathbf{w}_x, \lambda) = \mathbf{w}_x^T \mathbf{C} \mathbf{w}_x + \sigma^2 - 2\mathbf{w}_x^T \mathbf{c} + 2\lambda(\mathbf{w}_x^T \mathbf{I} - 1) \tag{2.115}$$

em que o número real λ é o multiplicador de Lagrange. A derivada da equação 2.115 é:

$$L(\mathbf{w}_x, \lambda) = 2(\mathbf{C} \mathbf{w}_x - \mathbf{c} + \lambda \mathbf{I}) \tag{2.116}$$

Sendo assim, para se obter o valor ótimo para \mathbf{w}_x a equação 2.116 deve ser igual a zero. Dessa forma, pode-se montar o seguinte sistema de equações:

$$\begin{cases} \mathbf{w}_x^T \mathbf{I} = 1 \\ \mathbf{C} \mathbf{w}_x + \lambda \mathbf{I} = \mathbf{c} \end{cases} \tag{2.117}$$

A equação 2.117 pode ser escrita na forma matricial:

$$\begin{bmatrix} \mathbf{C} & \mathbf{I} \\ \mathbf{I}^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{w}_x \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{c} \\ 1 \end{bmatrix} \tag{2.118}$$

Portanto, o problema de otimização é resolvido encontrando-se as incógnitas de um sistema linear do tipo $Ax = b$, e o vetor de pesos w_x pode ser obtido pela resolução da equação 2.118. O erro de posição do TCP ao longo do eixo X_B pode ser estimado substituindo-se o valor de w_x na equação 2.107 [6]. Os erros ao longo dos eixos Y_B e Z_B , são obtidos através do mesmo método.

2.4.3.3 Compensação de erros

O cálculo dos erros no modelo de identificação proposto nesse artigo, não leva em consideração os parâmetros cinemáticos do robô. Dessa forma, não é necessário modificar os parâmetros cinemáticos do robô durante o processo de calibração [6]. Após realizada a estimação dos erros de posição do TCP, podem-se modificar as coordenadas de posição do robô com as equações:

$$p'_x(P_o) = p_x(P_o) - \hat{e}_x(P_o) \quad (2.119)$$

$$p'_y(P_o) = p_y(P_o) - \hat{e}_y(P_o) \quad (2.120)$$

$$p'_z(P_o) = p_z(P_o) - \hat{e}_z(P_o) \quad (2.121)$$

em que $p_x(P_o)$, $p_y(P_o)$ e $p_z(P_o)$ são as coordenadas nominais do posicionamento do TCP ao longo dos eixos X_B , Y_B e Z_B , respectivamente. $p'_x(P_o)$, $p'_y(P_o)$ e $p'_z(P_o)$ são as coordenadas modificadas do posicionamento do efetuador terminal ao longo dos eixos X_B , Y_B e Z_B , respectivamente. Dessa forma, a compensação de erros é feita enviando-se $p'_x(P_o)$, $p'_y(P_o)$ e $p'_z(P_o)$ como comandos para o controlador do robô.

Capítulo 3

Desenvolvimento Experimental

3.1 Introdução

Este capítulo tem como objetivo mostrar como o sistema da calibração de câmera do modelo de restrição de alinhamento radial (RAC) foi implementado, de como são feitas as calibrações das coordenadas do centro da imagem e da razão μ .

3.2 Implementação do Modelo RAC de Calibração de Câmera

O modelo RAC possui dois estágios para determinar os parâmetros extrínsecos e intrínsecos da câmera. Primeiramente são determinados a matriz de rotação \mathbf{R} , e os parâmetros T_x e T_y do vetor \mathbf{T} . No segundo estágio, os parâmetros remanescentes são calculados. O algoritmo apresentado nesta seção, foi baseado no modelo apresentado na seção 3.2, que pode ser visto de forma detalhada em [5].

Nesse algoritmo é necessário um conjunto de pontos coplanares cujas coordenadas no sistema global são conhecidas. Uma placa de calibração como mostrada na figura 3.1 foi utilizada para essa tarefa. A escolha do sistema de coordenadas global $\{xw, yw, zw\}$ é livre, mas é interessante escolher uma posição em que $zw = 0$ para facilitar os cálculos. Deve-se ainda escolher uma posição em que T_y seja diferente de zero, pois, caso contrário, invalidaria o processo de calibração [5].

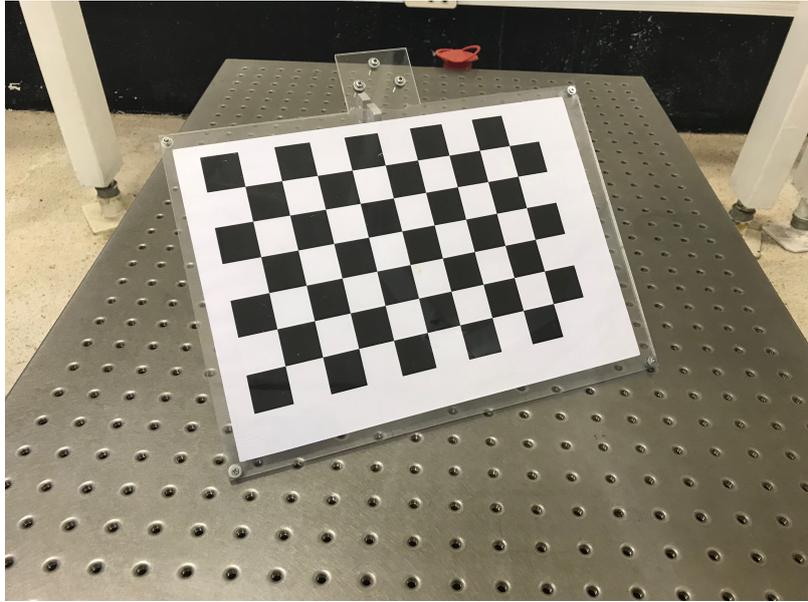


Figura 3.1: Placa de calibração com pontos coplanares utilizada no trabalho.

A placa de calibração utilizada nesse trabalho possui padrão xadrez com espaçamento de 25 mm entre os as bordas dos quadrados. A figura 3.2 mostra o padrão da placa da figura 3.1.

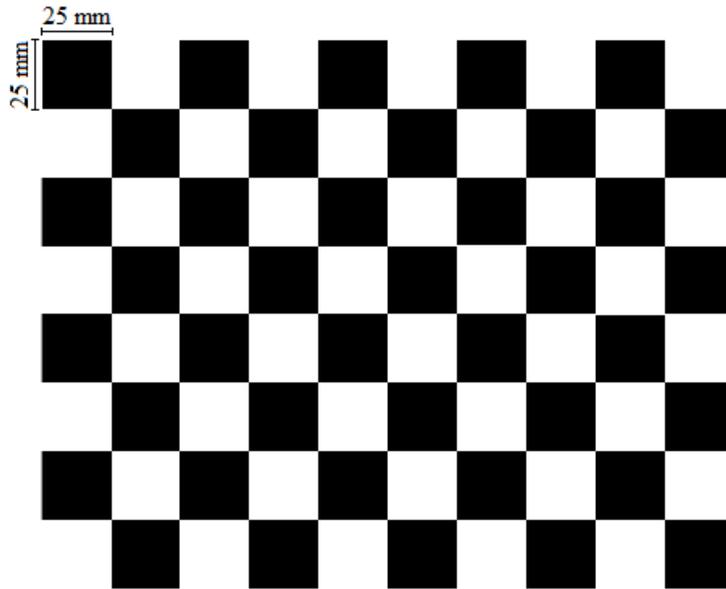


Figura 3.2: Padrão da placa de calibração.

Para que se possa calcular a matriz de rotação \mathbf{R} e os parâmetros T_x e T_y , assume-se que o centro da imagem (C_x, C_y) e a razão dos fatores de escala são conhecidos a priori.

O primeiro passo é calcular as coordenadas da imagem (X_i, Y_i) para os N pontos de calibração da placa, em que $i = 1, 2, 3, \dots, N$. O cálculo é feito da seguinte forma:

$$X_i = X_{f,i} - C_x \quad (3.1)$$

$$Y_i = Y_{f,i} - C_y \quad (3.2)$$

em que $(X_{f,i}, Y_{f,i})$ são as coordenadas em pixels na imagem do i -ésimo ponto da placa de calibração.

Logo após, são calculados parâmetros intermediários que irão auxiliar nos cálculos. Dividindo ambos os lados da equação 2.97 por T_y , e fazendo $z_w = 0$, é possível definir o seguinte sistema:

$$\begin{bmatrix} xw_i \cdot Y_i & yw_i \cdot Y_i & Y_i & -xw_i \cdot \mu \cdot X_i & -yw_i \cdot \mu \cdot X_i \end{bmatrix} \cdot \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{bmatrix} = \mu \cdot X_i \quad (3.3)$$

em que os parâmetros intermediários são definidos por:

$$[v_1 \quad v_2 \quad v_3 \quad v_4 \quad v_5] = \left[\frac{r_1}{T_y} \quad \frac{r_2}{T_y} \quad \frac{T_x}{T_y} \quad \frac{r_4}{T_y} \quad \frac{r_5}{T_y} \right] \quad (3.4)$$

Na equação 3.3, (xw_i, yw_i, zw_i) são as coordenadas globais do i -ésimo ponto de calibração. Vê-se que o número mínimo de equações para solução desse sistema é cinco. Isso implica que deve-se escolher pelo menos cinco pontos não-colineares na placa de calibração. Porém, esse número deve ser maior para que se possa ter um sistema sobre-determinado que possa ser resolvido pelo método **SVD** (Singular Value Decomposition - Decomposição em Valores Singulares), de modo a obter uma solução mais precisa. Tem-se ainda a restrição $T_y \neq 0$ para que o sistema possa ser resolvido. Isso implica que é necessário escolher o sistema de referência global $\{xw, yw, zw\}$ de forma adequada.

Os parâmetros intermediários irão auxiliar no cálculo de T_y . Primeiramente define-se uma matriz **C**, dada por:

$$\mathbf{C} = \begin{bmatrix} v_1 & v_2 \\ v_4 & v_5 \end{bmatrix} \quad (3.5)$$

Caso nenhuma coluna ou linha dessa matriz seja nula, então T_y pode ser calculado por:

$$T_y^2 = \frac{S_r - \sqrt{S_r^2 - 4 \cdot (v_1 \cdot v_5 - v_4 \cdot v_2)^2}}{2 \cdot (v_1 \cdot v_5 - v_4 \cdot v_2)^2} \quad (3.6)$$

em que $S_r = v_1^2 + v_2^2 + v_4^2 + v_5^2$.

Caso a matriz **C** possua alguma coluna ou linha nula, então T_y será calculado por:

$$T_y^2 = \frac{1}{v_i^2 + v_j^2} \quad (3.7)$$

em que v_i e v_j são os coeficientes não nulos da matriz \mathbf{C} .

Com o valor de T_y calculado, agora é necessário determinar o seu sinal. Para isso, considera-se que $T_y > 0$, e os seguintes cálculos são feitos:

$$\begin{aligned} r_1 &= v_1.T_y \\ r_2 &= v_2.T_y \\ r_4 &= v_4.T_y \\ r_5 &= v_5.T_y \\ T_x &= v_3.T_y \end{aligned} \tag{3.8}$$

Em seguida é escolhido um ponto de calibração qualquer, cujas coordenadas $(X_{f,i}, Y_{f,i})$ estão distantes do centro da imagem (C_x, C_y) , e possui coordenadas. Calculam-se as seguintes coordenadas:

$$\begin{cases} x = v_1.xw_i + r_2.yw_i + T_x \\ y = r_4.xw_i + r_5.yw_i + T_y \end{cases} \tag{3.9}$$

em que (xw_i, yw_i) são as coordenadas no sistema global do ponto de calibração arbitrário escolhido. O sinal de T_y é escolhido da seguintes forma:

- Se os sinais de x e X e de y e Y forem iguais, então mantém-se o sinal de T_y positivo.
- Se os sinais de x e X ou de y e Y forem diferentes, então o sinal de T_y será negativo, e os sinais de r_1, r_1, r_4, r_5 e T_x também são invertidos.

Agora é feito o cálculo da matriz \mathbf{R} . Dada uma submatriz 2x2 qualquer de uma matriz de rotação \mathbf{R} , existem duas soluções possíveis para essa matriz. Essas duas soluções irão gerar dois valores para fx , um positivo e outro negativo, nos próximos passos da calibração da câmera. Como o valor de fx é sempre positivo, pode-se usar essa restrição para eliminar uma das soluções possíveis da matriz \mathbf{R} . A primeira solução realizável é:

$$r_3 = \sqrt{1 - r_1^2 - r_2^2} \tag{3.10}$$

$$r_6 = -\text{sin}al(r_1.r_4 + r_2.r_5) \cdot \sqrt{1 - r_4^2 - r_5^2} \tag{3.11}$$

$$[r_7 \ r_8 \ r_9]^T = [r_1 \ r_2 \ r_3]^T \otimes [r_4 \ r_5 \ r_6]^T \tag{3.12}$$

O termo $\text{sin}al(r_1.r_4 + r_2.r_5)$ deve ser 1 se a operação $(r_1.r_4 + r_2.r_5)$ for positiva, ou -1 caso seja negativa. O símbolo \otimes representa o produto vetorial entre dois vetores. Se o sinal de fx , que será calculado mais a frente, for positivo essa solução da matriz \mathbf{R} será mantida, caso contrário a matriz será novamente calculada da seguinte forma:

$$r_3 = -\sqrt{1 - r_1^2 - r_2^2} \quad (3.13)$$

$$r_6 = \text{sinal}(r_1.r_4 + r_2.r_5).\sqrt{1 - r_4^2 - r_5^2} \quad (3.14)$$

$$[r_7 \ r_8 \ r_9]^T = [r_1 \ r_2 \ r_3]^T \otimes [r_4 \ r_5 \ r_6]^T \quad (3.15)$$

Com o cálculo da matriz \mathbf{R} e dos valores de T_x e T_y a primeira etapa da calibração está concluída. Na segunda etapa são calculados os parâmetros fx , fy , T_z e k em função dos valores já obtidos.

O valor de r pode ser obtido com a equação 2.92. Os valores de x_i , y_i e w_i são calculados da seguinte forma:

$$\begin{cases} x_i = r_1.xw_i + r_2.yw_i + T_x \\ y_i = r_4.xw_i + r_5.yw_i + T_y \\ w_i = r_7.xw_i + r_8.yw_i \end{cases} \quad (3.16)$$

O número mínimo de pontos de calibração para que a equação 3.16 tenha solução deve ser 3, mas pode-se escolher mais pontos de forma a gerar um sistema sobre-determinado que pode ser resolvido pelo método de decomposição em valores singulares (SVD).

Os valores obtidos com a resolução da equação 3.16 podem ser usados para calcular os parâmetros intrínsecos restantes:

$$\begin{cases} fy = fx.\mu \\ k = \frac{k.fx}{fx} \end{cases} \quad (3.17)$$

3.2.1 Transformação do Sistema da Câmera Para o Sistema de Coordenadas Global

Uma vez que os parâmetros externos da câmera (\mathbf{R} , T_x , T_y e T_z) foram determinados para uma determinada posição da câmera no sistema de coordenadas da câmera, é necessário estabelecer uma transformação inversa para determinar o vetor de translação (XW , YW , ZW) de qualquer ponto na placa de calibração para o centro óptico da câmera, representado no sistema de coordenadas global [13]. Como visto anteriormente na equação 2.65, a matriz \mathbf{R} é a matriz de rotação que alinha o sistema de coordenadas global com o sistema de coordenadas da câmera. Dessa forma:

$$\mathbf{T} = -\mathbf{R} \cdot \begin{bmatrix} XW + xw \\ YW + yw \\ ZW + zw \end{bmatrix} \quad (3.18)$$

Multiplicando ambos os lados da equação 3.18 por \mathbf{R}^{-1} , e reorganizando o resultado:

$$\begin{bmatrix} XW \\ YW \\ ZW \end{bmatrix} = -\mathbf{R}^{-1} \cdot \left[\mathbf{T} + \mathbf{R} \cdot \begin{bmatrix} xw \\ yw \\ zw \end{bmatrix} \right] \quad (3.19)$$

Definindo (x, y, z) como a representação de qualquer ponto (xw, yw, zw) no sistema de coordenadas da câmera, em que:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \left[\mathbf{T} + \mathbf{R} \cdot \begin{bmatrix} xw \\ yw \\ zw \end{bmatrix} \right] \quad (3.20)$$

obtêm-se:

$$\begin{bmatrix} XW \\ YW \\ ZW \end{bmatrix} = -\mathbf{R}^{-1} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (3.21)$$

O vetor (XW, YW, ZW) representa a distância de qualquer ponto na placa de calibração para o centro óptico, representado no sistema de coordenadas global [13]. A nova matriz de rotação representada no sistema de coordenadas global é dada por:

$$\mathbf{RW} = -\mathbf{R}^{-1}. \quad (3.22)$$

3.2.2 Calibração da Razão μ

Na seção 2.3.4 foi desenvolvido o modelo RAC que servirá de base para calibrar o valor da razão μ . Como os valores de fx e fy são fixos para um sistema de visão a razão μ poderá ser determinada uma única vez [19]. A razão entre as equações 2.99 e 2.100 é dada por:

$$\frac{X}{Y} = \frac{1}{\mu} \frac{r_1.xw + r_2.yw + r_3.zw + T_x}{r_4.xw + r_5.yw + r_6.zw + T_y} \quad (3.23)$$

em que (X, Y) são as coordenadas dos pontos de calibração em pixels na imagem da placa de calibração. Posicionando a placa de calibração de modo que $zw = 0$, a equação 3.23 se torna:

$$\mu.X.r_4.xw + \mu.X.r_5.yw + \mu.X.T_y - Y.r_1.xw - Y.r_2.yw - Y.T_x = 0 \quad (3.24)$$

Devido a incertezas nos valores das variáveis da equação 3.24 o seu valor não será exatamente igual a zero. Portanto, o valor correto de μ poderá ser encontrado buscando-se o valor de μ que minimiza o resíduo da equação 3.24 para cada imagem da placa de calibração. Com o valor de μ calibrado pode-se recalculer os parâmetros $[r_1, r_2, T_x, r_4, r_5, T_y]$, obtendo dessa forma valores mais precisos.

3.2.3 Calibração do Centro da Imagem

No modelo RAC de calibração, a restrição de alinhamento radial significa que os vetores $\overline{O_i P_d}$ e $\overline{O_i P_u}$ na imagem 2.9 são radialmente alinhados quando o centro da imagem é escolhido corretamente, independentemente do grau de distorção radial da lente [13].

Dessa forma, com um conjunto de i pontos coplanares na placa de calibração que possuem coordenadas $P_i = (xw_i, yw_i, zw_i)$ no sistema de coordenadas global, e $(X_{f,i}, Y_{f,i})$ no sistema de coordenadas da imagem, pode-se fazer a calibração do centro da imagem. A partir das equações 2.99 e 2.100 obtemos:

$$\frac{X_{f,i} - C_x}{1 - k.r^2} - fx \cdot \frac{r1.xw_i + r2.yw_i + r3.zw_i + Tx}{r7.xw_i + r8.yw_i + r9.zw_i + Tz} = 0 \quad (3.25)$$

$$\frac{Y_{f,i} - C_y}{1 - k.r^2} - fy \cdot \frac{r4.xw_i + r5.yw_i + r6.zw_i + Ty}{r7.xw_i + r8.yw_i + r9.zw_i + Tz} = 0 \quad (3.26)$$

em que:

$$r^2 = \mu^2 (X_{f,i} - C_x)^2 + (Y_{f,i} - C_y)^2 \quad (3.27)$$

As equações 3.25 e 3.26 não serão exatamente iguais a zero devido a incertezas dos parâmetros presentes nas equações. Dessa forma, o valor do centro da imagem pode ser escolhido buscando-se os parâmetros de C_x e C_y que minimizam os resíduos gerados pelas equações 3.25 e 3.26, respectivamente, para cada imagem capturada.

Capítulo 4

Resultados

4.1 Introdução

Nesta seção são apresentados os resultados da calibração da câmera, para isso é feita uma análise dos parâmetros intrínsecos e extrínsecos obtidos. Também é feita a análise de precisão da câmera, nesse caso o robô foi movido para diferentes posições e foi feita a comparação entre os deslocamentos apresentados pelo controlador do robô e os apresentados pelo sistema de visão desenvolvido. Por fim são mostrados os resultados obtidos com o modelo estatístico desenvolvido.

4.2 Experimento e Resultado da Calibração da Câmera

Para que se possam fazer medições de posicionamento do robô é necessário que se decida o *setup* do equipamento experimental. Existem duas formas principais para montagem do sistema, na primeira a câmera é fixada em um determinado ponto e a placa de calibração é montada no TCP do robô. Na segunda configuração, chamada de *hand-mounted*, a câmera é colocada no efetuador terminal do robô e a placa de calibração ficará em um ponto fixo. Nesse trabalho foi utilizada a configuração *hand-mounted*, como é mostrada na figura 4.1.

Deve-se levar em consideração que o robô apresenta erros de posicionamento, devido a sua estrutura mecânica. Tais erros influenciam nos resultados obtidos para a calibração da câmera. A tabela 4.1 mostra as incertezas presentes no sistema robótico medidas segundo a norma ISO 9283 para teste de desempenho de robôs manipuladores. Os dados se referem ao cálculo da média e desvio padrão dos erros de posicionamento entre trajetórias programadas e realizadas, segundo a norma citada. Pode-se ver que a precisão do robô é variável e depende da carga de operação.

Carga de operação (kg)	até 0.1	até 1.0	até 3.0	até 8.0
Exatidão de trajetórias (mm)	0.51	0.52	1.00	2.32
Repetitividade de trajetórias (mm)	0.30	0.21	0.14	0.10

Tabela 4.1: Exatidão e repetitividade de trajetórias do robô [2].

Como o robô está movimentando apenas uma câmera com poucas gramas, a exatidão de trajetórias e a repetitividade do robô serão 0.51 mm e 0.3 mm, respectivamente, como mostra a tabela 4.1.

O sistema de coordenadas global xw, yw, zw foi colocado na placa de calibração de modo que $zw = 0$, e a placa foi posicionada com aproximadamente 30° de inclinação. Com a câmera posicionada no TCP do robô é possível movimentá-la em três eixos relativos a base do robô, tais movimentos são necessários para analisar a precisão do sistema de medição.



Figura 4.1: Configuração hand-mounted.

Para aplicar o processo de calibração da câmera descrito na seção 3.2 foram coletadas 30 imagens em pontos aleatórios no volume de trabalho do robô. O volume de trabalho utilizado foi um cubo com dimensão (600mm x 500mm x 350mm), por conta do espaço físico limitado de movimentação do robô. Em pontos fora desse cubo a câmera não possui vista completa da placa de calibração, impedindo a medição de posicionamento.

Como explicitado na seção 3.2, para se medir o posicionamento da câmera em relação à placa, é necessário primeiramente coletar uma imagem da posição de medida desejada e extrair as coordenadas dos pontos de calibração da imagem obtida. Os pontos de calibração da placa estão localizados nas interseções dos quadrados. Para extrair tais pontos foi utilizada uma *toolbox* do software MATLAB¹.

O procedimento para extrair as coordenadas $(X_{f,i}, Y_{f,i})$ dos pontos de calibração da placa é simples quando se utiliza a *toolbox*. Primeiramente é necessário carregar para o software a imagem que se necessita extrair as coordenadas, e logo após é feita a seleção da área da placa que será utilizada na calibração. A figura 4.2 mostra uma imagem da placa e a extração da região de

¹Disponível em: http://www.vision.caltech.edu/bouguetj/calib_doc

interesse para a calibração da câmera.

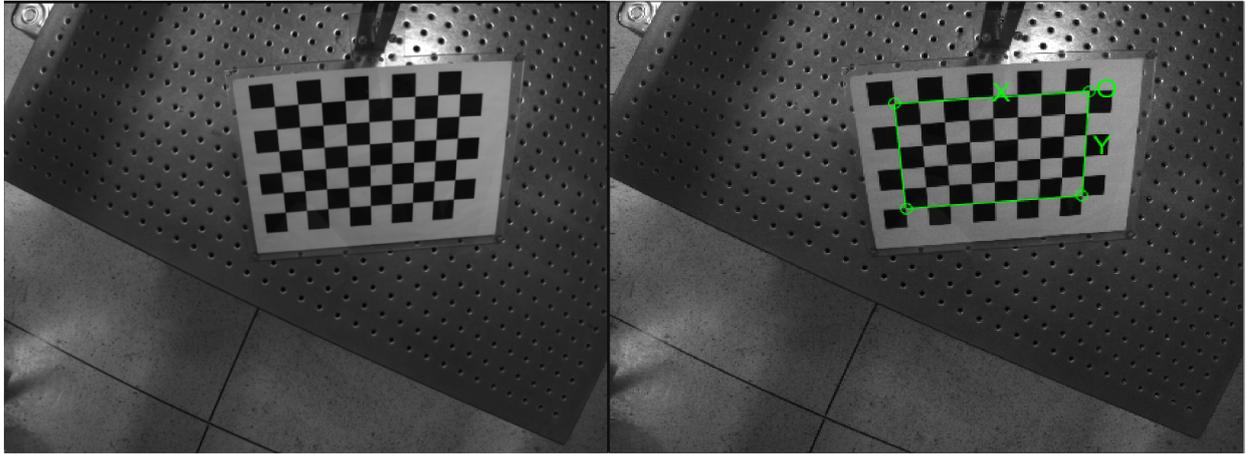


Figura 4.2: Seleção de região da imagem.

Essa mesma região selecionada foi utilizada em todas as imagens analisadas nesse trabalho, com ela foi possível obter 54 diferentes pontos de calibração na imagem, o que proporcionou uma boa precisão para o modelo. Vale lembrar que o número mínimo de pontos para calibrar a câmera é igual a cinco, mas quanto maior o número de pontos melhor será a precisão do modelo [20]. A figura 4.3 mostra o resultado da extração das coordenadas $(X_{f,i}, Y_{f,i})$ feita pelo software, dos i pontos de calibração.

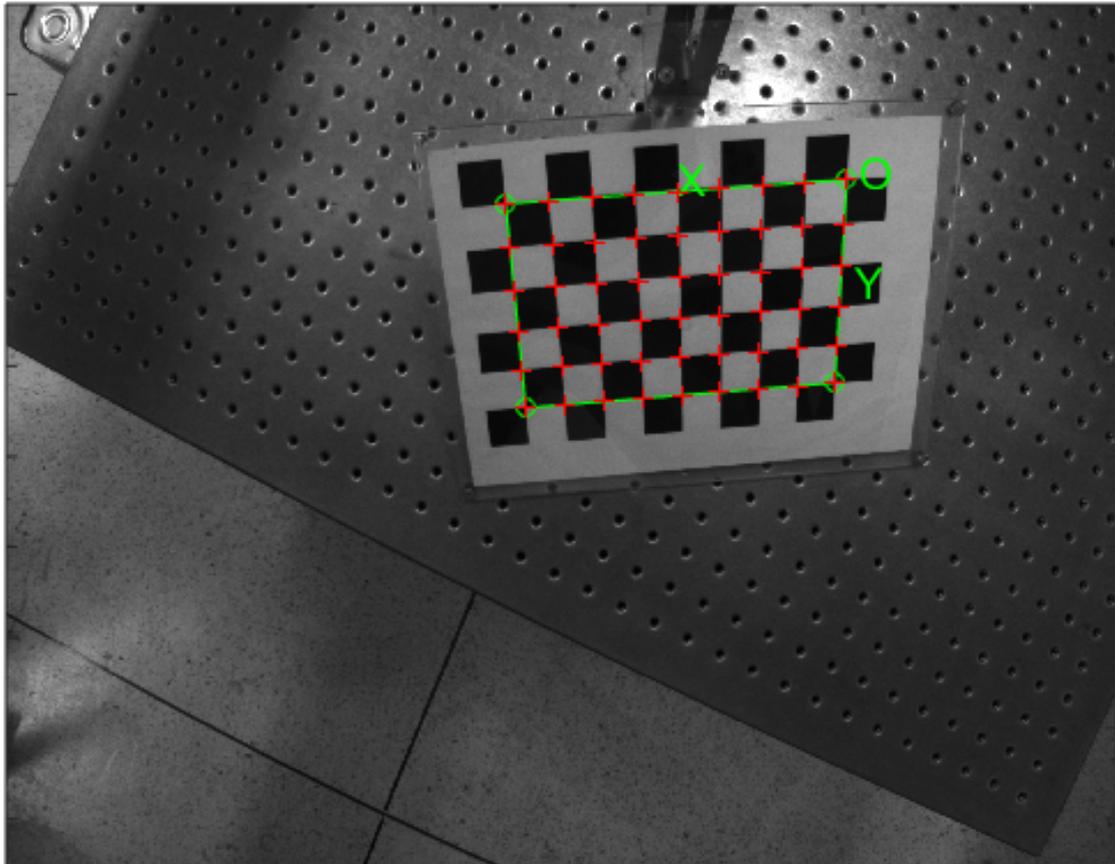


Figura 4.3: Pontos de calibração extraídos da imagem.

4.2.1 Parâmetros Intrínsecos Obtidos

Para que se possa obter os parâmetros intrínsecos da câmera com a aplicação do modelo RAC de calibração, é necessário que se conheça o centro da imagem (C_x, C_y) e a razão μ previamente. Dessa forma, foram escolhidas as coordenadas do centro da imagem como sendo $(1296, 972)$, que representa o centro do sensor CCD em pixels. Mas, como esse centro pode variar por conta das imperfeições existentes na lente da câmera [5], é necessário que se aplique o algoritmo de otimização desse parâmetro descrito na seção 3.2.3.

Também é necessário que se faça uma escolha inicial para a razão μ , a partir da equação 2.88 pode-se fazer essa escolha. Como (s_u, s_v) são os tamanhos efetivos dos pixels em milímetros nas direções horizontal e vertical do sensor CCD respectivamente, e como pode ser visto na tabela 2.1 que os pixels do sensor possuem dimensões $(2.2 \mu\text{m} \times 2.2 \mu\text{m})$, então um valor inicial para a razão pode ser $\mu = 1$. Na tabela 2.1 não são contabilizadas incertezas no valor do tamanho do pixel, logo é necessário otimizar esse parâmetro fazendo uso do algoritmo descrito na seção 3.2.2.

A partir das 30 imagens da placa de calibração coletadas no volume de trabalho do robô foi possível aplicar o algoritmo descrito na seção 3.2 para encontrar os valores otimizados do centro da imagem e da razão μ . A tabela 4.2 mostra os resultados encontrados.

Parâmetro	Valor Inicial	Valor Calibrado	Desvio Padrão
C_x (pixels)	1296	1285,191	2,121
C_y (pixels)	972	1004,557	7,116
μ	1	0,9942	0,042

Tabela 4.2: Calibração do centro da imagem e da razão μ .

Durante a análise dos dados obtidos foi percebido que fatores como a distância da câmera para a placa de calibração e a orientação da câmera, influenciam diretamente nos valores do centro da câmera e da razão μ . Dessa forma, como para cada uma das 30 imagens são obtidos valores diferentes para esses dois parâmetros, o valor escolhido foi a média entre os trinta diferentes.

Para analisar a precisão dos valores mostrados na tabela 4.2, na próxima seção será feita uma análise da precisão dos parâmetros extrínsecos do modelo. Como os parâmetros extrínsecos dependem bastante das coordenadas do centro da câmera e da razão μ , essa análise serve para provar que de fato a escolha foi coerente.

Os parâmetros intrínsecos restantes são a distância focal em pixels f_x e o coeficiente de distorção radial k da lente da câmera. A determinação desses parâmetros também foi feita utilizando o algoritmo descrito na seção 3.2. Para cada uma das 30 imagens da placa de calibração serão calculados valores diferentes para f_x e k , dessa forma o valor escolhido foi a média entre os 30 valores encontrados no processo de calibração. A tabela 4.3 mostra os resultados numéricos encontrados.

Foi verificado no experimento que o posicionamento e orientação da câmera bem como as coordenadas globais da placa de calibração afetam a distância focal, já o coeficiente de distorção radial é pouco afetado por essas razões.

Parâmetro	Valor Calibrado	Desvio Padrão
fx (pixeis)	2058,974	34,961
k	$4,820 \times 10^{-8}$	$2,7 \times 10^{-9}$

Tabela 4.3: Calibração da distância focal e do coeficiente de distorção radial.

Como mostrado na tabela 4.3 a lente da câmera utilizada no experimento possui distância focal fixa de 4,5 mm. Foi visto anteriormente que o fator de escala s_u é o responsável por converter coordenadas da imagem em pixeis para coordenadas em milímetros, sendo assim, tal fator pode ser usado para calcular o valor da distância focal em milímetros f como definido pela equação 2.70. O valor de f será dado por:

$$f = \frac{fx}{s_u} = \frac{2058,974}{454,545} = 4,52974 \text{ mm} \quad (4.1)$$

em que $s_u = 1/0.0022 = 454,545 \text{ mm}^{-1}$. O valor de 0.0022 mm é o tamanho do pixel na horizontal definido na tabela 2.1.

O valor obtido no cálculo em 4.1 foi próximo do valor esperado de 4.5 mm. O valor ideal para o parâmetro fx pode ser calculado pela multiplicação do fator de escala s_u com a distância focal nominal da câmera 4.5 mm. O valor de fx mostrado na tabela 4.3 foi mantido por ser uma boa aproximação.

4.2.2 Parâmetros Extrínsecos Obtidos e Avaliação da Precisão da Câmera

Para analisar os parâmetros extrínsecos e a precisão da câmera, foram utilizadas as mesmas 30 imagens coletadas para se fazer a calibração dos parâmetros intrínsecos da câmera na seção 4.2.1. As avaliações de precisão foram feitas em cada um dos eixos do sistema de coordenadas da câmera $\{x, y, z\}$. Para isso, a câmera foi fixada no efetuador terminal e a placa de calibração foi mantida em uma posição fixa como dito anteriormente e pode ser visto na figura 4.1. Em cada uma das posições desejadas uma imagem foi capturada e as coordenadas do efetuador terminal do robô dadas pelo seu controlador foram salvas. Aqui será mantida a notação para o sistema de coordenadas do robô em relação a sua base $\{X_B, Y_B, Z_B\}$ usada na seção 2.2. A movimentação do robô foi feita utilizando-se o FlexPendant mostrado na figura 4.4.



Figura 4.4: FlexPendant usado para movimentar o robô.

A avaliação da precisão do sistema de medição foi feita a partir da análise dos deslocamentos entregues pelo controlador e pelo sistema de visão. O volume de trabalho foi o mesmo escolhido na seção 4.2. Com o vetor de translação \mathbf{T} calculado para cada imagem pelo modelo RAC, é possível encontrar os deslocamentos do efetuador terminal do robô e avaliar a precisão do sistema de visão. A primeira coluna da tabela 4.4 mostra qual deslocamento em questão está sendo calculado, por exemplo, $1 \rightarrow 2$ representa o deslocamento entre a posição 1 e a posição 2, que será calculado a partir das imagens 1 e 2 capturadas. Todos os dados mostrados na tabela 4.4 estão em milímetros.

Movimento	dX_B	dY_B	dZ_B	dX_C	dY_C	dZ_C	$e(x)$	$e(y)$	$e(z)$
1 → 2	266,6	44,8	175,8	265,931	45,321	174,913	0,669	0,521	0,887
2 → 3	109,2	78,9	132,3	109,352	78,236	133,071	0,152	0,664	0,771
3 → 4	94,7	131,9	236,9	95,002	132,653	237,129	0,302	0,753	0,229
4 → 5	354,3	101,4	84,3	354,265	102,052	85,256	0,035	0,652	0,956
5 → 6	286,2	149,5	78,9	286,111	150,352	79,173	0,089	0,852	0,273
6 → 7	93,5	199,5	109,7	93,356	198,925	110,391	0,144	0,575	0,691
7 → 8	66,7	238,5	170,7	67,336	237,999	171,225	0,636	0,501	0,525
8 → 9	4,4	261,4	248,2	4,93138	260,678	249,099	0,532	0,721	0,899
9 → 10	36,4	66,2	83	37,365	66,632	82,933	0,965	0,432	0,067
10 → 11	127,7	64,8	135,8	126,989	64,789	134,919	0,711	0,011	0,881
11 → 12	288,6	261	212,6	287,921	261,965	213,003	0,679	0,965	0,403
12 → 13	404,9	251,1	163,5	404,263	252,036	164,277	0,637	0,936	0,777
13 → 14	257,5	55,5	185,5	256,963	54,958	184,916	0,537	0,542	0,584
14 → 15	23,2	73,3	138,4	22,963	72,6555	137,298	0,237	0,644	1,102
15 → 16	252,8	136,4	210,8	251,986	137,965	211,521	0,814	1,565	0,721
16 → 17	395,5	198,3	106,6	394,993	197,256	107,616	0,507	1,044	1,016
17 → 18	89,2	220,5	104,8	90,365	219,852	105,112	1,165	0,648	0,312
18 → 19	341,9	22,9	211,8	342,753	23,037	211,818	0,853	0,137	0,017
19 → 20	445,7	0,4	150,3	444,554	1,319	151,356	1,146	0,919	1,055
20 → 21	390,7	201	117,4	388,985	201,152	118,398	1,715	0,152	0,998
21 → 22	277,3	240,5	147,3	277,296	241,538	146,726	0,004	1,038	0,574
22 → 23	249,1	188,7	82,6	249,562	189,462	83,124	0,462	0,762	0,524
23 → 24	81,8	109,9	97,2	81,3016	108,937	96,751	0,498	0,963	0,449
24 → 25	353	3,6	178,8	353,669	3,237	179,452	0,669	0,363	0,652
25 → 26	1	106,8	46,1	1,036	106,562	45,752	0,036	0,237	0,348
26 → 27	68,1	120	216,7	69,002	120,605	217,223	0,902	0,605	0,523
27 → 28	102,2	101,2	240,2	103,035	102,017	239,765	0,835	0,817	0,435
28 → 29	110	123,4	155,1	110,365	122,752	154,732	0,365	0,648	0,368
29 → 30	82,5	76,7	84,4	82,551	77,952	84,828	0,051	1,252	0,428

Tabela 4.4: Deslocamentos dados pelo controlador do robô, pelo sistema de visão e erro entre eles, todos os dados estão em milímetros

Na tabela 4.4 estão os deslocamentos encontrados, os parâmetros (dX_B, dY_B, dZ_B) representam

os deslocamentos entregue pelo controlador do robô nos eixos X_B , Y_B e Z_B , respectivamente. Os parâmetros (dX_C, dY_C, dZ_C) representam os deslocamentos entregue pelo sistema de visão computacional ao longo dos eixos x , y e z da câmera, respectivamente. E por fim, os valores $(e(x), e(y), e(z))$ são os erros entre os valores dos deslocamentos dados pelo controlador e pelo sistema de visão em cada um dos eixos. Os erros foram calculados a partir do módulo da diferença entre o valor de deslocamento dado pelo controlador e o dado pelo sistema de visão:

$$e(x) = |dX_B - dX_C| \quad (4.2)$$

$$e(y) = |dY_B - dY_C| \quad (4.3)$$

$$e(z) = |dZ_B - dZ_C| \quad (4.4)$$

A magnitude do erro de posicionamento pode ser calculada da seguinte forma para cada ponto:

$$|e| = \sqrt{e(x)^2 + e(y)^2 + e(z)^2} \quad (4.5)$$

A média e o desvio padrão dos erros podem ser vistos na tabela 4.5

Erro	Valor médio	Desvio Padrão
$e(x)(mm)$	0,564	0,402
$e(y)(mm)$	0,687	0,338
$e(z)(mm)$	0,899	1,565

Tabela 4.5: Média e desvio padrão dos erros.

Na tabela 4.5 pode-se ver que o valor médio dos três erros ficaram abaixo de 1 milímetro. Na prática é difícil medir a distância entre os pontos de calibração sem uma máquina de medição de alta precisão. Foi observado que pequenas variações nas distâncias entre os pontos de calibração no sistema de coordenadas global, podem causar grandes variações nos parâmetros extrínsecos da câmera, outros fatores como centro da câmera, razão μ também causam grande influência nos parâmetros extrínsecos. Dessa forma, é necessário reduzir ao máximo as fontes de erros no modelo RAC para que se possam reduzir os erros mostrados na tabela 4.5. Pôde-se também notar que o número de pontos na placa de calibração possui pouca influência sobre os parâmetros extrínsecos da câmera no modelo RAC.

Em geral os erros encontrados foram pequenos, como pode ser visto na tabela 4.4. Dessa forma, o sistema de visão pode ser usado para o objetivo desse trabalho, mas é necessário avaliar a precisão do robô caso necessário usar tal modelo para outras aplicações.

4.3 Experimento e Resultado do Estimador Estatístico

A avaliação da precisão do sistema de visão computacional feita na seção 4.2, mostrou que o sistema possui boa precisão e que pode ser usado na medição de posicionamento do robô, de forma

a auxiliar na avaliação do algoritmo estatístico descrito na seção 2.4. Para avaliar a precisão do modelo estatístico desenvolvido, foram coletados dados de posicionamento do robô em 120 pontos diferentes, utilizando-se o mesmo volume de trabalho do experimento de avaliação da precisão da câmera, um cubo com dimensão (60mm x 500mm x 300mm).

Como descrito na seção 2.4, o algoritmo estatístico tem como principal função estimar erros de posicionamentos para diferentes posições, com base na similaridade dos erros de posicionamento. Dessa forma, para cada um dos 120 amostrados foram medidos os erros de posicionamentos reais, dados pelo sistema de medição, e então foram feitas estimativas de erros para cada um desses pontos.

As figuras 4.5, 4.6 e 4.7, nos mostram os resultados encontrados dos erros de posicionamentos nos eixos X_B , Y_B e Z_B , respectivamente. Nessas figuras, para cada uma das 120 posições é mostrado o valor de erro medido e o valor do erro estimado.

Como descrito na seção 2.4 pelas equações 2.119, 2.120 e 2.121, a correção dos erros de posicionamento do robô, é feita compensando-se nas coordenadas do robô os valores de erros estimados via software.

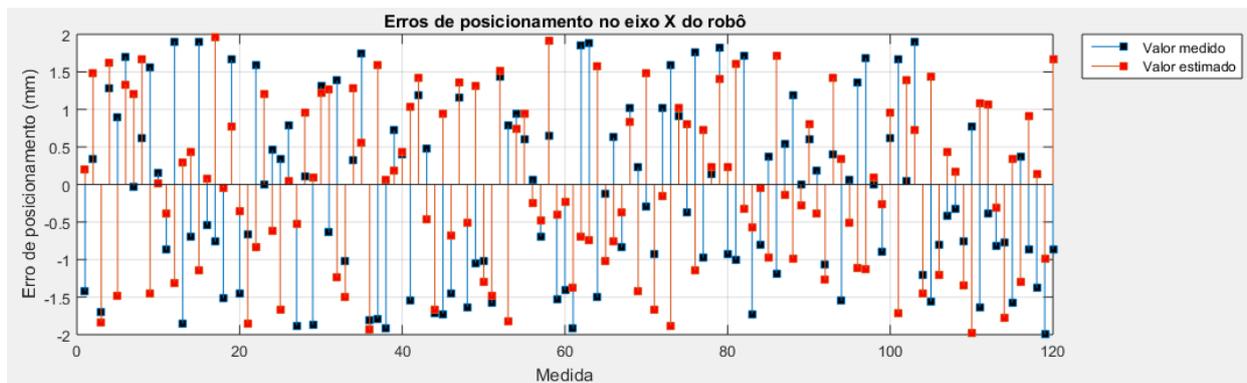


Figura 4.5: Erros de posicionamento no eixo X robô.

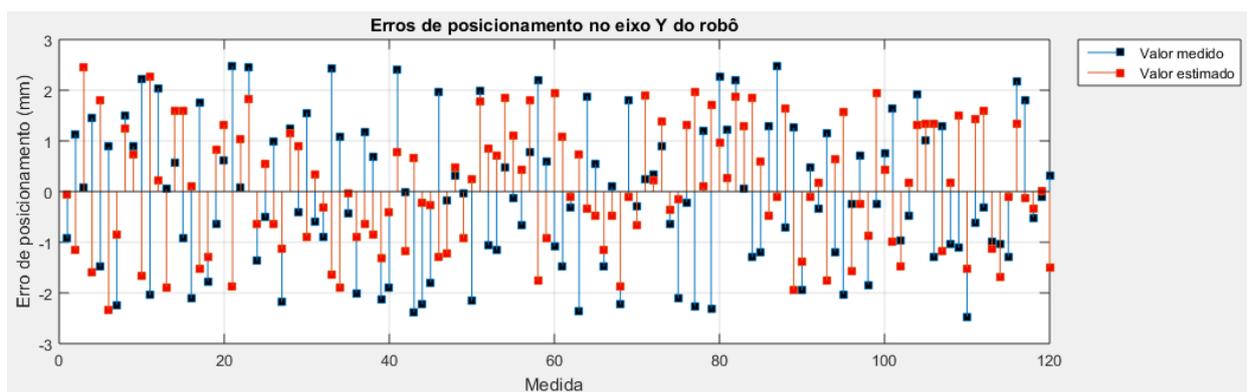


Figura 4.6: Erros de posicionamento no eixo Y robô.

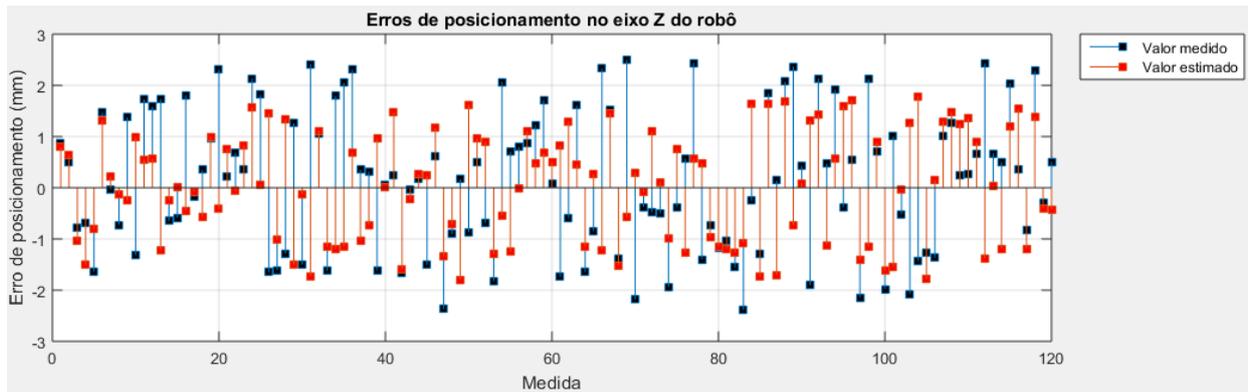


Figura 4.7: Erros de posicionamento no eixo Z robô.

Com um estimador estatístico de alta precisão, os valores dos erros estimados devem ser iguais aos valores dos erros medidos. A partir das figuras 4.5, 4.6 e 4.7, vemos o estimador não apresentou bem o comportamento esperado. Mas pode-se ver também que os valores estimados ficaram próximos dos valores medidos para a maioria dos pontos.

A figura 4.5 nos mostra que tanto os erros estimados quanto os erros medidos, se concentraram no intervalo de $[-2\text{mm}, 2\text{mm}]$, o mesmo fato ocorreu para os outros dois eixos como mostram as figuras 4.6 e 4.7, em que os erros medidos e estimados se concentraram no intervalo de $[-3\text{mm}, 3\text{mm}]$.

A principal influência na precisão do modelo estatístico é a quantidade de pontos coletados no volume de trabalho do robô, quando maior o número de pontos maior será a precisão dos valores estimados. Alguns trabalhos como [5] e [20], mostraram que a quantidade ideal de pontos para que se possa obter um bom modelo estatístico, são aproximadamente 1000 pontos no volume de trabalho do robô escolhido.

Também devemos considerar que nos erros obtidos é difícil saber se os erros estão de fato vindo da imprecisão do posicionamento do robô, ou do sistema de visão computacional. Dessa forma, quanto melhor a precisão do modelo estatístico e do sistema de medição baseado em visão, mais precisos serão os resultados.

Capítulo 5

Conclusões

O sistema de medição usado nesse trabalho foi o modelo RAC, após a implementação e avaliação dos resultados tal sistema se mostrou preciso e é viável utilizá-lo para a calibração do robô. Mas, os parâmetros extrínsecos da câmera dados pelo modelo são muito afetados por pequenas variações na distância entre os pontos da placa de calibração, nas coordenadas do centro da imagens e na razão μ . Dessa forma é necessário que esse valores sejam otimizados para que não afetem os resultados da calibração do robô, outras formas podem ser usadas para calibrar os fatores de escala. Para que o sistema de calibração seja mais preciso é necessário escolher um volume de trabalho maior do robô, isso irá fazer com que o modelo estatístico de estimativa de erros seja mais correto.

Quanto ao modelo estatístico desenvolvido, foi visto que a quantidade de pontos amostrados no volume de trabalho do robô possuem grande influência no resultado do valor estimado encontrado. Dessa forma, para que se possa obter maior precisão na estimativa de valores para os erros de posicionamento, é necessário que se colete em torno de 1000 pontos. Com um sistema mais preciso, é possível aplicar as compensações de coordenadas no controlador do robô, finalizando o processo de calibração.

Com base nos objetivos inicialmente sugeridos para esse trabalho, conclui-se que a maioria dos pontos propostos foram contemplados, mas com necessidade de melhorias.

Por fim, sugere-se para trabalhos futuros os seguintes pontos:

- Otimizar os valores dos parâmetros intrínsecos da câmera, tais como: centro da imagem e razão μ .
- Medir de forma mais precisa a distância entre os pontos de calibração da placa de calibração, de forma a reduzir os erros do sistema de visão.
- Coletar maior número de amostras no volume de trabalho do robô, de forma a aumentar a precisão do estimador estatístico.
- Aplicar as compensações de coordenadas no controlador do robô, para que o processo de calibração do robô possa ser feito de forma offline.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] MOTTA, J. M. S. T. da; GINANI, L. S. Theoretical and practical aspects of robot calibration with experimental verification. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, January 2011.
- [2] ABB ROBOTICS. *Product specification IRB 360*. Disponível em: <<https://new.abb.com/products/robotics/industrial-robots/irb-360>>.
- [3] WILLIAMS, R. L. The delta parallel robots: kinematics solutions. *Internet Publication*, <www.ohio.edu/people/williar4/html/pdf/DeltaKin.pdf>, Ohio University, January 2016.
- [4] SZELISKI, R. *Computer Vision: Algorithms and Applications*. First edition. London: Springer, 2010.
- [5] TSAI, R. Y. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE JOURNAL OF ROBOTICS AND AUTOMATION*, August 1987.
- [6] WEI, T. e. a. Positional error similarity for error compensation of industrial robots. *Chinese Journal of Aeronautics.*, 2016.
- [7] CORKE, P. *Robotics, Vision and Control: Fundamental Algorithms In MATLAB® Second, Completely Revised*. Berlin, Heidelberg - Germany: Springer, 2017.
- [8] CRAIG, J. J. *Introduction to robotics: mechanics and control*. Upper Saddle River, NJ, USA: Pearson/Prentice Hall, 2005.
- [9] BERNHARDT R.; ALBRIGHT, S. *Robot Calibration*. Cambridge, England: Chapman & Hall, 1993.
- [10] HENRIQUES R.V.B.; DUTRA, M. R. J. C. L. W. C. A. *ROBÓTICA INDUSTRIAL: Aplicação na Indústria de Manufatura e de Processos*. 1. ed. São Paulo, Brasil: Edgard Blücher, 2002.
- [11] MA, Y. et al. *An invitation to 3-D vision - From images to models*. New York: Berkeley, 2001.
- [12] HARTLEY, R.; ZISSERMAN, A. *Multiple view geometry in computer vision*. Second edition. Cambridge: Cambridge, 2004.

- [13] MOTTA, J. M. S. T. da. *Optimized Robot Calibration Using a Vision-Based Measurement System With a Single Camera*. Tese (Doutorado) — Cranfield University, England, 1999.
- [14] WENG J.; COHEN, P.; HERNIOU, M. Camera calibration with distortion models and accuracy evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, October 1992.
- [15] WEI, T. e. a. Determination of optimal samples for robot calibration based on error similarity. *Chinese Journal of Aeronautics.*, 2015.
- [16] GREGO, C. R. e. a. Agricultura de precisão: resultados de um novo olhar. *Cap 5: Geoestatística aplicada a agricultura de precisão*. Embrapa, Brasília, 2014.
- [17] FÉLIX V.B.; GONZATTO, O. R. D. H. M. Estimadores de semivariância: Uma revisão. *Departamento de estatística, Universidade estadual de Maringá, PR*, 2016.
- [18] SCHLATHER, T. G. Z. S. M. Analogies and correspondences between variograms and covariance functions. *NRCSE, Technical Report Series*, 2000.
- [19] GINANI, L. S. *Desenvolvimento de Um Sistema de Digitalização Tridimensional de Superfícies Baseado em Visão Computacional Com Varredura a Laser Para Uso em Metrologia Dimensional*. Dissertação (Mestrado) — Universidade de Brasília, Brasília, 2008.
- [20] TSAI, R. Y.; LENZ, R. K. Techniques for calibration of the scale factor and image center for high accuracy 3-d machine vision metrology. *IEEE transactions on pattern analysis and machine intelligence*, September 1988.

APÊNDICES

I. IMPLEMENTAÇÃO DO MODELO RAC NO MATLAB

Aqui está contido o código de implementação do modelo de câmera com restrição de alinhamento radial utilizado nesse trabalho. As linhas iniciadas com traços devem ser comentadas no software MATLAB. As coordenadas dos pontos de calibração da imagem podem ser obtidas com o auxílio de alguma *toolbox* de processamento de imagens do MATLAB.

```
1
2 clc
3
4 clear all
5
6 close all
7
8 — ESTAGIOS DE CALIBRACAO
9
10 — NOTACAO UTILIZADA
11
12 — [Tx Ty Tz]: Coordenadas globais da placa de calibracao
13
14 — [Xi Yi]: Coordenadas da imagem
15
16 — [Cx Cy]: Centro da imagem
17
18 — [Xfi Yfi]: Coordenadas em pixels na imagem do f-esimo
19 ponto de calibracao
20
21 — [xw yw zw]: Coordenadas globais
22
23 — mi: Razao dos fatores de escala
24
25 — Estagio 1: Calculo da matriz de rotacao R e dos parametros Tx e Ty
26
27 — Valores que devem ser conhecidos a priori:
28
29 — Centro da imagem (Declarar aqui o centro da imagem)
30
31 — Cx = Centro da imagem em x
32
```

```

33      — Cy = Centro da imagem em y
34
35
36 — 1. Calculo das coordenadas da imagem centrada (Xi,Yi), com n pontos
de calibracao.
37
38      — Definir aqui as coordenadas na imagem dos pontos de calibracao (
      Xfi(n), Yfi(n)
39
40      — Calculo da imagem centrada
41      for n=1:81
42
43      Xi(n) = Xfi(n) - Cx;
44
45      Yi(n) = Yfi(n) - Cy;
46
47      end
48
49 — 2. Calculo dos parametros intermediarios v = [v1 v2 v3 v4 v5];
50
51      for n=1:1:54
52
53      — Calculo das matrizes K e H, do sistema Kv=H.
54
55      Xi(n) = Xfi(n) - Cx;
56
57      Yi(n) = -Yfi(n) + Cy;
58
59      K(n,:) = [xwi(n)*Yi(n), ywi(n)*Yi(n), Yi(n), -xwi(n)*mi*Xi(n), -ywi
      (n)*mi*Xi(n)];
60
61      H(n,1) = mi*Xi(n);
62
63      end
64
65      — Calculo das incognitas pelo metodo SVD (Singular Values
      Decomposition).
66
67      [U,S,V] = svd(K, 'econ');
68
69      v = V*(S^-1)*(U')*H;
70

```

```

71
72 — 3. Calculo de R, Tx e TY
73
74 — Passo 1: Definicao da matriz C 2x2 como segue (rotacao em x e y).
75
76 C = [v(1) v(2);v(4) v(5)];
77
78 Sr = v(1)^2 + v(2)^2 + v(4)^2 + v(5)^2;
79
80 m = 0;
81
82
83 if (C(1)==0&&C(2)==0)|| (C(3)==0&&C(4)==0)|| (C(1)==0&&C(3)==0)|| (C(2)
    ==0&&C(4)==0)
84
85 — Caso a matriz C possua linhas ou colunas nulas.
86
87     for j=1:1:4
88
89         — Pega os elementos nao nulos para calcular Ty.
90
91         if (C(j)~=0)==true
92
93             m = m + 1;
94
95             L(m) = C(j);
96
97         end
98
99     end
100
101     Ty = sqrt(1/(L(1)^2+L(2)^2));
102
103 — Calculo de Ty
104
105 else
106
107     Ty = sqrt((Sr - sqrt( Sr^2 - 4*(v(1)*v(5)-v(4)*v(2))^2))/(2*(v(1)
        *v(5)-v(4)*v(2))^2));
108
109 end

```

```

109  — Passo 2: Determinacao do sinal de Ty
110
111  — Primeiramente considera-se que o sinal de Ty e positivo.
112
113  r1 = v(1)*Ty;
114
115  r2 = v(2)*Ty;
116
117  r4 = v(4)*Ty;
118
119  r5 = v(5)*Ty;
120
121  Tx = v(3)*Ty;
122
123  — Selecionando-se um ponto de calibracao arbitrario , calculamos:
124
125  x = r1*xwi(4) + r2*ywi(4) + Tx;
126
127  y = r4*xwi(4) + r5*ywi(4) + Ty;
128
129  — Inversao do sinal , caso os sinais de x e Xi ou y e Yi sejam
    diferentes
130
131  if (sign(x)~=sign(Xi(4))) || (sign(y)~=sign(Yi(4)))
132
133     Ty = Ty*(-1);
134
135     r1 = r1*(-1);
136
137     r2 = r2*(-1);
138
139     r4 = r4*(-1);
140
141     r5 = r5*(-1);
142
143     Tx = Tx*(-1);
144
145  end
146
147  — Passo 3: Calculo da matriz R
148
149  — Primeira solucao possivel para a matriz R:

```

```

150
151   r3 = sqrt(1-(r1)^2-(r2)^2);
152
153   r6 = -sign(r1*r4+r2*r5)*sqrt(1-(r4)^2-(r5)^2);
154
155   cross\_prod = cross([r1;r2;r3],[r4;r5;r6]);
156
157   r7 = cross\_prod(1);
158
159   r8 = cross\_prod(2);
160
161   r9 = cross\_prod(3);
162
163   R = [r1 r2 r3; r4 r5 r6; r7 r8 r9];
164
165 — Estagio 2: Calculos dos parametros k, fx e Tz
166
167   k = 1;
168
169   for n=1:1:54
170
171       xi(n) = r1*xwi(n) + r2*ywi(n) + Tx;
172
173       yi(n) = r4*xwi(n) + r5*ywi(n) + Ty;
174
175       wi(n) = r7*xwi(n) + r8*ywi(n);
176
177       r(n) = sqrt((mi^2)*(Xi(n)^2) + Yi(n)^2);
178
179 — Calculo das matrizes do sistema I*sol = J
180
181       if n~=1
182
183           k = k+2;
184
185       end
186
187       I(k,:) = [-Xi(n), xi(n), -xi(n)*(r(n)^2)];
188
189       I(k+1,:) = [-Yi(n), mi*yi(n), -mi*yi(n)*(r(n)^2)];
190
191       J(k,1) = [Xi(n)*wi(n)];

```

```

192
193     J(k+1,1) = [Yi(n)*wi(n) ];
194
195 end
196
197 — Calculo das incognitas pelo metodo SVD (Singular Values
    Decomposition).
198
199 [U,S,V] = svd(I, 'econ ');
200
201 sol = V*(S\^-1)*(U')*J;
202
203 Tz = sol(1);
204
205 fx = sol(2);
206
207 fy = mi*fx;
208
209 distorcao\_rad = sol(3)/sol(2);
210
211 — Nova matriz R, caso fx seja negativo.
212
213 if fx < 0
214
215     r3 = -sqrt(1-(r1)^2-(r2)^2);
216
217     r6 = sign(r1*r4+r2*r5)*sqrt(1-(r4)^2-(r5)^2);
218
219     cross\_prod = cross([r1;r2;r3],[r4;r5;r6]);
220
221     r7 = cross\_prod(1);
222
223     r8 = cross\_prod(2);
224
225     r9 = cross\_prod(3);
226
227     R = [r1 r2 r3; r4 r5 r6; r7 r8 r9];
228
229     k = 1;
230
231     for n=1:1:54
232

```

```

233         xi(n) = r1*xwi(n) + r2*ywi(n) + Tx;
234
235         yi(n) = r4*xwi(n) + r5*ywi(n) + Ty;
236
237         wi(n) = r7*xwi(n) + r8*ywi(n);
238
239         r(n) = sqrt((mi^2)*(Xi(n)^2) + Yi(n)^2);
240
241 — Calculo das matrizes do sistema I*sol = J.
242
243         if n~=1
244
245             k = k+2;
246
247         end
248
249         I(k,:) = [-Xi(n), xi(n), -xi(n)*(r(n)^2)];
250
251         I(k+1,:) = [-Yi(n), mi*yi(n), -mi*yi(n)*(r(n)^2)];
252
253         J(k,1) = [Xi(n)*wi(n)];
254
255         J(k+1,1) = [Yi(n)*wi(n)];
256
257     end
258
259 — Calculo das incognitas pelo metodo SVD (Singular Values
260     Decomposition).
261
262     [U,S,V] = svd(I, 'econ');
263
264     sol = V*(S^-1)*(U')*J;
265
266     Tz = sol(1);
267
268     fx = sol(2);
269
270     fy = mi*fx;
271
272     dist\_rad = sol(3)/sol(2);
273
274 end

```