



**Miguel Angelo Gaspar Pinto**

**Posicionamento e Calibração de  
um Manipulador Robótico Submarino  
com Uso de Visão Computacional**

**Dissertação de Mestrado**

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-graduação em Engenharia Elétrica do Departamento de Engenharia Elétrica da PUC-Rio.

Orientador: Prof. Ricardo Tanscheit  
Co-Orientador: Prof. Marco Antonio Meggiolaro

Rio de Janeiro  
Agosto de 2006



**Miguel Angelo Gaspar Pinto**

**Posicionamento e Calibração de um  
Manipulador Robótico Submarino com  
Uso de Visão Computacional**

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-Graduação em Engenharia Elétrica do Departamento de Engenharia Elétrica do Centro Técnico Científico da PUC-Rio. Aprovada pela Comissão Examinadora abaixo assinada.

**Prof. Ricardo Tanscheit**  
**Orientador**

Departamento de Engenharia Elétrica – PUC-Rio

**Prof. Marco Antonio Meggiolaro**  
**Co-Orientador**

Departamento de Engenharia Mecânica – PUC-Rio

**Prof. Raul Queiroz Feitosa**

Departamento de Engenharia Elétrica – PUC-Rio

**Prof. Pedro Magalhães Guimarães Ferreira**

Departamento de Engenharia Elétrica – PUC-Rio

**Prof. José Franco Machado do Amaral**  
UERJ

**Prof. José Eugenio Leal**  
Coordenador Setorial do Centro  
Técnico Científico

Rio de Janeiro, 31 de agosto de 2006

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, do autor e do orientador.

## Miguel Angelo Gaspar Pinto

Graduou-se em Engenharia de Controle e Automação (Pontifícia Universidade Católica) em 2004. Trabalhou por meio ano na área de controle antes de começar mestrado na área de Processamento de Sinais e Controle na Pós-Graduação da PUC-Rio. Suas áreas de interesse abrangem robótica, controle de sistemas, visão computacional e inteligência artificial.

### Ficha Catalográfica

Pinto, Miguel Angelo Gaspar

Posicionamento e calibração de um manipulador robótico submarino com uso de visão computacional / Miguel Ângelo Gaspar Pinto ; orientador: Ricardo Tanscheit ; co-orientador: Marco Antonio Meggiolaro. – Rio de Janeiro : PUC, Departamento de Engenharia Elétrica, 2006.

106 f. : il. ; 30 cm

Dissertação (mestrado) – Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Elétrica.

Inclui referências bibliográficas.

1. Engenharia elétrica – Teses. 2. Robótica. 3. Calibragem. 4. Visão computacional. 5. Controle. 6. Inteligência artificial. 7. Reconhecimento de padrões. 8. Rastreamento. I. Meggiolaro, Marco Antonio. II. Tanscheit, Ricardo. III. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Elétrica. VI. Título.

CDD: 621.3

À minha família por todo o apoio.

## Agradecimentos

- À CAPES, pelo apoio financeiro;
- Ao orientador Marco Antônio Meggiolaro, pela oportunidade;
- Ao orientador Ricardo Tansheit, pelo apoio;
- Ao professor Raul Feitosa, pela parte de visão computacional;
- À PUC-Rio pelo seu excelente corpo acadêmico que me deu as bases para este estudo;
- À minha família, que próxima ou distante, me incentivou nessa jornada;

## Resumo

Pinto, Miguel A.G; Tanscheit, Ricardo (Orientador). **Posicionamento e Calibração de um Manipulador Robótico Submarino com Uso de Visão Computacional**. Rio de Janeiro, 2006. 106p. Dissertação de Mestrado – Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

Muitos dos robôs industriais utilizados atualmente seguem uma programação baseada em rastreamento de trajetórias. O robô é guiado por um operador humano para localizações fixas onde ele atuará. Esses movimentos são, então, gravados na linguagem de programação residente no controlador do robô, de forma que este seja capaz de repetir as tarefas. Este método pode ser suficiente para, por exemplo, movimentar objetos entre locais fixos. Como o robô está treinado para se movimentar em posições fixas, todas as partes do manipulador, bem como todos os objetos que serão manipulados devem estar em posições bem definidas, ou uma nova programação deverá ser feita. Outra metodologia é a de teleoperação, na qual a movimentação de sistemas robóticos é executada em modo manual, no qual o operador trabalha em uma arquitetura mestre-escravo controlando direta e continuamente as posições do manipulador. Para essas tarefas é necessário apenas que o sistema possua um alto grau de repetibilidade, uma vez que quaisquer erros absolutos de posicionamento são visualmente compensados pelo operador humano. Porém em certas aplicações robóticas essas técnicas de programação de manipuladores são impraticáveis ou insatisfatórias. A primeira vem a ser impraticável no caso de alta variabilidade do ambiente onde a tarefa está sendo feita. O segundo método atribui ao robô uma precisão absoluta baixa, devido a própria deficiência da percepção humana. Este trabalho segue pelas tendências modernas de automação, as quais vêm colocando uma crescente ênfase em robôs guiados por sensores e programação off-line, automatizando total ou parcialmente muitas das tarefas a serem executadas. Sensores, como câmeras ligadas a um sistema de visão computacional, detectam diferenças entre a posição real do manipulador e a posição desejada. Estas diferenças são então enviadas para os controladores, para que estes corrijam a trajetória pré-programada. Os comandos de movimento do manipulador são programados off-line por um sistema de CAD, sem a necessidade de ativar o robô, permitindo maior velocidade em sua validação e na resolução de problemas. Apresentam-se neste trabalho metodologias e técnicas para o posicionamento do manipulador utilizando-se, para tanto, câmeras em sua extremidade. Uma vez posicionado o manipulador em relação ao espaço de coordenadas do mundo, é possível deslocá-lo com segurança e precisão em sua área de trabalho, o que é imprescindível para automatização de tarefas complexas. O trabalho está concentrado nas aplicações de técnicas de visão computacional à calibração de manipuladores. Como estudo de caso utiliza-se uma situação real, de um manipulador submarino de seis graus de liberdade, para intervenções submarinas em plataformas de petróleo. Abordam-se a calibração de câmeras, reconhecimento de padrões, correlação de padrões em imagens distintas, estereoscopia, cinemática direta e inversa de manipuladores e a união de todas estas técnicas para o posicionamento do manipulador em sua área de trabalho.

## Palavras-chave

Robótica; Calibragem; Visão Computacional; Controle; Inteligência Artificial; Reconhecimento de Padrões; Rastreamento.

## Abstract

Pinto, Miguel A.G; Tanscheit, R. (Advisor). **Positioning and Calibration of a Underwater Robotic Manipulator with Use of Computacional Vision**. Rio de Janeiro, 2006. 106p. MSc Dissertation – Departamento de Engenharia Elétrica, Pontificia Universidade Católica do Rio de Janeiro.

Many of today's industrial robots are still programmed to follow trajectories. The robot is guided by a human operator to the desired fixed application locations. These motions are recorded and are later edited, within the robotic language residing in the robot controller, and played back, for the robot to be able to repetitively perform its task. This methodology is enough to move objects between fixed locations. As the robot is trained to move within fixed positions, all manipulator parts, as well as all the objects which will be manipulated need to be in well defined positions, otherwise another program is needed. Another methodology would be teleoperation, where the robotic system's movements are executed in manual mode, having the operator working in a master-slave architecture, controlling direct and continuously the positions of the robot. For these tasks it is needed only for the system to have enough repeatability, once any absolute positioning errors can be visually compensated by the human operator. On the other side, in certain robotic applications, both programming techniques are either not practical or inefficient. The first, where the human operator teaches the trajectories to the robot, is not possible when there is high variance in the environment where the task is being performed. The second method, the teleoperation, has low absolute accuracy, due the deficiencies of human perception. This project follows modern tendencies of automation, which give increasing emphasis on robots guided by sensors and off-line programming, partially or completely automating many of the tasks to be executed. Sensors such as cameras eload to a system of computational vision detect differences between the actual and desired position of the manipulator. This information is sent to controllers to correct the pre-programated trajectory. The manipulator movement commands are programmed off-line by a CAD system, without need even to turn on the robot, allowing for greatest speed on its validation, as well as problem solving. This work presents methodologies and techniques which allow the precise positioning of the manipulator using cameras in its end-effector. Once it is positioned in relation with the world frame, it is possible to move the manipulator with safety and precision its work area, as is needed for automation of complex tasks. This work is focused on computational vision techniques applied for manipulator calibration. It is based on a real case of a subsea manipulator of six degrees of freedom, used for underwater interventions in oil exploring platforms. The subjects treated in this work include camera calibration, pattern recognition, position tracking, stereopsis, direct and inverse manipulator kinematics and the union of all techniques for manipulator positioning in the work area.

## Keywords

Robotics; Calibration; Computacional Vision; Control; Artificial Intelligence; Pattern Recognition; Tracking.

# Sumário

1. Introdução	14
1.1. Motivação	14
1.2. Objetivos do Trabalho	15
1.3. Descrição do Trabalho	15
1.4. Organização da Dissertação	20
2. Modelagem Cinemática para Calibração de Manipuladores	22
2.1. Introdução	22
2.2. Conceitos Básicos de Cinemática	23
2.3. Convenção de Modelagem de Denavit-Hartenberg	25
2.4. Método Clássico de Calibração de Manipuladores	29
2.5. Eliminação de Erros Redundantes	34
2.6. Representação Física dos Erros Redundantes	36
2.7. Medidas Parciais da Posição da Extremidade	37
2.8. Cinemática Inversa	38
2.8.1. Solubilidade	39
3. Visão Computacional	41
3.1. Introdução	41
3.2. Modelagem da Câmera	42
3.2.1. Modelo Pinhole	42
3.2.2. Parâmetros Intrínsecos	43
3.2.3. Parâmetros Extrínsecos	45
3.3. Calibração de Câmera	47
3.4. Emparelhamento de Pontos	49
3.4.1. Detecção de Extremos no Espaço de Escalas	50
3.4.2. Localização Precisa de Pontos-Chaves	53
3.4.3. Atribuição de Orientação	54
3.4.4. Descritor Local da Imagem	55
3.4.5. Invariância à Luminosidade	57
3.4.6. Relacionamento de Pontos-Chaves em Imagens Distintas	57
3.5. Reconstrução	58
4. Aplicação ao Manipulador TA-40	60
4.1. Introdução	60
4.2. Descrição do Manipulador	61
4.3. Cinemática do Manipulador	62
4.3.1. Juntas 1 e 2	62
4.3.2. Juntas 2 e 3	63
4.3.3. Juntas 3 e 4	64
4.3.4. Juntas 4 e 5	64
4.3.5. Juntas 5 e 6	66
4.3.6. Juntas 6	66
4.3.7. Parâmetros de Denavit-Hartenberg	67
4.4. Calibragem do Manipulador	68

4.5. Cinemática Inversa	70
4.6. Visão Computacional	74
4.7. Câmera	75
4.8. Correlação de Pontos	75
4.9. Calibragem da Base	78
5. Resultados	80
5.1. Introdução	80
5.2. Simulação Cinemática	80
5.2.1. Cinemática Direta-Inversa	80
5.2.2. Jacobiana de Identificação	81
5.3. Simulações em Visão	83
5.3.1. Triangulação em Distâncias	83
5.3.2. Triangulação em Função da Resolução	85
5.3.3. Simulação de Calibração da Base	86
6. Conclusões e Sugestões	89
6.1. Contribuições do Trabalho	89
6.2. Sugestões para Trabalhos Futuros	90
7. Referências Bibliográficas	92
8. Apêndice A	94
8.1. Posição da Extremidade	94
8.2. Jacobiana de Identificação	95
9. Apêndice B	104

## Lista de figuras

Figura 1 – Repetibilidade e precisão absoluta	16
Figura 2 – Coordenadas de referência dos diversos subsistemas (elos)	23
Figura 3 – Parâmetros de Denavit-Hartenberg	27
Figura 4 – Translação e Rotação do sistema devido ao erro no i-ésimo elo	30
Figura 5 – Erros Generalizados para i-ésimo Elo	31
Figura 6 – Esquema de Compensação de Erro	32
Figura 7 – Combinação linear de erros translacionais	36
Figura 8 – Combinação linear de erros translacionais	37
Figura 9 – Representação do modelo pinhole	42
Figura 10 – Geometria do modelo pinhole	43
Figura 11 – Modificações ao modelo pinhole com base na escala e cisalhamento dos pixels das câmeras. Figura (a) mostra os pixels segundo o modelo pinhole, figura (b) mostra um sistema mais realista onde escalas dos eixos x e y se diferenciam, figura (c) mostra o problema de cisalhamento	43
Figura 12 – Modelo pinhole modificado	44
Figura 13 – Transformação de coordenadas do mundo para a câmera	46
Figura 14 – Plataforma de Calibração	48
Figura 15 - Transformação do quadro do mundo para o quadro da câmera	48
Figura 16- Visão de um mesmo objeto em câmeras distintas deslocadas por uma distância fixa	49
Figura 17 – Imagem (à esquerda) submetida a gaussiana (no meio) e à diferença-do-gaussiano (à direita)	51
Figura 18 – Representação do procedimento de obtenção de diferentes diferenças de gaussianas para diversas octavas da imagem	52
Figura 19 – Máximo e mínimo das imagens de Diferença-do-Gaussiano são detectados comparando o pixel (marcado por X) com seus 26 vizinhos em 3x3 regions nas escalas corrente e adjacentes (marcados por círculos)	52
Figura 20 – Histograma de orientação do ponto-chave	55
Figura 21 – Ilustração da computação de descritores de pontos-chaves	56
Figura 22 – Triangulação	59
Figura 23 – TA40 e miniatura utilizada como “mestre” na tele-operação	61
Figura 24 – TA-40 e seu sistema de coordenadas	62
Figura 25 – Juntas 1 e 2	63
Figura 26 – Juntas 2 e 3	63
Figura 27 – Juntas 3 e 4	65
Figura 28 – Juntas 4 e 5	65
Figura 29 – Juntas 5 e 6	66
Figura 30 – Junta 6	67
Figura 31 – Eixos $O_2, O_3$ e $O_4$ sobre a forma de cinemática inversa de dois graus de liberdade	71
Figura 32 a/b – Imagem de painel obtidas com 10 segundos de diferença durante tarefa	76

Figura 33 a/b – Imagem de painel com os pontos obtidos pelo algoritmo SIFT	77
Figura 34 – Correlação de pontos	77
Figura 35 – Esquema do algoritmo para uma distância fixa, z	84
Figura 36 – Simulação de triangulação em função de distância de objetos à câmera	85
Figura 37 – Visões da simulação de calibração da base, com manipulador configurado com $\theta_1=26^\circ$ , $\theta_2=30^\circ$ , $\theta_3=20^\circ$ , $\theta_4=47^\circ$ , $\theta_5=60^\circ$ , $\theta_6=43^\circ$	87
Figura 38 – Visões de simulação de calibração da base, com manipulador configurado com $\theta_1=15^\circ$ , $\theta_2=-3^\circ$ , $\theta_3=-10^\circ$ , $\theta_4=10^\circ$ , $\theta_5=128^\circ$ , $\theta_6=0^\circ$	87
Figura 39 – Visões de simulação de calibração da base, com manipulador configurado com $\theta_1=0^\circ$ , $\theta_2=90^\circ$ , $\theta_3=-90^\circ$ , $\theta_4=0^\circ$ , $\theta_5=90^\circ$ , $\theta_6=0^\circ$	88
Figura 40 – Sistema de coordenadas do TA-40	94
Figura 41 – Imagens de Calibração tiradas da câmera esquerda e direita	104
Figura 42 – Pontos obtidos nas bordas da imagem	104
Figura 43 – Posição das câmeras em relação ao tabuleiro	105

## Lista de tabelas

Tabela 1 – Tabela de Parâmetros DH	67
Tabela 2 – Resultados da simulação de cinemática direta-inversa	81
Tabela 3 – Resultados da simulação com a Jacobiana de Identificação	82
Tabela 4 – Resultados da simulação de triangulação em distâncias	84
Tabela 5 – Medida de erro de triangulação a partir da resolução e distância da câmera (erros em milímetros).	85
Tabela 6 – Resultados da simulação de Calibração da Base	88

## Abreviações

ROV	Veículo Submarino Robótico
TA-40	Manipulador Robótico Submarino
GEC	Compensação de Erros Elásticos e Geométricos
SIFT	Transformada de Propriedades Invariantes à Escala
DH	Denavit-Hartenberg

# 1

## Introdução

### 1.1.

#### Motivação

A motivação para este estudo advém de um problema real de diversas aplicações em robótica, qual seja, o posicionamento preciso de robôs em ambientes dinâmicos. Manipuladores robóticos são empregados há décadas em ambientes industriais e nestas aplicações geralmente não há necessidade de uma calibração do manipulador devido ao baixo nível de precisão necessária e ao fato de toda sua área de trabalho compreender posições fixas. É o caso, por exemplo, de robôs que pintam carros em uma fábrica de automóveis. Porém, há aplicações em que a precisão é fundamental, como a inserção de componentes eletrônicos em uma placa de circuitos. Nesses casos, existe a necessidade de uma calibração do manipulador para compensar os erros de fábrica e aumentar a precisão. Métodos de calibração de manipuladores foram estudados por especialistas, tendo como uma das mais importantes referências o livro *Fundamentals of Manipulator Calibration* de Mooring, Roth e Driels (1991).

Certas tarefas executadas por manipuladores robóticos requerem que este reconheça sua área de trabalho e tome decisões baseado neste reconhecimento. É o caso de um manipulador robótico utilizado em intervenções submarinas. Tal manipulador desce a profundidades perigosas ao ser humano e deve fazer tarefas em um ambiente que muda a cada vez em que ele é transportado à sua área de trabalho. As dificuldades envolvidas no processo de calibragem de um manipulador em ambientes completamente mutáveis é um problema complexo e pouco tratado na literatura, motivando fortemente este estudo.

## 1.2.

### Objetivos do Trabalho

Os objetivos principais deste trabalho são:

- Estudar técnicas de calibragem de manipuladores seriais robóticos, modelagem cinemática direta e inversa.
- Estudar técnicas de visão computacional, especificamente modelos de câmeras, reconhecimento de padrões e métodos de rastreamento de pontos.
- Buscar métodos de posicionamento da extremidade do manipulador em relação ao seu ambiente de trabalho por uso de câmeras
- Aplicar as técnicas estudadas na calibração de um manipulador real, fazendo simulações para demonstrar a viabilidade de aplicar as técnicas em um ambiente real.

## 1.3.

### Descrição do Trabalho

É comum, em aplicações industriais de manipuladores robóticos, a programação de braços mecânicos por rastreamento de trajetórias, na qual um operador programa onde cada junta deverá se posicionar em períodos subsequentes do tempo. Desta forma, o manipulador poderá repetir os movimentos e efetuar a mesma tarefa repetidamente. Tal metodologia é suficiente para tarefas repetitivas onde o ambiente não muda e os objetos manipulados são movidos para posições constantes.

Porém, nem todas as atividades são realizadas em ambientes constantes e imutáveis. Este trabalho trata de um manipulador robótico com seis graus de liberdade (TA-40), utilizado para efetuar intervenções submarinas em painéis de controle submersos a profundidades perigosas à presença humana. A cada vez que há necessidade de intervenção, o manipulador é posicionado sobre um veículo

submarino robótico (Veículo Operado Remotamente - ROV) que o transporta até a área de trabalho. Uma vez atingido o seu destino, o submarino se fixa no local e permite que o manipulador realize suas tarefas.

Devido a questões que vão desde correntes marítimas até a imprecisão do operador que manipula o submarino robótico, a posição absoluta da base o manipulador em relação a sua área de trabalho nunca é igual à posição de uma intervenção submarina anterior. Assim, a programação por simples rastreamento de trajetórias não se aplica. Atualmente, faz-se a teleoperação do manipulador, na qual a movimentação do sistema robótico é executada em modo manual, e o operador trabalha em uma arquitetura mestre-escravo controlando direta e continuamente as posições do manipulador. Para essas tarefas é necessário apenas que o sistema possua repetibilidade suficiente, uma vez que quaisquer erros absolutos de posicionamento são visualmente compensados pelo operador humano. Define-se aqui repetibilidade como a capacidade de um sistema sair e retornar, com o menor erro possível, a uma posição arbitrária de seu volume de trabalho, sendo portanto uma medida de precisão relativa. A precisão absoluta mede os erros de posicionamento em relação a um referencial fixo na base do manipulador. A Figura 1 apresenta esquematicamente a diferença entre repetibilidade e precisão absoluta.

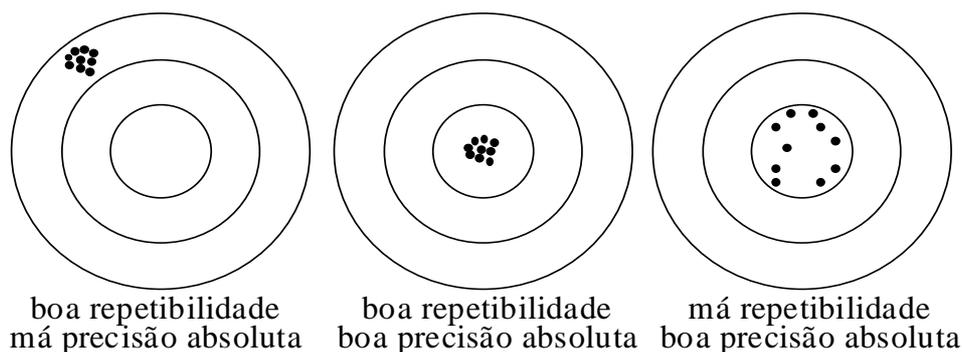


Figura 1: Repetibilidade e precisão absoluta

Em vista da aplicação direta do TA-40 em tarefas subaquáticas, percebe-se que ambas as técnicas de programação de manipuladores citadas anteriormente são impraticáveis ou insatisfatórias. A técnica na qual um operador humano ensina as trajetórias ao robô é impossibilitada devido a alta variabilidade do

ambiente. O manipulador, a cada vez que for levado ao seu ambiente de trabalho, deve ser treinado para um novo posicionamento, e portanto, perde-se toda a versatilidade. O método de teleoperação atribui ao robô uma precisão absoluta baixa, devido a própria deficiência da percepção humana.

Este trabalho segue tendências modernas de automação, as quais vêm colocando uma crescente ênfase em robôs guiados por sensores e programação off-line, automatizando total ou parcialmente muitas das tarefas a serem executadas. Sensores, como câmeras ligadas a um sistema de visão computacional detectam diferenças entre a posição real do manipulador e a posição desejada. Estas diferenças são então enviadas para os controladores para que estes corrijam a trajetória pré-programada. Os comandos de movimento do manipulador são programados off-line por um sistema de CAD. As trajetórias podem então ser testadas em um ambiente gráfico virtual, sem a necessidade de se ativar o robô. Assim, tarefas podem ser programadas à distância, sem o uso do robô, permitindo maior velocidade na validação de trajetórias para resolver problemas. Este estudo visa resolver o problema de calibração focado no manipulador TA-40, porém as técnicas de calibração aplicadas poderão ser utilizadas em outros manipuladores robóticos e em outros sistemas de posicionamento em ambientes diversos.

Para viabilizar estas técnicas é necessário que o robô tenha não apenas boa repetibilidade como também boa precisão absoluta, o que geralmente é difícil de ser alcançado em sistemas robóticos de grandes dimensões. Não é incomum que o modelo geométrico do robô, originário do projeto inicial deste durante sua fabricação, se diferencie do modelo real devido a tolerâncias de construção. Estas diferenças de modelo provocam perda de precisão e só podem ser revertidos a partir da calibração.

Calibragem de robôs é um processo no qual a precisão absoluta do sistema é melhorada através de modificações em seu software de controle de forma a compensar erros. Para melhor ajustar o modelo geométrico de um robô por calibragem, uma quantidade suficiente de medições precisa ser realizada. Essas medições consistem em posições das juntas internas do robô e de coordenadas de um ou mais pontos da estrutura do sistema em relação a um sistema de coordenadas designado. Técnicas de calibragem podem ser aplicadas especificamente para calibrar apenas a extremidade do manipulador, apenas sua

base (ou seja, no caso de intervenções submarinas, a posição do veículo submarino que contém o robô), apenas a posição de uma ferramenta acoplada a extremidade do manipulador, ou qualquer combinação dos três.

O processo de calibração pode ser dividido em quatro etapas distintas:

- Determinação do modelo matemático que representa a geometria e o movimento do manipulador (modelagem cinemática).
- Medições de posição e orientação da extremidade do manipulador em coordenadas do mundo (medidas de posição).
- Identificação das relações entre os ângulos das juntas e as posições da extremidade do robô (Identificação cinemática)
- Modificação dos comandos de controle para permitir que as tarefas sejam completadas com sucesso (Compensação Cinemática)

As etapas de seleção de modelo, identificação e compensação foram estudadas (Meggiolaro et al. M., 1999a; Meggiolaro M., 2000a), com a proposição de uma técnica denominada GEC (*Geometric and Elastic Error Compensation*) que modela qualquer tipo de erro no manipulador de uma forma unificada, apresentando melhores resultados em relação aos métodos tradicionais. O método GEC foi aplicado com sucesso, por exemplo, em dois sistemas robóticos, um manipulador capaz de posicionar pacientes em hospitais (Meggiolaro et al. M., 1998) e um manipulador hidráulico Schilling Titan II (Meggiolaro et al. M., 1999b) melhorando significativamente a precisão absoluta destes sistemas

Como o TA-40 tem sua base móvel, faz-se necessária uma calibração desta base em relação ao mundo sempre que o manipulador for ativado. A abordagem para esta calibração passa pelo uso de imagens do local. A calibragem por um sistema de câmeras é potencialmente rápida, automatizada e não-invasiva ao volume de trabalho. Há dois tipos de configuração para sistemas de medição baseados em imagens. O primeiro é fixar câmeras no ambiente próximo ao da tarefa que possam visualizar uma referência de calibragem presa na extremidade do robô. O segundo tipo de configuração é montar uma câmera ou um par de câmeras na extremidade do manipulador. A configuração de câmeras móveis

presas na extremidade do manipulador resolve o conflito entre alta precisão e grande volume de trabalho visualizável.

O uso de câmeras para calibração do manipulador passa por três estágios:

- Reconhecimento de padrões
- Acompanhamento de padrões
- Recuperação da cinemática da câmera

O primeiro estágio dedica-se a encontrar padrões nas imagens que sejam robustos a rotação, translação, mudança de escala e ruído. Diversos métodos são utilizados para este fim, sendo que o mais robusto é a identificação de cantos nas imagens. Este método, porém, dificulta o estágio seguinte, no qual se procura relacionar os mesmos pontos em imagens seguidas. É comum utilizar correlação para fazer este tracking, porém esta não é robusta a rotação ou a mudanças de escala, o que torna ambos os métodos insatisfatórios.

Para identificação dos pontos e posterior rastreamento será utilizada a Transformada de Propriedades Invariantes a Escala (SIFT-Scale Invariant Feature Transform) desenvolvido por David G Lowe (LOWE D. G, 2004), que descobre padrões em imagens subsequentes e dá a cada ponto um conjunto de parâmetros para que estes possam ser descobertos em imagens subsequentes. Este método é robusto a rotação, translação, escala e ruído, permitindo a sua aplicação neste trabalho.

A última etapa da calibração da câmera visa encontrar a distância dos pontos capturados com relação à câmera, bem como obter a posição e orientação da câmera em relação ao mundo. Uma vez que a câmera está instalada na extremidade do manipulador, obtêm-se assim a posição e orientação da extremidade em relação à base, bem como as distâncias dos pontos vistos à extremidade. O algoritmo utilizado para tanto se chama triangulação.

Após a calibragem do sistema, tarefas de teleoperação automática ou semi-automática são factíveis. Além disso, o conhecimento da posição absoluta do manipulador e do ambiente de trabalho ao seu redor permite gerar um sistema de visualização 3-D virtual que reflita as configurações do sistema em tempo real. Visualização virtual permite ao teleoperador visualizar áreas obscurecidas

utilizando câmeras virtuais. As câmeras virtuais também permitem magnificar as bordas de peças a serem inseridas, facilitando a execução destas tarefas. Em manipuladores com sensores de força e torque em sua extremidade, é possível ainda visualizar no próprio ambiente virtual medições de forças de contato entre o robô e o equipamento, como pode ser visto em um sistema de ambientes virtuais desenvolvido para o uso em tarefas de manutenção em usinas nucleares (Meggiolaro M., 1999). A partir da teoria e dos conhecimentos apresentados nesta dissertação, será possível viabilizar a operação de manipuladores robóticos em ambientes reais dinâmicos de forma facilitada.

#### 1.4.

#### **Organização da Dissertação**

Esta dissertação está dividida em seis capítulos, descritos a seguir:

O capítulo 2 resume toda a teoria necessária para o processo de calibração de um manipulador robótico. Este capítulo explica com detalhes os conceitos básicos de modelagem cinemática de manipuladores planos, incluindo as equações de Rodrigues, Transformações Homogêneas e Convenção de Modelagem de Denavit-Hartenberg. Também apresenta as técnicas de calibração e de compensação de erros que serão utilizadas posteriormente para a calibragem do TA-40. Por fim, neste capítulo são discutidos os requisitos e o método para se obter a cinemática inversa de manipuladores.

O Capítulo 3 inclui a teoria e os conceitos de visão computacional que serão utilizados para fazer o reconhecimento do ambiente de trabalho e o posicionamento da base do manipulador em relação ao mundo. Entre os tópicos abordados estão o Modelo de Câmera, explicitado pelos parâmetros intrínsecos e extrínsecos, bem como uma técnica de Calibração da Câmera. Também neste capítulo, encontra-se um método de emparelhamento de pontos em imagens obtidas por câmeras distintas (algoritmo SIFT) e o algoritmo de triangulação necessário para posicionar o manipulador.

O Capítulo 4 descreve ao caso particular do manipulador TA-40, empregando as técnicas e algoritmos descritos nos capítulos dois e três para fazer a calibragem total deste manipulador. O capítulo se divide em uma rápida

apresentação do manipulador TA-40, seguido da descrição do modelo de Denavit-Hartenberg para o mesmo. Em seguida, é demonstrada a aplicação dos algoritmos de calibração para encontrar as matrizes de erro. Por fim, mostra-se o uso das técnicas de visão computacional para obter o posicionamento da base do manipulador.

O Capítulo 5 possui resultados de diversas simulações feitas para testar os algoritmos no caso específico do TA-40.

O Capítulo 6 é composto de observações e comentários dos resultados e propostas de novos projetos.

## 2

# Modelagem Cinemática para Calibração de Manipuladores

### 2.1.

#### Introdução

A modelagem cinemática é o primeiro passo para o processo de calibragem do manipulador. O modelo cinemático do manipulador permite determinar, a partir dos ângulos de cada junta, as posições e orientações das juntas e da extremidade do manipulador. O processo de calibração inclui a mudança dos parâmetros do modelo cinemático para compensar erros.

Uma forma matematicamente conveniente para descrever posições e orientações dos diversos eixos do sistema é a utilização de transformações homogêneas. Inicialmente, estabelecem-se sistemas de coordenadas de referência para os diversos subsistemas, como visto na Figura 2, obtida do manual técnico do manipulador. A seguir, adicionam-se sistemas de coordenadas locais para cada parte móvel. No caso de um manipulador robótico, um sistema de coordenadas local é adicionado a cada eixo do robô, tendo sua base como o sistema de coordenadas inicial. Um conjunto de matrizes de transformação 4x4 relaciona cada sistema de coordenadas local com a de seu vizinho.

Além da cinemática direta, pode-se também fazer uma modelagem da cinemática inversa, na qual se obtêm os ângulos de cada junta a partir do conhecimento da posição e da orientação da extremidade do manipulador. Esta modelagem normalmente é mais complexa e nem sempre possui solução. Para tal modelagem, faz-se necessário inicialmente obter, pela cinemática direta, as matrizes de transformação homogênea.

Este capítulo está organizado da seguinte forma: a seção 2.2 apresenta conceitos básicos de cinemática, necessários para a modelagem cinemática do TA-40 (incluindo as equações de Rodrigues, transformações homogêneas e matrizes de rotação gerais). A seção 2.3 inclui a convenção de modelagem de

Denavit-Hartenberg (D-H). A seção 2.4 demonstra a metodologia de calibração do TA-40. A seção 2.5 finaliza o capítulo com a cinemática inversa do manipulador.

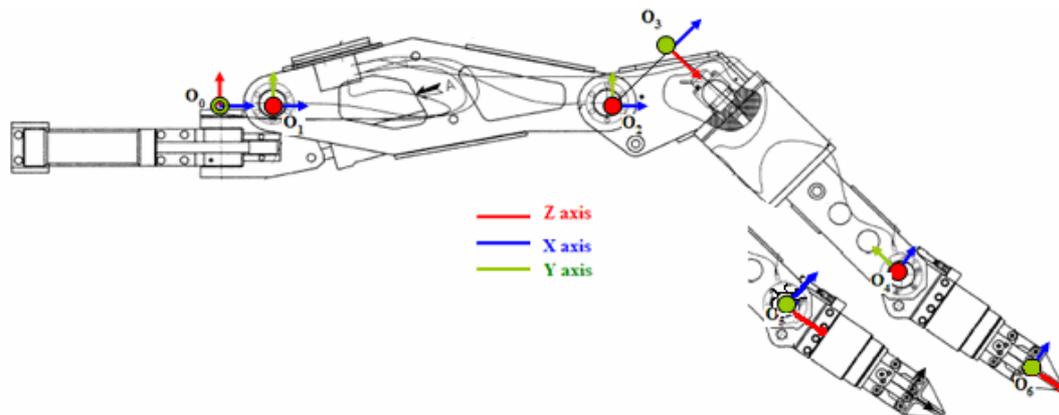


Figura 2: Coordenadas de referência dos diversos subsistemas (elos)

## 2.2.

### Conceitos Básicos de Cinemática

A movimentação de um corpo rígido no espaço pode ser convertida em uma translação e uma rotação em relação a um eixo. Se este eixo for caracterizado como um vetor chamado  $k$ , o movimento de um corpo rígido define-se por uma rotação de um ângulo  $\theta$  ao redor deste eixo e uma translação de distância  $d$  paralela a  $k$ . Os parâmetros  $(d, \theta)$  definem a movimentação em relação ao eixo  $k$ .

Considera-se  $P$  um ponto arbitrário do corpo rígido e  $S$  um ponto arbitrário no eixo  $k$ . Considera-se também  $r_s$  a posição de  $S$  em relação a um eixo de coordenadas de referência, e  $r_p^{(i)}$  a posição do ponto  $P$  em relação ao mesmo eixo de referência no momento  $i$ . Pela equação de Rodrigues (Murray, R. M.; 1994), é possível relacionar os pontos  $r_p^{(i)}$  e  $r_p^{(i+1)}$  através dos parâmetros de  $r_s$ ,  $k$ ,  $d$  e  $\theta$ :

$$r_p^{(i+1)} = (r_p^{(i)} - r_s) \cos \phi + k \times (r_p^{(i)} - r_s) \sin \phi + [(r_p^{(i)} - r_s) \cdot k] k (1 - \cos \phi) + r_s + dk \quad \text{Eq. 2.2.1}$$

Na forma matricial:

$$\begin{bmatrix} r_{p,x}^{(i+1)} \\ r_{p,y}^{(i+1)} \\ r_{p,z}^{(i+1)} \end{bmatrix} = R \begin{bmatrix} r_{p,x}^{(i)} \\ r_{p,y}^{(i)} \\ r_{p,z}^{(i)} \end{bmatrix} + \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}, \quad \text{Eq. 2.2.2}$$

Onde a matriz  $R$  de rotação 3x3 é definida por:

$$R \equiv \begin{bmatrix} n_x & p_x & b_x \\ n_y & p_y & b_y \\ n_z & p_z & b_z \end{bmatrix} = \begin{bmatrix} k_x^2 v\phi + c\phi & k_x k_y v\phi + k_z s\phi & k_x k_z v\phi - k_y s\phi \\ k_x k_y v\phi + k_z s\phi & k_y^2 v\phi + c\phi & k_y k_z v\phi - k_x s\phi \\ k_x k_z v\phi - k_y s\phi & k_y k_z v\phi - k_x s\phi & k_z^2 v\phi + c\phi \end{bmatrix} \quad \text{Eq. 2.2.3}$$

Onde  $v\phi = (1 - \cos \phi)$ ,  $s\phi = \sin \phi$  e  $c\phi = \cos \phi$ .

A matriz  $R$  é a matriz geral de rotação de um ângulo  $\phi$  em torno de um eixo  $k$ . Essa matriz pode ser denotada por  $Rot(k, \phi)$ . Os três vetores unitários  $n$ ,  $p$  e  $b$  são conhecidos respectivamente como normal, orientação e proximidade, respectivamente.

A Equação (2.2.2) pode ser escrita de forma mais compacta através da matriz de transformação homogênea  $T$ .

$$\begin{bmatrix} r^{(i+1)} \\ 1 \end{bmatrix} = T \begin{bmatrix} r^{(i)} \\ 1 \end{bmatrix} \quad \text{Eq. 2.2.4}$$

Onde:

$$T = \begin{bmatrix} R & p \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \quad \text{Eq. 2.2.5}$$

Pode-se, assim, obter matrizes de transformação homogêneas para algumas movimentações básicas de um corpo rígido. Considerando  $d=0$  e uma rotação de  $\theta$  em relação ao eixo  $x$ , ou seja  $k=(1,0,0)^T$ :

$$Rot(x, \theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta & 0 \\ 0 & \sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Igualmente, pode-se encontrar a rotação de um ângulo  $\Psi$  ao redor do eixo  $y$ :

$$Rot(y, \Psi) = \begin{bmatrix} \cos \Psi & 0 & \sin \Psi & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \Psi & 0 & \cos \Psi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

E a rotação de um ângulo  $\varphi$  ao redor do eixo  $z$ :

$$Rot(z, \varphi) = \begin{bmatrix} \cos \varphi & -\sin \varphi & 0 & 0 \\ \sin \varphi & \cos \varphi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

A matriz de translação sobre os eixos  $x$ ,  $y$  e  $z$ , por distâncias  $a$ ,  $b$  e  $c$ , respectivamente, pode ser obtida pela combinação das três translações básicas:

$$Trans(a, b, c) = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

A sequência de dois movimentos  $T_1$  e  $T_2$  é obtida pela multiplicação de ambas as matrizes  $T_1 T_2$ . Esta transformação é vista fisicamente como a transformação do eixo em uma localização  $P_1$  para um eixo em  $P_2$ .

### 2.3.

#### Convenção de Modelagem de Denavit-Hartenberg

Um manipulador é usualmente constituído por um grupo de corpos rígidos, ou elos, conectados por juntas. No caso do TA-40, como em muitos manipuladores industriais, cada elo se conecta a dois outros membros. Logo, cada

elo tem dois eixos. A modelagem de elos de Denavit-Hartenberg (DH) caracteriza a estrutura cinemática do manipulador em termos de parâmetros físicos de cada elo.

A convenção de DH permite encontrar uma transformação da base do manipulador até sua extremidade, fazendo-se necessário definir as coordenadas dos elos e utilizar uma técnica sistemática que permita descrever a cinemática do robô de  $n$  graus de liberdade de forma única.

A Figura 3 mostra um par de elos adjacente, elo  $i-1$  e elo  $i$ , e suas juntas associadas, juntas  $i+1$  e  $i$ . A linha  $\overline{H_i O_i}$  na figura é chamada de **normal comum** entre os eixos  $i$  e  $i+1$ . A relação entre dois elos pode ser descrita pela posição e orientação relativa entre os dois sistemas de coordenadas ligados aos dois elos. Na notação DH, a origem do  $i$ -ésimo sistema de coordenadas está localizado na interseção entre os eixos  $i+1$  e a normal comum entre os eixos das juntas  $i$  e  $i+1$ , como mostrado na Figura 3. Nota-se que o sistema de coordenadas do elo  $i$  está na junta  $i+1$  e não na junta  $i$ . O eixo  $x_i$  tem sua direção na extensão da linha formada pela normal comum, enquanto que o eixo  $z_i$  está na direção do eixo da junta  $i+1$ . Finalmente, o eixo  $y_i$  é escolhido como eixo resultante.

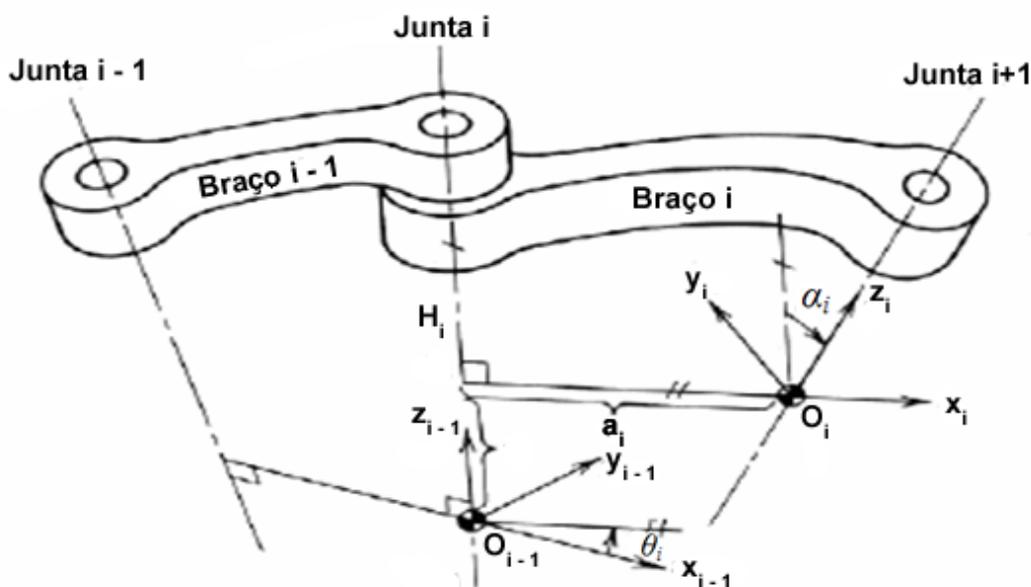


Figura 3: Parâmetros de Denavit-Hartenberg

A posição relativa entre os dois sistemas de coordenadas pode ser completamente deduzida a partir dos seguintes parâmetros:

$a_i$  – tamanho da normal comum

$d_i$  – distancia entre o ponto  $O_{i-1}$  e  $H_i$

$\alpha_i$  – ângulo entre as juntas  $i$  e o eixo  $z_i$  no sentido da mão direita

$\theta_i$  – ângulo entre o eixo  $x_i$  e a normal comum medida sobre o eixo  $z_i$  usando o sentido da mão direita.

Os dois primeiros parâmetros são constantes e determinados pela geometria do elo, enquanto que os dois últimos variam de acordo com o tipo de movimento da junta. No caso de juntas revolutas, o valor de  $\theta_i$  indica o movimento angular da junta, enquanto que o parâmetro  $d_i$  se manterá constante. Por outro lado, em juntas prismáticas, o contrário ocorre, sendo  $d_i$  um parâmetro variável.

A convenção de modelagem DH inicialmente associa um eixo de coordenadas  $\{x_i, y_i, z_i\}$ ,  $i=0, \dots, n$  para cada elo. A base, a extremidade do robô, e o ambiente são denotados respectivamente como eixos de coordenadas dos elos 0,  $n$  e  $n+1$ . Para um manipulador serial, o número de graus de liberdade é exatamente  $n$ . Para uma junta de revolução, a origem do sistema de coordenadas  $i-1$  está na junta  $i$ . Para uma junta prismática, por outro lado, a origem pode ser decidida arbitrariamente. A direção positiva do eixo  $i$  define o vetor unitário  $z_{i-1}$ .

A transformação do sistema de coordenadas do mundo para o da extremidade é:

$$T_n^0 = A_0 A_1 A_2 \dots A_i \dots A_{n-1} A_n \quad \text{Eq. 2.3.1}$$

Onde  $A_i$  é a matriz de transformação homogênea entre o sistema de coordenadas  $i$  com respeito ao sistema de coordenadas  $i-1$ . A transformação de elos  $A_i$  é representada, portanto, pelos seus quatro parâmetros  $(\theta_i, a_i, d_i, \alpha_i)$ . A construção das matrizes  $A_i$  para elos internos  $i=1, 2, \dots, n-1$  é mostrada a seguir. Se a  $i$ -ésima junta é revoluta, as seguintes transformações são necessárias para passar do sistema de coordenadas  $i-1$  ao sistema  $i$ .

- Rotacionar o sistema de coordenadas  $i-1$  ao redor do eixo  $z_{i-1}$  em um ângulo  $\theta_i$ , de modo que o eixo  $x$  do sistema de coordenadas em movimento seja paralelo ao eixo  $x_i$ .
- Transladar o sistema de coordenadas pelo eixo  $z_{i-1}$  na distância  $d_i$ , para a origem do sistema em movimento alcançar o ponto de interseção entre o eixo da  $i$ -ésima junta e a normal comum.
- Transladar o sistema de coordenadas pelo eixo  $x_i$  na distância  $a_i$ , para a origem do sistema em movimento coincidir com a do sistema de coordenadas  $i$ .
- Rotacionar o sistema de coordenadas  $i-1$  ao redor do eixo  $x_{i-1}$  em um ângulo  $\alpha_i$ , de modo que o eixo em movimento esteja agora coincidente com o eixo de coordenadas  $i$ .

$$A_i = Rot(z, \theta_i) Trans(0, 0, d_i) Trans(a_i, 0, 0) Rot(x, \alpha_i) \quad \text{Eq. 2.3.2}$$

$$A_i = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & 0 & a_i \sin \theta_i \\ 0 & 0 & -\cos \theta_i \sin \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Eq. 2.3.3}$$

Similarmente, se a  $i$ -ésima junta for prismática, por definição  $a_i=0$  e a variável passa a ser  $d_i$

$$A_i = Rot(z, \theta_i) Trans(0, 0, d_i) Rot(x, \alpha_i) \quad \text{Eq. 2.3.4}$$

No caso de juntas consecutivas com eixos paralelos, a convenção DH estipula que a normal comum que satisfaz  $d_i=0$  é selecionada.

O sistema de coordenadas do mundo e da extremidade são usualmente definidos pelo usuário. O sistema de coordenadas da base, por outro lado, é comumente definido pelo fabricante.

## 2.4.

### Método Clássico de Calibração de Manipuladores

Uma vez obtidas as matrizes homogêneas que modelam o manipulador a partir dos parâmetros de DH, seria possível determinar a posição através de uma simples multiplicação destas matrizes a partir das coordenadas da base. Porém, como já foi dito antes, o manipulador possui diversos erros, que apesar de serem pequenos em cada elo, provocam um efeito grande na extremidade.

Os erros podem ser classificados como “repetitivos” e “randômicos” (Slocum A., 1992). Erros repetitivos são aqueles nos quais os valores numéricos e sinais são constantes durante uma configuração do manipulador e peso carregado. Exemplos de erro repetitivo é um erro resultante das tolerâncias de fabricação dos componentes do manipulador e deflexões devido ao carregamento de cargas pesadas. Erros randômicos são erros cujo valor numérico e sinal mudam sem previsibilidade. Um exemplo de um erro randômico ocorre no escorregar dos dentes de uma engrenagem do atuador. A calibração que será feita sobre o TA-40 corrigirá apenas os erros repetitivos.

Como visto na seção anterior, a cinemática do manipulador é modelada por um conjunto de matrizes homogêneas que transformam o sistema de coordenadas de um elo para outro. A posição e orientação do sistema de coordenadas de referência  $F_i$ , com respeito ao anterior  $F_{i-1}$ , é definido por uma matriz 4x4  $A_i$ , que possui a forma geral:

$$A_i = \begin{bmatrix} \mathbf{R}_i & \mathbf{T}_i \\ 0 & 1 \end{bmatrix} \quad \text{Eq. 2.4.1}$$

O termo  $\mathbf{R}_i$  é uma matriz 3x3 de rotação, entre o sistema de coordenadas  $F_i$  e o sistema  $F_{i-1}$ , e  $\mathbf{T}_i$  é um vetor 3x1 que translada o centro do sistema  $F_i$  até o sistema  $F_{i-1}$ , como visto na Figura 4. Os elementos da matriz  $A_i$  dependem dos parâmetros de DH para aquele elo.

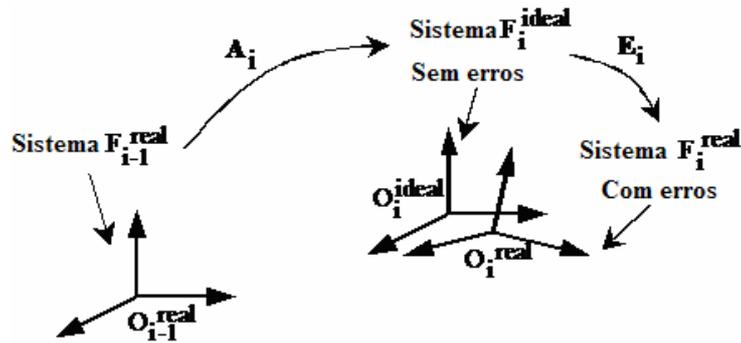


Figura 4: Translação e Rotação do sistema devido ao erro no i-ésimo elo

Os erros são expressos como mais uma matriz de transformação homogênea que desloca a posição final do elo subsequente em relação ao ideal. Os efeitos destes erros nas juntas, por menores que sejam, são acumulados de junta em junta podendo provocar um erro considerável na extremidade.

A matriz homogênea 4x4  $E_i$  produz o deslocamento da posição e orientação do sistema de coordenadas real  $F_i^{real}$  com respeito a sua localização ideal  $F_i^{ideal}$ . A Equação 2.4.2 mostra o formato da matriz de erros  $E_i$ .

$$E_i = \begin{bmatrix} 1 & -\varepsilon_r & \varepsilon_s & \varepsilon_x \\ \varepsilon_r & 1 & -\varepsilon_p & \varepsilon_y \\ -\varepsilon_s & \varepsilon_p & 1 & \varepsilon_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Eq. 2.4.2}$$

Os seis parâmetros  $\varepsilon_{x,b}$ ,  $\varepsilon_{y,b}$ ,  $\varepsilon_{z,b}$ ,  $\varepsilon_{s,b}$ ,  $\varepsilon_{r,i}$  e  $\varepsilon_{p,i}$  são chamados parâmetros de erro generalizados, os quais podem ser função da geometria do sistema e das variáveis das juntas. Para um manipulador de  $n$  graus de liberdade, existem  $6(n+1)$  erros generalizados que podem ser escritos na forma de um vetor de dimensão  $6(n+1) \times 1$ :  $\varepsilon = [\varepsilon_{x,0}, \dots, \varepsilon_{x,b}, \varepsilon_{y,b}, \varepsilon_{z,b}, \varepsilon_{s,b}, \varepsilon_{r,b}, \varepsilon_{p,b}, \dots, \varepsilon_{p,n}]^T$ , com  $i = 0 \dots n$  (supondo que ambos o manipulador e sua base estão sendo calibrados). Se o manipulador é calibrado em relação à sua própria base, então a matriz de erro  $E_0$  (que modela os erros da localização da base) é eliminada, reduzindo o número de erros generalizados para  $6n$ . Os erros generalizados que dependem da geometria do sistema podem ser calculados dos erros físicos elo a elo.

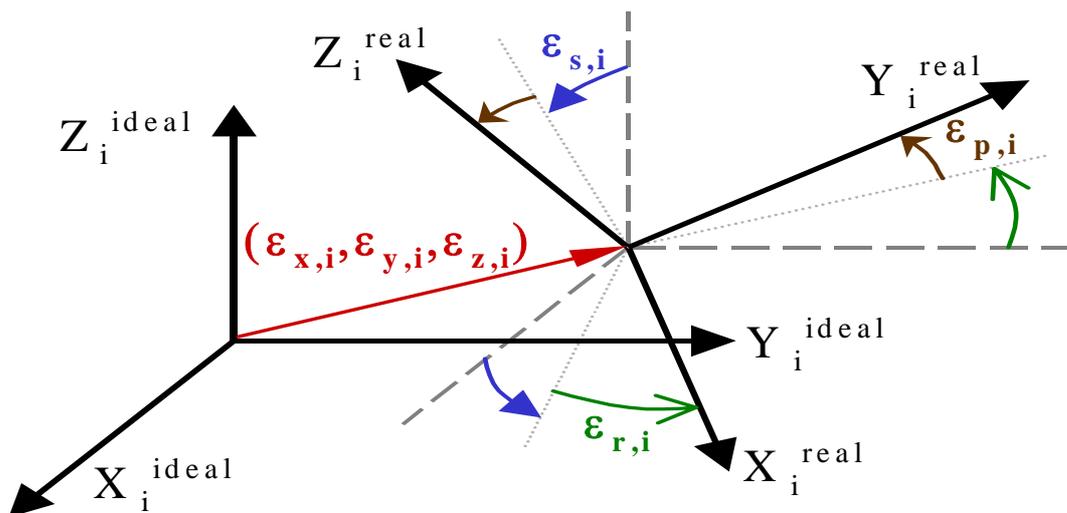


Figura 5: Erros Generalizados para i-ésimo Elo

Definido o erro como uma matriz de transformação homogênea, pode-se estender a Equação 2.3.1 de forma a incluí-los:

$$T_n^0(q, \varepsilon) = E_0 A_0 E_1 A_1 E_2 A_2 \dots A_{n-1} E_{n-1} A_n E_n \quad \text{Eq. 2.4.3}$$

A matriz  $T_n^0$  é uma a matriz homogênea 4x4 na forma da Equação (2.2.5) e descreve a posição e orientação da extremidade do manipulador com respeito ao sistema de coordenadas inercial da base como função dos parâmetros de configuração  $q$  e do vetor de erros generalizados  $\varepsilon$ .

É possível definir o vetor 6x1 da posição e orientação da extremidade  $\mathbf{X}^{real}$  como os componentes translacionais e os componentes formados pelos três ângulos rotacionais da matriz  $T_n^0$ . Se for utilizada a diferença entre a posição real e a ideal, ter-se-á erro de posição e orientação da extremidade do manipulador  $\Delta\mathbf{X}$ :

$$\Delta\mathbf{X} = \mathbf{X}^{real} - \mathbf{X}^{ideal} \quad \text{Eq. 2.4.4}$$

Considerando-se que os erros generalizados são pequenos,  $\Delta\mathbf{X}$  pode ser calculado pela seguinte equação linear em  $\varepsilon$ :

$$\Delta\mathbf{X} = \mathbf{J}_e \varepsilon \quad \text{Eq. 2.4.5}$$

Onde  $J_e$  é a matriz Jacobiana  $6 \times 6(n+1)$  do erro da extremidade  $\Delta X$  com respeito aos elementos do vetor de erros generalizados  $\varepsilon$ , também conhecida como a matriz Jacobiana de Identificação (Zhuang H., pp.805-810, 1999). Assim como no caso dos erros generalizados,  $J_e$  depende da configuração do sistema, geometria e cargas carregadas.

No processo de calibração, deve-se encontrar a matriz Jacobiana de Identificação a partir dos parâmetros cinemáticos, e tomar um conjunto de medidas da posição real da extremidade do manipulador em diversas configurações. Uma vez de posse destes dados, utiliza-se a Eq. 2.4.5 para obter os erros generalizados  $\varepsilon$  com uma simples inversão da matriz jacobiana. Uma vez de posse dos erros generalizados, estes podem ser utilizados para calcular a posição real do manipulador em qualquer configuração. Pode-se também utilizar a Equação 2.4.5 para compensar os erros. A Figura 6 mostra um esquema do algoritmo de compensação de erros baseado na Equação 2.4.5. O método de encontrar  $\varepsilon$  a partir de medidas experimentais é descrito a seguir.

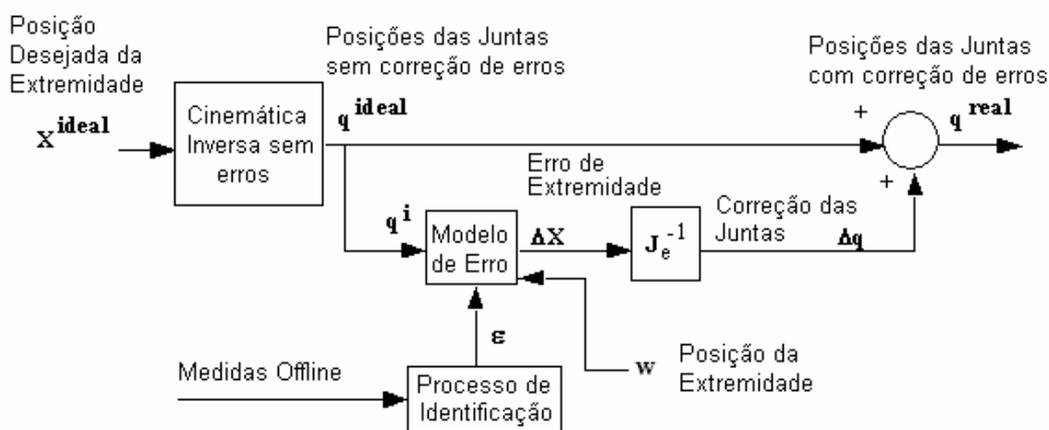


Figura 6: Esquema de Compensação de Erro

Para calcular os erros generalizados  $\varepsilon$  considera-se que alguns componentes do vetor  $\Delta X$  podem ser medidos em um número finito de diferentes configurações do manipulador. Porém, como coordenadas de posições são muito mais simples de se medir do que orientações, em muitos casos apenas as três coordenadas das posições de  $\Delta X$  são medidas.

Supondo que todas as 6 componentes de  $\Delta X$  podem ser medidas, para um manipulador de  $n$  graus de liberdade,  $6(n+1)$  erros generalizados  $\boldsymbol{\varepsilon}$  podem ser calculados medindo-se  $\Delta X$  em  $m$  configurações diferentes, definidas como  $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_m$ , e então escrevendo a Equação 2.4.5  $m$  vezes:

$$\Delta \mathbf{X}_t = \begin{bmatrix} \Delta \mathbf{X}_1 \\ \Delta \mathbf{X}_2 \\ \dots \\ \Delta \mathbf{X}_m \end{bmatrix} = \begin{bmatrix} \mathbf{J}_e(\mathbf{q}_1) \\ \mathbf{J}_e(\mathbf{q}_2) \\ \dots \\ \mathbf{J}_e(\mathbf{q}_m) \end{bmatrix} \cdot \boldsymbol{\varepsilon} = \mathbf{J}_t \cdot \boldsymbol{\varepsilon} \quad \text{Eq. 2.4.6}$$

Na Equação 2.4.6,  $\Delta \mathbf{X}_t$  é o vetor  $m \times 1$  formado por todos os vetores medidos  $\Delta X$  em  $m$  configurações diferentes e  $\mathbf{J}_t$  é a matriz  $6m \times 6(n+1)$  formada pelas  $m$  matrizes Jacobianas de Identificação  $\mathbf{J}_e$  em  $m$  configurações, denominada aqui de Jacobiano de Identificação Total. Para reduzir os efeitos de ruído de medidas,  $m$  é, em geral, muito maior do que  $n$ .

Considerando-se que os erros generalizados  $\boldsymbol{\varepsilon}$  são repetitivos e constantes, uma estimativa única por mínimos quadrados  $\hat{\boldsymbol{\varepsilon}}$  pode ser calculada por:

$$\hat{\boldsymbol{\varepsilon}} = (\mathbf{J}_t^T \mathbf{J}_t)^{-1} \mathbf{J}_t^T \cdot \Delta \mathbf{X}_t \quad \text{Eq. 2.4.7}$$

Porém, se a matriz de identificação jacobiana  $J_e(q_i)$  contiver colunas linearmente dependentes, a Equação 2.4.7 vai produzir estimativas com baixa precisão devido ao fraco condicionamento da matriz (Hollerbach, pp. 573-591, 1996). Isto ocorre quando há redundância no modelo de erro, e neste caso não é possível distinguir a contribuição de cada componente dos erros generalizados sobre o erro total.

## 2.5.

**Eliminação de Erros Redundantes**

Como visto na seção anterior, os erros redundantes causam baixa precisão ao processo de calibração e precisam ser eliminados. A seguir é apresentado um método analítico para efetuar tal eliminação.

Primeiramente, definem-se as combinações lineares dos coeficientes,  $\mathbf{J}_{x,i}$ ,  $\mathbf{J}_{y,i}$ ,  $\mathbf{J}_{z,i}$ ,  $\mathbf{J}_{s,i}$ ,  $\mathbf{J}_{r,i}$  e  $\mathbf{J}_{p,i}$ , que são colunas de  $\mathbf{J}_e$  associadas aos componentes dos erros generalizados  $\varepsilon_{x,i}$ ,  $\varepsilon_{y,i}$ ,  $\varepsilon_{z,i}$ ,  $\varepsilon_{s,i}$ ,  $\varepsilon_{r,i}$  e  $\varepsilon_{p,i}$ , respectivamente ( $i$  entre 0 e  $n$ ). As combinações lineares dos coeficientes são expressas a partir dos parâmetros DH do manipulador. A Equação 2.4.5 pode ser reescrita como:

$$\Delta X = \begin{bmatrix} J_{X0} & \dots & J_{Xi} & J_{Yi} & J_{Zi} & J_{Si} & J_{Ri} & J_{Pi} & \dots & J_{Pn} \end{bmatrix} \begin{bmatrix} \varepsilon_{X0} \\ \dots \\ \varepsilon_{Xi} \\ \varepsilon_{Yi} \\ \varepsilon_{Zi} \\ \varepsilon_{Si} \\ \varepsilon_{Ri} \\ \varepsilon_{Pi} \\ \dots \\ \varepsilon_{Pn} \end{bmatrix} \quad \text{Eq. 2.5.1}$$

Para cada elo  $i$ , entre 1 e  $n$ , as seguintes combinações lineares são sempre válidas (Meggiolaro M, Apêndice A, 2000):

$$\mathbf{J}_{z,(i-1)} \equiv \sin \alpha_i \mathbf{J}_{y,i} + \cos \alpha_i \mathbf{J}_{z,i} \quad \text{Eq. 2.5.2}$$

$$\mathbf{J}_{r,(i-1)} \equiv a_i \cos \alpha_i \mathbf{J}_{y,i} - a_i \sin \alpha_i \mathbf{J}_{z,i} + \sin \alpha_i \mathbf{J}_{s,i} + \cos \alpha_i \mathbf{J}_{r,i} \quad \text{Eq. 2.5.3}$$

Se a junta  $i$  for prismática, combinações adicionais para as colunas de  $\mathbf{J}_e$  são encontradas:

$$\mathbf{J}_{x,(i-1)} \equiv \mathbf{J}_{x,i} \quad \text{Eq. 2.5.4}$$

$$\mathbf{J}_{y,(i-1)} \equiv \cos \alpha_i \mathbf{J}_{y,i} - \sin \alpha_i \mathbf{J}_{z,i} \quad \text{Eq. 2.5.5}$$

As combinações lineares mostradas acima estão sempre presentes, independentemente dos valores de  $a_i$  e  $\alpha_i$ , mesmo em casos especiais (como  $a_i=0$ ). Se a posição completa da extremidade (posição e orientação) for medida, então as Equações 2.5.2-2.5.5 são as únicas combinações lineares do elo  $i$ .

Para obter uma matriz Jacobiana de Identificação não singular, chamada aqui de  $G_e$ , as colunas  $J_{z,(i-1)}$  e  $J_{r,(i-1)}$  devem ser eliminadas da matriz  $J_e$  para todos os valores de  $i$  entre 1 e  $n$ . Se a junta  $i$  for prismática, então as colunas  $J_{x,(i-1)}$  e  $J_{y,(i-1)}$  devem ser também eliminadas. Para um manipulador de  $n$  graus de liberdade com  $r$  juntas rotativas e  $p$  ( $p = n-r$ ) juntas prismáticas, um total de  $2r+4p$  colunas devem ser eliminadas do Jacobiano de Identificação  $J_e$  para formar a submatriz  $G_e$ . Isso significa que  $2r+4p$  erros generalizados não podem ser obtidos medindo-se a posição da extremidade.

Por definição, os parâmetros de erros dependentes eliminados por  $\boldsymbol{\varepsilon}$  não afetam o erro da extremidade, resultando na identidade:

$$\Delta \mathbf{X} = \mathbf{J}_e \boldsymbol{\varepsilon} \equiv \mathbf{G}_e \boldsymbol{\varepsilon}^* \quad \text{Eq. 2.5.6}$$

Usando a identidade acima e as combinações lineares das colunas de  $J_e$  das Equações 2.5.2-2.5.5, é possível obter todos os relacionamentos entre o conjunto de erros generalizados  $\boldsymbol{\varepsilon}$  e seus subconjuntos independentes,  $\boldsymbol{\varepsilon}^*$  (ver Apêndice A). Se a junta  $i$  for revolvente ( $i$  entre 1 e  $n$ ), então os erros generalizados  $\varepsilon_{z,(i-1)}$  e  $\varepsilon_{r,(i-1)}$  são eliminados, e seus valores são incorporados aos parâmetros de erro independentes  $\varepsilon_{y,i}^*$ ,  $\varepsilon_{z,i}^*$ ,  $\varepsilon_{s,i}^*$  and  $\varepsilon_{r,i}^*$ :

$$\begin{cases} \varepsilon_{y,i}^* \equiv \varepsilon_{y,i} + \varepsilon_{z,(i-1)} \sin \alpha_i + \varepsilon_{r,(i-1)} \cdot a_i \cos \alpha_i \\ \varepsilon_{z,i}^* \equiv \varepsilon_{z,i} + \varepsilon_{z,(i-1)} \cos \alpha_i - \varepsilon_{r,(i-1)} \cdot a_i \sin \alpha_i \\ \varepsilon_{s,i}^* \equiv \varepsilon_{s,i} + \varepsilon_{r,(i-1)} \sin \alpha_i \\ \varepsilon_{r,i}^* \equiv \varepsilon_{r,i} + \varepsilon_{r,(i-1)} \cos \alpha_i \end{cases} \quad \text{Eq. 2.5.7}$$

Se a junta  $i$  for prismática, os erros translacionais  $\varepsilon_{x,(i-1)}$  e  $\varepsilon_{y,(i-1)}$  são eliminados, e seus valores são incorporados aos parâmetros de erros independentes  $\varepsilon_{x,i}^*$ ,  $\varepsilon_{y,i}^*$  e  $\varepsilon_{z,i}^*$ . Neste caso, a Equação 2.5.7 passa a ser:

$$\begin{cases} \varepsilon_{x,i}^* \equiv \varepsilon_{x,i} + \varepsilon_{x,(i-1)} \\ \varepsilon_{y,i}^* \equiv \varepsilon_{y,i} + \varepsilon_{y,(i-1)} \cos \alpha_i + \varepsilon_{z,(i-1)} \sin \alpha_i + \varepsilon_{r,(i-1)} \cdot a_i \cos \alpha_i \\ \varepsilon_{z,i}^* \equiv \varepsilon_{z,i} - \varepsilon_{y,(i-1)} \sin \alpha_i + \varepsilon_{z,(i-1)} \cos \alpha_i - \varepsilon_{r,(i-1)} \cdot a_i \sin \alpha_i \\ \varepsilon_{s,i}^* \equiv \varepsilon_{s,i} + \varepsilon_{r,(i-1)} \sin \alpha_i \\ \varepsilon_{r,i}^* \equiv \varepsilon_{r,i} + \varepsilon_{r,(i-1)} \cos \alpha_i \end{cases} \quad \text{Eq. 2.5.8}$$

Se o vetor  $\varepsilon^*$  contendo os erros independentes for constante, a matriz  $G_e$  pode ser usada para substituir  $J_e$  na Equação 2.5.5, e a Equação 2.5.7 é utilizada para calcular um estimador para os erros generalizados independentes  $\varepsilon^*$ , completando o processo de identificação. Porém, se fatores não-geométricos forem considerados, são necessários mais parâmetros do modelo em  $\varepsilon^*$  como função da configuração do sistema anteriormente ao processo de identificação.

## 2.6.

### Representação Física dos Erros Redundantes

O conceito de erros redundantes pode ser melhor compreendido se visto fisicamente. Eles representam erros em juntas distintas que provocam o mesmo resultado na extremidade do manipulador.

A Equação 2.5.2 demonstra que um erro translacional no eixo Z da junta  $i-1$  possui o mesmo resultado na posição e orientação da extremidade que uma combinação de erros translacionais nos eixos Z e Y na junta  $i$  (ver figura 7).

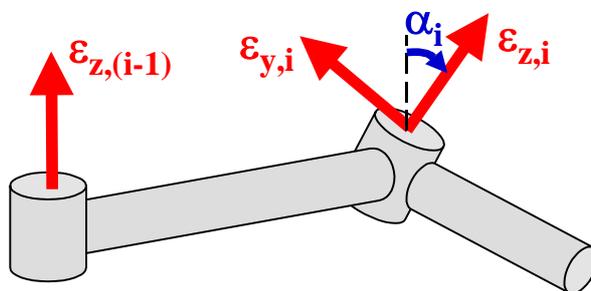


Figura 7: Combinação linear de erros translacionais

Outro exemplo ocorre quando um erro rotacional ao longo do eixo Z na junta  $i-1$  provoca o mesmo efeito que uma combinação de erro rotacional e translacional sobre os eixos Y e Z da junta  $i$ . Este resultado pode ser visto na Equação 2.3.3 e pode ser visto exemplificado na Figura 8.

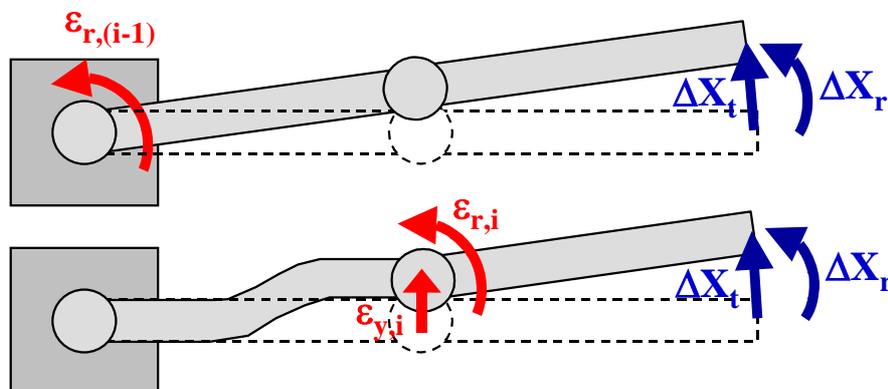


Figura 8: Combinação linear de erros translacionais

## 2.7.

### Medidas Parciais da Posição da Extremidade

Todas as equações obtidas até agora consideram que se tenham medidas completas da posição e orientação da extremidade do manipulador em diversas configurações. Porém conseguir mensurar as orientações da extremidade pode ser pouco prático e portanto vale a pena discutir como chegar aos mesmos resultados portando apenas da posição da extremidade. Neste caso, são formadas combinações lineares adicionais, tendo as três últimas colunas de  $\mathbf{J}_e$  como vetores nulos (Meggiolaro M, Apêndice A, 2000):

$$\mathbf{J}_{s,n} \equiv \mathbf{J}_{r,n} \equiv \mathbf{J}_{p,n} \equiv \mathbf{0} \quad \text{Eq. 2.6.1}$$

A Equação 2.6.1 exprime o fato de que os três erros rotacionais da extremidade,  $\varepsilon_{s,n}$ ,  $\varepsilon_{r,n}$  e  $\varepsilon_{p,n}$  não influenciam na posição da mesma (eles apenas afetam a orientação, as quais não estão sendo medidas). Como resultado, esses erros generalizados não são obtíveis.

Se a última junta for prismática, então nenhuma outra combinação linear é encontrada. Porém, se a última junta é revoluta e o tamanho deste elo  $a_n$  é zero, então mais três combinações lineares estão presentes:

$$\mathbf{J}_{s,(n-1)} \equiv d_n \mathbf{J}_{x,(n-1)} \quad \text{Eq. 2.6.2}$$

$$\mathbf{J}_{p,(n-1)} \equiv -d_n \mathbf{J}_{y,(n-1)} \quad \text{Eq. 2.6.3}$$

$$\mathbf{J}_{r,(n-1)} \equiv \mathbf{0} \quad \text{Eq. 2.6.4}$$

Significando que os efeitos de  $\varepsilon_{s,(n-1)}$  e  $\varepsilon_{p,(n-1)}$  não podem ser distinguidos daqueles causados por  $\varepsilon_{x,(n-1)}$  e  $\varepsilon_{y,(n-1)}$ , e também o erro generalizado  $\varepsilon_{r,(n-1)}$  não é obtível. Se ambos os tamanhos do elo  $a_n$  e a distância da junta  $d_n$  forem zero, então a origem das coordenadas  $n-1$  e  $n$  coincidem na extremidade. Nesse caso, as Equações 2.3.9-2.3.12 podem ser recursivamente aplicadas às coordenadas  $n-1$ ,  $n-2$ , e assim por diante, desde que a origem de todos os sistemas de coordenadas se mantenham na posição da extremidade.

## 2.8.

### Cinemática Inversa

A cinemática direta, encontrada a partir dos parâmetros de Denavit-Hartenberg, permite localizar a posição da extremidade do manipulador em relação à base. As matrizes de transformação encontradas, além disso, têm outra finalidade. A partir delas é possível encontrar a cinemática inversa do manipulador.

A cinemática inversa gera os ângulos de cada junta a partir da posição final da extremidade do manipulador. Com o uso da cinemática inversa passa a ser possível a automatização de tarefas, com um programa capaz de automaticamente calcular as posições de cada junta para movimentar a extremidade até a localização desejada.

Na cinemática direta, a posição da extremidade é determinada unicamente pelos deslocamentos das juntas. Na cinemática inversa, por outro lado, o problema passa a ser mais complexo uma vez que diversas soluções possíveis existem para uma mesma posição da extremidade, ou pode ocorrer de não existirem soluções

para um conjunto particular de posições da extremidade, considerando a estrutura do manipulador. Além destes fatores, uma vez que a equação cinemática é composta por diversas equações não-lineares formadas por funções trigonométricas, não é possível derivar uma solução geral, sendo a cinemática inversa produzida caso-a-caso a partir das equações da cinemática direta. Quando a solução para a cinemática inversa não pode ser calculada analiticamente, esta deve ser obtida por métodos numéricos.

Um braço robótico deve possuir pelo menos seis graus de liberdade para permitir a localização de sua extremidade em um ponto arbitrário e com uma orientação arbitrária no espaço. Manipuladores com menos que seis graus de liberdade podem não permitir tal posicionamento arbitrário. Por outro lado, se o manipulador possuir mais de seis graus de liberdade, existem infinitas possíveis soluções baseadas nas equações cinemáticas, sendo chamados manipuladores redundantes. Considere o exemplo do braço humano, o qual possui sete graus de liberdade, excluindo os dedos. Mesmo com a mão fixada numa parede, é possível mover o cotovelo continuamente sem causar mudanças na localização da mão nem do membro. Isto implica que existem infinitos conjuntos de deslocamentos de juntas que levam a mesma posição da mão.

### **2.8.1.**

#### **Solubilidade**

Nem sempre soluções fechadas para a cinemática inversa podem ser obtidas de forma analítica. Uma alternativa para o método analítico seria aplicar métodos numéricos baseados em algoritmos iterativos, como o Método de Newton-Raphson. Porém, o poder computacional necessário para tais métodos iterativos geralmente é muito maior que o necessário quando se possui uma solução-fechada. Devido a essa complexidade, métodos numéricos são geralmente considerados pouco práticos. Considere um manipulador necessitando mover sua extremidade por uma trajetória. O número de pontos ao longo da trajetória precisa ser transformado em deslocamento de juntas, necessitando portanto de rápida computação. Em particular, o tempo computacional é crucial se a transformação deve ser feita em tempo real.

A existência de soluções fechadas depende da estrutura cinemática do braço robótico. A estrutura cinemática do braço robótico é geralmente desenvolvida para que o problema da cinemática inversa seja possível de se resolver, para evitar complexidade computacional.

Portanto, uma questão importante é descobrir o que faz uma estrutura cinemática passível de solução. (Pieper, D.L., 1968) mostra que uma condição suficiente para que uma estrutura cinemática de um braço robótico de seis graus de liberdade tenha cinemática inversa analítica é que os eixos das juntas de três juntas revolutas consecutivas se interceptem em um único ponto para todas as configurações do braço.

Quando as três últimas juntas interceptam seus eixos em um único ponto, as três juntas são comumente referenciadas como pulso esférico. (Pieper, D. L., 1968) lista todas as possíveis estruturas cinemáticas para manipuladores de seis graus de liberdade que se encaixam neste caso. No capítulo 4 será discutida a cinemática inversa para o manipulador TA-40.

## 3

### Visão Computacional

#### 3.1.

##### Introdução

A calibração do manipulador TA-40 no seco permite que se tenha uma boa precisão na posição e orientação da extremidade do manipulador em relação à sua base, corrigindo os erros estáticos intrínsecos ao sistema. Em várias aplicações de manipuladores este nível de calibragem é suficiente, pois a base está sempre fixa na mesma posição. É o caso de um manipulador fixado no chão e que manipula objetos, os quais não mudam sensivelmente de posição.

A aplicação desta dissertação possui um desafio a mais para a calibração. O manipulador estará fixado sobre um robô submarino, que o levará até sua área de trabalho. Assim, dificilmente a base do manipulador estará no mesmo local a cada vez em que ele for levado para sua área de trabalho, pois o braço mecânico provavelmente não irá se atracar duas vezes na mesma exata posição no fundo do oceano, além de não haver garantias de que a base não se movimentará junto com o robô uma vez atracada. Logo, um método para se obter uma medida da posição da base do manipulador em relação à sua área de trabalho se faz essencial. Uma maneira consiste no uso de duas câmeras conectadas à garra do manipulador. Essa escolha é apropriada em virtude da redução do custo de câmeras na última década e pela existência de diversos algoritmos que permitirão um posicionamento com bastante precisão da base e dos objetos que serão manipulados.

Neste capítulo, será inicialmente feita uma breve descrição do modelo matemático da câmera e de sua calibração. Também será abordado um método para encontrar pontos-chave na imagem obtida por uma das câmeras e como correlacioná-las com pontos-chave encontrados nas imagens obtidas da outra câmera, procedimento este necessário para a obtenção da posição dos objetos no espaço tridimensional.

### 3.2.

#### Modelagem da Câmera

##### 3.2.1.

##### Modelo Pinhole

Um dos modelos de câmera mais difundidos é o *pinhole*, representado esquematicamente na Figura 9. Segundo este modelo, um raio de luz proveniente de um ponto  $M$  no espaço atravessa um orifício  $C$  e incide em um plano no ponto  $m$ . Dá-se o nome de plano de imagem ao plano de incidência dos raios, enquanto que o orifício  $C$  é chamado de centro da câmera. Um parâmetro intrínseco da câmera, chamado distância focal, pode ser medido pela distância do plano de imagem ao centro da câmera, e é geralmente representado por  $f$ .

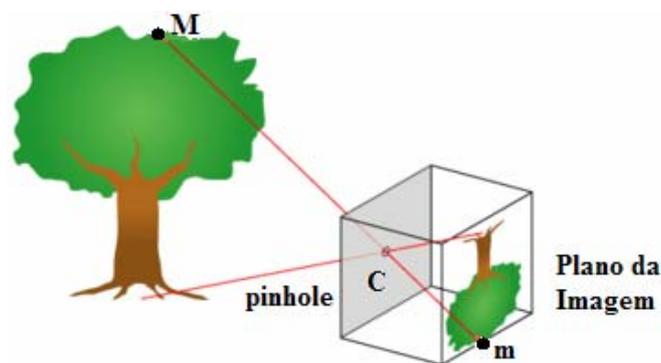


Figura 9: Representação do modelo pinhole.

A Figura 10 representa a geometria do modelo pinhole, tendo o quadro da câmera no espaço 3D com sua origem no orifício  $C$ . Também é representado o plano da imagem normalizado em um sistema de coordenadas em duas dimensões com origem no ponto  $\hat{C}$ . Por semelhança de triângulos, é possível demonstrar que o ponto  $P=(C_x, C_y, z_0)$  é mapeado no plano da imagem normalizado pelo ponto  $\hat{p}=(\hat{u}, \hat{v})=(fC_x/z_0, fC_y/z_0)$ . É possível visualizar os pontos  $P$  e  $\hat{p}$  em coordenadas homogêneas, pela seguinte equação:

$$\begin{cases} \hat{u} = \frac{C_x}{Z} \\ \hat{v} = \frac{C_y}{Z} \end{cases} \Leftrightarrow \hat{p} = \begin{bmatrix} \hat{u} \\ \hat{v} \\ 1 \end{bmatrix} = \frac{1}{z} \begin{bmatrix} Id & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} {}^c P \\ 1 \end{bmatrix} \quad \text{Eq. 3.2.1}$$

$$\text{Sendo } z = \frac{f}{z_0}.$$

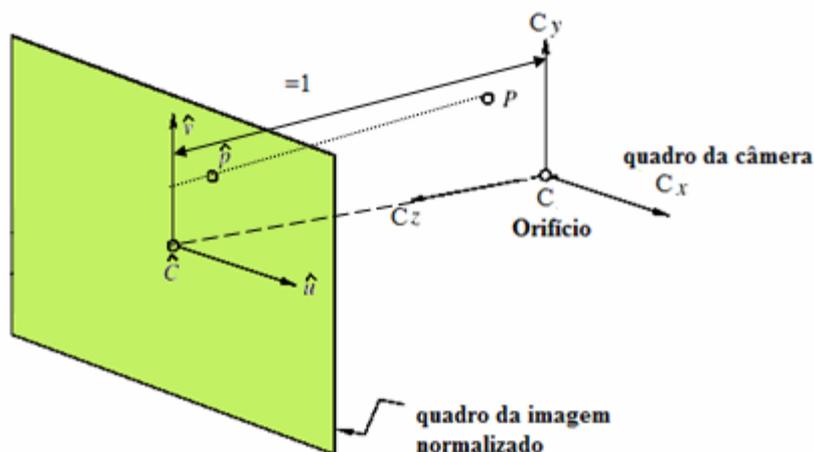


Figura 10: Geometria do modelo pinhole.

### 3.2.2.

#### Parâmetros Intrínsecos

Na Equação 3.2.1, supõe-se que a origem do sistema de coordenadas do plano da imagem é a origem da imagem, porém dependendo do programa empregado, a imagem que é digitalmente expressa como uma matriz, pode ter sua origem em uma posição diferente da do plano da imagem. O modelo pinhole também considera escalas iguais para os eixos x e y, o que não é sempre verdade para câmeras CCDs usadas na prática. A Figura 11 mostra a diferença entre o modelo pinhole e um modelo mais realista, onde os eixos x e y têm escalas distintas e as câmeras possuem cisalhamento de pixels.

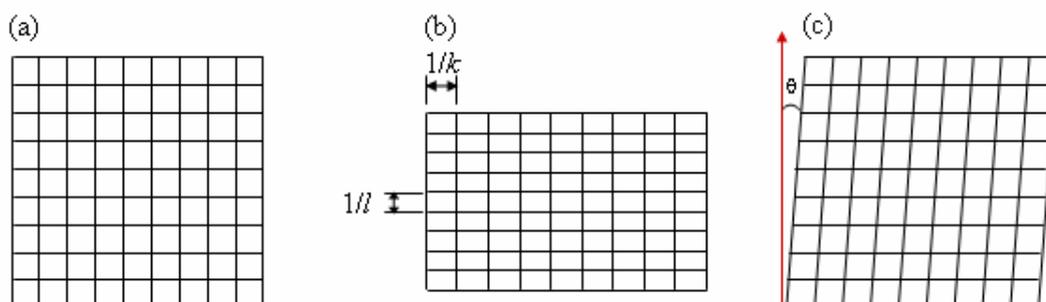


Figura 11: Modificações ao modelo pinhole com base na escala e cisalhamento dos pixels das câmeras. Figura (a) mostra os pixels segundo o modelo pinhole, figura (b) mostra um sistema mais realista onde escalas dos eixos x e y se diferenciam, figura (c) mostra o problema de cisalhamento

Além disto, a lente pode possuir distorções que precisam ser modeladas. Para se adequar a essa realidade, a Equação 3.2.1 deve ser alterada, criando-se na modelagem o quadro da retina física da imagem, após o quadro da imagem normalizada, conforme visto na Figura 12.

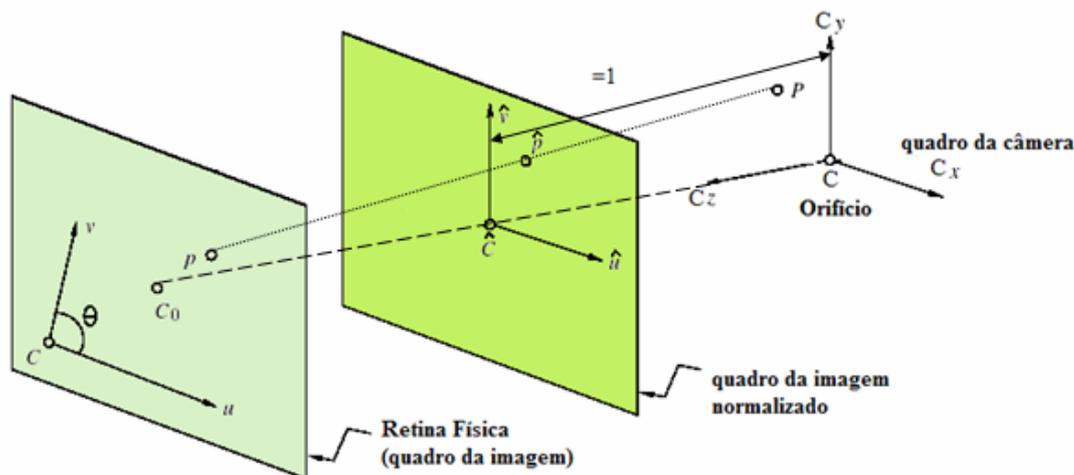


Figura 12: Modelo pinhole modificado.

Considerando-se que os pixels da câmera podem ter escalas diferentes nos eixos  $x$  e  $y$ , com  $1/k$  e  $1/l$  sendo as dimensões horizontal e vertical do pixel, tem-se:

$$\begin{cases} u = k f \hat{u} \\ v = l f \hat{v} \end{cases} \quad \text{Eq. 3.2.2}$$

Além disto, considerando-se que a origem do quadro da imagem não se encontra no centro do quadro, mas sim deslocado em  $(u_0, v_0)$ :

$$\begin{cases} u = \alpha \hat{u} + u_0 \\ v = \beta \hat{v} + v_0 \end{cases} \quad \text{Eq. 3.2.3}$$

, onde  $\alpha = kf$  e  $\beta = lf$

Por fim, considera-se que a matriz de sensores pode se encontrar distorcida de um ângulo  $\theta$ :

$$\begin{cases} u = \alpha \hat{u} - \alpha \cot \theta \hat{v} + u_0 \\ v = \beta \hat{v} + v_0 \end{cases} \quad \text{Eq. 3.2.4}$$

Matricialmente, pode-se definir uma matriz  $K$  dos parâmetros intrínsecos da câmera, onde:

$$\mathbf{p} = \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} \alpha & -\alpha \cot \theta & u_0 \\ 0 & \beta / \text{sen} \theta & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \hat{u} \\ \hat{v} \\ 1 \end{pmatrix} = \mathbf{K} \hat{\mathbf{p}} \quad \text{Eq. 3.2.5}$$

A matriz de parâmetros intrínsecos  $K$  transforma as coordenadas no quadro da imagem normalizado para o quadro da retina física da imagem. Por fim, pode-se utilizar esta matriz para mapear os pontos no quadro da câmera para o quadro da retina física da imagem por meio de matrizes homogêneas:

$$\mathbf{p} = \mathbf{K} \frac{1}{z} (\text{Id } \mathbf{0}) {}^c\mathbf{P} = \frac{1}{z} (\mathbf{K} \mathbf{0}) {}^c\mathbf{P} \quad \text{Eq. 3.2.6}$$

Na Equação 3.2.6, o ponto  ${}^c\mathbf{P}$  representa o ponto  $P$  em coordenadas homogêneas no sistema de coordenadas da câmera.

### 3.2.3.

#### Parâmetros Extrínsecos

Em geral, é mais conveniente expressar os pontos no espaço em termos de um ponto de referência no mundo, sendo o mundo definido como o quadro do espaço objeto. As coordenadas do mundo estão relacionadas com as coordenadas do centro da câmera por uma rotação e uma translação. Assim, se  ${}^w\mathbf{O}$  é o centro do sistemas de coordenadas do mundo e  ${}^c\mathbf{O}$  o centro do sistema de coordenadas da câmera, pode-se mapear a localização de um ponto  $P$  representado no quadro no mundo para o quadro da câmera. Este processo está representado na Figura 13.

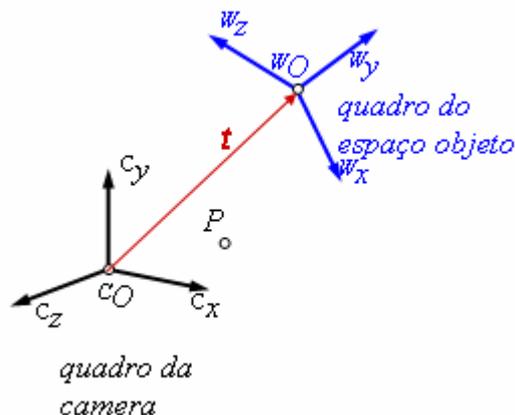


Figura 13: Transformação de coordenadas do mundo para a câmera.

Essa mudança de coordenadas pode ser descrita como uma transformação de matrizes, conforme a Equação 3.2.6:

$${}^c\mathbf{P} = \begin{pmatrix} {}^c\mathbf{R}_w & {}^c\mathbf{O}_w \\ \mathbf{0}^T & 1 \end{pmatrix} {}^w\mathbf{P} = \begin{pmatrix} {}^c\mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix} {}^w\mathbf{P} \quad \text{Eq. 3.2.7}$$

Os seis parâmetros das matrizes de rotação e translação são chamados de parâmetros extrínsecos da câmera, e mapeiam a posição de um ponto no sistema de coordenadas do mundo até a posição no sistema de coordenadas da câmera. Combinando as Equações 3.2.6 e 3.2.7, obtém-se um mapeamento completo de um ponto no quadro do mundo para o quadro da imagem.

$$\mathbf{p} = \frac{1}{z} (\mathbf{K} \mathbf{0}) \begin{pmatrix} {}^c\mathbf{R}_w & \mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix} {}^w\mathbf{P} = \frac{1}{z} \mathbf{K} \begin{pmatrix} {}^c\mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix} {}^w\mathbf{P} \quad \text{Eq. 3.2.8}$$

A matriz que representa o mapeamento é denominada Matriz de Projeção Perspectiva (M):

$$\mathbf{M} = \mathbf{K} \begin{pmatrix} {}^c\mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix} \rightarrow \mathbf{p} = \frac{1}{z} \mathbf{M} {}^w\mathbf{P} \quad \text{Eq. 3.2.9}$$

### 3.3.

#### Calibração de Câmera

Como visto na seção 5.2, um ponto no espaço 3D pode ser mapeado em um ponto no plano da imagem através da Equação (3.2.9). Contudo, a matriz de projeção da câmera ( $M$ ) deve ser conhecida. Essa matriz é uma característica particular de cada câmera e depende do processo de fabricação. Além disso, essa matriz pode ser alterada durante a execução normal do sistema por uma mudança na distância focal da câmera, e pode também variar de acordo com as condições do ambiente (variações mecânicas, térmicas). Por isso, a única forma de determinar a matriz de projeção corretamente é através da calibração da câmera.

A calibração da câmera é um procedimento distinto da calibração do manipulador. Enquanto que, no segundo caso, procura-se a matriz de erros de cada elo do manipulador, no caso da calibração de câmeras procura-se obter os parâmetros intrínsecos e extrínsecos da câmera. Como na aplicação serão utilizadas duas câmeras acopladas ao manipulador, a calibração tem como objetivo saber a posição de uma câmera em relação à outra.

Existem diversos métodos de calibração de câmeras presentes na literatura, sendo um processo bem testado, com diversos programas já implementados utilizando algoritmos diferenciados. A base dos algoritmos é a resolução de um sistema equações formado por diversos pontos coletados pelas câmeras, tendo como incógnitas os doze componentes da matriz de projeção  $M$ . A solução pode ser feita de forma linear ou não-linear. Não entra no escopo desta dissertação ir a fundo nos algoritmos, porém podem ser encontradas informações nas referências (Forsyth, D. A., pp.43-59, 2003).

É possível fazer o reconhecimento das posições de pontos no plano do mundo e relacioná-los facilmente com pontos no plano da câmera utilizando-se plataformas de calibragem como as mostradas na Figura 14. Conhecendo a distância dos lados dos quadrados, obtém-se facilmente a posição de cada ponto em relação ao outro.

Os resultados da calibragem levam a uma matriz de projeção que considera a origem do plano do mundo no centróide dos pontos observados na plataforma de calibração, conforme pode ser visto na Figura 15. Assim, a matriz

M considera uma translação e uma rotação em relação a essa origem, que pode não ser conveniente na prática.



Figura 14: Plataforma de Calibração

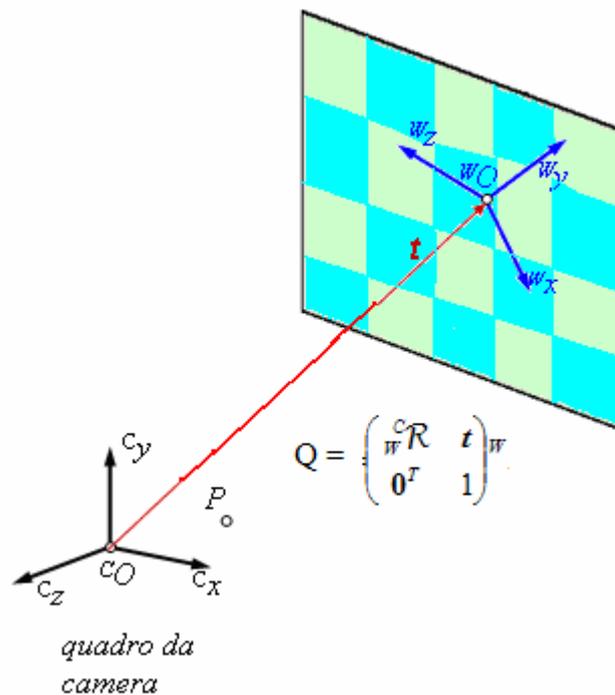


Figura 15: Transformação do quadro do mundo para o quadro da câmera

O manipulador robótico possuirá duas câmeras robóticas acopladas a sua extremidade, apresentando uma translação e rotação entre si. Esta translação e rotação serão futuramente importantes para posicionar os objetos vistos pelas câmeras e posicionar a base do manipulador em relação a pontos captados do ambiente de trabalho pelas câmeras. A calibração de câmeras será revista no capítulo 4 onde a aplicação das técnicas de visão computacional vai ser apresentada.

### 3.4.

#### Emparelhamento de Pontos

Um dos principais problemas de visão computacional é encontrar semelhanças em imagens. Tal problema ocorre em reconhecimento de objetos e cenas, resolução de estruturas 3D em múltiplas imagens, correspondência estéreo e busca de movimento. Na calibração da câmera, faz-se uso de uma plataforma de calibração que facilita o processo, mas no fundo do oceano o problema de emparelhar pontos em duas imagens distintas é bem mais complexo. A Figura 16 mostra como um mesmo objeto pode ser visto de formas diferentes por câmeras posicionadas de forma distinta.

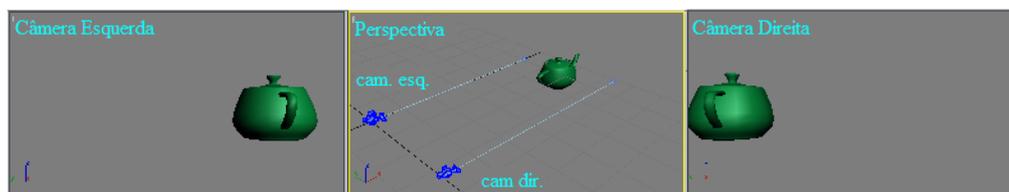


Figura 16: Visão de um mesmo objeto em câmeras distintas deslocadas por uma distância fixa.

Na aplicação, existirá um par de câmeras estéreo, vendo a mesma imagem a não ser por uma diferença de posição entre elas. Elas estarão sujeitas a ruídos na imagem causadas por um ambiente onde há pouca iluminação e distorções por refração. Para que seja possível obter a posição do manipulador em relação ao seu ambiente, será preciso encontrar pontos em ambas as imagens que sejam os mesmos no mundo real.

Nesta seção será apresentado um algoritmo para extrair pontos em uma imagem que possuem diversas propriedades que os habilitam a serem descobertos em outras imagens de um mesmo objeto ou cena. O método que será apresentado obtém, para cada ponto encontrado, propriedades robustas a variações na imagem, tais quais rotação, mudança de escala, iluminação e ponto de vista da câmera. Além disto, as propriedades obtidas dos pontos por esse método tem elevada robustez a ruído. O método é capaz de distinguir pontos diferentes a partir destas propriedades, com alta probabilidade de acerto. Essa metodologia é denominada Transformada de Propriedades Invariantes à Escala (“Scale Invariant Feature

Transform”, SIFT), pois transforma dados da imagem em coordenadas invariantes à escala e a outras características locais, como luminosidade e rotação.

Um aspecto importante desta metodologia é que ela gera um grande número de características que cobrem densamente a imagem sobre uma ampla gama de escalas e localizações. A quantidade de propriedades é particularmente importante para o reconhecimento de objetos, pois garante menor possibilidade de erros quando se faz a busca do mesmo objeto na nuvem de pontos de outras imagens. A seguir, será explicado de forma resumida cada estágio do algoritmo.

### 3.4.1.

#### Detecção de Extremos no Espaço de Escalas

A primeira etapa do algoritmo é buscar pontos que sejam invariantes a mudanças de escala da imagem, possibilitando a detecção de pontos com a câmera próxima ou distante do objeto de interesse. Tal objetivo é obtido procurando os pontos em todas as possíveis escalas, utilizando-se de uma função chamada de **espaço de escala**, que no caso é a função **gaussiana**. Assim, o espaço de escalas  $L(x,y,\sigma)$  é formado pela filtragem da imagem  $I(x,y)$  por funções gaussianas  $G(x,y,\sigma)$ , conforme visto abaixo:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad \text{Eq. 3.4.1}$$

Onde \* simboliza a convolução em x e y. A função gaussiana  $G(x,y,\sigma)$  é expressa por:

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \quad \text{Eq. 3.4.2}$$

A eficiência da busca por pontos-chaves é aumentada substituindo-se a função  $L(x,y,\sigma)$  por uma função de **Diferença-do-Gaussiano**, formada pela subtração de dois espaços de escala, separadas por um fator multiplicativo k, conforme mostrado na Equação 3.4.3. Esta função se aproxima da função **Laplaciano do Gaussiano** normalizado em escala  $\sigma^2 \Delta^2 G$ , que permite

(Lindeberg 1994) a possui pontos invariantes à escala em seus máximos e mínimos.

$$D(x, y, \sigma) = [G(x, y, k\sigma) - G(x, y, \sigma)] * I(x, y) = L(x, y, k\sigma) - L(x, y, \sigma) \quad \text{Eq. 3.4.3}$$

A Figura 17 mostra uma imagem submetida à funções  $L(x, y, \sigma)$  e  $D(x, y, \sigma)$ , com  $\sigma=2$ .



Figura 17: Imagem (à esquerda) submetida a gaussiana (no meio) e à diferença-do-gaussiano (à direita).

A Figura 18 demonstra o algoritmo para criação da diferença do gaussiano:

1- A imagem inicial sofre convoluções incrementais com Gaussianos para produzir imagens separadas por um fator de escala  $k$  no espaço de escala (empilhados na coluna esquerda)

2- Cada octava (grupo de cinco imagens que sofrem seguidas convoluções) do espaço de escala é dividida por um número inteiro,  $s$ , de forma que  $k = 2^{1/s}$ , produzindo assim  $s+3$  imagens na octava de forma que a detecção de extremos cubra toda octava.

3- Imagens em escalas adjacentes são subtraídas para produzir as imagens da diferença do gaussiano mostradas na direita.

4- Uma vez processada a octava, é reduzida a amostragem da imagem (*downsample*) com duas vezes o valor inicial de  $\sigma$  (duas imagens no topo da pilha), voltando-se ao passo 1.

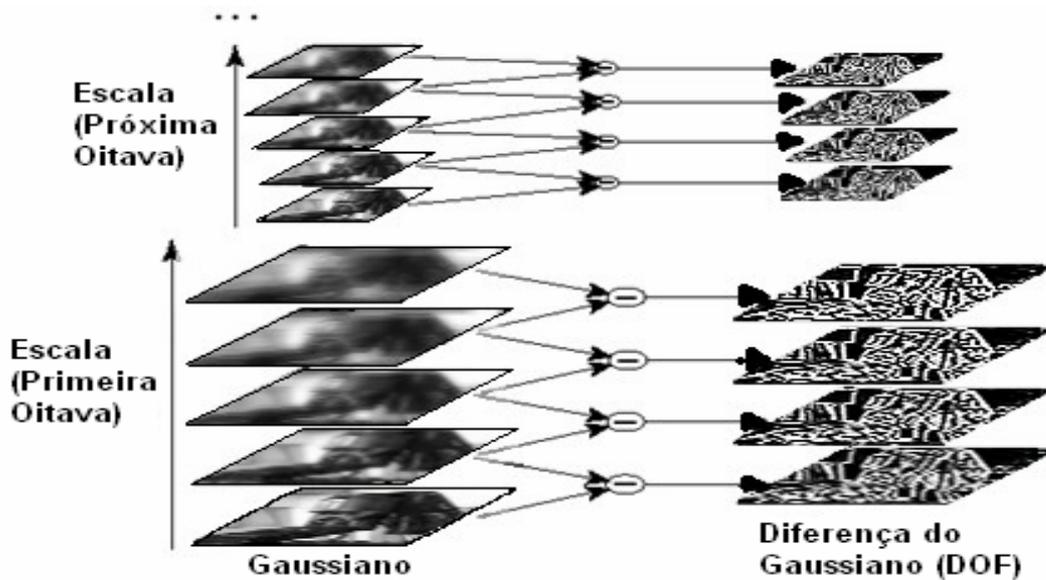


Figura 18: Representação do procedimento de obtenção de diferentes diferenças de gaussianas para diversas octavas da imagem.

Os pontos-chave são localizados nos máximos e mínimos locais de cada imagem  $D(x,y,\sigma)$ , que podem ser obtidos comparando-se a intensidade de cada ponto com as intensidades de seus seis pontos vizinhos na sua escala, e com os nove pontos vizinhos na escala acima e abaixo, como visto na Figura 19. Procuram-se pontos com valores de intensidade maiores ou menores que todos os seus vizinhos. É importante ressaltar que é possível encontrar um ponto máximo ou mínimo em uma imagem após ter sua amostragem reduzida, e portanto a sua posição nesta imagem reduzida não é equivalente à sua posição na imagem original, que possui mais pixels, sendo necessária uma interpolação para descobrir a posição na imagem original.

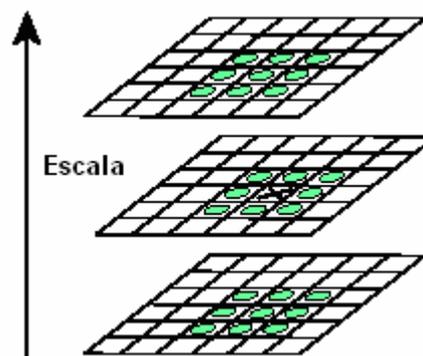


Figura 19: Máximo e mínimo das imagens de Diferença-do-Gaussiano são detectados comparando o pixel (marcado por X) com seus 26 vizinhos em 3x3 regions nas escalas corrente e adjacentes (marcados por círculos).

### 3.4.2.

#### Localização Precisa de Pontos-Chaves

Os pontos encontrados na seção anterior devem ser filtrados para retirar aqueles que possuem pouco contraste (sendo sensíveis a ruído) e aqueles próximos de extremidades. A eliminação de pontos sensíveis a ruídos é obtida utilizando uma expansão de Taylor sobre a função Diferença-do-Gaussiano deslocada de modo a ter sua origem no ponto de amostragem (Brown, M. and Lowe):

$$D(x) = D + \frac{\partial D^T}{\partial x} x + \frac{1}{2} x^T \frac{\partial^2 D}{\partial x^2} \quad \text{Eq. 3.4.4}$$

onde  $D$  e suas derivadas são calculadas no ponto de amostragem e  $x=(x,y,\sigma)^T$  é o deslocamento deste ponto. A localização do extremo,  $\hat{x}$ , é dada por:

$$\hat{x} = - \frac{\partial^2 D^{-1}}{\partial x^2} \frac{\partial D}{\partial x} \quad \text{Eq. 3.4.5}$$

O valor da função no extremo,  $D(\hat{x})$ , é útil para rejeição de extremos instáveis com baixo contraste. Pode-se obter este resultado substituindo a Equação 3.4.5 na Equação 3.4.4, descartando-se valores muito baixos de  $D(\hat{x})$  :

$$D(\hat{x}) = D + \frac{1}{2} \frac{\partial D^T}{\partial x} \hat{x} \quad \text{Eq. 3.4.6}$$

A eliminação de pontos-chaves próximos da extremidade é feita usando-se uma matriz Hessiana 2x2,  $H$ , na localização e escala do ponto-chave na função  $D$ . A Hessiana representa uma segunda derivada, permitindo mensurar o tamanho das curvaturas de  $D$  em seus autovalores. A curvatura próxima a pontos cujo gradiente é paralelo a extremidades possui baixa amplitude, mas em pontos com gradiente perpendicular a essas extremidades a amplitude é elevada.

$$H = \begin{bmatrix} D_{XX} & D_{XY} \\ D_{XY} & D_{YY} \end{bmatrix} \quad \text{Eq. 3.4.7}$$

Sendo  $\alpha$  o autovalor com maior magnitude e  $\beta$  o de menor magnitude, pode-se computar a soma e o produto destes autovalores:

$$\begin{aligned} Tr(H) &= D_{XX} + D_{YY} = \alpha + \beta \\ Det(H) &= D_{XX}D_{YY} - D_{XY}^2 = \alpha\beta \end{aligned} \quad \text{Eq. 3.4.8}$$

Caso o determinante seja negativo, as curvaturas têm sinais diferentes e o ponto é descartado, não sendo um extremo. Sendo  $r$  a razão entre o maior e o menor autovalor,  $\alpha=r\beta$ :

$$\frac{Tr(H)^2}{Det(H)} = \frac{(\alpha + \beta)^2}{\alpha\beta} = \frac{(r\beta + \beta)^2}{r\beta^2} = \frac{(r+1)^2}{r} \quad \text{Eq. 3.4.9}$$

A equação 3.4.9 depende apenas da razão entre os autovalores. A quantidade  $(r+1)^2/r$  oferece uma medida de quanto os autovalores são distintos. Eliminam-se pontos próximos a extremidades descartando-se pontos abaixo de um determinado *threshold*:

$$\frac{Tr(H)^2}{Det(H)} < \frac{(r+1)^2}{r} \quad \text{Eq. 3.4.10}$$

Assim, eliminam-se pontos-chave que não são estáveis, apesar de estarem próximos de extremidades.

### 3.4.3.

#### Atribuição de Orientação

A invariância ao ruído e à escala já são um passo importante para os pontos-chave, porém eles devem ser invariantes à rotação da câmera. Para se obter essa invariância, atribui-se uma orientação baseada nas características locais da imagem.

Para tanto, calcula-se para cada amostragem da imagem,  $L(x,y)$ , na escala do ponto-chave, a magnitude,  $m(x,y)$  e a orientação  $\theta(x,y)$  do gradiente usando-se diferenças de pixels.

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \quad \text{Eq. 3.4.11}$$

$$\theta(x, y) = \tan^{-1} \left( \frac{(L(x, y+1) - L(x, y-1))}{(L(x+1, y) - L(x-1, y))} \right) \quad \text{Eq. 3.4.12}$$

Na região vizinha ao ponto-chave, forma-se um histograma de orientações composto pelas orientações dos pontos amostrados. Esse histograma possui 36 pontos, cobrindo os 360 graus de orientação, conforme visto na figura 20.

O pico deste histograma é utilizado para definir a sua orientação. No caso de múltiplos picos de elevada amplitude, o ponto-chave receberá múltiplas orientações, tornando-o ainda mais estável para futura identificação. Ao final, uma parábola é usada para interpolar os três valores do histograma mais próximos do pico, de forma a se ter uma melhor precisão de sua posição.

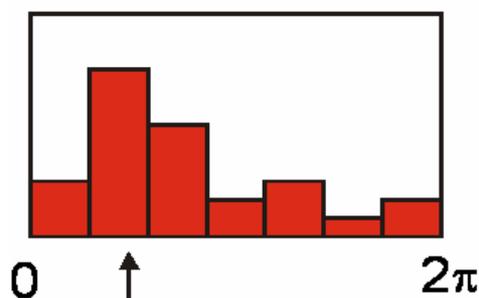


Figura 20: Histograma de orientação do ponto-chave.

#### 3.4.4.

#### Descritor Local da Imagem

As seções anteriores apresentaram procedimentos para posicionamento de pontos-chaves, e atribuir-lhes escala, resistência a ruído e orientações, permitindo a imposição de um sistema de coordenadas 2D na qual descrever uma região da imagem localmente, e prover invariância a esses parâmetros. O próximo passo será de atribuir a esses pontos-chave descritores invariantes a mudanças de iluminação e ponto de vista 3D, além de torná-los bem distinguíveis.

As magnitudes e orientações dos gradientes são amostradas ao redor da localização do ponto-chave, utilizando a escala do ponto-chave para escolher o nível de suavização Gaussiano para a imagem.

Para que os descritores tenham invariância a rotação, as coordenadas e as orientações dos descritores são rotacionadas relativo a orientação do ponto-chave. Este procedimento está mostrado pelas pequenas setas em cada amostra da localização no lado esquerdo da Figura 21.

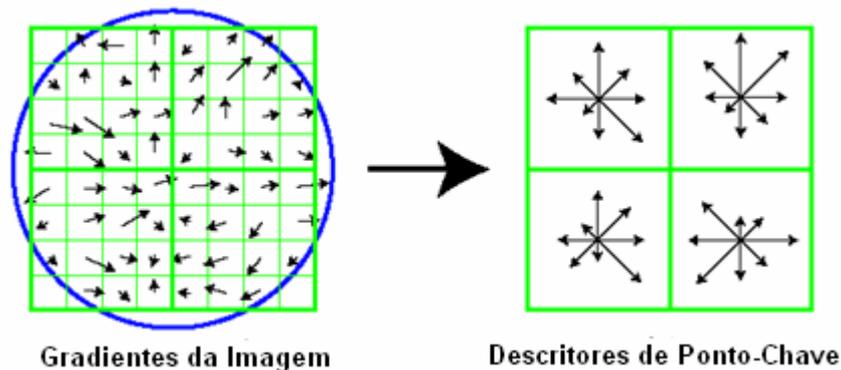


Figura 21: Ilustração da computação de descritores de pontos-chaves

Uma janela de suavização gaussiana de  $\sigma$  igual à metade da largura da janela do descritor é usada para dar pesos à magnitude do gradiente em cada ponto amostrado, conforme visto na esquerda da Figura 18. Esse gaussiano evita mudanças súbitas do descritor com pequenas mudanças na posição da janela e também reduz a ênfase nos gradientes longe do centro do descritor, que são mais afetados por erros.

Uma vez feita a suavização dos gradientes, criam-se histogramas de orientação sobre 4x4 regiões de amostragem. O lado direito da Figura 18 mostra os quatro histogramas formados pela amostragem da esquerda. Na figura estão representadas oito direções para cada histograma e o tamanho de cada seta representa a magnitude referente a respectiva entrada do histograma.

Para evitar efeitos de borda na qual o descritor abruptamente muda enquanto a amostra se desloca suavemente indo de um histograma para outro ou de uma orientação para outra, uma interpolação trilinear é usada para distribuir o valor de cada amostra de gradiente para a sua posição no histograma adjacente.

O descritor é formado de um vetor contendo os valores de todas as orientações dos histogramas, correspondentes aos tamanhos das setas no lado direito da Figura 18.

### 3.4.5.

#### **Invariância à Luminosidade**

Duas imagens de um mesmo objeto podem possuir variações de luminosidade que modifiquem os descritores obtidos sensivelmente. Devido a este motivo deve-se fazer um tratamento dos descritores para torná-los menos sensíveis a essas variações.

Os descritores são invariantes a mudanças homogêneas de brilho da imagem, uma vez que esta variação representa uma adição de todos os pixels da imagem por uma constante e os descritores são calculados por diferenças de pixels. Quanto a mudanças homogêneas de contraste, representada pela multiplicação de todos os pixels por uma constante, é corrigida normalizando os descritores.

Variações não-lineares, causadas quer por saturação das câmeras, quer por efeito de iluminação de superfícies tridimensionais em diferentes orientações, podem provocar elevada influência sobre as magnitudes dos descritores, mas pouco influenciam na orientação. Reduz-se este efeito impondo um valor máximo às magnitudes. Assim sendo, reduz-se a influência da magnitude na correspondência dos pontos em imagens distintas e aumenta-se a importância dos descritores de orientação.

### 3.4.6.

#### **Relacionamento de Pontos-Chaves em Imagens Distintas**

Uma vez aplicado o algoritmo SIFT sobre duas imagens distintas do mesmo objeto, o algoritmo retornará um conjunto de pontos-chaves em cada imagem e seus respectivos descritores. Se faz necessário relacionar os pontos de forma robusta. Diversos métodos possíveis são buscar pontos com menor distância euclidiana, distância de *Mahalanobis*, máxima probabilidade, entre outros. O procedimento escolhido consiste no uso de transformada Hough (Ballard D. , 1982) para agrupar os pontos a partir de suas propriedades (localização, orientação e descritores) em um espaço de transformação 2D, e em seguida utilizar o algoritmo RANSAC (Random Sample Consensus) (Fischler M. A., 1981) para melhorar as estimativas.

A transformada de Hough é um método usualmente utilizado para detectar linhas ou curvas especificadas por uma função paramétrica. Ela cria uma matriz acumuladora  $n$ -dimensional onde acumulam-se “votos” para os parâmetros corretos das linhas e curvas encontradas na imagem.

O uso da transformada Hough para relacionar pontos é parecida. Cada ponto-chave especifica quatro parâmetros: localização 2D, escala e orientação. Sendo assim, cria-se uma matriz acumuladora (espaço de transformada 2D) em translações ( $t_1$ ,  $t_2$ ), log escala ( $\log s$ ) e rotação ( $\theta$ ). Discretizam-se os tamanhos dos eixos de com um oitavo do tamanho da imagem para translação, um oitavo de escala para escala e um oitavo de radianos para rotação.

A transformada Hough identifica agrupamentos de propriedades de pontos-chaves usando cada propriedade de um ponto de uma imagem para votar entre todos os outros pontos da outra imagem aquele que é mais consistente.

Os resultados da transformada Hough são refinados usando RANSAC. Este método tem a vantagem de ser altamente insensitivo a pontos anormalmente distantes da maioria dos pontos amostrados, mas falhará se este grupo de pontos anormalmente distantes for muito grande. Por isso é usado Hough antes para agrupar as propriedades.

### 3.5.

#### Reconstrução

O objetivo final do uso de câmeras na calibração do braço mecânico é de obter as posições de pontos identificados nas imagens nas coordenadas reais deles no mundo. Tendo-se esta posição dos pontos, é possível descobrir a posição do manipulador em relação à sua área de trabalho para que ele possa trabalhar automaticamente.

Este processo chama-se reconstrução e dentre diversos algoritmos de reconstrução presentes na literatura, o escolhido foi o algoritmo de Triangulação. Outros métodos como Estrutura do Movimento e Geometria Epipolar foram testados mas trouxeram poucos resultados positivos.

O método da triangulação é puramente geométrico. Tendo-se um par de câmeras calibradas e um conjunto de pontos  $p$  na primeira imagem corretamente relacionado a pontos  $p'$  na segunda imagem, e sendo a posição das câmeras  $O$  e

$O'$  respectivamente, encontra-se a posição do ponto no espaço 3D na interseção dos raios  $R=Op$  e  $R'=O'p'$ , conforme visto na Figura 22. Na prática esta interseção não ocorre devido a erros de calibração e de localização dos pontos. Para solucionar este problema se calcula o ponto médio do segmento que mais aproxima os dois raios  $R$  e  $R'$ .

$$\overrightarrow{OP_1} + \overrightarrow{P_1P_2} + \overrightarrow{P_2O'} = \overrightarrow{OO'} \quad \text{Eq. 3.4.13}$$

Expandindo a Equação 3.4.13 com a posição relativa das câmeras:

$$z\hat{p} + c(\hat{p}xR\hat{p}') + z'R\hat{p}' = t \quad \text{Eq. 3.4.14}$$

Onde  $R$  é a matriz de rotação entre as câmeras e  $t$  é o vetor de translação. Substituindo em 3.4.14 os parâmetros intrínsecos ( $K$  e  $K'$ ):

$$zK^{-1}p + c(K^{-1}pxRK'^{-1}p) + z'RK'^{-1}\hat{p}' = t \quad \text{Eq. 3.4.15}$$

Resolvendo o sistema da Equação 3.4.15 encontra-se os parâmetros  $z$  e  $c$ . O ponto  $P$  é obtido pela equação:

$$P = zK^{-1}p + c(K^{-1}pxRK'^{-1}p)/2 \quad \text{Eq. 3.4.16}$$

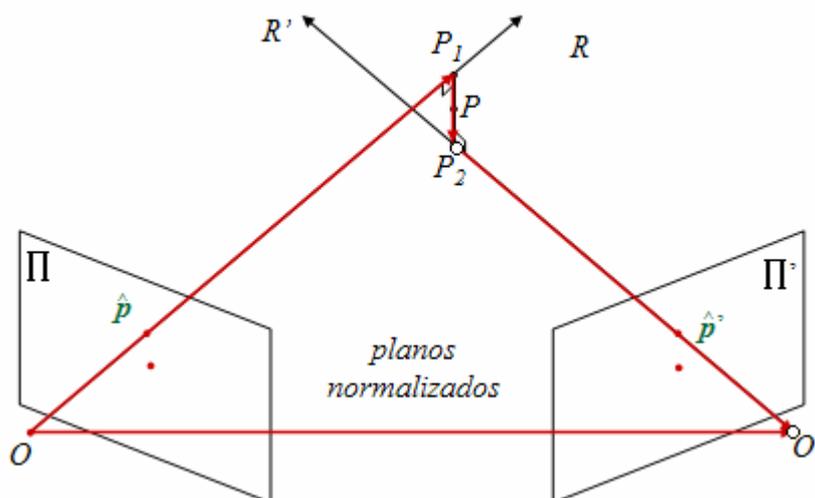


Figura 22: Triangulação

## 4

### Aplicação ao Manipulador TA-40

#### 4.1.

##### Introdução

Nos capítulos anteriores, foram discutidas diversas técnicas e algoritmos para a calibração de um manipulador robótico de forma genérica. Este capítulo tem como objetivo aplicar essas técnicas em um caso real.

O manipulador TA-40 é um braço robótico utilizado pela PETROBRAS em intervenções submarinas. Ele atua em profundidades que impossibilitam a intervenção humana e funciona fixado sobre um robô submarino (ROV) que o transporta até a sua área de trabalho. Atualmente, controlado por tele-operação, o manipulador possui pouca precisão e, gerando um alto grau de incerteza e perda de repetibilidade necessárias à automação. A calibração deste manipulador é não apenas uma grande oportunidade de estudo mas trará resultados positivos na exploração de petróleo, permitindo automatizar procedimentos.

Neste capítulo, será inicialmente feita uma breve descrição do manipulador TA-40, cuja compreensão é indispensável para o procedimento. Em seguida, serão obtidos o seu modelo cinemático, sua cinemática inversa e as matrizes de identificação de erro. Segue uma demonstração que simula o uso do algoritmo SIFT atuando em imagens vistas pelo manipulador, bem como a correlação de pontos. Por fim, todos os algoritmos são unidos de forma a se ter uma calibração completa do manipulador.

## 4.2.

### Descrição do Manipulador

O manipulador híbrido TA-40 é um braço robótico controlado capaz de levantar uma carga máxima de 210kg a 1950 mm de alcance. Ele está configurado com angulação de ombro, do cotovelo, rotação do braço inferior, angulação do punho, rotação de garra e abertura e fechamento de garra. A movimentação é dirigida por válvulas hidráulicas, atuadores rotatórios e motores hidráulicos, possuindo seis graus de liberdade.

Criado para atuar em ambientes hostis ao ser humano, o TA-40 é capaz de operar a profundidades de 3000 metros e sob efeito de radiação. Sua operação é feita por meio da manipulação de uma miniatura do manipulador, o braço mestre, visualizada na Figura 23.

O procedimento adotado atualmente no uso do TA-40 em intervenções submarinas algumas vezes requer a construção de uma réplica (maquete) da tarefa a ser desenvolvida. Quando uma trajetória para a tarefa é descoberta, o TA-40 é deslocado para sua área de trabalho por meio de um ROV e sua tarefa é executada.

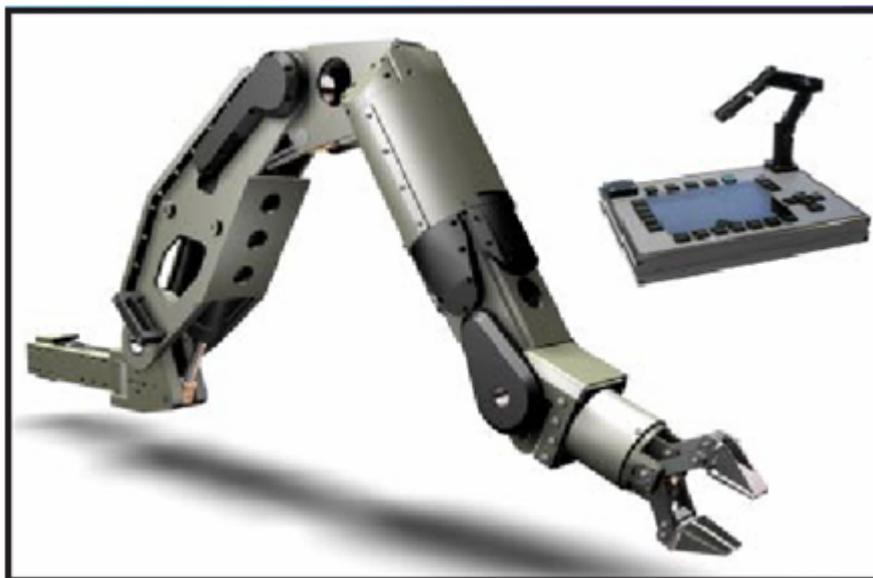


Figura 23: TA40 e miniatura utilizada como “mestre” na tele-operação

Com a calibração do TA-40, a solução para um problema que necessite de uma intervenção submarina pode ser desenvolvida “*offline*”, em um ambiente virtual, reduzindo assim o tempo de desenvolvimento de uma solução e reduzindo custos.

### 4.3.

#### Cinemática do Manipulador

A cinemática do manipulador é pré-requisito para a calibração, quer seja na obtenção da matriz de erros quer seja na calibração da base. A teoria necessária para compreensão desta sessão está no capítulo 2, mais precisamente em 2.5. A Figura 24 mostra o manipulador com seus seis sistemas de coordenadas. A seguir obtêm-se parâmetros de Denavit-Hartenberg para cada sistema de coordenadas.

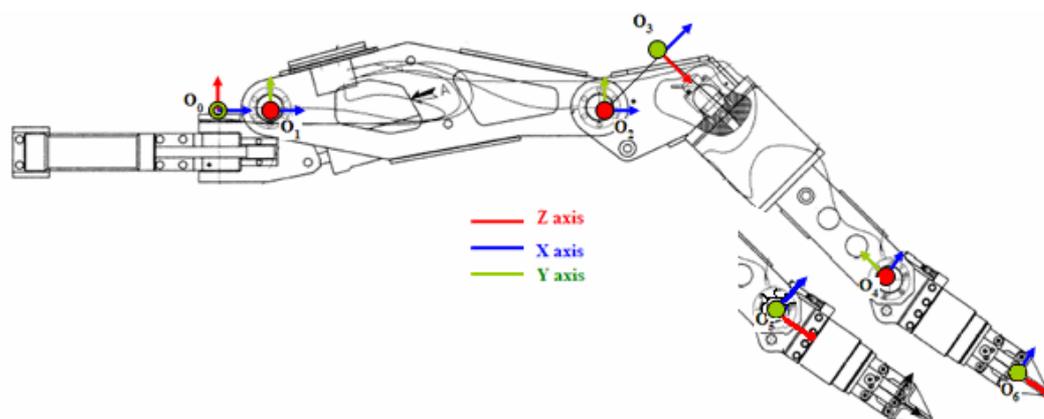


Figura 24: TA-40 e seu sistema de coordenadas

#### 4.3.1.

##### Juntas 1 e 2

A Figura 25 mostra as juntas 1 e 2 com seus respectivos sistemas de coordenadas. A junta 1 está na base do manipulador. Seu eixo  $z_0$  está sobre o eixo de rotação e o eixo  $x_0$  e  $x_1$  seguem a normal comum entre as juntas 1 e 2 (linha pontilhada). A normal comum entre  $z_0$  e  $z_1$  possui 115 milímetros ( $a_1=115$ ). Além disto, o eixo  $z_1$  está rotacionado em  $90^\circ$  no sentido horário em relação ao eixo  $z_0$ , sendo o parâmetro ( $\alpha_1=-90^\circ$ ). A distância entre o eixo de coordenadas  $O_0$  e a normal comum é nula ( $d_1=0$ ).

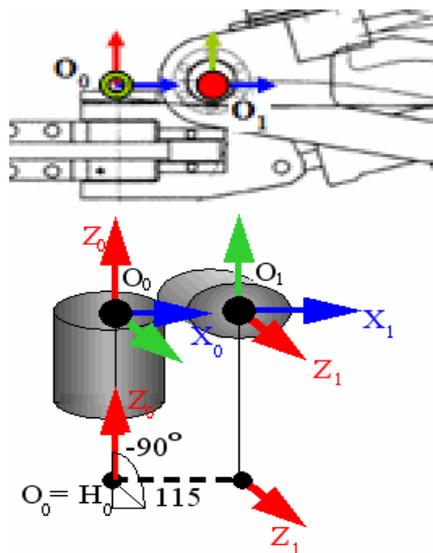


Figura 25: Juntas 1 e 2

#### 4.3.2.

#### Juntas 2 e 3

A Figura 26 mostra as juntas 2 e 3 com seus respectivos sistemas de coordenadas. O eixo  $z_1$  está sobre o eixo de rotação e o eixo  $x_2$  segue a normal comum entre as juntas 2 e 3 (linha pontilhada). A normal comum entre  $z_1$  e  $z_2$  possui 753 milímetros ( $a_2=753$ ). Os eixos  $z_1$  e  $z_2$  têm a mesma orientação e sentido ( $\alpha_2=0^\circ$ ). A distância entre o eixo de coordenadas  $O_1$  e a normal comum é nula ( $d_2=0$ ).

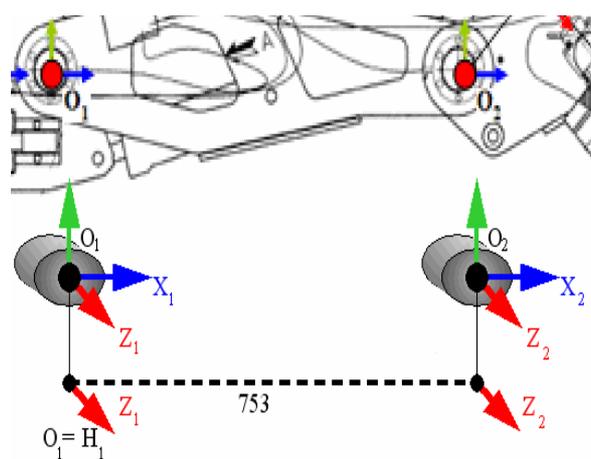


Figura 26: Juntas 2 e 3

### 4.3.3.

#### Juntas 3 e 4

Na Figura 27 (juntas 3 e 4), o eixo  $z_2$  está sobre o eixo de rotação e o eixo  $x_3$  segue a normal comum entre as juntas 3 e 4 (linha pontilhada). A normal comum entre  $z_2$  e  $z_3$  possui 188 milímetros ( $a_2=188$ ). A posição do eixo  $O_3$  se encontra fora dos limites reais do manipulador. Existe uma rotação horária de  $90^\circ$  entre os eixos  $z_2$  e  $z_3$  ( $\alpha_3=90^\circ$ ). A distância entre o eixo de coordenadas  $O_2$  e a normal comum é nula ( $d_3=0$ ).

### 4.3.4.

#### Juntas 4 e 5

Na Figura 28 (juntas 4 e 5), o eixo  $z_3$  está sobre o eixo de rotação, e seu sentido vai em direção ao eixo  $z_4$ , o que permite a escolha da posição da normal comum em qualquer ponto da linha pontilhada que une os dois eixos. Foi escolhida a posição da normal comum sobre o sistema de coordenadas  $O_4$  a 747 milímetros de distância ao sistema  $O_3$  ( $d_4=747$ ); a normal comum possui tamanho nulo ( $a_4=0$ ), o que permite a escolha da direção de  $x_4$  igual à direção de  $x_3$ , facilitando os cálculos. Existe uma rotação anti-horária de  $90^\circ$  entre os eixos  $z_3$  e  $z_4$  ( $\alpha_4=-90^\circ$ ).

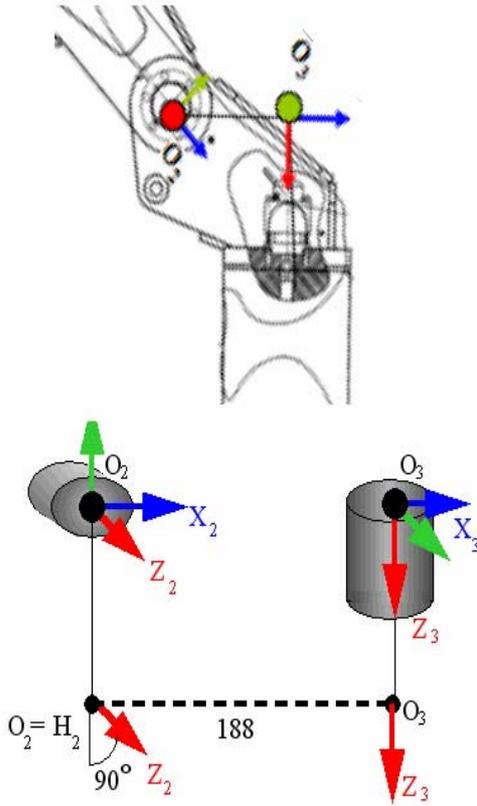


Figura 27: Juntas 3 e 4

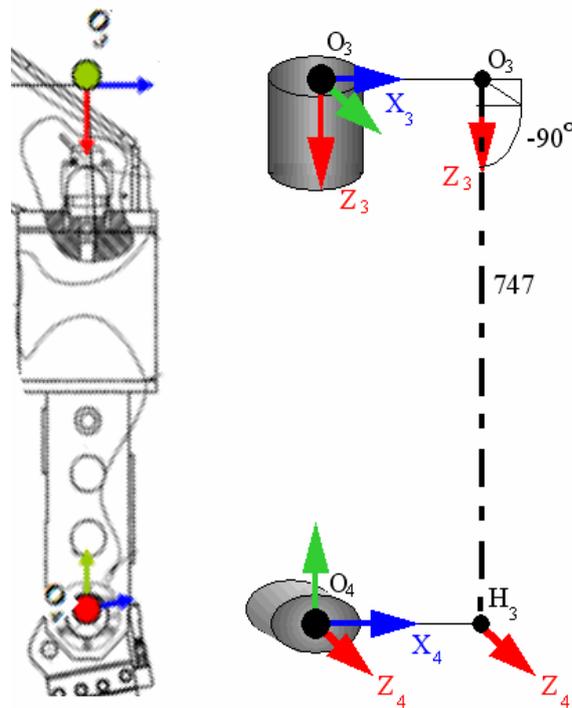


Figura 28: Juntas 4 e 5

#### 4.3.5.

#### Juntas 5 e 6

A Figura 29 mostra as juntas 5 e 6. Ambos os eixos estão no mesmo ponto (a distância entre eles é nula) nas direções de seus respectivos eixos de rotação e não há normal comum ( $d_5=0$ ,  $a_5=0$ ). O eixo  $x_5$  pode ter sua direção escolhida de forma a facilitar os cálculos. Há uma rotação de  $90^\circ$  entre o eixo  $z_4$  e o eixo  $z_5$  ( $\alpha_5=90^\circ$ ).

#### 4.3.6.

#### Juntas 6

A Figura 30 mostra as junta 6. O eixo  $z_5$  está sobre o eixo de rotação, e seu sentido vai em direção ao eixo  $z_6$ , o qual teve a orientação escolhida de modo que o eixo  $z_6$  tivesse a mesma orientação que o eixo  $z_5$ . A orientação dos eixos permite a escolha da posição da normal comum em qualquer ponto da linha pontilhada que une os dois eixos. Foi escolhida a posição da normal comum sobre o sistema de coordenadas  $O_6$  a 360 milímetros de distância ao sistema  $O_5$  ( $d_6=360$ ); a normal comum possui tamanho nulo ( $a_6=0$ ). Não existe rotação entre os eixos  $z_5$  e  $z_6$  ( $\alpha_6=0^\circ$ ).

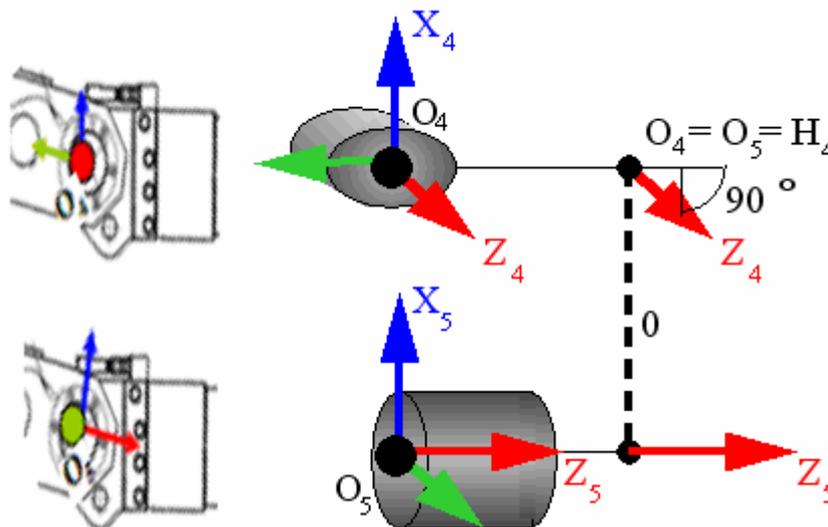


Figura 29: Juntas 5 e 6

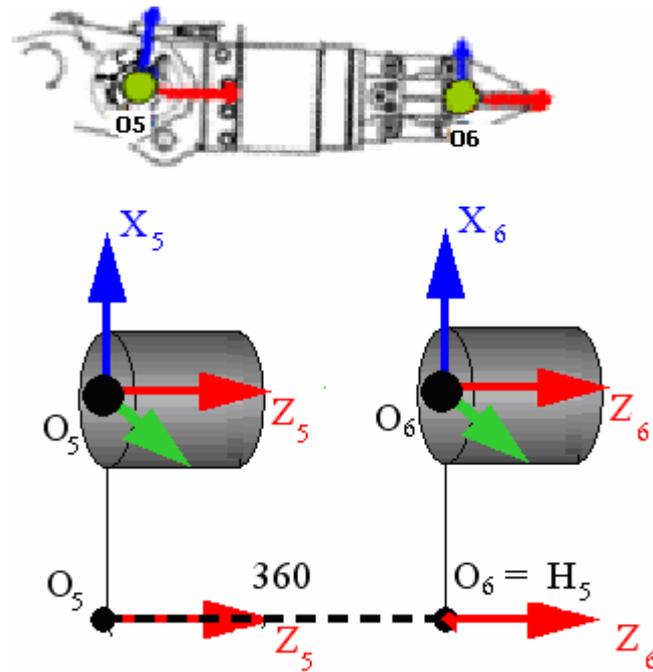


Figura 30: Junta 6

#### 4.3.7.

#### Parâmetros de Denavit-Hartenberg

A Tabela 1 apresenta todos os parâmetros de Denavit-Hartenberg (DH). A partir deles, pode-se criar as matrizes  $A_i$  que ligam uma junta à seguinte (Equação 4.3.3) e a partir da Equação 4.4.1, obter a posição da extremidade do manipulador.

Tabela 1: Tabela de Parâmetros DH

$Elo\ i$	$a_i$	$d_i$	$\alpha_i$	$\theta_i$
1	115	0	90	$\theta_1$
2	753	0	0	$\theta_2$
3	188	0	90	$\theta_3$
4	0	747	-90	$\theta_4$
5	0	0	90	$\theta_5$
6	0	360	0	$\theta_6$

#### 4.4.

#### Calibragem do Manipulador

Esta seção demonstrará a aplicação do método explicitado na seção 2.4, onde se busca a matriz de erros do manipulador baseado no cálculo de sua matriz jacobiana de identificação. Conforme visto, o manipulador possui erros, geralmente originários de tolerâncias de fabricação, que são pequenos em cada junta, cuja acumulação provoca um grande erro na extremidade. Estes erros em cada junta podem ser explicitados pela Equação 4.4.1

$$E_i = \begin{bmatrix} 1 & \varepsilon_p & \varepsilon_r & \varepsilon_x \\ \varepsilon_p & 1 & \varepsilon_s & \varepsilon_y \\ \varepsilon_r & \varepsilon_s & 1 & \varepsilon_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Eq. 4.4.1}$$

O primeiro passo para identificar os erros é calcular a matriz jacobiana de identificação. Para tal é necessária a cinemática direta do manipulador, calculada na seção 4.3. Adicionando-se os erros à Equação 2.3.1, encontra-se a Equação 2.4.3, que, para o TA-40, é composta por 12 matrizes:

$$T_6^0(q, \varepsilon) = E_0 A_0 E_1 A_1 E_2 A_2 E_3 A_3 E_4 A_4 E_5 A_5 E_6 A_6 \quad \text{Eq. 4.4.2}$$

A matriz jacobiana de identificação é encontrada derivando-se a Equação 4.4.2 em relação a cada um dos erros, e em seguida igualando todos os erros da matriz resultante a zero. Este último procedimento visa reduzir a complexidade de cálculos posteriores, considerando-se os erros individuais em cada junta muito pequenos.

$$J_e = \frac{\partial T_6^0(q, \varepsilon)}{\partial \varepsilon_{p0}, \partial \varepsilon_{r0}, \partial \varepsilon_{s0}, \partial \varepsilon_{x0}, \partial \varepsilon_{y0}, \partial \varepsilon_{z0}, \dots, \partial \varepsilon_{p6}, \partial \varepsilon_{r6}, \partial \varepsilon_{s6}, \partial \varepsilon_{x6}, \partial \varepsilon_{y6}, \partial \varepsilon_{z6}} \quad \text{Eq. 4.4.3}$$

A matriz jacobiana resultante possui 42 colunas e está explicitada no Apêndice A. Esta matriz foi calculada de forma que se utilizem apenas medidas

da posição da extremidade, ignorando sua orientação. Assim, como esperado, as três últimas colunas são nulas. Encontrada a matriz jacobiana, conforme explicado no capítulo 2, as colunas linearmente dependentes devem ser retiradas. Os erros generalizados  $\varepsilon_{z,(i-1)}$  e  $\varepsilon_{r,(i-1)}$ , para  $i=1..6$ , são eliminados, e seus valores são incorporados aos parâmetros de erro independentes  $\varepsilon_{y,b}^*$ ,  $\varepsilon_{z,b}^*$ ,  $\varepsilon_{s,i}^*$  e  $\varepsilon_{r,i}^*$ . Também são retiradas as três últimas colunas, referentes à orientação da extremidade. A matriz jacobiana reduzida resultante ( $G_e$ ) possui 25 colunas, tendo sido reduzida em 17 colunas.

Foi desenvolvido um teste, no qual se calculava a posição da extremidade do manipulador, em diversas configurações possíveis, sob efeito de erros generalizados em todas as juntas. Em seguida, estes erros foram calculados a partir da posição da extremidade obtida e da matriz  $G_e$ . Os erros generalizados calculados foram iguais aos erros utilizados para calcular a posição da extremidade do manipulador, porém a diferença entre os erros estavam substancialmente elevados nos erros generalizados  $e_{x6}, e_{y6}, e_{z6}, e_{r6}, e_{p6}$  e  $e_{s6}$ .

O resultado foi provocado pela existência de duas colunas linearmente dependentes, que provocou singularidades na inversão da matriz  $G_e$ . Estas colunas linearmente dependentes não foram reduzidas pelo método analítico devido ao fato de terem sido causadas pelo uso apenas da posição da extremidade do manipulador, ignorando a orientação. Para que a matriz  $G_e$  seja inversível, deve-se retirar tais colunas linearmente dependentes.

Neste caso particular, onde a orientação do manipulador não é considerada, a Equação 2.6.1 implica que os erros generalizados  $\varepsilon_{s,n}$ ,  $\varepsilon_{r,n}$  e  $\varepsilon_{p,n}$  não afetam medidas da extremidade. Portanto, se a última junta é revoluta (como no caso do TA-40) e  $a_n \neq 0$  então a seguinte equação deve ser utilizada para compensar a jacobiana (Meggiolaro, Apêndice A, 2000):

$$\begin{cases} \varepsilon_{x,(n-1)}^* = \varepsilon_{x,(n-1)} + \varepsilon_{s,(n-1)} \cdot d_n \\ \varepsilon_{y,(n-1)}^* = \varepsilon_{y,(n-1)} - \varepsilon_{p,(n-1)} \cdot d_n \end{cases} \quad \text{Eq. 4.4.4}$$

O que implica que  $\varepsilon_{r,(n-1)}$  não afeta a posição da extremidade.

## 4.5.

### Cinemática Inversa

A cinemática inversa é calculada caso-a-caso, baseada na configuração do manipulador, tendo como seu pré-requisito o cálculo de sua cinemática direta. Como o manipulador possui os eixos de suas três últimas juntas (juntas 4, 5 e 6) se encontrando num único ponto, a solução analítica é possível de ser encontrada.

Considere-se a Equação 4.5.1 que modela a posição da extremidade do manipulador em relação à base pela multiplicação de matrizes de transformação homogêneas obtidas pelos parâmetros DH:

$$T_6^0(q) = A_0 A_1 A_2 A_3 A_4 A_5 A_6 \quad \text{Eq. 4.5.1}$$

A Equação 4.4.3 pode ser manipulada para se encontrar a posição do eixo cinco  $P_5^1$ , com base no eixo 1:

$$P_5^1 = P_4^1 = (A_1^0)^{-1} T_n^0 (A_6^5)^{-1} = A_2^1 A_3^2 A_4^3 A_5^4 \quad \text{Eq. 4.5.2}$$

A Equação 4.5.2 retorna uma matriz com a posição e a orientação do eixo 5 em relação ao eixo 1. É importante notar que os eixos 4 e 5 possuem a mesma posição; portanto, a junta 4, por rodar em torno de seu próprio eixo, não tem efeito sobre esta posição. A seguir são expandidos ambos os lados da Equação 4.5.2:

$$A_2^1 A_3^2 A_4^3 A_5^4 = P_5^1 = \begin{bmatrix} & a_2 c_2 + a_3 c_{23} + d_4 s_{23} \\ R & a_2 s_2 + a_3 s_{23} + d_4 c_{23} \\ & 0 \\ 0 & 1 \end{bmatrix} \quad \text{Eq. 4.5.3}$$

$$(A_1^0)^{-1} T(A_6^5)^{-1} = \begin{bmatrix} & (x - b_1 a_6) c_1 + (y - b_2 a_6) s_1 - a_1 \\ R & z - b_3 a_6 \\ & (x - b_1 a_6) s_1 - (y - b_2 a_6) c_1 \\ 0 & 1 \end{bmatrix} \quad \text{Eq. 4.5.4}$$

$$P_5^1 = P_4^1 = \begin{bmatrix} a_2 c_2 + a_3 c_{23} + d_4 s_{23} \\ a_2 s_2 + a_3 s_{23} + d_4 c_{23} \\ 0 \end{bmatrix} = \begin{bmatrix} (x - b_1 a_6) c_1 + (y - b_2 a_6) s_1 - a_1 \\ z - b_3 a_6 \\ (x - b_1 a_6) s_1 - (y - b_2 a_6) c_1 \end{bmatrix} = \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} \quad \text{Eq. 4.5.5}$$

Resolvendo-se a linha três na Equação 4.5.5, obtém-se o ângulo da primeira junta:

$$\theta_1 = \tan^{-1}\left(\frac{y - b_2 a_6}{z - b_1 a_6}\right) + k\pi \quad \text{Eq. 4.5.6}$$

Igualando-se as duas primeiras linhas da Equação 4.5.5, tem-se:

$$\begin{pmatrix} a_2 c_2 + a_3 c_{23} + d_4 s_{23} \\ a_2 s_2 + a_3 s_{23} + d_4 c_{23} \end{pmatrix} = \begin{pmatrix} x' \\ y' \end{pmatrix} \quad \text{Eq. 4.5.7}$$

A Equação 4.5.7 pode ser simplificada para um manipulador plano de dois graus de liberdade, pois a distância entre os eixos 2 e 4 é constante, conforme visto na Figura 31.

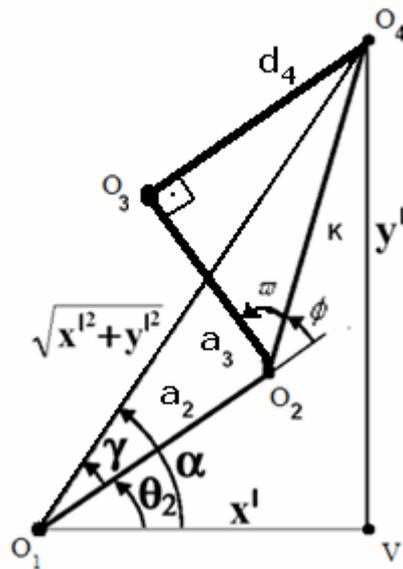


Figura 31: Eixos  $O_2, O_3$  e  $O_4$  sobre a forma de cinemática inversa de dois graus de liberdade

A partir da Equação 4.5.7 e pela análise dos triângulos formados pelos pontos  $O_1\hat{V}O_4$  e  $O_2\hat{O}_3O_4$  obtém-se o seguinte sistema de equações:

$$\begin{aligned}
k &= \sqrt{a_3^2 + d_4^2} \\
a_2 c_2 + k \cdot \frac{a_3}{k} c_{23} + k \cdot \frac{d_4}{k} s_{23} &= a_2 c_2 + k \cos(\theta_2 + \varpi) = x' \\
a_2 s_2 + k \cdot \frac{a_3}{k} s_{23} + k \cdot \frac{d_4}{k} c_{23} &= a_2 s_2 + k \sin(\theta_2 + \varpi) = y' \\
\varpi &= \theta_3 - \phi
\end{aligned} \tag{Eq. 4.5.8}$$

Este sistema é típico de um manipulador plano de dois graus de liberdade, cuja solução, com os ângulos  $\theta_2$  e  $\theta_3$ , é:

$$\theta_2 = \tan^{-1}\left(\frac{y'}{x'}\right) + \cos^{-1}\left(\frac{x'^2 + y'^2 + a_2^2 - k^2}{2a_2\sqrt{x'^2 + y'^2}}\right) + \phi \tag{Eq. 4.5.9}$$

$$\theta_3 = -\cos^{-1}\left(\frac{x'^2 + y'^2 - a_2^2 - k^2}{2a_2k}\right) + \tan^{-1}\left(\frac{d_4}{a_3}\right) \tag{Eq. 4.5.10}$$

Obtidos os ângulos dos três primeiros eixos, e a posição da extremidade do manipulador, é possível criar um sistema de equações que retorne o ângulo dos eixos 4 e 5. Para tanto, deve ser considerada a posição da junta 6 em relação à junta 3. Partindo da Equação 4.5.1, chega-se à seguinte resolução:

$$A_4^3 A_5^4 A_6^5 = P_6^3 = \begin{bmatrix} R & a_6 c_4 s_5 \\ & a_6 s_5 s_4 \\ & d_4 + a_6 c_5 \\ 0 & 1 \end{bmatrix} \tag{Eq. 4.5.11}$$

$$(A_3^0)^{-1} T = \begin{bmatrix} & -a_3 - a_2 c_3 + z s_{23} + y s_1 c_{23} + x c_1 c_{23} - a_1 c_{23} \\ R & x s_1 - y c_1 \\ & -z c_{23} + y s_1 s_{23} - a_2 s_3 + x c_1 s_{23} - a_1 s_{23} \\ 0 & 1 \end{bmatrix} \tag{Eq. 4.5.12}$$

$$P_6^3 = \begin{bmatrix} -a_3 - a_2 c_3 + z s_{23} + y s_1 c_{23} + x c_1 c_{23} - a_1 c_{23} \\ x s_1 - y c_1 \\ -z c_{23} + y s_1 s_{23} - a_2 s_3 + x c_1 s_{23} - a_1 s_{23} \end{bmatrix} = \begin{pmatrix} x'' \\ y'' \\ z'' \end{pmatrix} = \begin{bmatrix} a_6 c_4 s_5 \\ a_6 s_5 s_4 \\ d_4 + a_6 c_5 \end{bmatrix} \tag{Eq. 4.5.13}$$

Note-se que os valores da Equação 4.5.12 são todos constantes, uma vez que os ângulos das juntas 1 a 3 já foram calculados anteriormente. A partir da Equação 4.5.13, os ângulos dos eixos 4 e 5 são:

$$\theta_4 = \tan^{-1}\left(\frac{y''}{x''}\right) + k\pi \quad \text{Eq. 4.5.14}$$

$$\theta_5 = a \tan 2\left(\frac{y''}{a_6 s_4}, \frac{z'' - d_4}{a_6}\right) \quad \text{Eq. 4.5.15}$$

Onde  $z = a \tan 2(y, x)$  é definida tal que  $\sin(z) = x$  e  $\cos(z) = y$ .

Obtidos os ângulos das cinco primeiras juntas, já se tem a posição da extremidade, uma vez que a junta 6 implica apenas em uma mudança de orientação da extremidade. Assim, pode-se obter o ângulo do eixo 6 a partir da posição final da extremidade:

$$A_6^0 = \begin{bmatrix} n_x & p_z & b_x & x \\ n_y & p_y & b_y & y \\ n_z & p_z & b_z & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Eq. 4.5.16}$$

$$n_6 = \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} = \begin{bmatrix} [(c_1 c_{23} c_4 + s_1 s_4) c_5 - c_1 s_{23} s_5] c_6 + (-c_1 c_{23} s_4 + s_1 c_4) s_6 \\ [(s_1 c_{23} c_4 + c_1 s_4) c_5 - s_1 s_{23} s_5] c_6 + (-s_1 c_{23} s_4 + c_1 c_4) s_6 \\ (s_{23} c_4 c_5 + c_{23} s_5) c_6 - s_{23} s_4 s_6 \end{bmatrix} \quad \text{Eq. 4.5.17}$$

Para facilitar a visualização, as seguintes constantes são definidas:

$$\begin{aligned} \mu_1 &= [(c_1 c_{23} c_4 + s_1 s_4) c_5 - c_1 s_{23} s_5] \\ \mu_2 &= (-c_1 c_{23} s_4 + s_1 c_4) \\ \mu_3 &= [(s_1 c_{23} c_4 + c_1 s_4) c_5 - s_1 s_{23} s_5] \\ \mu_4 &= (-s_1 c_{23} s_4 + c_1 c_4) \\ \mu_5 &= (s_{23} c_4 c_5 + c_{23} s_5) \\ \mu_6 &= -s_{23} s_4 \end{aligned}$$

Conhecidos os ângulos das juntas 1 a 5, chega-se a:

$$s_6 = \frac{n_x \mu_3 - n_y \mu_1}{\mu_2 \mu_3 - \mu_4 \mu_1}$$

$$c_6 = \frac{n_x \mu_4 - n_y \mu_2}{\mu_1 \mu_4 - \mu_2 \mu_3}$$
Eq. 6.5.18

Resolvida a Equação 4.5.18, obtém-se o ângulo da junta seis:

$$\theta_6 = a \tan 2(s_6, c_6) + 2k\pi$$
Eq. 4.5.19

As equações da cinemática inversa do manipulador possuem vários termos definidos por  $\tan^{-1}$  e  $\cos^{-1}$ , o que leva a múltiplas soluções, num total de oito. Além disto, existem questões a serem vistas sobre a existência de soluções.

Considerando-se a Equação 4.5.6, referente ao eixo  $I$ , há a princípio a possibilidade de duas soluções. Não obstante, limitações físicas de movimento desta junta permitem que ela se mova em ângulos entre  $-10^\circ$  a  $90^\circ$ , permitindo apenas as soluções em que este ângulo se encontra entre  $-90^\circ$  e  $90^\circ$ .

Quanto às Equações 4.5.9 e 4.5.10, respectivamente referentes aos ângulos dos eixos 2 e 3, têm-se a princípio duas soluções possíveis. Sabendo-se que o manipulador funciona com o cotovelo voltado para cima devido às restrições das juntas, retira-se essa ambiguidade, permitindo-se apenas soluções nas quais  $\Phi$  é negativo.

Na Equação 4.5.14, referente à junta 4, há a possibilidade de singularidade quando o ângulo da junta 5 é nulo. Neste caso, os ângulos das juntas 4 e 6 são redundantes; um deles deve ser fixado (4), e o outro rotacionado para um ajuste fino da orientação da garra.

## 4.6.

### Visão Computacional

Esta seção mostra como é feita a ligação entre os algoritmos descritos no capítulo 3 e a cinemática do manipulador, de forma a se ter uma solução completa de calibragem, na qual o manipulador com uso de câmeras posiciona-se para efetuar suas tarefas.

Os testes de calibração da base do manipulador foram simulados em um ambiente virtual. O motivo principal para tal é o fato de que o ambiente virtual retorna posições precisas de todos os objetos. Foi utilizado o programa 3DStudio© para criação de câmeras e do modelo do manipulador.

#### **4.7.**

##### **Câmera**

Para testar o algoritmo completo de calibração do manipulador foi utilizado um par de câmeras com lentes de 35 milímetros e distância focal de 100mm. As câmeras foram posicionadas a uma distância de 20 milímetros uma da outra, ambas fixadas na extremidade do manipulador.

Assim como no modelo real, ambas as câmeras passaram por um processo de calibração. Foram tiradas oito fotos de uma plataforma de calibração (Figura 12) para cada câmera e em seguida foi utilizado o algoritmo de calibração presente na seção 3, capítulo 3. Os resultados para ambas as câmeras estão mostrados na Tabela 6, presente no Apêndice B:

Como mostrado na seção 3.3, os resultados da calibragem levam a uma matriz de projeção que considera a origem do plano do mundo no centróide dos pontos observados na plataforma de calibração. Assim, a matriz  $M$  considera uma translação e rotação em relação a essa origem, que pode não ser conveniente na prática. Assim, aplica-se a inversa da matriz de parâmetros extrínsecos da câmera esquerda sobre ambas as câmeras, levando o centro do mundo para a câmera esquerda, que está acoplada em uma posição conhecida do manipulador. Os pontos encontrados posteriormente pela triangulação serão posicionados em relação às coordenadas da câmera esquerda.

#### **4.8.**

##### **Correlação de Pontos**

Como visto no capítulo 3, encontrar pontos correspondentes em ambas as imagens é imprescindível para a triangulação e, portanto, para o posicionamento dos pontos da imagem. O método SIFT foi escolhido para encontrar os pontos

estáveis na imagem, e, através de transformada Hough e RANSAC faz-se o relacionamento dos pontos.

A Figura 31 apresenta duas imagens retiradas a um intervalo de 4 segundos uma da outra, enquanto o TA-40 fazia uma intervenção submarina em um painel de controle. A Figura 32 foi obtida de uma intervenção submarina em um painel de controle, sendo que as Figura 32-a e 32-b foram obtidas com dez segundos de diferença enquanto o manipulador se movia. A Figura 33-a mostra 229 pontos obtidos pelo algoritmo SIFT aplicado à imagem 32-a, enquanto que a Figura 33-b mostra 125 pontos encontrados na Figura 32-b. A Figura 34 mostra a correlação de pontos. Cada reta numerada na imagem tem origem em um ponto descoberto por SIFT na Figura 33-a e seu fim em um ponto encontrado por SIFT na Figura 33-b. Pode-se constatar que, apesar de muitos pontos encontrados em ambas as imagens serem ignorados pelo algoritmo, os cerca de vinte pontos que se correlacionam direcionam o movimento de forma coerente, pois as retas são visualmente paralelas.

Esta simulação permite que se constate que o algoritmo é capaz de encontrar um número relevante de pontos nas imagens oferecidas pelas câmeras nas profundidades da área de trabalho do TA-40 e que é capaz de acompanhá-los em imagens consecutivas.

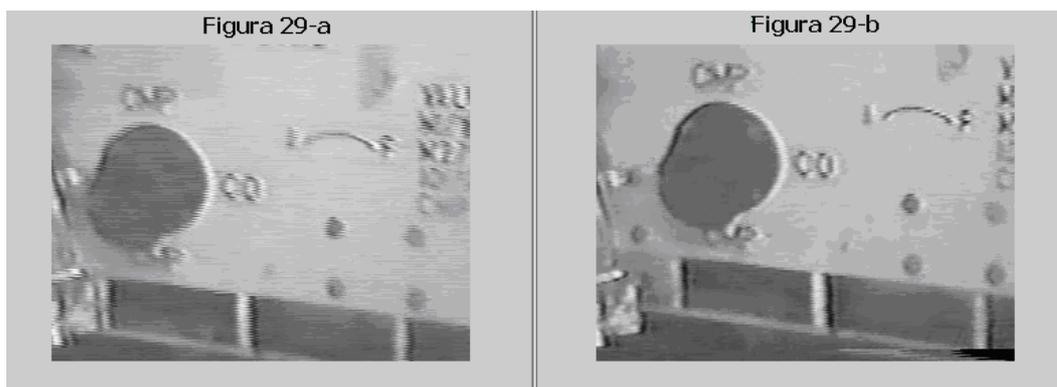


Figura 32 a/b: Imagem de painel obtidas com 10 segundos de diferença durante tarefa

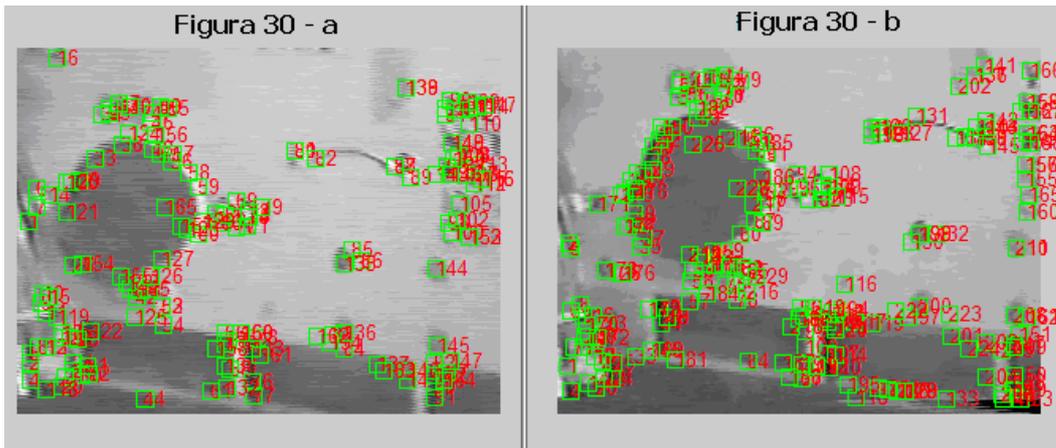


Figura 33 a/b: Imagem de painel com os pontos obtidos pelo algoritmo SIFT

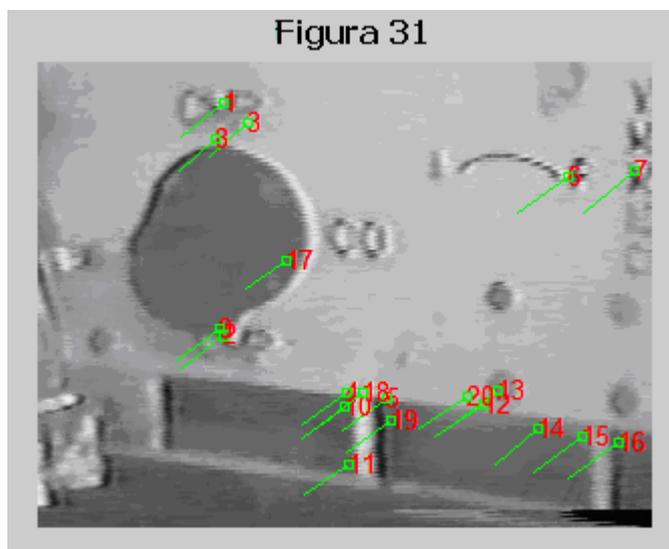


Figura 34: Correlação de pontos

## 4.9.

### Calibragem da Base

O grande objetivo de se utilizar processamento de imagens no TA-40 é o fato de que cada vez que este é levado a sua área de trabalho, os objetos que ele manipulará estarão em posições absolutas distintas da última operação. Assim, o manipulador necessita saber onde está sua base em relação aos objetos próximos que ele irá manipular.

Ao se observar os algoritmos dos capítulos 2 e 3, respectivamente de cinemática e visão computacional, percebe-se que ambos utilizam as matrizes de transformação homogêneas para modelar as posições. Assim, é possível a partir das imagens do ambiente, obter matrizes de transformação homogêneas que definam a posição de pontos neste ambiente em relação às câmeras, e em seguida utilizar as matrizes obtidas na cinemática e calibração para obter a posição destes pontos em relação à base do manipulador.

Tendo-se duas câmeras calibradas acopladas à extremidade do manipulador em uma posição fixa, obtêm-se imagens da área de trabalho com ambas as câmeras. A partir do uso do algoritmo SIFT, as imagens passam a ser convertidas em conjuntos de pontos  $p_L = (\rho_1, \rho_2, \rho_3, \dots, \rho_n)$  na câmera esquerda e  $p_R = (\rho'_1, \rho'_2, \rho'_3, \dots, \rho'_m)$  na câmera direita,  $\rho_i = (u, v)$ , coordenadas na imagem. Por transformada Hough e RANSAC, serão encontrados pontos em  $p_L$  que se relacionem a pontos em  $p_R$ .

Os pontos correlacionados são utilizados na triangulação para se obter a posição dos mesmos no espaço tridimensional,  $P^C = (q_1, q_2, q_3, \dots, q_s)$ , em relação a posição de uma das câmeras, sendo  $q_i = (x, y, z)$ . A posição da câmera em relação à base do manipulador é conhecida, uma vez que se conhece a cinemática direta do manipulador, bem como os ângulos das juntas, seus erros e a posição da câmera em relação à extremidade, determinada por uma matriz de transformação homogênea,  $A_C$  (Equação 4.6.1). A Equação 6.6.2 transforma as posições dos pontos  $P^C$  em relação à câmera para sua posição em relação à base do manipulador.

$$T_C^0(q, \varepsilon) = E_0 A_0 E_1 A_1 E_2 A_2 E_3 A_3 E_4 A_4 E_5 A_5 E_6 A_6 A_C \quad \text{Eq. 4.6.1}$$

$$P^0 = T_C^0 P^C = E_0 A_0 E_1 A_1 E_2 A_2 E_3 A_3 E_4 A_4 E_5 A_5 E_6 A_6 A_C P^C \quad \text{Eq. 4.6.2}$$

Por fim, pode-se movimentar a extremidade do manipulador para qualquer dos pontos desejados utilizando a cinemática inversa. A seguinte metodologia é utilizada para se conseguir a posição de cada objeto da imagem em relação à posição da base:

- 1- Obtém-se um par de imagens com ambas as câmeras.
- 2- Utiliza-se o SIFT, seguido de transformada Hough e RANSAC, para relacionar pontos em ambas as imagens.
- 3- Utiliza-se triangulação sobre os pontos para obter a posição real dos pontos em relação à câmera esquerda,  $P^C$ .
- 4- Calculam-se, a partir dos ângulos das juntas, erros e pela matriz conhecida  $A_C$ , a posição e orientação da câmera em relação à base  $T_C^0$ .
- 5- Utiliza-se a equação 4.6.2 para obter a posição dos pontos em relação à base.
- 6- Utiliza-se a cinemática inversa do manipulador para mover a extremidade para qualquer posição.

Através do uso deste algoritmo é possível ao manipulador posicionar-se em qualquer ambiente sem a intervenção humana. Ao operador, será oferecida apenas a tarefa de escolher posições da imagem onde ele deseja que a ferramenta esteja, e o algoritmo calculará automaticamente os ângulos das juntas necessários para alcançar o objetivo.

## 5

### Resultados

#### 5.1.

##### Introdução

O objetivo deste capítulo é apresentar os resultados de diversas simulações feitas no decorrer do projeto. Tais simulações têm o objetivo de testar os algoritmos presentes no capítulo anterior e verificar se eles se aplicam ao problema prático de calibragem do TA-40.

As simulações aqui feitas procuram obter resultados dos algoritmos aplicados em conjunto para o funcionamento do TA-40.

#### 5.2.

##### Simulação Cinemática

Nesta seção, serão expostos duas simulações. A primeira testa a cinemática direta e inversa, utilizando os parâmetros de Denavit-Hartenberg. A segunda testa a matriz jacobiana de identificação, analisando sua capacidade de encontrar erros presentes nas juntas do manipulador.

##### 5.2.1

##### Cinemática Direta-Inversa

Esta simulação tem como objetivo testar principalmente a cinemática inversa.

O procedimento para esta simulação é o de utilizar os parâmetros de Denavit-Hartenberg presentes na Tabela 1 de forma a calcular as seis matrizes de transformação homogêneas  $A_i$ . A partir destas matrizes e dos ângulos das seis juntas,  $\theta$ , utilizando a Equação 2.3.1 calcula-se a posição e orientação da extremidade com relação à base do manipulador,  $T_6^0(\theta)$ .

Com posse do algoritmo de cálculo da cinemática inversa, calculado na seção 4.5, calcula-se os ângulos das seis juntas a partir da posição da extremidade calculada pela cinemática direta. Os ângulos obtidos pela cinemática inversa deve ser igual ao utilizado na cinemática direta. A simulação é feita com diversos ângulos dentro da área de trabalho o manipulador, de forma a buscar falhas no cálculo da cinemática inversa.

A Tabela 2 mostra o resultado da simulação para nove ângulos obtidos aleatoriamente. O erro máximo encontrado entre os ângulos utilizados e os ângulos calculados pela cinemática inversa em 500 rodadas de simulação foi inferior a 0.3 graus.

Tabela 2: Resultados da simulação de cinemática direta-inversa

Ângulos das Juntas (graus)						Posição da Extremidade (mm)			Ângulo Obtido das Juntas (graus)						Diferença dos ângulos (graus)					
θ1	θ2	θ3	θ4	θ5	θ6	X	Y	Z	θ1	θ2	θ3	θ4	θ5	θ6	e1	e2	e3	e4	e5	e6
17	19	59	51	28	167	1888	436	289	17	19	59	51	28	167	0	0	0	0	0	0
45	52	16	81	64	10	1298	830	467	45	52	16	81	64	10	0	0	0	0	0	0
15	-3	-11	9	128	0	1167	272	-659	15	-3	-11	9	128	0	0	0	0	0	0	0
26	30	21	47	60	-43	1662	553	87	26	30	21	47	60	-43	0	0	0	0	0	0
0	90	-90	0	90	0	663	0	6	0	90	-90	0	90	0	0	0	0	0	0	0
6	72	39	21	55	2	1061	12	1503	6	72	39	21	55	2	0	0	0	0	0	0
24	63	33	8	86	10	1084	432	1292	24	63	33	8	86	10	0	0	0	0	0	0
57	26	44	25	71	19	1081	1417	488	57	26	44	25	71	19	0	0	0	0	0	0
63	15	54	26	33	17	955	1704	176	63	15	54	26	33	17	0	0	0	0	0	0

## 5.2.2

### Jacobiana de Identificação

Esta simulação tem como objetivo testar a matriz Jacobiana de Identificação calculada na seção 4.4, que pode ser encontrada algebricamente no Apêndice A. A matriz Jacobiana de Identificação,  $J_e$ , é utilizada para calcular os erros presentes no manipulador. O procedimento desta simulação é criar 42 erros randômicos, inserir no cálculo da extremidade do manipulador, e utilizar a jacobiana para obter os erros criados.

Inicialmente, criam-se os erros generalizados  $\varepsilon_{xi}$ ,  $\varepsilon_{yi}$ ,  $\varepsilon_{zi}$ ,  $\varepsilon_{pi}$ ,  $\varepsilon_{si}$ ,  $\varepsilon_{ri}$ , para  $i=0..6$ , sendo criados 42 erros. Estes erros são criados aleatoriamente, com valores na ordem de milímetros e mili-radianos. Os erros generalizados  $\varepsilon_{z,(i-1)}$  e  $\varepsilon_{r,(i-1)}$ , para  $i=1..6$ , são eliminados, e seus valores são incorporados aos parâmetros de erro independentes  $\varepsilon_{y,i}^*$ ,  $\varepsilon_{z,i}^*$ ,  $\varepsilon_{s,i}^*$  e  $\varepsilon_{r,i}^*$ .

Calcula-se a partir dos erros generalizados e do ângulo das juntas a posição da extremidade do manipulador (Eq. 2.4.1). Calcula-se também a posição ideal, sem erros, utilizando-se apenas a cinemática direta e a posição das juntas. A diferença entre a posição real e ideal é calculada,  $\Delta X$ .

Utilizando-se a Equação 2.4.5 calculam-se os erros generalizados a partir de  $\Delta X$  e da pseudo-inversa do Jacobiano de Identificação. Os erros calculados devem ser os mesmos erros utilizados. A Tabela 3 apresenta os erros estimados ( $E_{estim}$ ), os erros utilizados ( $E_{real}$ ) e a diferença entre eles em uma simulação. Em 500 simulações, o maior erro obtido foi da ordem de  $10^{-13}$ , demonstrando que o cálculo está adequado.

Tabela 3: Resultados da simulação com a Jacobiana de Identificação

Erro	Erros Reais ( $E_{real}$ )	Erros Estimados ( $E_{estim}$ )	Diferença ( $E_{real}-E_{estim}$ )
Ex0	0,000875	0,000875	2,93E-18
Ey0	0,000835	0,000835	3,03E-16
Es0	1,54E-05	1,54E-05	-1,89E-18
Ep0	9,79E-06	9,79E-06	3,85E-19
Ex1	0,000616	0,000616	2,77E-15
Ey1	0,000995	0,000995	1,50E-16
Es1	2,03E-05	2,03E-05	-3,69E-18
Ep1	1,25E-05	1,25E-05	2,02E-19
Ex2	0,000515	0,000515	-1,96E-15
Ey2	0,007311	0,007311	-2,32E-15
Es2	1,43E-05	1,43E-05	2,27E-18
Ep2	1,03E-05	1,03E-05	-1,30E-19
Ex3	0,00013	0,00013	1,78E-13
Ey3	0,000875	0,000875	1,09E-13
Es3	2,61E-05	2,61E-05	-1,24E-16
Ep3	9,80E-06	9,80E-06	8,92E-17
Ex4	0,000455	0,000455	-7,16E-14
Ey4	0,002819	0,002819	-6,90E-17
Es4	8,97E-07	8,97E-07	3,72E-17
Ep4	1,72E-05	1,72E-05	8,61E-17
Ex5	0,008772	0,008772	2,27E-14
Ep5	-2,32E-06	-2,32E-06	-3,66E-17
Ex6	0,000542	0,000542	2,07E-16
Ey6	6,86E-06	6,86E-06	4,38E-16
Ez6	0,001104	0,001104	1,47E-15

## 5.3

### Simulações em Visão

O algoritmo de triangulação já foi utilizado em diversos trabalhos e é relativamente confiável quando os pontos utilizados são bem correlacionados. O algoritmo SIFT é adequado para a obtenção de pontos estáveis, e, em conjunto com RANSAC e a transformada Hough, é uma ferramenta robusta para correlacionar pontos. São formuladas a seguir três simulações: a primeira para testar o algoritmo de triangulação dentro de distâncias pertencentes à aplicação, a segunda para testar diferentes resoluções, e a última para testar se o algoritmo permite realmente encontrar posições de pontos na imagem em relação à base do manipulador.

#### 5.3.1

##### Triangulação em Distâncias

O manipulador TA-40 pode alcançar uma distância máxima de dois metros da base. Assim, foi testada a capacidade do algoritmo de triangulação para calcular posições de objetos em imagens cuja distância para a câmera fosse de até dois metros.

De forma semelhante ao realizado para as simulações de cinemática, primeiramente foi criado um conjunto de pontos a distâncias conhecidas da câmera esquerda ( $P_R$ ). O algoritmo foi testado para dez distâncias distintas, variando entre 50 milímetros e 2 metros no eixo Z. Para cada distância, foram criados  $n$  pontos com valores nos eixos X e Y variando aleatoriamente, mas sempre possíveis de ser visualizados em ambas as câmeras.

Criados os pontos, foi utilizada a Equação 3.2.9 para se obter a posição em que cada ponto se encontra no quadro da imagem. O parâmetro  $z$  da Equação 3.2.9 foi igual à distância do conjunto de pontos à câmera esquerda no eixo Z. A Matriz de Projeção Perspectiva ( $M$ ) foi calculada com os parâmetros intrínsecos obtidos na calibração (Apêndice B) e com parâmetros extrínsecos conforme a Equação 5.3.1 (câmeras afastadas em cinquenta milímetros no eixo X e paralelas entre si).

$\begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 50 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	Eq. 5.3.1
--	-----------

Uma vez determinados os pontos no plano da imagem, eles são utilizados no processo de triangulação. São utilizados os mesmos parâmetros intrínsecos e extrínsecos para obtenção de um conjunto de pontos calculados no espaço tridimensional em relação à câmera esquerda ( $P_P$ ). Uma medida de erro ( $P_E$ ) para cada distância ( $d$ ) é obtida calculando-se a distância euclidiana entre os vetores  $P_R$  e  $P_P$ , para cada ponto, e em seguida calculando-se o erro médio quadrático:

$$P_E(d) = \sqrt{\sum_{i=1}^n P_R(i, d)^2 - P_P(i, d)^2 / n} \quad \text{Eq. 5.3.2}$$

A Figura 35 mostra uma representação da simulação para uma distância. A tabela 4 mostra os resultados das simulações.

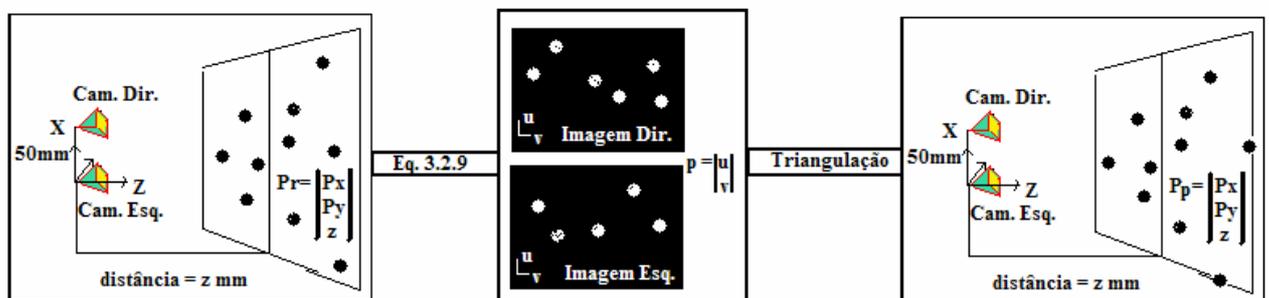


Figura 35: Esquema do algoritmo para uma distância fixa, z.

Tabela 4: Resultados da simulação de triangulação em distâncias

Distancia(mm)	$P_E$ (mm)	Erro Máximo
100,0	3,86	3,93
150,0	6,64	5,92
200,0	6,84	11,65
300,0	18,03	22,83
500,0	32,67	43,63
800,0	48,41	67,41
1000,0	57,66	68,12
1300,0	98,56	102,293
1600,0	121,86	142,36
2000,0	203,02	319,30

Percebe-se que o erro aumenta com a distância das esferas à câmera, como esperado. Porém espera-se que o manipulador, uma vez acoplado na sua área de trabalho, deva estar a uma distância entre sua extremidade e os objetos a serem movidos muito inferior a dois metros, com erros na ordem de centímetros. A Figura 36 mostra os erros médios quadráticos em diversas distâncias de acordo com o eixo.

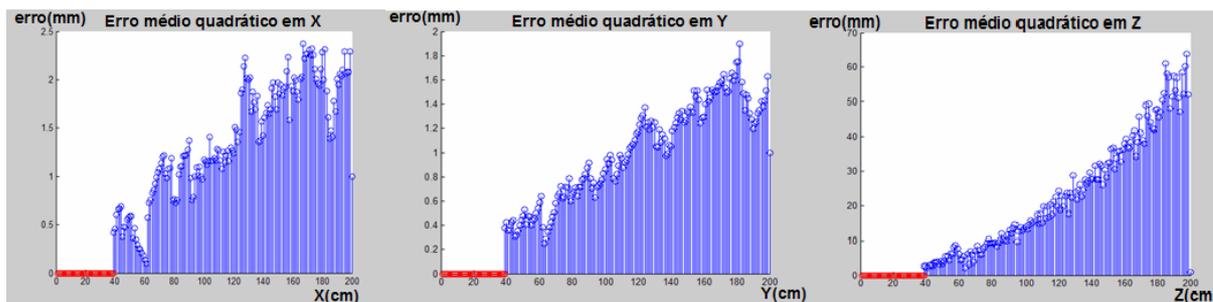


Figura 36: Simulação de triangulação em função de distância de objetos à câmera.

### 5.3.2

#### Triangulação em Função da Resolução

O algoritmo SIFT é mais rápido de ser processado se a imagem tiver dimensões menores, porém isto provoca uma redução da resolução, o que interfere na triangulação. Para testar esta interferência, foi feita a mesma simulação anterior com diversas resoluções de câmeras. Tendo sido feitas 500 simulações para cada resolução, com câmeras distanciadas em 100 milímetros, a Tabela 4 mostra os resultados para algumas distâncias.

Tabela 5: Medida de erro de triangulação a partir da resolução e distância da câmera (erros em milímetros).

Resolução/Distância(mm)	1000	1500	2000
320x240	17,8	41,3	73,5
405x254	15,0	32,9	62,9
640x480	8,0	18,2	32,4
811x508	7,6	18,4	31,3

A simulação comprovou que o algoritmo de triangulação possui erros maiores quanto maior a distância, mas também que este erro aumenta em

resoluções menores. No entanto uma resolução de 640x480 pode ser suficiente para oferecer ao algoritmo erros pequenos, trazendo o benefício de uma maior rapidez na computação.

### 5.3.3

#### **Simulação de Calibração da Base**

A última simulação testa o posicionamento do manipulador em relação aos objetos vistos pela câmera. Nesta simulação, foi criada uma réplica do manipulador em ambiente virtual e adicionadas em posições fixas de sua extremidade duas câmeras virtuais calibradas. Em seguida, foram inseridas esferas de cores diferenciadas no ambiente virtual, com posições fixas. O sistema de coordenadas do mundo é fixado na base do manipulador, enquanto que as posições reais das esferas em relação a esta base são medidas. Em seguida, são obtidas imagens das esferas em diversas configurações do TA-40, feita a triangulação para calcular as posições das esferas em relação a câmera esquerda, e utilizada a Equação 4.6.2 para obter tais posições em relação à base do manipulador.

É utilizada a Equação 5.3.1 para calcular o erro médio quadrático da posição real de cada esfera, com relação à calculada. Os resultados estão na Tabela 5, enquanto que as Figuras 37, 38 e 39 mostram imagens capturadas em posições distintas do manipulador.

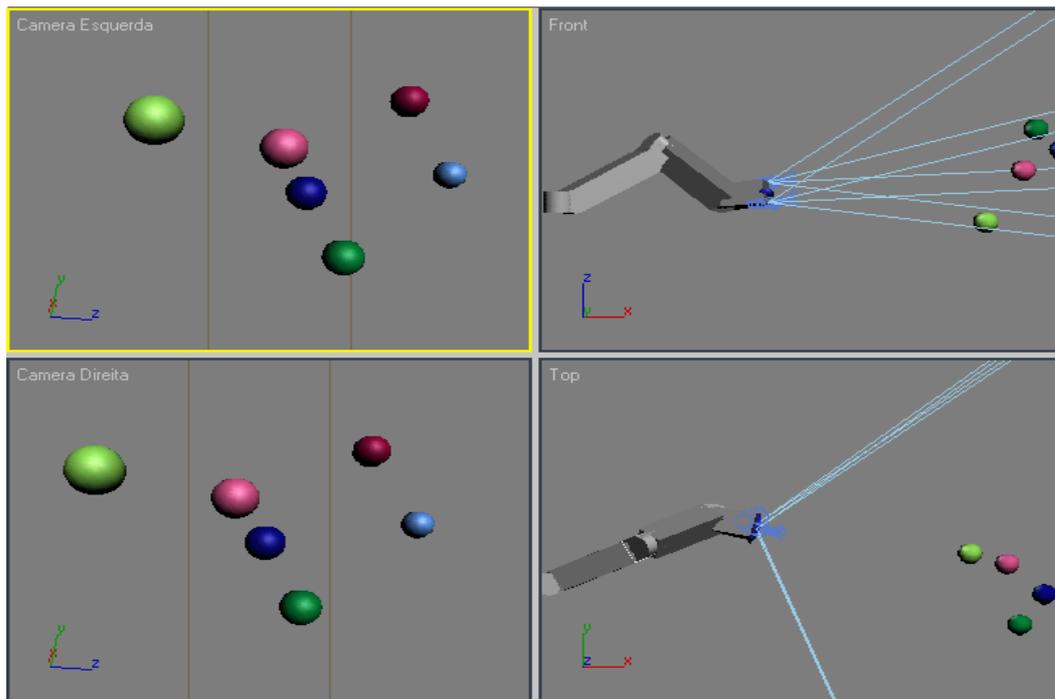


Figura 37: Visões da simulação de calibração da base, com manipulador configurado com  $\theta_1=26^\circ$ ,  $\theta_2=30^\circ$ ,  $\theta_3=20^\circ$ ,  $\theta_4=47^\circ$ ,  $\theta_5=60^\circ$ ,  $\theta_6=43^\circ$ .

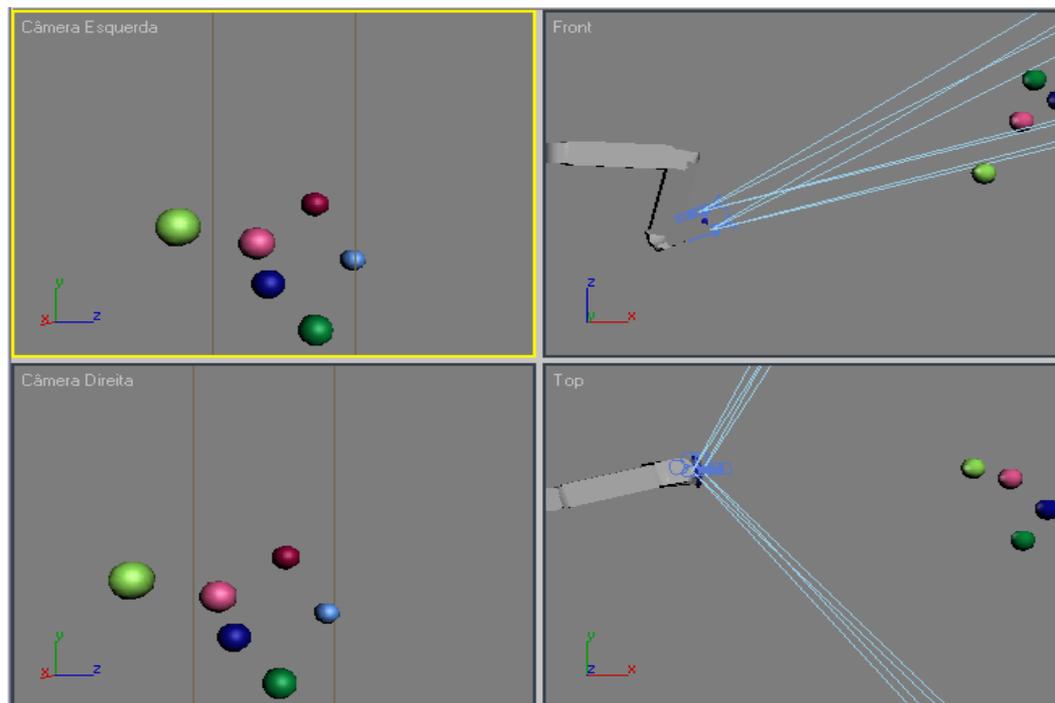


Figura 38: Visões de simulação de calibração da base, com manipulador configurado com  $\theta_1=15^\circ$ ,  $\theta_2=-3^\circ$ ,  $\theta_3=-10^\circ$ ,  $\theta_4=10^\circ$ ,  $\theta_5=128^\circ$ ,  $\theta_6=0^\circ$ .

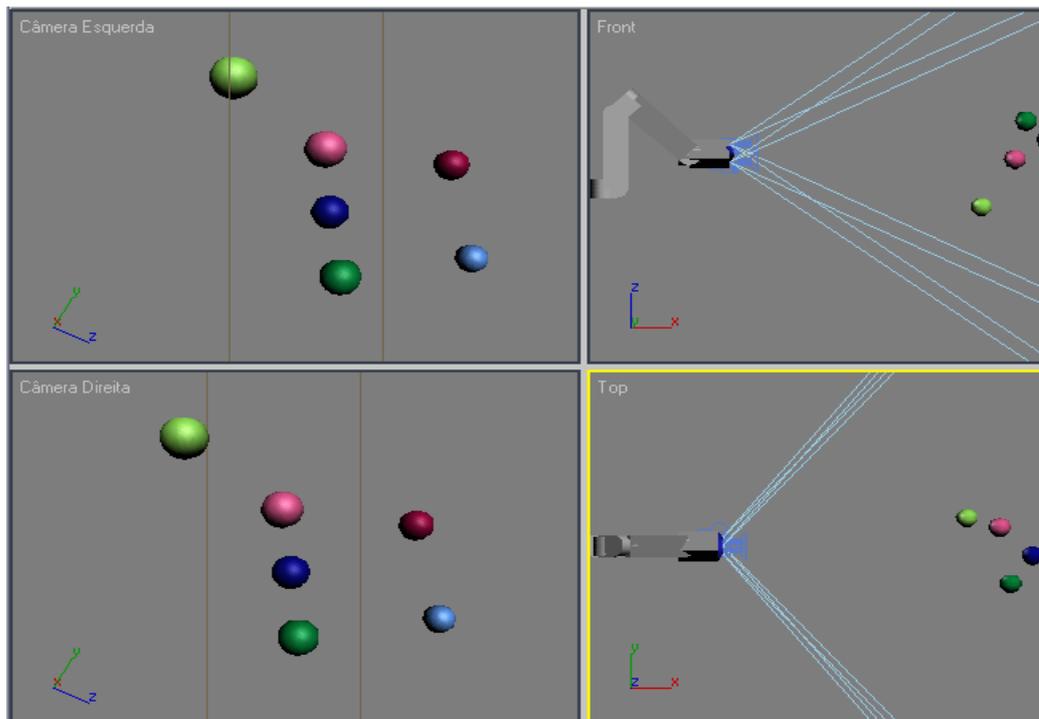


Figura 39: Visões de simulação de calibração da base, com manipulador configurado com  $\theta_1=0^\circ$ ,  $\theta_2=90^\circ$ ,  $\theta_3=-90^\circ$ ,  $\theta_4=0^\circ$ ,  $\theta_5=90^\circ$ ,  $\theta_6=0^\circ$ .

Tabela 6: Resultados da simulação de Calibração da Base

Angulo das Juntas ( $^\circ$ )						Posicao da Extremidade (mm)			Erro (mm)
$\theta_1$	$\theta_2$	$\theta_3$	$\theta_4$	$\theta_5$	$\theta_6$	X	Y	Z	
16,9	18,5	58,7	51,2	27,6	167,3	187,6	37,7	36,8	12,4
44,6	51,8	15,9	81,4	64	10	132,8	92,3	44,4	29,4
15,3	-3	-10,7	9,4	128,2	0	120,9	27,3	-74,9	13,2
25,9	29,7	20,5	47,4	60	-42,6	167,2	53,7	-1,1	12,6

## 6

### Conclusões e Sugestões

#### 6.1.

##### Contribuições do Trabalho

Este trabalho contribuiu no desenvolvimento de técnicas de calibragem, sobretudo na área relacionada à calibração da base do manipulador em relação ao seu ambiente de trabalho. As técnicas abordadas são gerais a qualquer manipulador, e portanto podem beneficiar não apenas usuários do TA-40, como a PETROBRAS.

Os algoritmos presentes no capítulo de visão computacional são bem conhecidos e já utilizados em diversas aplicações, porém o algoritmo SIFT em particular é recente, sendo aplicado basicamente em reconhecimento de padrões. O uso do algoritmo SIFT na calibração do manipulador é uma contribuição substancial deste trabalho.

A junção dos algoritmos de visão computacional ao de calibração cinemática permite que o manipulador passe a ter conhecimento do seu ambiente e a atuar com menos interferência do operador. É um passo à frente em direção a uma automação completa de braços mecânicos, onde os manipuladores não necessitarão mais ter suas trajetórias programadas previamente pelo operador, mas passarão a ser criadas no momento das tarefas de acordo com as dificuldades impostas pelo ambiente.

Quanto à aplicação prática, a calibração do manipulador TA-40, este projeto teve muitas contribuições. Foram calculadas a cinemática direta, inversa e a matriz jacobiana de identificação para este manipulador em particular. Também foi desenvolvida uma metodologia para que o manipulador conheça a posição de sua base em relação aos objetos a serem manipulados.

Com base no que foi desenvolvido, passa a ser possível desenvolver um programa que permita serem feitas tarefas em um ambiente virtual. Neste programa, haveria um modelo virtual do manipulador, criado com base em sua cinemática direta, incluindo os erros. Também haveria a posição ideal dos objetos que seriam manipulados. Neste ambiente virtual seriam criadas estratégias para completar tarefas, sem a necessidade de ativar fisicamente o manipulador. A movimentação neste ambiente virtual será feita baseada na cinemática inversa, deixando ao operador apenas a tarefa de definir que objetos serão manipulados.

Também poderá ser desenvolvido um programa para a atuação real do manipulador. Este programa receberia as imagens das câmeras e mediria as posições dos objetos sobre os quais o manipulador deve atuar. Em seguida, o manipulador compara a posição real da base com a posição ideal da base, calculada *offline*. O programa seria capaz de recalcular os ângulos necessários para recriar as tarefas no ambiente real, e através do uso conjunto da cinemática inversa e da visão computacional, desenvolver sua tarefa com o mínimo de esforço humano.

Com base nas simulações, foi possível escolher parâmetros para a câmera que melhor se adequam à aplicação, e garantir que os algoritmos possuam bons resultados nas distâncias em que o manipulador atuará. Espera-se que os programas desenvolvidos por base nestes algoritmos sejam confiáveis e robustos.

## 6.2.

### **Sugestões para Trabalhos Futuros**

Este projeto foi capaz de buscar algoritmos e metodologias necessárias para calibrar um manipulador serial que necessite atuar em ambientes mutáveis. Foram apresentadas também a aplicação destes algoritmos a um caso real. Porém os algoritmos foram testados em ambientes virtuais. Durante o desenvolvimento das técnicas, o TA-40 estava indisponível ao uso, e portanto não foi possível fazer testes em laboratório.

É notório que algoritmos baseados em visão computacional necessitem ser aperfeiçoados quando utilizados em imagens reais. Uma sugestão para um futuro trabalho é testar os algoritmos aqui desenvolvidos em um ambiente real. Também

é sugerida a criação de um programa completo para a atuação do manipulador por base nos algoritmos aqui apresentados.

O algoritmo de triangulação aqui utilizado teve um desempenho aceitável nas distâncias até dois metros, possuindo erros substanciais quanto maior a distância. Uma alternativa a este método que pode ser estudada é o uso de triangulação probabilística, que provavelmente apresentará resultados superiores a distâncias maiores.

## Referências Bibliográficas

AIT-AIDER O.; HOPPENOT, P. and COLLE, E. **Adaptation of Lowe's Camera Pose Recovery Algorithm to Mobile Robot Self-Localisation** - Robotica 2002.

ASADA, H. and SLOTINE, J.-J. **Robot Analysis and Control**, Wiley, New York, 1986.

BALLARD, D. and BROWN, C. **Computer Vision**, Prentice-Hall, Chap 4, 1982.

BROWN, M. and LOWE, D.G. **Invariant Features from Interest Point Groups**, Vision Conference, Cardiff, Wales, pp. 656-665, 2002.

DENAVID, J. and HARTENBERG, R.S. **A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices**, ASME J. Applied Mechanics, pp.215-221, 1955.

FISCHLER M.A. and BOLLES R.C. **Random Sample Consensus; A paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography**. Comm. Of the ACM, Vol 24, pp. 381-395, 1981

FORSYTH, D.A. and PONCE J. **Computer Vision, a Modern Approach**, ed. Prentice Hall, 2003.

HOLLERBACH, J.M. and WAMPLER, C.W. **The Calibration Index and Taxonomy for Robot Kinematic Calibration Methods**, *International Journal of Robotics Research*, Vol. 15, No. 6, pp. 573-591, 1996.

LOWE, D.G. **Distinctive Image Features from Scale-Invariant Keypoints**, University of British Columbia Vancouver, B.C., Canada, 2004.

MEGGIOLARO, M.; SCRIFFIGNANO, G. and DUBOWSKY, S. **Manipulator Calibration Using A Single Endpoint Contact Constraint**, *Proceedings of the 2000 ASME Design Engineering Technical Conference*, Maryland, 2000.

MEGGIOLARO, M. **Achieving Fine Absolute Positioning Accuracy in Large Powerful Manipulators**, PhD thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, Massachusetts, 2000a

MEGGIOLARO, M. and DUBOWSKY, S. **An Analytical Method to Eliminate the Redundant Parameters in Robot Calibration**, *Proceedings of the International Conference on Robotics and Automation (ICRA '2000)*, IEEE, San Francisco, pp. 3609-3615, 2000b.

MEGGIOLARO, M.; DUBOWSKY, S. and MAVROIDIS, C. **Calibrating for Geometric and Elastic Errors in Large Manipulators Including Elimination of Redundant Error Parameters**, submitted to the *International Journal of Robotics Research*, 2000c.

MEGGIOLARO, M.; JAFFE, P. and DUBOWSKY, S. **Achieving Fine Absolute Positioning Accuracy in Large Powerful Manipulators**, *Proceedings of the International Conference on Robotics and Automation (ICRA '99)*, pp.2819-2824, IEEE, Detroit, 1999b.

MEGGIOLARO, M.; JAFFE, P.; IAGNEMMA, K. and DUBOWSKY, S. **A Force-Updated Kinematic Virtual Viewing System With Application To Nuclear Power Plant Maintenance**, *Proceedings of the Tenth World Congress on The Theory of Machine and Mechanisms (IFTOMM)*, Finland, 1999a.

MEGGIOLARO, M.; MAVROIDIS, C. and DUBOWSKY, S. **Identification and Compensation of Geometric and Elastic Errors in Large Manipulators: Application to a High Accuracy Medical Robot**, *Proceedings of the 1998 ASME Design Engineering Technical Conference*, Atlanta, 1998.

MOORING, B.W.; ROTH S.Z. and DRIELS, M. **Fundamentals of Manipulator Calibration**, John Wiley & Sons, 1991.

MURRAY, R.M.; LI, Z. and SASTRY, S.S. **A Mathematical Introduction to Robotic Manipulation**. Boca Raton, FL: CRC Press, 1994.

PIEPER, D.L. **The kinematics of manipulators under computer control**. Stanford Artificial Intelligence Laboratory, Stanford University, AIM 72, 1968.

ROTH, Z.S.; MOORING, B.W. and RAVANI, B. **An Overview of Robot Calibration**, *IEEE Southcon Conference*, Vol.RA-3, No.5, pp. 377-85, 1987.

SCHEFFER, B. **Geometric Control and Calibration Methods of Industrial Robots**, *12<sup>th</sup> International Symposium on Industrial Robotics*, Renault, France, 1976.

SCHRÖER, K. **Theory of Kinematic Modelling and Numerical Procedures for Robot Calibration**. In Bernhardt, R., Albright, S.L. (eds.): *Robot Calibration*. London: Chapman & Hall, pp.157-196, 1993.

SLOCUM, A. **Precision Machine Design**, Prentice Hall, Englewood Cliffs, 1992.

ZHUANG, H.; MOTAGHEDI, S.H. and ROTH, Z.S. **Robot Calibration with Planar Constraints**, *Proc. IEEE International Conference of Robotics and Automation*, Detroit, Michigan, pp.805-810, 1999.

ZHUANG, HANQI; ROTH, S. ZVI. **Camera-Aided Robot Calibration**, ed. CRC Press, Inc, 1996.

## 8

### Apêndice A

Neste apêndice estão as matrizes relacionadas à cinemática direta do manipulador TA-40, bem como as matrizes da Jacobiana de Identificação, para a formulação dos erros generalizados.

Considere os parâmetros D.H.  $q_1$ ,  $q_2$ ,  $q_3$ ,  $q_4$ ,  $q_5$  e  $q_6$  para o manipulador robótico TA-40, definido na Figura 40.

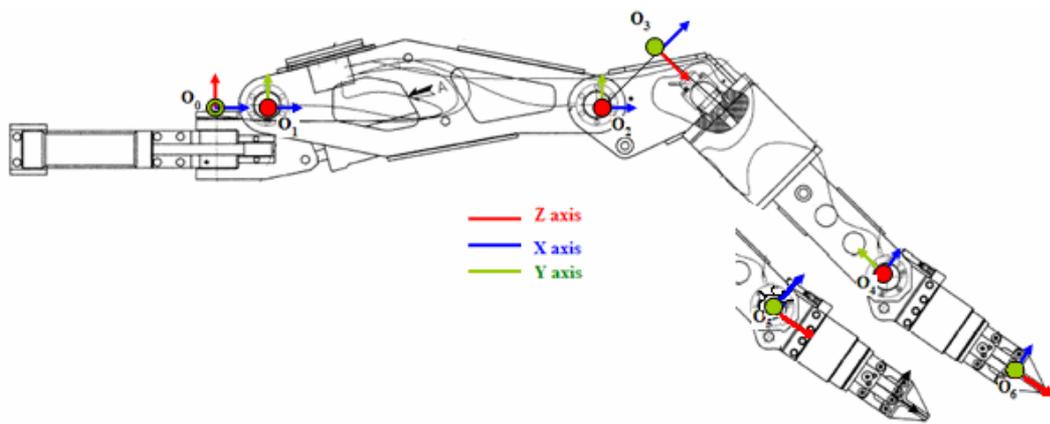


Figura 40: Sistema de coordenadas do TA-40.

#### 8.1.

##### Posição da Extremidade

$$\begin{aligned} X = & (((\cos(q_1) \cdot \cos(q_2) \cdot \cos(q_3) - \cos(q_1) \cdot \sin(q_2) \cdot \sin(q_3)) \cdot \cos(q_4) \\ & + \sin(q_1) \cdot \sin(q_4)) \cdot \sin(q_5) - (-\cos(q_1) \cdot \cos(q_2) \cdot \sin(q_3) \\ & - \cos(q_1) \cdot \sin(q_2) \cdot \cos(q_3)) \cdot \cos(q_5)) \cdot d_6 \\ & + (\cos(q_1) \cdot \cos(q_2) \cdot \sin(q_3) + \cos(q_1) \cdot \sin(q_2) \cdot \cos(q_3)) \cdot d_4 \\ & + \cos(q_1) \cdot \cos(q_2) \cdot a_3 \cdot \cos(q_3) - \cos(q_1) \cdot \sin(q_2) \cdot a_3 \cdot \sin(q_3) \\ & + \cos(q_1) \cdot a_2 \cdot \cos(q_2) + a_1 \cdot \cos(q_1) \end{aligned}$$

$$\begin{aligned}
Y = & ((\sin(q1) \cdot \cos(q2) \cdot \cos(q3) - \sin(q1) \cdot \sin(q2) \cdot \sin(q3)) \cdot \cos(q4) \\
& - \cos(q1) \cdot \sin(q4)) \cdot \sin(q5) - (-\sin(q1) \cdot \cos(q2) \cdot \sin(q3) \\
& - \sin(q1) \cdot \sin(q2) \cdot \cos(q3)) \cdot \cos(q5)) \cdot d6 \\
& + (\sin(q1) \cdot \cos(q2) \cdot \sin(q3) + \sin(q1) \cdot \sin(q2) \cdot \cos(q3)) \cdot d4 \\
& + \sin(q1) \cdot \cos(q2) \cdot a3 \cdot \cos(q3) - \sin(q1) \cdot \sin(q2) \cdot a3 \cdot \sin(q3) \\
& + \sin(q1) \cdot a2 \cdot \cos(q2) + a1 \cdot \sin(q1)
\end{aligned}$$

$$\begin{aligned}
Z = & ((\sin(q2) \cdot \cos(q3) + \cos(q2) \cdot \sin(q3)) \cdot \cos(q4) \cdot \sin(q5) \\
& - (-\sin(q2) \cdot \sin(q3) + \cos(q2) \cdot \cos(q3)) \cdot \cos(q5)) \cdot d6 \\
& + (\sin(q2) \cdot \sin(q3) - \cos(q2) \cdot \cos(q3)) \cdot d4 + \sin(q2) \cdot a3 \cdot \cos(q3) \\
& + \cos(q2) \cdot a3 \cdot \sin(q3) + a2 \cdot \sin(q2)
\end{aligned}$$

## 8.2.

### Jacobiana de Identificação

A seguir está a Matriz Jacobiana de Identificação, composta por quatro linhas e quarenta e duas colunas. Com o objetivo de auxiliar a aplicação, a jacobiana será exposta na forma de algoritmo.

```

function Je = jacobiana_identificacao(q1,q2,q3,q4,q5,q6)
%matriz jacobiana de identificação
%entradas = seis ângulos das juntas do manipulador

%calcula-se os senos e cossenos antecipadamente para aumentar a velocidade
c1 = cos(q1);c2 = cos(q2);c3 = cos(q3);c4 = cos(q4);c5 = cos(q5);c6 = cos(q6);
s1 = sin(q1);s2 = sin(q2);s3 = sin(q3);s4 = sin(q4);s5 = sin(q5);s6 = sin(q6);
%parâmetros DH do manipulador TA-40 em milímetros
a1=0.115;a2=0.753;a3=0.188;d4=0.747;a6=0.360;

% JX,0
Je(1,1) = 1;
Je(2,1) = 0;
Je(3,1) = 0;

```

%J<sub>Y,0</sub>

$$Je(1,2) = 0;$$

$$Je(2,2) = 1;$$

$$Je(3,2) = 0;$$

% J<sub>Z,0</sub>

$$Je(1,3) = 0;$$

$$Je(2,3) = 0;$$

$$Je(3,3) = 1;$$

%J<sub>S,0</sub>

$$Je(1,4) = c3*a3*s2 + s3*a3*c2 - d4*c3*c2 - a6*c5*c3*c2 + a6*c5*s3*s2 \\ + d4*s3*s2 + a6*s5*c4*s3*c2 + s2*a2 + a6*s5*c4*c3*s2;$$

$$Je(2,4) = 0;$$

$$Je(3,4) = -c3*a3*c2*c1 - c1*a1 - a6*c5*s3*c2*c1 - a6*c5*c3*s2*c1 - a6*s5*s4*s1 \\ + s3*a3*s2*c1 - d4*s3*c2*c1 - d4*c3*s2*c1 - a6*s5*c4*c3*c2*c1 - \\ c2*a2*c1 + a6*s5*c4*s3*s2*c1;$$

%J<sub>R,0</sub>

$$Je(1,5) = s3*a3*s2*s1 - s1*a1 - a6*c5*s3*c2*s1 + a6*s5*c4*s3*s2*s1 \\ - a6*s5*c4*c3*c2*s1 - d4*c3*s2*s1 - c3*a3*c2*s1 \\ + a6*s5*s4*c1 - d4*s3*c2*s1 - a6*c5*c3*s2*s1 - c2*a2*s1;$$

$$Je(2,5) = c3*a3*c2*c1 + c1*a1 + a6*c5*s3*c2*c1 + a6*c5*c3*s2*c1 + a6*s5*s4*s1 \\ - s3*a3*s2*c1 + d4*s3*c2*c1 + d4*c3*s2*c1 + a6*s5*c4*c3*c2*c1 \\ + c2*a2*c1 - a6*s5*c4*s3*s2*c1;$$

$$Je(3,5) = 0;$$

%J<sub>P,0</sub>

$$Je(1,6) = 0;$$

$$Je(2,6) = -c3*a3*s2 - s3*a3*c2 + d4*c3*c2 + a6*c5*c3*c2 - a6*c5*s3*s2 - d4*s3*s2 \\ - a6*s5*c4*s3*c2 - s2*a2 - a6*s5*c4*c3*s2;$$

$$Je(3,6) = -s3*a3*s2*s1 + s1*a1 + a6*c5*s3*c2*s1 - a6*s5*c4*s3*s2*s1 \\ + a6*s5*c4*c3*c2*s1 + d4*c3*s2*s1 + c3*a3*c2*s1 \\ - a6*s5*s4*c1 + d4*s3*c2*s1 + a6*c5*c3*s2*s1 + c2*a2*s1;$$

%J<sub>X,1</sub>

$$Je(1,7) = c1;$$

$$Je(2,7) = s1;$$

$$Je(3,7) = 0;$$

%J<sub>Y,1</sub>

$$Je(1,8) = 0;$$

$$Je(2,8) = 0;$$

$$Je(3,8) = 1;$$

%J<sub>Z,1</sub>

$$Je(1,9) = s1;$$

$$Je(2,9) = -c1;$$

$$Je(3,9) = 0;$$

%J<sub>S,1</sub>

$$Je(1,10) = s3*a3*s2*s1 - a6*c5*s3*c2*s1 + a6*s5*c4*s3*s2*s1 - a6*s5*c4*c3*c2*s1 \\ - d4*c3*s2*s1 - c3*a3*c2*s1 + a6*s5*s4*c1 - d4*s3*c2*s1 - a6*c5*c3*s2*s1 \\ - c2*a2*s1;$$

$$Je(2,10) = c3*a3*c2*c1 + a6*c5*s3*c2*c1 + a6*c5*c3*s2*c1 + a6*s5*s4*s1 \\ - s3*a3*s2*c1 + d4*s3*c2*c1 + d4*c3*s2*c1 + a6*s5*c4*c3*c2*c1 \\ + c2*a2*c1 - a6*s5*c4*s3*s2*c1;$$

$$Je(3,10) = 0;$$

%J<sub>R,1</sub>

$$Je(1,11) = -c3*a3*s2*c1 - d4*s3*s2*c1 + d4*c3*c2*c1 - s2*a2*c1 - s3*a3*c2*c1 \\ - a6*s5*c4*c3*s2*c1 - a6*s5*c4*s3*c2*c1 - a6*c5*s3*s2*c1 \\ + a6*c5*c3*c2*c1;$$

$$Je(2,11) = -d4*s3*s2*s1 - a6*s5*c4*s3*c2*s1 + d4*c3*c2*s1 - a6*s5*c4*c3*s2*s1 \\ - s3*a3*c2*s1 - c3*a3*s2*s1 - s2*a2*s1 + a6*c5*c3*c2*s1 - a6*c5*s3*s2*s1;$$

$$Je(3,11) = c3*a3*c2 + d4*c3*s2 + d4*s3*c2 - s3*a3*s2 + a6*s5*c4*c3*c2 \\ + c2*a2 + a6*c5*c3*s2 + a6*c5*s3*c2 - a6*s5*c4*s3*s2;$$

%J<sub>P,1</sub>

$$\begin{aligned} \text{Je}(1,12) &= d4*s3*s2*s1 + a6*s5*c4*s3*c2*s1 - d4*c3*c2*s1 + a6*s5*c4*c3*s2*s1 \\ &\quad + s3*a3*c2*s1 + c3*a3*s2*s1 + s2*a2*s1 - a6*c5*c3*c2*s1 \\ &\quad + a6*c5*s3*s2*s1; \end{aligned}$$

$$\begin{aligned} \text{Je}(2,12) &= -c3*a3*s2*c1 - d4*s3*s2*c1 + d4*c3*c2*c1 - s2*a2*c1 - s3*a3*c2*c1 \\ &\quad - a6*s5*c4*c3*s2*c1 - a6*s5*c4*s3*c2*c1 - a6*c5*s3*s2*c1 \\ &\quad + a6*c5*c3*c2*c1; \end{aligned}$$

$$\text{Je}(3,12) = -a6*s5*s4;$$

%J<sub>X,2</sub>

$$\text{Je}(1,13) = c2*c1;$$

$$\text{Je}(2,13) = c2*s1;$$

$$\text{Je}(3,13) = s2;$$

%J<sub>Y,2</sub>

$$\text{Je}(1,14) = -s2*c1;$$

$$\text{Je}(2,14) = -s2*s1;$$

$$\text{Je}(3,14) = c2;$$

%J<sub>Z,2</sub>

$$\text{Je}(1,15) = s1;$$

$$\text{Je}(2,15) = -c1;$$

$$\text{Je}(3,15) = 0;$$

%J<sub>S,2</sub>

$$\text{Je}(1,16) = -a6*s5*c4*c3*s1 - a6*c5*s3*s1 - c3*a3*s1 - d4*s3*s1 + a6*s5*s4*c2*c1;$$

$$\text{Je}(2,16) = a6*c5*s3*c1 + a6*s5*c4*c3*c1 + a6*s5*s4*c2*s1 + d4*s3*c1 + c3*a3*c1;$$

$$\text{Je}(3,16) = a6*s5*s4*s2;$$

%J<sub>R,2</sub>

$$\text{Je}(1,17) = -c3*a3*s2*c1 - d4*s3*s2*c1 + d4*c3*c2*c1 - s3*a3*c2*c1$$

$$- a6*s5*c4*c3*s2*c1 - a6*s5*c4*s3*c2*c1$$

$$- a6*c5*s3*s2*c1 + a6*c5*c3*c2*c1;$$

$$\begin{aligned} \text{Je}(2,17) = & -d4*s3*s2*s1 - a6*s5*c4*s3*c2*s1 + d4*c3*c2*s1 - a6*s5*c4*c3*s2*s1 \\ & - s3*a3*c2*s1 - c3*a3*s2*s1 + a6*c5*c3*c2*s1 - a6*c5*s3*s2*s1; \end{aligned}$$

$$\begin{aligned} \text{Je}(3,17) = & c3*a3*c2 + d4*c3*s2 + d4*s3*c2 - s3*a3*s2 + a6*s5*c4*c3*c2 \\ & + a6*c5*c3*s2 + a6*c5*s3*c2 - a6*s5*c4*s3*s2; \end{aligned}$$

%J<sub>P,2</sub>

$$\text{Je}(1,18) = s3*a3*s1 - d4*c3*s1 + a6*s5*c4*s3*s1 - a6*c5*c3*s1 + a6*s5*s4*s2*c1;$$

$$\text{Je}(2,18) = a6*s5*s4*s2*s1 + a6*c5*c3*c1 - s3*a3*c1 - a6*s5*c4*s3*c1 + d4*c3*c1;$$

$$\text{Je}(3,18) = -a6*s5*s4*c2;$$

%J<sub>X,3</sub>

$$\text{Je}(1,19) = -s3*s2*c1 + c3*c2*c1;$$

$$\text{Je}(2,19) = -s3*s2*s1 + c3*c2*s1;$$

$$\text{Je}(3,19) = c3*s2 + s3*c2;$$

%J<sub>Y,3</sub>

$$\text{Je}(1,20) = s1;$$

$$\text{Je}(2,20) = -c1;$$

$$\text{Je}(3,20) = 0;$$

%J<sub>Z,3</sub>

$$\text{Je}(1,21) = s3*c2*c1 + c3*s2*c1;$$

$$\text{Je}(2,21) = s3*c2*s1 + c3*s2*s1;$$

$$\text{Je}(3,21) = -c3*c2 + s3*s2;$$

%J<sub>S,3</sub>

$$\begin{aligned} \text{Je}(1,22) = & -d4*s3*s2*c1 + d4*c3*c2*c1 - a6*s5*c4*c3*s2*c1 - a6*s5*c4*s3*c2*c1 \\ & - a6*c5*s3*s2*c1 + a6*c5*c3*c2*c1; \end{aligned}$$

$$\begin{aligned} \text{Je}(2,22) = & -d4*s3*s2*s1 - a6*s5*c4*s3*c2*s1 + d4*c3*c2*s1 \\ & - a6*s5*c4*c3*s2*s1 + a6*c5*c3*c2*s1 - a6*c5*s3*s2*s1; \end{aligned}$$

$$\begin{aligned} \text{Je}(3,22) = & d4*c3*s2 + d4*s3*c2 + a6*s5*c4*c3*c2 + a6*c5*c3*s2 + a6*c5*s3*c2 \\ & - a6*s5*c4*s3*s2; \end{aligned}$$

%J<sub>R,3</sub>

$$Je(1,23) = a6*s5*c4*s1+a6*s5*s4*s3*s2*c1-a6*s5*s4*c3*c2*c1;$$

$$Je(2,23) = -a6*s5*c4*c1+a6*s5*s4*s3*s2*s1-a6*s5*s4*c3*c2*s1;$$

$$Je(3,23) = -a6*s5*s4*s3*c2-a6*s5*s4*c3*s2;$$

%J<sub>P,3</sub>

$$Je(1,24) = -d4*s1+a6*s5*s4*c3*s2*c1-a6*c5*s1+a6*s5*s4*s3*c2*c1;$$

$$Je(2,24) = a6*c5*c1+a6*s5*s4*c3*s2*s1+a6*s5*s4*s3*c2*s1+d4*c1;$$

$$Je(3,24) = -a6*s5*s4*c3*c2+a6*s5*s4*s3*s2;$$

%J<sub>X,4</sub>

$$Je(1,25) = c4*c3*c2*c1+s4*s1-c4*s3*s2*c1;$$

$$Je(2,25) = c4*c3*c2*s1-s4*c1-c4*s3*s2*s1;$$

$$Je(3,25) = c4*s3*c2+c4*c3*s2;$$

%J<sub>Y,4</sub>

$$Je(1,26) = -s3*c2*c1-c3*s2*c1;$$

$$Je(2,26) = -s3*c2*s1-c3*s2*s1;$$

$$Je(3,26) = c3*c2-s3*s2;$$

%J<sub>Z,4</sub>

$$Je(1,27) = s4*s3*s2*c1-s4*c3*c2*c1+c4*s1;$$

$$Je(2,27) = -c4*c1-s4*c3*c2*s1+s4*s3*s2*s1;$$

$$Je(3,27) = -s4*s3*c2-s4*c3*s2;$$

%J<sub>S,4</sub>

$$Je(1,28) = -a6*s5*c4*s1-a6*s5*s4*s3*s2*c1+a6*s5*s4*c3*c2*c1;$$

$$Je(2,28) = a6*s5*c4*c1-a6*s5*s4*s3*s2*s1+a6*s5*s4*c3*c2*s1;$$

$$Je(3,28) = a6*s5*s4*s3*c2+a6*s5*s4*c3*s2;$$

%J<sub>R,4</sub>

$$Je(1,29) = -a6*s5*s3*c2*c1+a6*c5*s4*s1-a6*s5*c3*s2*c1+a6*c5*c4*c3*c2*c1 \\ -a6*c5*c4*s3*s2*c1;$$

$$\begin{aligned} \text{Je}(2,29) = & -a6*c5*s4*c1 - a6*s5*s3*c2*s1 - a6*s5*c3*s2*s1 \\ & - a6*c5*c4*s3*s2*s1 + a6*c5*c4*c3*c2*s1; \end{aligned}$$

$$\text{Je}(3,29) = a6*c5*c4*c3*s2 - a6*s5*s3*s2 + a6*c5*c4*s3*c2 + a6*s5*c3*c2;$$

%J<sub>P,4</sub>

$$\text{Je}(1,30) = a6*c5*s4*c3*c2*c1 - a6*c5*c4*s1 - a6*c5*s4*s3*s2*c1;$$

$$\text{Je}(2,30) = a6*c5*c4*c1 + a6*c5*s4*c3*c2*s1 - a6*c5*s4*s3*s2*s1;$$

$$\text{Je}(3,30) = a6*c5*s4*s3*c2 + a6*c5*s4*c3*s2;$$

%J<sub>X,5</sub>

$$\text{Je}(1,31) = c5*c4*c3*c2*c1 + c5*s4*s1 - s5*c3*s2*c1 - s5*s3*c2*c1 - c5*c4*s3*s2*c1;$$

$$\begin{aligned} \text{Je}(2,31) = & -c5*s4*c1 - c5*c4*s3*s2*s1 + c5*c4*c3*c2*s1 - s5*c3*s2*s1 \\ & - s5*s3*c2*s1; \end{aligned}$$

$$\text{Je}(3,31) = s5*c3*c2 - s5*s3*s2 + c5*c4*s3*c2 + c5*c4*c3*s2;$$

%J<sub>Y,5</sub>

$$\text{Je}(1,32) = s4*s3*s2*c1 - s4*c3*c2*c1 + c4*s1;$$

$$\text{Je}(2,32) = -c4*c1 - s4*c3*c2*s1 + s4*s3*s2*s1;$$

$$\text{Je}(3,32) = -s4*s3*c2 - s4*c3*s2;$$

%J<sub>Z,5</sub>

$$\begin{aligned} \text{Je}(1,33) = & c5*s3*c2*c1 + c5*c3*s2*c1 + s5*c4*c3*c2*c1 \\ & - s5*c4*s3*s2*c1 + s5*s4*s1; \end{aligned}$$

$$\begin{aligned} \text{Je}(2,33) = & -s5*c4*s3*s2*s1 + c5*c3*s2*s1 + c5*s3*c2*s1 \\ & - s5*s4*c1 + s5*c4*c3*c2*s1; \end{aligned}$$

$$\text{Je}(3,33) = c5*s3*s2 - c5*c3*c2 + s5*c4*s3*c2 + s5*c4*c3*s2;$$

%J<sub>S,5</sub>

$$\begin{aligned} \text{Je}(1,34) = & -a6*s5*s3*c2*c1 + a6*c5*s4*s1 - a6*s5*c3*s2*c1 + a6*c5*c4*c3*c2*c1 \\ & - a6*c5*c4*s3*s2*c1; \end{aligned}$$

$$\begin{aligned} \text{Je}(2,34) = & -a6*c5*s4*c1 - a6*s5*s3*c2*s1 - a6*s5*c3*s2*s1 \\ & - a6*c5*c4*s3*s2*s1 + a6*c5*c4*c3*c2*s1; \end{aligned}$$

$$\text{Je}(3,34) = a6*c5*c4*c3*s2 - a6*s5*s3*s2 + a6*c5*c4*s3*c2 + a6*s5*c3*c2;$$

%J<sub>R,5</sub>

$$Je(1,35) = 0$$

$$Je(2,35) = 0$$

$$Je(3,35) = 0$$

%J<sub>P,5</sub>

$$Je(1,36) = -a6*c4*s1-a6*s4*s3*s2*c1+a6*s4*c3*c2*c1;$$

$$Je(2,36) = a6*c4*c1-a6*s4*s3*s2*s1+a6*s4*c3*c2*s1;$$

$$Je(3,36) = a6*s4*c3*s2+a6*s4*s3*c2;$$

%J<sub>X,6</sub>

$$Je(1,37) = s6*s4*s3*s2*c1-s6*s4*c3*c2*c1-c6*s5*s3*c2*c1 \\ -c6*s5*c3*s2*c1+s6*c4*s1-c6*c5*c4*s3*s2*c1 \\ +c6*c5*s4*s1+c6*c5*c4*c3*c2*c1;$$

$$Je(2,37) = -c6*c5*c4*s3*s2*s1-c6*c5*s4*c1-s6*s4*c3*c2*s1 \\ -s6*c4*c1+s6*s4*s3*s2*s1-c6*s5*c3*s2*s1 \\ +c6*c5*c4*c3*c2*s1-c6*s5*s3*c2*s1;$$

$$Je(3,37) = c6*c5*c4*c3*s2-s6*s4*s3*c2-c6*s5*s3*s2 \\ +c6*c5*c4*s3*c2+c6*s5*c3*c2-s6*s4*c3*s2;$$

%J<sub>Y,6</sub>

$$Je(1,38) = c6*s4*s3*s2*c1+c6*c4*s1+s6*s5*c3*s2*c1-c6*s4*c3*c2*c1 \\ -s6*c5*s4*s1+s6*s5*s3*c2*c1-s6*c5*c4*c3*c2*c1+s6*c5*c4*s3*s2*c1;$$

$$Je(2,38) = s6*c5*c4*s3*s2*s1-c6*c4*c1+s6*c5*s4*c1-s6*c5*c4*c3*c2*s1 \\ +s6*s5*s3*c2*s1+c6*s4*s3*s2*s1+s6*s5*c3*s2*s1-c6*s4*c3*c2*s1;$$

$$Je(3,38) = s6*s5*s3*s2-c6*s4*c3*s2-s6*c5*c4*c3*s2-s6*s5*c3*c2-c6*s4*s3*c2 \\ -s6*c5*c4*s3*c2;$$

%J<sub>Z,6</sub>

$$Je(1,39) = c5*s3*c2*c1+c5*c3*s2*c1+s5*c4*c3*c2*c1-s5*c4*s3*s2*c1 \\ +s5*s4*s1;$$

$$Je(2,39) = -s5*c4*s3*s2*s1+c5*c3*s2*s1+c5*s3*c2*s1-s5*s4*c1 \\ +s5*c4*c3*c2*s1;$$

$$Je(3,39) = c5*s3*s2-c5*c3*c2+s5*c4*s3*c2+s5*c4*c3*s2;$$

%J<sub>S,6</sub>

$$Je(1,40) = 0;$$

$$Je(2,40) = 0;$$

$$Je(3,40) = 0;$$

%J<sub>R,6</sub>

$$Je(1,41) = 0;$$

$$Je(2,41) = 0;$$

$$Je(3,41) = 0;$$

%J<sub>P,6</sub>

$$Je(1,42) = 0;$$

$$Je(2,42) = 0;$$

$$Je(3,42) = 0;$$

A jacobiana de identificação, conforme calculada acima com 42 colunas, possui colunas linearmente dependentes e portanto não pode ser invertida. Para tanto, deletam-se as colunas: 3 5 9 11 15 17 21 23 27 29 32 33 34 35 40 41 42. A matriz reduzida possui 25 colunas linearmente independentes.

## Apêndice B

Neste apêndice estão as matrizes relacionadas a calibragem de ambas as câmeras que foram utilizadas nas simulações de visão computacional. Para a calibração, as câmeras foram posicionadas a uma distância fixa entre si, e foram obtidas imagens de uma plataforma de calibração com as duas câmeras posicionadas em posições e angulações diversas em relação a própria plataforma.

A Figura 41 mostra as imagens da plataforma, obtidas na câmera esquerda e direita. A Figura 42 mostra uma imagem com os cantos do tabuleiro de chadrez obtidos para identificação. A Figura 43 mostra as posições de onde foram posicionadas as câmeras.

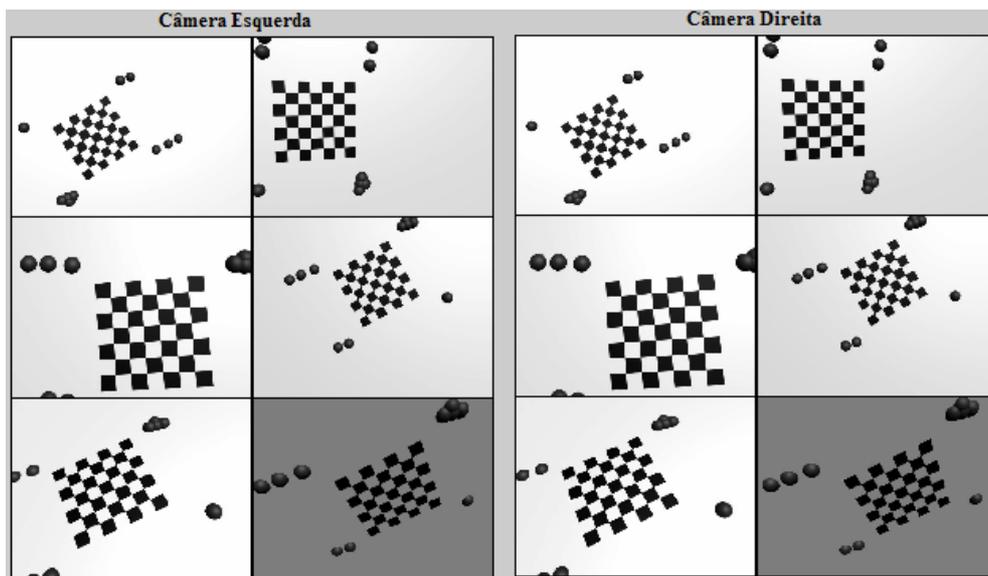


Figura 41: Imagens de Calibração tiradas da câmera esquerda e direita

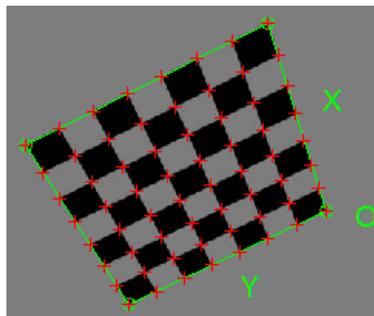


Figura 42: Pontos obtidos nas bordas da imagem

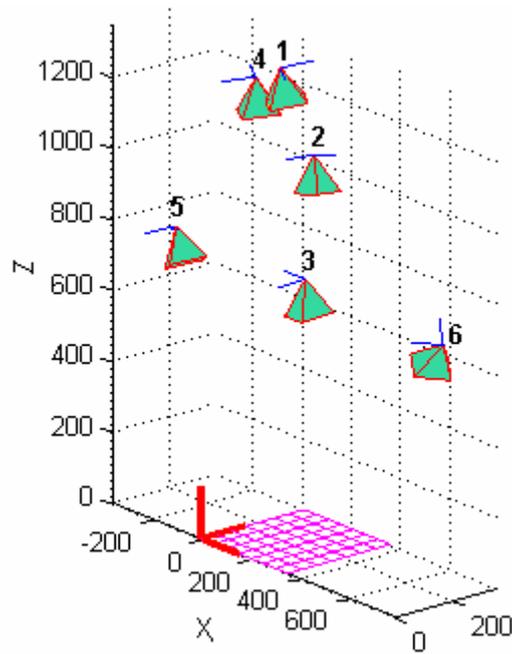


Figura 43: Posição das câmeras em relação ao tabuleiro.

Com base nestas imagens, foi utilizado o algoritmo de calibração, resultando nos seguintes parâmetros:

Câmera Esquerda:

Distância Focal: [622.722223 622.734845]

Ponto Principal: [319.645527 240.020607]

Rotação:: [0.995639 0.000059 -0.093285

0.000144 -1.000000 0.000901

-0.093285 -0.000910 -0.995639]

Translação: [-181.450138 95.014091 772.891183]

-----

Câmera Direita:

Distância focal: [622.003404 621.914437]

Ponto Principal: [319.848589 239.641100]

Rotação: [0.995651 -0.000019 -0.093157

0.000049 -1.000000 0.000734

-0.093157 -0.000735 -0.995651]

Translação: [-181.689910 115.513783 771.803544]

Os parâmetros extrínsecos (posição e orientação das câmeras) estão expressos em relação ao sistema de coordenadas do mundo, conforme explicado na seção 3.3. Pode-se a partir de uma transformação homogênea transferir o sistema de coordenadas do mundo para a posição da câmera esquerda, resultando nos seguinte parâmetros extrínsecos:

Câmera Esquerda:

Rotação:: [1 0 0

0 1 0

0 0 1]

Translação: [0 0 0]

-----

Câmera Direita:

Rotação: [1 0 0

0 1 0

0 0 1]

Translação: [50 0 0]