



DISSERTAÇÃO DE MESTRADO

**CALIBRAÇÃO CINEMÁTICA DE UM MANIPULADOR
INDUSTRIAL UTILIZANDO MÉTODO DE OTIMIZAÇÃO
PARAMÉTRICA DO MODELO DE ERROS**

PATRÍCIA DA SILVA OLIVEIRA

Brasília, Julho de 2019

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

**UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA MECÂNICA**

**CALIBRAÇÃO CINEMÁTICA DE UM MANIPULADOR INDUSTRIAL
UTILIZANDO MÉTODO DE OTIMIZAÇÃO PARAMÉTRICA DO
MODELO DE ERROS**

PATRÍCIA DA SILVA OLIVEIRA

**DISSERTAÇÃO SUBMETIDA AO DEPARTAMENTO DE
ENGENHARIA MECÂNICA DA FACULDADE DE TECNOLOGIA DA
UNIVERSIDADE DE BRASÍLIA COMO PARTE DOS REQUISITOS
NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM
SISTEMAS MECATRÔNICOS**

APROVADA POR:

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

Prof. Dr. Guilherme C. de Carvalho, PPMEC/UnB
Membro Interno

Prof. Dra. Marcela R. Machado, PCMEC/UnB
Membro Externo

BRASÍLIA/DF, 16 de Julho de 2019

Dedicatória

O temor do Senhor é o princípio do conhecimento, mas os insensatos desprezam a sabedoria e a disciplina. Pv. 1.7

À Deus, pela saúde, força, conhecimento, sabedoria e disciplina.

Aos meus amados pais, Francisco das Chagas e Maria do Rosário, por terem me propiciado a oportunidade de estudar.

À minha irmã Talita, por toda força e coragem demonstrados mediante os problemas de saúde que tem enfrentado.

Ao meu amor, Pedro Lopes, por todo o carinho, respeito e confiança.

PATRÍCIA DA SILVA OLIVEIRA

Agradecimentos

Agradeço primeiramente a Deus, por ser o meu baluarte e minha fortaleza, e por todo o entendimento e sabedoria a mim concedidos, possibilitando-me alcançar êxito na execução deste trabalho.

Aos meus queridos e amados pais, Francisco das Chagas e Maria do Rosário, por todo o amor e dedicação para comigo e por acreditarem no meu potencial.

Ao meu namorado, Pedro Lopes, pela sua compreensão e afabilidade, assim como pelas palavras de intrepidez todas as vezes em que pensei em desistir mediante as dificuldades.

Ao meu querido orientador, professor José Maurício Santos Torres Motta, pela sua imensurável paciência, pelo conhecimento compartilhado e pela confiança a mim depositada.

De forma especial, aos meus colegas de turma e de laboratório, Gilberto de Oliveira Costa e Leandro Pereira de Sousa, pelas inúmeras contribuições de seus conhecimentos nas áreas de robótica e programação.

Ao colega e aluno de doutorado, Cristhian Riaño, por ter disponibilizado parte do seu tempo para medir os quadrados do padrão de medição utilizado neste trabalho.

Aos técnicos de laboratório, Cláudio Pereira, Artur Alves e Alexandre Silveira, pela colaboração empreendida na montagem dos equipamentos experimentais.

Aos colegas do Laboratório de Robótica e Visão, Marlete Maria da Silva e Juan Sebastian Toquica Arenas, pelas palavras de apoio e pela amizade.

Um agradecimento especial ao Programa de Pós-Graduação em Sistemas Mecatrônicos (PPMEC), bem como à Universidade de Brasília, por disponibilizar espaço físico, computadores, robô, câmera e ferramentas, necessários para o desenvolvimento deste trabalho.

Por fim, deixo meus sinceros agradecimento à CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior), pelo apoio financeiro concedido por meio de bolsa de pesquisa, e à todos que, de alguma forma, contribuíram para a conclusão desta pesquisa.

PATRÍCIA DA SILVA OLIVEIRA

RESUMO

O número de aplicações envolvendo robôs tem aumentado no últimos anos em virtude dos investimentos em pesquisa robótica e dos avanços tecnológicos nas áreas de informática e eletrônica. Entretanto, um dos maiores gargalos da robótica industrial corresponde às diferenças existentes entre o modelo de controle e o modelo real do robô, limitando o uso da programação *off-line* (principal responsável por reduzir o tempo de inatividade da máquina). O sucesso dessa programação está associado à boa repetibilidade e precisão dos manipuladores, sendo este fator mais crítico. De maneira geral, existem duas formas de aproximar o modelo nominal do modelo real do robô. A primeira consiste em fabricar o robô mais próximo do projeto original, e, portanto, do modelo; a segunda maneira está associada ao melhoramento do modelo. Esse último método é denominado de calibração e tem sido importante na produção de robôs, bem como na operação e integração de robôs em sistemas de fabricação, uma vez que se trata de um método mais simples de ser implementado e com custo relativamente mais baixo. Nesse contexto, este trabalho tem como objetivo realizar a calibração cinemática do manipulador IRB-140, utilizando um método de otimização paramétrica do modelo de erros, a fim de melhorar sua precisão e, portanto, viabilizar a sua programação *off-line*. Para tanto, primeiramente foi desenvolvido e parametrizado um modelo cinemático; posteriormente, foi executada a etapa de identificação, onde foram detectados e eliminados os parâmetros que provocavam redundâncias e/ou deficiências no posto do Jacobiano; por último, com o modelo otimizado, foram identificados os valores dos parâmetros cinemáticos. A avaliação de desempenho do robô ocorreu por meio de técnicas de simulação computacional. Dentre os resultados obtidos com a simulação, chegou-se a um modelo otimizado com 24 parâmetros, proporcionando um número de condicionamento igual a 54. Além disso, foi possível averiguar que, acima de 30 pontos de medição, a melhoria na precisão do modelo de calibração é irrelevante.

ABSTRACT

The number of applications involving robots has increased in recent years due to investments in robotics research and technological advances in the areas of computing and electronics. However, one of the biggest bottlenecks in industrial robotics is the differences between the control model and the real robot model, limiting the use of offline text programming (primarily responsible for reducing machine downtime). The success of this programming is associated with the good repeatability and precision of the handlers, being this most critical factor. Generally speaking, there are two ways of approximating the nominal model to the actual robot model. The first is to manufacture the robot closest to the original design, and therefore the model; The second way is associated with model improvement. This latter method is called calibration and has been important in robot production as well as the operation and integration of robots in manufacturing systems as it is a simpler method to implement and relatively lower cost. In this context, this work aims to perform the kinematic calibration of the IRB-140 manipulator, using a parametric error model optimization method, in order to improve its accuracy and, therefore, make its *offline* programming possible. To this end, a kinematic model was first developed and parameterized; subsequently, the identification step was performed, where the parameters that caused redundancies and/or deficiencies in the Jacobian post were detected and eliminated. Finally, with the optimized model, the values of the kinematic parameters were identified. The performance evaluation of the robot occurred through computational simulation techniques. Among the results obtained with the simulation, we obtained an optimized model with 24 parameters, providing a conditioning number of 54. In addition, it was possible to verify that, over 30 measuring points, the improvement in the accuracy of the Calibration is irrelevant.

SUMÁRIO

RESUMO	i
ABSTRACT	ii
LISTA DE FIGURAS	iv
LISTA DE TABELAS	vi
1 Introdução	1
1.1 Estrutura da Dissertação.....	3
2 Revisão Bibliográfica	5
2.1 Programação <i>Off-Line</i>	5
2.2 Calibração de Manipuladores Industriais.....	8
2.3 Modelagem cinemática	10
2.3.1 Conceitos fundamentais de cinemática	10
2.3.2 Requisitos do modelo cinemático	17
2.3.3 Métodos de modelagem cinemática baseados em parâmetros geométricos de elos.....	18
2.3.4 Parametrizações	23
2.3.5 Singularidades e continuidade do modelo.....	24
2.3.6 Minimalidade	29
2.4 Métodos Numéricos para Identificação de Parâmetros.....	30
2.4.1 Método de Levenberg-Marquardt	33
2.4.2 Problemas inerentes ao processo de identificação.....	35
2.5 Otimização do Modelo.....	35
2.5.1 Decomposição em valores singulares	35
2.5.2 Número de condicionamento	36
2.5.3 Escalonamento das colunas do Jacobiano	37
2.5.4 Dependências e não identificabilidades	38
2.5.5 Modelos ótimos.....	39
2.6 Sistemas de Medição	40

2.7	Implementação e Compensação de Posicionamento.....	44
2.8	Problemas de Calibração de Robôs	46
2.8.1	Calibração <i>hand-eye</i> (mão-olho)	46
2.8.2	Calibração <i>eye-to-hand</i> (olho-a-mão)	51
2.8.3	Calibração envolvendo vários robôs e sensor	53
3	Metodologia	57
3.1	Aspectos Metodológicos	57
3.2	Modelagem Cinemática do IRB-140	57
3.2.1	O manipulador industrial IRB-140.....	57
3.2.2	Atribuição dos sistemas de coordenadas.....	59
3.2.3	Transformações homogêneas para o IRB-140	60
3.3	Identificação de Parâmetros.....	63
3.3.1	Modelo cinemático para identificação de parâmetros	63
3.4	Otimização do Modelo de Identificação	67
3.5	Simulação do Processo de Calibração.....	67
4	Resultados e Discussão	73
4.1	Otimização do Modelo de Calibração.....	73
4.2	Processo de Simulação da Calibração.....	75
5	Considerações Finais	82
5.1	Conclusões	82
5.2	Trabalhos Futuros	83
5.3	Trabalhos Publicados.....	83
	REFERÊNCIAS BIBLIOGRÁFICAS	85

LISTA DE FIGURAS

2.1	Diagrama representativo dos tipos de calibração de robôs. Fonte: Adaptado de [10].	9
2.2	Tipos de juntas do par inferior. Fonte: [20].	11
2.3	Relação entre as cinemáticas direta e inversa. Fonte: Produzido pela Autora.	12
2.4	Parâmetros de Denavit-Hartenberg definidos para a junta i e elo i . Fonte: [30].	13
2.5	Transformações entre sistemas de coordenadas em uma cadeia cinemática aberta. Fonte: [28].	14
2.6	Manipulador com pulso esférico. Fonte: [28].	16
2.7	Modelo de parametrização proposto por Denavit-Hartenberg. Fonte: [11].	19
2.8	Representação de um elo com eixos de juntas paralelos. Fonte: [31].	20
2.9	Juntas rotacionais quase-paralelas. Fonte: [31].	21
2.10	Juntas prismáticas quase-paralelas. Fonte: [31].	22
2.11	Processo de medição com teodolitos. Fonte: [22].	43
2.12	Compensação de erros <i>off-line</i> : X_{targ} é a posição do <i>target</i> ; f_N é o modelo nominal do robô; q_N são as posições nominais das juntas; f_{cont} é o modelo do controlador; X_{comp} é a posição de compensação; q_{real} são as posições reais das juntas; f_{corr} é o modelo corrigido; X_{real} é a posição real. Fonte: Adaptado de [22].	45
2.13	Transformações na calibração <i>hand-eye</i> entre diferentes sistemas de coordenadas para as posições 1 e 2 do robô. Fonte: [50].	47
2.14	Método de calibração simultâneo proposto por Zhuang et al. [48]. Fonte: [49].	49
2.15	Configuração <i>eye-to-hand</i> na captura do objeto com auxílio de câmera. Fonte: [82].	51
2.16	Configuração geral de um sistema <i>eye-to-hand</i> com um ponteiro a laser montado no elemento terminal do robô e um objeto plano colocado na frente da câmera. Fonte: [83].	52
2.17	Vista superior de três robôs móveis com as calibrações desconhecidas formuladas como $AXB = YCZ$. Fonte: [86].	54
2.18	Transformações envolvidas no problema de calibração simultânea <i>hand-eye, tool-flange e robot-robot</i> . Fonte: Adaptado de [81].	55
2.19	Calibração de um sistema robótico híbrido (robô paralelo acoplado a um robô serial) formulado como $AXB = YCZ$. Fonte: [87].	56
3.1	Espaço de trabalho do IRB-140. Fonte: [89].	58

3.2	Sistemas de coordenadas e parâmetros geométricos do manipulador IRB-140. Fonte: Produzido pela Autora.	60
3.3	Fluxograma do algoritmo de calibração contendo a etapa de otimização do modelo. Fonte: Produzido pela Autora.	68
3.4	Fluxograma do algoritmo de simulação do processo de calibração. Fonte: Produzido pela Autora.	69
4.1	Número de condicionamento da matriz Jacobiana calculado durante a etapa de identificação como uma função do número de parâmetros. Fonte: Produzido pela Autora.	74
4.2	Número de iterações para convergência do algoritmo de Levenberg-Marquardt como uma função do número de parâmetros. Fonte: Produzido pela Autora.	75
4.3	Erro de posicionamento do modelo mínimo em função do erro do sistema de medição. Fonte: Produzido pela Autora.	77
4.4	Precisão do elemento terminal em função do número de pontos de medição. Fonte: Produzido pela Autora.	77
4.5	Curvas de precisão do elemento terminal para um modelo otimizado e um modelo não-otimizado em função do número de pontos de medição. Fonte: Produzido pela Autora.	78
4.6	Precisão de posicionamento obtido a partir da aplicação do modelo de calibração otimizado. Fonte: Produzido pela Autora.	79
4.7	Histograma dos desvios no TCP. Fonte: Produzido pela Autora.	80
4.8	Avaliação do modelo calibrado a cada iteração. Fonte: Produzido pela Autora.	81

LISTA DE TABELAS

2.1	Relação entre o tipo de programação e a precisão do robô.	7
2.2	Número de soluções <i>versus</i> a_i diferente de zero.	15
3.1	Especificações técnicas do robô IRB-140.	58
3.2	Parâmetros de modelagem do manipulador IRB-140.	61
3.3	Transformações elementares para o IRB-140.	61
3.4	Transformações paramétricas para o IRB-140.	66
3.5	Desvios adicionados aos parâmetros.	71
4.1	Dados obtidos durante a etapa de otimização do modelo cinemático.	73
4.2	Número de condicionamento para o modelo não otimizado.	79

Capítulo 1

Introdução

Durante muito tempo os robôs têm sido aplicados na indústria de manufatura com o intuito de substituir a mão-de-obra humana em tarefas simples, repetitivas e perigosas. Os investimentos em pesquisa robótica, bem como o avanço tecnológico crescente nas áreas de informática e eletrônica, melhoraram de forma significativa a precisão dos robôs, permitindo ampliar o número de suas aplicações. Todavia, um dos maiores desafios na robótica industrial de hoje é a discrepância entre o modelo nominal e o modelo real do robô, impossibilitando o uso da programação *off-line* na indústria [1], cuja importância tem crescido significativamente devido à necessidade de minimizar o tempo de inatividade da máquina, e, assim, melhorar a taxa de utilização do robô [2].

A programação *off-line* pode ser definida como a técnica de gerar um programa para o robô sem usar uma máquina real, apresentando inúmeras vantagens em relação ao método *on-line*. Entretanto, fazer uma programação *off-line* bem sucedida implica que os robôs tenham boa repetibilidade e exatidão[2]. Segundo Mooring et al. [3], o conceito de repetibilidade está associado à habilidade do manipulador retornar a uma posição que foi armazenada no espaço das juntas. A exatidão, por outro lado, é a capacidade do robô se mover para uma posição definida no espaço da tarefa.

Apesar dos conceitos de exatidão e repetibilidade muitas vezes serem confundidos, a repetibilidade está na ordem de 0,1 mm para boa parte dos robôs atuais, e a exatidão está na ordem de milímetros ou mesmo de centímetros, de forma que a razão entre exatidão/repetibilidade esteja frequentemente na faixa de 3 para 20 [4]. Segundo Karan e Vukobratović [5], é desejável que tanto a exatidão quanto a repetibilidade atinjam padrões elevados, porém, como a exatidão normalmente é inferior à repetibilidade, a baixa exatidão de um robô é considerada um problema mais sério.

Nesse sentido, existem duas formas para aproximar o modelo nominal do modelo real do robô, a fim de viabilizar a programação *off-line*. A primeira consiste em fabricar o robô mais próximo do projeto original (e, portanto, do modelo), implicando em menores tolerâncias de fabricação, ou, talvez, um novo *design*; a segunda maneira está associada ao melhoramento do modelo, ge-

ralmente aumentando sua complexidade [6]. Este último processo é denominado de calibração, e desempenha um papel cada vez mais importante na produção de robôs, assim como na operação e integração de robôs em sistemas integrados de fabricação, pois é um método mais simples de ser implementado e com custo relativamente mais baixo [7]. Segundo Ginani e Motta [1], a calibração corrige a falta de exatidão no posicionamento estático de robôs, e minimiza o risco de ter que modificar os programas de aplicações em virtude de pequenas mudanças (desgaste de peças, desvios dimensionais ou tolerâncias e efeitos de substituição de componentes) no sistema do robô. Isso é relevante, principalmente em aplicações que envolvam uma grande quantidade de pontos de tarefa.

De acordo com Karan e Vukobratović [5], a principal fonte de erros de posicionamento do robô está diretamente relacionada a seus parâmetros cinemáticos ou modelo nominal de sua geometria, que diz exatamente onde o robô será posicionado em função de seus parâmetros de controle. Campos [8] acrescenta que as imprecisões de fabricação, montagem, temperatura e desgaste dos elementos fazem com que o modelo real do robô seja diferente do seu modelo nominal, concebido inicialmente na fase preliminar de projeto. Com isso, Wang et al. [9] afirmam que uma das formas mais eficientes para melhorar a precisão do robô é a calibração precisa de seus parâmetros cinemáticos.

O processo de calibração de um robô compreende a modelagem, medição, identificação numérica das características físicas reais de um robô e implementação de um novo modelo. Contudo, esse procedimento pode se tornar caro e extenso, conforme o equipamento de medição e a quantidade de dados que precisam ser coletados. Dessa maneira, dentre as decisões a serem tomadas antes de realizar a calibração, estão aquelas relacionadas a qual instrumento de medição utilizar e ao número de medições a serem feitas [6].

Os métodos de calibração dos robôs requerem que o modelo cinemático seja completo, mínimo e contínuo, além de apresentar parâmetros identificáveis [10], [11], [6]. Motta e McMaster [6] reforçam que é necessário empregar um procedimento de otimização baseado em uma função objetivo, visando eliminar parâmetros dependentes e/ou redundantes do modelo cinemático e, conseqüentemente, melhorar a precisão de posicionamento e reduzir o número de pontos de medição.

Segundo Motta et al. [12], nas últimas décadas diferentes métodos de calibração de robôs foram publicados, no qual parte significativa dessas publicações está relacionada com a calibração estática, que permite realizar a compensação dos erros de posição no modelo cinemático, utilizando, todavia, um grande número de pontos de medição. Wang et al [13] ressaltam que, atualmente, a calibração de robôs ainda consiste em uma área ativa de pesquisa.

Nesse contexto, a presente pesquisa tem como objetivo geral implementar um sistema de calibração cinemática de um manipulador industrial, baseado em um método de otimização paramétrica do modelo de erros.

Com o intuito de alcançar o objetivo geral, foram definidos, como objetivos específicos:

- Obter as equações da cinemática direta para o manipulador IRB-140;
- Realizar a modelagem cinemática desse robô, usando os modelos de parametrização propostos por Denavit-Hartenberg e Hayati-Mirmirani;
- Eliminar os parâmetros dependentes e/ou redundantes do modelo cinemático a partir da técnica de otimização por decomposição em valores singulares do Jacobiano;
- Identificar o modelo nominal desse robô por meio do algoritmo de Levenberg-Marquardt;
- Simular o processo de calibração proposto;
- Avaliar o número de condicionamento do Jacobiano como uma função do número de parâmetros no modelo;
- Averiguar a precisão do elemento terminal como uma função do número de pontos de medição;
- Analisar a melhoria na identificação de parâmetros devido ao procedimento de otimização do modelo cinemático.

Portanto, a principal contribuição deste trabalho está relacionada à otimização do modelo cinemático durante o procedimento de identificação dos parâmetros, que inclui três etapas, por intermédio do qual dependências e/ou redundâncias são eliminadas do modelo: decomposição em valores singulares, escalonamento das colunas do Jacobiano e análise do número de condicionamento da matriz Jacobiana. Esse método proporciona melhorias na precisão do manipulador, e reduz o número de pontos a serem medidos quando comparado com um modelo de calibração não otimizado [14].

1.1 Estrutura da Dissertação

A presente dissertação está organizada em cinco capítulos, incluindo esta Introdução. A seguir, os demais capítulos são descritos brevemente.

O Capítulo 2 apresenta os fundamentos teóricos pertinentes à programação *off-line* e calibração de manipuladores. Além disso, são detalhados os métodos de modelagem cinemática, os princípios de parametrização do modelo, assim como os principais métodos numéricos de identificação dos parâmetros cinemáticos e os sistemas de medição comumente empregados no processo de calibração. Também são explicitados os problemas típicos de calibração de robôs. Adicionalmente, são abordados os principais trabalhos encontrados na literatura sobre as etapas de identificação de parâmetros e sistemas de medição (consideradas mais relevantes no processo

de calibração estática), bem como os trabalhos relativos aos problemas de calibração *hand-eye*, *eye-to-hand* e com múltiplos robôs e sensor.

O Capítulo 3 discorre sobre a metodologia adotada para atingir os objetivos propostos, e descreve sucintamente as equações da cinemática direta do manipulador IRB-140, assim como a modelagem cinemática livre de singularidades e o modelo de identificação de parâmetros para este robô. Neste capítulo também é detalhado o procedimento de simulação do processo de calibração.

O Capítulo 4 mostra e discute todos os resultados obtidos com a simulação da calibração cinemática do manipulador industrial IRB-140.

Por fim, o Capítulo 5 apresenta as considerações finais do trabalho e sugestões para trabalhos futuros.

Capítulo 2

Revisão Bibliográfica

Esse capítulo tem como objetivo apresentar os princípios básicos necessários à compreensão da presente pesquisa. Para tanto, inicialmente são apresentados os tipos de programação de robôs, com suas respectivas características, além da importância da programação *off-line*.

Em seguida, é definido o processo de calibração de robôs e suas etapas são descritas minuciosamente, iniciando pela modelagem cinemática, cuja seção compreende uma breve revisão sobre os conceitos de cinemática direta e inversa, apresentação dos requisitos do modelo cinemático, especificação dos métodos de modelagem cinemática, assim como das parametrizações, singularidades e continuidade do modelo e minimalidade. Na seção seguinte é abordada a etapa do processo de calibração denominada de métodos numéricos para identificação dos parâmetros, onde são discutidos os métodos lineares e não lineares para solucionar o problema de identificação, com ênfase no método de Levenberg-Marquardt. Além disso, os problemas de identificação de parâmetros também são pormenorizados.

Na próxima seção é detalhado um método para otimizar o modelo cinemático de calibração, utilizando a ferramenta numérica de Decomposição em Valores Singulares (SVD). Logo após, são mostrados diferentes sistemas de medição que podem ser usados na terceira etapa do processo de calibração de robôs. Depois é caracterizada a quarta etapa, formalmente intitulada de implementação e compensação de posicionamento, apresentando duas técnicas: uma baseada no conhecimento do modelo cinemático inverso corrigido, e outra denominada de *Método de Redefinição de Posicionamento*.

Por fim, na última seção são mostrados os principais problemas de calibração de robôs.

2.1 Programação *Off-Line*

Em um ambiente industrial, um robô pode ser programado de duas formas. A primeira e mais comum consiste em fazer com que um operador movimente o manipulador, utilizando um

equipamento chamado de *teach pendant*, por meio de um conjunto de configurações de juntas necessárias para execução da tarefa. Nesse método, cada configuração é armazenada no controlador do robô, e o operador manipula os valores das juntas para que o elemento terminal alcance a posição desejada. Esse tipo de programação é denominado de *teaching method* e caracteriza-se por ser bastante demorado, exigir um operador qualificado e gerar atrasos nas linhas de produção. A segunda forma de programação, chamada de programação *off-line*, compreende o desenvolvimento do programa do robô longe da linha de fabricação e a transferência deste para o robô. Esta programação pode ser realizada em laboratório, utilizando outro robô, ou com uma simulação do robô e seu ambiente de trabalho no computador [3].

Segundo Mooring et al. [3], a utilização da programação *off-line* é restringida pelas discrepâncias entre a repetibilidade e a exatidão dos manipuladores industriais, que são mecanismos empregados para avaliar a capacidade de um robô alcançar uma dada posição no espaço. No *teaching method*, o operador garante que o elemento terminal esteja na posição apropriada em cada ponto da tarefa. Consequentemente, a conversão do espaço da tarefa para o espaço da junta é feita em cada ponto-chave da tarefa, com a confirmação visual do operador. Como esses pontos-chaves são previamente ensinados, a capacidade do robô de atingir essas posições é medida pela sua repetibilidade. Por outro lado, a programação *off-line* pressupõe que uma descrição correta de uma posição no espaço de juntas pode ser obtida a partir da descrição do espaço de tarefas. A fim de ilustrar esse conceito, supondo que um manipulador tenha sido configurado em laboratório e usado para programar uma tarefa, em cada ponto de interesse dessa tarefa, o efetuador final foi movido e os deslocamentos de juntas foram gravados. Posteriormente, o programa gerado é movido para outro robô no chão de fábrica. Devido à imprecisão característica dos manipuladores, o robô irá para uma posição diferente, o que implica em dizer que programas não podem ser transferidos de uma máquina para outra.

Portanto, o sucesso da programação *off-line* está diretamente associado à repetibilidade e à exatidão dos manipuladores industriais [15], que, segundo Placzek e Piszczek [16], são características importantes que afetam o funcionamento dos robôs. De acordo com Driels e Pathre [17], a exatidão corresponde à capacidade do robô de se mover para um local no seu espaço de trabalho (especificado em termos de suas coordenadas globais), enquanto que a repetibilidade refere-se à capacidade do robô de retornar a um determinado local após ter sido previamente ensinado. Conforme Gao et al. [18], em aplicações industriais, a exatidão de posicionamento é um fator crítico, principalmente nas áreas de soldagem, montagem, etc. Uma outra característica de desempenho especificada por muitos fabricantes é a resolução, definida por Driels e Pathre [17] como a menor mudança de posição que o robô pode acarretar no elemento terminal. Contudo, como este conceito é difícil de quantificar, porque o menor movimento varia ao longo do espaço de trabalho, uma outra definição de resolução está associada à digitalização dos vários sinais que entram e saem do controlador do robô, ou seja, a resolução para uma determinada junta traduz-se pela menor mudança de sinal que pode ser detectado e atuado pelo controlador [3].

Nesse contexto, é importante ressaltar que usuários de robôs recorreram a uma terceira classe de programação chamada de *workspace feedback*. Dessa forma, para partes da tarefa que não demandam alta precisão, pode-se usar a *teaching method* ou a programação *off-line*; para aquelas que exigem alta precisão, o espaço de trabalho é instrumentado de maneira a fornecer a posição real do elemento terminal ao controlador, que poderá fazer qualquer correção necessária para execução da tarefa [3].

De maneira geral, os métodos de programação apresentados relacionam-se com a exatidão do manipulador. Na Tabela 2.1 é possível verificar essas relações.

Tabela 2.1: Relação entre o tipo de programação e a precisão do robô.

Tipo de programação	Aspecto mais significativo da precisão	Nível de exatidão
<i>teaching method</i>	Repetibilidade	Moderado a alto
<i>workspace feedback</i>	Resolução	Muito alto
programação <i>off-line</i>	Exatidão	Baixo

Fonte: Adaptado de [3].

O interesse pela programação *off-line* tem crescido gradativamente desde o seu advento no início da década de 1980. Tal fato é justificado pelas vantagens proporcionadas por este método de programação, dentre as quais pode-se citar a redução do tempo necessário para programar um sistema robótico automatizado. Nos métodos tradicionais de ensino de manipuladores, denominados de programação *on-line*, o robô é retirado da produção sempre que programas são alterados ou gerados [19]. Portanto, um sistema de programação *off-line* (que não provoca a parada do equipamento de produção, ou seja, o robô), pode ser definido como um ambiente de programação de robôs que foi estendido por meio de computação gráfica, de maneira que o desenvolvimento de programas para o manipulador pode ser feito sem acesso direto ao mesmo [20].

De acordo com Young e Pickin [19], o uso da programação *off-line* diminui o tempo de programação em até 85% para pequenos programas em lote, e reduz o tempo ocioso de um manipulador de 12 a 18 horas para 6,5 horas. A relevância dessa programação pode ainda ser observada quando se considera uma linha de soldagem típica com 30 robôs e 40 pontos de soldagem, que leva aproximadamente, 400 horas para ensinar os robôs a executarem suas tarefas [15]. Todavia, é importante ressaltar que, apesar desses benefícios, existem problemas relacionados com a precisão do modelo do robô e de seu respectivo espaço de trabalho, o que implica na limitação do emprego desse método.

Diante disso, é notória a necessidade de aplicar métodos objetivando melhorar a precisão dos manipuladores. Neste contexto, destaca-se o método de calibração de robôs, que é fácil de ser implementado, e o custo é relativamente baixo [7].

2.2 Calibração de Manipuladores Industriais

A calibração de robôs consiste em um conjunto de procedimentos para melhorar a exatidão do robô por meio da alteração do seu software de controle [21]. Do ponto de vista matemático, a calibração de robôs é definida como a sequência de estimativas não lineares de parâmetros, cujos erros desconhecidos são identificados pela minimização de uma função custo [12], e tornam o modelo matemático mais próximo dos dados experimentais considerados. Portanto, a calibração se resume à solução de um conjunto de equações não lineares, que pode ser resolvido por diferentes métodos.

Duelen e Schröer [22] definem o problema central da calibração como a seleção adequada de um modelo estático do robô, onde é importante a alta exatidão de posicionamento associada com baixas velocidades do robô. Roth et al. [23] propuseram uma classificação hierárquica, onde os níveis de modelagem e calibração são separados em três:

1. Nível de calibração de junta: neste nível, o principal objetivo é estabelecer o relacionamento correto entre o sinal do sensor de junta e o deslocamento da junta real;
2. Calibração do modelo cinemático: o objetivo deste nível de calibração é melhorar o modelo cinemático, assumindo que o robô é formado por elos idealmente rígidos e juntas não elásticas.
3. Calibração não geométrica (não cinemática): neste nível, os desvios do modelo cinemático ideal devido a efeitos, tais como conformidade de junta, folga e conformidade de elos, são considerados.

Segundo Liu et al. [7], o processo de calibração de robôs divide-se em duas categorias: calibração cinemática (compensação de erros geométricos) e calibração dinâmica (compensação de erros não geométricos), indicados na Fig.2.1.

De acordo com Motta [24], os erros geométricos correspondem a cerca de 90 % de todos os erros do robô, o que justifica o fato de o processo de calibração cinemático, também chamado de estático, ser amplamente empregado. De maneira geral, manipuladores seriais apresentam erros de posicionamento entre 2 e 10 mm (mesmo sendo novos), e, após serem calibrados de forma adequada, esses erros podem ser reduzidos para valores inferiores a 0,5 mm.

Conforme Motta [6], a calibração compreende quatro etapas:

1. Modelagem Cinemática: obtenção de um modelo matemático que representa a geometria do robô e seu respectivo movimento;
2. Medição: medição da posição e orientação do efetuador final em coordenadas globais;

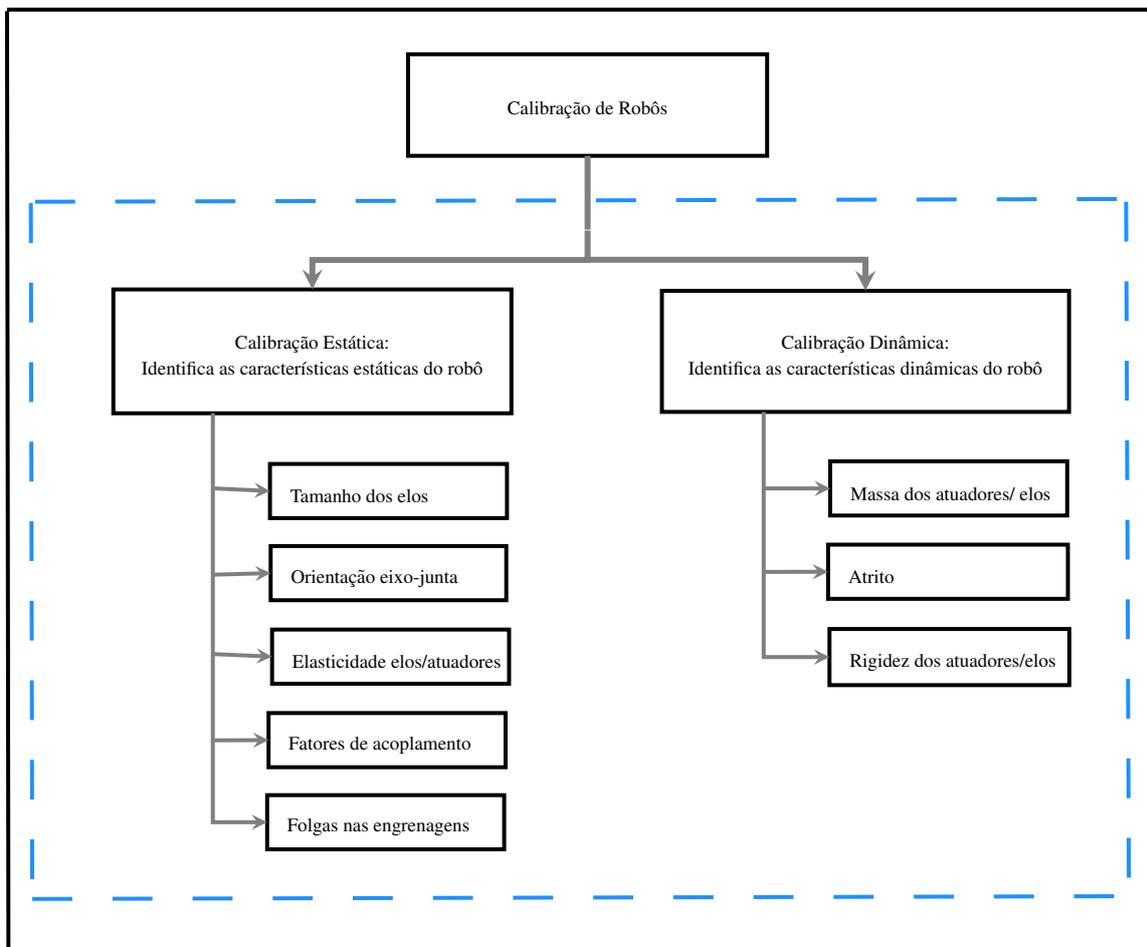


Figura 2.1: Diagrama representativo dos tipos de calibração de robôs. Fonte: Adaptado de [10].

3. Identificação dos Parâmetros: identificação dos valores dos parâmetros do modelo;
4. Implementação ou Compensação: modificação dos comandos de controle, a fim de possibilitar uma execução bem-sucedida de uma tarefa programada.

Motta e McMaster [25] descrevem, de forma mais sucinta, que a modelagem cinemática consiste na obtenção de um modelo matemático representativo da geometria e do movimento do robô; durante a etapa de medição, no entanto, características geométricas específicas do robô são medidas, com o auxílio de equipamentos de medição com precisão conhecida; no terceiro passo (identificação), por outro lado, os erros dos parâmetros do modelo são identificados; por fim, o modelo é corrigido no sistema de controle do robô, visando obter êxito na realização de tarefas programadas.

As quatro etapas características da calibração serão detalhadas nas próximas seções.

2.3 Modelagem cinemática

O primeiro passo para realizar a calibração de um robô denomina-se modelagem cinemática, e consiste em obter o modelo cinemático que associa as saídas dos sensores das juntas do robô com a posição do seu elemento terminal [21]. Dessa forma, esse modelo é constituído por equações não lineares que relacionam os parâmetros geométricos nominais e as variáveis de juntas com a posição e orientação do efetuador final do robô [26]. É importante destacar que a modelagem cinemática pode ser estendida para englobar uma célula de trabalho robótica inteira. Neste caso, as posições relativas de vários componentes do sistema são estabelecidos de forma única [21].

2.3.1 Conceitos fundamentais de cinemática

A cinemática é a ciência que estuda o movimento sem considerar as forças que o causam, ocupando-se da posição, velocidade, aceleração e de todas as derivadas de ordem superior da posição (tanto em relação ao tempo quanto em relação às outras variáveis). Portanto, o estudo da cinemática de manipuladores corresponde a todas as propriedades do movimento que sejam geométricas e baseadas no tempo [20].

De acordo com Ginani e Motta [1], um robô pode ser definido como um conjunto de elos que conectam o elemento terminal a sua base, com cada elo conectado ao próximo por meio de uma junta. A conexão entre um par de elos, quando o movimento relativo caracteriza-se por duas superfícies que deslizam uma sobre a outra, é denominado de **par inferior** [20], ilustrado na Fig. 2.2. Uma parte significativa dos manipuladores possuem juntas rotacionais ou prismáticas (também chamadas de deslizantes), cada uma apresentando um grau de liberdade. Em circunstâncias onde o manipulador é construído com uma junta com n graus de liberdade, ele

poderá ser modelado com n juntas com um grau de liberdade cada, e conectadas com $n-1$ elos de comprimento zero [20].

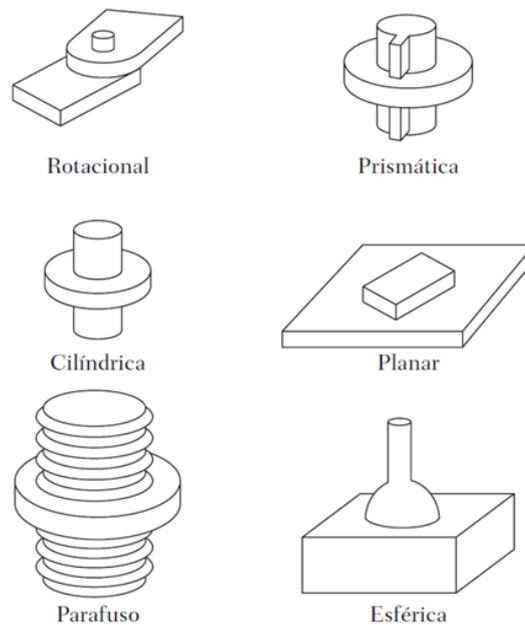


Figura 2.2: Tipos de juntas do par inferior. Fonte: [20].

Como manipuladores industriais possuem geometria complexa, o estudo de sua cinemática tem início com a atribuição de sistemas de coordenadas às várias partes do mecanismo, seguida pela descrição entre eles a partir de matrizes de transformações homogêneas. Um dos problemas centrais da cinemática consiste em compreender como a localização dos sistemas de referência se altera à medida que o mecanismo se articula [20]. Segundo Spong et al. [27], uma matriz de transformação homogênea é uma matriz de ordem 4×4 , constituída por uma matriz de rotação e um vetor de translação, cuja forma geral está representada na Eq. 2.1.

$$\mathbf{T}_N^0 = \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \vec{n} & \vec{s} & \vec{a} & \vec{p} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

Na Equação 2.1, $\mathbf{n} = (n_x, n_y, n_z)^T$, $\mathbf{s} = (s_x, s_y, s_z)^T$ e $\mathbf{a} = (a_x, a_y, a_z)^T$ são vetores que representam, a direção de x_1 , y_1 e z_1 , respectivamente, no sistema $o_0x_0y_0z_0$. O termo $\mathbf{p} = (p_x, p_y, p_z)^T$ corresponde ao vetor que relaciona a origem o_0 com a origem o_1 , expresso no sistema $o_0x_0y_0z_0$.

O modelo cinemático de posição pode ser solucionado por meio da cinemática direta ou inversa, que, por sua vez, depende das variáveis de entrada e de saída. Enquanto a cinemática direta permite calcular a posição e a orientação do elemento terminal a partir dos valores das variáveis

de juntas do mecanismo, conforme explícito pela Eq. 2.2, a cinemática inversa é empregada para computar os valores das variáveis de juntas para determinados valores de posição e orientação do elemento terminal, de acordo com a Eq. 2.3 [26].

$$[x, y, z, \alpha, \beta, \theta]^T = g(q_1, \dots, q_n) \quad (2.2)$$

$$[q_1, \dots, q_n]^T = f(x, y, z, \alpha, \beta, \theta) \quad (2.3)$$

Nas Equações 2.2 e 2.3, x , y e z são as coordenadas de posição e α , β e θ são as coordenadas de orientação (ambas do efetuador), enquanto que q_1, \dots, q_n são as variáveis das juntas do mecanismo. A Figura 2.3 demonstra a relação existente entre a cinemática direta e a cinemática inversa.

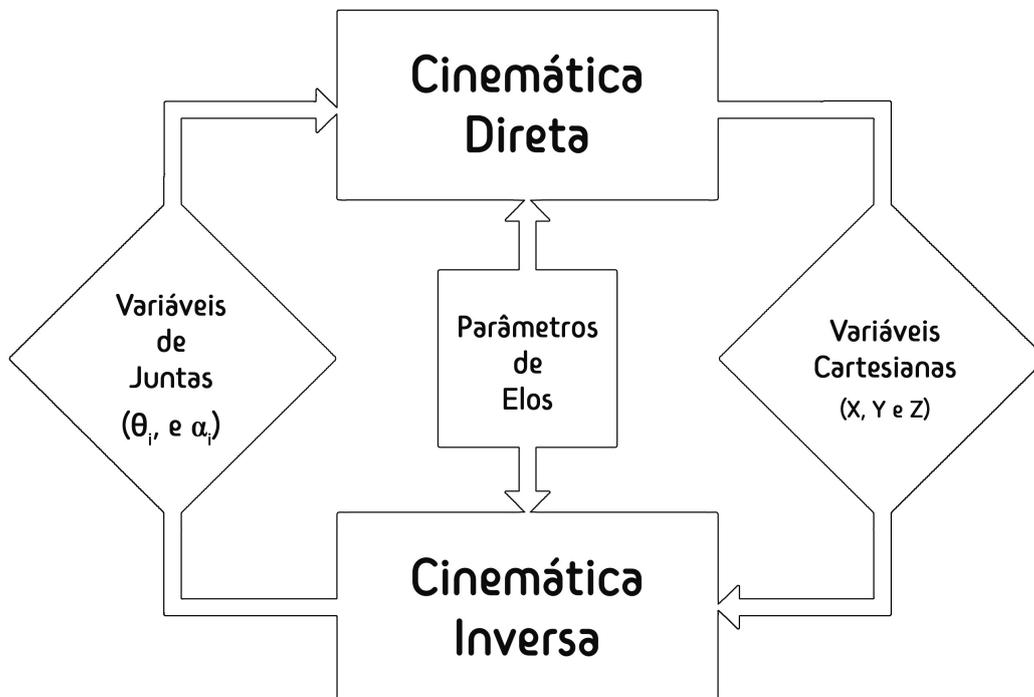


Figura 2.3: Relação entre as cinemáticas direta e inversa. Fonte: Produzido pela Autora.

As equações da cinemática direta são obtidas a partir da atribuição de sistemas de referência a cada elo do manipulador, utilizando uma determinada convenção para definir a posição e a

orientação relativa de dois elos consecutivos [28]. Um método amplamente usado para estabelecer sistemas de referências em aplicações robóticas é a convenção de Denavit-Hartenberg (DH) [27], [28], detalhado em [29].

Segundo Jazar [30], um sistema de coordenadas obtido pela notação de DH possui quatro parâmetros de elos, conforme indicado na Fig. 2.4.

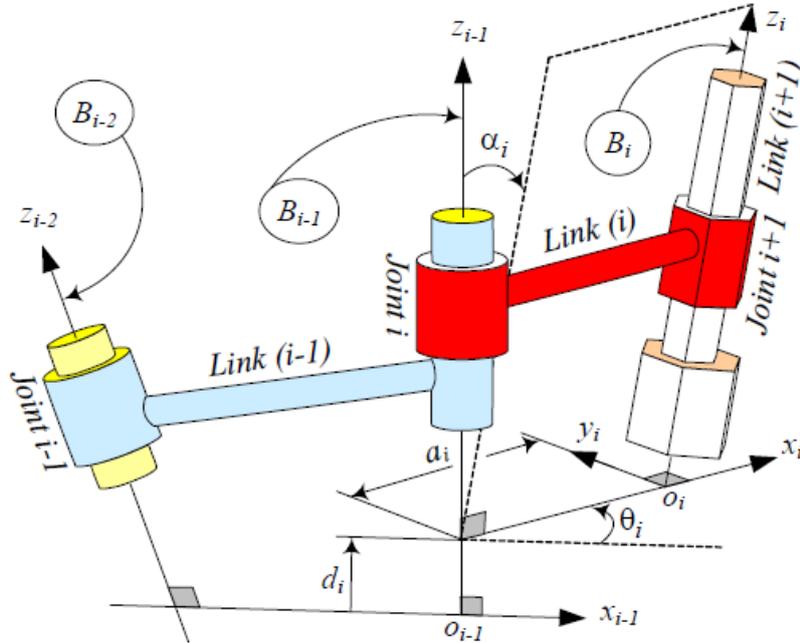


Figura 2.4: Parâmetros de Denavit-Hartenberg definidos para a junta i e elo i . Fonte: [30].

Os parâmetros de elos mostrados na Fig. 2.4 são interpretados como:

- a_i : distância, ao longo de x_i , de O_i à intersecção entre z_{i-1} e z_i , se z_{i-1} e z_i se cruzarem; ou a distância mais curta entre z_{i-1} e z_i , se estes forem paralelos ou não coplanares;
- α_i : ângulo, em torno de x_i , de z_{i-1} a z_i ;
- d_i : distância, ao longo de z_{i-1} , de O_{i-1} à intersecção entre z_{i-1} e x_i ; ou a distância mais curta entre x_{i-1} e x_i , se estes forem paralelos ou não coplanares;
- θ_i : ângulo, em torno de z_{i-1} , de x_{i-1} a x_i

Por meio dessa notação, cada elo pode ser descrito por uma transformação do sistema de coordenadas atual para o sistema de coordenadas anterior, de acordo com a Eq. 2.4.

$$T_i^{i-1} = R_z(\theta_i) \cdot T_z(d_i) \cdot T_x(a_i) \cdot R_x(\alpha_i) \quad (2.4)$$

Multiplicando os termos do lado direito da Eq. 2.4, tem-se a forma geral da matriz de transformação T_i^{i-1} , indicada pela Eq. 2.5.

$$T_i^{i-1} = \begin{bmatrix} \cos(\theta_i) & -\text{sen}(\theta_i) \cdot \cos(\alpha_i) & \text{sen}(\theta_i) \cdot \text{sen}(\alpha_i) & a_i \cdot \cos(\theta_i) \\ \text{sen}(\theta_i) & \cos(\theta_i) \cdot \cos(\alpha_i) & -\cos(\theta_i) \cdot \text{sen}(\alpha_i) & a_i \cdot \text{sen}(\theta_i) \\ 0 & \text{sen}(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$

Uma vez definidos os sistemas de referência para cada elo, e, conseqüentemente, obtidos seus respectivos parâmetros pela notação de DH, as equações da cinemática direta são obtidas pela substituição destes parâmetros na Eq. 2.5. A transformação isolada que relaciona a orientação e a translação do sistema de coordenadas N ao sistema de coordenadas O , conforme representado na Fig. 2.5, é obtida pela Eq. 2.6 [20], [28].

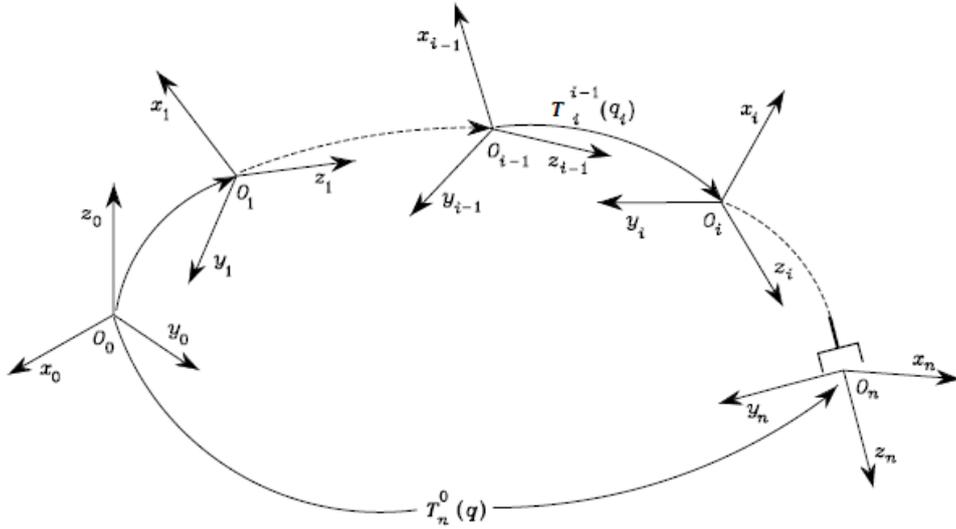


Figura 2.5: Transformações entre sistemas de coordenadas em uma cadeia cinemática aberta. Fonte: [28].

$$\mathbf{T}_n^0(\mathbf{q}) = \mathbf{T}_1^0(q_1) \cdot \mathbf{T}_2^1(q_2) \cdots \mathbf{T}_n^{n-1}(q_n) \quad (2.6)$$

A forma como são concatenadas as transformações de elos em um manipulador, indicada na Eq. 2.6, demonstra que o cálculo da função cinemática direta é recursivo e obtido pelo produto das matrizes de transformação homogênea $T_i^{i-1}(q_i)$ (para $i = 1, \dots, n$), onde cada matriz é função de uma única variável de junta. Como cada junta conecta dois elos consecutivos, é plausível considerar a descrição cinemática entre elos adjacentes, e em seguida obter a descrição geral da cinemática do manipulador de maneira recursiva [28].

O problema da cinemática inversa, por outro lado, é bem mais complexo que o da direta, devido às seguintes razões [28]:

- As equações a serem solucionadas normalmente são não-lineares e, portanto, nem sempre é possível obter uma solução analítica;
- Podem existir múltiplas soluções;
- Para um manipulador cinematicamente redundante, podem existir infinitas soluções;
- Dependendo da estrutura do manipulador, pode não haver soluções admissíveis.

De acordo com Siciliano et al. [28], a existência de soluções é garantida somente se a posição e a orientação do elemento terminal estiverem no espaço de trabalho destre, ou seja, no volume de espaço que o efetuador do robô consegue alcançar em todas as orientações. O número de soluções, entretanto, depende do número de graus de liberdade e do número de parâmetros de DH não nulos; em geral, quanto maior o número de parâmetros não nulos, maior o número de soluções admissíveis. Por exemplo, para um manipulador com seis juntas rotacionais, a Tab. 2.2 mostra a relação entre o número de soluções e quantidade de parâmetros de comprimentos de elos (a_i) iguais a zero.

Tabela 2.2: Número de soluções *versus* a_i diferente de zero.

a_i	Número de soluções
$a_1 = a_3 = a_5 = 0$	≤ 4
$a_3 = a_5 = 0$	≤ 8
$a_3 = 0$	≤ 16
Todos $a_i \neq 0$	≤ 16

Fonte: Adaptado de [20].

Analisando a Tabela 2.2 é possível constatar que, quanto maior a quantidade de parâmetros diferentes de zero, maior é o número máximo de soluções. Portanto, para um manipulador de seis graus de liberdade com juntas rotacionais e completamente genérico, existem até dezesseis soluções possíveis [20]. A existência de várias soluções para uma mesma junta, implica a necessidade de estabelecer um critério para escolha de uma única solução (por exemplo: o caso do cotovelo para cima/ cotovelo para baixo). É importante ressaltar que, a existência de limites de juntas mecânicas pode reduzir o número de soluções permitidas para a estrutura real.

As estratégias de solução da cinemática inversa podem ser divididas em dois grandes grupos: soluções de forma fechada e soluções numéricas. A solução de forma fechada é um método baseado em expressões analíticas ou na solução de um polinômio de grau 4 ou menor, de forma que cálculos não iterativos são suficientes para determinar a solução. Ela se divide em dois

Com efeito, uma vez que a posição e a orientação do elemento terminal são especificados em termos de \mathbf{p}_e e $\mathbf{R}_e = [\mathbf{n}_e \mathbf{s}_e \mathbf{a}_e]$, a posição do punho pode ser calculada por meio da Eq. 2.7.

$$\mathbf{p}_W = \mathbf{p}_e - d_6 \cdot \mathbf{a}_e \quad (2.7)$$

Portanto, a cinemática inversa para um manipulador de seis graus de liberdade com pulso esférico pode ser resolvida de acordo com os seguintes passos:

1. Calcular a posição do pulso \mathbf{p}_W , de acordo com a Eq. 2.7;
2. Solucionar a cinemática inversa para θ_1, θ_2 e θ_3 ;
3. Calcular a matriz de rotação \mathbf{R}_3^0 ;
4. Calcular \mathbf{R}_6^3 a partir da Eq. 2.8;

$$\mathbf{R}_6^3(\theta_4, \theta_5, \theta_6) = (\mathbf{R}_3^0)^{-1}(\theta_4, \theta_5, \theta_6) \mathbf{R}_6^0 \quad (2.8)$$

5. Solucionar a cinemática inversa para a orientação $(\theta_4, \theta_5, \theta_6)$.

Fundamentado neste desacoplamento cinemático, é possível resolver, separadamente, a cinemática inversa para o braço e para o pulso esférico [28].

2.3.2 Requisitos do modelo cinemático

No processo de calibração, a etapa de identificação de parâmetros exige que o modelo cinemático obedeça a três requisitos [11], [24], [1].

- **Completude:** caracteriza-se como a capacidade de um modelo cinemático descrever todas as possíveis configurações geométricas espaciais de um robô [1];
- **Continuidade:** está relacionada com a sensibilidade do modelo, isto é, pequenas mudanças em valores de juntas devem resultar em pequenas mudanças nos parâmetros descritivos [11];
- **Minimalidade:** corresponde ao menor número de parâmetros que um modelo cinemático deve possuir, a fim de evitar redundâncias [11], [1].

Segundo Schröer et al. [11], a continuidade do modelo é importante, pois garante que os métodos de otimização não linear funcionem de forma confiável. O quesito de minimalidade, por

outro lado, não é tão significativo, uma vez que a precisão matemática do seu conteúdo só é evidente quando se analisam técnicas numéricas para identificação de parâmetros e solução de problemas de identificabilidade e confiabilidade de valores de parâmetros. Portanto, declarações pertinentes à precisão matemática da minimalidade do modelo, só podem ser realizadas para relações geométricas definidas pelo modelo cinemático, não se aplicando ao modelo de deformações elásticas que engloba partes não geométricas do modelo do robô.

O emprego do modelo cinemático na calibração de robôs demanda algumas condições adicionais [11]:

- Se o sistema de coordenadas atual não é o sistema de referência da base do robô, mas um sistema pertencente a uma outra junta, a parametrização deve iniciar com uma rotação ou uma translação em torno do eixo z ;
- A parametrização do TCP para a última junta e da primeira junta para a base do robô deve ser distinta daquela aplicada às demais juntas adjacentes;
- É necessário que as localizações geométricas das juntas, descritas pelo modelo cinemático, coincidam com suas localizações físicas, garantindo, dessa forma, a integração de um modelo de deformação-elástico.

2.3.3 Métodos de modelagem cinemática baseados em parâmetros geométricos de elos

Segundo [2], a inexistência de um único método de modelagem cinemática, que pode ser aplicado aos diferentes tipos de geometrias dos robôs, está relacionada a razões topológicas fundamentais concernentes a mapeamentos de vetores euclidianos para esferas. Todavia, problemas topológicos relacionados à cinemática inversa e singularidades, podem ser solucionados pela disponibilidade de vários métodos que permitem definir funções inversas em situações específicas.

As primeiras abordagens de calibração surgiram com o intuito de identificar os parâmetros utilizados nos controladores dos robôs. Como nos algoritmos de controle eram empregadas equações da cinemática inversa, erros de comprimentos dos elos e de posição da junta zero conseguiam ser identificados, enquanto pequenos desvios nas orientações relativas das juntas (que interferem diretamente na precisão da posição do robô), não eram identificados [11]. A fim de solucionar este problema, Denavit e Hartenberg [29], propuseram uma notação baseada em matrizes, e que usa quatro parâmetros de transformações, para obter a cinemática de mecanismos do par inferior. Do ponto de vista matemático, esse modelo está representado na Eq. 2.9.

$$\mathbf{T}(\theta, d, a, \alpha) = \mathbf{R}_z(\theta) \cdot \mathbf{T}_z(d) \cdot \mathbf{T}_x(a) \cdot \mathbf{R}_x(\alpha) \quad (2.9)$$

Muitos autores utilizaram unicamente essa convenção na modelagem cinemática de manipuladores até perceberem que juntas sucessivas e quase paralelas violam o quesito de continuidade do modelo, como pode ser observado na Fig. 2.7. Isso sugere que, nas proximidades de uma configuração de juntas paralelas, existem configurações de juntas que produzem grandes valores aleatórios nos parâmetros de DH [11].

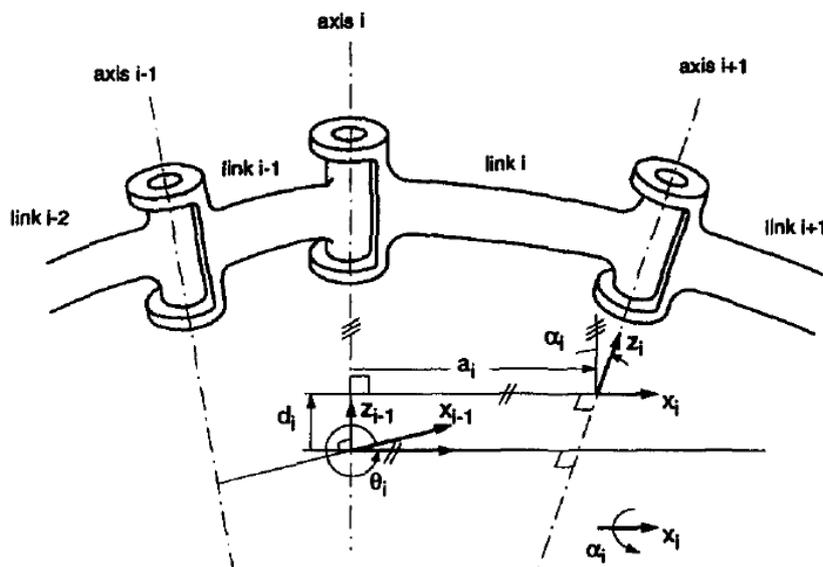


Figura 2.7: Modelo de parametrização proposto por Denavit-Hartenberg. Fonte: [11].

Em decorrência das limitações características da convenção de DH, Hayati e Mirmirani, em 1985, propuseram uma parametrização para duas juntas rotativas ou prismáticas, e que são paralelas ou quase-paralelas, popularmente chamada de convenção HM [11].

Em seu trabalho, Hayati e Mirmirani reforçam que, no caso de juntas consecutivas, cujos eixos são paralelos ou quase-paralelos, pequenas variações nos parâmetros de elos não correspondem a pequenas variações no sistema de coordenadas dos elos [31]. Isso pode melhor ser compreendido por meio de um exemplo, ilustrado na Fig. 2.8.

Aplicando as regras de Denavit e Hartenberg ao elo da Fig. 2.8, verifica-se que os parâmetros de elos são, respectivamente: $\theta_i = 0$, $d_i = 0$, $a_i = L$ e $\alpha_i = 0$. Admitindo que, devido às tolerâncias do processo de fabricação, o eixo z_i ficou desalinhado por um pequeno ângulo β , o novo eixo será chamado de z'_i , e interceptará o eixo z_{i-1} em alguma distância da origem do sistema de coordenadas $i - 1$. Partindo dessa premissa, os parâmetros reais do elo i são: $\theta_i = -90^\circ$, $d_i = -f$ (onde f é um escalar positivo, medido de 0 até o ponto onde z_{i-1} intercepta z'_i), $a_i = 0$ e $\alpha_i = -\beta$. Desse exemplo, pode-se aferir que, um pequeno erro no alinhamento do eixo z provoca um grande erro nos parâmetros de elos [31].

Para determinar de maneira correta esses parâmetros, Hayati e Mirmirani [31] propuseram novas regras, que encontram-se listadas a seguir:

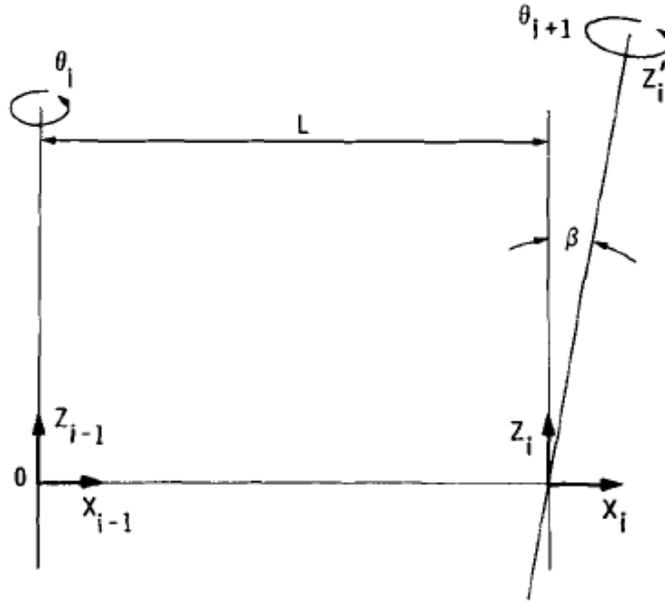


Figura 2.8: Representação de um elo com eixos de juntas paralelos. Fonte: [31].

1. Inicialmente, deve-se criar um plano perpendicular ao eixo z_{i-1} , conforme se observa na Fig. 2.9, onde a intersecção do plano com o eixo da junta $i+1$ é O_i ;
2. Depois, deve-se rotacionar o sistema de coordenadas $i-1$ em torno de z_{i-1} para alinhar x_{i-1} com a linha que conecta os pontos O_{i-1} e O_i ;
3. Após isso, é necessário transladar a origem do último sistema de coordenadas para O_i ;
4. Por último, e não menos importante, realizam-se duas rotações em torno dos eixos x e y do sistema de coordenadas resultante, a fim de alinhar o eixo z com o eixo da junta $i+1$.

Essas transformações podem ser expressas, matematicamente, pela Eq. 2.10, ou, na forma matricial, pela Eq. 2.11, onde $C = \cos()$ e $S = \sin()$.

$$T_i^{i-1} = R_z(\theta) \cdot T_{x'}(d) \cdot R_{x''}(a) \cdot R_{y'''}(\beta) \quad (2.10)$$

$$T_i^{i-1} = \begin{bmatrix} C(\theta_i)C(\beta_i) - S(\theta_i)S(\alpha_i)S(\beta_i) & -S(\theta_i)C(\alpha_i) & C(\theta_i)S(\beta_i) + S(\theta_i)S(\alpha_i)C(\beta_i) & a_iC(\theta_i) \\ S(\theta_i)C(\beta_i) + C(\theta_i)S(\alpha_i)S(\beta_i) & C(\theta_i)C(\alpha_i) & S(\theta_i)S(\beta_i) - C(\theta_i)S(\alpha_i)C(\beta_i) & a_iS(\theta_i) \\ -C(\alpha_i)S(\beta_i) & S(\alpha_i) & C(\alpha_i)C(\beta_i) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.11)$$

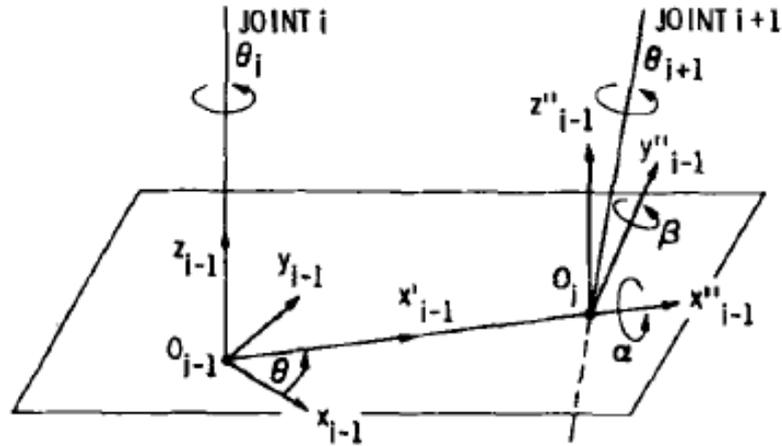


Figura 2.9: Juntas rotacionais quase-paralelas. Fonte: [31].

De maneira semelhante, equações são obtidas quando juntas prismáticas consecutivas possuem eixos paralelos ou quase-paralelos [31]. Nesta circunstância, o sistema de coordenadas i é obtido a partir das seguintes regras:

1. Primeiro, marca-se um ponto O_i no eixo da junta $i+1$, e desenha-se uma linha que passa por O_i e seja paralela ao eixo da junta prismática i , como esboçado na Fig. 2.10;
2. A partir do método de Denavit-Hartenberg, deve-se encontrar o sistema de coordenadas $i-1$, de tal maneira que o seu eixo z seja colinear com a linha desenhada na etapa anterior;
3. Depois, a origem do sistema de coordenadas $i-1$ precisa ser transladada ao longo do eixo z até o ponto O_i ;
4. Por fim, e da mesma forma que para juntas rotacionais, o eixo z precisa ser alinhado com o eixo da junta $i+1$ por meio de duas rotações sobre os eixos x e y dos sistemas de coordenadas resultantes.

Do ponto de vista matemático, estas regras podem ser escritas na forma da Eq. 2.12.

$$T_i^{i-1} = T_z(d) \cdot R_{x'}(\alpha) \cdot R_{y''}(\beta) \quad (2.12)$$

A convenção proposta por Hayati e Mirmirani [31], com suas respectivas transformações espaciais, garante que pequenas variações na localização e orientação de juntas podem ser modeladas por pequenas variações nos parâmetros. Nos casos onde os eixos de juntas adjacentes são perpendiculares ou quase perpendiculares, a notação de DH, por sua vez, é utilizada [31].

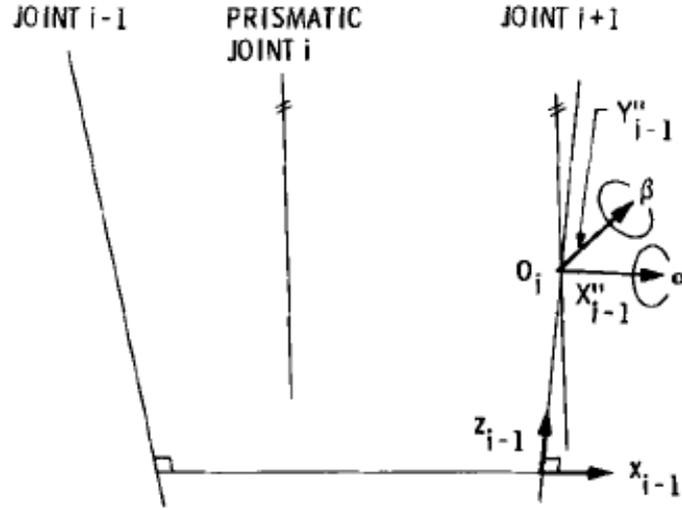


Figura 2.10: Juntas prismáticas quase-paralelas. Fonte: [31].

Veitschegger e Wu [32] desenvolveram um modelo para juntas paralelas consecutivas e consideraram erros de segunda ordem, objetivando descrever a precisão de posicionamento do robô. Esse modelo combina as abordagens de DH e HM ao incluir a transformação de rotação $R_y(\beta)$ às transformações de DH. Dessa forma, pequenas variações na posição e orientação de dois elos adjacentes sempre podem ser modelados por pequenas variações nos cinco parâmetros de elos: θ_i , α_i , a_i , d_i e β_i . Essa parametrização, assim como a correspondente matriz de transformação, estão especificadas, respectivamente, pela Eq. 2.13 e Eq. 2.14.

$$T_i^{i-1} = R_z(\theta) \cdot T_z(d) \cdot T_{x'}(a) \cdot R_{x'}(\alpha) \cdot R_{y''}(\beta) \quad (2.13)$$

$$T_i^{i-1} = \begin{bmatrix} C(\theta_i)C(\beta_i) - S(\theta_i)S(\alpha_i)S(\beta_i) & -S(\theta_i)C(\alpha_i) & C(\theta_i)S(\beta_i) + S(\theta_i)S(\alpha_i)C(\beta_i) & a_iC(\theta_i) \\ S(\theta_i)C(\beta_i) + C(\theta_i)S(\alpha_i)S(\beta_i) & C(\theta_i)C(\alpha_i) & S(\theta_i)S(\beta_i) - C(\theta_i)S(\alpha_i)C(\beta_i) & a_iS(\theta_i) \\ -C(\alpha_i)S(\beta_i) & S(\alpha_i) & C(\alpha_i)C(\beta_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.14)$$

Zhuang et al. [33] também propuseram um método para modelagem cinemática de manipuladores, denominado de Modelo Paramétrico Contínuo e Completo (CPC), cuja continuidade é obtida por uma linha de representação livre de singularidades baseada em quatro parâmetros. A completude, por outro lado, é alcançada pela adição de dois parâmetros elos, a fim de permitir a atribuição arbitrária dos sistemas de coordenadas. Uma vantagem dessa convenção, é que a transformação do sistema de coordenadas global para o sistema de coordenadas da base do robô assim como a transformação do sistema de coordenadas da ferramenta para o sistema de coordenadas do último elo podem ser modeladas com as mesmas regras aplicadas aos elos internos

da cadeia cinemática aberta. Além disso, considerando que todos os parâmetros redundantes podem ser eliminados no modelo CPC, é possível construir um modelo de erro linearizado do robô, no qual os parâmetros de erros são independentes e compreendem todo o espaço de erros geométricos. Essas características tornam o modelo CPC eficaz para a calibração de robôs. É importante destacar que a convenção CPC possibilita obter uma solução analítica, constituída de dois estágios, para o problema de identificação dos parâmetros cinemáticos.

Posteriormente, em 1993, Zhuang et al. [34], apresentaram uma modificação do modelo CPC, chamado de MCPC, que o aproxima do modelo DH. Essa parametrização produz um modelo de erro simplificado, pois emprega dois ângulos para acomodar a rotação necessária para cada transformação de elo, ao invés de vetores de direção nos eixos das juntas. Somado a isso, para a última transformação de elo, são empregados três parâmetros angulares e três parâmetros lineares. Assim como o modelo CPC, o MCPC consegue descrever completamente a geometria e o movimento do robô em um sistema de coordenadas de referência. Seu modelo de erro possui um número mínimo de parâmetros para abranger todo o espaço de erro geométrico, e, uma vez selecionado adequadamente o eixo da ferramenta, ele se torna livre de singularidades. A singularidade proveniente da ferramenta ocorre quando seu eixo é perpendicular ao eixo da última junta. Neste caso, isso pode ser evitado alterando o eixo z do sistema de coordenadas da ferramenta ou inserindo um sistema de coordenadas intermediário, que é fixado no sistema de coordenadas da ferramenta e tem uma orientação distinta em relação a este sistema.

Como pode ser verificado, os modelos cinemáticos discutidos não podem ser generalizados e aplicados para todas as geometrias de robôs existentes, entretanto, diferentes métodos podem ser usados em conjunto, com o intuito de determinar se as funções inversas podem ou não ser definidas em vários subconjuntos de espaços operacionais, além de possibilitar construir essas funções em situações específicas [25]. Baseado nisso e objetivando assegurar que o modelo de identificação seja mínimo, completo e contínuo, Schröer et al. [11] desenvolveram um conjunto de 17 modelos de parametrizações, incluindo critérios simples para decidir quais parametrizações devem ser empregadas para realizar a modelagem de um robô. Nas próximas subseções serão apresentados detalhes dessa abordagem livre de singularidade.

2.3.4 Parametrizações

A estrutura cinemática dos robôs pode ser descrita por um conjunto \mathbf{E} de transformações, chamado de Grupo Euclidiano, que preserva comprimentos e ângulos, e permite especificar a relação espacial entre os sistemas de coordenadas de juntas consecutivas, os sistemas da base e da primeira junta, e entre o último sistema de coordenadas e do TCP. O conjunto \mathbf{E} consiste em um múltiplo diferenciável de dimensão 6, obtido pelo produto do espaço vetorial real tridimensional com o grupo de rotações espaciais, conforme mostrado pela Eq. 2.15.

$$\mathbf{E} = \mathbb{R}^3 \times SO(3) \quad (2.15)$$

Uma parametrização do Grupo Euclidiano \mathbf{E} pode ser definida como um sêxtuplo $P = (P_1, \dots, P_6) \in \mathbf{P}^6$ de transformações elementares junto com um subconjunto não vazio, aberto e conectado W de $\mathbb{R}e^6$, de maneira que, para o mapeamento φ_P definido por P , e representado pela Eq. 2.16, a imagem de W através de φ_P é um submúltiplo de seis dimensões de \mathbf{E} , indicado pela Eq. 2.17.

$$\varphi_P = \prod_{i=1}^6 P_i(p_i) \quad (2.16)$$

$$\dim(\varphi_P(W)) = 6 \quad (2.17)$$

Na Equação 2.17, o subconjunto W contém todos os parâmetros de rotação e translação. A definição apresentada anteriormente é válida para qualquer representação especial dos elementos de \mathbf{E} , como é o caso das matrizes de transformação homogêneas. Por isso, para uma melhor compreensão deste trabalho, φ_P será referido com T .

2.3.5 Singularidades e continuidade do modelo

Como já mencionado, em um modelo de parametrização, um ponto de descontinuidade corresponde a uma transformação cujos parâmetros descritivos sofrem grandes alterações com pequenas mudanças na posição e orientação [2]. Sob a perspectiva matemática, a continuidade do modelo corresponde à continuidade da função inversa \mathbf{T}^{-1} , onde \mathbf{T} é o produto das transformações elementares (rotação e translação) entre as juntas. Fundamentado nisso, a singularidade da parametrização pode ser representada como uma transformação $\mathbf{T}_s \in \mathbf{E}$, onde o vetor de parâmetros $\mathbf{p} \in \mathbb{R}^6$ existe, de tal maneira que o posto do Jacobiano $\mathbf{J}_s = dTs/dp$ é menor que 6 e está indicado na Eq. 2.18, [2].

$$\mathbf{T} = \mathbf{T}_s \text{ e o posto}(\mathbf{J}_s) < 6 \quad (2.18)$$

Em outras palavras, as singularidades da parametrização \mathbf{T} pode ser investigada a partir da detecção dos zeros do determinante $\det(\mathbf{J}^T \cdot \mathbf{J})$ em função do vetor \mathbf{p} [2].

Segundo Motta [2], a partir das parametrizações do Grupo Euclidiano, os resultados enumerados abaixo podem ser demonstrados:

1. O conjunto de singularidades nunca é vazio e engloba o limite do conjunto de transformações não pertencentes à imagem de parametrização;

2. A continuidade do modelo é uma propriedade local.

Uma breve investigação das principais convenções de modelagem cinemática será apresentada a seguir.

a. Grupo Euclidiano

A transformação entre elos no Grupo Euclidiano é formada por uma parte translacional e por uma parametrização *roll-pitch-yaw* [14], sendo representada pela Eq. 2.19.

$$P = (T_x, T_y, T_z, R_z, R_y, R_x) \quad (2.19)$$

Estendendo essa parametrização para cada elo, tem-se a Eq. 2.20.

$$T = T_x(p_x), T_y(p_y), T_z(p_z), R_z(\gamma), R_y(\beta), R_x(\alpha) \quad (2.20)$$

Na Equação 2.20, p_x , p_y e p_z são coordenadas de translação e α , β e γ são coordenadas de rotação ao longo dos eixos, x , y e z , respectivamente. Expressando a Equação 2.20 na forma matricial, obtém-se a Eq. 2.21, onde $C = \cos()$ e $S = \sin()$.

$$\mathbf{T} = \begin{bmatrix} C\gamma C\beta & C\gamma S\beta S\alpha - S\gamma C\alpha & C\gamma S\beta C\alpha + S\gamma S\alpha & P_x \\ S\gamma C\beta & S\gamma S\beta S\alpha + C\gamma C\alpha & S\gamma S\beta C\alpha - C\gamma S\alpha & P_y \\ -S\beta & C\beta S\alpha & C\beta C\alpha & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.21)$$

A Equação 2.22 corresponde a uma forma mais simples da Eq. 2.21.

$$\mathbf{T} = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.22)$$

A ausência de parâmetros de rotação no vetor \mathbf{p} faz com que o Jacobiano se decomponha em duas submatrizes: uma de ordem 9×3 e outra de ordem 3×3 , como representado pela Eq. 2.23, cujas singularidades são os zeros da equação $\cos(\beta) = 0$ [11].

$$\mathbf{D}(x, y, z, \gamma, \beta, \alpha)\mathbf{T}(x, y, z, \gamma, \beta, \alpha) = \begin{bmatrix} \mathbf{O}_{9 \times 3} & \mathbf{D}(\gamma, \beta, \alpha)\mathbf{R}(\gamma, \beta, \alpha) \\ \mathbf{I}_3 & \mathbf{O}_{3 \times 3} \end{bmatrix} \quad (2.23)$$

Dessa forma, o conjunto de singularidades é composto por todas as matrizes de transformação homogêneas que possuem a forma da Eq. 2.24.

$$\mathbf{T}_s = \begin{bmatrix} 0 & o_x & a_x & p_x \\ 0 & o_y & a_y & p_y \\ \pm 1 & 0 & 0 & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.24)$$

A Equação 2.24 é obtida para $\text{Cos}(\gamma) = 0$ e $\text{Sen}(\beta) = \pm 1$. Todavia, recomenda-se um ângulo entre os eixos z menor que 45° ($\alpha, \beta < 45^\circ$) [11].

b. Convenção de Denavit-Hartenberg

A convenção de DH é uma parametrização em quatro dimensões de \mathbf{R} e, portanto, é empregada para modelar transformações entre juntas rotativas subsequentes. Seu conjunto de singularidades está indicado na Eq. 2.25 [11].

$$\mathbf{T}_s = \begin{bmatrix} n_x & o_x & 0 & p_x \\ n_y & o_y & 0 & p_y \\ n_z & o_z & \pm 1 & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.25)$$

A Equação 2.25 refere-se a um conjunto bidimensional, onde todos os elementos de \mathbf{R} representam juntas rotativas paralelas. Isso pode ser verificado a partir da observação da terceira coluna, que indica um eixo comum (z) no eixo anterior [2].

c. Convenção de Hayati

A convenção de Hayati possui um conjunto de singularidades complicado, uma vez que é constituído de duas partes, representadas na forma matricial pelas Eq. 2.26 e 2.27 [11], [2].

$$\mathbf{T}_s = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.26)$$

$$\mathbf{T}_s = \begin{bmatrix} n_x & o_x & \lambda_x & p_x \\ n_y & o_y & \lambda_y & p_y \\ n_z & o_z & \lambda_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.27)$$

Entretanto, essa parametrização não é completa, pois o conjunto que não pertence à imagem de parametrização não está vazio. O conjunto que não pode ser parametrizado está indicado na Eq. 2.28 [2].

$$\mathbf{T}_o = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & 0 & p_z \neq 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.28)$$

Estas equivalem a juntas rotativas, cujos eixos são ortogonais ou possuem uma orientação arbitrária e interceptam a origem do sistema de referência 2.27 [11].

d. Convenção de Veitschegger

A convenção de Veitschegger, por sua vez, possui 5 dimensões e não pode ser reduzida a uma parametrização mínima de \mathbf{R} , ou seja, a um modelo cinemático mínimo. Assim como a convenção de Hayati, possui dois conjuntos de singularidades, que estão especificadas pelas Eq. 2.29 e 2.30 [11], [11].

$$\mathbf{T}_s = \begin{bmatrix} n_x & o_x & a_x & 0 \\ n_y & o_y & a_y & 0 \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.29)$$

$$\mathbf{T}_s = \begin{bmatrix} n_x & o_x & a_x & \lambda a_x \\ n_y & o_y & a_y & \lambda a_y \\ n_z & o_z & 0 & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.30)$$

Essa convenção não engloba boas características das convenções de Denavit-Hartenberg e Hayati, uma vez que não é possível torná-la mínima [2].

e. Síntese

A Equação 2.31 contém os modelos completos e mínimos, discutidos até o momento, com seus respectivos intervalos de aplicação. No que se refere a estes intervalos, é importante destacar que a diferença entre ortogonal e paralelo é definido por um limite de 45° [10].

$$\left\{ \begin{array}{l} C \perp J_R : P = (R_z, T_z, T_x, R_x), \text{ convenção de DH} \\ C \parallel J_R : P = (R_z, T_x, R_x, R_y), \text{ convenção de Hayati} \\ C \perp J_T : P = (R_z, R_x), P = (R_z, R_y) \\ C \parallel J_T : P = (R_x, R_y), P = (R_y, R_x) \\ C \perp TCP : P = (T_x, T_y, T_z, R_z, R_y, R_z) \\ C \parallel TCP : P = (T_x, T_y, T_z, R_z, R_y, R_x) \end{array} \right. \quad (2.31)$$

Na Equação 2.31, as transformações podem ser modeladas a partir do sistema de coordenadas atual, chamado de C , para: um sistema de coordenadas alvo, cuja junta pode ser rotativa J_R ,

prismática J_T , ou o próprio TCP [11].

Respaldo no que foi mostrado até aqui, é possível obter um conjunto com 17 parametrizações, detalhadas a seguir, no qual alguns deles não são únicos, pois existem outras soluções possíveis capazes de satisfazer os requisitos. Transformações elementares que fazem parte de um submodelo do modelo mínimo, completo e contínuo a ser identificado estão destacadas em negrito [14].

- Transformação do sistema de coordenadas da base do robô para a primeira junta, quando esta é de translação:

$$B \perp J_T : P = (T_x, T_y, T_z, \mathbf{R}_z, \mathbf{R}_x) \quad (2.32)$$

$$B \parallel J_T : P = (T_x, T_y, T_z, \mathbf{R}_x, \mathbf{R}_y) \quad (2.33)$$

- Transformação do sistema de coordenadas da base do robô para a primeira junta, quando esta é de rotação:

$$B \perp J_R : P_x = (\mathbf{T}_y, \mathbf{T}_z, \mathbf{R}_z, \mathbf{R}_x, T_z) \quad (2.34)$$

$$B \perp J_R : P_y = (\mathbf{T}_x, \mathbf{T}_z, \mathbf{R}_z, \mathbf{R}_x, T_z) \quad (2.35)$$

$$B \parallel J_R : P_z = (\mathbf{T}_x, \mathbf{T}_y, \mathbf{R}_x, \mathbf{R}_y, T_z) \quad (2.36)$$

A Equação 2.34 é usada quando o eixo da junta estiver próximo do eixo x do sistema de coordenadas da base. Por outro lado, se o eixo da junta estiver próximo do eixo y do sistema de coordenadas da base, utiliza-se a Eq. 2.35.

- Transformação entre sistemas de coordenadas de juntas consecutivas:

$$J_R \perp J_R : P = (\mathbf{R}_z, \mathbf{T}_z, \mathbf{T}_x, \mathbf{R}_x, T_z) \quad (2.37)$$

$$J_R \parallel J_R : P = (\mathbf{R}_z, \mathbf{T}_x, \mathbf{R}_x, \mathbf{R}_y, T_z) \rightarrow \text{Suposição: os eixos das juntas não são idênticos.} \quad (2.38)$$

$$J_T \perp J_R : P = (\mathbf{T}_z, \mathbf{R}_z, \mathbf{T}_x, \mathbf{R}_x, T_z) \quad (2.39)$$

$$J_T \parallel J_R : P = (T_z, \mathbf{T}_x, \mathbf{T}_y, \mathbf{R}_x, \mathbf{R}_y, T_z) \quad (2.40)$$

$$J_T \perp J_T : P = (T_z, T_x, T_y, \mathbf{R}_z, \mathbf{R}_x) \quad (2.41)$$

$$J_T \parallel J_T : P = (T_z, T_x, T_y, \mathbf{R}_x, \mathbf{R}_y) \quad (2.42)$$

$$J_R \perp J_T : P = (\mathbf{R}_z, T_x, T_y, T_z, \mathbf{R}_x) \quad (2.43)$$

$$J_R \parallel J_T : P = (R_z, T_x, T_y, T_z, \mathbf{R}_x, \mathbf{R}_y) \quad (2.44)$$

- Transformação da última junta para o TCP:

$$J_T \perp TCP : P = (\mathbf{T}_z, \mathbf{T}_y, \mathbf{T}_x, [\mathbf{R}_z, \mathbf{R}_y, \mathbf{R}_z]) \quad (2.45)$$

$$J_T \parallel TCP : P = (\mathbf{T}_z, \mathbf{T}_y, \mathbf{T}_x, [\mathbf{R}_z, \mathbf{R}_y, \mathbf{R}_x]) \quad (2.46)$$

$$J_R \perp TCP : P = (R_z, \mathbf{T}_x, \mathbf{T}_y, \mathbf{T}_z, [\mathbf{R}_z, \mathbf{R}_y, \mathbf{R}_z]) \quad (2.47)$$

$$J_R \parallel TCP : P = (R_z, \mathbf{T}_x, \mathbf{T}_y, \mathbf{T}_z, [\mathbf{R}_z, \mathbf{R}_y, \mathbf{R}_x]) \quad (2.48)$$

Nas Equações 2.45, 2.46, 2.47 e 2.48, os parâmetros que estão entre colchetes só conseguem ser identificados quando medidas de orientação do TCP estiverem disponíveis na etapa de identificação [14], [11], [2].

2.3.6 Minimalidade

Para um modelo cinemático mínimo e completo, o número de parâmetros é determinado pela Eq. 2.49 [11].

$$n = 4 \cdot r + 2 \cdot t + 6 \quad (2.49)$$

onde r equivale ao número de juntas rotacionais, e t corresponde ao número de juntas prismáticas do manipulador. Na Equação 2.49, n também é o número máximo de parâmetros independentes do modelo cinemático completo.

Também é importante distinguir o número mínimo de parâmetros do modelo cinemático completo e o número de parâmetros identificáveis n_i . Se medidas de posição de juntas q e de posição e orientação do sistema de coordenadas do TCP estão disponíveis, o número máximo de parâmetros identificáveis n_i é igual ao número mínimo de parâmetros obtido pela aplicação da Eq. 2.49. Todavia, se houver somente dados de posição disponíveis, o número máximo de parâmetros identificáveis é calculado pela Eq. 2.50 [11].

$$n = 4 \cdot r + 2 \cdot t + 3 \quad (2.50)$$

Isso ocorre porque a orientação do sistema de coordenadas do TCP em relação ao último sistema de referência não pode ser identificada. Algumas abordagens consideram que o número de parâmetros em um modelo cinemático completo e mínimo é igual a $6 \cdot (N + 1)$ [11].

2.4 Métodos Numéricos para Identificação de Parâmetros

Dentre as etapas do processo de calibração, a identificação de parâmetros é a mais relevante, pois influencia diretamente no resultado final [7]. Matematicamente, a calibração de robôs consiste em uma sequência de estimativas não lineares de parâmetros, cujos parâmetros de erros desconhecidos são identificados pela minimização de uma função de custo [12], e tornam o modelo matemático mais próximo dos dados experimentais analisados. Para realizar essa estimativa, é necessário, inicialmente, considerar o manipulador robótico como uma função que, a partir de uma série de parâmetros de modelagem e das variáveis de juntas, determina uma posição específica. Isso implica em dizer que o problema da calibração fica reduzido à solução de um sistema de equações não lineares, que pode ser solucionado por diferentes métodos.

A identificação dos valores dos parâmetros do modelo do robô geralmente é considerada como um problema de otimização não linear de mínimos quadrados, que pode ser abordado de duas formas distintas [35]. Karan e Vukobratović [5], afirmam que a primeira forma de solucionar o problema é o método dos mínimos quadrados lineares, onde é considerado um modelo de erro linearizado, conforme mostra a Eq. 2.51.

$$\mathbf{e}_j = \mathbf{J}_j \delta \alpha, j = 1, \dots, k \quad (2.51)$$

Na Equação 2.51, \mathbf{e}_j corresponde ao vetor de erros de posicionamento, \mathbf{J}_j é a matriz Jacobiana do manipulador e $\delta \alpha$ é um vetor que contém os parâmetros a serem calibrados.

O sistema de equações representado pela Eq. 2.51 representa um sistema linear determinado que pode ser solucionado por técnicas conhecidas, como: algoritmo Hi'FI de Lawson e Hanson, ou decomposição QR [36]. Como a Equação 2.51 contém uma aproximação linear, o sistema é repetidamente resolvido enquanto as soluções convergem.

A segunda abordagem, por outro lado, é o método dos mínimos quadrados não lineares, onde o problema é tratado como um modelo de regressão não linear [5]. Os resíduos para esse modelo são expressos pela Eq. 2.52.

$$e_j(\alpha) = x_j - x(f(q_j, \alpha)), j = 1, \dots, k \quad (2.52)$$

onde x_j é a posição do robô, obtida por meio de medição, e $x(f(q_j, \alpha))$ descreve a posição e a orientação q_j atual do elemento terminal do robô, no qual α é o vetor dos parâmetros do modelo. A menor estimativa quadrada é o valor de α que minimiza a função objetivo descrita pela Eq. 2.53, no qual e_j^T é o vetor transposto do erro de posicionamento. Esta abordagem é empregada no algoritmo de Levenberg-Marquardt [37] para solucionar problemas de mínimos quadrados.

$$\sum_{j=1}^k = e_j^T \cdot e_j \quad (2.53)$$

A simulação de experimentos conduzidos por Pathre e Driels [35] mostrou que o método linear é quatro a oito vezes mais rápido que o método não linear. Por outro lado, o algoritmo de Levenberg-Marquardt é essencial para a convergência e para determinar os valores de parâmetros para os quais uma boa estimativa não está disponível [5].

Para Majarena et al. [26], o problema de otimização pode ser resolvido por meio de três métodos. O primeiro refere-se àqueles baseados em gradientes, normalmente empregados quando a função objetivo a ser minimizada é aproximada por meio de uma expansão em série de Taylor de primeira ordem. O segundo método, por sua vez, considera a expansão em série de Taylor de segunda ordem. Nesta abordagem, aparece a matriz Hessiana, formada por componentes que são as segundas derivadas da função objetivo em relação ao vetor de parâmetros. Esta matriz deve ser inversível e, caso a mesma apresente problemas relativos à singularidade, deve-se escolher um modelo matemático adequado e um conjunto de dados para a otimização, ou, então, utilizar métodos de otimização que evitem essa singularidade. Um dos métodos de mínimos quadrados mais empregado e que leva em consideração os termos de segunda ordem, é o Método de Gauss-Newton. Neste método, a matriz Hessiana deverá ser positiva ou semi-positiva, isto é, todos os seus valores precisam ser positivos ou positivos e zero, o que nem sempre será alcançado para qualquer modelo matemático e para qualquer conjunto de valores da função objetivo. Além disso, nem sempre a convergência será obtida.

O terceiro método de que trata Majarena et al. [26], corresponde ao algoritmo desenvolvido por Levenberg e Marquardt, sendo uma alternativa para os problemas numéricos presentes nos

dois métodos citados anteriormente. O método acrescenta um valor positivo λ nos elementos da diagonal principal da matriz Hessiana, obtendo uma matriz não singular e, portanto, reversível. A escolha deste valor implica em considerar a relação entre a velocidade de convergência do método, e a inversibilidade da matriz que deverá ser alcançada a cada iteração. É por isso que o algoritmo de Levenberg-Marquardt resulta da combinação dos dois métodos anteriores: quando o valor de λ aumenta, o algoritmo se comporta de forma semelhante ao método do gradiente; porém, quando os parâmetros se aproximam do seu valor ideal, o valor de λ diminui, comportando-se de forma similar ao método de Gauss-Newton, tanto na direção de busca, como no incremento de parâmetros.

Na literatura especializada, é possível encontrar alguns métodos alternativos que podem ser usados para evitar os problemas inerentes ao método de Gauss-Newton, como é o caso da decomposição em valores singulares (SVD), abordada por Wampler et al. [38], Agheli e Nategh [39], Bai e Teo [40]. Outra solução é o uso da decomposição QR para a identificação dos parâmetros [41].

O método de Levenberg-Marquardt foi amplamente empregado na calibração de robôs paralelos pelos autores Ryu e Rauf [42], Besnard e Khalil [41], Rauf e Ryu [43], Renaud et al. [44] e Huang et al. [45].

Goswami et al. [46] aplicaram o algoritmo de Levenberg-Marquardt, devido à sua robustez e habilidade para lidar com sistemas singulares, na calibração do manipulador PUMA 560, cuja cadeia cinemática tem seis graus de liberdade. Esse algoritmo também foi usado por Motta [25] na calibração dos manipuladores PUMA 500, IRB-2400 e IRB-6400. Ginani e Motta [1] empregaram o método de Levenberg-Marquardt na identificação dos parâmetros do robô IRB-2000, e, mais recentemente, Motta et al. [12], utilizaram este algoritmo na calibração de um robô de cinco graus de liberdade, desenvolvido para fazer reparos em pás de turbinas hidráulicas.

Zhuang e Roth [21] também propuseram um método numérico para identificação dos parâmetros. Esse método aproxima um modelo de erro cinemático por diferenças finitas, e segundo os autores, tem como vantagens, a facilidade de implementação e o fato de que, para muitas aplicações práticas, pode ser usado para substituir um modelo de erro diferencial sem perda de precisão ou estabilidade.

Horand e Dornaika [47], por sua vez, propuseram um método simultâneo iterativo baseado em quaternions e em otimização não-linear com o algoritmo de Levenberg-Marquardt. Estes dois métodos necessitam de uma estimativa da solução inicial e, dependendo da precisão das estimativas, pode não convergir para o ótimo global.

Zhuang et al. [48] ao estenderem a calibração *hand-eye* para também incluir a calibração da transformação *robot-world*, empregaram quaternions para derivar as soluções lineares explícitas. Dornaika e Horand [49] sugeriram duas novas soluções para o problema tratado por Zhuang et al. [48]: um método fechado, que usa a álgebra de quaternion e uma função de erro quadrático associada a essa representação; e um método baseado na minimização com restrições não-lineares, que

resolve rotação e translação de forma simultânea, a partir do algoritmo de Levenberg-Marquardt.

Daniilidis [50], em seu trabalho sobre calibração *hand-eye* usando quaternions duplos como função dos parâmetros de *screw*, descreve a solução do problema via Decomposição de Valores Singulares (SVD).

Em uma abordagem mais recente, Malti e Barreto [51], aplicaram decomposição em valores singulares para deduzir o quaternion correspondente à rotação R , enquanto a parte dupla do quaternion foi solucionada através de métodos clássicos de resolução de sistemas.

Segundo Bernhardt e Albright [10], os métodos numéricos empregados na identificação dos parâmetros do modelo constituem o núcleo do procedimento de calibração do robô. Motta e McMaster [6] acrescentam que a parte mais significativa dos procedimentos de identificação de parâmetros está relacionada com os métodos numéricos. Determinados procedimentos, cujos modelos de parâmetros são identificados a partir de várias posições do elemento terminal do robô, exigem a aplicação de métodos de otimização numérica. Esses métodos usam a linearização local do modelo não linear do robô e, em seguida, solucionam o problema de forma iterativa pelo método dos mínimos quadrados não linear.

De acordo com Majarena et al. [26], as técnicas baseadas em mínimos quadrados demandam menor custo de processamento computacional, desde que o valor inicial seja próximo da solução ideal para o conjunto de parâmetros. Entretanto, métodos baseados em algoritmos genéticos e redes neurais, são empregados para otimização e identificação de parâmetros quando não se sabe se os valores iniciais estão próximos da solução ideal. Somado a isso, esses métodos têm natureza estocástica, o que evita problemas na definição da direção de busca, inerentes aos métodos tradicionais de mínimos quadrados.

2.4.1 Método de Levenberg-Marquardt

Segundo Lima e Sampaio [52], o método de Levenberg-Marquardt assemelha-se ao método de Gauss-Newton, com a única diferença que é introduzido um fator de amortecimento μ .

Uma melhor compreensão do método de Levenberg-Marquardt implica em considerar, inicialmente, uma função não-linear representada pela Eq. 2.54, onde $\mathbf{X} \in \mathbb{R}^M$ e $\mathbf{Y} \in \mathbb{R}^N$ são vetores e $M \geq N$.

$$F(\mathbf{Y}) = \mathbf{X} \tag{2.54}$$

Com frequência é necessário estimar o vetor $\hat{\mathbf{Y}}$ que melhor se ajusta a um vetor $\hat{\mathbf{X}}$ medido. Esse problema pode ser reformulado como, dado o vetor $\hat{\mathbf{X}}$, determinar o valor de $\hat{\mathbf{Y}}$ que minimiza o erro $\|\epsilon\|$, sujeito à Eq. 2.55.

$$\hat{\mathbf{X}} = F(\hat{\mathbf{Y}}) + \epsilon \quad (2.55)$$

A partir de uma estimativa inicial para $\hat{\mathbf{Y}}$, o método de Newton refina a solução, assumindo a setença indicada na Eq. 2.56, onde \mathbf{J} é o Jacobiano, ou seja, $\mathbf{J} = \partial\mathbf{X}/\partial\mathbf{Y}$, e Δ corresponde a um pequeno incremento de $\hat{\mathbf{Y}}$. Portanto, a minimização do erro $\|\epsilon\|$ equivale a minimizar a expressão presente na Eq. 2.57, que é idêntica à expressão matemática da Eq. 2.58 [53].

$$F(\hat{\mathbf{Y}} + \Delta) = F(\hat{\mathbf{Y}}) + \mathbf{J}\Delta \quad (2.56)$$

$$\|\epsilon - \mathbf{J}\Delta\| \quad (2.57)$$

$$\mathbf{J}^T \mathbf{J}\Delta = \mathbf{J}^T \epsilon \quad (2.58)$$

A solução refinada, $\hat{\mathbf{Y}}_r$, é dada pela Eq. 2.59, que pode ser melhorada iterativamente.

$$\hat{\mathbf{Y}}_r = \hat{\mathbf{Y}} + \Delta \quad (2.59)$$

Em 1944, Levenberg [37], com o intuito de acelerar a convergência do algoritmo de Newton, propôs uma alteração para a Eq. 2.58, conforme indicado na Eq. 2.60.

$$(\mathbf{J}^T \mathbf{J} + \mathbf{I}\mu)\Delta = \mathbf{J}^T \epsilon \quad (2.60)$$

Inicialmente, na Equação 2.60, o valor de μ é igual a 10^{-3} . Entretanto, a cada iteração esse valor é modificado. Se a solução da Eq. 2.60 conduzir a uma redução do resíduo, μ é dividido por 10; caso contrário, multiplica-se μ por 10 [53].

Devido às instabilidades numéricas do método de Levenberg quando μ aumenta, Marquardt [54] propôs uma alteração no algoritmo outrora desenvolvido por Levenberg [37]. Nessa modificação, cada componente do gradiente é ponderado de acordo com sua curvatura, o que produz uma tendência de convergência na direção onde o gradiente é menor. Em termos matemáticos, isso corresponde à Eq. 2.61, e é conhecido como método de Levenberg-Marquardt, cuja complexidade aumenta com o número de variáveis [53].

$$(\mathbf{J}^T \mathbf{J} + \text{diag}(\mathbf{J}^T \mathbf{J})\mu)\Delta = \mathbf{J}^T \epsilon \quad (2.61)$$

2.4.2 Problemas inerentes ao processo de identificação

A identificação de parâmetros apresenta alguns problemas característicos que precisam ser resolvidos, de forma a garantir uma solução confiável para a calibração. Estes problemas estão descritos a seguir [10]:

1. As dependências entre parâmetros que provocam deficiências no posto do Jacobiano nem sempre podem ser constatadas a partir da inspeção do modelo do robô, uma vez que este engloba equações transcendentais. Como consequência, essas deficiências de posto produzem várias soluções para o problema numérico;
2. Problemas de identificação de modelos que envolvem parâmetros cinemáticos, de atuadores e elasticidade são mal condicionados do ponto de vista numérico, pois os valores a serem identificados correspondem a dimensões numéricas muito distintas, dependendo do tipo de robô e parâmetro;
3. Relações entre parâmetros dificultam a identificação, além de aumentar a sua incerteza. Mais uma vez, em virtude da complexidade do modelo, essas relações não podem ser determinadas diretamente a partir do próprio modelo;
4. Uma parcela mais significativa da incerteza dos parâmetros do modelo robótico identificado é proveniente da solução numérica, sobretudo de erros estocásticos típicos do sistema mecânico, enquanto que uma parcela menor advém dos erros de medição.

Esses problemas podem ser solucionados por meio de algumas técnicas de identificação numérica de parâmetros, baseadas no escalonamento do modelo de calibração [10].

2.5 Otimização do Modelo

A otimização do modelo cinemático, cuja finalidade é melhorar a etapa de identificação dos parâmetros por meio da eliminação de dependências ou redundâncias, compreende a decomposição em valores singulares, o escalonamento das colunas do Jacobiano e a análise do número de condicionamento da matriz Jacobiana [6]. Cada um desses dispositivos serão detalhados nas próximas subseções.

2.5.1 Decomposição em valores singulares

Em sistemas lineares onde o número de equações é igual ao número de incógnitas, é possível encontrar a solução empregando, por exemplo, os métodos de eliminação gaussiana ou fatoração

LU. Todavia, alguns sistemas podem perder a existência e/ou unicidade quando a matriz do sistema é singular, isto é, não tem inversa. Neste caso, uma ou mais linhas podem ser escritas como combinação linear das outras. Em procedimentos numéricos, pode acontecer agravantes decorrentes da maneira como os números são representados na máquina, bem como do acúmulo de erros de arredondamento. Um método numérico bastante eficiente na álgebra linear para lidar com sistemas de equações cujas matrizes são singulares, ou quase-singulares, é a Decomposição em Valores Singulares, conhecida como SVD (sigla herdada da sua denominação em inglês) [55].

Portanto, a SVD, empregada por [38], [40] e [39], consiste em uma técnica numérica utilizada na álgebra linear de problemas mal condicionados. Além disso, a SVD permite tratar problemas sobredeterminados (mais equações que incógnitas), fornecendo um diagnóstico de onde está a dificuldade para solucionar o problema, assim como provendo possíveis “soluções”, em algum sentido, quando a solução convencional não existe [55].

A ferramenta mais importante para analisar e otimizar o modelo de calibração e, consequentemente, o processo de identificação dos parâmetros, é a Decomposição em Valores Singulares (SVD) da matriz Jacobiana linearizada pelo método dos mínimos quadrados [14], [6], [12]. Aplicando essa ferramenta na matriz Jacobiana \mathbf{J} usada na identificação dos parâmetros, tem-se a Eq. 2.62.

$$\mathbf{J}_{3m \times p} = \mathbf{U}_{3m \times 3m} \cdot \mathbf{S}_{3m \times p} \cdot \mathbf{V}_{p \times p}^T \quad (2.62)$$

Na Equação 2.62, \mathbf{U} e \mathbf{V} são matrizes ortogonais unitárias e \mathbf{S} é uma matriz diagonal, cujos elementos da diagonal principal são os valores singulares não negativos, que podem ser organizados em ordem decrescente [14], [6]. A matriz \mathbf{S} está representada na Eq. 2.63.

$$\mathbf{S} = \begin{bmatrix} S_1 & & & & \\ & S_2 & & & \\ & & S_3 & & 0 \\ & & & \ddots & \\ & & & & S_r \\ & 0 & & & & 0 \end{bmatrix} \quad (2.63)$$

2.5.2 Número de condicionamento

Em problemas que envolvem matrizes, o número de condicionamento é importante, pois atua como um fator de amplificação do erro e análise de perturbação [10]. O número de condicionamento da matriz Jacobiana k_J é definido pela Eq. 2.64.

$$k(\mathbf{J}) = \|\mathbf{J}\| \cdot \|\mathbf{J}^+\| \quad (2.64)$$

Onde \mathbf{J}^+ corresponde à pseudo inversa de \mathbf{J} . Se a norma da matriz é derivada da norma vetorial Euclidiana, o número de condicionamento pode ser diretamente calculado pela Eq. 2.65, que relaciona o maior (S_1) e o menor (S_r) valores singulares diferentes de zero.

$$k(\mathbf{J}) = \frac{S_1}{S_r} \quad (2.65)$$

2.5.3 Escalonamento das colunas do Jacobiano

Segundo Bernhardt e Albright [10], problemas que envolvem mínimos quadrados são mal condicionados em virtude das diferenças de ordem de grandeza nos valores dos parâmetros. Por exemplo, um dado parâmetro de elasticidade pode ser da ordem de 10^{-6} e um parâmetro do atuador pode ter ordem igual a 10^{-9} , enquanto um parâmetro cinemático possui ordem de 10^{-3} . Isso produz um grande número de condicionamento da matriz Jacobiana (entre 10^6 e 10^9).

Neste sentido, o escalonamento das colunas do Jacobiano são adicionadas ao procedimento de otimização da identificação a fim de melhorar o condicionamento do problema. Modelos baseados em escalonamento de fatores podem ser estimados a partir de erros do robô (aproximadamente 1 mm). Esses modelos pode ser obtidos a partir da função $T(p, q)$, pois na aproximação de primeira ordem, a Eq. 2.66 é válida para o parâmetro p_i [14].

$$\frac{\partial}{\partial p_i} T(p, q) \cdot \Delta p_i = T(p + \Delta p_i, q) - T(p, q) \quad (2.66)$$

Na Equação 2.66, p_i corresponde ao parâmetro, Δp_i é o vetor de erros dos parâmetros e q é a posição do robô. Dessa forma, para uma posição q do robô, o valor singular escalonado é calculado pela Eq. 2.67, e refere-se à mudança de parâmetro que, quando empregado uma aproximação de primeira ordem, provoca um desvio de 1 mm no efetuator [14].

$$\sigma p_i(q) = \frac{10^{-3}}{\left\| \frac{\partial}{\partial p_i} T(p, q) \right\|} \quad (2.67)$$

Na Equação 2.67, $\sigma p_i(q)$ é o valor singular do parâmetro p_i na posição q . O termo no denominador é a norma da coluna i da matriz Jacobiana de identificação [6].

Se os valores $\sigma p_i(q)$ são calculados para um grande conjunto de posições, $q = [q_1, \dots, q_m]$, sem restrições de posição e orientação, os valores

$$\sigma p_i = \min_{q \in \{q_1, \dots, q_m\}} \{ \sigma p_i(q) \} \cong \min_{q \in \mathbb{R}^N} \{ \sigma p_i(q) \} \quad (2.68)$$

estão muito perto do desvio mínimo entre todas as possíveis posições. Os valores σp_i são chamados de valores extremos e são usados no escalonamento das colunas do Jacobiano [12].

2.5.4 Dependências e não identificabilidades

A confiabilidade e a precisão dos valores dos parâmetros do modelo identificados está diretamente associada à ausência de deficiências de posto e ao pequeno número de condicionamento do Jacobiano J , assim como à independência dos parâmetros do modelo, ou seja, sem redundância. Como já mencionado, a primeira etapa para a solução numérica do problema de identificação de parâmetros, consiste em obter um modelo cinemático do robô, que seja completo, mínimo e contínuo. Entretanto, a inclusão de parâmetros de elos, elásticos ou de atuadores, provoca o aparecimento de dependências e não identificabilidades que não são visíveis pela simples inspeção do modelo, mas que podem ser determinados por meio da decomposição em valores singulares do problema de mínimos quadrados escalonados e linearizados [10].

A Equação 2.62 pode ser reescrita na forma da Eq. 2.69.

$$\mathbf{J} \cdot \mathbf{V} = \mathbf{U} \cdot \mathbf{S} \quad (2.69)$$

Como as matrizes \mathbf{U} e \mathbf{V} são ortogonais, tem-se a Eq. 2.70

$$\|\mathbf{J} \cdot v_i\| = s_i \quad (2.70)$$

Se a matriz \mathbf{J} tem posto r , então $s_{r+1} = \dots = s_n = 0$. Cada posto deficiente corresponde a um valor singular igual a zero, e, conseqüentemente, a uma coluna do vetor v_i de \mathbf{V} que corresponde a uma combinação linear das colunas da matriz Jacobiana. A investigação do tamanho das entradas de v_i ($i = r+1, \dots, n$) possibilita decidir quais colunas do Jacobiano \mathbf{J} contribuem mais fortemente para a dependência linear das colunas de \mathbf{J} [10]. Aplicando isso ao modelo de parâmetros, tem-se as seguintes proposições:

1. Se o Jacobiano escalonado apresentar deficiências de posto, ou seja, $\text{posto}(\mathbf{J}) = r < n$, os parâmetros do modelo que são redundantes ou não identificáveis, são determinados por meio das entradas com os maiores valores absolutos dos vetores $v_i, i = r + 1, \dots, n$. Esse resultado pode ser estendido para englobar quase-dependências ou não identificabilidades, que provocam uma grande número de condicionamento [10];
2. Se o Jacobiano escalonado \mathbf{J} com $\text{posto}(\mathbf{J}) = r \leq n$ tem um grande número de condicionamento, os parâmetros que são quase-redundantes ou não identificáveis, são determinados a partir das entradas com o maior valor absoluto do vetor v_r [10].

Dessa forma, a investigação das colunas da matriz \mathbf{V} constituídas por pequenos valores singulares, é uma ferramenta essencial para obter modelos de calibração ótimos [10].

2.5.5 Modelos ótimos

No procedimento numérico de identificação de parâmetros é importante que os valores identificados sejam estáveis, precisos e confiáveis, uma vez que o objetivo é obter uma solução do problema de otimização não linear com [10]:

- Máxima melhoria da precisão de posicionamento;
- Confiabilidade máxima dos valores dos parâmetros identificados.

Os requisitos listados acima são contraditórios, porém, o modelo otimizado, sempre que possível, deve atender a ambos. É importante ressaltar que, o processo de otimização tratado nesta seção não é o mesmo procedimento de otimização da etapa de identificação numérica, mas sim um processo que modifica o modelo cinemático do robô que será empregado na identificação numérica de parâmetros [14].

O número de condicionamento, cujo conceito já foi discutido, é influenciado tanto pelo modelo do robô quanto pelo conjunto de posições considerado na identificação dos parâmetros. Por isso, é comum usar um gerador aleatório para determinar um conjunto de posições. Para assegurar a identificação, são aplicados critérios para analisar a diversidade de posições, bem como a distribuição igual de posições definida nas faixas de trabalho das juntas. Somado a isso, um procedimento numérico pode ser usado para investigar o efeito da visibilidade e das condições ambientais no número de condicionamento do problema de identificação [14].

Segundo Schröer [14], a otimização de modelos tem como pilares fundamentais a estimativa de erros e a seguinte regra: uma redução do modelo melhora o condicionamento numérico e diminui a precisão de posicionamento, ao passo que uma expansão do modelo aumenta o número de condicionamento e melhora a precisão de posicionamento. Dessa forma, a otimização é aplicável nas condições expressas a seguir:

- Somente para o ponto em que a precisão não venha a ser reduzida;
- Quantas vezes forem necessárias para que o número de condicionamento do problema de mínimos quadrados lineares seja menor que 100, pois pesquisadores da área de matemática já provaram que isso é necessário para produzir resultados confiáveis [14], [12].

De acordo com Schröer [14], o processo de otimização compreende:

1. O escalonamento do modelo para reduzir o número de condicionamento numérico k_J na ordem de centenas;
2. Determinar quase-dependências e não identificabilidades por meio da investigação da coluna do vetor v_r , referente ao menor valor singular s_r ;

3. Calcular os valores de sensibilidade;
4. Selecionar um parâmetro para ser excluído do procedimento de identificação, de modo que:
 - Tenha uma grande entrada em v_r ;
 - Não seja relevante para a precisão da posição;
 - Apresente uma variância que provoca uma grande incerteza na posição do TCP, conforme expresso pelos seus valores de sensibilidade.

Esse processo pode ser organizado em duas etapas [10]:

- 1º Eliminar as deficiências de posto;
- 2º Reduzir o número de condicionamento k_J para abaixo de 100;

Consequentemente, pode-se deduzir que um modelo ótimo O é derivado de um modelo completo M , a partir da exclusão de um pequeno número de parâmetros desse modelo durante a etapa de identificação [14].

2.6 Sistemas de Medição

Segundo Vasconcelos Filho [56], a etapa de medição, no que se refere a exatidão, é o procedimento mais crítico da calibração de robôs. Enquanto na calibração estática, envolve a medição do posicionamento estático do robô, na calibração dinâmica, engloba características como velocidade e aceleração de determinada junta durante o movimento.

Campos [8], afirma que os sistemas de medição são baseados em duas premissas: exatidão e repetibilidade. Goswami [46] ressalta a importância de distinguir entre a exatidão e a repetibilidade de um manipulador robótico, e define a repetibilidade de um robô como a precisão com a qual seu ponto terminal consegue atingir uma posição particular, sob comandos repetidos do mesmo conjunto de juntas angulares. A folga entre engrenagens, bem como a precisão do sensor e do servo, são exemplos de fatores que afetam a repetibilidade do robô.

Em relação à exatidão, Goswami [46] define como sendo a proximidade com a qual a posição real do robô corresponde à posição prevista pelo seu controlador. Um robô pode apresentar, simultaneamente, alta repetibilidade e baixa exatidão. Uma elevada repetibilidade é de primordial relevância para aplicações robóticas, como: pintura por pulverização e soldagem. Tarefas que envolvem programação *off-line*, por outro lado, dependem, fundamentalmente, da exatidão, além da repetibilidade do robô.

De acordo com Duelen e Schröer [22], os valores dos *encoders* (entrada do sistema) podem ser obtidos diretamente do controlador do robô. A posição e a orientação do ponto central da ferramenta (saída do sistema), por outro lado, precisam ser medidos por equipamentos de medição. Para fins de calibração, é necessário medir somente as posições do ponto central da ferramenta, uma vez que todos os parâmetros do modelo podem ser identificados a partir destes valores de posição. Isso é importante porque a medida de orientação é mais difícil de ser obtida e menos precisa que a medida de posição.

O êxito do processo de calibração depende do equipamento de medição. Do ponto de vista ideal, ele deve permitir a medição de posições alcançadas em um espaço de trabalho amplo, com uma exatidão cerca de dez vezes melhor que a exatidão de posicionamento do robô desejada [57]. Karan e Vukobratović [5] afirmam que é desejável que a instrumentação de medição facilite a detecção não tátil, de forma que seja eliminada qualquer influência de medição nas características de desempenho do robô. Dentre os dispositivos empregados para a medição dos valores de coordenadas do elemento terminal, destacam-se:

- *Laser tracker* – segundo Jia-Qing et al. [58], é um dispositivo de medição com grande tamanho e alta exatidão, formado por um *laser tracker*, um controlador, computador, um refletor e acessórios de medição. Com alta exatidão e eficiência, o dispositivo pode obter rastreamento em tempo real e com simples operação, porém, o custo financeiro para adquirir o equipamento é alto. Majarena et al. [26] destaca que o *laser tracker* possui precisão volumétrica de aproximadamente dezenas de micrometros;
- Câmera - equipamento de medição bastante empregado na calibração dos parâmetros cinemáticos de robôs. Devido seu baixo custo de aquisição e simples operação, a técnica de medição baseada em câmera é amplamente aplicada no âmbito industrial. Como consequência do intenso desenvolvimento da técnica de medição por câmera, muitos estudiosos têm escolhido esse dispositivo na calibração cinemática [58]. Majarena et al. [26] acrescentam que as câmeras, assim como os sensores de imagem 3-D, são compactas, robustas e apresentam flexibilidade e portabilidade. O rápido desenvolvimento desses dispositivos nas últimas décadas melhoraram de forma significativa a exatidão desses equipamentos;
- Infravermelho – baseia-se em uma análise estereoscópica através de duas câmeras localizadas a uma distância do diodo emissor de luz (LED) infravermelho, que está fixo no elemento terminal do robô. Três LEDs podem ser ligados e desligados em sequência, de forma que é possível localizar três pontos fixos do elemento terminal. Desta forma, a posição absoluta e a orientação de um sistema de coordenadas fixo no elemento terminal do robô pode ser conhecido. A resolução deste dispositivo é cerca de 1/4000 da escala completa. Este valor depende da localização relativa das duas câmeras, e sua exatidão é sensível às reflexões [59];
- Máquinas de Medição de Coordenadas (MMC) – consistem em um mecanismo de três

graus de liberdade, contendo os três eixos prismáticos ortogonais. São dispositivos fabricados para garantir movimento preciso ao longo dos eixos desejados, e estão instrumentados para determinar os deslocamentos das juntas para um alto grau de exatidão [3]. O uso de máquinas de medição de coordenadas para medições de contato garante uma exatidão alta, da ordem de 0,01 mm [60], [61], [62], [63]. Essas máquinas estão disponíveis em vários tamanhos e níveis de exatidão (esta pode estar na ordem de dez milésimos de polegada). Em geral, são projetadas para inspeção de peças e montagens, e, como a maioria das MMCs suficientemente grandes para a calibração de robôs são muito caras, torna-se inviável seu uso no processo de calibração [3]. De acordo com Jia-Qing et al. [58], as MMCs fornecem medições precisas para a posição de um corpo rígido, por meio da medição de três pontos não colineares no corpo. Entretanto, essas máquinas são volumosas e caras, sendo que inúmeros erros afetam sua exatidão, como: erros dos eixos da máquina e erros do sistema de sondagem. Além da medição por contato, a máquina de medição de coordenadas articuladas (MMCA), por sua vez, é um tipo de sistema de medição sem contato, cujo ponto medido resulta de uma série de transformações de coordenadas;

- Teodolitos – antigamente, a técnica mais comum de medição sem contato baseava-se no uso de um sistema de teodolitos [64], [65], [22], que, além de facilitar as medidas, também permite obtê-las com exatidão na ordem de 0,05 mm [66]. Ao se combinar os teodolitos com uma visão de baixa resolução para rastreamento automático, foco e centralização, é possível construir um sistema de alta qualidade para medições de calibração automática [35]. A Figura 2.11 ilustra o processo de medição com teodolitos.

Renders et al. [59], afirma que, apesar dos teodolitos fornecerem uma exatidão muito boa e permitirem medições ao longo da área referente ao espaço de trabalho do robô, trata-se de uma técnica de medição demorada e cansativa, embora existam teodolitos, cujo rastreamento é automático. Somado a isso, existe o fato de que esse equipamento mede somente ângulos, e não comprimentos, o que pode ocasionar certos problemas (necessidade de comprimento de referência em particular).

Uma técnica alternativa de medição sem contato foi aplicada por An et al. [67], e por Gateholm [68], que realizaram experimentos de calibração usando sistemas de câmeras equipados com detectores óptico-eletrônicos e diodos LED adicionais como alvos fixados no elemento terminal do robô; esta técnica produz resultados satisfatórios, porém em um espaço de trabalho menor. Existem também técnicas de calibração menos caras e menos precisas, que podem ser empregadas, tal como sondas de medição para medições precisas da posição e orientação do elemento terminal [69]. De forma mais extrema, Kim et al. [70], usou uma régua simples, ao invés de um sistema de medição 3D, para estimar os parâmetros cinemáticos.

Lau et al. [71], por outro lado, utilizaram a técnica de interferometria por *laser tracker* para realizar experimentos metrológicos em robôs. O sistema engloba um interferômetro a laser com um

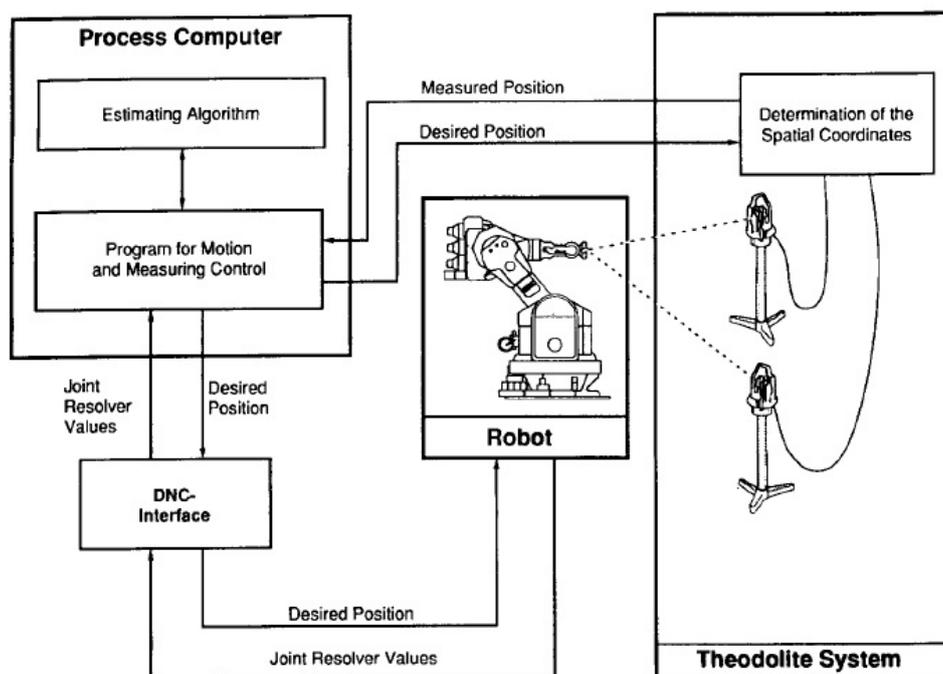


Figura 2.11: Processo de medição com teodolitos. Fonte: [22].

feixe orientável. O feixe rastreia o alvo montado no elemento terminal do robô, cuja localização encontra-se em um sistema de coordenadas esféricas. Chen e Chao [72], empregaram o método de triangulação com três teodolitos, com o intuito de calibrar o robô PUMA 760, através da localização do elemento terminal do robô por triangulação. Mooring e Padavala [61] usaram uma máquina de medição de coordenadas com um apalpador e um projeto do elemento terminal, especialmente concebido, visando calibrar um robô PUMA, por meio da medição da ferramenta em cada observação.

Everett e Ives [73] projetaram e implementaram um sensor para automatizar a calibração de robôs. O sensor opera em cooperação com esferas de precisão, que são montadas no espaço de trabalho, e cujas posições relativas são medidas previamente através de uma máquina de medição de coordenadas. O sensor está posicionado sobre essas esferas de forma automática e precisa, e seu posicionamento é determinado por três feixes de luzes. Isso elimina problemas experimentados por outros dispositivos de curto alcance, uma vez que atua de maneira digital e sem contato. Neste trabalho, os autores descrevem, além da construção do sensor, a calibração e os resultados de testes de desempenho.

Motta [2] discorre sobre resultados de calibração, usando um braço de medição por coordenadas (ITG ROMER), com 0,087 mm de exatidão, como um sistema de medição em um robô ABB IRB-2000. Além disso, apresenta e discute um sistema de medição 3-D baseado em visão computacional, desenvolvido especificamente para os requisitos de calibração de robôs, e que se mostra uma alternativa viável para sistemas de medição de alto custo de aquisição. O sistema de

medição é portátil, preciso e de baixo custo, sendo formado por uma única câmera CCD montada no flange da ferramenta do robô, visando medir a posição do elemento terminal do robô em relação a um sistema de coordenadas externo. A distorção radial das lentes foi incluída no modelo fotogramétrico, e fatores de escala e centros de imagens são obtidos com técnicas inovadoras, fazendo uso de uma abordagem *multiview*.

Wang et al. [74], propuseram um método de calibração para um manipulador robótico, com um sistema de medição baseado em uma câmera e um algoritmo de redes neurais. Os erros de posição em vários pontos dentro do espaço de calibração foram obtidos com o auxílio de uma câmera instalada no elemento terminal do robô a ser calibrado. Mais recentemente, Sun et al. [75], apresentaram uma abordagem de calibração cinemática de um manipulador paralelo de três graus de liberdade rotacionais, baseada em rastreamento a laser. Motta et al. [12], por sua vez, empregaram um braço de medição (ITG ROMER), com precisão informada pelo fabricante de 0,087 mm, para medir as posições, em vários locais dentro da área de trabalho, da ponta da tocha de solda montada na flange de um robô.

Zhang et al. [76], por sua vez, apresentaram um procedimento de auto calibração de robôs baseado em visão estéreo, utilizando somente uma câmera estéreo montada em um local fixo e um marcador plano conectado ao elemento terminal do robô. O procedimento de calibração proposto compreendia quatro etapas: seleção de configuração de destino, planejamento de trajetória, estimativa de posição baseada em visão computacional e identificação dos parâmetros cinemáticos. Desta forma, a calibração utiliza somente informações das imagens obtidas da câmera, e é conduzida automaticamente sem qualquer intervenção manual. Os resultados provenientes de simulações e experimentos com o robô de seis graus de liberdade UR5 provaram boa precisão e estabilidade no processo de calibração proposto.

Filion et al. [77] investigaram o desempenho de um sistema de fotogrametria portátil disponível no mercado, o MaxSHOT 3D da Creaform's, na calibração do robô FANUC LR Mate 200iC de seis graus de liberdade. Para demonstrar a eficiência desse dispositivo, os autores o compararam com um *laser tracker* fabricado pela FARO. Portanto, o robô foi calibrado com cada um dos dois equipamentos de medição, e, em seguida, a precisão da posição do robô foi comparada. Os resultados obtidos mostraram que o MaxSHOT 3D melhora a precisão da posição para 0,197 mm, enquanto o *laser tracker* melhora para 0,147 mm, o que permite aferir a semelhança nos desempenhos de calibração dos dois dispositivos de medição.

2.7 Implementação e Compensação de Posicionamento

A maneira mais evidente de implementar os resultados da calibração é substituir o modelo incorreto pelo calibrado. Entretanto, este último é mais complexo que o nominal, uma vez que algumas suposições feitas para robôs que possuem eixos paralelos e ortogonais, assim como pulso

esférico, não podem ser válidas quando pequenas variações são incluídas no modelo cinemático corrigido. Consequentemente, a solução da cinemática inversa para o modelo calibrado torna-se difícil de ser resolvida [3].

O problema da cinemática inversa *on-line* pode ser minimizado pela simplificação do método de calibração, por meio da adição de várias constantes aos atuadores de junta como vários parâmetros do modelo, possibilitando obter uma solução mais fácil. Em detrimento disso, limita-se a precisão do robô. Além disso, a implementação da cinemática inversa *on-line* baseia-se no fato de que existe uma pequena diferença entre a solução real e nominal, o que proporciona agilidade em determinar a solução usando algoritmos numéricos de busca. Todavia, se parâmetros não geométricos, como deformações elásticas (cálculo de torques e forças) são utilizados, não é possível simplificar o modelo, e a compensação *on-line*, nessa situação, demandaria muito esforço computacional por parte do controlador do robô [22].

Tendo em vista as desvantagens da cinemática inversa *on-line*, pode-se empregar uma solução *off-line*, desde que não sejam requeridas trajetórias de alta precisão para o robô. Os métodos de compensação *off-line* que utilizam os resultados de calibração permitem melhorar a precisão absoluta de posição, sem a necessidade de modificar os algoritmos e parâmetros de controle do robô, contanto que estes sejam conhecidos. Dessa forma, a partir do modelo cinemático do robô (corrigido pelo procedimento de calibração) e do modelo cinemático nominal, no qual os algoritmos de controle são baseados, as posições de compensação podem ser determinadas e carregadas no controlador do robô [25].

Duelen e Schröer [22] reforçam que, como os controladores dos robôs não podem fazer uso direto dos dados de calibração, os métodos de compensação de erros *off-line* devem ser aplicados para usar os resultados provenientes do processo de calibração e, assim, permitir a execução de programas gerados *off-line*. A Figura 2.12 mostra a técnica de compensação *off-line* proposta pelos autores.

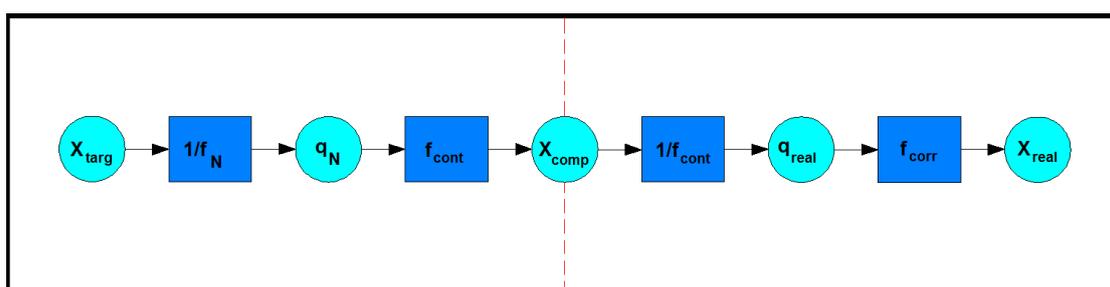


Figura 2.12: Compensação de erros *off-line*: X_{targ} é a posição do *target*; f_N é o modelo nominal do robô; q_N são as posições nominais das juntas; f_{cont} é o modelo do controlador; X_{comp} é a posição de compensação; q_{real} são as posições reais das juntas; f_{corr} é o modelo corrigido; X_{real} é a posição real. Fonte: Adaptado de [22].

Na Figura 2.12, posições nominais da junta são obtidas por meio do modelo nominal inverso e

das posições do *target*. Utilizando o modelo corrigido, novas posições do *target* são determinadas e, em seguida, as posições de compensação são definidas. Depois, baseado na inversa do modelo corrigido, bem como das posições de compensação, são calculadas novas posições de juntas que, quando inseridas no modelo do controlador do robô, o movimentam para a posição real desejada [22].

Existem várias técnicas de compensação de posicionamento disponíveis na literatura, dentre as quais se destaca o *Método de Redefinição de Posicionamento*, uma vez que não requer o conhecimento do modelo cinemático inverso corrigido [56].

De acordo com Vasconcelos Filho et al. [56], esse método emprega um modelo linear para prever os erros de posicionamento do robô e calcular a posição compensada P_c , para a qual o robô deve se mover de maneira a alcançar a posição desejada P_d , conforme demonstrado na Eq. 2.71.

$$P_c = P_d [Trans(-d_x, -d_y, -d_z) \cdot Rot(x, -\phi_x) \cdot Rot(y, -\phi_y) \cdot Rot(z, -\phi_z)] \quad (2.71)$$

Na Equação 2.71, $\begin{bmatrix} d_x & d_y & d_z & \phi_x & \phi_y & \phi_z \end{bmatrix}^T$ são os erros de posicionamento previstos pelo modelo calibrado [56].

2.8 Problemas de Calibração de Robôs

2.8.1 Calibração *hand-eye* (mão-olho)

A calibração *hand-eye* pode ser definida como o cálculo da posição relativa e da orientação entre a flange do robô e uma câmera (ou qualquer outro sensor) montado rigidamente na flange. De acordo com Dannilidis [50], essa calibração pode ser descrita por meio de matrizes de transformação homogênea. A Figura 2.13 ilustra uma aplicação do método de calibração *hand-eye*, onde \mathbf{X} é a transformação da câmera para a flange, \mathbf{A}_i é a matriz de transformação da câmera para o sistema de coordenadas externo, e \mathbf{B}_i é a matriz de transformação da base do robô para a garra na i -ésima posição. A matriz \mathbf{A}_i é obtida através de técnicas de calibração de câmeras, enquanto que a matriz de transformação \mathbf{B}_i é determinada a partir da cinemática direta do robô com as leituras dos ângulos das juntas.

Do ponto de vista matemático, a calibração *hand-eye* consiste em solucionar a variável \mathbf{X} , a partir da Eq. 2.72, que foi formulada pela primeira vez por Shiu e Ahmad [78] e Tsai e Lenz [79].

$$\mathbf{A} \cdot \mathbf{X} = \mathbf{X} \cdot \mathbf{B} \quad (2.72)$$

Onde $\mathbf{A} = \mathbf{A}_2 \cdot \mathbf{A}_1^{-1}$ e $\mathbf{B} = \mathbf{B}_2^{-1} \cdot \mathbf{B}_1$.

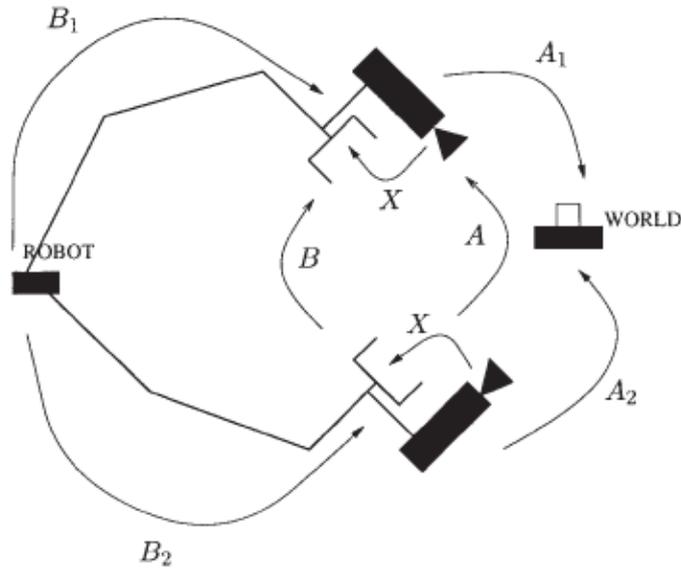


Figura 2.13: Transformações na calibração *hand-eye* entre diferentes sistemas de coordenadas para as posições 1 e 2 do robô. Fonte: [50].

Ao analisar a Fig. 2.72, percebe-se que, para uma posição, existem duas transformações desconhecidas: da base do robô para o sistema de coordenadas externo, e da câmera para a flange X. Para eliminar a primeira transformação, Shiu e Ahmad [78] mostraram que, pelo menos dois movimentos relativos com eixos rotacionais não-paralelos são necessários. Segundo Heller et al. [80], na prática, vários movimentos relativos são executados, produzindo um conjunto de matrizes $A_i, B_i, i = 1, \dots, n$, expresso pela Eq. 2.73, e, conseqüentemente, um sistema sobredeterminado. A menos que seja possível medir A_i e B_i com perfeita precisão, esse sistema, em geral, vem acompanhado de ruído.

$$A_i \cdot X = X \cdot B_i \quad (2.73)$$

Onde $i = 1, \dots, n$. A Equação 2.73 pode ser reescrita na forma da Eq. 2.74.

$$\begin{pmatrix} \mathbf{R}_{A_i} & \mathbf{t}_{A_i} \\ 0^T & 1 \end{pmatrix} \cdot \begin{pmatrix} \mathbf{R}_X & \mathbf{t}_X \\ 0^T & 1 \end{pmatrix} = \begin{pmatrix} \mathbf{R}_X & \mathbf{t}_X \\ 0^T & 1 \end{pmatrix} \cdot \begin{pmatrix} \mathbf{R}_{B_i} & \mathbf{t}_{B_i} \\ 0^T & 1 \end{pmatrix} \quad (2.74)$$

A Equação 2.74, por sua vez, pode ser decomposta em uma matriz de rotação, expressa pela Eq. 2.75, e em um vetor de translação, evidenciado pela Eq. 2.76.

$$\mathbf{R}_{A_i} \cdot \mathbf{R}_X = \mathbf{R}_X \cdot \mathbf{R}_{B_i} \quad (2.75)$$

$$\mathbf{R}_{A_i} \cdot \mathbf{t}_X + \mathbf{t}_{A_i} = \mathbf{R}_X \cdot \mathbf{t}_{B_i} + \mathbf{t}_X \quad (2.76)$$

Daniilidis [50], aponta que a calibração *hand-eye* é necessária em várias tarefas de ação sensorial. Para o caso de uma câmera montada em uma garra ou um veículo, por exemplo, pode-se estimar a posição de um alvo para determinar as coordenadas da câmera. Entretanto, os comandos de controle podem ser expressos somente no sistema de coordenadas da flange ou do veículo, e, mesmo que o critério de controle desejado seja dado em coordenadas da câmera, é indispensável saber o efeito de um movimento do robô no sistema de coordenadas da câmera.

Outra aplicação onde se faz necessário o uso da calibração *hand-eye* é na reconstrução estéreo, na qual se coloca uma câmera montada na flange de um robô em várias posições, compartilhando o mesmo campo de visão. Todavia, a reconstrução 3-D implica no conhecimento da orientação relativa do sistema de coordenadas da câmera, porém, as únicas transformações conhecidas são aquelas das coordenadas do robô.

O método de calibração *hand-eye*, conforme afirma Wu et al. [81], tem sido um problema clássico, ao utilizar um sensor (uma câmera, por exemplo) montado no elemento terminal do robô para realizar várias tarefas. Esse problema tem sido amplamente estudado por vários pesquisadores, com destaque para Daniilidis [50], que propôs a calibração *hand-eye* usando quaternions duplos. Para tanto, o autor introduziu uma linguagem para formalizar a abordagem de *screw* na calibração *hand-eye*. Em seu trabalho, o autor desenvolveu o primeiro algoritmo de solução simultânea, para a rotação e a translação, sem minimização não-linear. Este programa foi implementado e comparado com outros dois algoritmos, na qual o primeiro consiste em uma solução simultânea para a rotação e a translação, porém, envolvendo uma minimização não-linear; enquanto que o segundo resolve de forma linear, primeiro a rotação, e, depois, a translação. Os três métodos foram comparados em simulações, com variações no nível de ruído para as posições da câmera e da junta. O método de quaternions duplos se mostrou superior, não só pela solução simultânea dos movimentos, mas também pelo fato de usar somente informações necessárias para o problema de calibração *hand-eye*. Baseado em quaternions duplos, foi possível provar que os parâmetros de *screw* desconhecidos, incluindo a rotação e a translação, podem ser simultaneamente recuperados, usando a técnica algébrica de decomposição em valores singulares (SVD).

Dornaika e Horaud [49], em uma breve publicação, apresentaram duas novas soluções para aquela época: a solução de forma fechada e a otimização não-linear, a fim de tentar resolver a equação de matrizes homogêneas, representada pela Eq. 2.77.

$$\mathbf{A} \cdot \mathbf{X} = \mathbf{Z} \cdot \mathbf{B} \quad (2.77)$$

A Equação 2.77 foi proposta por Zhuang et al. [48] para estimar, simultaneamente, as transformações rígidas do sistema de coordenadas externo para o sistema de coordenadas da base do robô, e do sistema de coordenadas do elemento terminal para o sistema de coordenadas da câmera. Nesse

método, Zhuang et al. [48] usa quaternions para derivar soluções lineares para X e Z , onde X é a transformação da câmera para a flange e Z é a transformação rígida da base do robô para o sistema de coordenadas externo, conforme se observa na Fig. 2.14.

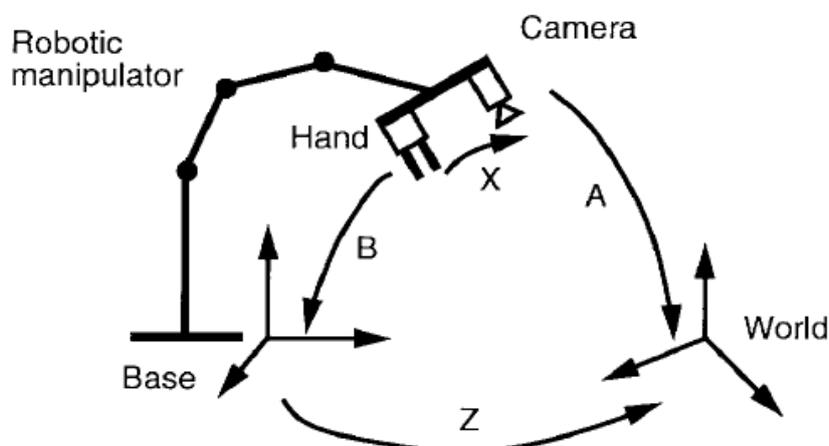


Figura 2.14: Método de calibração simultâneo proposto por Zhuang et al. [48]. Fonte: [49].

Como já mencionado, Dornaika e Horaud [49] descreveram uma solução fechada e uma solução não-linear para o sistema de equações matriciais da Eq. 2.77. As principais diferenças entre o método linear de Zhuang et al. [48] e o método de solução fechada de Dornaika e Horaud [49], é que o primeiro soluciona linearmente as componentes de dois quaternions e, o segundo, normaliza estes quaternions, de forma que eles representem rotações. O método fechado, entretanto, resolve diretamente para dois quaternions unitários e, portanto, a restrição que esses quaternions devem representar duas rotações é construída no método de resolução. O método linear não se aplica para determinadas configurações especiais, ao contrário da solução fechada.

Dessa forma, Dornaika e Horaud [49], realizaram uma análise de sensibilidade para a solução linear sugerida por Zhuang et al. [48] e para as duas soluções propostas por eles (fechada e otimização não-linear). Em seu trabalho, eles constataram, por meio de simulações e experimentos reais, que o método de otimização não-linear (que resolve rotações e translações simultaneamente), tem desempenho melhor do que o método fechado, que, por sua vez, funciona melhor que o método linear. Porém, é importante destacar que não existem evidências de que, com os dados reais, o método fechado sempre funcionará melhor que o método linear. Além disso, a solução obtida com o método linear ou fechado pode ser usada para inicializar o método de minimização não-linear.

Recentemente, Heller et al. [80], propuseram um conjunto de métodos iterativos para resolver a calibração *hand-eye* e *robot-world*, com base nas Eq. 2.73 e 2.77, que não requerem estimativas iniciais e fornecem soluções globais ótimas. Esses métodos solucionam a parte rotacional e translacional simultaneamente, através da formulação dos problemas de calibração *hand-eye* e *robot-*

world como problemas de otimização polinomial multivariável em conjuntos semi-algébricos, e resolvendo-os pelo método das relaxações da matriz linear convexa.

Tanto para a calibração *hand-eye*, quanto para a calibração simultânea *hand-eye* e *robot-world*, Heller et al. [80], apresentaram três formulações: a primeira baseada na parametrização ortonormal, a segunda com base na parametrização com quaternion e, a terceira, baseada na parametrização com quaternion duplo.

Malti e Barreto [51], sugeriram um método de calibração *hand-eye* robusto para a endoscopia médica assistida por computador, sendo motivados pelo fato de que a câmera endoscópica para navegação cirúrgica e visualização 3-D requer estimativas precisas e estáveis da calibração dos parâmetros.

De acordo com Malti e Barreto [51], na literatura existem vários métodos para resolver a Eq. 2.72, que se subdividem em três grupos:

- Estimativa separada: a Eq. 2.72 é decomposta em uma parte rotacional e uma parte translacional. A parte referente à rotação é solucionada primeiro, seguida pela solução da parte translacional;
- Estimativa simultânea: consiste em resolver simultaneamente a rotação e a translação, representando o problema por meio de *screw*;
- Estimativa que inclui os parâmetros intrínsecos da câmera: a transformação *hand-eye* e os parâmetros intrínsecos da câmera são simultaneamente estimados em um passo de otimização iterativa global. Em seu trabalho, os autores assumiram que os parâmetros intrínsecos da câmera foram previamente estimados, e, por isso, não consideraram este método para análise comparativa.

Malti e Barreto [51] acrescentam que a calibração *hand-eye* apresenta características peculiares quando aplicada no procedimento de endoscopia. A primeira característica está relacionada ao fato dos valores das componentes de translação dos movimentos de **A** e **B** serem pequenos (abaixo de 10 cm); a segunda, considera que a calibração deve ser realizada por uma pessoa que não é especialista na sala de operação, o que exige que o método seja robusto com um número mínimo de movimentos. Considerando as condições acima, os autores observaram, por meio de experimentos, que as estimativas separada e simultânea não produzem resultados satisfatórios para esse tipo de aplicação. O primeiro método fornece boas estimativas para a rotação, porém, os resultados para a translação são imprecisos para pequenos movimentos. No segundo método, a rotação permite restringir a estimativa da translação, o que produz melhores resultados. Entretanto, como a rotação e a translação são calculadas simultaneamente, esta última tem um impacto negativo na primeira, e os resultados para a rotação são mais imprecisos que aqueles obtidos para o primeiro método.

Diante disso, o trabalho de Malti e Barreto [51] tem como maior contribuição o algoritmo de calibração *hand-eye* robusto para pequenos deslocamentos e número mínimo de movimentos. O método proposto pelos autores supera as limitações dos métodos anteriores por apresentar as vantagens destes, uma vez que usa quaternion duplo na formulação do problema (pois oferece uma representação estável da translação como um deslocamento ao longo do eixo de rotação), e estima separadamente a rotação e a translação (o ruído na translação afeta gravemente a rotação em uma estimativa conjunta). A translação é estimada de forma robusta pela representação de quaternion por *screw*. Além disso, as restrições envolvidas são levadas em consideração durante o processo de estimativa. Os resultados das simulações realizadas mostraram a precisão e a estabilidade da abordagem proposta pelos autores, quando efetuada com uma pequena amplitude e número de movimentos. Do ponto de vista de uma calibração endoscópica, os resultados experimentais reais demonstraram a alta precisão e repetibilidade do algoritmo proposto pelos autores.

2.8.2 Calibração *eye-to-hand* (olho-a-mão)

Segundo Pan et al. [82], na configuração *eye-to-hand*, a câmera é fixada em uma posição distante do robô, de maneira que tenha uma visão global do espaço de trabalho, e a calibração consiste em determinar a transformação entre o sistema de coordenadas da câmera e o sistema de coordenadas da base do robô. Na Figura 2.15 é mostrado uma descrição detalhada desta configuração.

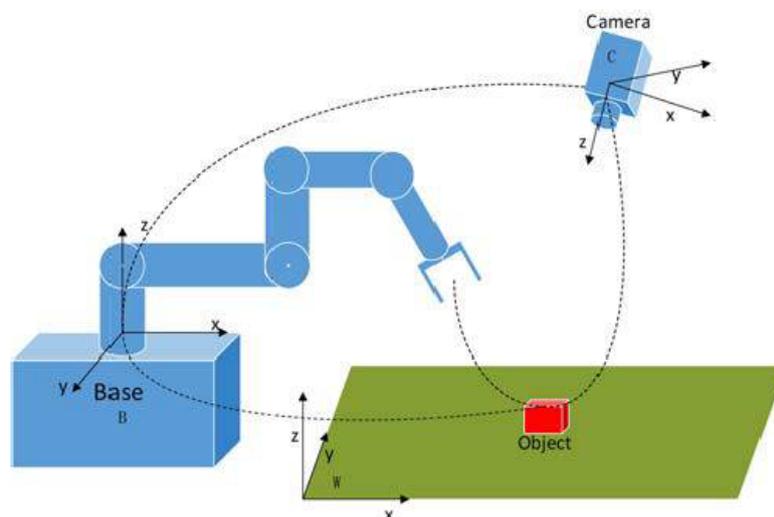


Figura 2.15: Configuração *eye-to-hand* na captura do objeto com auxílio de câmera. Fonte: [82].

Pan et al. [82] estudaram o problema da calibração *eye-to-hand* na captura de objetos com o auxílio de câmera. Para tanto, eles separaram a Eq. 2.72 em uma parte rotacional e uma parte translacional. A matriz de rotação é representada pelo método do quaternion unitário, e a

solução fechada é encontrada na sequência, de forma que o vetor de translação seja facilmente encontrado. A fim de melhorar a precisão e o desempenho da calibração, neste trabalho, levaram-se em consideração os impactos gerados pelos ruídos, e implementaram-se etapas de retificação. Simulações e experimentos reais de calibração *eye-to-hand* foram realizados com o robô Reinovo, e, com os resultados obtidos, foi possível atestar o desempenho predominante do método de calibração proposto pelos autores sobre outros métodos.

Hu e Chang [83] desenvolveram um algoritmo de calibração, capaz de estimar com precisão e, simultaneamente, os parâmetros intrínsecos da câmera, o posicionamento entre a câmera e o robô e a posição do plano de trabalho. Eles aplicaram essa técnica com o auxílio de um objeto plano e de um ponteiro a laser montado no elemento terminal do manipulador, conforme ilustrado na Fig. 2.16.

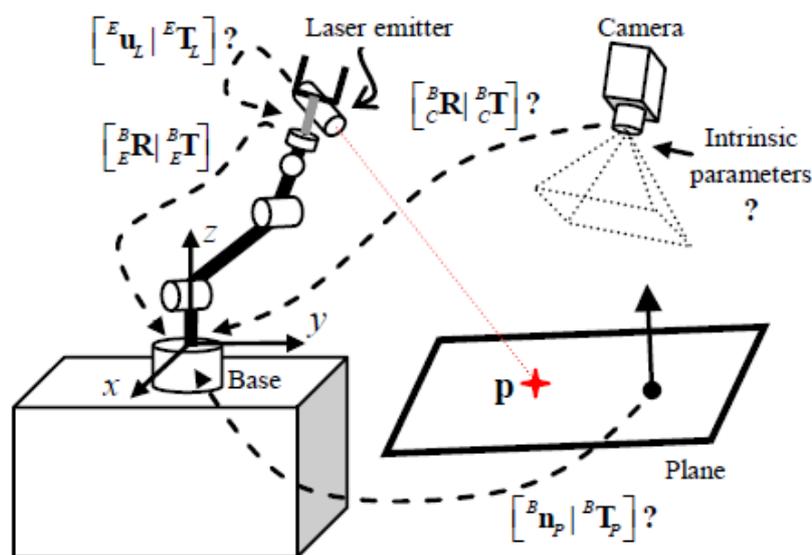


Figura 2.16: Configuração geral de um sistema *eye-to-hand* com um ponteiro a laser montado no elemento terminal do robô e um objeto plano colocado na frente da câmera. Fonte: [83].

Quando o robô é manipulado e o raio de laser projetado em um plano de orientações desconhecidas, uma quantidade de posições da imagem, relacionadas aos pontos de luz, são extraídas das imagens da câmera. Como o laser é montado rigidamente, e o objeto plano é fixo, as medições que a câmera adquire a partir do feixe de laser projetado no objeto plano devem obedecer às restrições geométricas. Baseada nessas restrições, a solução fechada é desenvolvida mediante o desacoplamento das equações não-lineares para calcular todos os valores iniciais. Como consequência, o método de calibração proposto não precisa de nenhuma estimativa inicial dos parâmetros desconhecidos. Hu e Chang [83] também implementaram um método de otimização não-linear para refinar a estimativa e melhorar a precisão dos valores provenientes da solução fechada. A vantagem de usar o ponteiro a laser, é que esta técnica pode ser aplicada para situações em que a câmera não consegue visualizar o manipulador robótico. Neste trabalho, foram realizados simulações e

experimentos reais, visando validar o algoritmo proposto.

Hu e Chang [84], em uma publicação mais recente, apresentaram um método para calibrar simultaneamente as relações *eye-to-hand* e *workspace-to-robot*, usando um laser com formato de linha montado no manipulador e que é projetado no plano de trabalho, ao invés de empregar um laser de ponto único. Esse método também leva em consideração os parâmetros intrínsecos da câmera. Novamente, pela forma como o sistema está organizado (com o laser rígido e com o plano fixo em cada orientação), os parâmetros geométricos e as medições satisfazem as restrições não-lineares e as soluções dos parâmetros podem ser estimadas. Com base nessas restrições, a solução fechada proposta pelos autores produz uma solução global, estimando os valores iniciais dos parâmetros intrínsecos da câmera e das transformações entre posições, que são refinados por meio de otimização não-linear. Neste trabalho foram realizadas simulações para validar as etapas do método de calibração e analisar seu desempenho em várias condições. Os resultados experimentais mostraram que o método proposto tem excelente eficiência quando a câmera não consegue visualizar o manipulador. Além disso, os resultados provenientes das simulações e dos experimentos foram confrontados com os resultados obtidos usando um laser de ponto único, o que evidenciou a maior precisão ao empregar-se um laser de linha, uma vez que este fornece mais informações geométricas com relação a cada posição do manipulador.

Carlson et al. [85] propuseram um método *eye-to-hand* robusto e linear para a calibração cinemática de um sensor de *scanner a laser* montado no elemento terminal do manipulador industrial IRB-140 da ABB. Esse sensor é usado em aplicações de digitalização robótica 3-D e rastreamento de costura de solda. O método baseia-se unicamente na resolução de conjuntos lineares de equações, aceitando uma estimativa inicial grosseira dos parâmetros cinemáticos, que, por sua vez, são refinados por um processo iterativo. As colocações dos planos de calibração são assumidas desconhecidas e suas localizações são encontradas junto com a matriz de transformação desejada. As incertezas inerentes às estimativas iniciais são tratadas, e o erro de estimativa converge para abaixo do nível do ruído de medição.

2.8.3 Calibração envolvendo vários robôs e sensor

Esse problema é caracterizado por envolver mais de um robô e pela necessidade de calibrar a posição entre cada um deles [86]. A equação matricial a ser solucionada é formulada pela Eq. 2.78.

$$A \cdot X \cdot B = Y \cdot C \cdot Z \quad (2.78)$$

Na Equação 2.78, as transformações representam diferentes relações, conforme o sistema robótico a ser calibrado. A Figura 2.17 ilustra um desses sistemas, que mostra três robô móveis, cada um montado com uma câmera na flange que olha para um *target* de outro robô.

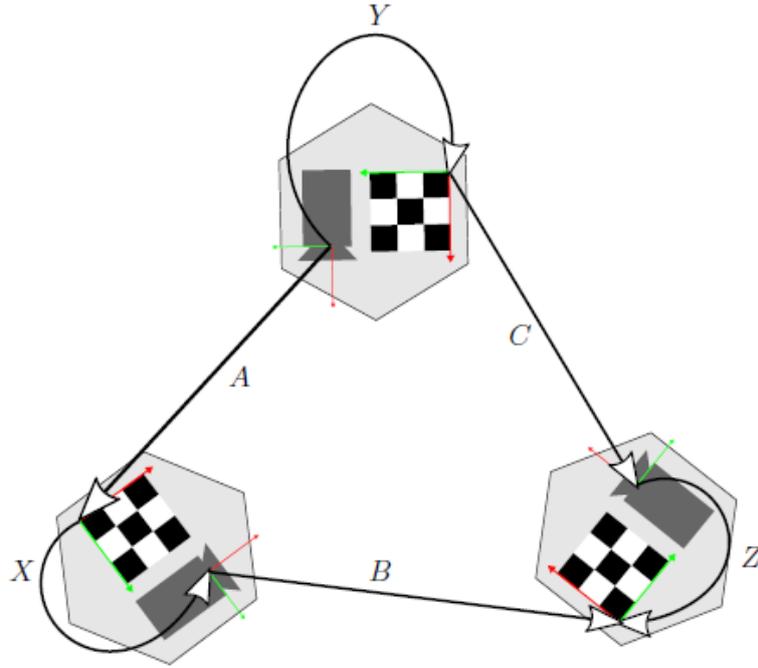


Figura 2.17: Vista superior de três robôs móveis com as calibrações desconhecidas formuladas como $AXB = YCZ$. Fonte: [86].

Na Figura 2.17, A , B e C são transformações da câmera para o *target* entre robôs diferentes, enquanto X , Y e Z são as transformações da câmera para o *target* dentro do mesmo robô.

As incógnitas presentes na Eq. 2.78 são determinadas a partir da solução separada das componentes de rotação e translação [81], [86]. Portanto, expandindo as matrizes de transformação, obtém-se as Eq. 2.79 e 2.80.

$$\mathbf{R}_A \cdot \mathbf{R}_X \cdot \mathbf{R}_B = \mathbf{R}_Y \cdot \mathbf{R}_C \cdot \mathbf{R}_Z \quad (2.79)$$

$$\mathbf{R}_A \cdot \mathbf{R}_X \cdot \mathbf{t}_B + \mathbf{R}_A \cdot \mathbf{t}_X + \mathbf{t}_A = \mathbf{R}_Y \cdot \mathbf{R}_C \cdot \mathbf{t}_Z + \mathbf{R}_Y \cdot \mathbf{t}_C + \mathbf{t}_Y \quad (2.80)$$

Onde \mathbf{R} é uma matriz de rotação de ordem 3×3 , e \mathbf{t} é um vetor de translação de ordem 3×1 . Primeiramente são solucionadas as rotações \mathbf{R}_X , \mathbf{R}_Y e \mathbf{R}_Z presentes na Eq. 2.79; depois esses valores são usados como dados de entrada na Eq. 2.80, a fim de determinar as translações \mathbf{t}_X , \mathbf{t}_Y e \mathbf{t}_Z .

Goh [86] acrescenta que, pode-se resolver para X e Z simultaneamente, e usar o resultado para obter Y .

Wu et al. [81] propuseram um novo método para solucionar o problema de calibração *hand-eye*, *tool-flange* e *robot-robot*, aplicado a tarefas onde robôs devem executá-las simultaneamente.

O método formula o problema na forma da Eq. 2.78. Quando comparado com os métodos de solução expressos pelas Eq. 2.72 e 2.77, os autores verificaram que o método proposto reduz tanto a dificuldade na aquisição de dados, quanto a imprecisão provocada pela propagação e acúmulo de erros. Esse trabalho apresenta como principais contribuições: o uso de uma solução analítica, que fornece uma estimativa inicial para a solução a ser obtida iterativamente; uma interpretação matemática completa, além de incluir a análise de solvabilidade e provas relacionadas; experimentos realizados em robôs reais, demonstrando a eficácia do método proposto. A Figura 2.18 apresenta as transformações presentes no método proposto por Wu et al. [81].

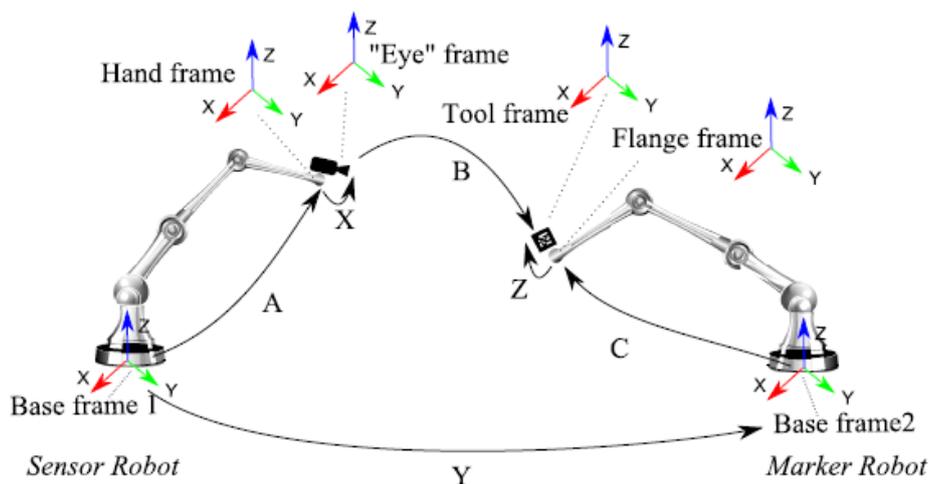


Figura 2.18: Transformações envolvidas no problema de calibração simultânea *hand-eye*, *tool-flange* e *robot-robot*. Fonte: Adaptado de [81].

Yan et al. [87] propuseram um sistema robótico híbrido, cuja calibração é modelada por meio da Eq. 2.78, e está representado na Fig. 2.19.

Analisando a Figura 2.19 é possível verificar que o robô híbrido consiste em um manipulador paralelo fixado na flange de um robô serial, e que existe uma câmera estacionária visualizando um *target* na ferramenta do robô paralelo. Neste sistema, as incógnitas correspondem às transformações entre a base do robô paralelo e a flange do robô serial, X , a transformação entre o sistema de coordenadas da câmera e a base do robô serial, Y , e a transformação entre o *target* e a flange do robô paralelo, Z . As transformações A e B são obtidas da cinemática direta dos robôs, enquanto que a transformação C é fornecida pela calibração da câmera a partir de imagens capturas do *target*.

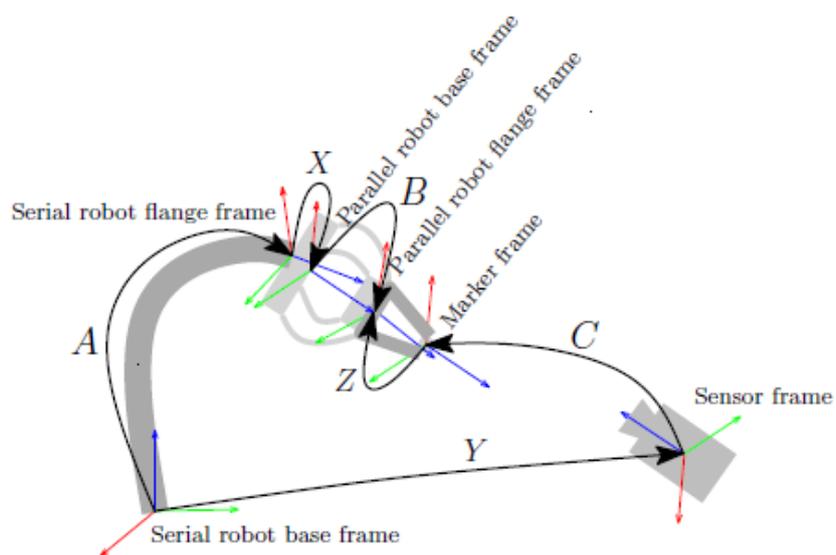


Figura 2.19: Calibração de um sistema robótico híbrido (robô paralelo acoplado a um robô serial) formulado como $AXB = YCZ$. Fonte: [87].

Capítulo 3

Metodologia

3.1 Aspectos Metodológicos

Segundo Gil [88], o tipo de pesquisa científica pode ser classificada a partir de dois pontos de vista: quanto aos objetivos e quanto aos procedimentos técnicos. Em relação aos objetivos, a pesquisa abordada nesta dissertação de mestrado se classifica como uma pesquisa exploratória, uma vez que objetiva desenvolver um método de calibração de robôs que inclui uma etapa de otimização paramétrica do modelo de erros; e, quanto aos procedimentos técnicos, a mesma se enquadra como uma pesquisa bibliográfica, pois compreende uma fundamentação teórica constituída por conceitos (relacionados aos principais tópicos envolvidos no trabalho) obtidos de livros e artigos científicos.

3.2 Modelagem Cinemática do IRB-140

3.2.1 O manipulador industrial IRB-140

O método de calibração proposto neste trabalho foi implementado em um manipulador robótico IRB-140. Esse robô é compacto, possui seis eixos, com uma carga útil de 6 kg, e é fabricado pela Asea Brown-Boveri (ABB) para indústrias de manufatura que utilizam automação flexível baseada em robôs. O IRB-140 pode ser montado no chão, invertido ou na parede em qualquer ângulo, sendo adequado para várias aplicações, dentre as quais se destacam: soldagem a arco, montagem, limpeza ou pulverização e manuseio de materiais. Além disso, pode ser operado manualmente, por meio do *teach-pendant*, ou remotamente, utilizando a linguagem de programação de alto nível denominada de RAPID [89]. Na Tabela 3.1 é possível verificar algumas especificações técnicas desse robô, enquanto que a Fig. 3.1 ilustra seu espaço de trabalho.

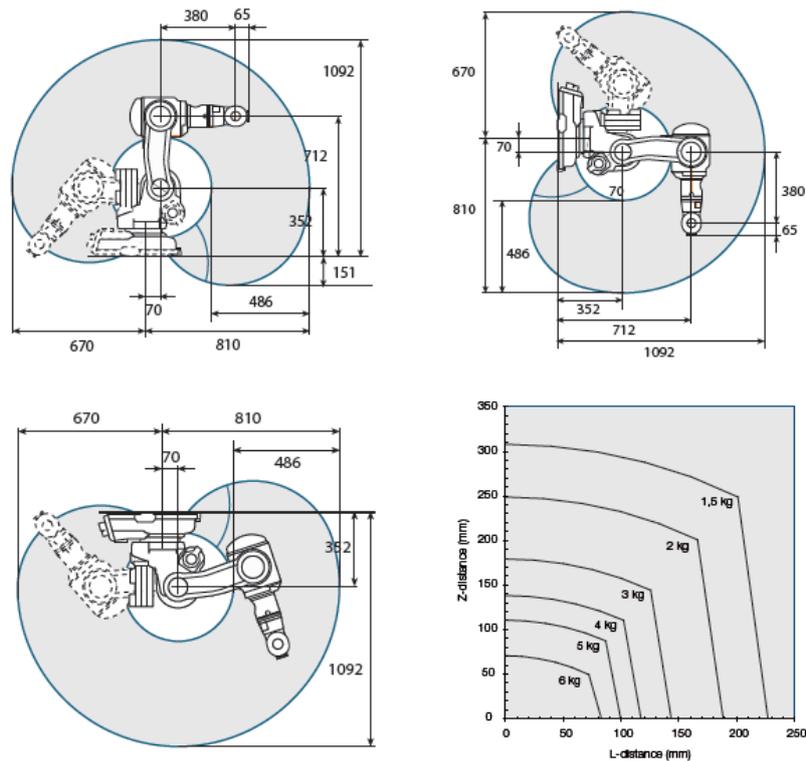


Figura 3.1: Espaço de trabalho do IRB-140. Fonte: [89].

Tabela 3.1: Especificações técnicas do robô IRB-140.

Tipo de movimento	Range de movimento
Eixo 1: movimento de rotação	+ 180° a - 180°
Eixo 2: movimento do braço	+ 110° a - 90°
Eixo 3: movimento do braço	+ 50° a - 230°
Eixo 4: movimento do pulso	+ 200° a - 200°
Eixo 5: movimento de torção	+ 115° a - 115°
Eixo 6: movimento de rotação	+ 400° a -400°
Repetibilidade de posição	0,03 mm
Resolução	aproximadamente 0,01 ° em cada eixo

Fonte: Produzido pela Autora.

3.2.2 Atribuição dos sistemas de coordenadas

A primeira etapa da modelagem cinemática implica na atribuição de sistemas de coordenadas ortogonais para cada elo do robô, cujos eixos obedecem a regra da mão direita [15]. Uma maneira de conferir esses sistemas consiste em mover o manipulador para a posição zero, onde todas as variáveis de juntas são zero. Esse procedimento permite verificar se as posições do modelo são as mesmas empregadas pelo controlador, evitando introduzir erros constantes para as variáveis de juntas (juntas de posição).

Dessa forma, para a construção do modelo cinemático direto do IRB-140, os sistemas de coordenadas foram atribuídos de maneira a fazer o eixo z coincidir com o eixo da junta. Com isso, se a junta for prismática, a direção do eixo z corresponde à direção do movimento, e seu sentido está ao longo da junta; se a junta for de revolução, entretanto, o sentido do eixo z está na direção positiva de rotação em torno desse eixo. O sentido positivo de rotação de cada junta pode ser facilmente encontrado pela movimentação do robô e leitura das posições das juntas no visor do controlador.

O sistema de coordenadas da base b , por outro lado, foi atribuído com eixos paralelos ao sistema de coordenadas universal w . A origem do sistema de coordenadas da base coincide com a origem da primeira junta, o que pressupõe que o eixo da primeira junta é normal ao plano $x-y$ [1].

As origens dos sistemas de coordenadas das juntas foram colocadas obedecendo aos seguintes critérios: se os eixos das juntas de um elo se cruzam, então a origem do sistema de referência fixado ao elo é inserido na intersecção dos eixos de juntas; se os eixos de juntas de um elo são paralelos ou não se cruzam, então a origem do sistema de coordenadas fixado ao elo é colocado na junta distal. Com isso, o sistema de coordenadas i é estabelecido na junta $i+1$, isto é, a junta que conecta o elo i ao elo $i+1$; e, se a origem de um sistema de coordenadas é descrito em relação a outro sistema, englobando mais de uma direção, então ela deve ser movida para ter sua posição relativa descrita por apenas uma direção, se possível. Baseado nisso, é possível garantir que as origens dos sistemas de coordenadas sejam descritas usando um número mínimo de parâmetros de elos [12].

Os eixos x ou y têm sua direção de acordo com a convenção utilizada para parametrizar as transformações entre os elos e o outro eixo, x ou y , pode ser determinado por meio da regra da mão direita.

Um sistema de coordenadas foi fixado ao último elo, especificamente na ferramenta, de maneira que o eixo z tenha a mesma direção que o eixo z do sistema de coordenadas da última junta ($n-1$). A localização e a orientação do efetuador ou do sistema de coordenadas da ferramenta são definidas conforme as convenções do controlador.

Da convenção apresentada, parâmetros geométricos de comprimento são estabelecidos: p_{ni}

(distância entre os sistemas de coordenadas $i-1$ e i), e n é o eixo paralelo localizado no sistema de coordenadas $i-1$. Aplicando as regras dispostas acima para o manipulador IRB-140, obtém-se os sistemas de coordenadas e as respectivas características geométricas para cada junta, conforme indicado na Fig.3.2.

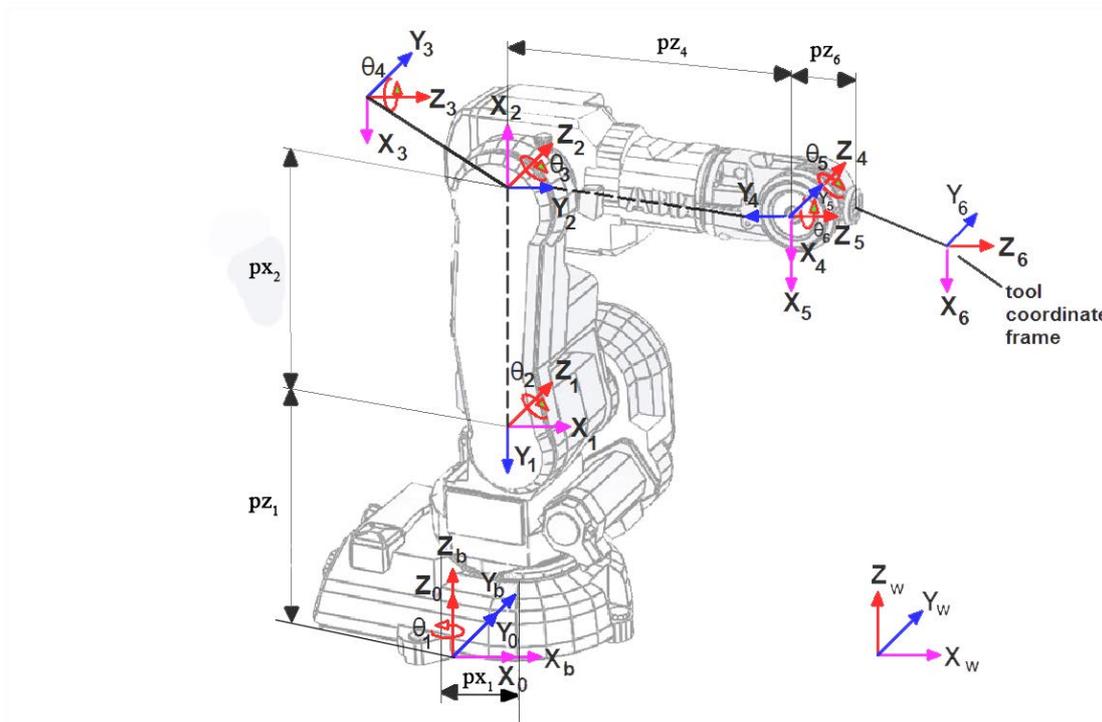


Figura 3.2: Sistemas de coordenadas e parâmetros geométricos do manipulador IRB-140. Fonte: Produzido pela Autora.

Conforme se observa na Fig. 3.2, o IRB-140 é formado por juntas rotativas, sendo que somente dois eixos internos da sua cadeia cinemática são paralelos.

3.2.3 Transformações homogêneas para o IRB-140

As transformações homogêneas para o modelo cinemático foram determinadas de maneira a assegurar que o modelo seja completo, contínuo e mínimo, o que significa que a convenção de Denavit e Hartenberg (DH) foi usada para juntas rotativas perpendiculares, enquanto que a convenção de Hayati e Mirmirani (HM) foi empregada para juntas rotativas paralelas, pois juntas sucessivas paralelas violam os requisitos do modelo ao usar a parametrização de DH [11]. Levando isso em consideração, assim como os sistemas de coordenadas presentes na Fig. 3.2 e as dimensões nominais do manipulador (disponíveis em suas especificações técnicas [89]), foram obtidos e organizados na Tab. 3.2 os parâmetros de modelagem para o IRB-140.

Uma vez que o sistema de coordenadas da ferramenta tem a mesma orientação que o sistema

Tabela 3.2: Parâmetros de modelagem do manipulador IRB-140.

Elo	Convenção	$T_x[mm]$	$T_y[mm]$	$T_z[mm]$	$R_x[^\circ]$	$R_y[^\circ]$	$R_z[^\circ]$	Faixa de Operação
0	–	0	0	0	0	0	0	0° a 0°
1	DH	70	–	352	-90	–	0	- 180° a + 180°
2	DH	360	–	0	0	–	-90	-90° a +110 °
3	HM	0	–	–	-90	0	0	-230° a +50°
4	DH	0	–	380	90	–	0	- 200° a + 200°
5	DH	0	–	0	-90	–	0	- 115° a + 115°
6	–	0	0	65 + 120	–	–	0	- 400° a + 400°

Fonte: Produzido pela Autora.

de coordenadas do último elo, porém está transladado ao longo do eixo z, essa distância foi mensurada, sendo igual a 120 mm, e adicionada ao elo 6, como se pode constatar na Tab. 3.2.

Dessa forma, as transformações elementares para o IRB-140 podem ser agrupadas em um conjunto de parametrizações, representado na Tab. 3.3.

Tabela 3.3: Transformações elementares para o IRB-140.

Transformação	Tipo de Junta	Parametrização
Universal para Base	w-b	$[T_x(px_b).T_y(py_b).T_z(pz_b).R_z(\theta_b).R_y(\beta_b).R_x(\alpha_b)]$
Junta 0 para Junta 1	$J_R \perp J_R$	$[R_z(\theta_1).T_z(pz_1).T_x(px_1).R_x(\alpha_1)]$
Junta 1 para Junta 2	$J_R \perp J_R$	$[R_z(\theta_2).T_z(pz_2).T_x(px_2).R_x(\alpha_2)]$
Junta 2 para Junta3	$J_R \parallel J_R$	$[R_z(\theta_3).T_x(px_3).R_x(\alpha_3).R_y(\beta_3)]$
Junta 3 para Junta 4	$J_R \perp J_R$	$[R_z(\theta_4).T_z(pz_4).T_x(px_4).R_x(\alpha_4)]$
Junta 4 para Junta 5	$J_R \perp J_R$	$[R_z(\theta_5).T_z(pz_5).T_x(px_5).R_x(\alpha_5)]$
Junta 5 para Junta 6	$J_R \parallel TCP$	$[R_z(\theta_6).T_x(px_6).T_y(py_6).T_z(pz_6)]$

Fonte: Produzido pela Autora.

Na Tabela 3.3, w é o sistema de referência universal, b é o sistema de referência da base, R_x , R_y e R_z são matrizes de rotação em torno, respectivamente, dos eixos x , y e x . Os termos T_x , T_y e T_z são matrizes de translação ao longo dos eixos x , y e x , respectivamente. O símbolo \parallel indica que as juntas são paralelas, enquanto que o símbolo \perp expressa o perpendicularismo entre as juntas. A expressão p_{xi} indica uma translação no eixo x , da origem da junta i para a junta $i + 1$; p_{yi} representa uma translação ao longo do eixo y , da origem da junta i para a junta $i + 1$; p_{zi} refere-se a uma translação no eixo z , da origem da junta i para a junta $i + 1$; θ_i representa uma rotação da junta em torno do eixo z ; β_i corresponde a uma rotação em torno do eixo y ; e α_i refere-se a uma torção geométrica em torno do eixo x da junta $i+1$ relacionada à junta i .

As matrizes de transformações homogêneas do IRB-140 são obtidas com base nos dados de modelagem presentes na Tab. 3.3, e estão definidas nas Eq. 3.1, 3.2, 3.3, 3.4, 3.5, 3.6 e 3.7.

$$\mathbf{T}_b^w = \begin{bmatrix} C(\beta_b)C(\theta_b) & S(\alpha_b)S(\beta_b)C(\theta_b) - C(\alpha_b)S(\theta_b) & S(\alpha_b)S(\theta_b) + C(\alpha_b)S(\beta_b)C(\theta_b) & px_b \\ C(\beta_b)S(\theta_b) & C(\alpha_b)C(\theta_b) + S(\alpha_b)S(\beta_b)S(\theta_b) & C(\alpha_b)S(\beta_b)S(\theta_b) - S(\alpha_b)C(\theta_b) & py_b \\ -S(\beta_b) & C(\beta_b)S(\alpha_b) & C(\alpha_b)C(\beta_b) & pz_b \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.1)$$

$$\mathbf{T}_1^0 = \begin{bmatrix} C(\theta_1) & -C(\alpha_1)S(\theta_1) & S(\alpha_1)S(\theta_1) & C(\theta_1)px_1 \\ S(\theta_1) & C(\alpha_1)C(\theta_1) & -C(\theta_1)S(\alpha_1) & S(\theta_1)py_1 \\ 0 & S(\alpha_1) & C(\alpha_1) & pz_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.2)$$

$$\mathbf{T}_2^1 = \begin{bmatrix} C(\theta_2) & -C(\alpha_2)S(\theta_2) & S(\alpha_2)S(\theta_2) & C(\theta_2)px_2 \\ S(\theta_2) & C(\alpha_2)C(\theta_2) & -C(\theta_2)S(\alpha_2) & S(\theta_2)py_2 \\ 0 & S(\alpha_2) & C(\alpha_2) & pz_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.3)$$

$$\mathbf{T}_3^2 = \begin{bmatrix} C(\beta_3)C(\theta_3) - S(\alpha_3)S(\beta_3)S(\theta_3) & -C(\alpha_3)S(\theta_3) & S(\beta_3)C(\theta_3) + C(\beta_3)S(\alpha_3)S(\theta_3) & C(\theta_3)px_3 \\ C(\beta_3)S(\theta_3) + S(\alpha_3)S(\beta_3)C(\theta_3) & C(\alpha_3)C(\theta_3) & S(\beta_3)S(\theta_3) - C(\beta_3)S(\alpha_3)C(\theta_3) & S(\theta_3)py_3 \\ -C(\alpha_3)S(\beta_3) & S(\alpha_3) & C(\alpha_3)C(\beta_3) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.4)$$

$$\mathbf{T}_4^3 = \begin{bmatrix} C(\theta_4) & -C(\alpha_4)S(\theta_4) & S(\alpha_4)S(\theta_4) & C(\theta_4)px_4 \\ S(\theta_4) & C(\alpha_4)C(\theta_4) & -C(\theta_4)S(\alpha_4) & S(\theta_4)py_4 \\ 0 & S(\alpha_4) & C(\alpha_4) & pz_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.5)$$

$$\mathbf{T}_5^4 = \begin{bmatrix} C(\theta_5) & -C(\alpha_5)S(\theta_5) & S(\alpha_5)S(\theta_5) & C(\theta_5)px_5 \\ S(\theta_5) & C(\alpha_5)C(\theta_5) & -C(\theta_5)S(\alpha_5) & S(\theta_5)py_5 \\ 0 & S(\alpha_5) & C(\alpha_5) & pz_5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.6)$$

$$\mathbf{T}_{TCP}^5 = \begin{bmatrix} C(\theta_6) & -S(\theta_6) & 0 & C(\theta_6)px_6 - S(\theta_6)py_6 \\ S(\theta_6) & C(\theta_6) & 0 & C(\theta_6)py_6 + S(\theta_6)px_6 \\ 0 & 0 & 1 & pz_6 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.7)$$

3.3 Identificação de Parâmetros

A parte mais significativa do processo de identificação de parâmetros está relacionada aos métodos numéricos. Os procedimentos nos quais os parâmetros do modelo são identificados por meio de várias posições realizadas pelo elemento terminal do robô exigem métodos de otimização numérica. De maneira geral, esses métodos utilizam a linearização local do modelo não linear do robô e, em seguida, uma solução iterativa do problema de mínimos quadrados não lineares [6].

3.3.1 Modelo cinemático para identificação de parâmetros

As equações cinemáticas do manipulador são determinadas a partir do produto entre as transformações homogêneas do sistema de coordenadas da base para o último sistema de coordenadas, como expresso pela Eq.3.8.

$$\hat{\mathbf{T}}_N^0 = \hat{\mathbf{T}}_N^0(\mathbf{p}) = \mathbf{T}_1^0 \cdot \mathbf{T}_2^1 \dots \mathbf{T}_N^{N-1} = \prod_{i=1}^N \mathbf{T}_i^{i-1} \quad (3.8)$$

Na Equação 3.8, N corresponde ao número de juntas (ou sistemas de coordenadas), e \mathbf{p} é o vetor de parâmetros geométricos do modelo do robô, sendo definido como $\mathbf{p} = [p_1^T p_2^T \dots p_n^T]^T$. A transformação real para cada junta pode ser obtida por meio da Eq.3.9.

$$\mathbf{A}_i^{i-1} = \mathbf{T}_i^{i-1} + \Delta \mathbf{T}_i; \Delta \mathbf{T}_i = \Delta \mathbf{T}_i(\Delta \mathbf{p}_i) \quad (3.9)$$

O termo $\Delta \mathbf{p}_i$, na Eq. 3.9, é o vetor de erros dos parâmetros de elos para a junta i . Levando isso em consideração, a Eq. 3.8 passa a ser escrita na forma da Eq. 3.10.

$$\hat{\mathbf{A}}_N^0 = \prod_{i=1}^N (\mathbf{T}_i^{i-1} + \Delta \mathbf{T}_i) = \prod_{i=1}^N \mathbf{A}_i^{i-1} \quad (3.10)$$

Portanto, a transformação completa do manipulador com a adição do vetor de erros dos parâmetros passa a ser escrita na forma da Eq.3.11.

$$\hat{\mathbf{A}}_N^0 = \hat{\mathbf{T}}_N^0 + \Delta \hat{\mathbf{T}}, \Delta \hat{\mathbf{T}} = \Delta \hat{\mathbf{T}}(\mathbf{q}, \Delta \mathbf{p}) \quad (3.11)$$

Na Equação 3.11, $\Delta \mathbf{p} = [\Delta p_1^T \Delta p_2^T \dots \Delta p_n^T]^T$ é o vetor de erros dos parâmetros do manipulador e \mathbf{q} é o vetor de variáveis de juntas descrito como $[\theta_1^T, \theta_2^T, \theta_N^T]^T$. O termo $\Delta \hat{\mathbf{T}}$ é uma função não-linear do vetor $\Delta \mathbf{p}$. Para m posições medidas, pode-se afirmar que [15]:

$$\hat{\mathbf{A}} = \hat{\mathbf{A}}_N^0 = \hat{\mathbf{A}}(\mathbf{q}, \mathbf{p}) \quad (3.12)$$

A matriz $\hat{\mathbf{A}} : \mathbb{R}^n \times \mathbb{R}^N$ é uma função de dois vetores de dimensões n e N , onde n é o número de parâmetros e N é o número de juntas (incluindo a ferramenta). Com base nisso, têm-se as Eq. 3.13 e Eq. 3.14.

$$\hat{\mathbf{A}} = \hat{\mathbf{A}}_N^0 = \hat{\mathbf{A}}(\mathbf{q}, \mathbf{p}) = (\hat{\mathbf{A}}(q_1, \mathbf{p}), \dots, \hat{\mathbf{A}}(q_m, \mathbf{p}))^T : \mathbb{R}^n \times \mathbb{R}^{mN} \quad (3.13)$$

$$\Delta\hat{\mathbf{T}} = \Delta\hat{\mathbf{T}}(\mathbf{q}, \Delta\mathbf{p}) = (\Delta\hat{\mathbf{T}}(q_1, \Delta\mathbf{p}), \dots, \Delta\hat{\mathbf{T}}(q_m, \Delta\mathbf{p}))^T : \mathbb{R}^n \times \mathbb{R}^{mN} \quad (3.14)$$

Uma breve análise das equações acima possibilita verificar que todas as matrizes e vetores em negrito são funções de m . O processo de identificação consiste em calcular os valores dos parâmetros do modelo, $\mathbf{p}^* = \mathbf{p} + \Delta\mathbf{p}$, que resultam em um ajuste ideal entre as posições reais (obtidas por meio de instrumentos de medição), e as posições calculadas pelo modelo cinemático, ou seja, consiste na solução de um sistema de equações não-lineares representados pela Eq. 3.15.

$$\mathbf{B}(\mathbf{q}, \mathbf{p}^*) = \mathbf{M}(\mathbf{q}) \quad (3.15)$$

Na Equação 3.15, $\mathbf{B}(\mathbf{q}, \mathbf{p}^*)$ é um vetor formado pelas componentes de posição e orientação de $\hat{\mathbf{A}}$, $\mathbf{M}(\mathbf{q})$, por sua vez, corresponde às componentes medidas, indicadas na Eq. 3.16.

$$\mathbf{M}(\mathbf{q}) = (\mathbf{M}(q_1), \dots, \mathbf{M}(q_m))^T, \in \mathbb{R}^{\phi m} \quad (3.16)$$

O termo ϕ , na Eq. 3.16 é o número de equações advindas de cada posição de medição. Caso sejam considerados somente dados de posição (como é o caso do presente trabalho), cada medição de posição fornecerá três equações, e $\mathbf{B}(\mathbf{q}, \mathbf{p}^*)$ incluirá somente as componentes de posição de $\hat{\mathbf{A}}$. Como se trata de um problema onde se está tentando ajustar dados a um modelo não linear, os mínimos quadrados não-lineares é o método numérico de solução mais recomendado, sobretudo quando m é muito maior que n [15]. Neste caso, a Eq. 3.15 pode ser reescrita na forma da Eq. 3.17.

$$\mathbf{B}(\mathbf{q}, \mathbf{p}^*) = \mathbf{M}(\mathbf{q}) = \mathbf{B}(\mathbf{q}, \mathbf{p}) + \mathbf{C}(\mathbf{q}, \Delta\mathbf{p}) \quad (3.17)$$

O vetor \mathbf{C} presente na Eq. 3.17 representa o movimento diferencial formado pelas componentes de posição e rotação de $\Delta\hat{\mathbf{T}}$. Da definição da matriz Jacobiana e desprezando suas derivadas de segunda ordem, obtém-se uma expressão matemática para $\mathbf{C}(\mathbf{q}, \Delta\mathbf{p})$, indicada na Eq. 3.18.

$$\mathbf{C}(\mathbf{q}, \Delta\mathbf{p}) = \mathbf{J} \cdot \Delta\mathbf{p} \quad (3.18)$$

Substituindo a Equação 3.18 na Eq. 3.17 e reorganizando os termos, tem-se o modelo matemático representado pela Eq. 3.19.

$$\mathbf{M}(\mathbf{q}) - \mathbf{B}(\mathbf{q}, \mathbf{p}) = \mathbf{J} \cdot \Delta \mathbf{p} \quad (3.19)$$

A notação indicada nas Eq. 3.20, Eq. 3.21 e Eq. 3.22 permitem reescrever a Eq. 3.19 de acordo com a Eq. 3.23.

$$\mathbf{b} = \mathbf{M}(\mathbf{q}) - \mathbf{B}(\mathbf{q}, \mathbf{p}), \in \mathbb{R}^{\phi m} \quad (3.20)$$

$$\mathbf{J} = \mathbf{J}(\mathbf{q}, \Delta \mathbf{p}), \in \mathbb{R}^{\phi m \times n} \quad (3.21)$$

$$x = \Delta \mathbf{p}, \in \mathbb{R}^n \quad (3.22)$$

$$\mathbf{r} = \mathbf{J} \cdot x - \mathbf{b}, \in \mathbb{R}^{\phi m} \quad (3.23)$$

A Equação 3.23 pode ser resolvida pelo método dos mínimos quadrados lineares na forma da Eq. 3.24.

$$\mathbf{J} \cdot x = \mathbf{b} \quad (3.24)$$

Um método eficaz (no sentido de ser globalmente convergente) para solucionar problemas de mínimos quadrados não lineares é o algoritmo proposto por Levenberg-Marquardt (algoritmo LM), que combina as melhores propriedades do método do gradiente descendente e do método de Gauss-Newton. Dessa forma, o método LM, como é conhecido, é capaz de convergir rapidamente quando distante do mínimo (característica típica do método do gradiente), enquanto que assemelha-se ao método de Gauss-newton, à medida que o mínimo é aproximado com rápida convergência local. Outra vantagem do método LM é o fato dele ser bem definido, mesmo quando o Jacobiano não possui posto completo [6], [15], [1]. Com base na Eq. 3.24, este método pode ser formulado pela Eq. 3.25) [15], [12].

$$\mathbf{x}_{j+1} = \mathbf{x}_j - [\mathbf{J}(x_j)^T \cdot \mathbf{J}(x_j) + \mu_j \cdot \mathbf{I}]^{-1} \cdot \mathbf{J}(x_j)^T \cdot \mathbf{b}(x_j) \quad (3.25)$$

Portanto, cada iteração do algoritmo compreende as três etapas descritas abaixo:

1. Cálculo do Jacobiano do robô ($\mathbf{J}(x_j)$);
2. Cálculo do vetor $\Delta \mathbf{x}_j$ a partir da relação $\Delta \mathbf{x}_j = - [\mathbf{J}(x_j)^T \cdot \mathbf{J}(x_j) + \mu_j \cdot \mathbf{I}]^{-1} \cdot \mathbf{J}(x_j)^T \cdot \mathbf{b}(x_j)$;
3. Atualização de $\mathbf{x}_{j+1} = \mathbf{x}_j + \Delta \mathbf{x}_j$ e $j = j + 1$.

Nos passos listados acima, \mathbf{x}_j é o vetor de parâmetros do modelo cinemático na iteração j e $\Delta\mathbf{x}_j$ é o vetor que contém as correções a serem inseridas no modelo. O termo μ_j , conforme recomendações de Marquardt [54], é obtido a partir das relações matemáticas expressas na Eq. 3.26

$$\begin{cases} \mu_j = 0.001 \\ \mu_j = \lambda \cdot 0.001 \rightarrow \text{if } |\mathbf{b}(x_{j+1})| \geq |\mathbf{b}(x_j)| \\ \mu_j = \frac{0.001}{\lambda} \rightarrow \text{if } |\mathbf{b}(x_{j+1})| \leq |\mathbf{b}(x_j)| \\ 2.5 < \lambda < 10 \end{cases} \quad (3.26)$$

Os modelos cinemáticos parametrizados para calibração de robôs devem apresentar três propriedades importantes: completude, continuidade e minimalidade [11] e [12]. A observação desses requisitos do modelo permite evitar singularidades na matriz Jacobiana de identificação. Considerando isso, bem como o modelo cinemático desenvolvido aqui, é possível construir o modelo de identificação para o manipulador IRB-140, constituído pelas transformações presentes na Tab. 3.3 com adição de erros aos parâmetros geométricos. Esse modelo pode ser visualizado na Tab. 3.4

Tabela 3.4: Transformações paramétricas para o IRB-140.

Tipo de Junta	Parametrização D-H
w- b	$[T_x(px_b + \delta px_b).T_y(py_b + \delta py_b).T_z(pz_b + \delta pz_b).R_z(\alpha_b + \delta\alpha_b).R_y(\beta_b + \delta\beta_b).R_x(\gamma_b + \delta\gamma_b)]$
$J_R \perp J_R$	$[R_z(\theta_1 + \delta\theta_1).T_z(pz_1 + \delta pz_1).T_x(px_1 + \delta px_1).R_x(\alpha_1 + \delta\alpha_1)]$
$J_R \perp J_R$	$[R_z(\theta_2 + \delta\theta_2).T_z(pz_2 + \delta pz_2).T_x(px_1 + \delta px_1).R_x(\alpha_1 + \delta\alpha_1)]$
$J_R \parallel J_R$	$[R_z(\theta_3 + \delta\theta_3).T_x(px_3 + \delta px_3).R_x(\alpha_3 + \delta\alpha_3).R_y(\beta_3 + \delta\beta_3)]$
$J_R \perp J_R$	$[R_z(\theta_4 + \delta\theta_4).T_z(pz_4 + \delta pz_4).T_x(px_4 + \delta px_4).R_x(\alpha_4 + \delta\alpha_4)]$
$J_R \perp J_R$	$[R_z(\theta_5 + \delta\theta_5).T_z(pz_5 + \delta pz_5).T_x(px_5 + \delta px_5).R_x(\alpha_5 + \delta\alpha_5)]$
$J_R \parallel TCP$	$[R_z(\theta_6 + \delta\theta_6).T_x(px_6 + \delta px_6).T_y(py_6 + \delta py_6).T_z(pz_6 + \delta pz_6)]$

Fonte: Produzido pela Autora.

As sentenças matemáticas presentes na Tab. 3.4 correspondem aos elementos da Eq. 3.9, e são definidas pelo produto de matrizes de rotação e translação para cada transformação homogênea entre os sistemas de referência do robô, e entre os sistemas de referência universal e da base, ambos ilustrados na Fig. 3.2. Os parâmetros da Tab. 3.4 que não estão indicados na Fig. 3.2 possuem valores iniciais nulos. Uma análise da Tab. 3.4 possibilita constatar que o número de parâmetros do modelo é igual a 30 (seis parâmetros da base e vinte e quatro parâmetros pertencentes aos elos internos da cadeia cinemática), atestando a conformidade com a Eq.2.49.

3.4 Otimização do Modelo de Identificação

A otimização do modelo é processada em duas etapas. Primeiro é verificado se o número de condicionamento do Jacobiano $k(\mathbf{J})$ está acima de 100; o segundo passo identifica qual parâmetro do modelo produz deficiências no posto do Jacobiano. Com isso, um modelo ótimo é derivado do modelo completo pela exclusão de um pequeno número de parâmetros do modelo do processo de identificação, até que $k(\mathbf{J})$ seja inferior a 100 [12]. A Figura 3.3 ilustra as etapas do processo de calibração do robô com a inclusão do passo de otimização do modelo cinemático de identificação.

Em termos práticos, a maior parte dos sistemas de medição disponíveis empregados em metrologia robótica, tais como sistemas *laser*, ultrassom, câmeras CCD, teodolitos, etc., possuem uma precisão média que varia entre 0,05 mm a 0,5 mm, dependendo do seu tipo ou tecnologia. Dessa forma, quando são adicionados ruídos de medição maiores devido ao uso sistemas de medição menos precisos, o valor de $k(\mathbf{J})$ aumenta, de maneira que o limite de $k(\mathbf{J}) = 100$ não será alcançado com a exclusão de um pequeno número de parâmetros. Se o $k(\mathbf{J})$ apresentar pouca melhoria com a exclusão do próximo parâmetro, o processo de otimização deve ser interrompido. É importante frisar que, como a otimização objetiva excluir, um por vez, cada parâmetro do modelo parametrizado, independente do número de parâmetros, sempre haverá uma melhoria no número de condicionamento. Entretanto, se um parâmetro diferente daquele identificado pela rotina de otimização for excluído do modelo, não haverá garantia de que o número de condicionamento irá melhorar ou de que haverá melhoria na precisão do modelo cinemático [12].

3.5 Simulação do Processo de Calibração

Nesta etapa da pesquisa foi realizada uma simulação computacional da calibração do manipulador IRB-140 (de seis graus de liberdade), utilizando a técnica de otimização mostrada anteriormente.

A simulação computacional do processo de calibração compreende todas as etapas de uma calibração real, ou seja, modelagem cinemática, identificação dos parâmetros a partir de posições selecionadas (representando medições) e avaliação da calibração. Foram adotados dois modelos cinemáticos: *Simulated Pose Model* (SPM) e *Estimated Pose Model* (EPM). O primeiro modelo inclui os parâmetros geométricos e a adição de erros aleatórios nas variáveis de juntas. Os desvios aleatórios das variáveis de juntas são considerados como a principal causa de desvios do robô, e podem ser medidos experimentalmente e estaticamente para determinar a repetibilidade do robô [6]. O segundo modelo, por outro lado, corresponde ao modelo para identificação de parâmetros e contém todas as fontes de erros geométricos identificadas no primeiro modelo, com exceção dos erros dos parâmetros que foram eliminados, a fim de melhorar a identificabilidade do modelo. A simulação do processo de calibração foi realizada de acordo com o algoritmo apresentado na Fig. 3.4.

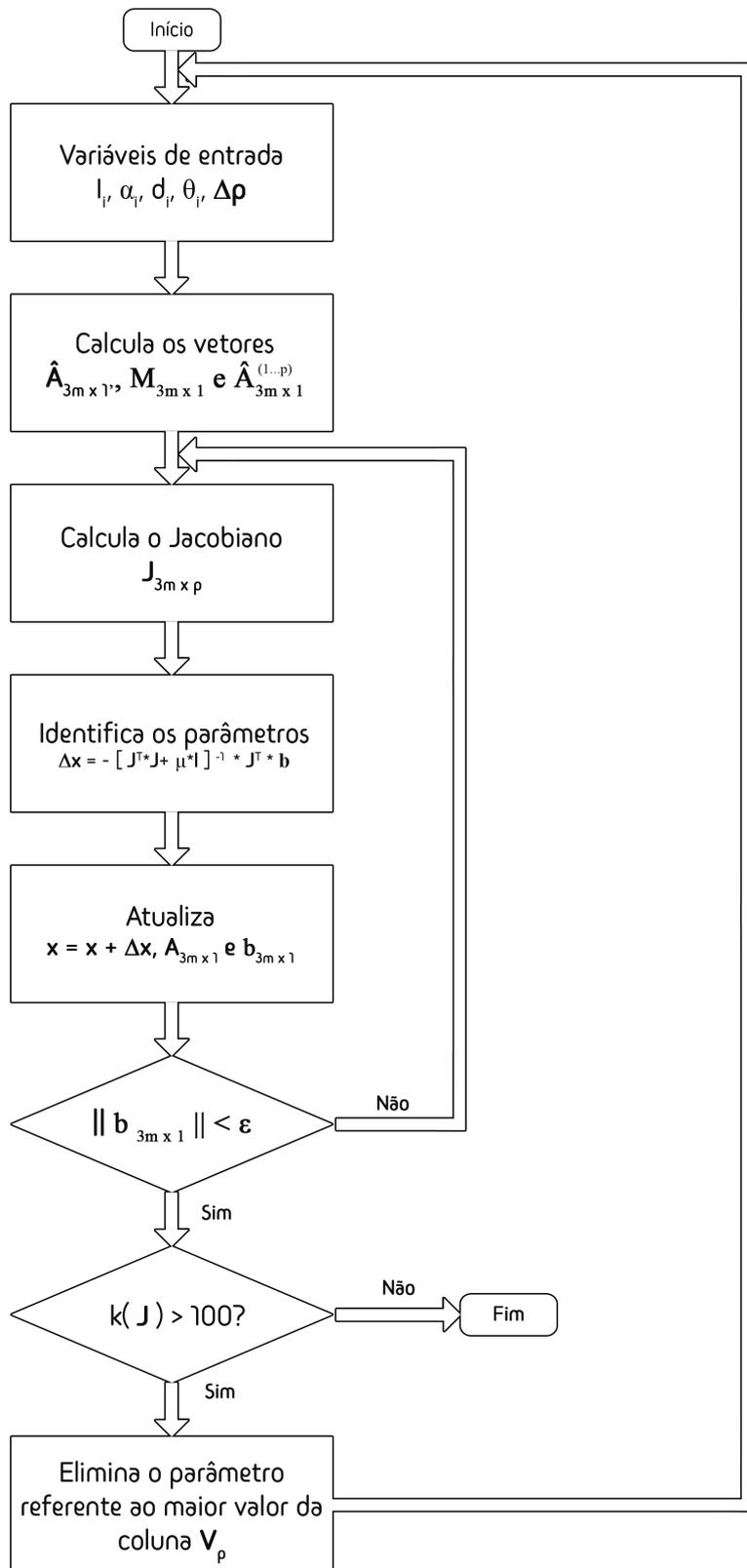


Figura 3.3: Fluxograma do algoritmo de calibração contendo a etapa de otimização do modelo.
 Fonte: Produzido pela Autora.

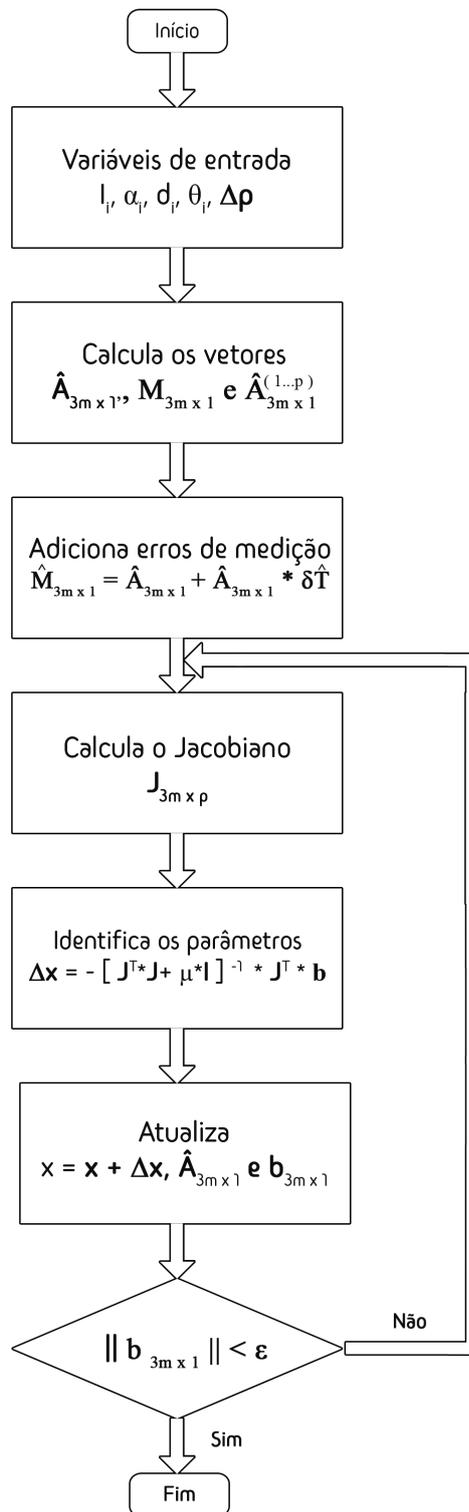


Figura 3.4: Fluxograma do algoritmo de simulação do processo de calibração. Fonte: Produzido pela Autora.

Nesta simulação foram utilizados somente dados de posição, devido às limitações dos sistemas de medição. Em termos matemáticos, cada ponto de medição forneceu três equações de medida. Com o intuito de minimizar os ruídos de medição, o número de equações precisa ser duas ou três vezes maior que o número de parâmetros a serem estimados. Para o manipulador IRB-140, o número máximo de parâmetros é 30, o que acarretou na escolha de 30 pontos de medição [21]. Para fins práticos, quanto menor o número de pontos de medição utilizados na calibração, menor é o tempo de inatividade da máquina em uma linha de produção, diminuindo os custos.

Os erros geométricos constantes a serem adicionados aos valores dos parâmetros cinemáticos do robô foram selecionados aleatoriamente entre ± 1 mm, para os parâmetros de translação, e ± 0.0005 rad, para os parâmetros de orientação, de acordo com os desvios comuns encontrados na literatura para esta classe de robôs. Os valores nominais dos parâmetros e os valores de erros assumidos encontram-se na Tab. 3.5.

Os erros característicos do sistema de medição, por sua vez, foram incorporados ao modelo por meio da Eq. 3.27 [6].

$$\hat{\mathbf{M}} = \hat{\mathbf{A}}_N^0 + \hat{\mathbf{A}}_N^0 \cdot \delta\hat{\mathbf{T}} \quad (3.27)$$

Na Equação 3.27, $\delta\hat{\mathbf{T}}$ é a matriz de transformação de movimento diferencial (erros de medição) para as três componentes de posição, calculada por meio da Eq. 3.28; $\hat{\mathbf{A}}_N^0$ é a transformação real do manipulador com adição de desvios aos parâmetros, apresentada na Eq. 3.10; e $\hat{\mathbf{M}}$ é a mesma transformação, porém, com adição de erros de medição.

$$\delta\hat{\mathbf{T}} = \begin{bmatrix} 0 & 0 & 0 & \gamma_x \\ 0 & 0 & 0 & \gamma_y \\ 0 & 0 & 0 & \gamma_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.28)$$

Os elementos que formam a quarta coluna da Eq. 3.28 são gerados de maneira randômica e com distribuição de probabilidade gaussiana, de acordo com a Eq. 3.29.

$$\gamma = p + q \cdot randn(i, 1) \quad (3.29)$$

Na Equação 3.29, o termo *randn* produz um vetor coluna de números aleatórios, onde o número de linhas i desse vetor é igual a três vezes o número de pontos medidos. A variável p , por sua vez, é a média, e q é o desvio padrão do sistema de medição, calculado a partir da Eq. 3.30.

$$q = \sigma_r + \sigma_s \quad (3.30)$$

Tabela 3.5: Desvios adicionados aos parâmetros.

Parâmetro	Valor nominal	Valor de erro assumido
$px_b[mm]$	*	0,05
$py_b[mm]$	*	0,035
$pz_b[mm]$	*	-0,08
$\alpha_b[rad]$	0	0,0001
$\beta_b[rad]$	0	0,0015
$\gamma_b[rad]$	0	-0,0001
$\alpha_1[rad]$	$-\pi/2$	-0,0001
$pz_1[mm]$	352	1,0
$\theta_1[rad]$	0	0,0003
$px_1[mm]$	70	0,08
$\alpha_2[rad]$	0	0,0001
$pz_2[mm]$	0	-0,05
$\theta_2[rad]$	$-\pi/2$	-0,0005
$px_2[mm]$	360	-1,0
$\alpha_3[rad]$	$-\pi/2$	0,0003
$beta_3[rad]$	0	-0,0001
$\theta_3[rad]$	0	0,0005
$px_3[mm]$	0	-0,04
$\alpha_4[rad]$	$\pi/2$	-0,0001
$pz_4[mm]$	380	1,0
$\theta_4[rad]$	0	-0,0004
$px_4[mm]$	0	0,02
$\alpha_5[rad]$	$-\pi/2$	0,0015
$pz_5[mm]$	0	-0,25
$\theta_5[rad]$	0	-0,0035
$px_5[mm]$	0	-0,05
$py_6[mm]$	0	0,01
$pz_6[mm]$	65 + 120 (ferramenta)	-1,0
$\theta_6[rad]$	0	0,0003
$px_6[mm]$	0	-0,02

Fonte: Produzido pela Autora.

Os termos σ_r e σ_s presentes na Eq. 3.30, são, respectivamente, o desvio padrão dos erros do manipulador em cada direção e o desvio padrão do sistema de medição. A primeira parcela da Eq. 3.30 se relaciona com a repetibilidade do manipulador por meio da Eq. 3.31.

$$R_p = \sqrt{\sigma_{r_x}^2 + \sigma_{r_y}^2 + \sigma_{r_z}^2} \quad (3.31)$$

O termo R_p , presente na Eq. 3.31, corresponde à repetibilidade de posicionamento do robô, sendo igual a 0,03 mm para o IRB-140, e as parcelas σ_{r_x} , σ_{r_y} e σ_{r_z} são os desvios padrão para cada componente do vetor de posição. Considerando que $\sigma_{r_x} = \sigma_{r_y} = \sigma_{r_z} = \sigma_r$, essa equação pode ser reescrita na forma da Eq. 3.32.

$$\sigma_r = \frac{R_p}{\sqrt{3}} \quad (3.32)$$

Para analisar a influência do erro de medição sobre o erro de posicionamento do manipulador, foi variado a média de medição p e assumido um desvio padrão de medição q constante, de maneira que a parcela do desvio padrão relacionada ao sistema de medição, σ_s , foi considerada igual a 0,1 mm.

Capítulo 4

Resultados e Discussão

Este capítulo apresenta e discute os resultados obtidos com o emprego das técnicas de otimização para melhorar a identificação dos parâmetros. Adicionalmente, são avaliados os resultados oriundos da simulação computacional do processo de calibração, onde foram considerados tanto os erros geométricos do modelo cinemático nominal, como os erros inerentes aos sistemas de medição.

4.1 Otimização do Modelo de Calibração

A aplicação do método de otimização do modelo, detalhado anteriormente, permitiu identificar os parâmetros redundantes e construir o modelo EPM a partir da análise do número de condicionamento do Jacobiano e da ferramenta numérica de decomposição em valores singulares. Os resultados experimentais estão mostrados na Tab. 4.1.

Tabela 4.1: Dados obtidos durante a etapa de otimização do modelo cinemático.

N°	30	29	28	27	26	25	24
K	$1,37 \times 10^9$	$4,56 \times 10^8$	$3,72 \times 10^6$	$2,70 \times 10^6$	345,05	111,59	53,67
Iter.	5	5	5	5	5	4	4
Par.	θ_1	β_3	pz_1	θ_6	px_5	pz_5	-

Fonte: Produzido pela Autora.

Na Tabela 4.1, N° é o número de parâmetros presentes no modelo; K corresponde ao número de condicionamento, $Iter.$ é o número de iterações e $Par.$ é o parâmetro identificado como redundante. Analisando os dados presentes na Tab. 4.1, verifica-se que o processo de otimização do modelo iniciou com o número máximo de parâmetros (30). Cada parâmetro eliminado (indicados na quarta linha da Tab. 4.1), correspondem ao menor valor singular do modelo, observado a partir da maior entrada na última coluna da matriz V , obtida da SVD do Jacobiano corrigido pelos

valores singulares escalonados.

Durante o processo de otimização, com frequência dois parâmetros apresentaram o mesmo maior valor no intervalo $0 < v_i < 1$, onde v_i é a entrada na linha da última coluna de V . Por exemplo, no modelo com 29 parâmetros foram identificadas duas linhas na última coluna de V com os maiores valores numericamente iguais a 0,7071. Essas linhas correspondiam aos parâmetros θ_3 e β_3 , o que significa que eles são mutuamente redundantes. Neste caso, o parâmetro eliminado foi o β_3 , pois diferentemente do θ_3 , apresenta valor nulo no modelo nominal, ou seja, não é necessário para o modelo nominal.

A Figura 4.1 ilustra a evolução do número de condicionamento em função do número de parâmetros do modelo.

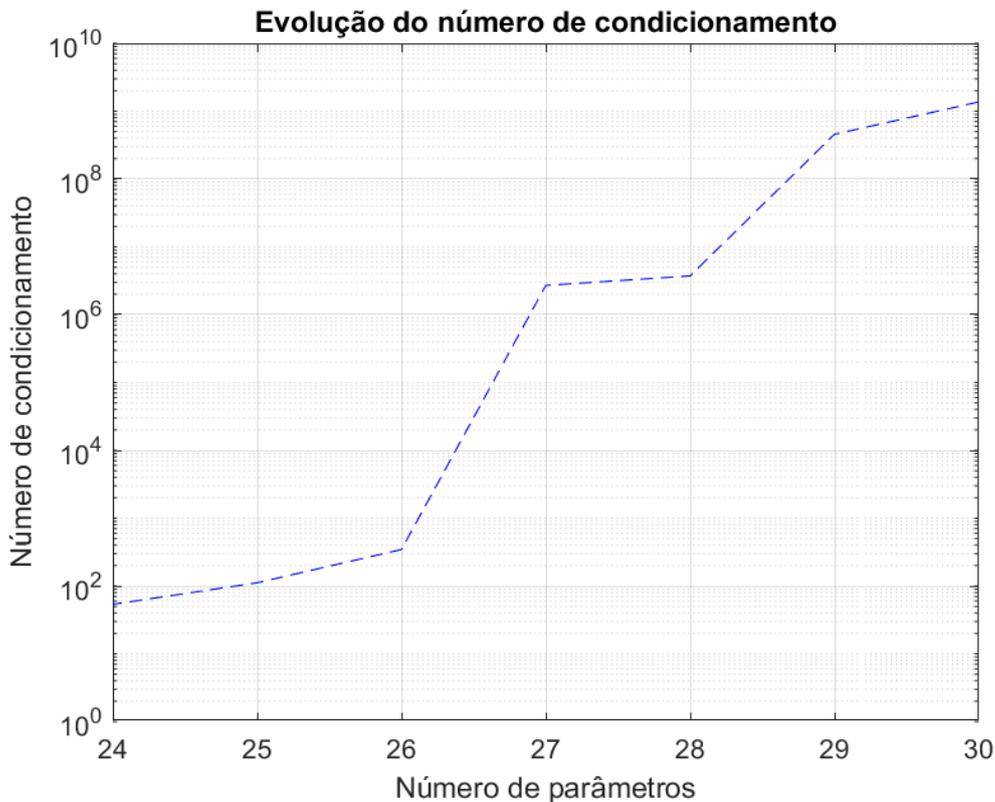


Figura 4.1: Número de condicionamento da matriz Jacobiana calculado durante a etapa de identificação como uma função do número de parâmetros. Fonte: Produzido pela Autora.

Os resultados mostrados na Tab. 4.1 e na Fig. 4.1 evidenciam que, à medida que um parâmetro é removido do modelo, o número de condicionamento do Jacobiano também reduz. Dessa forma, chegou-se a um modelo com 24 parâmetros (valor abaixo do número máximo de parâmetros identificáveis, calculado pela Eq. 2.50), cujo número de condicionamento é aproximadamente igual a 54, que é inferior a 100. Além disso, altos valores no número de condicionamento (como ocorre para o modelo de 30 parâmetros, que apresenta esse valor na ordem de 10^9), indica uma

solução mal condicionada, o que pode levar a um maior número de iterações para a convergência do algoritmo de Levenberg-Marquardt, ou a possibilidade de um mínimo local ser alcançado mesmo com uma boa convergência, ou ambas as situações. Isso pode ser observado na Fig. 4.2, onde se observa que o número de iterações é maior para os modelos com 30, 29, 28, 27 e 26 parâmetros, que, por sua vez, apresentam números de condicionamento muito superior a 100.

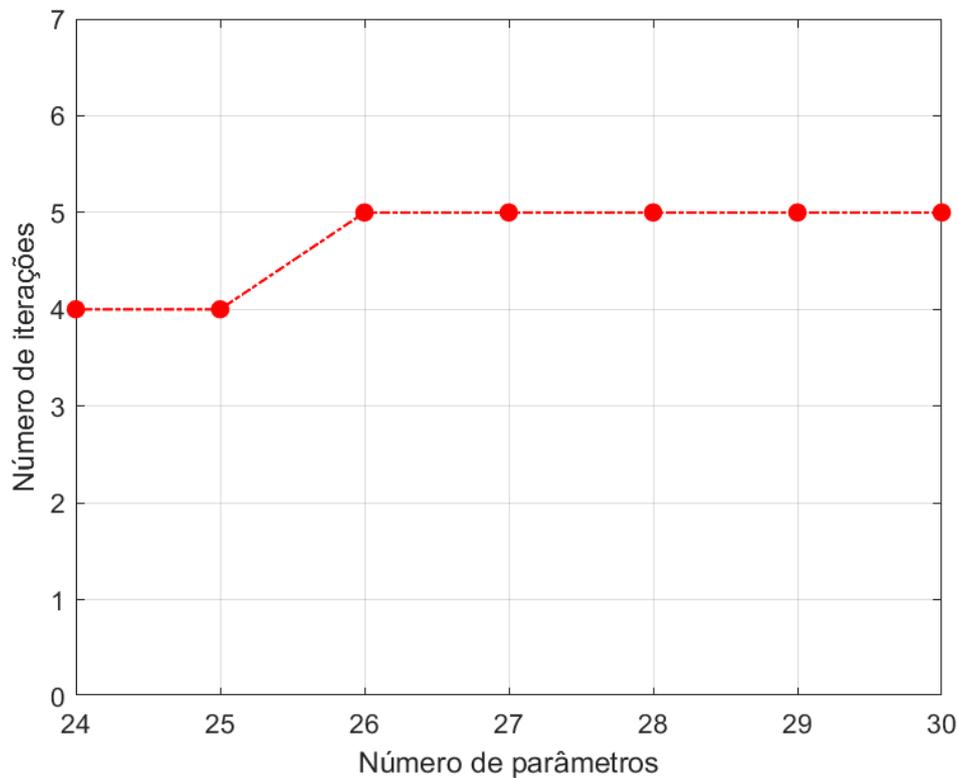


Figura 4.2: Número de iterações para convergência do algoritmo de Levenberg-Marquardt como uma função do número de parâmetros. Fonte: Produzido pela Autora.

De forma geral, o método de otimização demonstrou a sua eficácia, identificando os parâmetros que provocavam deficiências no posto do Jacobiano. Isso possibilitou eliminá-los, permitindo obter um modelo com o menor número de parâmetros possíveis.

4.2 Processo de Simulação da Calibração

O processo de calibração simulado computacionalmente foi avaliado por meio da precisão de posicionamento do efetuador na região considerada, sendo expressa, numericamente, pela média e desvio padrão dos erros, calculados, respectivamente, pelas Eq. 4.1 e 4.2.

$$\bar{E} = \frac{\sum_{i=1}^n |\mathbf{M}_i - \hat{\mathbf{A}}_i|}{n} \quad (4.1)$$

$$\sigma = \sqrt{\frac{\sum_{i=1}^n \left(|\mathbf{M}_i - \hat{\mathbf{A}}_i| - \bar{E} \right)^2}{n}} \quad (4.2)$$

Nas Equações 4.1 e 4.2, \mathbf{M} é um vetor contendo as componentes de posição mensuradas, $\hat{\mathbf{A}}$ é a transformação do manipulador calculada com a adição do vetor de erros dos parâmetros, e n é o número de pontos de medição.

Durante a simulação, foi utilizado o mesmo modelo cinemático do robô real, porém, com valores de parâmetros geométricos modificados para aqueles obtidos na etapa de identificação de erros. Além disso, foram gerados erros aleatórios, visando simular o sistema de medição e deixar o procedimento mais próximo do que seria uma calibração real.

Na simulação do processo de calibração, inicialmente foi averiguada a influência do erro de medição adotado sobre o erro de posicionamento, utilizando o modelo mínimo de 24 parâmetros. A Figura 4.3 ilustra os resultados obtidos e demonstra que, quanto maior o erro relacionado ao sistema de medição, maior será o erro médio e o desvio padrão associado ao posicionamento do robô. Somado a isso, verifica-se que, após a calibração, o desvio padrão do manipulador (cerca de 0,02 mm) é acrescido do desvio padrão do erro de medição.

Dentre os fatores que influenciam nos resultados da calibração de robôs, destaca-se o número de pontos de medição. Como nesta simulação foram utilizados somente dados de posição, devido às limitações dos sistemas de medição, em termos matemáticos, cada ponto de medição forneceu três equações de posição. Com o intuito de minimizar os ruídos de medição, o número de equações precisa ser duas ou três vezes maiores que o número de parâmetros a serem estimados[21]. Levando isso em consideração, e sabendo que o número máximo de parâmetros para o IRB-140 é 30, adotaram-se 30 pontos de medição como o valor padrão. Entretanto, avaliou-se a influência da quantidade de pontos de medição na precisão do modelo mínimo calibrado, considerando conjuntos de dados com 10, 20, 30, 40, 50 e 60 pontos.

A Figura 4.4 apresenta as curvas para os erros médio e máximo de posição, assim como a curva do desvio padrão do erro, em função de diferentes conjuntos de medições.

A observação da Fig. 4.4 permite aferir uma melhoria irrelevante na precisão para um conjunto acima de 30 pontos, ou seja, quando o número de posições medidas excede esse valor, os erros médios de posições de calibração tendem a ser estáveis e menores que 0,5 mm. Isso demonstra que um conjunto com 30 pontos é suficiente para realizar a calibração do manipulador.

Adicionalmente, também foi analisado a melhoria na etapa de identificação devido às técnicas de otimização do modelo. Para tanto, foram confrontadas a curva de precisão do modelo não

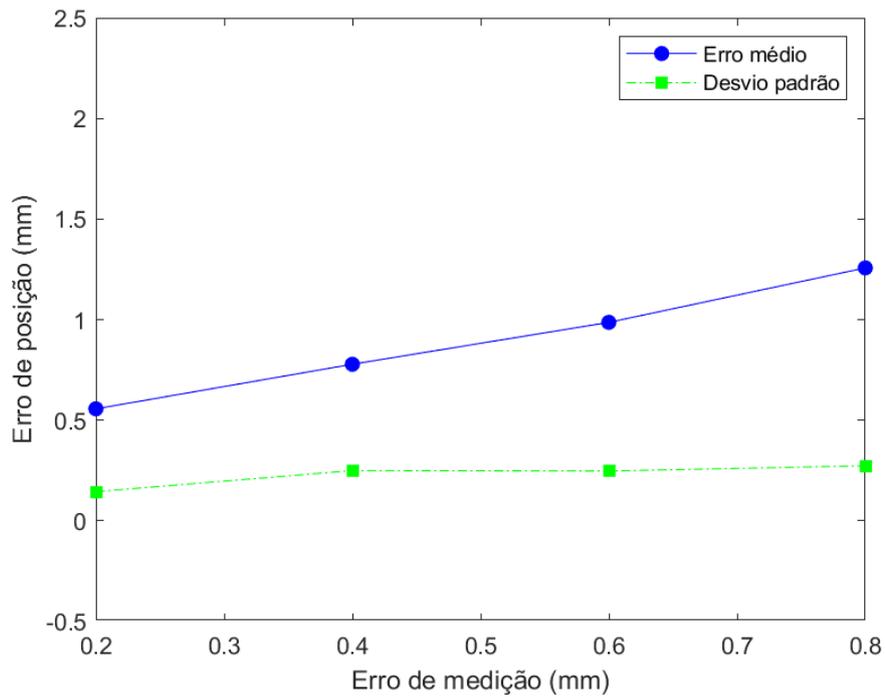


Figura 4.3: Erro de posicionamento do modelo mínimo em função do erro do sistema de medição. Fonte: Produzido pela Autora.

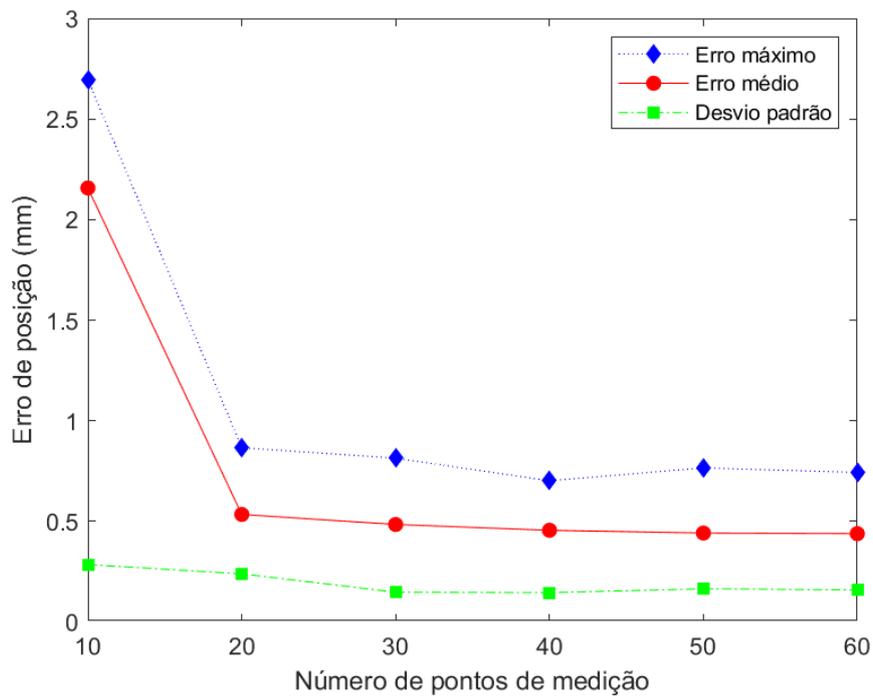


Figura 4.4: Precisão do elemento terminal em função do número de pontos de medição. Fonte: Produzido pela Autora.

otimizado de 30 parâmetros com a curva do modelo otimizado de 24 parâmetros em função dos mesmos conjuntos de medições discutidos anteriormente. Os resultados obtidos estão ilustrados na Fig. 4.5.

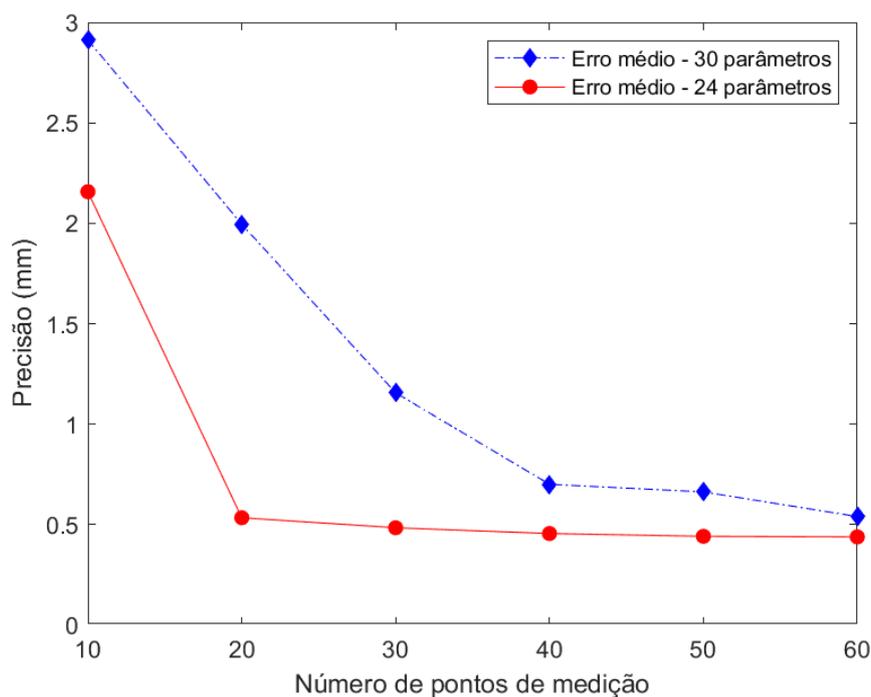


Figura 4.5: Curvas de precisão do elemento terminal para um modelo otimizado e um modelo não-otimizado em função do número de pontos de medição. Fonte: Produzido pela Autora.

Na Figura 4.5 é possível verificar que, para os conjuntos de dados considerados, os valores de erro médio são maiores para o modelo não-otimizado com 30 parâmetros do que para o modelo ótimo com 24 parâmetros, sendo mais acentuados para pequenos volumes de dados, ou seja, para 10 e 20 pontos de medição. Além disso, atesta-se que o modelo com 30 parâmetros tem uma diminuição significativa no valor do erro a partir de 40 pontos de medição. Esses resultados demonstram que o procedimento de otimização diminui o número de pontos a serem medidos e melhora a precisão do modelo de calibração. Os valores do número de condicionamento para o modelo não-otimizado, para cada volume de dados de medição, estão presentes na Tab. 4.2.

A Tabela 4.2 mostra que os valores do número de condicionamento do Jacobiano para o modelo não otimizado é da ordem de 10^{24} ou acima dessa magnitude, isto é, superior a 100. Inspeccionando a Tabela 4.2, observa-se que o número de condicionamento sofre uma pequena diminuição entre os volumes com 10 e 40 pontos medidos, aumentando novamente nos volumes com 50 e 60 pontos, o que indica uma solução mal condicionada.

A Figura 4.6 mostra o desvio no efetuador final para cada ponto do conjunto de 30 medições, calculado a partir do modelo mínimo calibrado com adição de erros aleatórios para simular o

Tabela 4.2: Número de condicionamento para o modelo não otimizado.

Número de pontos medidos	K
10	$2,41 \times 10^{26}$
20	$1,12 \times 10^{25}$
30	$9,99 \times 10^{24}$
40	$6,43 \times 10^{24}$
50	$8,71 \times 10^{25}$
60	$1,03 \times 10^{25}$

Fonte: Produzido pela Autora.

sistema de medição. A investigação da curva da Fig.4.6 possibilita constatar que o erro máximo de posição é 0,74 mm, valor inferior à precisão exigida na indústria automotiva (cerca de 2,5 mm), em peças fabricadas por robôs programados *off-line* [90].

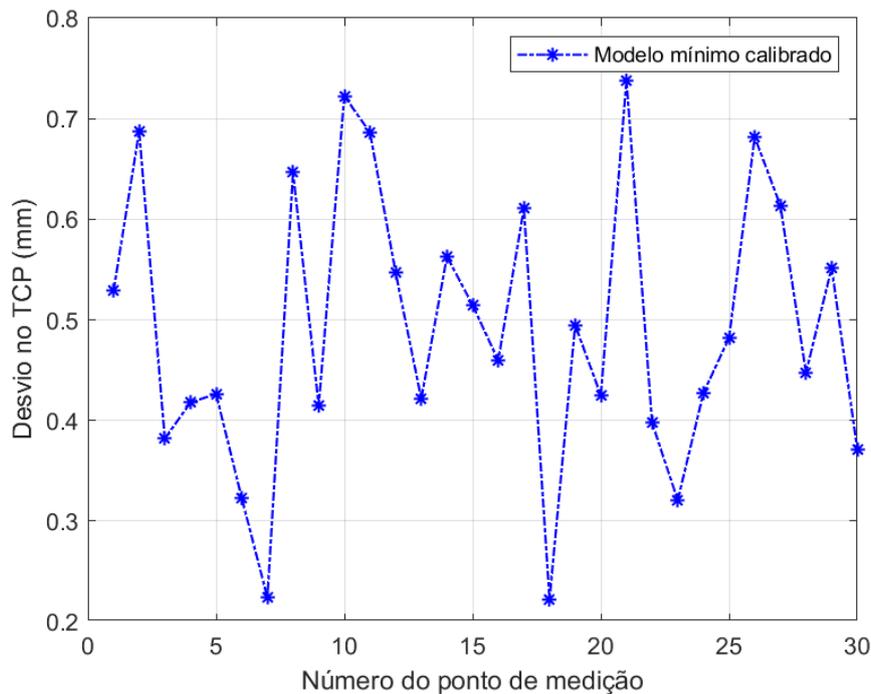


Figura 4.6: Precisão de posicionamento obtido a partir da aplicação do modelo de calibração otimizado. Fonte: Produzido pela Autora.

A distribuição de frequências para os desvios no efetuador, ilustrados anteriormente, está indicada na Fig. 4.7. Nesta representação gráfica em colunas (histograma), é possível identificar que o valor mais comum de erro, dentre as posições mensuradas, é igual a 0,4 mm. A dispersão do erro, por outro lado, ocorre entre 0,23 mm e 0,74 mm.

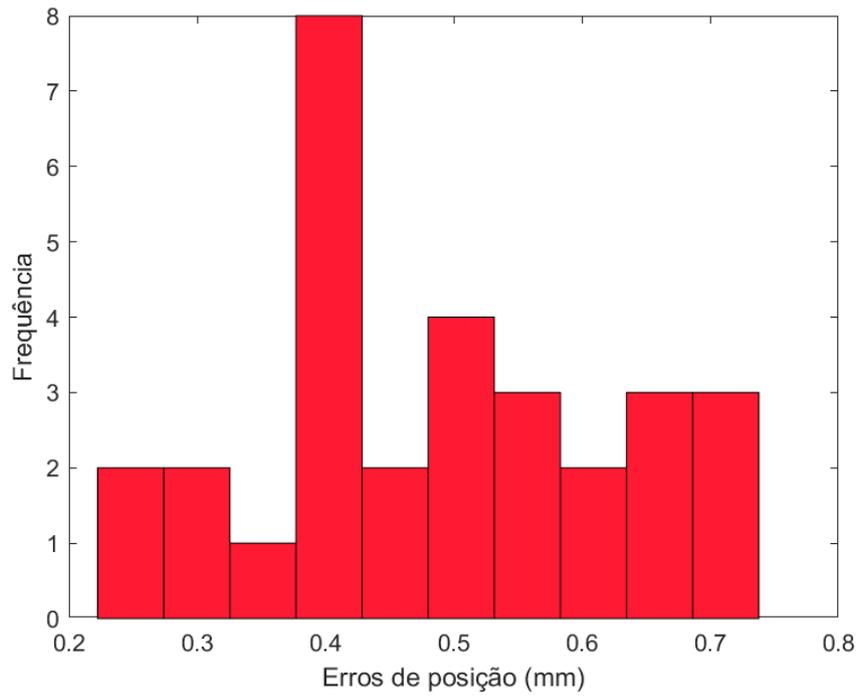


Figura 4.7: Histograma dos desvios no TCP. Fonte: Produzido pela Autora.

A precisão do modelo mínimo calibrado também foi analisada, e os resultados podem ser visualizados na Fig. 4.8, onde o erro médio e o desvio padrão a cada iteração foram calculados, respectivamente, pelas Eq. 4.1 e 4.2.

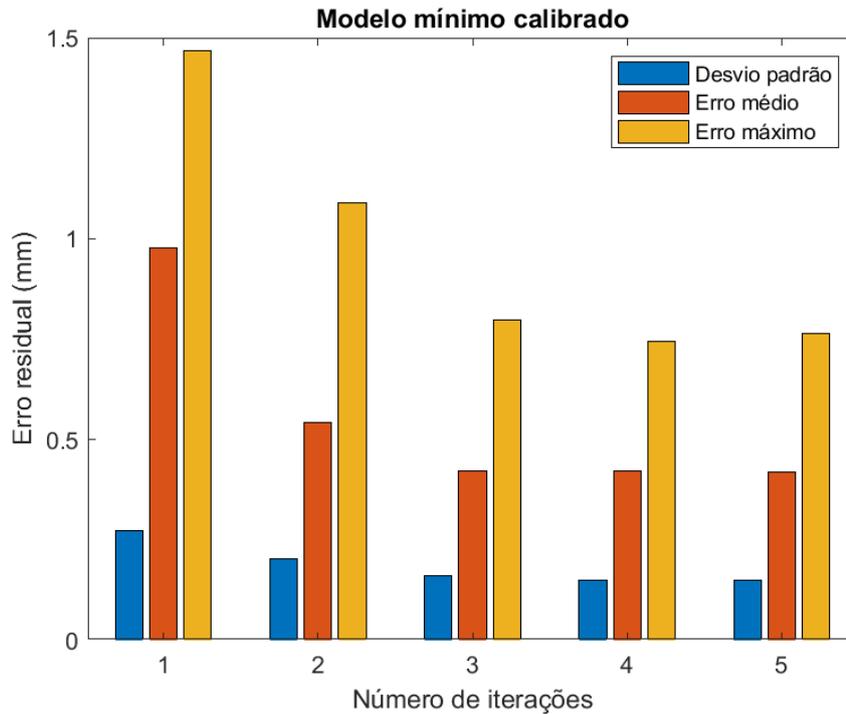


Figura 4.8: Avaliação do modelo calibrado a cada iteração. Fonte: Produzido pela Autora.

Os resultados presentes na Fig. 4.8 exprimem a precisão de posicionamento do efetuador, calculada como a média dos valores absolutos dos erros residuais entre a posição simulada e a posição esperada (nominal) do efetuador, a cada iteração do modelo mínimo calibrado. Além disso, também estão representados o valor máximo e o desvio padrão obtidos a cada repetição do algoritmo de solução. Analisando o gráfico, verifica-se que os erros de posição foram reduzidos de aproximadamente 1,0 mm (na primeira iteração) para 0,4 mm (na última iteração).

Capítulo 5

Considerações Finais

5.1 Conclusões

Neste trabalho foi implementado um sistema de calibração de robôs que utiliza técnicas de otimização do modelo matemático para identificação dos parâmetros, visando eliminar deficiências e redundâncias inerentes ao modelo cinemático. Para tanto, previamente foi realizada uma revisão bibliográfica pormenorizada sobre programação *off-line*, calibração de manipuladores, modelagem cinemática, identificação de parâmetros, sistemas de medição e otimização do modelo. Na metodologia foi detalhada a modelagem cinemática do manipulador IRB-140, incluindo a atribuição de sistemas de coordenadas aos elos e a obtenção das transformações elementares empregando as convenções de DH e HM. De posse dessas transformações, foi construído o modelo de identificação com adição de erros aos parâmetros geométricos do manipulador industrial IRB-140. Ainda na metodologia, foi apresentado o algoritmo de calibração contendo a etapa de otimização do modelo, assim como as considerações práticas para realizar a simulação computacional das etapas do processo de calibração.

A otimização do modelo cinemático possibilitou reduzir o número de parâmetros no modelo de 30 para 24, de maneira que o número de condicionamento caiu de 10^9 para 54. Além disso, foi verificado que a presença de parâmetros redundantes demanda um maior número de iterações para convergência do algoritmo de Levenberg-Marquardt.

Na simulação computacional da calibração, inicialmente foi avaliado a influência do erro de medição adotado no erro de posicionamento do modelo mínimo de 24 parâmetros, empregando um desvio padrão constante. Dentre os resultados obtidos, verificou-se um aumento no erro médio e no desvio padrão relacionados ao posicionamento do robô, à medida que o erro inerente ao sistema de medição era aumentado.

Durante a simulação foram introduzidos erros aleatórios no modelo cinemático nominal, e erros para simular o sistema de medição. Com isso, foi possível averiguar a influência da quan-

tidade de pontos de medição na precisão da calibração, atestando que, acima de 30 pontos, é insignificante a melhoria em termos de precisão. Paralelamente, também se avaliou a melhoria na identificação em virtude das técnicas de otimização, confrontando as curva de precisão do modelo não otimizado de 30 parâmetros com a curva do modelo otimizado de 24 parâmetros. Desse resultado, foi constatado que, para os volumes de dados considerados, o erro para o modelo não otimizado é sempre maior que o erro para o modelo otimizado, sendo que o modelo não otimizado tem uma redução significativa do erro somente a partir de 40 pontos de medição. Isso implica em dizer que o procedimento de otimização dos parâmetros diminui o número de pontos a serem medidos e melhora a precisão do modelo.

Dispondo do modelo mínimo calibrado e 30 pontos de medição, também foram investigados os desvios no TCP para cada ponto de medição, permitindo detectar um erro máximo de 0,74 mm, que é inferior ao erro admitido pela indústria automotiva no que se refere a peças fabricadas por manipuladores programados *off-line* [90]. O histograma de desvios no TCP mostrou que, para a simulação realizada com o conjunto de 30 pontos de medição, o valor mais frequente de erro foi igual a 0,4 mm, e que a dispersão do erro ocorreu entre 0,23 mm e 0,74 mm.

Por fim, foi realizada uma análise da precisão de posicionamento do efetuador, a cada iteração, utilizando, para tanto, o modelo mínimo calibrado. O gráfico gerado permitiu atestar que os erros de posição foram reduzidos de cerca de 1,0 mm (na primeira iteração), para 0,4 mm (na última iteração).

5.2 Trabalhos Futuros

Com base no conhecimento e experiência adquiridos neste trabalho, bem como nos resultados obtidos, pretende-se aplicar o método aqui desenvolvido para realizar a calibração cinemática de um manipulador paralelo, objetivando obter um modelo que seja completo e contínuo e que, simultaneamente, apresente o menor número de parâmetros.

Posteriormente, almeja-se implementar um algoritmo de calibração simultânea *hand-eye*, *tool-flange* e *robot-robot* baseado no método de otimização paramétrica do modelo de erros. A fim de analisar a precisão e a convergência do modelo cinemático de calibração, serão realizadas simulações e experimentos reais com dois robôs, que consistirá em uma câmera montada no manipulador serial IRB-140, e um padrão de medição (tabuleiro de xadrez) fixado no efetuador do manipulador paralelo IRB-360.

5.3 Trabalhos Publicados

- Artigo publicado em conferência

J. S. Toquica, P. S. Oliveira, J. M. S. T. Motta, and D. L. Borges, “A proposal to solve the inverse kinematics problem of a parallel robot configuration with neural networks”, in *Proceedings 2018 CIE 48. The 48th International Conference on Computers and Industrial Engineering*, vol. 1. CIE 48, 2018, pp. 1-16.

- **Artigo submetido para revista**

J. S. Toquica, P. S. Oliveira, Witenberg S. R. Souza, J. M. S. T. Motta, and D. L. Borges, “An Analytical and a Deep Learning Model for Solving the Inverse Kinematic Problem of an Industrial Parallel Robot”, *Computer & Industrial Engineering*, pp. 1-29, 2019.

- **Artigo completo submetido para congresso**

P. S. Oliveira and J. M. S. T. Motta, “Kinematics calibration of a 6-D.O.F. Manipulator using a model optimization routine”, in *[2019] 25th International Congress of Mechanical Engineering. ABCM*, 2019.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] L. S. Ginani and J. M. S. Motta, “Theoretical and practical aspects of robot calibration with experimental verification,” *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol. 33, no. 1, pp. 15–21, 2011.
- [2] J. M. S. Motta, “Robot calibration: Modeling measurement and applications,” in *Industrial Robotics: Programming, Simulation and Applications*. IntechOpen, 2006.
- [3] B. W. Mooring, Z. S. Roth, and M. R. Driels, *Fundamentals of manipulator calibration*. Wiley New York, 1991.
- [4] J. O. Berg, “Robot calibration for off-line programming,” *Industrial Robot: An International Journal*, vol. 18, no. 2, pp. 29–31, 1991.
- [5] B. Karan and M. Vukobratović, “Calibration and accuracy of manipulation robot models—an overview,” *Mechanism and Machine Theory*, vol. 29, no. 3, pp. 479–500, 1994.
- [6] J. M. S. Motta and R. S. McMaster, “Modeling, optimizing and simulating robot calibration with accuracy improvement,” *Journal of the Brazilian Society of Mechanical Sciences*, vol. 21, no. 3, pp. 384–401, 1999.
- [7] Y. Liu, J. Wu, L. Wang, and J. Wang, “Parameter identification algorithm of kinematic calibration in parallel manipulators,” *Advances in Mechanical Engineering*, vol. 8, no. 9, p. 1687814016667908, 2016.
- [8] B. A. N. Campos, “Simulador para modelagem e calibração de robôs industriais,” *Dissertação de Mestrado em Sistemas Mecatrônicos, Publicação ENM. DM-09 A/06, Departamento de Engenharia Mecânica, Universidade de Brasília, Brasília, DF, 153p*, 2006.
- [9] W. Wei, W. Lei, and Y. Chao, “Design of a two-step calibration method of kinematic parameters for serial robots,” *Chinese Journal of Mechanical Engineering*, vol. 30, no. 2, pp. 438–448, 2017.
- [10] R. Bernard and S. Albright, *Robot calibration*, 1st ed. Springer Science & Business Media, 1993.

- [11] K. Schröer, S. L. Albright, and M. Grethlein, “Complete, minimal and model-continuous kinematic models for robot calibration,” *Robotics and Computer-Integrated Manufacturing*, vol. 13, no. 1, pp. 73–85, 1997.
- [12] J. M. S. Motta, C. H. Llanos-Quintero, and R. Coral Sampaio, “Inverse kinematics and model calibration optimization of a five-dof robot for repairing the surface profiles of hydraulic turbine blades,” *International Journal of Advanced Robotic Systems*, vol. 13, no. 3, p. 114, 2016.
- [13] H. Wang, X. Lu, Z. Hu, and Y. Li, “A vision-based fully-automatic calibration method for hand-eye serial robot,” *Industrial Robot: An International Journal*, vol. 42, no. 1, pp. 64–73, 2015.
- [14] K. Schröer, “Theory of kinematic modelling and numerical procedures for robot calibration,” *Robot Calibration*, vol. 157196, 1993.
- [15] J. M. S. Motta, G. C. De Carvalho, and R. McMaster, “Robot calibration using a 3d vision-based measurement system with a single camera,” *Robotics and Computer-Integrated Manufacturing*, vol. 17, no. 6, pp. 487–497, 2001.
- [16] M. Placzek and L. Piszczek, “Testing of an industrial robot’s accuracy and repeatability in off and online environment,” *EKSPLLOATACJA I NIEZAWODNOSC-MAINTENANCE AND RELIABILITY*, vol. 20, no. 3, pp. 455–464, 2018.
- [17] M. R. Driels and U. S. Pathre, “Vision-based automatic theodolite for robot calibration,” *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3, pp. 351–360, 1991.
- [18] G. Gao, G. Sun, J. Na, Y. Guo, and X. Wu, “Structural parameter identification for 6 dof industrial robots,” *Mechanical Systems and Signal Processing*, vol. 113, pp. 145–155, 2018.
- [19] K. Young and C. G. Pickin, “Accuracy assessment of the modern industrial robot,” *Industrial Robot: An International Journal*, vol. 27, no. 6, pp. 427–436, 2000.
- [20] J. J. Craig, *Introduction to robotics: mechanics and control*, 3rd ed. Pearson Education India, 2009.
- [21] H. Zhuang and Z. S. Roth, *Camera-aided robot calibration*. CRC press, 1996.
- [22] G. Duelen and K. Schröer, “Robot calibration—method and results,” *Robotics and Computer-Integrated Manufacturing*, vol. 8, no. 4, pp. 223–231, 1991.
- [23] Z. Roth, B. Mooring, and B. Ravani, “An overview of robot calibration,” *IEEE Journal on Robotics and Automation*, vol. 3, no. 5, pp. 377–385, 1987.

- [24] J. M. S. T. d. Motta, “An investigation of singularities in robot kinematic chains aiming at building robot calibration models for off-line programming,” *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol. 27, no. 2, pp. 200–204, 2005.
- [25] J. Motta, “Optimised robot calibration using a vision-based measurement system with a single camera,” 1999.
- [26] A. C. Majarena, J. Santolaria, D. Samper, and J. J. Aguilar, “An overview of kinematic and calibration models using internal/external sensors or constraints to improve the behavior of spatial parallel mechanisms,” *Sensors*, vol. 10, no. 11, pp. 10 256–10 297, 2010.
- [27] M. W. Spong, S. Hutchinson, M. Vidyasagar *et al.*, *Robot modeling and control*, 1st ed., 2006.
- [28] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: modelling, planning and control*. Springer Science & Business Media, 2010.
- [29] R. S. Hartenberg and J. Denavit, “A kinematic notation for lower pair mechanisms based on matrices,” *Journal of applied mechanics*, vol. 77, no. 2, pp. 215–221, 1955.
- [30] R. N. Jazar, *Theory of applied robotics: kinematics, dynamics, and control*. Springer Science & Business Media, 2010.
- [31] S. Hayati and M. Mirmirani, “Improving the absolute positioning accuracy of robot manipulators,” *Journal of Robotic Systems*, vol. 2, no. 4, pp. 397–413, 1985.
- [32] W. Veitschegger and C.-H. Wu, “Robot accuracy analysis based on kinematics,” *IEEE Journal on Robotics and Automation*, vol. 2, no. 3, pp. 171–179, 1986.
- [33] H. Zhuang, Z. S. Roth, and F. Hamano, “A complete and parametrically continuous kinematic model for robot manipulators,” *IEEE Transactions on Robotics and Automation*, vol. 8, no. 4, pp. 451–463, 1992.
- [34] H. Zhuang, L. K. Wang, and Z. S. Roth, “Error-model-based robot calibration using a modified cpc model,” *Robotics and computer-integrated manufacturing*, vol. 10, no. 4, pp. 287–299, 1993.
- [35] U. Pathre and M. Driels, “Simulation experiments in parameter identification for robot calibration,” *The International Journal of Advanced Manufacturing Technology*, vol. 5, no. 1, pp. 13–33, 1990.
- [36] C. L. Lawson and R. J. Hanson, *Solving least squares problems*. Siam, 1995, vol. 15.
- [37] K. Levenberg, “A method for the solution of certain non-linear problems in least squares,” *Quarterly of applied mathematics*, vol. 2, no. 2, pp. 164–168, 1944.

- [38] C. W. Wampler, A. Morgan, and A. Sommese, "Numerical continuation methods for solving polynomial systems arising in kinematics," *Journal of mechanical design*, vol. 112, no. 1, pp. 59–68, 1990.
- [39] M. Agheli and M. Nategh, "Identifying the kinematic parameters of hexapod machine tool," *World Academy of Science, Engineering and Technology*, vol. 52, pp. 380–385, 2009.
- [40] S. Bai and M. Y. Teo, "Kinematic calibration and pose measurement of a medical parallel manipulator by optical position sensors," *Journal of Robotic Systems*, vol. 20, no. 4, pp. 201–209, 2003.
- [41] S. Besnard and W. Khalil, "Identifiable parameters for parallel robots kinematic calibration," in *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)*, vol. 3. IEEE, 2001, pp. 2859–2866.
- [42] J. Ryu and A. Rauf, "A new method for fully autonomous calibration of parallel manipulators using a constraint link," in *2001 IEEE/ASME International Conference on Advanced Intelligent Mechatronics. Proceedings (Cat. No. 01TH8556)*, vol. 1. IEEE, 2001, pp. 141–146.
- [43] A. Rauf and J. Ryu, "Fully autonomous calibration of parallel manipulators by imposing position constraint," in *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)*, vol. 3. IEEE, 2001, pp. 2389–2394.
- [44] P. Renaud, N. Andreff, J.-M. Lavest, and M. Dhome, "Simplifying the kinematic calibration of parallel mechanisms using vision-based metrology," *IEEE Transactions on robotics*, vol. 22, no. 1, pp. 12–22, 2006.
- [45] T. Huang, D. G. Chetwynd, D. J. Whitehouse, and J. Wang, "A general and novel approach for parameter identification of 6-dof parallel kinematic machines," *Mechanism and machine theory*, vol. 40, no. 2, pp. 219–239, 2005.
- [46] A. Goswami, A. Quaid, and M. Peshkin, "Complete parameter identification of a robot from partial pose information," in *[1993] Proceedings IEEE International Conference on Robotics and Automation*. IEEE, 1993, pp. 168–173.
- [47] R. Horaud and F. Dornaika, "Hand-eye calibration," *The international journal of robotics research*, vol. 14, no. 3, pp. 195–210, 1995.
- [48] H. Zhuang, Z. S. Roth, and R. Sudhakar, "Simultaneous robot/world and tool/flange calibration by solving homogeneous transformation equations of the form $ax = yb$," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 4, pp. 549–554, 1994.
- [49] F. Dornaika and R. Horaud, "Simultaneous robot-world and hand-eye calibration," *IEEE transactions on Robotics and Automation*, vol. 14, no. 4, pp. 617–622, 1998.

- [50] K. Daniilidis, “Hand-eye calibration using dual quaternions,” *The International Journal of Robotics Research*, vol. 18, no. 3, pp. 286–298, 1999.
- [51] A. Malti and J. P. Barreto, “Robust hand-eye calibration for computer aided medical endoscopy,” in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 5543–5549.
- [52] R. LIMA and R. SAMPAIO, “Identificação de parâmetros pelo método dos mínimos quadrados não linear,” *Relatório final da Iniciação Científica. Rio de Janeiro*, 2009.
- [53] J. A. de França, M. B. d. M. França, M. H. Koyama, and T. P. da Silva, “Uma implementação do algoritmo levenberg-marquardt dividido para aplicações em visão computacional,” *Semina: Ciências Exatas e Tecnológicas*, vol. 30, no. 1, p. 51, 2009.
- [54] D. W. Marquardt, “An algorithm for least-squares estimation of nonlinear parameters,” *Journal of the society for Industrial and Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.
- [55] M. C. C. Cunha, *Métodos Numéricos*, 2nd ed. Campinas - SP: Editora da Unicamp, 2000.
- [56] E. P. Vasconcelos Filho, L. Ginani, J. M. S. T. Motta, G. C. D. Carvalho, C. Llanos, and W. V. D. Britto, “Desenvolvimento do modelo cinemático de calibração de um robô de geometria esférica para soldagem gmaw posicional multicamadas,” in *[2009] Congresso Brasileiro de Engenharia de Fabricação*. ABCM, 2009.
- [57] B. C. Jiang, J. Black, and R. Duraisamy, “A review of recent developments in robot metrology,” *Journal of Manufacturing Systems*, vol. 7, no. 4, pp. 339–357, 1988.
- [58] X. Jia-Qing, X. Sun-Han *et al.*, “Review on kinematics calibration technology of serial robots [j],” *International Journal of Precision Engineering and Manufacturing*, vol. 15, no. 8, pp. 1759–1774, 2014.
- [59] J.-M. Renders, E. Rossignol, M. Becquet, and R. Hanus, “Kinematic calibration and geometrical parameter identification for robots,” *IEEE Transactions on robotics and automation*, vol. 7, no. 6, pp. 721–732, 1991.
- [60] M. Ishii, S. Sakane, M. Kakikura, and Y. Mikami, “Kinematic calibration of a robot manipulator for integrated robot systems,” in *Proc. 20th ISIR*, 1989, pp. 993–1000.
- [61] B. W. Mooring and S. Padavala, “The effect of kinematic model complexity on manipulator accuracy,” in *Proceedings, 1989 International Conference on Robotics and Automation*. IEEE, 1989, pp. 593–598.
- [62] N. Vira and T. Estler, “Use of a contact-type measurement device to detect robots’ hand positions,” *ISA Transactions*, vol. 29, no. 4, pp. 21–40, 1990.

- [63] J.-H. Borm and C.-H. Meng, "Determination of optimal measurement configurations for robot calibration based on observability measure," *The International Journal of Robotics Research*, vol. 10, no. 1, pp. 51–63, 1991.
- [64] J. Caenen and J. Angue, "Identification of geometric and nongeometric parameters of robots," in *Proceedings., IEEE International Conference on Robotics and Automation*. IEEE, 1990, pp. 1032–1037.
- [65] R. P. Judd and A. B. Knasinski, "A technique to calibrate industrial robots with experimental verification," *IEEE Transactions on robotics and automation*, vol. 6, no. 1, pp. 20–30, 1990.
- [66] J. F. Jarvis, "Calibration of theodolites," in *Proceedings. 1988 IEEE International Conference on Robotics and Automation*. IEEE, 1988, pp. 952–954.
- [67] C. H. An, C. G. Atkeson, and J. Hollerbach, "Model-based control of a robot manipulator, 1988," *MIT Press, Cambridge, Massachusetts*), vol. 249, no. 250, pp. 9–18.
- [68] M. Gatenholm, "Robot calibration and performance specification," in *Proc. 19th ISIR*, 1988, pp. 1359–1367.
- [69] J. Ziegert and P. Datsoris, "Basic considerations for robot calibration," in *Proceedings. 1988 IEEE International Conference on Robotics and Automation*. IEEE, 1988, pp. 932–938.
- [70] D.-H. Kim, K. Cook, and J.-H. Oh, "Identification and compensation of a robot kinematic parameter for positioning accuracy improvement," *Robotica*, vol. 9, no. 1, pp. 99–105, 1991.
- [71] K. Lau, R. Hocken, and W. Haight, "Automatic laser tracking interferometer system for robot metrology," *Precision Engineering*, vol. 8, no. 1, pp. 3–8, 1986.
- [72] J. Chen and L.-M. Chao, "Positioning error analysis for robot manipulators with all rotary joints," *IEEE Journal on Robotics and Automation*, vol. 3, no. 6, pp. 539–545, 1987.
- [73] L. J. Everett and T. W. Ives, "A sensor used for measurements in the calibration of production robots," *IEEE transactions on robotics and automation*, vol. 12, no. 1, pp. 121–125, 1996.
- [74] D. Wang, Y. Bai, and J. Zhao, "Robot manipulator calibration using neural network and a camera-based measurement system," *Transactions of the Institute of Measurement and Control*, vol. 34, no. 1, pp. 105–121, 2012.
- [75] T. Sun, Y. Zhai, Y. Song, and J. Zhang, "Kinematic calibration of a 3-dof rotational parallel manipulator using laser tracker," *Robotics and Computer-Integrated Manufacturing*, vol. 41, pp. 78–91, 2016.
- [76] X. Zhang, Y. Song, Y. Yang, and H. Pan, "Stereo vision based autonomous robot calibration," *Robotics and Autonomous Systems*, vol. 93, pp. 43–51, 2017.

- [77] A. Filion, A. Joubair, A. S. Tahan, and I. A. Bonev, “Robot calibration using a portable photogrammetry system,” *Robotics and Computer-Integrated Manufacturing*, vol. 49, pp. 77–87, 2018.
- [78] Y. C. Shiu and S. Ahmad, “Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form $ax = xb$,” *IEEE Transactions on Robotics and Automation*, vol. 5, no. 1, pp. 16–29, 1989.
- [79] R. Y. Tsai and R. K. Lenz, “A new technique for fully autonomous and efficient 3d robotics hand/eye calibration,” *IEEE Transactions on robotics and automation*, vol. 5, no. 3, pp. 345–358, 1989.
- [80] J. Heller, D. Henrion, and T. Pajdla, “Hand-eye and robot-world calibration by global polynomial optimization,” in *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2014, pp. 3157–3164.
- [81] L. Wu, J. Wang, L. Qi, K. Wu, H. Ren, and M. Q.-H. Meng, “Simultaneous hand-eye, tool-flange, and robot-robot calibration for comanipulation by solving the $AXB = YCZ$ problem,” *IEEE Transactions on robotics*, vol. 32, no. 2, pp. 413–428, 2016.
- [82] H. Pan, N. L. Wang, and Y. S. Qin, “A closed-form solution to eye-to-hand calibration towards visual grasping,” *Industrial Robot: An International Journal*, vol. 41, no. 6, pp. 567–574, 2014.
- [83] J.-S. Hu and Y.-J. Chang, “Calibration of an eye-to-hand system using a laser pointer on hand and planar constraints,” in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 982–987.
- [84] —, “Automatic calibration of hand–eye–workspace and camera using hand-mounted line laser,” *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 6, pp. 1778–1786, 2012.
- [85] F. B. Carlson, R. Johansson, and A. Robertsson, “Six dof eye-to-hand calibration from 2d measurements using planar constraints,” in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 3628–3632.
- [86] Z. Goh, “Calibration of spatial relationships between multiple robots and sensors,” Ph.D. dissertation, Johns Hopkins University, 2016.
- [87] S. Yan, S.-K. Ong, and A. Y. Nee, “Registration of a hybrid robot using the degradation-kronecker method and a purely nonlinear method,” *Robotica*, vol. 34, no. 12, pp. 2729–2740, 2016.
- [88] A. C. Gil, *Como elaborar projetos de pesquisa*, 5th ed. São Paulo: Atlas, 2010.

- [89] A. Robotics, “Product manual irb 140,” *ABB Robotics Products AB publication*, no. M2000, pp. 7564–1, 2003.
- [90] G. Stark, E. Benz, and M. Hüttendorfer, “Calibration experiences in industry,” *Internal summary of EPSRIT II CAR (5220)*, KUKA Schweißanlagen+ Roboter GmbH, 1992.