

ESTE EXEMPLAR CORRESPONDE A REDAÇÃO FINAL
DA TESE DEFENDIDA POR RENATO DE
CAMPOS E APROVADA PELA
COMISSÃO JULGADORA EM 20/09/93.

João Maurício Rosário
ORIENTADOR

Universidade Estadual de Campinas
Faculdade de Engenharia Mecânica
Departamento de Projeto Mecânico

TESE DE MESTRADO

IMPLEMENTAÇÃO DE UM ALGORITMO PARA ESTIMATIVA DE PARÂMETROS
DO MODELO CINEMÁTICO DE ROBÔS

Aluno: Renato de Campos

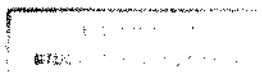
Renato de Campos

Orientador: João Maurício Rosário

João Maurício Rosário

30/93

Trabalho apresentado à comissão de
Pós-graduação da Faculdade de
Engenharia Mecânica como parte dos
requisitos para obtenção do título
de Mestre em Engenharia Mecânica.



UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA

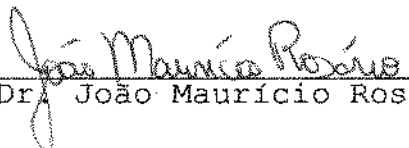
TESE DE MESTRADO

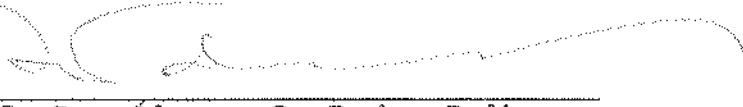
IMPLEMENTAÇÃO DE UM ALGORITMO PARA ESTIMATIVA DE PARÂMETROS
DO MODELO CINEMÁTICO DE ROBÔS

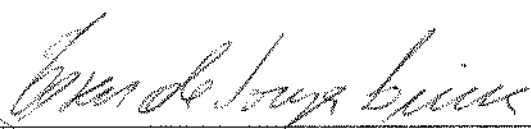
Autor: Renato de Campos

Orientador: Prof. Dr. João Maurício Rosário

Aprovado por:


Prof. Dr. João Maurício Rosário, Presidente


Prof. Dr. Alvaro G. Badan Palhares


Dr. César de Souza Lima

Campinas, 20 de setembro de 1993

CONTEÚDO

INTRODUÇÃO	1
1. PROGRAMAÇÃO DE UM ROBÔ E MEDIDA DE PRECISÃO DE POSICIONAMENTO ..	3
1.1 - Programação On-line e Programação Off-line	3
1.2 - Procedimentos para Calibração de um Robô	7
1.2.1. Tabulação de Erros versus Identificação de Parâmetros..	8
1.2.2. Identificação Junta por Junta versus Modelagem do Erro de Posicionamento do Elemento Terminal	9
1.2.3. Técnicas de Medidas de Posicionamento	9
1.3 - Projeto de Automação de Operações Submarinas	15
2 - MODELAGEM GEOMÉTRICA DO SISTEMA E SEU ERRO	18
2.1 - Método "Modificado" de Denavit-Hartenberg	19
2.2 - Modelagem do Erro de Posicionamento	23
2.3 - Modelagem Geométrica através de Vetores Locais	25
3 - ALGORITMO PROPOSTO PARA ESTIMATIVA DE PARÂMETROS	30
4 - VALIDAÇÃO DO ALGORITMO PROPOSTO A PARTIR DE SIMULAÇÕES	35
4.1 - Metodologia	35
4.2 - Análise da Influência de Variáveis do Procedimento.....	41
4.3 - Simulação do Experimento Prático	46

5 - EXPERIMENTOS E RESULTADOS	50
5.1 - Determinação da Repetibilidade	51
5.2 - Estimativa de Parâmetros	52
5.2.1. Metodologia	52
5.2.2. Resultados	55
5.3 - Determinação da Exatidão	59
5.4 - Conclusões	60
 CONCLUSÃO	 61
ANEXO A	63
ANEXO B	66
ANEXO C	88
ANEXO D	94
ANEXO E	100
 BIBLIOGRAFIA	 105

DEDICO ESTA DISSERTAÇÃO AOS MEUS PAIS

AGRADECIMENTOS

Ao Professor João Maurício Rosário pela orientação e apoio durante o desenvolver da tese.

Ao Dr. César de Souza Lima e ao Prof. Dr. Álvaro G. Badan Palhares por aceitarem participar da banca julgadora e contribuir com a tese.

Ao pessoal do Laboratório de Automação e Robótica pela ajuda e companherismo durante os trabalhos.

Aos meus irmãos que me incentivaram na realização desta tese.

Ao Departamento de Projeto Mecânico (DPM/FEM) e ao Centro de Tecnologia pelo suporte técnico.

Ao CNPq e CENPES/Petrobrás pelo apoio financeiro.

RESUMO

Atualmente, as tarefas a serem executadas por robôs estão se tornando mais complexas. Nesse contexto encontra-se, por exemplo, a integração de robôs aos sistemas CIM (Computer Integrated Manufacturing), e a necessidade de se definir trajetórias matematicamente, via um sistema CAD (programação off-line), sendo inviável a programação por aprendizagem. Com a programação off-line, a exatidão do sistema robótico torna-se essencial para o sucesso da operação programada. Este trabalho visa o estudo e implementação de um procedimento para estimativa de parâmetros geométricos e calibração para robôs, consistindo basicamente de quatro fases:

1. Modelagem geométrica do sistema robótico e de seu erro através do jacobiano de sua matriz de posicionamento;
2. Sequência de medidas de posicionamento por um sistema externo;
3. Estimativa de parâmetros através de um algoritmo que tem como entrada as medidas geradas na segunda fase e um valor inicial arbitrário para os parâmetros, que deve convergir usando o critério de mínimos quadrados;
4. Implementação dos valores dos parâmetros estimados no software de controle e programação off-line do robô.

São realizadas simulações do algoritmo de estimativa, onde é analisado a influência de variáveis do procedimento, com relação à precisão na estimativa. É efetuado um experimento prático de estimativa dos parâmetros do robô MANUTEC r3, tendo como dispositivo de medidas, uma mesa tridimensional. É realizado um outro procedimento visando a estimativa de parâmetros do mesmo robô, porém com as medidas feitas através de uma ferramenta e peça de calibração dedicada para esta finalidade. Os resultados obtidos utilizando-se estes dois dispositivos são comparados.

São propostas duas diferentes soluções visando a correção de erros de posicionamento entre um robô e seu painel de operação em intervenções submarinas.

INTRODUÇÃO

Os sistemas robóticos estão, com o passar do tempo, mais complexos e automáticos. Consequentemente a programação por aprendizagem, não satisfaz a maioria das aplicações nas quais as características mencionadas estão presentes. Por exemplo, a utilização de robôs numa produção gerenciada por um sistema CIM, minimiza a intervenção humana na programação do robô, e assim as trajetórias passam a serem descritas matematicamente através de um sistema CAD (programação off-line). Neste tipo de programação, a precisão do robô é de extrema importância.

Este trabalho é dividido em cinco capítulos e objetiva o estudo e implementação de um algoritmo para estimativa de parâmetros para robôs, com a intenção de calibrá-lo e aumentar sua exatidão de posicionamento.

No capítulo 1 é realizado uma introdução sobre programação off-line, métodos de estimativa de parâmetros e calibração, e comentários relativos ao projeto de automação de operações submarinas entre a UNICAMP, CENPES-Petrobrás, e GKSS da Alemanha.

No capítulo 2 é descrito a modelagem geométrica de um robô genérico através do método "modificado" de Denavit-Hartenberg, usado no software para estimativa, e uma modelagem através de vetores locais de rotação e translação, utilizado num software de programação off-line. Também é descrito uma relação entre erros de posicionamento e erros nos parâmetros geométricos do robô que é usado no algoritmo.

No capítulo 3 é apresentado o algoritmo para a estimativa proposto.

No capítulo 4 o software para a estimativa é testado, estudando-se a influência de variáveis do procedimento na precisão da estimativa dos parâmetros.

E finalmente no capítulo 5 são descritos os experimentos práticos, e realizado comparações entre dois diferentes tipos de procedimento de medidas visando a estimativa de parâmetros de um robô.

CAPÍTULO 1

PROGRAMAÇÃO DE UM ROBÔ E MEDIDA DE PRECISÃO DE POSICIONAMENTO

1.1 - PROGRAMAÇÃO ON-LINE E PROGRAMAÇÃO OFF-LINE

A introdução de robôs num ambiente industrial, inicialmente, veio atender à realizações de tarefas relativamente simples . Assim na fase de programação do robô, era suficiente a aprendizagem (programação on-line) através de um sistema master-slave ou teclas de um painel de operação, movimentando-o junta por junta até o objetivo a ser atingido e armazenando em computador dedicado as informações referentes a esta posição fornecidas pelos seus sensores. Através de um algoritmo de interpolação interno ao controlador do robô gera-se sua trajetória entre duas ou mais posições. Diante desse sistema, a mais importante característica que se desejava de um robô, quanto a sua performance no posicionamento, era sua repetibilidade. A repetibilidade de um robô refere-se à sua capacidade de retornar à um mesmo posicionamento, uma vez tenha sido gravada [39].

Durante o passar dos anos, as aplicações em robótica se ampliaram, sendo que em grande parte dessas aplicações, faz-se necessária o uso da programação off-line. Na programação off-line, as trajetórias podem ser especificadas em termos de coordenadas cartesianas e angulares, através de um software de visualização gráfica e matemática (CAD), no qual é implementado o modelo geométrico do sistema (robô, ferramentas dedicadas e ambiente de atuação). Podemos citar como exemplos:

- * Realização de tarefas precisas e complexas, como montagens e realização de trajetórias descritas matematicamente. O uso da programação on-line nestes casos, apresenta o inconveniente de exigir a repetição de numerosas vezes a aprendizagem, acarretando a perda de um tempo considerável e não garantindo a qualidade da operação. Ela é facilmente vulnerável às modificações do ambiente em que se trabalha, sendo que nem sempre pode se transferir um mesmo programa de um robô para outro similar.
- * Integração de robôs aos sistemas CIM (Computer Integrated Manufacturing), onde os programas são baseados na troca de informações com um sistema CAD/CAM, e onde se prevê constantes mudanças nos tipos de produtos a serem manufaturados, característica de sistemas flexíveis de manufatura.
- * Uso de robôs em ambientes hostís ou inacessíveis, onde o modelo geométrico desse ambiente é conhecido, como em instalações nucleares e operações submarinas.

A partir do momento que se programa através de um sistema off-line, não só a repetibilidade, como um alto grau de exatidão de posicionamento (posição e orientação), torna-se essencial para o sucesso da operação. Define-se como exatidão, a capacidade do robô fazer um posicionamento numericamente preciso, em relação a um posicionamento programado, dentro do seu volume de trabalho [39].

Robôs industriais apresentam, em geral, alta repetibilidade, mas deixam a desejar em exatidão. Possuem repetibilidade por volta de 1 mm e em alguns casos, são menores de 0.1 mm. Por outro lado sua exatidão pode chegar facilmente acima de 1 cm, sendo neste caso inconveniente a programação off-line [2], [5], [7], [25], [26].

Diferentes estudos podem ser realizados visando identificar os fatores que levam à imprecisão de um sistema robótico. Supondo um sistema constituído de um robô industrial, uma mesa de montagem, suporte de peças (figura 1.1), e a simples tarefa de encaixe de uma peça A em outra B, podemos relacionar as seguintes fontes de imprecisão no modelo geométrico do sistema considerado:

i) Erros relativos ao robô:

- tolerâncias de fabricação e montagem do robô (erros de paralelismo e perpendicularismo dos eixos sucessivos);
- erros no acoplamento da ferramenta no elemento terminal;
- backlash e conhecimento inexato dos fatores de redução;
- flexões e torções, sobretudo a nível dos sistemas de transmissão, devido à influência da carga;
- efeito da temperatura (dilatação dos corpos, fluídos de acionadores hidráulicos, variação das características dos componentes, etc.).
- erros em transdutores.
- erros do controle.

ii) Erros relativos ao sistema de trabalho:

- conhecimento inexato da posição e orientação do sistema de referência do suporte da peça B (R_{SB}) em relação ao referencial do robô (R_{base});
- conhecimento inexato do referencial da peça B (R_B) em relação ao referencial do suporte (R_{SB});
- conhecimento inexato do sistema de referência da ferramenta (R_F) em relação ao referencial da peça A (R_A).

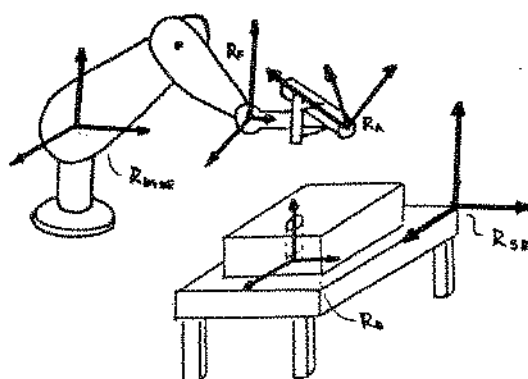


Figura 1.1 – Operação de montagem de uma peça A em outra B.

Admitindo que para a montagem ser perfeita, o referencial da peça A deve coincidir com o referencial da peça B, os erros descritos anteriormente dificultariam a realização de uma operação automatizada, programada através de um sistema off-line.

Com o objetivo de aumentar a exatidão do sistema, é proposto um método de calibração, consistindo de 4 fases:

1. Modelagem do sistema, onde deve se encontrar a melhor função que relacione os parâmetros do robô e o posicionamento do elemento terminal em relação a um sistema de coordenadas. Nesta fase, deve-se incluir no modelo, todos os parâmetros que podem influir significativamente na exatidão do posicionamento.
2. Obtenção de medidas de posicionamento, através de um dispositivo externo de medição em relação a um sistema de coordenadas arbitrário.
3. Estimativa dos parâmetros através de um algoritmo, que tem como entrada as medidas geradas na segunda fase e valores nominais dos parâmetros (arbitrário), e que faz uso do critério de mínimos quadrados (figura 1.2).
4. Correção do modelo geométrico, através da implementação dos valores dos parâmetros estimados no software de controle e de programação off-line do robô.

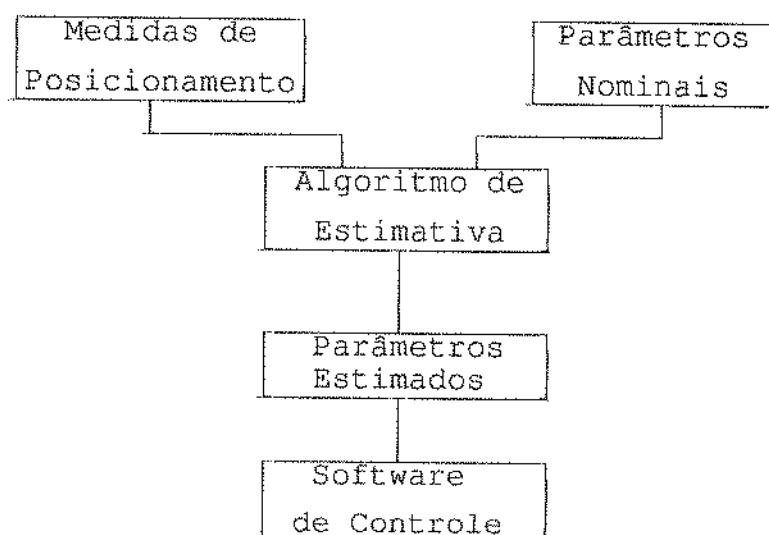


figura 1.2 - Proposta para estimativa de parâmetros e calibração.

1.2 - PROCEDIMENTOS PARA CALIBRAÇÃO DE UM ROBÔ

A literatura dedicada à análise da precisão para robôs, apresenta uma grande variedade de métodos para sua calibração (tipos de algoritmo para estimativa, modelagem, mecanismos externos de medição, etc), sendo difícil concluir qual o melhor método. Certamente a solução dependerá de cada caso, obtendo a melhor relação entre o desempenho necessário (precisão, automação, rapidez, etc), e os recursos disponíveis (equipamentos de medição, hardware, software, etc). Neste item é descrito algumas características de procedimentos usados na estimativa de parâmetros e calibração para robôs.

1.2.1 - Tabulação de Erros versus Estimativa de Parâmetros.

Um procedimento para corrigir o posicionamento de um robô pode ser realizado através da tabulação dos erros ($\bar{E}$) e posterior interpolação polinomial:

$$\bar{E} = [\Delta x, \Delta y, \Delta z]^t = \bar{F}(x, y, z) \quad (1.1)$$

Neste caso, para uma calibração precisa seriam necessárias medidas ao longo das três coordenadas (x, y e z), cobrindo todo volume de trabalho entre pequenas distâncias (3 a 5 mm por exemplo).

Pode-se também, tabular os erros, relacionando-os com deslocamentos angulares das juntas do robô. Neste caso, medidas para uma junta seriam realizadas com todas as outras paradas. Os erros para cada junta seriam interpolados, possibilitando a correção de erros de posição e também orientação, ao contrário do caso anterior:

$$\bar{E} = [\Delta x, \Delta y, \Delta z, \Delta \psi, \Delta \theta, \Delta \phi]^t = \bar{F}(\theta_1, \theta_2, \dots, \theta_n) \quad (1.2)$$

A tabulação de erros possui o inconveniente de exigir um enorme número de medidas, necessitar uma grande quantidade de memória computacional e tempo de processamento.

A estimativa de parâmetros que influenciam a exatidão de posicionamento e sua correção no controle através do modelo do robô, pode ser realizada com um menor número de medidas em relação a tabulação. Usando um rápido algoritmo para a estimativa, ela se apresenta como uma solução mais geral e propícia para a automação do procedimento.

1.2.2 - Identificação Junta por Junta versus Modelagem do Erro de Posicionamento do Elemento Terminal.

O procedimento de estimativa proposto por Stone [15], permite estimar parâmetros junta por junta. Ele realizou medidas para pequenos incrementos de rotação entre os limites angulares para a junta em análise, não se movendo as outras. Os dados eram tratados por um software, que usou o critério de mínimos quadrados para obter o plano de rotação e o centro de rotação no caso juntas rotacionais, e a linha de translação para juntas translacionais. Após, usou de critérios particulares para fixar referenciais e estimar seis parâmetros geométricos para cada junta. Ao mesmo tempo foi implementado um software para relacionar esses parâmetros com os parâmetros de Denavit-Hartenberg.

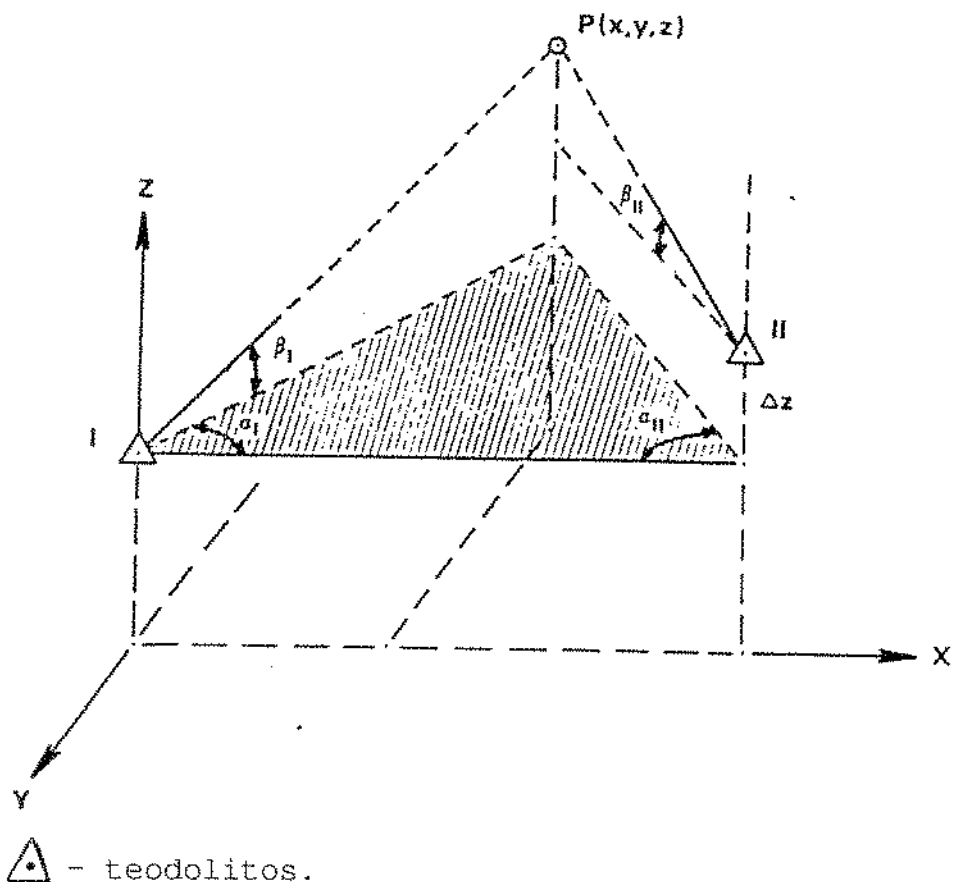
A estimativa de parâmetros baseado na modelagem do erro de posicionamento ("estimativa através do Jacobiano") [1], [5], [11], [17], [27], necessita de um menor número de medidas, estima simultaneamente todos os parâmetros, inclusive erros de posicionamento entre o referencial do robô e um referencial de operação de interesse, sendo mais rápido e automático. Este tipo de procedimento será o usado neste trabalho.

1.2.3 - Técnicas de Medidas de Posicionamento.

O instrumento de medida ideal para a calibração de um robô deve ser rápido, automático e preciso. Esse instrumento deve permitir medidas em todo o volume de trabalho e não necessitar de contato com o robô. É difícil um aparelho medir a orientação com o desempenho acima, levando a ter como solução, um mecanismo especial associado a um outro instrumento e conseqüentemente um aumento no número de medidas e processamento de informações. A seguir são descritos sucintamente algumas técnicas existentes para calibração de parâmetros de um robô.

A) Teodolitos

Um sistema de medida tridimensional pode ser montado através de dois ou mais teodolitos [2], [5], [10]. A coleta de um conjunto de medidas pode ser totalmente automatizada no caso de teodolitos eletrônicos e chegar a uma precisão de 0.1 mm. O princípio é baseado na intersecção de retas passando por um ponto no elemento terminal do robô (figura 1.3) e usando ângulos e relações trigonométricas, pode-se chegar à posição tridimensional do ponto.



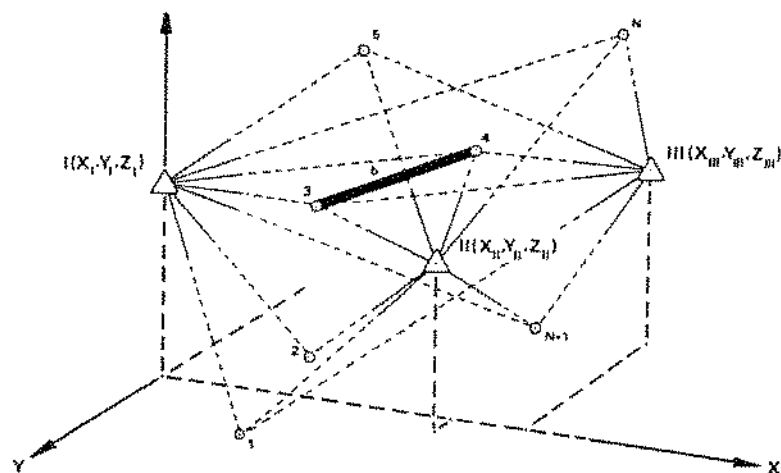
$\triangle$ - teodolitos.

$P(x,y,z)$ - posição do elemento terminal do robô.

$\alpha_I, \beta_I, \alpha_{II}, \beta_{II}$ - ângulos de visada dos teodolitos I e II.

Figura 1.3 - Intersecção de retas passando pelo elemento terminal do robô.

O sistema pode ser calibrado através de uma peça usinada com medidas precisas, como por exemplo uma barra (figura 1.4).



b - comprimento da barra.

(X_I, Y_I, Z_I) , (X_{II}, Y_{II}, Z_{II}) , $(X_{III}, Y_{III}, Z_{III})$ - posições dos teodolitos I, II, III.

N - número de visadas dos teodolitos.

Figura 1.4 - Calibração do sistema através de uma barra.

Pode ser explorada a obtenção de medidas de orientação através de três ou mais marcas de referência no elemento terminal, ao contrário de apenas uma, e introduzir as medidas em um programa que através de relações trigonométricas, forneçam a sua orientação.

B) Dispositivos utilizando Laser.

O procedimento de utilização do laser para fins de medidas de posicionamento pode ser muito parecido com o do teodolito, sendo que já existem teodolitos a laser. Além da grande precisão, ela pode ser uma solução mais adequada para determinadas situações. Nesta tese ela será explorada como uma solução para um dos problemas em estudo que será apresentado no item 1.3 deste capítulo.

C) Mesa Tridimensional

A mesa tridimensional de medidas possui o inconveniente de medir um conjunto de posições em apenas uma parte do volume de trabalho e necessitar de contato entre o elemento terminal do robô e aparelho de medida [26]. Esse aparelho pode fornecer precisão em torno de 0.1 mm. Durante as realizações experimentais, será utilizado um dispositivo desse tipo. A aquisição e tratamento das medidas pode ser automatizada como na figura 1.5 (não é o caso do aparelho usado neste trabalho).

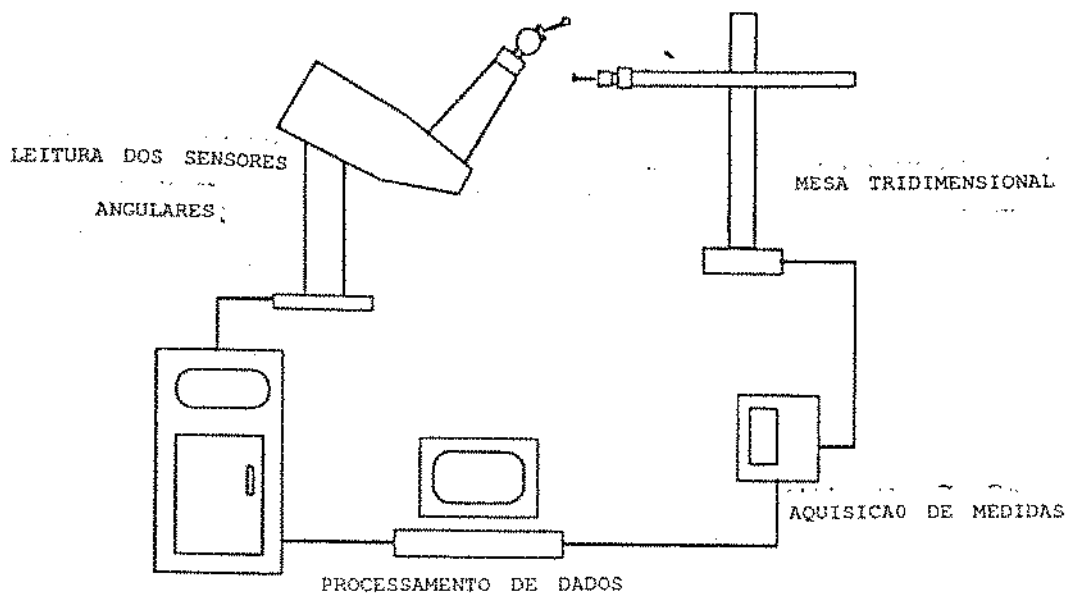


Figura 1.5 - Medidas através de uma mesa tridimensional com aquisição automática

D) Peça de Calibração

Medidas de posição podem ser realizadas através de uma ferramenta e uma peça de calibração usinadas de modo a obter uma boa precisão geométrica [7], [11]. Ao fazer a extremidade da ferramenta atingir determinados pontos da peça, pode-se conhecer exatamente a posição do elemento terminal em relação ao referencial da peça (figura 5.2). Neste trabalho, um exemplo desse método é testado e comparado com a mesa tridimensional (capítulo 5). O sistema pode ter sensores que permitam uma melhor viabilização deste procedimento.

E) Plataforma Dinamométrica

Uma plataforma dinamométrica pode ser construída através de uma placa onde são colocados 4 sensores de esforços em C_1 , C_2 , C_3 e C_4 (figura 1.6). Ao se aplicar uma força F pelo robô, vertical no ponto C , as leituras de F_1 , F_2 , F_3 e F_4 permitem determinar as coordenadas x e y do ponto de aplicação no referencial da placa:

$$x = \sum_{i=1}^4 F_i \cdot x_i / F \quad (1.3)$$

$$y = \sum_{i=1}^4 F_i \cdot y_i / F \quad (1.4)$$

A plataforma permite a calibração em apenas duas dimensões e deve se limitar a força normal para evitar deformações no robô.

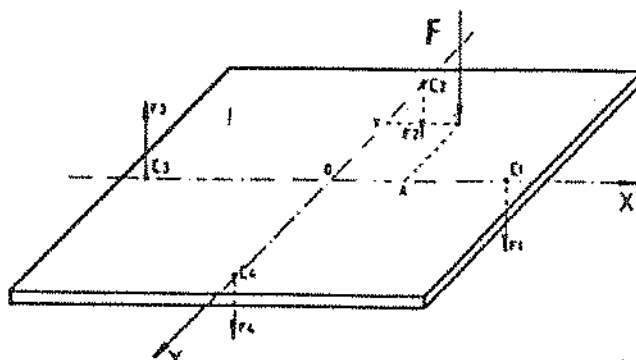


Figura 1.6 - Plataforma Dinamométrica.

F) Câmeras CCD.

Dentre os vários métodos encontrados na literatura pesquisada, aquele que mais se aproxima do método ideal, é um sistema (fornecido comercialmente) que usa 2 câmeras CCD (Coupled Charge Device) de grande precisão e um ponto luminoso na garra do robô (figura 1.7). Neste sistema não é necessário nenhum tipo de contato, e possibilita tanto uma análise estática como dinâmica de movimentos até 10 m/s. Através de uma alta e precisa frequência de aquisição de dados, detecta mudanças de posições de 25 μm (dados de fabricante). O tratamento desses dados através de um software fornece, automaticamente, resultados referentes a sua repetibilidade e exatidão, e gráficos relativos à posição e velocidade da trajetória, em um volume de trabalho de aproximadamente um metro cúbico.

O sistema mede posição com um ponto de reflexão no elemento terminal e também pode medir orientação com mais alguns pontos. É necessário um processamento adicional desses pontos. Para se obter um sistema de referência para as medidas, deve-se realizar durante o set-up, uma aquisição pelo sistema de câmeras de pontos luminosos fixos (por exemplo na base do robô), que irá definir esse sistema de referência para as medidas.

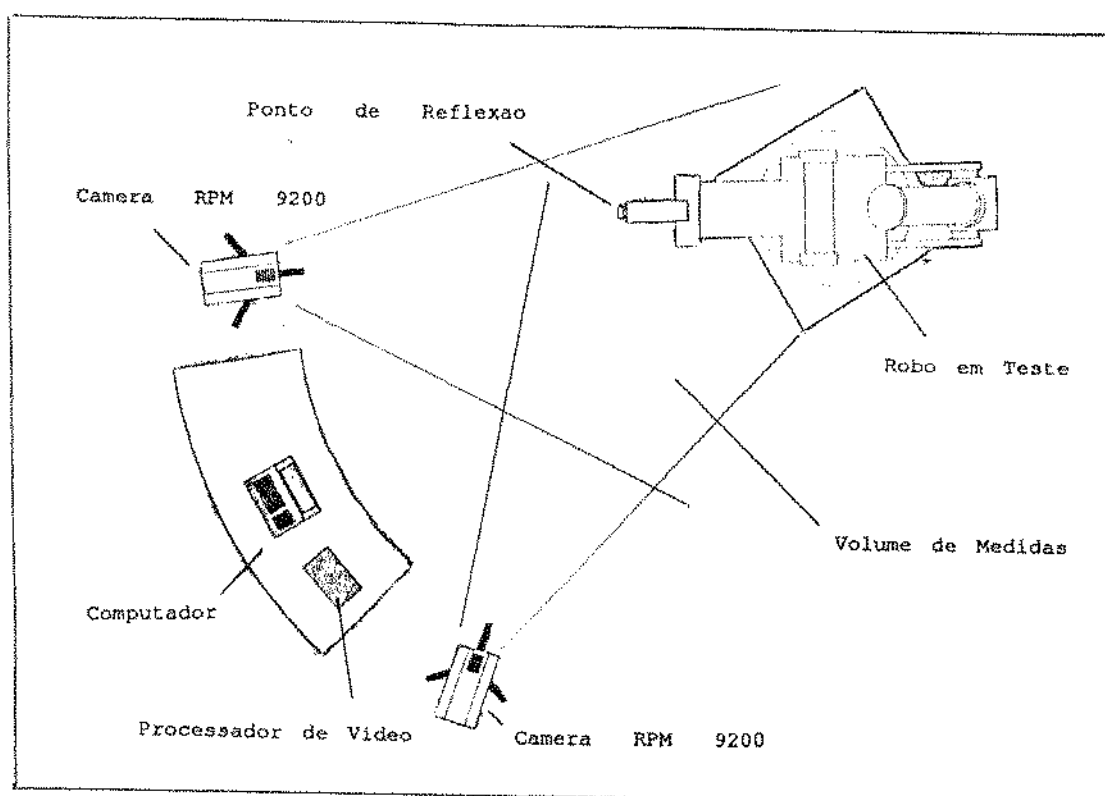


Figura 1.7 - Medidas usando câmeras CCD.

1.3 - PROJETO DE AUTOMAÇÃO DE OPERAÇÕES SUBMARINAS

Em particular, este trabalho faz parte de um projeto maior, sendo desenvolvido pela UNICAMP, GKSS (Instituto Tecnológico de Geesthacht - Alemanha) e a CENPES-Petrobrás, cujo objetivo final consiste na automação de operações submarinas, utilizando robôs industriais [30], [31], [32], [33].

Para executar essas operações, o robô, estando fixo em uma base móvel, deverá se posicionar através dessa base, apoiando-se sobre trilhos, diante de um painel o qual vai operar (figura 1.8 e 1.9). Apesar de controlado por sensores, prevê-se erros de posição e orientação da base em que é fixado o robô, em relação ao painel.

Existem dispositivos projetados para tolerar algum erro. Porém esses erros podem ser suficientes para inviabilizar a operação automatizada.

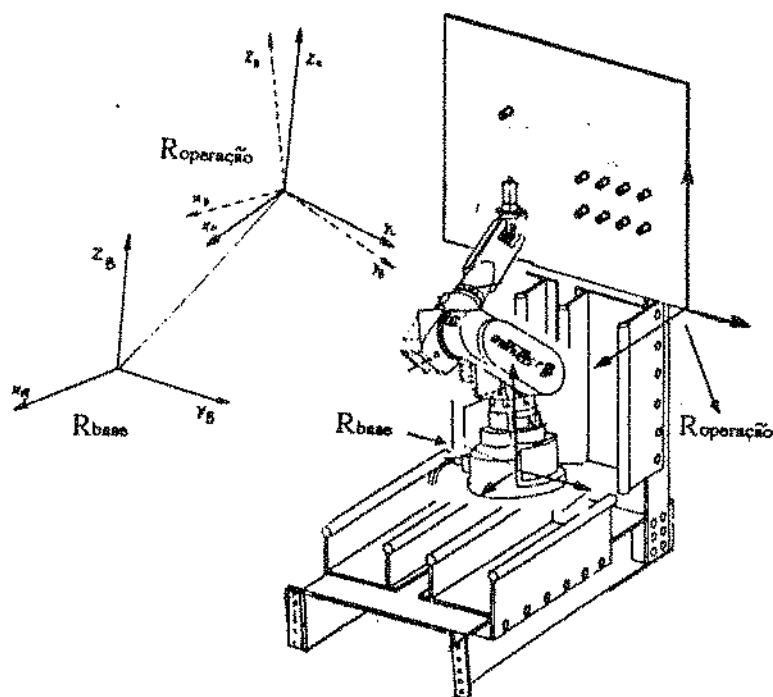


Figura 1.8 - Posicionamento do base móvel em relação ao painel.

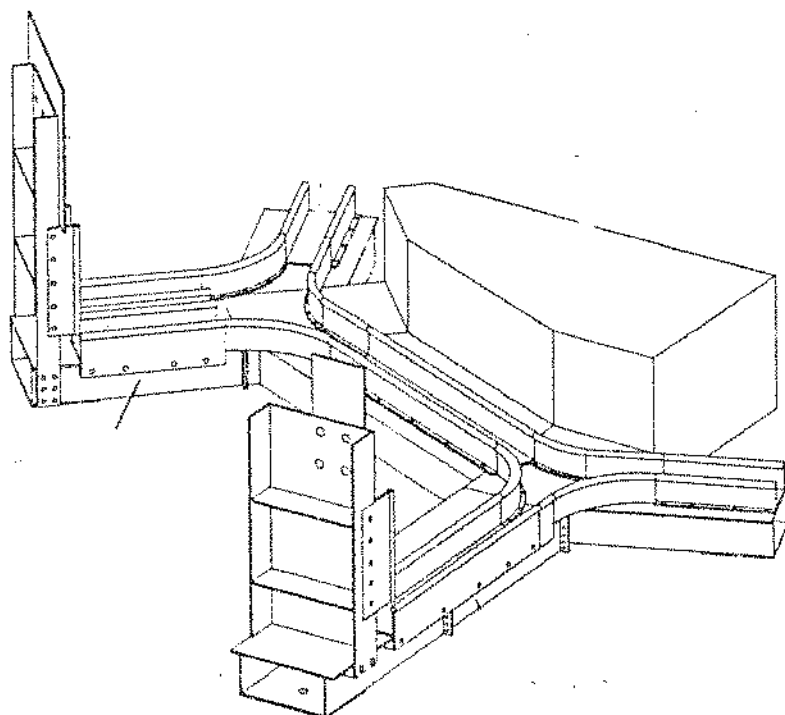


Figura 1.9 - Mock-up e trilhos de deslocamento do robô entre painéis

A estimativa desses erros é necessária para a correção das trajetórias geradas off-line através de uma programação via CAD, cujo modelo geométrico supõe posicionamento perfeito da base móvel.

Neste projeto será testado o desempenho do robô industrial MANUTEC r3 e do manipulador submarino KRAFT. Para o manipulador KRAFT, é desenvolvido um hardware e software de controle e programação (modelo direto, modelo inverso, etc.) [12], [13], que irá possibilitar obter as características de um robô e permitir a utilização de um software de visualização gráfica para especificar suas trajetórias em função de coordenadas cartesianas e angulares (programação off-line), e assim possibilitar, através do modelo geométrico do ambiente, implementar testes de colisão para pontos específicos do sistema (robô, base móvel e painel de operação).

Um dos critérios de desempenho a se analisar, é a precisão com a qual os robôs poderão executar as operações. Nesse sentido, serão realizados experimentos, visando a calibração, análise da repetibilidade e exatidão para os robôs, além da implementação de um software e mecanismo para estimativa dos erros de posicionamento entre robô e painel de operação anteriormente descritos. A estimativa desses erros possibilita a correção automática de trajetórias geradas off-line pelo software de visualização gráfica (CAD) [35], [36].

CAPÍTULO 2

MODELAGEM GEOMÉTRICA DO SISTEMA E SEU ERRO

Conforme descrito no capítulo 1 deste trabalho, uma modelagem na qual busca-se uma relação entre parâmetros e o posicionamento do robô, é o primeiro passo para a calibração do sistema. Já foram listados vários fatores que influenciam no erro do sistema e que devem ser considerados no modelo para a calibração ser a mais perfeita possível. Porém, neste trabalho vamos nos preocupar apenas com parâmetros geométricos, e que muitas vezes representam grande parte dos erros de posição e orientação de um robô industrial.

Modelagens envolvendo parâmetros não geométricos, como desalinhamento e folgas em redutores, flexões e torções em juntas podem ser encontrados nas referências [2], [5], [7] e [25].

Neste capítulo será apresentado a modelagem geométrica para um robô genérico através do método "modificado" de Denavit-Hartenberg, que será usado no programa para estimativa de parâmetros por ser compacto, rápido e de fácil implementação computacional. A partir dessa modelagem, é apresentada uma relação entre erros de posicionamento e erros nos parâmetros através do Jacobiano da matriz de transformação de coordenadas representando a posição e orientação do robô. No item 2.3, é descrito a modelagem usada no software de programação off-line.

2.1 - MÉTODO "MODIFICADO" DE DENAVIT-HARTENBERG

Para obter a equação que fornece o posicionamento do elemento terminal do robô em relação ao sistema de referência fixo à base do robô, parte-se da sistemática de Denavit-Hartenberg (DH) [19], [20], o [21] qual associa apenas 4 parâmetros, θ_i , a_i , d_i e α_i para descrever uma matriz de transformação homogênea A_{i+1}^i 4x4, entre referenciais de duas juntas robóticas sucessivas, $i + 1$ e i (figura 2.1). Essa matriz é descrita por 4 sucessivas transformações homogêneas:

$$A_{i+1}^i = \text{Rot}(z, \theta_i) * \text{Trans}(z, d_i) * \text{Trans}(x, a_i) * \text{Rot}(x, \alpha_i). \quad (2.1)$$

$$A_{i+1}^i(\bar{p}_i) = \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i & a_i c\theta_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.2)$$

$$\text{com } \bar{p}_i = \begin{bmatrix} \theta_i \\ d_i \\ a_i \\ \alpha_i \end{bmatrix}$$

onde s = função seno e c = função cosseno

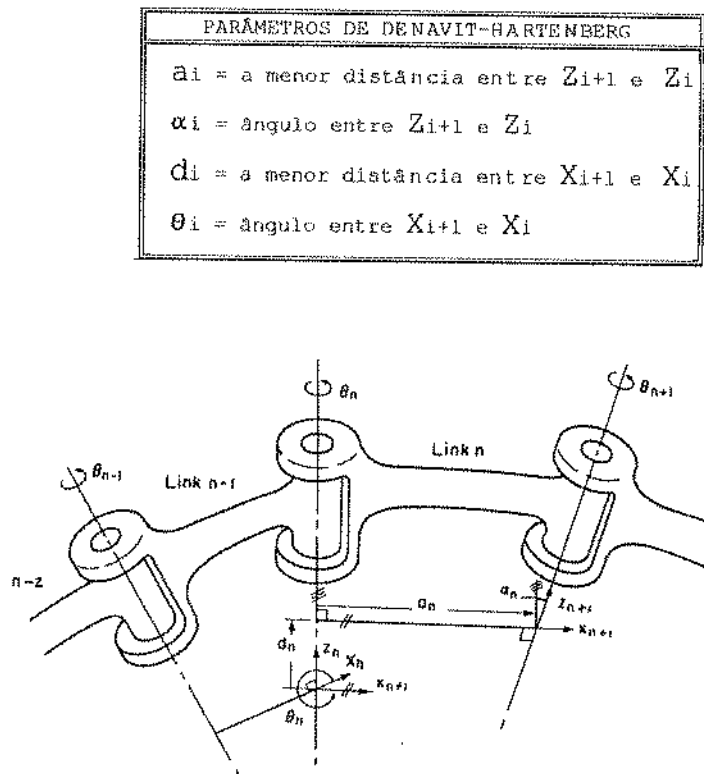


Figura 2.1 - Notação de Denavit-Hartenberg.

Para o propósito de estimar parâmetros, a notação de DH não se mostra adequada para descrição da relação entre juntas rotacionais sucessivas paralelas. É que neste caso a hipótese utilizada no algoritmo de estimativa, de que para pequenos erros de posicionamento pode-se associar pequenos erros nos parâmetros nem pode ser considerada.

Hayati [4] propôs uma modificação para DH, possibilitando descrever possíveis desvios angulares para duas juntas nominalmente paralelas. Ele propõe a multiplicação de mais uma matriz de rotação ($\text{Rot}(y, \beta_i)$) para a obtenção da matriz homogênea A_{i+1}^i .

Assim, para juntas rotacionais paralelas:

$$A_{i+1}^i = \text{Rot}(z, \theta_i) * \text{Trans}(z, d_i) * \text{Trans}(x, a_i) * \text{Rot}(x, \alpha_i) * \text{Rot}(y, \beta_i) \quad (2.3)$$

β_i - ângulo entre Y_{i+1} e Y_i , para 2 juntas paralelas.

$$A_{i+1}^i(\bar{\mathbf{p}}_i) = \begin{bmatrix} c\theta_i c\beta_i - s\theta_i s\alpha_i s\beta_i & -s\theta_i c\alpha_i & c\theta_i s\beta_i + s\theta_i s\alpha_i c\beta_i & a_i c\theta_i \\ s\theta_i c\beta_i + c\theta_i s\alpha_i s\beta_i & c\theta_i c\alpha_i & s\theta_i s\beta_i - c\theta_i s\alpha_i c\beta_i & a_i s\theta_i \\ -c\alpha_i s\beta_i & s\alpha_i & c\alpha_i c\beta_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

com $\bar{\mathbf{p}}_i = \begin{bmatrix} \theta_i \\ d_i \\ a_i \\ \alpha_i \\ \beta_i \end{bmatrix} \quad (2.4)$

Designando para o referencial fixo na base do robô, o índice "1", para o referencial fixo no elemento terminal, o índice "N+1" (N é o número de graus de liberdade do robô), a matriz de transformação homogênea que descreve a posição e orientação do elemento terminal em relação ao referencial da base é:

$$T_{N+1}^1(\bar{\mathbf{k}}') = A_2^1 A_3^2 \dots A_{N+1}^N \quad (2.5)$$

$\bar{\mathbf{k}}'$ - vetor de parâmetros : $\bar{\mathbf{k}}' = [\bar{\mathbf{p}}_1^t \ \bar{\mathbf{p}}_2^t \ \dots \ \bar{\mathbf{p}}_N^t]^t$

$$T_{N+1}^1 = \begin{vmatrix} n_x & s_x & a_x & : & p_x \\ n_y & s_y & a_y & : & p_y \\ n_z & s_z & a_z & : & p_z \\ \hline 0 & 0 & 0 & : & 1 \end{vmatrix} = \begin{vmatrix} & & & : & \\ & O_{xyz} & & : & \bar{P} \\ & & & : & \\ \hline 0 & 0 & 0 & : & 1 \end{vmatrix} \quad (2.6)$$

O_{xyz} - matriz 3x3 de orientação composto pelos vetores ortonormais $\bar{n}$ (normal), $\bar{s}$ (slide), $\bar{a}$ (approach).

$\bar{P} = [p_x \ p_y \ p_z]^t$ - vetor posição do elemento terminal em relação a base.

Com o objetivo de expressar o posicionamento em relação a um referencial de operação R_0 , é feita a multiplicação de T_{N+1}^1 por outra matriz de transformação A_1^0 , que descreve a posição (x_0, y_0, z_0) e orientação $(\bar{n}_0, \bar{s}_0, \bar{a}_0)$ do referencial da base em relação ao referencial de operação. A matriz de orientação composta pelos vetores ortonormais $\bar{n}_0, \bar{s}_0$ e $\bar{a}_0$, pode ser descrita em função dos ângulos de Euler ou dos ângulos Roll, Pitch e Yaw $(\phi_0, \theta_0, \psi_0)$ [6], [14], [21]. Pode-se escrever então:

$$A_1^0(\bar{P}_0) = \begin{vmatrix} n_{0x} & s_{0x} & a_{0x} & : & x_0 \\ n_{0y} & s_{0y} & a_{0y} & : & y_0 \\ n_{0z} & s_{0z} & a_{0z} & : & z_0 \\ \hline 0 & 0 & 0 & : & 1 \end{vmatrix} \quad \bar{P}_0 = \begin{vmatrix} x_0 \\ y_0 \\ z_0 \\ \phi_0 \\ \theta_0 \\ \psi_0 \end{vmatrix} \quad (2.7)$$

Assim:

$$T_{N+1}^0 = A_1^0(\bar{P}_0) \cdot T_{N+1}^1(\bar{k}') = T_{N+1}^0(\bar{k}) \quad (2.8)$$

$$\bar{k} = [\bar{P}_0^t \ \bar{P}_1^t \ \dots \ \bar{P}_N^t]^t$$

2.2 - MODELAGEM DO ERRO DE POSICIONAMENTO

Seja $\delta \mathbf{p}_i$, o vetor de erros relacionados com os parâmetros da matriz de transformação A_{i+1}^i do sistema. A matriz de transformação exata ${}^e A_{i+1}^i$, pode ser escrita como:

$${}^e A_{i+1}^i = A_{i+1}^i + dA_i \quad dA_i = dA_i(\delta \bar{\mathbf{p}}_i) \quad (2.9)$$

$\delta \bar{\mathbf{p}}_0 = [\delta x_0, \delta y_0, \delta z_0, \delta \phi_0, \delta \theta_0, \delta \psi_0]^t$: vetor de erros relativos aos parâmetros de posicionamento na matriz de transformação A_1^0 .

$\delta \bar{\mathbf{p}}_i = [\delta \theta_i, \delta d_i, \delta a_i, \delta \alpha_i]^t$: vetor de erros relativos aos parâmetros da matriz de transformação A_{i+1}^i no caso de juntas não paralelas.

$\delta \bar{\mathbf{p}}_i = [\delta \theta_i, \delta d_i, \delta a_i, \delta \alpha_i, \delta \beta_i]^t$: vetor de erros relativos aos parâmetros da matriz de transformação A_{i+1}^i no caso de juntas nominalmente paralelas.

A matriz de transformação exata do sistema é:

$${}^e T_{N+1}^0 = \prod_{i=0}^N (A_{i+1}^i + dA_i) \quad (2.10)$$

Pode-se demonstrar* que expandindo a equação anterior, e ignorando termos de segunda e maior ordem, podemos escrever a seguinte relação :

$$\delta \bar{\mathbf{T}} = J \cdot \delta \bar{\mathbf{k}} \quad (2.11)$$

$\delta \bar{\mathbf{k}}$ - vetor de erros dos parâmetros: $\delta \bar{\mathbf{k}} = [\delta \bar{\mathbf{p}}_0^t \ \delta \bar{\mathbf{p}}_1^t \dots \delta \bar{\mathbf{p}}_N^t]^t$

$\delta \bar{\mathbf{T}}$ - vetor de erros de posicionamento. $\delta \bar{\mathbf{T}} = [dx \ dy \ dz \ rx \ ry \ rz]^t$, onde dx, dy, dz são pequenos deslocamentos na direção x, y e z , e rx, ry, rz são pequenas rotações em torno dos eixos x, y e z .

* Demonstração no anexo D.

$$J = \begin{vmatrix} \frac{\partial p_x}{\partial \text{par}_1} & \frac{\partial p_x}{\partial \text{par}_2} & \dots & \frac{\partial p_x}{\partial \text{par}_{\text{npar}}} \\ \frac{\partial p_y}{\partial \text{par}_1} & \frac{\partial p_y}{\partial \text{par}_2} & \dots & \frac{\partial p_y}{\partial \text{par}_{\text{npar}}} \\ \frac{\partial p_z}{\partial \text{par}_1} & \frac{\partial p_z}{\partial \text{par}_2} & \dots & \frac{\partial p_z}{\partial \text{par}_{\text{npar}}} \\ \frac{\partial \psi_0}{\partial \text{par}_1} & \frac{\partial \psi_0}{\partial \text{par}_2} & \dots & \frac{\partial \psi_0}{\partial \text{par}_{\text{npar}}} \\ \frac{\partial \theta_0}{\partial \text{par}_1} & \frac{\partial \theta_0}{\partial \text{par}_2} & \dots & \frac{\partial \theta_0}{\partial \text{par}_{\text{npar}}} \\ \frac{\partial \phi_0}{\partial \text{par}_1} & \frac{\partial \phi_0}{\partial \text{par}_2} & \dots & \frac{\partial \phi_0}{\partial \text{par}_{\text{npar}}} \end{vmatrix}$$

(2.12)

par_i - parâmetro i.
npar - número de parâmetros.

O algoritmo para estimativa proposto baseia-se na equação 2.11.
A matriz J terá o número de colunas igual ao número de parâmetros a ser estimado. O número de linhas será igual à 6 no caso de se relacionar erros de parâmetros com erros de posição e orientação durante o processo de estimativa, ou igual a três no caso de se usar apenas medidas de posição durante a segunda fase do processo de estimativa proposto no item 1.1 desta dissertação.

2.3 - MODELAGEM GEOMÉTRICA ATRAVÉS DE VETORES LOCAIS

Ao contrário do modelo utilizado no programa de estimativa, onde o que interessa são apenas as coordenadas p_x , p_y , p_z , ψ , θ , ϕ do elemento terminal do robô, no software de programação off-line é necessário a determinação das coordenadas de vários pontos do robô para construir o desenho gráfico através de sólidos geométricos (paralelepípedos, cilindros e outros) que representam o robô e executar testes de colisão com o ambiente externo. Este modelo deve incluir uma matriz para correção de erros de posicionamento identificados com características semelhantes à matriz ΔT na equação D.10 do anexo D. Consequentemente, para o software de programação off-line, optou-se por modelar geometricamente o robô através de vetores locais de translação e rotação. A seguir é demonstrado simplifiadamente como este tipo de modelagem pode ser implementado em uma linguagem de programação visando a construção do software de visualização gráfica, tendo como exemplo o manipulador KRAFT.

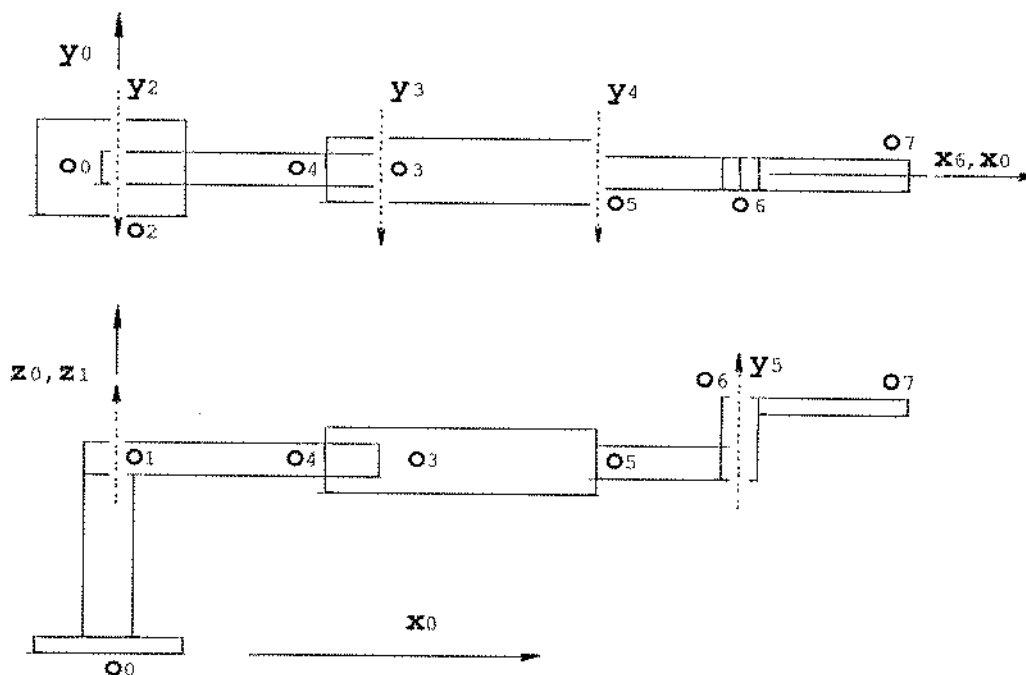


Figura 2.2 - Pontos de interesse para o software de visualização e teste de obstáculos (manipulador KRAFT).

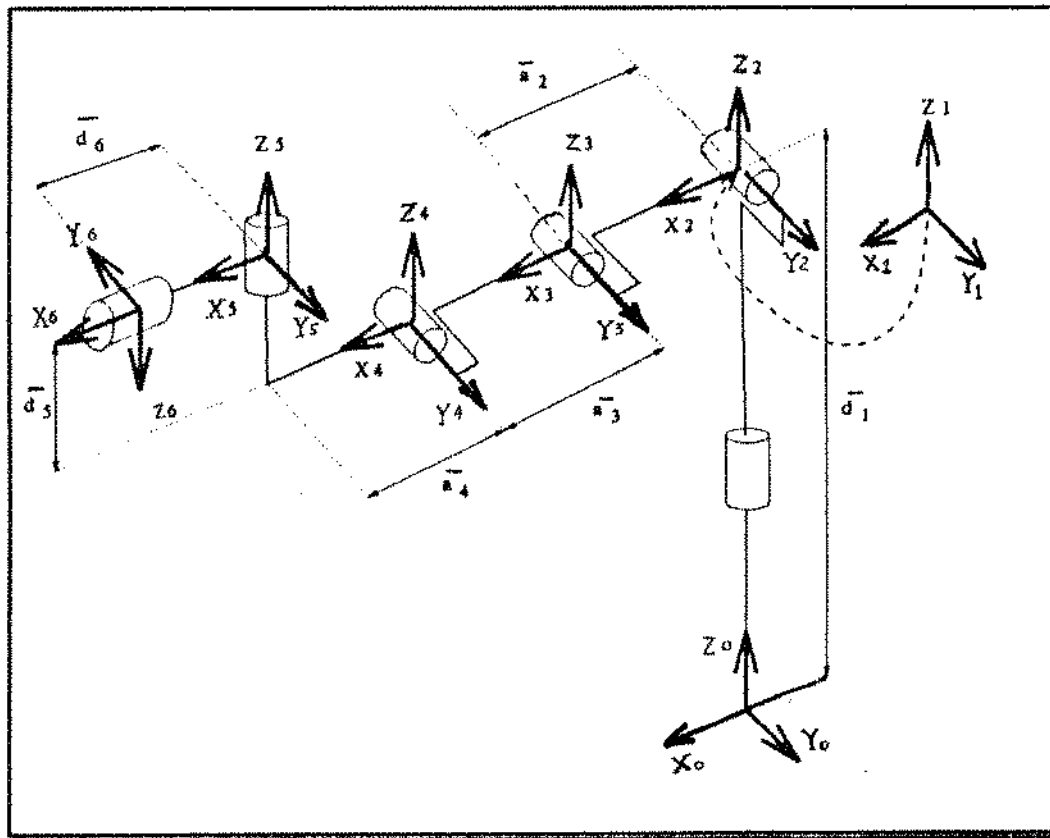


Figura 2.3 - Exemplo mostrando os referenciais e os vetores associados a cada link.

i) Inicia-se a modelagem arbitrando uma configuração inicial e fazendo-se as definições para os vetores de rotação locais do sistema (ver figura 2.2 e 2.3). É associado para cada link i um referencial local R_i . A definição desses referenciais locais para cada articulação i , é realizada através da matriz W_LOCAL , composta de vetores de rotação especificados em relação ao referencial R_{i-1} :

```
W_LOCAL : matriz := (
  ( 0 , 0 , 1 ), -- R1 fixo em O1 no link 1: rotação em relação a z0
  ( 0 , 1 , 0 ), -- R2 fixo em O1 no link 2: rotação em relação a y1
  ( 0 , 1 , 0 ), -- R3 fixo em O3 no link 3: rotação em relação a y2
  ( 0 , 1 , 0 ), -- R4 fixo em O5 no link 4: rotação em relação a y3
  ( 0 , 0 , 1 ), -- R5 fixo em O6 no link 5: rotação em relação a z4
  ( 1 , 0 , 0 )) -- R6 fixo em O7 no link 6: rotação em relação a x5
```

ii) Definição dos vetores de translação, para os pontos de interesse em cada link i , especificados em relação ao referencial R_i , através da matriz VET_LOC (ver figura 2.2 e 2.3):

```
VET_LOCAL : matriz := (
( 0 , 0 , d1 ),      -- Translação 00/01 em relação a z1
( 0 , c1 , 0 ),      -- Translação 00/02 em relação a y0
( a2 , 0 , 0 ),      -- Translação 01/03 em relação a x2
(-a21 , 0 , 0 ),     -- Translação 03/04 em relação a x3
( a3 , 0 , 0 ),      -- Translação 03/05 em relação a x3
( a4 , 0 , d5 ),     -- Translação 05/06 em relação a x4 e z4
( d6 , 0 , 0 )       -- Translação 06/07 em relação a x5
( Xfer, yfer, zfer)) -- Translação 07/08 em relação a x6,y6,z6*
```

* (Xfer, yfer, zfer) - Suposta extremidade de uma ferramenta.

iii) A seguir deve-se construir as matrizes de transformação $S[i]$, associadas a cada link i . Sendo θ_i a variação angular da junta i , as matrizes $S[i]$ serão função de θ_i e do vetor local de rotação $W_LOCAL[i]$:

```
S[1] = F (θ1,W_LOCAL[1])
S[2] = F (θ2,W_LOCAL[2])
.
.
.
S[6] = F (θ6,W_LOCAL[6])
```

iv) Com o objetivo de obter a visualização do robô em vários ângulos pelo software de programação, define-se:

M_V - matriz de Euler com variáveis do software que definem o ângulo de visualização gráfica.

v) Visando a correção do referencial da base devido à possíveis erros de posicionamento do robô em relação a um sistema de referencia arbitrado (referencial de operação - R_{op}), são definidos:
 RT_A - matriz de correção de erros de orientação estimados entre a base do robô R_0 e R_{op} .

VT_A - vetor de correção de erros de posição estimados entre R_0 e R_{op} .

A matriz RT_A e o vetor VT_A possuem características análogas à matriz ΔT (equação D.10 do anexo D).

vi) Definida as matrizes de transformação de coordenadas entre referenciais de links sucessivos ($S[i]$), a matriz de correção de orientação RT_A , e a matriz que determina o ângulo de visualização M_V pode-se calcular as matrizes de transformação $MG[j]$ entre os referenciais R_i e R_{op} (referencial de operação escolhido):

```
MG[1] := M_V * RT_A      -- correção entre R0 e Rop
MG[2] := MG[1] * S[1]    -- transformação entre R1 e Rop
MG[3] := MG[2] * S[2]    -- transformação entre R2 e Rop
.      =                  .
.      =                  .
MG[7] := MG[6] * S[6]    -- transformação entre R6 e Rop
```

vii) Determinação dos pontos de interesse O_i , em relação ao referencial de visualização e corrigidos pela matriz de erros estimados:

```
O(0) := MG[1] * VT_A ;
O(1) := O(0) + MG[2] * VET_LOC [1] ;
O(2) := O(0) + MG[2] * VET_LOC [2] ;
O(3) := O(1) + MG[3] * VET_LOC [3] ;
O(4) := O(3) + MG[4] * VET_LOC [4] ;
O(5) := O(3) + MG[4] * VET_LOC [5] ;
O(6) := O(5) + MG[5] * VET_LOC [6] ;
O(7) := O(6) + MG[6] * VET_LOC [7] ;
O(8) := O(7) + MG[7] * VET_LOC [8] ;
```

Assim é possível utilizar esses pontos para construir o desenho de visualização e possibilitar testes de colisão entre os diferentes componentes do sistema (robô, base móvel e painel de operação).

CAPÍTULO 3

ALGORITMO PROPOSTO PARA A ESTIMATIVA DE PARÂMETROS

Partindo do modelo linearizado, que associa pequenas variações de posicionamento, com erros nos parâmetros através do Jacobiano (equação 2.11), é proposto um método iterativo, que através de dados de uma sequência de medidas de posicionamento no espaço cartesiano e informações dos sensores angulares, possibilite obter a estimativa dos parâmetros de um sistema, usando o critério de mínimos quadrados (ver figura 3.1).

É construído então, um algoritmo que a cada iteração, um vetor estimador dos parâmetros $\hat{\mathbf{k}}$, deve convergir de um valor inicial $\hat{\mathbf{k}}_0$ (arbitrário) até um valor próximo do real, através de incrementos $\delta\hat{\mathbf{k}}$, tendo como condição de convergência, um valor pequeno para este incremento (δk_{final}).

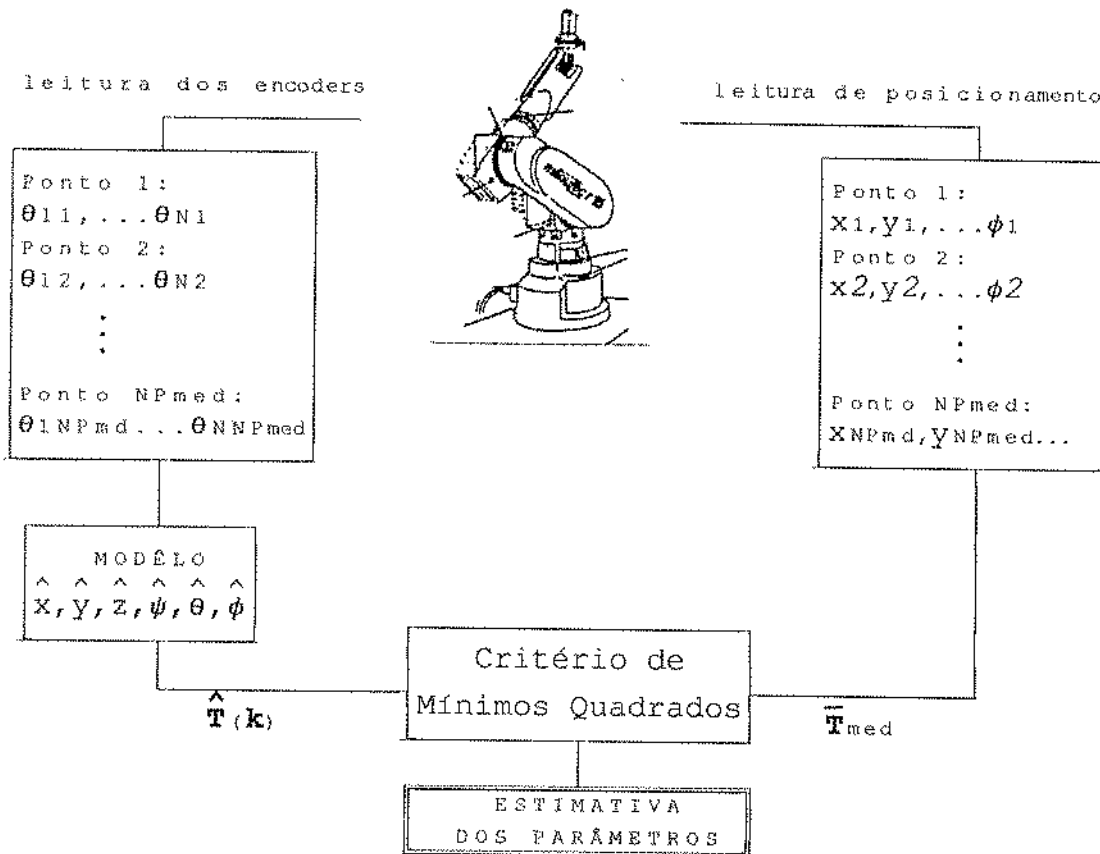


Figura 3.1 - Fluxo de dados durante o processo de estimativa.

Sendo:

$\hat{\mathbf{k}}_i$ - vetor estimador dos parâmetros na iteração i ;

$\bar{\mathbf{T}}_{med} = [x_m, y_m, z_m, \psi_m, \theta_m, \phi_m]^t$ - posição e orientação medida;

$\bar{\mathbf{T}}_{mod}(\hat{\mathbf{k}}_i) = [px, py, pz, \psi, \theta, \phi]^t$ - posição e orientação calculada pelo modelo na iteração i ;

$$\delta \bar{\mathbf{T}}(\hat{\mathbf{k}}_i) = \bar{\mathbf{T}}_{med} - \bar{\mathbf{T}}(\hat{\mathbf{k}}_i) \quad (3.1)$$

$$\delta \bar{\mathbf{T}}(\hat{\mathbf{k}}_i) = \mathbf{J}(\hat{\mathbf{k}}_i) \cdot \delta \hat{\mathbf{k}}_i \quad (3.2)$$

Sendo NPmed, o número de medidas:

$$\delta \bar{\mathbf{T}} = \begin{bmatrix} \delta \bar{\mathbf{T}}_1 \\ \delta \bar{\mathbf{T}}_2 \\ \vdots \\ \delta \bar{\mathbf{T}}_{NPmed} \end{bmatrix} = \begin{bmatrix} J_1(\hat{\mathbf{k}}) \\ J_2(\hat{\mathbf{k}}) \\ \vdots \\ J_{NPmed}(\hat{\mathbf{k}}) \end{bmatrix} \cdot \delta \hat{\mathbf{k}} = J(\hat{\mathbf{k}}) \cdot \delta \hat{\mathbf{k}} \quad (3.3)$$

$$\delta \hat{\mathbf{k}}_i = (J(\hat{\mathbf{k}}_i))^{-1} \cdot \delta \bar{\mathbf{T}} \quad (3.4)$$

$$\hat{\mathbf{k}}_{i+1} = \hat{\mathbf{k}}_i + \delta \hat{\mathbf{k}}_i \quad (3.5)$$

Para obter a inversão da matriz $J(\hat{\mathbf{k}})$ na equação 3.4 é utilizado o algoritmo de pseudo-inversão de Greville [3], que satisfaz o critério de mínimos quadrados.

ALGORITMO DE ESTIMATIVA	
0.	Inicialização do vetor estimador de parâmetros $\hat{\mathbf{k}}_0$.
1.	Cálculo de $\bar{\mathbf{T}}(\hat{\mathbf{k}}_i)$.
2.	Cálculo de $\delta \bar{\mathbf{T}} = \bar{\mathbf{T}}_{med} - \bar{\mathbf{T}}(\hat{\mathbf{k}}_i)$.
3.	Cálculo de $[J(\hat{\mathbf{k}}_i)]^{-1}$.
4.	Cálculo de $\delta \hat{\mathbf{k}}_i$ a partir de 2 e 3 (equação 3.4).
5.	$\hat{\mathbf{k}}_{i+1} = \hat{\mathbf{k}}_i + \delta \hat{\mathbf{k}}_i$.
6.	Caso $\delta k_i < \delta k_{final}$ for falsa para algum parâmetro retornar para 1. Caso contrário, fim da estimativa.

Figura 3.2 - Algoritmo de estimativa proposto.

Os valores dos parâmetros iniciais de $\hat{\mathbf{k}}_0$, referentes ao robô, podem ser considerados como sendo os valores fornecidos pelo fabricante. Os valores iniciais para o vetor $\bar{\mathbf{p}}_0$ (posição e orientação entre o referencial da base do robô e o referencial de operação) podem ser medidos ou calculados sem necessidade de precisão.

Tendo como entrada, tanto medidas de posição quanto de orientação ($\bar{T}_{med} = [x_m, y_m, z_m, \psi_m, \theta_m, \phi_m]^t$), o número de parâmetros que podem ser estimados (N_{pi}) pode ser calculado pela equação:

$$N_{pi} = 6 + 4.J_n + 5.J_p \tag{3.6}$$

J_n - número de juntas não paralelas.
 J_p - número de juntas nominalmente paralelas.

No caso de se usar medidas de posição e orientação, a matriz J na equação 3.3 será formada por $npmed$ submatrizes (ver equação 2.29) e terá o número de linhas igual à $6 \cdot NPmed$ e o número de colunas igual à $Npar$.

$NPmed$ - número de pontos medidos.
 $Npar$ - número de parâmetros a serem estimados.

Utilizando como entrada para o algoritmo, medidas apenas de posição ($\bar{T}_{med} = [x_m, y_m, z_m]^t$), o número de parâmetros que podem ser estimados são :

$$N_{pi} = 2 + 4.J_n + 5.J_p \tag{3.7}$$

Neste caso a matriz J (eq. 3.3) será formada também por $npmed$ submatrizes porém com apenas três linhas (ver equação 3.8). O número de linhas será igual à $3 \cdot NPmed$ e o número de colunas igual à $Npar$.

$$J_p = \begin{vmatrix} \frac{\partial p_x}{\partial par_1} & \frac{\partial p_x}{\partial par_2} & \dots & \frac{\partial p_x}{\partial par_{Npar}} \\ \frac{\partial p_y}{\partial par_1} & \frac{\partial p_y}{\partial par_2} & \dots & \frac{\partial p_y}{\partial par_{Npar}} \\ \frac{\partial p_z}{\partial par_1} & \frac{\partial p_z}{\partial par_2} & \dots & \frac{\partial p_z}{\partial par_{Npar}} \end{vmatrix} \tag{3.8}$$

p - varia de 1 até $npmed$.

Ao se fazer medidas apenas de posição para o procedimento de estimativa, deve-se garantir que os eixos dos referenciais da base do robô e do dispositivo de medidas sejam paralelos, não podendo haver variações na orientação entre esses referenciais. Consequentemente três dos quatro parâmetros que não são estimados devido à falta de informação sobre a orientação (observar equações 3.6 e 3.7), são os ângulos de orientação entre o referencial da base do robô e do referencial de medidas (ou de operação), que foram supostos serem conhecidos (eixos paralelos: $\psi_0 = \theta_0 = \phi_0 = 0$). O quarto parâmetro que não pode ser estimado deve ser escolhido entre a dimensão do primeiro link d_1 e a posição do referencial de medidas (ou operação) z_0 com relação ao referencial da base do robô na direção do eixo z .

Isto é devido à suposição feita acima (referenciais da base do robô e de medidas com eixos paralelos), e à notação de Denavit-Hartenberg (DH), que define seus parâmetros ao longo dos eixos dos referenciais associados à cada link. O fato de DH definir d_1 na direção z do referencial da base leva a que esses dois parâmetros (d_1 e z_0) se fixem na mesma direção, gerando vetores linearmente dependentes na matriz J e fazendo com que o algoritmo para estimativa não consiga fazer a distinção entre um e outro parâmetro.

Devido a restrição a que impõe os dispositivos de medidas usado durante os experimentos (impossibilidade de medir orientação), é construído um programa para estimativa baseado no algoritmo apresentado e tendo como entrada apenas medidas de posição.

No capítulo seguinte são realizadas simulações com o objetivo de testar o algoritmo para a estimativa e algumas variáveis dentro do processo.

CAPÍTULO 4

VALIDAÇÃO DO ALGORITMO PROPOSTO A PARTIR DE SIMULAÇÕES

4.1 - METODOLOGIA

Com o objetivo de testar o algoritmo de estimativa e estudar a influência de algumas variáveis dentro do processo são realizadas várias simulações de estimativa. Nestas simulações os valores das medidas de posicionamento, que num processo real, deveriam ser fornecidas por um dispositivo externo de medição, são gerados através do modelo do robô com um vetor de parâmetros supostos exatos ($\bar{k}_e$). Os erros de posicionamento ($\delta\bar{T}$), serão então, a diferença entre os valores fornecidos pelo modelo com parâmetros supostamente exatos ($\bar{k}_e$) gerando $\bar{T}_{med}$, e pelo modelo com o vetor estimador de parâmetros ($\hat{k}$) que gera $\bar{T}_{mod}$.

$$\delta\bar{T} = \bar{T}_{med}(\bar{k}_e) - \bar{T}_{mod}(\hat{k})$$

As variáveis dentro do processo de estimativa a serem analisadas são:

A - Porcentagem de Erro no Vetor Inicial dos Parâmetros - Per(%).

Dados valores para o vetor de parâmetros supostamente exatos ($\bar{k}_e$), o vetor inicial de parâmetros estimadores ($\hat{k}_0$) para o algoritmo será calculado do seguinte modo:

- Parâmetros cuja dimensão é linear (d_i e a_i):

$$K_{0_{Li}} = (1 \pm \text{Per}/100) * K_{e_{Li}} \quad (4.1a)$$

L_i - varia de 1 até o número de parâmetros lineares.

- Parâmetros cuja dimensão é angular (θ_i e α_i):

Para se obter uma maior consistência no significado da variação dos valores iniciais para os parâmetros que variam angularmente, a variável Per será aplicada de modo distinto a esses parâmetros não lineares. Arbitrado um erro δK_{An} para o parâmetro angular $K_{e_{An}}$, o valor inicial deste parâmetro será:

$$K_{0_{An}} = K_{e_{An}} + (1 \pm \text{Per}/100) * \delta K_{An} \quad (4.1b)$$

N_L - varia de 1 até o número de parâmetros não lineares.

O erro é escolhido positivo ou negativo aleatoriamente. Espera-se com esta variável (Per) saber como a convergência do algoritmo se comporta com relação aos valores iniciais dos parâmetros.

B) Número de Pontos Medidos - NPmed.

Um grande número de medidas pode diminuir a influência dos erros de medidas. O número mínimo de pontos deve ser aquele que possibilite o sistema possuir no mínimo um número de equações igual ao número de parâmetros a estimar. O número máximo de pontos considerados foi de 60.

C) Variação Angular entre os Limites das Juntas - Var (%).

Com o objetivo de obter a influência da variação angular em cada junta, são realizadas simulações, onde pode-se gerar um posicionamento ($\bar{T}_{med}$), movendo-se as juntas dentro de 0 a 100% de seus respectivos ângulo entre limites de junta ($\Delta\theta_i$). Assim os ângulos θ_{1j} , θ_{2j} , ..., θ_{nj} gerados aleatoriamente pelo programa de simulação e que gera T_{med}' , devem estar dentro dessa variação. Variar dentro de 100% significa poder atingir todo volume de trabalho do robô.

$$\theta_{ij}' = R_{ij} * Var * \Delta\theta_i \quad (4.2)$$

i - varia de 1 até o número de graus de liberdade do robô.

j - varia de 1 até o número de pontos medidos (NPmed).

R_{ij} - número randomico entre -1 e 1 para cada valor de θ_{ij} .

$\Delta\theta_i = (\theta_{maxi} - \theta_{mini})$ - ângulo entre limites da junta i (anexo A).

D) Erro de Medidas - $E_{max} (mm)$.

Simulações variando-se erros de medidas (R_{med}), são interessantes pelo fato de fornecer informações que possibilite estimar a precisão que um instrumento de medida deve possuir num processo de estimativa real. Sendo $\bar{T}_{med}'$ o vetor de medidas calculado pelo modelo com os parâmetros supostamente exatos, os valores do vetor $\bar{T}_{med}$ que entrará na simulação do processo de estimativa será:

$$\bar{T}_{med} = \bar{T}_{med}' + \bar{R}_{med} \quad (4.3)$$

Os valores que compõe $\bar{R}_{med}$ são gerados aleatoriamente da seguinte forma:

$$R_{med_{nm}} = R_{nm} * E_{max} \quad (4.4)$$

R_{nm} - número randomico entre -1 e 1, para cada valor das medidas.

E_{max} - erro máximo suposto nas medidas.

nm - varia de 1 até o número de medidas.

E) Ruído no Sinal dos Encoders - $R_{max} (graus)$.

Num processo prático de estimativa é necessária a leitura dos sensores de posição angular para o cálculo do posicionamento do robô pelo modelo com os valores estimadores dos parâmetros ($\bar{T}_{mod}(\hat{k})$). A existência de ruído nestas informações pode influenciar na estimativa, sendo então necessária a introdução de mais uma variável a ser estudada. Nas simulações os valores dos ângulos (θ_{1j}' , θ_{2j}' , ..., e θ_{Nj}'), calculados pela equação 4.2 para gerar $\bar{T}_{med}$, são acrescidos de um erro devido ao ruído para gerar $\bar{T}_{mod}$:

$$\theta_{ij} = \theta_{ij}' + R_{ij} * R_{max} \quad (4.5)$$

θ_{ij}' - ângulo de junta gerada aleatoriamente (equação 4.2) para obtenção dos dos vetores $\bar{T}_{med}$.

R_{ij} - números gerados aleatoriamente entre -1 e 1 para cada valor de θ_{ij}' .

R_{max} - ruído máximo suposto.

F) Incremento Final - δk_{final} .

Como condição necessária para convergência do algoritmo determina-se um valor δk_{final} conforme apresentado na figura 3.2 do capítulo 3. São realizadas algumas simulações para poder associar o valor de δk_{final} com a precisão de estimativa.

As simulações foram realizadas, utilizando-se os valores numéricos e modelo do robô MANUTEC r3, o qual foi objeto de um processo real de estimativa (capítulo 5). A partir dessas simulações torna-se possível a validação da metodologia proposta. Sendo que no experimento prático, o dispositivo de medidas utilizado não permite obter medidas angulares, para se obter coerência entre o experimento e as simulações, serão gerados pelo software de simulação apenas valores de posição ($T_{med} = (x_m, y_m, z_m)^t$). Isto implica na possibilidade de estimar apenas dois parâmetros de posicionamento (x_0 e y_0) e não todos os seis ($x_0, y_0, z_0, \psi_0, \theta_0, \phi_0$) conforme exposto no capítulo 3. Os planos xy dos referenciais de medidas e da base do robô são considerados coplanares e conseqüentemente z_0 é igual a zero.

Foram escolhidos como parâmetros próprios do robô a identificar, as dimensões lineares d_1 , a_2 , d_4 e d_6 (comprimento dos links segundo notação Denavit-Hartenberg [6], [20]), e os valores dos offsets dos encoders das juntas 1, 2, 3, 4 e 5 (θ_{10} , θ_{20} , θ_{30} , θ_{40} e θ_{50}). Os off-sets das juntas são os valores dos ângulos de junta na posição de inicialização do robô.

O offset θ_{60} não será estimado na simulação e durante o experimento prático, devido à geometria da ferramenta utilizada nas medidas (figuras 5.1 e 5.2). A extremidade da ferramenta, a qual é referência para as medidas, é alinhada com o eixo de rotação da junta 6. Caso esta extremidade estivesse a uma distância normal, o offset θ_{60} poderia ser estimado através das medidas.

Sendo que o robô Manutec é de alta precisão, é suposto que os outros parâmetros geométricos e não geométricos não possuem grandes erros e portanto não influem em erros no posicionamento. Assim considera-se que os desvios angulares de paralelismo sejam nulos ($\beta_i = 0$), validando a hipótese usada para a convergência do algoritmo (equação 2.11). As conclusões sobre os parâmetros geométricos considerados fornecidas pelas simulações podem ser generalizadas para os outros parâmetros geométricos que consideramos não variar.

O vetor de parâmetros a ser estimado é:

$$\bar{\mathbf{k}} = (d_1, d_4, d_6, a_2, \theta_{10}, \theta_{20}, \theta_{30}, \theta_{40}, \theta_{50}, x_0, y_0)^T$$

O modelo geométrico do robô calculado segundo Denavit-Hartenberg e fornecendo valores de posição (p_x , p_y , p_z) em relação ao referencial de medidas é (ver anexo A):

$$\begin{aligned} p_x &= \{(C_4S_5d_6).C_{23} + (C_5d_6 + d_4).S_{23} + a_2C_2\}.C_1 - \{S_4S_5d_6\}.S_1 - x_0 \\ p_y &= \{(C_4S_5d_6).C_{23} + (C_5d_6 + d_4).S_{23} + a_2C_2\}.S_1 + \{S_4S_5d_6\}.C_1 - y_0 \\ p_z &= -(C_4S_5d_6).S_{23} + (C_5d_6 + d_4).C_{23} - a_2S_2 + d_1 \end{aligned} \quad (4.6)$$

A matriz do jacobiano J será então:

$$J = \begin{vmatrix} \partial p_x / \partial d_1 & \partial p_x / \partial d_4 & \dots & \partial p_x / \partial y_0 \\ \partial p_y / \partial d_1 & \partial p_y / \partial d_4 & \dots & \partial p_y / \partial y_0 \\ \partial p_z / \partial d_1 & \partial p_z / \partial d_4 & \dots & \partial p_z / \partial y_0 \end{vmatrix} \quad (4.7)$$

4.2 - Análise da Influência de Variáveis do Procedimento.

Procurando obter resultados principalmente qualitativos, são realizados os conjuntos A, B, C, D, E, e F de simulações onde analisa-se as variáveis descritas anteriormente (Per , NP_{med} , Var , E_{max} , R_{max} , δk_{final}), mudando-se os valores para a variável em questão e fixando as demais. Os resultados dados pelo programa de simulação são apresentados a partir de tabelas no Anexo B. A primeira coluna das tabelas refere-se aos parâmetros a serem estimados. A segunda coluna corresponde aos valores de parâmetros considerados exatos ($\bar{k}_e$) e usados no modelo que gera a posição medida ($\bar{T}_{med}$). Os valores supostos exatos de d_1 , d_4 , d_6 , e a_2 , são dados fornecidos pelo fabricante do robô. Os supostos valores exatos de θ_{10} a θ_{50} são calculados do seguinte modo:

$$\theta_{i0} = \theta_{in} + \delta\theta_i \quad (4.8)$$

θ_{in} - valor angular de inicialização fornecida pelo encoder.

$\delta\theta_i$ - valor arbitrário de erro dos encoders para as simulações.

Para essas simulações os valores de erros angulares para para o cálculo de θ_{10} , θ_{20} , θ_{30} , θ_{40} e θ_{50} , são escolhidos serem de :

ENCODER	$\delta\theta_i$ (graus)	θ_{in} (graus)	θ_{i0} (graus)
1	1	28.436	29.436
2	-1	69.043	68.043
3	2	-128.925	-126.925
4	1	-59.027	-58.027
5	-2	24.996	22.996

Tabela 4.1

As distâncias entre o referencial da base do robô e o referencial de medidas x_0 e y_0 , poderiam também ser arbitrários no caso de simulações e neste caso são valores próximos dos utilizados nos experimentos práticos. Assim em cada simulação os parâmetros a serem estimados deverão convergir para os valores supostamente exatos e serão os mesmos para as simulações dos conjuntos A à F.

$\bar{k}_e = (665, 730, 160.7, 500, 29.436,$
 $68.043, -126.925, -58.027, 22.996, -408, 1470)^t$

Na terceira coluna são apresentados os valores do vetor inicial de parâmetros nominais $\hat{k}_0$, resultantes da equação 4.1a e 4.1b. O vetor de parâmetros estimados pelo algoritmo k é listado na quarta coluna. Para o conjunto de simulações A até F, a quarta coluna fornece a precisão de estimativa (PI) em função dos parâmetros supostamente exatos $(k_e)^{-}$ e os valores estimados $(\bar{k})$, calculados da seguinte forma:

• Parâmetros lineares ($d_1, d_4, d_6, a_2, x_0, y_0$):

$PI (\%) = \text{abs}[(k_e - k) * 100 / k_e]$ (4.9a)

• Parâmetros angulares (θ_{10} , θ_{20} , θ_{30} , θ_{40} e θ_{50}):

$$PI(k) = \text{abs}[(k_e - k) * 100 / \delta k_e] \quad (4.9b)$$

δk_e - erro arbitrado para os valores dos offsets (ver tabela 4.1).

As simulações realizadas (anexo B) permitem a obtenção das seguintes conclusões:

A) Erros no Vetor Inicial de Parâmetros Nominais (Simulações A1 a A5)

Durante a análise da influência dos valores do vetor inicial de parâmetros nominais ($\bar{k}_0$), é observado uma grande tolerância quanto a um valor arbitrário inicial. Houve boa convergência, notando-se nestas primeiras simulações, melhores resultados na convergência dos seis parâmetros lineares d_1 , d_4 , d_6 , a_2 , x_0 e y_0 . Esta característica permanece nas demais simulações e é devido ao fato de que esses seis parâmetros variam linearmente na relação entre erros diferenciais de posicionamento e erros nos parâmetros (equação 2.11), ao contrário dos outros cinco parâmetros referentes aos valores angulares dos offsets dos encoders. Pode se observar isto ao se calcular a matriz jacobiana J .

Variando $\hat{k}_0$ entre 1% e 2500% de k_e (conforme equações 4.1a e 4.1b) observou-se a mesma convergência. Já para uma variação de 3000%, observou-se a não convergência de θ_{30} e θ_{40} , mostrando ser um limite para uma estimativa nas condições supostas (número de pontos, erros nas medidas, ruído nos encoders,...). Porém este valor limite é muito grande, sendo que a estimação de valores iniciais adequados para o algoritmo não deve ser problema para se garantir uma convergência. O número de iterações aumenta pouco com o aumento dos erros no vetor de parâmetros iniciais.

B) Número de Pontos Medidos (B1 a B4).

Baseando-se nas simulações B1 (60 pontos) e B2 (20 pontos), a diminuição de pontos medidos, não provocou uma pior convergência de todos os parâmetros, sendo que alguns melhoraram, ao contrário do que se esperava. Uma possível explicação é que apesar da diminuição do número de informações, o conjunto menor de pontos usados na identificação B2 possuíam valores de posição ($\overline{T_{med}}$) e ângulos de juntas θ_{ij} que com seus respectivos erros e ruídos aleatórios, proporcionaram uma melhor estimativa de alguns parâmetros. Em função das simulações B1 e B2 e outras, pode se notar que, aumentando-se as medidas, a partir de um certo número, não existe melhora significativa na precisão de estimativa. Já com a diminuição para menos de 20 pontos, a estimativa é debilitada mais acentuadamente (simulações B3 e B4). O número de iterações é maior com a diminuição dos pontos.

C) Variação Angular entre Limites das Juntas (C1 a C4).

Fazendo comparações entre as simulações do conjunto C, percebe-se a importância de se variar os ângulos das juntas. Nas simulações C1 (Var = 1%) e C2 (Var = 10%) nota-se uma pequena precisão nas estimativas, principalmente nas dos offsets das juntas, mostrando-se a necessidade de se variar razoavelmente os movimentos de cada junta para se obter uma estimativa satisfatória. Como acontece com o número de pontos, a partir de um valor de Var, a precisão de estimativa não aumenta tão significativamente.

D) Erro nas Medidas de Posição (D1 a D4).

Conhecer a precisão com a qual deve se realizar medidas de posicionamento para que a estimativa dos parâmetros em questão, alcance a precisão necessária é muito importante. Pode ser notado nas simulações D1, D2, D3 e D4, que existe uma relação direta entre o aumento do erro nas medidas e a diminuição da precisão no processo de estimativa. O número de iterações é igual em todas simulações.

E) Ruído na Leitura dos Encoders (E1 a E4).

O aumento do ruído nos encoders, provocou grande perda de precisão na estimativa, principalmente nos valores dos offsets, Quando este valor ultrapassou 0.1 grau os erros aumentaram significativamente (comparar E1 e E2 com E3 e E4) podendo assumir este valor como um valor limite para o ruído em um processo de estimativa nas condições impostas.

F) Condição de Convergência (F1 a F4).

A simulação F4 mostra o quanto o algoritmo converge rapidamente, sendo que para uma diferença de 100% entre parâmetros nominais e o exato, apenas 3 iterações foram suficientes para se obter precisão menores de 1% na estimativa em quase todos parâmetros. Isto é devido aos grandes incrementos (δk_i) nas primeiras iterações. Para assegurar a precisão necessária, deve-se dar à δk_{final} o valor da precisão que se quer e supõe poder obter (por exemplo $\delta k_{final}=0.01$ no caso de precisão de décimos de milímetros, ou décimos de graus). Valores de δk_{final} menores à necessária, irá gastar um maior tempo de processamento com um maior número de iterações (simulações F1, F2, F3 e F4).

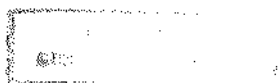
4.3 - Simulação do Experimento Prático (simulações G1 a G5).

Visando representar ao máximo o procedimento de estimativa realizado durante os experimentos práticos (capítulo 5), é apresentado o conjunto de simulações G (resultados no anexo B). Os valores das seis variáveis analisadas nas simulações anteriores são determinadas de acordo com o experimento.

Ga) Diferença entre o Vetor Inicial de Parâmetros Nominais ($\bar{k}_0$) e o de Parâmetros supostamente Exatos ($\bar{k}_e$).

Para os valores supostamente exatos dos parâmetros d_1 , d_4 , d_6 e a_2 , serão usados os valores de fabricação do robô, fornecido pelo fabricante e também usados nas simulações anteriores. Estima-se que os erros devem estar dentro de uma faixa de 1% destes valores de fabricação. Os parâmetros de posição (x_0 e y_0) foram medidos com boa precisão e seus erros estão dentro de 1% das medidas. Superestimando o erro, para a simulação escolhe-se para estes seis parâmetros lineares uma faixa de 5% de erro. Devido à alta precisão do robô, espera-se pouco erro. O erro na inicialização do robô ($\delta\theta_1$), isto é, dos offsets das juntas é estimado estar em torno de 6 impulsos do encoder. Como o valor correspondente em graus para um impulso no encoder de menor resolução é de 0.0025, é escolhido ser de 0.015 graus (negativo ou positivo aleatoriamente) o valor do erro $\delta\theta_1$ que irá compor o valor supostamente exato de θ_{10} (equação 4.8).

Os valores iniciais para o valor dos offsets dos encoders é definido como sendo os valores atualmente fornecidos por eles (θ_{1n}).



Gb) Número de Pontos Medidos - NPmed.

O número de pontos medidos no experimento é de 53. É simulado o experimento com 53 (simulação G1), 20 (G4) e 8 pontos (G5) para posterior comparação.

Gc) Variações entre os Limites das Juntas - Var (%).

A mesa tridimensional, usado como dispositivo de medidas no experimento, permitiu que em média as juntas se movessem dentro de 40% da variação angular possível das juntas, sendo este o valor usado nas simulações do conjunto G.

Gd) Erros nas medidas - E_{max} (mm).

A precisão de leitura da mesa tridimensional é de 0.05 mm. Porém o erro total das medidas estima-se em torno de 0.3 mm, devido a falta de precisão ao se fazer a extremidade da ferramenta tocar o mesmo ponto de referência na mesa tridimensional. Nas simulações o erro (E_{max}) recebe o valor estimado de 0.3 mm (G1, G4 e G5) e menores nas simulações G2 e G3 (0.1 e 0.01 mm respectivamente), permitindo tirar conclusões sobre o efeito da precisão do dispositivo de medidas empregado.

Ge) Ruído nos Encoders - R_{max} (graus).

Os encoders do robô são de alta precisão, e o ruído estima-se ser muito pequeno. Simularemos o experimento supondo não haver ruído.

Gf) Condição de Convergência - δk_{final} .

Para a convergência das medidas lineares na precisão de um décimo de milímetro, o valor para δk_{final} de 0.1 seria suficiente. Porém, baseando nos parâmetros dos offsets escolhe-se o valor de 0.0001.

Assim os valores representativos do experimento são:

Número de pontos medidos	53
Variação angular das juntas	40%
Erro nas medidas	0.3 mm
Ruído nos Encoders	0.0 graus
Condição de convergência	0.001

Tabela 4.2

Valores supostamente exatos (ke)
d1 = 665 mm
d4 = 730 mm
d6 = 160.7 mm
a2 = 500 mm
X0 = -408 mm
Y0 = 1470 mm
θ01 = 28.451 graus
θ02 = 69.028 graus
θ03 = -128.940 graus
θ04 = -59.012 graus
θ05 = 25.002 graus

Tabela 4.3

Valores Iniciais dos Parâmetros Parâmetros Nominais (ko)
d1 = 631.75 mm
d4 = 693.5 mm
d6 = 152.6 mm
a2 = 525 mm
X0 = -428.4 mm
Y0 = 1396.5 mm
θ01 = 28.436 graus
θ02 = 69.043 graus
θ03 = -128.925 graus
θ04 = -59.027 graus
θ05 = 24.996 graus

Tabela 4.4

Baseado na simulação G1, observa-se que com a precisão de 0.3 mm, pode-se obter a precisão de estimativa de 0.1 mm para os parâmetros lineares d_1 , d_4 , d_6 , a_2 , x_0 e y_0 , o que pode ser considerado satisfatório. Já para os valores dos offsets dos encoders que não se comportam tão bem como os parâmetros lineares, não se conseguiu uma estimativa satisfatória. Uma melhor estimativa destes parâmetros dependem de uma maior precisão de medidas como mostram as simulações G2 e G3. Este fato leva a concluir que os valores dos offsets estimados no experimento (capítulo 5) não devem ser considerados.

A redução dos pontos, conforme simulações G4 e G5, diminui a precisão de estimativa de todos os parâmetros, mesmo assim com os seis parâmetros lineares (d_1 , d_4 , d_6 , a_2 , x_0 , y_0) com uma razoável precisão de alguns décimos de milímetros.

A maior conclusão com relação a estas simulações é de que não se pode estimar os pequenos erros de inicialização que são de uma precisão bem maior em relação à precisão do sistema de medidas utilizadas para o processo de estimativa.

CAPÍTULO 5

EXPERIMENTOS E RESULTADOS

Para a execução da parte experimental foi utilizado o robô industrial MANUTEC r3, que possui 6 graus de liberdade, movido por motores de corrente contínua e considerado de alta precisão.

Pelo fato de ser um robô preciso, deve possuir pequenos erros de fabricação. Assim espera-se que os valores dos parâmetros geométricos estimados sejam próximos destes valores de fabricação. Depois de aplicar e validar o procedimento de estimativa para o robô MANUTEC r3, espera-se poder aplicá-lo no manipulador submarino KRAFT. Após então, implementar no seu software de controle, desenvolvido para a automação e programação de suas operações [12], [13], seus parâmetros estimados. Isto deverá resultar em uma boa na exatidão de posicionamento.

Objetivando obter informações sobre a precisão do robô MANUTEC, é realizado durante os experimentos a determinação de sua repetibilidade e exatidão, seguindo a norma ISO/TC 184/SC 2/WG. Para a determinação da exatidão, faz-se necessária a determinação de um referencial de medidas, através de um processo de estimativa. Consequentemente esse processo deverá ser anterior ao cálculo da exatidão e as mesmas medidas são utilizadas para as duas finalidades.

5.1 - DETERMINAÇÃO DA REPETIBILIDADE

Com o objetivo de avaliar o desempenho do robô MANUTEC quanto a sua repetibilidade, foi desenvolvido um experimento, cuja metodologia e fórmulas utilizadas são descritas no anexo C.

• Resultados

i) Repetibilidade Máxima

Posição
$r_{Max} = 7.6 \cdot 10^{-2} \text{ mm}$
Orientação
$\theta_{x_{max}} = 5.2 \cdot 10^{-2} \text{ graus}$
$\theta_{y_{max}} = 1.8 \cdot 10^{-2} \text{ graus}$
$\theta_{z_{max}} = 1.8 \cdot 10^{-2} \text{ graus}$

Tabela 5.1

ii) Repetibilidade Estática

Posição
$r_{Est} = 9.0 \cdot 10^{-2} \text{ mm}$
Orientação
$\theta_x = 2.3 \cdot 10^{-2} \text{ graus}$
$\theta_y = 1.1 \cdot 10^{-2} \text{ graus}$
$\theta_z = 1.1 \cdot 10^{-2} \text{ graus}$

Tabela 5.2

5.2 - ESTIMATIVA DE PARÂMETROS

5.2.1 - Metodologia

Com o objetivo de avaliar o desempenho de instrumentos de medidas, é realizado dois procedimentos. O primeiro com uma mesa tridimensional e o segundo com uma peça de calibração, visando a identificação de parâmetros do mesmo robô.

Apesar de possuir uma precisão de leitura de 0.05 mm, estima-se que o erro de medidas para a mesa tridimensional seja de 0.3 mm. Este erro é devido à falta de precisão ao se fazer tocar a extremidade da ferramenta do robô no mesmo ponto de referência do dispositivo de medidas para todas as medidas. Esta mesa pode realizar medidas em um volume de 1.0 x 1.0 x 1.7 m³ (figura 5.1).

A peça de calibração possui 12 furos cônicos medidos com a precisão de aproximadamente 0.1 com relação do seu referencial (figura 5.2). Porém, devido às limitações geométricas do robô, foram atingidos apenas 8 furos. Estima-se o erro de medida no caso da peça de calibração em 0.4 mm.

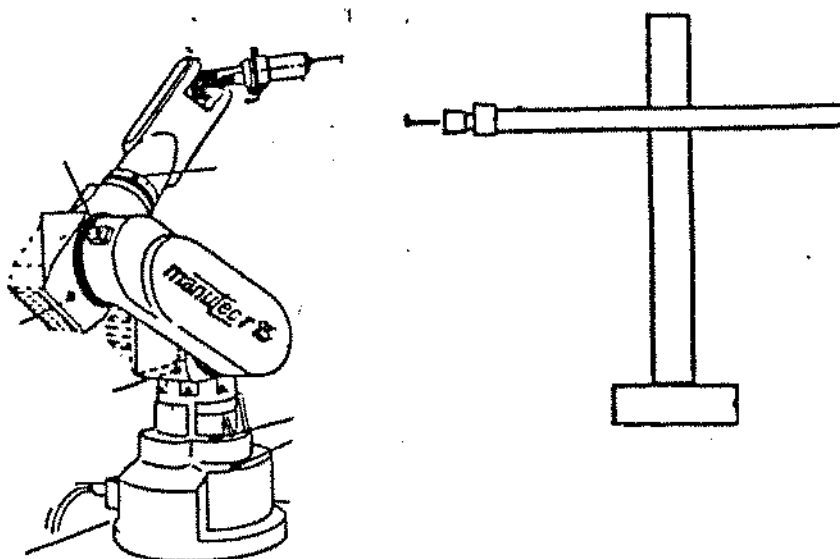


Figura 5.1 - Mesa tridimensional de medidas.

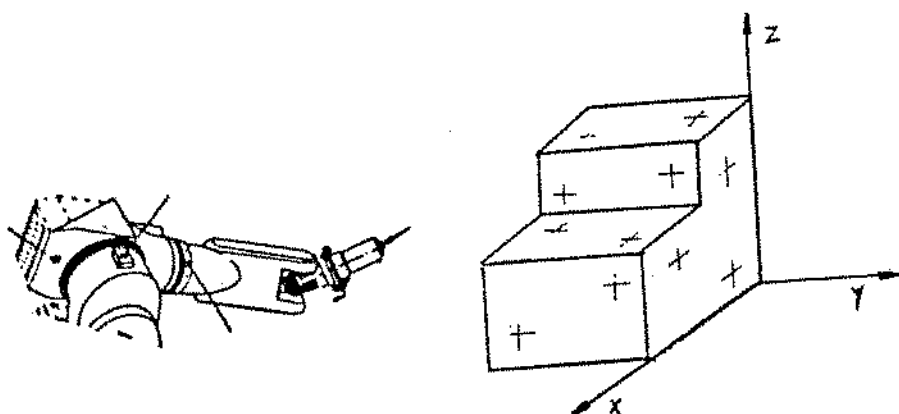


Figura 5.2 - Peça de calibração.

Foram então, obtidas apenas a posição da extremidade de uma ferramenta destinada à calibração, fixa no elemento terminal do robô, em relação a um referencial de medidas (ou de operação), o qual considera-se de mesma orientação do referencial da base. Para eliminar parte da influência do erro nas medidas no experimento com a mesa tridimensional mediu-se três vezes 53 pontos.

Conforme discutido no capítulo 4, apenas os parâmetros d_1 , d_4 , d_6 e a_2 , θ_{10} , θ_{20} , θ_{30} , θ_{40} e θ_{50} , do robô serão considerados para o objetivo de estimativa de parâmetros e calibração.

Como mencionado anteriormente (capítulo 3), devido ao fato de usar somente medidas de posição, não se pode estimar simultaneamente os valores dos parâmetros d_1 e z_0 . Esses parâmetros formam vetores linearmente dependentes na matriz J. Porém na prática, a obtenção de apenas um ou outro não deve criar problemas para fins de programação do robô na maioria dos casos. Assim o vetor de parâmetros a estimar será:

$$\mathbf{k} = (d_1, d_4, d_6, a_2, \theta_{10}, \theta_{20}, \theta_{30}, \theta_{40}, \theta_{50}, x_0, y_0)^t$$

Deve-se ressaltar que no valor de d_6 estimado está somada à dimensão da ferramenta utilizada para as medidas. A ferramenta usada nas medidas com a mesa tridimensional tem dimensão diferente da usada com a peça de calibração. Essas dimensões foram medidas com uma precisão de um décimo de milímetro e para se obter o valor da dimensão o link 6 (último link), deve-se subtrair as dimensões das ferramentas:

$$D_6 = d_6 - D_{ferr}$$

D_{ferr} - dimensão da ferramenta.

D_6 - dimensão do link 6.

d_6 - dimensão total (link e ferramenta) estimada.

5.2.2 - Resultados

I - Mesa tridimensional

Usando o programa de estimativa, desenvolvido através do algoritmo apresentado no capítulo 3 e tendo como entradas os 53 pontos medidos pela mesa tridimensional, foram obtidos como parâmetros exatos ($\mathbf{K_e}$) os valores da tabela 5.3. Dimensão da ferramenta usada:

$D_{ferr} = 160.7 \text{ mm}$

VETOR INICIAL DOS PARÂMETROS ($\bar{\mathbf{k}}_0$)	VETOR DE PARÂMETROS ESTIMADOS ($\bar{\mathbf{k}}_e$)	ABS ($\bar{\mathbf{k}}_e - \bar{\mathbf{k}}_0$)
$d_1 = 665 \text{ mm}$	$d_1 = 663.933 \text{ mm}$	1.067 mm
$d_4 = 730 \text{ mm}$	$d_4 = 730.107 \text{ mm}$	0.107 mm
$d_6 = 160.7 \text{ mm}$	$d_6 = 160.256 \text{ mm}$	0.444 mm
$a_2 = 500 \text{ mm}$	$a_2 = 500.174 \text{ mm}$	0.174 mm
$\theta_{10} = 28.436 \text{ graus}$	$\theta_{10} = 28.4379 \text{ graus}$	0.0019 graus
$\theta_{20} = 69.043 \text{ graus}$	$\theta_{20} = 69.0403 \text{ graus}$	0.0027 graus
$\theta_{30} = -128.925 \text{ graus}$	$\theta_{30} = -128.9287 \text{ graus}$	0.0037 graus
$\theta_{40} = -59.027 \text{ graus}$	$\theta_{40} = -59.0238 \text{ graus}$	0.0042 graus
$\theta_{50} = 24.996 \text{ graus}$	$\theta_{50} = 25.0148 \text{ graus}$	0.0188 graus
$x_0 = -408$	$x_0 = -410.101 \text{ mm}$	2.101 mm
$y_0 = 1470$	$y_0 = 1470.151 \text{ mm}$	0.151 mm

$\delta k_{final} = 0.001$

Número de pontos : 53

Número de iterações : 2

Tabela 5.3

Foram também realizadas estimativas utilizando 20 e 8 pontos aleatoriamente escolhidos dentre os 53 pontos medidos através da mesa tridimensional (Tabelas 5.4 e 5.5).

VECTOR INICIAL DOS PARÂMETROS ($\bar{k}_0$)	VECTOR DE PARÂMETROS ESTIMADOS ($\bar{k}_e$)	ABS ($\bar{k}_e - \bar{k}_0$)
$d_1 = 665 \text{ mm}$	$d_1 = 664.389 \text{ mm}$	0.611 mm
$d_4 = 730 \text{ mm}$	$d_4 = 729.288 \text{ mm}$	0.712 mm
$d_6 = 160.7 \text{ mm}$	$d_6 = 160.449 \text{ mm}$	0.251 mm
$a_2 = 500 \text{ mm}$	$a_2 = 500.270 \text{ mm}$	0.270 mm
$\theta_{10} = 28.436 \text{ graus}$	$\theta_{10} = 28.4394 \text{ graus}$	0.0034 graus
$\theta_{20} = 69.043 \text{ graus}$	$\theta_{20} = 69.0413 \text{ graus}$	0.0017 graus
$\theta_{30} = -128.925 \text{ graus}$	$\theta_{30} = -128.9259 \text{ graus}$	0.0009 graus
$\theta_{40} = -59.027 \text{ graus}$	$\theta_{40} = -59.0165 \text{ graus}$	0.0105 graus
$\theta_{50} = 24.996 \text{ graus}$	$\theta_{50} = 25.0343 \text{ graus}$	0.0383 graus
$x_0 = -408$	$x_0 = -410.009 \text{ mm}$	2.009 mm
$y_0 = 1470$	$y_0 = 1469.437 \text{ mm}$	0.563 mm

$\delta k_{\text{final}} = 0.001$
Número de Pontos : 20
Número de iterações : 2

Tabela 5.4

VETOR INICIAL DOS PARÂMETROS ($\bar{k}_0$)	VETOR DE PARÂMETROS ESTIMADOS ($\bar{k}_e$)	ABS ($\bar{k}_e - \bar{k}_0$)
$d_1 = 665$ mm	$d_1 = 663.542$ mm	1.458 mm
$d_4 = 730$ mm	$d_4 = 729.663$ mm	0.337 mm
$d_6 = 160.7$ mm	$d_6 = 160.749$ mm	0.049 mm
$a_2 = 500$ mm	$a_2 = 500.010$ mm	0.010 mm
$\theta_{10} = 28.436$ graus	$\theta_{10} = 28.4361$ graus	0.0001 graus
$\theta_{20} = 69.043$ graus	$\theta_{20} = 69.0418$ graus	0.0012 graus
$\theta_{30} = -128.925$ graus	$\theta_{30} = -128.9316$ graus	0.0066 graus
$\theta_{40} = -59.027$ graus	$\theta_{40} = -59.0318$ graus	0.0048 graus
$\theta_{50} = 24.996$ graus	$\theta_{50} = 25.0258$ graus	0.0298 graus
$x_0 = -408$	$x_0 = -409.872$ mm	1.872 mm
$y_0 = 1470$	$y_0 = 1470.815$ mm	0.815 mm

$\delta k_{final} = 0.001$
Número de Pontos : 8
Número de iterações : 2

Tabela 5.5

II - Peça de Calibração

Através do mesmo programa utilizado nas estimativas anteriores, porém com apenas 8 pontos obtidos através da peça de calibração temos como resultados os valores dados pela tabela 5.6.

Dimensão da ferramenta utilizada:

$D_{ferr} = 178.5 \text{ mm}$

VETOR INICIAL DOS PARÂMETROS ($\bar{k}_0$)	VETOR DE PARÂMETROS ESTIMADOS ($\bar{k}_e$)	ABS ($\bar{k}_e - \bar{k}_0$)
$d_1 = 665 \text{ mm}$	$d_1 = 666.114 \text{ mm}$	1.114 mm
$d_4 = 730 \text{ mm}$	$d_4 = 730.553 \text{ mm}$	0.553 mm
$d_6 = 175.4 \text{ mm}$	$d_6 = 175.581 \text{ mm}$	0.181 mm
$a_2 = 500 \text{ mm}$	$a_2 = 501.381 \text{ mm}$	1.381 mm
$\theta_{10} = 28.436 \text{ graus}$	$\theta_{10} = 28.4425 \text{ graus}$	0.0065 graus
$\theta_{20} = 69.043 \text{ graus}$	$\theta_{20} = 69.0500 \text{ graus}$	0.0070 graus
$\theta_{30} = -128.925 \text{ graus}$	$\theta_{30} = -128.9324 \text{ graus}$	0.0074 graus
$\theta_{40} = -59.027 \text{ graus}$	$\theta_{40} = -59.0247 \text{ graus}$	0.0023 graus
$\theta_{50} = 24.996 \text{ graus}$	$\theta_{50} = 24.9986 \text{ graus}$	0.0026 graus
$x_0 = -178.8$	$x_0 = -183.141 \text{ mm}$	4.341 mm
$y_0 = 965.5$	$y_0 = 967.441 \text{ mm}$	1.959 mm

$\delta k_{final} = 0.001$

Número de Pontos : 8

Número de iterações : 2

Tabela 5.6

5.3 - DETERMINAÇÃO DA EXATIDÃO DE POSIÇÃO

Na determinação da exatidão do robô MANUTEC r3 foi utilizado para cálculo, os 53 pontos medidos através da mesa tridimensional. As fórmulas utilizadas se encontram no anexo C. Tendo sido estimado a posição do referencial do robô em relação ao referencial de medidas (referindo-se aos parâmetros x_0 e y_0), os valores de posição medidos ($\bar{T}_{med}$) são corrigidos em relação aos valores de posição dados pelo modelo ($\bar{T}_{mod}$).

• Resultados

I - Erro Máximo (E_{max})

$E_{max} = 3.6 \text{ mm}$

II - Erro Médio (E_{med}) e Desvio do Erro (D_{erro})

$E_{med} = 2.8 \text{ mm}$
$D_{erro} = 0.9 \text{ mm}$

5.4 - CONCLUSÕES

Dentre as quatro estimativas realizadas (tabelas 5.3, 5.4, 5.5 e 5.6), deve-se admitir como a mais correta a estimativa 1 (tabela 5.3), pois a mesma foi realizada com um maior número de medidas em um maior volume de trabalho do robô. Mesmo assim, a diminuição em quase 7 vezes do número de medidas (estimativa 1 em relação à estimativa 3 e 4), não fez com que os valores estimados tivessem grande diferença para os parâmetros lineares (d_1 , d_4 , d_6 , a_2 , x_0 e y_0).

Ressalta-se que no valor do parâmetro d_1 estimado, esta incluso o erro na direção z (erro de z_0), o que pode explicar o erro elevado (mais de 1 mm) na estimativa 1 entre o valor do fabricante (nominal) e o estimado.

Como era esperado e ficou demonstrado através das simulações apresentadas, os valores lineares tiveram uma convergência segura. Já para os parâmetros dos offsets, baseado nas simulações e na baixa precisão dos dispositivos de medidas empregado com relação à precisão dos encoders, os valores dos offsets estimados não devem ser considerados.

Os resultados apresentados pela estimativa 4, através da peça de calibração, apesar de permitir variar pouco as juntas (principalmente as primeiras), obteve-se valores próximos em relação às estimativas com a mesa tridimensional. Apesar do grande esforço ao se fazer a ferramenta tocar exatamente os pontos de referência na peça de calibração, com um possível aumento de pontos de medidas, este tipo de dispositivo pode ser uma alternativa de simples construção e pequeno custo para uma calibração onde não é necessária uma grande precisão.

CONCLUSÕES E PERSPECTIVAS

Este trabalho teve como objetivo a implementação de um procedimento para estimativa de parâmetros e calibração para robôs industriais utilizando uma sequência de medidas de posicionamento, modelagem geométrico do robô e um algoritmo que faz uso do critério de mínimos quadrados para se obter convergência.

Os modelos, tanto para o software de estimativa (modelo "modificado" de Denavit-Hartenberg) como para o software de programação off-line (vetores locais) mostraram-se satisfatórios para o propósito de representar parâmetros geométricos. Para o caso de parâmetros não geométricos, seria necessária uma modelagem particular. A modelagem para o erro de posicionamento do robô através do Jacobiano, usado no software de estimativa também se mostrou eficiente.

Com a realização de simples simulações do algoritmo de estimativa proposto, pode-se testar a convergência para os parâmetros geométricos considerados. A partir dessas simulações pode-se explorar a influência de variáveis do procedimento de calibração.

A estimativa usando a mesa tridimensional como dispositivo de medidas possui a vantagem de obter medidas em um maior volume de trabalho (significando maior variação dos ângulos de junta) e obtendo-se parâmetros mais exatos. A peça de calibração resultou em valores próximos, sendo com um maior número de pontos de medidas, ela pode ser uma alternativa simples na falta de um melhor dispositivo de medição.

Diante da validação do método através do robô Manutec R3, o qual já se sabia com alguma precisão os valores de seus parâmetros, pretende-se aplicá-lo ao manipulador Kraft, cujos valores atuais de seus parâmetros são imprecisos, com o objetivo de implantar em seu modelo direto e inverso os valores a estimar. Ao contrário do robô Manutec onde o acesso a alteração de seus parâmetros é complexa, no caso do manipulador será possível testes de exatidão antes (com os parâmetros imprecisos) e após o procedimento de estimativa e calibração (com os parâmetros estimativa implementados no seu controle) e poder avaliar o efeito desse procedimento.

Após uma pesquisa introdutória sobre mecanismos de medidas em águas submarinas chegou-se à duas propostas para implementação de um mecanismo automático para estimativa e correção de posição e orientação entre a base móvel e painel de operações através de um sistema laser. A partir dessas propostas, deve-se direcionar estudos para obter detalhes técnicos visando a melhor solução para o problema exposto.

ANEXO A

ROBÔ MANUTEC r-3

Este robô possui 6 juntas rotacionais e os parâmetros de Denavit-Hartenberg (D.H.) fornecidos pelo seu fabricante são apresentados na tabela A1. Através das matrizes de passagens da tabela A2 (observar a matriz na equação 2.2), pode-se obter o modelo do robô em função de d_1 , d_4 , d_6 , a_2 e de seus ângulos de junta.

JUNTA	θ (gr)	d (mm)	α (gr)	a (mm)	range (graus)	REPRESENTAÇÃO
1	θ_1	665.0	-90.0	0.0	± 165	
2	θ_2	0.0	0.0	500.0	-20/+220	
3	θ_3	0.0	90.0	0.0	-225/+45	
4	θ_4	730.0	-90.0	0.0	± 190	
5	θ_5	0.0	90.0	0.0	± 120	
6	θ_6	100.0	0.0	0.0	± 265	

TABELA A1: Robô MANUTEC r3 - Parametros de D.H.

$A_2^1 = \begin{bmatrix} C_1 & 0 & -S_1 & 0 \\ S_1 & 0 & C_1 & 0 \\ 0 & -1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$A_3^2 = \begin{bmatrix} C_2 & -S_2 & 0 & a_2.C_2 \\ S_2 & C_2 & 0 & a_2.S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
$A_4^3 = \begin{bmatrix} C_3 & 0 & S_3 & 0 \\ S_3 & 0 & -C_3 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$A_5^4 = \begin{bmatrix} C_4 & 0 & -S_4 & 0 \\ S_4 & 0 & C_4 & 0 \\ 0 & -1 & 0 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
$A_6^5 = \begin{bmatrix} C_5 & 0 & S_5 & 0 \\ S_5 & 0 & -C_5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$A_7^6 = \begin{bmatrix} C_6 & -S_6 & 0 & 0 \\ S_6 & C_6 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

TABELA A2: Robô MANUTEC r3 - Matrizes de passagem A_i^{i-1}

C_i: COS (θ_i) , S_i: sin (θ_i)

$$T_6^0 = A_2^1 \cdot A_3^2 \cdot A_4^3 \cdot A_5^4 \cdot A_6^5 \cdot A_7^6 = \begin{bmatrix} \underline{n} & \underline{s} & \underline{a} & \underline{p} \\ & 0 & & 1 \end{bmatrix} \quad (a1)$$

. Orientação

$$\begin{aligned} n_x &= \{ (C_1C_{23}C_4 - S_1S_4) . C_5 - C_1S_{23}S_5 \} . C_6 + \{ -C_1C_{23}S_4 - S_1C_4 \} . S_6 \\ n_y &= \{ (S_1C_{23}C_4 + C_1S_4) . C_5 - S_1S_{23}S_5 \} . C_6 + \{ -S_1C_{23}S_4 + C_1C_4 \} . S_6 \\ n_z &= \{ -S_{23}C_4C_5 - C_{23}S_5 \} . C_6 + S_{23}S_4S_6 \\ \\ s_x &= -\{ (C_1C_{23}C_4 - S_1S_4) . C_5 - C_1S_{23}S_5 \} . S_6 + \{ -C_1C_{23}S_4 - S_1C_4 \} . C_6 \\ s_y &= -\{ (S_1C_{23}C_4 + C_1S_4) . C_5 - S_1S_{23}S_5 \} . S_6 + \{ -S_1C_{23}S_4 + C_1C_4 \} . C_6 \\ s_z &= \{ S_{23}C_4C_5 + C_{23}S_5 \} . S_6 + S_{23}S_4C_6 \\ \\ a_x &= (C_1C_{23}C_4 - S_1S_4) . S_5 + C_1S_{23}C_5 \\ a_y &= (S_1C_{23}C_4 + C_1S_4) . S_5 + S_1S_{23}C_5 \\ a_z &= -S_{23}C_4S_5 + C_{23}C_5 \end{aligned} \tag{a2}$$

. Posição

$$\begin{aligned} p_x &= \{ (C_4S_5d_6) . C_{23} + (C_5d_6 + d_4) . S_{23} + a_2C_2 \} . C_1 - \{ S_4S_5d_6 \} . S_1 \\ p_y &= \{ (C_4S_5d_6) . C_{23} + (C_5d_6 + d_4) . S_{23} + a_2C_2 \} . S_1 + \{ S_4S_5d_6 \} . C_1 \\ p_z &= - (C_4S_5d_6) . S_{23} + (C_5d_6 + d_4) . C_{23} - a_2S_2 + d_1 \end{aligned} \tag{a3}$$

JUNTA	1	2	3	4	5	6
IMP/GRAU	700	1400	400	660	528	660

TABELA A3 : RELAÇÃO ENTRE IMPULSÕES DOS ENCODERS e 1 GRAU

ANEXO B

As tabelas deste anexo são referentes às simulações do capítulo 4.

• Definições (conforme capítulo 4):

Per - porcentagem de erros nos valores do vetor inicial de parâmetros (equações 4.9a e 4.9b).

NPmed - número de pontos medidos para o processo de estimativa.

Var - porcentagem de variação entre limites de junta do robô.

E_{max} - erro máximo de medidas suposto.

R_{max} - ruído máximo na leitura dos encoders suposto.

δk_{final} - condição para convergência (ver algoritmo no capítulo 3).

k_e - valores do vetor de parâmetros supostamente exatos.

k_0 - valores do vetor inicial de parâmetros.

k - valores dos parâmetros estimados.

PI - Precisão de estimativa.

Simulações A

* Influência da porcentagem de erro no vetor inicial - Per (%).

Per (%)	NPmed	Var (%)	E _{max} (mm)	R _{max} (graus)	δk _{final}
Per	30	50	0.1	0.01	0.001

SIMULACAO	A1	A2	A3	A4	A5
Per (%)	1	50	500	2500	3000

SIMULAÇÃO A1 : Per = 1 %

PARAMETRO	Vetor $\overline{k_e}$	Vetor $\overline{k_0}$	Vetor $\overline{k}$	PI (%)
d ₁ (mm)	665	671.65	664.66	0.052
d ₄ (mm)	730	722.70	729.95	0.007
d ₆ (mm)	160.7	162.31	160.63	0.041
a ₂ (mm)	500	505.00	499.87	0.026
θ ₁₀ (gr)	29.436	29.446	29.437	0.142
θ ₂₀ (gr)	68.043	68.053	68.009	3.402
θ ₃₀ (gr)	-126.925	-126.945	-126.963	1.892
θ ₄₀ (gr)	-58.027	-58.037	-58.006	2.117
θ ₅₀ (gr)	22.996	22.976	22.968	1.404
x ₀ (mm)	-408	-612.00	-408.02	0.005
y ₀ (mm)	1470	1484.70	1469.96	0.003

Números de iterações: 3

SIMULAÇÃO A2 : Per = 50 %

PARAMETRO	Vetor $\overline{k_e}$	Vetor $\overline{k_0}$	Vetor $\overline{k}$	PI (%)
d1 (mm)	665	997.50	664.66	0.052
d4 (mm)	730	365.00	729.95	0.007
d6 (mm)	160.7	80.35	160.63	0.041
a2 (mm)	500	750.00	499.87	0.026
θ_{10} (gr)	29.436	29.936	29.437	0.142
θ_{20} (gr)	68.043	68.543	68.009	3.402
θ_{30} (gr)	-126.925	-127.925	-126.963	1.892
θ_{40} (gr)	-58.027	-57.527	-58.006	2.117
θ_{50} (gr)	22.996	21.996	22.968	1.404
x0 (mm)	-408	-612.00	-408.02	0.005
y0 (mm)	1470	2205.00	1469.96	0.003

Números de iterações: 4

SIMULAÇÃO A3 : Per = 500 %

PARAMETRO	Vetor $\overline{k_e}$	Vetor $\overline{k_0}$	Vetor $\overline{k}$	PI (%)
d1 (mm)	665	3990.00	664.66	0.052
d4 (mm)	730	-2920.00	729.95	0.007
d6 (mm)	160.7	-642.80	160.63	0.041
a2 (mm)	500	3000.00	499.87	0.026
θ_{10} (gr)	29.436	34.436	29.437	0.142
θ_{20} (gr)	68.043	73.043	68.009	3.402
θ_{30} (gr)	-126.925	-136.945	-126.963	1.892
θ_{40} (gr)	-58.027	-63.027	-58.006	2.117
θ_{50} (gr)	22.996	12.996	22.968	1.404
x0 (mm)	-408	-2448.00	-408.02	0.005
y0 (mm)	1470	8820.00	1469.96	0.003

Números de iterações: 5

SIMULAÇÃO A4 : Per = 2500 %

PARAMETRO	Vetor $\vec{k_e}$	Vetor $\vec{k_0}$	Vetor $\vec{k}$	PI (%)
d1 (mm)	665	17290.00	664.66	0.052
d4 (mm)	730	-17520.00	729.95	0.007
d6 (mm)	160.7	-3856.80	160.63	0.041
a2 (mm)	500	13000.00	499.87	0.026
θ_{10} (gr)	29.436	54.436	29.437	0.142
θ_{20} (gr)	68.043	93.043	68.009	3.402
θ_{30} (gr)	-126.925	-176.925	-126.963	1.892
θ_{40} (gr)	-58.027	-83.027	-58.006	2.117
θ_{50} (gr)	22.996	-27.004	22.968	1.404
x0 (mm)	-408	-10608.00	-408.02	0.005
y0 (mm)	1470	38220.00	1469.96	0.003

Números de iterações: 9

SIMULAÇÃO A5 : Per = 3000 %

PARAMETRO	Vetor $\vec{k_e}$	Vetor $\vec{k_0}$	Vetor $\vec{k}$	PI (%)
d1 (mm)	665	20615.00	664.66	0.052
d4 (mm)	730	-21170.00	729.95	0.007
d6 (mm)	160.7	-4660.30	160.63	0.041
a2 (mm)	500	15500.00	499.87	0.026
θ_{10} (gr)	29.436	59.436	29.437	0.142
θ_{20} (gr)	68.043	98.043	68.009	144003.0
θ_{30} (gr)	-126.925	-186.925	-1567.956	72001.7
θ_{40} (gr)	-58.027	-88.027	-1496.061	2.117
θ_{50} (gr)	22.996	-37.004	22.968	1.404
x0 (mm)	-408	-12648.00	-408.02	0.005
y0 (mm)	1470	45570.00	1469.96	0.003

Números de iterações: 11

Simulações B

Influência do número de pontos - NPmed:

Per (%)	NPmed	Var (%)	E _{max} (mm)	R _{max} (graus)	δk _{final}
50	NPmed	50	0.1	0.01	0.001

SIMULACAO	B1	B2	B3	B4
NPmed	60	20	10	4

SIMULAÇÃO B1 : NPmed = 60 pontos

PARAMETRO	Vetor $\overline{k_e}$	Vetor $\overline{k_0}$	Vetor $\overline{k}$	PI (%)
d ₁ (mm)	665	0.0	665.15	0.022
d ₄ (mm)	730	0.0	730.03	0.004
d ₆ (mm)	160.7	321.4	160.74	0.028
a ₂ (mm)	500	0.0	500.05	0.016
θ ₁₀ (gr)	29.436	30.436	29.437	0.169
θ ₂₀ (gr)	68.043	67.043	68.048	1.286
θ ₃₀ (gr)	-126.925	-128.925	-126.923	0.555
θ ₄₀ (gr)	-58.027	-57.027	-58.035	0.818
θ ₅₀ (gr)	22.996	20.976	23.037	1.058
x ₀ (mm)	-408	-816.0	-408.02	0.004
y ₀ (mm)	1470	2940.0	1469.96	0.000

Números de iterações: 4

SIMULAÇÃO B2 : Npmed = 20 pontos

PARAMETRO	Vetor $\overline{k_e}$	Vetor $\overline{k_0}$	Vetor $\overline{k}$	PI (%)
d1 (mm)	665	0.0	665.06	0.009
d4 (mm)	730	0.0	730.07	0.009
d6 (mm)	160.7	321.4	160.74	0.027
a2 (mm)	500	0.0	500.05	0.010
θ_{10} (gr)	29.436	30.436	29.437	0.006
θ_{20} (gr)	68.043	67.043	68.048	0.521
θ_{30} (gr)	-126.925	-128.925	-126.923	0.081
θ_{40} (gr)	-58.027	-57.027	-58.035	0.829
θ_{50} (gr)	22.996	20.976	23.037	2.052
x0 (mm)	-408	-816.0	-408.02	0.004
y0 (mm)	1470	2940.0	1469.96	0.000

Números de iterações: 4

Simulação B3 : NPmed = 10 pontos

PARAMETRO	Vetor $\overline{k_e}$	Vetor $\overline{k_0}$	Vetor $\overline{k}$	PI (%)
d1 (mm)	665	0.0	665.38	0.057
d4 (mm)	730	0.0	730.11	0.016
d6 (mm)	160.7	321.4	160.78	0.052
a2 (mm)	500	0.0	500.16	0.032
θ_{10} (gr)	29.436	30.436	29.441	0.525
θ_{20} (gr)	68.043	67.043	68.084	4.050
θ_{30} (gr)	-126.925	-128.925	-126.887	1.877
θ_{40} (gr)	-58.027	-57.027	-58.029	0.212
θ_{50} (gr)	22.996	20.976	23.042	2.299
x0 (mm)	-408	-816.0	-407.90	0.024
y0 (mm)	1470	2940.0	1469.97	0.002

Números de iterações: 5

Simulação B4 : NPmed = 4 pontos

PARAMETRO	Vetor $\overline{k_e}$	Vetor $\overline{k_0}$	Vetor $\overline{k}$	PI (%)
d1 (mm)	665	0.0	670.44	0.818
d4 (mm)	730	0.0	730.50	0.069
d6 (mm)	160.7	321.4	160.78	0.020
a2 (mm)	500	0.0	502.52	0.504
θ_{10} (gr)	29.436	30.436	29.500	6.432
θ_{20} (gr)	68.043	67.043	68.613	56.965
θ_{30} (gr)	-126.925	-128.925	-126.365	28.009
θ_{40} (gr)	-58.027	-57.027	-58.126	9.902
θ_{50} (gr)	22.996	20.976	23.202	10.276
x0 (mm)	-408	-816.0	-407.17	0.203
y0 (mm)	1470	2940.0	1469.18	0.055

Números de iterações: 5

Simulações C

Influência da variação nos ângulos de junta - Var (%)

Per (%)	NPmed	Var (%)	E _{max} (mm)	R _{max} (graus)	δk_{final}
50	30	Var	0.1	0.01	0.001

SIMULACAO	C1	C2	C3	C4
Var (%)	1	10	20	100

SIMULAÇÃO C1 : Var = 1%

PARAMETRO	Vetor $\overline{k_e}$	Vetor $\overline{k_0}$	Vetor $\overline{k}$	PI (%)
d1 (mm)	665	1330.0	676.49	1.728
d4 (mm)	730	0.0	728.77	0.169
d6 (mm)	160.7	0.0	158.06	1.642
a2 (mm)	500	1000.0	500.16	1.220
θ_{10} (gr)	29.436	30.436	29.441	4.792
θ_{20} (gr)	68.043	67.043	68.084	116.015
θ_{30} (gr)	-126.925	-128.925	-126.887	53.377
θ_{40} (gr)	-58.027	-59.027	-58.029	8.679
θ_{50} (gr)	22.996	20.976	23.042	32.423
X0 (mm)	-408	0.0	-407.90	0.009
y0 (mm)	1470	2940.0	1469.97	0.091

Números de iterações: 5

SIMULAÇÃO C2 : Var = 10 %

PARAMETRO	Vetor $\overline{k_e}$	Vetor $\overline{k_0}$	Vetor $\overline{k}$	PI (%)
d1 (mm)	665	1330.0	666.09	0.164
d4 (mm)	730	0.0	729.82	0.024
d6 (mm)	160.7	0.0	160.36	0.212
a2 (mm)	500	1000.0	500.59	0.117
θ_{10} (gr)	29.436	30.436	29.444	0.792
θ_{20} (gr)	68.043	67.043	68.165	11.229
θ_{30} (gr)	-126.925	-128.925	-126.826	4.948
θ_{40} (gr)	-58.027	-59.027	-58.044	1.682
θ_{50} (gr)	22.996	20.976	22.910	4.290
X0 (mm)	-408	0.0	-408.02	0.004
y0 (mm)	1470	2940.0	1469.86	0.010

Números de iterações: 4

SIMULAÇÃO C3 : Var = 20 %

PARAMETRO	Vetor $\overline{k_e}$	Vetor $\overline{k_0}$	Vetor $\overline{k}$	PI (%)
d1 (mm)	665	1330.0	665.51	0.076
d4 (mm)	730	0.0	729.91	0.012
d6 (mm)	160.7	0.0	160.43	0.117
a2 (mm)	500	1000.0	500.28	0.056
θ_{10} (gr)	29.436	30.436	29.442	0.552
θ_{20} (gr)	68.043	67.043	68.095	5.242
θ_{30} (gr)	-126.925	-128.925	-126.881	2.208
θ_{40} (gr)	-58.027	-59.027	-58.053	2.350
θ_{50} (gr)	22.996	20.976	22.956	2.012
x0 (mm)	-408	0.0	-408.13	0.003
y0 (mm)	1470	2940.0	1469.93	0.005

Números de iterações: 4

SIMULAÇÃO C4 : Var = 100 %

PARAMETRO	Vetor $\overline{k_e}$	Vetor $\overline{k_0}$	Vetor $\overline{k}$	PI (%)
d1 (mm)	665	1330.0	664.88	0.019
d4 (mm)	730	0.0	730.03	0.003
d6 (mm)	160.7	0.0	160.51	0.005
a2 (mm)	500	1000.0	499.97	0.006
θ_{10} (gr)	29.436	30.436	29.435	0.080
θ_{20} (gr)	68.043	67.043	68.032	1.094
θ_{30} (gr)	-126.925	-128.925	-126.938	0.664
θ_{40} (gr)	-58.027	-59.027	-58.027	0.011
θ_{50} (gr)	22.996	20.976	23.004	0.380
x0 (mm)	-408	0.0	-408.00	0.000
y0 (mm)	1470	2940.0	1469.99	0.000

Números de iterações: 4

Simulações D

Influência dos erros de medidas - Emax (mm)

Per (%)	NPmed	Var (%)	Emax (mm)	Rmax (graus)	δk_{final}
50	30	50	Emax	0.01	0.001

SIMULACAO	D1	D2	D3	D4
Emax (mm)	0.01	0.1	1	10

SIMULAÇÃO D1 : Emax = 0.01 mm

PARAMETRO	Vetor $\vec{k_e}$	Vetor $\vec{k_0}$	Vetor $\vec{k}$	PI (%)
d1 (mm)	665	1330.0	665.03	0.005
d4 (mm)	730	0.0	730.02	0.002
d6 (mm)	160.7	0.0	160.67	0.017
a2 (mm)	500	1000.0	500.02	0.005
θ_{10} (gr)	29.436	30.436	29.436	0.007
θ_{20} (gr)	68.043	67.043	68.048	0.503
θ_{30} (gr)	-126.925	-128.925	-126.922	0.163
θ_{40} (gr)	-58.027	-59.027	-58.017	0.976
θ_{50} (gr)	22.996	20.976	23.012	0.788
x0 (mm)	-408	0.0	-407.98	0.003
y0 (mm)	1470	2940.0	1469.99	0.001

Números de iterações: 4

SIMULAÇÃO D2 : Emax = 0.1 mm

PARAMETRO	Vetor $\overline{k_e}$	Vetor $\overline{k_0}$	Vetor $\overline{k}$	PI (%)
d1 (mm)	665	1330.0	665.06	0.009
d4 (mm)	730	0.0	730.03	0.004
d6 (mm)	160.7	0.0	160.65	0.029
a2 (mm)	500	1000.0	500.06	0.011
θ_{10} (gr)	29.436	30.436	29.438	0.183
θ_{20} (gr)	68.043	67.043	68.050	0.690
θ_{30} (gr)	-126.925	-128.925	-126.921	0.196
θ_{40} (gr)	-58.027	-59.027	-58.040	1.347
θ_{50} (gr)	22.996	20.976	23.010	0.688
x0 (mm)	-408	0.0	-408.00	0.016
y0 (mm)	1470	2940.0	1469.98	0.001

Números de iterações: 4

SIMULAÇÃO D3 : Emax = 1 mm

PARAMETRO	Vetor $\overline{k_e}$	Vetor $\overline{k_0}$	Vetor $\overline{k}$	PI (%)
d1 (mm)	665	1330.0	665.33	0.049
d4 (mm)	730	0.0	730.13	0.018
d6 (mm)	160.7	0.0	160.45	0.153
a2 (mm)	500	1000.0	500.39	0.079
θ_{10} (gr)	29.436	30.436	29.457	2.078
θ_{20} (gr)	68.043	67.043	68.069	2.555
θ_{30} (gr)	-126.925	-128.925	-126.915	0.518
θ_{40} (gr)	-58.027	-59.027	-58.273	24.611
θ_{50} (gr)	22.996	20.976	22.990	0.314
x0 (mm)	-408	0.0	-408.20	0.048
y0 (mm)	1470	2940.0	1469.87	0.009

Números de iterações: 4

SIMULAÇÃO D4 : Emax = 10 mm

PARAMETRO	Vetor $\overline{k_e}$	Vetor $\overline{k_0}$	Vetor $\overline{k}$	PI (%)
d1 (mm)	665	1330.0	667.92	0.440
d4 (mm)	730	0.0	731.22	0.167
d6 (mm)	160.7	0.0	158.57	1.327
a2 (mm)	500	1000.0	503.73	0.746
θ_{10} (gr)	29.436	30.436	29.646	20.956
θ_{20} (gr)	68.043	67.043	68.247	20.425
θ_{30} (gr)	-126.925	-128.925	-126.859	3.320
θ_{40} (gr)	-58.027	-59.027	-60.627	259.973
θ_{50} (gr)	22.996	20.976	22.785	10.554
x0 (mm)	-408	0.0	-410.10	0.515
y0 (mm)	1470	2940.0	1468.76	0.084

Números de iterações: 4

Simulações E

Influência ruído nas medidas dos encoders - Rmax (graus)

Per (%)	NPmed	Var (%)	Emax (mm)	Rmax (graus)	δk_{final}
50	30	50	0.1	Rmax	0.001

SIMULACAO	E1	E2	E3	E4
Rmax (GRAUS)	0.001	0.01	0.1	1

SIMULAÇÃO E1 : Rmax = 0.001 graus

PARAMETRO	Vetor $\vec{k}_e$	Vetor $\vec{k}_0$	Vetor $\vec{k}$	PI (%)
d1 (mm)	665	1330.0	665.03	0.005
d4 (mm)	730	0.0	730.01	0.002
d6 (mm)	160.7	0.0	160.68	0.015
a2 (mm)	500	1000.0	500.04	0.008
θ_{10} (gr)	29.436	30.436	29.438	0.208
θ_{20} (gr)	68.043	67.043	68.046	0.256
θ_{30} (gr)	-126.925	-128.925	-126.924	0.052
θ_{40} (gr)	-58.027	-59.027	-58.052	2.457
θ_{50} (gr)	22.996	20.976	22.995	0.031
x0 (mm)	-408	0.0	-408.02	0.005
y0 (mm)	1470	2940.0	1469.99	0.001

Números de iterações: 4

SIMULAÇÃO E2 : Rmax = 0.01 graus

PARAMETRO	Vetor $\vec{k}_e$	Vetor $\vec{k}_0$	Vetor $\vec{k}$	PI (%)
d1 (mm)	665	1330.0	665.06	0.009
d4 (mm)	730	0.0	730.03	0.004
d6 (mm)	160.7	0.0	160.65	0.029
a2 (mm)	500	1000.0	500.06	0.011
θ_{10} (gr)	29.436	30.436	29.438	0.1826
θ_{20} (gr)	68.043	67.043	68.050	0.690
θ_{30} (gr)	-126.925	-128.925	-126.921	0.196
θ_{40} (gr)	-58.027	-59.027	-58.060	1.347
θ_{50} (gr)	22.996	20.976	23.009	0.688
x0 (mm)	-408	0.0	-408.01	0.002
y0 (mm)	1470	2940.0	1469.98	0.001

Números de iterações: 4

SIMULAÇÃO E3 : Rmax = 0.1 graus

PARAMETRO	Vetor $\vec{k}_e$	Vetor $\vec{k}_0$	Vetor $\vec{k}$	PI (%)
d1 (mm)	665	1330.0	665.30	0.050
d4 (mm)	730	0.0	730.18	0.024
d6 (mm)	160.7	0.0	160.43	0.167
a2 (mm)	500	1000.0	500.22	0.044
θ_{10} (gr)	29.436	30.436	29.435	0.079
θ_{20} (gr)	68.043	67.043	68.091	4.768
θ_{30} (gr)	-126.925	-128.925	-126.895	1.500
θ_{40} (gr)	-58.027	-59.027	-57.928	9.860
θ_{50} (gr)	22.996	20.976	23.155	7.927
x0 (mm)	-408	0.0	-407.87	0.031
y0 (mm)	1470	2940.0	1469.90	0.007

Números de iterações: 4

Simulação E4 : Rmax = 1 graus

PARAMETRO	Vetor $\vec{k}_e$	Vetor $\vec{k}_0$	Vetor $\vec{k}$	PI (%)
d1 (mm)	665	1330.0	665.15	0.023
d4 (mm)	730	0.0	731.78	0.243
d6 (mm)	160.7	0.0	158.45	1.395
a2 (mm)	500	1000.0	500.54	0.108
θ_{10} (gr)	29.436	30.436	29.399	3.662
θ_{20} (gr)	68.043	67.043	68.251	20.765
θ_{30} (gr)	-126.925	-128.925	-126.882	2.137
θ_{40} (gr)	-58.027	-59.027	-56.703	132.356
θ_{50} (gr)	22.996	20.976	24.693	84.868
x0 (mm)	-408	0.0	-406.48	0.372
y0 (mm)	1470	2940.0	1469.22	0.053

Números de iterações: 5

Simulações F

Influência do incremento final do algoritmo - δk_{final}

Per (%)	NPmed	Var (%)	E _{max} (mm)	R _{max} (graus)	δk_{final}
50	30	50	0.1	0.01	δk_{final}

SIMULACAO	E1	E2	E3	E4
δk_{final}	0.00001	0.001	0.1	10

SIMULAÇÃO F1 : $\delta k_{final} = 0.00001$

PARAMETRO	Vetor $\vec{k}_e$	Vetor $\vec{k}_0$	Vetor $\vec{k}$	PI (%)
d ₁ (mm)	665	1330.0	665.06	0.009
d ₄ (mm)	730	0.0	730.03	0.004
d ₆ (mm)	160.7	0.0	160.65	0.029
a ₂ (mm)	500	1000.0	500.06	0.011
θ ₁₀ (gr)	29.436	30.436	29.438	0.183
θ ₂₀ (gr)	68.043	67.043	68.050	0.690
θ ₃₀ (gr)	-126.925	-128.925	-126.921	0.196
θ ₄₀ (gr)	-58.027	-59.027	-58.040	1.347
θ ₅₀ (gr)	22.996	20.976	23.010	0.688
x ₀ (mm)	-408	0.0	-408.01	0.002
y ₀ (mm)	1470	2940.0	1469.98	0.001

Números de iterações: 5

SIMULAÇÃO F2 : $\delta k_{final} = 0.001$

PARAMETRO	Vetor $\overline{k_e}$	Vetor $\overline{k_0}$	Vetor $\overline{k}$	PI (%)
d1 (mm)	665	1330.0	665.06	0.009
d4 (mm)	730	0.0	730.03	0.004
d6 (mm)	160.7	0.0	160.65	0.029
a2 (mm)	500	1000.0	500.06	0.011
θ_{10} (gr)	29.436	30.436	29.438	0.183
θ_{20} (gr)	68.043	67.043	68.050	0.690
θ_{30} (gr)	-126.925	-128.925	-126.921	0.196
θ_{40} (gr)	-58.027	-59.027	-58.040	1.347
θ_{50} (gr)	22.996	20.976	23.010	0.688
x0 (mm)	-408	0.0	-408.01	0.002
y0 (mm)	1470	2940.0	1469.98	0.001

Números de iterações: 4

SIMULAÇÃO F3 : $\delta k_{final} = 0.1$

PARAMETRO	Vetor $\overline{k_e}$	Vetor $\overline{k_0}$	Vetor $\overline{k}$	PI (%)
d1 (mm)	665	1330.0	665.06	0.009
d4 (mm)	730	0.0	730.03	0.004
d6 (mm)	160.7	0.0	160.65	0.029
a2 (mm)	500	1000.0	500.06	0.011
θ_{10} (gr)	29.436	30.436	29.438	0.183
θ_{20} (gr)	68.043	67.043	68.050	0.690
θ_{30} (gr)	-126.925	-128.925	-126.921	0.196
θ_{40} (gr)	-58.027	-59.027	-58.040	1.347
θ_{50} (gr)	22.996	20.976	23.010	0.688
x0 (mm)	-408	0.0	-408.01	0.002
y0 (mm)	1470	2940.0	1469.98	0.001

Números de iterações: 4

SIMULAÇÃO F4 : $\delta k_{final} = 10$

PARAMETRO	Vetor $\overline{k_e}$	Vetor $\overline{k_0}$	Vetor $\overline{k}$	PI (%)
d1 (mm)	665	1330.0	665.06	0.009
d4 (mm)	730	0.0	730.03	0.003
d6 (mm)	160.7	0.0	160.65	0.029
a2 (mm)	500	1000.0	500.06	0.011
θ_{10} (gr)	29.436	30.436	29.438	0.182
θ_{20} (gr)	68.043	67.043	68.050	0.693
θ_{30} (gr)	-126.925	-128.925	-126.921	0.197
θ_{40} (gr)	-58.027	-59.027	-58.040	1.361
θ_{50} (gr)	22.996	20.976	23.010	0.682
x0 (mm)	-408	0.0	-408.01	0.002
y0 (mm)	1470	2940.0	1469.98	0.001

Números de iterações: 3

Simulações G

Estas tabelas correspondem às simulações do conjunto G representando o experimento prático descrito do capítulo 5 (ver capítulo 4).

SIMULAÇÃO G1

NPmed	Var (%)	E _{max} (mm)	R _{max} (graus)	δk _{final}
53	40	0.3	0.00	0.001

PARAMETRO	Vetor $\overline{k_e}$	Vetor $\overline{k_0}$	Vetor $\overline{k}$	abs ($\overline{k_e} - \overline{k}$)
d ₁ (mm)	665	631.75	665.03	0.03
d ₄ (mm)	730	693.5	730.07	0.07
d ₆ (mm)	160.7	152.7	160.67	0.03
a ₂ (mm)	500	525.0	500.04	0.04
θ ₁₀ (gr)	28.451	28.436	28.4521	0.00112
θ ₂₀ (gr)	69.028	69.043	69.0312	0.00317
θ ₃₀ (gr)	-128.940	-128.925	-128.9391	0.00087
θ ₄₀ (gr)	-59.042	-59.027	-59.0620	0.02003
θ ₅₀ (gr)	24.981	24.996	25.0095	0.02854
x ₀ (mm)	-408	-428.4	-408.05	0.05
y ₀ (mm)	1470	1396.5	1469.99	0.01

Números de iterações: 3

SIMULAÇÃO G2

NPmed	Var (%)	E _{max} (mm)	R _{max} (graus)	δk_{final}
53	40	0.1	0.00	0.0001

PARAMETRO	Vetor $\overline{k_e}$	Vetor $\overline{k_0}$	Vetor $\overline{k}$	abs ($\overline{k_e}-\overline{k}$)
d ₁ (mm)	665	631.75	665.03	0.03
d ₄ (mm)	730	693.5	730.03	0.03
d ₆ (mm)	160.7	152.7	160.69	0.01
a ₂ (mm)	500	525.0	500.06	0.06
θ ₁₀ (gr)	28.451	28.436	28.4511	0.00004
θ ₂₀ (gr)	69.028	69.043	69.0281	0.00011
θ ₃₀ (gr)	-128.940	-128.925	-128.9400	0.00003
θ ₄₀ (gr)	-59.042	-59.027	-59.0421	0.00067
θ ₅₀ (gr)	24.981	24.996	24.9811	0.00095
x ₀ (mm)	-408	-428.4	-408.00	0.00
y ₀ (mm)	1470	1396.5	1470.00	0.00

Números de iterações: 3

SIMULAÇÃO G3

NPmed	Var (%)	E _{max} (mm)	R _{max} (graus)	δ _{k final}
53	40	0.01	0.00	0.0001

PARAMETRO	Vetor $\overline{k_e}$	Vetor $\overline{k_0}$	Vetor $\overline{k}$	abs($\overline{k_e}-\overline{k}$)
d ₁ (mm)	665	631.75	665.00	0.00
d ₄ (mm)	730	693.5	730.03	0.03
d ₆ (mm)	160.7	152.7	160.70	0.00
a ₂ (mm)	500	525.0	500.00	0.00
θ ₁₀ (gr)	28.451	28.436	28.4511	0.00004
θ ₂₀ (gr)	69.028	69.043	69.0281	0.00011
θ ₃₀ (gr)	-128.940	-128.925	-128.9400	0.00003
θ ₄₀ (gr)	-59.042	-59.027	-59.0421	0.00067
θ ₅₀ (gr)	24.981	24.996	24.9811	0.00095
x ₀ (mm)	-408	-428.4	-408.00	0.00
y ₀ (mm)	1470	1396.5	1470.00	0.00

Números de iterações: 3

SIMULAÇÃO G4

NPmed	Var (%)	E _{max} (mm)	R _{max} (graus)	δk _{final}
20	40	0.3	0.00	0.001

PARAMETRO	Vetor $\vec{k_e}$	Vetor $\vec{k_0}$	Vetor $\vec{k}$	abs ($\vec{k_e}-\vec{k}$)
d ₁ (mm)	665	631.75	664.84	0.16
d ₄ (mm)	730	693.5	729.93	0.07
d ₆ (mm)	160.7	152.7	160.55	0.15
a ₂ (mm)	500	525.0	500.00	0.00
θ ₁₀ (gr)	28.451	28.436	28.4625	0.01145
θ ₂₀ (gr)	69.028	69.043	69.0078	0.02018
θ ₃₀ (gr)	-128.940	-128.925	-128.9669	0.02694
θ ₄₀ (gr)	-59.042	-59.027	-59.1744	0.13235
θ ₅₀ (gr)	24.981	24.996	24.9352	0.04582
X ₀ (mm)	-408	-428.4	-408.06	0.06
Y ₀ (mm)	1470	1396.5	1469.99	0.01

Números de iterações: 3

SIMULAÇÃO G5

NPmed	Var (%)	E _{max} (mm)	R _{max} (graus)	δk _{final}
8	40	0.3	0.00	0.001

PARAMETRO	Vetor $\overline{k_e}$	Vetor $\overline{k_0}$	Vetor $\overline{k}$	abs ($\overline{k_e}-\overline{k}$)
d ₁ (mm)	665	631.75	665.25	0.25
d ₄ (mm)	730	693.5	729.88	0.12
d ₆ (mm)	160.7	152.7	160.43	0.27
a ₂ (mm)	500	525.0	500.18	0.18
θ ₁₀ (gr)	28.451	28.436	28.4828	0.03180
θ ₂₀ (gr)	69.028	69.043	69.0292	0.00119
θ ₃₀ (gr)	-128.940	-128.925	-128.9437	0.00373
θ ₄₀ (gr)	-59.042	-59.027	-59.2558	0.21376
θ ₅₀ (gr)	24.981	24.996	24.9527	0.02835
x ₀ (mm)	-408	-428.4	-408.10	0.10
y ₀ (mm)	1470	1396.5	1469.81	0.19

Números de iterações: 3

ANEXO C

c.1) Medidas de Repetibilidade

c.1.1) Metodologia

As medidas para a repetibilidade devem ser realizadas em relação a um sistema de referência de eixos paralelos aos eixos do robô (figura C.1).

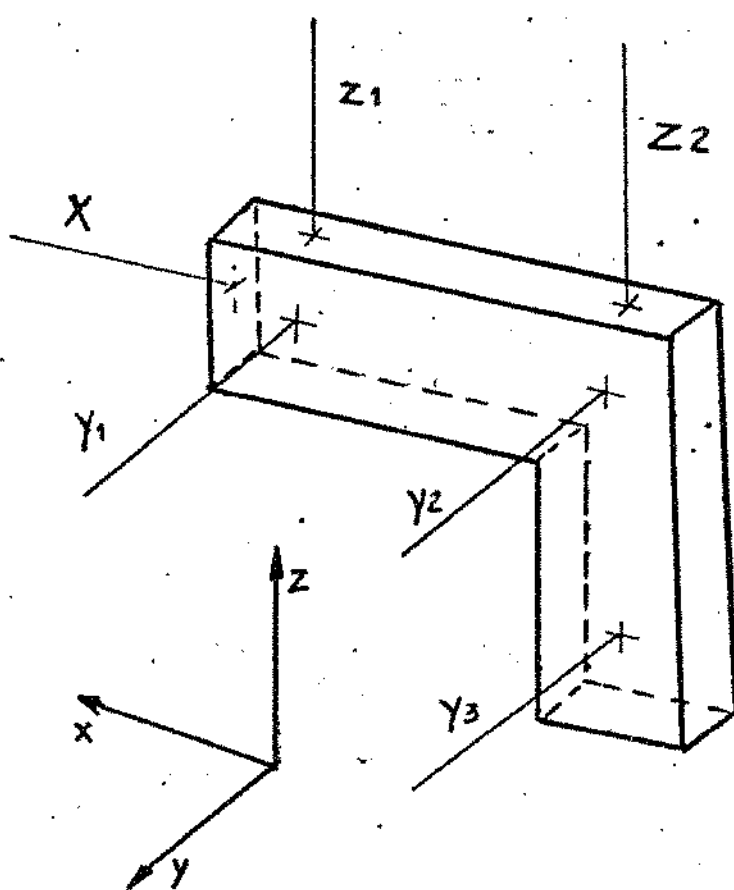


figura C.1 - Esquadro e posições dos eixos de coordenadas e medidas.

O material utilizado foi:

- 1) 6 relógios comparadores (precisão de centésimos de milímetros);
- 2) 1 esquadro, fixado no elemento terminal do robô (figura C.1).

Posicionou-se e orientou-se o elemento terminal com o esquadro em um ponto $P_c = (x_p, y_p, z_p, \theta_{xp}, \theta_{yp}, \theta_{zp})$.

Os relógios comparadores foram posicionados de modo a medirem as variações de posição ao longo dos eixos X, Y_1, Y_2, Y_3, Z_1 e Z_2 , conforme esquema da figura C.1.

Zerou-se os relógios e repetiu-se a posição através da mesma trajetória 30 vezes e anotou-se as variações $dx, dy_1, dy_2, dy_3, dz_1$ e dz_2 dos relógios.

c.1.2) Cálculos

Definições:

C_1 - distância entre os eixos Y_3 e Y_2 .

C_2 - distância entre os eixos Z_2 e Z_1 .

C_3 - distância entre os eixos Y_2 e Y_1 .

dx_i - variação de posição em relação ao eixo X do robô.

$dy_i = dy_{2i}$ - variação de posição em relação ao eixo Y do robô.

$dz_i = dz_{2i}$ - variação de posição em relação ao eixo Z do robô.

dexi - variação angular em torno do eixo X.

$$\text{dexi} = \arctan (\text{dy3i} - \text{dy2i}) / \text{C1} \quad (\text{rad})$$

døyi - variação angular em torno do eixo Y.

$$\text{døyi} = \arctan (\text{dz2i} - \text{dz1i}) / \text{C2} \quad (\text{rad})$$

døzi - variação angular em torno do eixo Z.

$$\text{døzi} = \arctan (\text{dy2i} - \text{dy1i}) / \text{C3} \quad (\text{rad})$$

Equações:

I.1) REPETIBILIDADE MÁXIMA DE POSIÇÃO:

$$r_{\max} = \max_{i=1}^N \sqrt{(\text{dxi} - \overline{\text{dx}})^2 + (\text{dyi} - \overline{\text{dy}})^2 + (\text{dzi} - \overline{\text{dz}})^2} \quad (\text{c.1})$$

I.2) REPETIBILIDADE MÁXIMA DE ORIENTAÇÃO :

$$\text{dexmax} = \max_{i=1}^N (\text{dexi} - \overline{\text{dex}}) \quad (\text{c.2})$$

$$\text{døymax} = \max_{i=1}^N (\text{døyi} - \overline{\text{døy}}) \quad (\text{c.3})$$

$$\text{døzmax} = \max_{i=1}^N (\text{døzi} - \overline{\text{døz}}) \quad (\text{c.4})$$

II) REPETIBILIDADE DE POSICIONAMENTO MONODIRECIONAL ESTÁTICO:

II.1) Posição :

$$\text{rest} = \overline{\text{D}} + 3.S_0 \quad (\text{c.5})$$

$$\overline{\text{D}} = \frac{1}{N} \sum_{i=1}^N \text{Di} \quad (\text{c.6})$$

$$D_i = \sqrt{(dx_i - \overline{dx})^2 + (dy_i - \overline{dy})^2 + (dz_i - \overline{dz})^2} \quad (c.7)$$

$$S_{D^*} = \sqrt{\frac{\sum_{i=1}^N (D_i - \overline{D})^2}{N - 1}} \quad (c.8)$$

II.2) Para a orientação:

$$S_{\theta x} = \sqrt{\frac{\sum_{i=1}^N (d\theta x_i - \overline{d\theta x})^2}{N - 1}} \quad (c.9)$$

$$S_{\theta y} = \sqrt{\frac{\sum_{i=1}^N (d\theta y_i - \overline{d\theta y})^2}{N - 1}} \quad (c.10)$$

$$S_{\theta z} = \sqrt{\frac{\sum_{i=1}^N (d\theta z_i - \overline{d\theta z})^2}{N - 1}} \quad (c.11)$$

C2) Exatidão

c.2.1) Metodologia

Na determinação da exatidão de posição do robô MANUTEC R3 foi utilizado para cálculo, os 53 pontos medidos através da mesa tridimensional e tendo sido identificado a posição do referencial do robô em relação ao referencial de medidas, os valores das medidas são corrigidos em relação aos valores de posição dados pelo modelo. A exatidão angular não foi medida.

c.2.2) Cálculos

Usando as variações de posição ($dT = (dx, dy, dz)^t$) em torno dos 53 pontos programados para ser atingido pela ponta da ferramenta. Sendo:

$T_{pi} = (x_{pi}, y_{pi}, z_{pi})^t$ - Posição programada.

$T_{medi} = (x_{mi}, y_{mi}, z_{mi})^t$ - Posição medida.

$dT_i = T_{pi} - T_{medi} = (dx_i, dy_i, dz_i)^t$ - Erro de posição.

i - varia de 1 até o número de pontos medidos.

I - Erro Máximo:

$$E_{max} = \max_{i=1}^N \sqrt{(x_{pi} - x_{mi})^2 + (y_{pi} - y_{mi})^2 + (z_{pi} - z_{mi})^2} \quad (c.12)$$

II - Erro Médio de Posição:

$$E_{med} = \sqrt{\overline{dx}^2 + \overline{dy}^2 + \overline{dz}^2} \quad (c.13)$$

onde:

$$\overline{dx} = \frac{1}{N} \sum_{i=1}^n (xp_i - xm_i) \quad (c.14)$$

$$\overline{dy} = \frac{1}{N} \sum_{i=1}^n (yp_i - ym_i)^2 \quad (c.15)$$

$$\overline{dz} = \frac{1}{N} \sum_{i=1}^n (zp_i - zm_i)^2 \quad (c.16)$$

$$Derro = \sqrt{\frac{\sum_{i=1}^N (E_i - E_{med})^2}{N - 1}} \quad (c.17)$$

onde:

$$E_i = \sqrt{(xp_i - xm_i)^2 + (yp_i - ym_i)^2 + (zp_i - zm_i)^2}$$

ANEXO D

MODELAGEM DO ERRO DE POSICIONAMENTO

Neste ítem será descrita a obtenção de uma equação que relaciona erros diferenciais de parâmetros com erros diferenciais de posicionamento através do Jacobiano. Esta relação será usada no algoritmo de estimativa. Durante a obtenção desta relação, é demonstrado matematicamente, a influência do erro de cada parâmetro no erro de posicionamento e a construção da matriz do jacobiano [1], [9], [21].

Seja $\delta \bar{\mathbf{p}}_i$, o vetor de erros relacionados com os parâmetros da matriz de transformação A_{i+1}^i do sistema. A matriz de transformação exata ${}^e A_{i+1}^i$, pode ser escrita como:

$${}^e A_{i+1}^i = A_{i+1}^i + dA_i \quad dA_i = dA_i(\delta \bar{\mathbf{p}}_i) \quad (D.1)$$

$\delta \bar{\mathbf{p}}_0 = [\delta x_0, \delta y_0, \delta z_0, \delta \phi_0, \delta \theta_0, \delta \psi_0]^t$: vetor de erros relativos aos parâmetros de posicionamento na matriz de transformação A_1^0 .

$\delta \bar{\mathbf{p}}_i = [\delta \theta_i, \delta d_i, \delta a_i, \delta \alpha_i]^t$: vetor de erros relativos aos parâmetros da matriz de transformação A_{i+1}^i no caso de juntas não paralelas.

$\delta \bar{\mathbf{p}}_i = [\delta \theta_i, \delta d_i, \delta a_i, \delta \alpha_i, \delta \beta_i]^t$: vetor de erros relativos aos parâmetros da matriz de transformação A_{i+1}^i no caso de juntas nominalmente paralelas.

A matriz de transformação exata do sistema é:

$${}^e T_{N+1}^0 = \prod_{i=0}^N (A_{i+1}^i + dA_i) \quad (D.2)$$

Expandindo a equação anterior, e ignorando termos de segunda e maior ordem, podemos escrever a matriz de transformação exato do sistema ${}^eT_{N+1}^0$ como:

$${}^eT_{N+1}^0 = T_{N+1}^0 + dT = T_{N+1}^0 + \sum_{i=0}^N (A_1^0 \dots A_i^{i-1} dA_i A_{i+2}^{i+1} \dots A_{N+1}^N) \quad (D.3)$$

Seja

$$dA_i = A_{i+1}^i \Delta A_i \quad (D.4)$$

onde ΔA_i é uma transformação representando a mudança na matriz de transformação A_{i+1}^i , expressa na coordenada i . Usando esta relação na equação (D.3) temos:

$$\begin{aligned} dT &= \sum_{i=0}^N (A_1^0 \dots A_{i+1}^i) \Delta A_i (A_{i+2}^{i+1} \dots A_{N+1}^N) \\ &= T_{N+1}^0 \sum_{i=0}^N (A_{i+2}^{i+1} \dots A_{N+1}^N)^{-1} \Delta A_i (A_{i+2}^{i+1} \dots A_{N+1}^N) \end{aligned} \quad (D.5)$$

Definindo $dT = T_{N+1}^0 \Delta T$, através da equação D.5 podemos escrever:

$$\Delta T = \sum_{i=0}^N [U_{i+1}]^{-1} \Delta A_i U_{i+1} \quad (D.6)$$

$$U_{i+1} = \prod_{k=i+1}^N A_{k+1}^k \quad U_{N+2} = I \quad (D.7)$$

$$\Delta T_i = [U_{i+1}]^{-1} \Delta A_i U_{i+1} \quad (D.8)$$

$$\Delta T = \sum_{i=0}^N \Delta T_i \quad (D.9)$$

ΔT_i - matriz representando a mudança na matriz de transformação de coordenadas T_{n+1}^0 devido à mudança de transformação de coordenadas ΔA_i em cada matriz de transformação A_{i+1}^i .

ΔT - mudança total na matriz de transformação T_{N+1}^0 .

O fato de se poder representar a mudança ΔT da matriz de transformação T_{N+1}^0 como a soma das mudanças ΔT_i relativas às mudanças nas matrizes de transformação de coordenadas a cada referencial $i+1$ é devido à linearização feita na equação (D.2). ΔT e ΔA são chamadas "matrizes de operação diferencial" e segundo Paul [6] são do tipo:

$$\Delta T = \begin{bmatrix} 0 & -r_z & r_y & dx \\ r_z & 0 & -r_x & dy \\ -r_y & r_x & 0 & dz \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (D.10)$$

onde dx , dy e dz são pequenos deslocamentos em T_{N+1}^0 , e r_x , r_y e r_z são pequenas rotações em torno do referencial "N+1". Para esses pequenos deslocamentos e rotações, podemos associar à matriz ΔT um vetor de translações e rotações diferenciais $\delta \bar{T}$.

$$\delta \bar{T} = \begin{bmatrix} dx \\ dy \\ dz \\ r_x \\ r_y \\ r_z \end{bmatrix} \quad (D.11)$$

A diferença entre a matriz de transformação exata ${}^eA_{i+1}^i$ e a nominal A_{i+1}^i (equação D.1) pode ser escrita através de derivadas parciais:

$$dA_i = \frac{\partial A_i}{\partial \theta_i} \delta \theta_i + \frac{\partial A_i}{\partial d_i} \delta d_i + \dots + \frac{\partial A_i}{\partial \beta_i} \delta \beta_i \quad (D.12)$$

Sendo $dA = A_{i+1}^i \Delta A_i$, através da equação D.12 podemos escrever:

$$\Delta A_i = Q_{\theta_i} \delta \theta_i + \dots + Q_{\beta_i} \delta \beta_i \quad (D.13)$$

onde Q são matrizes de operações diferenciais do tipo de ΔT (equação D.10):

$$Q_{\theta_i} = [A_{i+1}^i]^{-1} \frac{\partial A_i}{\partial \theta_i} \quad , \quad \dots \quad , \quad Q_{\beta_i} = [A_{i+1}^i]^{-1} \frac{\partial A_i}{\partial \beta_i}$$

Sendo ΔA_{θ_i} a mudança na matriz de transformação de coordenadas A_{i+1}^i devido ao erro no parâmetro θ_i e assim para d_i , α_i e β_i , através da equação D.13 temos:

$$\Delta A_i = \Delta A_{\theta_i} + \Delta A_{d_i} + \Delta A_{\alpha_i} + \Delta A_{\beta_i} \quad (D.14)$$

mostrando a influência de cada parâmetro na mudança da matriz de transformação A_{i+1}^i e consequentemente na matriz T_{N+1}^0 . A mudança na transformação T_{N+1}^0 , devido ao parâmetro θ_i do link i pode ser obtido através das equações (D.14) e (D.8).

$$\Delta T_{\theta_i} = [U_{i+1}]^{-1} \Delta A_{\theta_i} [U_{i+1}] = [U_{i+1}]^{-1} Q_{\theta_i} \delta \theta_i [U_{i+1}] \quad (D.15)$$

e do mesmo modo para os outros parâmetros.

Dada a aproximação $\Delta T \approx \delta \bar{T}$, a coluna do Jacobiano correspondente ao erro do parâmetro θ_i pode ser definida como:

$$\bar{J}_{\theta_i} = \frac{\delta \bar{T}}{\delta \theta_i} = [U_{i+1}]^{-1} Q_{\theta_i} [U_{i+1}] \quad (D.16)$$

$\bar{J}_{\theta_i} = \frac{\delta \bar{T}}{\delta \theta_i}$ - representa o quanto varia T_{N+1}^0 devido a um determinado erro no parâmetro θ_i , e assim para os outros parâmetros. A submatriz do Jacobiano J^i correspondente aos parâmetros do link i pode ser escrita:

$$J^i = [\bar{J}_{\theta_i} \quad \bar{J}_{d_i} \quad \bar{J}_{a_i} \quad \bar{J}_{\alpha_i} \quad \bar{J}_{\beta_i}] \quad (D.17)$$

i - varia de 1 até o número de graus de liberdade.

A submatriz J^0 correspondente aos parâmetros de posicionamento $\bar{p}_0$, será:

$$J^0 = [\bar{J}_{x_0} \quad \bar{J}_{y_0} \quad \bar{J}_{z_0} \quad \bar{J}_{\psi_0} \quad \bar{J}_{\theta_0} \quad \bar{J}_{\phi_0}] \quad (D.18)$$

A matriz Jacobiana correspondente à matriz de transformação T_{N+1}^0 no ponto $(\bar{k})$ é :

$$J = [J^0 \quad J^1 \quad J^2 \quad \dots \quad J^N] \quad (D.19)$$

$\bar{k}$ - vetor de parâmetros: $\bar{k} = [\bar{p}_0^t \quad \bar{p}_1^t \quad \dots \quad \bar{p}_N^t]^t$

Observando as equações D.8, D.14, e lembrando da aproximação $\Delta \mathbf{T} \approx \delta \mathbf{T}$, podemos escrever a seguinte relação:

$$\delta \mathbf{T} = \mathbf{J} \cdot \delta \mathbf{k} \quad (\text{D.20})$$

$\delta \mathbf{k}$ - vetor de erros dos parâmetros: $\delta \mathbf{k} = [\delta \mathbf{p}_0^t \ \delta \mathbf{p}_1^t \ \dots \ \delta \mathbf{p}_N^t]^t$
 $\delta \mathbf{T}$ - vetor de erros de posicionamento.

$$\mathbf{J} = \begin{vmatrix} \partial p_x / \partial \text{par}_1 & \partial p_x / \partial \text{par}_2 & \dots & \partial p_x / \partial \text{par}_{n_{\text{par}}} \\ \partial p_y / \partial \text{par}_1 & \partial p_y / \partial \text{par}_2 & \dots & \partial p_y / \partial \text{par}_{n_{\text{par}}} \\ \partial p_z / \partial \text{par}_1 & \partial p_z / \partial \text{par}_2 & \dots & \partial p_z / \partial \text{par}_{n_{\text{par}}} \\ \partial \psi_0 / \partial \text{par}_1 & \partial \psi_0 / \partial \text{par}_2 & \dots & \partial \psi_0 / \partial \text{par}_{n_{\text{par}}} \\ \partial \theta_0 / \partial \text{par}_1 & \partial \theta_0 / \partial \text{par}_2 & \dots & \partial \theta_0 / \partial \text{par}_{n_{\text{par}}} \\ \partial \phi_0 / \partial \text{par}_1 & \partial \phi_0 / \partial \text{par}_2 & \dots & \partial \phi_0 / \partial \text{par}_{n_{\text{par}}} \end{vmatrix} \quad (\text{D.21})$$

par_i - parâmetro i .
 n_{par} - número de parâmetros.

O algoritmo para estimativa proposto baseia-se na equação D.20. A matriz $\mathbf{J}$ terá o número de colunas igual ao número de parâmetros a ser estimado. O número de linhas será igual à 6 no caso de se relacionar erros de parâmetros com erros de posição e orientação durante o processo de estimativa ou igual à três no caso de se usar apenas medidas de posição durante a segunda fase do processo de estimativa proposto no item 1.1 desta dissertação.

ANEXO E

SISTEMAS DE INSPEÇÃO E MEDIDAS EM ÁGUAS SUBMARINAS

Recentemente vários sistemas de inspeção e medidas em águas profundas tem sido pesquisados e testados, estimulado principalmente pela exploração de petróleo. Neste contexto, a técnica mais explorada atualmente são sistemas a laser e cameras CCD (charge coupled device). Vários tipos de lasers e métodos de tratamento de suas informações são estudados e utilizados, acarretando um continuo refinamento dos resultados na precisão atualmente obtida, viabilizando-a para diversas outras aplicações.

As aplicações são variadas e podem ser simples como a utilização de dois lasers em cameras fotográficas, útil no cálculo de distâncias, ajuste do foco da câmera ou a obtenção, a partir de feixes paralelos com uma distância definida, da dimensão do objeto fotografado (figura E.1).

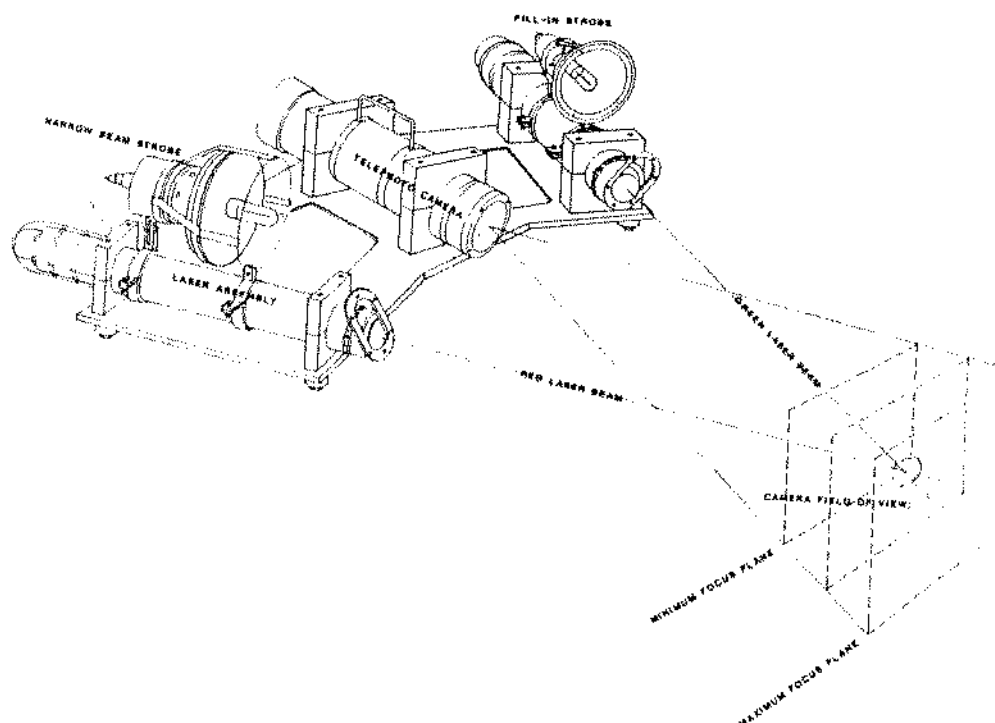


Figura E.1 - Uso de feixes de laser para medidas em fotos.

Direcionando para aplicações mais complexas, o laser tem sido empregado na geração de imagens bidimensionais e pesquisado para imagens tridimensionais. Um sistema usando sensores óticos e laser pode gerar imagens e fornecer informações geométricas e de posição, as quais são de grande importância na automação e qualidade em operações submarinas, geralmente executadas através de veículos comandados remotamente (ROV). Exemplos desses sistemas, como o Spotrange e Spotscan-2D, são fornecidos comercialmente pelo fabricante norueguês Seatex.

O Spotrange utiliza um preciso emissor de ondas acústicas de alta frequência, com um laser instalado coaxialmente. Geralmente acoplado a um ROV, esse sistema possibilita fazer medidas angulares e de distâncias, ajuda no posicionamento durante a construção de estruturas, estudo do relevo submarino e inspeção de tubulações. Através de uma câmera CCD, um vídeo e um software especializado, pode-se obter vários tipos de medidas off-line, através de imagens digitalizadas do ambiente focalizado (figura E.2).

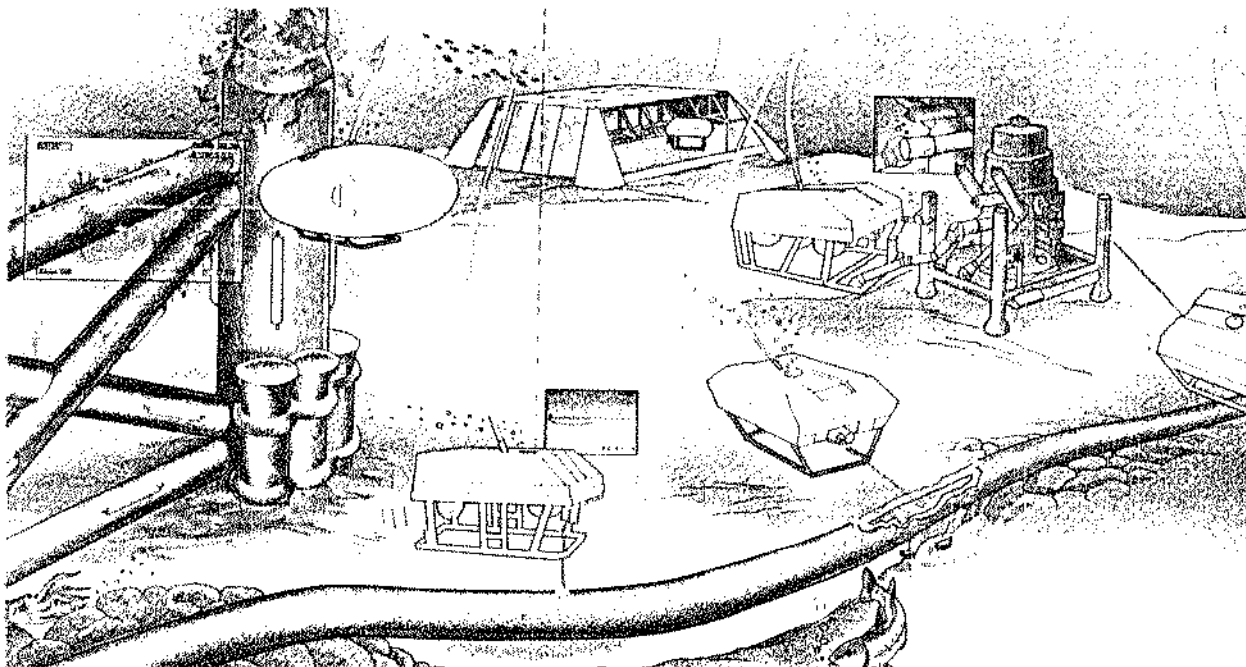


Figura E.2 - Utilização do Spotrange num ROV.

O SpotsScan 2D usa uma câmera CCD e laser verde. Emitindo laser sob determinado ângulo, a câmera capta informações do objeto a ser inspecionado. Essas informações são tratadas e os resultados podem ser obtidos em tempo real (figura E.3). Suas principais aplicações são no mapeamento topográfico do relevo submarino e na inspeção de estruturas e tubulações.

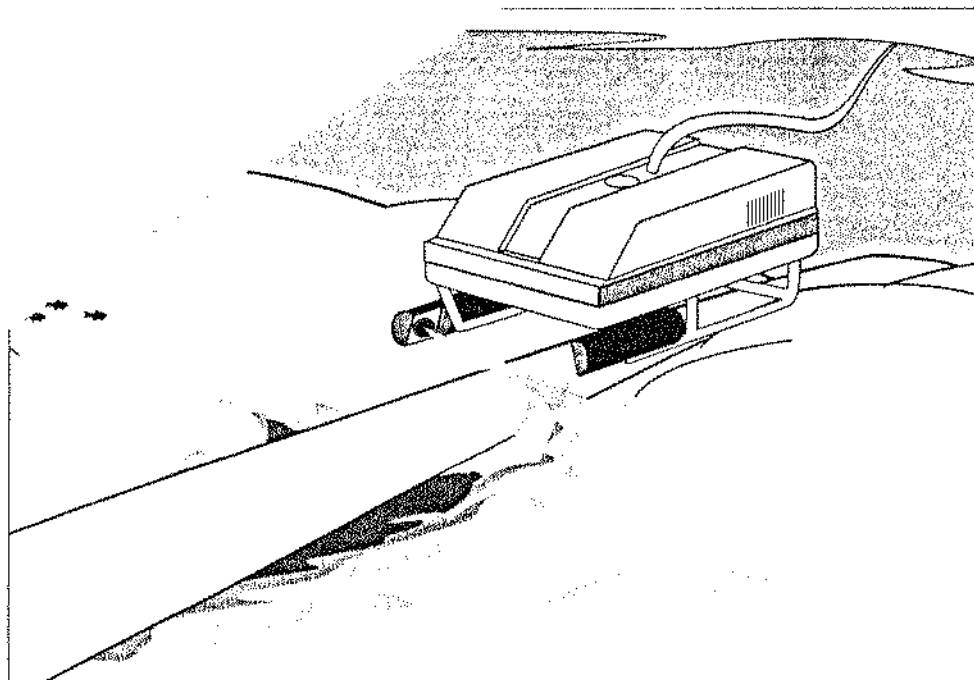


Figura E.3 - Inspeção de tubulações através do Spotscan 2D.

Baseado na pesquisa introdutória sobre mecanismos de medidas em águas profundas, e sem uma maior preocupação com detalhes técnicos, é proposto à seguir, dois possíveis sistemas para a obtenção de medidas geométricas e detecção de erros de posicionamento do sistema robô, base móvel e painel de operação referido no item 1.3 deste trabalho.

A primeira proposta, seria o acoplamento de um laser no elemento terminal do robô para medidas de distâncias. O robô após ter se posicionado em frente ao painel, deve se mover para que o laser atinja três ou mais sensores óticos fixos em posições definidas no painel de operação. Através das medidas entre os sensores do painel, medidas feitas pelo sistema laser entre o elemento terminal e os sensores óticos, informações sobre a posição e orientação do elemento terminal fornecidas pelo modelo geométrico do robô, pode-se, através de relações trigonométricas, obter os erros de posicionamento com relação ao painel de operação.

Outra proposta seria o uso de duas câmeras CCD, como no exemplo F do item 1.2 desta dissertação. Pontos luminosos, fixos no painel de operação, funcionariam como um sistema de referência para as imagens geradas pelas câmeras. No elemento terminal do robô, também seria acoplado um ponto luminoso. Medidas de distâncias entre o elemento terminal e os pontos luminosos no painel, obtidas através do tratamento de imagens do sistema com o robô em algumas configurações, devem levar ao cálculo de posições relativas. Através de um programa de estimativa pode-se determinar os erros desconhecidos.

A melhor solução deve partir da viabilidade técnica dos sistemas propostos e de informações gerais do projeto de automação de operações submarinas pela Petrobrás.

BIBLIOGRAFIA

- [1] Driels, M. R., Pathre U. S.:
"Significance of Observation Strategy on the Design of Robot Calibration Experiments",
Journal of Robotics Systems, pp 198-223, 1990.
- [2] Duelen G.; Schröer K.:
"Robot Calibration - Method and Results",
Robotics & Computer-Integrated Manufacturing ,
Vol 8, n 4, pp 223-231, 1991.
- [3] Gorla, B., Renaud, M.
"Modeles des Robots Manipulateurs"
Cepadues Editions, 1984.
- [4] Hayati, S.A.
"Robot Arm Geometric Link Parameter Estimation"
IEEE Control System Society, Decision and
Control, 22nd Conference, San Antonio, Texas, Dec 14-16,
Vol 3, pp 1011-1016, 1983.
- [5] Judd R. P., Knasinski, A. B.
"A Technique to Calibrate Industrial Robots whith
Experimental Verification",
Proc. IEEE Int. Conf. on Robotics and Automation,
pp 351-357, 1987.

-
- [6] Paul P. P.
"Robots Manipulators: Mathematics, Programing and Control",
MIT Press, Cambridge, MA, 1981.
- [7] Veitschegger, W. K., Wu, C.-H.
"A Method for Calibration and Compensating Robot
Kinematic Errors"
Proc. IEEE Int. Conf. on Robotics and Automation,
pp 39-44, 1987.
- [8] Veitschegger, W. K., Wu, C.-H
"Robot Accuracy Analysis Based on Kinematics"
IEEE Journal of Robotics and Automation
Vol RA-2, n 3, pp 171-179, setembro 1986.
- [9] Whitney, D. E., Lozinski, C. A., Rourke, J. M.
"Industrial Robot Calibration, Method and Results "
Proc. of 1984 ASME Conference on Computers and
Engineering, pp 92-100, 1984.
- [10] Gottwald, R.
"An automated non contact 3-D measuring system for industrial
aplication"
Kern & Co AG, Suíça.
- [11] Charles, Philippe
"Robotisation D'un Poste de Travail: Commande et
Environnement de Programmation"
Tese de doutorado, L'Institut Natioonal Polytechnique de
Grenoble, França, 1985.

-
- [12] Cruz, Joselito M.
"Projeto e Implementação de um Sistema de Geração Automática de Trajetória para Manipuladores"
Tese de Mestrado, Unicamp, Brasil, 1993.
- [13] Coutinho, Luis A. F.
"Um Ambiente Integrado de Desenvolvimento de Software Aplicado à Robótica"
Tese de Mestrado, Unicamp, Brasil, 1993.
- [14] Rivin, E. I.
"Mechanical Design of Robots"
Prentice Hall, 1987.
- [15] Stone, H. W.
"Kinematic Modeling, Identification, and Control of Robotic Manipulators"
Kluwer Academic Publishers, 1987.
- [16] Yao, Y. L., Yusoff M. R. Mohd
"A CAD-Based Errors Mapping and Layout Facility For Precision Robotic Operations"
Robotics & Computer-Integrated Manufacturing, Vol 9, N 6,
pág 505-511, 1992
- [17] Yonghong, B., Yi, Wang
"An Identification Method For Geometric Parameters Estimation of Robot Manipulators"
IFAC Identification an System Parameter Estimation, PRC 1988
pág. 555-559

-
- [18] Bennett, D. J.; Hollerbach, J. M.
"Identifying the Kinematics of Robots and for their Tasks"
IEEE International Conference on Robotics and Automation
Proceedings 1989, pág. 580-586
- [19] Craig, J. J.
"Introduction to Robotics Mechanics & Control"
Ed. Addison-Wesley Publishing Company - 1986
- [20] Denavit, J.; Hartenberg, R. S.
"A Kinematic Notation for Lower Pair Mechanisms Based on
Matrices"
Journal of Applied Mechanics, June 1985, pág. 215-221
- [21] Fu, K. S.; Gonzalez, R. C.; Lee, G. S. G.
"ROBOTICS: Control, Sensing, Vision and Intelligence"
Ed. McGraw-Hill Book Company 1987
- [22] Gousenes, L.
"Strategies for Solving Collision-Free Trajectories Problems
for Mobile and Manipulator Robots"
The International Journal of Robotics Research
Vol. 3, n° 4, Winter 1984, pág. 51-65

-
- [23] Kusiak, A.
"Programming, Off-Line Languages"
International Encyclopedia of Robotics-Applications and
Automation, Editor R. C. Dorf, 1988
Ed. Willey-Interscience Publication-John Wiley & Sons
Vol. 2, pág. 1235-1250
- [24] MacDonald, A. C.
"Robot Technology - Theory Design and Applications"
Ed. Prentice - Hall New Jersey 1986
- [25] Mooring, B. W.; Padavala, S. S.
"The Effect of Kinematic Model Complexity on Manipulator
Accuracy"
Proceedings of the IEEE Internacional Conference on Robotics
and Automation, Scottsdale, Arizona, Maio 1989, pág. 593-598
- [25] Puskorius, G. V., FeldKamp, L. A.
"Global Calibration of a Robot/Vision System"
IEEE, 1987
- [26] Zhuang, H., Routh, Zvi S.
"Robot Calibration Using the CPC Model"
Robotics and Computer-Integrated Manufacturing, Vol 9, No 3,
pág. 227-237, 1992
- [27] Wu, C. -H.
"The Kinematic error model for the design of robot manipulator"
American Control Conference, June 1983, pp. 497-502.

-
- [28] Wu, C. .H., Lee C. C.
"Estimation of the Acuracy of Robot Manipulator"
IEEE Trans. Automatic Control AC-30 (3) : 304-306, 1985.
- [29] Wu, Chi-Haur
"A Kinematic CAD Tool for the Design and Control of a Robot Manipulator"
The International Journal of Robotics Research , Vol 3 n° 1
Spring 1984, pág. 58-67
- [30] Rosário, J.M.; Messina, L.C.P.; Aust, E.:
"Development of An off-line Program for Underwater Work by using Industrial Robots",
Third (1993) Int. Offshore and Polar Eng. Conference, Singapore, junho de 1993.
- [31] Rosário, J.M.; Messina, L.C.P.; Aust, E.:
"Automated Intervention of Subsea Work with aid an Off-Line-Programming and Simulation System for Industrial Robot",
OMAE 1993 11th Int. Conference on Offshore Mechanics and Artic Engineering, Glasgow, Scotland, junho de 1993.
- [32] Rosário, J.M.; Weber, H.I.; Aust, E.; Schultieiss, G.F.:
"Development for Deep Water Robot Work",
Brasil Offshore'91 Engineering, Pentech Press publication, sept. 1991.

-
- [33] Rosário, J.M.; Weber, H.I.; Morooka, C.:
"Development and Control of Advanced Robots for Underwater Tasks"
5 ICAR, PISA,
Italia, Jun. 1991, pp 1792-1795.
- [34] Rosário, J.M.
"A systematic methode the kinematic analysis of industrial robots and the MANUTEC robot -r15",
GKSS, Alemanha, 1990.
- [35] Rosário, J.M.; Cazenave, A.; Blanchard, P.: "Génération de
"Trajectoires hors-ligne pour robots capables d'intervenir
en millieu hostile",
HISPANO-SUIZA, França, 1989
- [36] Cazenave, A.; Rosário, J.M.; Blanchard, P.:
"Etude et Development d'un Logiciel de Programmation
Hors-Ligne et Création de Trajectoires pour le Robot
ISIS-2",
HISPANO-SUIZA, Projet 4/SPE/1989, Paris, France, 1989.989.
- [37] Rosário, J.M.; Pelissou, P.; Bouzid, L.
"Mise en oeuvre d'un dispositif de mesure d'une longueur
à partir d'un système incrémental générateur d'impulsions",
ISMCM, 1987.

- [38] Guyomard, H.; Rosário, J.M.
"Les armoires de commande en robotique",
ISMCM, 1987.

- [39] Norma ISO/TC 184/SC 2/WG