

**UNIVERSIDADE DO ESTADO DE SANTA CATARINA
BACHARELADO EM CIÊNCIA COMPUTAÇÃO**

Rudimar Anor Zwirtes

**CINEMÁTICA INVERSA PARA CONTROLE DA
ABORDAGEM DE ÓRGÃOS TERMINAIS DE
ROBÔS MANIPULADORES**

Trabalho de Conclusão de Curso submetido à Universidade Estadual de Santa Catarina como parte dos requisitos para a obtenção do grau de Bacharel em Ciência da Computação

Orientador :
Marcelo da Silva Hounsell

Joinville, Novembro de 2004

CINEMÁTICA INVERSA PARA CONTROLE DA ABORDAGEM DE ÓRGÃOS TERMINAIS DE ROBÔS MANIPULADORES

Rudimar Anor Zwirtes

Este Trabalho de Conclusão de Curso foi julgado adequado para a obtenção do título de Bacharel em Ciência da Computação Área de Concentração (Sistemas de Conhecimento) e aprovada em sua forma final pelo Curso de Ciência da Computação Noturno do CCT/UDESC.

Cinara T. Menegazzo

Banca Examinadora

Marcelo da Silva Hounsell

Alexandre Gonçalves

Rogério Eduardo da Silva

Os pequenos atos que se executam são
melhores que todos aqueles grandes
que se planejam.

George C. Marshall (1880-1959)

Agradeço à Deus, pelo fôlego de vida,
à minha família pelos princípios
e à minha esposa pelo amor incondicional.

SUMÁRIO

| | |
|---|-----------|
| 1. INTRODUÇÃO | 10 |
| 2. ROBÓTICA | 13 |
| 2.1 MOVIMENTAÇÃO DOS ROBÔS..... | 14 |
| 2.2 TIPOS DE ROBÔS | 16 |
| 2.3 ESPAÇO DE TRABALHO..... | 18 |
| 3. CINEMÁTICA..... | 20 |
| 3.1 O PROBLEMA DA CINEMÁTICA DIRETA | 20 |
| 3.1.1 Representação de Denavit-Hartenberg (DH)..... | 22 |
| 3.2 O PROBLEMA DA CINEMÁTICA INVERSA | 25 |
| 3.2.2 Propriedades gerais das soluções | 28 |
| 3.2.2 Unicidade de soluções | 31 |
| 4. ROBÔ SCORBOT ER-4PC..... | 33 |
| 4.1 CINEMÁTICA DIRETA DO SCORBOT ER4PC | 35 |
| 4.2 CINEMÁTICA INVERSA DO SCORBOT ER4PC..... | 38 |
| 4.2.1 Junta da base | 38 |
| 4.2.2 Junta do Cotovelo..... | 39 |
| 4.2.3 Junta do Ombro | 41 |
| 4.2.4 Junta do Punho..... | 42 |
| 4.2.5 Junta de Rotação | 42 |
| 5. REALIDADE VIRTUAL..... | 43 |
| 5.1 O QUE É REALIDADE VIRTUAL?..... | 43 |
| 5.2 ALGUNS CONCEITOS E TERMOS DE RV | 45 |
| 5.2.1 Navegação e Interação..... | 45 |
| 5.2.2 Imersão e Presença..... | 46 |
| 5.2.3 Realidade Virtual Imersiva e Não Imersiva | 46 |
| 5.3 VRML..... | 47 |
| 6. IMPLEMENTAÇÃO..... | 49 |
| 6.1 IMPLEMENTAÇÃO DO CONTROLE DE ABORDAGEM | 50 |
| 6.1.1 Implementação da cinemática inversa | 50 |
| 6.1.2 Implementação das possíveis soluções..... | 53 |
| 6.1.3 Implementação do controle da abordagem..... | 55 |
| 6.1.4 Implementação da Interface | 57 |
| 7. CONCLUSÕES | 61 |
| 7.1 TRABALHOS FUTUROS | 61 |
| REFERÊNCIAS BIBLIOGRÁFICAS..... | 63 |
| ANEXOS..... | 64 |

LISTA DE FIGURAS

| | |
|---|----|
| FIGURA 1 – Tipos de Juntas | 15 |
| FIGURA 2 – Juntas do punho | 16 |
| FIGURA 3 – Configurações básicas de robôs | 17 |
| FIGURA 4 – Espaço de trabalho das configurações básicas..... | 19 |
| FIGURA 5 – Parâmetros Cinemáticos | 21 |
| FIGURA 6 – Solução gráfica do problema da cinemática direta | 25 |
| FIGURA 7 – Relação dos problemas da cinemática direta e inversa..... | 27 |
| FIGURA 8 – Exemplo de robô redundante | 31 |
| FIGURA 9 – Múltiplas soluções para robôs não redundantes | 32 |
| FIGURA 10 – Robô manipulador SCORBOT-ER4PC..... | 33 |
| FIGURA 11 – Envelope de Trabalho Scrobot-ER4PC | 35 |
| FIGURA 12 – Esquema gráfico do procedimento de DH..... | 36 |
| FIGURA 13 – Dispositivos Especiais para RV (Capacete)..... | 44 |
| FIGURA 14 – Simulador do grupo LARVA..... | 50 |
| FIGURA 15 – Código da função da cinemática inversa em Java | 52 |
| FIGURA 16 – Solução Cotovelo acima do robô Scrobot-ER4PC | 54 |
| FIGURA 17 – Solução Cotovelo abaixo do robô Scrobot-ER4PC | 54 |
| FIGURA 18 – Exemplo de relação entre coordenadas da base e ferramenta..... | 55 |
| FIGURA 19 – Composição do sistema de coordenadas da ferramenta..... | 56 |
| FIGURA 20 - Posições da Ferramenta do Robô para Controle da Abordagem..... | 58 |
| FIGURA 21 – Novo Simulador | 58 |
| FIGURA 22 – Parte do código Java para determinar opção de múltipla escolha | 59 |
| FIGURA 23 – Parte do código Java para escolha do tipo de abordagem e chamada da função calcularInversa. | 60 |
| FIGURA 24 – Parte do código Java para definição do objeto listaAbordagem..... | 60 |

LISTA DE ABREVIATURAS

DH – Denavit - Hartenber

DOF – *Degrees Of Freedom*

ER4PC – *Eshed Robotek for Personal Computer*

FMS – *Flexible Manufacturing Systems*

HMD – *Head Mounted Display*

HTML – *HiperText Markup Language*

LARVA – Laboratório de Realidade Virtual Aplicada

NC – *Numeric Control*

RV – Realidade Virtual

VRML – *Virtual Reality Modeling Language*

WWW – World Wide Web

RESUMO

O presente trabalho visa a implementação da cinemática inversa para o modelo virtual do robô manipulador SCORBOT-ER4PC que faz parte de um simulador que está sendo desenvolvido pelo Laboratório de Realidade Virtual Aplicada– LARVA da Universidade do Estado de Santa Catarina – UDESC. A cinemática é o campo da robótica que estuda o comportamento da posição e orientação do robô no espaço em relação a um sistema de coordenadas. A cinemática é dividida em direta e inversa, onde a direta é responsável em determinar, a partir, dos valores dos ângulos de cada elemento do robô, a posição e a orientação da garra do robô, enquanto a inversa é responsável em determinar, a partir de uma posição no espaço um conjunto de valores de ângulos que os elementos devem ter para o robô alcançar o ponto desejado. O cálculo da cinemática inversa do modelo do robô se dá a partir das características construtivas do robô e do cálculo da cinemática direta. Dispondo da função da cinemática inversa uma interface criada no simulador possibilita ao usuário interagir com o modelo virtual e analisar algumas possíveis soluções para o problema da cinemática inversa de robôs manipuladores. As possíveis soluções são levantadas de acordo com a configuração do modelo em estudo e também, das singularidades existentes no sistema do modelo, como volume do envelope de trabalho e limites construtivos de cada junta constituinte.

ABSTRACT

The present work seeks implementation of the inverse kinematics for the manipulator's robot virtual model SCORBOT-ER4PC that is part of a environment of simulation that is being developed by the Laboratory of Virtual Reality Applied-LARVA of the State University of Santa Catarina - UDESC. The kinematics is divided in direct and inverse, where the direct kinematics is responsible to resolve, from de values of the angles of each robot's element, the position and orientation of robot's end-effector, while the inverse kinematics is responsible to resolve, from the position in the space, one set of values of angles of each robot's element that reach the desired position. The resolution of the inverse kinematics of the model start, from the robot's features and of the direct kinematics' resolution. Having the function of the inverse kinematics, an interface created in the simulator makes possible the user to interact with the virtual model and to analyze some possible solutions for the problem of the robots manipulators' inverse kinematics. The possible solutions are lifted up in agreement with the configuration of the model in study and also, of the existent singularities in the system of the model, as volume of the work envelope and constructive limits of each joint.

1. INTRODUÇÃO

Este trabalho visa realizar um estudo da cinemática inversa de robôs manipuladores, utilizando o caso do robô SCORBOT-ER4PC e implementar uma função de cinemática inversa juntamente com uma interface para o controle de abordagem de órgãos terminais que será inserido no simulador do robô citado pertencente ao grupo LARVA.

A cinemática inversa é a base para a movimentação dos robôs manipuladores, ou seja, todas as tarefas realizadas pelo robô necessitam do cálculo da cinemática inversa, sendo assim é considerada a função mais importante dentro do estudo da cinemática de robôs manipuladores.

A principal importância deste trabalho é a disponibilização de uma ferramenta de ensino a estudantes de robótica. Com um ambiente de simulação os usuários poderão usufruir de recursos visuais e de interatividade que propiciarão o entendimento dos conceitos fundamentais que são estudados em cursos de fundamentos de robótica. Segundo Hounsell e Pimentel (HOUNSELL e PIMENTEL 2003) o uso de ambientes virtuais para ensino de robótica propiciam vários benefícios que auxiliam o usuário, como a sensação de espaço e proporção, a variação do cenário, onde o usuário pode estar em cenários diferentes realizando tarefas diferentes, levantando situações que no mundo real dificilmente haveria possibilidade de se realizar um treinamento ou aprendizado, a segurança do robô e do usuário, onde os equipamentos e todos os investimentos realizados estão protegidos e principalmente é garantida a integridade física do usuário.

Também é verificada a importância da cinemática inversa nos simuladores em aplicações industriais para o desenvolvimento da programação *off-line*, ou seja, não há necessidade de se retirar um robô de sua estação de trabalho para realizar a programação de alguma tarefa, como é o caso da programação *on-line* onde, através de ferramentas como *teach-pendant*, é ensinado ao robô, as posições, orientações e respectivos movimentos (REDEL et al, 2003). Utilizando-se da programação *off-line*, não se possui as ferramentas de ensino, sendo necessário disponibilizar ao usuário uma certa flexibilidade para implementar o programa, e este é o objetivo do controle de abordagem implementado neste trabalho pois com ele o usuário poderá escolher qual a

melhor maneira, por exemplo, de agarrar um determinado objeto que possua alguma restrição geométrica.

Na área da simulação também o uso da cinemática inversa é importante fator para estudos cinemáticos de projetos de novos robôs, onde uma simulação pode ser realizada durante o projeto do modelo real, antes da construção física do robô, verificando o bom funcionamento do robô (FLÜCKIGER et al, 1996).

Diante disto a seguir são mostrados os objetivos deste projeto:

Objetivo Geral:

Desenvolver uma interface que facilite a definição e a experimentação do posicionamento e orientação do órgão terminal de um robô manipulador a um determinado objeto via cinemática inversa.

Objetivos Específicos:

- Implementar a cinemática inversa para o simulador ER4PC do LARVA;
- Modificar a interface gráfica de forma a poder controlar e variar a abordagem do órgão terminal;
- Levantar e estudar técnicas de resolução da cinemática inversa considerando aspectos gráficos somente (não levando em consideração restrições e otimizações quanto a forças e energias).

O trabalho será dividido em capítulos de forma a organizar as informações para facilitar a compreensão. O capítulo 2 traz alguns conceitos básicos de robótica como os movimentos dos robôs, tipos de robôs e espaço de trabalho, que são importantes para fundamentação de outros conceitos apresentados a seguir. No capítulo 3 é realizada uma revisão bibliográfica do estudo da cinemática, onde serão abordados os procedimentos da resolução dos problemas das cinemáticas direta e inversa. Com a revisão concluída, no capítulo 4 é realizado todos os procedimentos do capítulo 3 para o robô em estudo, ou seja, o SCORBOT-ER4PC. No capítulo 5 são mostrados alguns conceitos de realidade virtual bem como a linguagem de programação de ambientes virtuais, chamada VRML (*Virtual Reality Modeling Language*). No capítulo 6 é mostrada a

implementação realizada, ou seja, a interface do simulador com a função da cinemática inversa, e por fim são realizadas considerações finais com propostas para futuros trabalhos.

2. ROBÓTICA

Uma das primeiras menções feitas ao desenvolvimento de organismos artificiais com o intuito de prover funções que auxiliassem os seres humanos foi descrita em uma novela de ficção científica muito conhecida escrita por Mary Shelley intitulada *Frankenstein*, publicada em 1817. A história trata dos esforços de um cientista Dr. Frankenstein, para criar um monstro humanóide, que aterroriza a comunidade local (GROOVER, 1988).

A robótica possui várias definições, segundo vários pontos de vista. Porém o termo ou a palavra “robô” no português é derivada pela palavra Checa “robotá” fazendo menção a uma peça teatral do início de 1920, de autoria de Karel Capek, intitulada *Os Robôs Universais de Rossum*. No vocabulário Checo a palavra “robotá” significa servidão ou trabalhador forçado. Desde então o termo tem sido aplicado para uma grande variedade de dispositivos mecânicos, ou seja, qualquer dispositivo que opere com algum grau de autonomia, usualmente sobre um controle computacional, tem sido chamada de *Robô*.

Uma definição de robô muito aceita no mundo é do Instituto de Robótica da América (RIA) (SPONG, 1989):

“Um robô é um manipulador multifuncional reprogramável projetado para mover materiais, peças, ferramentas, ou dispositivos especiais através de variáveis de movimento programadas para a realização de uma variedade de tarefas.”

Outra definição pode ser encontrada em (SCHILLING, 1990):

“Um robô é um dispositivo mecânico controlado por software que usa sensores para guiá-lo ou usa ferramentas na sua garra, que através de movimentos programados dentro de um espaço de trabalho, manipula objetos físicos”.

Em virtude desta definição mais técnica de robô pode-se afirmar que a robótica nasceu através do casamento de duas outras tecnologias recentes na época de 1940 e 1950 durante a segunda grande guerra mundial, a **Teleoperação** e o **Comando Numérico**.

O comando numérico (NC) foi desenvolvido para máquinas operatrizes. Como o nome sugere, o comando numérico envolve o controle das ações de uma máquina operatriz por meio de números, provendo grande precisão dos processos.

A teleoperação, ou sistemas mestre-escravo, foram desenvolvidos durante a segunda guerra mundial para manipulação de materiais radioativos. Neste processo o operador se mantém em um lugar seguro, e neste local, através de um circuito fechado de televisão ou visualmente também, pode guiar os movimentos de dispositivo remoto.

Surgem, a partir, deste momento os primeiros robôs manipuladores, sendo que podemos destacar o dispositivo robótico patenteado pelo britânico Cyril Walter Fenward em 1954 e o dispositivo de Transferência Programada de Artigos de George C. Devol, também em 1954.

Em 1962, Joseph Engelberger um físico formando pela Universidade de Columbia, juntamente com Geroge C. Devol fundaram a primeira companhia de robótica a *Unimation Company* e iniciaram a comercialização dos robôs manipuladores com o nome de **Unimate**. Os primeiros exemplares foram utilizados na industria automobilística (GROOVER, 1988).

Muitos outros projetos se desenvolveram com êxito também. Podemos citar o braço manipulador Stanford, da Universidade de Stanford, o Cincinnati Milacron, o robô PUMA, o SCARA, dentre outros.

Atualmente muitas companhias fabricam robôs manipuladores como por exemplo, Fanuc, ABB, Kuka, Kawasaki.

Dentro do ramo da robótica encontramos o estudo da cinemática dos robôs. O estudo da cinemática provê uma análise dos movimentos do robô em relação ao espaço geométrico. Para tanto, alguns conceitos, como os tipos de movimentos dos robôs manipuladores, os tipos de juntas que compõem estes robôs e em relação a sua configuração ou forma construtiva, o espaço geométrico que o robô pode atuar ou trabalhar, são necessários para melhor compreensão dos assuntos que seguirão

2.1 MOVIMENTAÇÃO DOS ROBÔS

Robôs industriais são projetados para realizarem trabalho produtivo. O trabalho é realizado pela habilidade do robô mover seu corpo, braço e punho através de uma série de movimentos e posições. Fixado no punho está o efetuator, garra ou ferramenta, o qual é usado pelo robô para realizar uma tarefa específica. Os movimentos do robô podem ser divididos em duas grandes categorias: movimento do braço e do corpo, e

movimento do punho. Os movimentos individuais das juntas associados com estas duas categorias geralmente são referenciados em termos de *graus de liberdade* (**DOF – Degrees Of Freedom**), e um robô industrial típico é equipado com 4 a 6 DOF. Três juntas são normalmente associadas com a ação do braço e duas ou três juntas são geralmente usadas para atuar o punho. Os membros rígidos que conectam as várias juntas são chamados de *elos ou links*. Em qualquer cadeia *elo-junta-elo* chamamos o elo mais perto da base de elo de entrada. O elo de saída é o elo que se move em relação ao elo de entrada.

As juntas usadas no projeto de robôs são tipicamente as juntas *linear* ou *rotacional*. Juntas lineares envolvem um movimento de deslizamento ou translacional dos elos conectados. Este movimento pode ser conseguido de várias maneiras (por um pistão, um mecanismo telescópico e um movimento relativo ao longo de um trilho). Podemos referir uma junta linear como uma junta do tipo **L** (Linear). O termo junta prismática é algumas vezes encontrada na literatura no lugar de junta linear.

Há, pelo menos, três tipos de juntas rotacionais que podem ser distinguidas nos manipuladores robóticos: a junta **R** (Rotacional), onde o eixo de rotação é perpendicular ao eixo dos dois elos conectados; a junta **T** (Torção), onde o eixo de rotação da junta é paralelo a ambos os elos; e a junta **V** (reVolução), no qual o elo de entrada é paralelo ao eixo de rotação e o elo de saída é perpendicular ao eixo de rotação. As quatro juntas são mostradas na fig. 1, com os seus respectivos movimentos relativos entre os elos.

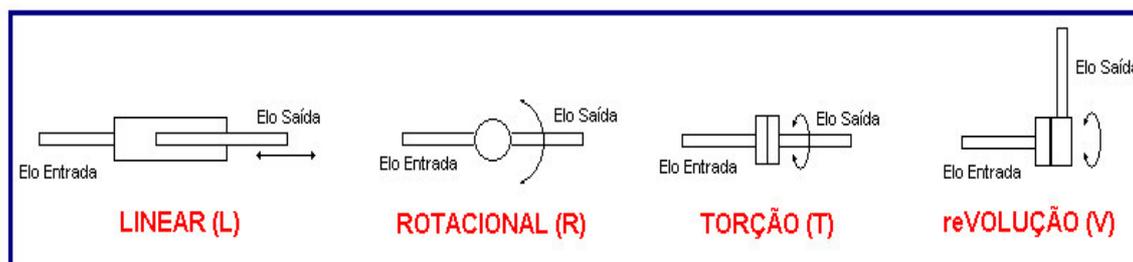


FIGURA 1 – Tipos de Juntas
Fonte: GROOVER, 1998

As juntas do braço e corpo são projetadas para habilitar o robô a mover sua ferramenta para uma *posição* desejada dentro dos limites do tamanho do robô e dos movimentos da junta.

O movimento do punho é projetado para habilitar o robô a *orientar* a ferramenta em relação à tarefa a ser realizada. Por exemplo, a mão deve ser orientada num ângulo

apropriado em relação a um objeto, a fim de agarrá-lo. Para resolver este problema de orientação, o punho é normalmente provido de 3 DOF. A fig. 2 mostra o punho e seus respectivos movimentos.

1. **Rolamento (Roll):** É a rotação do mecanismo do punho sobre o eixo do braço;
2. **Arfagem (Pitch):** É a rotação para cima ou para baixo do punho;
3. **Guinada (Yaw):** É a rotação para direita ou para esquerda do punho.

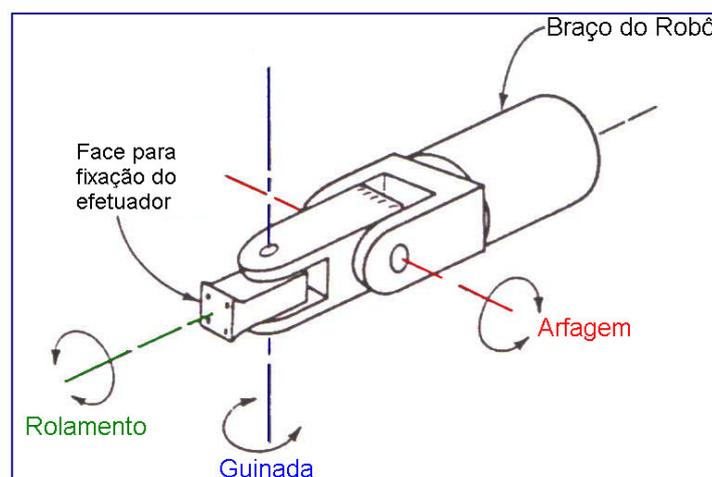


FIGURA 2 – Juntas do punho
Fonte: GROOVER, 1998

2.2 TIPOS DE ROBÔS

Robôs industriais são disponibilizados em uma larga variedade de tamanhos, formas e configurações físicas. A vasta maioria dos robôs, disponíveis comercialmente hoje, possui uma das quatro configurações básicas:

1. Configuração Polar (Esférico);
2. Configuração Cilíndrica;
3. Configuração de Coordenadas Cartesianas;
4. Braço Articulado (Revoluto)

A configuração polar é mostrada na fig. 3.a. Esta configuração usa um braço telescópico que pode ser erguido ou abaixado sobre um pivô horizontal. O pivô é montado sobre uma base rotativa. Estas várias juntas provem ao robô uma capacidade de mover seu braço dentro de um espaço esférico, e por isto também recebe o nome de robô de coordenadas esféricas.

A configuração cilíndrica, fig. 3.b usa uma coluna vertical e um elemento deslizante que pode ser movido ao longo da coluna, para cima e para baixo. O braço do robô é fixado no elemento deslizante que pode ser movido radialmente em relação à coluna. Pela rotação da coluna, o robô é capaz de realizar um espaço de trabalho que se aproxima a um cilíndrico.

O robô de coordenadas cartesianas, fig. 3.d usa três elementos lineares para construir os eixos x,y,z . Outros nomes são, algumas vezes, usados para esta configuração, incluindo robô xyz e robô retilinear. Pelo movimento dos três eixos um em relação ao outro, o robô é capaz de operar dentro de um espaço de trabalho retangular.

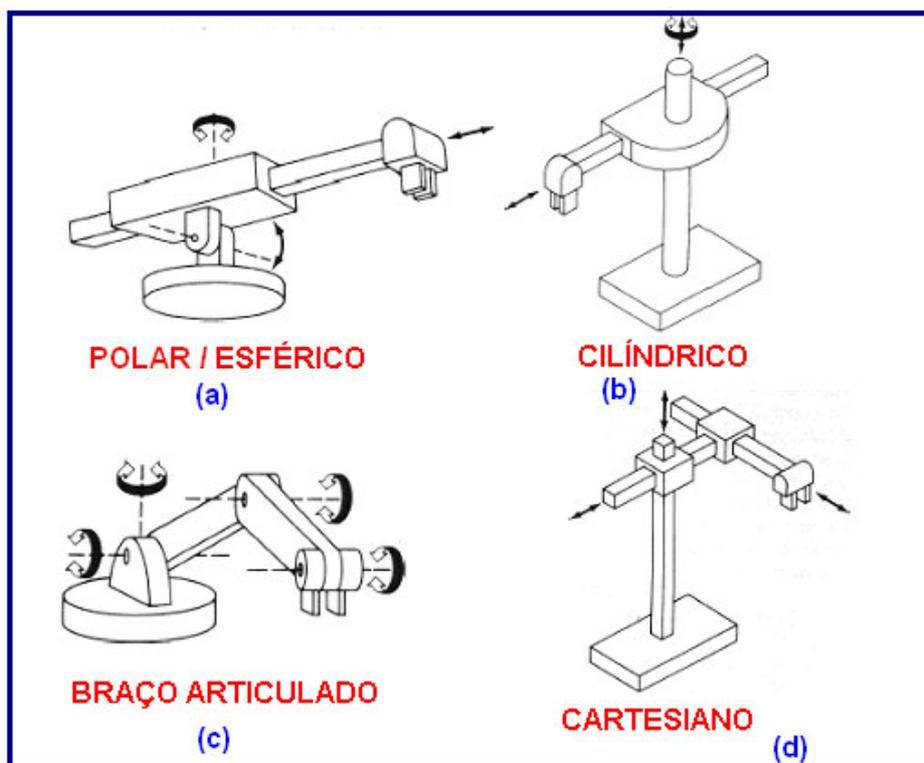


FIGURA 3 – Configurações básicas de robôs
Fonte: GROOVER, 1998

O robô braço articulado possui uma configuração similar ao braço humano, fig. 3.c. Ele consiste de componentes lineares correspondentes ao braço e antebraço humano, montados num pedestal vertical. Estes componentes são conectados por duas juntas rotacionais correspondentes ao ombro e o cotovelo. Um punho é fixado no final do antebraço, com isto provendo várias juntas adicionais.

Há relativas vantagens e desvantagens nestas quatro anatomias básicas do robô simplesmente devido às suas geometrias. Em termos de repetição de movimento, o robô cartesiano possui vantagens devido a sua estrutura rígida. Em termos de distância de alcance, a configuração polar e o braço articulado são superiores. As configurações cilíndrica e cartesiana são projetados para suportar cargas maiores.

2.3 ESPAÇO DE TRABALHO

Espaço de trabalho é o termo que refere ao espaço dentro do qual o robô pode alcançar com seu punho. A convenção de usar o punho para definir o espaço de trabalho do robô é adotada para evitar a complicação de diferentes tamanhos de efetadores/garras que podem ser aplicados no punho do robô. O efetador é uma adição ao robô básico e não deve ser contado como parte do espaço de trabalho do robô. Um efetador longo montado no punho adicionará uma extensão significativa ao robô comparado com um efetador menor. Também o efetador fixado no punho não poderá ser capaz de alcançar certos pontos dentro do espaço de trabalho normal do robô devido a particulares combinações de limites de juntas do robô.

O espaço de trabalho é determinado pelas seguintes características básicas:

- A condição física do robô;
- O tamanho do corpo, braço, e componentes do punho;
- Os limites dos movimentos das juntas do robô.

A influência da configuração física na geometria do espaço de trabalho é mostrada na fig. 4. Um robô de coordenadas polares possui um espaço de trabalho que é uma esfera parcial, fig. 4.d, um robô de coordenadas cilíndricas possui um envelope de trabalho cilíndrico, fig. 4.a, um robô de coordenadas cartesianas possui um espaço de

trabalho retangular, fig. 4.b, e um robô do tipo braço articulado possui um espaço de trabalho aproximadamente a uma esfera, fig. 4.c.

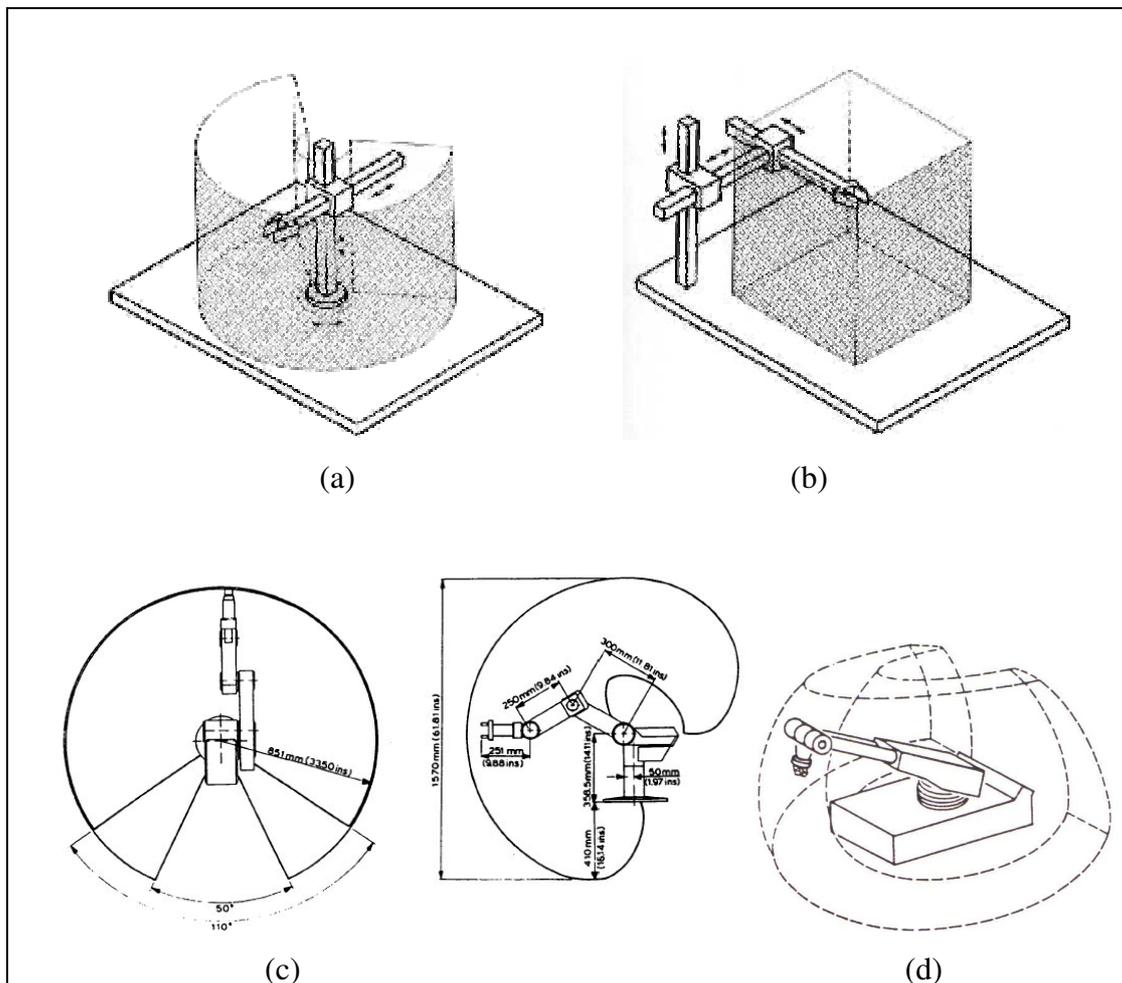


FIGURA 4 – Espaço de trabalho das configurações básicas
Fonte: GROOVER, 1998

O tamanho de cada espaço de trabalho é influenciado pelas dimensões dos componentes do braço e pelos limites dos movimentos das suas juntas. Usando uma configuração cilíndrica, por exemplo, limites na rotação da coluna sobre a base, determinarão qual porção de um cilindro completo o robô poderá alcançar com seu punho.

3.CINEMÁTICA

O estudo da cinemática é basicamente dividido em dois problemas, o problema da cinemática direta e o da cinemática inversa.

3.1 O PROBLEMA DA CINEMÁTICA DIRETA

O objetivo da cinemática direta é controlar a posição e a orientação de ferramenta num espaço tridimensional. A ferramenta pode ser programada para seguir uma trajetória planejada ou manipular objetos dentro de um espaço de trabalho. Para realizar a programação do movimento da ferramenta, deve-se primeiramente formular a relação entre as variáveis das juntas e a posição e orientação da ferramenta. Isto é o que se refere o problema da cinemática direta, como pode ser formalmente definido:

Problema da Cinemática Direta: Dado um vetor de variáveis de juntas de um manipulador robótico, determinar a posição e orientação da ferramenta em relação à base do robô.

O resultado da cinemática direta provém da equação do braço, para cada tipo de robô, conforme a configuração do mesmo. Para tanto é necessário conhecer os parâmetros cinemáticos envolvidos em cada elemento do robô. Cada par de elos é conectado por juntas ou de revolução ou prismáticas. A posição relativa e a orientação, destes dois elos sucessivos, podem ser representadas por dois parâmetros de junta, conforme mostrado na fig. 5.a .

Aqui a junta k conecta o elo $k-1$ ao elo k . Os parâmetros associados com a junta k são definidos em relação a z^{k-1} , o qual é alinhado como o eixo da junta k . O primeiro parâmetro de junta, θ_k , é chamado de *ângulo da junta*. Ele é a rotação sobre z^{k-1} necessária para fazer o eixo x^{k-1} paralelo com o eixo x^k . O segundo parâmetro, d_k é chamado de *distância do ângulo*. Ele é a translação ao longo de z^{k-1} necessária para fazer o eixo x^{k-1} intersecionar com o eixo x^k . Então θ_k é a rotação sobre o eixo da junta k , enquanto d_k é a translação ao longo do eixo da junta k . Para cada junta, é necessário que um desses parâmetros seja fixo enquanto o outro seja variável. O parâmetro da junta

variável depende do tipo de junta, para juntas de revolução, o ângulo da junta θ_k é variável e a distância da junta d_k é fixa. Portanto dois elos rotacionam um em relação ao outro sobre o eixo da junta k . Para juntas prismáticas, a distância da junta k é variável e o ângulo da junta é fixo. Neste caso, os dois elos translacionam um em relação ao outro ao longo do eixo da junta k .

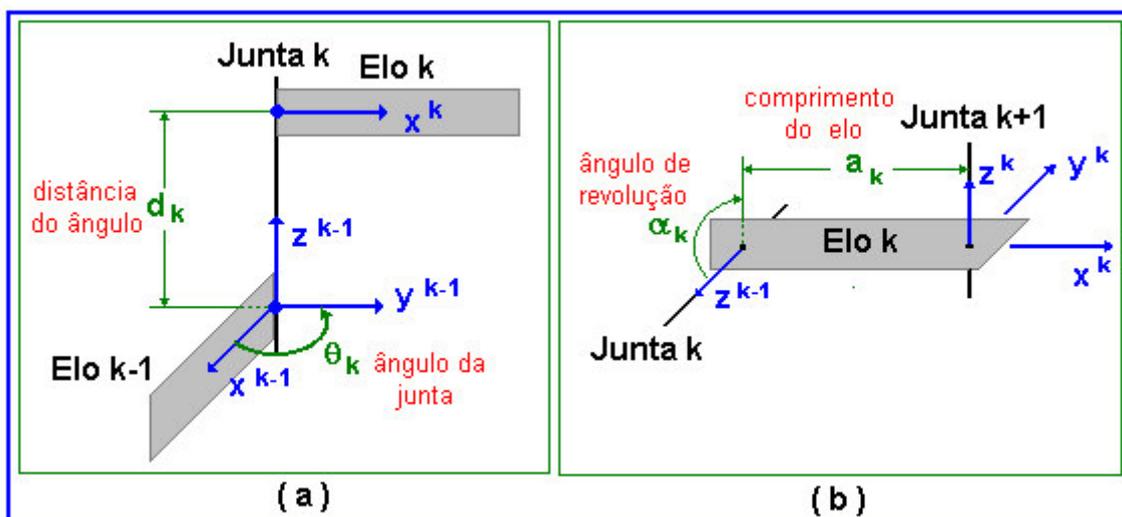


FIGURA 5 – Parâmetros Cinemáticos
 Fonte – SCHILLING, 1990

Assim como há uma junta conectando elos adjacentes, há também um elo entre juntas sucessivas. A posição e orientação relativa dos eixos de duas juntas sucessivas podem ser especificadas por dois *parâmetros de elos*, como mostrado na fig. 5.b. Aqui o elo k conecta a junta k à junta $k+1$. Os parâmetros associados com o elo k são definidos em relação a x^k , o qual é uma normal comum entre o eixo da junta k e o eixo da junta $k+1$. O primeiro parâmetro de elo, a_k , é chamado de *comprimento do elo*. Ele é a translação ao longo de x^k necessária para fazer o eixo z^{k-1} interseccionar com o eixo z^k . O segundo parâmetro de elo, α_k , é chamado de *ângulo de revolução do elo*. Ele é a rotação sobre x^k necessária para fazer o eixo z^{k-1} paralelo com o eixo z^k (SCHILLING,1990).

Diferente dos parâmetros de junta, os dois parâmetros de elos são sempre constantes e são especificados como partes do projeto mecânico. Para um manipulador robótico de n eixos, os quatro parâmetros cinemáticos constituem um conjunto mínimo necessário para especificar a configuração cinemática do robô. Para cada eixo, três parâmetros são fixos e dependem do projeto mecânico, enquanto o quarto parâmetro é

uma junta variável.

3.1.1 Representação de Denavit-Hartenberg (DH)

Devanit e Hartenberg propuseram uma notação sistemática para atribuir um sistema de coordenadas ortonormal com a regra da mão direita, um para cada elo numa cadeia cinemática aberta de elos (SCHILLING,1990). Uma vez que estes sistemas de coordenadas fixados ao elo são atribuídos, transformações entre sistemas de coordenadas adjacentes podem ser representadas por uma simples matriz de transformação de coordenadas homogêneas 4x4 padronizada. Para atribuir o sistema de coordenadas para os elos do robô, considere L_k ser o sistema associado com o elo k , ou seja:

$$L_k \hat{=} \{x^k, y^k, z^k\} \quad 0 \leq k \leq n$$

O sistema de coordenadas L_k será fixado ao fim distal do elo k para $0 \leq k \leq n$. Isto coloca o último sistema de coordenadas, L_n , na ponta da ferramenta. Os conjuntos de coordenadas são atribuídos aos elos usando o seguinte procedimento (SCHILLING,1990):

0. Enumere as juntas de 1 a n começando com a base e terminando com ângulos *yaw*, *pitch* e *roll*, nesta ordem.
1. Atribuir o conjunto de coordenadas ortonormal com a regra da mão direita L_0 à base do robô, sendo que z^0 deverá ser alinhado com o eixo da junta 1. Faça $k=1$.
2. Alinhe z^k com o eixo da junta $k+1$.
3. Localize a origem de L_k na interseção dos eixos z^k e z^{k-1} . Se eles não se intersecionarem, use a interseção de z^k com a normal comum entre z^k e z^{k-1} .
4. Estabeleça x^k sendo ortonormal a ambos z^k e z^{k-1} .
5. Estabeleça y^k para formar o conjunto de coordenadas L_k utilizando a regra da mão direita.
6. Faça $k=k+1$. Se $k < n$, vá para o passo 2, senão, continue.
7. Selecione a origem de L_n na ponta da ferramenta. Alinhe z^n com o vetor aproximação, y^n com o vetor deslizamento e x^k com o vetor normal da

ferramenta. Faça $k=1$.

8. Localize o ponto b^k na interseção dos eixos x^k e z^{k-1} . Se eles não se interseccionam use a interseção de x^k com a comum normal entre x^k e z^{k-1} .
9. Faça θ_k como o ângulo da rotação de x^{k-1} e x^k medido sobre z^{k-1} .
10. Faça d_k como a distância da origem do conjunto L_{k-1} ao ponto b^k medido ao longo de z^{k-1} .
11. Faça a_k como a distância do ponto b^k até a origem do conjunto L_k medido ao longo de x^k .
12. Faça α_k como o ângulo de rotação entre z^{k-1} e z^k medido sobre x^k .
13. Faça $k=k+1$. Se $k \leq n$, vá para o passo 8, senão pare.

Uma vez que um sistema de coordenadas de elos é atribuído usando o algoritmo DH, pode-se então, transformar de um sistema de coordenadas k para o de coordenadas $k-1$ usando uma matriz de transformação homogênea de coordenadas. Pela multiplicação de várias destas matrizes de transformação juntas, chega-se na matriz de transformação de coordenadas compostas na qual transforma ou mapeia as coordenadas da ferramenta dentro da coordenada de base.

Há quatro passos envolvidos na construção da matriz de transformação homogênea os quais mapeiam o sistema de coordenadas k dentro do de coordenadas $k-1$. Cada um desses passos está associado com um dos quatro parâmetros cinemáticos. Para determinar a matriz de transformação, deve-se sucessivamente rotacionar e translacionar o sistema de coordenadas $k-1$ para coincidir com o conjunto de coordenadas k . Usando os passos 8 a 12 do algoritmo DH, isto envolve as quatro operações fundamentais sumarizadas na tab. 1.

TABELA 1 – Operações de transformação de coordenadas
Fonte – SCHILLING, 1990

| OPERAÇÃO | DESCRISÃO |
|----------|---|
| 1 | Rotacionar L_{k-1} sobre z^{k-1} por θ_k |
| 2 | Translacionar L_{k-1} ao longo de z^{k-1} por d_k |
| 3 | Translacionar L_{k-1} ao longo de x^{k-1} por a_k |
| 4 | Rotacionar L_{k-1} sobre x^{k-1} por α_k |

Utilizando-se das transformações homogêneas compostas tem-se a seguinte matriz para cada transformação do sistema de coordenadas k-1 para o de coordenadas k:

$${}^k T_{k-1} = \begin{bmatrix} \cos \theta_k & -\cos \alpha_k \cdot \text{sen} \theta_k & \text{sen} \alpha_k \cdot \text{sen} \theta_k & a_k \cdot \cos \theta_k \\ \text{sen} \theta_k & \cos \alpha_k \cdot \cos \theta_k & -\text{sen} \alpha_k \cdot \cos \theta_k & a_k \cdot \text{sen} \theta_k \\ 0 & \text{sen} \alpha_k & \cos \alpha_k & d_k \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.1)$$

Portanto, para realizar a transformação do conjunto de coordenadas de k=0 até k=n necessita-se realizar a multiplicação de todas as matrizes:

$$T_{base}^{ferramenta} = T_{base}^1 \cdot T_1^2 \cdot \dots \cdot T_{ferramenta-1}^{ferramenta} \quad (3.2)$$

Como resultado às sucessivas multiplicações das matrizes de transformações homogêneas temos uma matriz resultante, que é chamada de equação do braço (SCHILLING, 1990):

$$T_{base}^{ferramenta}(q) = \left[\begin{array}{ccc|c} R(q) & & & p(q) \\ \hline 0 & 0 & 0 & 1 \end{array} \right] \quad (3.3)$$

Para cada valor do vetor de junta q, que é composto pelas juntas variáveis do robô, com dimensão correspondente ao número de graus de liberdade, a equação do braço $T(q)$ pode ser resolvida. Na equação 3.3 a sub-matriz 3x3 $R(q)$ na superior esquerda especifica a orientação da ferramenta, enquanto a sub-matriz 3x1 $p(q)$ na superior direita especifica a posição da ponta da ferramenta. As três colunas de $R(q)$ indicam a direção dos três vetores unitários do sistema de coordenadas da ferramenta em relação à base, como pode ser visto na fig.6 onde os vetores r^1 , r^2 e r^3 são os vetores de $R(q)$. Similarmente, $p(q)$ especifica as coordenadas da ponta da ferramenta em relação à base. A solução do problema da cinemática direta na equação 3.2, é mostrada na fig. 6.

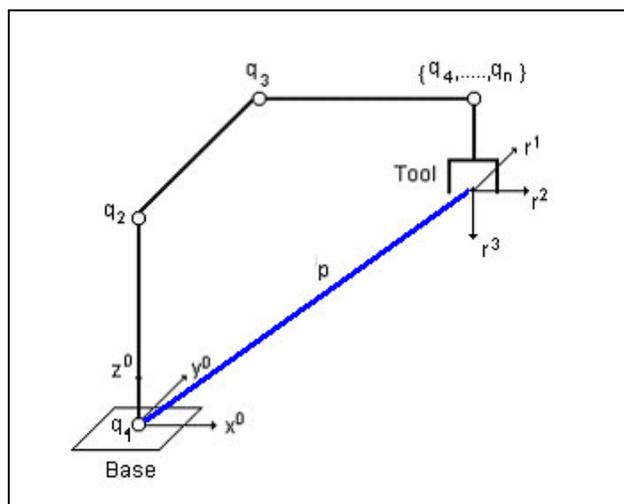


FIGURA 6 – Solução gráfica do problema da cinemática direta
 Fonte: SCHILLING, 1990

3.2 O PROBLEMA DA CINEMÁTICA INVERSA

A chave para a solução do problema da Cinemática Direta foi o algoritmo de Denavit-Hartenberg, um procedimento sistemático para determinar os elos de coordenadas para um robô manipulador. Sucessivas transformações entre conjuntos de coordenadas adjacentes, começando na ponta da ferramenta retornando até a base do robô, então levando para a equação do braço. A equação do braço representa a posição p e a orientação R da ferramenta na base do robô como uma função de variáveis de juntas q , conforme mostrado na fig. 6.

Por conveniência, iremos referenciar a posição e a orientação da ferramenta coletivamente como a *configuração* da ferramenta. A solução para o problema da cinemática direta pode ser expressa da seguinte forma: R representa a *rotação* e p representa a *translação* da ferramenta em relação à base, conforme a equação 3.3.

A solução para o problema da cinemática direta é útil pois mostra explicitamente a dependência da configuração da ferramenta nas variáveis das juntas. Isto pode ser utilizado, por exemplo, na determinação do tamanho e da forma do espaço de trabalho. Em particular, supondo que Q representa a faixa de valores no espaço R^n que as variáveis das juntas podem assumir. Referencia-se Q como *junta-espaço de trabalho* do robô. Tipicamente, o espaço de trabalho é um poliedro convexo em R^n da seguinte forma geral:

$$Q = \{q \in \mathbb{R}^n : q^{\min} \leq Cq \leq q^{\max}\} \quad (3.4)$$

Aqui q^{\min} e q^{\max} são vetores constantes em \mathbb{R}^n os quais representam *limites das juntas* e C é uma matriz de par de juntas. Para o mais simples caso, a Matriz $C = I$, neste caso q_k^{\min} e q_k^{\max} representam os limites inferiores e superiores, respectivamente, para a junta k . Mais genericamente, se a matriz C possui alguns termos de fora da diagonal, então a equação 3.4 representa as restrições de uma combinação linear das variáveis das juntas. Neste caso o número de restrições m pode exceder o número de variáveis n . O conjunto de posições no espaço Cartesiano \mathbb{R}^3 que são alcançados pela ponta da ferramenta pode ser investigado pela verificação dos valores $p(q)$ com q varia sobre Q . Similarmente, o conjunto de orientações alcançáveis pela ferramenta pode ser determinado pela verificação dos valores de $R(q)$ como q varia sobre a junta-espaço de trabalho Q .

Talvez o mais importante benefício provido pela solução do problema da cinemática direta é que ela se situa como base para a solução do problema da cinemática inversa. O vetor das variáveis das juntas q é restringido pelo sub-conjunto Q de \mathbb{R}^n . Referenciamos o vetor espaço \mathbb{R}^n neste caso com *espaço das juntas*. Similarmente, os parâmetros da configuração da ferramenta $\{R, p\}$ pode ser associado com um sub-conjunto W de \mathbb{R}^6 . Referenciamos o vetor espaço \mathbb{R}^6 neste caso como *espaço configuração da ferramenta*. O espaço configuração da ferramenta é de sexta dimensão pois configurações arbitrárias da ferramenta podem ser especificadas pelo uso de três coordenadas de posição (p_1, p_2, p_3) junto com três coordenadas de orientação (*yaw, pitch, roll*). Solucionando o problema da cinemática direta é equivalente a encontrar mapeamento do espaço das juntas para o espaço da configuração da ferramenta, enquanto solucionando o problema da cinemática inversa é equivalente a encontrar o mapeamento do espaço da configuração da ferramenta de volta para o espaço das juntas.

Uma sumarização desta relação entre as duas formas do problema da cinemática é mostrado na fig. 7.

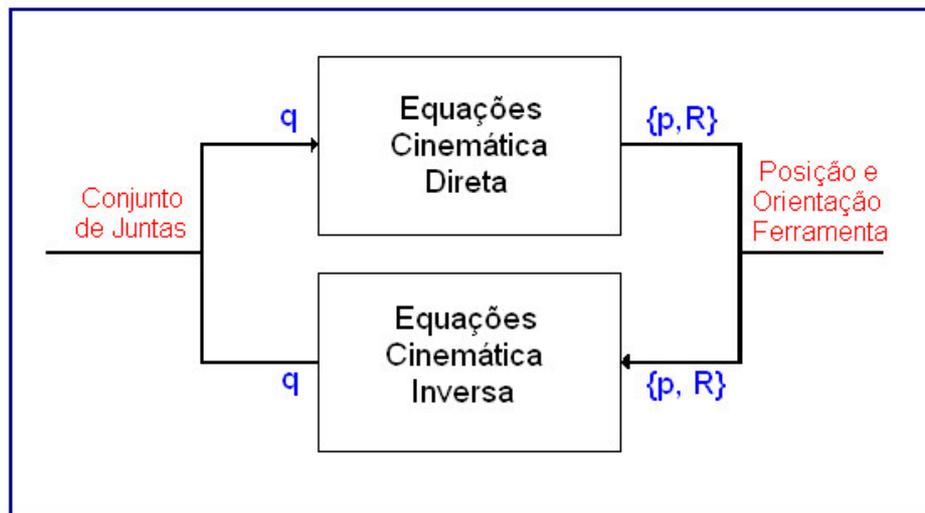


FIGURA 7 – Relação dos problemas da cinemática direta e inversa
 Fonte: SCHILLING, 1990

O problema da cinemática inversa é mais difícil que o problema da cinemática direta, pois não há um procedimento sistemático explícito como o algoritmo DH (SCHILLING, 1990). Um resultado disto é que cada robô, ou classe de robôs, tem que ser tratados separadamente.

No entanto, a solução para o problema da cinemática inversa é mais útil do que da cinemática direta, pois no controle da trajetória do robô a cinemática inversa se faz necessária para escolha da melhor configuração das juntas para um movimento de um ponto a outro com o mínimo de esforço, ou seja, com o movimento de menos juntas possíveis. Além disso, a chave para fazer robôs mais versáteis implica em usar *feedbacks* de sensores externos tais como o de visão. Sensores externos alimentam informações sobre a localização e orientação de objetos em termos de variáveis do espaço de configuração.

Estas informações são necessárias na determinação dos valores apropriados das juntas do robô, para que o robô possa realizar determinada tarefa em locais sujeitos a interferência de outros equipamentos no espaço de trabalho do robô. Portanto devemos encontrar o mapeamento da especificação de entrada do espaço configuração da ferramenta dentro de uma especificação de saída do espaço de junta. Isto é o problema da cinemática inversa. Caracterizamos este problema mais formalmente como o seguinte (SCHILLING, 1990):

Problema da Cinemática Inversa: Dado uma posição p e uma orientação R desejada para a ferramenta, encontre as variáveis das juntas q as quais satisfaçam a equação do braço 3.3.

A solução para o problema da cinemática inversa é muito útil mesmo quando sensores externos não são empregados. Um caso, por exemplo, é o problema de manter o elemento final ou ferramenta seguindo um caminho em linha reta. Certos tipos de operações de montagem, pintura ou colagem requerem que caminhos em linha reta sejam seguidos ou pelo menos obtenham-se uma aproximação destes. Uma trajetória em linha reta é naturalmente formulada no espaço de configuração da ferramenta. Então é necessário encontrar uma trajetória correspondente no espaço das juntas a qual irá produzir um movimento linear da ponta da ferramenta. Este é um especial caso do problema da cinemática inversa.

3.2.2 Propriedades gerais das soluções

Embora uma solução geral tem sido obtida para o problema da cinemática direta, os detalhes de uma solução explícita para o problema da cinemática inversa, dependem do robô ou da classe de robôs que estão sendo investigada. No entanto, há certas características da solução que são as mesmas no geral.

Primeiramente, examinaremos as condições sob as quais as soluções para o problema da cinemática inversa existam. Claramente, se uma posição p da ponta da ferramenta está fora do espaço de trabalho, então não pode haver solução. Além disto, quando p está dentro do espaço de trabalho, pode haver certas orientações R da ferramenta as quais não são realizáveis sem violar um ou mais limites das variáveis das juntas. Se o robô possui menos que três graus de liberdade para a orientação da ferramenta, então nem todas as classes de orientação não são realizáveis. Para examinar este assunto, consideremos a seguinte formulação, na equação 3.5, mais detalhada da equação do braço.

$$T_{base}^{ferramenta}(q) = \begin{bmatrix} R_{11} & R_{12} & R_{13} & p_x \\ R_{21} & R_{22} & R_{23} & p_y \\ R_{31} & R_{32} & R_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.5)$$

Sendo que a última linha da equação do braço é sempre constante, a equação do braço constitui um sistema de 12 simultâneas equações algébricas não lineares com n componentes desconhecidos de q . No entanto, as 12 equações são de forma alguma independentes umas das outras. Sendo que R é uma matriz de transformação de coordenadas representando uma pura rotação de um elo ortonormal à outro segundo $R^{-1} = R^T$. Mas se $R^T.R=I$, então as três colunas de R formam um conjunto ortonormal. Esta mutua ortogonalidade coloca três restrições nas colunas de R , que são:

$$\begin{aligned} r^1.r^2 &= 0 \\ r^1.r^3 &= 0 \\ r^2.r^3 &= 0 \end{aligned} \quad (3.6)$$

Estas restrições vêm dos termos diagonais de $R^T.R=I$. Os termos diagonais ditam que cada coluna de R deve também ser um vetor unitário. Isto adiciona mais três restrições para as colunas de R :

$$\|r^k\| = 1 \quad 1 \leq k \leq 3 \quad (3.7)$$

Então as 12 restrições implícitas na equação do braço atualmente representam apenas 6 restrições independentes em n componentes desconhecidos do vetor das variáveis das juntas q . Se quisermos uma solução geral para o problema da equação do braço, para qual um q possa ser encontrado o qual gera uma configuração da ferramenta, então o número de desconhecidos deve ser o número de restrições independentes, que é (SCHILLING, 1990):

$$\text{Manipulador Geral} \rightarrow n \geq 6$$

O limite mínimo no número de eixos n é uma necessidade, mas não uma condição suficiente para a existência de uma solução para o problema da cinemática inversa quando configurações da ferramenta arbitrárias são especificadas. Claramente, a posição da ferramenta deve ser dentro do espaço de trabalho do robô, e a orientação da ferramenta deve ser tal que nenhum dos limites nas variáveis das juntas sejam violadas. Quando estas restrições adicionais nos valores de p e R são satisfeitas, não há garantias que uma *expressão de forma fechada* para a solução do problema da cinemática inversa pode ser obtida.

A estratégia geral para a solução do problema da cinemática inversa simplifica quando o robô possui um punho esférico. Para ver isto, considere o caso de um robô com n eixos onde $4 \leq n \leq 6$. Suponha que o último eixo é um eixo de rotação da ferramenta, e supondo que o robô tenha um punho esférico, o que significa que os $n - 3$ eixos no fim do braço se interseccionam no mesmo ponto. Para esta classe de robôs, o problema da cinemática inversa pode ser decomposto em dois subproblemas menores pelo particionamento do problema original no punho. Dado a posição da ponta da ferramenta p e a orientação da ferramenta R , a posição do punho p^{punho} pode ser inferido de p retornando ao longo do vetor aproximação (SCHILLING, 1990).

$$p^{punho} = p - d_n \cdot r^3 \quad (3.8)$$

Aqui a distância da junta d_n representa o comprimento da ferramenta para um robô de n eixos tão longo quanto o último eixo é um eixo de rotação da ferramenta. O vetor aproximação r^3 é simplesmente a terceira coluna da matriz de rotação R . Uma vez que a posição do punho p^{punho} é obtida de $\{p, R, d_n\}$, as primeiras três variáveis de juntas $\{q_1, q_2, q_3\}$ que são usadas para posicionar o punho podem então ser obtidas da seguinte *equação do braço reduzida*:

$$T_{base}^{punho}(q_1, q_2, q_3) i^4 = \begin{bmatrix} p - d_n \cdot r^3 \\ 1 \end{bmatrix} \quad (3.9)$$

A quarta coluna de T representa as coordenadas homogêneas da origem do elo do punho L_3 relativo à base L_0 . Sendo que as coordenadas do punho dependem apenas das

variáveis das juntas $\{q_1, q_2, q_3\}$, estas variáveis das juntas dos eixos maiores podem ser resolvidas por separação usando a equação 3.9. Uma vez que as variáveis das juntas $\{q_1, q_2, q_3\}$ são encontradas, seus valores podem ser substituídos dentro da equação geral do braço na equação 3.5 e podem ser resolvidas as variáveis das juntas remanescentes $\{q_4, \dots, q_n\}$.

3.2.2 Unicidade de soluções

A existência de solução para o problema da cinemática inversa não é a única resposta que precisa ser respondida. Quando a solução existe, tipicamente ela não é única. Isto é, múltiplas soluções podem ser levantadas em vários métodos. Por exemplo, alguns robôs são projetados com n eixos onde $n > 6$. Para estes robôs, existem muitas soluções para o problema da cinemática inversa. Os robôs com mais de 6 eixos são chamados de *robôs cinematicamente redundantes*, pois eles possuem mais graus de liberdade que os necessários para estabelecer configurações de ferramentas arbitrários. Estes graus de liberdade extras adicionam flexibilidade para os manipuladores. Por exemplo, um robô redundante pode ser usado para alcançar algum objeto ao redor de algum objeto e manipulá-lo. Aqui alguns dos graus de liberdade podem ser usados para evitar o obstáculo enquanto os demais dos graus de liberdade são usados para configurar a ferramenta. Considere, por exemplo, a visão de cima de um robô redundante SCARA mostrado na fig. 8. Sendo que apenas duas juntas são necessárias para estabelecer uma posição horizontal da ferramenta, a Segunda junta do cotovelo do robô SCARA é redundante (SCHILLING, 1990).

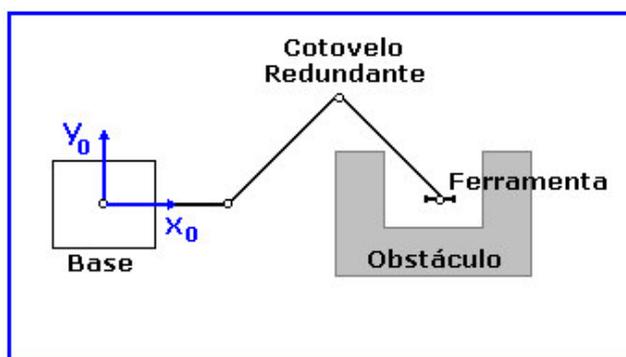


FIGURA 8 – Exemplo de robô redundante
Fonte: SCHILLING, 1990

Quando o robô não é cinematicamente redundante, há geralmente circunstâncias na qual a solução do problema da cinemática inversa não é única. Várias soluções distintas podem ser encontradas quando o tamanho do espaço de trabalho Q é suficientemente grande. Como um caso a ser visto, considere um robô de coordenadas articulado mostrado na fig. 9. Se os limites da taxa de deslocamento para as juntas do ombro, cotovelo e do *pitch* são suficientemente grandes, então duas soluções distintas existem para uma simples tarefa de colocar a ponta da ferramenta em frente do robô.

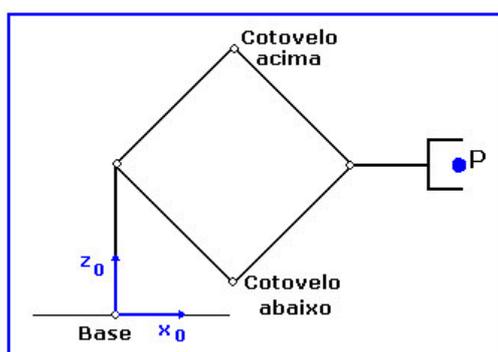


FIGURA 9 – Múltiplas soluções para robôs não redundantes
Fonte: SCHILLING, 1990

São chamadas as duas soluções na fig. 9 de *solução cotovelo acima* e *solução cotovelo abaixo*. No espaço da configuração da ferramenta as duas soluções são idênticas, pois elas produzem o mesmo p e R , mas no espaço das juntas são claramente distintas. Tipicamente a solução cotovelo acima é preferida, pois reduz as chances de uma colisão entre os elos do braço e obstáculos em repouso no espaço de trabalho (SCHILLING, 1990).

4. ROBÔ SCORBOT ER-4PC

O robô Scrobot ER-4PC é um robô didático fabricado pela empresa Intelitek de Israel. Os robôs didáticos são fabricados com o uso restrito a ensino, portanto não possuindo todas as características dos robôs utilizados em aplicações industriais, mas reflete em muito a maioria das funcionalidades.

A sua configuração é a do tipo braço antropomórfico, ou seja, é similar a um braço humano, onde todas as suas juntas são do tipo revolução. Este robô possui 5 graus de liberdade, e portanto não é um robô cinematicamente redundante, diante disso não realiza certas tarefas que os robôs industriais com 6 graus de liberdade realizam, por exemplo, o Scrobot não realiza o movimento *yaw*, ou seja, o movimento de guinada ou de rotação lateral do punho da ferramenta.

A fig. 10 mostra um robô Scrobot ER-4PC juntamente com seu controlador e seu *teach pendant*.

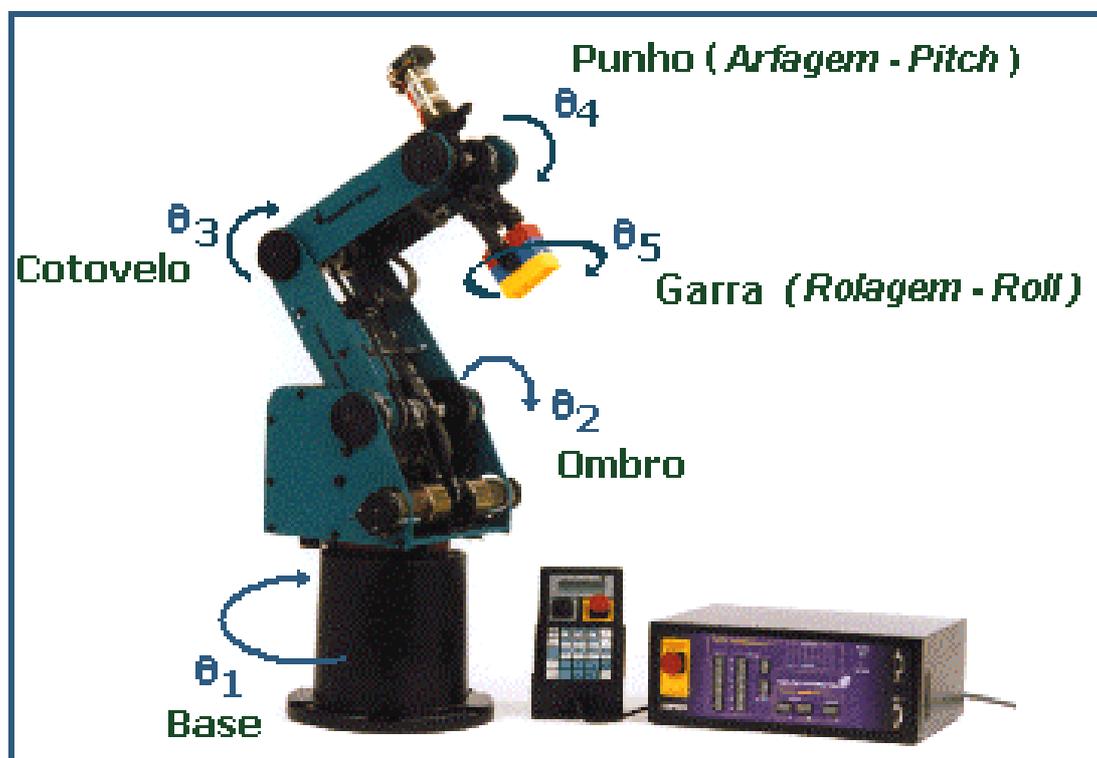


FIGURA 10 – Robô manipulador SCORBOT-ER4PC
Fonte: Intelitek, 2004

Na tab. 2 encontram-se dados técnicos construtivos do robô, onde podem ser encontrados os limites mecânicos de cada junta e na fig. 11 observa-se o envelope de trabalho do robô devido a sua configuração das juntas. Estes dados são importantes para o tratamento das singularidades da solução da cinemática inversa, pois se o ponto desejado estiver fora do envelope de trabalho o robô não conseguirá alcançá-lo, ou se um ponto desejado estiver dentro do envelope de trabalho porém algum ângulo de junta calculado estiver fora dos limites estipulados pelo projeto do robô este também não conseguirá alcançá-lo.

TABELA 2 – Dados Técnicos construtivos Scorbot-ER4PC
Fonte: Intelitek, 2004

| BRAÇO MECÂNICO | |
|-------------------------------|---|
| Estrutura mecânica | Braço vertical articulado; estrutura aberta |
| Graus de liberdade | 5 eixos rotacionais |
| Capacidade de carga | 2,1 kg |
| MOVIMENTO DOS EIXOS | |
| Eixo 1 – Base | 310 ° |
| Eixo 2 – Ombro | +135/35 ° |
| Eixo 3 – Cotovelo | ±130 ° |
| Eixo 4 – Punho | ±130 ° |
| Eixo 5 - Garra | ±570 ° |
| CARACTERÍSTICAS GERAIS | |
| Alcance | 610 mm com pinça |
| Velocidade | 700 mm/seg |
| Repetibilidade | ±0,18 mm |
| Realimentação | Encoders óticos incrementais |
| Home | Micro interruptor em cada eixo |
| Atuadores | Servomotor 12 VDC em cada eixo |
| Pinça | Servomotor DC, 2 dedos paralelos |
| Abertura das pinças | 65/75 mm |
| Transmissão | Engrenagem correia dentada |
| Peso | 10,8 kg |

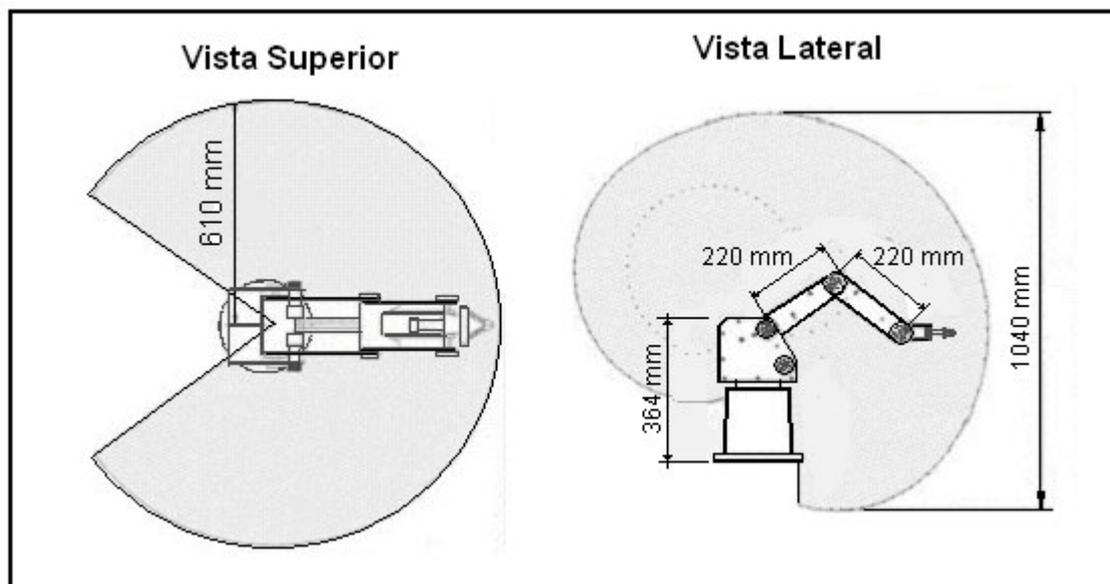


FIGURA 11 – Envelope de Trabalho Scrobot-ER4PC
Fonte: Intelitk, 2004

4.1 CINEMÁTICA DIRETA DO SCORBOT ER4PC

Para a resolução do problema da cinemática direta foi realizado o procedimento de Denavit-Hartenberg(DH) para a construção da equação do braço para o modelo específico de robô.

O robô possui 5 graus de liberdade, portanto 5 juntas variáveis, sendo que todas são juntas de revolução, sendo assim, segundo a definição dos parâmetros cinemáticos já discutidos anteriormente temos os ângulos das juntas θ_k variáveis e os demais parâmetros, d_k , a_k e α_k fixos.

Na fig. 12 é mostrada a configuração do modelo do robô Scrobot com 5 graus de liberdade, após realização do procedimento de DH com todos os parâmetros cinemáticos envolvidos.

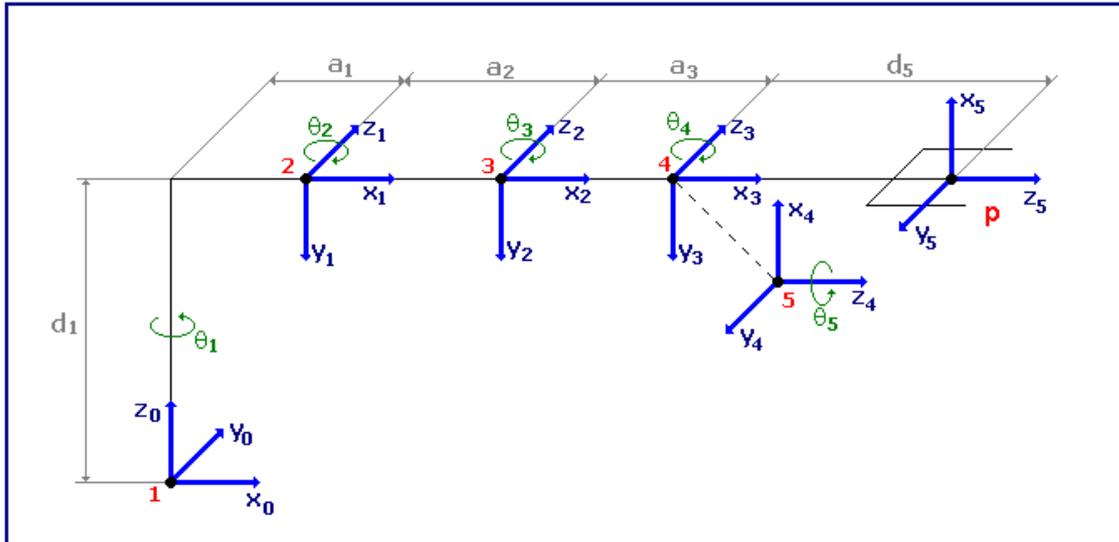


FIGURA 12 – Esquema gráfico do procedimento de DH
Fonte: SCHILLING, 1990

Na tab. 3 é sumarizado os parâmetros cinemáticos do robô Scorbot.

TABELA 3 – Parâmetros cinemáticos do robô Scorbot-ER4PC
Fonte: SCHILLING, 1990

| JUNTA | α_i | a_i | d_i | θ_i | Nome |
|-------|------------|-------|-------|------------|----------|
| 1 | $-\pi/2$ | 45 | 360 | θ_1 | Base |
| 2 | 0 | 220 | 0 | θ_2 | Ombro |
| 3 | 0 | 220 | 0 | θ_3 | Cotovelo |
| 4 | $-\pi/2$ | 0 | 0 | θ_4 | Punho |
| 5 | 0 | 0 | 125 | θ_5 | Rotação |

Tendo em mãos os parâmetros cinemáticos de todas as juntas, e utilizando a matriz de transformação de coordenadas homogêneas que transforma o sistema de coordenadas k-1 para o sistema de coordenadas k da equação 3.1, substitui-se para todas as transformações os parâmetros cinemáticos de cada junta e tem-se:

$${}^0T_1 = \begin{bmatrix} \cos \theta_1 & 0 & -\text{sen } \theta_1 & a_1 \cdot \cos \theta_1 \\ \text{sen } \theta_1 & 0 & \cos \theta_1 & a_1 \cdot \text{sen } \theta_1 \\ 0 & -1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.1)$$

$${}^1T_2 = \begin{bmatrix} \cos \theta_2 & -\text{sen } \theta_2 & 0 & a_2 \cdot \cos \theta_2 \\ \text{sen } \theta_2 & \cos \theta_2 & 0 & a_2 \cdot \text{sen } \theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.2)$$

$${}^2T_3 = \begin{bmatrix} \cos \theta_3 & -\text{sen } \theta_3 & 0 & a_3 \cdot \cos \theta_3 \\ \text{sen } \theta_3 & \cos \theta_3 & 0 & a_3 \cdot \text{sen } \theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.3)$$

$${}^3T_4 = \begin{bmatrix} \cos \theta_4 & 0 & -\text{sen } \theta_4 & 0 \\ \text{sen } \theta_4 & 0 & \cos \theta_4 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.4)$$

$${}^4T_5 = \begin{bmatrix} \cos \theta_5 & -\text{sen } \theta_5 & 0 & 0 \\ \text{sen } \theta_5 & \cos \theta_5 & 0 & 0 \\ 0 & 0 & 1 & d_5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.5)$$

Com todas as matrizes montadas, é realizada a multiplicação sucessiva das matrizes desde a base até a ferramenta, conforme a equação 3.2.

Como resposta a essas multiplicações de matrizes de quarta ordem obtém-se a seguinte matriz resultante.

$${}^0T_5 = \begin{bmatrix} r^1 & r^2 & r^3 & p \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.6)$$

onde:

$$\begin{aligned} r^1_x &= [(\cos \theta_1 \cdot \cos \theta_{234} \cdot \cos \theta_5) + (\text{sen } \theta_1 \cdot \text{sen } \theta_5)] \\ r^1_y &= [(\text{sen } \theta_1 \cdot \cos \theta_{234} \cdot \cos \theta_5) - (\cos \theta_1 \cdot \text{sen } \theta_5)] \\ r^1_z &= (-\text{sen } \theta_{234} \cdot \cos \theta_5) \\ r^2_x &= [-(\cos \theta_1 \cdot \cos \theta_{234} \cdot \text{sen } \theta_5) + (\text{sen } \theta_1 \cdot \cos \theta_5)] \\ r^2_y &= [-(\text{sen } \theta_1 \cdot \cos \theta_{234} \cdot \text{sen } \theta_5) - (\cos \theta_1 \cdot \cos \theta_5)] \end{aligned}$$

$$\begin{aligned}
r_z^2 &= (\text{sen } \theta_{234} \cdot \text{sen } \theta_5) \\
r_x^3 &= -\text{cos } \theta_1 \cdot \text{sen } \theta_{234} \\
r_y^3 &= -\text{sen } \theta_1 \cdot \text{sen } \theta_{234} \\
r_z^3 &= -\text{cos } \theta_{234} \\
p_x &= \text{cos } \theta_1 \cdot (a_1 + a_2 \cdot \text{cos } \theta_2 + a_3 \cdot \text{cos } \theta_{23} - d_5 \cdot \text{sen } \theta_{234}) \\
p_y &= \text{sen } \theta_1 \cdot (a_1 + a_2 \cdot \text{cos } \theta_2 + a_3 \cdot \text{cos } \theta_{23} - d_5 \cdot \text{sen } \theta_{234}) \\
p_z &= d_1 - a_2 \cdot \text{sen } \theta_2 - a_3 \cdot \text{sen } \theta_{23} - d_5 \cdot \text{cos } \theta_{234}
\end{aligned}$$

Como solução para o problema da cinemática direta tem-se então a sub-matriz 3x3 composta pelos vetores r^1 , r^2 e r^3 como a orientação da ferramenta e a sub-matriz 3x1 p como a posição no espaço na ponta da ferramenta.

4.2 CINEMÁTICA INVERSA DO SCORBOT ER4PC

Há dois métodos que podem ser aplicados para resolver o problema da cinemática inversa. Um modelo seria pela abordagem numérica, utilizando funções de otimização com subseqüentes iterações.

Outro método faz o uso da abordagem analítica/geométrica, onde é explorada a equação do braço, que foi resultado do problema da cinemática direta. Este método é considerado mais rápido do que os métodos numéricos, e eles podem ser usados para identificar múltiplas soluções. O principal inconveniente do método analítico está no fato de que a solução é totalmente dependente da configuração do robô. Sendo assim, uma análise separada deverá ser realizada para cada tipo de robô ou classe de robôs.

A solução analítica para o problema da cinemática inversa começa com a utilização da equação 4.6 que é a equação do braço para o robô manipulador.

4.2.1. Junta da base

Sendo que não existe o movimento rotação lateral, o conhecido movimento *yaw*, a junta mais fácil de se encontrar é o ângulo da base θ_1 . Inspeções nas expressões de p_x e p_y na equação 2.6 revelam que eles possuem um fator em comum.

Dividindo p_y por p_x temos como solução.

$$\frac{p_y = \text{sen } \theta_1 \cdot (a_1 + a_2 + a_3 \cdot \cos \theta_{23} - d_5 \cdot \text{sen } \theta_{234})}{p_x = \cos \theta_1 \cdot (a_1 + a_2 \cdot \cos \theta_2 + a_3 \cdot \cos \theta_{23} - d_5 \cdot \text{sen } \theta_{234})}$$

$$\left(\frac{p_y}{p_x} \right) = \left(\frac{\text{sen } \theta_1}{\cos \theta_2} \right) = \tan \theta_1$$

$$\theta_1 = \arctan \left(\frac{p_y}{p_x} \right)$$

$$\theta_1 = \arctan 2(p_y, p_x) \quad (4.7)$$

A função $\arctan 2(y,x)$ acima denota uma versão de quatro quadrantes da função \arctan . Ela permite cobrir ângulos que estão dentro do intervalo $[-\pi, \pi]$. Esta função é usada repetidamente em derivações sucessivas. Ela pode ser implementada utilizando a função de dois quadrantes comum \arctan como mostra a tab. 4, onde a notação **sgn** representa a função signum, ou sign, sendo que $\text{sgn}(\alpha)$ retorna -1 ou 1 dependendo se o valor de α é negativo ou positivo, respectivamente (SCHILLING,1990).

TABELA 4 – Sumarização da Função ArcTan2
Fonte: SCHILLING, 1990

| Arctan para 4 quadrantes | | |
|--------------------------|-----------|------------------------|
| Caso | Quadrante | Arctan2(y,x) |
| X>0 | 1,4 | Arctan(y/x) |
| X=0 | 1,4 | [sgn(y).π/2] |
| X<0 | 2,3 | Arctan(y/x)+[sgn(y).π] |

4.2.2 Junta do Cotovelo

A junta do cotovelo θ_3 é a junta mais difícil de se extrair das equações, pois ela é fortemente acoplada com os ângulos do ombro e o *pitch*. Sendo assim, será iniciada a resolução isolando uma variável intermediária chamada, θ_{234} , chamada de *ângulo global*

da ferramenta. O ângulo θ_{234} será igual a soma dos ângulos θ_2 , θ_3 e θ_4 é o ângulo *pitch* medido em relação a superfície de trabalho ou o plano x^0y^0 .

Para a resolução do problema da cinemática inversa é necessário ter pelo menos 6 equações, sendo assim é utilizado o vetor r^3 da equação 4.6 para auxílio na resolução. Então inspecionando as três equações do vetor r^3 podemos encontrar uma relação com os ângulos 2,3 e 4.

$$r^3_x = -\cos \theta_1 \cdot \text{sen } \theta_{234} \quad (4.8)$$

$$r^3_y = -\text{sen } \theta_1 \cdot \text{sen } \theta_{234} \quad (4.9)$$

$$r^3_z = -\cos \theta_{234} \quad (4.10)$$

Multiplicando a equação 4.8 por $\cos \theta_1$ e a equação 4.9 por $\text{sen } \theta_1$, e somando os as duas igualdades, temos:

$$\text{sen } \theta_{234} = -(\cos \theta_1 \cdot r^3_x + \text{sen } \theta_1 \cdot r^3_y) \quad (4.11)$$

Dividindo a equação 4.11 por 4.10 temos:

$$\frac{\text{sen } \theta_{234}}{\cos \theta_{234}} = \frac{-(\cos \theta_1 \cdot w_x + \text{sen } \theta_1 \cdot w_y)}{-w_z}$$

$$\tan \theta_{234} = \frac{-(\cos \theta_1 \cdot w_x + \text{sen } \theta_1 \cdot w_y)}{-w_z}$$

$$\theta_{234} = \arctan 2[-(\cos \theta_1 \cdot w_x + \text{sen } \theta_1 \cdot w_y), -w_z] \quad (4.12)$$

Precisa-se definir duas outras variáveis intermediárias para auxiliar na resolução dos ângulos 2, 3 e 4. A partir da matriz de posição da equação 4.6 isolam-se dois termos conforme as equações 4.13 e 4.14 (TSAI, 1999).

$$a_2 \cdot \cos \theta_2 + a_3 \cdot \cos \theta_{23} = k_1 \quad (4.13)$$

$$a_2 \cdot \text{sen } \theta_2 + a_3 \cdot \text{sen } \theta_{23} = k_2 \quad (4.14)$$

Com isto, a matriz de posição p será:

$$p_x = \cos \theta_1 \cdot (a_1 + k_1 - d_5 \cdot \text{sen } \theta_{234}) \quad (4.15)$$

$$p_y = \text{sen } \theta_1 \cdot (a_1 + k_1 - d_5 \cdot \text{sen } \theta_{234}) \quad (4.16)$$

$$p_z = d_1 - k_2 - d_5 \cdot \cos \theta_{234} \quad (4.17)$$

Multiplicando a equação 4.15 por $\cos \theta_1$ e a equação 4.16 por $\sin \theta_1$ e somando as duas equações obtemos k_1 e k_2 como sendo:

$$k_1 = p_x \cdot \cos \theta_1 + p_y \cdot \sin \theta_1 - a_1 + d_5 \cdot \sin \theta_{234} \quad (4.18)$$

$$k_2 = -p_z + d_1 - d_5 \cdot \cos \theta_{234} \quad (4.19)$$

Elevando ao quadrado ambos os lados das equações 4.18 e 4.19, temos que:

$$a_2^2 + a_3^2 + 2 \cdot a_2 \cdot a_3 \cdot \cos \theta_3 = k_1^2 + k_2^2$$

Isolando o ângulo temos:

$$\theta_3 = \pm \arccos \left[\frac{k_1^2 + k_2^2 - a_2^2 - a_3^2}{2 \cdot a_2 \cdot a_3} \right] \quad (4.20)$$

Devido à existência de componentes elevados ao quadrado, o ângulo do cotovelo pode ter duas soluções, as conhecidas cotovelo acima e cotovelo abaixo com o valor do θ_3 sendo positivo ou negativo respectivamente.

4.2.3 Junta do Ombro

Expandindo as equações 4.13 e 4.14 temos (TSAI, 1999):

$$\begin{aligned} (a_2 + a_3 \cdot \cos \theta_3) \cdot \cos \theta_2 - (a_3 \cdot \sin \theta_3) \cdot \sin \theta_2 &= k_1 \\ (a_3 \cdot \sin \theta_3) \cdot \cos \theta_2 + (a_2 + a_3 \cdot \cos \theta_3) \cdot \sin \theta_2 &= k_2 \end{aligned}$$

Resolvendo estas equações isolamos o $\cos \theta_2$ e $\sin \theta_2$:

$$C2 = \cos \theta_2 = \left(\frac{k_1 \cdot (a_2 + a_3 \cdot \cos \theta_3) + k_2 \cdot a_3 \cdot \sin \theta_3}{a_2^2 + a_3^2 + 2 \cdot a_2 \cdot a_3 \cdot \cos \theta_3} \right) \quad (4.21)$$

$$S2 = \sin \theta_2 = \left(\frac{-k_1 \cdot a_3 \cdot \sin \theta_3 + k_2 \cdot (a_2 + a_3 \cdot \cos \theta_3)}{a_2^2 + a_3^2 + 2 \cdot a_2 \cdot a_3 \cdot \cos \theta_3} \right) \quad (4.22)$$

Aplicando a função Atan2 obtemos o ângulo do ombro:

$$\theta_2 = \arctan 2(S2, C2) \quad (4.23)$$

4.2.4 Junta do Punho

Com os ângulos do cotovelo e do ombro encontrados isolamos o ângulo do unho θ_4 do termo genérico θ_{234} . Com isto, tem-se:

$$\theta_4 = \theta_{234} - \theta_2 - \theta_3 \quad (4.24)$$

4.2.5 Junta de Rotação

A junta de rotação, θ_5 , é uma junta que está totalmente desacoplada com a posição desejada, mas acoplada sim com a orientação, como podemos observar nas equações 4.7 para a resolução de θ_1 , 4.12 para a resolução de θ_{234} , 4.20 para a resolução de θ_3 , 4.23 para a resolução de θ_2 e 4.24 para a resolução de θ_4 , onde não é encontrado o termo da junta de rotação θ_5 .

O intuito deste trabalho é dar a liberdade ao usuário de escolher a sua forma de abordagem e com isso esta junta pode ser uma escolha pessoal do usuário e não uma solução matemática.

Portanto, esta junta será mantida com o valor de 0 (zero) graus, para todas as operações, e também poderá, em trabalhos futuros ser liberada para escolha do usuário.

5. REALIDADE VIRTUAL

Durante a última década a palavra *virtual* se tornou uma das palavras mais usadas na língua inglesa. Hoje, há universidades virtuais, escritórios virtuais, exposições virtuais, túneis de vento virtuais, atores virtuais, estúdios virtuais, museus virtuais, doutores virtuais – e tudo isto por causa da Realidade Virtual (VINCE, 1999).

Realidade Virtual (RV) surgiu em meados de 1980, e foi o motivo de uma série de conferências, exposições, programas de televisão, e debates filosóficos sobre o significado de Realidade.

5.1 O QUE É REALIDADE VIRTUAL?

Os primeiros sistemas RV descreviam uma tecnologia computacional que habilitava um usuário a olhar através de um visor especial chamado de *Head-Mounted Display* (HMD) e em vez de verem um mundo normal, eles viam um mundo gerado por computador. Uma das mais percebidas vantagens desta abordagem foi o método de integração do usuário com imagens virtuais. Primeiramente, os movimentos da cabeça do usuário são monitorados eletronicamente e realimentam o computador criando as imagens, conforme o usuário move sua cabeça, os objetos em cena se tornam estacionários – apenas como eles fazem na vida real. A maioria dos HMDs previne os usuários de verem o mundo real, o qual, junto com uma visão estereoscópica, imergem os usuários rapidamente num mundo substituto.

Se fosse possível construir um sistema que habilitasse ao usuário mover-se sobre o mundo virtual e interagir-se com o próprio mundo, então se encontrariam aplicações extraordinárias. Deveria ser possível ir a um teatro virtual, fazer um *test drive* num carro virtual, e interagir com atores virtuais numa televisão 3D. Os projetistas de carros poderiam projetar carros conceito e explorá-los num nível virtual muito antes de serem construídos. Cirurgiões poderiam praticar operações num cadáver virtual. As aplicações são ilimitadas.

A RV representa um paradigma de interface entre o usuário e o computador, onde o principal objetivo dessa forma de interface é fazer com que o usuário sinta-se dentro

da mesma (sensação de imersão). A interface tradicional humano computador é uma interface em duas dimensões, x e y, onde são representados elementos como: botões, ícones, letras, imagens, vídeos, áudios e números. Uma interface em realidade virtual é uma interface em 3 dimensões onde não só existem as dimensões x e y do monitor, mas a dimensão z, ou seja, a dimensão de profundidade.

Juntamente com essa nova idéia de interface surgiram também dispositivos especiais que a complementavam como capacetes, luvas, coletes e óculos. Esses dispositivos são os responsáveis pela interação do usuário com as aplicações em realidade virtual. Nesse caso o usuário fica quase que totalmente imerso na interface podendo navegar e interagir com o espaço virtual. Segundo Kirner (KIRNER, 2004), “realidade virtual é uma forma técnica avançada de interface, onde o usuário pode realizar **imersão**, **navegação** e **interação** em um ambiente sintético tridimensional gerado por computador, utilizando canais multi-sensoriais”. Na fig. 13 é mostrado um exemplo de um equipamento de realidade virtual.

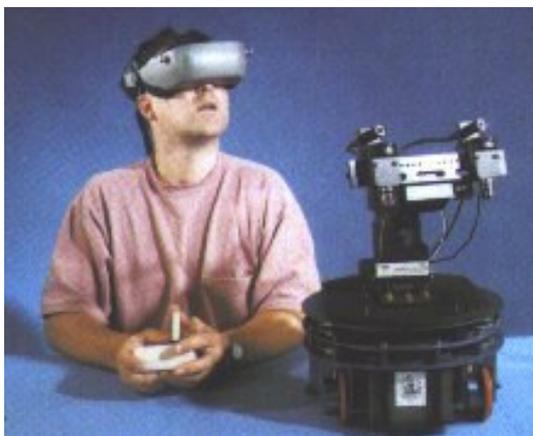


FIGURA 13 – Dispositivos Especiais para RV (Capacete)
Fonte: KIRNER, 2004

Com a Realidade Virtual o ser humano pode vivenciar uma realidade que não existe ou enxergar a própria realidade de uma forma diferente e mais sofisticada de modo que possa facilitar o seu trabalho e/ou aprimorá-lo. A realidade virtual pode ser considerada hoje, a forma mais avançada de interface usuário-máquina.

Um mundo virtual é um ambiente desenvolvido com interface tridimensional - realidade virtual para aplicações específicas controlado interativamente pelo

computador. No mundo virtual os dados não são apenas números ou arquivos, mas também objetos “concretos” como paredes, cadeiras, mesas, e objetos abstratos como temperatura, velocidade e cor. No mundo virtual o usuário realiza operações com dados-objetos como visualização, análise e manipulação. Essas operações podem refletir no comportamento do próprio mundo. O suporte à essa interação é dada por dispositivos periféricos como luvas, capacetes, coletes e óculos.

5.2 ALGUNS CONCEITOS E TERMOS DE RV

A definição de um dicionário típico para a palavra *virtual* é ser alguma coisa em efeito mas não em forma – e as palavras *realidade virtual* são utilizadas como uma visão da realidade sem requerer o físico de porcas e parafusos, ou tudo o que não é necessário para construir realidade (VINCE, 1999).

O que é referido como realidade é baseado em alguma coisa que é chamado de mundo físico externo. Este universo – tudo o que é – pode apenas ser explorado por nossos sensores, e depende-se de do aprendizado do passado para descrever certas experiências em termo de cores, sons, temperatura, cheiro, sabor, toque, etc. Portanto, uma realidade virtual parece sugerir uma realidade que é acreditável, e também não existe fisicamente. Mas o que fazemos com o termo acreditável? Bem acreditar significa *aceitar como real ou verdadeiro*. Mas por quanto tempo podemos estar iludidos por uma realidade alternativa? Um segundo? Um minuto? Uma hora? Além disso, o que representa ser verdade para nós?

5.2.1 Navegação e Interação

Basicamente, RV é a utilização de computadores para criar imagens de cenas 3D com as quais alguém pode navegar e interagir. A palavra navegar implica a habilidade de mover-se ao redor e explorar características de uma cena 3D tal como uma construção; enquanto interação implica a habilidade de selecionar e mover objetos na cena, tal como uma cadeira.

Para navegar e interagir requer-se gráficos em tempo real, os quais implicam

computadores velozes. Navegação e interação com o mundo real tem certas vantagens se estamos equipados com visão estereoscópica, e a mesma vantagem é realizada se nossas imagens geradas por computador são também estereoscópicas. Naturalmente, tal fato demanda uma capacidade maior de processamento do computador.

5.2.2 Imersão e Presença

Em 1965 Ivan Sutherland publicou um artigo “*The Ultimate Display*”, que descreveu como um dia, o computador proverá uma janela dentro de um mundo virtual. Em 1968 ele construiu um HMD que apresentava ao usuário visões direita e esquerda de uma cena 3D gerada por computador, tal que quando a cabeça do usuário se movia, a cena virtual permanecia estacionária. As imagens foram de longe fatos reais – elas foram simples linhas desenhadas. Mas como elas eram estereoscópicas o usuário recebia uma impressão de estar olhando para um objeto sólido 3D. A Realidade Virtual havia nascido (VINCE, 1999).

Infelizmente, as décadas de 1960 e 1970 RV ficaram dormentes, principalmente pela dificuldade de hardware que fosse de baixo custo e eficiente suficiente para as cargas pesadas das imagens. Durante a década de 1980, gráficos em tempo real se tornaram uma realidade, e RV se tornou novamente uma realidade comercial. Inicialmente, os primeiros sistemas RV comprimiam um sistema computacional em tempo real, um HMD, e uma luva interativa. O HMD imerge o usuário num mundo virtual pela privação do usuário ver um mundo real. A imersão aumenta a sensação da presença dentro do mundo virtual, e para algumas pessoas, a imersão distingue sistemas RV de outros tipos de sistemas gráficos em tempo real nascido (VINCE, 1999). Para esta comunidade, um sistema RV tem provido um usuário com uma visão em primeira-pessoa. Olhando para uma tela de uma estação não era realidade virtual – era apenas gráficos computacionais.

5.2.3 Realidade Virtual Imersiva e Não Imersiva

Durante a década de 1990 muitos relaxam em suas visões duras de o que compreendia os sistemas RV, e aceitaram uma definição mais ampla para RV. Os

sistemas PC foram dotados com a capacidade de mostrar imagens em tempo real de ambientes 3D que poderiam ser navegados e suportando interações, e também eles poderiam ser configurados para suportar um HMD. Obviamente, eles eram um sistema RV – mas não tão poderosos quanto seus colegas mais caros.

Diante disto, dois tipos de realidade virtual foram definidas, a imersiva e a não-imersiva. A realidade virtual imersiva é feita com o uso de periféricos de interface RV como capacetes, luvas e salas de projeção. O usuário fica totalmente imerso no mundo virtual sem qualquer visualização do mundo real. A grande vantagem desse tipo de RV é o fator imersividade que dá mais realismo e resultado coerente às aplicações, bem como a possibilidade de forte interação com os sentidos humanos. É a realidade virtual plena. Já a realidade virtual não-imersiva é feita através do uso de monitores. Nesse caso o usuário entra em contato com o mundo virtual através da tela do computador sem, no entanto, perder contato visual com o mundo real. Uma das vantagens desse tipo de realidade virtual é o fato da mesma poder dispor dos avanços da tecnologia computacional, embora o fator não-imersividade seja uma desvantagem.

5.3 VRML

Em 1994, na conferência anual da *World Wide Web(WWW)* em Genebra na Suíça, Tim Barness-Lee e Dave Roggett organizaram uma sessão para discutir interfaces de realidade virtual na *WWW*. Foi aí que surgiu a idéia da criação de uma linguagem “*Markup*” – análoga ao HTML - para a especificação de mundos 3D e *hyperlinks* chamada inicialmente de VRML (*Virtual Reality Markup Language*). O Termo “*Markup*” viria, depois, a ser substituído por *Modeling* (modelagem), ficando a VRML conhecida como linguagem de modelagem de realidade virtual.

VRML é uma linguagem interpretada por *browsers* que descreve objetos tridimensionais formados através de primitivas geométricas. Essas primitivas são predefinidas na linguagem, embora os programadores possam definir objetos complexos criados a partir das primitivas anteriores. Estes objetos podem receber atributos e propriedades como cor, textura e mapeamento, transparência, brilho, deformação, translações e rotações.

Na primeira versão do VRML não existia nenhum tipo de movimento e

interação entre os usuários e os objetos. O VRML 1.0 produz apenas mundos estáticos, sem movimentos, onde basicamente, a única interação com o usuário é feita através de *links*. Na segunda versão do VRML foram introduzidas noções de movimentos através de interpolação e alguns *scripts* foram incorporados à linguagem. A linguagem passou a ter suporte a *scripts* Java, o que abriu novas possibilidades para aplicações de alto nível de interatividade com outros sistemas e com os próprios usuários.

Uma das grandes vantagens do VRML é que ela é uma linguagem totalmente independente de plataforma. Apesar de existirem diversos compiladores para diversas plataformas, sempre que se migra o programa, muitas alterações são feitas. Após todas estas alterações, é necessário compilar o código fonte novamente. Uma linguagem totalmente independente de plataforma, pode migrar de uma plataforma para outra sem a necessidade de se alterar uma só linha de código fonte. Dentre algumas das linguagens que vem realmente cumprindo essa promessa, podemos citar a Java e a VRML.

O VRML é uma linguagem interpretada por um *browser* VRML ou qualquer *browser* que tenha o *plug in* para interpretar esse formato de arquivo. Existem vários *plug ins* VRML disponíveis no mercado, inclusive para plataforma Linux. A maioria de acesso gratuito e tamanho pequeno. Dentre os mais eficientes podemos citar o *Cortona* da *Parallel Graphics*, o *Cosmo Player* da *Platinum Technologies* e o *Blaxxun Interactive*.

Outra grande vantagem da segunda versão da linguagem VRML é a capacidade de interação com diversos tipos de mídia. No VRML a propriedade cor dos objetos pode ser substituída pela propriedade textura, é nessa propriedade que fica a porta para o suporte das mídias no VRML. As texturas podem ser arquivos JPG, GIF, BMP bem como vídeos MPG. Como a textura representa a aparência dos objetos, pode-se então exibir imagens e vídeos nos mundos virtuais.

O VRML 2.0 também dá suporte a arquivos de áudio nos formatos mais conhecidos da internet como o WAV. O VRML incorpora arquivos como som de fundo do ambiente podendo-se definir o nível de volume e as dimensões aonde o som será ouvido. O som escutado no ambiente VRML é, assim como seus objetos, tri-dimensional. A proximidade ou afastamento da fonte sonora faz com que o usuário tenha a sensação de aumento ou diminuição do volume do som, bem como sua transferência de intensidade de um ouvido para o outro.

6. IMPLEMENTAÇÃO

O simulador do grupo LARVA é um ambiente virtual 3D que apresenta um robô manipulador Scorbot-ER4PC que pertence à Universidade do Estado de Santa Catarina – UDESC.

A interface do simulador compreende em uma área de visualização 3D, que está inserido *o plug in* para VRML, pode-se constatar isto na fig. 15, na caixa 1. Além da visualização dos movimentos realizados pelo robô, também é disponibilizado, ao usuário, uma interação com o robô virtual através do *mouse*, ou seja, ao manter pressionado o botão esquerdo do *mouse* sobre uma junta e arrasta-lo, a junta realizará um movimento que será determinado pela direção que é arrastado o *mouse*.

Há também, uma interface para entrada e saída de dados, como pode ser vista na caixa 2 da fig. 14, onde o usuário pode movimentar as juntas através de botões, ou definir diretamente numa caixa de texto o valor dos ângulos correspondente de cada junta. Ao alterar os valores dos ângulos das juntas, quer seja através dos botões ou pelas caixas de texto, o robô virtual irá automaticamente assumir os valores desejados e a interface irá fornecer, através de caixas de texto, os valores das componentes da posição da ponta da ferramenta.

Outra opção que o simulador disponibiliza é a criação de um simplificado programa para simulação dos movimentos do robô, como pode ser visto na caixa 3 da fig. 14.

Na versão atual, em termos de funções cinemáticas, o simulador dispõe apenas da resolução da cinemática direta, sendo de duas formas, uma através da hierarquia gráfica modelada, onde ao movimentar o robô através das juntas tem-se uma noção da posição final da ferramenta. A outra forma é através da função matemática implementada no simulador que realiza o cálculo da cinemática direta para cada configuração de junta especificada pelo usuário.

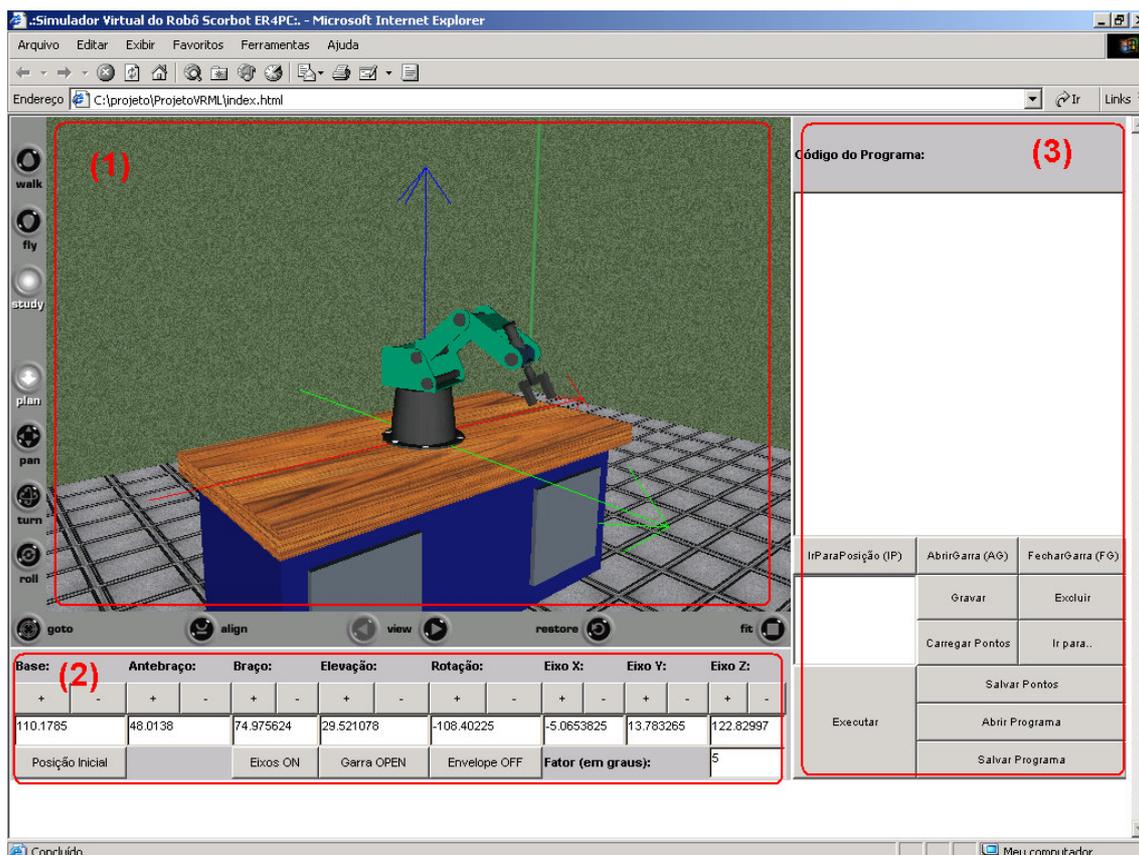


FIGURA 14 – Simulador do grupo LARVA
Fonte: HOUNSELL e REDEL, 2003

6.1 IMPLEMENTAÇÃO DO CONTROLE DE ABORDAGEM

A implementação do controle de abordagem para o robô manipulador Scorbot-ER4PC foi dividida em etapas, que são: *implementação da função cinemática inversa*, *implementação das soluções possíveis* e *o controle da abordagem*.

6.1.1 Implementação da cinemática inversa

A implementação da função da cinemática inversa para o robô SCORBOT-ER4PC foi realizada em três etapas. Na primeira etapa foi criada uma planilha eletrônica com o *software* Microsoft Excel, contendo as funções da cinemática direta e inversa. A planilha pode ser vista no anexo 1, onde foi implementada a primeira versão para o cálculo das duas cinemáticas e foram realizados testes como por exemplo, entrar

com os valores dos ângulos das juntas e a planilha realizar o cálculo da cinemática direta e após o cálculo da cinemática inversa e retornar os valores dos ângulos das juntas. Com isto pôde-se constatar a reciprocidade das funções, ou seja, o cálculo da cinemática inversa retorna os mesmos valores dos ângulos das juntas que são alimentadas na planilha.

Na segunda etapa foi implementada uma função em VRML para o cálculo da cinemática inversa, que pode ser vista no anexo 2. Porém, a função arco tangente 2 que é necessária para o cálculo não é função matemática disponível pelo VRML, sendo assim foi necessária a inclusão de uma função específica para o cálculo deste arco tangente. A função pode ser vista no anexo 3.

E por fim, como terceira etapa, foi implementada uma função em Java que visa, basicamente, realizar os cálculos matemáticos para definição dos valores dos ângulos das juntas. Na figura 15 pode ser observada a função *calcularInversa*. Esta função recebe sete parâmetros de entrada, que são: a posição no espaço da ponta da ferramenta – *posX*, *posY* e *posZ*; o vetor orientação que foi definido para o controle da abordagem, ou seja, o vetor r^3 – *orX*, *orY* e *orZ*; e uma variável booleana para determinação das múltiplas soluções, que no caso, são as soluções cotovelo acima e cotovelo abaixo – *braço*.

Para a definição das variáveis, na implementação, foi adotado seguir com a mesma notação já utilizada na versão inicial. Com isto, a nova notação para os ângulos das juntas que será adotada, de aqui em diante na implementação com sua respectiva notação utilizada até agora, será:

- *angBase* – Base - θ_1 ;
- *angantebraco* – Ombro - θ_2 ;
- *angBraco* – Cotovelo - θ_3 ;
- *angelevacao* – Punho - θ_4 .

```

public void calcularInversa( float posX, float posY, float posZ, float orX, float orY,
float orZ, boolean braco)
{
    float a1=45; float a2=220; float a3=220; float d1=360; float d5=125;
    // Cálculo da Junta da Base  $\theta_1 \rightarrow$  Equação 4.7
    float angBase = (float) Math.atan2(posY, posX);

    if (posX= =1)
    {
        posX=Math.cos(angBase);
        posY=Math.sen(angBase);
    }
    else if (posX= = -1)
    {
        posX= - Math.cos(angBase);
        posY= - Math.sen(angBase);
    }
    // Cálculo da variável auxiliar  $\theta_{234} \rightarrow$  Equação 4.12
    float auxA = (float) ((orX*Math.cos(angBase))+orY*Math.sin(angBase));
    float auxB = -orZ;
    float ang234= (float) Math.atan2(auxA, auxB);

    // Cálculo das variáveis auxiliares  $k_1$  (Equação 4.18) e  $k_2$  (Equação 4.19)
    float k1 = (float) ((posX*Math.cos(angBase))+posY*Math.sin(angBase))-
a1+(d5*Math.sin(ang234));
    float k2 = (float) (-posZ+d1-(d5*Math.cos(ang234)));

    // Cálculo da Junta do Cotovelo  $\theta_3 \rightarrow$  Equação 4.20
    float angBraco = (float) (((braco == true) ? 1 : (-1)) *
Math.acos((Math.pow(k1,2)+Math.pow(k2,2)-Math.pow(a2,2)-
Math.pow(a3,2))/(2*a2*a3)));

    // Cálculo das variáveis auxiliares  $C_2$  (Equação 4.21) e  $S_2$  (Equação 4.22)
    float c2 = (float)
(((k1*(a2+(a3*Math.cos(angBraco))))+(k2*a3*Math.sin(angBraco)))/(Math.pow(
a2,2)+Math.pow(a3,2)+(2*a2*a3*Math.cos(angBraco))));
    float s2 = (float)
((-k1*a3*Math.sin(angBraco))+k2*(a2+(a3*Math.cos(angBraco))))/(Math.pow(
220,2)+Math.pow(a3,2)+(2*a2*a3*Math.cos(angBraco))));

    // Cálculo da Junta do Ombro  $\theta_2 \rightarrow$  Equação 4.23
    float angantebraco = (float) (Math.atan2(s2,c2)+ Math.PI/2);

    // Cálculo da Junta do Punho  $\theta_4 \rightarrow$  Equação 4.24
    float angelevacao = ang234 - angBraco - angantebraco + (float) Math.PI/2;
}

```

FIGURA 15 – Código da função da cinemática inversa em Java
Fonte: Simulador Larva

6.1.2 Implementação das possíveis soluções

Segundo Schilling (1990), pode-se condensar o número de possíveis soluções para o caso da cinemática inversa em dois. Matematicamente é possível encontrar até oito soluções, porém como foi definido que a junta de rotação terá seu valor constante de 0° , não poderá ser utilizado a outra solução que seria 180° .

Outra opção de solução seria a junta da base que poderia haver uma solução θ_1 e $\theta_1 + \pi$, porém esta última solução não é viável pois o robô estaria trabalhando “de costas” para o objeto.

Então para o robô Scorbot-ER4PC será considerado duas possíveis soluções que são as conhecidas *cotovelo acima e cotovelo abaixo*.

As figs. 16 e 17 mostram as soluções cotovelo acima – fig.16, onde o ângulo θ_3 tem seu valor positivo, e cotovelo abaixo – fig. 17, onde o ângulo θ_3 tem seu valor negativo.

Para uma mesma posição no espaço, que nestes exemplos das fig. 18 e 19 é:

$$\rightarrow p_x = -52,40 \quad \rightarrow p_y = 90,76 \quad \rightarrow p_z = 64,78$$

Encontramos as duas soluções sendo:

$$\text{Cotovelo acima: } \theta_1 = 120; \quad \theta_2 = 45; \quad \theta_3 = 45; \quad \theta_4 = 40; \quad e \quad \theta_5 = 0;$$

$$\text{Cotovelo abaixo: } \theta_1 = 120; \quad \theta_2 = 90; \quad \theta_3 = -45; \quad \theta_4 = 85; \quad e \quad \theta_5 = 0;$$

No código em Java, da fig. 15 pode-se observar no cálculo do ângulo *angBraco*, que é o ângulo θ_3 é multiplicado pelo resultado de uma função que retorna 1 se a variável booleana é verdadeira e -1 se falsa. Sendo que o ângulo $\theta_{2,3,4}$ apenas depende da junta da base e do vetor de orientação, a diferença no cálculo destas duas soluções está na variação dos três ângulos, *ombro, cotovelo e punho*.

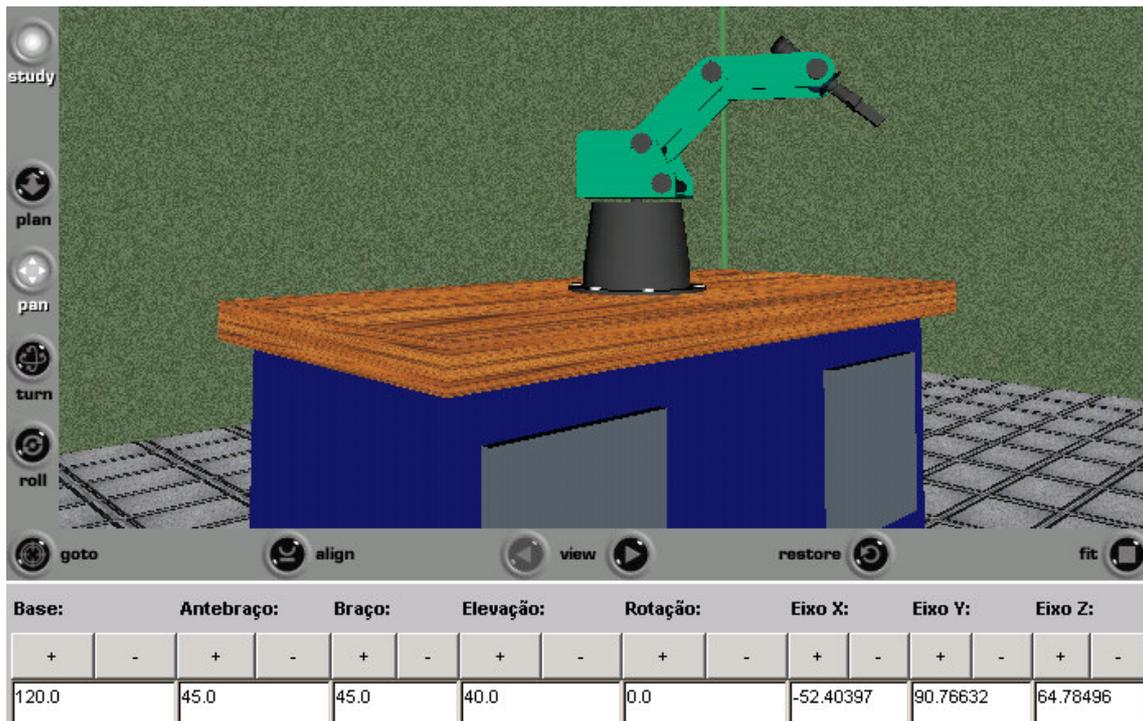


FIGURA 16 – Solução Cotovelo acima do robô Scorbot-ER4PC
Fonte: Simulador LARVA

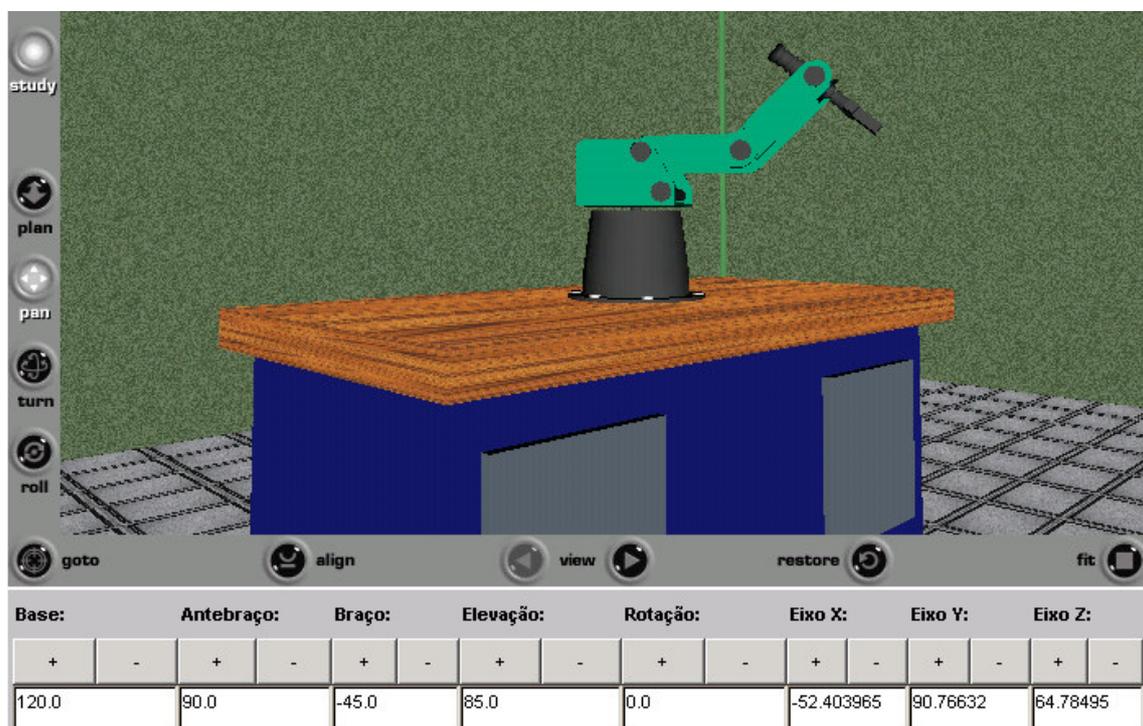


FIGURA 17 – Solução Cotovelo abaixo do robô Scorbot-ER4PC
Fonte: Simulador LARVA

6.1.3 Implementação do controle da abordagem

Para a implementação do controle da abordagem, serão analisadas, antes de iniciar a implementação, algumas questões sobre os vetores que farão parte das informações fornecidas pelo usuário, ou seja o vetor posição e o vetor orientação.

O vetor posição é o vetor mais simples pois compreende simplesmente a posição no espaço desejada pelo usuário. Já o vetor orientação será o vetor que irá ser tratado no controle da abordagem e que será foco do presente estudo.

Primeiramente, precisamos definir como será este vetor de abordagem. A resposta da cinemática direta compreende numa matriz 3x3 de orientação, que é a orientação do sistema de coordenadas da ferramenta em relação ao sistema de coordenadas da base. Do exemplo da fig.18 pode-se tirar as seguintes conclusões, onde x_0, y_0 e z_0 formam o conjunto de coordenadas da base e x_5, y_5 e z_5 formam o conjunto de coordenadas da ponta da ferramenta:

- O vetor x_5 possui a mesma direção e sentido que y_0 ;
- O vetor y_5 possui a mesma direção e sentido que x_0 ;
- O vetor z_5 possui a mesma direção e sentido contrário a z_0 .

| | x_5 | y_5 | z_5 |
|-------|-------|-------|-------|
| x_0 | 0 | 1 | 0 |
| y_0 | 1 | 0 | 0 |
| z_0 | 0 | 0 | -1 |

FIGURA 18 – Exemplo de relação entre coordenadas da base e ferramenta
Fonte: SCHILLING, 1990

Segundo o algoritmo de DH, é nomeado o sistema de coordenadas da ferramenta como o seguinte:

- vetor aproximação = z_n ;
- vetor deslizamento = y_n ;
- vetor normal = x_n .

Na fig. 19 verifica-se a composição do sistema de coordenadas da ferramenta.

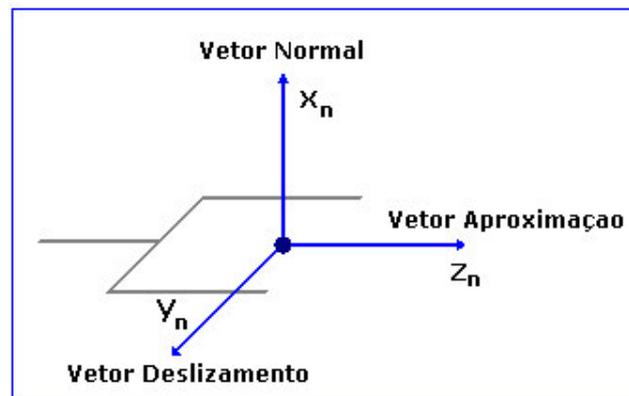


FIGURA 19 – Composição do sistema de coordenadas da ferramenta
Fonte: SCHILLING, 1990

Diante disso, será escolhido o vetor z_n para o controle da abordagem, que nada mais é que o vetor orientação já definido r^3 . Serão escolhidas algumas opções para a abordagem da ferramenta para que o usuário possa analisar, visto que pode-se definir inúmeras formas de abordagem para um determinado ponto no espaço. As opções, fornecidas pelo simulador serão as seguintes, desde que não haja restrições devido a singularidades do sistema, que serão vistas mais a diante:

- Abordagem de cima;
- Abordagem de baixo;
- Abordagem de frente;
- Abordagem de trás;

Da resposta da cinemática direta temos as seguintes equações para o vetor de orientação r^3 .

$$r^3_x = -\cos \theta_1 \cdot \text{sen } \theta_{234}$$

$$r^3_y = -\text{sen } \theta_1 \cdot \text{sen } \theta_{234}$$

$$r^3_z = -\cos \theta_{234}$$

Nas abordagens de cima e de baixo, não temos nenhuma componente de r_x^3 e r_y^3 e assim para a abordagem de cima r_z^3 será igual a -1 , ou seja, mesma direção porém com sentido contrario a z_0 . Para o caso da abordagem de baixo r_z^3 será igual a 1 , ou seja mesma direção e sentido que z_0 . Isto pode ser verificado no vetor o onde o fator $\text{sen}\theta_{234}$ é igual a zero ou 180 dependendo da abordagem de cima ou de baixo. Na fig. 20.a percebe-se o caso de uma abordagem de cima onde o vetor orientação, que nesta figura está representada como z_5 é constituído apenas por componentes de z_0

Para as abordagens de frente e de trás observando as figs. 20.b e 20.c constata-se que o vetor r^3 não possui nenhuma componente do vetor z_0 e uma combinação do vetor x_0 e y_0 . Esta combinação como podemos observação na fig. 20.c é dependente do ângulo da base θ_1 .

Sendo que o ângulo θ_1 é dependente apenas das posições p_y e p_x , quando o usuário definir a posição desejada, o sistema poderá definir as componentes do vetor orientação. Neste caso o fator $\text{sen}\theta_{234}$ será igual a 1 ou -1 dependendo da abordagem ser de frente ou de trás.

Como pode ser verificado nas configurações apresentadas na fig. 20 não há a possibilidade de haver uma abordagem pela lateral, apenas de frente, de trás, de cima, de baixo e inclinado, isto é porque a ferramenta apenas consegue trabalhar a sua aproximação no plano xz .

A razão disto está no fato de que o robô Scorbot-ER4PC possui apenas 5 DOF's, faltando uma junta no punho, a junta da guinada(*yaw*), que é uma junta de revolução que rotaciona a ferramenta sobre o eixo do vetor normal ou, neste caso, o vetor x_5 , realizando uma rotação lateral da ferramenta.

6.1.4 Implementação da Interface

Diante da função do cálculo da cinemática inversa, o simulador teve de ser alterado para comportar a interface com o usuário. Na fig. 21 são mostradas as alterações realizadas no simulador. Na caixa 1 é mostrado um botão para definição das soluções *cotovelo acima* e *cotovelo abaixo*. A descrição do botão está informando *braço acima* e *braço abaixo* para continuar com a mesma nomenclatura utilizada na versão anterior do simulador.

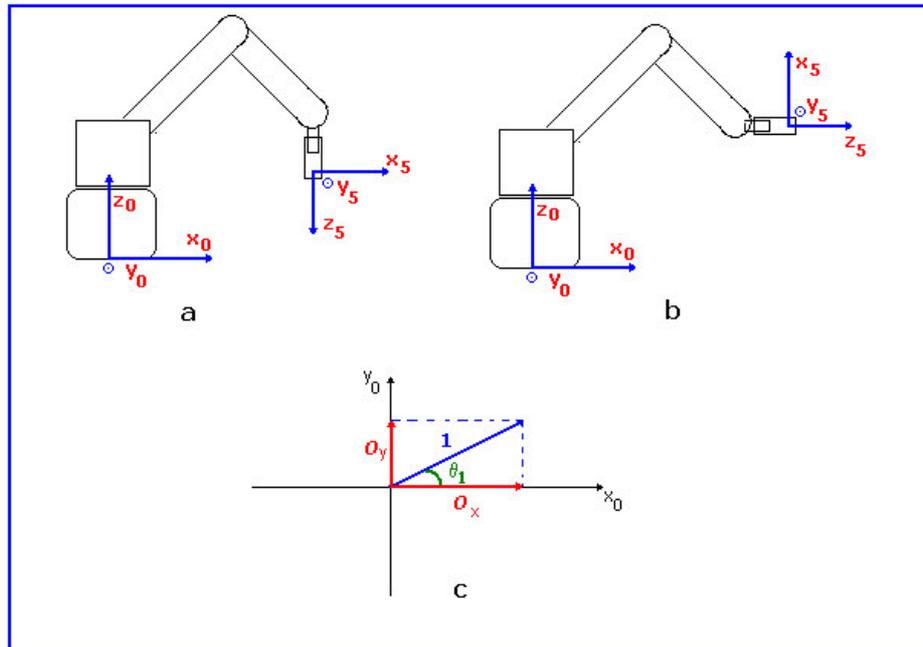


FIGURA 20 - Posições da Ferramenta do Robô para Controle da Abordagem

..Simulador Virtual do Robô Scorbot ER-4PC.. - Microsoft Internet Explorer

Arquivo Editar Exibir Favoritos Ferramentas Ajuda

Endereço C:\Meus documentos\Rudimar\pessoal\tcc\ProjetoVRMLJavaV2\index.html

walk
fly
study
plan
pan
turn
roll
goto align view restore fit

Código do Programa:

IfParaPosição (IP) AbrirGarra (AG) FecharGarra (FG)

Gravar Excluir

Carregar Pontos Ir para..

Executar Salvar Pontos

Abriir Programa

Salvar Programa

| Base: | Antebraço: | Braço: | Elevação: | Rotação: | Eixo X: | Eixo Y: | Eixo Z: |
|-----------------|---------------------------------|-----------|-------------|-------------|-------------------|-----------|-----------|
| + | - | + | - | + | - | + | - |
| 107.97234 | 33.34859 | 76.512955 | 69.631805 | 0.0 | 115.388245 | 355.71268 | 344.03537 |
| Posição Inicial | Cima Baixo Frente Trás | Eixos ON | Garra CLOSE | Braço ACIMA | Fator (em graus): | | 5 |

(2) (1)

Concluído Meu computador

FIGURA 21 – Novo Simulador
Fonte: Simulador LARVA

Ao pressionar o botão de opção das múltiplas soluções um código em Java é acionado para setar uma variável booleana chamada *bracoAcima*, conforme é mostrado na fig. 22. Como pode ser observado no código da fig. 22 quando o botão, aqui chamado de, *bracoInversa* for pressionado é setado a variável *bracoAcima* e alterada a descrição do botão.

```
(.....)
// Ao pressionar o botão para seleção da solução...
else if (event.getSource() == this.bracoInversa)
{
    this.bracoAcima = !this.bracoAcima;
    if (this.bracoAcima)

this.bracoInversa.setLabel(PrincipalControles.BRACO_ACIMA);
    else
    this.bracoInversa.setLabel(PrincipalControles.BRACO_ABAIXO);
}
(.....)
```

FIGURA 22 – Parte do código Java para determinar opção de múltipla escolha
Fonte: Simulador LARVA

Para a escolha do tipo de abordagem que o robô irá realizar, foi incorporado à interface um objeto lista contendo as quatro opções de abordagem. Na fig. 21 pode ser observado, na caixa 2, a lista contendo as quatro opções de abordagem, para escolha do usuário.

Na fig. 23 é mostrado parte do código em Java que realiza a captura do vetor orientação definido pelo usuário. Sendo que já existia uma classe chamada *Eixo* na qual é definido um vetor de três posições, foi instanciado um objeto chamado *abord* para receber o vetor orientação definido pela seleção na lista.

A lista, conforme mostra o código na fig. 24, foi criada já com os valores definidos do vetor orientação, conforme a seleção do usuário, como, por exemplo, pode ser observada a seleção “cima” cria uma instancia de *Eixo* chamada *temp* com um vetor de orientação (0,0,-1).

Podem ser verificados na fig. 24, que para chamar a função *calcularInversa*, é necessário que seja pressionada a tecla <ENTER> em qualquer uma das três caixas de texto da posição desejada. Neste momento, os valores das caixas de texto são

armazenados, o vetor orientação é armazenado, através da seleção da lista, e então é chamada a função *calcularInversa* passando todos os parâmetros necessários.

```
(.....)
// Ao pressionar a tecla enter dentro de uma das caixas de texto da posição desejada.
else if ((evt.getSource() == this.posicaoX) || (evt.getSource() == this.posicaoY) ||
(evt.getSource() == this.posicaoZ))
{
    // Armazena os valores do vetor da posição desejada, nas caixas de texto
    float posX = Float.valueOf(this.posicaoX.getText()).floatValue();
    float posY = Float.valueOf(this.posicaoY.getText()).floatValue();
    float posZ = Float.valueOf(this.posicaoZ.getText()).floatValue();

    // Armazena no objeto abord o vetor de abordagem, a partir da seleção do
    //usuário na lista.
    Eixo abord = (Eixo)this.listaAbordagem.getObjetoSel();

    // Chama função calcularInversa, passando todos os parâmetros necessários
    this. robo.calcularInversa(posX, posY, posZ, abord.getX().floatValue(),
abord.getY().floatValue(), abord.getZ().floatValue(), this.bracosAcima);
    this.atualizaTexto();
}
(.....)
```

FIGURA 23 – Parte do código Java para escolha do tipo de abordagem e chamada da função *calcularInversa*.

Fonte: Simulador LARVA

```
(.....)
this.addComponente((this.listaAbordagem = new RList()), 2, 3, 2, 1);
Eixo temp = new Eixo(new Float(0),new Float(0), new Float(-1));
temp.setString("Cima");
this.listaAbordagem.add(temp);
temp = new Eixo(new Float(0),new Float(0), new Float(1));
temp.setString("Baixo");
this.listaAbordagem.add(temp);
temp = new Eixo(new Float(1),new Float(0), new Float(0));
temp.setString("Frente");
this.listaAbordagem.add(temp);
temp = new Eixo(new Float(-1),new Float(0), new Float(0));
temp.setString("Trás");
this.listaAbordagem.add(temp);
(.....)
```

FIGURA 24 – Parte do código Java para definição do objeto *listaAbordagem*

Fonte: Simulador LARVA

7. CONCLUSÕES

O estudo da cinemática inversa é fundamental para realização de tarefas de movimentação dos robôs manipuladores, pois auxilia o controlador a realizar movimentos que consumam menos energia, ou seja, realizar um movimento, partindo de um ponto para outro, movimentando um menor número de juntas possíveis. Outra importância da cinemática inversa está na sua utilização em linguagens de programação de robôs *off-line*. Quando não se possui um *teach pendant*, que é o principal recurso de programação das linguagens *on-line*, para ensinar o robô a sua posição e orientação no espaço, o ambiente necessita de recursos para definição dos pontos de operação. Os objetivos de implementar a cinemática inversa e a interface foram alcançados, provendo ao novo simulador a opção da interação simulador-usuário para o problema da cinemática inversa. Muitas dificuldades foram encontradas durante a realização do trabalho, principalmente devido a pouca bibliografia existente e a pouca experiência por parte do acadêmico, com isto tornando lenta a seqüência do cronograma e como conseqüência limitando o tempo para o refinamento do simulador.

O trabalho aqui apresentado foi de grande valia, para todos, pois foram pesquisados vários conceitos relativos a cinemática de robôs manipuladores, e foram várias as discussões sobre aspectos importantes na resolução do problema da cinemática inversa. Após conclusão dos trabalhos pôde ser percebido o amadurecimento dos conceitos de robótica por parte do aluno e a satisfação de ter realizado o trabalho definido com êxito.

Este trabalho ainda contribuiu no sentido de documentar a aplicação da solução analítica de forma completa e detalhada a um robô real com seus parâmetros e seqüência na íntegra. Desta feita, tem-se uma exemplificação completa de como resolver a cinemática inversa para um determinado robô, exemplo este que não existe na literatura desta forma. O desenvolvimento relatado está correto pois as implementações algorítmicas confirmaram a coerência ente a cinemática direta e inversa.

7.1 TRABALHOS FUTUROS

Como trabalhos futuros alguns aspectos podem ser melhorados otimizando o simulador, como a implementação de um controle de abordagem para ângulos

inclinados, disponibilizando ao usuário maior flexibilidade para a utilização da cinemática inversa. Dentro desta flexibilidade, também dispor uma forma de manipular diretamente no ambiente virtual o ponto p e o robô ir calculando a cinemática inversa, e inclusive realizando testes de limites dos ângulos, pois foram encontradas algumas situações em que a resposta da cinemática inversa retornava NaN , tipicamente uma resposta devido a divisões por zero. Deve-se ir testando cada cálculo de cada junta para seguir os cálculos se houver uma situação de divisão por zero ou outra singularidade.

Para trabalhos futuros que necessitem da cinemática inversa pode ser sugerida a definição de uma trajetória, ou seja, movimentar um robô de um ponto para o outro seguindo uma linha reta, por exemplo, ou realizando uma curva.

E com a realização desta trajetória, pode-se realizar estudos de métodos de controle de trajetória para realizar estes movimentos. Existem vários tipos de controle de trajetória, como o de menor energia, menor tempo, menor espaço percorrido, e com o simulador podem ser realizados estudos comparativos entre métodos. O simulador neste caso seria uma ferramenta de análise cinemática apenas, pois no estudo destes métodos outros aspectos são envolvidos principalmente aspectos dinâmicos.

Outro trabalho interessante seria a inclusão de uma nova junta, por exemplo uma junta virtual de guinada, provendo ao robô 6 DOF's e com isto se tornando um robô cinematicamente redundante, e como consequência com múltiplas soluções, podendo com isto, até realizar um estudo comparativo entre a solução analítica/geométrica, aqui apresentada, e a solução numérica, pois quando se aumenta complexidade construtiva do robô aumenta também a complexidade algébrica, e assim a solução numérica se torna mais interessante.

ambém um trabalho que envolve mais questões gráficas, seria um controle de “pegada” ou captura de um objeto pelo robô virtual, ou seja, geralmente não existe uma fidelidade na inclusão de novos filhos dentro de uma hierarquia de objetos virtuais, com isto, por exemplo, quando o robô virtual pegar um objeto, simplesmente este objeto será incorporado à hierarquia do robô. Mas será que o objeto foi pego de maneira correta? Ou será que o robô real, ao realizar este procedimento irá realmente pegar o objeto? Diante disto os movimentos e tarefas do robô virtual, no simulador, irá desenvolver cada vez a fielmente as operações realizadas pelo robô real.

REFERÊNCIAS BIBLIOGRÁFICAS

- CAREY, Rikk e BELL, Gavin. **The annotated VRML 2.0 reference manual**. Addison-Wesley Developers Press, c1997. ISBN 0201419742;
- FLÜCKIGER, Lorenzo et al. **Generic robotic kinematic generator for virtual environment interfaces**. *SPIE Telem manipulator and Telepresence Technologies III, Vol. 2901, pp 186-195, Boston, Nov. 1996*;
- GROOVER, Mikell P. **Robótica: Tecnologia e Programação**. McGraw-Hill, 1998;
- HOUNSELL, Marcelo da Silva e PIMENTEL, Alexandre. **On the use of virtual reality to teach robotics**. *3rd International Conference on Enigneering and Computer Education*. March, 16-19, 2003, São Paulo, Brazil;
- INTELITEK. <http://www.sti-sl.es/scorboter4u.htm>, acessado em 04/11/2004;
- KIRNER, Cláudio. <http://www.dc.urscar.br/~grv/tutrv/tutrv.htm> **Sistemas de Realidade Virtual**, acessado em 04/11/2004;
- REDEL, Rubens et al. **Criteria Determination and Prioritization for a “Good” Robot’s Off-line Programming Language**. *VI Induscon*. Outubro, 2004, pg 1444-1449, Joinville, Brazil;
- SCHILLING, Robert Jr. **Fundamentals of Robotics Analysis and Control**. Englewood Cliffs Prentice Hall. Cop, 1990;
- SPONG, M. W e VIDYASAGAR, M. **Robot Dynamics and Control**. John Wiley & Sons, New York, 1989;
- TSAI, Lung-Wen. **Robot analysis: the mechanic of serial and parallel manipulators**. John Wiley & Sons, 1999. ISBN 0-471-32593-7;
- VINCE, John. **Essential virtual reality fast: how to understand the techniques and potential of virtual reality**. Springer-Verlag. London, 1999. ISBN 1852330120;

ANEXOS

ANEXO 1 – Planilha Microsoft Excel do cálculo da cinemática SCORBOT-ER4PC

The screenshot shows a Microsoft Excel spreadsheet titled 'Microsoft Excel - INVERSA.XLS'. The spreadsheet is organized into several sections:

- Entrada Cinemática Direta (Yellow background):**

| base | AnteBraço | Braço | Elevação | Rotação |
|------------|-------------|------------|-------------|-----------|
| 12 | 78 | 98 | 100 | 60 |
| 0,20943951 | 1,361356817 | 1,71042267 | 1,745329252 | 1,0471976 |
- Saída Inversa (Green background):**

| | | | | |
|------------|-------------|---------|----------|-----------|
| 0,20943951 | 1,361356817 | 1,71042 | -4,53786 | 1,0471976 |
| 12 | 78 | 98 | -260 | 60 |
- Matriz Resultante Cinemática Direta (Orange background):**

| | | | |
|----------|-------|-------|----------|
| 0,23118 | 0,02 | 0,97 | -4,3120 |
| -0,83623 | -0,51 | 0,21 | -0,9165 |
| 0,49726 | -0,86 | -0,10 | 116,3950 |
- Entrada Cin. Inversa (Cyan background):**

| posição | orientação |
|-----------|------------|
| -4,31195 | 0,23118 |
| -0,91653 | -0,83623 |
| 116,39505 | 0,49726 |
- Trigonometric Values:**

| | |
|------------|-------------|
| cos2 | sen2 |
| 0,20791169 | 0,978147601 |
- Constants:**

| | |
|-------------|----------|
| k1 | k2 |
| -173,723519 | 230,5389 |

The spreadsheet also shows a status bar at the bottom with 'Plan1 / Plan2 / Plan3 /' and 'Pronto'.

ANEXO 2 – Código da função para o cálculo da Cinemática Inversa em VRML

```

function atualizaAngulo()
{
    Atan2(ponto[1],ponto[0]);
    angbase=atan2;

    angrotacao=Math.asin((orientaX*Math.sin(angbase)-
        (orientaY*Math.cos(angbase)));

    auxY=(orientaX*Math.cos(angbase))+(orientaY*Math.sin(angbase));
    auxX= -orientaZ;
    Atan2(auxX,auxY);
    ang234=atan2;
    k1=((ponto[0]*Math.cos(angbase))+(ponto[1]*Math.sin(angbase))-
        45+(125*Math.sin(ang234)));
    k2 = (-ponto[2]+360-(125*Math.cos(ang234)));

    angbraco=Math.acos((Math.pow(k1,2)+Math.pow(k2,2)-Math.pow(220,2)-
        Math.pow(220,2))/(2*220*220));
    c2=(((k1*(220+(220*Math.cos(angbraco))))+(k2*220*Math.sin(angbraco)))/(2*M
        ath.pow(220,2)+(2*220*220*Math.cos(angbraco))));
    s2=(((k1*220*Math.sin(angbraco))+k2*(220+(220*Math.cos(angbraco))))/
        (2*Math.pow(220,2)+(2*220*220*Math.cos(angbraco))));
    Atan2(s2,c2);

    angantebraco = atan2;

    angelevacao = ang234-angbraco-angantebraco;

    angbase = Math.round(180*angbase/Math.PI);
    angrotacao = Math.round(180*angrotacao/Math.PI);
    angantebraco=Math.round(180*angantebraco/Math.PI);
    angbraco=Math.round(180*angbraco/Math.PI);
    angelevacao=Math.round(180*angelevacao/Math.PI);

}

```

ANEXO 3 – Código do cálculo da função Atan2() em VRML

```
function Atan2(y,x)
{
  if (x>0)
  {
    atan2=Math.atan(y/x);
  }
  if (x==0)
  {
    if (y>0)
    {
      atan2=Math.PI/2;
    }
    if (y<0)
    {
      atan2=-Math.PI/2;
    }
  }
  if (x<0)
  {
    if (y>0)
    {
      atan2=Math.atan(y/x)+Math.PI;
    }
    if (y<0)
    {
      atan2=Math.atan(y/x)-Math.PI;
    }
  }
}
```