

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CURSO DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA

DESENVOLVIMENTO DE UM SISTEMA PARA ANÁLISE
E MODELAMENTO DINÂMICO DE ROBÔS INDUSTRIAIS.

DISSERTAÇÃO SUBMETIDA À UNIVERSIDADE FEDERAL DE SANTA CATARINA
PARA A OBTENÇÃO DO GRAU DE MESTRE EM ENGENHARIA MECÂNICA

JONNY CARLOS DA SILVA

FLORIANÓPOLIS, SETEMBRO DE 1990.

DESENVOLVIMENTO DE UM SISTEMA PARA ANÁLISE E
MODELAMENTO DINÂMICO DE ROBÔS INDUSTRIAIS.

JONNY CARLOS DA SILVA

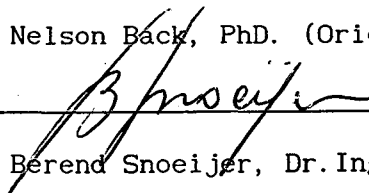
ESTA DISSERTAÇÃO FOI JULGADA ADEQUADA PARA OBTENÇÃO DO TÍTULO DE

MESTRE EM ENGENHARIA

ESPECIALIDADE ENGENHARIA MECÂNICA, ÁREA DE CONCENTRAÇÃO
PROJETO, APROVADA EM SUA FORMA FINAL PELO CURSO DE PÓS-GRADUAÇÃO EM ENGENHARIA
MECÂNICA.



Prof. Nelson Back, PhD. (Orientador)



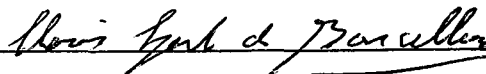
Prof. Berend Snoeijer, Dr. Ing.

Coordenador do Curso.

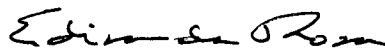
BANCA EXAMINADORA.



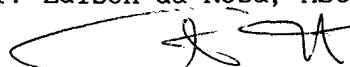
Prof. Nelson Back, PhD.



Prof. Clóvis Sperb de Barcellos, PhD.



Prof. Edison da Rosa, MSc.



Prof. Carlos Alberto Martin, Dr. Ing.

DEDICATÓRIA

Dedico este trabalho a Minha Família, em especial aos Meus Pais,
pelo apoio e compreensão durante sua execução.

AGRADECIMENTOS.

Ao Prof. Nelson Back, pela dedicação e estímulo demonstrados como Orientador deste trabalho, sobretudo pelo senso de pioneirismo necessário para o desenvolvimento de uma nova linha de pesquisa.

Aos Professores Clóvis Sperb de Barcellos, Edison da Rosa e Carlos Alberto Martin, pelas sugestões apresentadas como membros da banca examinadora.

Aos funcionários do Curso de Pós-Graduação em Engenharia Mecânica da UFSC, pois são também responsáveis pelo bom nível que o mesmo obteve ao longo de sua existência.

Aos colegas, Jorge Luís Erthal e Fernando Antonio Forcelini, como companheiros de turma, pelas discussões sobre os diversos temas relacionados à robótica.

Aos colegas, Maurício Küster e Jun Fonseca, como responsáveis pelo gerenciamento do Laboratório de CAE/CAD do Departamento de Engenharia Mecânica da UFSC.

Aos colegas, Jairo, Celso e Cristian, pela atenção dedicada à área de robótica como bolsistas, tendo favorecido a apresentação de seminários relativos a este tema.

À Comunidade Bahá'í, pelo espírito de Família e elevados princípios de vida transmitidos.

A todos que direta ou indiretamente auxiliaram a realização deste trabalho.

Muito Obrigado.

SUMÁRIO

Simbologia.....	vii
Resumo.....	viii
Abstract.....	ix
Capítulo 1. Introdução.....	1
Capítulo 2. Definição dos parâmetros geométricos e cinemáticos.	
2.1. Estrutura do robô.....	5
2.2. Convenção de Denavit-Hartenberg.....	9
2.3. Cinemática direta.....	11
2.4. Cinemática inversa.....	13
2.5. Cálculo do jacobiano.....	15
Capítulo 3. Formulação dinâmica de robôs.	
3.1. Contexto da dinâmica na robótica.....	19
3.2. Modelamento segundo Newton-Euler.....	21
3.3. Modelamento segundo Lagrange-Euler.....	25
3.4. Aplicações da análise dinâmica.....	32
3.4.1. Eficiente simulação dinâmica.....	32
3.4.2. Cálculo da capacidade de carga.....	34
Capítulo 4. Aplicação da simulação no projeto de sistemas.	
4.1. Introdução.....	38
4.2. Simulação na robótica.....	40
4.3. Arquitetura de um sistema.....	42
4.4. Aspectos da programação de robôs.....	42
Capítulo 5. Desenvolvimento do sistema.	
5.1. Considerações gerais.....	45
5.2. Modelamento geométrico.....	48
5.3. Modelamento cinemático.....	49
5.4. Modelamento dinâmico.....	53
5.5. Programação de trajetória.....	56 55
5.6. Simulação de operação.....	67

Capítulo 6. Análise de resultados.	
6.1. Apresentação dos objetivos.....	73
6.2. Simulação de um robô cilíndrico.....	73
6.3. Simulação de um robô articulado.....	83
6.4. Análise do efeito dos perfis de velocidade sobre o cálculo dos torques e da energia.....	92
6.5. Avaliação das cargas estáticas.....	95
Capítulo 7. Conclusão e recomendações.....	96
Referências bibliográficas.....	98
Anexo: Manual do usuário.	

SIMBOLOGIA

θ_i - ângulo que determina a orientação relativa entre os eixos das junta i e $i-1$. (graus)

d_i - comprimento da junta i , medido sobre o eixo da junta anterior. (m)

α_i - ângulo de torção da ligação i . (graus)

a_i - comprimento da ligação i . (m)

q_i - coordenada generalizada que identifica a posição da junta i , medida em radianos ou metros, dependendo do tipo de junta.

\dot{q}_i - velocidade da junta i . (rad/s ou m/s)

\ddot{q}_i - aceleração da junta i . (rad/s² ou m/s²)

τ_i - torque generalizado sobre a junta i . (Nm ou N)

\underline{x} - vetor posição do efetuador. (m)

\underline{b} - vetor direcionamento do efetuador. (adm)

\underline{n} - vetor orientação do efetuador. (adm)

Δt - incremento de tempo adotado na simulação. (segundos)

T - tempo da simulação.

$[H]$ - tensor de inércia do robô.

H_{ij} - elemento do tensor de inércia. (kg.m² ou kg)

$r_{0,ci}$ - posição do centróide da ligação i , em coordenadas inerciais. (m)

v_{ci} - velocidade linear do centróide da ligação i . (m/s)

\dot{v}_{ci} - aceleração linear do centróide da ligação i . (m/s²)

ω_i - velocidade angular da ligação i . (rad/s)

$\dot{\omega}_i$ - aceleração angular da ligação i . (rad/s²)

RESUMO

O trabalho apresenta o desenvolvimento de um sistema computacional, que permite analisar as configurações de robôs industriais mais utilizadas, considerando entre outros aspectos, os acoplamentos dos movimentos entre as ligações, avaliados através do modelo dinâmico, na formulação lagrangeana. O sistema possibilita a programação de trajetória, para obtenção das curvas características de posição, velocidade, aceleração, e torque de cada junta durante o movimento do robô. Também são avaliadas a influência das cargas externas aplicadas à extremidade livre do robô, o teste da capacidade de carga dos atuadores e o comportamento da energia na simulação.

A interface gráfica permite: a observação em perspectiva isométrica da cadeia de ligações do robô durante toda a trajetória; a simulação de uma manipulação na qual o robô interage com o ambiente de objetos, sendo as instruções fornecidas de forma textual de acordo com cada tarefa; e a análise das curvas mencionadas para cada operação executada. Estas funções do sistema visam fornecer uma ferramenta de CAE/CAD para o estudo da robótica.

ABSTRACT

The work presents the development of a computer system which permits to analyse the kinds of manipulators arms more used, considering the coupling of movements among the links, evaluated through the dynamic modeling based upon the lagrangian formulation. The system allows the trajectory programming, to obtain the graphics of position, velocity, acceleration and torque for each joint during a robot movement. Other aspects are presented, such as, the influence of extern loads applied to the end effector, the test of carrying capacity of the actuators and the energy consumed.

The graphic interface permits to observe the motion of the linkage of the robot during the trajectory; the simulation of a pick-and-place operation, in which the robot interacts with others elements in its workspace, with instruction being given on task level programming form; and the analysis of the graphics for each operation. These functions of the system aim to give a tool of CAE/CAD to the study of robotics.

1. INTRODUÇÃO.

A crescente busca da indústria visando maior produtividade, menores custos de produção, melhor qualidade e substituição do operador em tarefas monótonas e perigosas, tem favorecido a aplicação dos robôs nas operações de manufatura em geral, especialmente nas indústrias automotiva e de componentes elétricos [1].

Embora não seja focado no presente trabalho, o estudo da robótica, como tecnologia emergente, é de suma importância, sobretudo, em uma sociedade que se encontra numa fase de transição tecnológica, na qual a robótica é avaliada mais como fonte de problemas sociais do que pelos seus benefícios.

A maior aplicação dos robôs tem favorecido o aumento das atividades de pesquisa e desenvolvimento sobre os diversos campos relacionados à robótica, objetivando uma melhor performance destes equipamentos.

Como área de estudo, considera-se a robótica como sendo formada a partir de duas áreas existentes, as tecnologias de comando numérico e manipulação remota. Contudo, a robótica envolve questões de projeto e controle substancialmente distintas destas, devido sobretudo às características de mobilidade, destreza, acoplamentos e intensa interação entre o robô e os equipamentos envolvidos [2].

Historicamente, é considerada como primeira patente de um robô o dispositivo denominado PAT (Programmed Article Transfer), projetado por George Devol, fabricado pela UNIMATION e instalado na FORD em 1961 na operação de descarregamento de uma fundição [3]. Desde então, diversos estudos tem sido desenvolvidos sobre a tecnologia, programação e aplicação de robôs, tais como, robôs com bases móveis, linguagens de programação em alto nível, sensoriamento. Atualmente, robôs estão sendo utilizados nas mais variadas tarefas, a saber: operações de soldagem, pintura, acabamento, manipulação de materiais e carga de máquinas.

Em virtude do amplo espectro de conhecimentos envolvidos com a robótica, esta é apresentada como uma interface de outras áreas como as engenharias mecânica, elétrica e industrial e a computação, tornando-a uma área interdisciplinar e de amplas fronteiras para pesquisa e desenvolvimento.

Existem algumas definições de robôs industriais. Segundo Mikell Groover, um robô industrial é uma máquina multi-tarefa, programável, possuindo certas características antropomórficas [5]. Esta definição mostra-se restrita, pois como será visto no decorrer deste trabalho, nem todos os robôs parecem braços mecânicos. Conforme a JIRA (Japan Industrial Robot Association), seis classes de robôs são apresentadas, a saber, manipuladores manuais, manipuladores com sequência fixa, manipuladores com sequência variável, robôs repetidores, robôs de controle numérico e robôs inteligentes [4].

Segundo a RIA (Robot Institute of America) um robô industrial é um manipulador reprogramável e multifuncional projetado para mover materiais, partes, ferramentas ou dispositivos especiais através de movimentos predefinidos para desenvolver uma variedade de tarefas [3,4]. Esta definição compreende apenas as três últimas classes propostas pela JIRA e será escolhida para o presente trabalho.

Como uma das principais aplicações da robótica no contexto atual, observa-se a integração da manufatura ou CIM (Computer Integrated Manufacturing). Nesta atividade o robô realiza algumas funções, tais como, assistir outros equipamentos, executar tarefas de média habilidade como montagem ou alta habilidade como soldagem por costura.

Devido ao número de diferentes tipos de robôs existentes no mercado, a seleção da melhor configuração para uma dada tarefa torna-se consideravelmente complexa. Do ponto de vista de desempenho, questões como, capacidade de carga, consumo de energia, repetibilidade, precisão e espaço de trabalho, devem ser avaliadas, cujo processo é praticamente inviável caso não seja utilizado o computador como ferramenta na análise, devido às características específicas de cada configuração e a tecnologia envolvida no

projeto destes equipamentos.

Entre os aspectos a serem avaliados, o desempenho dinâmico dos robôs mostra-se de grande importância, sobretudo referindo-se a operações em altas velocidades e/ou com altas cargas, pois nestas condições as propriedades de massa e inércia são relevantes [2]. O modelo dinâmico do robô é representado por um sistema de equações acoplado e não linear que descreve a relação entre os torques aplicados pelos atuadores e as grandezas de posição, velocidade e aceleração.

O presente trabalho apresenta o desenvolvimento de um sistema de simulação, que possibilita a análise do modelo dinâmico das principais configurações utilizadas, considerando os acoplamentos dos movimentos entre as ligações, obtenção das curvas que descrevem o comportamento das grandezas mencionadas ao longo do tempo, bem como as curvas para seleção dos atuadores e a energia consumida durante a trajetória. O sistema possui uma interface gráfica que permite a observação tridimensional do modelo do robô durante toda a trajetória.

Tendo em vista a complexidade do modelo, são apresentadas no segundo capítulo: a terminologia, a convenção e os princípios envolvidos no projeto mecânico do robô, considerando a geometria e cinemática da estrutura.

As propriedades inerciais e as formulações da dinâmica do robô, métodos de Newton-Euler e de Lagrange, são estudadas no terceiro capítulo e apresentadas algumas aplicações da análise dinâmica no contexto da robótica.

O capítulo seguinte trata da simulação como ferramenta no projeto de sistemas, bem como os requisitos necessários à sua aplicação em relação à robótica, apresenta também alguns aspectos referentes à programação de robôs. O quinto capítulo apresenta o desenvolvimento do sistema, descrevendo as etapas relativas ao modelamento e aos modos de programação permitidos, como também os módulos de análise e simulação.

O sexto capítulo analisa os resultados obtidos, comparando as diversas configurações simuladas para uma determinada tarefa e propondo suas

aplicações. O sétimo capítulo apresenta uma conclusão, recomendações e tendências da robótica para futuros trabalhos. Acompanha este texto um manual do usuário.

2.1 ESTRUTURA DO ROBÔ.

Um robô tem como uma de suas funções posicionar e orientar um objeto localizado em seu dispositivo terminal, podendo ser uma garra, pistola de pintura ou outro tipo de ferramenta.

A estrutura geral do manipulador é composta pelos seguintes subsistemas:

- . Estrutura mecânica que compreende uma cadeia aberta formada por sucessivos corpos (ligações) conectados por juntas, que podem ser prismáticas, permitindo uma translação relativa entre as ligações, ou rotacionais que permitem uma rotação relativa entre as ligações.

- . Os atuadores aplicam torques ou forças sobre as juntas, promovendo o movimento das ligações.

- . Os sensores são responsáveis pelas informações enviadas ao controlador do robô, tanto sobre o estado do robô (informações de posição e/ou velocidade) como as relacionadas com ambiente em volta do robô.

- . O sistema de controle executa o comando dos atuadores na realização das tarefas predeterminadas pelo sistema de decisão e pelas informações dos sensores.

- . O sistema de decisão interpreta os comandos do operador, na solução das tarefas, geração de planos e mensagens para o operador e a preparação dos dados para o sistema de controle.

- . O sistema de comunicação processa as mensagens do sistema de decisão para o operador via comandos alfanuméricos, displays gráficos, teachingbox, etc [6].

No presente trabalho são abordadas com maior ênfase, a estrutura mecânica, os atuadores e o sistema de comunicação. Os dois primeiros são representados e modelados por equações que descrevem seus comportamentos e o sistema de comunicação é avaliado pela interface gráfica do simulador.

Nesta abordagem tanto as ligações como as juntas são

consideradas rígidas, embora estudos estejam sendo realizados considerando a flexibilidade de tais elementos [6]. A forma como a cadeia é elaborada, ou seja, a combinação em sequência entre juntas prismáticas e rotacionais, definirá o tipo de robô.

Devido a estrutura cinemática em série apresentada pelo robô, a posição do órgão terminal ou efetuator, é uma função da posição de todas as ligações. Este efeito também ocorre entre as ligações, ou seja, o deslocamento de uma junta afeta a posição de todas as ligações que a seguem, e dependendo da configuração provocará esforços também nas juntas anteriores a esta, este efeito será analisado no decorrer deste trabalho.

Na classificação de um robô, um parâmetro a ser considerado é o número de graus de liberdade, correspondente ao número de parâmetros independentes que definem completamente a posição da cadeia [3]. Considerando que cada junta tenha apenas um grau de liberdade, a mobilidade do robô, ou seja, os graus de liberdade da cadeia são medidos pelo número de juntas.

Os modelos de robôs considerados, no presente trabalho, apresentam no máximo seis juntas, as quais compreendem a maioria das aplicações industriais. No espaço tridimensional três graus de liberdade (GDL) são necessários para posicionar um corpo, e as restantes três juntas possibilitam a orientação desejada.

Estas configurações ($GDL \leq 6$) são conhecidas como não redundantes, pois a maioria das posições e orientações são conseguidas, por uma única combinação dos parâmetros de juntas, contudo em algumas posições existe redundância.

Embora as configurações com maior número de graus de liberdade compliquem significativamente a programação, o controle e o projeto, estas são desejáveis para contornar possíveis obstáculos existentes no espaço de trabalho do robô [7].

Outro fator importante na especificação de um robô é sua capacidade de carga (payload). Este parâmetro depende do projeto do robô, da

forma e condições da superfície do objeto a ser segurado, da posição do robô (ou seja, da inércia da carga e do robô) e da velocidade de transporte. Alguns fabricantes fornecem duas capacidades de carga, normal e máxima, ou estática e dinâmica. Deve-se verificar sobre que condições estes valores são determinados [5] e qual efetuador que será utilizado, pois a capacidade de carga depende do efetuador e da carga.

Basicamente, existem quatro configurações de robôs no mercado, a saber: cartesiano, cilíndrico, esférico e articulado, que são obtidas variando-se o tipo e a sequência das juntas. Na análise da configuração considera-se apenas as três primeiras juntas que determinam a forma do espaço de trabalho.

A configuração cartesiana apresenta as três primeiras juntas prismáticas, com o movimento destas juntas descreve-se um espaço de forma paralelepípedica. Outros nomes aplicados a esta configuração são robô xyz ou linear ou pórtico. Como características, esta configuração possui elevadas rigidez e capacidade de carga, satisfatórias precisão e repetibilidade.

A configuração cilíndrica tem como espaço de trabalho um cilindro, obtido devido a disposição das juntas na sequência de uma rotacional e duas prismáticas. Este modelo é utilizado em operações de manipulação e montagem, apresenta boa rigidez.

A configuração esférica (ou polar) possui duas juntas rotacionais e uma prismática na direção radial, formando um espaço de trabalho limitado por uma calota esférica, apresenta menor rigidez que as duas anteriores, porém maior versatilidade.

Dos modelos mencionados, o articulado é o que mais se aproxima de um braço humano, por não possuir nenhuma junta prismática, que caracteriza o efeito telescópico. Este tipo é de grande aplicação nas atuais tarefas, apesar da baixa rigidez e pouca precisão comparado com os outros tipos. Os modelos mencionados são apresentados na figura 2.1.

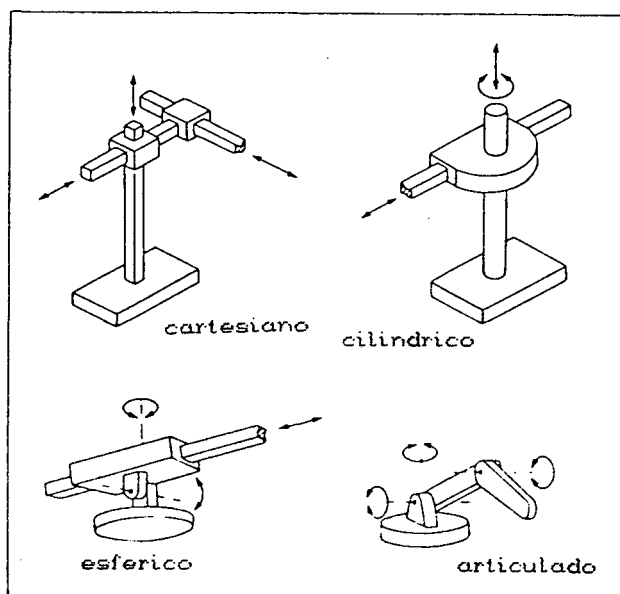


Figura.2.1 Configurações representadas.

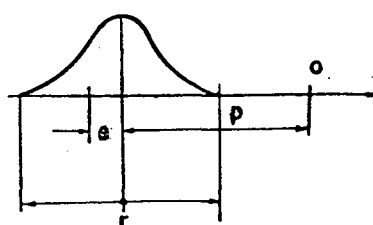
Além destas, existe uma outra configuração, obtida como uma variação da cilíndrica, pois apresenta as duas primeiras juntas rotacionais em eixos verticais e uma terceira junta prismática na mesma direção, conferindo a esta configuração uma satisfatória rigidez nesta direção, é denominada SCARA (Selective Compliance Assembly Robot Arm). Como a sigla sugere, a sua maior aplicação é na operação de montagem.

O espaço de trabalho do robô compreende todos os pontos alcançados pelo seu pulso, a última ligação da cadeia antes do efetuator. A forma deste espaço depende do tipo de configuração. Nos catálogos, o espaço de trabalho é representado por duas vistas, uma superior e outra vista lateral.

Com relação ao tipo de juntas do pulso, tem-se : junta tipo "bend" significa que a ligação seguinte a junta gira e se translada em relação ao eixo da junta, pois a extensão da ligação não é colinear ao eixo da junta, ao contrário do que ocorre com a junta tipo "roll", na qual a ligação

posterior apenas gira em torno do eixo da junta devido a colinearidade.

Precisão e repetibilidade são também aspectos relevantes na robótica, apesar de existir relação entre eles, são fatores distintos. A precisão é importante tanto nas operações ponto a ponto como na geração de trajetória, é definida como a diferença entre a posição desejada a ser atingida pelo efetuador e o centro da curva de distribuição das coordenadas reais do efetuador. Repetibilidade é a largura desta curva (fig 2.2), ou seja, é uma medida da capacidade do manipulador de reposicionar, seguidas vezes, o efetuador no mesmo ponto [7]. Tanto a precisão como a repetibilidade são influenciadas por fricção, histerese, folgas, flexibilidade das ligações, juntas e atuadores.



o- posição desejada.

p- precisão obtida.

r- repetibilidade.

e- erro de repetibilidade.

Fig.2.2.Representação de precisão e repetibilidade.

A alteração da posição objetivo no espaço de trabalho pode ainda causar a variação destes parâmetros, que também são afetados pelo tempo de estabilização permitido para cada operação de posicionamento, que é o tempo necessário para o amortecimento das vibrações originadas do movimento do robô.

2.2 CONVENÇÃO DE DENAVIT-HARTEMBERG.

Como mencionado, a estrutura mecânica do robô é formada por uma cadeia de ligações conectadas por juntas. Visando representar este encadeamento entre corpos, é adotado um conjunto de sistemas de coordenadas (cartesiano e tridimensional) localizados, um por ligação, de tal forma que a posição relativa entre as ligações é determinada, relacionando à posição e

orientação entre cada sistema, conforme a convenção de Denavit-Hartenberg [2].

Nesta convenção, o par composto por uma ligação e a respectiva junta é, cinematicamente, definido por quatro parâmetros, sendo dois deles parâmetros de junta e os outros relativos à ligação (Fig 2.3).

De acordo com esta convenção, o eixo z_i é orientado na direção da junta $i+1$, enquanto o eixo x_i está na direção da normal comum aos eixos z_{i-1} e z_i , prolongamento do comprimento da ligação a_i .

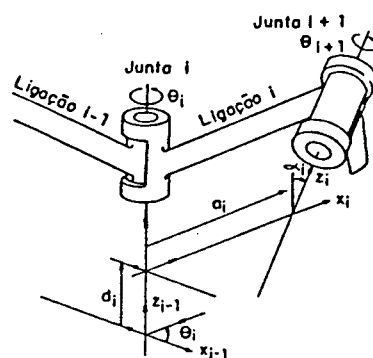


Figura 2.3. Definição dos parâmetros de Denavit-Hartenberg.

Na determinação dos eixos locais, dois sistemas merecem observações. O primeiro sistema considera-se fixo à base, com o eixo z_0 paralelo ao eixo da junta 1 e os eixos x_0 e y_0 são escolhidos pela regra da mão direita. No caso do último sistema o eixo x_n deve interceptar o eixo da última junta com um ângulo reto.

Dois tipos de juntas são utilizados na cadeia do robô, para junta i rotacional, o parâmetro θ_i é a variável da junta e d_i é constante. Caso a junta i seja prismática d_i é a variável e θ_i torna-se constante.

Como mostrado na figura 2.3, os parâmetros a_i e α_i são medidos sobre o sistema i , enquanto θ_i e d_i são relativos ao sistema $i-1$, caracterizando assim a relação entre os dois sistemas, pela matriz de transformação homogênea [2].

$$A_{i-1,i} = \begin{bmatrix} \cos\theta_i - \cos\alpha_i \cdot \text{sen}\theta_i & \text{sen}\alpha_i \cdot \text{sen}\theta_i & a_i \cdot \cos\theta_i \\ \text{sen}\theta_i & \cos\alpha_i \cdot \cos\theta_i - \text{sen}\alpha_i \cdot \cos\theta_i & a_i \cdot \text{sen}\theta_i \\ 0 & \text{sen}\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

Esta matriz homogênea é obtida com uma rotação θ_i em torno de z_{i-1} , uma translação de d_i ao longo do eixo z_{i-1} , uma translação de a_i ao longo de x_i e uma rotação de α_i em torno de x_i . Aplicando estas quatro transformações sucessivamente, obtém-se a matriz de transformação homogênea $A_{i-1,i}$.

O efeito de encadeamento entre as ligações é representado então pela pós-multiplicação das sucessivas matrizes homogêneas, esta concatenação representa o acoplamento entre o movimento de cada junta e a posição (e orientação) das ligações posteriores a esta junta.

$$A_{0,j} = A_{0,1} \cdot A_{1,2} \cdot A_{2,3} \dots A_{j-1,j} \quad (2.2)$$

Com o propósito de generalizar a notação, as matrizes de transformação homogêneas são determinadas como funções das coordenadas generalizadas do sistema q_s que representam as variáveis (θ ou d) de cada junta, o vetor formado por todas as coordenadas, caracteriza a posição da cadeia do robô. Esta equação é denominada a equação cinemática do manipulador e governa o comportamento cinemático da cadeia.

2.3. CINEMÁTICA DIRETA.

Como mencionado, a posição e orientação do efetuador é uma função da posição de cada junta, ou seja, do vetor de coordenadas generalizadas do robô.

Esta relação não é biunívoca, contudo é possível a determinação

da posição e orientação do efetuador partindo das coordenadas generalizadas, aplicando-se a cinemática direta.

Analisando a expressão 2.2, observa-se que a matriz $A_{0,j} = A(q_1, q_2, \dots, q_j)$, para um robô de seis graus de liberdade é da forma $A_{0,6} = A(q_1, q_2, q_3, q_4, q_5, q_6)$ e estabelece a relação entre o último sistema de coordenadas (sistema 6) e o sistema de base (sistema 0) através do vetor de coordenadas do robô.

Nesta matriz a quarta coluna representa a posição do efetuador e as três primeiras colunas representam os cossenos diretores dos vetores unitários do sistema de coordenadas do efetuador (determinando a orientação) expressos em relação ao sistema de base, esta submatriz 3x3 é ortonormal, ou seja, as colunas são ortogonais entre si e as normas euclidianas destes vetores são iguais a um.

A representação de um vetor em coordenadas homogêneas caracteriza-se por apresentar uma dimensão a mais além da dimensão do vetor, ou seja, no caso apresentado o vetor é tridimensional, porém em coordenadas homogêneas é representado por 4 componentes, sendo a última componente (elemento 4,4) igual a unidade, representando o fator de escala. Assim um vetor $3i+4j+5k$ (onde i, j, k são os vetores unitários nas direções x, y, z respectivamente) pode ser representado por $[3, 4, 5, 1]^t$ ou $[6, 8, 10, 2]^t$ ou $[-30, -40, -50, -10]^t$, onde o índice t denota a transposta do vetor linha, ou o vetor coluna correspondente [8].

Na equação 2.2 as pós-multiplicações identificam que as transformações são realizadas, sucessivamente, sobre os sistemas locais e não com relação à base. No presente trabalho são consideradas apenas transformações de translação e rotação, não envolvendo extensão ou escalamento de vetores.

As coordenadas generalizadas apresentadas como argumentos das matrizes de transformação homogêneas são valores finitos dos deslocamentos angulares ou lineares de acordo com o tipo de junta, porém outras relações

cinemáticas serão descritas com relação aos valores infinitesimais destes deslocamentos, neste caso as relações são de velocidades como será apresentado.

Um outro problema de fundamental importância é a determinação das posições das juntas (coordenadas generalizadas do robô) a partir da posição e orientação do efetuador, denominada cinemática inversa.

2.4 CINEMÁTICA INVERSA.

Operacionalmente, existem dois espaços de coordenadas distintos que são relacionados ao robô, um é o espaço cartesiano, também conhecido como espaço operacional, no qual a posição é determinada por três parâmetros e a orientação por ângulos ou seus cossenos diretores. A programação neste espaço é a de maior aplicação, por ser determinada em coordenadas cartesianas, independente da configuração do robô. O outro espaço é referente às juntas do manipulador, ou seja, ao vetor de coordenadas do robô. A programação neste caso é mais complexa, pois necessita de informações específicas para cada configuração.

A obtenção da cinemática inversa tem como pré-requisito a cinemática direta (Eq. 2.2) e a solução analítica para este problema depende diretamente da configuração do robô. Uma condição suficiente para a existência da solução analítica, determina que três eixos consecutivos devem se interceptar em um único ponto (pulso esférico), neste caso a posição deste ponto é função apenas das 3 primeiras coordenadas, sendo assim o problema de determinar seis incógnitas é dividido em dois problemas de três incógnitas cada um [2].

A análise da cinemática inversa é desenvolvida em várias referências [9,10]; Existem basicamente duas linhas de estudos, os métodos numéricos e os analíticos.

Computacionalmente, os métodos analíticos são mais rápidos,

contudo não apresentam a generalidade dos métodos numéricos que compreendem algoritmos como Newton-Raphson [9] ou análise da posição zero [11]. A característica iterativa destas técnicas, torná-as inviáveis no processo de programação empregado neste trabalho, no qual as posições e orientações do efetuador são variáveis com o tempo ao longo da trajetória.

A técnica de Paul [8], utiliza a expressão 2.2, e pela aplicação de pré e pós-multiplicações, obtém expressões de igualdade matricial, das quais as coordenadas generalizadas podem ser determinadas.

O sistema de coordenadas do efetuador com relação à base é dado por, sendo esta matriz $T_{0,6}$ descrita na forma :

$$T_{0,6} = A_{0,1}.A_{1,2}.A_{2,3}.A_{3,4}.A_{4,5}.A_{5,6} \quad (2.3)$$

$$T_{0,6} = \begin{bmatrix} n_x & o_x & b_x & p_x \\ n_y & o_y & b_y & p_y \\ n_z & o_z & b_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.4)$$

Na equação (2.4), o vetor \underline{b} pertence a direção z do sistema de coordenadas do efetuador que determina a direção de aproximação, o vetor \underline{n} é normal ao plano do efetuador e o vetor \underline{o} é obtido pela forma :

$$\underline{o} = \underline{b} \times \underline{n} \quad (\text{produto vetorial}) \quad (2.5)$$

A orientação é especificada por uma sequência de rotações sobre os eixos x,y,z. Os ângulos de Euler descrevem uma possível orientação em termos de uma rotação ψ sobre o eixo z, uma rotação θ sobre o eixo y' modificado, e uma rotação de ϕ sobre o eixo x'' (modificado após a segunda rotação) (Fig. 2.4).

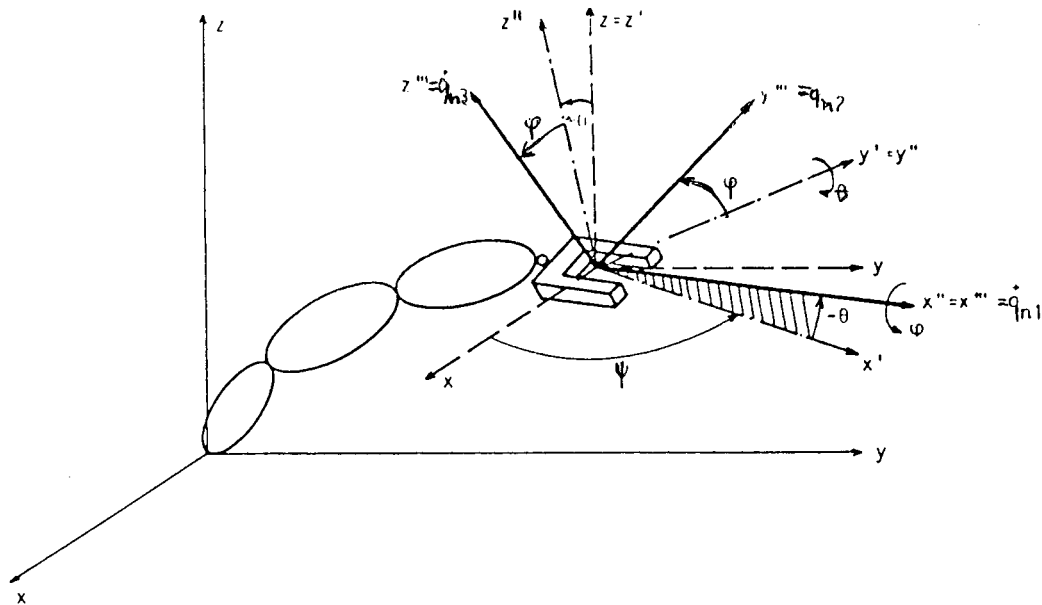


Figura 2.4. Representação dos ângulos de Euler.

Outra maneira de especificar a orientação é a denominada tombamento, rolamento e guinamento (pitch, roll, yaw), formada por sucessivas rotações sobre os eixos x, y, z do sistema de coordenadas de base.

Alguns métodos numéricos são baseados na relação diferencial entre as variáveis de posição e orientação do efetuador, esta transformação é definida matriz jacobiana do manipulador, e também é aplicada na obtenção do tensor de inércia do manipulador.

2.5. CÁLCULO DA MATRIZ JACOBIANA.

Representando o vetor de coordenadas generalizadas do robô por \underline{q} e o vetor de posição e orientação do efetuador em coordenadas de base por \underline{x} , pode-se dizer que:

$$\underline{x} = f(\underline{q}) \quad (2.6)$$

Diferenciando esta expressão, tem-se :

$$\delta \underline{x} = (\delta f / \delta \underline{q}) \delta \underline{q} \quad (2.7)$$

Ou na forma matricial:

$$\delta \underline{X} = [J] \cdot \delta \underline{Q} \quad (2.8)$$

A expressão 2.8 determina que a matriz Jacobiana terá o número de colunas igual a ordem do vetor \underline{q} e o número de linhas igual a ordem do vetor \underline{x} . Para um manipulador de 6 graus de liberdade, a matriz jacobiana é da ordem 6x6.

Dividindo-se a expressão 2.8 pelo incremento infinitesimal de tempo δt , tem-se :

$$\delta \underline{X} / \delta t = [J] \cdot \delta \underline{Q} / \delta t \text{ ou } V = [J] \dot{\underline{q}} \quad (2.9)$$

onde $\dot{\underline{q}}$ representa o vetor velocidade das juntas.

As três primeiras linhas da matriz Jacobiano são referentes às velocidades lineares do efetuador, enquanto as outras três referem-se às velocidades angulares. Cada coluna, representa a velocidade linear e angular gerada por cada junta correspondente. Particionando a matriz jacobiana de acordo com esta definição, tem-se :

$$[J] = \begin{bmatrix} J_{11} & J_{12} & J_{13} & J_{1n} \\ J_{a1} & J_{a2} & J_{a3} & J_{an} \end{bmatrix} \quad (2.10)$$

onde J_{1i} e J_{ai} são vetores (3x1) relacionados com as velocidades produzidas pela junta i sobre o efetuador.

Utilizando J_{1i} , pode-se determinar a velocidade linear do efetuador por:

$$V_e = \sum J_{1i} \dot{q}_i \quad (2.11)$$

Se a junta i é prismática, esta produz uma velocidade no

efetuador na mesma direção do eixo da junta. Caso seja rotacional a velocidade é obtida pelo produto vetorial da velocidade angular (ω_i) da junta e o vetor posição relativa ($r_{i-1,e}$) esta definição é mostrada na figura 2.5.

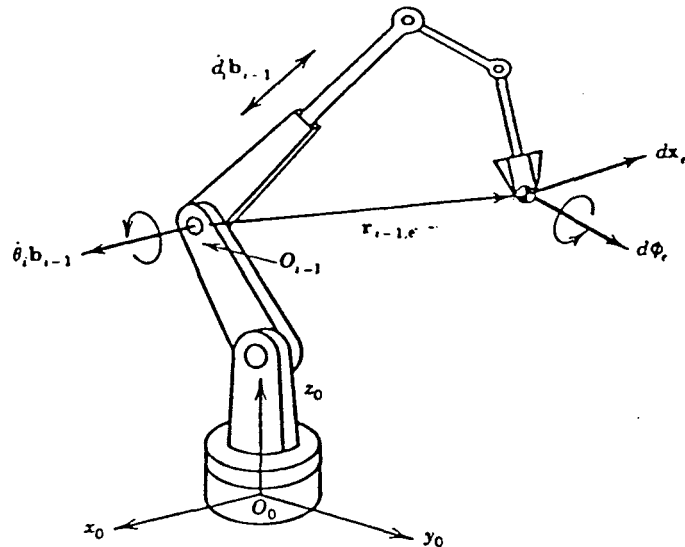


Figura 2.5 Definição física da matriz jacobiana.

Aplicando a mesma análise para a velocidade angular do efetuador tem-se que:

$$\omega_e = \sum J_{ai} \dot{q}_i \quad (2.12)$$

Quando a junta i é prismática, esta não produz velocidade angular sobre o efetuador, logo $J_{ai} = 0$. Para o caso de junta rotacional a velocidade angular produzida é exatamente a velocidade da junta i [2].

Equacionando as relações acima de acordo com a fig 2.5, tem-se:

$$\begin{bmatrix} J_{11} \\ J_{ai} \end{bmatrix} = \begin{bmatrix} b_{i-1} \\ 0 \end{bmatrix} \text{ para junta } i \text{ prismática.} \quad (2.13)$$

$$\begin{bmatrix} J_{11} \\ J_{ai} \end{bmatrix} = \begin{bmatrix} b_{i-1} \times r_{i-1,e} \\ b_{i-1} \end{bmatrix} \text{ para junta } i \text{ rotacional.} \quad (2.14)$$

A aplicação relacionada à cinemática inversa é desenvolvida a partir da equação (2.9), invertendo-se a matriz jacobiana, tem-se:

$$\dot{q} = [J]^{-1} \cdot V \quad (2.15)$$

Esta relação, porém é restrita às configurações não-singulares, pois havendo a singularidade, ocorre a degeneração da matriz jacobiana, ou seja, seus vetores colunas são linearmente dependentes entre si, implicando em uma perda de mobilidade do efetuador em alguma direção. Este efeito é percebido quando o movimento de uma junta não produz efeito sobre o efetuador, ou quando mais de uma junta pode exercer o mesmo efeito sobre o efetuador. Neste caso, outras técnicas devem ser aplicadas para obtenção da cinemática inversa [6].

Embora neste trabalho seja desenvolvido o cálculo da matriz jacobiana como apresentado, isto é devido à sua aplicação no modelo dinâmico para a determinação das velocidades das ligações. A cinemática inversa é obtida pelo método analítico, o qual utiliza relações geométricas entre a posição e orientação do efetuador e as coordenadas generalizadas do robô [9]. Este método é mais eficiente computacionalmente, mas devido ao fato da não generalidade, pois as relações geométricas são específicas para cada configuração, foram implementados algoritmos de cinemática inversa para cada tipo de modelo.

3.1-CONTEXTO DA DINÂMICA NA ROBÔTICA.

Neste capítulo são estudadas as formulações da dinâmica de manipuladores, considerando suas aplicações no projeto de robôs, alguns algoritmos implementados sobre a análise dinâmica, bem como as vantagens e desvantagens de cada formulação.

Na robótica, o modelo dinâmico relaciona as posições, velocidades e acelerações das juntas do robô, com os torques que devem ser aplicados pelos atuadores de acordo com a trajetória programada, considerando as características de massas e inércias das ligações, bem como as da carga manipulada. Neste sentido, o modelo dinâmico envolve um sistema de equações diferenciais.

Basicamente, esta relação ocorre de duas maneiras, a saber: estabelecer as posições, velocidades e acelerações das juntas partindo-se dos torques aplicados pelos atuadores, denominada dinâmica direta; ou, caso contrário, uma vez dada a trajetória (posições, velocidades e acelerações) calcular os torques necessários de acordo com estas variáveis, conhecida como a dinâmica inversa. Estes métodos são representados na fig. 3.1.

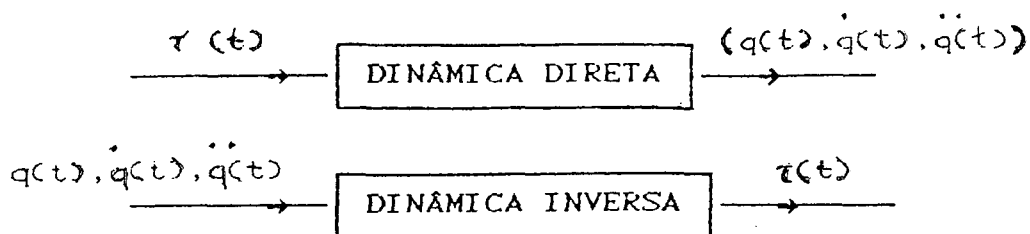


Figura 3.1.Representação da dinâmica direta e inversa.

No presente trabalho, é enfatizado o segundo caso, pois as formulações apresentadas tratam da obtenção do modelo dinâmico, no qual se determinam os torques pelo processo mencionado de dinâmica inversa.

Atráves da dinâmica direta, o modelo dinâmico determina as variáveis cinemáticas, pelo cálculo das acelerações, velocidades e posições que podem ser atingidas pelo robô, obtendo deste modo, sua real trajetória, sendo assim, este processo é aplicado no modelo de controle do robô [12].

Dependendo da complexidade desejada, o modelo permite considerar as características dos atuadores (constantes elétricas), efeitos dissipativos de potência e estratégias de controle [6].

Devido às rápidas mudanças da configuração do robô durante seu movimento e a dependência entre o modelo dinâmico e este fator, os termos do modelo devem ser calculados em altas frequências, alguns autores estabelecem valores superiores a 60 Hz [10,19]. Por este aspecto, nas aplicações da formulação dinâmica em tempo real, a preocupação predominante é a eficiência computacional [12].

Sendo assim, estudos têm sido realizados no sentido de simplificar os algoritmos que representam as formulações do modelo dinâmico, analisando a influência das variáveis envolvidas sobre o número de operações aritméticas [13], tipos de estruturas de robôs que simplificam o modelo dinâmico [14], e redução dos cálculos do modelo, eliminando operações repetitivas pela aplicação de procedimentos algébricos [1].

No presente contexto, o objetivo da formulação dinâmica é tornar compreensível a importância dos parâmetros envolvidos no projeto mecânico de robôs, tais como: dimensões, massas e inércias das ligações e suas interações durante o movimento.

Neste sentido, o modelo deve ser preciso o suficiente para representar satisfatoriamente o dispositivo a ser projetado, e ainda ser simples o bastante para uso prático tanto no projeto como no controle de manipuladores [15].

As formulações mais utilizadas na obtenção do modelo dinâmico de robôs são a formulação de Newton-Euler e a formulação de Lagrange. A seguir apresentam-se uma descrição dos princípios envolvidos nestes métodos e alguns

algoritmos utilizados para melhorar a eficiência computacional das suas implementações.

3.2-MODELAMENTO SEGUNDO NEWTON-EULER.

Como observado no capítulo anterior, formulação cinemática, a estrutura mecânica de um robô é formada por sucessivas ligações conectadas por juntas, que são os elementos mecânicos que transmitem os esforços (forças ou torques), recebidos do sistema atuador-redutor, para as ligações, realizando desta maneira o movimento relativo entre as mesmas.

Devido às características mecânicas de robôs, os esforços (daqui em diante torques generalizados) são calculados de acordo com um modelo matemático que possibilita ao robô realizar certas tarefas. A maneira como este modelo é obtido, determina a diferença entre as 2 formulações citadas.

A formulação segundo NEWTON-EULER está baseada nos princípios da mecânica newtoniana para a translação de corpos rígidos e nas equações de Euler relativas ao movimento rotacional de um corpo em relação ao seu centro de gravidade [16].

O procedimento na formulação de Newton-Euler para obtenção das equações do movimento compreende, primeiramente, estabelecer as relações que definem as velocidades e acelerações lineares e angulares relativas aos centróides de cada ligação (eqs. 3.1, 3.2 e 3.3), determinadas em coordenadas inerciais, e resolvê-las, partindo-se da base para a extremidade do manipulador (resolução da cinemática).

$$\omega_i = \begin{cases} \omega_{i-1} + \dot{q}_i \cdot z_{i-1} & \text{para junta } i \text{ rotacional.} \\ \omega_{i-1} & \text{para junta } i \text{ prismática.} \end{cases} \quad (3.1)$$

$$\dot{\omega}_i = \begin{cases} \dot{\omega}_{i-1} + \ddot{q}_i \cdot z_{i-1} + \omega_i \times (\dot{q}_i \cdot z_{i-1}) & \text{junta } i \text{ rotacional.} \\ \dot{\omega}_{i-1} & \text{junta } i \text{ prismática.} \end{cases} \quad (3.2)$$

$$\dot{v}_{ci} = \begin{cases} \omega_i \times r_{i-1,ci} + \omega_i \times (\omega_i \times r_{i-1}) + v_{i-1} & \text{junta i rotacional.} \\ q_i \cdot z_{i-1} + \omega_i \times r_{i-1,ci} + 2 \cdot \omega_i \times (q_i \cdot z_{i-1}) + & \\ \omega_i \times (\omega_i \times r_{i-1,ci}) + v_{i-1} & \text{junta i prismática.} \end{cases} \quad (3.3)$$

onde: q_i - velocidade da junta i (m/s ou rad/s);

z_{i-1} - vetor que identifica a direção da junta i;

$\dot{\omega}_i$ - aceleração angular da ligação i ;

ω_i - velocidade angular da ligação i ;

\dot{v}_{ci} - aceleração linear do centróide;

$r_{i-1,ci}$ - posição do centróide da ligação i até a junta i-1 e $r_{i,ci}$ - posição do centróide da ligação i até a junta i.

O símbolo \times , como por exemplo, em $\omega_i \times (q_i \cdot z_{i-1})$, indica o produto vetorial em todas as equações apresentadas. A representação destas grandezas será realizada juntamente com as forças e momentos sobre a ligação.

Este procedimento ocorre para cada configuração do robô durante a trajetória, sendo assim determinadas, as velocidades e acelerações de todas as ligações do robô a cada instante [2,17].

Nesta formulação, é realizado o equilíbrio dinâmico de forças e momentos de cada ligação, cujas equações representam o acoplamento entre as sucessivas ligações durante o movimento do robô (eq. 3.4 e 3.5). Os efeitos de inércia são incluídos em ambas as equações, como termos negativos, pela aplicação do princípio de D'Alembert, enfocando um sistema dinâmico analogamente a um estático [16]. Tal procedimento é representado na fig. 3.2.

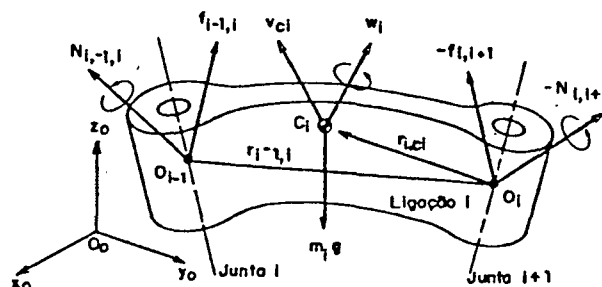


Figura 3.2. Diagrama de corpo livre da ligação i.

$$f_{i-1,i} - f_{i,i+1} + m_i \cdot g - m_i \cdot v_{ci} = 0 \quad (3.4)$$

$$N_{i-1,i} - N_{i,i+1} + (r_{i,ci} \times f_{i,i+1}) - (r_{i-1,ci} \times f_{i-1,i}) - I_i \cdot \omega_i - \omega_i \times (I_i \cdot \omega_i) = 0 \quad (3.5)$$

onde:

$f_{i-1,i}$ - força exercida sobre a ligação i pela ligação i-1;

$N_{i-1,i}$ - momento exercido sobre a ligação i pela ligação i-1 ;

I_i - tensor de inércia da ligação i, dado em relação a um sistema de coordenadas paralelo ao inercial, localizado sobre o centróide da ligação i. As componentes deste tensor são os momentos de inércia da ligação, nas direções dos eixos inerciais.

Nas equações acima, os termos $f_{i,i+1}$ e $N_{i,i+1}$, apresentam-se como negativos, pois referem-se às reações da ligação i+1 sobre a ligação i, que está sendo analisada.

Como a estrutura é uma cadeia, este sistema de equações possui inerente recursividade, ou seja, o torque generalizado exercido pela ligação i sobre a ligação i+1, depende do torque exercido pela ligação i-1 sobre a ligação i.

Computacionalmente, a determinação das forças e momentos inicia-se na extremidade livre do manipulador, onde $f_{n,n+1}$ e $N_{n,n+1}$ são nulos caso não exista carga ou contato entre o manipulador e o ambiente, caso contrário assumem o valor correspondente à carga de trabalho, onde o índice n+1 refere-se à carga ou às forças de contato que atuam sobre a última ligação do robô. Resolvendo-se as equações (3.4) e (3.5), com i variando de n a 1, onde n é o número de juntas do robô, são determinados os momentos ou as forças de acoplamento entre as ligações, cujas componentes nas direções das juntas, constituem os torques generalizados. Este procedimento está representado na figura 3.3.

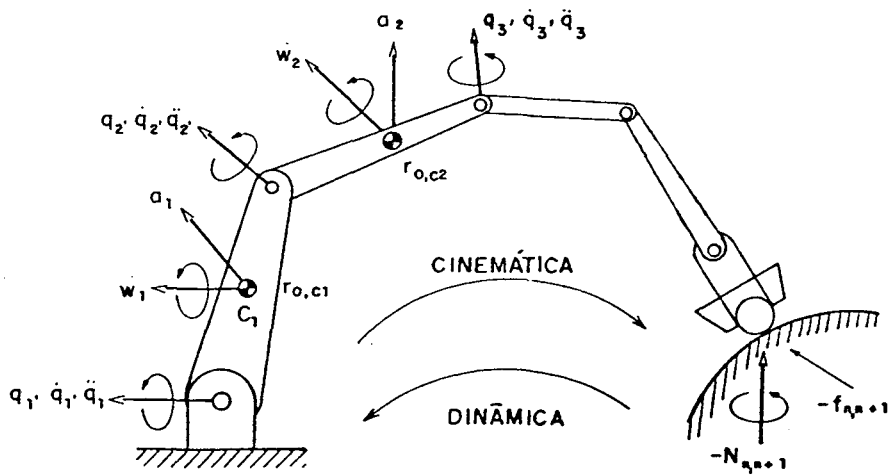


Figura 3.3. Recursividade na cinemática e dinâmica.

Como mencionado, esta formulação não apresenta os torques generalizados τ_s explicitamente relacionados às coordenadas generalizadas q_s , pois estes torques são obtidos como componentes dos efeitos de acoplamento entre as ligações sobre os eixos das juntas, conforme mostra a equação (3.6)

$$\tau_i = \begin{cases} N_{i-1,i} \cdot z_{i-1} + \mu_i \cdot \dot{q}_i & \text{para junta } i \text{ rotacional.} \\ f_{i-1,i} \cdot z_{i-1} + \mu_i \cdot \dot{q}_i & \text{para junta } i \text{ prismática.} \end{cases} \quad (3.6)$$

Nestas equações, os termos relativos às velocidades \dot{q}_i referem-se aos torques devido ao atrito viscoso, sendo μ_i o coeficiente de atrito para a junta i .

Devido à recursividade mencionada, esta formulação mostra-se eficiente computacionalmente, pois o número de operações aritméticas é proporcional ao número de graus de liberdade do robô. No entanto, a recursividade deste método não conduz, diretamente, a um sistema fechado de equações, relacionando as coordenadas generalizadas q_s aos torques generalizados τ_s , entradas e saídas do modelo, respectivamente.

Nesta formulação, os efeitos inerciais, centrífugos, de

Coriolis, gravitacionais e externos não se apresentam separadamente, posto que todos estes efeitos são transmitidos de uma ligação para outra, pelos momentos e forças de acoplamento.

Estes aspectos tornam esta formulação pouco apropriada para a síntese de controle e análise dinâmica de robôs [2,10,18].

Apesar disto, alguns algoritmos foram desenvolvidos utilizando esta formulação, para a dinâmica direta, no cálculo dos coeficientes de um sistema fechado de equações [12]. Estes métodos serão discutidos posteriormente.

3.3. MODELAMENTO SEGUNDO LAGRANGE-EULER.

Nesta formulação aplica-se os princípios da mecânica Lagrangeana para obtenção das equações do movimento, que dependem da determinação da energia lagrangeana, L , do sistema como função das suas coordenadas generalizadas, as quais no modelamento de robôs são seus graus de liberdade. São utilizadas para representação da cadeia cinemática as matrizes homogêneas 4×4 (Eq. 2.1), referenciadas no capítulo 2.

O efeito de encadeamento entre as ligações é representado pela pós-multiplicação das matrizes homogêneas (eq.2.2).

Assumindo que as ligações e juntas são rígidas e sem folgas, ou seja, desconsiderando as deflexões da estrutura, a configuração do robô é função apenas das coordenadas generalizadas, ou seja, das variáveis de juntas do robô [15].

A mecânica lagrangeana baseia-se na obtenção da função de Lagrange, a qual depende da energia cinética e potencial do sistema, sendo assim, como primeiro passo deve-se determinar a energia cinética do manipulador em função das coordenadas generalizadas. Como esta formulação fundamenta-se no princípio do trabalho virtual, as forças de restrição ou vínculos, existentes na estrutura do robô, não estão envolvidas na formulação

[16].

As equações do movimento são obtidas com base na equação de Lagrange.

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \left(\frac{\partial L}{\partial q_i} \right) = \tau_i \quad (3.7)$$

onde: L= Função de Lagrange.

L=E (Energia cinética) -P (Energia potencial);

q_i = coordenada generalizada ou variável da junta i ;

\dot{q}_i = velocidade generalizada da junta i ;

τ_i = torque ou força na junta i.

$Q_i = \tau_i + \tau_{dis,i}$, representa os torques generalizados obtidos considerando-se os efeitos dissipativos $\tau_{disp,i}$.

A energia cinética do manipulador expressa-se como a soma das energias cinéticas das ligações, como mostra a equação (3.8), onde as grandezas mencionadas são referentes ao sistema inercial. Como sistema inercial, é considerado o sistema de coordenadas fixo à base do manipulador.

$$E = \sum_{j=1}^n \frac{1}{2} \left[M_j v_{cj}^t \cdot v_{cj} + \omega_j^t \cdot I_j \cdot \omega_j \right] \quad (3.8)$$

M_j - massa da ligação j ;

onde: v_{cj} - velocidade do centro de massa desta ligação;

ω_j - velocidade angular desta ligação;

I_j - tensor de inércia da ligação j, definido no item anterior.

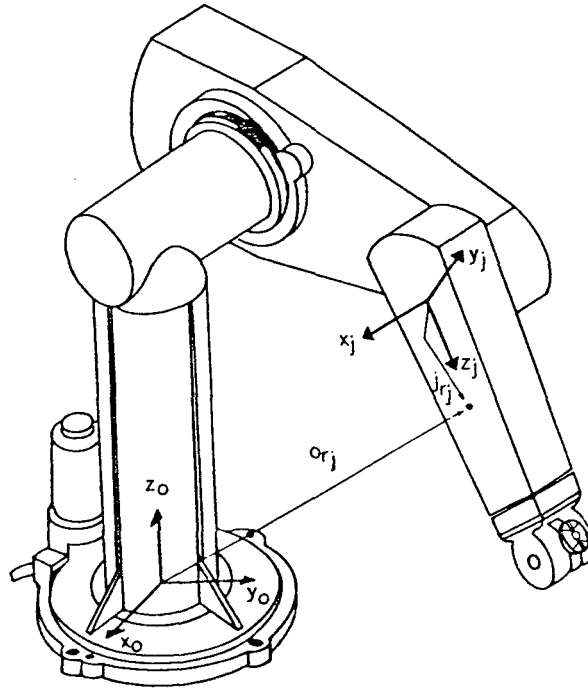


Fig. 3.4. Representação da posição do centróide de uma ligação.

Na determinação das velocidades da equação 3.8, representa-se o vetor posição do centróide da ligação j em coordenadas inerciais por $r_{o,cj}$, como mostra a figura 3.4.

Como mencionado anteriormente, a posição de uma ligação é função das juntas que a antecedem, sendo assim, as velocidades lineares e angulares de uma ligação, também serão dependentes das velocidades das juntas q_i anteriores a esta ligação. Aplicando-se a regra da cadeia obtém-se a velocidade do centro de gravidade da ligação j , como segue:

$$v_{cj} = \sum_{i=1}^j \left(\frac{\partial A_{o,j}}{\partial q_i} \dot{q}_i \right) \cdot r_{j,cj} \quad (3.9)$$

Onde:

$$\frac{\partial A_{o,j}}{\partial q_i} \cdot r_{j,cj} = J_{j,i}$$

-representa o efeito da velocidade da junta i q_i , sobre a velocidade do centróide da ligação j , ou seja, a i -ésima coluna da matriz jacobiana da ligação j ;

$r_{j,cj}$ - representa o vetor posição do centróide da ligação j em

relação ao sistema j , sendo um fator constante.

No cálculo da velocidade angular, tem-se $q_i = \theta_i$ para junta i rotacional, ou $q_i = d_i$ para junta prismática, na eq. 3.10.

$$\omega_j = \sum_{i=1}^j \dot{q}_i \cdot Ja_{j,i} \quad (3.10)$$

Onde:

$Ja_{j,i}$ - representa a i -ésima coluna da matriz jacobiana da junta j , referente às 3 últimas linhas da equação 2.14.

Substituindo-se as expressões de v_{c_j} e ω_j na equação (3.8) e agrupando os termos da matriz jacobiana com os termos de massa e inércia, tem-se a equação (3.11).

$$E = \frac{1}{2} \dot{q}^t [H] \dot{q} = \frac{1}{2} \sum_{j=1}^n \sum_{i=1}^n H_{ij} \dot{q}_i \dot{q}_j \quad (3.11)$$

Onde: $[H]$ define o tensor de inércia do manipulador e $H_{ij}(\{q\})$ representa o efeito da aceleração da junta j sobre o torque na junta i , sendo função da posição do robô, ou seja do vetor de coordenadas generalizadas $\{q\}$.

$$H = \sum_{j=1}^n (M_j [J1_j]^t [J1_j] + [Ja_j]^t [I_j] [Ja_j]) \quad (3.12)$$

Na expressão (3.12) a primeira parcela, refere-se à inércia de translação, enquanto que a inércia relativa à rotação das ligações está representada na segunda parcela.

Esta matriz caracteriza-se por ser positiva definida e simétrica, na qual os termos fora da diagonal demonstram o acoplamento mecânico entre as juntas [12].

Por definição a energia potencial do manipulador é obtida como a soma das energias potenciais de cada ligação, representada pela equação (3.13).

$$P = \sum_{j=1}^n M_j [g]^t r_{0,cj} \quad (3.13)$$

onde: $[g]^t$ - representa o vetor aceleração da gravidade expresso no sistema inercial.

$r_{0,cj}$ - define a posição do centróide da ligação j em relação ao sistema inercial.

Para determinação das equações do movimento do manipulador deve-se substituir as expressões da energia cinética (3.11) e energia potencial (3.13) na equação de Lagrange (3.7), observando que o tensor de inércia $[H]$ e os termos da equação (3.13) não dependem das velocidades generalizadas \dot{q}_s . Fazendo-se esta substituição, tem-se:

$$\tau_i = \frac{d}{dt} \left[\frac{\partial}{\partial \dot{q}_i} \left(\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n H_{ij} \dot{q}_i \dot{q}_j - \sum_{j=1}^n M_j [g]^t r_{0,cj} \right) \right] - \frac{\partial}{\partial q_i} \left[\frac{1}{2} \left(\sum_{i=1}^n \sum_{j=1}^n H_{ij} \dot{q}_i \dot{q}_j - \sum_{j=1}^n M_j [g]^t r_{0,cj} \right) \right] \quad (3.14)$$

Desenvolvendo-se a expressão acima, obtem-se a equação de Lagrange (3.15) para dinâmica de manipuladores, separando os termos relacionados aos efeitos inerciais, centrífugos e de Coriolis (termos não-lineares) e gravitacionais.

$$\tau_i = \sum_{j=1}^n H_{ij} \dot{q}_j + \sum_{j=1}^n \sum_{k=1}^n h_{ijk} \dot{q}_j \dot{q}_k + G_i \quad (3.15)$$

$$\text{Onde: } h_{ijk} = \frac{\partial H_{ij}}{\partial \dot{q}_k} - \frac{1}{2} \frac{\partial H_{jk}}{\partial \dot{q}_i} \quad (3.15-a)$$

representa a influência da aceleração centrífuga da junta j ($k=j$) ou o efeito

de Coriolis ($k \neq j$) sobre o torque na junta i .

$$G_i = \sum_{j=1}^n M_j [g]^t \frac{\partial r_{o,cj}}{\partial q_i} \quad (3.15-b)$$

representa a componente de torque na junta i que compensa o peso das ligações.

Analisando a expressão 3.15-a, observa-se que as não-linearidades do modelo, relacionados aos produtos de velocidades, surgem em função da variação do tensor de inércia com a configuração. Este aspecto é estudado para a obtenção de robôs, cuja inércia seja invariante com a configuração, durante o movimento [14]. O cálculo dos coeficientes h_{ijk} leva em consideração as observações de [19], apresentadas como:

$$h_{ijk} = h_{ikj} \quad (3.15-c)$$

$$h_{ijk} = -h_{kji}, \quad (j < i, k) \quad (3.15-d)$$

$$h_{iji} = 0, \quad (j < i) \quad (3.15-e)$$

Como a expressão (3.15) foi baseada na equação de Lagrange, para a dinâmica de robôs, as cargas externas são adicionadas a equação mencionada através do princípio do trabalho virtual [2]. Aplicando-se um vetor de forças e momentos $\{F_{ext}\}$ ao efetuador do robô, o trabalho virtual realizado é representado por δW . Este trabalho pode ser calculado como :

$$\delta W = \{F_{ext}\}^t \cdot \delta p \quad (3.16)$$

onde: δp representa o deslocamento virtual, devido a carga externa.

Este trabalho também é realizado a nível das juntas do manipulador, como função dos torques realizados pelas juntas e os deslocamentos virtuais produzidos nestas (eq.3.17).

$$\delta W = \{\tau\}^t \cdot \delta q \quad (3.17)$$

onde: δq representa o vetor de deslocamentos virtuais das juntas do robô, relacionados aos deslocamentos virtuais δp .

Pela equação (2.8), pode-se relacionar os deslocamentos virtuais $\delta \underline{q}$, àqueles realizados no efetuador do robô $\delta \underline{p}$, através da matriz jacobiana. Substituindo-se esta expressão na equação (3.17) e comparando este resultado com a equação (3.16), estabelece-se a relação entre as forças externas e os torques realizados pelas juntas para compensá-los eq.(3.18), como segue:

$$\{\tau_{ext}\} = [J]^t \cdot \{F_{ext}\} \quad (3.18)$$

Esta relação é adicionada à expressão 3.15, afim de obter a equação geral em sua forma final (eq.3.19).

$$\tau_i = \sum_{j=1}^n H_{ij} \ddot{q}_j + \sum_{i=1}^n \sum_{j=1}^n h_{ijk} \dot{q}_j \dot{q}_k + G_i + [J]^t \{F_{ext}\} \quad (3.19)$$

onde $[J]^t$ - identifica a transposta da matriz jacobiana do manipulador.

A principal desvantagem da formulação Lagrangeana é o tempo computacional necessário ao cálculo dos termos H_{ij} e h_{ijk} na equação (3.19), que dependem da matriz jacobiana da configuração e das matrizes homogêneas, tornando esta formulação, na forma apresentada, não aplicável à simulação em tempo real, posto que este processo deve ser realizado para cada configuração do robô, ou seja, a cada incremento na simulação [2,19].

Sendo o cálculo do tensor de inércia, a principal etapa na obtenção do modelo dinâmico do robô, e a variação deste fator a causa de não-linearidades, como demonstrado pela equação 3.15a, a análise deste aspecto torna-se relevante para o projeto de robôs com modelos dinâmicos simplificados.

Basicamente, esta análise consiste em estabelecer condições para o projeto da cadeia geométrica, de forma que o robô obtido não possua acoplamento mecânico entre as ligações e tenha sua inércia invariante,

eliminando assim, o cálculo deste fator a cada mudança da configuração e suas não-linearidades.

Estudos realizados, neste contexto, limitaram-se aos robôs articulados, e os resultados foram obtidos apenas para estruturas com 3 graus de liberdade, não considerando as possíveis alterações na carga [14].

Como principais vantagens desta formulação tem-se: o método conduz a um sistema fechado de equações (eq. 3.19), no qual os torques são funções explícitas das posições, velocidades e acelerações das juntas; os diferentes efeitos (inerciais, centrífugos e de Coriolis, gravitacionais e externos) são obtidos separadamente. Isto viabiliza a análise dinâmica, com a utilização de diferentes variáveis para armazenar cada termo da equação (3.19); efeitos de atrito viscoso também podem ser incluídos no modelo, uma vez que dependem apenas das velocidades e dos coeficientes de atrito de cada junta; e a estrutura fechada do sistema obtido (relação explícita entre entradas e saídas) favorece a síntese de controle, ou seja, pode ser aplicado eficientemente, tanto para dinâmica inversa como para direta.

3.4. APLICAÇÕES DA ANÁLISE DINÂMICA.

3.4.1. EFICIENTE SIMULAÇÃO DINÂMICA.

Em 1982, Walker & Orin [12] propuseram um método para simulação dinâmica que aplicava os princípios da formulação Newtoniana para a obtenção dos torques no modelo e a estrutura da formulação Lagrangeana na realização da dinâmica direta, ou seja, estabelecer as acelerações baseadas no modelo dinâmico, visando realizar a simulação do controle, utilizando as vantagens das duas formulações. Devido à sua característica, aplicação das duas formulações, este método será apresentado, embora não tenha sido implementado no sistema desenvolvido, o qual utilizou apenas a formulação lagrangeana, aplicada à dinâmica inversa.

O método utiliza rotinas para a implementação das equações

recursivas (3.1 a 3.6) da formulação de Newton-Euler, tendo como entradas, as posições, velocidades e acelerações programadas das juntas e as cargas externas, e como saídas os torques aplicados às juntas, devido a todos os efeitos, inclusive a gravidade. Em uma das rotinas SUB1, todos os efeitos são considerados, enquanto que na outra SUB2, desconsidera-se os efeitos gravitacionais e de velocidades. Sendo esta última utilizada posteriormente, para o cálculo dos efeitos inerciais.

Pela análise da equação (3.19), observa-se que as acelerações \ddot{q}_{sreal} podem ser calculadas, caso as componentes de torque (não-lineares, gravitacionais e externos) e os termos do tensor de inércia sejam conhecidos, através da inversão do tensor de inércia [H] (eq.3.20).

$$\ddot{q}_{sreal} = [H]^{-1} \{ \tau - (\tau_{nl} + \tau_g + \tau_{ext}) \} \quad (3.20)$$

onde: τ_{nl} - torques relacionados às velocidades;

τ_g - torques relacionados à gravidade;

τ_{ext} - torques devido à carga externa.

Para isto, o método divide-se em 3 fases: primeiramente a dinâmica inversa é realizada através da formulação de Newton-Euler apresentada (os torques totais τ são calculados de acordo com a trajetória programada); numa segunda etapa os torques devido aos efeitos não-lineares, gravitacionais e externos são calculados, através da rotina SUB1, porém sendo chamadas com as posições, velocidades, cargas externas instantâneas e com acelerações nulas, ou seja, não calculando os torques inerciais; e finalmente os termos da matriz [H] são calculados, chamando-se a rotina SUB2, com as posições instantâneas, acelerações unitárias, velocidades nulas, sem gravidade e sem cargas externas. Neste caso a rotina é chamada n vezes (onde n é o número de juntas do robô), sendo para cada vez acessada considerando uma aceleração unitária e as demais nulas.

Exemplificando, quando a rotina é acessada tendo como entradas: as posições instantâneas das juntas, a aceleração da junta 1 unitária e as

demais acelerações nulas, com velocidades de todas as juntas nulas, sem cargas externas e gravidade, devolve como saídas os torques em todas as juntas de acordo com esta condição, ou seja, produzir uma aceleração de 1 (m/s^2 ou rad/s^2) na primeira junta, este vetor é a primeira linha (ou coluna) da matriz [H]. Repetindo-se este processo para todas as juntas, obtém-se todo o tensor de inércia [H].

Com a inversão desta matriz e sua substituição na equação (3.20), obtém-se as acelerações reais das juntas fornecidas pelo modelo, por integração numérica determinam-se as velocidades e posições alcançadas, as quais são comparadas com aquelas programadas para efeito de controle da trajetória.

Com este método, torna-se possível a simulação dinâmica sem o cálculo das matrizes homogêneas e jacobianas, calculando o tensor de inércia do robô de forma indireta, o que aumenta a eficiência computacional do algoritmo.

3.4.2. CÁLCULO DA CAPACIDADE DE CARGA DINÂMICA.

A capacidade de carga de um robô é um dos índices de desempenho, utilizado na seleção do tipo de manipulador para diferentes aplicações.

Existem alguns conceitos sobre a capacidade de carga e sua utilização em robótica. Por exemplo, a capacidade de carga estática (SLCC) é o menor valor da carga estática que pode ser manipulada por um robô, calculada, dentre os valores de todos os pontos da trajetória programada a ser percorrida, tendo em vista que este parâmetro depende da configuração em que se encontra o robô. Esta definição não envolve os efeitos inerciais e de velocidades atuantes durante o movimento.

Caso os efeitos inerciais sejam considerados, o conceito passa a ser de capacidade de carga dinâmica (DLCC), a qual em função das condições

de posição, velocidade e aceleração em um determinado ponto da trajetória pode ser menor que a capacidade de carga estática, dependendo se a influência da carga é adicionada aos efeitos inerciais, ou seja, gera forças e momentos que estão na mesma direção destes efeitos. Contudo, considerando que o robô encontra-se em repouso nos pontos inicial e final da trajetória, a capacidade de carga dinâmica será limitada pela capacidade de carga estática destes dois pontos, ou seja, será menor ou igual a capacidade de carga estática nestes dois pontos [20].

Na determinação da capacidade de carga dinâmica de um manipulador em uma dada trajetória, realiza-se a superposição dos efeitos da dinâmica da carga e do robô. Sendo assim, considera-se que se a carga está firmemente segura pelo efetuador, ambos podem ser tratados como um único corpo rígido, cujo centróide deve ser calculado em função dos centros de massa do efetuador e da carga.

Tendo estabelecido esta relação, através da formulação de Newton-Euler e do princípio de D'Alembert, determina-se as forças e momentos devido à inércia do efetuador e da carga, pela equação (3.21).

$$\{F_c\} = \{ \{ (m_e + m_1)(\dot{v}_c - g) \} ; \{ ([I_c] \dot{\omega}_c + \omega_c \times [I_c] \omega_c) \} \}^t \quad (3.21)$$

onde: $\{F_c\}$ - representa o vetor de 6 componentes, sendo as 3 primeiras $\{ (m_e + m_1)(\dot{v}_c - g) \}^t$ referentes às forças de inércia devido à translação do conjunto (efetuador e carga) e as 3 últimas $\{ ([I_c] \dot{\omega}_c + \omega_c \times [I_c] \omega_c) \}^t$ relativas à inércia de rotação;

m_e - massa do efetuador;

m_1 - massa da carga;

\dot{v}_c - aceleração do centróide do corpo composto pelo efetuador e carga;

ω_c - velocidade angular do corpo composto em torno do seu centróide;

$\dot{\omega}_c$ - aceleração angular do corpo composto em torno do seu

centróide;

I_c - momento de inércia do corpo composto em relação ao seu centróide.

Com a determinação de $\{F_c\}$, ou seja, as forças e momentos referentes à inércia da carga e do efetuador, pode-se através da eq. 3.18, obter os torques nas juntas que compensam estes efeitos (eq.3.22).

$$\{\tau_c\} = [J]^t \{F_c\} \quad (3.22)$$

De acordo com este método, os efeitos de inércia da carga e do efetuador são calculados separadamente dos demais esforços decorrentes das inércias das outras ligações, suas velocidades e gravidade. Desta forma o torque total em cada junta é obtido como a soma destas 2 parcelas.

$$\{\tau\} = \{\tau_{nc}\} + \{\tau_c\} \quad (3.23)$$

onde: τ_{nc} refere-se a todas as componentes de torque menos os efeitos devido ao efetuador e à carga, esta componente é calculada, simulando-se paralelamente um robô descrevendo a mesma trajetória, porém sem efetuador e carga, ou seja, com suas massas e inércias nulas.

Observa-se portanto, que as componentes τ_{nc} independem da carga transportada, sendo assim, são calculadas apenas uma única vez e armazenada durante a trajetória.

Em função dos torques limites dos atuadores, são computados os torques disponíveis para a trajetória, como a diferença entre os torques limites e os torques necessários.

$$\{\tau_{disp}\} = \{\tau_{lim}\} - \{\tau\} \quad (3.24)$$

Este procedimento é repetido para todos os pontos da

trajetória, no qual os torques limites são determinados em função das curvas características dos atuadores, exemplificada na figura 3.5. Com os torques disponíveis são calculados os coeficientes de carga, que identificam a relação entre os torques fornecidos devido à carga e ao efetuador e os torques disponíveis (eq.3.25).

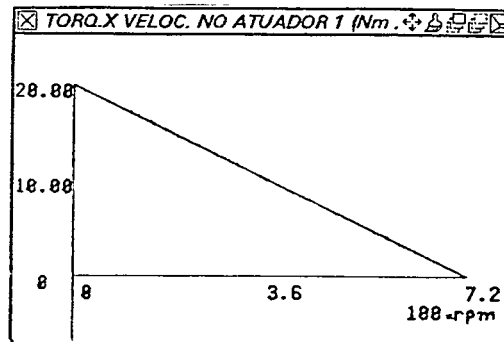


Figura 3.5-Curva de torque x velocidade de um motor elétrico.

$$C_j = \min \{ (\tau_{disp})_i / (\tau_c)_i, i=1,2,\dots,n \} \quad (3.25)$$

onde: C_j - representa o coeficiente de carga para o j -ésimo ponto da trajetória, onde j varia de 1 até m , refere-se ao número de pontos da trajetória, e i varia de 1 a n , referente ao número de juntas do robô.

Sendo assim, a capacidade de carga dinâmica para esta trajetória, pode ser determinada por:

$$m_{DLCC} = \min \{ C_j \cdot (m_e + m_1), j=1,2,3,\dots,m \} \quad (3.26a)$$

$$I_{DLCC} = \min \{ C_j \cdot (I_1), j=1,2,3,\dots,m \} \quad (3.26b)$$

Em muitas aplicações, a capacidade de carga dinâmica é definida apenas pelo valor da massa (m_{DLCC}), tendo em vista que para a maioria dos robôs disponíveis comercialmente, a carga de trabalho é menor que 10 % do peso total do robô e pode ser considerada como uma massa pontual [20].

4.1. INTRODUÇÃO.

Neste capítulo são apresentados alguns aspectos sobre a utilização de simuladores como ferramenta de apoio na atividade de projeto, bem como sua aplicação no contexto da robótica, enfatizando o modelamento e a programação de robôs.

Na engenharia, simulação é a representação e avaliação de um certo sistema através de um modelo que contenha algumas de suas características físicas de importância para o entendimento do seu comportamento. Frequentemente o uso de modelos auxilia na busca de soluções para problemas complexos, sem os quais seria muito difícil, se não impossível, obtê-las. Sendo assim, a simulação atua como fator na economia de tempo, recursos materiais e humanos na solução de problemas [21].

Em função destes aspectos a simulação tem sido utilizada em diversas áreas da engenharia, como por exemplo: na engenharia biomédica, através da simulação gráfica são elaboradas imagens em 3D de órgãos abdominais, a partir de dados bidimensionais, para auxiliar em atividades cirúrgicas [22]; técnicas de simulação a eventos discretos são aplicadas para a análise e controle do fluxo de informações e materiais em sistemas de produção [23]; e em projetos de complexos sistemas, nos quais a não utilização da simulação comprometeria a confiabilidade do projeto ou sua viabilidade econômica [21].

Na simulação de sistemas físicos, diferentes tipos de modelos podem ser utilizados, a saber, modelos analógicos, modelos icônicos e modelos matemáticos. No presente trabalho, tendo em vista a utilização do computador, são implementados para a análise do comportamento dos robôs, os modelos matemáticos representados pelas equações descritas nos capítulos anteriores e outras a serem mencionadas.

Um modelo matemático adequado deve ter como características: realismo na previsão de desempenho; mínimo de complexidade; termos separados

para ações ou fenômenos separados; manipulação direta das expressões e fácil verificação [24].

Com a crescente complexidade dos sistemas físicos, o projeto deixou de ser uma atividade individual, tornando-se um trabalho de equipe, composta por especialistas de diversas áreas. Este aspecto, provocou uma divisão da tarefa de projeto, que teve como consequência a necessidade do estabelecimento de regras e interfaces bem definidas entre as diversas fases do projeto [25].

Estas condições implicaram na necessidade de requisitos, que devem ser atingidos pelos sistemas CAD, tais como: capacidade de entrada e saída gráfica; descrição de objetos e métodos de projeto; banco de dados integrado e uma estrutura de diálogo homem-máquina adequada.

Com a maior utilização do computador, o emprego de modelos matemáticos tornou-se de grande aplicação, para analisar a variação com o tempo de características do modelo, inclusive o efeito de elementos não-lineares, cuja análise através da manipulação matemática sem o computador seria praticamente inviável, devido a complexidade existente. Além deste fator, o computador permite precisão e armazenamento de informações, o que garante melhor documentação e acompanhamento do projeto.

Neste trabalho, será aplicada a simulação dinâmica na análise do comportamento dos robôs. Este procedimento pode ser resumido em quatro etapas: definição do problema, concepção do modelo matemático, cálculo e interpretação dos resultados [26].

No contexto computacional, a simulação dinâmica tem como característica a variação dos parâmetros que formam o modelo matemático do sistema durante a simulação. Neste sentido, este conceito é aplicado no presente trabalho, devido a modificação dos coeficientes que descrevem o modelo dinâmico do robô no decorrer de uma operação, como apresentado no capítulo anterior. Sendo assim, tópicos sobre a simulação aplicada á robótica serão apresentados no próximo item.

4.2. APLICAÇÃO DA SIMULAÇÃO NA ROBÓTICA.

Uma das principais aplicações de simuladores é a análise prévia do sistema a ser projetado antes da sua construção, o que atua como redução do custo total do projeto. Na robótica, este fator envolve a seleção do tipo de robô, modelamento de seus parâmetros, layout da configuração de trabalho, teste da programação, entre outros aspectos [27].

Consideráveis gastos podem ser evitados, caso se trabalhe nos estágios iniciais do projeto com um modelo matemático do robô e do ambiente de trabalho. No caso de um sistema implantado, a elaboração de um programa para verificar dados criados durante o planejamento da manufatura permite a programação das tarefas sem tirar o sistema de operação [4,28].

Através da simulação do robô em seu ambiente de trabalho, permite-se observar se os pontos a serem atingidos pelo efetuador do robô pertencem ao espaço de trabalho; se colisões são possíveis entre o robô e os outros objetos existentes e se o tempo de operação estimado está de acordo com as limitações do robô. Sendo assim, a simulação pode ser considerada, também, como uma ferramenta de apoio na tarefa de depuração do programa executado pelo operador.

Com o intuito de atingir tais objetivos, é fundamental que o sistema simule o robô de maneira a mostrar sua operação, se possível graficamente [29].

Os recentes avanços nas técnicas de projeto auxiliado por computador (CAD) tem capacitado construir e mostrar modelos matemáticos de objetos, visualizar e manipular estes modelos tridimensionais de maneira mais eficiente, viabilizando assim a simulação de robôs [30].

Algumas das tarefas que podem ser simuladas com a computação gráfica, incluem: geração de trajetórias livres de colisão; otimização do layout da célula de trabalho e do tempo de ciclo e demonstração do movimento do robô por animação computacional.

Dependendo dos modelos utilizados, tanto para representar o robô como os objetos no espaço de trabalho, a simulação poderá determinar qual o melhor modelo de robô a ser escolhido para uma dada tarefa. Este processo envolve questões tais como, o menor tempo de ciclo de trabalho admissível e o consumo de energia, pois diversos ciclos de uma tarefa serão executados.

O sistema deve permitir a programação "off-line" do robô e ser independente do robô simulado, tal que diferentes tipos de robôs possam ser analisados, sem a modificação do sistema [31].

O modelo matemático a ser utilizado deve satisfazer dois requisitos básicos: deve ser preciso o suficiente para prover informações que possam ser aplicadas ao sistema real e deve ser compacto e facilmente compreensível, permitindo um processamento simples e com custo razoável.

Em função destes requisitos, os simuladores são desenvolvidos baseados em modelos genéricos, com os quais os robôs existentes podem ser simulados, ou diferentes configurações de robôs podem ser modeladas. Dependendo destes modelos, o sistema terá condições de analisar as limitações do espaço de trabalho de cada robô, a integração deste com outros elementos existentes no seu ambiente de trabalho ou até as características de massa e inércia do robô e dos outros elementos.

Sendo assim, os simuladores podem considerar: os modelos geométricos, com os quais diferentes geometrias são projetadas; modelos cinemáticos que correspondem a determinação das diferentes posições a serem assumidas pela cadeia do robô, através da cinemática inversa, como também a relação entre posições, velocidades e acelerações das juntas do robô, durante a trajetória, e finalmente, os modelos dinâmicos, com os quais as propriedades de massa e inércia das ligações podem ser analisadas, diferentes estratégias de controle avaliadas, ou ainda a flexibilidade das ligações modelada [6].

O modelamento cinemático descreve as características do movimento do robô pela relação tempo e espaço. Está subdividido na componente específica que depende do tipo de robô e na componente independente do

robô, que define as características da trajetória, ou seja, os perfis de velocidade [27].

A especificação das coordenadas de cada ligação do robô é realizada através do conceito de estruturas, as quais consistem em um ponto no espaço e três eixos de coordenadas ortogonais, cuja origem está neste ponto. Este conceito é uma representação gráfica dos sistemas de coordenadas obtidos pelas matrizes concatenadas $A_{0,j}$ ($j= 1$ a n), apresentadas na equação (2.2).

Em função do modelo dinâmico implementado pode-se avaliar a capacidade de carga de um robô, para isto torna-se necessário a representação dos modelos dos atuadores, com os quais os limites de torque e velocidade das juntas do robô são avaliados. Neste sentido, recomenda-se apresentar uma mensagem caso o movimento simulado exceda os limites estabelecidos [32].

4.3. ARQUITETURA DE UM SISTEMA.

Um sistema dedicado ao estudo de robôs deve possuir, basicamente, uma estrutura contendo os seguintes subsistemas: [33]

- . Sistema de gerenciamento - responsável pelas funções de acesso à memória principal, transferência de dados de entrada e saída e periféricos (simulação dos sensores);

- . Módulo específico do robô- inclui o software que realiza as transformações de coordenadas, traduz as aplicações para os comandos específicos do robô, provê a geração de trajetória e considera os limites de posição, velocidade e torque das juntas do robô;

- . Linguagem de programação que é a interface do sistema, pela qual o programador define as ações a serem realizadas pelo robô.

4.4. ASPECTOS DA PROGRAMAÇÃO DE ROBÔS.

A programação de robôs é substancialmente diferente da

programação tradicional, pois na robótica, várias considerações devem ser realizadas, tais como: os objetos a serem manipulados possuem uma variedade de propriedades físicas (massas, forma, tipo de superfície, dimensões, etc); os robôs operam em complexos ambientes, interagindo com outros dispositivos; vários parâmetros devem ser fornecidos para a programação da trajetória.

Os métodos utilizados para a programação de robôs são classificados em duas categorias: programação orientada para o robô e programação orientada para a tarefa (ou objeto) [18].

Na programação orientada para o robô, uma tarefa é explicitamente descrita como uma sequência de movimentos a ser realizada pelo robô, sendo assim este é guiado pelo operador e controlado pelo sistema por toda a tarefa, com cada declaração do programa correspondendo a uma trajetória específica.

Por outro lado, na programação orientada para a tarefa, a operação é descrita como uma sequência de objetivos a serem atingidos, e portanto os movimentos do robô não são explicitamente especificados pelo programador.

No desenvolvimento de linguagens de programação de robôs, visando uma melhor aplicação no contexto de sistemas integrados, os maiores problemas encontrados são:

- .Dificuldade no entendimento geral do conceito do robô integrado ao ambiente de trabalho;

- .Lenta introdução de computadores mais poderosos, que possam integrar os controladores das máquinas e dos robôs, tendo semelhantes sistemas operacionais e interfaces;

- .Ausência de linguagens inteligentes de alto nível para controle de máquinas;

- .Elevado custo de sistemas de visão tridimensional que poderiam prover informações para um banco de dados interfaceado com o sistema de controle do robô e um sistema especialista;

.Alto custo da introdução de técnicas de modelamento sólido a serem usadas no projeto de componentes e processo de manufatura [34].

Devido as características específicas da programação de robôs, são realizados estudos para proporcionar maior facilidade ao usuário nesta tarefa, com a elaboração de linguagens de alto-nível. Afim de realizar este tipo de programação, é necessário fornecer ao sistema as informações básicas sobre as características do robô, seu ambiente de trabalho, objetos a serem manipulados e a especificação das tarefas realizadas.

As pesquisas na área de Inteligência Artificial e robôs inteligentes concentram-se na área técnico-científica e ainda não estão sendo utilizados no ambiente de fábrica [34].

5.1. CONSIDERAÇÕES GERAIS.

Como apresentado anteriormente o objetivo deste trabalho está centralizado no estudo do comportamento dinâmico de robôs industriais, tal tarefa foi realizada com o apoio de um simulador, o qual permite o modelamento de diferentes tipos de robôs, bem como a análise de algumas variáveis envolvidas na sua utilização.

Sendo assim, neste capítulo é apresentado o desenvolvimento deste simulador, enfocando seus aspectos conceituais, requisitos e objetivos.

O simulador tem como principal objetivo, numa primeira etapa, prover uma ferramenta de apoio para estudantes de robótica, enfatizando alguns aspectos da utilização de robôs, tais como: diferentes tipos de robôs; integração com o ambiente; limites de carga dos atuadores; análise da adequacidade de modelos para tarefas estabelecidas; seleção dos atuadores e planejamento de trajetória.

Afim de possibilitar, a inclusão das características de massa e inércia, a análise do acoplamento mecânico entre as ligações do robô durante seu movimento e o cálculo da energia consumida no decorrer da operação, o simulador foi desenvolvido baseado no modelo dinâmico genérico para robôs não redundantes (com até 6 graus de liberdade), o que envolve a maioria dos robôs existentes no mercado.

Esta formulação aplica os princípios da mecânica Lagrangeana para a obtenção do modelo dinâmico do robô, conforme descritos no capítulo 3 [2,18], a qual possibilita a observação dos efeitos que atuam sobre as juntas do robô separadamente, outras considerações sobre esta escolha serão apresentadas.

O sistema foi desenvolvido na estação gráfica do laboratório de CAE/CAD do Departamento de Engenharia Mecânica da UFSC. Em função das características desejadas para o sistema e da disponibilidade do compilador no "hardware" utilizado (Interpro 32CG), foi escolhida para o desenvolvimento do

sistema, a linguagem C.

Esta linguagem apresenta como principais vantagens, portabilidade entre computadores e sistemas operacionais, compartimentalização de códigos e dados, caracterizando-se como linguagem estruturada, eficiência com a geração de códigos concisos e rápidos, manipulação direta de bits, bytes e ponteiros, entre outros aspectos [35]. Além destes fatores, a linguagem C está sendo considerada na implementação do controle de robôs [36].

No projeto de sistemas CAD, a interface com o usuário é uma das principais características e uma fácil entrada de dados torna-se indispensável para sua aplicação.

Em função do número de parâmetros envolvidos na definição da cadeia geométrica do robô e suas características específicas, o sistema dispõe, como entrada de dados, de um arquivo de configurações previamente escolhidas, no qual estão os modelos de robôs mencionados no capítulo 2, (articulado, esférico, cilíndrico, scara e cartesiano) e possibilita a modificação de suas estruturas, dispensando o conhecimento prévio da cadeia cinemática e permitindo a formação de diferentes configurações, partindo-se das existentes.

De acordo com os objetivos propostos e visando permitir a simulação de diferentes modelos ou diversas simulações para o mesmo modelo, faz-se necessário a integração entre os diferentes módulos, bem como a interface gráfica. Sendo assim, o sistema foi elaborado conforme o diagrama apresentado na fig.5.1.

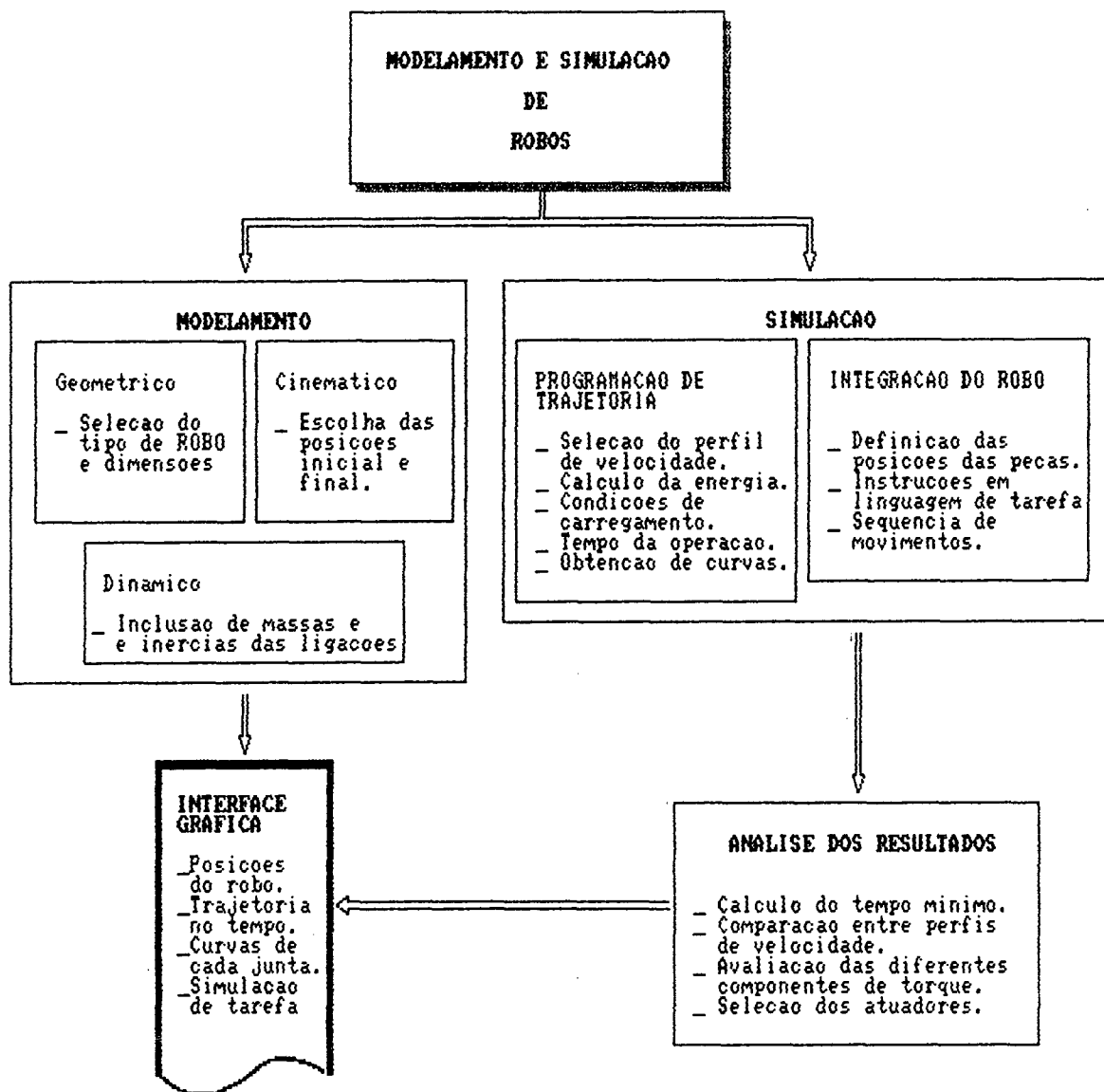


Figura 5.1. Diagrama do sistema.

O sistema utiliza a entrada de dados de forma conversacional e interativa, através de "menus", saída gráfica opcional e possui um "help on line", fazendo-o auto-explicativo. Suas principais funções, bem como o fluxo de informações, são coordenadas por um núcleo gerenciador do sistema. Isto é apresentado como um "menu" principal (fig. 5.2).


```

*****
*   MODELAMENTO DINAMICO DE ROBOS   *
*   EM DESENVOLVIMENTO             *
*   ***GRANTE-EMC-UFSC***          *
*****
*MODELAMENTO DINAMICO DE ROBOS*
1_ARQUIVO DE CONFIGURACOES
2_FORMULACAO DINAMICA
3_PROGRAMACAO DA TRAJETORIA
4_PARAMETROS DE SAIDA
5_OTIMIZACAO DE TRAJETORIA
6_SIMULACAO DE OPERACAO
7_AUXILIO AO USUARIO
8_FIM
Selecione opcao: █

```

Figura 5.2. Menu principal

5.2. MODELAMENTO GEOMÉTRICO.

Tendo em vista, o simulador ser aplicado a diferentes configurações, primeiramente, seleciona-se a geometria do robô a ser modelada, através da função ARQUIVO DE CONFIGURAÇÕES. Isto se realiza através de um menu (fig.5.3), no qual encontram-se os tipos de robôs que podem ser simulados. Os modelos de robôs apresentados nesta função correspondem às configurações básicas mais utilizadas, mencionadas anteriormente.

```

***ARQUIVO DE CONFIGURACOES ***
Configuracoes arquivadas.
1_ESFERICO.
2_ARTICULADO.
3_CILINDRICO.
4_SCARA.
5_CARTESIANO.
Selecione opcao: █

```

Figura 5.3. Configurações arquivadas.

Embora o número de graus de liberdade seja um valor constante para cada configuração, cadeias com um menor número de juntas podem ser modeladas, eliminando-se os últimos graus de liberdade, anulando os parâmetros de juntas.

Como mencionado anteriormente, a seleção do modelo através de

um arquivo diminui o número de dados necessários para a simulação, facilitando desta forma a execução de operações com diferentes modelos em um tempo relativamente curto, comparando-se com aquele no qual todos os parâmetros fossem acessados individualmente.

Exemplificando, na definição da cadeia geométrica tem-se que, para um robô de 6 juntas, são necessários 24 parâmetros geométricos e 6 fatores construtivos, referentes ao tipo de cada junta.

5.3. MODELAMENTO CINEMÁTICO.

Afim de possibilitar a observação da cadeia do robô, também, em movimento, estabeleceu-se como um dos requisitos do sistema, a obtenção da cinemática direta, detalhada no capítulo 2, a qual consiste em determinar a posição e a orientação do efetuador (ou garra), em função das coordenadas das juntas. Este procedimento é usado em diversos pontos do sistema.

Com o objetivo de ter diferentes níveis de programação de trajetória do robô, ou seja, em coordenadas de juntas, em coordenadas cartesianas e em instruções de tarefa, visando possibilitar, também a simulação sem o conhecimento prévio de robótica, tornou-se necessário o desenvolvimento dos algoritmos de cinemática inversa, relacionando a configuração do efetuador às coordenadas de juntas do robô.

Como os métodos genéricos baseiam-se em procedimentos numéricos, os quais demandam várias iterações para a determinação de uma única configuração do robô, optou-se pela obtenção da cinemática inversa através do método geométrico para cada modelo de robô considerado, embora, externamente o sistema mantenha a generalidade, ou seja, simulações de diferentes tipos de robôs seguem a mesma sequência de operações [8].

Realizada a escolha do tipo de robô a ser modelado, é apresentada uma tabela 5.1, contendo os respectivos parâmetros de DENAVIT & HARTENBERG. O sistema apresenta como opção, modificar quaisquer dos valores

apresentados na tabela, porém não os tipos das juntas, persistindo a configuração escolhida anteriormente.

Tipo de configuracao:ARTICULADO					
Numero de juntas=6					
CONFIGURACAO INICIAL					
NJUNT	TETA graus	ALFA graus	A mts	D mts.	JUNTA tipo
1	90.00	-90.00	0.00	1.00	rot
2	0.00	0.00	1.00	0.15	rot
3	90.00	90.00	-0.02	0.00	rot
4	0.00	-90.00	0.00	1.00	rot
5	0.00	90.00	0.00	0.00	rot
6	0.00	0.00	0.00	0.06	rot

Tabela 5.1 Parâmetros Denavit-Hartenberg.

Caso sejam alterados os parâmetros de juntas (d_i ou θ_i) a posição da cadeia está sendo modificada, caso os outros parâmetros sejam alterados, transforma-se a estrutura da cadeia.

Sendo assim, com a modificação dos parâmetros de juntas, também é realizada a escolha da posição inicial da simulação. Existe a opção de entrada com variáveis cartesianas (x,y,z) do efetuador e os cossenos diretores que identificam sua orientação. Neste caso, o sistema processa a cinemática inversa e apresenta a configuração escolhida. Esta modificação é acompanhada, pela apresentação da posição do robô, como mostrado na figura 5.4. A seleção da posição final é realizada da mesma forma que a inicial, ou seja, com parâmetros de junta ou com as coordenadas do efetuador.

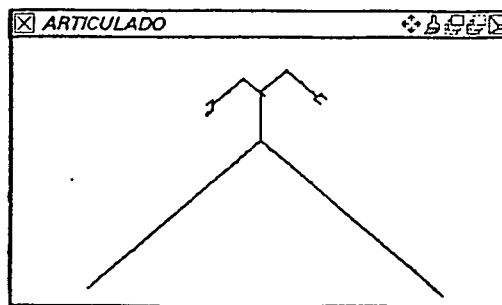


Figura 5.4. Representação de um robô em 2 posições.

Em função da limitação do espaço de trabalho de algumas

configurações, (articulado, esférico e scara), caso a posição escolhida esteja fora deste espaço, o sistema devolverá uma mensagem de erro, bem como instruções para uma escolha adequada.

A representação geométrica é realizada em "wire frame", e as ligações são identificadas por cores. O processo de desenhar a configuração de um robô na tela, utiliza as matrizes concatenadas $A_{0,j}$ descritas no capítulo 2, que relacionam as posições das diferentes ligações do robô, através dos seus sistemas de coordenadas, com o sistema inercial da base do robô.

Atráves dos parâmetros a_i e d_i , são desenhadas as ligações. Inicialmente, identifica-se a origem do sistema de coordenadas inercial como referência, e deste ponto, o sistema desenha cada ligação, primeiro comparando se os parâmetros a_i e d_i (para cada ligação) são não nulos. Neste caso a ligação será representada por 2 linhas. A primeira linha parte da posição do sistema $i-1$ (vetor que corresponde a 4^a coluna da matriz $A_{0,i-1}$) e atinge o ponto de cruzamento entre as 2 linhas, obtido pela soma do vetor mencionado (posição do sistema $i-1$) com o vetor dado por $d_i \cdot z_{i-1}$ (que corresponde a 3^a coluna da matriz $A_{0,i-1}$). A segunda linha parte deste ponto e atinge a posição do sistema de coordenadas i (correspondente a 4^a coluna da matriz A_i), representando cada ligação em função dos seus sistemas de coordenadas, como mostrado na figura 5.5. Caso um dos parâmetros (a_i ou d_i) seja nulo, o procedimento é uma simplificação do exposto, visto que a ligação será representada por uma linha.

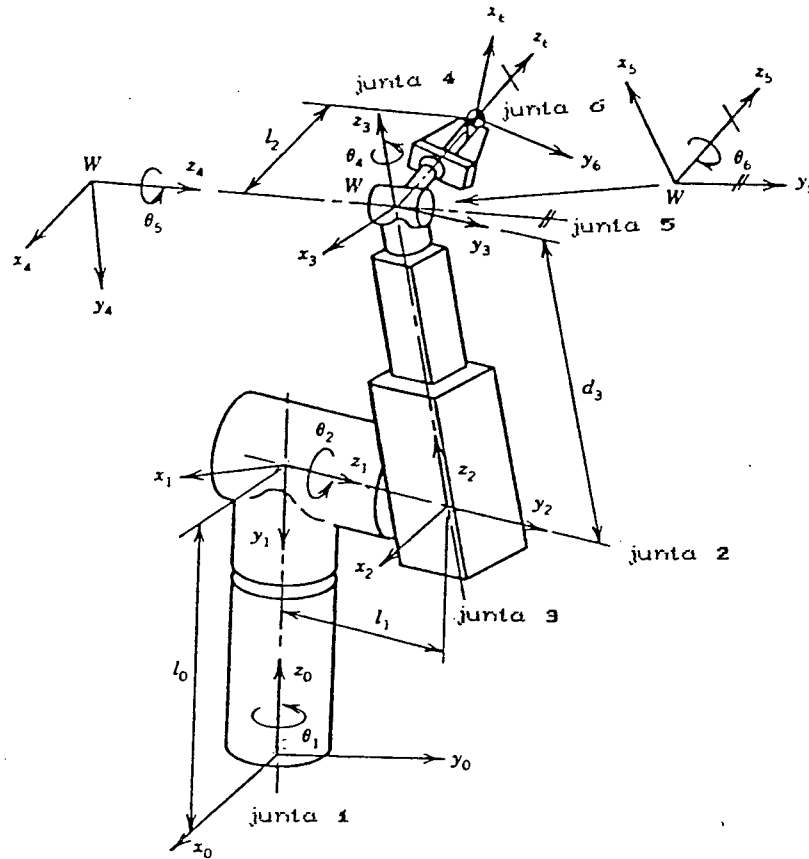


Figura 5.5. Ligações de um robô e seus sistemas de coordenadas.

Após identificados os pontos que caracterizam cada ligação em coordenadas tridimensionais estes são convertidos para coordenadas bidimensionais considerando a perspectiva isométrica, apresentada em [25].

Afim de evidenciar a distinção entre as sucessivas ligações, estas são desenhadas com diferentes cores, fazendo a variável "cor" relacionar-se ao índice de cada ligação. O procedimento descrito, repete-se para cada ligação, realizando desta maneira o encadeamento entre os corpos.

Como o sistema apresenta as diversas configurações do robô durante a simulação com animação, fez-se necessário o armazenamento das variáveis que representam os eixos dos sistemas de coordenadas das ligações a cada incremento da simulação.

5.4. MODELAMENTO DINÂMICO.

Tendo concluído as formulações geométrica e cinemática, ou seja, a definição da cadeia geométrica e suas posições (inicial e final), dando sequência à simulação, realiza-se o modelamento dinâmico da cadeia do robô, através da opção FORMULAÇÃO DINÂMICA.

Nesta função, duas opções são apresentadas, a inclusão apenas das massas, concentradas no centro geométrico de cada ligação (neste caso o sistema modela as ligações como hastes), considerando a posição do centro de gravidade da ligação como mostrado na fig. 5.6, sendo os momentos de inércia calculados pelas expressões (5.1), (5.2) e (5.3), ou a inclusão, das matrizes de inércia referentes ao sistema de coordenadas paralelo ao sistema de cada ligação e localizado no seu centróide.

$$I_{xx} = m_1 \cdot (d/2)^2 + m_2 \cdot d^2/12 \quad (5.1)$$

$$I_{yy} = m_1 \cdot [(a^2/12) + (d/2)^2] + m_2 \cdot [(d^2/12) + (a/2)^2] \quad (5.2)$$

$$I_{zz} = m_1 \cdot (a^2/12) + m_2 \cdot (a/2)^2 \quad (5.3)$$

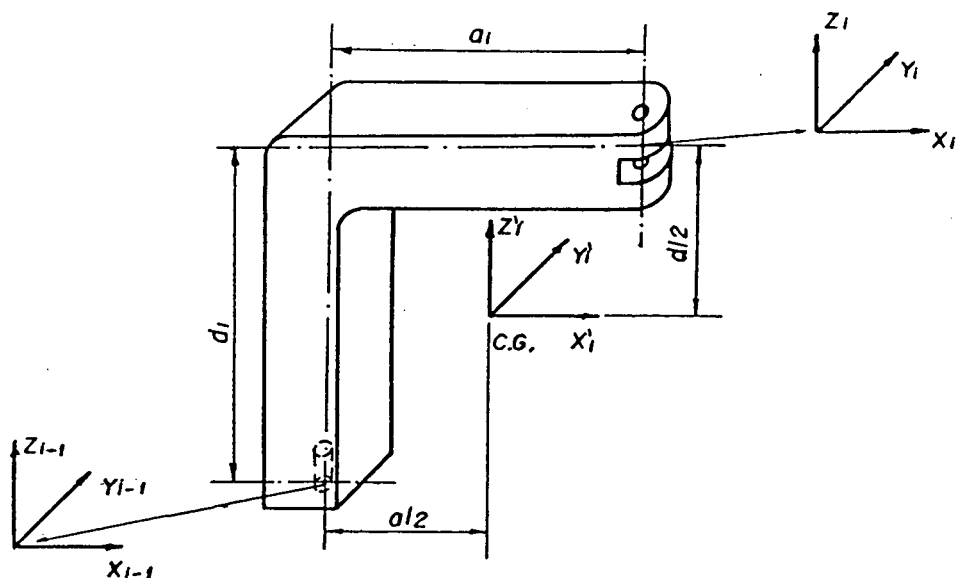


Fig. 5.6. Ligação e seu sistema de referência.

Caso o ângulo de torção da ligação α seja ∓ 90 graus os momentos

I_{yy} e I_{zz} são trocados entre si, onde: $m_1 = a.M/(a+d)$ e $m_2 = d.M/(a+d)$, sendo assim, $m_1 + m_2 = M$, massa total da ligação.

Como mencionado anteriormente, o sistema deve permitir a inclusão das massas e inércias do robô, e o cálculo de torques e energia, ou seja, a simulação dinâmica. Na obtenção do modelo dinâmico de robôs, os métodos mais utilizados são: as formulações de Newton-Euler e de Lagrange.

Visando possibilitar a análise, considerando os diversos fatores que afetam o comportamento dinâmico dos robôs, tais como, efeitos inerciais, acoplamento entre as ligações, cargas gravitacionais e externas, é escolhida a formulação Lagrangeana, posto que nesta, estes efeitos se apresentam separadamente e sua estrutura favorece a implementação de um algoritmo de controle [13,18].

O modelamento segundo a formulação de Newton-Euler é uma alternativa razoável. Caso seja realizado o projeto das ligações do robô, tendo em vista que nesta formulação cada ligação é analisada separadamente e os esforços (forças e momentos), em cada ligação, são calculados separadamente.

O sistema devolve as matrizes de inércia de cada ligação em coordenadas locais (em relação ao sistema no centro de gravidade da ligação) ou globais (em relação ao sistema de coordenadas inercial). Os jacobianos do modelo escolhido, também são calculados e fornecidos, os quais relacionam as velocidades dos centros de gravidade de cada ligação às velocidades das juntas anteriores e são utilizados no cálculo da energia cinética das ligações como apresentado no capítulo 3.

Outro importante fator no projeto de robôs, o tensor de inércia, matriz $[H]$ da equação 3.12, é apresentado nesta função, fornecendo informações sobre o acoplamento mecânico entre as ligações do robô. Por exemplo, analisando-se os elementos fora da diagonal nesta matriz, observa-se quais as ligações que estão, mecanicamente, acopladas entre si, a saber: o elemento da 1^a linha e 2^a coluna desta matriz, demonstra quanto de torque

generalizado (N_m ou N) a junta 1 terá que fornecer para suportar uma aceleração de 1 (rad/s^2 ou m/s^2) na junta 2, ou vice-versa.

Em função deste aspecto, podem surgir não-linearidades no modelo, ou seja, torques devido aos produtos de velocidades, que correspondem ao termo $(h_{ijk} \dot{q}_j \dot{q}_k)$ apresentado na equação (3.19), os quais serão analisados nas componentes quadráticas das curvas de torque obtidas na simulação. Também as componentes gravitacionais são obtidas neste módulo do sistema.

Todos os valores fornecidos neste módulo referem-se à configuração inicial do robô, ou seja, a primeira configuração da simulação, tendo em vista que todos estes termos são alterados com a mudança da configuração no decorrer da simulação.

Este módulo visa fornecer dados sobre o projeto mecânico do robô, tais como, o tensor de inércia do robô, os momentos de inércia de cada ligação e os jacobianos do modelo escolhido na sua configuração inicial.

5.5. PROGRAMAÇÃO DE TRAJETÓRIA.

Concluída a fase de modelamento, este pode ser reinicializado, em qualquer um dos módulos do sistema (geométrico, cinemático ou dinâmico), ou dando sequência à avaliação do modelo escolhido, é realizada a simulação, através da PROGRAMAÇÃO DA TRAJETÓRIA.

Nesta função são apresentadas as características limites de torque e velocidade dos atuadores, juntamente com os fatores de redução e eficiência das transmissões. Todos estes valores podem ser modificados para cada simulação. A seguir o tempo da operação é incluído e a simulação pode ser realizada com ou sem o efeito da gravidade e das cargas externas.

Este módulo foi baseado na teoria de simulação de eventos discretos, a qual consiste em representar um sistema dinâmico contínuo, como uma sequência de eventos definidos pelas variáveis de estado do sistema nos

pontos discretizados [6].

Tal procedimento é utilizado na simulação de sistemas físicos em computadores digitais. No presente trabalho, realiza-se a discretização do tempo total da simulação em intervalos de tempo Δt , considerando como variáveis de estado $[q, \dot{q}, \ddot{q}]^t$, isto significa que o sistema de equações diferenciais que representa o modelo dinâmico, representado pela eq. (3.19), é resolvido para cada incremento de tempo, afim de determinar os torques $\{\tau\}$ [6].

Escolhidos estes fatores, o perfil de velocidade das juntas deve ser selecionado, podendo ser triangular, parabólico, trapezoidal (mas do mesmo tipo para todas as juntas) ou a programação pode ser realizada considerando o espaço cartesiano.

Os perfis de velocidade são funções que relacionam as velocidades das juntas ao tempo do percurso e ao deslocamento, que cada junta terá. Estas funções estabelecem como as juntas serão aceleradas e desaceleradas durante a trajetória. Foram implementados para se avaliar comparativamente, os torques nas juntas com a variação deste parâmetro.

O perfil de velocidade triangular é determinado pelo cálculo da aceleração necessária, para um deslocamento Δq com um tempo de T . Este perfil é definido por acelerações constantes das juntas durante a primeira metade da trajetória, e desacelerações constantes, de mesma intensidade das acelerações, na segunda metade da trajetória, considerando que as juntas partem das posições iniciais e atingem as posições finais com velocidades nulas, estas relações são apresentadas a seguir:

$$a_1 = \frac{4 \times \Delta q_1}{T^2} \quad (5.4)$$

$$\text{onde: } \ddot{q}_1(t) = \begin{cases} a_1 & \text{para } t < T/2; \\ -a_1 & \text{para } t > T/2; \end{cases} \quad (5.5)$$

A equação (5.5) determina as acelerações generalizadas \ddot{q}_1

durante a simulação de uma dada trajetória, sendo a primeira parte correspondente ao movimento acelerado, e a segunda, a relativa às desacelerações de todas as juntas, para que estas alcancem às coordenadas especificadas com velocidades nulas.

A equação (5.5) é integrada para se obter as velocidades $\dot{q}_i(t)$ e estas também são integradas para se obter as posições q_i . Isto está representado nas equações (5.6) e (5.7).

$$\dot{q}_i(t+\Delta t) = \dot{q}_i(t) + \ddot{q}_i(t) \cdot \Delta t \quad (5.6)$$

$$q_i(t+\Delta t) = q_i(t) + \dot{q}_i(t) \cdot \Delta t + \frac{1}{2} \ddot{q}_i(t) \cdot \Delta t^2 \quad (5.7)$$

Onde Δt representa o incremento de tempo utilizado na simulação, ou seja, $\Delta t = T / (\text{número de etapas na discretização})$.

No sistema, o cálculo se processa na ordem apresentada, acelerações, velocidades e posições das juntas. Observando que neste tipo de programação não está sendo analisada a trajetória percorrida pelo efetuador, pois a interpolação realiza-se a nível de juntas do robô.

O perfil de velocidade parabólico baseia-se em uma parábola do segundo grau, na qual as condições de contorno são os deslocamentos totais das juntas Δq_i , o tempo de percurso T e as velocidades iniciais e finais de todas as juntas nulas $\dot{q}_i(0) = \dot{q}_i(T) = 0$. Substituindo-se as condições de contorno na equação da parábola para as velocidades, processando a integração e derivação analíticas, para se obter, respectivamente, as posições e acelerações, tem-se as seguintes equações:

$$\ddot{q}_i(t) = \frac{6 \times \Delta q_i - 12 \times \Delta q_i \times t}{T^3} \quad (5.8)$$

$$\dot{q}_i(t) = \frac{6 \times \Delta q_i \times t}{T^2} - \frac{6 \times \Delta q_i \times t^2}{T^3} \quad (5.9)$$

$$q_i(t) = q_i(0) + 3 \frac{\Delta q_i}{T^2} t^2 - 6 \frac{\Delta q_i}{3 \times T^3} t^3 \quad (5.10)$$

Onde: Δq_i e T são, respectivamente, o deslocamento total da junta i (para cada trecho) e o tempo para este deslocamento, constantes para cada trajetória. As equações acima são funções exatas obtidas analiticamente de acordo com as condições de contorno, mencionadas anteriormente.

O perfil de velocidade trapezoidal é uma variação da implementação do perfil triangular, no qual em um intervalo de tempo durante a trajetória as acelerações de todas as juntas são nulas (eq.5.11). A razão para se simular com este perfil, é a observação de que durante o tempo de acelerações nulas, ou seja, sem os efeitos inerciais, apenas as outras componentes de torque da equação (3.19) são consideradas. Neste modo de programação, as velocidades e posições são obtidas por integração numérica, conforme as equações 5.6 e 5.7.

$$a_i = \frac{\Delta q_i}{t_a (T - t_a)} \quad (5.11-a)$$

$$q_i = \begin{cases} a_i t & t < t_a \\ 0 & t_a < t < T - t_a \\ -a_i t & t > T - t_a \end{cases} \quad (5.11-b)$$

onde: t_a é o tempo de aceleração durante o percurso, que para o sistema foi adotado como $0.3 \times T$.

Nos tipos de perfis mencionados, não existe o acompanhamento da posição do efetuador no espaço de trabalho, pois todos se referem ao nível de juntas e a relação, entre os deslocamentos das juntas e os do efetuador, é não linear para a maioria das configurações de robôs existentes, conforme mencionado no capítulo 2.

Contudo, caso se deseje programar o robô em função das coordenadas do efetuador no espaço de trabalho e sua trajetória em linha reta

entre as posições iniciais e finais, existe a opção do perfil cartesiano, neste caso será questionado sobre a orientação do efetuador durante a trajetória, tendo duas opções : manter a orientação inicial ao longo da trajetória ou calcular a posição separadamente da orientação. Isto implica que as três primeiras coordenadas são obtidas através da cinemática inversa, sem considerar as restantes coordenadas do robô, estas são interpoladas no espaço de juntas, segundo o perfil triangular.

A programação com este perfil se processa com a discretização do tempo entre as posições inicial e final, e o cálculo da aceleração cartesiana do efetuador (eq. 5.12), ou seja, aceleração que o efetuador deve ter para percorrer a distância ΔS , entre as posições inicial e final, em um tempo T, partindo do repouso e alcançando o objetivo também com velocidade nula.

$$\underline{a} = 4 \times \Delta S / T^2 \quad (5.12)$$

$$a(t) = \begin{cases} \underline{a} , t < T/2 \\ -\underline{a} , t > T/2 \end{cases} \quad (5.12-a)$$

A equação da aceleração cartesiana é integrada ao longo do tempo para se obter as velocidades do efetuador eq. (5.13), e destas são obtidas, pelo mesmo processo, suas posições ao longo da trajetória (eq.5.14).

$$\underline{v}(t+\Delta t) = \underline{v}(t) + \underline{a}(t) \times \Delta t \quad (5.13)$$

$$\underline{s}(t+\Delta t) = \underline{s}(t) + \underline{v}(t) \times \Delta t + \frac{1}{2} \underline{a}(t) \times \Delta t^2 \quad (5.14)$$

Onde: $\underline{a}(t)$, $\underline{v}(t)$ e $\underline{s}(t)$ são os vetores das acelerações, velocidades e posições do efetuador no tempo, em coordenadas inerciais.

A cada incremento Δt na simulação, com o vetor $\underline{s}(t)$ é processada a cinemática inversa, considerando a orientação e direcionamento iniciais do efetuador, no primeiro caso mencionado, ou seja, o robô

movimenta-se ao longo da trajetória entre os pontos inicial e final, mantendo durante a trajetória a orientação e direcionamento iniciais definidos. E no segundo caso, o efetuador percorre a trajetória em linha reta, com sua orientação interpolada no espaço de juntas.

Com isto, são obtidas as posições das juntas $q_1(t)$ ao longo da trajetória, por derivação numérica, calculam-se as velocidades $\dot{q}_1(t)$ e as acelerações das juntas $\ddot{q}_1(t)$, como segue:

$$\dot{q}_1(t) = \frac{q_1(t) - q_1(t-\Delta t)}{\Delta t} \quad (5.15)$$

$$\ddot{q}_1(t) = \frac{q_1(t+\Delta t) - 2 \times q_1(t) + q_1(t-\Delta t)}{\Delta t^2} \quad (5.16)$$

No cálculo das acelerações das juntas, foi aplicado o método da diferença central (eq.5.16). Por este método de derivação, as acelerações são determinadas com base nas posições, e não nas velocidades, melhorando a precisão em relação ao método de Euler, embora não sendo incondicionalmente estável [37].

Este procedimento visa simplificar a programação em coordenadas cartesianas, tendo em vista que a orientação relativa do efetuador no espaço é determinada pelo vetor direcionamento \underline{b} e pelo vetor orientação \underline{n} , conforme figura 5.7. Tal simplificação foi considerada, também, por satisfazer os requisitos das simulações admitidas para os testes de comprovação do modelo dinâmico, detalhadas no próximo capítulo.

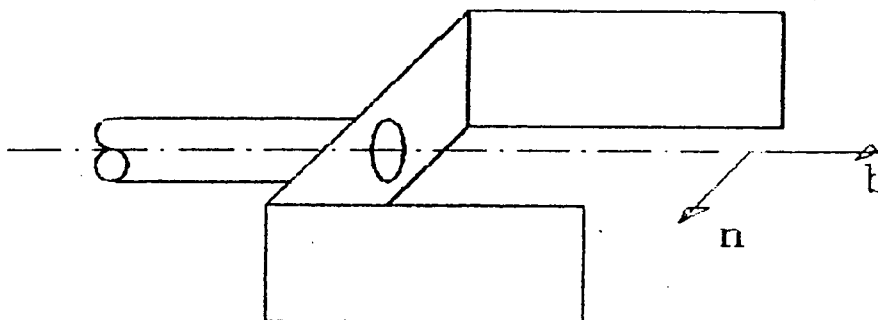


Figura 5.7. Efetuador com vetores \underline{b} e \underline{n} .

Outro método de especificar a orientação do efetuador é através dos ângulos de Euler, o qual não foi implementado no presente trabalho [16].

A escolha entre os perfis de velocidade apresentados, afeta os valores de torques e da energia, ou seja, para as mesmas posições inicial e final, o mesmo tempo de simulação, os esforços nas juntas dependem do perfil selecionado, este aspecto será avaliado no próximo capítulo.

Com o objetivo de ampliar as possíveis tarefas a serem simuladas, bem como realizar as simulações escolhidas para os testes de comprovação, foi desenvolvida a opção de programar trajetórias com configurações intermediárias, permitindo até 3 destas configurações. O tempo, como também, o perfil de velocidade podem ser escolhidos, independentemente, para cada trecho. Como mencionado anteriormente, os perfis de velocidade implementados realizam a programação de trajetória, considerando as velocidades inicial e final nulas, sendo assim, em cada configuração intermediária, o robô passa por uma condição de repouso instantâneo.

Com isto, é possível avaliar numa mesma operação, características distintas de cada intervalo e simular diferentes modos de programação entre as configurações intermediárias, ou seja, em linha reta ou interpolação no espaço de juntas.

Após a seleção do perfil de velocidade, o sistema entra no

módulo de simulação dinâmica no qual, com as posições, velocidades e acelerações de todas as juntas, a cada incremento da simulação, são calculadas as matrizes jacobianas, os momentos de inércia de cada ligação, tensor de inércia do robô. A partir destes elementos são calculadas as componentes de torques como demonstrado no capítulo 3.

Este módulo do sistema envolve a implementação das equações (3.19) para obtenção do tensor de inércia, (3.15-a) referente aos termos centrífugos e de Coriolis, (3.15-b) termos gravitacionais, (3.18) cálculo dos torques externos e a equação (3.19) que adiciona as parcelas calculadas, determinando assim o torque total em cada junta.

A seguir é apresentado um fluxograma com as etapas deste módulo do sistema, mostrando a diferença entre os tipos de programação fig. 5.8.

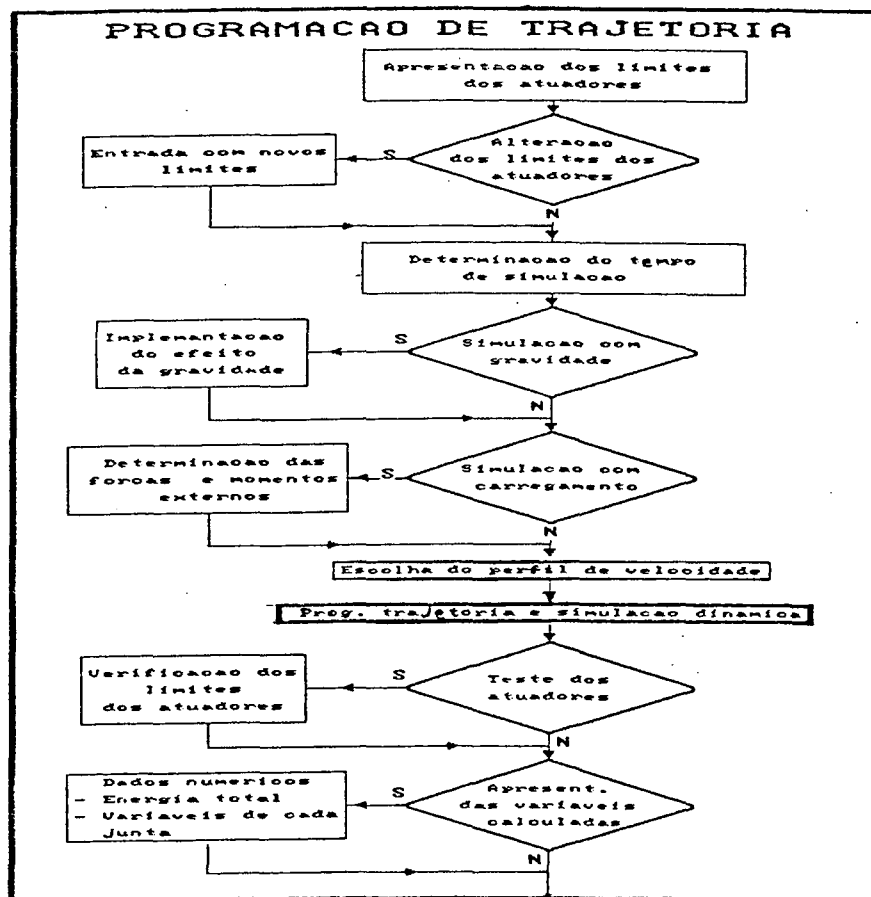


Fig. 5.8 a. Função Programação de trajetória.

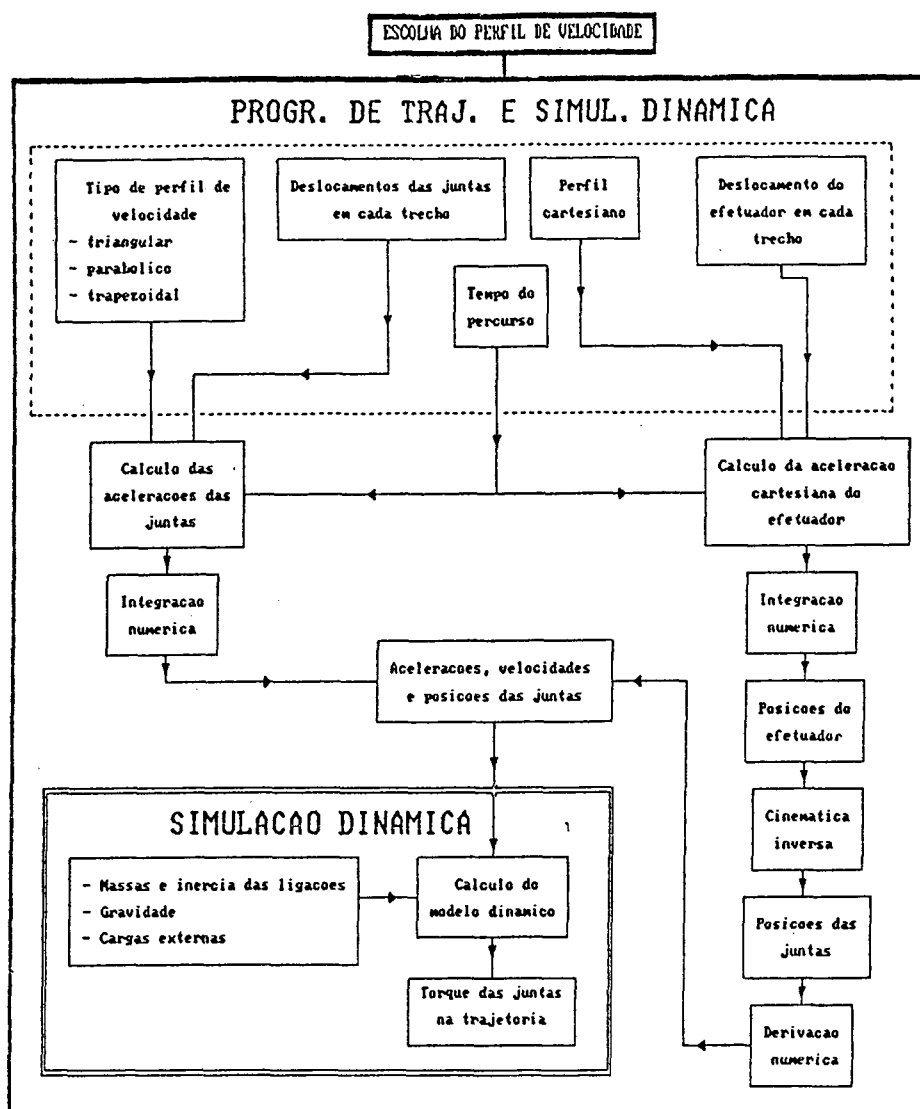


Figura 5.8b. Fluxograma da Programação de trajetória.

A energia é calculada em função dos torques e dos deslocamentos das juntas a cada incremento, considerando o trabalho mecânico realizado por cada junta. A cada etapa é calculado o incremento de energia correspondente ao trabalho realizado por todas as juntas, usando a equação (5.17).

$$dE(e) = \sum_{i=1}^n [q_i(e) - q_i(e-1) \tau_i(e) + \tau_i(e-1)] / 2 \quad (5.17)$$

$$E(e) = E(e-1) + dE(e) \quad (5.18)$$

onde: n- indica o número de juntas do robô;

$q_i(e)$ - representa a coordenada generalizada da junta i na etapa "e"

da simulação;

$\tau_i(e)$ - representa o torque generalizado na junta i na etapa "e" da simulação;

$dE(e)$ -indica o incremento de energia (trabalho mecânico) da etapa "e", para se permitir melhor precisão, utiliza-se a média dos torques entre a etapa "e" e a etapa anterior "e-1", em cada junta do robô e,

$E(e)$ - representa a energia consumida durante a operação até a etapa "e", incluindo esta.

Após este módulo, a simulação terá sido realizada e todos os valores, de posição, velocidade, aceleração, torque de todas as juntas e a energia consumida, para todos os pontos da trajetória, estão armazenados na memória até que uma nova simulação seja executada.

Ainda nesta função, estes valores são apresentados em forma de tabela e a opção de TESTE DOS ATUADORES pode ser escolhida, afim de verificar se, com os valores limites de torque e velocidade dos atuadores e o tempo da operação, é possível ser realizada esta trajetória.

Nesta opção, o sistema compara os valores de torque e velocidade obtidos nas juntas com a simulação, modificados pelas reduções e eficiências das transmissões, com os valores limites determinados pelas curvas características dos atuadores.

Caso os valores calculados sejam maiores que os permitidos, o sistema caracteriza como teste negativo e exibe uma mensagem com a junta responsável pela limitação, bem como o tempo no qual esta ocorreu. Em caso contrário, o teste é considerado positivo.

A comparação se processa, com a representação da função que relaciona diretamente, os torques às velocidades dos atuadores, tendo em vista que ambos são computados em relação ao tempo. Para o teste positivo, esta curva deve estar limitada pelas curvas de torque x velocidade de cada atuador, um exemplo é mostrado na figura abaixo (fig. 5.9). Nesta figura os tempos T_1 e T_2 referem-se a diferentes simulações realizadas, sendo $T_1 > T_2$.

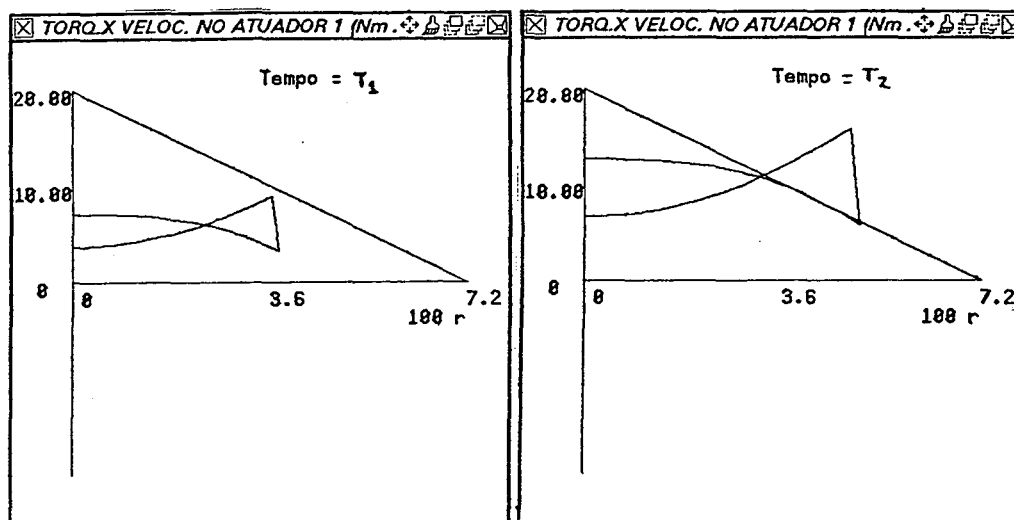


Figura 5.9. Exemplo gráfico de teste de um atuador.

Terminado este módulo o sistema retorna ao menu principal. Neste ponto, as opções podem ser: desconsiderando-se a simulação executada, redefinir o modelo geométrico e/ou os seus parâmetros cinemáticos e dinâmicos, ou ainda o tempo da trajetória (repetindo alguns passos mencionados anteriormente) ou, desejando analisar os valores calculados da operação realizada, deve-se optar pela função PARÂMETROS DE SAÍDA.

Esta função exhibe todos os gráficos referentes as grandezas mencionadas (posições, velocidades, acelerações, torques e a energia consumida), como também a trajetória descrita pelo robô durante a operação.

A escolha dos gráficos é feita pelo menu da fig. 5.10. A trajetória do robô, pela opção 1, é observada com animação, ou apenas os pontos percorridos pelo efetuador no espaço.

```

****PARAMETROS DE SAIDA****

Este modulo permto a obtencao de curvas
caracteristicas do manipulador ,tais como:
comportamento da velocidade,aceleracao,posicao,
trajetoria ou torque com o tempo,tendo como
entrada os dados do perfil de velocidade selecionado
no modulo anterior (programacao de trajetoria).

1_TRAJETORIA
2_POSICAO X TEMPO.
3_VELOCIDADE X TEMPO.
4_ACELERACAO X TEMPO.
5_TORQUE X TEMPO.
6_ENERGIA X TEMPO.
7_CURVAS DE TORQUE .

Selecione a curva:█

```

Figura. 5.10.Parâmetros de saída.

As opções referentes aos gráficos de posição, velocidade, aceleração e torque em função do tempo, respectivamente, 2,3,4, e 5 da figura 5.10, conduzem, no sistema, a outra questão, isto é, qual a junta que se quer analisar, uma vez respondida, surge na tela o gráfico correspondente.

A opção 6 fornece a energia total consumida (trabalho realizado) na operação. A opção 7 direciona para um outro menu no qual a análise das componentes de torque é realizada, ou seja, o estudo comparativo entre as parcelas da equação (3.19), a saber, torque inercial, não-linear (centrífugos e de Coriolis), gravitacional e externo sobre cada junta, e as curvas de torque x velocidade dos atuadores também são examinadas.

Esta análise torna-se possível devido ao fato de que com a formulação Lagrangeana, as componentes de torque são calculadas separadamente e armazenadas em diferentes espaços da memória, ou seja, utilizando diferentes variáveis para cada uma destas componentes a cada etapa da simulação. Este é o principal módulo para a tarefa de análise.

O sistema permanece nesta função até que não sejam solicitados mais gráficos, sendo assim, retorna para o menu principal.

Como todos os valores estão armazenados na memória, esta função poderá ser acessada novamente, com os dados da última simulação executada.

Caso a limitação de carga do modelo do robô seja avaliada, deve-se escolher a opção 5 na fig. 5.2, OTIMIZAÇÃO DE TRAJETÓRIA. Nesta função, a partir do teste positivo, ou seja, as limitações de torque e velocidade foram satisfeitas na última operação realizada, o sistema decrementa o tempo total da operação, executa a simulação e o teste dos atuadores, até que seja obtido um teste negativo. Sendo assim, obtem-se o tempo mínimo necessário para esta trajetória de acordo com as limitações do modelo.

Com a diminuição do tempo da trajetória, para as mesmas posições inicial e final, serão necessários maiores valores de aceleração e velocidade das juntas do robô, durante a simulação, como mostram as equações dos perfis de velocidade, pela análise da equação (3.19), isto requer maiores valores de torques dos atuadores, o que implica em alguma violação dos limites de carga dos atuadores com a sucessiva diminuição do tempo de operação.

5.6. SIMULAÇÃO DE OPERAÇÃO.

As operações descritas até o presente, requerem um certo conhecimento de robótica para sua execução, pois as instruções fornecidas ao sistema, referem-se ao movimento relativo das juntas do robô ou a determinação da posição e orientação do efetuador no espaço de trabalho.

Contudo, este tipo de programação possui algumas desvantagens, tais como : relativa complexidade, pois necessita de experiência do programador; a programação pode se tornar tediosa, especialmente para tarefas complexas e as declarações podem não ser facilmente compreensíveis [22].

Com o intuito de reduzir esta limitação, foi desenvolvida a função SIMULAÇÃO DE OPERAÇÃO, a qual relaciona o robô a um ambiente de mesas e peças. Esta função baseia-se no conceito de programação a nível de tarefa, na qual as instruções são fornecidas, ao sistema em declarações de alto nível (linguagem em forma textual), ou seja, tipo de declaração facilmente

compreendida por um operador (ex .: pega e coloca).

Sendo assim, o sistema converte estas declarações em instruções relativas ao movimento das juntas do robô (baixo nível) e processa o sequenciamento das tarefas: aproximação, carregamento, manipulação e descarregamento da peça.

Apenas a posição inicial e os parâmetros dinâmicos, incluídos nas etapas anteriores, serão utilizados nesta função, posto que a posição final depende da tarefa a ser simulada.

Em função da posição da peça a ser manipulada, o sistema calcula a posição de aproximação, considerando uma distância de 200 mm acima da peça, a orientação de aproximação é fixa (efetuador direcionado em y e orientado no eixo x), Com a posição e orientação iniciais e as calculadas para a aproximação, o sistema realiza a programação de trajetória do primeiro trecho, utilizando, implicitamente, as funções de perfil de trajetória anteriormente apresentadas.

Este procedimento repete-se entre a posição de aproximação e a posição da peça, para realizar o carregamento. A configuração de aproximação está mostrada na figura 5.11.

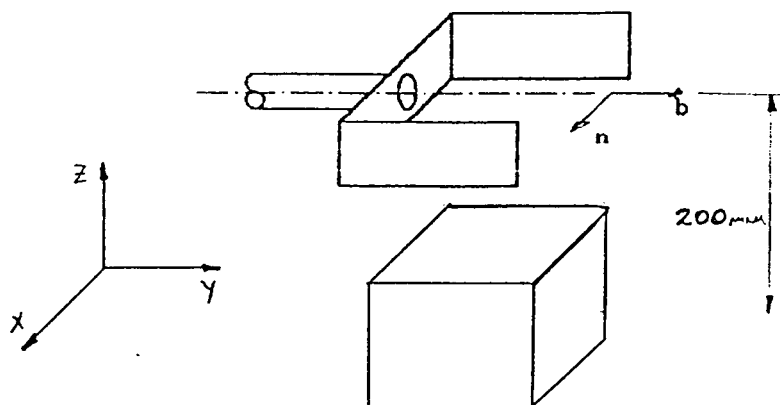


Figura 5.11 Aproximação do efetuador em relação à peça.

A manipulação ocorre em 2 partes, do instante de carregamento

da peça ao ponto de aproximação do objetivo, e deste para o ponto de descarregamento.

Neste período, o sistema de coordenadas da peça manipulada, passa a ser função do sistema de coordenadas do efetuador, o qual por sua vez, é obtido como função da matriz $A_{0,n}$, apresentada na equação (2.2).

Este procedimento visa possibilitar a representação gráfica da peça "sendo manipulada" pelo robô.

O descarregamento consiste na liberação do sistema de coordenadas da peça em relação ao do efetuador e a realização da trajetória do ponto objetivo para 200 mm acima deste.

Afim de facilitar a entrada de dados deste módulo, as dimensões das peças e mesas são constantes, porém as massas das peças são variáveis.

Segue-se uma descrição sequencial das instruções que são fornecidas ao sistema nesta função. Primeiramente, o ambiente é formado pelo robô na posição inicial e 2 mesas, uma com 2 marcações e outra sem objetos.

O sistema solicita posicionar 2 peças (blocos), sobre a mesa sem marcas, pelas coordenadas (x,y) de cada peça, obtendo na tela a fig. 5.12.

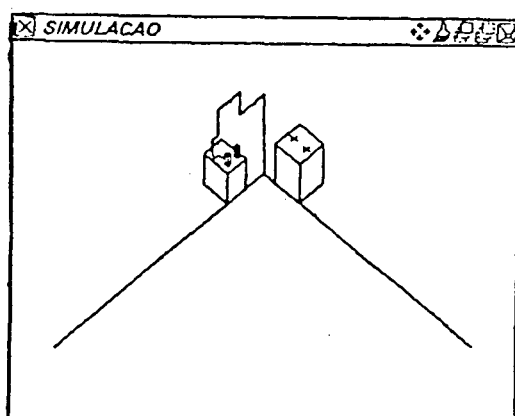


Fig. 5.12. Montagem da simulação.

Na simulação dinâmica do carregamento, ou seja, considerando a massa da peça manipulada durante o processamento do modelo dinâmico, esta é adicionada à massa do efetuador, e admite-se ser pontual (sem inércia rotacional). Tal hipótese é considerada para pequenas massas [20].

Foram utilizados os recursos gráficos de modo que as peças são identificadas como vermelha e verde.

A seguir são incluídas as massas das peças, após isto, o sistema entra em ciclo, no qual devem ser selecionadas: a peça a ser manipulada; em qual posição deve ser colocada e os tempos relativos a cada movimento, conforme mostra a fig. 5.13.

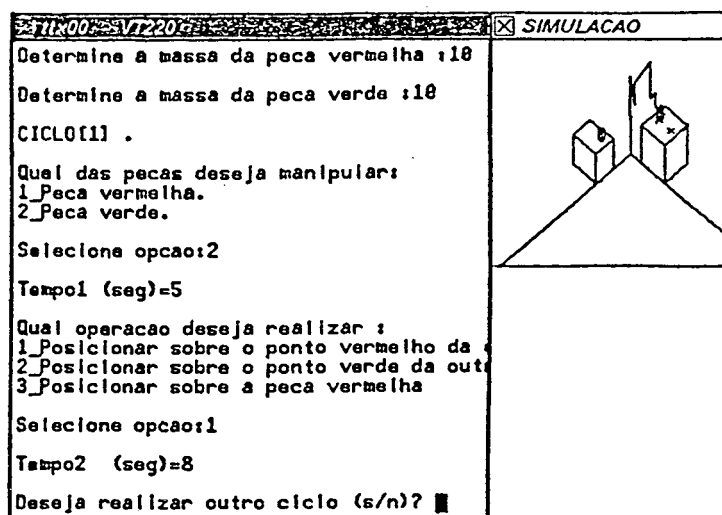


Fig. 5.13. Ciclo de simulação.

Com os dados fornecidos, o sistema processa a programação das várias trajetórias, em função da peça a ser manipulada e da posição objetivo, estabelecendo o sequenciamento das tarefas: aproximação; carregamento; manipulação e descarregamento da peça.

Após o processamento, é mostrada na tela a operação realizada, considerando a velocidade da animação, uma função dos tempos relativos previamente estabelecidos.

A animação ocorre com a plotagem (e eliminação) das sucessivas configurações, obtidas com a simulação, através das trajetórias implicitamente programadas pelo sistema e armazenadas como variáveis que representam as posições das ligações determinadas pelas matrizes concatenadas $A_{o,j}$ ($j= 1$ a n), indexadas a cada incremento da simulação.

Após cada ciclo, o sistema possibilita a continuação da

simulação, em caso afirmativo, um outro ciclo se inicia, considerando a mudança provocada pelo ciclo anterior, ou seja, a modificação da posição da peça manipulada. Os valores relativos ao último ciclo realizado ficam armazenados na memória e são analisados na função PARÂMETROS DE SAÍDA.

Exemplificando a relativa complexidade envolvida em uma operação do tipo apresentado, caso esta fosse realizada com instruções relativas às coordenadas do efetuador, tem-se que, para cada configuração-chave, a saber, aproximação da peça, carregamento, aproximação do objetivo e descarregamento da peça, seriam necessários 3 parâmetros de posição e 6 parâmetros referentes aos vetores direcionamento e orientação do efetuador.

Além disto, estes dados deveriam ser previamente calculados pelo programador, posto que seriam funções da peça e do objetivo escolhidos. Tais aspectos, tornariam a realização desta simulação bastante complexa e sujeita a erros, em função das declarações necessárias.

Esta função não pretende representar um ambiente complexo, tal como o existente em uma unidade de fabricação, pois isto está fora do objetivo deste trabalho, mas sim, apresentar uma metodologia básica utilizada para simplificar uma simulação de um robô. Envolvendo desta forma os 5 elementos básicos de um sistema para simulação de robôs [38], a saber:

.Modelamento do espaço de trabalho, realizado com a representação de outros objetos além do robô, o que possibilita a integração entre diferentes partes do sistema;

.Modelamento do robô, considerando as características geométricas, cinemáticas e dinâmicas do manipulador, bem como a influência da carga manipulada, sobre as juntas do robô;

.Planejamento da tarefa, realiza-se pelo sistema, em função de determinadas opções, o sequenciamento das sub-tarefas que formam a tarefa desejada;

.Programação de trajetória- determina para cada trecho da

operação, referente a cada sub-tarefa, os valores de posição, velocidade, aceleração e torque em cada junta do robô, necessários a cada intervalo;

.Simulação gráfica- atua como verificação "off-line" da simulação programada, neste sentido, pode ser considerada como ferramenta de depuração do programa, pela observação de características, tais como, não colisão entre o robô e os outros objetos no ambiente, observação de todos os pontos atingidos pelo efetuador durante a simulação, entre outros.

Com o intuito de proporcionar dados sobre a utilização do sistema, foi desenvolvida a função AUXILIO AO USUÁRIO, a qual refere-se a um "help on line". Esta função fornece informações sobre as diversas partes do sistema e sua aplicação, tais como, características das configurações arquivadas, modelamentos cinemático e dinâmico, seleção dos perfis de velocidade e programação da trajetória, otimização em relação ao tempo e simulação de operação. Esta função pode ser acessada sempre que o sistema estiver no "menu" principal, sem alterar os passos até então realizados, ou seja, após consultar esta função, volta-se ao sistema, continuando com todas as operações já executadas.

6.1. APRESENTAÇÃO DOS OBJETIVOS.

Este capítulo tem por objetivo apresentar os exemplos de robôs simulados com o sistema desenvolvido, analisar os resultados obtidos, bem como outras aplicações possíveis do sistema, tais como, cálculo do tempo mínimo de uma trajetória e avaliação da energia com diferentes perfis de velocidade.

Devido ao fato de não existir sistemas semelhantes à disposição para comparação, os resultados foram comparados com exemplos encontrados na literatura pesquisada, levando em consideração, os aspectos de generalidade, com relação ao modelo dinâmico desenvolvido, isto é, o mesmo modelo é utilizado para diferentes tipos de robôs e similaridade entre os testes simulados e os exemplos comparativos.

Nos testes realizados as trajetórias foram programadas considerando o perfil de velocidade cartesiano apresentado no capítulo 5, em alguns trechos considerando-se a orientação inicial do efetuador mantida e em outros desprezando este fator. O objetivo é obter para comparação as curvas de torque no tempo, para validação do modelo dinâmico.

A seguir serão apresentadas as simulações de dois tipos de robôs para fins de comparação das curvas de torque e de um terceiro tipo para análise das curvas de torque, com a variação do perfil de velocidade utilizado na programação da trajetória.

6.2. SIMULAÇÃO DE UM ROBÔ CILÍNDRICO.

Como descrito no capítulo 2, o modelo cilíndrico apresenta a primeira junta rotacional na vertical e duas juntas prismáticas, sendo uma na vertical e a outra na direção radial.

Esta simulação foi baseada em condições obtidas da referência [6], que consiste na movimentação de um robô cilíndrico, o qual apresenta 4 graus de liberdade, conforme mostrada na figura 6.1. Esta configuração tem

como parâmetros de Denavit-Hartenberg, a tabela 6.1.

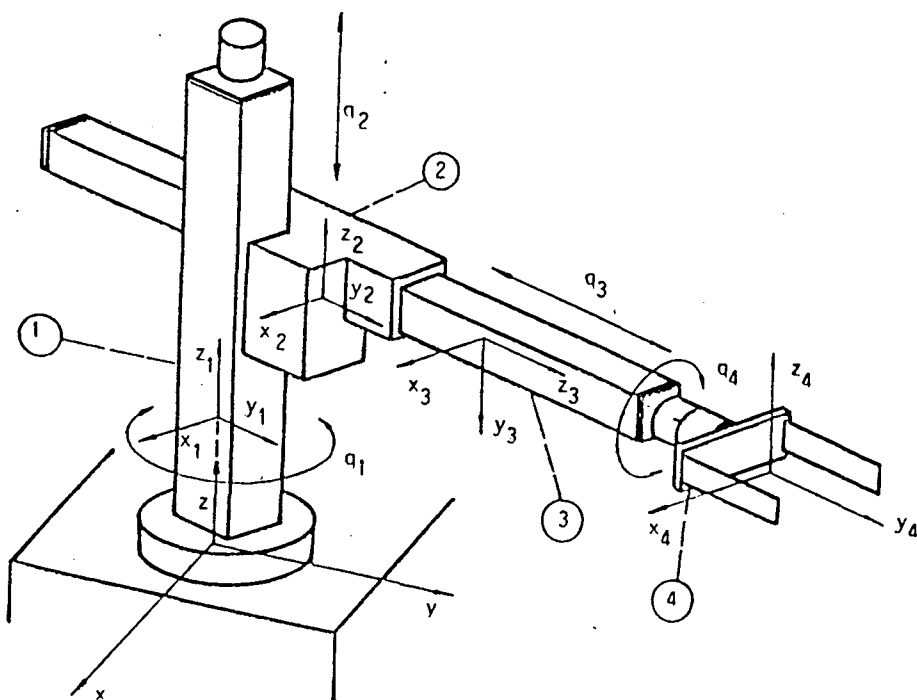


Figura 6.1. Manipulador cilíndrico [6].

ligação	θ (graus)	α (graus)	a(m)	d(m)
1	θ_1	0.0	0.0	0.2
2	0.0	-90.0	0.1	d2
3	0.0	-90.0	0.0	d3
4	θ_4	0.0	0.0	0.07

Tabela 6.1. Parâmetros de Denavit-Hartenberg.

O modelo dinâmico implementado para a simulação deste exemplo na literatura, aplica o princípio da equação de Appel [6], diferente do implementado no sistema desenvolvido, baseado na formulação de Lagrange.

A simulação está mostrada na figura 6.2, equivalendo a uma operação na qual o robô passa por duas configurações intermediárias (A_1 e A_2) e voltando a configuração inicial ($A_3 = A_0$), sendo as trajetórias programadas com o perfil cartesiano, o que corresponde a uma linha reta no espaço a ser

ligação	1	2	3	4
M (kg)	10.0	7.0	4.15	3.5
I _x (kg.m ²)	0.0	0.0	0.0	0.02
I _y (kg.m ²)	0.0	0.055	0.318	0.011
I _z (kg.m ²)	0.029	0.0	0.0	0.011

Tabela 6.2. Valores de massa e inércia das ligações.

Nesta função obtem-se o tensor de inércia do robô para a configuração inicial (eq.6.1), o qual mostra o acoplamento entre os torques generalizados das juntas (1 e 3), caracterizado pelo elemento H_{13} , como também a não existência de acoplamento entre as demais juntas.

$$[H] = \begin{bmatrix} 1.851 & 0.0 & -0.765 & 0.0 \\ 0.0 & 14.65 & 0.0 & 0.0 \\ -0.765 & 0.0 & 7.65 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.11 \end{bmatrix} \quad (6.1)$$

Tendo em vista a simulação desejada, na função PROGRAMAÇÃO DE TRAJETÓRIA foi adotado tempo total da simulação de 4,5 s. A opção escolhida para a trajetória foi com duas configurações intermediárias, sendo estas estabelecidas em parâmetros de junta, e os tempos correspondentes entre cada trecho da simulação de 1,5 s, incluídos nesta função. A simulação foi realizada considerando o efeito da gravidade e sem cargas externas, posto que estas foram incluídas nas massas e inércias da última ligação.

Como mencionado anteriormente, toda a simulação foi realizada com o perfil cartesiano entre as configurações intermediárias (fig. 6.3). Para isto, a programação foi desenvolvida desconsiderando-se a orientação, tendo em vista que a junta 4 é acionada com um perfil de velocidade triangular e o efetuator desloca-se com o vetor de direcionamento variando na direção radial

durante a trajetória nos primeiros dois trechos.

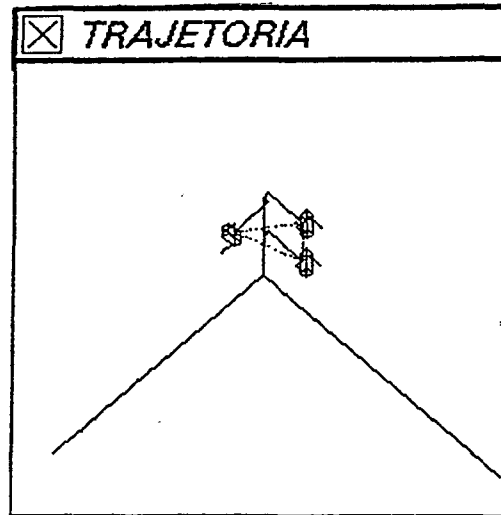


Fig. 6.3. Representação da simulação realizada pelo sistema.

A seguir são apresentadas as curvas de posição, de velocidade e de torque das juntas correspondentes a esta simulação, respectivamente, figuras 6.4, 6.5 e 6.6.

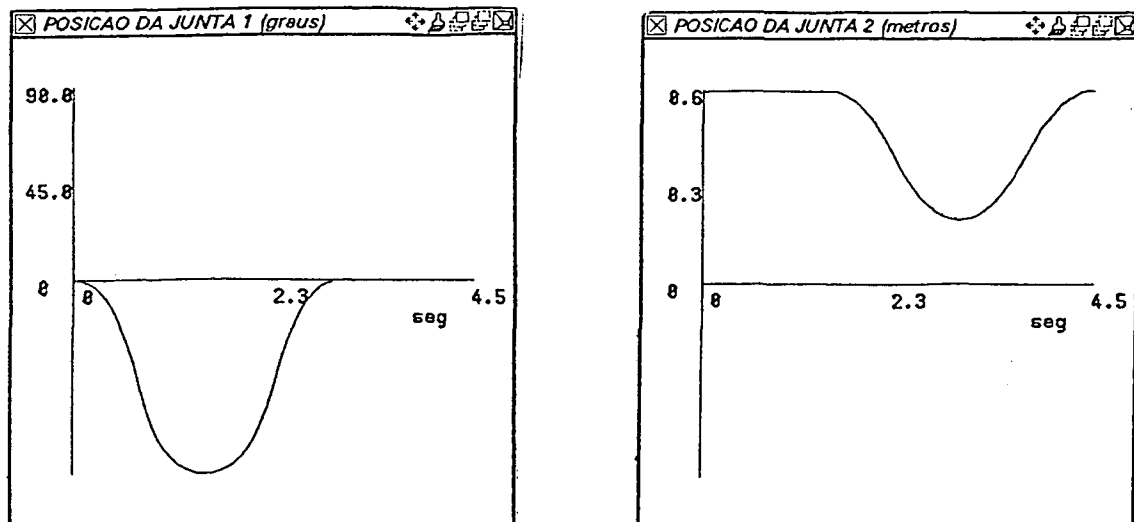


Figura 6.4a. Curvas de posição das juntas 1 e 2.

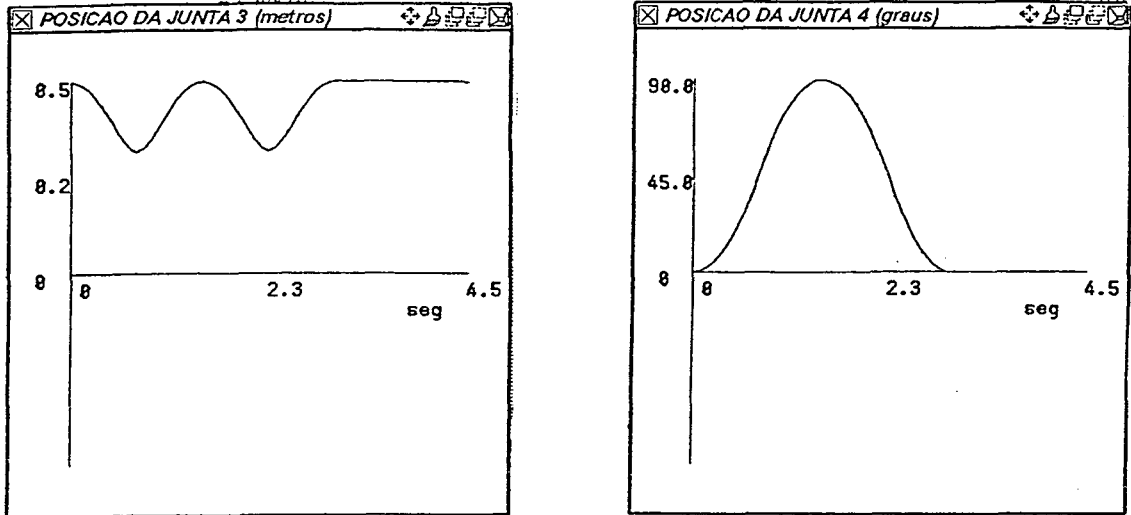


Figura 6.4b. Curvas de posição das juntas 3 e 4.

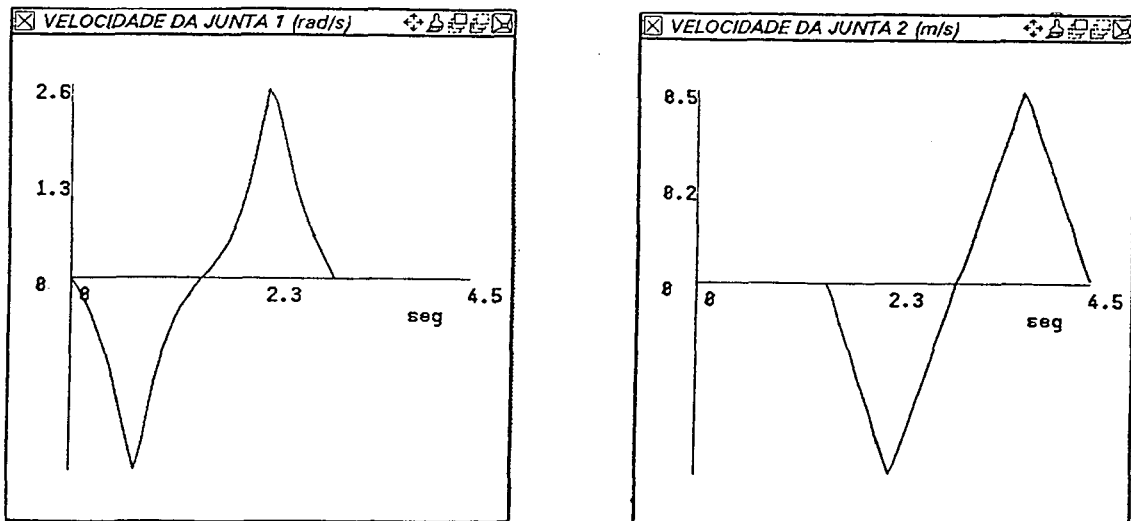


Fig. 6.5-a. Curvas de velocidade das juntas 1 e 2.

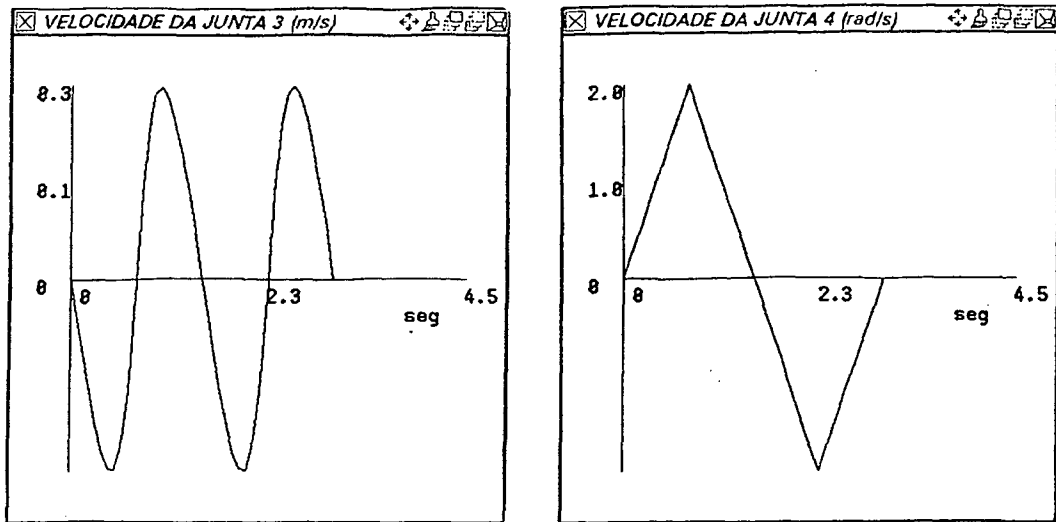


Figura 6.5-b. Curvas de velocidade das juntas 3 e 4.

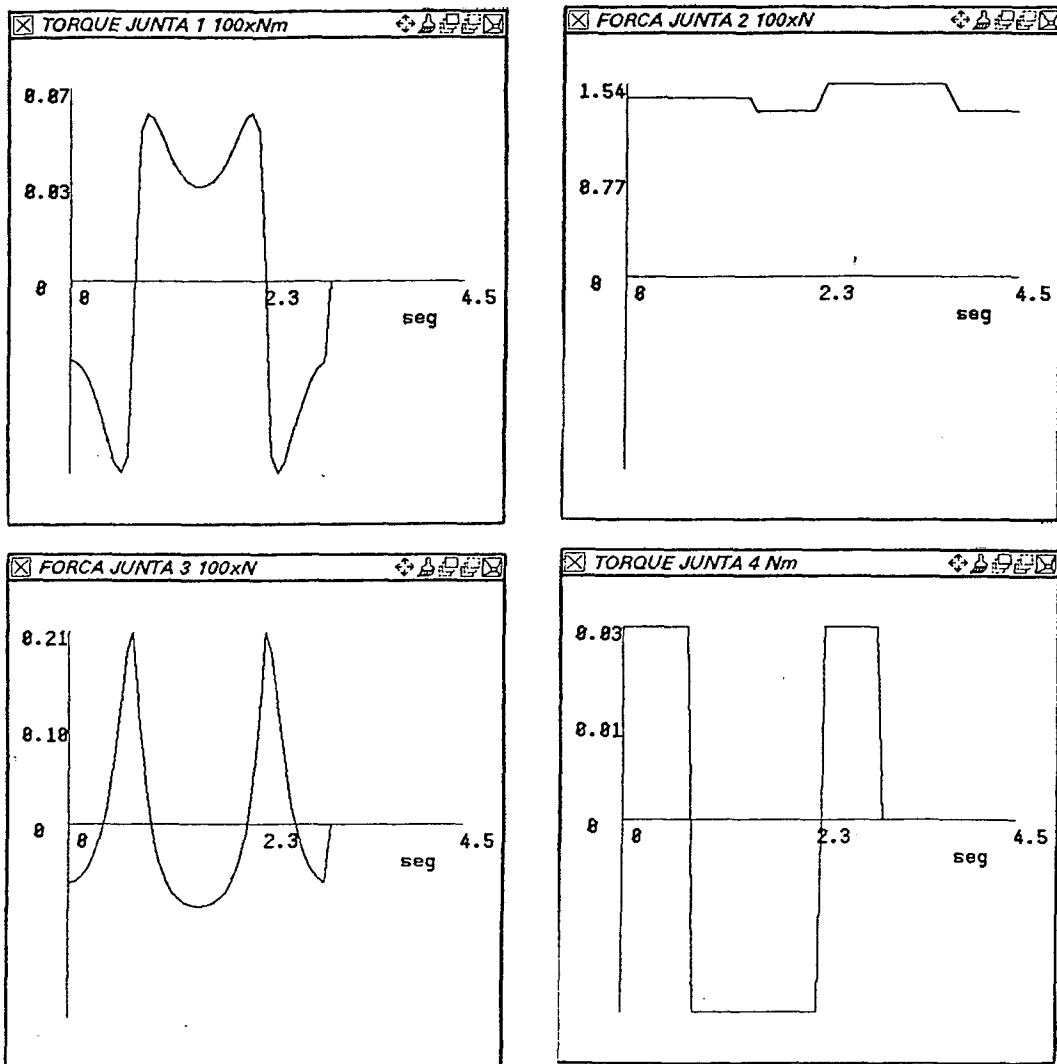


Figura 6.6. Curvas de torque das juntas.

Como mostram as curvas de velocidade (figs. 6.5-a e b), as juntas 2 e 4 comportam-se com perfil de velocidade triangular, o que é coerente tendo em vista que a junta 2 é prismática e na vertical, o que implica que nos trechos 2 e 3 da simulação a componente da aceleração cartesiana do efetuador na direção vertical é igual a aceleração da junta 2. Isto é explicado, tendo em vista que a trajetória com perfil cartesiano é programada considerando um perfil de velocidade triangular no espaço cartesiano, como apresentado no capítulo 5, e a junta 2 por ser prismática e na vertical desloca-se em relação a este espaço. A junta 4, como apresentado, é programada com perfil de velocidade triangular, em função de não se considerar a orientação do efetuador durante a trajetória. Tal efeito, é equivalente a acelerações em degrau e o fato de não existir acoplamento entre as juntas 2 e 4 e as demais juntas do robô, (como mostra os termos do tensor de inércia eq.6.1), gera curvas de torque para estas juntas também na forma de degrau, como demonstra a fig.6.6.

Com relação às juntas 1 e 3, observa-se que suas velocidades apresentam-se como curvas deformadas, isto ocorre em função da trajetória programada (linha reta) implicar em acelerações não constantes nestas juntas. A curva de velocidade da junta 3 em forma ondulatória, demonstra que esta junta deve ser recolhida e estendida, nos dois primeiros trechos da simulação, afim de manter o efetuador em linha reta. Devido ao baixo acoplamento mecânico existente entre estas juntas, suas curvas de torque seguem a forma das curvas de aceleração de cada junta, e apresentam uma forma simétrica em relação aos dois primeiros trechos, o que explica-se pelo fato destes serem correspondentes a deslocamentos iguais em tempos iguais para estas juntas. A seguir, são apresentadas as curvas de posição e de torque da literatura [6], para efeito de comparação.

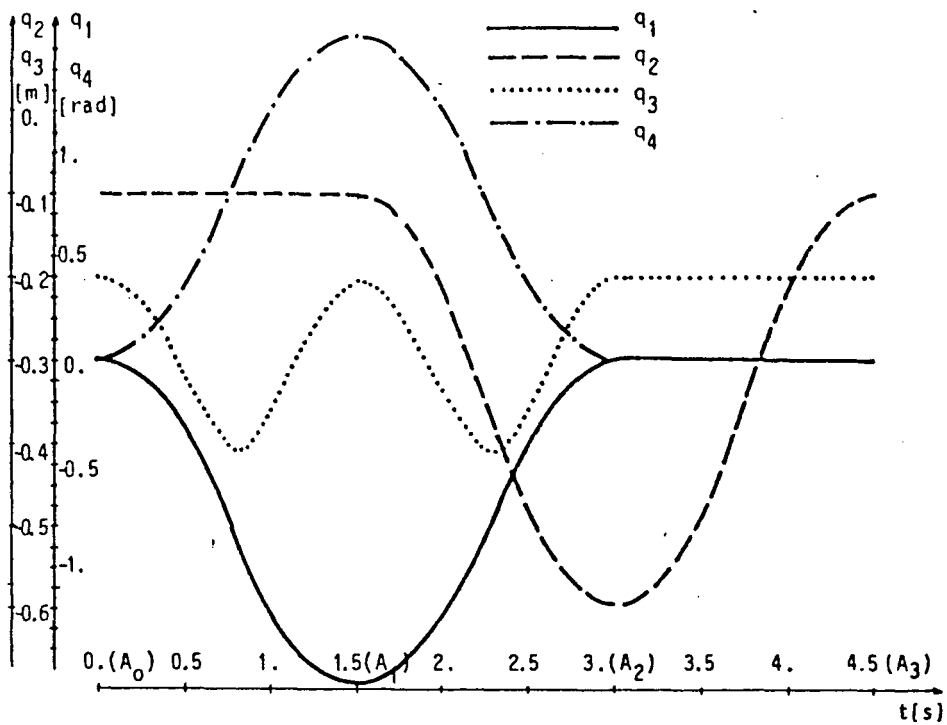


Figura 6.7. Curvas de posição obtidas da literatura [6].

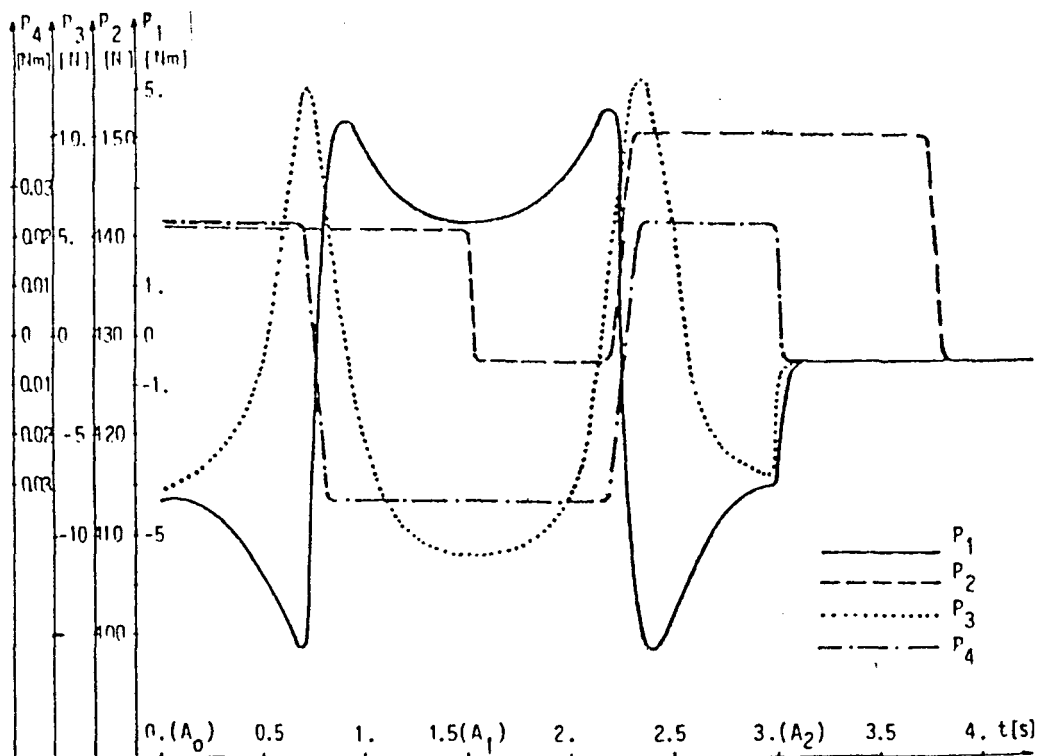


Figura 6.8 Curvas de torque da literatura.

Nas curvas de posição, as variáveis correspondentes às juntas rotacionais 1 e 4 são expressas em radianos, com as outras coordenadas de posição, juntas 2 e 3, em metros. Pela comparação observa-se que estas curvas conferem com as obtidas pelo sistema.

Na figura 6.8, as variáveis P_1, P_2, P_3 e P_4 , referem-se aos torques generalizados nas juntas, sendo expressos conforme a escala apresentada. Estas curvas são equivalentes às obtidas pelo sistema fig.6.6, tal aspecto, confere confiabilidade ao modelo dinâmico implementado.

Embora não exista para comparação o comportamento da energia consumida nesta simulação realizada, apresenta-se a curva referente a esta variável (fig. 6.9), esta mostra que o maior consumo de energia ocorre nos trechos 2 e 3 da simulação, devido a descida e subida da estrutura, provocada pelo acionamento da junta 2, pois, além da aceleração desta junta, a força aplicada sobre ela deve compensar o peso das ligações que a seguem.

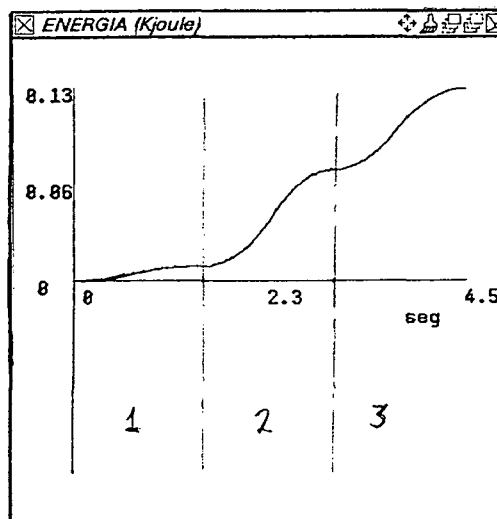


Figura. 6.9. Curva de Energia total consumida.

6.3. SIMULAÇÃO DE UM ROBÔ ARTICULADO.

O modelo de robô escolhido para a segunda simulação, também obtido da referência [6], possui todas as juntas rotacionais e 5 graus de liberdade, a representação da cadeia cinemática é mostrada na figura 6.10.

Este modelo, pela suas características mecânicas (acoplamento entre as juntas), tem um comportamento distinto do anteriormente apresentado, além disto, é projetado para elevadas cargas de trabalho como indicam os valores de massa apresentados posteriormente.

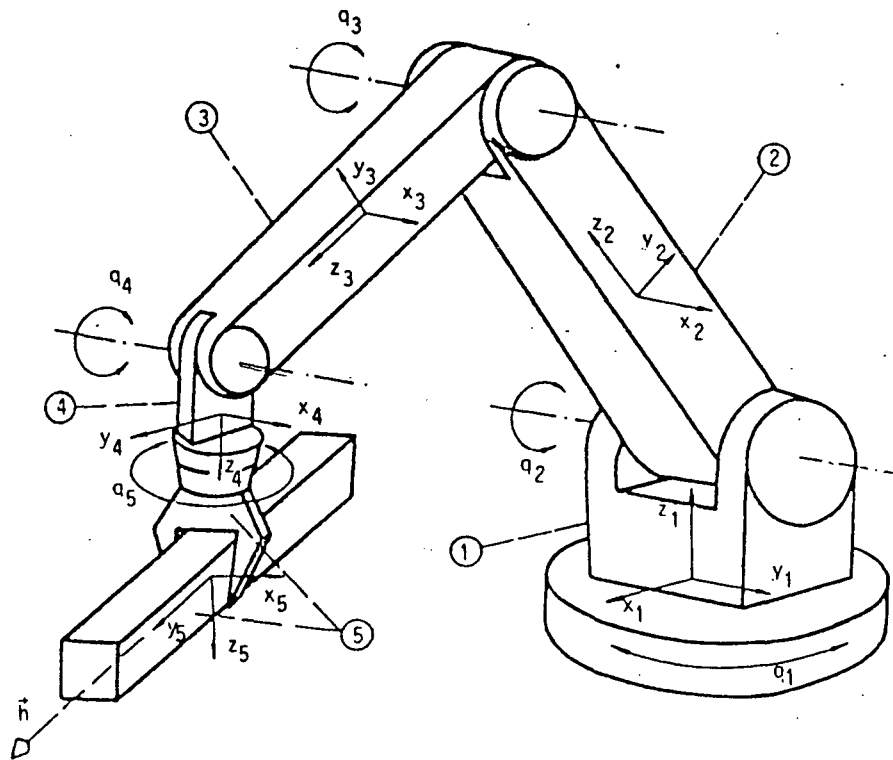


Figura.6.10. Robô articulado [6].

Os parâmetros geométricos da cadeia e os valores de massa e inércia das ligações são apresentados, respectivamente, nas tabelas 6.5 e 6.6. Sendo que na segunda tabela, os valores foram reavaliados para compatibilizar a convenção com os sistemas de coordenadas adotados.

ligação	θ (graus)	α (graus)	a(m)	d(m)
1	θ_1	-90.0	0.0	0.4
2	θ_2	-180.0	1.50	0.0
3	θ_3	0.0	1.50	0.0
4	θ_4	-90.0	0.0	0.0
5	θ_5	0.0	0.0	0.4

Tabela 6.5. Parâmetros de Denavit-Hartenberg.

ligação	1	2	3	4	5
M(kg)	-	125.0	98.0	10.0	270.0
I_x (kg.m ²)	0.0	2.80	2.80	0.05	38.0
I_y (kg.m ²)	15.0	31.0	26.0	0.05	3.0
I_z (kg.m ²)	0.0	31.0	26.0	0.05	38.0

Tabela 6.6 Massas e momentos de inércia das ligações.

A análise do tensor de inércia desta configuração, (eq.6.2), mostra um considerável acoplamento mecânico entre as juntas 2, 3 e 4, porém baixo acoplamento entre as juntas 1 e 5, isto é característica de robôs articulados com juntas paralelas.

$$H = \begin{bmatrix} 843.11 & 0.0 & 0.0 & 0.0 & -38.0 \\ 0.0 & 971.54 & -460.0 & -48.85 & 0.0 \\ 0.0 & -460.0 & 900.27 & 119.0 & 0.0 \\ 0.0 & -48.85 & 119.0 & 48.85 & 0.0 \\ -38.0 & 0.0 & 0.0 & 0.0 & 38.0 \end{bmatrix} \quad (6.2)$$

A simulação, a ser realizada, está representada na fig. 6.11 e, como no exemplo anterior, possui as configurações inicial e final idênticas. Também foi executada com duas configurações intermediárias e o módulo de programação foi o perfil cartesiano.

Diferentemente da simulação anterior, nesta a orientação do efetuador é mantida durante o primeiro trecho da operação, sendo as coordenadas do efetuador na posição inicial: $p=(1.50;0.0;0.0)$, $b=(0.0;0.0;-1.0)$ e $n=(0.0;1.0;0.0)$. Para a primeira configuração intermediária, foram adotados os seguintes valores de posição e orientação: $p=(1.50;1.50;0.0)$, $b=(0.0;0.0;-1.0)$ e $n=(0.0;1.0;0.0)$, com os vetores \underline{b} e \underline{n} idênticos aos da configuração inicial. Sendo que no segundo trecho da operação a orientação não é mantida, afim de atingir a segunda configuração intermediária, cujas coordenadas são: $p=(0.0;2.0;0.8)$, $b=(0.0;0.0;-1.0)$ e $n=(-1.0;0.0;0.0)$.

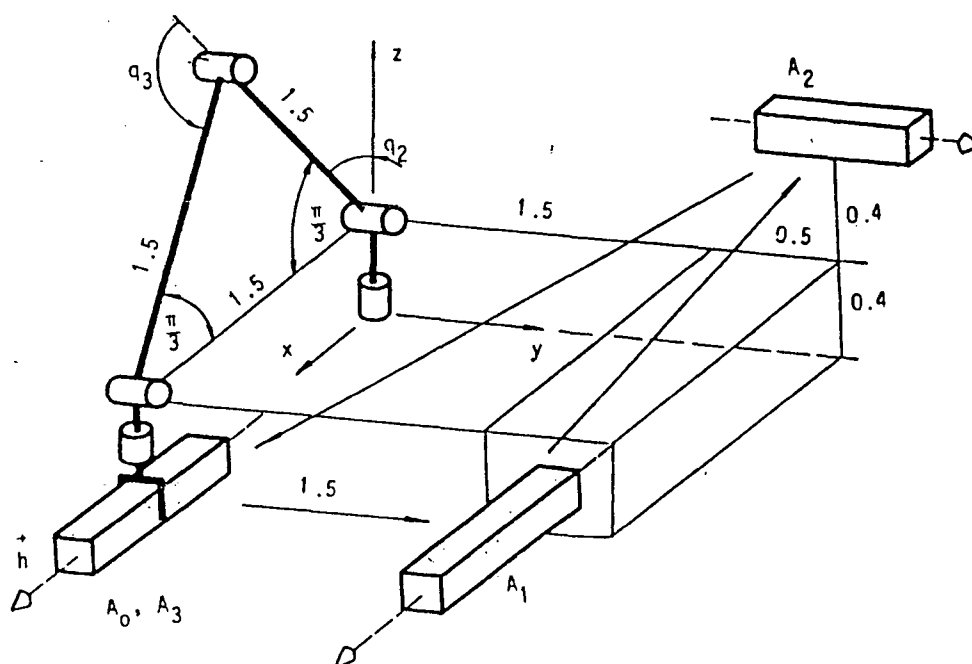


Figura 6.11. Esquema da simulação [6].

A seguir é apresentada na fig. 6.12, a simulação obtida com o sistema, a qual demonstra os pontos percorridos pelo efetuador formando uma linha reta no espaço, entre as posições do efetuador estabelecidas, o que caracteriza a programação no espaço cartesiano. A distribuição dos pontos no espaço é realizada em intervalos de tempo constantes.

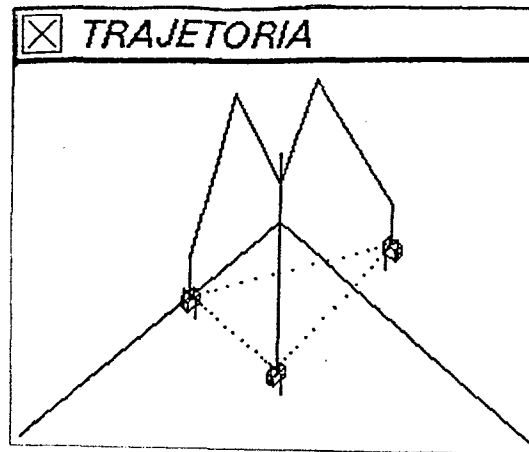


Figura 6.12 Simulação realizada pelo sistema.

As curvas de posição obtidas pelo sistema são apresentadas na fig. 6.13 e as de torque na fig.6.14, para comparação, com os valores obtidos da literatura [6], apresentados na figuras 6.15 e 6.16.

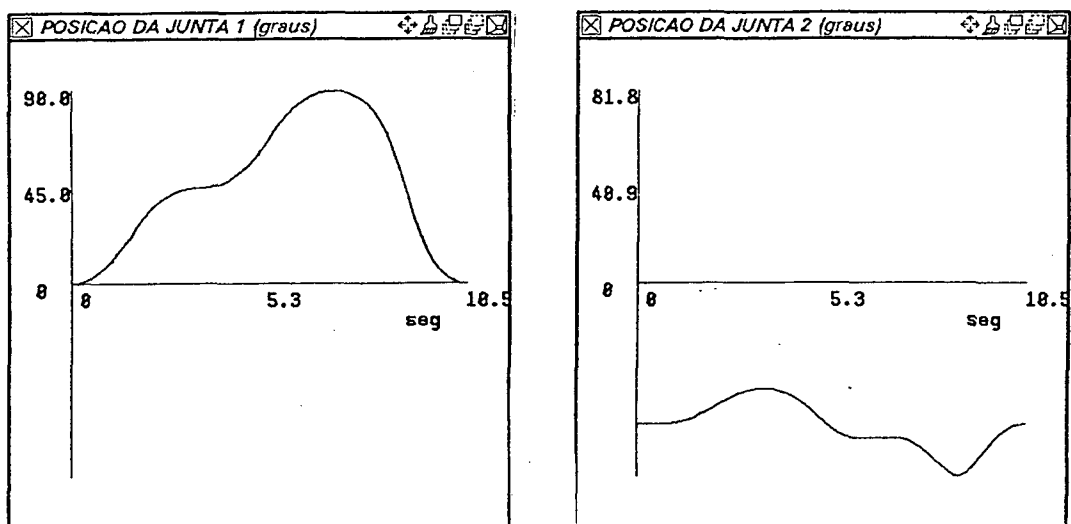


Figura 6.13-a. Curvas de posição das juntas 1 e 2.

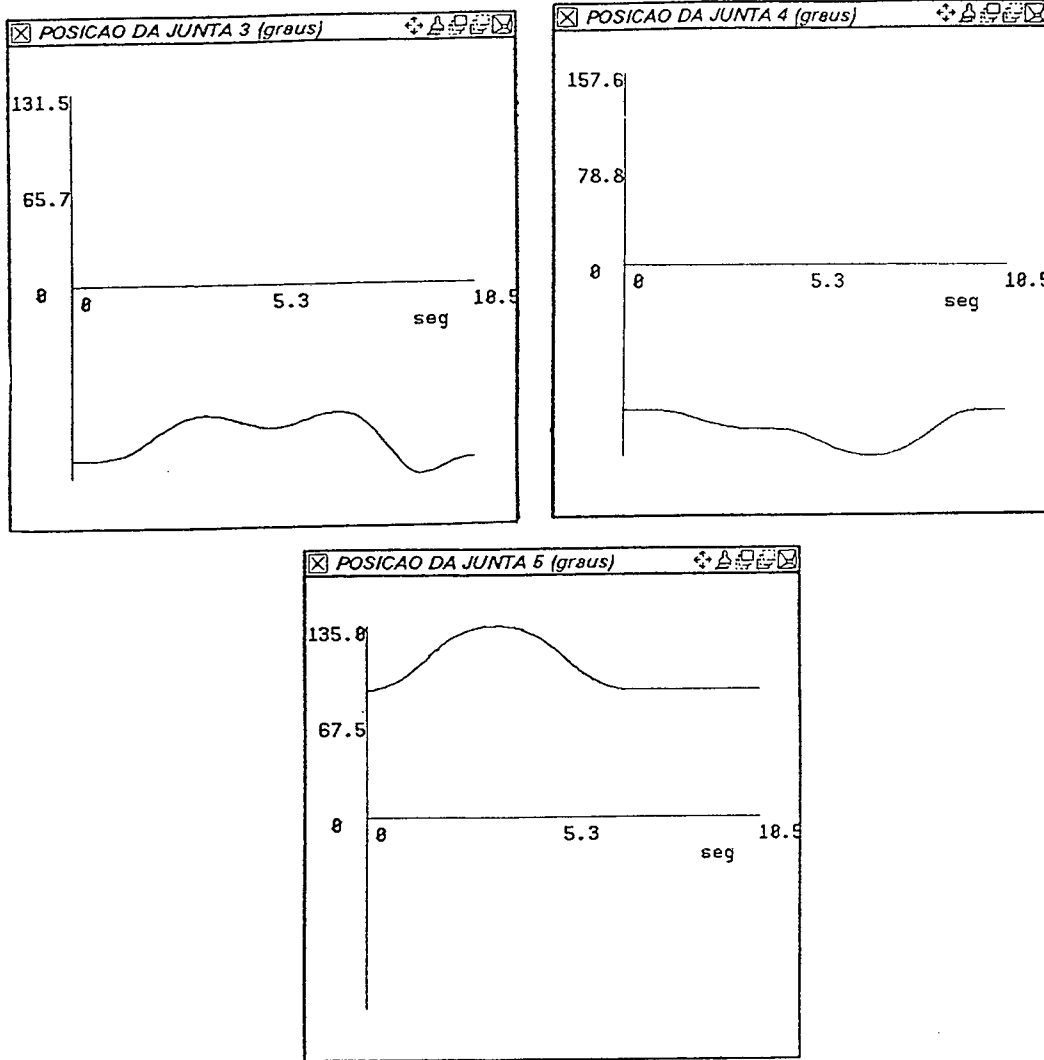


Figura 6.13-b. Curvas de posição das juntas 3, 4 e 5.

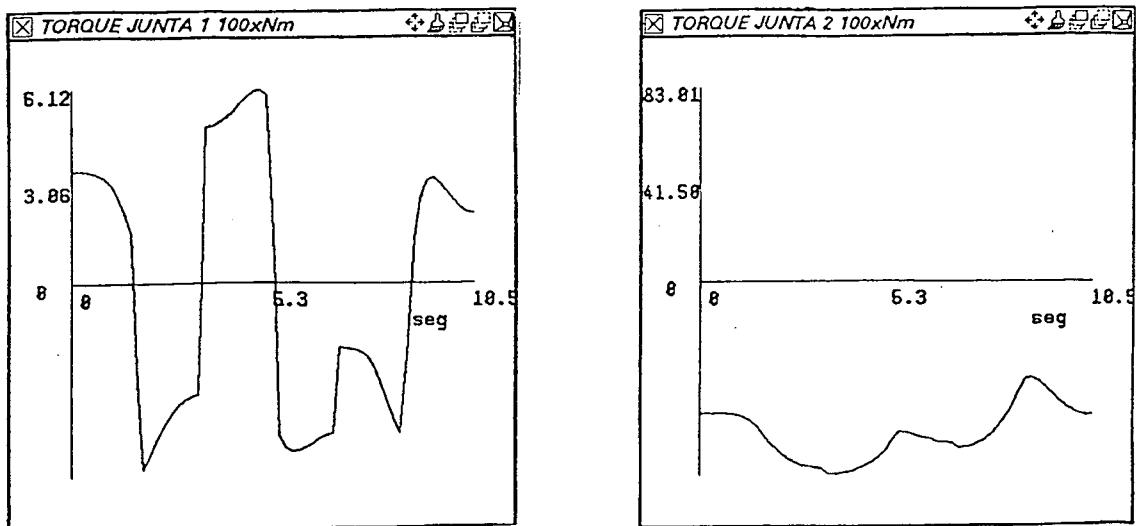


Figura 6.14-a. Curvas de torque das juntas 1 e 2.

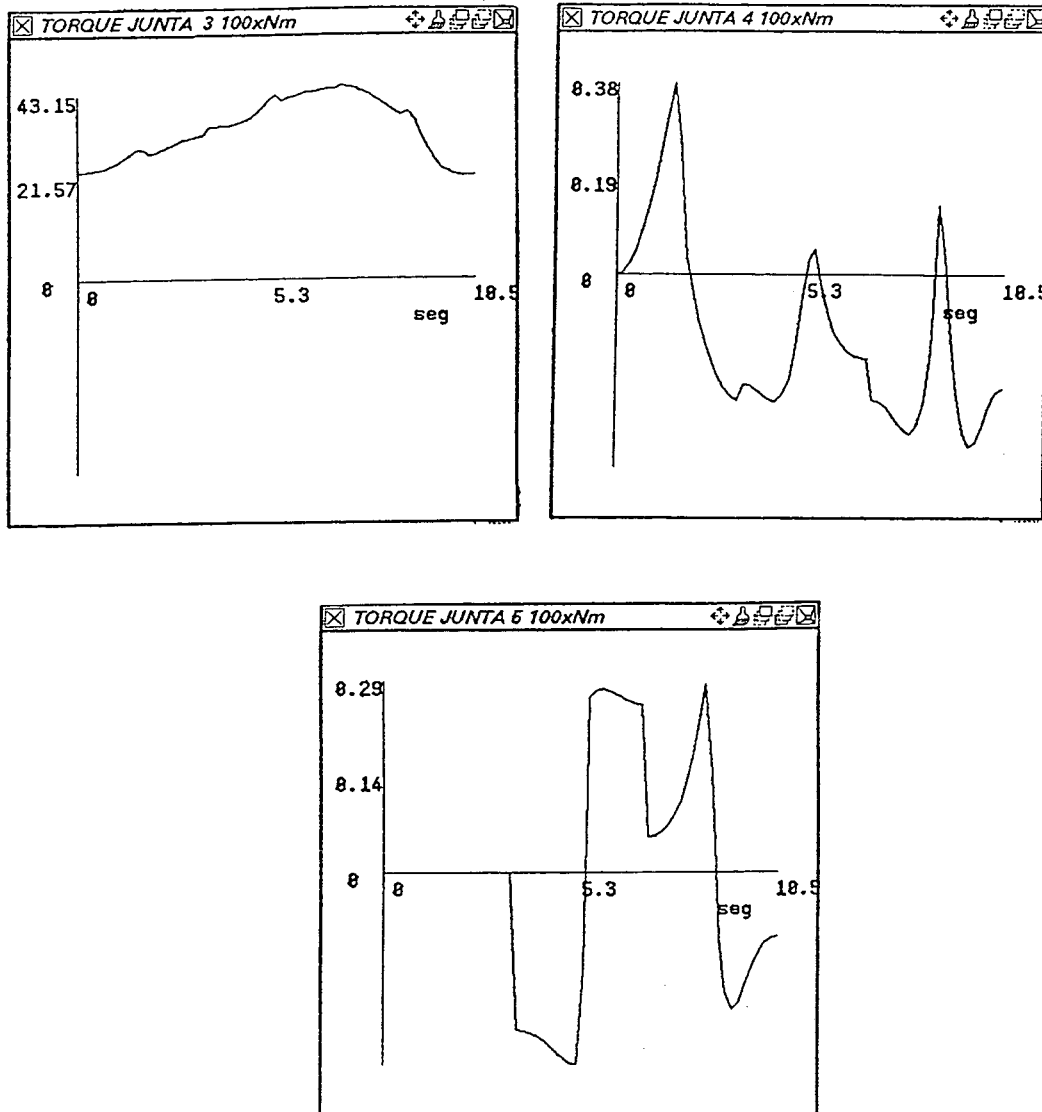


Figura 6.14-b. Curvas de torque das juntas 3, 4 e 5.

Pela análise das curvas de posição fig.6.13, observa-se que as mesmas conferem com as obtidas da literatura, A discrepância em relação à junta 2 é explicada pelo sistema de coordenadas adotado no exemplo, o qual não segue a notação de Denavit-Hartenberg, pois como mostra a fig.6.11, a coordenada q_2 é medida em relação ao eixo z na vertical (o que corresponde a -30° na configuração inicial), contudo de acordo com a convenção mencionada, esta coordenada deve ser medida em relação ao eixo na horizontal (correspondendo a -60° na configuração inicial). Tal fato, origina também a inversão da curva de torque da junta 2, vide fig. 6.15 e 6.16.

Com relação ao torque da junta 5, observa-se que no primeiro trecho, seu valor é nulo, isto se explica pelo fato do acoplamento mecânico existente entre as juntas 1 e 5 (termos H_{15} e H_{51}) ser oposto ao momento de inércia sobre a junta 5 (H_{55}), indicando que no primeiro trecho, a própria inércia da carga sobre o efetuator manteria o mesmo na orientação estabelecida. Deve-se observar que tal análise se processa desconsiderando-se o atrito na junta, o qual no caso real deveria ser compensado.

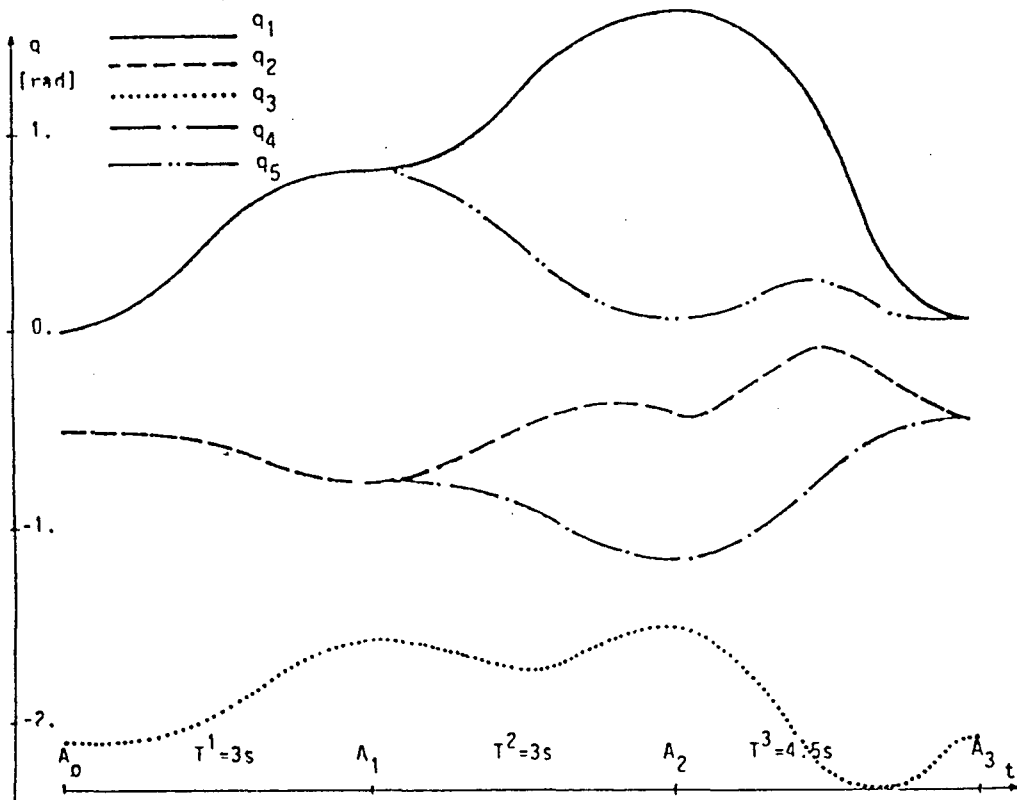


Fig.6.15. Curvas de posição obtidas da literatura [6].

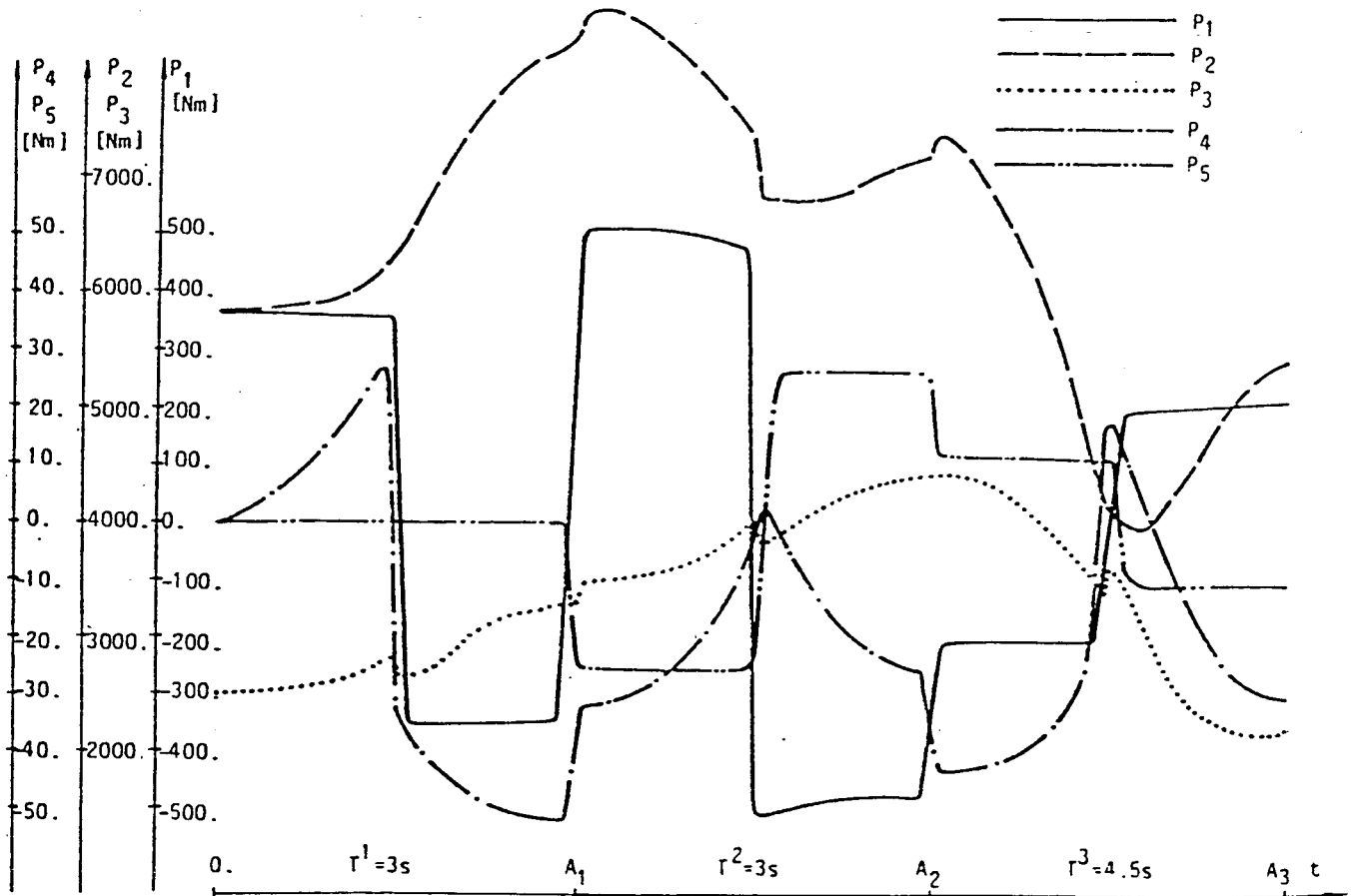


Fig.6.16. Curvas de torque obtidas da literatura [6].

Embora as curvas de torque das juntas 1 e 5 (fig.6.13) não tenham sido obtidas com a mesma forma da comparada, isto é devido ao comportamento das acelerações (destas juntas), que são calculadas por diferenciação numérica, o que acarreta instabilidade nos pontos de mudança de sinal. Tal comportamento não foi apresentado nas demais juntas pelo fato destas terem pequenos deslocamentos ao longo da trajetória, como mostram as curvas da figura 6.13.

Este aspecto, não compromete, significativamente, o modelo dinâmico implementado, pois o valor da energia obtido nesta simulação foi de 15672,126 J, o que corresponde a um desvio de 0.0774% em relação ao da literatura (15660 J), tal aspecto fornece confiabilidade ao sistema. Com a observação das curvas das componentes de torque, apresentadas na fig. 6.17, mostra-se que os efeitos inerciais tem maior influência nas juntas 1 e 5,

enquanto as demais juntas são mais afetadas pelos efeitos gravitacionais, sendo que estes não atuam sobre as juntas 1 e 5, pois estas juntas estão direcionadas na vertical.

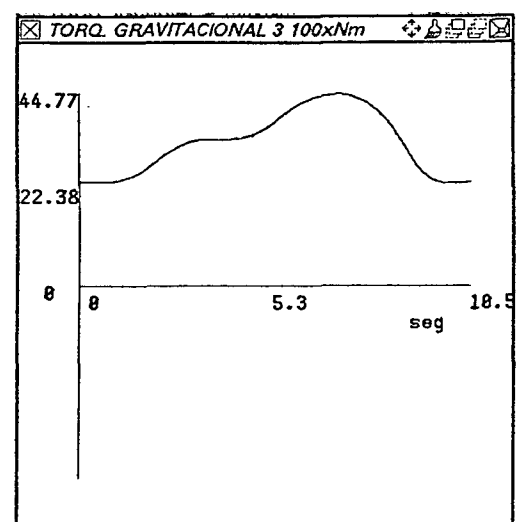
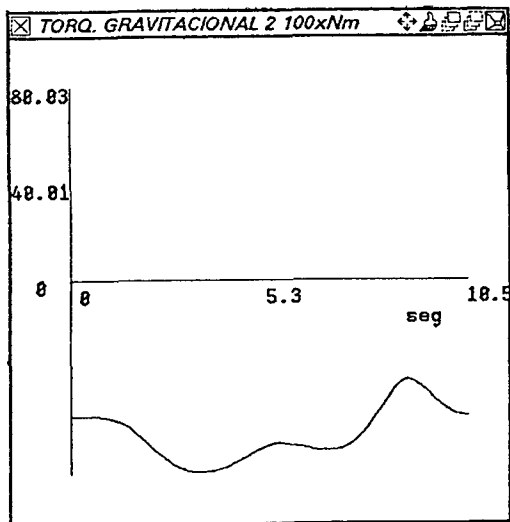
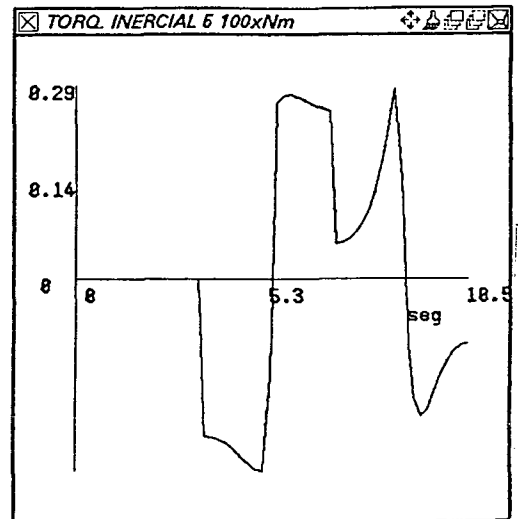
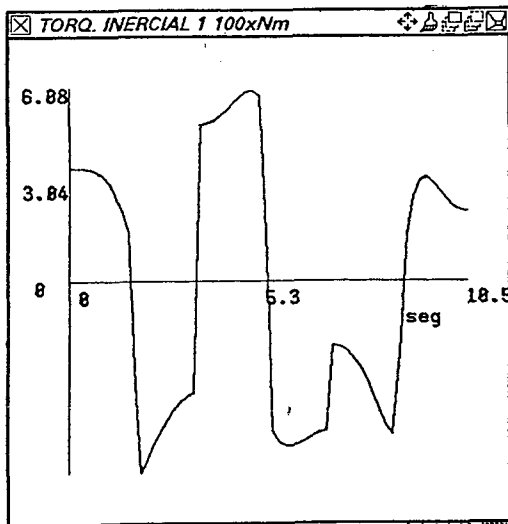


Fig.6.17. Curvas das componentes de torque.

6.4. ANÁLISE DO EFEITO DOS PERFIS DE VELOCIDADE SOBRE O CÁLCULO DOS TORQUES E DA ENERGIA.

Com o objetivo de avaliar o comportamento dos torques e da energia, calculados em função da variação dos perfis de velocidade, foi escolhido o modelo SCARA, com posição inicial conforme fig.6.17 e posição final fig. 6.18.

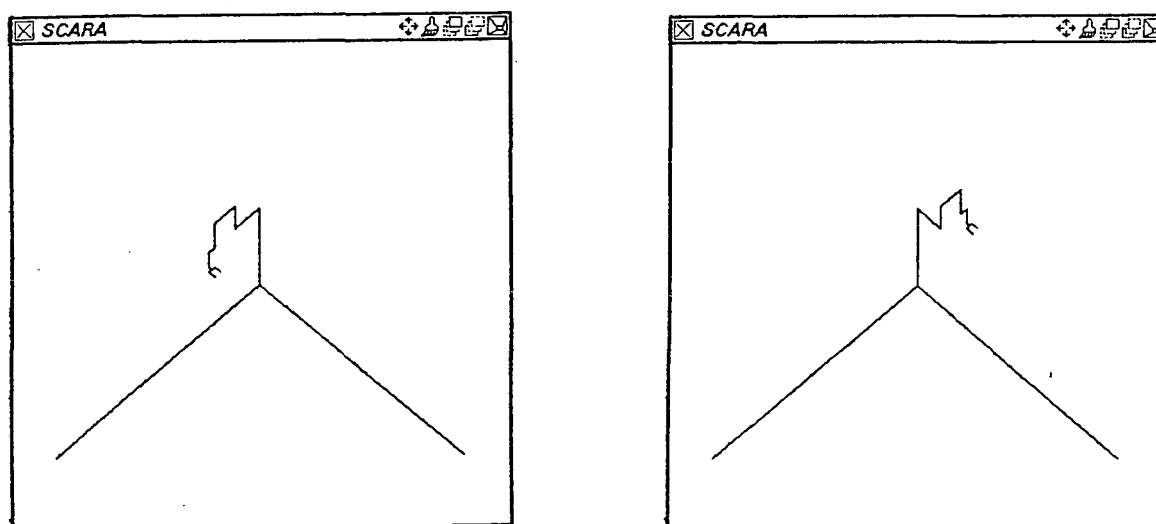


Fig.6.17. Posição inicial. Fig.6.18 Posição final do SCARA.

Como opção de formulação dinâmica adotou-se para as massas das ligações, de 1 a 6, os seguintes valores: 60,40,30,20,10 e 5 kg, e as mesmas foram modeladas como hastes. A trajetória foi programada com o tempo de 6 segundos, considerando o efeito da gravidade e sem cargas externas.

A simulação foi realizada, como mostra a trajetória da fig.6.19, a qual consiste em uma rotação de 90 graus na 1^a e 2^a juntas, e simultaneamente, uma rotação de 180 graus na 4^a junta, mantendo o efetuador na mesma orientação, e um prolongamento da junta 3.

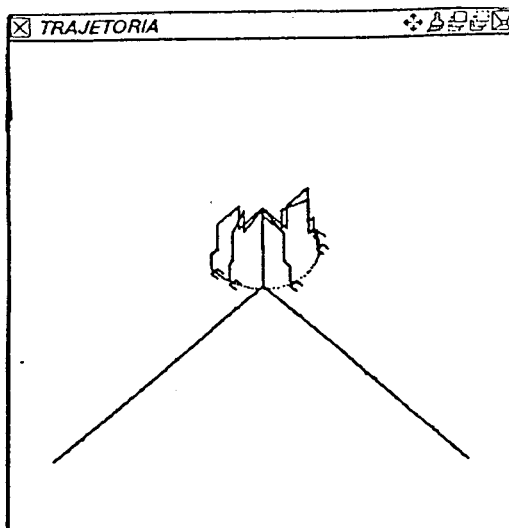


Fig. 6.19. Trajetória realizada pelo robô SCARA.

Inicialmente, para efeito comparativo, foi escolhido o perfil de velocidade triangular, obtendo-se como comportamento de velocidade da junta 1 a fig.6.20, este é igual ao da junta 2, pois ambas sofrem o mesmo deslocamento no mesmo tempo. Para esta simulação teve-se um consumo de energia de 296,79 J.

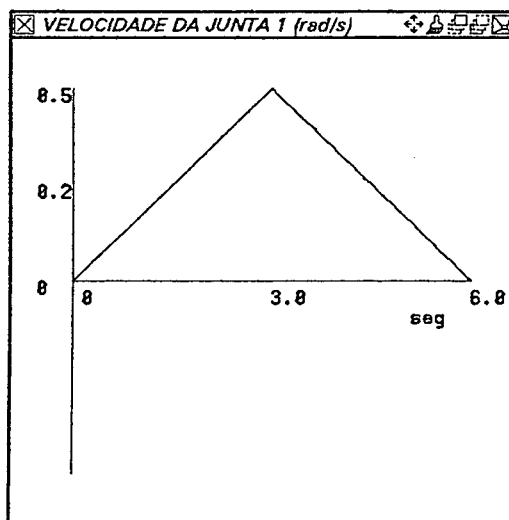


Fig. 6.20. Velocidade da junta 1 com perfil triangular.

Posteriormente, repetiu-se a simulação, considerando os fatores anteriores de posições, massas, tempo e carga, desta vez com o perfil parabólico, apresentado na fig.6.21, obtendo o valor de energia 251,62 J.

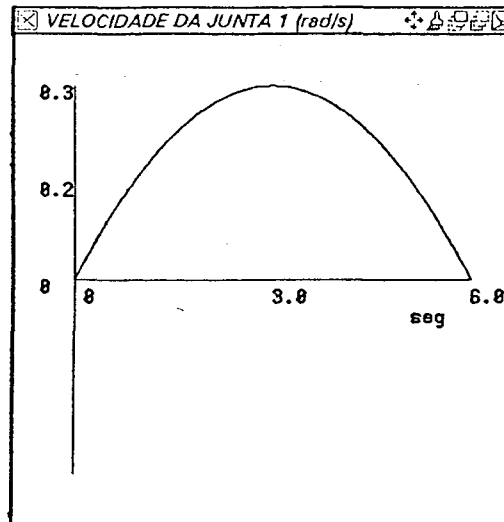


Fig.6.21. Velocidade da junta 1 com perfil parabólico.

Em ambos os casos simulados, os testes dos atuadores foram positivos e foi feita a otimização relativa ao tempo. Na primeira simulação, com perfil triangular, obteve-se o tempo mínimo de 5,0 s com a energia consumida de 370 J, e na segunda, com perfil parabólico, o tempo reduziu-se para 3,8 s, com o valor de energia de 430 J. Considerando-se o tempo de 5 s com perfil parabólico, a energia obtida é de 306 J.

Este estudo comparativo prova que, com a mudança do perfil de velocidade de triangular para parabólico (ou seja, de um degrau de aceleração para uma rampa), o mesmo modelo obtém ganho de energia na ordem de 16 % para o mesmo tempo, ou uma redução no tempo mínimo em torno de 24 %.

O decréscimo dos valores de torque e, conseqüentemente da energia, é devido ao fato de que com a mudança do perfil de velocidade, de triangular para parabólico, tanto as velocidades máximas a serem atingidas como as acelerações diminuem, com isto as componentes de torques inerciais e não-lineares sofrem redução em seus valores. Este aspecto implica em uma menor solicitação dos atuadores, o que diminui também o tempo mínimo conseguido pelo mesmo modelo.

Esta conclusão está de acordo com as pesquisas que mostram os perfis de velocidade ótimos como formas próximas de parábolas deformadas [9], obtidas através de programação dinâmica, que não foi abordada no presente contexto.

6.5. AVALIAÇÃO DAS CARGAS ESTÁTICAS.

Apesar do sistema ter sido desenvolvido visando o modelamento e a análise dinâmica, pode também ser aplicado no estudo do comportamento estático dos robôs, o qual é apenas um caso específico do dinâmico.

Isto compreenderia a avaliação dos efeitos gravitacionais e das cargas externas sobre as juntas de um robô. Tal análise é diferente da avaliação destas componentes em uma simulação dinâmica, já mencionada anteriormente. Tendo em vista que estes efeitos são modificados com a alteração da configuração do robô.

A avaliação dos efeitos estáticos é realizada escolhendo-se as configurações inicial e final da cadeia geométrica idênticas, processando-se o modelamento dinâmico, com momentos de inércia estabelecidos ou apenas com as massas. E na programação de trajetória, escolhendo-se um tempo de simulação qualquer, incluindo as cargas externas se desejado, porém selecionando qualquer opção para o perfil de velocidade exceto "trajetória com configurações intermediárias", pois o robô deve permanecer estático.

Sendo assim serão processados os cálculos de todas as velocidades e acelerações nulas, e os torques obtidos serão correspondentes apenas aos efeitos estáticos mencionados.

Este estudo é de interesse na fase de projeto preliminar de um robô, na qual deseja-se entre outros fatores, ter uma estimativa do seu peso total, bem como da influência do peso das ligações sobre as diversas juntas.

Deve-se observar que tais efeitos podem ser obtidos separadamente, ou seja, calcular a influência do peso de apenas uma ligação sobre as demais juntas do robô, bastando-se para isto, na fase de modelamento dinâmico incluir apenas a massa da ligação que se deseja analisar e as demais nulas.

O sistema desenvolvido satisfaz aos objetivos propostos, sobretudo, com relação à sua aplicação como ferramenta de apoio para o estudo da robótica, sendo este aspecto comprovado no decorrer deste trabalho. Dentre os pontos mais evidenciados apresentam-se:

.O estudo do comportamento dinâmico dos robôs foi possibilitado, através da implementação da formulação de Lagrange, a qual se mostrou apropriada para esta aplicação. Considera-se que o modelo obteve satisfatório desempenho, de acordo com os testes realizados.

.O sistema possui versatilidade com relação aos modos de programação implementados, os quais possibilitam simulações de diferentes robôs.

.Com a simplificação das tarefas, obtida através da interface gráfica, torna-se possível a interação do robô com outros elementos de seu ambiente, bem como a simulação gráfica de uma operação, utilizando-se menor número de instruções.

Estes aspectos conferem ao sistema apresentado uma possibilidade para sua ampliação, sendo assim, seguem algumas recomendações para futuros trabalhos.

.Implementação da formulação de Newton-Euler, visando avaliar os esforços nas ligações no decorrer da trajetória.

.Inclusão de graus de liberdade relativos à flexibilidade das ligações e juntas, afim de simular os efeitos estruturais e possibilitar a avaliação de diferentes materiais de fabricação com adequadas propriedades mecânicas em função das aplicações de cada robô.

.Desenvolvimento de algoritmos de controle, aplicando as técnicas apresentadas no capítulo 3, tendo em vista a importância do controle no estudo dos robôs. Com a sofisticação do modelo matemático dos atuadores e a

inclusão das suas características eletromecânicas.

.Melhoramento da interface gráfica do sistema, utilizando modelos sólidos para as ligações do robô. .Inclusão na função simulação de operação de outros elementos ativos no ambiente de trabalho do robô, e com a aplicação de inferência, limitar as ações do robô em função das mudanças provocadas no decorrer da tarefa, esta aplicação possibilitaria a simulação de diferentes níveis de inteligência, conforme a complexidade do ambiente.

.Implementação do algoritmo de cálculo da capacidade de carga dinâmica, com o intuito de auxiliar no dimensionamento dos atuadores a serem utilizados no robô.

.Melhoramento do cálculo dos coeficientes relativos aos termos não lineares, tendo em vista a importância destes no estudo de robôs com elevadas cargas e/ou com altas velocidades de trabalho, pois nestas condições ocorrem rápidas mudanças nas características de inércia dos robôs, acarretando desvios nas curvas de torque. Neste sentido, recomenda-se a utilização de alocação dinâmica para as variáveis da função simulação, afim de permitir uma melhor discretização do tempo de cada operação.

.Desenvolvimento de uma versão do sistema para microcomputador, o que envolve a redefinição de algumas variáveis e funções, de acordo com a capacidade de memória disponível.

.Utilização de outros métodos de diferenciação numérica, com características de estabilidade incondicional, visando a obtenção de melhores curvas de velocidade e aceleração na programação da trajetória em relação ao espaço cartesiano.

.Implementação do cálculo do espaço de trabalho do robô de acordo com os limites de deslocamentos de cada junta.

[1]- RAO, S. S. & BHATTI, P.K., "Optimization in the design and control of robotic manipulators: a survey. Applied mechanical review, vol 42, ASME, april, 1989.

[2]- ASADA, H. & SLOTINE, J-J. E., Robot analysis and control. John Wiley and Sons, Inc, 1986.

[3]- Groover, M. P., et alli, Industrial robotics: technology, programming and applications, Mc Graw Hill, N.Y.,1986.

[4]- Teicholz, E., CAD/CAM HANDBOOK, Mc Graw Hill, 1985.

[5]- Asfahl, C.R., Robots and manufacturing automation. John Wiley, 1985.

[6]- Vukobratovic, M. and Potkonjak, V., Applied dynamic and CAD of manipulation robots. Scientific fundamental of robotics, vol-6, Springer-Verlag, Berlin, 1985.

[7]- Rivin, E. I., Mechanical design of robots. McGraw Hill,1988.

[8]- Paul, R. P., Robot manipulators:mathematics, programming, and control, MIT Press, Cambridge MA, 1984.

[9]- Vukobratovic, M. and Potkonjak, V., Kinematics and trajectory synthesis of manipulation robots. Scientific fundamental of robotics, vol-3, Springer-Verlag, Berlin, 1985.

[10]- Craig, J. J., Intorduction of robotics: Mechanical and control, Addison- Wesley Publishing Company, 1986.

[11]- Kazerorian, K., "On the numerical inverse kinematics of robotic manipulators." Transactions of the ASME, vol 109, pags 8-13, march 1987.

[12]-Walker, M.W. and Orin, D.E., "Efficient dynamic computer simulation of robotic mechanisms." Journal of dynamic systems, measurement and control, ASME, vol 104, sept.1982. pp. 205-211.

[13]-Hollerbach, J.M.,"A recursive lagrangian formulation of manipulator dynamics and comparative study of dynamic formulation complexity." IEEE SMC-10 no-11,nov. 1980.

[14]- Toumy, K.Y. and Asada, H., "The design of open-loop manipulators arms with decoupled and configuration-invariant inertia tensors." Journal of dynamic systems, measurement and control, ASME, vol 109, pp.269-275, sept.1987.

[15]- Thomas, M. and Tesar, D., "Dynamic modelling of serial manipulators arms." Journal of dynamic systems, measurement and control, ASME, vol 104, pp 218-228, sept. 1982.

[16]-Greenwood, D.T., Principles of dynamics, Prentice-Hall, 1965.

[17]-Luh, J.Y.S., Walker, M.W. and Paul, R.P.C., "On line computational scheme for mechanical manipulators." Journal of dynamic systems, measurement and control, ASME, 1980.

[18]-Fu, K.S., Gonzalez, R.C. and Lee, C.S.G., Robotics: Control, sensing, vision and intelligence, McGraw Hill, N.Y., 1987.

[19]-Li, C.J., "A new method of dynamic for robot manipulators." IEEE SMC, vol 18, no 1, jan. 1988.

[20]-Wang, L.T. and Ravani, B., "Dynamic load carrying capacity of mechanical manipulators- part I: problem formulation." Journal of dynamic systems, measurement and control, ASME, march 1988.

[21]- Stepheson, R.E., Computer simulation for engineers. Hartcourt Brace Jovanovich, Inc, 1971.

[22]-Dohi, T. et alli, "Computer aided surgery system (CAS): Development of surgical simulation and planning system with three dymensional graphic reconstruction." IEEE, 1990.

[23]-Valette, R. e Silva, M., "A rede de petri: uma ferramenta para a automação fabril." Tutoriais do IV CONAI, São Paulo, julho 1990.

[24]-Back, N., Metodologia de projeto de produtos industriais, Guanabara dois, Rio de Janeiro, 1983.

[25]-Tozzi, C.L., Projeto auxiliado por computador, Papirus, Campinas, 1986.

[26]-Takase, C.N., "Aplicações da simulação dinâmica no controle de processos." Tutoriais do IV CONAI, São Paulo, julho de 1990.

[27]-Worn, H. and Stark,G., "Robot applications supported by CAD simulation."Robotics & CIM, vol.3,no 1 ,pag55-62, 1987.

[28]-Marklund,P-O. and Norlin, P.,"Using CAD/CAM for robotics systems."The industrial robot-16th ISIR/ Robotex, oct. 1986,pag30.

[29]- Callan,J.F.,"The simulation and programming of multiple robot systems".Robotics engineering, april 1986.

[30]-Haffenden,A.,"Robotics simulation in practice."The industrial robot, december 1985.

[31]-Four steps to simulating robotic workcells. Robotics engineering. aug/86.

[32]-Craig,J.J.,"Anatomy of an off-line programming system.".Robotics today, febr.1985. [33]-Ránky, P.G. and Ho,C.Y., Robot modelling control and applications with software. Springer-Verlag, Berlin,1985.

[34]-Sata,T. et alli,"An approach to robot-based systems with software applications."Computers in industry,7, pags 211-225,1986.

[35]-Schidt,H., Inteligência artificial utilizando linguagem C, Mc Graw Hill, São Paulo, 1989.

[36]-Dillman,R."Artificial intelligence and robotics." Tutoriais do IV CONAI, São Paulo, julho 1990.

[37]-Cook, R.D.,Concepts and applications of finite element analysis. John Wiley & Sons, 1981.

[38]-Leu,M.C. "Robotics software systems", Robotics & CIM, vol2, no. 1, pp 1-12 ,1985.

Manual do usuario

**SIMULADOR PARA ANÁLISE E MODELAMENTO
DINÂMICO DE ROBÔS INDUSTRIAIS**

MANUAL DO USUÁRIO

ÍNDICE

Prefácio.....	A-02
1- Introdução.....	A-03
1.1- Robótica.....	A-03
1.2- Simulador para Análise e Modelamento Dinâmico de Robôs Industriais.....	A-04
2- Como usar o simulador.....	A-05
2.1- Entrando no sistema.....	A-05
2.2- Função ARQUIVO DE CONFIGURAÇÕES.....	A-06
2.2.1- Alteração de parâmetros.....	A-08
2.2.2- Entrada com coordenadas cartesianas.....	A-09
2.3- Função FORMULAÇÃO DINÂMICA.....	A-11
2.3.1- Tensor de inércia das ligações.....	A-12
2.3.2- Matriz Jacobiana.....	A-13
2.3.3- Tensor de inércia do manipulador.....	A-13
2.4- Função PROGRAMAÇÃO DA TRAJETÓRIA.....	A-13
2.5- Função PARÂMETROS DE SAÍDA.....	A-17
2.6- Função OTIMIZAÇÃO DE TRAJETÓRIA.....	A-18
2.7- Função SIMULAÇÃO DE OPERAÇÃO.....	A-19
2.8- Função AUXÍLIO AO USUÁRIO.....	A-22
3- EXEMPLO DE APLICAÇÃO.....	A-23

PREFÁCIO

Este manual foi desenvolvido para acompanhar o Simulador para Análise e Modelamento Dinâmico de Robôs Industriais.

O manual descreve detalhadamente todas as funções do sistema, e traz um exemplo de aplicação do simulador.

Tem como objetivo permitir que o usuário compreenda com facilidade todo o sistema e tenha uma interação rápida com o simulador.

1-INTRODUÇÃO

1.1 Robótica

Como área de estudo, considera-se a robótica como sendo formada a partir de duas áreas existentes, a de tecnologia de comando numérico e a de manipulação remota. Contudo a robótica envolve questões de projeto e controle substancialmente distintas destas duas, devido sobretudo às características de mobilidade, destreza, acoplamentos envolvidos.

Em virtude do amplo aspecto de conhecimentos envolvidos com a robótica, esta é apresentada como uma interface de outras áreas como as engenharias mecânica, elétrica e industrial e a computação, fato este que torna a robótica uma área interdisciplinar de amplas fronteiras para pesquisa e desenvolvimento.

A definição de robô, adotada neste manual, é a dada pela RIA (Robot Institute of America), onde um robô é um manipulador reprogramável e multifuncional projetado para mover materiais, partes, ferramentas ou dispositivos especiais através de movimentos predefinidos para desenvolver uma variedade de tarefas. Esta definição compreende três classes de robôs:

- Robôs repetidores;
- Robôs de controle numérico;
- Robôs inteligentes.

Cinematicamente considera-se um robô como sendo composto de ligações interconectadas por juntas, formando uma cadeia cinemática aberta.

A crescente busca da industria visando maior produtividade, menores custos de produção, melhor qualidade e substituição do operador em atividades perigosas, tem favorecido a aplicação de robôs de manufatura em geral, especialmente nas industrias automotivas e de componentes elétricos. Atualmente os robôs estão sendo utilizados nas mais variadas tarefas, tanto de média como de alta habilidade, a saber: operações de soldagem, pintura, acabamento, montagem, manipulação de materiais, carga de máquinas e usinagem.

1.2 Simulador para Análise e Modelamento Dinâmico de Robôs Industriais.

Devido ao número de tipos de robôs existentes no mercado a seleção da melhor configuração para uma dada tarefa torna-se consideravelmente complexa. Do ponto de vista de desempenho, questões como, capacidade de carga, consumo de energia, repetibilidade, precisão e espaço de trabalho, devem ser avaliadas. Esse processo é praticamente inviável caso não seja utilizado o computador como ferramenta na análise, devido às características específicas de cada configuração e a tecnologia envolvida no projeto destes equipamentos.

Entre os aspectos a serem avaliados, o desempenho dinâmico dos robôs mostra-se de grande importância, sobretudo referindo-se a operações em altas velocidades e/ou com altas cargas, pois nestas condições as propriedades de massa e inércia são relevantes.

Frente a dificuldade de análise dinâmica dos robôs, criou-se o Simulador para Análise e Modelamento Dinâmico de Robôs Industriais.

O simulador foi desenvolvido em linguagem C, sob ambiente Unix, utilizando a estação de trabalho Intergraf, modelo Interpro 32 CG, do laboratório de CAE/CAD, no departamento de Engenharia Mecânica da UFSC.

Tem como principais objetivos: prover uma ferramenta de apoio para estudantes de robótica, enfatizando os vários aspectos da utilização de robôs, tais como, projeto das configurações, níveis de linguagem de programação, integração com o ambiente, limites de carga dos atuadores, auxiliar na análise da adequabilidade de configurações para tarefas estabelecidas, seleção dos atuadores e planejamento de trajetória.

O simulador consiste em um programa que permite analisar as configurações de robôs industriais mais utilizadas, considerando os acoplamentos entre as ligações, velocidade, aceleração, o torque de cada junta durante o movimento do manipulador, a influência das cargas externas, as curvas obtidas para a seleção dos atuadores e o cálculo da energia consumida durante a operação.

O programa possui uma interface gráfica com o usuário, permitindo a observação em perspectiva isométrica da cadeia de ligações do manipulador durante toda a trajetória. Possibilita também a simulação de uma manipulação na qual o robô interage com o ambiente de objetos, sendo as instruções fornecidas de forma textual de acordo com cada tarefa. Permite, ainda, a análise das curvas mencionadas para cada operação executada.

2-COMO USAR O SIMULADOR

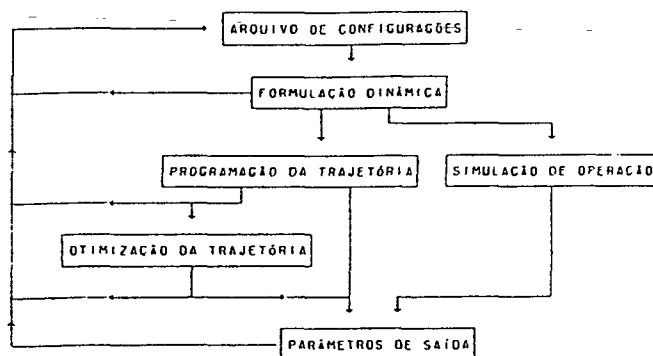
2.1 Entrando no sistema.

Inicializa-se o sistema digitando o comando MDR <enter>, em seguida é apresentado o "menu principal" que mostra as principais funções do simulador. (fig 2.1.1)

```
*****
*  MODELAMENTO DINAMICO DE ROBOS  *
*      EN DESENVOLVIMENTO        *
*  ***GRANTE-ENC-UFSC***         *
*****
*MODELAMENTO DINAMICO DE ROBOS*
1_AQUIVO DE CONFIGURACOES
2_FORMULACAO DINAMICA
3_PROGRAMACAO DA TRAJETORIA
4_PARAMETROS DE SAIDA
5_OTIMIZACAO DE TRAJETORIA
6_SIMULACAO DE OPERACAO
7_AUXILIO AO USUARIO
8_FIN
Seleccione opcao:█
```

fig 2.1.1 - Menu principal do sistema.

É permitido que qualquer sequência de funções seja realizada, mas apenas algumas tem sentido serem feitas. O fluxograma abaixo representa as possibilidades lógicas de operações.



Possibilidades de sequência de operações.

Apesar de todas as possibilidades mostradas serem possíveis, quando se entra no sistema pela primeira vez deve-se optar pelo arquivo de configurações, em seguida realizar a formulação dinâmica, podendo só então optar pela programação da trajetória ou pela simulação de operação.

As funções "Auxílio ao Usuário" e "Fim" podem ser escolhidas a qualquer momento, desde que se esteja no Menu Principal. A seguir serão detalhadas todas as funções.

2.2 Função ARQUIVO DE CONFIGURAÇÕES

O simulador aplica-se a diferentes configurações, assim como primeiro passo deve-se digitar "1"<enter>, função ARQUIVO DE CONFIGURAÇÕES. Esta apresenta um novo menu (fig 2.2.1), que mostra as configurações arquivadas. Uma delas deve ser escolhida.

```

***ARQUIVO DE CONFIGURACOES ***
Configuracoes arquivadas.
1_ESFERICO.
2_ARTICULADO.
3_CILINDRICO.
4_SCARRA.
5_CARTESIANO.
  
```

fig. 2.2.1 - Menu da função ARQUIVO DE CONFIGURAÇÕES.

Realizada a escolha do tipo de configuração e teclado <enter> é apresentada uma tabela contendo os parâmetros de DENAVIT & HARTEMBERG da

configuração escolhida e os índices de cada junta. (fig. 2.2.2)

Tipo de configuração: ARTICULADO					
Numero de juntas=6					
CONFIGURACAO INICIAL					
NJUNT	TETA	ALFA	A	D	JUNTA
	graus	graus	mts	mts	tipo
1	98.88	-98.88	8.88	1.88	rot
2	8.88	8.88	1.88	8.15	rot
3	98.88	98.88	-8.88	8.88	rot
4	8.88	-98.88	8.88	1.88	rot
5	8.88	98.88	8.88	8.88	rot
6	8.88	8.88	8.88	8.88	rot

fig. 2.2.2 - Tabela contendo os parâmetros de DENAVIT & HARTEMBERG.

Os parâmetros de DENAVIT & HARTEMBERG são valores que representam as características cinemáticas de cada junta e de cada ligação do manipulador. (fig 2.2.3)

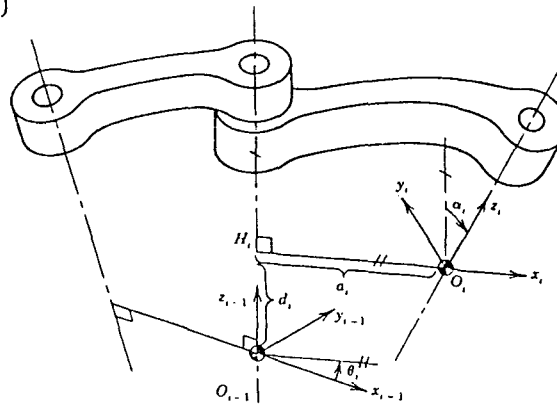


fig. 2.2.3 - Parâmetros de DENAVIT & HARTEMBERG.

Parâmetros de ligação (constantes):

a_i - distância entre os eixos de duas ligações consecutivas (Z_i e Z_{i-1}). É medida segundo uma linha que é perpendicular, mutuamente, aos dois eixos, e é chamada de comprimento da ligação;

α_i - Ângulo de torção da ligação. É o ângulo existente entre os eixos de duas juntas consecutivas. É medido do eixo " x_i " ao eixo " x_{i-1} ", num plano perpendicular ao segmento " a_i ", positivamente, segundo a regra da mão direita.

Parâmetros de junta.

θ_i - É o ângulo formado entre os segmentos " X_i " e " X_{i-1} ". É chamado de ângulo da junta, e é medido positivamente segundo a regra da mão direita,

sobre o eixo z_{i-1} .

d_i - É a distância entre duas ligações em sequência, medida sobre o eixo z_{i-1} .

Os parâmetros de junta são variáveis, se a junta for rotacional " θ_i " varia, se for prismática " d_i " varia.

A figura 2.2.4 apresenta um robô Stanford com a representação dos sistemas de coordenadas locais e os valores dos parâmetros de Denavit & Hartenberg para aquela configuração.

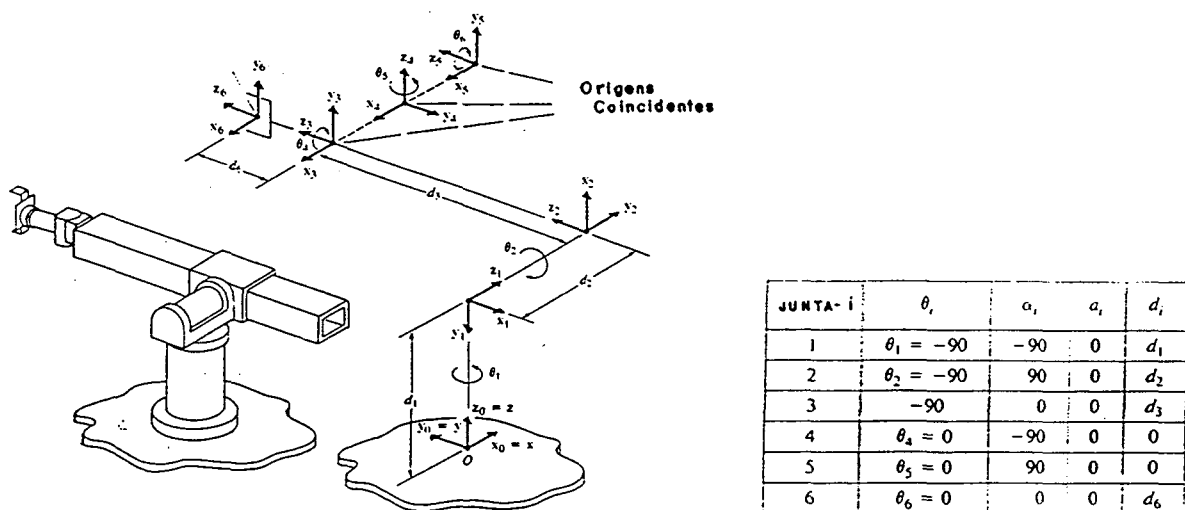


fig. 2.2.4.- Robô Stanford, tabela de parâmetros.

2.2.1 Alteração de parâmetros.

Em seguida o sistema pergunta se o usuário deseja rever algum parâmetro.

Todos os parâmetros podem ser alterados exceto o tipo de junta que permanece como a configuração inicial. Desejando-se alterar algum parâmetro deve-se digitar "s" <enter> caso contrário "n" <enter>.

Ao digitar-se "s" o sistema mostra na tela a configuração inicial e pergunta qual junta que será modificada. Após selecionar-se a junta, deve-se escolher qual o parâmetro que será alterado selecionando uma das opções que aparecem na tela. Seleciona-se o parâmetro e em seguida deve-se

digitar o seu novo valor. Após isto é apresentada na tela a configuração já modificada. Todo processo pode ser repetido digitando novamente "s" <enter>.

Caso a opção seguinte (Entrada com coordenadas cartesianas) não for escolhida, se estará determinando nesta etapa a posição inicial do efetuator.

Para determinar a posição final através de parâmetros de junta, será solicitado ao usuário, em uma etapa posterior, que forneça o valor dos parâmetros variáveis ("teta" ou "d") conforme o tipo de junta.

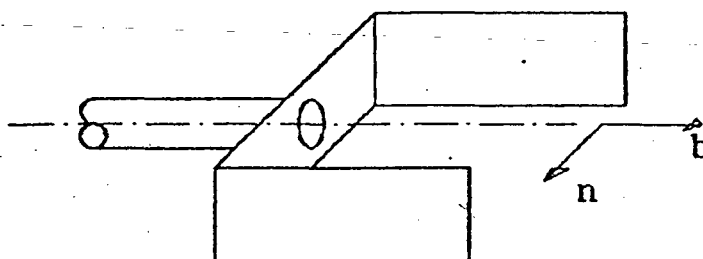
Vale lembrar que os parâmetros variáveis de junta estão sujeitos a alteração quando opta-se por entrar com a posição inicial através de coordenadas cartesianas.

2.2.2 Entrada com coordenadas cartesianas.

Terminado o processo de alteração dos parâmetros, o sistema pergunta se o usuário deseja entrar com as coordenadas cartesianas do efetuator (Deseja entrar com as coordenadas cartesianas do efetuator ? (s/n)). Para entrar com as coordenadas basta digitar "s" e teclar <enter>.

Caso a opção de entrada com coordenadas cartesianas for escolhida, deve-se inicialmente determinar os parâmetros de posição do efetuator no espaço (coordenadas X1, X2 e X3). Após serem fornecidos os valores de posição do efetuator, deve-se fornecer o direcionamento do efetuator.

As coordenadas de direcionamento do efetuator irão dar origem a um vetor que indica a direção em que a garra está apontada, são os parâmetros b1, b2 e b3. Especificada a direção do efetuator, em seguida serão pedidos os parâmetros n1, n2 e n3, que são as coordenadas do vetor que indica como está orientado o efetuator no espaço (fig. 2.2.5).



EFETUADOR

b - vetor de direcionamento

n - vetor de orientação

fig 2.2.5 - Efetuador, vetores direção e orientação.

Após cada coordenada fornecida deve-se teclar <enter> para que a próxima pergunta apareça no vídeo.

Estando completa a entrada das coordenadas cartesianas, em seguida aparecerá a tabela contendo os parâmetros de DENAVIT & HARTEMBERG com a configuração já alterada, a posição, direcionamento e orientação do efetuador, esta será a configuração inicial do manipulador.

Teclando novamente <enter>, será perguntado se o usuário deseja saída gráfica. Digitando "s" <enter> aparecerá uma janela mostrando a configuração inicial do robô em 3 dimensões.

Teclando <enter> será solicitada a configuração final. Se a opção de entrar com coordenadas cartesianas foi feita no início, a configuração final também será determinada em coordenadas cartesianas. Caso contrário deve-se fornecer um novo "d" (para junta translacional) ou um novo teta (no caso de junta rotacional), para cada junta. Deve-se teclar <enter> depois de cada valor fornecido.

Fornecendo as posições em parâmetros de junta, o sistema determina as posições em coordenadas cartesianas e vice-versa, desta maneira ao final da função as posições estão armazenadas em ambos sistemas de coordenadas.

Caso a saída gráfica tenha sido escolhida anteriormente será

apresentada na tela a configuração final a ser considerada para a simulação.

Processada a escolha da configuração e terminada as alterações e entrada de dados, a função mostra a configuração do robô. Também são mostradas a configuração final e inicial simultaneamente.

Ao final da função deve-se digitar qualquer caracter e teclar <enter>. Volta-se então para o menu principal.

2.3 Função FORMULAÇÃO DINÂMICA

Esta função fornece as matrizes de inércia de cada ligação em coordenadas locais ou globais, as matrizes jacobianas do modelo escolhido e o tensor de inércia do robô na **posição inicial**. Antes de iniciar a FORMULAÇÃO DINÂMICA, a função ARQUIVO DE CONFIGURAÇÕES deve estar concluída.

Para entrar na função FORMULAÇÃO DINÂMICA deve-se estar no menu principal, para então selecionar a opção "2" e teclar <enter>.

A primeira tela traz uma pequena descrição da função, em seguida são pedidas as massas de cada ligação. Após entrar com a massa de cada ligação teclar <enter>.

É possível realizar uma simulação dinâmica considerando apenas a massa de uma única ligação. Para tanto basta entrar com um valor nulo (0) como massa das ligações que devem ser desprezadas. Desta forma é possível analisar o efeito de uma única ligação sobre todas as outras.

O sistema permite entrar com os momentos de inércia locais de cada ligação, bastando para isso responder positivamente à pergunta apresentada na tela. Em seguida deve-se fornecer os valores.

Os momentos de inércia locais de cada ligação são determinados em relação ao seu sistema de coordenadas local correspondente. A figura 2.3.1 mostra um robô cilíndrico e a tabela de momentos de inércia locais de cada ligação.

Caso o usuário não possua os valores dos momentos de inércia, o sistema calcula os momentos de inércia de cada ligação como combinações de hastes.

Os momentos de inércia locais da ligação irão compor a matriz de inércia em coordenadas locais desta ligação.

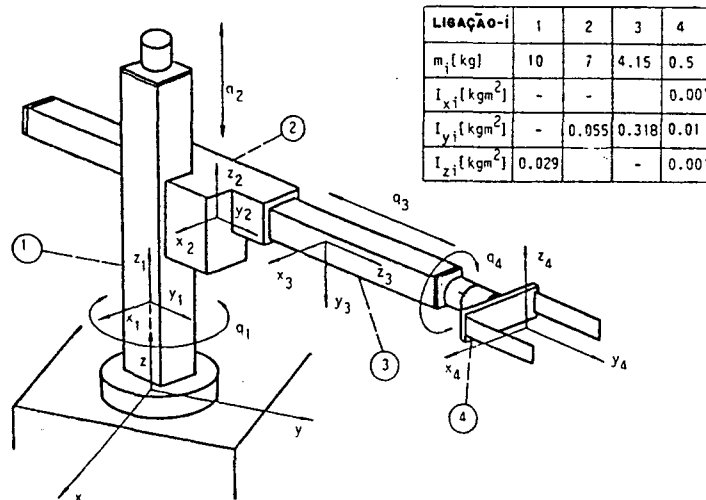


fig 2.3.1 - Robô cilíndrico e os momentos de inércia locais de suas ligações.

2.3.1 Tensor de inércia das ligações.

Depois de entrar com as massas das ligações tem-se a opção de ver as matrizes de inércia das ligações em coordenadas locais. Digitando "s" e teclando <enter> serão apresentados no vídeo as matrizes de inércia de cada ligação em coordenadas locais, podendo assim serem observados os valores dos momentos de inércia locais que compoem cada matriz.

Teclando <enter> após a apresentação dos tensores de inércia em coordenadas locais, possibilita-se a observação do tensor de inércia das ligações em coordenadas globais. Em caso afirmativo ("s" <enter>), serão mostrados na tela os tensores de inércia de cada ligação em coordenadas globais.

2.3.2 Matriz JACOBIANA.

Após terem sido mostrados na tela os tensores de inércia, teclando <enter> o sistema fornece a opção de apresentação da matriz JACOBIANO. Desejando analisar a matriz Jacobiano, deve-se responder afirmativamente à pergunta digitando "s" e teclando <enter>, a matriz será então apresentada.

2.3.3 Matriz de inércia do manipulador.

Logo após a apresentação da matriz Jacobiano tem-se a opção de observar-se na tela a matriz de inércia do manipulador na configuração inicial, para isto deve-se responder à pergunta apresentada digitando "s" e teclando <enter>. Em seguida aparecerá na tela a matriz de inércia do manipulador, nesta é possível observar os acoplamentos entre as ligações.

Como última opção desta função tem-se a possibilidade de verificar o valor correspondente ao torque gravitacional. Bastando responder sim à pergunta digitando "s". Teclar <enter> e em seguida selecionar a junta.

Pode-se verificar a componente de outra junta respondendo novamente positivamente a pergunta e teclando <enter>.

Para voltar ao menu principal ao final da função deve-se digitar qualquer caracter e teclar <enter>.

2.4 Função PROGRAMAÇÃO DA TRAJETÓRIA

Entra-se nesta função, a partir do menu principal, selecionando a opção "3". Antes de iniciar esta função, a FORMULAÇÃO DINÂMICA deve estar concluída.

Os valores limites de velocidade, torque, redução e eficiência dos atuadores são inicialmente apresentados em forma de tabela. O sistema pergunta se o usuário deseja modificar algum destes valores. Caso a resposta

seja "s"(sim), o sistema solicita que seja informada em seguida qual a junta que será alterada. Novo menu é apresentado deve-se determinar que valor deverá ser modificado:

- 1-Velocidade;
- 2-Torque;
- 3-Redução;
- 4-Eficiência.

Selecionada uma das opções, tecla-se <enter>, digita-se o novo valor e tecla-se novamente <enter>. Em seguida o sistema pergunta se o usuário deseja nova modificação.

Após a apresentação das características acima citadas deve-se determinar o tempo de percurso. O valor de tempo desejado deve ser fornecido depois do comando apresentado na tela, teclando <enter> em seguida. Este valor refere-se ao tempo que o manipulador levará para se deslocar de sua configuração inicial até a final.

A simulação acontecerá em um número de 60 etapas, cada uma delas corresponderá a um intervalo de tempo que dependerá do tempo de percurso determinado (tempo de percurso/60).

A pergunta seguinte trata da opção de simular-se com o efeito da gravidade, em caso afirmativo, este efeito é implementado na simulação.

Pode-se optar também por simular com carregamento externo. Desejando este efeito na simulação, é necessário determinar o valor das forças externas e o valor dos momentos externos. Inicialmente deve-se fornecer os valores das forças externas, primeiro na direção "X" (Fext[1]), em seguida na direção "Y" (Fext[2]), e por fim na direção "Z" (Fext[3]). Os valores dos momentos também devem ser fornecidos nas três direções (Mext[1], Mext[2],Mext[3]). Estes vetores atuarão no efetuador do robô durante a simulação e permanecerão constantes.

Terminado o procedimento de determinação do carregamento externo determina-se o perfil de velocidade em que a simulação irá acontecer.

As possibilidades são apresentadas na tela:

- 1- Perfil de velocidade triangular;
- 2- Perfil de velocidade parabólico;
- 3- Perfil de velocidade trapezoidal;
- 4- Trajetória com configurações intermediárias;
- 5- Trajetória cartesiana,

seleciona-se uma das opções e tecla-se <enter>.

Ao selecionar a opção "1", "2" ou "3" todas as juntas terão o mesmo tipo de perfil para aquela trajetória. Na opção "5- Trajetória cartesiana" a trajetória do robô determinará uma linha reta entre a posição inicial e final do efetuador.

Ao selecionar-se a opção TRAJETÓRIA COM CONFIGURAÇÕES INTERMEDIÁRIAS tem-se a possibilidade de simular diferentes trechos com perfis de velocidade diferentes para cada trecho.

Primeiramente, deve-se determinar o número de configurações intermediárias que se deseja, podendo variar de um a três. Selecionando o número de três configurações intermediárias se estará determinando quatro trechos distintos.

Em seguida o sistema pergunta se o usuário deseja entrar com as coordenadas cartesianas do manipulador das configurações intermediárias. Respondendo "s" (sim) o sistema pede para ser determinado o tempo correspondente ao trecho, fornece-se o valor e tecla-se <enter>. Deve-se tomar o cuidado de não fornecer um valor superior ao tempo total do percurso, o tempo do último trecho é calculado pelo sistema.

Após a determinação do tempo, deve-se determinar o vetor de posição do efetuador (x,y,z), o vetor direção (bx,by,bz) e o vetor orientação (nx,ny,nz) da garra, para esta configuração intermediária.

Caso a opção de entrar com as coordenadas cartesianas de cada trecho não seja desejada, digita-se "n" e tecla-se <enter>. Será pedido então para determinar-se os valores de "teta" para as juntas rotacionais, ou os

valores de "d" para as juntas prismáticas, para cada configuração intermediária.

O procedimento de determinação das configurações intermediárias e o tempo de cada trecho se repete para cada configuração intermediária. Terminada a entrada de dados, é executada a simulação.

Após terminar a simulação o usuário é perguntado se deseja testar os atuadores. Neste ponto o usuário tem a possibilidade de verificar se os valores limites, que foram dados para a velocidade, torque, redução e eficiência dos atuadores, permitem obter os valores resultantes da simulação. Respondendo "s"(sim) é realizada esta verificação dos atuadores. Se o resultado do teste for negativo, tem-se uma indicação que as características fornecidas aos atuadores são inferiores às exigidas na simulação. Neste caso, as características dos atuadores ou o tempo de percurso devem ser então alterados. O sistema ainda indica qual foi a junta restritiva e o instante em que a restrição ocorreu.

Ao fim da simulação realizada, todos os valores de posição, velocidade, aceleração, torque de todas as juntas e a energia consumida, são armazenados na memória.

Em seguida, o sistema fornece ao usuário o valor da energia que foi consumida durante o percurso, se assim o usuário desejar, ou seja, digitando "s" e teclando <enter>.

Depois é possível observar o comportamento das juntas, deve-se então selecionar a junta e em seguida escolher qual o item que deverá ser mostrado:

"1"-Torque ou força;

"2"-Posição;

"3"-Velocidade;

"4"-Aceleração.

Selecionada uma das opções, é em seguida apresentada na tela uma tabela com o valor da variável correspondente a cada instante.

O comportamento de outra junta pode ser observado respondendo de novo positivamente.

Terminada a função o sistema volta ao menu principal. Neste ponto o usuário pode optar por redefinir o tipo de robô e/ou os seus parâmetros cinemáticos e dinâmicos, determinar um novo tempo de trajetória e, se desejar analisar os valores calculados, deve escolher a opção "4" Parâmetros de saída.

2.5 Função PARÂMETROS DE SAÍDA

Esta função apresenta para uma análise todos os gráficos referentes as grandes mencionadas anteriormente e a trajetória descrita pelo robô durante a operação. A primeira tela mostra as opções que podem ser escolhidas. (fig. 2.5.1)

```
****PARAMETROS DE SAÍDA****  
  
Este modulo permite a obtencao de curvas  
caracteristicas do manipulador, tais como:  
comportamento da velocidade, aceleracao, posicao,  
trajetoria ou torque com o tempo, tendo como  
entrada os dados do perfil de velocidade selecionado  
no modulo anterior (programacao de trajetoria).  
  
1_TRAJETORIA.  
2_POSICAO.  
3_VELOCIDADE.  
4_ACELERACAO.  
5_TORQUE.  
6_ENERGIA.  
7_CURVAS DE TORQUE.  
  
Selecione a curva:█
```

fig 2.5.1 - Tela inicial da função.

Selecionando a opção "1" Trajetória, deve-se decidir entre a apresentação da trajetória completa (todas as configurações, digita-se "s") ou os pontos percorridos pelo efetuador (digita-se "n"). Optando pela apresentação da trajetória completa, nova pergunta é realizada, "Deseja animação? (s/n)". Digitando "s" o sistema irá apresentar uma tela com o manipulador movendo-se de sua configuração inicial para a final, passando pelas configurações intermediárias. Digitando "n" será apresentada uma tela com todas configurações da simulação simultaneamente.

As opções "2", "3", "4" e "5" exigem que seja determinada qual a

junta que se quer analisar, uma vez escolhida a junta surge na tela o gráfico correspondente.

A opção "6" mostra a curva correspondente a energia total consumida durante a simulação.

A opção "7" Curvas de Torque, apresenta um novo menu. (fig. 2.5.2)

```
1_TORQUE GRAVITACIONAL X TEMPO.  
2_TORQUE INERCIAL X TEMPO.  
3_TORQUE NAO-LINEAR X TEMPO.  
4_TORQUE X VELOCIDADE.  
5_TORQUE X VELOCIDADE NOS ATUADORES.
```

fig. 2.5.2 - Menu de curvas de torque.

Esta opção possibilita a análise das componentes de torque sobre cada junta e o exame das características torque X velocidade dos atuadores .

O sistema solicita que seja selecionada uma das curvas, e em seguida pergunta qual a junta que será analisada. Selecionada a junta , imediatamente é mostrado o gráfico.

Depois de cada opção selecionada deve-se teclar <enter>.

Esta é a principal função para análise do comportamento do manipulador durante a simulação. O sistema permanece nesta função até que o usuário decida não mais solicitar gráficos, sendo assim retorna para o menu principal.

Como todos os valores estão armazenados na memória, esta função poderá ser acessada novamente, com os dados da última simulação executada.

2.6 Função OTIMIZAÇÃO DE TRAJETÓRIA

Estando no menu principal optando pela opção "5" OTIMIZAÇÃO DE

TRAJETÓRIA, o sistema permite melhorar a trajetória descrita pelo manipulador, visando obter o tempo mínimo da possível para a trajetória programada, de acordo com as restrições escolhidas para o tipo de robô.

Se na função programação de trajetória o teste dos atuadores foi negativo não será possível realizar a otimização, já que as características dos atuadores são inferiores às exigidas na simulação.

Ao entrar nesta função o usuário é indagado se deseja todas as marcações de tempo durante os testes. Digitando "s" a cada teste é apresentado o tempo, digitando "n" será apresentado somente o tempo do último teste.

Em seguida será realizada a otimização em relação ao tempo. O tempo de percurso é então decrementado de 0,20 segundos e processa-se uma nova simulação. Esta operação se repete até que seja obtido um teste negativo. Ao final da função deve-se digitar qualquer caracter e teclar <enter> para voltar ao menu principal.

2.7. Função SIMULAÇÃO DE OPERAÇÃO

Pode-se ainda selecionar a opção "6" SIMULAÇÃO DE OPERAÇÃO. Esta função relaciona o robô ao ambiente de mesas e peças simulando uma operação de manipulação. Esta simulação é fácil de ser executada pelo usuário pois as instruções a serem fornecidas, são do tipo pega e coloca.

Durante esta função o robô está sendo programado novamente, apesar de ser com outro tipo de instrução, portanto a trajetória realizada anteriormente torna-se sem efeito.

Nesta função apenas a posição inicial e os parâmetros dinâmicos, incluídos nas etapas anteriores, serão utilizados.

Esta função apresenta um texto explicativo e uma saída gráfica. Nesta são mostras duas mesas e o robô escolhido na configuração inicial determinada anteriormente (fig 2.7.1).

Nesta funcao o usuario tem a opcao de simular operacoes de manipulacao de objetos, de acordo com as posicoes iniciais, que tambem sao escolhidas do usuario e atraves de instrucoes de alto nivel não necessitando a conhecimento previo sobre robotica.

O usuario deve inicialmente construir uma cena, ou seja, posicionar os 2 objetos de acordo com a posicao da mesa sem marcacoes, apos isto deve seleccionar a primeira peca a ser manipulada pelo robo juntamente com a opcao desejada, a seguir repete-se o procedimento para a segunda peca.

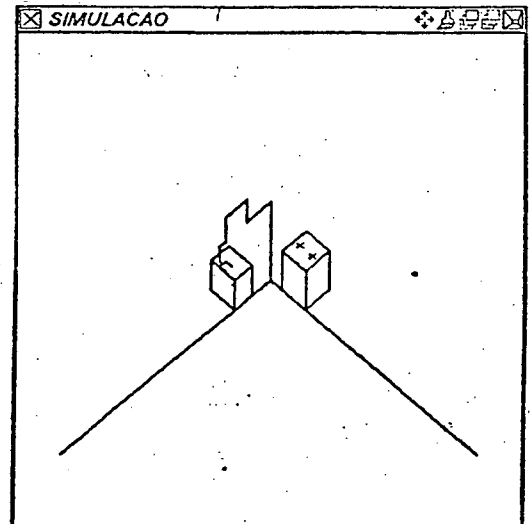


fig. 2.7.1 - Tela inicial da função.

Em seguida o sistema solicita que seja determinada a posição da peça vermelha. Os valores de "X" e "Y" que deverão ser fornecidos devem estar dentro da faixa de valores apresentada, caso contrário a peça será posicionada fora das mesas.

Terminado o processo de posicionamento da peça vermelha, será mostrado na janela a peça posicionada.

Todo processo se repete para uma peça verde. (fig 2.7.2)

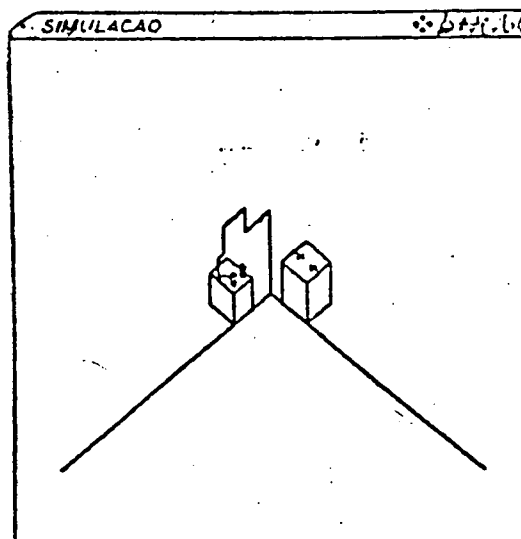


fig. 2.7.2 - Tela apresentando as peças posicionadas sobre a mesa.

O procedimento seguinte é o da determinação das massas das duas peças, inicialmente da peça vermelha e depois da peça verde. Digita-se o valor

de massa desejado e tecla-se <enter>.

Deve-se, depois de determinar as massas, selecionar a peça que será manipulada e em seguida o tempo que o manipulador deverá levar para se aproximar e pegar a peça selecionada (Tempo 1).

O sistema então pergunta sobre qual operação o usuário deseja realizar:

- 1- Posicionar sobre o ponto vermelho da outra mesa;
- 2- Posicionar sobre o ponto verde da outra mesa;
- 3- Posicionar sobre a peça verde(ou vermelha).

Digita-se a opção e tecla-se <enter>.

Em seguida determina-se o tempo que o manipulador levará para posicionar a peça na opção escolhida anteriormente (Tempo 2). Será então realizada a simulação, mostrando na saída gráfica o robô movimentando-se.

O sistema pergunta se o usuário deseja realizar novo ciclo. Respondendo "s" (sim), retorna ao ponto em que é pedido qual das peças que será manipulada, e de lá prossegue da forma anterior. Respondendo "n" (não), surge a opção de se obter algumas posições da última simulação realizada. Respondendo novamente "s" (sim) será mostrada algumas posições do robô da simulação anterior (fig 2.7.3). Em seguida o sistema retorna ao menu principal.

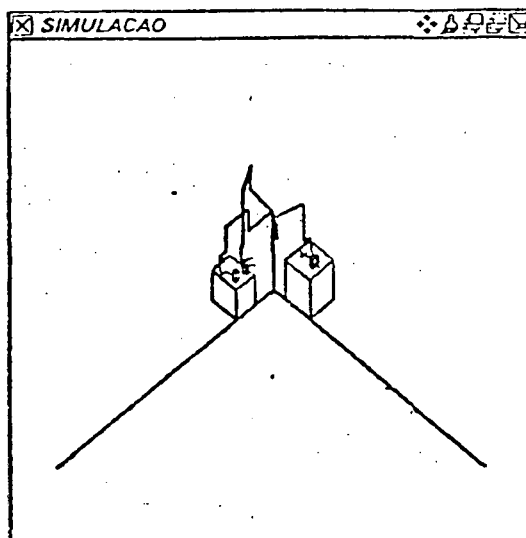


fig. 2.7.3 - Tela apresentando algumas posições do manipulador.

2.8 Função AUXÍLIO AO USUÁRIO

A função "7" AUXÍLIO AO USUÁRIO pode ser acessada somente do menu principal. Ela serve de apoio para esclarecer o que cada função do sistema irá executar.

Logo ao ser chamada a função apresenta um novo menu.

```
*****
Esta e' a funcao auxilio ao usuario
poderá ser acessada de acordo com o modulo
que o programa estiver executando.
Selecione a opcao, a ser acessada.

(a) Arquivo de Configuracoes.
(c) Formulacao Cinemática.
(d) Formulacao Dinamica.
(t) Programacao de Trajetoria.
(p) Parametros de Saida .
(o) Otimizacao.
(s) Simulacao de Operacao.
(r) Retorna ao sistema .
*****

Opcao:█
```

Menu da função AUXÍLIO AO USUÁRIO.

Este menu apresenta as alternativas que contem as explicações de cada função do sistema. Nesta função o usuário deverá digitar a letra correspondente à opção da qual ele deseja obter alguma informação. Após selecionar a letra e teclar <enter> será apresentado o texto explicativo referente à função escolhida. Teclando <enter> novamente retorna-se ao menu de opções onde nova letra pode ser selecionada.

Ao selecionar a letra "a" Arquivo de Configurações é apresentado um outro menu com todas as configurações arquivadas, devendo ser escolhida uma delas. Desejando retornar ao menu de opções após a apresentação do texto referente à configuração escolhida tecla-se <enter>.

Seleciona-se a letra "r" e tecla-se <enter> para retornar ao menu principal.

3 EXEMPLO DE APLICAÇÃO

Afim de esclarecer melhor a utilização do simulador será mostrado um exemplo de aplicação. Neste exemplo será observada a influência da escolha do perfil de velocidade ao realizar uma programação de trajetória com um robô do tipo "SCARA".

Na função ARQUIVO DE CONFIGURAÇÕES foi escolhido o tipo de robô e foram determinadas as posições inicial (figura 3.1) e final.

NJUNT	TETA graus	ALFA graus	A mts	D mts	JUNTA tipo
1	0.00	0.00	0.80	1.60	rot
2	0.00	180.00	0.70	0.45	rot
3	0.00	0.00	0.20	0.50	trans
4	0.00	0.00	0.00	0.10	rot
5	0.00	90.00	0.00	0.20	rot
6	0.00	0.00	0.00	0.10	rot

Posicao| direcionamento| orientacao inicial do efetuador :
(1.70,0.10,1.25) (0.00,1.00,0.00) (1.00,0.00,0.00)

fig. 3.1 - Tabela de parâmetros da posição inicial.

Alterou-se apenas o valor de teta das juntas 2 (dois) e 4 (quatro), e o valor de "d" da junta 3 (três) para formar a configuração final. (90°, 180° e 0,2m, respectivamente)

Na sequência entrou-se na função FORMULAÇÃO DINÂMICA. Determinou-se como massas das ligações os seguintes valores: 60, 40, 30, 20, 10 e 5 kg (para as ligações de 1 a 6 respectivamente). Não optou-se por entrar com os momentos de inércia locais, desta forma o sistema calculou estes valores considerando as ligações como sendo hastes.

Na função PROGRAMAÇÃO DE TRAJETÓRIA foram determinados como limites dos atuadores os valores apresentados na tabela a seguir.

Valores limites dos atuadores :			
Velocidades (rpm)	torques (Nm)	reducao	eficiencia
728.00	28.00	100	0.900
720.00	20.00	100	0.900
720.00	20.00	200	0.900
1200.00	10.00	100	0.900
1200.00	10.00	100	0.900
1200.00	10.00	100	0.900

Tabela de limites dos atuadores.

Determinou-se o tempo de 6,0 segundos como tempo de percurso para trajetória, e optou-se por simular-se com gravidade.

Inicialmente realizou-se a simulação escolhendo o perfil de velocidade triangular. A energia consumida com este perfil foi de 256,09 J.

Em seguida selecionou-se a função OTIMIZAÇÃO DE TRAJETÓRIA e realizou-se a otimização em relação ao tempo. Para o perfil de velocidade triangular o tempo mínimo foi de 5,0 segundos. Retornou-se à função PROGRAMAÇÃO DE TRAJETÓRIA onde realizou-se uma nova simulação, desta vez com o tempo mínimo obtido, e resultou um valor de 284,64 J para energia consumida.

Após a realização da última simulação entrou-se na função PARÂMETROS DE SAÍDA para obter uma saída gráfica da trajetória descrita pelo robô durante a simulação ("1-Trajétoria" fig 3.2) e uma curva que representa a velocidade da junta 1 durante a trajetória (fig 3.3).

Novamente foi selecionada a função PROGRAMAÇÃO DE TRAJETÓRIA, determinou-se o tempo de 6,0 segundos para a trajetória e alterou-se o perfil de velocidade, desta vez perfil parabólico. Para este perfil resultou o valor de 228,78 J de energia consumida durante o percurso.

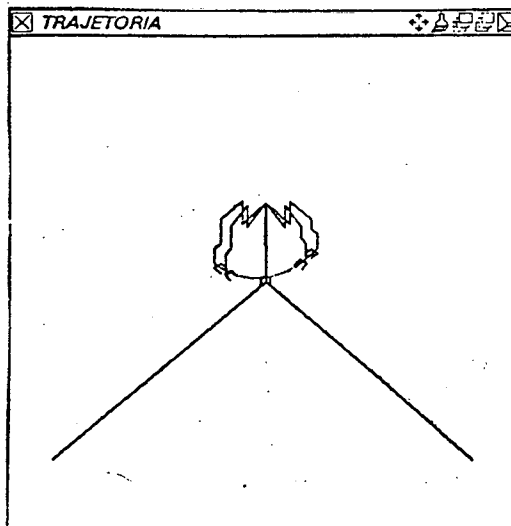


fig. 3.2 - Trajetória descrita pelo manipulador.

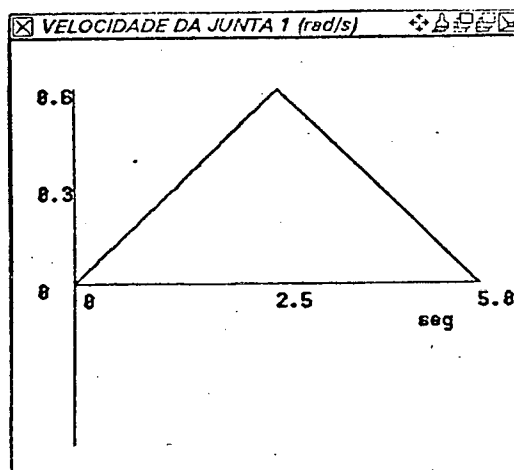


fig. 3.3 - Velocidade da junta 1, perfil triangular.

Realizou-se a otimização em relação ao tempo para o perfil parabólico. Obteve-se como tempo mínimo o valor de 3,8 segundos.

Realizou-se nova simulação para o perfil de velocidade parabólico com o tempo de 3,8 segundos resultante da otimização, obteve-se então o valor de 284,86 J para a energia consumida durante o percurso. A figura 3.4 mostra a curva de velocidade apresentada nesta simulação.

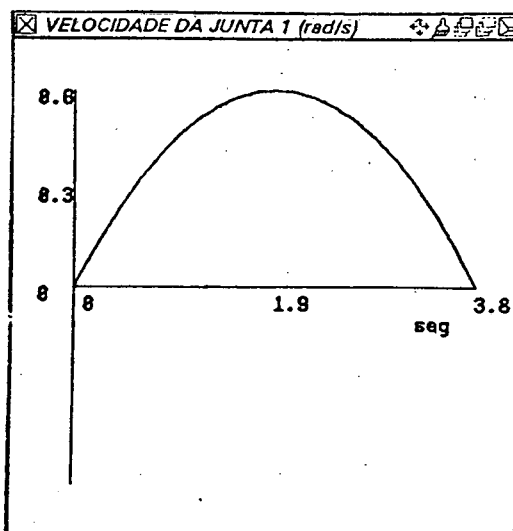


fig. 3.4 - Velocidade da junta 1, perfil parabólico.

Foram realizados os testes dos atuadores, com resultados positivos. Finalmente, são apresentadas as seguintes conclusões:

Para o perfil triangular otimizado obteve-se o valor de 284,64 J de energia consumida com um tempo de 5,0 segundos. Para o perfil parabólico o valor da energia consumida foi de 284,86 J com um tempo mínimo de 3,8 segundos. Simulando novamente o perfil parabólico com o tempo de 5,0 segundos, obtem-se como resultado o valor de 245,30 J para energia consumida.

Este exemplo mostra que com a mudança do perfil de velocidade de triangular para parabólico há uma redução do tempo mínimo de percurso ou, simulando com o mesmo tempo de percurso, há um ganho de energia.

Este exemplo representa apenas uma aplicação do simulador, outros estudos foram realizados.