

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA
DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO E
AUTOMAÇÃO INDUSTRIAL

Este exemplar corresponde à versão final da tese
defendida por José Tarcísio Franco de
Camargo pela Comissão
Julgadora em 27 / 01 / 75
Léo Pini Magalhães
Orientador

ANIMAÇÃO MODELADA POR COMPUTADOR:
TÉCNICAS DE CONTROLE DE MOVIMENTO EM ANIMAÇÃO

Dissertação de doutorado
apresentada à Faculdade de
Engenharia Elétrica da
Universidade Estadual de
Campinas como parte dos
requisitos para a obtenção
do título de Doutor em
Engenharia Elétrica.

AUTOR: JOSÉ TARCÍSIO FRANCO DE CAMARGO
ORIENTADOR: PROF. DR. LÉO PINI MAGALHÃES

JANEIRO DE 1995

BANCA DE AVALIAÇÃO

Presidente:

Prof. Dr. Léo Pini Magalhães (UNICAMP-FEE-DCA)

Membros externos:

Prof. Dr. José Luis Encarnação (T. H. Darmstadt - GRIS - Alemanha)

Prof. Dr. Bruno Feijó (PUC-RJ / Departamento de Infomática)

Membros internos:

Profa. Dra. Wu Shin Ting (UNICAMP-FEE-DCA)

Prof. Dr. Rafael Santos Mendes (UNICAMP-FEE-DCA)

27/01/1995

AGRADECIMENTOS:

À FAPESP, pelo apoio financeiro concedido através de minha bolsa de doutorado.

Aos meus queridos orientadores de mestrado e doutorado e, principalmente, amigos:

Prof. Dr. Léo Pini Magalhães

Prof. Dr. Rui Fragassi Souza

Ao grande amigo e companheiro de pesquisas:

Alberto Barbosa Raposo

Aos meus pais:

José Franco de Camargo e

Therezinha Aparecida Villani de Camargo

À minha querida esposa:

Eliana Anunciato Franco de Camargo

RESUMO

A utilização de computadores no desenvolvimento de animações ou simulações torna-se cada vez maior atualmente. A cada dia que passa, novas e mais complexas tarefas são atribuídas ao computador neste campo de aplicações.

Dessa forma, este trabalho procura apresentar um conjunto de ferramentas úteis ao desenvolvimento de animações ou simulações modeladas por computador. Especificamente, apresentamos um conjunto de técnicas de controle de movimento de objetos rígidos articulados ou não.

São abordados dois níveis de controle neste trabalho: local e global. As técnicas de controle local, que caracterizam a descrição do movimento de um único objeto, são apresentadas sob a forma de modelos cinemáticos e dinâmicos. Por sua vez, o controle global, que caracteriza a interação entre objetos numa animação ou simulação, é apresentado através de um modelo de "Sistema Dinâmico a Eventos Discretos" (DEDS). Consequentemente, esta subdivisão permite-nos descrever as ações a serem executadas numa animação de forma mais clara e eficiente.

São também apresentados alguns exemplos de implementações possíveis para as técnicas apresentadas neste texto, disponíveis também em um disquete que acompanha este exemplar.

ABSTRACT

The use of computers in modelling animations or simulations has been increasing lately. Every day new and more complex tasks are attributed to computers in this application field.

In this way, this work presents a set of tools to develop computer modeled animations or simulations. Specifically, we present a set of motion control techniques that are useful for modelling the motion of (articulated) rigid objects.

Two levels of control are studied: local and global. Local control techniques are used to model the motion of each object individually and are presented here through kinematic and dynamic models. On the other hand, Global control techniques are responsible for the interactions among objects and are presented here through a "Discrete Event Dynamic Systems" (DEDS) model. In consequence, this subdivision allows us to describe the actions in the animation with a higher degree of efficiency.

Some implementations for these techniques are also presented.

GLOSSÁRIO DE NOTAÇÕES

O seguinte glossário refere-se às equações apresentadas ao longo deste texto.

${}^A P_{(3 \times 1)}$	Vetor representado no sistema de coordenadas {A}.
${}^A P_{(4 \times 1)}$	Vetor, expresso em sua forma homogênea, representado no sistema de coordenadas {A}.
${}^A R_B (3 \times 3)$	Matriz, representa a orientação entre os sistemas de coordenadas {A} e {B}.
${}^A T_B (4 \times 4)$	Matriz de transformação, que leva um vetor representado no sist. de coord. {B} para o sist. de coord. {A}.
$\vec{n}, \vec{o}, \vec{a}$	Vetores-coluna que definem a orientação de uma matriz de transformação.
\vec{p}	Vetor-coluna que define o vetor de posição de uma matriz de transformação.
${}^A P_{B(3 \times 1)}$	Vetor, representado no sist. de coord. {A}, que vai da origem deste sist. de coord. até a origem do sist. de coord. {B}.
${}^A V_Q$	Velocidade do ponto "Q" em relação ao sist. de coord. {A}.
${}^B V_{AORG}$	Velocidade linear da origem do sist. de coord. {A} em relação ao sist. de coord. {B}.
v_A	Velocidade linear da origem do sist. de coord. {A} em relação ao sist. de coord. referencial .
${}^B v_A$	Velocidade linear da origem do sist. de coord. {A}, em relação ao sist. de coord. referencial , porém representada no sist. de coord. {B}.
${}^B \dot{v}_A$	Aceleração linear da origem do sist. de coord. {A}, em relação ao sist. de coord. referencial , porém representada no sist. de coord. {B}.
${}^B \Omega_A$	Velocidade angular do sist. de coord. {A} em relação ao sist. de coord. {B}.
ω_A	Velocidade angular do sist. de coord. {A} em relação ao sist. de coord. referencial .
${}^B \omega_A$	Velocidade angular do sist. de coord. {A}, em relação ao sist. de coord. referencial , porém representada no sist. de coord. {B}.
${}^B \dot{\omega}_A$	Aceleração angular do sist. de coord. {A}, em relação ao sist. de coord. referencial , porém representada no sist. de coord. {B}.
a_i	Parâmetro de Denavit-Hartenberg. Comprimento do segmento "i", medido ao longo do eixo "x" do sist. de coord. preso ao segmento "i".
α_i	Parâmetro de Denavit-Hartenberg. Ângulo entre os eixos "z" dos sist. de coord. "i" e "i+1".

- θ_i Parâmetro de Denavit-Hartenberg. Ângulo entre os eixos "x" dos sist. de coord. "i-1" e "i". Também chamado de variável de junta ou grau de liberdade.
- $\dot{\theta}_i$ Velocidade do grau de liberdade "i".
- $\ddot{\theta}_i$ Aceleração do grau de liberdade "i".
- d_i Parâmetro de Denavit-Hartenberg. Distância entre os sist. de coord. "i-1" e "i".
- \vec{q} Vetor de variáveis de junta de um objeto articulado.
- A_i Matriz tensor de inércia de um objeto em relação ao sist. de coord. {A}
- F_i Força externa aplicada no centro de massa do segmento "i" de um objeto articulado.
- N_i Torque externo aplicado no centro de massa do segmento "i" de um objeto articulado.
- ${}^i f_1$ Força exercida na junta do segmento "i" de um objeto articulado, pelo segmento "i-1", representada no sist. de coord. {i}.
- ${}^i n_1$ Torque exercido na junta do segmento "i" de um objeto articulado, pelo segmento "i-1", representado no sist. de coord. {i}.
- I_3 Matriz identidade 3x3.

ÍNDICE

PARTE I:	
INTRODUÇÃO À ANIMAÇÃO POR COMPUTADOR	01
CAPÍTULO 1:	
INTRODUÇÃO	02
CAPÍTULO 2:	
Animação modelada por computador e suas técnicas de controle	06
2.1 - Introdução	06
2.2 - Métodos de controle do movimento	07
2.3 - Trabalhos já realizados na área / Um breve histórico	08
2.4 - Campo de aplicações	11
PARTE II:	
MÉTODOS DE CONTROLE LOCAL DO MOVIMENTO	12
CAPÍTULO 3:	
Representação de objetos no espaço tridimensional	13
3.1 - Obtenção de uma matriz de orientação entre sistemas de coordenadas	16
3.2 - Notação para sistemas de coordenadas variantes no tempo	17
CAPÍTULO 4:	
Modelo cinemático de um manipulador	20
4.1 - Parâmetros de um manipulador segundo Denavit-Hartenberg	20
4.2 - Geração de trajetórias para uma estrutura articulada	22
CAPÍTULO 5:	
Modelo dinâmico para o manipulador	27
5.1 - Propagação de velocidades	28
5.2 - Propagação de acelerações	29
5.3 - Tensor de inércia / Matriz inercial de um objeto rígido	30
5.3.1 - Teorema dos eixos paralelos	32
5.4 - Modelo dinâmico iterativo de Newton-Euler	33
5.4.1 - Equação de Newton	33
5.4.2 - Equação de Euler	33
5.5 - Algoritmo iterativo de Newton-Euler	35
5.5.1 - Estabelecendo condições iniciais para o algoritmo	36
CAPÍTULO 6:	
Modelo dinâmico para estruturas articuladas ramificadas	37
6.1 - Formulação dinâmica de Euler	38
6.2 - Características dos objetos articulados	44
6.2.1 - Limite de juntas / Torque de restauração	44
6.2.2 - Efeito "mola" / Torque de reação	45
6.2.3 - Perdas nas juntas / Torque de dissipação	45
CAPÍTULO 7:	
Controle ótimo em animação	47
7.1 - Ferramentas matemáticas	47
7.2 - Minimização de uma função sujeita a restrições	48
7.2.1 - 1º Algoritmo	50
7.2.2 - 2º Algoritmo: Algoritmo de projeção do gradiente	50

CAPÍTULO 8:	
Planejamento de trilhas / Desvio de obstáculos	56
8.1 - Um caso estudado	57
8.1.1 - Definição dos campos de potenciais	57
8.1.2 - Definição de uma trilha otimizada	59
PARTE III:	
CONTROLE GLOBAL DO MOVIMENTO	61
CAPÍTULO 9:	
Controle global de uma animação através do modelo de DEDS	62
9.1 - Projeto do mecanismo de controle global	63
9.1.1 - Modelo de locomoção cinemático	63
9.2 - Modelo de locomoção a eventos discretos	64
9.2.1 - Definição da sintaxe de ESMs	64
9.2.2 - Planejamento das ESMs para o modelo de locomoção	68
9.2.3 - 1º Caso: Movimento retilíneo ininterrupto	69
9.2.4 - 2º Caso: Movimento retilíneo com possibilidade de interrupção	69
PARTE IV:	
CASOS DE ESTUDO	72
CAPÍTULO 10:	
Implementações realizadas	73
10.1 - Exemplos implementados através do modelo de DEDS	73
10.1.1 - Simulação de um ator bípede	73
10.1.2 - Simulação de um sistema de cruzamento de ruas	74
10.2 - Planejamento de trilhas através de campos de potenciais	79
10.3 - Simulação de um braço articulado (robô) com cinco graus de liberdade	80
10.4 - Animação de uma luminária Luxo	83
10.5 - Simulação do modelo dinâmico de Euler	85
10.6 - Modelo de simulação simplificado para uma mão	85
PARTE V:	
CONCLUSÕES E BIBLIOGRAFIA	88
CAPÍTULO 11:	
Conclusões Gerais	89
11.1 - Considerações gerais	89
11.2 - Resultados apresentados	89
11.3 - Contribuições deste trabalho	91
11.4 - Perspectiva futura	91
BIBLIOGRAFIA	92
APÊNDICE:	
Utilização do disquete de simulações	96

PARTE I:
INTRODUÇÃO À ANIMAÇÃO POR COMPUTADOR

Na Parte I são introduzidos os conceitos fundamentais relativos à animação por computador: métodos de classificação e técnicas de controle. Também encontra-se apresentado um breve relato do desenvolvimento desta ciência e o estado atual da arte. Apresentamos, também, a proposta deste trabalho.

CAPÍTULO 1: INTRODUÇÃO:

Computação Gráfica é o termo largamente utilizado para designar, em português, *Graphics* ou *Computer Graphics* em inglês. Dentro desta vasta área de pesquisa em Ciência da Computação, existe um ramo de investigações que, atualmente, vem gerando grandes expectativas sendo comumente denominado Animação por Computador.

Existem diversas definições possíveis para o termo ANIMAÇÃO. Entre elas podemos citar:

- arte de "dar vida" a personagens; ou
- ilusão da sensação de movimento através da apresentação de uma sequência de quadros (imagens).

Entretanto, uma animação não precisa, necessariamente, envolver deslocamentos de personagens podendo também ser caracterizada por:

- uma metamorfose: quando um personagem¹ muda sua forma;
- uma mudança de cor: quando, por exemplo, um ator se ruboriza de vergonha;
- uma variação na intensidade luminosa: por exemplo, a luz ambiente diminui à medida em que o Sol se põe.

O computador, atualmente, constitui uma ferramenta poderosa para o desenvolvimento de animações. Dizemos que uma animação é *auxiliada por computador* quando a máquina apenas auxilia os animadores na criação de desenhos, no cálculo de movimentos, na escolha de cores, no controle da câmera que fotografa os quadros, etc. Uma animação pode ser considerada como *modelada por computador* quando o animador determina os atores, o ambiente e as ações a serem executadas, cabendo ao computador o *controle da animação*² como um todo. Através da animação modelada por computador podemos gerar desde animações até simulações:

- animações convencionais: desenhos animados tradicionais (*cartoons*), onde não existe qualquer compromisso com a realidade física;
- simulações: onde simulamos fenômenos físicos tais como o deslocamento do braço de um robô, um lançamento balístico, etc.

Durante o processo de criação de uma animação modelada por computador, necessitamos de, pelo menos, três ferramentas fundamentais:

- um modelador geométrico³, para que possamos definir as características geométricas das entidades envolvidas na animação tais como formato, etc.;
- um mecanismo de controle, para que possamos controlar as interações entre atores e o ambiente;
- um mecanismo de visualização (*rendering*), para que possamos visualizar e fotografar cada quadro gerado durante o processo de animação.

¹ Os objetos de nosso interesse são comumente denominados "personagens" ou "atores".

² Os termos "controle de uma animação" e "controle de movimento em animação" não devem ser interpretados da mesma forma que o termo homônimo clássico da engenharia. Em nosso caso, tais expressões referem-se a métodos que podem ser utilizados na descrição de movimentos de objetos e na interação entre estes.

³ Existem métodos sem modelagem de objetos, por exemplo, animação estocástica, "fuzzy", etc. Entretanto, tais exceções não são abordadas neste trabalho.

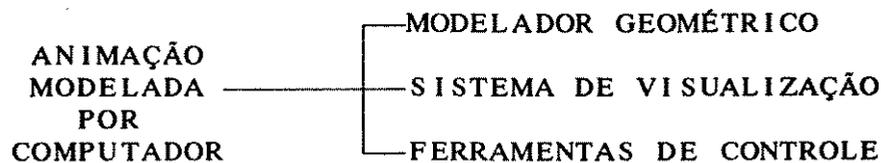


Fig. 1.1: Componentes de um sistema para animação modelada por computador.

Baseado em técnicas apresentadas em [Thalmann (1985)], encontra-se disponível na Faculdade de Engenharia Elétrica da Universidade Estadual de Campinas um sistema cinemático para a animação de objetos no espaço tridimensional, denominado *TOOKIMA (A TOOL KIT for scripting computer Modeled Animation)* [Hounsell (1992)]. Este sistema define um conjunto de ferramentas para descrição algorítmica de animações de objetos modelados por computador utilizando a filosofia quadro-a-quadro (keyframe).

O *TOOKIMA* é composto basicamente por três sub-sistemas. O sistema de modelagem de objeto é o Geomod (modelador geométrico de objetos por fronteiras). O sistema de Visualização é o sistema Scanline e o sistema de descrição e controle de animação é o próprio *TOOKIMA*, que é responsável por integrar os dois anteriores, além de prover uma linguagem para sincronização temporal e descrição das diversas cenas que comporão a animação.

Entretanto, tendo em vista o fato de que o *TOOKIMA* foi concebido como um sistema puramente cinemático, novas pesquisas foram iniciadas após sua implementação no sentido de se desenvolver uma ferramenta de análise dinâmica, de modo a termos um maior "grau de realismo" nas animações geradas. Recentemente, um novo sistema denominado *ANIMADO* foi implementado de forma a complementar o sistema *TOOKIMA* [Rodrigues (1993)]. Através do sistema *ANIMADO* torna-se possível gerar simulações dinâmicas de objetos rígidos, além deste ser capaz de detectar e responder a colisões entre objetos. Note que os sistemas *TOOKIMA* e *ANIMADO* abordam o problema da animação modelada por computador de forma restrita, pois são capazes de modelar apenas alguns tipos de situações. Entretanto, a arquitetura "aberta" destes sistemas permite-nos acoplar aos mesmos novos módulos, de modo a termos maior flexibilidade quanto ao tipo de animações que podemos tratar.

A filosofia que orientou o desenvolvimento dos sistemas *TOOKIMA* e *ANIMADO* limitou-se à utilização de apenas conceitos cinemáticos ou dinâmicos para a descrição do movimento. Isto ocorreu devido ao fato de que, durante o período de desenvolvimento destes sistemas, a maior parte dos "cientistas da animação" havia abordado o problema de controle do movimento de objetos através da utilização do cálculo diferencial e das leis físicas, tais como as Leis Dinâmicas de Newton e de Euler. De fato, sem a aplicação das leis físicas que regem os movimentos dos objetos dificilmente seríamos capazes de simular os movimentos de qualquer objeto. Entretanto, a complexa natureza de diversos sistemas físicos nos leva a lidar com sistemas de equações diferenciais cada vez mais complexos. Mais ainda, não existe uma solução global que possa ser aplicada à animação⁴ de qualquer objeto. Ou seja, uma solução conveniente para determinado sistema pode não trazer bons resultados quando aplicada a um outro sistema.

⁴ Note que utilizamos indistintamente os termos "Animação" e "Simulação".

Portanto, acreditamos que, atualmente, a pesquisa por ferramentas de controle para animações (convencionais ou simulações) constitui uma vasta área de investigação científica neste ramo da ciência da computação. Dessa forma, dedicaremos este trabalho, integralmente, à apresentação de ferramentas de controle para animações.

Como alternativa para a solução deste problema (isto é, o problema de controle do fluxo de uma animação) sugerimos a divisão do modelo que controla os movimentos numa animação em diversas partes, denominadas *Blocos de Controle Local* e *Bloco de Controle Global*. Imagine-se um sistema a ser simulado composto por diversos "atores". Podemos atribuir a cada ator um bloco de controle local, que conterá as "regras de comportamento" de cada ator. Ou seja, as leis físicas que regem o movimento de um único ator estão embutidas em seu respectivo módulo de controle local. A natureza deste bloco sugere uma ferramenta matemática (tal como cálculo diferencial) para que seja definido o modelo de um ator e seu comportamento próprio.

Uma vez que o sistema sugerido é composto por muitos atores, deve haver interações entre atores e o animador. O bloco de controle global sugere uma abordagem lógica para a determinação de um modelo para estas interações. Ou seja, caso ocorra um determinado evento, quais atores serão afetados? Como um determinado ator deve responder? Dessa forma, dependendo daquilo que está ocorrendo no sistema, os atores se comportam de uma dada forma, que poderia ser diferente caso outro evento houvesse ocorrido. A figura 1.2 apresenta os componentes de um sistema de animação modelada por computador que dispõe de dois níveis de controle.

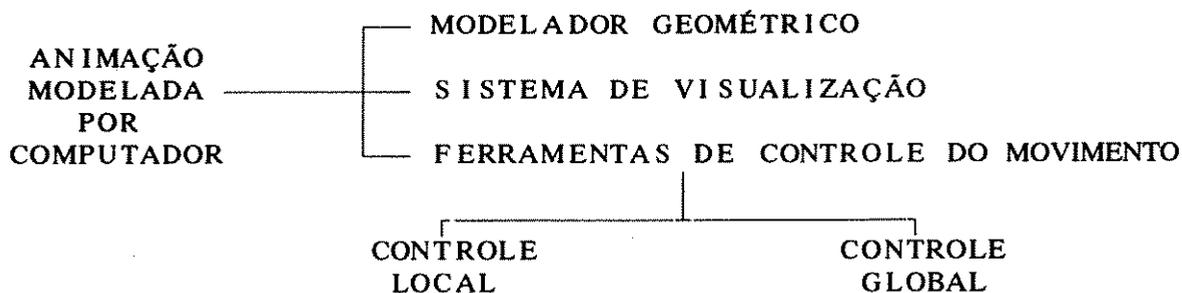


Fig. 1.2: Componentes de um sistema para animação modelada por computador com dois níveis de controle.

Neste trabalho apresentaremos algumas técnicas de controle local e global interessantes aos cientistas da animação. As técnicas desenvolvidas serão apresentadas com o auxílio de pequenas implementações, cabendo ao animador o desenvolvimento das particularidades de sua própria aplicação. O texto desta dissertação encontra-se dividido em cinco partes.

Ainda na primeira parte desta dissertação, no capítulo 2, apresentaremos um resumo da evolução dos métodos de controle do movimento já desenvolvidos.

Na segunda parte desta dissertação serão apresentadas algumas ferramentas de controle local próprias para o desenvolvimento de modelos de sistemas que dispõem apenas de objetos rígidos, podendo estes ser articulados. Inicialmente apresentaremos uma proposta para a representação de objetos no espaço 3-D. Na sequência, serão desenvolvidos modelos cinemáticos e dinâmicos que podem ser aplicados ao controle do movimento de objetos rígidos, articulados ou não. Finalizando esta etapa serão apresentados métodos para o planejamento de trilhas.

A terceira parte desta dissertação trata do problema de controle global de uma

animação. Apresentamos uma proposta para a solução deste problema de controle fundamentada no modelo de *Sistemas Dinâmicos a Eventos Discretos (DEDS - Discrete Event Dynamic System)*. Utilizaremos uma *Máquina de Estados Estendida (ESM - Extended State Machine)* para a representação do mecanismo de controle global de animações, visto que se trata de uma ferramenta interessante para o modelo de sistemas a eventos discretos, como é o caso de inúmeras animações.

A quarta parte deste trabalho apresenta um conjunto de problemas em animação modelada por computador e suas respectivas soluções, obtidas com o auxílio das ferramentas desenvolvidas nos capítulos anteriores.

A quinta parte deste trabalho contém a conclusão e a bibliografia desta dissertação.

Dessa forma, a contribuição deste trabalho encontra-se, em primeiro lugar, na realização de um *survey* reunindo as principais técnicas de controle em animação utilizadas no momento. Este *survey*, que reúne as partes I e II desta dissertação, constitui uma importante referência para os cientistas da animação no Brasil, visto que ainda nos encontramos em um estágio inicial desta ciência.

Por sua vez, como contribuição global, temos a proposta de formalização de animações baseadas no paradigma de eventos discretos e na divisão em controle local e controle global. O paradigma atual em animação encontra-se extremamente direcionado às restrições de espaço-tempo, [Witkin (1988)], enquanto que, neste trabalho, procuramos contribuir no sentido de estabelecer um paradigma complementar, fundamentado no modelo de *Sistemas Dinâmicos a Eventos Discretos*, através da utilização de conceitos tais como controle local, controle global e máquinas de estados.

CAPÍTULO 2: ANIMAÇÃO MODELADA POR COMPUTADOR E SUAS TÉCNICAS DE CONTROLE

2.1 - INTRODUÇÃO

Classificações de animação por computador auxiliam-nos a melhor organizar esta área, caracterizada por uma rápida inovação. Também podem nos ajudar na análise de diferenças (e pontos comuns) entre as várias técnicas de animação e, ainda, melhor entender a evolução deste campo.

Os primeiros modelos computacionais definidos em animação por computador eram basicamente geométricos. Sendo que animação por computador deriva da animação tradicional, os primeiros esforços imitavam os métodos tradicionais. Tais esforços concentravam-se, basicamente, no resultado gráfico da animação. Nesta primeira etapa do desenvolvimento desta ciência destacam-se os trabalhos de James Korein [Korein (1982)] e Nadia e Daniel Thalmann [Thalmann (1985)] que se fundamentavam na utilização de técnicas cinemáticas para o desenvolvimento de animações. Posteriormente, para se obter movimentos mais realísticos, foram introduzidos os modelos baseados em leis físicas. O modelo dinâmico desenvolvido por William Armstrong e apresentado em [Armstrong (1985)] constitui um marco fundamental nesta etapa de transição de paradigmas. Estes modelos, entretanto, possuem um problema: todos os atores comportam-se basicamente segundo as mesmas regras. Dessa forma, visto que os seres humanos possuem comportamentos relativamente distintos entre si, alguns pesquisadores recentemente introduziram o conceito de modelos comportamentais de modo a garantir a individualidade dos atores.

De forma a tentar melhor classificar esta área, que se encontrava em desenvolvimento vertiginoso, P. M. Isaacs propôs, em [Isaacs (1987)], a classificação das técnicas de animação em três classes: animação *keyframe*, animação procedimental e simulação dinâmica.

A animação *keyframe* é derivada diretamente das técnicas de animação tradicional, nas quais o animador determina o que deve ser apresentado em cada quadro e, portanto, especifica explicitamente a "cinemática" do sistema. O computador é então utilizado como ferramenta de apoio na determinação de quadros intermediários (*in-between*) que interpolarão os quadros-chave (*keyframes*) determinados pelo usuário. Embora este tipo de sistema forneça quase que completamente o controle da cena ao animador, tal método é deficiente na criação de sequências dinamicamente corretas (realismo) além de ser muito trabalhoso.

A animação procedimental fundamenta-se na capacidade do computador em determinar a cinemática de uma sequência baseando-se em instruções implícitas ao invés de posições explícitas. Uma classe de métodos procedimentais consiste na "cinemática inversa", onde o objetivo final de uma sequência é especificado pelo animador, sendo os movimentos consequentes determinados por algoritmos. Em alguns casos, alguns princípios dinâmicos são utilizados para se obter um certo grau de realismo. Embora através de métodos procedimentais tenham sido obtidos bons resultados, este método também sofre de deficiências na sua integridade dinâmica.

Sistemas destinados à simulação dinâmica de objetos articulados tem surgido recentemente na literatura relativa a robótica e bio-mecânica [Craig (1989)]. Alguns destes métodos incorporam conceitos de restrições cinemáticas e seus mecanismos de controle associados. Fundamentados na formulação dinâmica do movimento, estes métodos mostram-se muito eficientes quando se necessita de um alto grau de realismo no movimento ou a obediência (simulação) de um comportamento físico conhecido.

A simulação dinâmica requer como entrada as características físicas e comportamentais do objeto, bem como as restrições do ambiente. O modelo físico do objeto pode incluir descrições de todas as suas articulações bem como suas conectividades. O estado inicial do sistema contém a posição inicial e velocidade do elemento modelado em um certo instante inicial. Funções comportamentais mostram o estado atual do sistema com relação a forças externas ou a movimentos específicos.

Atualmente, com a evolução dos modelos de controle de movimento, surgiram maiores desenvolvimentos no relacionamento entre atores e seus ambientes. O surgimento de técnicas tais como inteligência artificial e programação orientada para objetos tornou possível um avanço significativo na interação de atores com o seu ambiente, de atores com atores e de atores com o animador. Desta forma, estes novos conceitos de interação provocaram a reformulação dos métodos de classificação acima descritos de modo a que estes se tornassem mais abrangentes.

Estes tipos de interação e as três categorias de modelos anteriormente citados deram origem à seguinte classificação proposta por Nadia e Daniel Thalmann [Thalmann (1991)]. Cada caso na fig. 2.1 representa um método existente e potencial para o controle de animação, com o incremento de sua complexidade podendo ser observado de cima para baixo e da esquerda para a direita. A seguir apresentamos a classificação apresentada em [Thalmann (1991)].

	MCMs* Geométricos	MCMs Físicos	MCMs Comportamentais
Único A tor	Rotoscopia Téc. <i>keyframe</i> Restr. cinemát.	Dinâmica	Mod. muscular Sist. facial
Interface A tor-Ambiente	Contorna obst. Detecta inters.	Mod. colisões Mod. de formação	Sensores Desvio de obst. baseado em visão
Interface A tor-A tor	Colisões geom. entre atores	Colisões dinam. entre atores	Comun. emoções entre atores
Interface Animador-A tor	Proj. de trajet. com <i>dataglove</i>	Entr. de forças e realimentação	Visão via camera de vídeo

(*) MCM: Método de Controle do Movimento

Fig. 2.1: Classificação proposta em [Thalmann (1991)].

2.2 - MÉTODOS DE CONTROLE DO MOVIMENTO

Podemos classificar cenas animadas por computador envolvendo atores sintéticos em função do método de controle do movimento (MCM) e dos tipos de interação entre os atores [Thalmann (1991)]. Um método de controle de movimento (MCM) determina como um ator é animado. Podemos caracterizar um MCM de acordo com o tipo de informação privilegiada na animação de um ator sintético. Por exemplo, num sistema *keyframe* para um corpo articulado, a informação principal são os ângulos das articulações. Num sistema baseado nas leis físicas de dinâmica, a informação privilegiada consiste num conjunto de forças e torques. É claro que, na solução de equações dinâmicas, também são tratadas informações tipo ângulo, entretanto, estas são considerados informações derivadas.

Informações privilegiadas para o controle do movimento de atores concentram-se em três categorias: geométricas, físicas e comportamentais. Estas dão origem às três correspondentes categorias de MCMs.

MCMs GEOMÉTRICOS: No primeiro grupo de MCMs, a informação privilegiada tem natureza geométrica. Tipicamente o movimento é determinado em função de coordenadas, ângulos e outras características relacionadas à forma. Embora MCMs geométricos concentrem-se principalmente na determinação de movimento de esqueletos, eles também podem ser aplicados no cálculo de deformações de corpos.

MCMs FÍSICOS: Um MCM físico utiliza leis e características físicas como base para cálculo de movimentos. As informações privilegiadas para estes MCMs incluem características físicas tais como massa, inércia e rigidez. As leis físicas envolvidas auxiliam no controle do movimento de esqueletos, mas também possuem um importante papel no cálculo de deformações de corpos.

MCMs COMPORTAMENTAIS: Um MCM comportamental determina o movimento de um ator em termos de seu comportamento, frequentemente definido como o modo através do qual os seres vivos agem. Comumente descreve-se o comportamento em termos de linguagem natural por intermédio de características sociais, psicológicas e fisiológicas. Infelizmente, tais termos não apresentam, necessariamente, facilidades de redução a movimentos de um ou dois músculos ou juntas. Termos para a descrição de variações na personalidade entre atores são também considerados como informação privilegiada por MCMs comportamentais.

INTERFACE PARA ATORES: Como visto, o controle do movimento de atores depende do processamento de informações geométricas, físicas ou comportamentais. Além disso, devemos considerar o relacionamento entre um ator e o resto de seu universo, ou seja, a interface para o ator. Esta interface pode ser descrita por quatro casos básicos:

- Um único ator: tal ator se encontra sozinho na cena e não interage com qualquer outro objeto.
- Interface ator-ambiente: o ator encontra-se em movimento em um ambiente, estando ciente da existência deste ambiente.
- Interface ator-ator: as ações realizadas por um ator são reconhecidas por outros e podem mudar o comportamento de um segundo ator.
- Interface ator-animador: não apenas o animador pode enviar informações ao ator, mas também o ator pode responder, enviando informações ao animador.

Dessa forma, para cada uma das interfaces acima descritas, podemos dispor de um tipo de MCM (geométrico, físico ou comportamental), ajustando o melhor controlador a cada caso.

2.3 - TRABALHOS JÁ REALIZADOS NA ÁREA / UM BREVE HISTÓRICO

A utilização de computadores no desenvolvimento de animações constitui uma atividade relativamente recente. O salto também recente de tecnologia no desenvolvimento de *hardware* pode justificar este fato. Alguns pesquisadores começaram a se interessar pelo desenvolvimento de técnicas de animação por computador no início da década de 80. [Korein (1982)] apresenta um dos primeiros estudos relatados sobre a animação de objetos articulados. Entre seus objetos de interesse figuravam as cadeias planares articuladas, sendo que suas técnicas de controle fundamentavam-se na descrição cinemática do movimento.

Em 1985, a publicação de [Armstrong (1985)] provocou um grande salto qualitativo nesta área de investigação. William W. Armstrong e Mark W. Green apresentaram, em seu trabalho, uma formulação dinâmica para a simulação do movimento de objetos articulados. Através desta formulação é possível gerar animações com alto grau de realismo de estruturas articuladas tridimensionais dispostas em cadeias lineares ou em cadeias tipo "árvore".

Paul M. Isaacs apresentou, em 1987, um Sistema de Movimento Dinâmico (DYNAMO - DYNAMIC MOTION SYSTEM) capaz de realizar simulações com objetos rígidos e articulados [Isaacs (1987)]. Tal sistema apresentou três métodos para a obtenção de controle de movimento em simulações: restrições cinemáticas, funções de comportamento e dinâmica inversa. A combinação de especificações cinemáticas e dinâmicas permite ao animador pensar sobre cada parte da animação do modo mais apropriado resultando em um elevado grau de realismo do movimento. Esta nova técnica para a geração de animações consolidou-se como um paradigma, já em 1988, conforme demonstram os resultados apresentados em [Isaacs (1988)] e [Wilhelms (1988)].

Técnicas baseadas em restrições dinâmicas foram também abordadas em 1988 por Ronen Barzel [Barzel (1988)]. Utilizando restrições dinâmicas foram construídos objetos através da especificação de restrições geométricas; o modelo montava a si mesmo à medida em que seus elementos se moviam de acordo com as restrições. Os elementos individuais eram corpos rígidos que agiam de acordo com leis físicas e, portanto, apresentavam comportamento realístico. Para se implementar as restrições, um conjunto de "forças restritoras" foi encontrado, o qual fez com que os corpos agissem de acordo com restrições; encontrar tais "forças restritoras" consistiu num problema de dinâmica inversa.

Seguindo uma filosofia muito semelhante, Andrew Witkin propôs, em 1988, um método para animação de objetos rígidos baseado em restrições de espaço-tempo (*Spacetime Constraints*) [Witkin (1988)]. Segundo o método proposto, o animador deve determinar o que o objeto deve fazer através de restrições espaciais ("onde") e temporais ("quando"), além dos recursos disponíveis para a realização do movimento (forças e torques). As exigências contidas nesta descrição, combinadas com as leis da física, constituem um problema de otimização. A solução para o problema deve ser um movimento fisicamente válido, satisfazendo e otimizando as restrições. O paradigma das restrições de espaço-tempo é ainda abordado em outros trabalhos tais como [Schroeder (1990)], [Witkin (1990)b] e [Ngo (1993)].

Técnicas de controle ótimo para o movimento de objetos sob restrições dinâmicas também são utilizadas em animação, conforme mostra o trabalho de Lynne S. Brotman, [Brotman (1988)]. O método desenvolvido neste trabalho modela o movimento de objetos e seus respectivos ambientes através de equações diferenciais obtidas por intermédio da mecânica clássica. Dessa forma, de modo a satisfazer as restrições impostas pelos *keyframes* é aplicado o controle externo, de modo a se obter o maior grau de realismo no movimento.

Michael van de Panne introduziu, em 1990, o conceito de SSC (*State Space Controller*) [Panne (1990)]. Um SSC define um conjunto de torques e forças de controle que guiam um objeto até um objetivo (posição) específico a partir de um grande domínio de configurações iniciais, de modo a otimizar tempo e energia gastos. Um objetivo é caracterizado por um conjunto de "estados-destino" e, dependendo da natureza deste conjunto, diversas classes de movimento são possíveis.

Em 1991, R. C. Zeleznik propôs um sistema orientado para objetos para a integração de técnicas interativas de animação [Zeleznik (1991)]. Neste seu trabalho foi apresentado um sistema bastante versátil que facilita a integração de uma variedade de paradigmas de simulação e animação. O sistema permite o modelamento de

diversos objetos que mudam de formato, aparência e comportamento ao longo do tempo.

Uma recente abordagem para a animação de objetos, fundamentada em leis físicas, consiste na "Rede de Sensores-Atuadores" [Panne (1993)]. De acordo com este novo paradigma, são associados alguns sensores e atuadores a um objeto definido pelo usuário. Dessa forma, torna-se possível determinar automaticamente alguns modos de locomoção para este objeto.

Além das técnicas de controle de objetos rígidos acima citadas, técnicas destinadas ao controle da animação de objetos deformáveis foram pesquisadas paralelamente em [Witkin (1987)] e [Witkin (1990)a].

Por sua vez, de maneira complementar aos trabalhos acima citados, [D.Kalra (1992)] preocupa-se em abordar o problema das interações existentes entre objetos numa animação. Em seu trabalho é proposto um esquema orientado a tempo e eventos para o tratamento das interações entre atores numa animação.

Especificamente com relação ao Brasil, o estudo da animação por computador constitui ainda um recente campo de pesquisas. [Hounsell (1992)] e [Silva (1992)] apresentam sistemas de animação modelada por computador fundamentados na descrição cinemática do movimento de objetos rígidos. [Lemos (1992)] e [Rodrigues (1993)] apresentam, por sua vez, sistemas para a modelagem de animações de objetos rígidos fundamentados no equacionamento dinâmico da simulação a ser realizada.

Sistemas mais específicos também vem sendo abordados, como é o caso de [Amorim (1992)] que descreve um sistema para a representação do movimento humano e [Rigo (1993)] que apresenta um sistema gráfico interativo para modelagem e animação facial.

O desenvolvimento de técnicas de controle de animações mais sofisticadas também pode ser considerado um objetivo de interesse comum entre diversos pesquisadores. [Camargo (1993)] apresenta um conjunto de ferramentas de controle local e global úteis ao desenvolvimento de animações de objetos rígidos, articulados ou não. [Watt (1993)] apresenta técnicas para o controle de animações de objetos rígidos ou deformáveis.

O estudo de mecanismos adequados ao controle da interação entre atores numa animação (controle global) também vem recebendo atenção especial em alguns trabalhos. [Feijó (1993)] apresenta um modelo para a animação comportamental de atores baseado em lógica. [Camargo (1994)] apresenta um modelo de simulação fundamentado no conceito de "Sistema Dinâmico a Eventos Discretos".

Podemos ainda citar contribuições de várias outras áreas de pesquisa ao desenvolvimento da animação modelada por computador. Diversos trabalhos realizados em robótica constituíram uma rica fonte de experiências para o desenvolvimento de técnicas de controle local de objetos articulados. Resultados apresentados em [Paul (1981)], [Goldenberg (1985)], [Morgan (1985)], [McInnis (1986)], [Craig (1989)] e [D'Eleuterio (1991)] servem como base para o desenvolvimento de simulações cinemáticas ou dinâmicas de objetos articulados. Da mesma forma, técnicas desenvolvidas para o planejamento de trilhas de robôs e prevenção de colisões apresentadas em [Crowley (1985)], [Gilbert (1985)], [Takahashi (1989)], [Gewali (1990)], [Cameron (1990)], [Hwang (1992)], [Fardanesh (1992)] e [Shaffer (1992)] são perfeitamente aplicáveis aos sistemas de animação de nosso interesse. Por sua vez, modelos desenvolvidos para o estudo de sistemas dinâmicos a eventos discretos, tais como aqueles apresentados em [Ho (1989)], [Cohen (1989)] e [Ostroff (1989)], são utilizados para o desenvolvimento de técnicas de controle global de animações.

2.4 - CAMPO DE APLICAÇÕES

Não existe um limite bem definido para o campo de possíveis aplicações para a "ciência" da animação/simulação por computador. Uma vasta faixa de possibilidades inclui as artes, a robótica, a bio-engenharia, etc. Entretanto, existe atualmente uma grande tendência para o desenvolvimento de sistemas de animação que sejam capazes de simular diversos fenômenos da natureza.

Já em 1985 Michael Girard havia discutido um modelo de locomoção para estruturas bípedes ou quadrúpedes [Girard (1985)]. Atualmente, a equipe do Prof. Norman Badler, da Universidade da Pensilvânia, EUA, tem apresentado importantes trabalhos sobre a simulação de movimentos humanos, conforme descrito em [Badler (1993)]. Outros fenômenos biológicos tais como a simulação do movimento de músculos faciais [P.Kalra (1992)] e a animação do movimento de crescimento de plantas [Stam (1993)] são também abordados atualmente.

Fenômenos tais como deslocamento de fluidos, turbulência gasosa e movimento do vento estão sendo atualmente simulados com uma certa frequência, conforme apresentam os trabalhos [Sakas (1992)], [Shinya (1992)] e [Prusinkiewicz (1993)].

Estão sendo também desenvolvidas algumas interessantes aplicações em realidade virtual. [Bergman (1993)] apresenta um sistema para a visualização e animação de moléculas relativamente complexas. [Taylor (1993)] apresenta um sistema interativo, denominado "NANOMANIPULADOR", capaz de permitir ao usuário a interação, em escala atômica, com superfícies de materiais em processamento.

Este capítulo apresentou de forma sucinta os diferentes esforços já realizados na área de Animação por Computador. Pretendemos com isso facilitar ao leitor o correto enquadramento dos temas da dissertação no contexto mais amplo da Animação.

PARTE II: MÉTODOS DE CONTROLE LOCAL DO MOVIMENTO

Na Parte II é introduzido um conjunto de técnicas que podem ser aplicadas ao controle local de uma animação. São também apresentados modelos cinemáticos e dinâmicos que podem ser aplicados a sistemas compostos por objetos rígidos, articulados ou não.

CAPÍTULO 3: REPRESENTAÇÃO DE OBJETOS NO ESPAÇO TRIDIMENSIONAL

Antes de partirmos para o desenvolvimento das técnicas de controle local é interessante definir uma convenção para a representação dos objetos de nosso interesse. As definições encontradas neste capítulo são derivadas das convenções propostas em [Craig (1989)] e [McInnis (1986)]. Considere a figura abaixo:

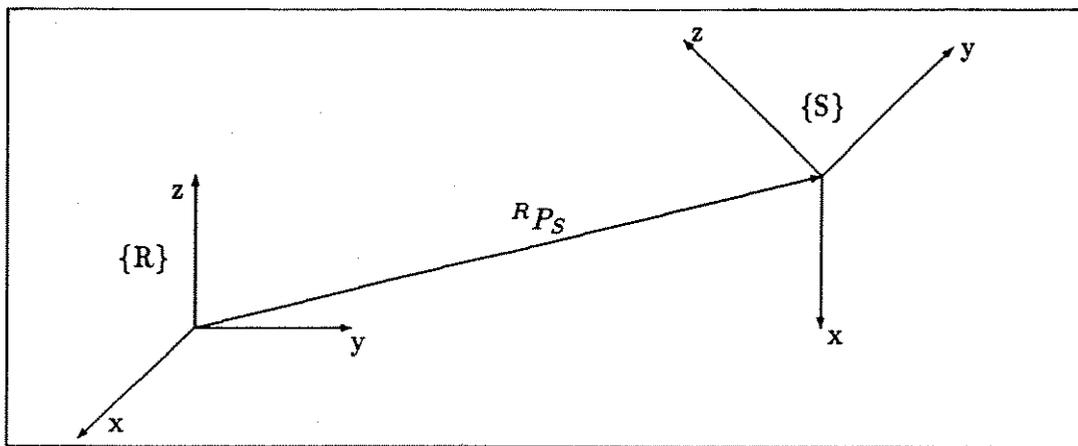


Fig. 3.1: Esquema de representação de sistemas de coordenadas no espaço 3-D.

Seja:

$R_{V_{(3 \times 1)}}$ um vetor qualquer, representado no sistema de coordenadas {R}.

$S_{V_{(3 \times 1)}}$ um vetor qualquer, representado no sistema de coordenadas {S}.

$R_{T_{S(4 \times 4)}}$ uma transformação de coordenadas, que leva um vetor representado no sistema de coordenadas {S} para o sistema de coordenadas {R}.

Para estabelecermos uma relação entre vetores e transformações de coordenadas, e melhor descrever as translações, vamos considerar, quando realizarmos o produto entre uma matriz de transformação e um vetor, o vetor expresso na forma homogênea:

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}_{(4 \times 1)}$$

sendo ¹ "x", "y" e "z" as coordenadas do vetor em seu devido sistema de coordenadas.

¹É comum a substituição do termo "sistema de coordenadas" por outros, como é o caso do termo "frame" ou a abrevitura "S.C.". Utilizaremos tais denominações conforme conveniência.

A transformação ${}^R_S T$ é inversível e pode ser representada na forma:

$${}^R_S T = \begin{bmatrix} \begin{bmatrix} R_R \\ S_T \end{bmatrix}^T & -\begin{bmatrix} R_R \\ S_T \end{bmatrix}^T \cdot R_{P_S} \\ \hline 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.5)$$

Assim sendo:

$$S_V = {}^S_T \cdot R_V \quad (3.6)$$

$$S_V = \begin{bmatrix} R_R \\ S_T \end{bmatrix}^T \cdot R_V - \begin{bmatrix} R_R \\ S_T \end{bmatrix}^T \cdot R_{P_S} \quad (3.7)$$

Podemos concatenar diversas transformações como produto entre matrizes:

$$\begin{matrix} A_T \\ D \end{matrix} = \begin{matrix} A_T \cdot B_T \cdot C_T \\ B \\ C \\ D \end{matrix} \quad (3.8)$$

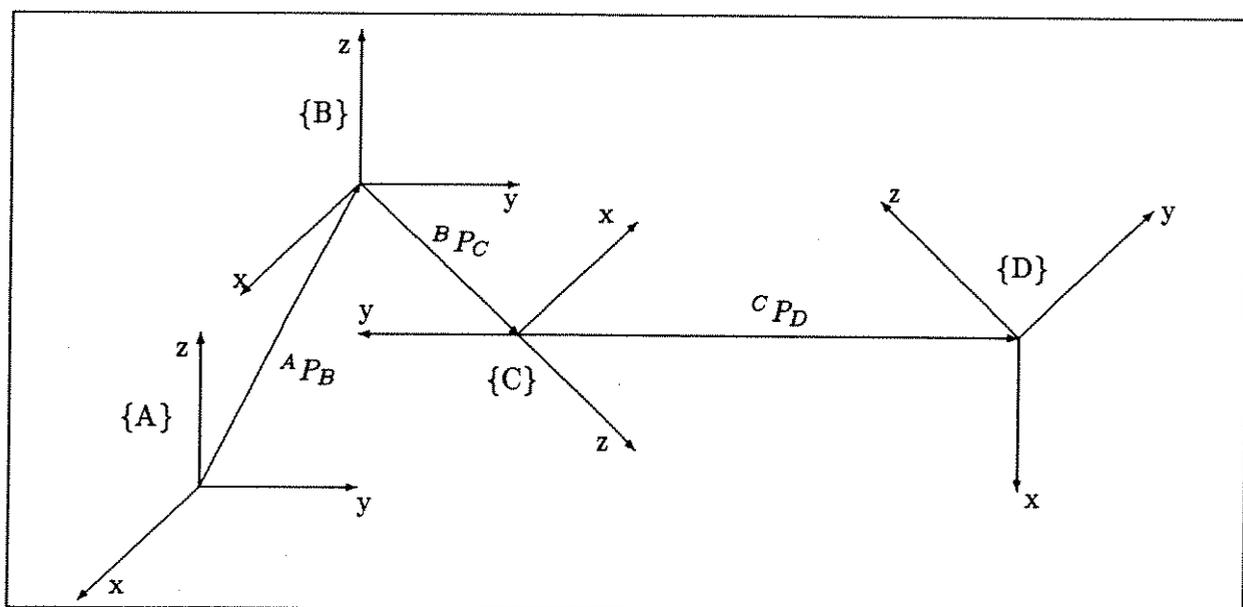


Fig. 3.2: Concatenando diversos sistemas de coordenadas.

3.1 - OBTENÇÃO DE UMA MATRIZ DE ORIENTAÇÃO ENTRE SISTEMAS DE COORDENADAS

Considere dois S.C. {A} e {B}, inicialmente coincidentes, sendo {A} o S.C. referencial e {B} o S.C. que deve ser rotacionado. É possível a representação de uma matriz de orientação através da "notação de rotação de Euler", onde a rotação será representada em relação ao próprio S.C. que está girando. Dessa forma, considere a seguinte sequência de passos:

- primeiro, gire o S.C. {B} em torno de seu próprio eixo "x" de um ângulo " α ",
- em segundo lugar, gire o S.C. {B} em torno de seu próprio eixo "y" de um ângulo " β " e,
- finalmente, gire o S.C. {B} em torno de seu próprio eixo "z" de um ângulo " γ ".

Formalmente temos:

$${}^A R_{xyz}(\alpha, \beta, \gamma) = R_x(\alpha) \cdot R_y(\beta) \cdot R_z(\gamma) = \quad (3.9)$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\text{sen}\alpha \\ 0 & \text{sen}\alpha & \cos\alpha \end{bmatrix} \cdot \begin{bmatrix} \cos\beta & 0 & \text{sen}\beta \\ 0 & 1 & 0 \\ -\text{sen}\beta & 0 & \cos\beta \end{bmatrix} \cdot \begin{bmatrix} \cos\gamma & -\text{sen}\gamma & 0 \\ \text{sen}\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} =$$

$$= \begin{bmatrix} (\cos\beta \cdot \cos\gamma) & (-\cos\beta \cdot \text{sen}\gamma) & (\text{sen}\beta) \\ (\text{sen}\alpha \cdot \text{sen}\beta \cdot \cos\gamma + \cos\alpha \cdot \text{sen}\gamma) & (-\text{sen}\alpha \cdot \text{sen}\beta \cdot \text{sen}\gamma + \cos\alpha \cdot \cos\gamma) & (-\text{sen}\alpha \cdot \cos\beta) \\ (-\cos\alpha \cdot \text{sen}\beta \cdot \cos\gamma + \text{sen}\alpha \cdot \text{sen}\gamma) & (\cos\alpha \cdot \text{sen}\beta \cdot \text{sen}\gamma + \text{sen}\alpha \cdot \cos\gamma) & (\cos\alpha \cdot \cos\beta) \end{bmatrix}$$

Podemos também representar a orientação entre dois S.C. através da rotação de um S.C. em relação a um vetor, por intermédio dos "parâmetros de Euler", conforme mostra a fig. 3.3. De acordo com a fig. 3.3, o S.C. {B} pode ser rotacionado de um ângulo qualquer, digamos θ , ao redor do vetor ${}^A \Omega_B$, de acordo com a regra da mão direita, na forma:

$$\begin{aligned} \epsilon_1 &= \Omega_x \cdot \text{sen}(\theta/2) \\ \epsilon_2 &= \Omega_y \cdot \text{sen}(\theta/2) \\ \epsilon_3 &= \Omega_z \cdot \text{sen}(\theta/2) \\ \epsilon_4 &= \cos(\theta/2) \end{aligned} \quad (3.10)$$

Sendo que Ω_x , Ω_y e Ω_z são as componentes do vetor ${}^A\Omega_B$, a matriz de orientação será dada por:

$${}^A_R_B = \begin{bmatrix} (1 - 2.\epsilon_2^2 - 2.\epsilon_3^2) & 2.(\epsilon_1.\epsilon_2 - \epsilon_3.\epsilon_4) & 2.(\epsilon_1.\epsilon_3 + \epsilon_2.\epsilon_4) \\ 2.(\epsilon_1.\epsilon_2 + \epsilon_3.\epsilon_4) & (1 - 2.\epsilon_1^2 - 2.\epsilon_3^2) & 2.(\epsilon_2.\epsilon_3 - \epsilon_1.\epsilon_4) \\ 2.(\epsilon_1.\epsilon_3 - \epsilon_2.\epsilon_4) & 2.(\epsilon_2.\epsilon_3 + \epsilon_1.\epsilon_4) & (1 - 2.\epsilon_1^2 - 2.\epsilon_2^2) \end{bmatrix} \quad (3.11)$$

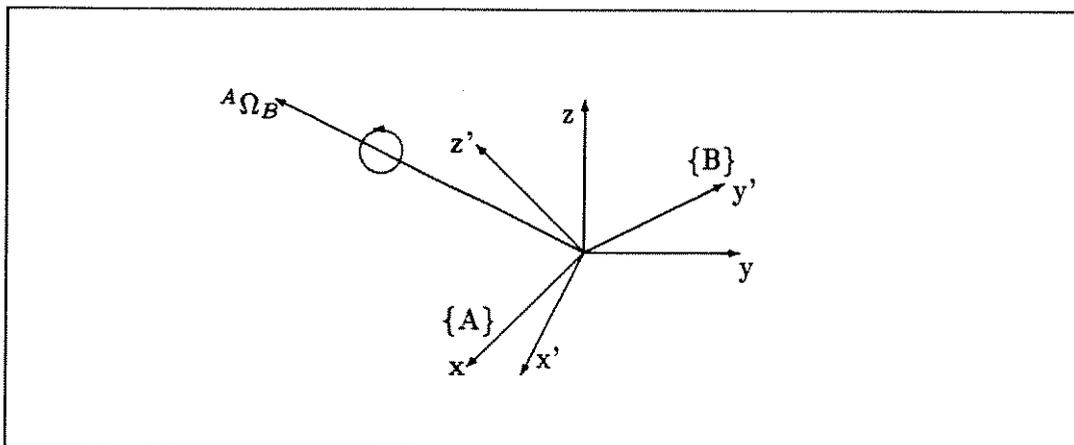


Fig. 3.3: Rotação entre sistemas de coordenadas.

3.2 - NOTAÇÃO PARA SISTEMAS DE COORDENADAS VARIANTES NO TEMPO

Considere a fig. 3.4, onde são apresentados dois S.C. {A} e {B}. Considere também o ponto Q, representado no S.C. {B} (BQ).

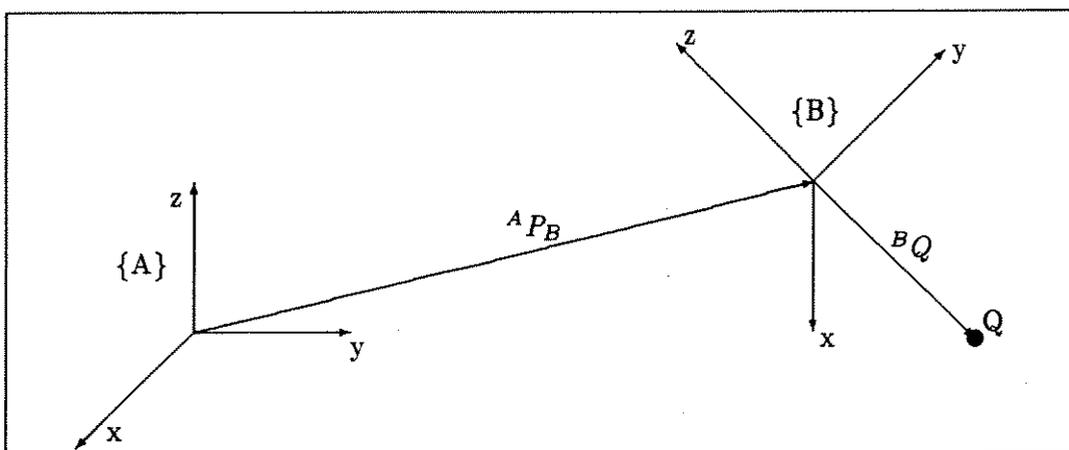


Fig. 3.4: Representação para S.C. variantes no tempo.

Caso o ponto ${}^B Q$ esteja em movimento, seu vetor velocidade será dado por:

$${}^B V_Q = \frac{d({}^B Q)}{dt} = \lim_{\Delta t \rightarrow 0} \frac{{}^B Q(t+\Delta t) - {}^B Q(t)}{\Delta t} \quad (3.12)$$

Caso desejássemos expressar a velocidade do ponto "Q" em termos do S.C. {A}, deveríamos fazer:

$${}^A \left[{}^B V_Q \right] = {}^A R_B \cdot {}^B V_Q \quad (3.13)$$

Por outro lado, se existe variação na distância entre as origens dos S.C. {A} e {B} (os S.C. estão em movimento), então devemos considerar um novo termo:

$${}^A \left[{}^B V_Q \right] = {}^A V_{BORG} + {}^A R_B \cdot {}^B V_Q \quad (3.14)$$

sendo ${}^A V_{BORG}$ a velocidade entre as origens dos S.C. {A} e {B}, medida no S.C. {A}.

Devemos notar que, para que a equação (3.14) seja válida, não é permitida a ocorrência de alterações na orientação entre os S.C. {A} e {B}. Ou seja, ${}^A R_B = cte.$

Consideremos agora que os S.C. {A} e {B} possuem origens coincidentes e que esta condição não se altera com o passar do tempo. Dessa forma: ${}^A P_B = 0 = cte$, como mostra a fig. 3.3.

Devemos também notar que, embora as origens permaneçam coincidentes, existe movimento entre os S.C. {A} e {B}, visto que existe uma velocidade angular ${}^A \Omega_B$ entre os mesmos, dada em termos do S.C. {A}.

A velocidade angular descreve a variação da orientação do S.C. {B} relativa ao S.C. {A}. A direção do vetor velocidade angular indica o eixo de rotação instantâneo de {B} relativo a {A}; a magnitude do vetor velocidade angular indica a velocidade da rotação.

Consideremos novamente a fig. 3.4. Caso exista velocidade linear (${}^A V_{BORG}$) e velocidade angular ${}^A \Omega_B$ entre os S.C. {A} e {B} a velocidade do ponto Q será dada por:

$${}^A V_Q = {}^A V_{BORG} + {}^A R_B \cdot {}^B V_Q + \left[{}^A \Omega_B \times {}^A R_B \cdot {}^B Q \right] \quad (3.15)$$

sendo:

${}^A\Omega_B \times {}^A R_B \cdot {}^B Q$ = componente de rotação, caso o S.C. {B} "gire" em relação ao {A}, representado no S.C. {A};

${}^A R_B \cdot {}^B V_Q$ = velocidade do ponto Q em relação ao S.C. {B}, representada no S.C. {A} e

${}^A V_{BORG}$ = velocidade da origem do S.C. {B} em relação ao S.C. {A}.

Vamos chamar o Sistema de Coordenadas Inercial de {U}. Quando desejarmos representar a velocidade angular de um S.C. {A} qualquer em relação a {U}, utilizaremos a notação ω_A .

Caso dois S.C. estejam se movendo, um em relação ao outro, é possível que exista contínua alteração na matriz de orientação que relaciona os dois sistemas. Para analisarmos esta situação, consideremos o S.C. {A} em movimento, em relação ao S.C. inercial, {U}.

Neste caso temos:

$$\frac{d}{dt} \begin{bmatrix} U \\ A \end{bmatrix} \begin{bmatrix} R \\ \end{bmatrix} = \begin{bmatrix} \omega_A \end{bmatrix}^* \cdot \begin{bmatrix} U \\ A \end{bmatrix} \begin{bmatrix} R \\ \end{bmatrix} \quad (3.16)$$

onde:

$$\begin{bmatrix} \omega_A \end{bmatrix}^* = \begin{bmatrix} 0 & \omega_z & -\omega_y \\ -\omega_z & 0 & \omega_x \\ \omega_y & -\omega_x & 0 \end{bmatrix} \quad \text{é o dual do vetor } \omega_A = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \quad (3.17)$$

Uma vez que tenhamos definido uma convenção para a notação a ser utilizada na descrição dos sistemas a serem simulados, podemos proceder à modelagem dos sistemas a serem simulados. No capítulo seguinte definiremos uma metodologia para a representação e simulação de objetos articulados no espaço tridimensional.

CAPÍTULO 4: MODELO CINEMÁTICO DE UM MANIPULADOR

Manipulador é um termo comum em robótica utilizado para designar estruturas rígidas dotadas de articulações como, por exemplo, o braço de um robô. Em [Koren (1982)] temos apresentada uma primeira aplicação do modelo cinemático de um manipulador em Computação Gráfica. Um problema bastante comum em robótica consiste em se determinar os torques nas juntas das articulações para que o manipulador realize determinado movimento. Este tipo de problema pode ser dividido em duas partes:

i-) Em primeiro lugar a trajetória do robô deve ser devidamente especificada. Ou seja, por quais pontos do espaço 3-D deve se deslocar a "mão" ou o "braço" do robô? Qual a orientação a ser seguida pela "mão" (*end-effector*) do robô? Para a trajetória desejada, quais as posições e orientações dos demais segmentos articulados do robô? Para solucionarmos estas questões devemos desenvolver um modelo cinemático para o robô, conforme apresentaremos neste capítulo.

ii-) Além da devida especificação da trajetória, devemos determinar quais torques devem ser aplicados a cada junta pois serão os torques produzidos pelos motores das juntas os responsáveis pelo deslocamento final desejado. Neste ponto nos deparamos com um problema cuja solução envolve a determinação do modelo dinâmico do robô, que será tratado no próximo capítulo.

4.1 - PARÂMETROS DE UM MANIPULADOR SEGUNDO DENAVIT-HARTENBERG [Girard (1985)]

Consideremos os segmentos de uma estrutura articulada, apresentados na fig. 4.1:

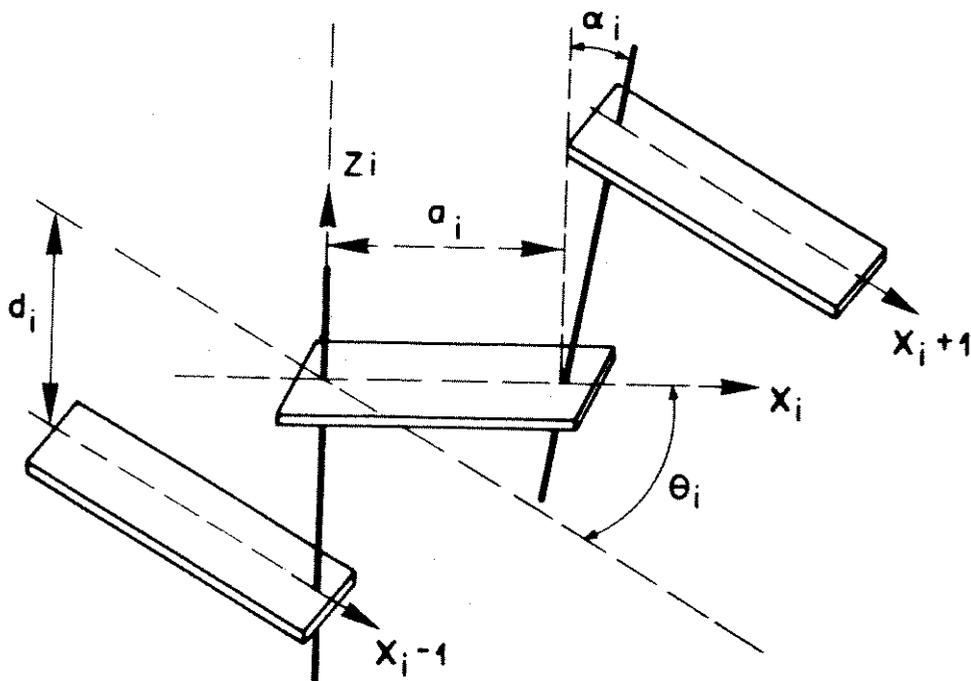


Fig. 4.1: Parâmetros de Denavit-Hartenberg para uma estrutura articulada.

Temos, segundo [Girard (1985)], que:

a_i é o comprimento do segmento "i" (medido ao longo do eixo "x" do S.C. "preso" ao segmento "i"),

α_i é o ângulo entre os eixos "z" dos S.C. dos segmentos "i" e "i+1",

θ_i é o ângulo entre os eixos "x" dos S.C. dos segmentos "i-1" e "i",

d_i é a distância entre os segmentos "i-1" e "i".

Como pode ser notado pela fig. 4.1, para melhor representar as características de cada segmento, associamos a cada segmento do manipulador um S.C. próprio (e fixo) a este, conforme mostra a fig. 4.2:

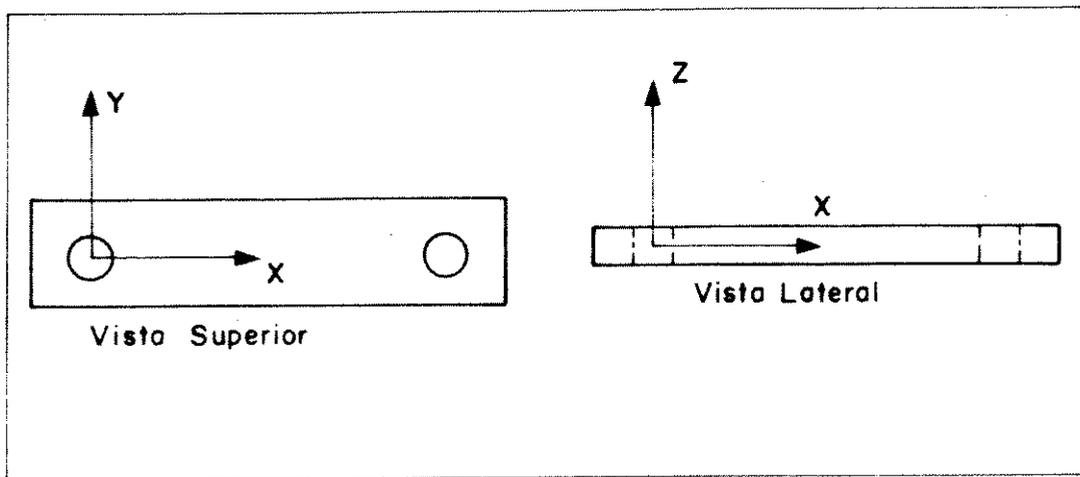


Fig. 4.2: Sistema de Coordenadas de um segmento.

Se associarmos a cada segmento do objeto articulado um S.C., então poderemos definir também transformações que relacionam tais S.C.. Logo, seja a transformação entre dois segmentos adjacentes, "i-1" e "i":

$${}^{i-1}T_i = \begin{bmatrix} \vec{n}_i & \vec{o}_i & \vec{a}_i & \vec{p}_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.1)$$

onde:

$\vec{n}_i, \vec{o}_i, \vec{a}_i$, são vetores-coluna que definem a orientação do segmento,

\vec{p} equivale ao vetor-coluna de posição, que vai da origem do S.C. {i-1} até a origem do S.C. {i}.

e ainda:

$$\begin{aligned} n_{ix} &= \cos\theta_i & o_{ix} &= -\cos\alpha_i \cdot \text{sen}\theta_i & a_{ix} &= \text{sen}\alpha_i \cdot \text{sen}\theta_i & p_{ix} &= a_i \cdot \cos\theta_i \\ n_{iy} &= \text{sen}\theta_i & o_{iy} &= \cos\alpha_i \cdot \cos\theta_i & a_{iy} &= -\text{sen}\alpha_i \cdot \cos\theta_i & p_{iy} &= a_i \cdot \text{sen}\theta_i \\ n_{iz} &= 0 & o_{iz} &= \text{sen}\alpha_i & a_{iz} &= \cos\alpha_i & p_{iz} &= d_i \end{aligned} \quad (4.2)$$

Concatenando as transformações entre quaisquer segmentos teremos:

$${}^j_T = {}^j_{j+1}T \cdot {}^{j+1}_{j+2}T \cdot \dots \cdot {}^{i-1}_iT \quad (4.3)$$

Notemos que, devido à sua estrutura rígida, quando o objeto sofrer algum deslocamento, apenas o parâmetro " θ " sofrerá variação.

4.2 - GERAÇÃO DE TRAJETÓRIAS PARA UMA ESTRUTURA ARTICULADA

Considere a estrutura articulada representada na fig. 4.3.

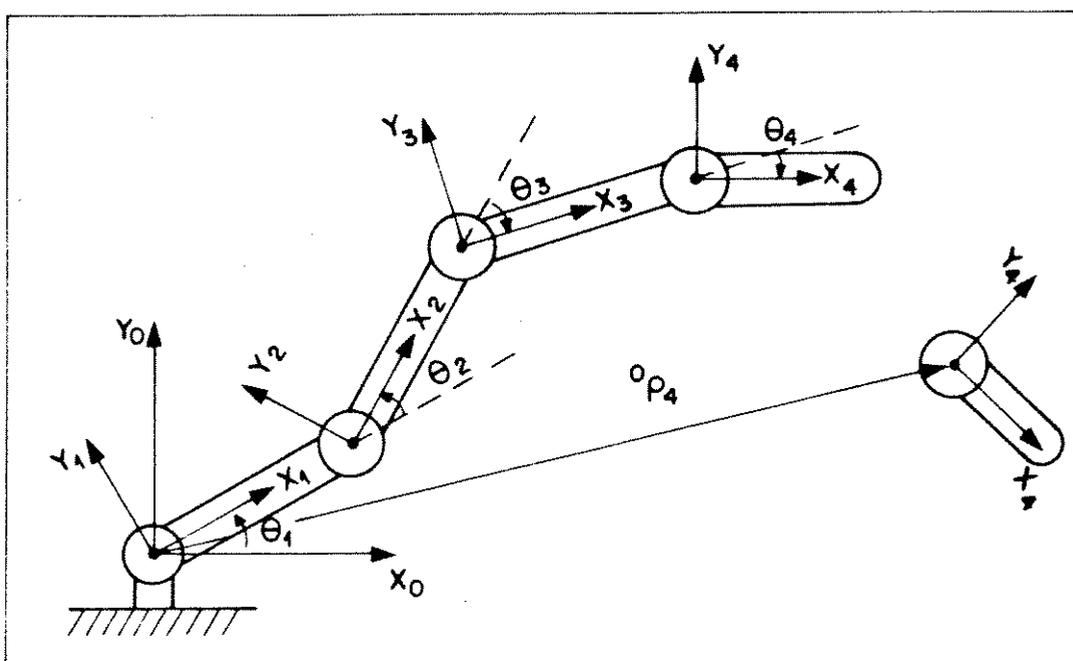


Fig. 4.3: Posições inicial e final de uma estrutura articulada.

Para a figura acima, consideremos completamente conhecida sua configuração inicial; ou seja, todos os parâmetros do objeto são conhecidos ($\alpha_i = 0$ e $d_i = 0$), inclusive as variáveis de junta: θ_1 , θ_2 , θ_3 e θ_4 .

Com relação à configuração final do objeto, ou seja, o estado do objeto após sua movimentação, consideremos conhecidas a posição e a orientação do último S.C. (fixo ao último segmento) em relação ao S.C. inercial {0}.

Dessa forma, sendo conhecidas a posição e a orientação final do último segmento (associado ao S.C. {4}) em relação ao S.C. inercial, {0}, podemos definir a transformação final que relaciona os respectivos S.C. .

$${}^0_T{}^4_{final} = \begin{bmatrix} \vec{n}^f & \vec{o}^f & \vec{a}^f & \vec{p}^f \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.4)$$

Analisando cuidadosamente a expressão (4.4), podemos concluir que, embora a matriz de transformação se encontre numericamente definida, cada um de seus elementos está relacionado a uma ou mais variáveis de junta.

Através das informações acima, podemos propor o seguinte problema cinemático inverso: Como determinar os valores para as variáveis de junta ($\theta_1, \theta_2, \theta_3$ e θ_4) que definem a configuração final do objeto ?

Para a solução deste tipo de problema propomos um procedimento iterativo, onde os valores das variáveis de junta são constantemente atualizados até que se chegue à solução do problema.

Partindo-se de uma estimativa inicial qualquer, podemos definir para o objeto em questão a transformação relativa à "k-ésima" iteração que relaciona o último S.C., (4), com o S.C. inercial, (0), como sendo:

$${}^0_4T^k = \begin{bmatrix} \vec{n}^k & \vec{o}^k & \vec{a}^k & \vec{p}^k \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.5)$$

Sendo o respectivo vetor de variáveis de junta dado por:

$$q^k = (\theta_1^k, \theta_2^k, \dots, \theta_n^k)^T \quad (4.6)$$

Podemos ainda definir uma relação entre as transformações ${}^0_4T^{final}$ e ${}^0_4T^k$ na forma:

$$r_1 = \vec{n}^k \cdot (\vec{p}^f - \vec{p}^k) \quad (4.7.a)$$

$$r_2 = \vec{o}^k \cdot (\vec{p}^f - \vec{p}^k) \quad (4.7.b)$$

$$r_3 = \vec{a}^k \cdot (\vec{p}^f - \vec{p}^k) \quad (4.7.c)$$

$$r_4 = (\vec{a}^k \cdot \vec{o}^f - \vec{a}^f \cdot \vec{o}^k)/2 \quad (4.7.d)$$

$$r_5 = (\vec{n}^k \cdot \vec{a}^f - \vec{n}^f \cdot \vec{a}^k)/2 \quad (4.7.e)$$

$$r_6 = (\vec{o}^k \cdot \vec{n}^f - \vec{o}^f \cdot \vec{n}^k)/2 \quad (4.7.f)$$

Sendo:

$$r(q) = (r_1, r_2, r_3, r_4, r_5, r_6) \quad (4.8)$$

conhecido como sendo o vetor residual da "k-ésima" iteração.

Quando $r(q) = 0$, o manipulador estará na posição e orientação final desejadas.

Segundo [Goldenberg (1985)], para chegarmos a $r(q) = 0$, o "vetor" "q" pode ser atualizado da seguinte forma:

$$q^{(k+1)} = q^{(k)} + \delta^{(k)} \quad (4.9)$$

sendo $\delta^{(k)} = [\delta_1, \delta_2, \dots, \delta_n]^T$ a solução para o sistema linear:

$$r_j[q^{(k)}] + \sum_{i=1}^n J_{ji}^{(k)} \cdot \delta_i^{(k)} = 0, \quad j = 1, \dots, 6 \quad (4.10)$$

e J_{ji} são os elementos da matriz Jacobiano, "J", para um objeto com "n" graus de liberdade, que é dada por:

$$J_{ji}^{(k)} = \frac{\partial r_j^{(k)}}{\partial \theta_i^{(k)}} \quad (4.11)$$

Notemos que, ao invés de associarmos a uma junta um grau de liberdade, vamos associar um S.C. a um grau de liberdade.

A solução deste problema geralmente depende da inversão da matriz Jacobiano. Entretanto, nem sempre o Jacobiano será uma matriz quadrada, o que o tornará não inversível. Neste caso, para que ocorra convergência no processo iterativo de cálculo do vetor de variáveis de estado, devemos calcular a pseudo-inversa do Jacobiano, J^+ , ao invés de sua inversa. Para uma matriz Jacobiano de dimensão $m \times n$ e posto (rank) p a sua pseudo-inversa será:

$$J^+ = \begin{cases} (J^T \cdot J)^{-1} \cdot J & ; \text{ se: } m > n = p \\ J^T \cdot (J \cdot J^T)^{-1} & ; \text{ se: } p = m < n \end{cases} \quad \text{ou} \quad (4.12)$$

Muitas vezes, o método de Newton-Raphson modificado é utilizado, ao invés do primeiro caso:

$$q^{(k+1)} = q^{(k)} + \lambda^{(k)} \cdot \delta^{(k)} \quad (4.13)$$

onde $\lambda^{(k)}$ é um número positivo escolhido de tal forma que a seguinte inequação seja satisfeita:

$$G[q^{(k+1)}] < G[q^{(k)}] \quad (4.14)$$

onde:

$$G(q) = \sum_{j=1}^6 [r_j(q)]^2 \quad (4.15)$$

Considere, agora, a fig. 4.4:

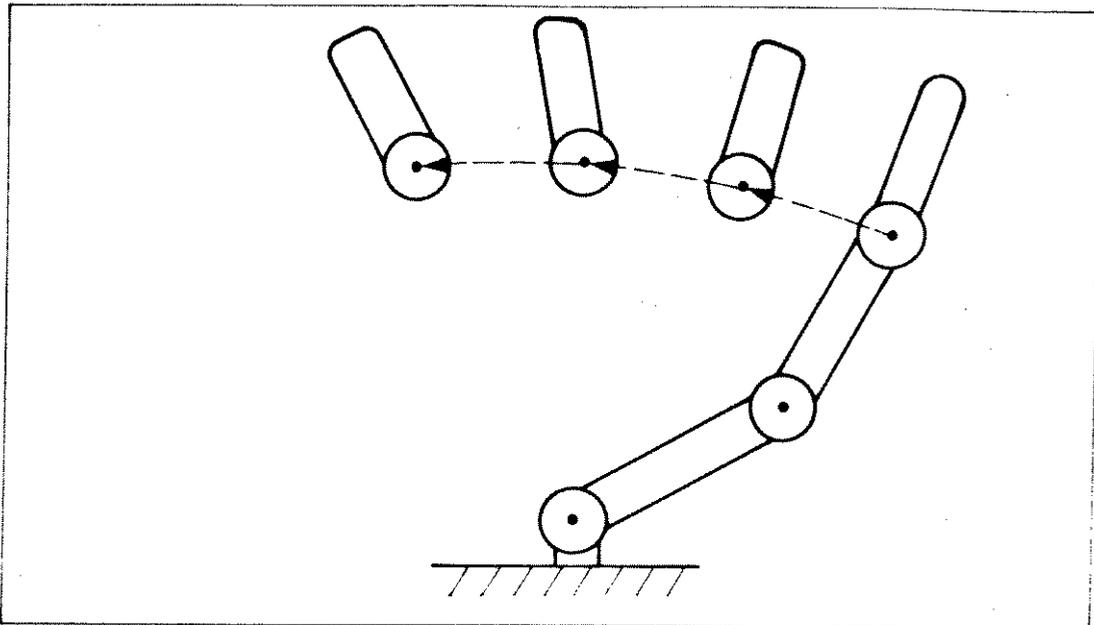


Fig. 4.4: Marcação da trajetória do objeto.

Definiremos a marcação da trajetória do objeto através da marcação de posições-chave de seu extremo livre. Ou seja, para cada quadro da sequência definimos a posição e a orientação do extremo livre¹ do manipulador; a seguir, aplicamos o algoritmo acima descrito para que sejam obtidas as soluções para as variáveis das juntas.

Notemos que, até o momento, estabelecemos as condições cinemáticas para a simulação de movimentos de objetos articulados, como é o caso do manipulador. Para que o sistema seja completo, resta-nos estabelecer um modelo dinâmico para o mesmo.

Embora já tenhamos desenvolvido um método para a geração de sequências animadas, também podemos estar interessados em descobrir quais forças ou torques são capazes de provocar os deslocamentos desejados. Dessa forma, faz-se necessário o desenvolvimento de um modelo dinâmico para o manipulador, para que possamos investigar quais forças ou torques no objeto produzem o movimento esperado.

A seção 10.3 desta dissertação apresenta um modelo para a implementação de um braço articulado (robô) com cinco graus de liberdade. O item 1 do apêndice descreve como utilizar a implementação deste modelo presente no disquete anexo.

¹ Consideramos o extremo livre de um objeto articulado como sendo o segmento mais extremo do mesmo, capaz de realizar alguma tarefa.

Por outro lado, a seção 10.4 desta dissertação apresenta um exemplo de animação cinemática "direta", onde as variáveis de junta do objeto são diretamente "acionadas" pelo programa. O item 2 do apêndice descreve como utilizar a implementação deste modelo, também presente no disquete anexo.

CAPÍTULO 5: MODELO DINÂMICO PARA O MANIPULADOR

Reconsidere o manipulador estudado no capítulo anterior. Como vimos, para esta estrutura rígida e articulada, apenas as variáveis das juntas " θ_i " estão sujeitas a variações durante a movimentação do objeto. Uma vez que a sequência de posições angulares para a junta "i" tenha sido gerada: $\theta_i^0 \rightarrow \theta_i^1 \rightarrow \theta_i^2 \rightarrow \dots \rightarrow \theta_i^m$, podemos calcular as respectivas velocidade e aceleração da junta, em relação ao sistema de coordenadas {i-1}, na forma:

$$\dot{\theta}_i^{(k)} = \frac{\theta_i^{(k)} - \theta_i^{(k-1)}}{\Delta t} \quad (5.1)$$

e

$$\ddot{\theta}_i^{(k)} = \frac{\theta_i^{(k+1)} - 2.\theta_i^{(k)} + \theta_i^{(k-1)}}{\Delta t^2} \quad (5.2)$$

sendo Δt o intervalo de tempo entre duas amostras consecutivas.

Dessa forma, para cada variável de junta θ_i , dispomos de uma sequência bem definida de posições, velocidades e acelerações.

Considere, agora, a fig. 5.1 e as definições dos itens seguintes:

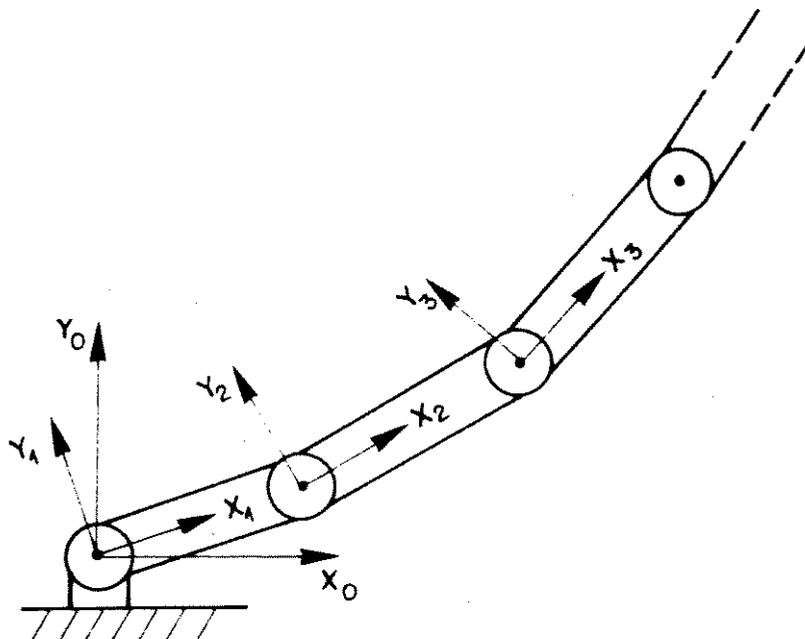


Fig. 5.1: Um manipulador e seus segmentos.

5.1 - PROPAGAÇÃO DE VELOCIDADES

A velocidade do segmento {i+1} será a velocidade do segmento {i} mais qualquer nova componente adicionada pela junta i+1. Dessa forma:

1-) A velocidade angular do S.C. preso ao segmento {i+1} é a mesma do segmento {i} mais uma nova componente causada pela velocidade angular da junta i+1. Seja:

${}^i\omega_{i+1}$ → a velocidade angular do S.C. {i+1} em relação ao S.C. inercial, porém representada no S.C. {i}.

${}^i\omega_i$ → a velocidade angular do S.C. {i} em relação ao S.C. inercial, porém representada no S.C. {i}.

Portanto, se a junta for rotacional¹:

$${}^i\omega_{i+1} = {}^i\omega_i + {}^{i+1}R \cdot \dot{\theta}_{i+1} \cdot {}^{i+1}\hat{Z}_{i+1} \quad (5.3.a)$$

$${}^{i+1}\omega_{i+1} = {}^{i+1}R \cdot {}^i\omega_i + \dot{\theta}_{i+1} \cdot {}^{i+1}\hat{Z}_{i+1} \quad (5.3.b)$$

onde:

$\dot{\theta}_{i+1}$ é um escalar e

$${}^{i+1}\hat{Z}_{i+1} = [0 \ 0 \ 1]^T$$

Se a junta i+1 for prismática²:

$${}^{i+1}\omega_{i+1} = {}^{i+1}R \cdot {}^i\omega_i \quad (5.3.d)$$

2-) A velocidade linear da origem do S.C. {i+1} é a mesma da origem do S.C. {i} mais uma nova componente causada pela velocidade do segmento {i}. Seja:

${}^iV_{i+1}$ → a velocidade linear da origem do S.C. {i+1} em relação ao S.C. inercial, porém representada no S.C. {i}.

iV_i → a velocidade linear da origem do S.C. {i} em relação ao S.C. inercial, porém representada no S.C. {i}.

¹Uma junta é considerada rotacional quando seu grau de liberdade gira ao redor do eixo z do sist. de coord. preso a ela. Todos as estruturas articuladas apresentadas nesta dissertação são compostas por juntas rotacionais.

²Uma junta é considerada prismática quando seu grau de liberdade desliza ao longo do eixo x do sist. de coord. preso a ela.

Portanto, se a junta for rotacional:

$${}^i\mathbf{v}_{i+1} = {}^i\mathbf{v}_i + {}^i\boldsymbol{\omega}_i \times {}^i\mathbf{P}_{i+1} \quad (5.4.a)$$

$${}^{i+1}\mathbf{v}_{i+1} = {}^{i+1}\mathbf{R} \cdot \left[{}^i\mathbf{v}_i + {}^i\boldsymbol{\omega}_i \times {}^i\mathbf{P}_{i+1} \right] \quad (5.4.b)$$

Se a junta for prismática:

$${}^{i+1}\mathbf{v}_{i+1} = {}^{i+1}\mathbf{R} \cdot \left[{}^i\mathbf{v}_i + {}^i\boldsymbol{\omega}_i \times {}^i\mathbf{P}_{i+1} \right] + \dot{d}_{i+1} \cdot {}^{i+1}\hat{\mathbf{z}}_{i+1} \quad (5.4.c)$$

5.2 - PROPAGAÇÃO DE ACELERAÇÕES

1-) A aceleração angular, para uma junta rotacional, pode ser expressa por:

$${}^{i+1}\dot{\boldsymbol{\omega}}_{i+1} = {}^{i+1}\mathbf{R} \cdot {}^i\dot{\boldsymbol{\omega}}_i + {}^{i+1}\mathbf{R} \cdot {}^i\boldsymbol{\omega}_i \times \dot{\theta}_{i+1} \cdot {}^{i+1}\hat{\mathbf{z}}_{i+1} + \ddot{\theta}_{i+1} \cdot {}^{i+1}\hat{\mathbf{z}}_{i+1} \quad (5.5.a)$$

Se a junta i+1 for prismática:

$${}^{i+1}\dot{\boldsymbol{\omega}}_{i+1} = {}^{i+1}\mathbf{R} \cdot {}^i\dot{\boldsymbol{\omega}}_i \quad (5.5.b)$$

Ou seja, a aceleração angular da junta "i+1" será igual à aceleração angular da junta "i" mais a aceleração provocada pela variação da componente de junta ($\dot{\theta}_{i+1}$).

2-) A aceleração linear, para uma junta rotacional, pode ser expressa por:

$${}^{i+1}\dot{\mathbf{v}}_{i+1} = {}^{i+1}\mathbf{R} \cdot \left[({}^i\dot{\boldsymbol{\omega}}_i \times {}^i\mathbf{P}_{i+1}) + {}^i\boldsymbol{\omega}_i \times ({}^i\boldsymbol{\omega}_i \times {}^i\mathbf{P}_{i+1}) + {}^i\dot{\mathbf{v}}_i \right] \quad (5.6.a)$$

Se a junta i+1 for prismática:

$${}^{i+1}\dot{\mathbf{v}}_{i+1} = {}^{i+1}\mathbf{R} \cdot \left[({}^i\dot{\boldsymbol{\omega}}_i \times {}^i\mathbf{P}_{i+1}) + {}^i\boldsymbol{\omega}_i \times ({}^i\boldsymbol{\omega}_i \times {}^i\mathbf{P}_{i+1}) + {}^i\dot{\mathbf{v}}_i \right] + 2 \cdot {}^{i+1}\boldsymbol{\omega}_{i+1} \times \dot{d}_{i+1} \cdot {}^{i+1}\hat{\mathbf{z}}_{i+1} + \ddot{d}_{i+1} \cdot {}^{i+1}\hat{\mathbf{z}}_{i+1} \quad (5.6.b)$$

Portanto, a velocidade linear da junta "i+1" é igual à velocidade linear da junta "i" mais as componentes decorrentes da velocidade angular da junta "i" e, se a junta "i+1" é prismática, a componente decorrente da variação do parâmetro "d" da junta "i+1".

Podemos definir, desde já, a aceleração linear do centro de massa de cada segmento, que será utilizada pelo algoritmo da seção 5.5, através da equação:

$${}^i\dot{v}_c = {}^i\dot{\omega}_i \times {}^iP_c + {}^i\omega_i \times ({}^i\omega_i \times {}^iP_c) + {}^i\dot{v}_i \quad (5.7)$$

onde, devemos imaginar um S.C. {C} com sua origem presa ao centro de massa do objeto (segmento) "i" e com a mesma orientação do S.C. {i}; sendo iP_c a distância entre as origens dos S.C. {i} e {C}.

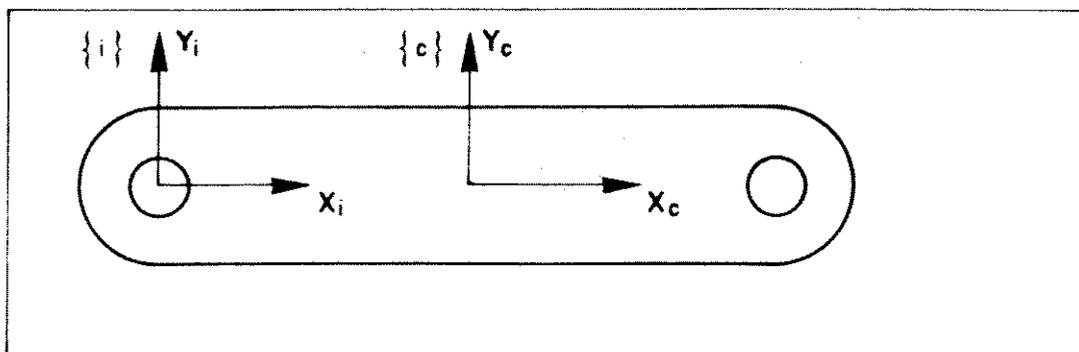


Fig. 5.2: Um segmento de um manipulador e seus sistemas de coordenadas.

5.3 - TENSOR DE INÉRCIA / MATRIZ INERCIAL DE UM OBJETO RÍGIDO

Considere o objeto da fig. 5.3, que apresenta densidade uniforme:

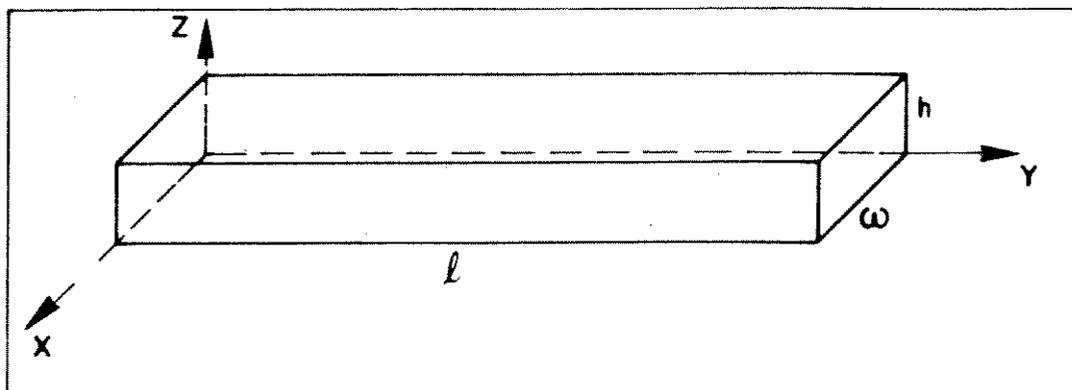


Fig. 5.3: Parâmetros dimensionais de um objeto rígido.

O tensor de inércia constitui uma importante informação pois define a distribuição da massa de um objeto rígido em relação a um S.C. Em nosso desenvolvimento, o S.C. adotado estará sempre preso ao objeto. Para o exemplo da fig. 5.3 seja a matriz inercial do objeto dada por [Craig (1989)]:

$${}^A I = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix} \quad (5.8)$$

onde:

$$I_{xx} = \int \int \int_V (y^2 + z^2) \cdot \rho \, dv \quad (5.9.a)$$

$$I_{yy} = \int \int \int_V (x^2 + z^2) \cdot \rho \, dv \quad (5.9.b)$$

$$I_{zz} = \int \int \int_V (x^2 + y^2) \cdot \rho \, dv \quad (5.9.c)$$

$$I_{xy} = \int \int \int_V x \cdot y \cdot \rho \, dv \quad (5.9.d)$$

$$I_{xz} = \int \int \int_V x \cdot z \cdot \rho \, dv \quad (5.9.e)$$

$$I_{yz} = \int \int \int_V y \cdot z \cdot \rho \, dv \quad (5.9.f)$$

sendo que:

- cada ponto dentro do objeto é localizado através de um vetor do tipo: ${}^A P = [x, y, z]^T$;
- a densidade volumétrica do objeto é $\rho = dm/dv$ e
- $dv = dx dy dz$.

O tensor de inércia para o objeto da fig. 5.3 é dado por:

$$I_{xx} = \int_0^h \int_0^l \int_0^w (y^2 + z^2) \cdot \rho \, dx dy dz = \left[\frac{l^2}{3} + \frac{h^2}{3} \right] \cdot h \cdot l \cdot w \cdot \rho = \frac{m}{3} \cdot (l^2 + h^2)$$

Analogamente, para I_{yy} e I_{zz} teremos:

$$I_{yy} = \frac{m}{3} \cdot (w^2 + h^2) \quad e \quad I_{zz} = \frac{m}{3} \cdot (l^2 + w^2)$$

Calculando para I_{xy} chegaremos a:

$$I_{xy} = \int_0^h \int_0^l \int_0^w x \cdot y \cdot \rho \, dx dy dz = \frac{w \cdot l \cdot (w \cdot l \cdot h \cdot \rho)}{4} = \frac{m \cdot w \cdot l}{4}$$

Analogamente, para I_{xz} e I_{yz} teremos:

$$I_{xz} = \frac{m \cdot h \cdot w}{4} \quad \text{e} \quad I_{yz} = \frac{m \cdot h \cdot l}{4}$$

Portanto, o tensor de inércia do objeto é:

$$A_I = \begin{bmatrix} m \cdot (l^2 + h^2)/3 & -m \cdot w \cdot l/4 & -m \cdot h \cdot w/4 \\ -m \cdot w \cdot l/4 & m \cdot (w^2 + h^2)/3 & -m \cdot h \cdot l/4 \\ -m \cdot h \cdot w/4 & -m \cdot h \cdot l/4 & m \cdot (l^2 + w^2)/3 \end{bmatrix}$$

5.3.1 - TEOREMA DOS EIXOS PARALELOS

O teorema dos eixos paralelos relaciona o tensor de inércia com um S.C. cuja origem se encontra no centro de massa do objeto e cuja orientação coincide com um S.C. previamente preso ao objeto. Reconsidere a fig. 5.2. Seja ${}^A P_c = (x_c, y_c, z_c)^T$ um vetor que localiza o centro de massa do objeto em relação ao S.C. {A}. Então:

$$A_{I_{zz}} = C_{I_{zz}} + m \cdot (x_c^2 + y_c^2) \quad (5.10.a)$$

$$A_{I_{xy}} = C_{I_{xy}} - m \cdot x_c \cdot y_c \quad (5.10.b)$$

sendo os demais produtos e momentos de inércia calculados permutando-se x , y e z nas equações (5.10.a) e (5.10.b).

O Teorema dos Eixos Paralelos pode ser formalizado em uma forma matricial como sendo:

$$A_I = C_I + m \cdot [({}^A P_c)^T \cdot A_{P_c} \cdot I_3 - A_{P_c} \cdot ({}^A P_c)^T] \quad (5.11)$$

sendo I_3 uma matriz identidade 3×3 .

Para o exemplo da fig. 5.3 temos:

$$C_I = \begin{bmatrix} m.(h^2 + l^2)/12 & 0 & 0 \\ 0 & m.(w^2 + h^2)/12 & 0 \\ 0 & 0 & m.(l^2 + w^2) \end{bmatrix}$$

Tanto em robótica, quanto em "animação", é comum gerarmos previamente a trajetória do objeto (θ , $\dot{\theta}$ e $\ddot{\theta}$) segundo o modelo cinemático exposto no capítulo anterior e, posteriormente, segundo o modelo dinâmico iterativo de Newton-Euler por exemplo [Craig (1989)], calcularmos o torque necessário nas juntas para que o deslocamento desejado seja efetivamente obtido.

5.4 - MODELO DINÂMICO ITERATIVO DE NEWTON-EULER

5.4.1 - EQUAÇÃO DE NEWTON

Para um corpo de massa total m , cujo centro de massa está se acelerando com aceleração igual a \dot{v}_c , a força equivalente F , atuando em seu centro de massa, será dada por:

$$F = m \cdot \dot{v}_c \quad (5.12)$$

5.4.2 - EQUAÇÃO DE EULER

Se um corpo (como o da fig. 5.3, por exemplo) encontra-se em rotação com velocidade angular ω e aceleração angular $\dot{\omega}$, então, o momento N que está atuando no corpo para causar este tipo de movimento será dado por:

$$N = C_I \cdot \dot{\omega} + \omega \times C_I \cdot \omega \quad (5.13)$$

Desta forma, para um objeto articulado, podemos calcular as forças e torques inerciais que atuam nos respectivos centros de massa de cada segmento aplicando as equações do modelo Newton-Euler. A força e o torque, respectivamente, para um segmento "i" são dadas por:

$$F_i = m \cdot \dot{v}_{c_i} \quad (5.14.a)$$

$$N_i = C_{I_i} \cdot \dot{\omega}_i + \omega_i \times C_{I_i} \cdot \omega_i \quad (5.14.b)$$

onde $\{C_i\}$ possui sua origem no centro de massa do segmento "i", além de possuir a mesma orientação do S.C. $\{i\}$ do próprio segmento.

Seja:

${}^i f_i \rightarrow$ a força exercida na junta do segmento "i" pelo segmento "i-1" representada no S.C. {i}.

${}^i n_i \rightarrow$ o torque exercido na junta do segmento "i" pelo segmento "i-1" representado no S.C. {i}.

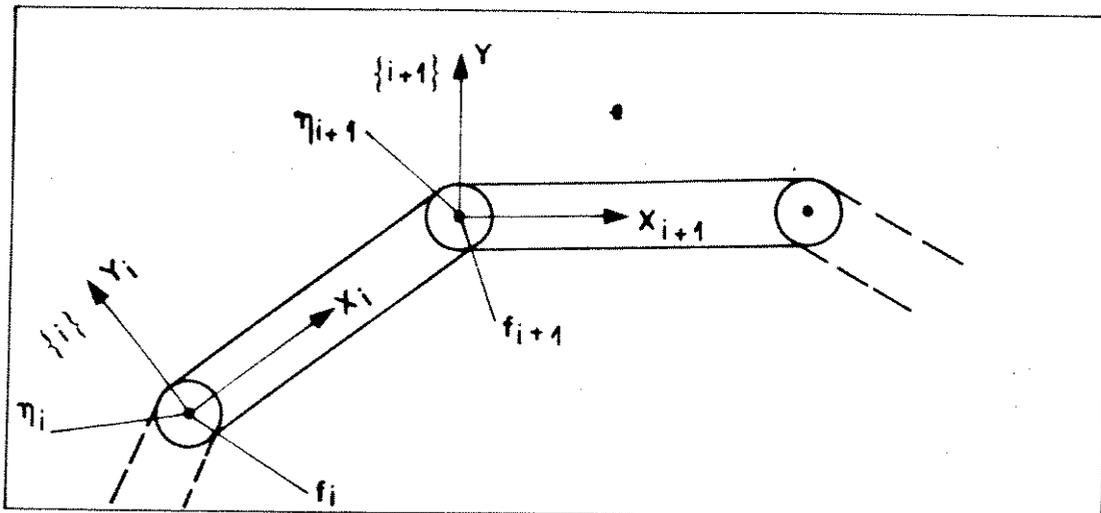


Fig. 5.4: Forças e torques aplicados a um objeto articulado.

Podemos então reescrever as equações de Newton-Euler para que as forças e torques sejam calculadas de forma iterativa entre segmentos de um manipulador:

$${}^i f_i = {}^i F_i + {}_{i+1}^i R \cdot {}^{i+1} f_{i+1} \quad (5.15.a)$$

$${}^i n_i = {}^i N_i + {}_{i+1}^i R \cdot {}^{i+1} n_{i+1} + {}^i P_c \times {}^i F_i + {}^i P_{i+1} \times {}_{i+1}^i R \cdot {}^{i+1} f_{i+1} \quad (5.15.b)$$

Notemos que as iterações ocorrem a partir do extremo do manipulador, indo em direção à sua base.

Finalmente, o torque que o atuador (motor) da junta "i" deve fornecer para que o movimento desejado seja realizado será dado por:

$$\tau_i = ({}^i n_i)^T \cdot {}^i \hat{Z}_i, \text{ se a junta "i+1" for rotacional} \quad (5.16.a)$$

$$\tau_i = ({}^i f_i)^T \cdot {}^i \hat{Z}_i, \text{ se a junta "i+1" for prismática} \quad (5.16.b)$$

5.5 - ALGORITMO ITERATIVO DE NEWTON-EULER

Para um robô (manipulador) de n segmentos, FAÇA:

1-) A partir do S.C. inercial, {0}, até o S.C. do último segmento, {n}, CALCULE:

$${}^{i+1} \omega_{i+1} = {}^{i+1} R_i \cdot {}^i \omega_i + \dot{\theta}_{i+1} \cdot {}^{i+1} \hat{Z}_{i+1}$$

$${}^{i+1} \dot{\omega}_{i+1} = {}^{i+1} R_i \cdot {}^i \dot{\omega}_i + {}^{i+1} R_i \cdot {}^i \omega_i \times \dot{\theta}_{i+1} \cdot {}^{i+1} \hat{Z}_{i+1} + \ddot{\theta}_{i+1} \cdot {}^{i+1} \hat{Z}_{i+1}$$

$${}^{i+1} \dot{v}_{i+1} = {}^{i+1} R_i \cdot [{}^i \dot{\omega}_i \times {}^i P_{i+1} + {}^i \omega_i \times ({}^i \omega_i \times {}^i P_{i+1}) + {}^i \dot{v}_i]$$

$${}^{i+1} \dot{v}_{c_{i+1}} = {}^{i+1} \dot{\omega}_{i+1} \times {}^{i+1} P_{c_{i+1}} + {}^{i+1} \omega_{i+1} \times ({}^{i+1} \omega_{i+1} \times {}^{i+1} P_{c_{i+1}}) + {}^{i+1} \dot{v}_{i+1}$$

$${}^{i+1} F_{i+1} = m_{i+1} \cdot {}^{i+1} \dot{v}_{c_{i+1}}$$

$${}^{i+1} N_{i+1} = C(i+1)_{I_{i+1}} \cdot {}^{i+1} \omega_{i+1} + {}^{i+1} \omega_{i+1} \times C(i+1)_{I_{i+1}} \cdot {}^{i+1} \omega_{i+1}$$

2-) A partir da extremidade do robô (manipulador), até o S.C. do primeiro segmento, {1}, CALCULE:

$${}^i f_i = {}^i F_i + {}^{i+1} R_i \cdot {}^{i+1} f_{i+1}$$

$${}^i n_i = {}^i N_i + {}^{i+1} R_i \cdot {}^{i+1} n_{i+1} + {}^i P_{c_i} \times {}^i F_i + {}^i P_{i+1} \times {}^{i+1} R_i \cdot {}^{i+1} f_{i+1}$$

$$\tau_i = ({}^i n_i)^T \cdot {}^i \hat{Z}_i$$

O conjunto de equações do algoritmo acima é adequado para um objeto composto apenas por juntas rotacionais. Devem ser substituídas as respectivas equações para o caso de juntas prismáticas.

5.5.1 - ESTABELECENDO CONDIÇÕES INICIAIS PARA O ALGORITMO

Estando o S.C. inercial "parado", temos que:

$${}^0\omega_0 = {}^0\dot{\omega}_0 = [0, 0, 0]^T$$

A inclusão de um "termo de gravidade" no sistema nos leva a:

$${}^0\dot{v}_0 = [0, g, 0]^T, \quad g = \text{gravidade local.}$$

Se existe uma força f_e e um torque τ_e sendo aplicados ao extremo do robô, então:

$${}^{n+1}f_{n+1} = f_e$$

$${}^{n+1}\tau_{n+1} = \tau_e$$

Se a extremidade do robô encontra-se livre no espaço, então:

$${}^{n+1}f_{n+1} = {}^{n+1}\tau_{n+1} = 0$$

Podemos notar que, até o momento, nos limitamos ao estudo de objetos articulados compostos por uma sequência não-ramificada de segmentos. Dessa forma, trataremos, no capítulo 6, da classe dos objetos articulados ramificados.

CAPÍTULO 6: MODELO DINÂMICO PARA ESTRUTURAS ARTICULADAS RAMIFICADAS

Nos capítulos 4 e 5 tratamos de estruturas articuladas que são caracterizadas por uma **cadeia de segmentos**. Entretanto, uma grande variedade de objetos articulados admite **cadeias ramificadas**, caracterizando estruturas tipo **árvore**. Um exemplo bastante comum seria o esqueleto humano. [Armstrong (1985)] foi um dos pioneiros a tratar este tipo de problema aplicado à Computação Gráfica. Ainda na segunda metade da década de 80, foram propostas soluções alternativas de forma mais genérica em [Isaacs (1987)] e [Isaacs (1988)].

O modelo dinâmico apresentado neste capítulo foi desenvolvido por [Wilhems (1988)] de modo a ser aplicável a estruturas articuladas ramificadas ou não. Este tipo de abordagem para a solução do problema de controle de movimento de uma estrutura tipo **árvore** nos leva a um algoritmo iterativo que pode ser implementado em computador de maneira relativamente fácil.

Considere a estrutura articulada em árvore apresentada na fig. 6.1.

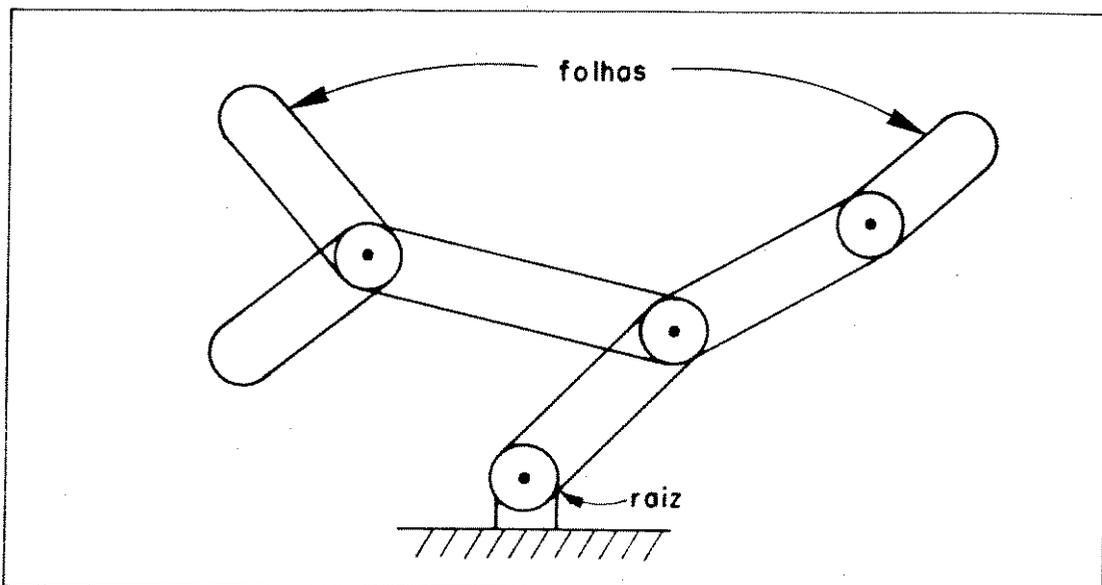


Fig. 6.1: Exemplo de estrutura tipo árvore.

Uma estrutura tipo **árvore** consiste, então, de um objeto articulado cujas juntas podem conectar mais de dois segmentos. Da mesma forma que o manipulador estudado nos capítulos 4 e 5, associamos a cada segmento um S.C. que ficará fixo no seu respectivo segmento.

A **raiz** da estrutura será escolhida arbitrariamente e será considerada uma referência para o objeto. As **folhas** da árvore são os segmentos extremos da estrutura.

Temos que, para dois segmentos adjacentes, será chamado de **pai** aquele que estiver mais próximo da raiz, sendo o segundo chamado de **filho**. Deve ficar claro desde já que um segmento não pode ter dois pais.

Para a análise das estruturas tipo **árvore** podemos utilizar a formulação dinâmica de Euler, conforme apresentado em [Wilhems (1988)] e [Craig (1989)].

6.1 - FORMULAÇÃO DINÂMICA DE EULER

Considere o objeto rígido da fig. 6.2. As equações vetoriais de Euler que descrevem o movimento deste objeto no espaço tridimensional são:

$$F = m \cdot [a + \dot{\omega} \times P_c + \omega \times (\omega \times P_c)] \quad (6.1.a)$$

$$N = I \cdot \dot{\omega} + m \cdot P_c \times a + \omega \times I \cdot \omega \quad (6.1.b)$$

onde:

"m" representa a massa do objeto,

"P_c" é um vetor que vai da origem do S.C. fixo no objeto até o centro de massa do mesmo,

"F" representa a soma das forças que atuam no centro de massa do objeto,

"N" representa a soma dos torques que atuam no centro de massa do objeto,

"I" é a matriz tensor de inércia (3×3) do objeto relativa ao seu centro de massa,

"a" é o vetor aceleração linear do corpo,

"ω" é o vetor velocidade angular do corpo e

" $\dot{\omega}$ " é o vetor aceleração angular do objeto.

Sendo que, por conveniência, os valores acima devem ser medidos em relação a um sistema de coordenadas fixo no corpo.

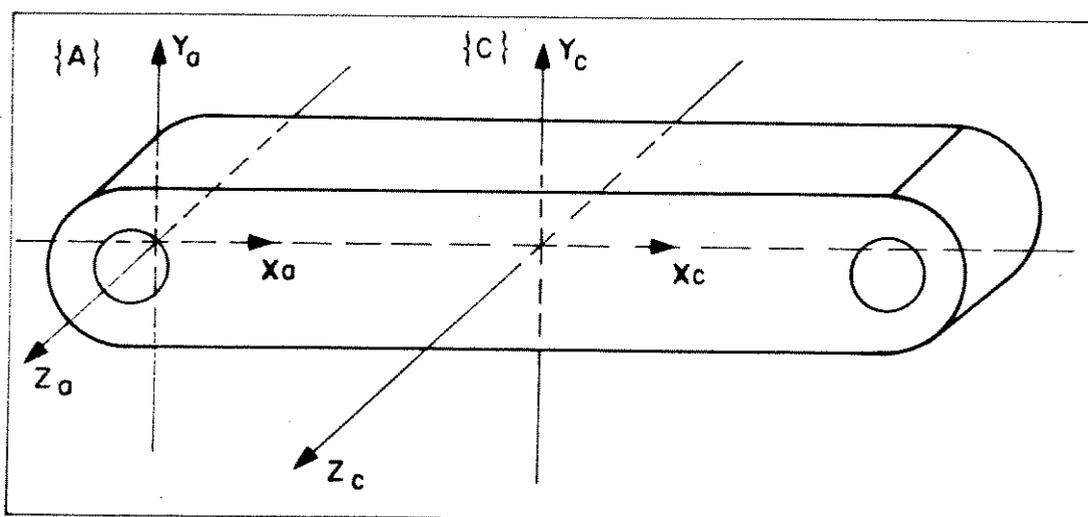


Fig. 6.2: Um segmento de uma estrutura tipo árvore.

Em um objeto articulado, para que dois segmentos adjacentes (pai e filho) sejam mantidos unidos através de uma junta, devemos considerar um conjunto de equações de restrição além das equações de Euler para cada conjunto pai-filho:

$$f_p = F_p + {}^P_f R \cdot f_f \quad (6.2.a)$$

$$\eta_p = N_p + {}^P_f R \cdot \eta_f + {}^P P_f \times {}^P_f R \cdot f_f \quad (6.2.b)$$

onde:

" f_p " é a força aplicada na origem do S.C. preso ao segmento "pai",

" f_f " é a força aplicada na origem do S.C. preso ao segmento "filho",

" η_p " é o torque aplicado na origem do S.C. preso ao segmento "pai",

" η_f " é o torque aplicado na origem do S.C. preso ao segmento "filho"

" ${}^P_f R$ " é a matriz de orientação existente entre os segmentos "pai" e "filho" e

" ${}^P P_f$ " é o vetor que vai da origem do S.C. preso ao segmento "pai" até a origem do S.C. preso ao segmento "filho".

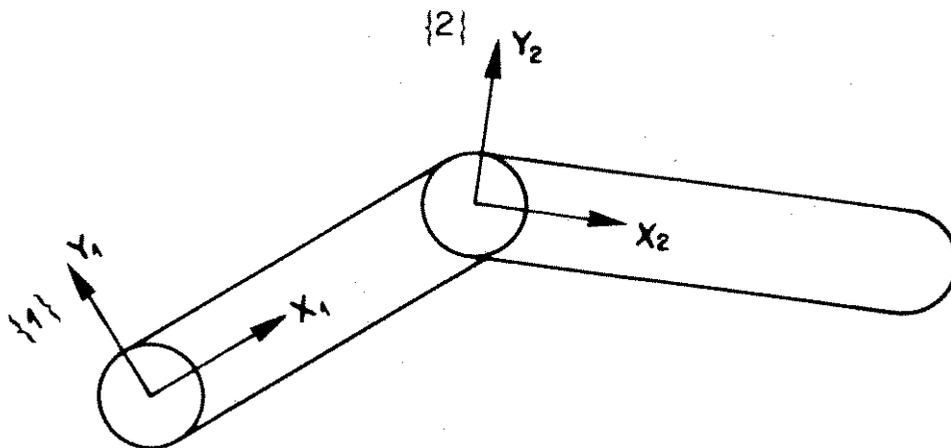


Fig. 6.3: Dois segmentos unidos através de uma junta rotacional.

Dessa forma, para o objeto da fig. 6.3 temos:

$$F_1 = m_1 \cdot [a_1 + \dot{\omega}_1 \times Pc_1 + \omega_1 \times (\omega_1 \times Pc_1)] \quad (6.3.a)$$

$$N_1 = I_1 \cdot \dot{\omega}_1 + m_1 \cdot Pc_1 \times a_1 + \omega_1 \times I_1 \cdot \omega_1 \quad (6.3.b)$$

$$F_2 = m_2 \cdot [a_2 + \dot{\omega}_2 \times Pc_2 + \omega_2 \times (\omega_2 \times Pc_2)] \quad (6.3.c)$$

$$N_2 = I_2 \cdot \dot{\omega}_2 + m_2 \cdot Pc_2 \times a_2 + \omega_2 \times I_2 \cdot \omega_2 \quad (6.3.e)$$

$$f_1 = F_1 + {}^1_2R \cdot f_2 \quad (6.3.e)$$

$$\eta_1 = N_1 + {}^1_2R \cdot \eta_2 + {}^1P_2 \times {}^1_2R \cdot f_2 \quad (6.3.f)$$

Supondo que as forças e torques externos são aplicados nas origens dos S.C. presos aos segmentos 1 e 2, e considerando a existência de apenas dois segmentos, podemos ainda escrever que:

$$F_1 = f_1 - {}^1_2R \cdot f_2 \quad (6.4.a)$$

$$N_1 = \eta_1 - {}^1_2R \cdot \eta_2 - {}^1P_2 \times {}^1_2R \cdot f_2 \quad (6.4.b)$$

$$F_2 = f_2 \quad (6.4.c)$$

$$N_2 = \eta_2 \quad (6.4.d)$$

Os conjuntos de equações (6.3) e (6.4) podem ser reescritos na seguinte forma matricial:

$$\begin{bmatrix} a_1 \\ \dot{\omega}_1 \\ a_2 \\ \dot{\omega}_2 \end{bmatrix} = \begin{bmatrix} m_1 & -Pc_1 \times & 0 & 0 \\ m_1 \cdot Pc_1 \times & I_1 & 0 & 0 \\ 0 & 0 & m_2 & -Pc_2 \times \\ 0 & 0 & m_2 \cdot Pc_2 \times & I_2 \end{bmatrix}^{-1} \begin{bmatrix} f_1 - {}^1_2R \cdot f_2 - \omega_1 \times (\omega_1 \times Pc_1) \\ \eta_1 - {}^1_2R \cdot \eta_2 - {}^1P_2 \times {}^1_2R \cdot f_2 - \omega_1 \times I_1 \cdot \omega_1 \\ f_2 - \omega_2 \times (\omega_2 \times Pc_2) \\ \eta_2 - \omega_2 \times I_2 \cdot \omega_2 \end{bmatrix} \quad (6.5)$$

que expandida resulta em:

$$\begin{bmatrix} a_{1x} \\ a_{1y} \\ a_{1z} \\ \dot{\omega}_{1x} \\ \dot{\omega}_{1y} \\ \dot{\omega}_{1z} \\ a_{2x} \\ a_{2y} \\ a_{2z} \\ \dot{\omega}_{2x} \\ \dot{\omega}_{2y} \\ \dot{\omega}_{2z} \end{bmatrix} = \begin{bmatrix} m_1 & 0 & 0 & 0 & P_{c1z} & -P_{c1y} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & m_1 & 0 & -P_{c1z} & 0 & P_{c1x} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & m_1 & P_{c1y} & -P_{c1x} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -m.P_{c1z} & m.P_{c1y} & I_{1xx} & -I_{1xy} & -I_{1xz} & 0 & 0 & 0 & 0 & 0 & 0 \\ m.P_{c1z} & 0 & -m.P_{c1x} & -I_{1xy} & I_{1yy} & -I_{1yz} & 0 & 0 & 0 & 0 & 0 & 0 \\ -m.P_{c1y} & m.P_{c1x} & 0 & -I_{1xz} & -I_{1yz} & I_{1zz} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & m_2 & 0 & 0 & 0 & P_{c2z} & -P_{c2y} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & m_2 & 0 & -P_{c2z} & 0 & P_{c2x} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & m_2 & P_{c2y} & -P_{c2x} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -m.P_{c2z} & m.P_{c2y} & I_{2xx} & -I_{2xy} & -I_{2xz} \\ 0 & 0 & 0 & 0 & 0 & 0 & m.P_{c2z} & 0 & -m.P_{c2x} & -I_{2xy} & I_{2yy} & -I_{2yz} \\ 0 & 0 & 0 & 0 & 0 & 0 & -m.P_{c2y} & m.P_{c2x} & 0 & -I_{2xz} & -I_{2yz} & I_{2z} \end{bmatrix}^{-1}$$

$$f_{1x}^{-1}R_{11} \cdot f_{2x}^{-1}R_{12} \cdot f_{2y}^{-1}R_{13} \cdot f_{2z}^{-1}P_{1y} \cdot \omega_{1x} \cdot \omega_{1y} \cdot Pc_{1y} + \omega_{1y}^2 \cdot Pc_{1x} + \omega_{1z}^2 \cdot Pc_{1x} - \omega_{1x} \cdot \omega_{1z} \cdot Pc_{1z}$$

$$f_{1y}^{-1}R_{21} \cdot f_{2x}^{-1}R_{22} \cdot f_{2y}^{-1}R_{23} \cdot f_{2z}^{-1}P_{1z} \cdot \omega_{1y} \cdot \omega_{1z} \cdot Pc_{1z} + \omega_{1z}^2 \cdot Pc_{1y} + \omega_{1x}^2 \cdot Pc_{1y} - \omega_{1x} \cdot \omega_{1y} \cdot Pc_{1x}$$

$$f_{1z}^{-1}R_{31} \cdot f_{2x}^{-1}R_{32} \cdot f_{2y}^{-1}R_{33} \cdot f_{2z}^{-1}P_{1x} \cdot \omega_{1x} \cdot \omega_{1z} \cdot Pc_{1x} + \omega_{1x}^2 \cdot Pc_{1z} + \omega_{1y}^2 \cdot Pc_{1z} - \omega_{1y} \cdot \omega_{1z} \cdot Pc_{1y}$$

$$\eta_{1x}^{-1}R_{11} \cdot \eta_{2x}^{-1}R_{12} \cdot \eta_{2y}^{-1}R_{13} \cdot \eta_{2z}^{-1}P_{2y} \cdot ({}^1R_{31} \cdot f_{1x} + {}^1R_{32} \cdot f_{1y} + {}^1R_{33} \cdot f_{1z}) +$$

$$+ {}^1P_{2z} \cdot ({}^1R_{21} \cdot f_{1x} + {}^1R_{22} \cdot f_{1y} + {}^1R_{23} \cdot f_{1z}) - \omega_{1y} \cdot (-I_{1xz} \cdot \omega_{1x} - I_{1yz} \cdot \omega_{1y} + I_{1zz} \cdot \omega_{1z}) +$$

$$+ \omega_{1z} \cdot (-I_{1xy} \cdot \omega_{1x} + I_{1yy} \cdot \omega_{1y} - I_{1yz} \cdot \omega_{1z})$$

$$\eta_{1y}^{-1}R_{21} \cdot \eta_{2x}^{-1}R_{22} \cdot \eta_{2y}^{-1}R_{23} \cdot \eta_{2z}^{-1}P_{2z} \cdot ({}^1R_{11} \cdot f_{1x} + {}^1R_{12} \cdot f_{1y} + {}^1R_{13} \cdot f_{1z}) +$$

$$+ {}^1P_{2x} \cdot ({}^1R_{31} \cdot f_{1x} + {}^1R_{32} \cdot f_{1y} + {}^1R_{33} \cdot f_{1z}) - \omega_{1z} \cdot (-I_{1xx} \cdot \omega_{1x} - I_{1xy} \cdot \omega_{1y} + I_{1xz} \cdot \omega_{1z}) +$$

$$+ \omega_{1x} \cdot (-I_{1xz} \cdot \omega_{1x} + I_{1yz} \cdot \omega_{1y} - I_{1zz} \cdot \omega_{1z})$$

$$\eta_{1z}^{-1}R_{31} \cdot \eta_{2x}^{-1}R_{32} \cdot \eta_{2y}^{-1}R_{33} \cdot \eta_{2z}^{-1}P_{2x} \cdot ({}^1R_{21} \cdot f_{1x} + {}^1R_{22} \cdot f_{1y} + {}^1R_{23} \cdot f_{1z}) +$$

$$+ {}^1P_{2y} \cdot ({}^1R_{11} \cdot f_{1x} + {}^1R_{12} \cdot f_{1y} + {}^1R_{13} \cdot f_{1z}) - \omega_{1x} \cdot (-I_{1xy} \cdot \omega_{1x} - I_{1yy} \cdot \omega_{1y} + I_{1yz} \cdot \omega_{1z}) +$$

$$+ \omega_{1y} \cdot (-I_{1xx} \cdot \omega_{1x} + I_{1xy} \cdot \omega_{1y} - I_{1xz} \cdot \omega_{1z})$$

$$f_{2x}^{-1}P_{2x} \cdot \omega_{2y} \cdot Pc_{2y} + \omega_{2y}^2 \cdot Pc_{2x} + \omega_{2z}^2 \cdot Pc_{2x} - \omega_{2x} \cdot \omega_{2z} \cdot Pc_{2z}$$

$$f_{2y}^{-1}P_{2y} \cdot \omega_{2z} \cdot Pc_{2z} + \omega_{2z}^2 \cdot Pc_{2y} + \omega_{2x}^2 \cdot Pc_{2y} - \omega_{2y} \cdot \omega_{2x} \cdot Pc_{2x}$$

$$f_{2z}^{-1}P_{2z} \cdot \omega_{2x} \cdot Pc_{2x} + \omega_{2x}^2 \cdot Pc_{2z} + \omega_{2y}^2 \cdot Pc_{2z} - \omega_{2z} \cdot \omega_{2y} \cdot Pc_{2y}$$

$$\eta_{2x}^{-1}P_{2y} \cdot (-I_{2xz} \cdot \omega_{2x} - I_{2yz} \cdot \omega_{2y} + I_{2zz} \cdot \omega_{2z}) + \omega_{2z} \cdot (-I_{2xy} \cdot \omega_{2x} + I_{2yy} \cdot \omega_{2y} - I_{2yz} \cdot \omega_{2z})$$

$$\eta_{2y}^{-1}P_{2z} \cdot (-I_{2xx} \cdot \omega_{2x} - I_{2xy} \cdot \omega_{2y} + I_{2xz} \cdot \omega_{2z}) + \omega_{2x} \cdot (-I_{2xz} \cdot \omega_{2x} + I_{2yz} \cdot \omega_{2y} - I_{2zz} \cdot \omega_{2z})$$

$$\eta_{2z}^{-1}P_{2x} \cdot (-I_{2xy} \cdot \omega_{2x} - I_{2yy} \cdot \omega_{2y} + I_{2yz} \cdot \omega_{2z}) + \omega_{2y} \cdot (-I_{2xx} \cdot \omega_{2x} + I_{2xy} \cdot \omega_{2y} - I_{2xz} \cdot \omega_{2z})$$

Através da simulação da expressão matricial acima podemos determinar por completo o comportamento dinâmico do objeto articulado ao longo do tempo. Para tanto, consideraremos conhecidas, num determinado instante de tempo, a posição (e orientação) e a velocidade (linear e angular) do objeto. Para determinarmos a posição e velocidade do objeto no "próximo instante de tempo" devemos então calcular a aceleração atual (linear e angular) e, através de integração numérica, determinar os próximos valores para as variáveis de posição, orientação e velocidade.

Logo, através do método de integração numérica de Euler, temos:

$$\omega_i^{(k+1)} = \omega_i^{(k)} + \dot{\omega}_i^{(k)} \cdot \Delta t \quad (6.6.a)$$

$$v_i^{(k+1)} = v_i^{(k)} + a_i^{(k)} \cdot \Delta t \quad (6.6.b)$$

$$S_i^{(k+1)} = S_i^{(k)} + v_i^{(k)} \cdot \Delta t + 0,5 \cdot a_i^{(k)} \cdot (\Delta t)^2 \quad (6.6.c)$$

sendo Δt = intervalo de amostragem.

Outro método de integração (mais eficiente) que pode ser utilizado é o método de Runge-Kutta [Boyce (1985)].

Dessa forma, a primeira matriz da expressão (6.5) representa as "incógnitas" (acelerações) do objeto a ser simulado. Já a segunda matriz, que é constante e pode ser previamente calculada, representa a configuração do objeto sendo comumente chamada de **matriz de massa**. Finalmente, a terceira matriz é composta pelas forças e torques que estão atuando no objeto.

O item 3 do apêndice desta dissertação descreve uma possível implementação para o modelo proposto na fig. 6.3 através das equações (6.3) a (6.6).

Como um novo exemplo, considere a fig. 6.4, que apresenta uma estrutura em árvore.

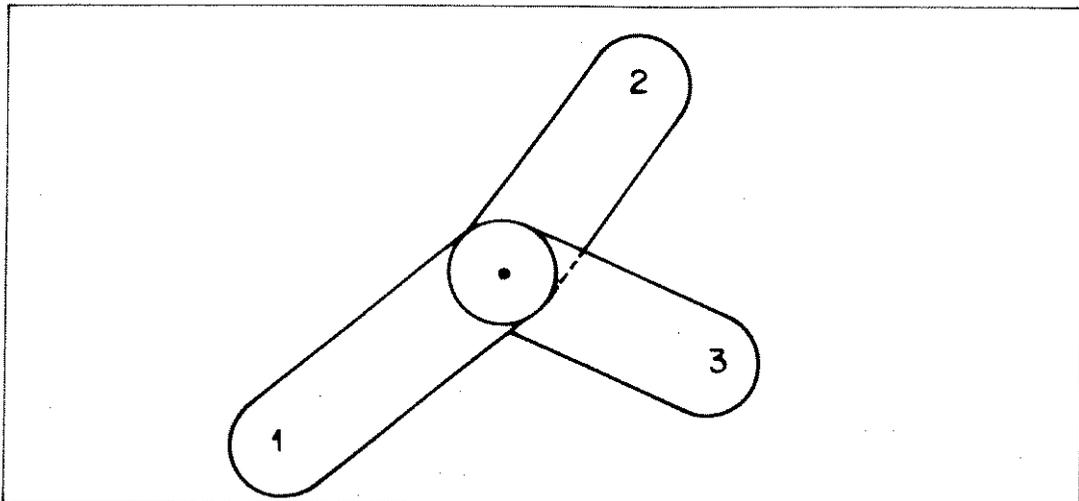


Fig. 6.4: Três segmentos unidos por intermédio de uma junta rotacional.

As equações dinâmicas que modelam este objeto são:

$$F_1 = m_1 \cdot [a_1 + \dot{\omega}_1 \times Pc_1 + \omega_1 \times (\omega_1 \times Pc_1)] \quad (6.7.a)$$

$$N_1 = I_1 \cdot \dot{\omega}_1 + m_1 \cdot Pc_1 \times a_1 + \omega_1 \times I_1 \cdot \omega_1 \quad (6.7.b)$$

$$F_2 = m_2 \cdot [a_2 + \dot{\omega}_2 \times Pc_2 + \omega_2 \times (\omega_2 \times Pc_2)] \quad (6.7.c)$$

$$N_2 = I_2 \cdot \dot{\omega}_2 + m_2 \cdot Pc_2 \times a_2 + \omega_2 \times I_2 \cdot \omega_2 \quad (6.7.d)$$

$$F_3 = m_3 \cdot [a_3 + \dot{\omega}_3 \times Pc_3 + \omega_3 \times (\omega_3 \times Pc_3)] \quad (6.7.e)$$

$$N_3 = I_3 \cdot \dot{\omega}_3 + m_3 \cdot Pc_3 \times a_3 + \omega_3 \times I_3 \cdot \omega_3 \quad (6.7.f)$$

$$f_1 = F_1 + {}^1_2R \cdot f_2 + {}^1_3R \cdot f_3 \quad (6.7.g)$$

$$\eta_1 = N_1 + {}^1_2R \cdot \eta_2 + {}^1_{P_2} \times {}^1_2R \cdot f_2 + {}^1_3R \cdot \eta_3 + {}^1_{P_3} \times {}^1_3R \cdot f_3 \quad (6.7.h)$$

As equações (6.1) a (6.7) definem a dinâmica de estruturas articuladas em árvore. Entretanto, o modelo definido por estas equações não leva em consideração características comuns presentes em objetos articulados como, por exemplo, o limite de junta, o atrito existente, a "tensão muscular" (efeito mola), etc. Dessa forma, na seção seguinte discutiremos como acrescentar estes efeitos aos objetos articulados.

6.2 - CARACTERÍSTICAS DOS OBJETOS ARTICULADOS

6.2.1 - LIMITE DE JUNTAS / TORQUE DE RESTAURAÇÃO

Na seção anterior abordamos um modelo onde as juntas poderiam se movimentar livremente. No entanto, em muitos casos não é permitido que as juntas desloquem-se livremente além de um certo limite. Consideremos, por exemplo, uma junta rotacional "i" entre os segmentos "i-1" e "i". Consideremos também uma restrição na forma: "o ângulo formado entre os segmentos "i-1" e "i" pode variar de $\theta_i^l < \theta_i < \theta_i^r$ ".

A melhor maneira de mantermos esta restrição para que o movimento se pareça o mais natural possível, segundo [Armstrong (1985)], consiste em se adicionar um torque de restauração à junta "i" conforme mostra a fig. 6.5:

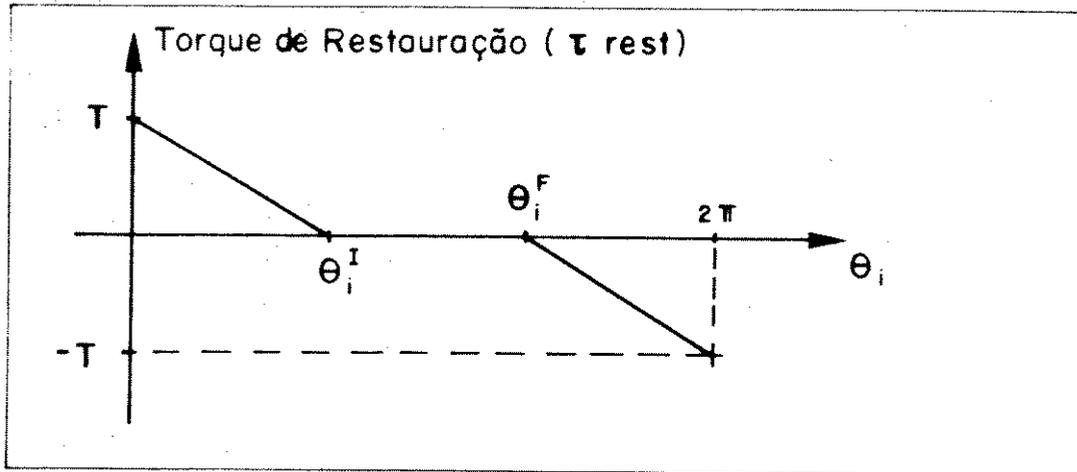


Fig. 6.5: Torque de restauração aplicado a uma junta rotacional.

6.2.2 - EFEITO "MOLA" / TORQUE DE REAÇÃO

Em alguns casos talvez desejássemos obter um efeito tipo "mola" na junta de algum segmento. Tal efeito é obtido, conforme [Isaacs (1987)], adicionando-se um torque à junta na forma:

$$\tau_{\text{mola}} = k_m \cdot (\text{deslocamento da posição de repouso da junta}) \quad (6.8)$$

sendo k_m a constante de "elasticidade" da junta.

6.2.3 - PERDAS NAS JUNTAS (AMORTECIMENTO VISCOZO) / TORQUE DE DISSIPACÃO

Certos sistemas podem apresentar perdas de energia nas juntas durante o movimento. Este efeito, segundo [Isaacs (1987)], pode ser representado por um torque adicionado à junta na forma:

$$\tau_{\text{amort.}} = k_a \cdot (\text{velocidade angular do segmento}) \quad (6.9)$$

sendo k_a a constante de "amortecimento" da junta.

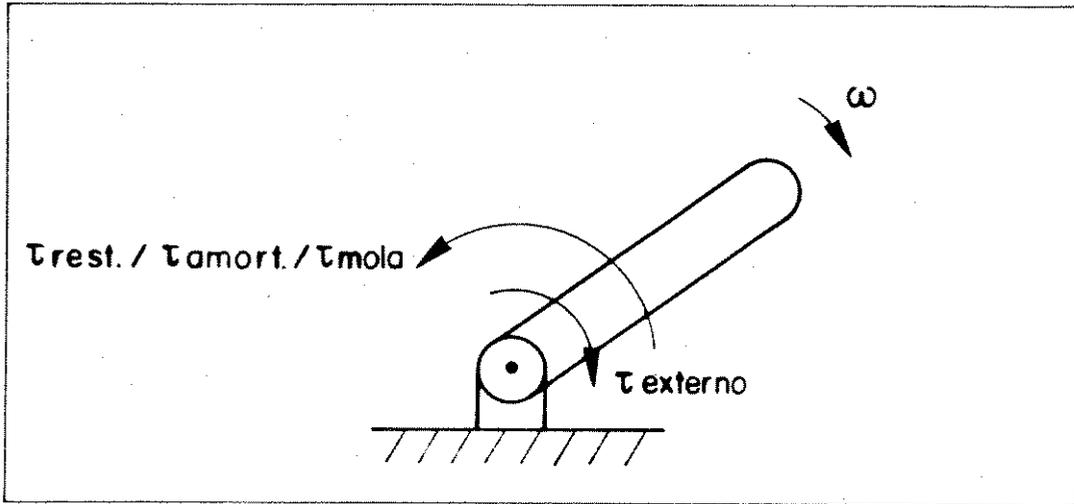


Fig. 6.6: Torques tipo "mola", tipo "amortecimento" e tipo "restauração" aplicados a uma junta.

Para a fig. 6.6, temos:

$$\tau = \tau_{externo} - \tau_{restauração} - \tau_{mola} - \tau_{amortecimento}$$

A seção 10.6 apresenta um modelo de simulação simplificado do movimento dos dedos de uma mão. O modelo desenvolvido integra as técnicas apresentadas nos capítulos 4, 5 e 6. O item 4 do apêndice desta dissertação descreve como utilizar a implementação presente no disquete anexo.

Até o presente momento estudamos métodos de controle de movimento sob dois pontos de vista:

- 1º) Dada uma trajetória definida completamente pelo "usuário" do sistema, como determinar o conjunto de variáveis de estado do objeto para o movimento desejado ?
- 2º) Atribuindo-se valores para as variáveis de estado do objeto, qual o movimento obtido ?

Note que, sob estes dois pontos de vista, não se encontra definido o conceito de otimização do movimento a ser realizado. Ou seja, não é possível definir perfeitamente qual o conjunto de variáveis de estado que otimizam o movimento a ser realizado. Portanto, discutiremos nos capítulos 7 e 8, ainda dentro do contexto de "Controle Local", alguns mecanismos destinados à otimização do movimento de objetos rígidos.

CAPÍTULO 7: CONTROLE ÓTIMO EM ANIMAÇÃO

De acordo com o modelo apresentado nos capítulos 4 e 5, podemos gerar trajetórias para uma estrutura articulada e então calcular os torques necessários nas juntas para que o movimento seja efetivamente realizado. Por sua vez, de acordo com o modelo apresentado no capítulo 6, podemos aplicar forças e torques a uma estrutura articulada e então observar a trajetória gerada.

Uma outra possível opção seria: "dada uma trajetória que vai do ponto "A" ao ponto "B" no espaço tridimensional, num certo intervalo de tempo, qual o melhor conjunto de forças e torques (segundo algum critério de otimização) que satisfaz o movimento desejado?". Neste caso estamos diante de um problema de otimização (controle ótimo) bastante comum em animação que será estudado neste capítulo.

Alguns trabalhos referentes a este tipo de problema tem sido publicados desde o final da década de 80. [Witkin (1988)] e [Ngo (1993)] abordam o tema sob o rótulo de "Restrições de Espaço-Tempo". [Barzel (1988)] apresenta uma visão mais abrangente do ponto de vista da dinâmica do problema. [Panne (1990)] inova este tipo de abordagem através do conceito de "Controlador de Espaço de Estados". Finalmente, em [Gill (1981)] e [Brotman (1988)] são apresentadas ferramentas matemáticas de grande importância para o desenvolvimento desta classe de problemas.

7.1 - FERRAMENTAS MATEMÁTICAS

Seja: $F(x) = F(x_1, x_2, \dots, x_n)$; $x \in \mathbb{R}^n$; $F(x) \in \mathbb{R}^n$

onde "x" é representado pelo vetor $x=[x_1, x_2, \dots, x_n]$.

Podemos então definir:

1-) Vetor Gradiente de $F(x)$:

$$\nabla F(x) = g(x) = \left[\frac{\partial F}{\partial x_1}, \frac{\partial F}{\partial x_2}, \dots, \frac{\partial F}{\partial x_n} \right]^T \quad (7.1)$$

2-) Matriz Hessiano de $F(x)$:

$$\nabla^2 F(x) = G(x) = \begin{bmatrix} \frac{\partial^2 F}{\partial x_1 \partial x_1} & \frac{\partial^2 F}{\partial x_1 \partial x_2} & \dots & \frac{\partial^2 F}{\partial x_1 \partial x_n} \\ \vdots & & & \\ \frac{\partial^2 F}{\partial x_n \partial x_1} & \frac{\partial^2 F}{\partial x_n \partial x_2} & \dots & \frac{\partial^2 F}{\partial x_n \partial x_n} \end{bmatrix} \quad (7.2)$$

Note que: $\frac{\partial^2 F}{\partial x_i \partial x_j} = \frac{\partial^2 F}{\partial x_j \partial x_i} \Rightarrow \nabla^2 F(x)$ é simétrica.

3-) Norma Euclidiana de $F(x)$:

$$\| F(x) \| = (x_1^2 + x_2^2 + \dots + x_n^2)^{1/2} \quad (7.3)$$

Para uma função vetorial do tipo:

$$C(x) = [c_1(x), c_2(x), \dots, c_t(x)]; \quad x \in \mathbb{R}^n; \quad C(x) \in \mathbb{R}^n$$

podemos definir:

4-) Matriz Jacobiano de $C(x)$:

$$A(x) = \begin{bmatrix} \frac{\partial c_1}{\partial x_1} & \frac{\partial c_1}{\partial x_2} & \dots & \frac{\partial c_1}{\partial x_n} \\ \frac{\partial c_2}{\partial x_1} & \frac{\partial c_2}{\partial x_2} & \dots & \frac{\partial c_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial c_t}{\partial x_1} & \frac{\partial c_t}{\partial x_2} & \dots & \frac{\partial c_t}{\partial x_n} \end{bmatrix}_{t \times n} \quad (7.4)$$

7.2 - MINIMIZAÇÃO DE UMA FUNÇÃO SUJEITA A RESTRIÇÕES

Considere, por exemplo, o objeto da fig. 7.1. Trata-se de um corpo rígido, de massa "m", que deve ser levado do ponto x_I ao ponto x_F em "T" unidades de tempo.

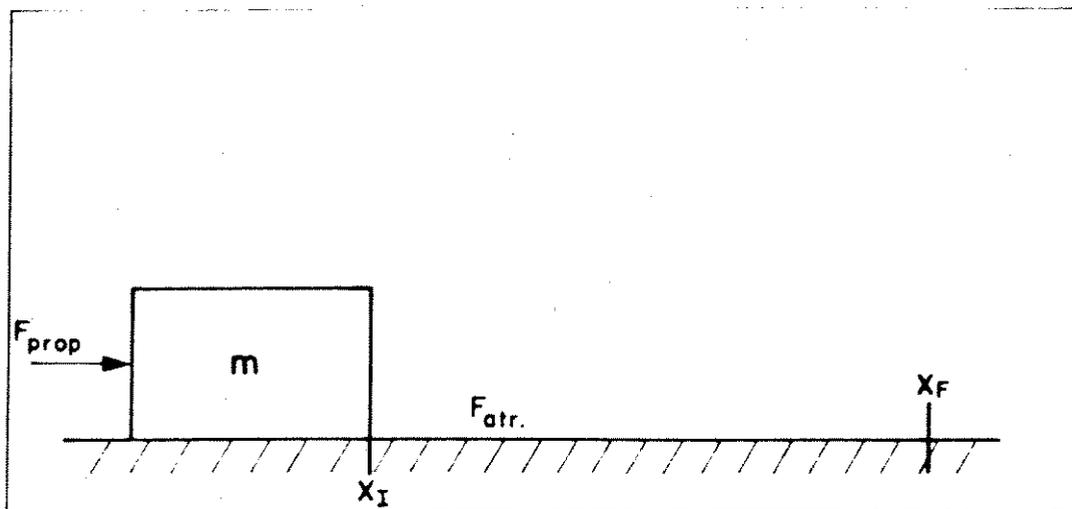


Fig. 7.1: Exemplo de um sistema sob restrições a ser otimizado.

Note que o objeto está sujeito a uma força propulsora $F_{\text{prop}} = F(t)$ e a uma força de atrito $F_{\text{atr}} = -b \cdot dx/dt$; sendo "b" o coef. de atrito.

De acordo com a 2ª Lei de Newton, esta "simulação" poderia ser modelada na forma:

$$F_{\text{prop}} + F_{\text{atr}} = m \cdot \vec{a}$$

ou seja: $m \cdot \frac{d^2x}{dt^2} + b \cdot \frac{dx}{dt} - F(t) = 0$

O problema anteriormente formulado poderia então ser enunciado na forma:

Como fazer com que o objeto vá de x_1 a x_F em T unidades de tempo, "gastando o mínimo" da força propulsora ?

Ou seja:

$$\begin{aligned} & \text{Minimize } \int_0^T [F(t)^2] dt , \\ & \text{sujeito a: } x(t=0) - x_1 = 0 \\ & \quad \quad \quad x(t=T) - x_F = 0 \\ & \quad \quad \quad m \cdot \frac{d^2x}{dt^2} + b \cdot \frac{dx}{dt} - F(t) = 0 \end{aligned}$$

Optamos por minimizar $\int_0^T [F(t)^2] dt$ por considerar uma boa aproximação para a "taxa de consumo" de F(t).

Dessa forma, um problema genérico de minimização pode ser re-escrito na forma:

$$\begin{aligned} & \text{Minimize } F(x), x \in \mathbb{R}^n , \\ & \text{estando } F(x) \text{ sujeita às restrições:} \\ & C(x) = [c_1(x), c_2(x), \dots, c_t(x)] \\ & \text{sendo: } c_i(x) = 0 ; \quad i = 1, 2, \dots, t. \end{aligned}$$

Para a solução deste tipo de problema serão propostos dois algoritmos baseados em técnicas de otimização apresentadas em [Gill (1981)]. O primeiro deles é adequado à solução de problemas relativamente simples. O segundo, por sua vez, é apropriado a sistemas de maior complexidade.

7.2.1 - 1º ALGORITMO

Seja x^* uma solução ótima que buscamos para o problema P1.

Faça:

1º Passo: Construa uma função do tipo:

$$B(x,r) = F(x) - r \cdot \sum_{i=1}^t \ln[c_i(x)] \quad (7.5)$$

onde "r" é uma variável auxiliar.

2º Passo: Obtenha uma expressão para $x(r)$ fazendo:

$$\frac{\partial B(x,r)}{\partial x} = 0 \quad (7.6)$$

3º Passo: Obtenha a solução x^* fazendo:

$$x^* = \lim_{r \rightarrow 0} x^*(r) \quad (7.7)$$

Segundo [Gill (1981)] a função $B(x,r)$ também pode ser expressa como sendo:

$$B(x,r) = F(x) + r \cdot \sum_{i=1}^t \left[\frac{1}{c_i(x)} \right] \quad (7.8)$$

Como pode ser observado, $B(x,r)$ representa uma combinação entre a função a ser minimizada e o conjunto de restrições a ser considerado no processo de minimização. Neste caso, os processos desenvolvidos nos passos 2º e 3º do algoritmo buscam a projeção do vetor oposto ao gradiente de $F(x)$ (portanto um vetor contrário ao crescimento de $F(x)$) no sub-espaço das restrições.

7.2.2 - 2º ALGORITMO: ALGORITMO DE PROJEÇÃO DO GRADIENTE

Para a solução do problema P1 vamos agora considerar o seguinte processo iterativo:

Seja:

$X^k = [x_1^k, x_2^k, \dots, x_n^k]$ o valor iterativo atual e

$X^{k+1} = [x_1^{k+1}, x_2^{k+1}, \dots, x_n^{k+1}]$ o próximo valor iterativo.

Para que o problema se mantenha consistente devemos ter:

$$C(X^k) = C(X^{k+1}) = 0 \quad (7.9.a)$$

$$F(X^{k+1}) < F(X^k) \quad (7.9.b)$$

Ou seja, $F(X^{k+1})$ deve sofrer um "decréscimo suficiente" de $F(X^k)$.

Para o vetor $C(X^{k+1})$ podemos fazer a seguinte aproximação, em série de Taylor, em torno de X^k :

$$C(X^{k+1}) \cong C(X^k) + A(X^k) \cdot s^k; \quad s^k \in \mathbb{R}^n \quad (7.10)$$

Dessa forma, o problema agora é encontrar o "vetor" s^k que satisfaça as condições impostas e então fazer:

$$X^{k+1} = X^k + s^k \quad (7.11)$$

Como primeira aproximação para X^k (ou seja, X^0) vamos procurar por algum vetor que satisfaça:

$$C(X^0) = 0 \quad (7.12)$$

Dispondo de uma solução inicial, proceder como o algoritmo que segue:

1º Passo: Escrever o Jacobiano das restrições na forma:

$$A^k = [V \ U] \quad (7.13)$$

sendo V uma matriz $t \times t$ que abrange as primeiras " t " colunas da matriz A^k ; U é a matriz que abrange as colunas restantes de A^k .

2º Passo: Fazer:

$$Z^k = \begin{bmatrix} -(V^{-1}) \cdot U \\ I \end{bmatrix} \quad (7.14)$$

sendo Z^k uma matriz cujas $(n-t)$ colunas formam uma base para o conjunto de vetores ortogonais às linhas de A^k e I é uma matriz identidade " $t \times t$ ".

3º Passo: Fazer:

$$p_z = -(Z^k)^T \cdot g^k(X^k) \quad (7.15)$$

sendo $g^k(X^k)$ o gradiente de $F(X^k)$.

Note que $-p_z$ é o gradiente de $F(X^k)$ projetado no sub-espço de vetores ortogonais ao gradiente das restrições.

4º Passo: Fazer:

$$p^k = Z^k \cdot p_z \quad (7.16)$$

p^k é uma possível solução para s^k , entretanto, para que $F(X^k)$ sofra um "decréscimo suficiente", devemos proceder ao quinto passo:

5º Passo: Fazer:

$$s^k = \alpha^k \cdot p^k \quad (7.17)$$

sendo α^k um escalar positivo, escolhido de modo a termos:

$$F(X^k + \alpha^k \cdot p^k) < F(X^k) \quad e \quad (7.18.a)$$

$$C(X^k + \alpha^k \cdot p^k) = 0 \quad (7.18.b)$$

Para que o decréscimo seja suficiente para haver convergência, α^k deve ser escolhido de modo a satisfazer:

$$0 < -\mu_1 \cdot \alpha^k \cdot (g^k)^T \cdot p^k \leq F(X^k) - F(X^{k-1}) \leq -\mu_2 \cdot \alpha^k \cdot (g^k)^T \cdot p^k \quad (7.19)$$

sendo μ_1 e μ_2 escalares arbitrários que satisfazem:

$$0 < \mu_1 \leq \mu_2 \leq 1 \quad (7.20)$$

6º Passo: Finalmente, fazer:

$$X^{k+1} = X^k + s^k \quad (7.21)$$

O algoritmo deve ser interrompido quando s^k for suficientemente pequeno, não provocando grande variação em $F(X)$. Neste ponto tem-se a solução ótima para o problema em X^k .

Retomando o exemplo proposto no início da seção 7.2, podemos discretizar $x(t)$ e $F(t)$, representando estas funções por uma sequência de amostras x_i e F_i , $0 \leq i \leq n$, tomadas a um intervalo de tempo constante Δt .

Assumindo que $F(t)$ é constante entre um instante de amostragem e outro "vizinho", a função a ser minimizada pode ser re-escrita como:

$$\int_0^T [F(t)^2] dt \cong \Delta t \cdot \sum_{i=0}^n F_i^2 = f(v)$$

Considerando também que:

$$\frac{dx_i}{dt} \cong \frac{x_i - x_{i-1}}{\Delta t} \quad \text{e} \quad \frac{d^2 x_i}{dt^2} \cong \frac{x_{i+1} - 2 \cdot x_i + x_{i-1}}{\Delta t^2}$$

então as restrições podem ser aproximadas por:

$$x_0 - x_I = 0$$

$$x_n - x_F = 0$$

$$\frac{m}{\Delta t^2} \cdot (x_{i+1} - 2 \cdot x_i + x_{i-1}) + \frac{b}{\Delta t} \cdot (x_i - x_{i-1}) - F_i = 0$$

Notemos que a terceira equação de restrição deve ser expandida em várias outras equações para todas as variáveis independentes do problema: $\{x_0, x_1, \dots, x_n, F_0, F_1, \dots, F_n\} = v$.

Veja que: $u = 2 \cdot n + 2 \Rightarrow v \in \mathbb{R}^u$

Em sua forma discretizada, o problema pode ser formulado como:

$$\begin{aligned} \text{Minimize: } f(v) &= \Delta t \cdot (F_0^2 + F_1^2 + \dots + F_n^2) \\ \text{sendo: } v &= \{x_0, x_1, \dots, x_n, F_0, F_1, \dots, F_n\} \\ \text{sujeita às restrições:} \\ x_0 - x_I &= 0 \\ x_n - x_F &= 0 \\ \frac{m}{\Delta t^2} \cdot (x_2 - 2 \cdot x_1 + x_0) + \frac{b}{\Delta t} \cdot (x_1 - x_0) - F_1 &= 0 \\ \frac{m}{\Delta t^2} \cdot (x_3 - 2 \cdot x_2 + x_1) + \frac{b}{\Delta t} \cdot (x_2 - x_1) - F_2 &= 0 \\ &\vdots \\ \frac{m}{\Delta t^2} \cdot (x_n - 2 \cdot x_{n-1} + x_{n-2}) + \frac{b}{\Delta t} \cdot (x_{n-1} - x_{n-2}) - F_{n-1} &= 0 \end{aligned}$$

A partir deste ponto podemos aplicar o algoritmo da seção 7.2.2 ao problema descrito em sua forma discretizada para obtermos a solução desejada.

Modelos mais complexos, como aqueles descritos através das equações de Euler apresentadas no cap. 6, também podem ser adaptados a esta metodologia. Os mesmos resultados podem ser obtidos através da generalização do modelo dinâmico para o objeto da fig. 7.2.

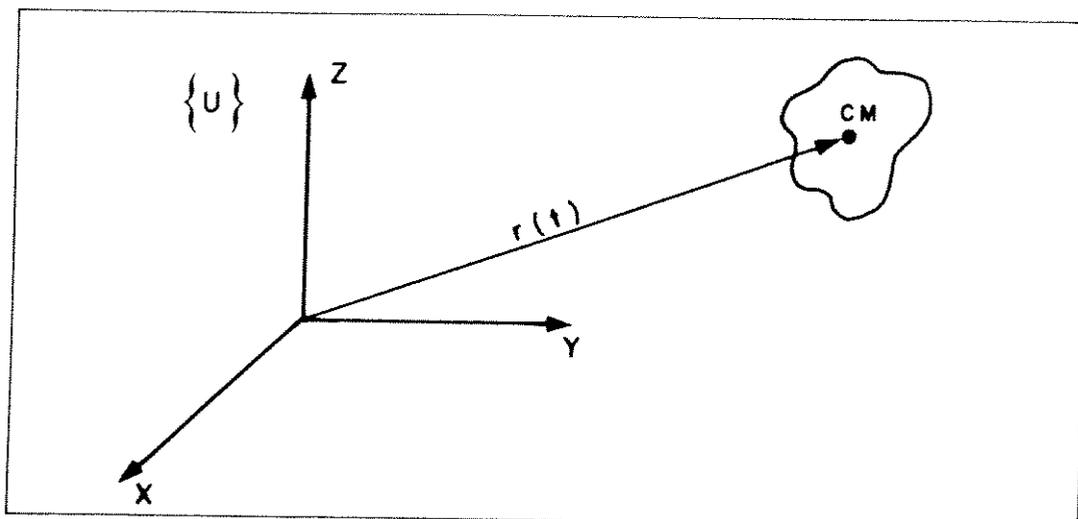


Fig. 7.2: Objeto genérico no espaço tridimensional.

Seja:

$r(t)^T = [x(t), y(t), z(t)]^T$ um vetor que determina a posição do centro de massa do objeto, no tempo, em relação ao S.C. inercial.

$\theta(t) = [\theta_x(t), \theta_y(t), \theta_z(t)]^T$ um vetor que determina os ângulos de rotação do objeto em relação aos eixos "x", "y" e "z" do S.C. inercial.

m - a massa do objeto.

I - o tensor de inércia do objeto em relação ao S.C. inercial.

$\sum F$ - representa o vetor soma das forças que atuam no objeto, relativas ao S.C. inercial.

$\sum M$ - representa a soma dos torques que atuam no objeto, relativas ao S.C. inercial.

Então:

$$\sum F = m \cdot \frac{d^2 r(t)}{dt^2} \quad (7.22.a)$$

$$\sum M = I \cdot \frac{d^2 \theta(t)}{dt^2} \quad (7.22.b)$$

A formulação apresentada neste capítulo gerou, em meados da década de 80, a filosofia de controle de animação denominada *Space-Time Constraints*, ou seja, uma filosofia de planejamento de animações orientada a restrições espaciais e temporais.

Entretanto, como pode ser notado através de um estudo mais cuidadoso, esta metodologia implica em uma automatização muito grande do processo de geração de uma animação. Por exemplo, a descrição da trajetória associada ao movimento é quase que completamente determinada pelo algoritmo de otimização. Dessa forma, para que o animador exerça influência sobre a trajetória a ser gerada, este deve determinar um conjunto relativamente grande de restrições.

Uma alternativa a este problema é descrita no capítulo 8, onde apresentamos um método para o planejamento de trilhas em animação por computador.

CAPÍTULO 8: PLANEJAMENTO DE TRILHAS / DESVIO DE OBSTÁCULOS

Modelos cinemáticos e dinâmicos são amplamente utilizados para o cálculo do movimento de qualquer objeto em animação por computador. Visto que buscamos uma "representação realística" para o movimento de algum objeto, temos nas leis físicas as melhores ferramentas para se modelar a realidade de nosso cotidiano. De fato, praticamente todos os paradigmas desenvolvidos para se modelar uma animação por computador são fundamentados na utilização de leis físicas, tais como as equações dinâmicas de Newton e Euler.

Dessa forma, de acordo com grande parte das técnicas de controle em animação, o animador deve especificar "o quê" um ator deve fazer, por exemplo "mova-se daqui para lá" (restrição espacial), e "como" a ação deve ser realizada, por exemplo "mova-se o mais rápido possível mas não desperdice energia" (restrição temporal-energética). Mais ainda, o animador deve especificar as outras restrições presentes no sistema a ser modelado, tais como as leis físicas que regem o movimento, etc. A seguir, de acordo com as restrições impostas, a trilha para o movimento é gerada por alguma técnica de otimização.

Isto significa que o animador tem consciência da tarefa atribuída ao ator, mas a trilha a ser propriamente seguida pelo mesmo pode ser uma incógnita. Analisando as restrições impostas ao movimento é possível extrair todas as informações sobre a trilha gerada, mas isto nem sempre é uma tarefa trivial. Mais ainda, a determinação de caminhos livres de colisões pode exigir um conjunto muito complexo de restrições em alguns casos. Portanto, o paradigma acima descrito refere-se muito mais a "como se especificar as ações desejadas" do que a "como se planejar o movimento".

Para se solucionar o problema do planejamento do movimento de um ator sugerimos, então, um mecanismo de planejamento prévio da trilha a ser seguida. Dessa forma, a avaliação da trilha a ser seguida pelo ator deve ser realizada antes da aplicação de algum modelo cinemático ou dinâmico ao movimento. Uma vez que uma trilha conveniente tenha sido calculada de acordo com algum conjunto de restrições, tais como "não se aproxime muito deste objeto" ou "encontre o caminho mínimo entre estes dois pontos", algum modelo cinemático ou dinâmico pode ser aplicado ao sistema para se gerar o movimento final do ator.

Esta estratégia fornece ao animador um maior controle da animação como um todo, visto que ele não apenas sabe o que deve ser feito, mas também é consciente de tudo aquilo que ocorre durante o movimento. Mais ainda, dividindo o modelo inicial em duas partes, ou seja, um planejamento prévio da trilha a ser seguida e posterior aplicação de um modelo fisicamente válido, tornamos o controle da animação como um todo mais fácil de ser manipulado.

Existem desenvolvidos, atualmente, diversos métodos para a determinação de trilhas (*path planning*) de objetos rígidos evitando-se colisões (*collision avoidance*). E. G. Gilbert apresentou um método em [Gilbert (1985)] que utiliza funções de distância para o cálculo de uma trilha ótima; a idéia principal consiste em expressar o impedimento de colisões com outros obstáculos em termos das distâncias entre partes que apresentam grande probabilidade de colisão. O. Takahashi, por sua vez, apresentou um método para se evitar colisões entre figuras planas (polígonos) no espaço 2-D utilizando Diagramas de Voronoi [Takahashi (1989)]. L. P. Gewali considerou o problema de planejamento de trilhas no espaço 3-D na presença de obstáculos verticais [Gewali (1990)].

Um método muito interessante para o problema do impedimento de colisões ou planejamento de trilhas é apresentado por Y. K. Hwang em [Hwang (1992)]. Neste método o planejamento de trilhas no espaço 3-D é obtido através da representação de obstáculos por intermédio de campos potenciais.

O algoritmo para planejamento de trilhas que apresentaremos a seguir é parcialmente baseado na estratégia de atribuição de "potenciais" aos obstáculos presentes numa região do espaço. Para se encontrar um conjunto de possíveis trilhas, de acordo com algum conjunto de restrições impostas ao ambiente onde se passa a animação, sugerimos a atribuição de "funções potenciais" aos obstáculos presentes no ambiente. Uma vez que o potencial para cada ponto tenha sido calculado, podemos obter um conjunto de trilhas possíveis para o movimento desejado. A seguir, dados os pontos inicial e final para o movimento, através de uma "função de otimização" podemos encontrar uma trilha ótima para o movimento.

8.1 - UM CASO ESTUDADO

Discutiremos nosso método através de um exemplo relativamente simples. Para o exemplo em questão nos limitaremos a aplicar o algoritmo de planejamento de trilhas, visto que já foram apresentadas nos capítulos anteriores diversas técnicas cinemáticas e dinâmicas que podem ser aplicadas posteriormente ao cálculo da trilha do movimento.

8.1.1 - DEFINIÇÃO DOS CAMPOS DE POTENCIAIS

O método a seguir pode ser aplicado para regiões no espaço 3-D ou 2-D. Considere, por exemplo, a região planar apresentada na fig. 8.1.

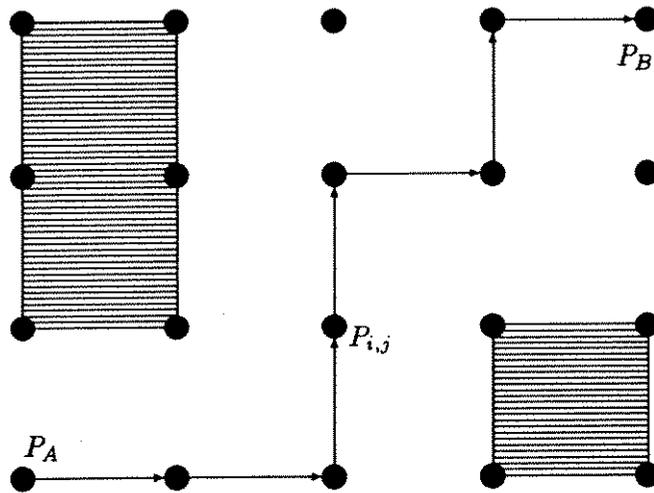


Fig. 8.1: Uma região planar a ser estudada.

Como primeiro passo, a região planar deve ser decomposta em uma rede de pontos. A rede resultante será, portanto, composta por pontos dentro de "obstáculos" ou no "espaço livre". A seguir, a cada um dos pontos da rede devemos atribuir um determinado potencial. Cabe ressaltar que o cálculo de potenciais que se fará para cada ponto tem um pressuposto particular, a analogia com potenciais de campos elétricos.

Iniciaremos este procedimento atribuindo um potencial P_i a cada ponto no interior ou na borda de um obstáculo "1". Notemos que, atribuindo potenciais diferentes a cada obstáculo podemos obter comportamentos diferentes para o movimento do ator, isto é, um potencial elevado fará com que o ator se distancie do obstáculo.

Podemos também considerar a fronteira que delimita a região planar como um obstáculo, atribuindo um potencial apropriado à mesma. Para os pontos no espaço livre inicialmente atribuiremos um potencial zero.

Visto que existem regiões com potenciais superiores em relação a outras, um campo de potenciais surgirá no espaço livre. Isto significa que devemos atualizar os valores inicialmente atribuídos aos pontos no espaço livre. Isto pode ser feito através de um processo iterativo que apresentaremos abaixo.

De acordo com a Equação de Laplace, em sua forma discretizada, apresentada em [Hayt (1981)], podemos avaliar o potencial de um ponto $P_{i,j}$ na região planar como sendo dado pela média dos potenciais de seus pontos vizinhos. Logo, para o exemplo apresentado na fig. 8.1, temos:

$$P_{i,j} = \frac{P_{i-1,j} + P_{i,j-1} + P_{i,j+1} + P_{i+1,j}}{4} \quad (8.1)$$

O algoritmo para a avaliação do campo de potencial de uma região é dado por:

1. Divida a região em uma rede de pontos.
2. Para cada ponto nesta região:
 - Se o ponto encontra-se no interior ou na borda de um obstáculo: atribua um valor para seu potencial.
 - Se o ponto encontra-se no espaço livre: atribua um potencial inicial igual a zero.
3. Repita, até que o potencial de cada ponto no espaço livre apresente convergência para algum valor:
 - Para cada ponto no espaço livre: atualize seu potencial de acordo com a eq. (8.1).

Até o momento, dispomos da topologia da região descrita através de um campo potencial. Isto significa que o problema inicial, isto é, "como evitar uma colisão", encontra-se representado pelo campo potencial presente no espaço livre. Portanto, sabemos que o ator estará se aproximando de um obstáculo quando o potencial dos pontos percorridos estiver aumentando. Ou seja, a probabilidade de colisão diminui à medida em que o potencial dos pontos percorridos também diminui.

Dessa forma, a trilha composta pelos pontos de menores potenciais certamente constitui a trilha com menor probabilidade de colisão. Entretanto, isto não implica que esta trilha é a mais curta entre dois pontos. Isto significa que necessitamos de algum critério/algoritmo de otimização de forma a descobirmos a "melhor trilha" entre dois pontos de uma região.

A seção seguinte discute esta etapa do problema, ou seja, "como encontrar a melhor trilha entre dois pontos mediante algum critério de otimização".

8.1.2 - DEFINIÇÃO DE UMA TRILHA OTIMIZADA

Para se avaliar uma trilha "ótima" entre dois pontos, vamos considerar uma regra básica em programação linear. De acordo com a fig. 8.2, se a trilha desenhada entre os pontos A e B é ótima, então a "sub-trilha" entre os pontos $P_{i,j}$ e B também é.

Isto significa que, de acordo com o exemplo da fig. 8.1, se $P_{i,j}$ se encontra na trilha ótima, e é necessário passar por algum vizinho de $P_{i,j}$ para se chegar a este ponto, então este ponto vizinho também se encontra na trilha ótima.

Existem diversos algoritmos para se calcular uma trilha ótima entre dois pontos. Um algoritmo de programação linear, como aqueles apresentados em [Gill (1981)] e [Gondran (1986)], são adequados ao método aqui desenvolvido porque já dispomos da região de nosso interesse representada através de uma rede. Podemos considerar esta região como um grafo onde os "nós" do grafo são representados pelos pontos discretizados da região. Os "arcos" do grafo são representados pela conectividade da região e os "pesos de transição" entre "nós" são dados por algum critério de otimização.

O critério de otimização que sugerimos é a soma ponderada do potencial do ponto com sua distância euclidiana até o ponto inicial da trilha. A função que representa o "custo" do ponto (i,j) , $C_{i,j}$, pode ser expressa por:

$$C_{i,j} = \alpha \cdot P_{i,j} + \beta \cdot D_{i,j} \quad (8.2)$$

onde:

α é o "peso" atribuído ao potencial do ponto (i,j) ;

$P_{i,j}$ é o potencial do ponto (i,j) ;

β é o "peso" atribuído à distância;

$D_{i,j}$ é a distância euclidiana entre os pontos (i,j) e o ponto inicial da trilha.

Para aplicarmos o algoritmo de otimização a ser apresentado a seguir, associamos, a cada ponto (i,j) no espaço livre, os seguintes atributos:

Point_Number:	Número associado ao ponto.
Father_Number:	Número do "pai" deste ponto.
X:	Coordenada "x" do ponto.
Y:	Coordenada "y" do ponto.
Z:	Coordenada "z" do ponto.
C:	"Custo" deste ponto.
Mark:	Indica se este ponto foi marcado.

O algoritmo que desenvolvemos para a otimização de uma trilha é representado por:

1. Para cada ponto no espaço livre:
defina Point_Number (≥ 1).
2. Defina os pontos inicial e final da trilha.
3. Selecione o ponto final, marque-o e faça-o ser o ponto atual.
Faça: Father_Number = 0 para o ponto final.
4. Para cada ponto (i, j), não marcado, no espaço livre vizinho ao ponto atual:
calcule o custo $C_{i,j}$.
5. Selecione o vizinho do ponto atual, que não esteja marcado, com menor custo. Faça seu Father_Number = Point_Number do ponto atual. Agora, marque o ponto selecionado e faça-o ser o ponto atual.
6. O ponto atual é o ponto inicial da trilha desejada ?
Sim: Fim do algoritmo.
Não: Volte ao passo 4.

Visto que estamos interessados na representação realística do movimento, uma vez que a trilha ótima tenha sido calculada, o próximo estágio é a aplicação de algum modelo físico ao movimento que deve ser executado de acordo com a trajetória gerada. Considerando-se que podemos estar modelando o movimento de objetos rígidos, aos invés de apenas entidades "puntiformes", sugerimos manter o maior eixo do objeto na mesma direção da trilha gerada.

Um exemplo para uma trajetória gerada é apresentado na fig. 8.2.

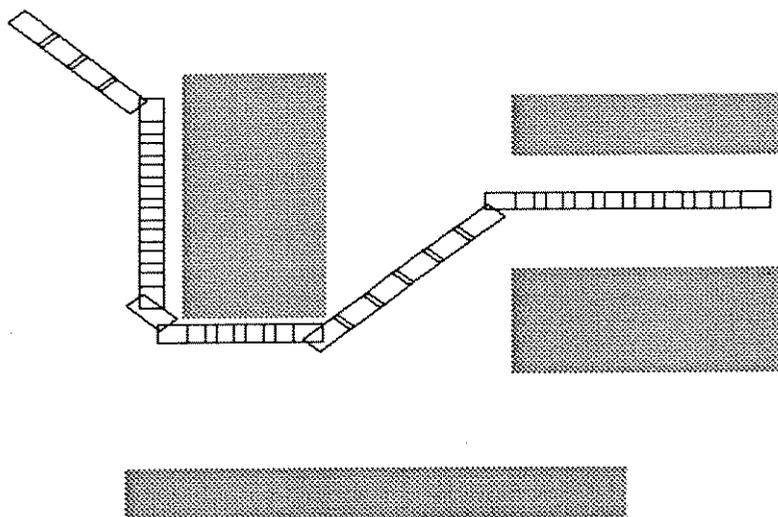


Fig. 8.2: Exemplo de trilha gerada através do algoritmo de planejamento de trilhas.

A seção 10.2 desta dissertação apresenta a implementação de um pequeno sistema para planejamento de trilhas / desvio de obstáculos. O item 5 do apêndice descreve como utilizar a implementação contida no disquete anexo.

**PARTE III:
CONTROLE GLOBAL DO MOVIMENTO**

Na parte III é introduzido o conceito de controle global em uma animação através do modelo de DEFS (Máquina de Estados Estendida).

CAPÍTULO 9: CONTROLE GLOBAL DE UMA ANIMAÇÃO ATRAVÉS DO MODELO DE DEDS

Conforme o capítulo 1, para que uma animação seja realizada estão envolvidos dois níveis de controle: local e global. Através do controle local realizamos o deslocamento efetivo de um determinado objeto, por outro lado, através do controle global controlamos a interação entre atores e entre um ator e o animador.

Como ilustração da interação entre os métodos de controle local e global discutiremos, neste capítulo, o controle de uma estrutura bípede. Podemos modelar um bípede através de uma estrutura articulada. De fato, são necessárias duas estruturas articuladas, idênticas no caso, para compormos o bípede. Analogamente, para um modelo que se assemelha a um ser humano devemos considerar mais duas estruturas articuladas para os seus braços.

Quando consideramos cada uma das estruturas articuladas individualmente, estaremos diante do controle local do ator. Ou seja, devemos solucionar o problema do posicionamento de cada uma das "pernas" ao longo do tempo isoladamente. Podemos encarar o problema do ponto de vista da cinemática inversa, ou seja, dadas as posições inicial e final de um objeto, devem ser calculadas as posições intermediárias para o objeto, de modo que o movimento pareça o menos robótico possível. Observe a fig. 9.1.

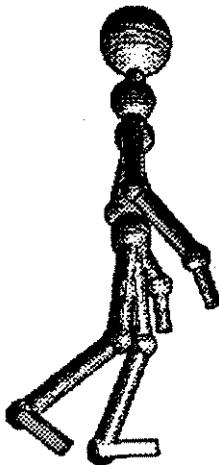


Fig. 9.1: Um bípede composto por dois pares de estruturas articuladas.

Por sua vez, não basta controlarmos o movimento de cada perna isoladamente, visto que, em um bípede, deve haver coordenação entre ambas as pernas para que a translação do ator seja efetivamente realizada. Por exemplo, enquanto caminha, não é permitido que ambas as pernas encontrem-se no ar simultaneamente. Dessa forma, quando estamos lidando com a interação que existe entre as pernas, estamos diante do problema de controle global do movimento.

Por simplicidade, ao modelarmos o movimento da estrutura articulada bípede, consideraremos que o mesmo pode apenas deslocar-se em linha reta. O controle local do movimento pode ser modelado com o auxílio de técnicas de cinemática inversa, conforme estudamos no cap. 4. Quanto ao controle global, este será modelado como uma Máquina de Estados Estendida (ESM), pois este tipo de sistema pode ser encarado como um Sistema Dinâmico a Eventos Discretos (DEDS). Neste trabalho abordaremos alguns conceitos relevantes sobre DEDS, entretanto, maiores detalhes podem ser encontrados em [Ho (1989)], [Cohen (1989)] e [Ostroff (1989)].

9.1 - PROJETO DO MECANISMO DE CONTROLE GLOBAL

A tarefa do mecanismo de controle global é coordenar o movimento das "pernas" em termos de suas respectivas posições e velocidades. O modelo de locomoção proposto em [Girard (1985)] não se baseia em DEDS, porém são descritos alguns parâmetros importantes tanto para o modelo cinemático quanto para o modelo de sistema a eventos discretos. Nas próximas seções discutiremos em detalhes estes modelos.

9.1.1 - MODELO DE LOCOMOÇÃO CINEMÁTICO

O modelo de locomoção discutido em [Girard (1985)] utiliza alguns parâmetros para descrever o deslocamento de um personagem dotado "de pernas".

- * Um padrão de locomoção descreve uma sequência de suspensões e descidas do pé. O padrão se repete à medida em que o personagem se move. Cada repetição de uma sequência é chamada de ciclo de locomoção.
- * O tempo (ou número de quadros) necessário para se completar um único ciclo equivale ao período "P" do ciclo.
- * A fase relativa da perna i, R_i , descreve a fração do período do ciclo de locomoção gasto antes que a perna "i" seja suspensa.
- * Durante cada período do ciclo de locomoção qualquer uma das pernas passará uma parte deste no solo. Tal fração é chamada de fator de suporte da perna i. Este fator pode ser utilizado para se distinguir entre o caminhar e o correr de um bípede.
- * O tempo em que um pé permanece no solo é chamado de tempo de suporte.
- * O tempo que uma perna permanece no ar é chamado de tempo de transferência.
- * O deslocamento é definido como a distância percorrida pelo personagem durante a fase de suporte do pé.

A fig. 9.2 ilustra o ciclo de locomoção de uma perna de um bípede.

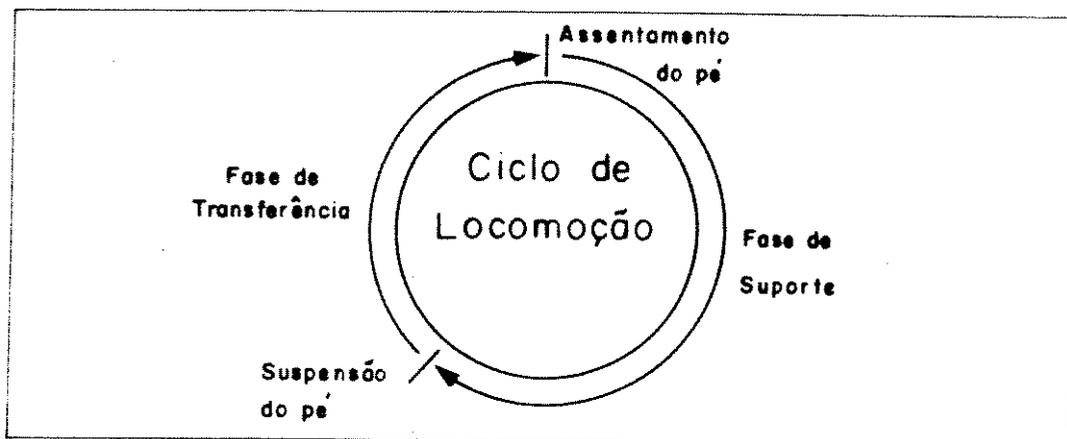


Fig. 9.2: Ciclo de locomoção de uma perna [Girard (1985)]

As seguintes equações descrevem o modelo cinemático de locomoção:

$$\text{Tempo de Suporte} = \frac{\text{deslocamento}}{\text{velocidade do personagem}} \quad (9.1)$$

$$\text{Fase de Suporte} = \frac{\text{Tempo de suporte}}{P} \quad (9.2)$$

$$\begin{aligned} \text{Tempo de Transferência} &= \frac{\text{comprimento do arco descrito} \\ &\quad \text{pelo deslocamento da perna}}{\text{velocidade da perna}} \quad (9.3) \\ &= P - \text{tempo de suporte} \end{aligned}$$

Note que, de fato, não existe nenhum controle interagindo com as duas pernas simultaneamente. Dessa forma, existem apenas regras cinemáticas que não garantem a coordenação das duas pernas. Na seção seguinte estruturaremos o controle do sistema através de um modelo a eventos discretos.

9.2 - MODELO DE LOCOMOÇÃO A EVENTOS DISCRETOS

Quando recorremos a um modelo de sistema dinâmico a eventos discretos para o exemplo em questão, desejamos que, efetivamente, exista um mecanismo de controle interagindo entre as pernas do bípede. As relações desenvolvidas e os parâmetros especificados na seção 9.1.1 são úteis ao modelo, em especial, devemos reconsiderar a fig. 9.2.

Visto que modelaremos o sistema como um conjunto de ESMs, é interessante apresentar, inicialmente, o método para se modelar sistemas com o auxílio de ESMs.

9.2.1 - DEFINIÇÃO DA SÍNTEXE DE ESMs

Definição: Uma ESM básica é uma quintupla (X, Y, C, L, A) onde, segundo [Ostroff (1989)]:

→ "X" é um conjunto singular $[x]$ onde "x" é uma variável de atividade com uma classe associada $\text{tipo}(x)$. Os elementos de $\text{tipo}(x)$ são chamados atividades ou estados.

→ "Y" é um conjunto de variáveis de dados onde cada variável de dado $y \in Y$ tem associada uma classe $\text{tipo}(y)$.

→ "C" é um conjunto de canais de comunicação. Um canal de comunicação pode ser considerado como sendo uma linha de comunicação uni-direcional conectando dois processos, através do qual um sinal ou dado pode ser transmitido.

→ "L" é um conjunto de rótulos de eventos.

→ "A" é um conjunto de ações básicas.

ACÕES BÁSICAS: Cada ação básica em A é dada por $[(x, E)]$, onde "E" é um evento dado pela quádrupla $(a_s, \text{guarda}, \text{operação}, a_d)$, onde:

- a_s é uma atividade (ou estado) de origem no tipo(x),
- a_d é uma atividade (ou estado) de destino no tipo(x),
- guarda é uma expressão booleana com as variáveis de dados.
- operação é descrita em detalhes abaixo:

OPERAÇÕES: Existem três tipos de operação:

1-) Uma operação de atribuição é dada por:

$\alpha[y:a]$ (Iê-se "a" é atribuído a "y") onde " α " é o rótulo de um evento, "y" é uma variável de dados e "a" é uma expressão. A expressão a deve ser do mesmo tipo da variável y. Atribuições simultâneas devem ser representadas na forma $\alpha[y_1: a_1, y_2: a_2]$.

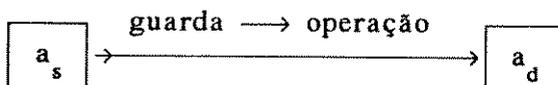
2-) Uma operação de envio é dada por:

$c!m$, onde "c" é um canal de comunicação em "C" e "m" é uma mensagem. Uma mensagem pode ser tanto um termo (isto é, uma expressão) com variáveis de dados ou de atividade, como um rótulo de um evento.

3-) Uma operação de recebimento é dada por:

$c?r$, onde "c" é um canal de comunicação e "r" pode ser tanto uma variável de dados como um rótulo de um evento.

O gráfico:



é uma representação de um evento. Se o guarda é omitido, então este é assumido como VERDADEIRO. Uma interpretação informal do evento acima é: "Se a ESM encontra-se atualmente na atividade a_s e, se o guarda é avaliado como sendo VERDADEIRO, então saltamos para o estado a_d enquanto executa-se a operação".

Operações de comunicação podem ser utilizadas tanto para comandar a execução de um evento específico como para se comunicar uma mensagem. Uma descrição informal desses tipos de comunicação é apresentada a seguir:

COMANDO PARA EXECUTAR UM EVENTO:

A operação de envio $c!\alpha$ em algum processo M_1 da ESM é lida como: "envie o comando "faça α " pelo canal "c" para algum outro processo (M_2 , por exemplo)". A operação de recebimento $c?\alpha$ no processo M_2 é lida como: "receba o comando "faça α " (de M_1) no canal "c" e execute o comando " α ". As operações de envio e recebimento, quando sincronizadas, resultam neste "comando e execução do evento α ".

COMUNICANDO UMA MENSAGEM:

Seja "m" um termo (o valor de "m" é uma mensagem) e seja "r" uma variável de dado. Então, a operação $c_1 m$ em algum processo M_1 é lida como: "envie a mensagem "m" pelo canal "c" (para algum processo M_2)". A operação correspondente $c_2 r$ em M_2 é lida como: "receba uma mensagem (de M_1) pelo canal "c" e armazene a mensagem em "r". As operações de envio e recebimento em conjunto resultam em uma atribuição distribuída de "m" para "r".

Uma operação de envio $c_1 m$ "casa-se" a uma operação de recebimento $c_2 r$ se $c_1 = c_2$ e, tanto "m" como "r" são um único rótulo de evento, ou "r" é uma variável de dados e "m" é um termo do mesmo tipo de "r" (isto é, $\text{tipo}(m) = \text{tipo}(r)$). Uma ação de envio (isto é, uma ação básica contendo uma operação de envio) em um ESM básico M não pode ter uma ação de recebimento correspondente em M pois não existe comunicação interna dentro do próprio ESM (um canal representa uma conexão de um-para-um de M para algum outro ESM).

Dois ESMs básicos devem sincronizar seus eventos quando os mesmos precisam interagir. A ação sincronizada, consistindo da participação simultânea de eventos componentes é chamada de interação. Segue a definição formal:

Definição:

ACÇÕES: - Uma ação pode ser tanto uma ação básica como uma interação. Uma interação pode ser tanto uma ação compartilhada como uma ação de comunicação, definidas como segue:

1-) Ações compartilhadas são definidas basicamente como segue.

Caso Base: Sejam A_i e A_j ações de atribuição (isto é, ações básicas com uma operação de atribuição) de diferentes ESMs possuindo o mesmo rótulo de evento " α ". Então, $A_i \cup A_j$ é uma ação compartilhada possuindo o mesmo rótulo de evento " α ".

Caso Induzido: Sejam A_i e A_j ações de atribuição ou ações já compartilhadas associadas a um mesmo evento " α ". Então, $A_i \cup A_j$ é também uma ação compartilhada associada ao mesmo rótulo de evento " α ".

2-) Sejam A_s e A_r ações de envio e recebimento correspondentes, respectivamente. Então, $A_s \cup A_r$ é uma ação de comunicação.

Se $A_i \cup A_j$ é uma interação, então A_i e A_j são chamadas de ações correspondentes.

Quando dois ESMs (M_1 e M_2 , por exemplo) operam em conjunto, então um novo ESM é formado onde:

$$M = M_1 \parallel M_2.$$

Caso não existam interações mútuas entre as ESMs, então as ações de M serão compostas pela união das ações de M_1 e M_2 . Por sua vez, se existem interações entre as máquinas que compõe M, então as ações de M devem refletir a sincronização de ações casadas entre as componentes. Para tanto, considere as definições seguintes.

Definição: Composição paralela de conjuntos de ações.

Se A_1 e A_2 são conjuntos de ações, então a composição paralela $A_1 || A_2$ é também um conjunto de ações definido como:

$$A_1 || A_2 \stackrel{\text{def}}{=} (A_1 \cup A_2 \cup \text{Interações}(A_1, A_2)) - \text{AçõesCasadas}(A_1, A_2)$$

onde:

$$\text{Interações}(A_1, A_2) \stackrel{\text{def}}{=} \{ A = A_1 \cup A_2 \mid A_1 \in A_1 \text{ e } A_2 \in A_2 \text{ e "A" é uma interação.} \}$$

$$\text{AçõesCasadas}(A_1, A_2) \stackrel{\text{def}}{=} \{ A \mid (\exists A') (A \cup A' \in \text{Interações}(A_1, A_2) \text{ e } A, A' \in A_1 \cup A_2) \}$$

O conjunto $\text{AçõesCasadas}(A_1, A_2)$ contém todas as ações componentes casadas utilizadas para construir interações sincronizadas em $\text{Interações}(A_1, A_2)$, as quais devem ser removidas da composição paralela de ESMs.

Definição: ESMs por composição paralela.

ESMs são geradas (a partir de ESMs básicas) pela aplicação das seguintes regras:

1. Qualquer ESM básica é uma ESM.

2. Seja $M_1 = (X_1, Y_1, C_1, L_1, A_1)$ e $M_2 = (X_2, Y_2, C_2, L_2, A_2)$ duas ESMs distintas, ou seja $X_1 \cap X_2 = \emptyset$ e $Y_1 \cap Y_2 = \emptyset$. Então:

$$M_1 || M_2 \stackrel{\text{def}}{=} (X_1 \cup X_2, Y_1 \cup Y_2, (C_1 \cup C_2) - (C_1 \cap C_2), L_1 \cup L_2, A_1 || A_2)$$

Portanto, os canais de $M = M_1 || M_2$ constituem a interface de comunicação de M com o ambiente externo. Os canais que conectavam as componentes de M foram eliminados pois constituíam apenas a comunicação interna na máquina.

9.2.2 - PLANEJAMENTO DAS ESMs PARA O MODELO DE LOCOMOÇÃO

Podemos considerar cada uma das pernas do bípede como sendo um ESM básico, sendo que, associados a uma perna "i", temos os seguintes estados:

- * S_i - Fase (estado) de suporte
- * T_i - Fase (estado) de transferência.

Também temos associados a cada perna "i" os seguintes eventos:

- * α_i - início da suspensão do pé "i",
- * β_i - fim da descida do pé "i".

Como especificação básica de controle temos: "Não é permitido que ambas as pernas encontrem-se na fase de transferência simultaneamente".

Para as pernas direita, P_D , e esquerda, P_E , podemos formalizar os seguintes ESMs básicos:

→ ESM P_E :

$$P_E = ([x_E], \emptyset, [m_E, c_E], [\alpha_E, \beta_E], A_E)$$

$$\text{tipo}(x_E) = [S_E, T_E]$$

$$A_E = [[(x_E, (T_E, \beta_E, [], S_E))] ; \\ [(x_E, (S_E, c_E ? \alpha_E, [], T_E))] ; \\ [(x_E, (S_E, \text{VERDADE}, m_E | x_E, S_E))] ; \\ [(x_E, (T_E, \text{VERDADE}, m_E | x_E, T_E))]] .$$

→ ESM P_D :

$$P_D = ([x_D], \emptyset, [m_D, c_D], [\alpha_D, \beta_D], A_D)$$

$$\text{tipo}(x_D) = [S_D, T_D]$$

$$A_D = [[(x_D, (T_D, \beta_D, [], S_D))] ; \\ [(x_D, (S_D, c_D ? \alpha_D, [], T_D))] ; \\ [(x_D, (S_D, \text{VERDADE}, m_D | x_D, S_D))] ; \\ [(x_D, (T_D, \text{VERDADE}, m_D | x_D, T_D))]] .$$

A partir de agora trataremos o problema de controle desta estrutura sob dois pontos de vista. Inicialmente será permitido ao ator mover-se em linha reta indefinidamente. Num segundo caso, permitiremos que o ator pare em um dado instante e possa, em seguida, retomar seu movimento retilíneo.

9.2.3 - 1º CASO: MOVIMENTO RETILÍNEO ININTERRUPTO

Neste caso, o controlador pode ser representado da seguinte forma¹:

→ ESM CONTROLADOR :

$$\text{CONTROLADOR} = ([x_c], [y_E, y_D], [c_E, c_D, m_E, m_D], [\alpha_E, \alpha_D], A_c)$$

$$\text{tipo}(x_c) = [L_1, L_2, L_3, L_4]$$

$$\text{tipo}(y_E) = [S_E, T_E]$$

$$\text{tipo}(y_D) = [S_D, T_D]$$

$$A_c = [[(x_c, (L_1, m_E ? y_E, [], L_2))] ; \\ [(x_c, (L_2, y_E = T_E, [], L_1))] ; \\ [(x_c, (L_2, y_E = S_E, c_D ! \alpha_D, L_3))] ; \\ [(x_c, (L_3, m_D ? y_D, [], L_4))] ; \\ [(x_c, (L_4, y_D = T_D, [], L_3))] ; \\ [(x_c, (L_4, y_D = S_D, c_E ! \alpha_E, L_1))]] .$$

Graficamente, podemos representar o conjunto planta + controlador conforme indica a fig. 9.3.

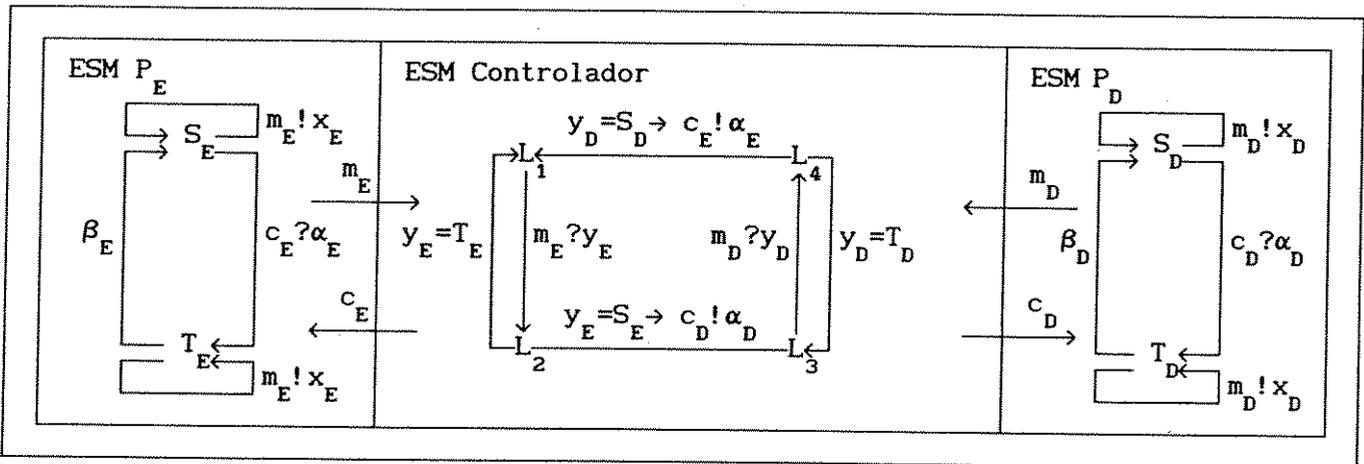


Fig. 9.3: Representação gráfica para o primeiro caso.

9.2.4 - 2º CASO: MOVIMENTO RETILÍNEO COM POSSIBILIDADE DE INTERRUPTÃO

Neste caso, antes de projetarmos o controlador devemos considerar a influência do "tronco" do ator sob a forma de um ESM.

Ao ESM tronco temos associados, por sua vez, dois eventos e dois estados, além de um canal de comunicação com o controlador.

¹ Cabe ressaltar o caráter heurístico desta proposta de controlador, possível devido à relativa simplicidade do problema.

Eventos:

- * γ - o tronco apresenta tendência a movimentar-se,
- * δ - o tronco apresenta tendência a parar.

Estados:

- * P - o tronco está parado,
- * M - o tronco está em movimento.

Dessa forma, para o tronco temos o seguinte ESM:

→ ESM TRONCO :

$$T = ([x_T], \emptyset, [m_T], [\gamma, \delta], A_T)$$

$$\text{tipo}(x_T) = [M, T]$$

$$A_D = [[(x_T, (M, \delta, [], P))] ; \\ [(x_T, (P, \gamma, [], M))] ; \\ [(x_T, (P, \text{VERDADE}, m_T | x_T, P))] ; \\ [(x_T, (M, \text{VERDADE}, m_T | x_T, M))]] .$$

Quanto ao controlador, este pode ser representado por:

→ ESM CONTROLADOR :

$$\text{CONTROLADOR} = ([x_c], [y_T, y_E, y_D], [c_E, c_D, m_T, m_E, m_D], [\alpha_E, \alpha_D], A_c)$$

$$\text{tipo}(x_c) = [L_1, L_2, L_3, L_4, L_5, L_6, L_7, L_8]$$

$$\text{tipo}(y_T) = [M, P]$$

$$\text{tipo}(y_E) = [S_E, T_E]$$

$$\text{tipo}(y_D) = [S_D, T_D]$$

$$A_c = [[(x_c, (L_1, m_T ? y_T, [], L_2))] ; \\ [(x_c, (L_2, y_T = P, [], L_1))] ; \\ [(x_c, (L_2, y_T = M, c_D ! \alpha_D, L_3))] ; \\ [(x_c, (L_3, m_T ? y_T, [], L_4))] ; \\ [(x_c, (L_4, y_T = P, [], L_1))] ; \\ [(x_c, (L_4, m_D ? y_D, [], L_5))] ; \\ [(x_c, (L_5, y_D = T_D, [], L_3))] ; \\ [(x_c, (L_5, y_D = S_D, c_E ! \alpha_E, L_6))] ; \\ [(x_c, (L_6, m_T ? y_T, [], L_7))] ; \\ [(x_c, (L_7, y_T = P, [], L_1))] ; \\ [(x_c, (L_7, m_E ? y_E, [], L_8))] ;$$

$$[(x_c, (L_8, y_E = T_E, [], L_6))] ;$$

$$[(x_c, (L_8, y_E = S_E, c_D ! \alpha_D, L_3))]] .$$

A representação gráfica para este conjunto planta + controlador é agora representada pela fig. 9.4 .

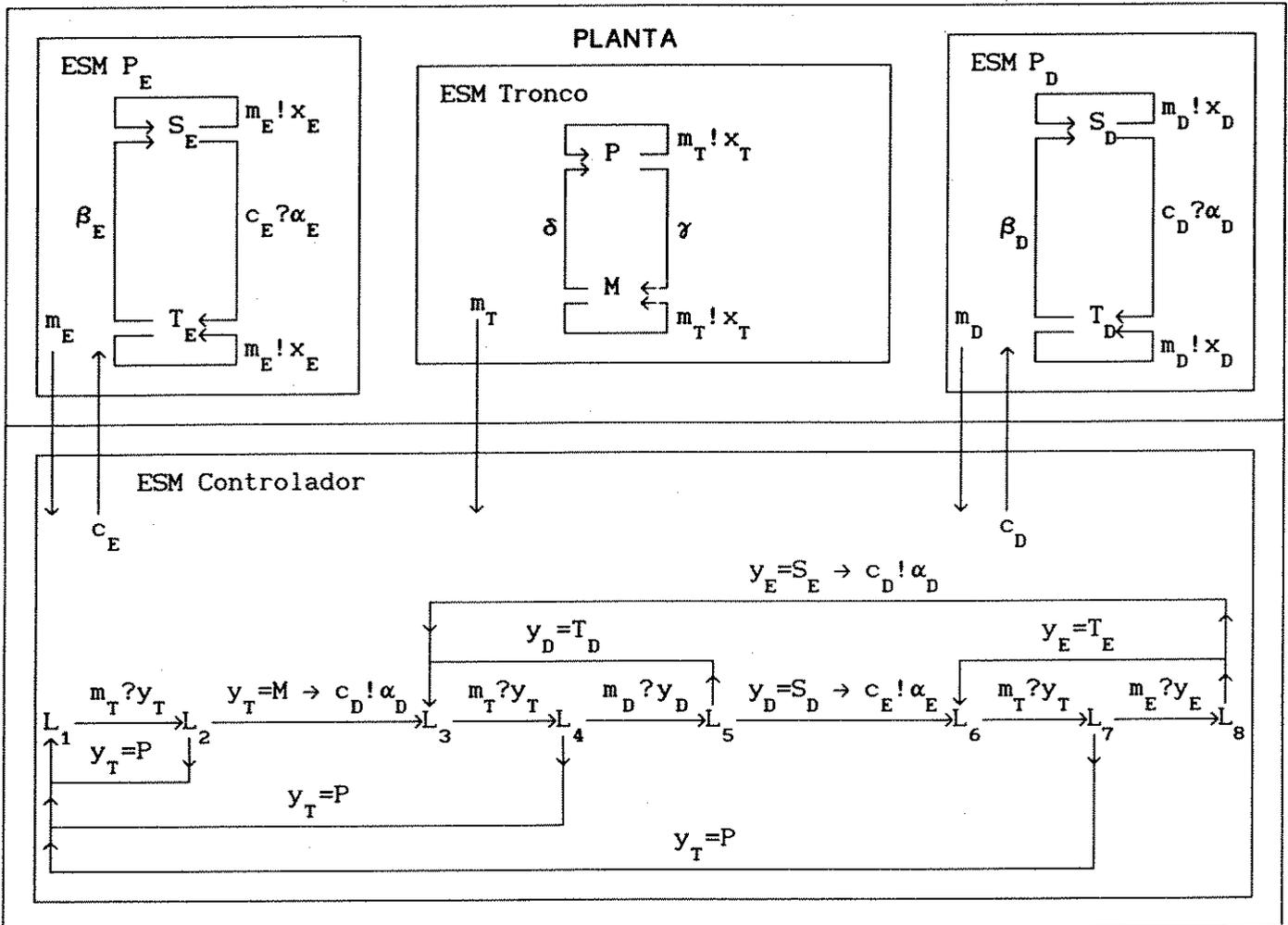


Fig. 9.4: Representação gráfica para a nova versão do sistema de controle global do caminhar de um bípede.

O modelo de controle global baseado em DEDS ainda necessita de estudos mais profundos. Neste primeiro estudo buscamos formalizar esta técnica como uma ferramenta de modelagem de sistemas. Entretanto, a codificação do modelo para alguma linguagem computacional ainda é realizada de forma heurística, não obedecendo a regras de codificação. Isto constitui um sério problema, pois, embora modelos de sistemas possam ser descritos por intermédio de ESMs de maneira relativamente fácil, estes não são facilmente codificáveis. Portanto, como pontos futuros a serem explorados podemos citar a pesquisa por técnicas de validação e projeto de controladores.

Modelos baseados em DEDS são apresentados na seção 10.1 desta dissertação. O item 6 do apêndice descreve como utilizar a implementação da simulação do caminhar de um esqueleto humano, contida no disquete anexo.

PARTE IV: CASOS DE ESTUDO

Na parte IV são introduzidos alguns pequenos exemplos que foram simulados através da utilização das técnicas de controle apresentadas nos capítulos anteriores. As implementações apresentadas nesta etapa encontram-se anexas a este trabalho em um disquete de computador.

CAPÍTULO 10: IMPLEMENTAÇÕES REALIZADAS

Não constitui interesse fundamental a este trabalho o desenvolvimento de um sistema de porte. Ao contrário, nosso interesse fundamenta-se no estudo de uma coleção de técnicas de controle de movimento, de modo a dispormos de um conjunto de ferramentas que sejam úteis à modelagem genérica de animações. Dessa forma, os exemplos apresentados neste capítulo constituem pequenas implementações que permitiram-nos avaliar a viabilidade das técnicas desenvolvidas.

Nas seções seguintes serão, portanto, apresentados os resultados obtidos neste trabalho.

10.1 - EXEMPLOS IMPLEMENTADOS ATRAVÉS DO MODELO DE DEDS

10.1.1 - SIMULAÇÃO DE UM ATOR BÍPEDE

De acordo com o modelo desenvolvido no cap. 9, implementamos o exemplo do bípede que é capaz de caminhar indefinidamente. A fig. 10.1 apresenta uma sequência de quadros que ilustra o movimento desta estrutura articulada.

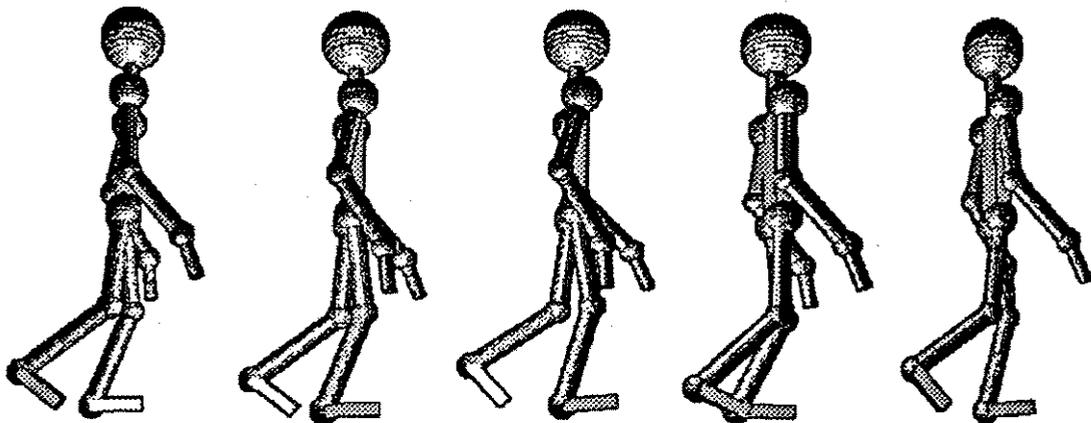


Fig. 10.1: Quadros gerados para o modelo do caminhar humano.

De acordo com a modularidade da linguagem de programação ANSI-C, foram implementadas Máquinas de Estados para o ator bípede como "sub-rotinas" de um "script" de animação. Neste caso, a sub-rotina "controle global" é responsável pelo controle de toda a estrutura, enviando "ordens" para as "sub-rotinas" dos braços e das pernas.

O item 6 do apêndice desta dissertação descreve como utilizar a implementação deste modelo presente no disquete anexo. As imagens deste exemplo foram geradas numa estação "Sun Sparc 2" através do sistema TOOKIMA, que foi comentado anteriormente.

10.1.2 - SIMULAÇÃO DE UM SISTEMA DE CRUZAMENTO DE RUAS

Como um segundo exemplo implementado através do modelo de DEDS temos a simulação de um sistema de cruzamento de ruas, conforme apresentado na fig. 10.2. Neste caso, dois semáforos devem controlar o fluxo de veículos que atravessam duas ruas perpendiculares. De acordo com as características de um sistema deste tipo, a qualquer instante de tempo podemos ter carros em ambas as ruas tentando atravessar o cruzamento. Portanto, sem os semáforos certamente observaríamos algumas (ou várias) colisões.

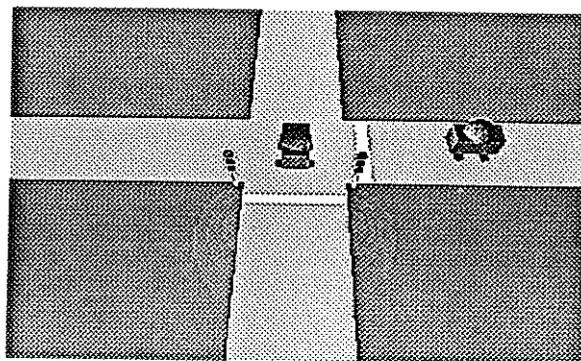


Fig. 10.2: Sistema de cruzamento de ruas.

Dessa forma, de acordo com o modelo de DEDS proposto, podemos ter como especificações básicas de controle do sistema as seguintes regras:

- 1-) Um carro movendo-se em uma das ruas pode acelerar se seu respectivo semáforo estiver apresentando luz verde.
- 2-) Um carro movendo-se em uma das ruas deve desacelerar se seu respectivo semáforo estiver apresentando luz amarela.
- 3-) Um carro movendo-se em uma das ruas deve desacelerar e parar defronte ao seu respectivo semáforo se este estiver apresentando luz vermelha.
- 4-) Se um semáforo estiver verde seu próximo estado será amarelo.
- 5-) Se um semáforo estiver amarelo seu próximo estado será vermelho.
- 6-) Se um semáforo estiver vermelho seu próximo estado será verde.
- 7-) Os dois semáforos presentes no cruzamento devem ter seus estados sincronizados.

Portanto, a partir das regras acima, podemos projetar ESMs para ambos os semáforos e para os carros que trafegam pelas ruas. Para tanto, vamos chamar as ruas do sistema de "Rua 1" e "Rua 2". Logo:

ESM SEMÁFORO NA RUA 1 (SEM1):

* Estados associados:

G_1 - Luz verde

Y_1 - Luz amarela

R_1 - Luz vermelha

* Eventos associados:

gy_1 - transição da luz verde para amarela

yr_1 - transição da luz amarela para vermelha

rg_1 - transição da luz vermelha para verde

* Canais associados:

m_1 - Envia informações para os carros que passam pela Rua 1

n_1 - Recebe informações do controlador

De forma análoga, para o semáforo na Rua 2 temos:

ESM SEMÁFORO NA RUA 2 (SEM2):

* Estados associados:

G_2 - Luz verde

Y_2 - Luz amarela

R_2 - Luz vermelha

* Eventos associados:

gy_2 - transição da luz verde para amarela

yr_2 - transição da luz amarela para vermelha

rg_2 - transição da luz vermelha para verde

* Canais associados:

m_2 - Envia informações para os carros que passam pela Rua 2

n_2 - Recebe informações do controlador

Analicamente, temos:

ESM SEM1:

$SEM1 = ([s_1], \emptyset, [m_1, n_1], [gy_1, yr_1, rg_1], A_{SEM1})$

$tipo(s_1) = [G_1, Y_1, R_1]$

$$A_{SEM1} = [[s_1 , (G_1 , VERDADE , m_1 | s_1 , G_1)] , \\ [s_1 , (Y_1 , VERDADE , m_1 | s_1 , Y_1)] , \\ [s_1 , (R_1 , VERDADE , m_1 | s_1 , R_1)] , \\ [s_1 , (G_1 , n_1 ? gy_1 , \emptyset , Y_1)] , \\ [s_1 , (Y_1 , n_1 ? yr_1 , \emptyset , R_1)] , \\ [s_1 , (R_1 , n_1 ? rg_1 , \emptyset , G_1)]]$$

ESM SEM2:

$$SEM2 = ([s_2] , \emptyset , [m_2, n_2] , [gy_2, yr_2, rg_2] , A_{SEM2})$$

$$tipo(s_2) = [G_2 , Y_2 , R_2]$$

$$A_{SEM2} = [[s_2 , (G_2 , VERDADE , m_2 | s_2 , G_2)] , \\ [s_2 , (Y_2 , VERDADE , m_2 | s_2 , Y_2)] , \\ [s_2 , (R_2 , VERDADE , m_2 | s_2 , R_2)] , \\ [s_2 , (G_2 , n_2 ? gy_2 , \emptyset , Y_2)] , \\ [s_2 , (Y_2 , n_2 ? yr_2 , \emptyset , R_2)] , \\ [s_2 , (R_2 , n_2 ? rg_2 , \emptyset , G_2)]]$$

Outra consideração importante a ser feita diz respeito ao intervalo de tempo de cada luz do semáforo. Vamos supor que o intervalo de tempo da luz amarela seja Δt_y , e para a luz verde Δt_g . Logo, o tempo de duração da luz vermelha deve ser $(\Delta t_y + \Delta t_g)$ e, podemos assumir que $\Delta t_y < \Delta t_g$.

Dessa forma, para o controlador que interage com ambos os semáforos temos:

* Eventos associados:

dt_y - passaram-se Δt_y unidades de tempo desde a última transição do controlador

dt_g - passaram-se Δt_g unidades de tempo desde a última transição do controlador

A definição analítica do controlador pode ser então dada por:

ESM CONTROLADOR:

$$CONTROLADOR = ([x_c] , \emptyset , [n_1, n_2] , [gy_1, yr_1, rg_1, gy_2, yr_2, rg_2] , A_{CTRL})$$

$$tipo(x_c) = [L_1 , L_2 , L_3 , L_4]$$

$$A_{CTRL} = [[[x_c , (L_1 , dt_y , (n_1 |yr_1 , n_2 |rg_2) , L_2)] , \\ [x_c , (L_2 , dt_g , n_2 |gy_2 , L_3)] , \\ [x_c , (L_3 , dt_y , (n_2 |yr_2 , n_1 |rg_1) , L_4)] , \\ [x_c , (L_4 , dt_g , n_1 |gy_1 , L)]]]$$

A estrutura deste sistema será completa com o projeto das ESMs para os carros que passam pelas ruas 1 e 2. Um carro trafegando em qualquer rua pode mover-se ou parar, dependendo do estado de seu semáforo. Dessa forma, temos associados os seguintes estados e eventos associados a um carro:

Para um carro na Rua 1 (CAR1):

* Estados associados:

M_1 - O carro está em movimento

S_1 - O carro está parado

* Eventos associados: gy_1 , yr_1 , rg_1

Para um carro na Rua 2 (CAR2):

* Estados associados:

M_2 - O carro está em movimento

S_2 - O carro está parado

* Eventos associados: gy_2 , yr_2 , rg_2

Analiticamente:

ESM CAR1:

$$CAR1 = ([c_1] , [] , m_1 , [gy_1, yr_1, rg_1] , A_{C1})$$

$$tipo(c_1) = [M_1 , S_1]$$

$$A_{C1} = [[c_1 , (S_1 , m_1 ?G_1 , [] , M_1)] , \\ [c_1 , (M_1 , m_1 ?Y_1 , [] , S_1)] , \\ [c_1 , (M_1 , m_1 ?R_1 , [] , S_1)]]$$

ESM CAR2:

$$CAR2 = ([c_2] , [] , m_2 , [gy_2, yr_2, rg_2] , A_{C2})$$

$$tipo(c_2) = [M_2 , S_2]$$

$$A_{C2} = \begin{bmatrix} [c_2, (S_2, m_2?G_2, [], M_2)] \\ [c_2, (M_2, m_2?Y_2, [], S_2)] \\ [c_2, (M_2, m_2?R_2, [], S_2)] \end{bmatrix}$$

Uma representação em diagrama de blocos para este sistema é apresentada na fig. 10.3.

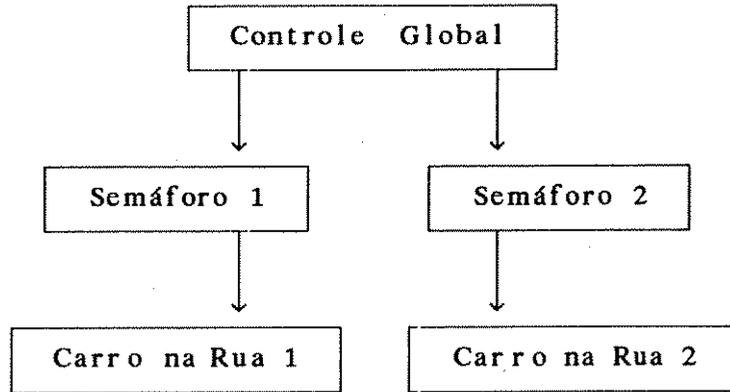


Fig. 10.3: Diagrama de blocos para o sistema de cruzamento de ruas.

Por sua vez, a fig. 10.4 representa a estrutura deste sistema sob a forma de ESMs.

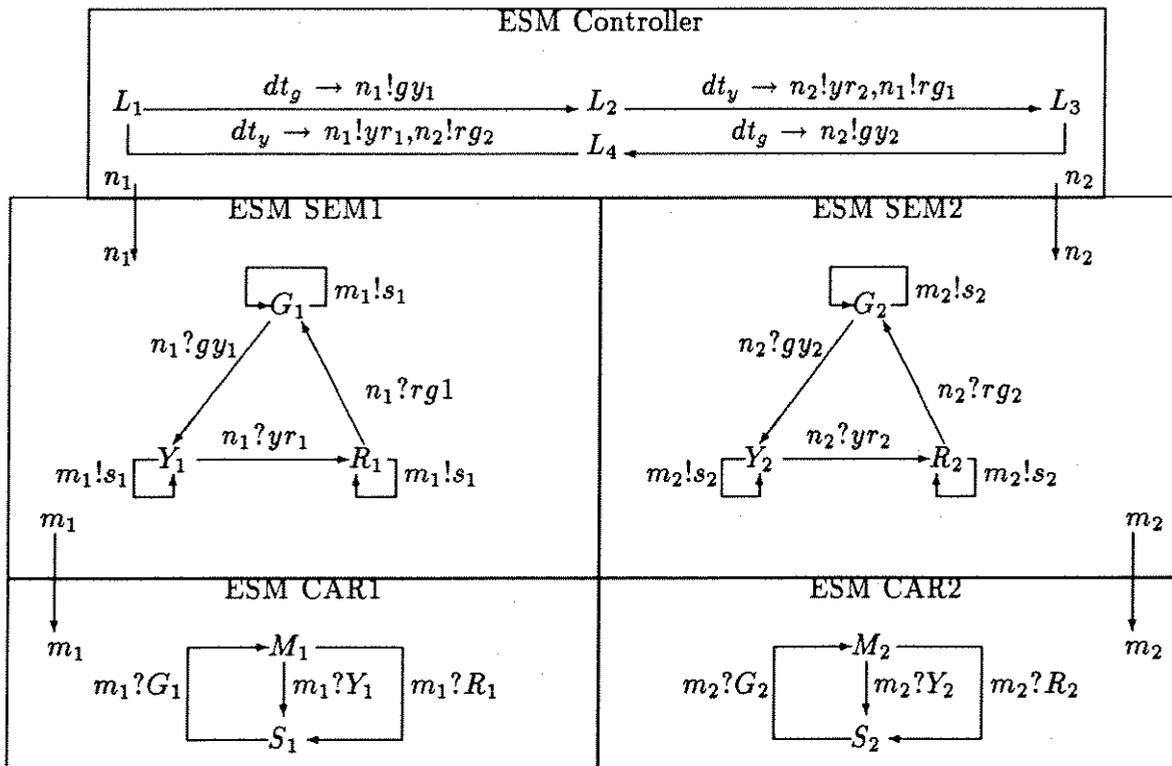


Fig. 10.4: Diagrama de Máquinas de Estados para o sistema.

Da mesma forma que para o exemplo da estrutura bípede, através da utilização do sistema TOOKIMA obtivemos as imagens seguintes, conforme apresentado na figura 10.5.

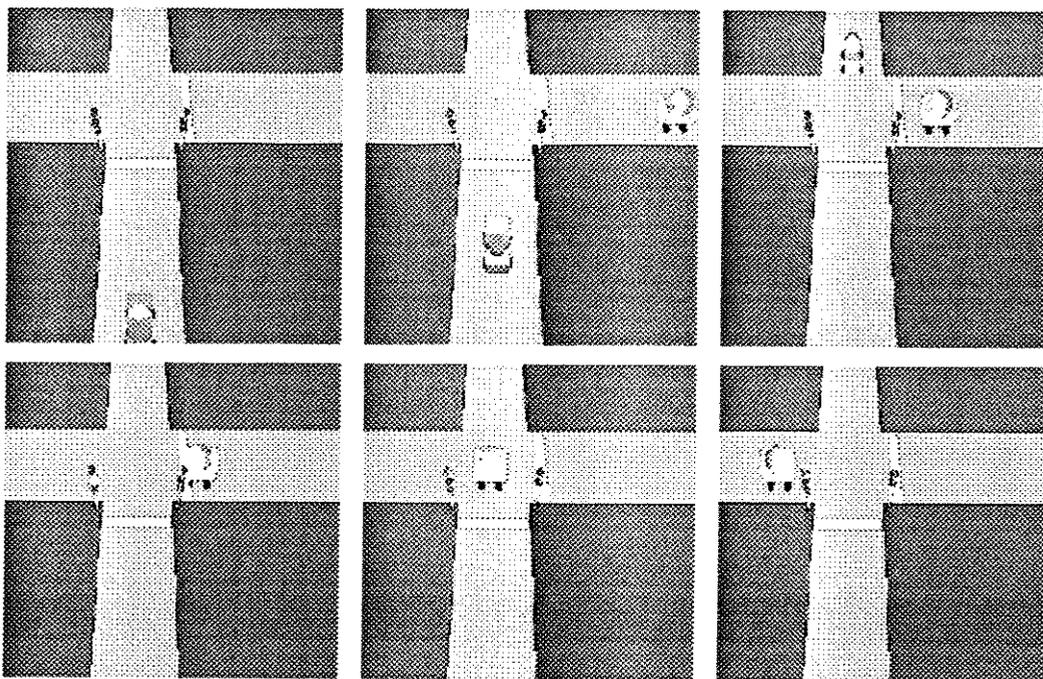


Fig. 10.5: Quadros gerados para o sistema de cruzamento de vias com o auxílio do TOOKIMA.

10.2 - PLANEJAMENTO DE TRILHAS ATRAVÉS DE CAMPOS DE POTENCIAIS

Ao contrário do exemplo desenvolvido na seção 10.1.2, onde o modelo de DEDS é utilizado como meio de controle de um sistema de cruzamento de vias, neste exemplo a prevenção de colisões é realizada através de um modelo de campos de potenciais.

Neste caso, conforme discutido no cap. 8, são associados potenciais aos "obstáculos" e à fronteira que compõe o sistema. Através da definição de um campo resultante, trilhas otimizadas são calculadas de acordo com algum critério de otimização, de tal forma que o objeto em movimento não colida com os obstáculos que compõem o ambiente.

Visto que a atribuição de potenciais aos obstáculos é arbitrária, podemos atribuir potenciais mais elevados (ou mais baixos) a determinados objetos. De forma, torna-se possível fazer com que o objeto em movimento afaste-se (ou aproxime-se) mais de determinado obstáculo em seu percurso.

As figuras 10.6 e 10.7 apresentam alguns exemplos de trilhas calculadas para um determinado ambiente. O item 5 do apêndice desta dissertação descreve como utilizar a implementação deste modelo presente no disquete anexo.

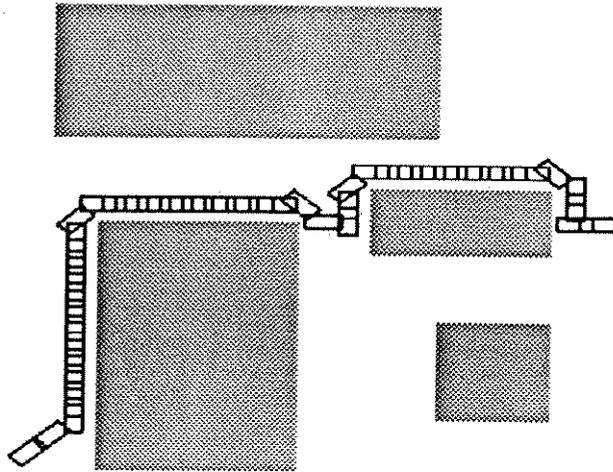


Fig. 10.6: Exemplo de trilha planejada através do modelo de campos de potenciais.

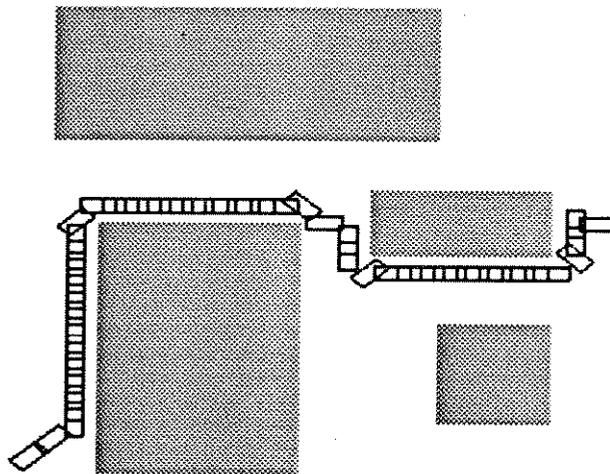


Fig. 10.7: Exemplo de trilha planejada através do modelo de campos de potenciais.

10.3 - SIMULAÇÃO DE UM BRAÇO ARTICULADO (ROBÔ) COM CINCO GRAUS DE LIBERDADE

A simulação apresentada nesta seção consiste na implementação do modelo de um robô YASUKAWA MOTOMAN L-3, apresentado em [Craig (1989)]. Através das técnicas desenvolvidas no capítulos 4, 5 e 6 torna-se possível desenvolver simulações interessantes para este robô bastante comum na indústria.

Visto que esta estrutura articulada apresenta cinco graus de liberdade, são então associados ao robô cinco sistemas de coordenadas variantes no tempo, conforme mostra a fig. 10.8. Obviamente, além dos cinco sistemas de coordenadas variantes no tempo, também é associado ao robô um sistema de coordenadas referencial inercial.

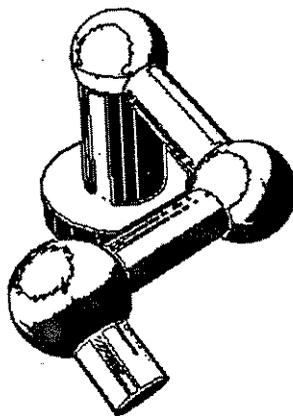


Fig. 10.8: Modelo estrutural de um robô YASUKAWA MOTOMAN L-3 [Craig (1989)].

De acordo com o modelo proposto, as transformações associadas entre os graus de liberdade do robô são dadas por:

$${}^0_1T = \begin{bmatrix} \cos\theta_1 & -\text{sen}\theta_1 & 0 & 0 \\ \text{sen}\theta_1 & \cos\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^1_2T = \begin{bmatrix} \cos\theta_2 & -\text{sen}\theta_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\text{sen}\theta_2 & -\cos\theta_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^2_3T = \begin{bmatrix} \cos\theta_3 & -\text{sen}\theta_3 & 0 & l_1 \\ \text{sen}\theta_3 & \cos\theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^3_4T = \begin{bmatrix} \cos\theta_4 & -\text{sen}\theta_4 & 0 & l_2 \\ \text{sen}\theta_4 & \cos\theta_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^4_5T = \begin{bmatrix} \cos\theta_5 & -\text{sen}\theta_5 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ \text{sen}\theta_5 & \cos\theta_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Onde: θ_1 , θ_2 , θ_3 , θ_4 e θ_5 representam os graus de liberdade do objeto e l_1 e l_2 representam o comprimento do primeiro e segundo segmentos, respectivamente.

Sendo que a concatenação destas matrizes nos leva à transformação entre o último S.C. do robô (extremo livre) e o S.C. referencial inercial:

$${}^0_5T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

onde:

$$r_{11} = \cos\theta_1 \cdot \cos(\theta_2 + \theta_3 + \theta_4) \cdot \cos\theta_5 - \text{sen}\theta_1 \cdot \text{sen}\theta_5$$

$$r_{21} = \text{sen}\theta_1 \cdot \cos(\theta_2 + \theta_3 + \theta_4) \cdot \cos\theta_5 + \cos\theta_1 \cdot \text{sen}\theta_5$$

$$r_{31} = -\text{sen}(\theta_2 + \theta_3 + \theta_4) \cdot \cos\theta_5$$

$$r_{12} = -\cos\theta_1 \cdot \cos(\theta_2 + \theta_3 + \theta_4) \cdot \text{sen}\theta_5 - \text{sen}\theta_1 \cdot \cos\theta_5$$

$$r_{22} = -\text{sen}\theta_1 \cdot \cos(\theta_2 + \theta_3 + \theta_4) \cdot \text{sen}\theta_5 + \cos\theta_1 \cdot \cos\theta_5$$

$$r_{32} = \text{sen}(\theta_2 + \theta_3 + \theta_4) \cdot \text{sen}\theta_5$$

$$r_{13} = \cos\theta_1 \cdot \text{sen}(\theta_2 + \theta_3 + \theta_4)$$

$$r_{23} = \text{sen}\theta_1 \cdot \text{sen}(\theta_2 + \theta_3 + \theta_4)$$

$$r_{33} = \cos(\theta_2 + \theta_3 + \theta_4)$$

$$p_x = \cos\theta_1 \cdot (l_1 \cdot \cos\theta_2 + l_2 \cdot \cos(\theta_2 + \theta_3))$$

$$p_y = \text{sen}\theta_1 \cdot (l_1 \cdot \cos\theta_2 + l_2 \cdot \cos(\theta_2 + \theta_3))$$

$$p_z = -l_1 \cdot \text{sen}\theta_2 - l_2 \cdot \text{sen}(\theta_2 + \theta_3)$$

As figuras 10.9 e 10.10 apresentam duas simulações obtidas com a implementação do modelo desta estrutura. O item 1 do apêndice desta dissertação descreve como utilizar a implementação do modelo cinemático inverso do robô, contida no disquete anexo.

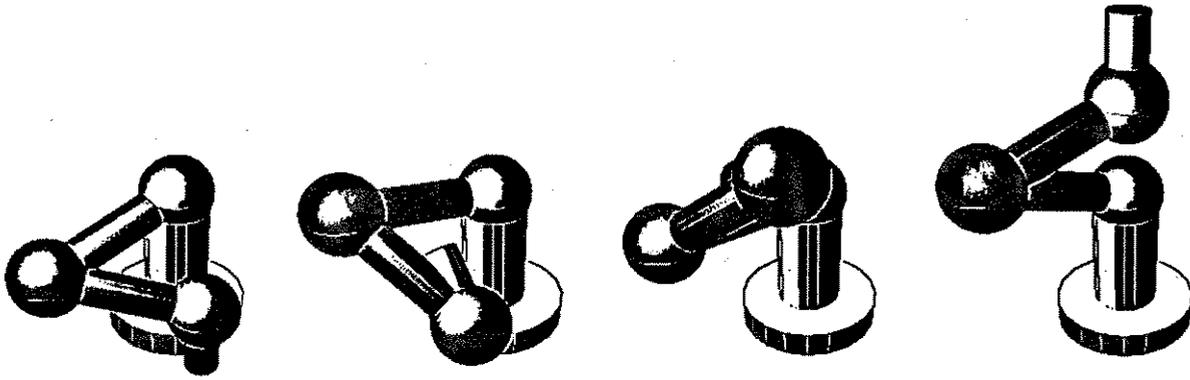


Fig. 10.9: Simulação de um robô YASUKAWA MOTOMAN L-3.

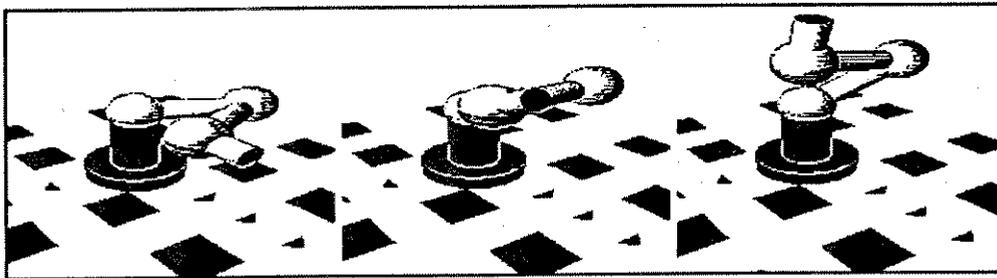


Fig. 10.10: Simulação de um robô YASUKAWA MOTOMAN L-3.

10.4 - ANIMAÇÃO DE UMA LUMINÁRIA LUXO

As ferramentas utilizadas para o desenvolvimento da simulação apresentada na seção 10.3 podem também ser utilizadas para a geração de animações convencionais (*cartoons*). Mais ainda, com algumas pequenas variações no modelo apresentado na seção anterior podemos construir uma "luminária Luxo" e transformá-la num verdadeiro personagem de desenhos animados. Um possível modelo para a Luxo é apresentado na fig. 10.11.

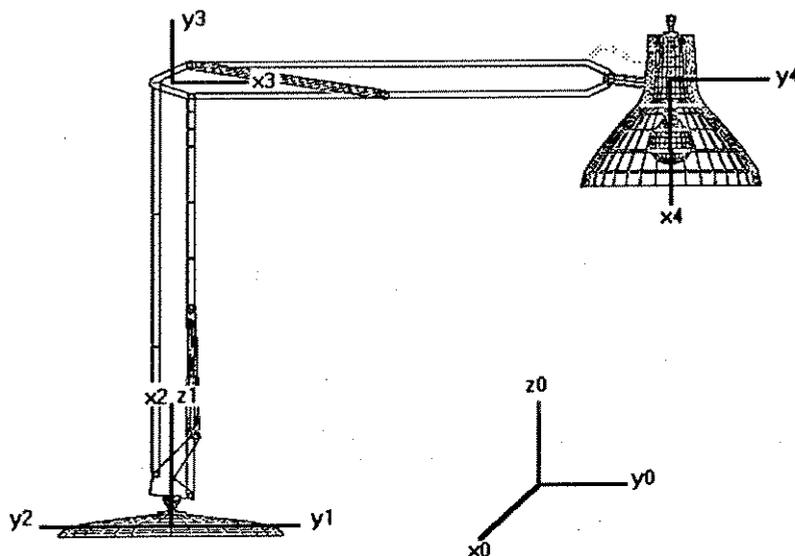


Fig. 10.11: Modelo para a luminária Luxo.

Note que, desta vez, a estrutura articulada apresenta apenas quatro graus de liberdade em juntas rotacionais. Entretanto, para dar "vida" ao modelo, vamos acrescentar mais três graus de liberdade translacionais. Dessa forma, as matrizes de transformação que definem o modelo para Luxo são dadas por:

$${}^0_1T = \begin{bmatrix} \cos\theta_1 & -\text{sen}\theta_1 & 0 & p1_x \\ \text{sen}\theta_1 & \cos\theta_1 & 0 & p1_y \\ 0 & 0 & 1 & p1_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^1_2T = \begin{bmatrix} \cos\theta_2 & -\text{sen}\theta_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\text{sen}\theta_2 & -\cos\theta_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^2_3T = \begin{bmatrix} \cos\theta_3 & -\text{sen}\theta_3 & 0 & l_1 \\ \text{sen}\theta_3 & \cos\theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^3_4T = \begin{bmatrix} \cos\theta_4 & -\text{sen}\theta_4 & 0 & l_2 \\ \text{sen}\theta_4 & \cos\theta_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Onde: $p1_x$, $p1_y$ e $p1_z$ representam os graus de liberdade translacionais do objeto.

As figuras 10.12 e 10.13 apresentam exemplos simulados para a luminária Luxo. O item 2 do apêndice desta dissertação descreve como utilizar a implementação deste modelo, contida no disquete anexo.



Fig. 10.12: Animação realizada com a luminária Luxo.

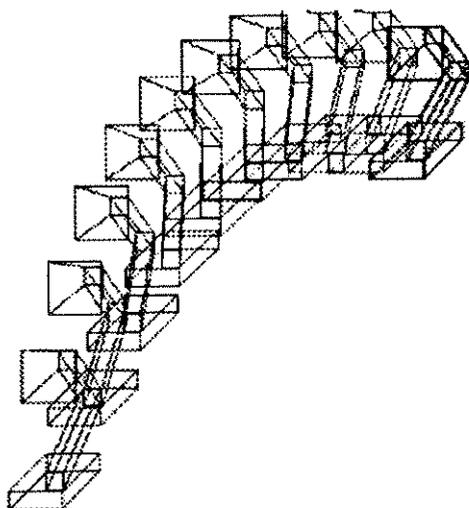


Fig. 10.13: Animação realizada com a luminária Luxo.

10.5 - SIMULAÇÃO DO MODELO DINÂMICO DE EULER

Uma simulação para o objeto da figura 6.3, descrito através das equações (6.3) a (6.6), foi implementado através da utilização do modelo dinâmico de Euler. O item 3 do apêndice desta dissertação descreve como utilizar a simulação contida no disquete anexo.

10.6 - MODELO DE SIMULAÇÃO SIMPLIFICADO PARA UMA MÃO

Na seção 10.1, um modelo de uma estrutura articulada "tipo árvore" foi simulado através do modelo de DEDS. Segundo este modelo a estrutura a ser simulada foi seccionada em quatro cadeias articuladas sem ramificações, no caso, dois braços e duas pernas. Entretanto, conforme apresentado nos capítulos 4, 5 e 6, também é possível desenvolver modelos articulados complexos através de técnicas cinemáticas e dinâmicas.

Nesta seção apresentamos um modelo simplificado para uma mão, construído a partir das técnicas apresentadas nos capítulos 4, 5 e 6. Para tanto, considere a fig. 10.14.

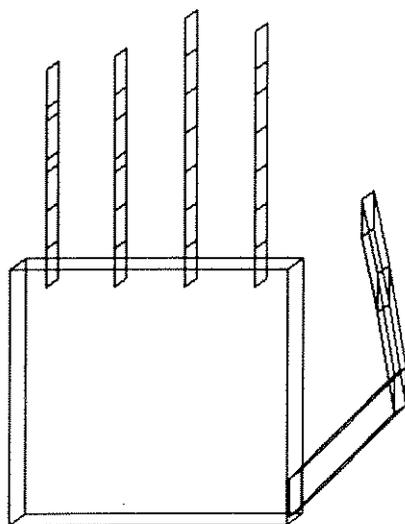


Fig. 10.14: Modelo simplificado para uma mão.

De acordo com a própria anatomia da mão humana, os dedos mínimo (D1), anular (D2), médio (D3) e indicador (D4) apresentam características muito semelhantes, distinguindo-se basicamente no comprimento e espessura das falanges. Por sua vez, o dedo polegar (D5) encontra-se rotacionado em relação aos demais dedos. Portanto, de acordo com o modelo simplificado apresentado na fig. 10.14, poderíamos definir o seguinte conjunto de transformações para esta estrutura articulada:

Para qualquer dedo "D_i", 0 ≤ i ≤ 4 :

$${}^0_1T(D_1) = \begin{bmatrix} \cos\theta_1 & -\text{sen}\theta_1 & 0 & x_0 \\ \text{sen}\theta_1 & \cos\theta_1 & 0 & y_0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^1_2T(D_1) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ \cos\theta_2 & -\text{sen}\theta_2 & 0 & 0 \\ \text{sen}\theta_2 & \cos\theta_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^2_3T(D_1) = \begin{bmatrix} \cos\theta_3 & -\text{sen}\theta_3 & 0 & l_1 \\ \text{sen}\theta_3 & \cos\theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^3_4T(D_1) = \begin{bmatrix} \cos\theta_4 & -\text{sen}\theta_4 & 0 & l_2 \\ \text{sen}\theta_4 & \cos\theta_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

e, para o dedo polegar (D5):

$${}^0_1T(D_5) = \begin{bmatrix} 0 & 0 & -1 & x_0 \\ \text{sen}\theta_1 & \cos\theta_1 & 0 & y_0 \\ \cos\theta_1 & -\text{sen}\theta_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^1_2T(D_5) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ \cos\theta_2 & -\text{sen}\theta_2 & 0 & 0 \\ \text{sen}\theta_2 & \cos\theta_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^2_3T(D_5) = \begin{bmatrix} \cos\theta_3 & -\text{sen}\theta_3 & 0 & l_1 \\ \text{sen}\theta_3 & \cos\theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^3_4T(D_5) = \begin{bmatrix} \cos\theta_4 & -\text{sen}\theta_4 & 0 & l_2 \\ \text{sen}\theta_4 & \cos\theta_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

sendo que:

x_0 e y_0 representam o desvio da base do respectivo dedo em relação ao referencial,

θ_1 , θ_2 , θ_3 e θ_4 representam as variáveis de junta e

l_1 e l_2 representam os comprimentos das primeira e segunda falanges.

As figuras 10.15 e 10.16 representam duas simulações realizadas com este modelo. O item 4 do apêndice desta dissertação descreve como utilizar a implementação contida em disquete.

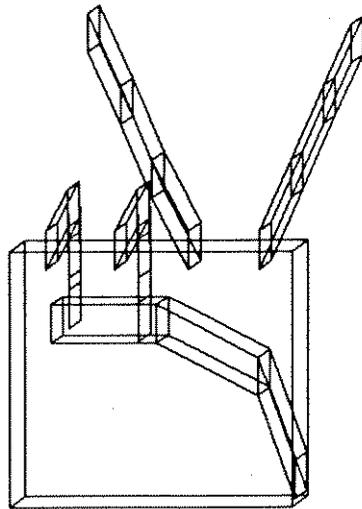


Fig. 10.15: Simulação do modelo simplificado para uma mão humana.

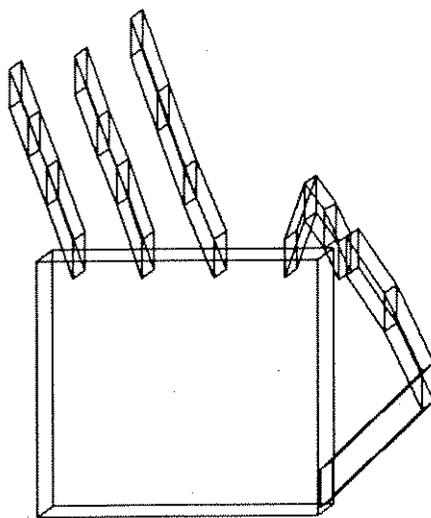


Fig. 10.16: Simulação do modelo simplificado para uma mão.

**PARTE V:
CONCLUSÕES E BIBLIOGRAFIA**

CAPÍTULO 11: CONCLUSÕES GERAIS

11.1 - CONSIDERAÇÕES GERAIS

O processo de modelagem de uma animação implica no desenvolvimento de ferramentas que representem o comportamento do sistema em análise. Existe, na prática, um grande número de sistemas com diferentes comportamentos e, portanto, diferentes modelos devem ser implementados. Dessa forma, podemos afirmar que não existe uma solução única que represente qualquer tipo de sistema, ou seja, ferramentas de animação específicas são necessárias para cada tipo de sistema.

Um "pacote" de animação que possa ser aplicado a qualquer tipo de sistema é irrealizável na prática. Tal sistema seria extremamente custoso, grande em demasia e, talvez, extremamente difícil de ser operado. Assim sendo, uma opção mais racional seria o estudo de casos isolados, tais como objetos rígidos, objetos articulados, corpos deformáveis, objetos sujeitos a restrições, etc. Uma vez que o modelo para o sistema em questão tenha sido desenvolvido, com o auxílio de uma ferramenta para a representação "visual" da sequência de movimentos, podemos obter o produto final desejado que é a visualização do sistema modelado. Esta é a filosofia que tem orientado nosso trabalho.

Este tipo de arquitetura aberta é extremamente interessante, visto que sempre um novo modelo pode ser adicionado à biblioteca de modelos e visualizado através do TOOKIMA ou qualquer outro sistema de animação. Dentro desta filosofia esperamos que este trabalho venha a servir como "semente" para novas idéias e perspectivas neste ramo de pesquisa.

11.2 - RESULTADOS APRESENTADOS

Os diversos modelos apresentados nesta dissertação constituem uma "biblioteca de técnicas de controle de movimento" que podem ser utilizadas para representar determinados sistemas físicos. Por sua vez, através do "pacote" TOOKIMA, citado anteriormente, promovemos a "tradução" dos modelos desenvolvidos sob a forma de equações matemáticas e programas para a forma visual desejada.

As formulações de controle local e controle global em animação constituem uma poderosa estratégia para a definição de animações complexas. Tal estratégia permite ao animador desenvolver suas simulações de maneira mais clara e precisa. Entretanto, para representarmos um modelo de maneira correta e, principalmente, concisa, devemos dispor de ferramentas adequadas para a representação dos sistemas a serem simulados. Para tanto, apresentamos um conjunto de convenções extremamente úteis aos nossos propósitos no capítulo 3.

Por sua vez, o capítulo 4 apresenta um modelo cinemático simples e eficiente para o desenvolvimento de animações de cadeias articuladas. O modelo apresentado permite ao animador gerar animações onde não exista qualquer compromisso com a realidade física que nos cerca e, também, gerar simulações físicas como, por exemplo, a simulação do movimento de um robô. Contudo, o modelo cinemático dispõe de um conjunto limitado de variáveis de estado do sistema a ser simulado, o que pode implicar em um menor controle da simulação por parte do animador.

Dessa forma, o modelo dinâmico, proposto no capítulo 5, vem a complementar o modelo cinemático. Neste modelo, o número de variáveis de estado a ser considerado é relativamente grande, o que permite ao animador manter um maior controle sobre o estado de sua simulação. Entretanto, o acréscimo no número de variáveis de estado do sistema também implica em maior dificuldade para a "operação" do sistema.

Através da generalização do modelo dinâmico apresentado no capítulo 5, sugerimos, no capítulo 6, um modelo dinâmico capaz de simular cadeias articuladas ramificadas. Grande parte das estruturas articuladas de nosso interesse são constituídas por cadeias ramificadas e, portanto, necessitam de um modelo apropriado para a geração de suas simulações.

Notemos que, um modelo capaz de tratar estruturas ramificadas pode ser diretamente formulado a partir daqueles apresentados nos capítulos 4 e 5, bastando, para tanto, acrescentar mais algumas restrições ao sistema (por exemplo, alguma "força de restrição" deve atuar sobre um conjunto de segmentos que devem permanecer unidos através de uma junta). Descrever animações através de modelos cinemáticos ou dinâmicos associados a um conjunto de restrições ao movimento desejado passou a ser, em meados da década de 80, uma das mais comuns ferramentas de animação.

A definição de "restrições em animação", por sua vez, nos orienta em direção às "técnicas de controle ótimo" em animação. Nas técnicas apresentadas até o capítulo 6, a solução para o problema do cálculo do movimento desejado encontra-se em um conjunto qualquer de variáveis de estado capaz de satisfazer às restrições impostas pelo animador. No capítulo 7, entretanto, sugerimos o refinamento das técnicas anteriormente propostas através da introdução do conceito de "controle ótimo" em animação. Dessa forma, o conjunto de variáveis de estado que satisfaz o movimento desejado para um objeto deve, também, satisfazer a critérios de otimização de determinados parâmetros da animação. Contudo, conforme ainda discutido no capítulo 7, esta técnica nos leva a um grau de automação da animação muito elevado, o que pode vir a reduzir a interferência do animador na simulação, tornando mais complexa a definição dos movimentos desejados.

Este problema pode ser contornado através da utilização das técnicas propostas no capítulo 8. O planejamento prévio da trilha a ser seguida por um objeto nos leva a um modelo (cinemático ou dinâmico) mais simples, sem que seja comprometida a validade do modelo. Problemas comuns em animação, tais como a prevenção e tratamento de colisões podem ser tratados de forma muito mais confortável através do planejamento prévio do movimento a ser realizado. O mapeamento do ambiente da animação através de campos de potencial e o algoritmo para a otimização de trilhas são ferramentas simples e eficientes para o tratamento deste tipo de problema.

Por outro lado, o tratamento das interações entre objetos numa animação constitui um problema de controle relativamente complexo. Para reduzirmos a complexidade de sistemas onde existe um alto grau de interação entre os objetos que os compõem, optamos por definir níveis de controle na animação. Dessa forma, um nível (ou bloco) de controle local deve ser associado a cada objeto que compõe o sistema a ser simulado, de modo a determinar as características que definem o movimento de cada objeto. Por sua vez, um nível (ou bloco) de controle global deve gerenciar as interações existente entre os objetos que fizerem parte do sistema.

A estrutura de um bloco de controle local sugere uma formulação matemática para o mesmo, como, por exemplo, os modelos cinemáticos e dinâmicos propostos anteriormente. Por sua vez, o bloco de controle global da animação sugere uma formulação lógica para o tratamento das interações entre objetos e entre estes e o animador. Dessa forma propomos, no capítulo 9, um modelo de sistema a eventos discretos capaz de modelar animações de diversos tipos. Especificamente, o modelo de máquinas de estados

estendida parece-nos adequado à formulação do bloco de controle global de uma animação. Entretanto, a utilização das ESM como ferramenta de modelo de animação necessita de estudos mais profundos, pois a codificação dos modelos em linguagem de computador não é simples.

Embora tenhamos definido capítulos específicos para as técnicas de controle local e global, existe uma certa flexibilidade nas técnicas apresentadas. É o caso das técnicas apresentadas nos capítulos de 3 a 8. Embora tenhamos apresentado tais técnicas de controle como "técnicas de controle local", é possível reutilizá-las para o controle global de uma animação. O exemplo de simulação da mão humana caracteriza esta perspectiva. Podemos desenvolver a simulação dos dedos através de um único modelo cinemático ou dinâmico, assim como é possível desenvolver um modelo através da sub-divisão de blocos de controle local e um bloco de controle global.

11.3 - CONTRIBUIÇÕES DESTE TRABALHO

No conjunto de técnicas apresentadas na Parte II encontramos a primeira contribuição deste trabalho. Embora tais técnicas já tenham sido divulgadas com frequência em países da Europa e América do Norte, o Brasil ainda carece de trabalhos que tratem deste assunto. Tal fato nos motivou a apresentar um tutorial sobre técnicas de controle de movimento em animação no VI SIBGRAPI (Simpósio Brasileiro de Computação Gráfica e Processamento de Imagens), em Recife, PE, no ano de 1993, conforme descrito em [Camargo (1993)].

As técnicas descritas na Parte II se encontram intimamente relacionadas ao paradigma da "Animação Orientada a Restrições de Espaço e Tempo". Podemos também afirmar que este paradigma vem sendo utilizado por grande parte dos "cientistas da animação". Entretanto, descrever algumas animações através do paradigma das restrições de espaço e tempo constitui uma tarefa extremamente complexa, como no caso do modelo de simulação do caminhar de um bípede. Neste exemplo, a divisão da animação em níveis de controle local e global contribui para a redução da complexidade do problema, além de permitir o desenvolvimento de um modelo bastante flexível, que pode atender a uma grande variedade de restrições.

Portanto, embora o conceito de controle local e controle global não seja inédito, a utilização de ferramentas de modelagem de sistemas dinâmicos a eventos discretos, como, por exemplo, as ESMs, constitui uma importante contribuição no sentido de se estabelecer um paradigma complementar ao das restrições de espaço e tempo. Dessa forma, o paradigma da "Animação Orientada a Eventos Discretos" constitui uma ferramenta de modelagem de animações extremamente eficaz para a solução de determinados problemas.

11.4 - PERSPECTIVA FUTURA

O desenvolvimento de ferramentas para modelagem de animações orientadas a eventos discretos constitui um vasto campo de pesquisa. Portanto, pretendemos realizar como etapa seguinte a este trabalho, um estudo mais abrangente sobre modelagem de sistemas de animação a eventos discretos. Especificamente, pretendemos abordar o problema do desenvolvimento de uma linguagem, para a formalização da descrição de sistemas, além de uma interface, para o desenvolvimento de animações.

BIBLIOGRAFIA

- [Amorim (1992)] - AMORIM, G et alli; "*KIKO: Sistema de Animação para Representação de Movimento Humano.*"; V SIBGRAPI - Simpósio Brasileiro de Computação Gráfica e Processamento de Imagens, pp. 243-250; Novembro, 1992.
- [Armstrong (1985)] - ARMSTRONG, W. W. et alli; "*The dynamics of articulated rigid bodies for the purpose of animation.*"; The Visual Computer, v. 1, n. 4, pp. 231-240; Springer-Verlag, December, 1985.
- [Badler (1993)] - BADLER, N. I. et alli; "*Simulating Humans: Computer Graphics, Animation, and Control.*"; Oxford University Press, June, 1993.
- [Barzel (1988)] - BARZEL, R. et alli; "*A modeling system based on dynamic constraints.*"; Computer Graphics, v. 22, n. 4, pp. 179-188; August, 1988.
- [Bergman (1993)] - BERGMAN, L. D. et alli; "*VIEW - An Exploratory Molecular Visualization System with User-Definable Interaction Sequences.*"; SIGGRAPH'93 Conference, pp. 117-126; August, 1993.
- [Boyce (1985)] - BOYCE, W. E. et alli; "*Equações diferenciais elementares e problemas de valores de contorno.*"; Guanabara Dois, pp. 315-353; 1985.
- [Brotman (1988)] - BROTMAN, L. S. et alli; "*Motion Interpolation by optimal control.*"; Computer Graphics, v. 22, n. 4, pp. 309-315; August, 1988.
- [Camargo (1993)] - CAMARGO, J. T. F. et alli; "*Técnicas de Controle em Animação - Um Tutorial.*"; Tutorial apresentado durante o VI SIBGRAPI - Simpósio Brasileiro de Computação Gráfica e Processamento de Imagens; Outubro, 1993.
- [Camargo (1994)] - CAMARGO, J. T. F. et alli; "*Modeling Motion Simulation With DEDS.*"; 13 th. IFIP Congress'94 - International Federation For Information Processing; August, 1994.
- [Cameron (1990)] - CAMERON, S.; "*Collision Detection by Four-Dimensional Intersection Testing.*"; IEEE Trans. on Robotics and Automation, v. 6, n. 3, pp. 291-303; June, 1990.
- [Cohen (1989)] - COHEN, G. et alli; "*Algebraic tools for the performance evaluation of discrete event systems.*"; Proc. IEEE, v. 77, n. 1, pp. 39-57; January, 1989.
- [Craig (1989)] - CRAIG, J. J.; "*Introduction to robotics: mechanics and control.*"; 2nd ed, Addison-Wesley Publishing Company, 1989.
- [Crowley (1985)] - CROWLEY, J. L.; "*Navigation for an Intelligent Mobile Robot*"; IEEE Journal of Robotics and Automation, v. RA-1, n. 1, pp. 31-41; March, 1985.
- [D'Eleuterio (1991)] - D'ELEUTERIO, G. M. T. et alli; "*The relationship Between Recursive Multibody Dynamics and Discrete-Time Optimal Control.*"; IEEE Journal of Robotics and Automation, v. 7, n. 6, pp. 743-749; December, 1991.
- [Fardanesh (1992)] - FARDANESH, B. et alli; "*A New Model-Based Tracking Controller for Robot Manipulators Using Trajectory Pattern Inverse Dynamics*"; IEEE Trans. on Robotics and Automation, v. 8, n. 2, pp. 279-285; April, 1992.

- [Feijó (1993)] - FEIJÓ, B. et alli; "*Animação Comportamental Baseada em Lógica*"; VI SIBGRAPI - Simpósio Brasileiro de Computação Gráfica e Processamento de Imagens, pp. 117-122; Outubro, 1993.
- [Gewali (1990)] - GEWALI, L. P. et alli; "*Path Planning in the Presence of Vertical Obstacles*"; IEEE Trans. on Robotics and Automation, v. 6, n. 3, pp. 331-341; June, 1990.
- [Gilbert (1985)] - GILBERT, E. G. et alli; "*Distance Functions and Their Application to Robot Path Planning in the Presence of Obstacles.*"; IEEE Journal of Robotics and Automation, v. RA-1, n. 1, pp. 21-31; March, 1985.
- [Gill (1981)] - GILL, P. E. et alli; "*Practical Optimization.*"; Academic Press Inc. Ltd., 1989.
- [Girard (1985)] - GIRARD, M. et alli; "*Computational modeling for the computer animation of legged figures.*"; ACM Computer Graphics, v. 19, n. 3, pp. 263-270; July, 1985.
- [Goldenberg (1985)] - GOLDENBERG, A. A. et alli; "*A complete generalized solution to the inverse kinematics of robots.*"; IEEE Journal of Robotics and Automation, v. RA-1, n. 2, pp. 14-20; March, 1985.
- [Gondran (1986)] - GONDRAN, M. et alli; "*Graphs and algorithms.*"; John Wiley and Sons; 1986.
- [Hayt (1981)] - HAYT Jr., W. H.; "*Engineering electromagnetics.*"; McGraw-Hill Inc., 1981.
- [Ho (1989)] - HO, Y.; "*Dynamics of Discrete Event Systems.*"; Proc. IEEE, v. 77, n. 1, pp. 3-6; January, 1989.
- [Hounsell (1992)] - HOUNSELL, M. S. et alli; "*TOOKIMA - animação cinemática.*"; V SIBGRAPI - Simpósio Brasileiro de Computação Gráfica e Processamento de Imagens; Novembro, 1992.
- [Hwang (1992)] - HWANG, Y. K. et alli; "*A Potential Field Approach to Path Planning.*"; IEEE Trans. on Robotics and Automation, v. 8, n. 1, pp. 23-32; February, 1992.
- [Isaacs (1987)] - ISAACS, P. M. et alli; "*Controlling dynamic simulation with kinematic constraints, behavior functions and inverse dynamics.*"; ACM Computer Graphics, v. 21, n. 4, pp. 215-224; July, 1987.
- [Isaacs (1988)] - ISAACS, P. M. et alli; "*Mixed methods for complex kinematic constraints in dynamic figure animation.*"; Visual Computer, v. 4, n. 6, pp. 296-305; 1988.
- [D.Kalra (1992)] - KALRA, D. et alli; "*Modeling with time and events in computer animation.*"; EUROGRAPHICS'92, v. 11, n. 3, pp. 45-58; 1992.
- [P.Kalra (1992)] - KALRA, P. et alli; "*Simulation of Facial Muscle Actions Based on Rational Free Form Deformations.*"; EUROGRAPHICS'92, v. 11, n. 3, pp. 59-69; 1992.
- [Korein (1982)] - KOREIN, J. U. et alli; "*Techniques for generating the goal-directed motion of articulated structures.*"; IEEE Computer Graphics and Applications, pp. 71-81; November, 1982.

- [Lemos (1992)] - LEMOS, R. R. et alli; "*Animação de Corpos Rígidos e Tratamento de Colisões Utilizando Análise Dinâmica.*"; V SIBGRAPI - Simpósio Brasileiro de Computação Gráfica e Processamento de Imagens, pp. 79-82; Novembro, 1992.
- [McInnis (1986)] - MCINNIS, B. C. et alli; "*Kinematics and dynamics in robotics: a tutorial based upon classical concepts of vectorial mechanics.*"; IEEE Journal of Robotics and Automation, v. RA-2, n. 4, pp. 181-187; December, 1986.
- [Morgan (1985)] - MORGAN, R. G. et alli; "*A Decentralized Variable Structure Control Algorithm for Robotic Manipulators*"; IEEE Journal of Robotics and Automation, v. RA-1, n. 1, pp. 57-65; March, 1985.
- [Ngo (1993)] - NGO, J. T. et alli; "*Spacetime Constraints Revised.*"; SIGGRAPH'93 Conference, pp. 343-350; August, 1993.
- [Ostroff (1989)] - OSTROFF, J. S.; "*Temporal logic for real time systems.*"; Research Studies Press Ltd., John Wiley & Sons Inc.; 1989.
- [Panne (1990)] - van de PANNE, M. et alli; "*Reusable motion synthesis using state-space controllers.*"; ACM Computer Graphics, v. 24. n. 4, pp. 225-234; August, 1990.
- [Panne (1993)] - van de PANNE, M. et alli; "*Sensor-Actuator Networks.*"; SIGGRAPH'93 Conference, pp. 335-342; August, 1993.
- [Paul (1981)] - PAUL, R.; "*Robot Manipulators: Mathematics, Programming and Control*"; MIT Press; 1981.
- [Prusinkiewicz (1993)] - PRUSINKIEWICZ, P. et alli; "*Animation of Plant Development.*"; SIGGRAPH'93 Conference, pp. 351-360; August, 1993.
- [Rigo (1993)] - RIGO, S. J. et alli; "*Desenvolvimento de um Sistema Gráfico Interativo para Modelagem e Animação Facial.*"; VI SIBGRAPI - Simpósio Brasileiro de Computação Gráfica e Processamento de Imagens, Volume de Comunicações, pp. 107-110; Outubro, 1993.
- [Rodrigues (1993)] - RODRIGUES, M. A. F. et alli; "*ANIMADO: Um Protótipo de um Sistema de Animação Modelado por Dinâmica*"; VI SIBGRAPI - Simpósio Brasileiro de Computação Gráfica e Processamento de Imagens, pp. 107-115; Outubro, 1993.
- [Sakas (1992)] - SAKAS, G. et alli; "*A Functional Approach to the Visual Simulation of Gaseous Turbulence.*"; EUROGRAPHICS'92, v. 11, n. 3, pp. 107- 117; 1992.
- [Schröder (1990)] - SCHRÖDER, P. et alli; "*The Virtual Erector Set: Dynamic Simulation with Linear Recursive Constraint Propagation*"; ACM Computer Graphics, v. 24, n. 4, pp. 23-31; August, 1990.
- [Shaffer (1992)] - SHAFFER, C. A. et alli; "*A Real-Time Robot Arm Collision Avoidance System.*"; IEEE Trans. on Robotics and Automation, v. 8, n. 2, pp. 149-160; April, 1992.
- [Shinya (1992)] - SHINYA, M. et alli; "*Stochastic Motion - Motion Under the Influence of Wind.*"; EUROGRAPHICS'92, v. 11, n. 3, pp. 119-128; 1992.
- [Silva (1992)] - SILVA, R. L. et alli; "*ANIMAKER: Sistema de Animação Modelada.*"; V SIBGRAPI - Simpósio Brasileiro de Computação Gráfica e Processamento de Imagens, pp. 259-268; Novembro, 1992.

- [Stam (1993)] - STAM, J. et alli; "*Turbulent Wind Fields for Gaseous Phenomena.*"; SIGGRAPH'93 Conference, pp. 369-376; August, 1993.
- [Takahashi (1989)] - TAKAHASHI, O. et alli; "*Motion Planning in a Plane Using Generalized Voronoi Diagrams*"; IEEE Trans. on Robotics and Automation, v. 5, n. 2, pp. 143-151; April, 1989.
- [Taylor (1993)] - TAYLOR, R. M. et alli; "*The NANOMANIPULATOR: A Virtual- Reality Interface for a Scanning Tunneling Microscope.*"; SIGGRAPH'93 Conference, pp. 127-134; August, 1993.
- [Thalmann (1985)] - THALMANN, N. M. et alli; "*Computer Animation: Theory and Practice*"; Springer-Verlag; 1985.
- [Thalmann (1991)] - THALMANN, N. M. et alli; "*Complex Models for Animating Synthetic Actors*"; IEEE Computer Graphics & Applications, pp. 32-44; September, 1991.
- [Watt (1993)] - WATT, A. et alli; "*Scripting and Artistic Control in Computer Animation*"; VI SIBGRAPI - Simpósio Brasileiro de Computação Gráfica e Processamento de Imagens, pp. 123-132; Outubro, 1993.
- [Wilhelms (1988)] - WILHELMS, J. et alli; "*Dynamic animation: interaction and control*"; Visual Computer, v. 4, n. 6, pp. 283-295; 1988.
- [Witkin (1987)] - WITKIN, A. et alli; "*Energy Constraints On Parametrized Models*"; Computer Graphics, v. 21, n. 4, pp. 225-232; July, 1987.
- [Witkin (1988)] - WITKIN, A. et alli; "*Spacetime constraints.*"; ACM Computer Graphics, v. 22, n. 4, pp. 159-168; August, 1988.
- [Witkin (1990)a] - WITKIN, A. et alli; "*Fast Animation and Control of Nonrigid Structures*"; Computer Graphics, v. 24, n. 4, pp. 243-252; August, 1990.
- [Witkin (1990)b] - WITKIN, A. et alli; "*Interactive Dynamics.*"; Computer Graphics, v. 24, n. 2, pp. 11-21; March, 1990.
- [Zelevnik (1991)] - ZELEVNIK, R. C. et alli; "*An Object-Oriented Framework for the Integration of Interactive Animation Techniques*"; Computer Graphics, v. 25, n. 4, pp. 105-112; July, 1991.

APÊNDICE: UTILIZAÇÃO DO DISQUETE DE SIMULAÇÕES

Neste apêndice apresentaremos, resumidamente, alguns exemplos que foram implementados através da utilização das técnicas discutidas nesta dissertação. Os exemplos aqui discutidos encontram-se também disponíveis sob a forma de programas, em um disquete anexo ao exemplar desta dissertação. Este disquete contém os programas-fonte escritos em linguagem "C", além de arquivos executáveis sob o sistema operacional DOS.

Aconselha-se a utilização de um microcomputador PC-AT "386" com co-processador ou PC-AT "486". Além disso, sugere-se a utilização de um monitor VGA.

Os programas executáveis são auto-explicativos o suficiente, de modo a permitir que qualquer pessoa possa utilizar o sistema. Os arquivos-fonte possuem comentários que permitem aos usuários introduzir modificações no sistema para novas experiências.

1 - MODÉLO CINEMÁTICO INVERSO PARA OBJETOS ARTICULADOS

O sistema "Inv-Kin" constitui um pequeno exemplo de sistema para a simulação do movimento de objetos articulados. O modelo utilizado neste exemplo deriva das técnicas apresentadas no capítulo 4. Para a utilização deste "pacote", descomprima o arquivo INV_KIN.ZIP. Este sistema dispõe dos seguintes arquivos:

- INV_KIN.C : Dada a posição e orientação do último segmento de um objeto articulado, em relação ao sistema de coordenadas referencial, são calculadas, iterativamente, todas as variáveis de junta desta estrutura no estado desejado. O controle do processo iterativo é "manual", ou seja, o usuário controla, passo-a-passo, a convergência para alguma solução. Algum arquivo com o modelo da estrutura articulada deve ser fornecido como entrada, para este exemplo temos o arquivo YASUKAWA.H.

- INV_KIN.EXE : Arquivo executável relativo ao programa INV_KIN.

- YASUKAWA.H : Modelo para um robô do tipo Yasukawa Motoman L-3, definido através de "equações de resíduos". Este arquivo serve como entrada para os programas INV_KIN e AUTO_GEN.

- AUTO_GEN.C : Uma vez especificada uma trajetória para o último segmento de uma estrutura articulada, este programa gera a respectiva trajetória das variáveis de junta do robô. Necessita de um arquivo de entrada que contenha o modelo da estrutura articulada. O resultado desta simulação (trajetória das variáveis de junta) é armazenado no arquivo JOINT.OUT.

- AUTO_GEN.EXE : Arquivo executável relativo ao programa AUTO_GEN.

- JOINT.OUT : Arquivo gerado pelo programa AUTO_GEN. Contém a trajetória das variáveis de junta para uma simulação especificada no programa AUTO_GEN. Este arquivo serve como entrada para o programa FRAME, que fará a conversão das variáveis de junta para o espaço 3-D.

- JOINT1.OUT, JOINT2.OUT e JOINT3.OUT : Constituem também simulações que foram geradas pelo programa AUTO_GEN.

- FRAME.C : Realiza a conversão da trajetória de cada uma das variáveis de junta do robô para o espaço tridimensional. Necessita, como entrada, de algum arquivo de

trajetória de juntas. Este programa gera, como saída, um arquivo que contém a conversão especificada, denominado MOTION.OUT.

- FRAME.EXE : Arquivo executável relativo ao programa FRAME.
- MOTION.OUT : Arquivo gerado pelo programa FRAME, que contém a descrição da trajetória das variáveis de junta do robô no espaço 3-D. Este arquivo serve como entrada ao programa VIS_SIM.
- MOTION1.OUT, MOTION2.OUT e MOTION3.OUT : Idem ao arquivo MOTION.OUT.
- VIS_SIM.C : Visualiza o movimento da estrutura articulada que está contido nos arquivos MOTION.OUT, MOTION1.OUT, etc.
- EGAVGA.BGI : "Driver" gráfico.

2 - MODELO DE ANIMAÇÃO PARA UMA LUMINÁRIA "LUXO"

Este sistema simula o movimento de uma luminária tipo "Luxo". O modelo implementado deriva, parcialmente, das técnicas apresentadas no capítulo 4. Para utilizá-lo descomprima o arquivo LUXO.ZIP. O sistema é composto pelos seguintes arquivos:

- LUXO.C : A partir de uma trajetória pré-estabelecida, este programa gera um conjunto de "quadros" para o movimento desejado para uma luminária tipo LUXO. Os quadros gerados são, por sua vez, armazenados no arquivo LUXO.OUT.
- LUXO.EXE : Arquivo executável relativo ao programa LUXO.
- LUXO.OUT : Neste arquivo temos armazenada uma simulação realizada pelo programa LUXO. Este arquivo pode ser utilizado como entrada para o programa VIS_SIM, que permite a visualização da animação gerada.
- JUMP.OUT, WALKING.OUT e SHY.OUT : Arquivos de dados que contém simulações realizadas com a luminária LUXO. Tais simulações também podem ser visualizadas através do programa VIS_SIM.
- VIS_SIM.C : Programa para a visualização da simulação gerada através do programa LUXO. Possui como entrada algum dos arquivos .OUT deste sistema.
- VIS_SIM.EXE : Arquivo executável do programa VIS_SIM.
- EGAVGA.BGI : "Driver" gráfico.

3 - SIMULAÇÃO DINÂMICA PARA OBJETOS ARTICULADOS

Este sistema vem a complementar o sistema "Inv-Kin", onde os objetos articulados são tratados apenas através de um modelo cinemático. No sistema "Dinâmica" encontra-se desenvolvido um modelo dinâmico para estruturas articuladas onde, dado um conjunto de forças e torques que atuam no objeto, é calculado o movimento descrito pela estrutura. O modelo proposto para esta simulação fundamenta-se nas técnicas desenvolvidas no capítulo 6. Para a utilização deste sistema descomprima o arquivo DINAMICA.ZIP. Os arquivos deste sistema são:

- DINAMICA.C : Realiza a simulação dinâmica de uma estrutura articulada. O modelo de um objeto é descrito através de uma forma matricial contida no programa fonte. O arquivo também contém a descrição das forças e torques que atuam no objeto. Finalmente, uma vez que o modelo tenha sido desenvolvido, este é simulado iterativamente pelo programa. A simulação resultante é armazenada no arquivo DYNAMIC.OUT.
- DINAMICA.EXE : Arquivo executável relativo ao programa DINAMICA.
- DYNAMIC.OUT : Contém uma simulação obtida através do programa DINAMICA. Pode ser utilizado como entrada para o programa VIS_SIM.
- SIM1.OUT, SIM2.OUT e SIM3.OUT : Arquivos que também contém simulações obtidas através do programa DINAMICA.
- VIS_SIM.C : Realiza a visualização de uma simulação obtida através do programa DINAMICA.
- VIS_SIM.EXE : Arquivo executável relativo ao programa VIS_SIM.
- EGAVGA.BGI : "Driver" gráfico.

4 - SIMULAÇÃO DOS MOVIMENTOS DOS DEDOS DE UMA MÃO

Este sistema descreve um modelo simplificado que simula o movimento dos dedos de uma mão durante a realização de alguns gestos. Trata-se de um modelo misto cinemático-dinâmico, onde as restrições são as posições e orientações iniciais e finais dos dedos da mão. O modelo simulado foi desenvolvido com base nas técnicas apresentadas nos capítulos 4, 5 e 6. Para a utilização do sistema descomprima o arquivo HAND.ZIP. Os arquivos que compõem o sistema são:

- HAND.C : Programa que simula os movimentos dos dedos de uma mão. Uma vez que sejam especificadas as posições e orientações iniciais e finais dos dedos da mão no arquivo INPUT.H o programa HAND interpola estes estados inicial e final gerando, iterativamente, uma trajetória "dinamicamente correta" para os dedos. A seguir, os resultados gerados são armazenados nos arquivos MINIMO.OUT, ANULAR.OUT, MEDIO.OUT, INDICADOR.OUT e POLEGAR.OUT; que servem como entrada para o programa VIS_SIM, que apresenta a simulação realizada.
- HAND.EXE : Arquivo executável relativo ao programa HAND.
- INPUT.H : Arquivo de entrada para o programa HAND. Contém as configurações iniciais e finais para os dedos da mão, portanto, especificando o gesto desejado a ser realizado.
- MINIMO.OUT, ANULAR.OUT, MEDIO.OUT, INDICADOR.OUT e POLEGAR.OUT : Arquivos de saída do programa HAND. Contém a simulação resultante para os gestos configurados no arquivo INPUT.H. Serve como entrada para o programa de visualização VIS_SIM.
- OKAY.ZIP e H_LOOSE.ZIP : Também contém simulações geradas através do programa HAND.
- VIS_SIM.C : Programa para a visualização da simulação gerada pelo programa HAND. Necessita dos arquivos .OUT como entrada.
- VIS_SIM.EXE : Arquivo executável relativo ao programa VIS_SIM.

- EGAVGA.BGI : "Driver" gráfico.

5 - PLANEJAMENTO DE TRILHAS / DESVIO DE OBSTÁCULOS

Neste exemplo utilizamos as técnicas desenvolvidas no capítulo 8. Para a utilização do sistema de planejamento de trilhas e desvio de obstáculos, descomprima o arquivo TRILHA.ZIP. Os arquivos contidos neste sistema são:

- TRILHA.C : Realiza o planejamento de trilhas e prevenção de colisões através da definição de campos de potenciais. Na primeira parte deste programa é definido o "ambiente" onde se passa a animação. A seguir, o programa contém os algoritmos de cálculo dos campos de potencial e otimização de trilhas. As saídas geradas por este programa são os arquivos OBSTAC.OUT e TRILHA.OUT.

- TRILHA.EXE : Arquivo executável do sistema de planejamento de trilhas. Quando executado, inicialmente calcula os campos potenciais associados ao ambiente que foi definido em seu programa fonte. Posteriormente, aguarda que o usuário defina os pontos inicial e final da trilha para que a mesma possa ser otimizada.

- OBSTAC.OUT : Contém a definição dos "obstáculos" presentes no "ambiente" da animação. Este arquivo é utilizado como entrada pelo programa VISUAL.

- TRILHA.OUT : Contém a trilha final definida pelo programa TRILHA.EXE. Este arquivo é utilizado como entrada pelo programa VISUAL.

- VISUAL.C : Apresenta, no monitor do computador, a simulação do movimento de um objeto qualquer através da trilha gerada no programa TRILHA.

- VISUAL.EXE : Arquivo executável do programa visual.

- EGAVGA.BGI : "Driver" gráfico.

6 - SIMULAÇÃO DE UM ESQUELETO HUMANO: O PACOTE "WALK"

Este sistema foi desenvolvido com base nas técnicas apresentadas nos capítulos 4 e 9. Para utilizar o sistema "Walk", descomprima o arquivo WALK.ZIP. Os arquivos contidos neste sistema são:

- WALK.C : Simula o movimento de um ator articulado com forma humana. Os resultados desta simulação são guardados no arquivo BODY.OUT.

- WALK.EXE : Arquivo executável resultado da compilação do arquivo WALK.C.

- BODY.H : Arquivo de entrada do programa WALK.C. Contém os dados de inicialização do ator articulado (posições inicial e final de braços, pernas e tronco).

- TIMER.H : Arquivo de entrada do programa WALK.C. Contém os dados de temporização da animação, tais como *frame-rate*, tempo total da animação, etc.

- DISPLAY.H : Arquivo de entrada do programa WALK.C. Contém informações sobre a configuração do monitor de vídeo do computador.

- BODY.OUT : Arquivo numérico gerado pelo programa WALK.C que contém informações sobre o estado do ator (posição e orientação de seus membros).

- BODY1.OUT, BODY2.OUT, BODY3.OUT, BODY4.OUT : Arquivos que também contém simulações de movimento do ator bípede. Todas estas simulações podem ser visualizadas através do programa WALKPLAY.

- WALKPLAY.C : Apresenta, no monitor de um microcomputador PC, as simulações geradas pelo programa WALK.

- WALKPLAY.EXE : Arquivo executável do programa WALKPLAY.C.

- EGAVGA.BGI : "Driver" gráfico.