

Uma Representação Flexível de Controladores para Animação Fisicamente Realista de Personagens Virtuais

por

Rubens Fernandes Nunes

Dissertação apresentada ao
Mestrado em Ciência da Computação
Universidade Federal do Ceará

Fortaleza, Ceará, Brasil.

Orientador: Creto Augusto Vidal, PhD

Abril 2006
© Rubens Fernandes Nunes, 2006



Universidade Federal do Ceará

Mestrado em Ciência da Computação
Departamento de Computação

UFC

Dissertação de Mestrado

**Uma Representação Flexível de Controladores para
Animação Fisicamente Realista de Personagens Virtuais**

Rubens Fernandes Nunes

Fortaleza – CE, 2006

Rubens Fernandes Nunes

**Uma Representação Flexível de Controladores para
Animação Fisicamente Realista de Personagens Virtuais**

Dissertação apresentada ao Mestrado de Ciências da Computação da Universidade Federal do Ceará (UFC), como requisito parcial para a obtenção do título de Mestre em Ciência da Computação

Orientador:
Creto Augusto Vidal

Fortaleza – CE, 2006

*A Deus e a todas as
pessoas que amo*

Agradecimentos

Agradeço principalmente a Deus, que me deu forças para superar os problemas que apareceram durante o mestrado. Agradeço por conseguir não deixá-los me desanimar nem prejudicar meu desempenho no final das contas. Agradeço por, nos momentos de desânimo, poder contar com muita paciência e força de vontade.

Agradeço ao meu orientador pelo conhecimento que vem me passando desde a minha graduação, pelo constante interesse e confiança demonstrados pelo meu trabalho e esforço, e pelas idéias e contribuições fundamentais para a realização deste trabalho.

Agradeço aos meus pais por todo o apoio, confiança, incentivo, carinho, educação, paz e segurança que me deram em toda minha vida. Agradeço pelo ambiente e situação enormemente favoráveis que me proporcionaram para que eu pudesse me dedicar exclusivamente ao meu trabalho.

Agradeço também às minhas irmãs pelo apoio, confiança, incentivo, carinho e paz que sempre me deram.

Agradeço à minha namorada pela compreensão, confiança e pelo apoio em momentos importantes. Agradeço por sua constante e importante presença, dedicação e pelos muitos carinhos recebidos. Sei o quanto ela torceu para que eu conseguisse.

Agradeço a meus amigos e a todos que torceram por mim, me incentivaram ou contribuíram de alguma forma para que eu conseguisse me tornar mestre!

Agradeço ao Mestrado em Ciência da Computação da Universidade Federal do Ceará pela oportunidade e ao CNPq pelo financiamento. Agradeço também a todos os autores do ODE por seus esforços para disponibilizar um simulador dinâmico de muita qualidade e gratuito.

Obrigado!

Sumário

Neste trabalho, é proposta uma representação de controladores que utiliza sensores de alto nível e tem uma estrutura geral e intuitiva que disponibiliza vários tipos de parâmetros para o animador. Essa estrutura permite, de acordo com a retroalimentação de seus sensores, que várias máquinas de estados atuem simultaneamente no modelo, ou em subconjuntos de seus atuadores. Também permite que, durante seu projeto, procedimentos com instruções gerais, de acordo com o movimento desejado, possam ser definidos e adequadamente associados a ela pelo animador, para serem executados automaticamente pelo controlador durante a simulação dinâmica.

Summary

We propose an improved representation of controllers which uses high-level sensors and possess a general and intuitive structure that offers the animator several types of parameters to manipulate. This structure, with the feedback signals provided by its sensors, allows that several state machines act simultaneously on the model, or in a subset of its actuators. It also permits that procedures with general instructions, in accordance with a specified motion, be defined during its design phase and be adequately associated with it by the animator, in order to be automatically executed by the controller during the dynamic simulation.

Índice

Agradecimentos.....	ii
Sumário	iii
Summary	iv
Índice	v
Lista de Figuras	viii
Lista de Tabelas.....	xi
Capítulo 1	1
Introdução	1
1.1 Motivação	1
1.2 Objetivos e Escopo.....	3
1.3 Estrutura da Dissertação.....	5
Capítulo 2.....	6
Problema de Controle em Sistemas Dinâmicos Atuados.....	6
2.1 Introdução.....	6
2.2 Abordagem Baseada em Trajetória.....	7
2.3 Abordagem Baseada em Controlador	8
2.3.1 Simulador Dinâmico.....	8
2.3.2 Sensores	10
2.3.3 Controlador	10
Controladores PD.....	11
2.4 Considerações Finais.....	12
Capítulo 3.....	14
Animação de Sistemas Dinâmicos Atuados Através de Controladores	14
3.1 Introdução.....	14
Resumo de Trabalhos Relacionados	14
3.2 Representação de Controladores.....	17
3.2.1 Representação Baseada em Estímulo-Resposta.....	17
3.2.2 Redes Sensores-Atuadores.....	19
3.2.3 Máquinas de Estados Finitos	21
Grafos de Controle de Pose	22
3.3 Atribuição de Valores aos Parâmetros dos Controladores	24
3.3.1 Modificação Manual dos Parâmetros	25

3.3.2 Modificação Automática dos Parâmetros.....	26
Critério de Otimização	27
3.4 Interação com o Animador	27
3.5 Composição de Controladores.....	30
3.6 Considerações Finais.....	32
Capítulo 4.....	34
Máquinas de Estados Sensoriais.....	34
4.1 Introdução.....	34
4.2 Estrutura do Controlador	35
4.2.1 Máquinas de Estados Componentes	36
4.2.2 Organização das Máquinas de Estados Componentes	36
4.2.3 Estrutura das Máquinas de Estados.....	38
4.2.4 Estrutura dos Sensores.....	40
4.2.5 Modelagem da Estrutura do Controlador	42
4.3 Tipos de Parâmetros.....	42
4.4 Atribuição de Valores aos Parâmetros do Controlador	43
4.4.1 Modificação Off-Line Manual dos Parâmetros	44
4.4.2 Modificação Off-Line Automática dos Parâmetros	44
4.4.3 Modificação On-Line Interativa dos Parâmetros	45
4.4.4 Modificação On-Line Automática dos Parâmetros.....	46
4.5 Considerações sobre a Utilização da Representação “Máquinas de Estados Sensoriais”	47
4.5.1 Composição de Movimentos.....	47
4.5.2 Ativação de Máquinas de Estados Sensoriais.....	48
4.5.3 Disparando Procedimentos	49
4.6 Considerações Finais.....	51
Capítulo 5.....	53
Implementação	53
5.1 Introdução.....	53
5.2 Simulação Dinâmica	53
5.3 Máquinas de Estados Sensoriais	55
5.3.1 Visão Geral do Processo de Controle.....	55
5.3.2 Executando as Máquinas de Estados.....	56
5.3.3 Médias Ponderadas nos Atuadores.....	56

5.3.4 Interpolação entre Poses	58
5.4 Modificação Automática	59
5.4.1 Modificador Randômico	59
5.4.2 Avaliador	60
5.4.3 Otimizador	61
5.5 Considerações Finais.....	61
Capítulo 6.....	63
Resultados	63
6.1 Introdução.....	63
6.2 Modelos	63
6.3 Descrição dos Testes	66
6.3.1 Humanóide Caminhando em uma Superfície Plana.....	66
6.3.2 Arremesso de uma Bola ao Cesto	74
6.3.3 Salto Mortal para Trás	77
6.4 Considerações Finais.....	81
Capítulo 7.....	84
Conclusões	84
Referências.....	86

Lista de Figuras

Figura 1.1: Estrutura de corpos rígidos articulada.....	1
Figura 2.1: Discretização da trajetória.	7
Figura 2.2: Esquema de interação entre simulador dinâmico, sensores e controlador.	8
Figura 2.3: Iterações do processo de controle.	9
Figura 2.4: Analogia com o ser humano (Santillana, 2006).	11
Figura 2.5: Mola e amortecedor simulados por controladores PD (Laszlo et al., 1997). 11	
Figura 3.1: Representação estímulo-resposta (Ngo & Marks, 1993a).....	18
Figura 3.2: Interior do controlador (representação estímulo-resposta).	19
Figura 3.3: Regiões sensitivas e suas respostas associadas (Ngo & Marks, 1993a).	19
Figura 3.4: Rede sensores-atuadores (Van de Panne & Fiume, 1993).	20
Figura 3.5: Interior do controlador (SAN).	21
Figura 3.6: Corrida e salto sobre cavalo (Hodgins et al., 1995).	22
Figura 3.7: Um PCG utilizado no caminhar humano (Laszlo, 1996).	22
Figura 3.8: Interior do controlador (PCG).	23
Figura 3.9: Topologias da estrutura de PCGs (Laszlo, 1996).	23
Figura 3.10: Processo de interação do controlador com o animador.	28
Figura 3.11: Ação contínua aplicada a uma luminária (Laszlo et al., 2000).	29
Figura 3.12: Ação discreta aplicada aos membros de um gato (Laszlo et al., 2000).	29
Figura 3.13: Paleta usada para os saltos ornamentais (Zhao & van de Panne, 2005). ...	30
Figura 3.14: Exemplo de um PCG gerado interativamente (Zhao, 2004).	30
Figura 3.15: Framework para composição de controladores (Faloutsos et al., 2001).	31
Figura 3.16: Animação através da composição de controladores (Faloutsos, 2002).	32
Figura 4.1: Estrutura da representação “máquinas de estados sensoriais”.	35
Figura 4.2: Conjunto de máquinas de estados atuando simultaneamente em subconjuntos dos atuadores de um modelo.	37
Figura 4.3: Vetor de estados cíclico.	39
Figura 4.4: Possível trecho de código C++ correspondente aos cálculos e testes, realizados por um sensor, responsáveis por determinar seu retorno.	41
Figura 4.5: Tipos de parâmetros vinculados à estrutura da representação “máquinas de estados sensoriais”.	42
Figura 4.6: Manutenção do equilíbrio com modificações on-line automáticas.	47

Figura 4.7: Animação de um humanóide ativando uma máquina de estados sensorial para proteger sua cabeça com os braços durante a queda.	48
Figura 4.8: Controlador disparando um procedimento associado a uma transição.....	51
Figura 5.1: GUI do ODE compilada no Dev-C++ 4.9.8.0 no Windows XP.....	54
Figura 5.2: Visão geral do processo de controle, utilizando a representação “máquinas de estados sensoriais”.....	55
Figura 5.3: Pseudo-algoritmo – Execução das máquinas de estados ativas e habilitadas.	56
Figura 6.1: Modelo de humanóide utilizado para caminhar em uma superfície plana...	63
Figura 6.2: Modelo de humanóide utilizado para arremessar uma bola ao cesto.	64
Figura 6.3: Modelos de humanóide utilizados para realizar um salto mortal para trás..	64
Figura 6.4: Configuração inicial do modelo de humanóide original.....	65
Figura 6.5: Máquina de estados constante responsável pelo movimento do tronco e das pernas de um humanóide ao caminhar.	67
Figura 6.6: Máquina de estados constante responsável pelo movimento dos braços de um humanóide ao caminhar.....	68
Figura 6.7: Comparação do caminhar antes e depois da otimização.....	71
Figura 6.8: Controlador otimizado (mostrado na Figura 6.7), interagindo com pequenas forças externas sendo aplicadas na cabeça do modelo.....	72
Figura 6.9: Ativação de uma máquina de estados sensorial durante a queda, para proteção do corpo com os braços.....	73
Figura 6.10: Máquina de estados constante responsável pelo movimento dos braços de um humanóide ao arremessar uma bola ao cesto.....	74
Figura 6.11: Comparação do arremesso de uma bola ao cesto antes e depois da otimização.....	76
Figura 6.12: Trajetória percorrida pela bola depois de arremessada ao cesto, utilizando o controlador otimizado.....	76
Figura 6.13: Máquina de estados constante responsável por fazer um humanóide impulsionar um outro, para que este realize um salto mortal.....	78
Figura 6.14: Máquina de estados constante responsável pelo movimento do tronco e das pernas de um humanóide ao executar um salto mortal, impulsionado por um outro.	79
Figura 6.15: Máquina de estados constante responsável pelo movimento dos braços de um humanóide ao executar um salto mortal, impulsionado por um outro.....	80

Figura 6.16: Quadros da animação de um humanóide executando um salto mortal para trás, impulsionado por um outro.	81
Figura 6.17: Trajetória realizada por um humanóide executando um salto mortal para trás, impulsionado por um outro.	81

Lista de Tabelas

Tabela 6.1: Propriedades dos corpos rígidos do modelo de humanóide original.....	65
Tabela 6.2: Propriedades das juntas do modelo de humanóide original.....	65
Tabela 6.3: Modificações automáticas nos parâmetros de um controlador utilizado para fazer um modelo de humanóide caminhar em uma superfície plana.	69

Capítulo 1

Introdução

1.1 Motivação

A animação por computador, uma das subáreas da computação gráfica, está voltada para a geração de seqüências de imagens construídas com o auxílio do computador a partir de modelos matemáticos dos objetos componentes de um cenário de forma que o observador perceba a dinâmica desses elementos de cenário. A animação de personagens virtuais vem ganhando um interesse especial devido às aplicações na indústria cinematográfica, na indústria de jogos de computador e na realidade virtual. Em um ambiente virtual, por exemplo, os personagens virtuais movimentam-se para explorar o ambiente e para interagir tanto com os objetos nele inseridos, quanto com outros personagens virtuais que compartilham esse ambiente. Em algumas situações, a forma realista com que esses personagens se movem é de fundamental importância para a boa aceitação da aplicação.

Os personagens virtuais geralmente são modelados como estruturas articuladas, isto é, estruturas constituídas de corpos rígidos simples conectados por juntas. A Figura 1.1 mostra um exemplo de uma estrutura articulada, representando um humanóide. A aparência do personagem pode ser trabalhada através da sobreposição de malhas e texturas à estrutura articulada. Entretanto, a aparência do personagem não está no escopo deste trabalho.

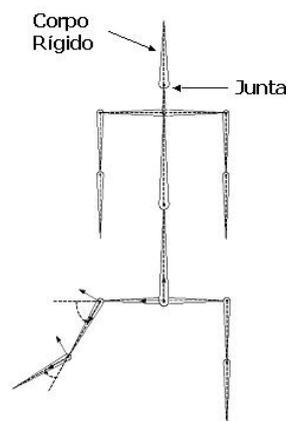


Figura 1.1: Estrutura de corpos rígidos articulada.

A utilização de física para gerar movimentos mais realísticos tem sido uma das direções de pesquisa em animação por computador e vem mostrando avanços significativos há aproximadamente vinte anos (Armstrong & Green, 1985; Wilhelms,

1986; Wilhelms & Barsky, 1985). Duas abordagens principais têm sido consideradas: a abordagem cinemática e a abordagem dinâmica.

A abordagem cinemática baseia-se apenas em informações de posição e velocidade (não há preocupação com as forças e os torques responsáveis por gerar o movimento). Duas técnicas cinemáticas principais são: *keyframing* e captura de movimentos (Zhao, 2004). A técnica de *keyframing* consiste em definir, em determinados instantes, as posições de todas as partes do modelo. Essas configurações-chaves são chamadas de *keyframes* (quadros-chaves). Os quadros intermediários são obtidos através da interpolação entre dois quadros-chaves consecutivos. A qualidade e o realismo da animação dependem muito das habilidades artísticas do animador, devido ao alto controle que essa técnica oferece. A técnica de captura de movimentos (Multon et al., 1999) consiste em obter o movimento a partir de dispositivos conectados a atores reais e mapeá-lo ao modelo virtual. Comparada com a técnica de *keyframing*, o realismo é obtido com um menor esforço por parte do animador, mas o controle oferecido é menor. Além da enorme quantidade de dados a ser tratada, a principal desvantagem dessa técnica é a difícil adaptação do movimento capturado a novos ambientes e situações. Outras técnicas cinemáticas são discutidas em (Costa-Filho, 2004).

Na abordagem dinâmica (fisicamente realista), as forças e torques que geram o movimento são os elementos principais a serem considerados. Essa abordagem considera a interação física com o ambiente e as propriedades físicas do modelo, tais como massas e momentos de inércia dos corpos. Todas as forças e torques (externos ou internos) que agem sobre o modelo influenciam diretamente na animação. A grande vantagem do uso da dinâmica é a garantia de que o sistema obedece às leis da física. O realismo físico é obtido de uma maneira mais automática do que através de técnicas cinemáticas, facilitando o trabalho do animador em determinadas seqüências. Entretanto, a geração do movimento tem um custo computacional elevado quando comparado ao custo das técnicas cinemáticas. No entanto, graças aos recentes avanços das técnicas numéricas empregadas nos simuladores físicos e aos avanços da tecnologia de hardware, já é possível realizar simulações dinâmicas em tempo real (Smith, 2000).

O problema de como controlar os movimentos enquanto as leis da física são respeitadas (problema de controle) continua exigindo resultados mais expressivos para que a dinâmica ganhe ainda mais espaço nos softwares de animação. Esse problema é característico de sistemas atuados, que são sistemas dinâmicos capazes de executar tarefas, criando e regulando seus próprios movimentos através de forças e torques

internos gerados por seus atuadores, sem depender exclusivamente de forças externas (Raibert & Hodgins, 1991). Esse problema, no contexto de animação fisicamente realista¹ de personagens virtuais, consiste em determinar as forças e os torques adequados a serem gerados em seus atuadores a fim de conseguir um movimento desejado. Neste trabalho, os personagens virtuais são modelados como sistemas atuados cujos atuadores simulam o funcionamento dos músculos e estão localizados nas articulações.

Uma abordagem possível para determinar as forças e os torques desenvolvidos pelos atuadores do modelo para executar o movimento desejado utiliza elementos controladores. Esses controladores são projetados para tipos específicos de movimentos e guiam o modelo durante a simulação dinâmica, enviando sinais aos atuadores para que eles modifiquem as forças e os torques por eles produzidos. A representação de um controlador define como essas forças e torques são determinados, e é de fundamental importância em seu projeto.

O projeto de um controlador é suficientemente complexo para ser desenvolvido manualmente e não menos complexo e caro computacionalmente quando desenvolvido automaticamente. O projeto automático de controladores visa tornar esse processo mais fácil e acessível aos animadores. Entretanto, controladores para modelos mais complexos e instáveis, utilizando as técnicas disponíveis atualmente, ainda não podem ser completamente projetados de maneira automática. Geralmente, dependendo da complexidade do modelo e do movimento desejado, apenas alguns dos parâmetros do controlador são escolhidos para serem gerados automaticamente. Assim, tanto o custo computacional quanto a eficácia do projeto automático de controladores depende muito de como o controlador é representado e desses parâmetros escolhidos, os quais geram o espaço de busca.

1.2 Objetivos e Escopo

Esta dissertação aborda o problema de controle da animação fisicamente realista de personagens virtuais. As técnicas disponíveis atualmente para a solução desse problema ainda não permitem que movimentos fisicamente realistas desejados sejam gerados de uma maneira suficientemente fácil, eficaz e rápida.

¹ Animações de personagens virtuais podem ser ditas fisicamente realistas levando-se em consideração que o realismo físico está limitado ao modelo dinâmico estudado, correspondendo apenas a um modelo simplificado do mundo real.

Mais especificamente, considerando a abordagem que utiliza controladores, este trabalho tem como objetivo propor uma representação flexível de controladores para animação fisicamente realista de personagens virtuais. A representação proposta utiliza sensores, para fornecerem feedback e ajudarem a manter o controle mais estável, e tem uma estrutura mais geral e intuitiva que disponibiliza vários tipos de parâmetros ao animador. Essa estrutura permite que, de acordo com a retroalimentação proveniente de seus sensores, várias máquinas de estados possam atuar simultaneamente no mesmo modelo, ou em subconjuntos de seus atuadores. Ela também permite que o animador, ao projetá-la, defina procedimentos com instruções gerais (de acordo com o movimento desejado) que a ela são associados adequadamente, para serem executados automaticamente pelo controlador durante a simulação dinâmica.

Em síntese, a representação de controladores proposta visa facilitar:

- ▶ O projeto dos controladores, disponibilizando uma estrutura mais intuitiva;
- ▶ A composição de movimentos, capacitando os controladores a produzir movimentos mais gerais e complexos; e
- ▶ O projeto automático dos controladores.

Especificamente em relação ao projeto automático, a representação proposta:

- ▶ Facilita a escolha dos parâmetros que devem ser modificados automaticamente;
- ▶ Fornece diversas variações de modificação automática; e
- ▶ Permite que a otimização possa ser realizada em vários estágios, onde possivelmente diferentes conjuntos de parâmetros são modificados.

Além das inovações propostas na representação de controladores, neste trabalho também é proposta uma estrutura mais geral e elaborada para os sensores utilizados, composta por parâmetros mais expressivos, e mais adequada à geração automática dos controladores, pois permite que o animador possa pensar no funcionamento do sensor em um nível mais alto de abstração.

O escopo deste trabalho é a animação fisicamente realista de estruturas de corpos rígidos articulados, ou, mais especificamente, o problema de controle dos atuadores associado à animação fisicamente realista de tais estruturas, vistas como sistemas dinâmicos atuados. Assim, os aspectos de renderização, a fim de melhorar a aparência dos personagens virtuais, e as considerações sobre deformações nos corpos não é o foco desta dissertação. A simulação dinâmica também não faz parte do escopo

deste trabalho. Conseqüentemente, para a realização dos testes, um simulador dinâmico já pronto foi utilizado (Smith, 2000).

1.3 Estrutura da Dissertação

Para a apresentação detalhada dos aspectos abordados nesta Introdução, o restante desta dissertação está estruturado como se segue. No Capítulo 2, é apresentada uma introdução ao problema de controle em sistemas dinâmicos atuados. No Capítulo 3, é discutida a utilização de controladores na animação de sistemas dinâmicos atuados. No Capítulo 4, é explicada a representação de controladores proposta. No Capítulo 5, são descritos os aspectos de implementação da representação proposta. No Capítulo 6, são apresentados os resultados dos testes realizados. E, finalmente, no Capítulo 7, são apresentadas as conclusões sobre o trabalho.

Capítulo 2

Problema de Controle em Sistemas Dinâmicos Atuados

2.1 Introdução

Sistemas dinâmicos podem ser divididos em duas categorias: sistemas não-atuados e sistemas atuados. Os sistemas dinâmicos não-atuados dependem unicamente de agentes externos para se movimentarem. As forças e os torques que determinam seu movimento são previamente especificados, e a animação resultante do sistema é gerada. Um exemplo simples de sistema não-atuado é o lançamento de um projétil sujeito apenas ao campo gravitacional. Sua simulação dinâmica é o bastante para gerar um movimento realista e natural. Sistemas não-atuados são também muito utilizados em movimentos secundários, que resultam diretamente da animação principal, como por exemplo, para animar modelos de roupas e cabelos, adicionando realismo à animação (Carignan et al., 1992; Magnenat-Thalmann, 1994).

Os sistemas dinâmicos atuados se diferenciam dos não-atuados principalmente por possuírem mecanismos de geração de forças e torques internos (atuadores) e por serem capazes de executar tarefas, por meio do controle de seus movimentos que pode ser obtido através da especificação adequada das forças e torques gerados pelo conjunto de seus atuadores, sem depender exclusivamente de forças externas (Raibert & Hodgins, 1991). Os atuadores são dispostos nas juntas do modelo articulado e funcionam como músculos, que convertem energia armazenada em forças e torques, que agem na estrutura mecânica do sistema.

A animação de sistemas dinâmicos atuados requer a solução de um problema de controle que, no contexto de animação fisicamente realista de personagens virtuais, consiste em determinar as forças e os torques adequados a serem gerados por seus atuadores a fim de conseguir um movimento desejado, o qual, ao contrário de sistemas não-atuados, deve ser previamente especificado. Especificar o movimento desejado não implica em fornecer exatamente as trajetórias de todas as juntas do modelo durante o intervalo de tempo da animação. Por exemplo, na animação de um humanóide, o movimento desejado pode ser o de caminhar. Assim, o problema de controle é determinar os torques adequados a serem gerados por todos os atuadores para conseguir um caminhar natural. Apenas a obediência às leis da física não garante o tipo de movimento especificado nem a naturalidade do movimento. Se nenhuma força fosse

gerada pelos atuadores do humanóide, ele provavelmente desabaria no chão e não seria capaz de dar nenhum passo, mesmo obedecendo à física.

Existem duas abordagens principais para tratar o problema de controle de sistemas dinâmicos atuados: a abordagem baseada em trajetória e a abordagem baseada em controlador.

2.2 Abordagem Baseada em Trajetória

Na abordagem baseada em trajetória, o movimento é analisado de uma forma global, onde todo o intervalo de tempo da animação estudado é discretizado em intervalos menores, geralmente do mesmo tamanho (Figura 2.1).

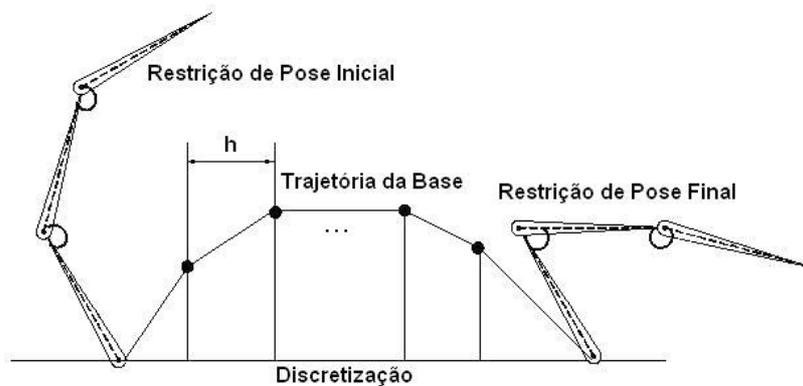


Figura 2.1: Discretização da trajetória.

O problema de controle é mapeado num problema de otimização. A trajetória completa descrita pelo modelo e as forças e os torques aplicados pelos atuadores em cada instante constituem o espaço de busca desse problema, o que significa que a trajetória escolhida é aquela que minimiza (ou maximiza) uma dada função objetivo, como por exemplo, o esforço gasto pelo modelo. As leis da física fazem parte das restrições impostas sobre o problema de otimização. Essas restrições devem ser consideradas entre todos os intervalos consecutivos da discretização (Witkin & Kass, 1988).

Essa abordagem permite um maior controle do movimento resultante, através da especificação de restrições explícitas da trajetória gerada, tais como restrições das configurações inicial e final, mostradas na Figura 2.1. Entretanto, o uso dessa abordagem impõe algumas limitações importantes ao processo de animação além das já impostas pelo problema de otimização. Por exemplo, como a própria trajetória faz parte do resultado obtido da otimização, um novo problema de otimização deve ser solucionado para obter uma nova trajetória a cada nova instância de movimento

desejado, implicando num enorme custo computacional. Além disso, as interações com o ambiente, tais como colisões, são difíceis de tratar.

2.3 Abordagem Baseada em Controlador

Na abordagem baseada em controlador, simulações dinâmicas do sistema atuado são realizadas incrementalmente. Geralmente, um incremento de tempo fixo é determinado e, a partir da configuração inicial no início do intervalo considerado e das forças e torques atuantes no sistema, uma simulação dinâmica é executada para obtenção da configuração no final do intervalo. Um controlador, projetado para um tipo específico de movimento, é utilizado para guiar o sistema, determinando as forças e os torques adequados a serem gerados por seus atuadores durante a simulação dinâmica, a fim de obter o movimento desejado. A cada passo, o controlador pode utilizar informações sensoriais como feedback para auxiliar no cálculo dos valores das forças e dos torques fornecidos ao simulador dinâmico, possibilitando um movimento resultante mais estável. A Figura 2.2 mostra o esquema de como o simulador dinâmico, os sensores (considerando que há feedback) e o controlador interagem.

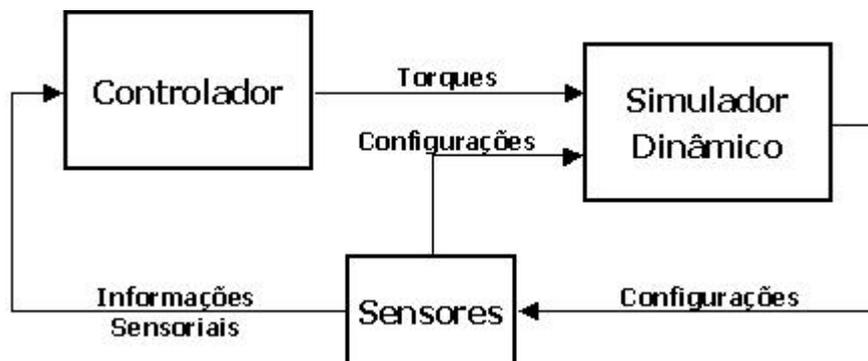


Figura 2.2: Esquema de interação entre simulador dinâmico, sensores e controlador.

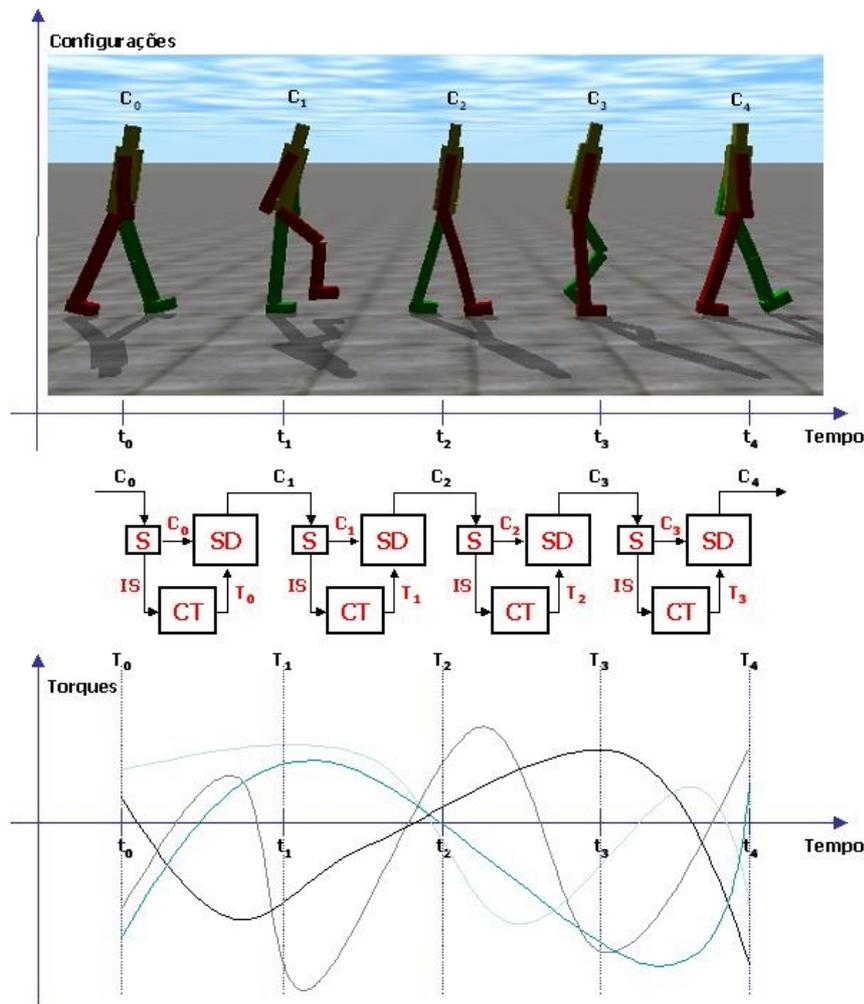
Um outro esquema, ilustrando o processo de controle iterativamente, é mostrado na Figura 2.3. As configurações mostradas são apenas ilustrativas e correspondem ao movimento obtido ao se tentar realizar um movimento desejado de caminhada.

Nesta seção, o simulador dinâmico, os sensores e o controlador são discutidos individualmente.

2.3.1 Simulador Dinâmico

O simulador dinâmico é composto por um compilador dinâmico e por um integrador numérico. Ele é responsável pelo cálculo das equações de movimento do sistema e pela integração dessas equações. O compilador dinâmico recebe as especificações do modelo

a ser animado e do ambiente de animação, determina as equações de movimento do sistema e repassa essas equações ao integrador numérico. O integrador numérico, de posse das equações de movimento, da configuração corrente do sistema (no início do intervalo de tempo especificado) e das forças atuantes, integra as equações de movimento para obter uma nova configuração do sistema no final do intervalo de tempo.



Legenda: S – Sensores; SD – Simulador Dinâmico; CT – Controlador; IS – Informações Sensoriais; T_i – Conjunto de valores de torques a serem gerados pelos atuadores no instante t_i.

Figura 2.3: Iterações do processo de controle.

Os valores das forças internas e externas aplicadas ao modelo fazem parte das equações de movimento e podem ser alterados, a cada iteração da simulação dinâmica, de acordo com as modificações que ocorrem no modelo ou no ambiente. As forças internas aplicadas ao modelo são provenientes das forças e dos torques gerados pelos atuadores. A cada iteração, o controlador fornece os valores das forças e dos torques atuais a serem gerados pelos atuadores, como ilustrado na Figura 2.3, e esses valores são atualizados nas equações de movimento do sistema.

2.3.2 Sensores

Os sensores são elementos dispostos em posições específicas da estrutura articulada com a finalidade de: perscrutar o ambiente externo para detectar condições para as quais o sistema dinâmico deverá reagir, e examinar variáveis internas do sistema dinâmico que disparem ações corretivas. Esse *feedback* fornecido pelos sensores distingue os chamados controladores de laço aberto (aqueles que não recebem *feedback* de sensores) dos controladores de laço fechado (aqueles que recebem *feedback* de sensores). O uso de *feedback*, ou seja, o repasse de informações sobre o modelo ou sobre as condições do ambiente, serve para corrigir ou ajustar o movimento do modelo, e fornecer um controle mais estável desse movimento. Numa simulação dinâmica com controladores e atuadores, os sensores examinam o estado de variáveis de configuração do sistema ao final de um passo de simulação e, após um processamento adequado, enviam as informações necessárias para o controlador e a configuração corrente de volta ao simulador dinâmico.

2.3.3 Controlador

No contexto de animação dinâmica de sistemas atuados, controladores podem ser definidos como entidades responsáveis pelo controle dos atuadores desses sistemas. Os controladores enviam sinais aos atuadores com os valores de forças e torques que devem ser gerados para produzir um movimento desejado. A Figura 2.4 faz uma analogia com o ser humano. Os sensores correspondem às regiões sensitivas localizadas pelo corpo, os atuadores correspondem aos músculos estriados esqueléticos e os controladores correspondem às regiões do cérebro responsáveis pelas ações musculares voluntárias (Webciência, 2006). Similarmente, essas regiões do cérebro enviam sinais aos músculos com as intensidades das contrações que devem ser realizadas, resultando em torques aplicados na estrutura esquelética que produzem um movimento desejado.

A simulação dinâmica resolve as equações da física que governam o movimento do sistema para determinadas condições iniciais e de contorno. No entanto, ela não tem responsabilidade sobre o controle do movimento. O simulador dinâmico sozinho, por exemplo, não faz um humanóide caminhar, pois não é ele quem define os torques adequados a serem aplicados pelos atuadores. O controlador é o responsável por gerar, para cada atuador, os torques, como função do tempo, que são enviados ao simulador dinâmico (vide Figura 2.3 – nessa figura as funções de torques são apenas ilustrativas e apenas quatro atuadores são contemplados). Entretanto, como na simulação dinâmica o

tempo é discretizado, uma função de torques gerada para um determinado atuador possui valores de torques apenas nos instantes da discretização.

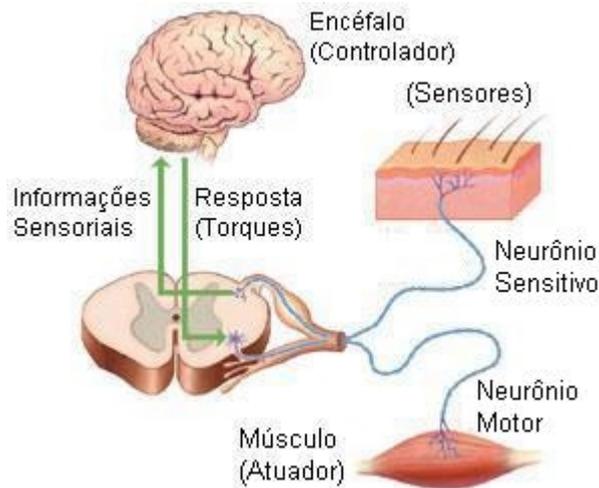
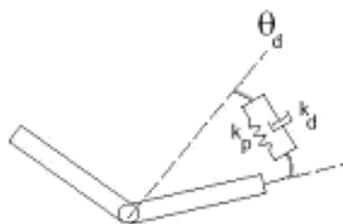


Figura 2.4: Analogia com o ser humano (Santillana, 2006).

Os controladores são projetados para computarem os torques adequados a serem gerados pelos atuadores a eles associados, a fim de que um movimento desejado seja obtido. Neste trabalho, adotam-se controladores do tipo *Proportional-Derivative* (PD).

Controladores PD

Em simulações, os atuadores nem sempre são modelados para funcionar como os músculos do ser humano, que ao se contraírem aplicam forças na estrutura óssea. Os atuadores podem, por exemplo, funcionar como motores que aplicam torques nas juntas do modelo. Controladores do tipo PD são geralmente utilizados para calcular os torques a serem gerados pelos atuadores para levar as juntas a assumirem ângulos desejados. Em um determinado instante, um conjunto de ângulos desejados de todas as juntas do modelo corresponde a uma configuração desejada.



θ_d – ângulo desejado a ser assumido pela junta.

Figura 2.5: Mola e amortecedor simulados por controladores PD (Laszlo et al., 1997).

Os controladores do tipo PD simulam uma mola e um amortecedor angulares situados entre o ângulo atual e o ângulo desejado da junta (Figura 2.5). A mola é definida em relação ao ângulo desejado e com um ângulo de relaxamento nulo, de tal

forma que o torque gerado pela mola é nulo se a junta já se encontra com o ângulo desejado. O torque a ser gerado pelo atuador é calculado a partir dos torques gerados pela mola e pelo amortecedor através da expressão

$$T = k_p (\theta_d - \theta) - k_d \dot{\theta}, \quad (2.1)$$

onde T é o torque a ser gerado pelo atuador, θ_d é o ângulo desejado da junta, θ é o ângulo atual da junta, $\dot{\theta}$ é a velocidade angular relativa entre os dois corpos conectados pela junta, k_p é a constante da mola e k_d é a constante do amortecedor.

Existem trabalhos que consideram k_p e k_d como parâmetros do controlador (Van de Panne & Fiume, 1993), mas neste trabalho essas constantes são mantidas fixas e fazem parte da especificação da modelagem do atuador. Os valores de θ e $\dot{\theta}$ são determinados pela configuração atual do modelo. Portanto, ao invés do valor do torque, apenas o valor do ângulo desejado da junta deve ser especificado a cada iteração.

Para determinar os torques, controladores do tipo PD executam cálculos simples que são suficientes para os propósitos de controle da animação de estruturas articuladas simuladas (Laszlo, 1996). Comparados com controladores do tipo PD, controladores do tipo PID executam mais cálculos, sendo capazes de calcular torques mais precisos, e são mais apropriadamente utilizados no controle da animação de robôs, onde atritos da estrutura mecânica nas juntas não podem ser desprezados (Bekey, 2005).

O controlador pode ter diferentes níveis de abstração de como gerar as funções de torques que são enviadas ao simulador dinâmico. Quando controladores do tipo PD são utilizados, correspondendo ao nível mais baixo de abstração do controlador, o problema de controle se reduz (tornando-se mais intuitivo) a obter funções de ângulos desejados adequados para todos os atuadores a fim de obter o movimento desejado.

2.4 Considerações Finais

Neste capítulo, foram apresentadas duas abordagens para solução do problema de controle de sistemas dinâmicos atuados: a abordagem baseada em trajetória e a abordagem baseada em controladores. A abordagem baseada em controladores possui várias vantagens sobre a abordagem baseada em trajetória. Na abordagem baseada em controlador, como o resultado a ser obtido é um controlador adequadamente projetado, que define o movimento exclusivamente em nível de forças e torques, não há imposição de restrições explícitas quanto à trajetória gerada. Esse menor controle direto sobre a trajetória resultante é compensado pela maior liberdade em nível de posição que resulta:

- ▶ Na possibilidade de reutilização de controladores em configurações ligeiramente diferentes do modelo ou do próprio ambiente (Van de Panne et al., 1990). Por exemplo, um controlador pode funcionar mesmo que a configuração inicial do modelo seja alterada ou que forças externas sejam aplicadas sobre o modelo;
- ▶ Na obtenção de animações não repetitivas, ao contrário do que acontece em animações obtidas por cinemática. Utilizando cinemática em um caminhar, por exemplo, um ciclo de passadas é gerado e o restante da animação é obtido repetindo-se esse ciclo de posições de forma que o ciclo no início da caminhada será idêntico a todos os ciclos posteriores; e
- ▶ Na possibilidade de que controladores, além de serem utilizados em ferramentas de animação, sejam também mais facilmente aplicáveis na área da robótica (MIT Leg).

Controladores também podem ser concatenados para gerar uma seqüência de movimentos mais complexa, a fim de produzir personagens autônomos com um grande repertório de habilidades específicas (Faloutsos et al., 2001). Composição é explicada com mais detalhes na Seção 3.5. Além dessas vantagens, dentre as técnicas dinâmicas, o uso de controladores apresenta um baixo custo computacional, considerando que os controladores já tenham sido projetados. Esse custo é determinado principalmente pelo custo da própria simulação dinâmica, a qual pode ser realizada em tempo real (Smith, 2000).

O principal problema do uso de controladores é a dificuldade de se projetar controladores adequadamente. Além disso, o projeto dos controladores geralmente exige um processo de busca demorado, e é elaborado com vistas a um movimento específico. Entretanto, a possibilidade de composição, ao utilizar mais de um controlador, pode compensar essa limitação.

Capítulo 3

Animação de Sistemas Dinâmicos Atuados Através de Controladores

3.1 Introdução

Na Seção 2.3, controladores são apresentados como caixas-pretas responsáveis por determinar as forças e os torques adequados a serem gerados pelos atuadores durante a simulação dinâmica. Neste capítulo, controladores e a forma como funcionam são discutidos em maiores detalhes, analisando-se o seu interior.

Controladores são representados através de uma estrutura, a base da representação, e de um conjunto de parâmetros a ela associados, que é a parte modificável da representação. A representação do controlador define como determinar as forças e os torques, e é de fundamental importância em seu projeto. As figuras 3.2, 3.5 e 3.8 ilustram o interior da caixa controlador, mostrada na Figura 2.2, considerando diferentes tipos de representação.

Para gerar um controlador, deve-se, primeiramente, determinar qual tipo de representação utilizar. Em seguida, deve-se, de acordo com o movimento desejado, modelar sua estrutura, a qual permanece imutável depois de modelada. Finalmente, deve-se, também de acordo com o movimento desejado, atribuir valores ao conjunto de parâmetros, que é definido pelo tipo de representação utilizado. Em relação à atribuição de valores ao conjunto de parâmetros, controladores podem ser projetados tanto manualmente, exigindo que o animador conheça bem os detalhes do comportamento a ser gerado, quanto automaticamente.

Resumo de Trabalhos Relacionados

As primeiras versões de controladores usavam funções explícitas de forças e torques especificadas pelo animador, baseadas em dados medidos ou observados (Laszlo, 1996). O animador atuava no nível mais baixo do projeto do controlador. Utilizando máquinas de estados projetadas manualmente, Raibert e Hodgins (1991) animam robôs bípedes e quadrúpedes e um modelo de canguru correndo e pulando e Hodgins e seus co-autores (1995) geram animações de corrida, ciclismo e salto sobre cavalo para modelos humanóides.

Alguns autores propõem representações de controladores as quais permitem que eles possam também ser projetados automaticamente utilizando otimização estocástica (Auslander et al., 1995; Hodgins & Pollard, 1997; Ngo & Marks, 1993a; Ngo & Marks, 1993b; Sims, 1994; Van de Panne, 1996; Van de Panne & Fiume, 1993; Van de Panne et al., 1994; Van de Panne & Lamouret, 1995). Ngo e Marks (1993a; 1993b) propõem uma representação baseada em respostas a estímulos. Pares constituídos de uma função de estímulo e de uma resposta são projetados considerando o modelo e o movimento desejado. A cada passo da simulação dinâmica, as funções de estímulo são avaliadas, e a função vencedora ativa a resposta correspondente, influenciando no movimento gerado. Van de Panne e Fiume (1993) representam o controlador conectando os sensores aos atuadores do modelo através de uma rede de conexões com peso. Os atuadores recebem os valores resultantes das operações internas dessa rede, em função dos sensores, e mapeiam em ângulos desejados a serem assumidos pelas juntas. Controladores do tipo PD calculam os torques correspondentes a esses ângulos desejados. Van de Panne e seus co-autores (1994) propõem grafos de controle de pose (Pose Control Graph - PCG), que são máquinas de estados finitos que possuem uma pose associada a cada estado. Os estados das máquinas de estados finitos correspondem a fases do controlador, determinando como ele deve agir. Transições entre estados são responsáveis por realizar a troca do estado ativo. Essas poses consistem de um conjunto de ângulos desejados a serem assumidos pelas juntas, também utilizando controladores do tipo PD.

Sims (1994) projeta automaticamente tanto o controlador quanto o modelo de estruturas articuladas controlado. Van de Panne e Lamouret (1995) utilizam forças externas para auxiliar o processo de otimização. Essas forças externas guiam o processo de otimização, mantendo o equilíbrio do modelo, e facilitam o aprendizado do movimento desejado. A idéia é semelhante a um bebê aprendendo a andar com o apoio dos pais para não cair. Aos poucos, os pais vão deixando de apoiá-lo até que o bebê consiga andar sozinho. Da mesma forma, as forças externas vão sendo eliminadas em estágios posteriores. Van de Panne (1996) sintetiza controladores parametrizados, interpolando linearmente os parâmetros de outros controladores já projetados. Um exemplo de parâmetro de interpolação é a velocidade do movimento gerado pelo controlador. A partir de dois controladores projetados para gerar movimentos com diferentes velocidades, controladores que geram movimentos com velocidades intermediárias podem ser rapidamente sintetizados por interpolação. Hodgins e Pollard

(1997) utilizam controladores já projetados para um determinado personagem e os adaptam automaticamente a novos personagens.

Controladores também podem ser projetados ou modificados durante a simulação dinâmica (on-line), através da interação com o animador ou automaticamente. No primeiro caso, interfaces apropriadas mapeiam informações de dispositivos de entrada, como mouse e teclado, nos parâmetros do controlador. Laszlo e seus co-autores (2000) guiam o movimento de vários modelos 2D interativamente, explorando ações de controle contínuas e discretas. Zhao e van de Panne (2005) apresentam uma interface mais elaborada e amigável, permitindo o controle de um maior número de parâmetros, e comprovam sua eficácia gerando animações em 3D de saltos ornamentais, saltos de esqui e *snowboarding*. A interface consiste de uma paleta usada para gerar poses e definir o tempo de transição entre elas interativamente. O controlador resultante dessa interação permite ajustes fora da simulação dinâmica (off-line), manuais ou automáticos, a fim de obter melhores resultados. Uma outra interface que explora a utilização de um *gamepad* como dispositivo de entrada também é apresentada. Já no segundo caso, modificações são realizadas automaticamente sem a intervenção do animador. Laszlo (1996) analisa o ciclo do movimento de caminhada gerado por um controlador PCG instável de laço aberto, projetado manualmente off-line, e modifica esse PCG automaticamente on-line, de acordo com o feedback obtido, a fim de estabilizar o movimento, mantendo sua periodicidade.

Nos trabalhos citados cujos controladores são projetados automaticamente, utiliza-se um processo de otimização estocástica, exigindo muitas tentativas de simulação e avaliação segundo uma função objetivo. A utilização de métodos de busca, tais como têmpera simulada e algoritmos genéticos, que essencialmente caminham randomicamente através de espaços de busca, possivelmente enormes, é exigida porque essa função objetivo quase nunca pode ser derivada com relação aos sinais enviados pelo controlador. Quando comparados aos métodos guiados por direções de gradiente, métodos de busca estocástica geralmente convergem mais lentamente para uma boa solução. Conseqüentemente, os espaços de busca utilizados devem ter tamanhos razoáveis para que resultados satisfatórios sejam obtidos. Grzeszczuk e seus co-autores (1998) introduzem o NeuroAnimator, uma eficiente alternativa de substituição à abordagem convencional de produzir animações fisicamente realistas através de simulação dinâmica. Na simulação dinâmica convencional, as equações de movimento são integradas numericamente. NeuroAnimators utilizam redes neurais que observam os

modelos físicos durante simulações dinâmicas convencionais e capturam características de seus movimentos, tornando-se capazes de sintetizar esses movimentos com um custo bem menor. Essa abordagem introduz também um método para projetar os controladores automaticamente, que explora essa rápida emulação, avaliando o controlador mais rapidamente em cada tentativa, e a estrutura das redes neurais, que permite utilizar informações de gradiente no processo de otimização, obtendo uma convergência mais rápida. Entretanto, para que os NeuroAnimators sejam capazes de emular os movimentos de um modelo físico, as redes neurais correspondentes têm que ser treinadas. O treinamento é demorado, pois utiliza o algoritmo de *backpropagation* e exige a execução de vários exemplos de simulação dinâmica do modelo físico.

Nas seções seguintes, alguns conceitos necessários são melhor definidos e os trabalhos relacionados são discutidos mais especificamente dentro de seus respectivos temas.

3.2 Representação de Controladores

Dentre as representações possíveis, três representações principais que podem ser projetadas automaticamente se destacam (Pina et al., 2000): a representação baseada em estímulo-resposta, a representação por redes sensores-atuadores, e a representação por máquinas de estados finitos. Para cada uma dessas representações, são discutidos as suas estruturas e os seus tipos de parâmetros, analisando-se o interior dos controladores. A análise desses exemplos de tipos de representação proporciona um melhor entendimento do funcionamento interno dos controladores.

3.2.1 Representação Baseada em Estímulo-Resposta

Essa representação se baseia em respostas que são ativadas devido a estímulos provenientes do modelo ou do ambiente. A partir do modelo e do movimento desejado, a estrutura do controlador é definida por pares constituídos de uma função de estímulo e de uma resposta. Essas funções de estímulo dependem de variáveis sensoriais, cujos valores são obtidos do estado atual do sistema. O esquema da Figura 3.1 ilustra o funcionamento dos controladores que possuem a representação estímulo-resposta.

Em cada passo da simulação dinâmica, as funções de estímulo são calculadas e a que retorna o maior valor positivo, considerada a vencedora, ativa a resposta correspondente. Apenas uma resposta pode estar ativa em cada iteração da simulação física. Uma resposta consiste de uma constante de tempo e de uma configuração

desejada a ser atingida pelo modelo. As respostas são aplicadas de acordo com a equação de movimento criticamente amortecida,

$$\tau^2 \ddot{\theta}_i + 2\tau \dot{\theta}_i + (\theta_i - \theta_i^0) = 0, \quad (3.1)$$

definida para cada atuador, onde τ é a constante de tempo, θ_i^0 é o valor desejado (referente à configuração desejada definida pela resposta correspondente) para o i -ésimo ângulo interno do modelo, e θ_i é o valor atual desse i -ésimo ângulo interno. O objetivo dessas equações de movimento é tentar levar o modelo a atingir a configuração desejada suavemente, mesmo quando a resposta ativa e, conseqüentemente, a configuração desejada são alteradas abruptamente. Portanto, essa representação também poderia ser implementada utilizando controladores do tipo PD, como é feito nas outras duas representações discutidas nas seções 3.2.2 e 3.2.3. A Figura 3.2 ilustra o interior do controlador com o molde de sua estrutura, de acordo com a Figura 2.2, quando a representação estímulo-resposta é a utilizada.

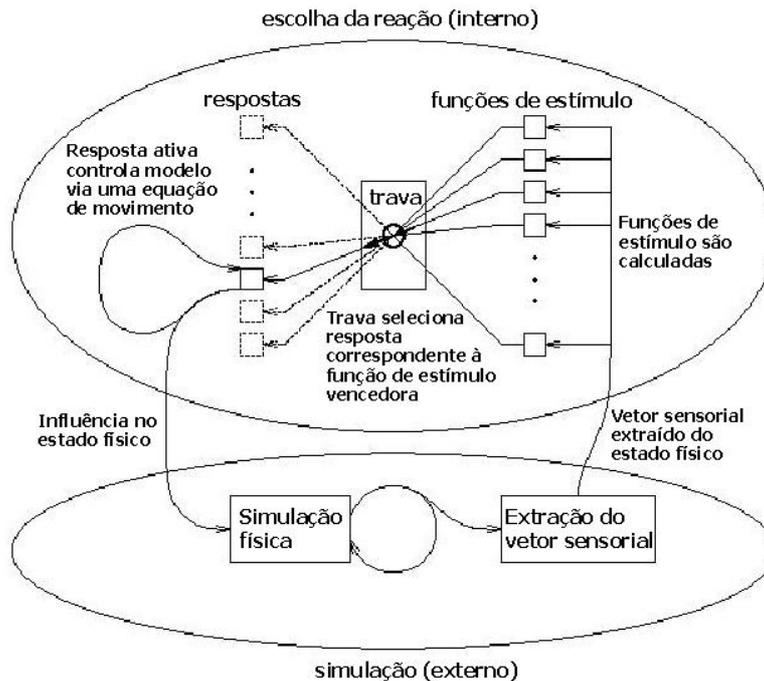


Figura 3.1: Representação estímulo-resposta (Ngo & Marks, 1993a).

Os tipos de parâmetros dessa representação são as configurações desejadas, associadas às respostas, e as posições e dimensões das regiões sensitivas. As regiões sensitivas são definidas dentro do espaço gerado pelas variáveis sensoriais e correspondem às regiões onde as funções de estímulo retornam valores positivos. A cada função de estímulo está associada uma região sensitiva. A Figura 3.3 mostra três exemplos de regiões sensitivas e suas respectivas respostas em uma situação em que o

movimento desejado é pular o mais alto possível. Essas regiões foram definidas manualmente num espaço gerado por duas variáveis sensoriais: a altura do centro de massa e a velocidade do centro de massa.

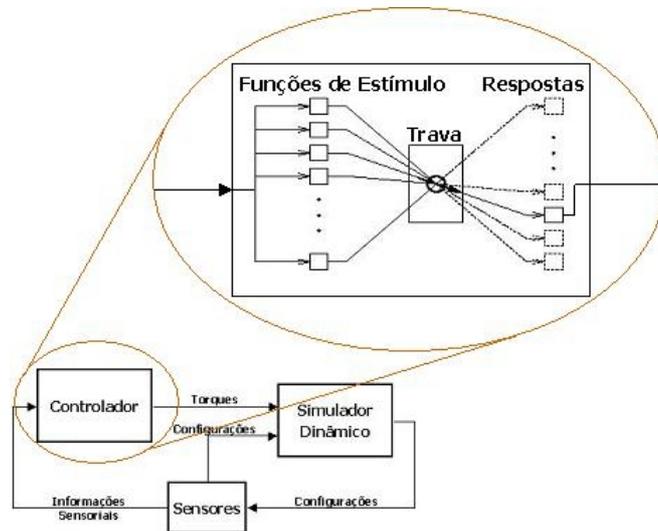


Figura 3.2: Interior do controlador (representação estímulo-resposta).

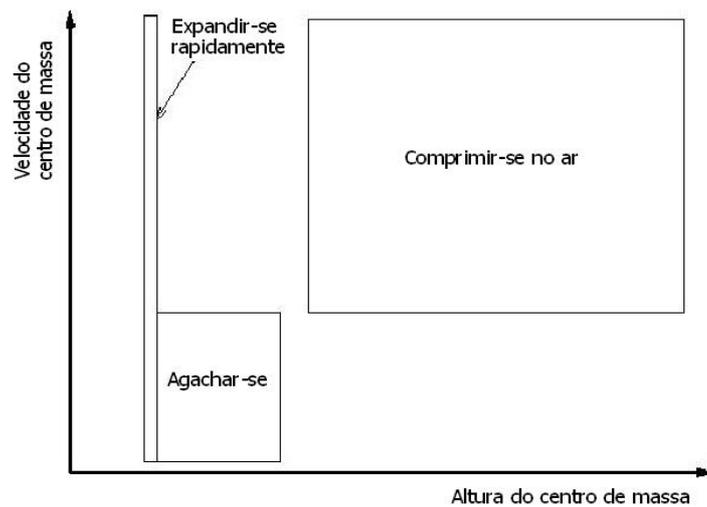


Figura 3.3: Regiões sensíveis e suas respostas associadas (Ngo & Marks, 1993a).

Ngo e Marks (1993a; 1993b) introduzem a representação estímulo-resposta e projetam controladores automaticamente, usando algoritmos genéticos para realizar a busca dos melhores valores para os parâmetros dos controladores. Auslander e seus co-autores (1995) dão continuidade a esse trabalho, comparando controladores baseados em sensores e em tempo, e conseguindo alguns bons resultados em modelos 3D.

3.2.2 Redes Sensores-Atuadores

Uma rede sensores-atuadores (Sensor-Actuator Network - SAN) representa controladores conectando os sensores aos atuadores do modelo através de uma rede de

conexões com pesos (Figura 3.4). A estrutura do controlador é definida por essa rede, que consiste de nós e conexões com pesos unidirecionais. Há três tipos de nós: nós sensores, nós escondidos e nós atuadores. Os nós sensores correspondem aos sensores e os nós atuadores aos atuadores do modelo. Nós escondidos são nós intermediários que influenciam as informações que vão dos nós sensores aos nós atuadores. O número de nós escondidos é aproximadamente igual ao número de nós sensores. Os nós sensores são conectados a todos os nós escondidos e a todos os nós atuadores. Todos os nós escondidos e nós atuadores são completamente interconectados.

Os valores dos nós sensores são os mesmos dos seus sensores associados. Em cada passo da simulação dinâmica, os valores dos nós sensores são verificados e transmitidos aos nós atuadores de acordo com os pesos das conexões da rede, através de operações realizadas pelos nós da rede. Os nós atuadores recebem os valores resultantes dessas operações e os mapeiam em ângulos desejados a serem assumidos pelas juntas. Controladores do tipo PD calculam os torques a serem gerados pelos atuadores a partir desses ângulos desejados. *Delays* são associados à operação dos nós da rede para tratar rápidas oscilações sensoriais. A Figura 3.5 ilustra o interior do controlador com um exemplo de estrutura modelada, de acordo com a Figura 2.2, quando uma SAN é utilizada para representá-lo. A SAN mostrada é apenas um exemplo específico.

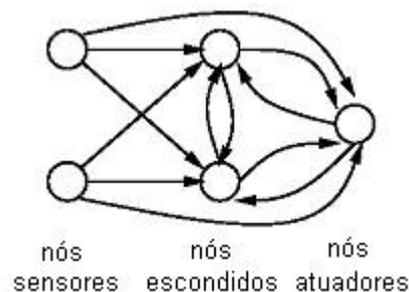


Figura 3.4: Rede sensores-atuadores (Van de Panne & Fiume, 1993).

Os tipos de parâmetros dessa representação são os pesos das conexões, os *delays*, os intervalos de ângulos desejados admissíveis definidos para cada atuador, as constantes das molas e dos amortecedores correspondentes aos controladores do tipo PD e os intervalos em que os sensores se tornam ativos.

Van de Panne e Fiume (1993) introduzem SANs como um tipo de representação de controladores, projetando controladores automaticamente, e realizam a busca dos melhores valores para os parâmetros dos controladores em duas fases. Na primeira fase, só os pesos são modificados. Já na segunda fase, usando os melhores pesos encontrados na primeira fase, pequenos ajustes são realizados nos demais parâmetros. Os parâmetros

a serem modificados a cada tentativa são escolhidos randomicamente. São testados dois métodos de busca: o primeiro consiste em sempre rejeitar os piores resultados e aceitar os melhores, tentando alcançar um ótimo local; o segundo, método de têmpera simulada, aceita alguns resultados piores a fim de evitar ótimos locais, tentando alcançar um ótimo global.

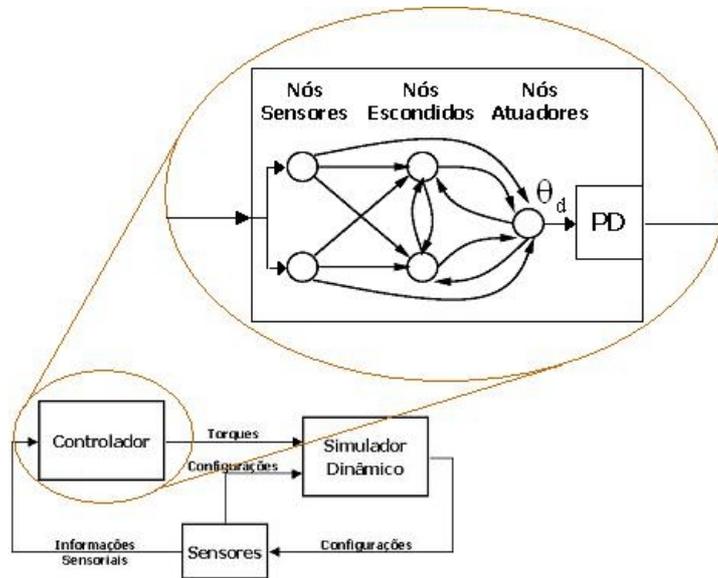


Figura 3.5: Interior do controlador (SAN).

Sims (1994) utiliza uma abordagem similar de representação, também conectando sensores a atuadores, e projeta automaticamente tanto o controlador quanto o modelo de estruturas articuladas controlado. Algoritmos genéticos são utilizados na busca, permitindo que o modelo e seu controlador evoluam em conjunto.

3.2.3 Máquinas de Estados Finitos

Ao se utilizar máquinas de estados finitos como representação de controladores, a estrutura do controlador é definida por estados e transições que os conectam. Os estados correspondem a fases do controlador, e determinam como o controlador deve agir. Transições entre estados são responsáveis por determinar qual estado deve estar ativo no decorrer da simulação dinâmica.

Comparada com as demais representações mencionadas anteriormente, essa representação permite uma melhor organização e é mais intuitiva para o animador. O animador tem uma maior facilidade de definir a estrutura desejada para um determinado movimento e de imaginar o resultado da ação de controle, o que ajuda na modificação dos parâmetros. Hodgins e seus co-autores (1995) usam máquinas de estados projetadas

manualmente para gerar animações de corrida, ciclismo e salto sobre cavalo (Figura 3.6).



Figura 3.6: Corrida e salto sobre cavalo (Hodgins et al., 1995).

Grafos de Controle de Pose

Grafos de controle de pose (Pose Control Graph – PCG) são máquinas de estados finitos, onde a cada estado está associada uma pose (Van de Panne, 1996). Uma pose consiste em um conjunto de ângulos desejados a serem assumidos pelas juntas do modelo. As transições são responsáveis por realizar a troca da pose ativa. Há dois tipos de transição: a temporal e a sensorial. Uma transição temporal define o número de iterações em que a pose atualmente ativa continuará ativa. A troca da pose ativa é realizada logo após a execução desse número de iterações, correspondente à simulação do tempo. Uma transição sensorial realiza a troca da pose ativa quando o sensor associado está ativo. A Figura 3.7 ilustra um exemplo de PCG aplicado a um modelo humanóide. As figuras desenhadas nos estados correspondem às poses a eles associadas.

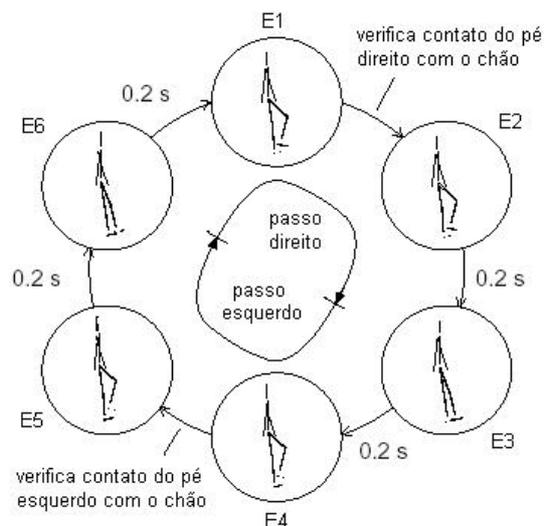


Figura 3.7: Um PCG utilizado no caminhar humano (Laszlo, 1996).

Um PCG pode ser considerado uma versão dinâmica da técnica de *keyframing*. A principal diferença é que, em um PCG, não há restrições explícitas quanto à trajetória

gerada. Uma pose define apenas uma configuração que se deseja alcançar, diferente da especificação de um quadro-chave, onde o modelo obrigatoriamente deve atingir a configuração especificada no instante determinado. Uma outra diferença é que uma pose não define posicionamento ou orientação global desejados. Em cada passo da simulação dinâmica, as informações da pose associada ao estado ativo são enviadas para os controladores do tipo PD, que calculam os torques a serem gerados pelos atuadores. Também é verificado se a próxima transição da máquina de estados deve ser realizada. A Figura 3.8 ilustra o interior do controlador com um exemplo de estrutura modelada, de acordo com a Figura 2.2, quando um PCG é utilizado para representá-lo. O PCG mostrado é apenas um exemplo específico.

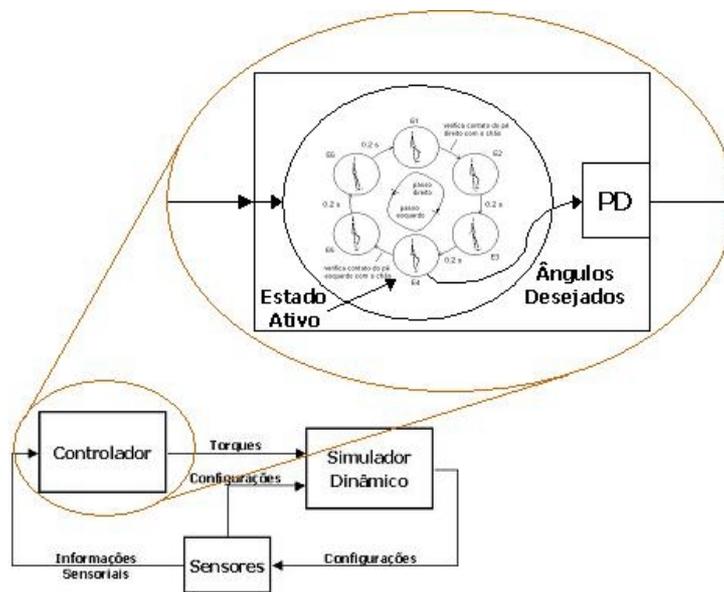


Figura 3.8: Interior do controlador (PCG).

Quanto à topologia da estrutura de PCGs, existem três categorias: cíclica, linear e mista (Figura 3.9). PCGs cíclicos são usados para movimentos periódicos (Van de Panne et al., 1994). Movimentos aperiódicos, como levantar, por exemplo, se encaixam na categoria linear. PCGs mistos são compostos por partes cíclicas e lineares e são usados para movimentos mais gerais.

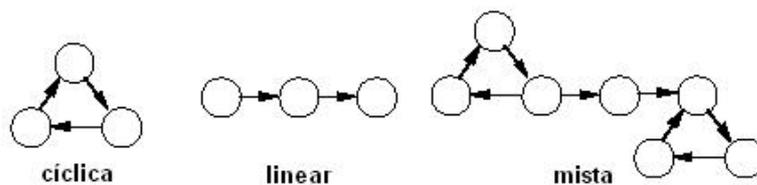


Figura 3.9: Topologias da estrutura de PCGs (Laszlo, 1996).

Os tipos de parâmetros dessa representação são os ângulos desejados das poses e os tempos das transições por tempo (Van de Panne, 1996). Van de Panne e seus co-

autores (1994) consideram os tempos de transição fixos, não fazendo parte dos parâmetros. Nesses trabalhos, os modelos utilizados e os movimentos gerados são relativamente simples e estáveis.

Van de Panne e seus co-autores (1994) introduzem a idéia de PCGs, projetando controladores automaticamente, e também realizam a busca dos melhores valores para os parâmetros dos controladores em duas fases. Na primeira fase é realizada uma busca mais global, modificando todos os parâmetros de uma vez. A segunda fase usa os melhores parâmetros encontrados na primeira fase e realiza pequenas modificações em alguns parâmetros de cada vez. Os parâmetros a serem modificados a cada tentativa são escolhidos randomicamente.

Laszlo (1996) introduz uma técnica de controle de laço fechado para estabilizar movimentos gerados por controladores de laço aberto. Um PCG usado para testes (o PCG mostrado na Figura 3.7) é projetado manualmente off-line, e, sozinho, não consegue estabilizar o movimento desejado para o modelo usado. Essa técnica analisa o ciclo do movimento e modifica os parâmetros desse PCG automaticamente on-line, de acordo com o feedback obtido, ao perceber variações consideráveis indicando a quebra da periodicidade do movimento. Com a aplicação dessa técnica, o modelo usado como exemplo consegue caminhar de maneira estável, e modelos de natureza dinâmica mais instável, comparados aos modelos utilizados em (Van de Panne et al., 1994), podem ser controlados.

3.3 Atribuição de Valores aos Parâmetros dos Controladores

Como já mencionado, a representação de um controlador consiste de uma estrutura, que é a base da representação, e de um conjunto de parâmetros modificáveis a ela associados. A estrutura permanece imutável após sua modelagem, que é feita visando um movimento específico. A geração do controlador fica completa após a atribuição de valores ao conjunto de parâmetros associados à sua estrutura. Entretanto, um diferente conjunto de valores pode ser atribuído, substituindo o atual, a fim de se tentar obter um melhor resultado.

Definida a representação do controlador e modelada sua estrutura, os valores dos parâmetros são especificados e podem ser avaliados e modificados. Dependendo do movimento desejado a ser gerado, os valores dos parâmetros podem ser modificados de diferentes maneiras, inclusive através de combinações. Portanto, o animador deve decidir qual dessas maneiras é a mais adequada para se modificar cada um dos

parâmetros, incluindo a opção de não modificá-lo. Essa decisão depende tanto da representação quanto do movimento desejado. A representação define os parâmetros passíveis de modificação que podem ser selecionados, e o movimento desejado indica qual a maneira mais adequada para modificar cada um dos parâmetros selecionados.

De uma maneira geral, os valores dos parâmetros podem ser modificados manualmente, pelo próprio animador, ou automaticamente, utilizando otimização estocástica. A avaliação dos valores atribuídos ao conjunto de parâmetros do controlador é realizada através da execução e análise da simulação dinâmica do modelo que é influenciado pelo controlador que está sendo avaliado. Essa análise pode ser realizada tanto pelo próprio animador, simplesmente visualizando o movimento gerado na tela do computador; quanto automaticamente, através de um critério de otimização.

Os valores dos parâmetros também podem ser atribuídos durante a simulação dinâmica (on-line): através da interação com o animador; ou automaticamente, utilizando feedback. Entretanto, esta seção trata da atribuição de valores aos parâmetros fora da simulação dinâmica (off-line). Interação com o animador é abordada na Seção 3.4 e modificação on-line automática é mencionada no final da Seção 3.2.3 e discutida na Seção 4.4.4.

3.3.1 Modificação Manual dos Parâmetros

A modificação manual dos valores dos parâmetros é realizada pelo próprio animador. Escolhas cuidadosas dos valores são exigidas e, conseqüentemente, o animador deve conhecer bem os detalhes do comportamento a ser gerado. Esses valores são geralmente baseados em dados medidos ou observados. A avaliação de um conjunto dos valores dos parâmetros do controlador, quando obtido através de modificação manual, geralmente é realizada, também pelo próprio animador, visualizando o movimento gerado na tela do computador.

Técnicas em que os valores dos parâmetros são modificados manualmente geralmente não servem para modelos ou movimentos arbitrários. A principal desvantagem da modificação manual é ser trabalhosa para o animador. Além disso, para alguns casos, pode levar até mais tempo do que a modificação automática. Entretanto, a atribuição manual de valores a alguns dos parâmetros geralmente é necessária para se conseguir o movimento desejado, principalmente para modelos mais complexos, os quais possuem um número elevado de parâmetros. Nesses modelos complexos, a escolha de um subconjunto de parâmetros para modificação manual diminui o número

de parâmetros cujos valores são atribuídos automaticamente e, conseqüentemente, diminui o espaço de busca utilizado na otimização estocástica. Os valores de alguns dos parâmetros também podem ser atribuídos manualmente com o objetivo de poderem ser utilizados como bons valores iniciais para a busca automática (Van de Panne, 1996).

3.3.2 Modificação Automática dos Parâmetros

O projeto de um controlador é suficientemente complexo para ser desenvolvido manualmente e não menos complexo e caro computacionalmente quando desenvolvido automaticamente. Entretanto, poder projetar controladores automaticamente, apesar do maior custo computacional exigido, é essencial para que controladores possam ser projetados de uma maneira fácil e acessível a qualquer usuário.

Os valores dos parâmetros são modificados automaticamente utilizando otimização estocástica. Modificações automáticas são realizadas atribuindo-se randomicamente valores aos parâmetros do controlador, podendo considerar valores iniciais ou apenas os seus intervalos de valores admissíveis (Van de Panne et al., 1994).

A avaliação dos valores dos parâmetros do controlador, obtidos através de modificação automática, é também realizada automaticamente, e exige a especificação de um critério de otimização, definido a partir do movimento desejado. Essa avaliação é realizada em relação a esse critério de otimização durante um número finito e pré-determinado de iterações de uma simulação dinâmica. Outra exigência é que cada uma dessas simulações deve ser realizada de maneira rigorosamente igual às outras (exceto pelos valores dos parâmetros do controlador) para que uma comparação adequada desses valores seja possível. A configuração inicial do modelo, por exemplo, deve ser a mesma em todas as simulações avaliadas. Um método de busca estocástico é empregado a fim de achar os valores dos parâmetros que sejam os mais adequados ao movimento desejado. Vários conjuntos de valores atribuídos automaticamente ao conjunto de parâmetros do controlador são avaliados, e aquele considerado o mais adequado ao movimento desejado é o que resulta no melhor valor de retorno (*fitness*) da avaliação, ou seja, o que gera o movimento mais apto, segundo o critério de otimização.

O processo de otimização estocástica geralmente exige várias tentativas de avaliação. Cada uma delas, por sua vez, exige a execução de uma simulação dinâmica. Isso acarreta um elevado custo computacional devido ao grande número de simulações exigidas na busca. Como o espaço de busca tem uma enorme influência nesse número de simulações, seu tamanho deve ser razoável para que a otimização estocástica obtenha

bons resultados. O espaço de busca aqui considerado é definido pelos parâmetros escolhidos para serem modificados automaticamente. Uma boa forma de diminuir o espaço de busca é analisar a simetria dos parâmetros (Van de Panne, 1996).

Critério de Otimização

O critério de otimização tenta guiar o avanço do comportamento obtido pelo controlador a fim de obter o comportamento desejado. A especificação de um critério adequado não é uma tarefa fácil e fornece apenas um controle de alto nível ao animador. Geralmente são utilizados critérios simples. Por exemplo, maximizar a distância percorrida pelo centro de massa do modelo ao final da simulação física traz bons resultados, quando utilizado para obter movimentos de locomoção. Esse critério confia na idéia de que quanto mais longe o controlador levar o modelo, mais natural será o modo de locomoção obtido, apesar de não haver garantias (Van de Panne & Fiume, 1993). Um outro exemplo de critério de otimização é maximizar a altura máxima atingida pela parte mais baixa do modelo, utilizado para obter um salto.

Penalidades podem ser atribuídas para auxiliar o avanço correto do movimento. Assim, por exemplo, quando um modelo que pretende caminhar cair no chão ele é penalizado. As penalidades podem ser parciais ou totais, chegando até a desconsideração completa do resultado. Um ponto importante a ser observado na avaliação automática de um controlador é que a simulação dinâmica é executada durante um número finito de iterações, o que não garante que o desempenho do controlador seja mantido durante um número maior de iterações. Outra observação importante é que não se utiliza necessariamente um critério de parada (convergência) para o processo de otimização, podendo-se analisar um número pré-determinado de tentativas.

3.4 Interação com o Animador

A interação com o animador ocorre quando suas ações interferem na animação. Em jogos, sua utilização é fundamental. Além dessa utilização, a possibilidade do animador acompanhar visualmente o efeito do controlador sobre o modelo durante a simulação dinâmica, e interferir no processo através da modificação de parâmetros do controlador, pode ser percebida como uma maneira mais simples de substituir sensores poderosos que seriam difíceis de serem projetados, sendo muito úteis no projeto de controladores. Como mencionado anteriormente, os valores dos parâmetros também podem ser atribuídos durante a simulação dinâmica (on-line). A atribuição interativa é

caracterizada por realizar atribuições on-line a partir da interação com o animador. Esses valores dos parâmetros atribuídos podem ser salvos para gerarem o mesmo movimento sem precisar novamente da interação com o animador.

Quanto ao uso de feedback, pelo fato da interação não utilizar informações do modelo ou do ambiente de uma forma direta, essa forma de laço recebe uma classificação diferente chamada de “usuário-no-laço”, pois o feedback é fornecido indiretamente pelo animador, o usuário. A Figura 3.10 mostra como o esquema da Figura 2.2 pode ser adaptado para incluir a interação entre o controlador e o animador. O animador recebe as informações visuais da tela do computador e retorna o feedback através de algum dispositivo de entrada. Uma interface apropriada recebe as informações do dispositivo de entrada e mapeia em valores adequados para os parâmetros do controlador, que atua na simulação dinâmica e gera a animação na tela do computador, mandando novamente informações visuais para o animador.

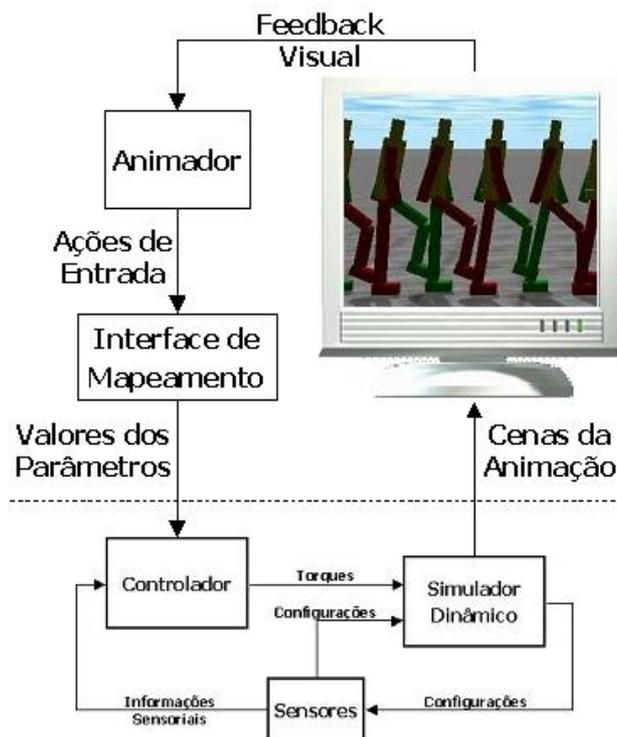


Figura 3.10: Processo de interação do controlador com o animador.

A maneira mais óbvia de modificar um conjunto de parâmetros é utilizar um dispositivo de entrada, pelo qual o animador executa suas ações, tendo um número de graus de liberdade equivalente a esse conjunto. Entretanto, quanto maior o conjunto de parâmetros, mais difícil se torna fazer o mapeamento entre os graus de liberdade dos dispositivos de entrada e os parâmetros do controlador. Mesmo com a disponibilidade de dispositivos de entrada com muitos graus de liberdade, não é razoável assumir que

um animador consiga aprender a manipular qualquer número de graus de liberdade. Portanto, para um grande número de parâmetros, um grau de liberdade de entrada é geralmente mapeado em mais de um parâmetro do controlador.

Interfaces de mapeamento apropriadas a diferentes movimentos de vários modelos 2D são projetadas em (Laszlo et al., 2000). As ações do animador estão associadas aos dispositivos de entrada e podem resultar em modificações dos valores dos parâmetros de maneira contínua (ações contínuas), ou de maneira discreta (ações discretas). O mouse e o teclado são exemplos de dispositivos de entrada de ação contínua e discreta respectivamente. A Figura 3.11 ilustra um exemplo de ação contínua, onde os ângulos desejados das duas juntas atuadas são os parâmetros do controlador, e as coordenadas da posição do mouse relativa à posição do centro da área de visualização da animação são mapeadas linearmente nesses parâmetros. Assim, a movimentação do mouse resulta em modificações contínuas apropriadas dos ângulos desejados dessas juntas. A Figura 3.12 ilustra um exemplo de ação discreta, onde há o mapeamento de uma tecla do teclado a uma pose pré-determinada de um dos membros. Para cada membro há seis poses correspondendo a seis teclas, devidamente localizadas no teclado. Controladores do tipo PD são utilizados nesses dois exemplos para gerar os torques correspondentes aos ângulos desejados.

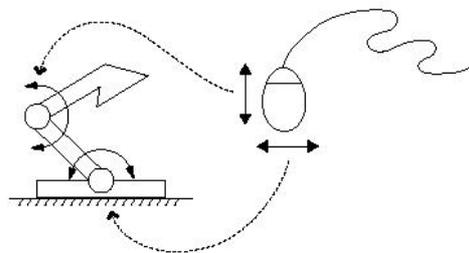


Figura 3.11: Ação contínua aplicada a uma luminária (Laszlo et al., 2000).

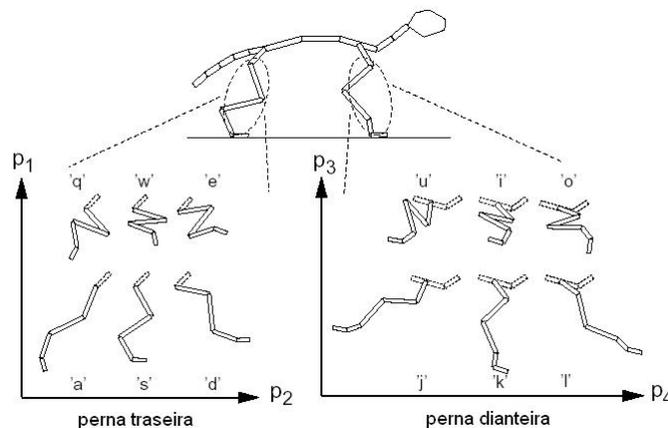


Figura 3.12: Ação discreta aplicada aos membros de um gato (Laszlo et al., 2000).

Zhao (2004) apresenta uma interface mais elaborada e amigável, permitindo o controle de um maior número de parâmetros, e comprova sua eficácia gerando animações em 3D de saltos ornamentais, saltos de esqui e *snowboarding*. A interface consiste de uma paleta usada para gerar poses e definir o tempo de transição entre elas interativamente. Diversas poses pré-projetadas são disponibilizadas na paleta e o animador vai selecionando, conforme o movimento desejado, as poses mais apropriadas para estarem ativas no instante analisado. Ao clicar com o mouse no botão correspondente à pose apropriada, o animador fornece a exata posição do clique relativa ao botão, definindo exatamente a pose ativa e o tempo de transição da pose anterior para a atual. Cada botão possui a idéia da ação desejada, e a pose exata, assim como outros parâmetros, pode ser diferente, dependendo da posição do clique dentro do botão. O controlador resultante dessa interação é um PCG que pode ser ajustado off-line manual ou automaticamente a fim de obter melhores resultados. A Figura 3.13 mostra a paleta usada para os saltos ornamentais e a Figura 3.14 ilustra um PCG gerado interativamente utilizando essa paleta. Zhao e van de Panne (2005) apresentam também uma outra interface que explora a utilização de um *gamepad* como dispositivo de entrada.

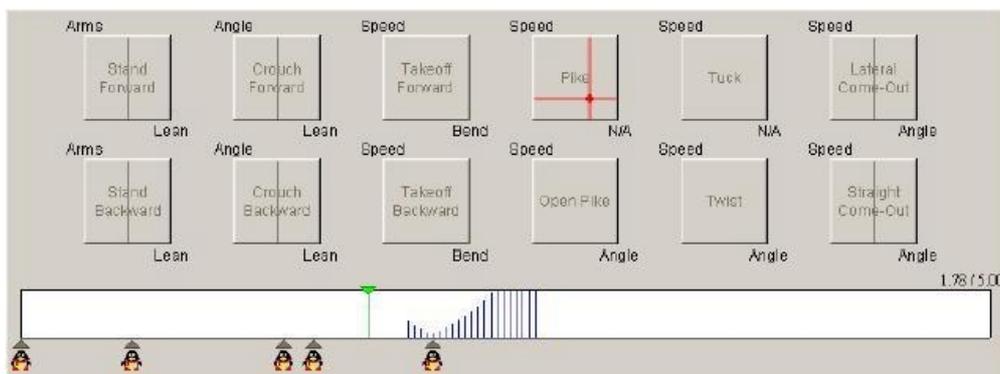


Figura 3.13: Paleta usada para os saltos ornamentais (Zhao & van de Panne, 2005).

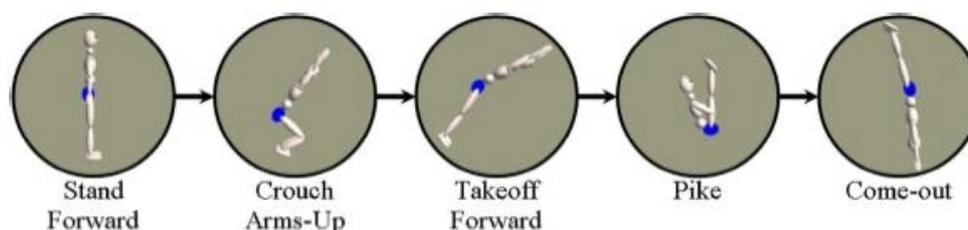


Figura 3.14: Exemplo de um PCG gerado interativamente (Zhao, 2004).

3.5 Composição de Controladores

A composição de controladores consiste em organizar as ações de mais de um controlador em seqüência cronológica, possibilitando que movimentos mais gerais e

complexos sejam gerados. Entretanto, nas transições entre os controladores, mesmo havendo garantia de obediência às leis da física, também há problema de controle. As transições entre os controladores devem ser realizadas de forma natural e suave. Através da composição de controladores, modelos com vasto repertório de movimentos podem ser obtidos com menor dificuldade.

Faloutsos e seus co-autores (2001) apresentam um *framework* de composição de controladores. Definido um modelo, o *framework* é composto por um conjunto de controladores individuais, situados em um nível mais baixo, e por um controlador supervisor, que decide qual controlador individual deve estar atuando no modelo em cada instante e o momento mais apropriado para que ele comece a atuar. Os controladores individuais funcionam como caixas pretas, não podendo sofrer alterações, e cada um é responsável por fornecer o controle adequado a um movimento específico correspondente. Esses controladores devem fornecer também as pré-condições em que eles funcionam apropriadamente e as pós-condições responsáveis por determinar seu sucesso. Ao estarem ativos, esses controladores devem ser capazes de determinar se estão operando satisfatoriamente, se obtiveram sucesso ou se falharam. Qualquer controlador com essas características pode ser adicionado ao conjunto de controladores individuais. A idéia é poder acoplar cada vez mais controladores ao repertório de movimentos do modelo.

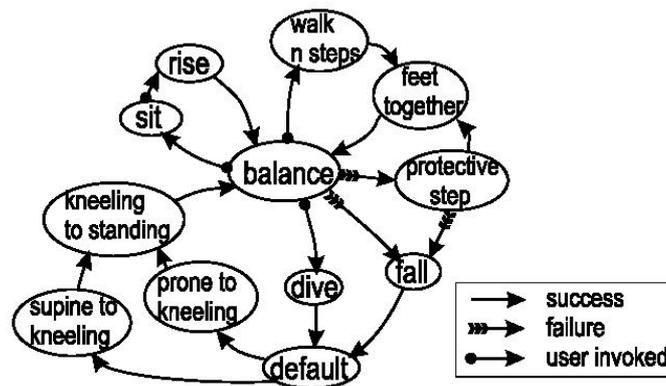


Figura 3.15: Framework para composição de controladores (Faloutsos et al., 2001).

As pré-condições correspondem ao conjunto de estados iniciais do modelo em que o controlador, sob condições normais de ambiente, consegue obter sucesso na geração do seu movimento. Elas funcionam como sensores que mandam informações para o controlador supervisor, ajudando-o a realizar as transições corretas nos instantes mais adequados. O controlador supervisor pode trocar o controlador ativo quando perceber que ele obteve sucesso, quando perceber que ele falhou ou através da interação

com o animador. A Figura 3.15 mostra o esquema do controle realizado por um controlador supervisor sobre um conjunto de controladores individuais projetados para um modelo humanóide 2D. Esses controladores individuais, capazes de gerar movimentos básicos tais como manter o equilíbrio, executar reações de proteção quando o equilíbrio é perdido e levantar de várias maneiras após uma queda, são utilizados para avaliar o *framework* apresentado. A Figura 3.16 mostra um exemplo de animação desse modelo 2D gerado utilizando esse *framework*.



Figura 3.16: Animação através da composição de controladores (Faloutsos, 2002).

3.6 Considerações Finais

Neste capítulo, controladores e a forma como funcionam foram discutidos em maiores detalhes. Alguns conceitos necessários à compreensão das contribuições deste trabalho foram introduzidos, e alguns trabalhos relacionados às idéias associadas a essas contribuições foram abordados.

Um dos principais problemas discutidos, nos trabalhos relacionados mencionados neste capítulo, e que ainda requer melhoramentos, é a dificuldade na geração dos controladores. Mais especificamente, a geração automática possui sérias limitações quando aplicada a modelos e movimentos mais complexos ou mais instáveis.

Devido ao alto custo computacional, controladores para modelos mais complexos não podem ser gerados de forma exclusivamente automática. Esse custo depende muito da representação e dos parâmetros escolhidos para serem obtidos automaticamente, os quais geram o espaço de busca. Portanto, o tipo de representação utilizado é de fundamental importância na geração dos controladores. Um ponto importante a ser analisado é a utilização de sensores, que, em PCGs, ainda não foi completamente explorada.

Capítulo 4

Máquinas de Estados Sensoriais

4.1 Introdução

O controlador ideal deve ser capaz de gerar movimentos que se aproximem ao máximo dos requisitos de movimento estabelecidos pelo animador, da maneira mais rápida e automática possível. É justamente a dificuldade de gerar controladores com esse grau de versatilidade o principal problema da utilização da técnica de controladores.

O projeto de um controlador é suficientemente complexo para ser desenvolvido manualmente e não menos complexo e caro computacionalmente quando desenvolvido automaticamente. No entanto, enquanto o projeto manual dos parâmetros do controlador oferece um maior controle sobre esses parâmetros, o projeto automático é realizado em um nível mais alto em que o animador não se responsabiliza por esse trabalho manual, exigindo um menor esforço de sua parte. Portanto, apesar do menor controle sobre os parâmetros e do maior custo computacional exigido, poder projetar controladores automaticamente é essencial para facilitar seu projeto e tornar esse processo acessível a qualquer usuário.

Controladores para modelos mais complexos e instáveis, utilizando as técnicas disponíveis atualmente, ainda não podem ser completamente projetados de maneira automática. Geralmente, dependendo da complexidade do modelo e do movimento desejado, apenas alguns dos parâmetros do controlador são escolhidos para serem gerados automaticamente. Assim, tanto o custo computacional quanto a eficácia do projeto automático de controladores depende muito de como o controlador é representado e desses parâmetros escolhidos, os quais geram o espaço de busca.

Neste capítulo, é proposta uma representação de controladores chamada máquinas de estados sensoriais, que se baseia em máquinas de estados finitos. Essa representação é bem organizada e intuitiva, e utiliza a idéia de PCGs.

Um ponto importante no uso de controladores é a utilização de sensores (feedback). Em PCGs, os sensores utilizados ainda requerem aprimoramentos, visto que suas estruturas não permitem que sejam tão explorados quando gerados automaticamente. Dos trabalhos relacionados discutidos no Capítulo 3, alguns não modificam automaticamente os sensores utilizados, e outros modificam automaticamente apenas os intervalos em que os sensores se tornam ativos (Van de

Panne & Fiume, 1993) ou as posições e dimensões de regiões sensitivas (Ngo & Marks, 1993a; Ngo & Marks, 1993b). A interação com o animador é uma forma de utilização externa do sensor visual do próprio animador e ajuda na obtenção de melhores resultados. Para conseguir resultados comparáveis aos obtidos com a intervenção do animador (intervenção que nem sempre é desejável), estruturas de sensores mais sofisticadas devem ser utilizadas. Uma estrutura mais geral e mais elaborada para os sensores utilizados também é proposta, permitindo que eles sejam melhor explorados quando gerados automaticamente.

No restante deste capítulo, são discutidos em detalhes: a estrutura dos controladores, incluindo a estrutura proposta para os sensores; os tipos de parâmetros disponibilizados pela representação proposta; e a forma como é realizada a atribuição de valores aos parâmetros do controlador.

4.2 Estrutura do Controlador

Modelos físicos controlados por PCGs geralmente são associados a um único PCG o qual controla simultaneamente todos os atuadores do modelo através do envio de sinais (Van de Panne, 1996). As poses do PCG são definidas por ângulos desejados nas juntas onde estão colocados os atuadores do modelo. Modelar a estrutura desse tipo de controlador consiste em organizar seus estados e transições, associando uma pose desejada a cada estado; e em definir, para cada transição, quais os estados que ela conecta e qual o tipo que a qualifica (transição temporal ou transição sensorial). Os tempos das transições temporais podem ser modificados, portanto, fazem parte dos parâmetros do controlador; já os sensores associados às transições sensoriais geralmente não podem ser modificados.

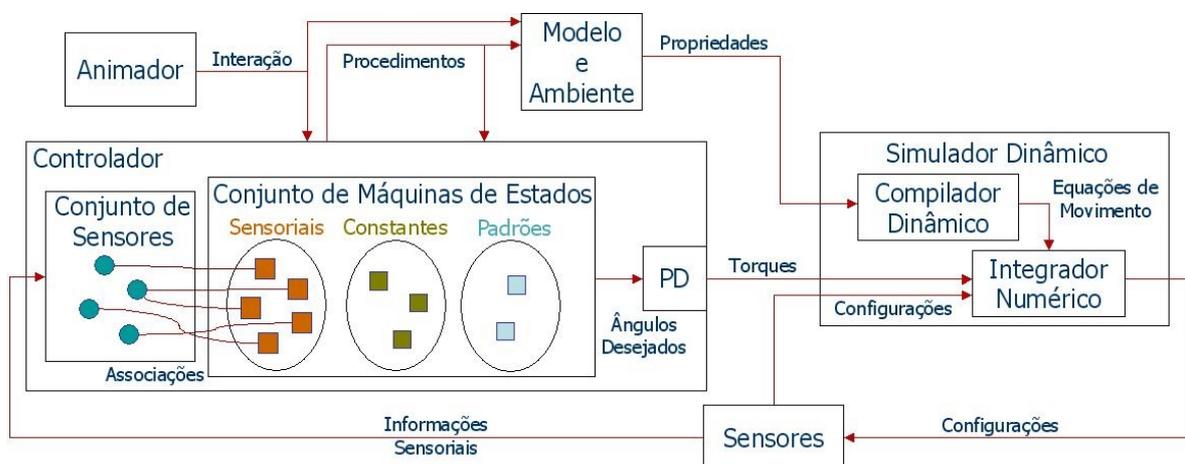


Figura 4.1: Estrutura da representação “máquinas de estados sensoriais”.

A estrutura do controlador proposta neste trabalho está ilustrada na Figura 4.1 (essa figura será melhor explorada na Seção 5.3.1). Sua representação, denominada “máquinas de estados sensoriais”, consiste de um conjunto de máquinas de estados, que podem atuar simultaneamente no modelo controlado, e de um conjunto de sensores associados a algumas dessas máquinas de estados. A possibilidade de definir um conjunto de máquinas de estados em um mesmo modelo é uma das diferenças fundamentais entre a estrutura proposta e a estrutura de PCGs convencionais.

4.2.1 Máquinas de Estados Componentes

Apesar de poderem atuar todas simultaneamente, as máquinas de estados da representação “máquinas de estados sensoriais” não atuam assim o tempo todo. Algumas permanecem ativas durante toda a simulação, mas outras são ativadas ou desativadas em função dos sinais recebidos do conjunto de sensores da estrutura. Em cada iteração da simulação dinâmica, todo o conjunto de sensores é analisado, determinando as máquinas de estados que estão ativas. Quanto à dependência desse conjunto de sensores, essas máquinas de estados são classificadas como:

- ▶ Constantes – quando não possuem nenhum sensor associado, e permanecem ativas durante todo o tempo;
- ▶ Sensoriais – quando possuem um sensor associado, e estão ativas quando esse sensor está ativo (um sensor pode ser associado a mais de uma dessas máquinas sensoriais);
- ▶ Padrões – quando dependem de todos os sensores para estarem ativas, embora não possuam nenhum associado. Estão ativas quando nenhum sensor está ativo.

4.2.2 Organização das Máquinas de Estados Componentes

O nível mais alto de abstração da estrutura de controle proposta consiste na organização de seu conjunto de máquinas de estados. É importante ressaltar que dado o modelo da estrutura articulada a ser animada e o tipo de movimento desejado (levantar, caminhar, correr, saltar, etc.), máquinas de estados são definidas para representar as mudanças de estados de partes do modelo durante o movimento especificado. Assim, uma máquina de estados pode estar associada à movimentação das pernas, uma outra máquina pode representar a movimentação do conjunto tronco e braços, e assim por diante. A divisão e organização das máquinas de estados para obtenção do movimento desejado é uma

questão de projeto do controlador. A utilização de mais de uma máquina de estados permite que controladores com a representação “máquinas de estados sensoriais” gerem tipos de movimentos diferentes. As máquinas de estados podem ser organizadas para atuar simultaneamente no modelo ou em seqüência cronológica (composição de movimentos). Essas máquinas de estados são semelhantes a PCGs e também enviam valores de ângulos desejados para controladores do tipo PD, que são utilizados para calcular os torques correspondentes a serem aplicados pelos atuadores.

Em função das partes do modelo atreladas a uma máquina de estados, ela pode afetar subconjuntos de atuadores do modelo (Oshita & Makinouchi, 2001). A Figura 4.2 ilustra várias máquinas de estados atuando simultaneamente num mesmo modelo. O modelo utilizado como exemplo é mostrado com todos os seus atuadores numerados. As máquinas de estados são ilustradas através de retângulos, e os números escritos no interior de cada uma delas indicam quais os atuadores que elas afetam. Os círculos mostrados no interior de uma das máquinas de estados ilustram seus estados, e indicam que os estados são independentes, podendo afetar quaisquer subconjuntos de atuadores.

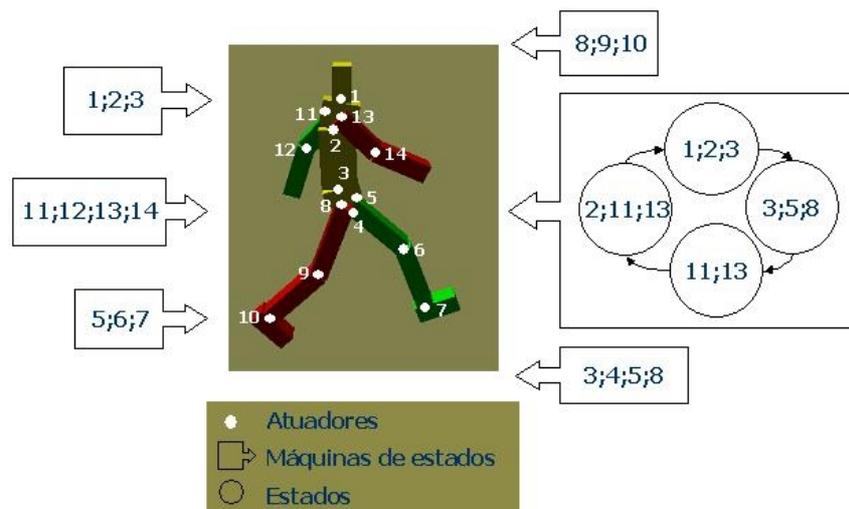


Figura 4.2: Conjunto de máquinas de estados atuando simultaneamente em subconjuntos dos atuadores de um modelo.

Na Figura 4.2, nota-se também que um atuador pode ser afetado simultaneamente por mais de uma máquina de estados, isto é, um controlador do tipo PD pode receber mais de um valor de ângulo desejado numa mesma iteração da simulação dinâmica, e então, calcular os torques correspondentes a todos esses valores recebidos e computar a média ponderada desses torques, para obter o torque resultante a ser aplicado pelo atuador. Os pesos utilizados nessa média são os respectivos pesos associados às máquinas de estados que enviam os sinais.

A principal vantagem de máquinas de estados atuando em subconjuntos dos atuadores de um modelo é poder gerar o movimento desejado por partes, sem precisar analisar todos os atuadores ao mesmo tempo. Dessa forma, diferentes partes do modelo podem ser tratadas com mais independência. Ajustes podem ser realizados posteriormente para que as máquinas de estados funcionem bem em conjunto. Por exemplo, em um caminhar, é apenas a parte inferior do modelo (quadril e membros inferiores) que necessita manter mais precisamente um padrão em seu movimento. O movimento da parte superior do modelo tem influência em seu equilíbrio, no entanto, o problema de controle associado a esse movimento permite maior liberdade de configurações, já que há várias configurações possíveis que não causam a perda do equilíbrio. Portanto, uma única máquina de estados pode ser projetada para a parte inferior, responsável por manter um padrão de passadas, enquanto algumas outras máquinas de estados podem ser projetadas para diferentes tipos de movimento da parte superior, tais como: simplesmente deixar os braços balançarem, acompanhando as passadas; bater palmas; pegar ou jogar algum objeto; e coçar a cabeça.

4.2.3 Estrutura das Máquinas de Estados

O nível mais baixo de abstração da estrutura de controle proposta consiste nas estruturas individuais das máquinas de estados componentes e nas estruturas dos sensores utilizados (Seção 4.2.4). A estrutura das máquinas de estados utilizadas neste trabalho possui características semelhantes às de PCGs, mas apresenta algumas peculiaridades. Uma semelhança com os PCGs é que essas máquinas de estados consistem de estados conectados por transições, aos quais estão associadas poses. Uma das peculiaridades é que essas poses, conforme mostrado na Figura 4.2, podem se referir a qualquer subconjunto dos atuadores do modelo.

As transições entre estados podem ser de três tipos: temporal, sensorial e por método. As transições temporais realizam a troca do estado ativo quando um determinado número de iterações (medida de tempo) é executado. As transições sensoriais trocam o estado ativo quando o sensor associado está ativo. As transições por método, por sua vez, trocam o estado ativo quando o método apropriado é executado, o que possibilita interação com o animador. Por exemplo, o método pode ser executado pressionando uma tecla, permitindo ao animador realizar a transição no instante mais conveniente, de acordo com sua percepção visual. Os sensores associados às transições

sensoriais fazem parte da estrutura de uma máquina de estados e possui estruturas iguais às dos sensores associados às máquinas de estados sensoriais (Seção 4.2.4).

Para que as transições entre a pose associada ao estado ativo corrente e aquela associada ao próximo estado ativo sejam suaves, interpolação linear dos ângulos desejados dessas duas poses é utilizada. Cada transição possui um atraso, correspondendo ao tempo de duração da interpolação entre os estados que ela conecta. A interpolação entre poses também evita que torques muito grandes sejam aplicados nas juntas do modelo e que, por consequência, instabilidades numéricas ocorram (Zhao, 2004). Além disso, ao torque gerado por cada atuador é associado um limite máximo, que é obedecido pelos controladores do tipo PD.

A topologia da estrutura das máquinas de estados é cíclica. Os estados são definidos em um vetor e transições devem ser definidas conectando os estados vizinhos, e o último estado (último elemento do vetor) deve ser conectado ao primeiro estado (primeiro elemento do vetor), como mostrado na Figura 4.3. Entretanto, essas máquinas de estados podem ser utilizadas como lineares, simplesmente definindo sua última transição de tal forma que ela nunca ocorra, como, por exemplo, definindo-a por método, sem nunca executá-lo. Máquinas de estados mistas podem ser simuladas utilizando mais de uma máquina de estados, o que é permitido na representação “máquinas de estados sensoriais”.

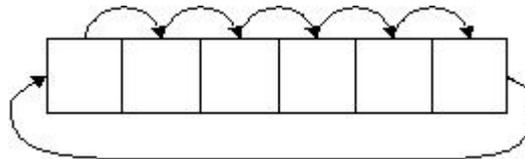


Figura 4.3: Vetor de estados cíclico.

A uma máquina de estados é associado um valor que determina o peso de sua influência no torque gerado por um atuador. Os pesos das máquinas de estados permitem utilizar uma média ponderada para o cálculo do torque resultante realizado pelos controladores do tipo PD. Dessa forma, umas máquinas de estados podem ter mais influência no movimento do que outras.

É possível também desabilitar ou habilitar uma máquina de estados. A habilitação de uma máquina de estados depende apenas do valor de um atributo que pode ser modificado a qualquer momento, e é independente de ela estar ativa ou não. Entretanto, para que uma máquina de estados atue no modelo, ela necessita estar ativa e habilitada. Poder desabilitar e habilitar máquinas de estados facilita a composição de

movimentos. Essas ações podem ser realizadas através da interação com o animador ou automaticamente.

4.2.4 Estrutura dos Sensores

A estrutura de sensores proposta neste trabalho é mais geral e elaborada do que as encontradas na literatura. Ela permite que os sensores sejam melhor explorados, principalmente quando gerados automaticamente. Em alguns trabalhos, o funcionamento dos sensores já é pré-especificado e o significado dos seus parâmetros se limita ao intervalo em que os sensores se tornam ativos (Ngo & Marks, 1993a; Ngo & Marks, 1993b; Van de Panne & Fiume, 1993). Assim como as estruturas utilizadas para representar controladores, a estrutura proposta para os sensores utilizados neste trabalho também possui parâmetros vinculados, cujos significados são determinados pelo próprio animador na definição do sensor. Essa característica não limita os significados dos parâmetros, e, portanto, permite utilizar parâmetros mais expressivos. Conseqüentemente, modificações nos valores desses parâmetros possibilitam alterações mais gerais e mais intuitivas no funcionamento do sensor, de acordo com o desejo do animador ao defini-lo. Isso facilita a especificação do movimento desejado pelo animador e permite que os sensores sejam melhor explorados na geração automática do controlador. Os parâmetros dos sensores fazem parte dos parâmetros do controlador.

Essa estrutura de sensores mais geral proposta consiste de três ingredientes: os cálculos e testes que determinam o retorno do sensor; os parâmetros do sensor; e o próprio retorno do sensor. Esses cálculos e testes, fazem parte do funcionamento interno do sensor, e são especificados a partir dos parâmetros do sensor e das configurações atuais do modelo e do ambiente. Portanto, os parâmetros influenciam diretamente o resultado retornado pelo sensor. Esses parâmetros são organizados em um vetor, pela ordem em que são definidos. O valor de retorno, que determina se o sensor está ativo ou não, é um número real. Isso permite que o valor de retorno seja melhor utilizado tanto em procedimentos atrelados a sensores associados a máquinas de estados sensoriais; quanto em procedimentos associados a transições sensoriais. Visto que esses procedimentos só são executados quando os sensores aos quais estão associados tornam-se ativos, a faixa de valores de entrada para esses procedimentos pode ser bastante ampla (incluindo até mesmo valores negativos) se apenas o valor de retorno igual a zero for associado ao estado inativo do sensor. Esse assunto é melhor discutido na Seção 4.4.4. Quanto aos sensores associados a máquinas de estados sensoriais, suas

estruturas consistem também de um mecanismo de retardo e podem possuir procedimento associado.

À guisa de exemplo, um sensor que verifica a altura do centro de gravidade do modelo é definido. Um sensor desse tipo pode ser utilizado para gerar um salto, por exemplo. A idéia principal de seu funcionamento é que ele esteja ativo quando o centro de gravidade do modelo estiver abaixo de certa altura, embora a altura mais adequada não seja conhecida pelo animador. Assim, para determinar o retorno desse sensor, o centro de gravidade do modelo é calculado e, em seguida, testado com um parâmetro definido para esse sensor, cujo significado é a altura mínima que o centro de gravidade pode atingir para que o sensor não esteja ativo. Esse sensor possui apenas um parâmetro. Se a altura do centro de gravidade for menor do que o valor do parâmetro, o seu retorno recebe um valor diferente de zero e o sensor fica ativo, caso contrário, o seu retorno recebe zero e o sensor fica inativo. A Figura 4.4 ilustra um possível trecho de código C++ correspondente aos cálculos e testes, realizados por esse sensor, responsáveis por determinar seu retorno.

```
alturaCOG = modelo->alturaCOG();  
if (alturaCOG < sensor.parametros[0].valor)  
{  
    sensor.retorno = 1.0; // ativo  
} else {  
    sensor.retorno = 0.0; // inativo  
}
```

Figura 4.4: Possível trecho de código C++ correspondente aos cálculos e testes, realizados por um sensor, responsáveis por determinar seu retorno.

Esta estrutura então permite que, nesse exemplo, a altura mais adequada possa ser encontrada modificando-se esse parâmetro automaticamente, permitindo que o animador possa pensar no funcionamento do sensor em um nível de abstração mais alto. A especificação precisa dos valores dos parâmetros do sensor, correspondente ao nível mais baixo, fica sob a responsabilidade da otimização. Embora, nesse exemplo, quando o sensor está ativo, o valor de retorno é sempre igual a 1.0; esse valor de retorno poderia ter sido definido como função da altura do centro de gravidade calculada e poderia ser utilizado pelo controlador para executar ações mais elaboradas.

Neste trabalho, esta estrutura de sensores é utilizada tanto nos sensores associados a máquinas de estados quanto nos sensores associados a transições.

4.2.5 Modelagem da Estrutura do Controlador

A estrutura de um controlador é modelada de acordo com o movimento desejado. A estrutura da representação apresentada neste trabalho é modelada definindo seu conjunto de máquinas de estados e seu conjunto de sensores. Para modelar a estrutura de cada máquina de estados, primeiramente, o número de estados é informado. Para cada estado, define-se uma pose, especificando-se quais atuadores ela afeta, e, em seguida, associa-se essa pose ao estado. Para cada transição, são escolhidos os estados que ela conecta (vide Figura 4.3) e o tipo da transição. Para cada transição sensorial, define-se um sensor e o associa à transição. Os sensores associados a transições são independentes dos sensores associados a máquinas de estados e não fazem parte do conjunto de sensores definido inicialmente. Por último, cada estado ou transição pode ser associado a um procedimento.

Além de se modelar cada máquina de estados individualmente, os seus tipos são definidos quanto à dependência do conjunto de sensores e a cada máquina de estados sensorial é associado um sensor pertencente a esse conjunto. Cada sensor desse conjunto pode também ser associado a um procedimento. A definição dos procedimentos também faz parte da modelagem da estrutura do controlador. Para modelar a estrutura dos sensores, os cálculos e testes responsáveis por determinar seus retornos são especificados.

4.3 Tipos de Parâmetros



Figura 4.5: Tipos de parâmetros vinculados à estrutura da representação “máquinas de estados sensoriais”.

Os parâmetros são responsáveis por completar a estrutura modelada, determinando exatamente as ações realizadas pelo controlador. Esta representação busca disponibilizar diferentes tipos de parâmetros mais expressivos e intuitivos. A Figura 4.5 apresenta uma

visão geral dos tipos de parâmetros disponíveis na representação “máquinas de estados sensoriais”. Os tipos de parâmetros, indicados pelas elipses, são mostrados encaixados no modelo de sua estrutura, fixando a idéia de como eles estão vinculados a ela. As transições por método não possuem exclusivamente nenhum tipo de parâmetro.

Os significados dos tipos de parâmetros mostrados na Figura 4.5 são discutidos a seguir:

- ▶ Peso – O peso de uma máquina de estados indica a influência que ela possui sobre os torques gerados pelos atuadores para os quais ela envia sinais;
- ▶ Habilitada – O atributo, “habilitada”, de uma máquina de estados indica se ela está habilitada ou desabilitada;
- ▶ Ângulos desejados – Os ângulos desejados de uma pose são os ângulos a serem assumidos pelas juntas do modelo, através dos torques aplicados pelos atuadores. Esses torques são calculados por controladores do tipo PD;
- ▶ Delay de interpolação – O *delay* de interpolação de uma transição corresponde ao tempo de duração da interpolação entre as poses dos estados que ela conecta;
- ▶ Tempo – O tempo de uma transição temporal corresponde ao número de iterações da simulação dinâmica que devem ser executadas logo após o estado anterior dessa transição tornar-se ativo, para que ela seja realizada;
- ▶ Parâmetros do sensor – Os significados dos parâmetros de um sensor associado a uma transição sensorial são determinados pelo próprio animador, na definição do sensor;
- ▶ Delay de oscilação – O *delay* de oscilação de um sensor associado a uma máquina de estados sensorial corresponde ao número mínimo de iterações em que o estado do sensor deve ser mantido, mesmo que seu sinal de retorno indique a mudança de seu estado;
- ▶ Parâmetros dos sensores – Os significados dos parâmetros dos sensores associados a máquinas de estados sensoriais são também determinados pelo próprio animador, na definição do sensor.

4.4 Atribuição de Valores aos Parâmetros do Controlador

Modelada a estrutura, os valores dos parâmetros são especificados e podem ser avaliados e modificados a fim de obter melhores resultados. Os parâmetros disponíveis

na representação “máquina de estados sensoriais” podem ser modificados de diferentes maneiras, que inclusive podem ser combinadas. Qual dessas maneiras (ou combinação) é a mais adequada para modificar cada um desses parâmetros deve ser especificada pelo animador, com base no movimento desejado a ser gerado. Nesta seção, são discutidas as maneiras possíveis de se modificar os parâmetros, especificamente quando se utiliza a representação proposta. Como já mencionadas na Seção 3.3, elas são: modificação off-line manual, modificação off-line automática, modificação on-line interativa e modificação on-line automática.

4.4.1 Modificação Off-Line Manual dos Parâmetros

A modificação off-line manual é utilizada, principalmente, logo após a especificação dos parâmetros, com o intuito de diminuir o espaço de busca da otimização estocástica ou para fornecer valores iniciais razoáveis. Embora controladores para modelos mais simples e estáveis possam ser completamente gerados automaticamente, diminuir o espaço de busca e partir de bons valores iniciais é essencial para que controladores para modelos mais complexos consigam obter bons resultados automaticamente. Os ângulos desejados geralmente são os parâmetros que requerem mais esforço manual, pois são os que mais tendem a contribuir para o crescimento do espaço de busca, podendo gerar espaços de busca enormes.

4.4.2 Modificação Off-Line Automática dos Parâmetros

Os parâmetros de controladores para modelos mais complexos não devem ser todos gerados automaticamente, pois a grande quantidade de parâmetros disponíveis envolvidos acarreta um grande espaço de busca, o que dificulta a obtenção de bons resultados. Para limitar o tamanho do espaço de busca automática, alguns parâmetros são gerados manualmente e retirados do processo automático.

A escolha dos parâmetros que devem ser modificados automaticamente é uma tarefa difícil. Por isso, a representação proposta oferece uma estrutura intuitiva que permite que essa escolha seja feita pelo animador de forma hierárquica. Assim, em primeira instância escolhem-se as máquinas de estados que terão seus parâmetros modificados automaticamente; em seguida, definem-se quais os estados que participarão da modificação automática; e finalmente, escolhem-se quais os parâmetros desses estados que serão modificados automaticamente. Maiores detalhes sobre esse assunto são discutidos na Seção 5.4.1.

A representação proposta também permite que a otimização possa ser realizada em vários estágios. Em cada estágio, um conjunto, possivelmente diferente, de parâmetros é modificado. Quando o conjunto de todos os parâmetros a serem modificados automaticamente está determinado, as modificações são efetivadas e o controlador é avaliado, executando-se uma simulação dinâmica. A quantidade de modificações e avaliações é determinada pelo animador. Ao se determinar automaticamente os valores vencedores, a escolha do conjunto de parâmetros que são modificados automaticamente pode ser alterada e um outro estágio da otimização pode ser realizado, utilizando os valores determinados no estágio anterior como valores iniciais, se isso for conveniente. Assim, o animador pode interagir com o processo de otimização, determinando quais parâmetros são modificados automaticamente em cada estágio. Maiores detalhes sobre esse assunto são discutidos na Seção 5.4.3.

Como parte da estrutura, intervalos de valores admissíveis para os parâmetros devem ser especificados. Os parâmetros então podem ser modificados todos¹ ao mesmo tempo de uma forma completamente randômica, correspondendo a uma busca global, onde apenas os limites dos intervalos são obedecidos. Além dos intervalos de valores admissíveis, valores iniciais também podem ser considerados a fim de realizar uma busca mais local. Além disso, os parâmetros não são necessariamente modificados todos ao mesmo tempo antes de ser realizada uma avaliação. Diferentes variações de modificação automática são possíveis e dependem da maneira como os valores iniciais são considerados e da quantidade de parâmetros que são modificados a cada avaliação. Van de Panne e seus co-autores (1994) investigam várias dessas variações. Os tipos de modificação automática utilizados neste trabalho são explicados na Seção 5.4.1.

4.4.3 Modificação On-Line Interativa dos Parâmetros

Alguns parâmetros são escolhidos para serem modificados durante a simulação dinâmica (modificações on-line). As modificações on-line interativas são as realizadas a partir da interação com o animador. Interfaces apropriadas devem associar as ações do animador às modificações dos parâmetros do controlador. Nesta representação, apesar da falta de uma interface amigável disponível, o animador pode habilitar ou desabilitar as máquinas de estados, facilitando a composição de movimentos; pode também, enquanto acompanha a simulação dinâmica, perceber e especificar os tempos mais apropriados de algumas transições, modificá-los e armazená-los; e pode criar animações

¹ Todos dentre os parâmetros já escolhidos para serem gerados automaticamente.

em tempo real, especificando on-line os valores dos parâmetros. Utilizar a interação com o animador em controladores é interessante para produzir animações fisicamente realistas em tempo real nas aplicações em jogos (Van de Panne & Lee, 2003; Zhao & van de Panne, 2005).

4.4.4 Modificação On-Line Automática dos Parâmetros

As modificações on-line automáticas são as realizadas sem a intervenção do animador durante a simulação dinâmica. Como já discutido anteriormente, Laszlo (1996) modifica on-line automaticamente o PCG responsável por fazer um modelo caminhar, analisando o ciclo do seu movimento. Neste trabalho, os procedimentos associados à estrutura do controlador podem possuir instruções para modificar seus parâmetros automaticamente. As modificações on-line automáticas são realizadas a partir de procedimentos disparados. Essas modificações devem ser planejadas, durante a modelagem da estrutura, ao se definir os procedimentos. Procedimentos podem ser associados às máquinas de estados sensoriais, aos estados e às transições. Todos os procedimentos podem modificar os parâmetros do controlador, independentemente de suas associações.

Para modificar os valores dos parâmetros do controlador, os procedimentos podem utilizar informações do modelo e do ambiente, assim como informações do próprio controlador, incluindo informações dos seus sensores. O retorno dos sensores é um valor real, o que permite um maior aproveitamento dessa informação.

Um procedimento vinculado a um sensor associado a uma máquina de estados sensorial pode ser utilizado, por exemplo, para manter o equilíbrio de um modelo, modificando automaticamente os ângulos desejados das poses dessa máquina de estados, em função do retorno desse sensor, o qual percebe quando o modelo perde o equilíbrio. No exemplo simplificado da Figura 4.6, um sensor associado a uma máquina de estados sensorial retorna o ângulo corrente entre o tronco e o quadril. Então, um procedimento vinculado a esse sensor atualiza contínua e automaticamente o ângulo desejado (θ_d) para o atuador correspondente, atribuindo-lhe o valor do ângulo corrente (retornado pelo sensor) multiplicado por -1 . Assim, torques correspondentes responsáveis por manter o equilíbrio são gerados, de acordo com a Equação 2.1. A idéia desse procedimento é tentar manter o valor desse ângulo igual a zero, para manter a postura ereta e o modelo continuar em pé.

Esse exemplo apenas ilustra como modificações on-line automáticas podem ser realizadas. A estratégia descrita pode nem mesmo ser suficiente para manter o equilíbrio

do modelo, embora seja eficaz em alguns casos. Através desse exemplo, ilustra-se também que modificações on-line automáticas permitem ações de controle contínuas, assim como explicado na Seção 3.4 e ilustrado na Figura 3.12.

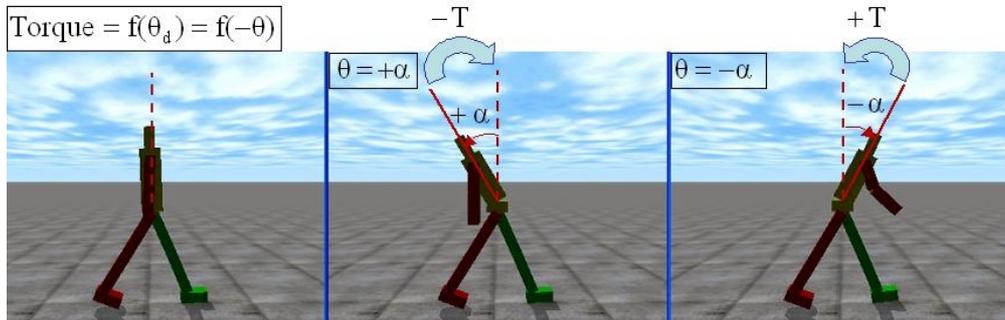


Figura 4.6: Manutenção do equilíbrio com modificações on-line automáticas.

4.5 Considerações sobre a Utilização da Representação “Máquinas de Estados Sensoriais”

4.5.1 Composição de Movimentos

Como já discutido no Capítulo 3, a composição de controladores é realizada com o objetivo de gerar seqüências de movimentos para obter animações mais complexas, pois controladores geralmente são projetados para um tipo específico de movimento. Por exemplo, um único PCG não é adequado para controlar um modelo tanto para caminhar quanto para realizar um salto sobre um obstáculo. Para isso, dois PCGs, um para cada tipo de movimento, são projetados e aplicados separadamente, de acordo com o movimento desejado. Faloutsos e seus co-autores (2001) apresentam um *framework* de composição que utiliza controladores como caixas pretas. Esses controladores fornecem pré-condições, que funcionam como sensores e são utilizadas por um controlador supervisor, como explicado na Seção 3.5. Esses controladores não podem ser modificados e só podem atuar um por vez no modelo.

Para que um controlador possa gerar vários tipos de movimento diferentes, uma estrutura mais geral, que permita a composição de movimentos, é necessária. Semelhante ao *framework* discutido, a estrutura da representação “máquinas de estados sensoriais” permite a composição de movimentos. A composição é realizada naturalmente, pois um controlador que possui essa representação pode ser analisado como um conjunto de controladores – as máquinas de estados – organizados em uma única estrutura. Entretanto, ao contrário do *framework* discutido, essas máquinas de estados não são caixas pretas e podem ser modificadas e melhoradas de acordo com o

movimento desejado. A possibilidade de modificação das máquinas de estados é essencial para que elas possam ser utilizadas em conjunto, atuando ao mesmo tempo.

4.5.2 Ativação de Máquinas de Estados Sensoriais

Comparando à representação estímulo-resposta (Seção 3.2.1), as ações das máquinas de estados sensoriais correspondem às respostas, enquanto seus sensores associados correspondem aos estímulos. Entretanto, uma diferença significativa é que mais de uma máquina de estados sensorial pode estar ativa ao mesmo tempo, enquanto que apenas uma resposta pode ser escolhida em cada iteração da simulação física. Além disso, essas máquinas podem controlar o modelo de uma maneira mais geral e expressiva do que as respostas produzidas na representação estímulo-resposta.

A Figura 4.7 exemplifica a utilização da representação “máquinas de estados sensoriais”, mostrando uma animação onde uma máquina de estados sensorial é ativada. O sensor associado a essa máquina calcula a distância horizontal do quadril em relação ao pé que está posicionado mais à frente, e é ativado quando essa distância indica que o quadril está à frente desse pé e ultrapassou um valor de tolerância. Inicialmente, máquinas de estados projetadas para um caminhar atuam no modelo. Durante a caminhada, uma força externa é aplicada na parte de trás da cabeça do modelo a fim de causar a perda de equilíbrio. O modelo tende a cair para frente e, no instante em que o sensor é ativado (distância do quadril ao pé da frente supera a tolerância), a máquina de estados sensorial associada começa a atuar no modelo para que os braços sejam erguidos e protejam a cabeça durante a queda. Se essa máquina não fosse ativada, o modelo continuaria apenas tentando andar e bater a cabeça no chão.

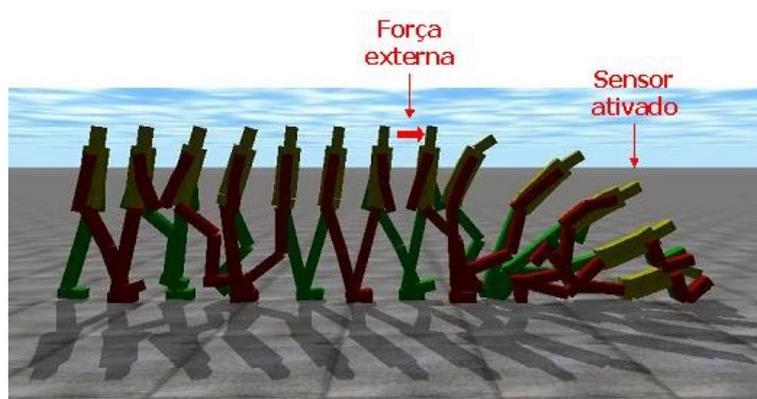


Figura 4.7: Animação de um humanóide ativando uma máquina de estados sensorial para proteger sua cabeça com os braços durante a queda.

Esse exemplo simples ilustra como os sensores influenciam a ativação das máquinas de estados sensoriais. Ao invés da simples proteção da cabeça, máquinas

produzindo diferentes ações de controle poderiam ter sido associadas ao sensor utilizado, assim como outros sensores poderiam ter sido utilizados.

Um cuidado importante no projeto das máquinas de estados é considerar as ações das demais máquinas de estados que atuam simultaneamente. É evidente que uma máquina de estados projetada para atuar sozinha não tem o mesmo resultado quando atua simultaneamente com outra máquina de estados. No exemplo mostrado, as máquinas responsáveis pelo caminhar poderiam ser definidas como padrões (estão ativas quando nenhum sensor está ativo), portanto, não atuariam juntamente com a máquina de estados sensorial. Entretanto, por simples questão de preferência, elas são definidas como constantes (permanecem ativas durante todo o tempo) e, assim, agem conjuntamente com a máquina de estados sensorial, resultando em um movimento combinado. O planejamento da atuação conjunta das máquinas de estados não torna seu projeto tão mais difícil.

Um sensor atrelado a uma máquina de estados também pode ser associado a um procedimento. Assim, quando tal sensor torna-se ativo, não só as máquinas de estados sensoriais às quais está atrelado tornam-se ativas, como também o procedimento associado a ele (se houver) é executado. Os procedimentos associados a sensores podem ser executados nas seguintes situações: apenas no instante em que o sensor torna-se ativo; ou em toda iteração da simulação dinâmica em que o sensor estiver ativo. A utilização de procedimentos na estrutura proposta é melhor discutida na Seção 4.5.3.

Um problema que pode ocorrer como consequência da utilização de sensores é a rápida oscilação de seus valores de retorno. Esse problema precisa ser tratado em algumas situações. Apesar dos retornos dos sensores que utilizam a estrutura proposta neste trabalho serem valores reais, há apenas dois estados possíveis para os sensores: ativo e inativo. Isso pode acarretar rápida oscilação indesejada dos estados dos sensores que é tratada através de mecanismos de retardo adicionados aos sensores associados às máquinas de estados sensoriais. Esses mecanismos de retardo definem o número mínimo de iterações em que o estado de um sensor deve ser mantido, mesmo que o sinal de retorno do sensor indique a mudança de seu estado.

4.5.3 Disparando Procedimentos

Procedimentos podem ser implementados e executados facilmente durante a simulação dinâmica, através da interação com o animador. Entretanto, a estrutura da representação “máquinas de estados sensoriais” permite que procedimentos com instruções gerais, de

acordo com o movimento desejado, sejam definidos e adequadamente associados a ela pelo animador durante seu projeto, para serem executados automaticamente pelo controlador durante a simulação dinâmica. Neste trabalho, disparar um procedimento significa executá-lo automaticamente durante a simulação dinâmica. Além dos sensores associados às máquinas de estados sensoriais, os estados e as transições das máquinas de estados também podem ser associados a esses procedimentos. Os estados disparam seus procedimentos associados em toda iteração da simulação dinâmica em que eles estão ativos. As transições disparam seus procedimentos associados apenas no instante em que realizam a troca do estado ativo.

A associação de procedimentos a estados é a idéia básica utilizada na representação de controladores por máquinas de estados, porém PCGs geralmente limitam as ações dos seus estados a tentar atingir uma determinada pose. Portanto, o diferencial da estrutura da representação “máquinas de estados sensoriais”, quanto à maneira e às opções de utilização de procedimentos, é permitir ao animador fornecer instruções mais gerais para o controlador e permitir ao controlador executar essas instruções nos momentos mais adequados.

Poder disparar um procedimento é muito útil para um controlador. Procedimentos podem alterar tanto as propriedades do modelo quanto as propriedades do ambiente. O ambiente pode ser alterado, por exemplo, aplicando-se forças externas ou modificando-se o coeficiente de atrito do chão. Procedimentos podem também alterar propriedades do controlador, tais como seus parâmetros (Seção 4.4.4). Além da vantagem de realizar essas alterações automaticamente, disparar procedimentos é essencial ao se avaliar um controlador automaticamente, pois os procedimentos precisam ser executados exatamente da mesma maneira em todas as simulações avaliadas, o que não é possível através da interação com o animador.

Através de uma animação onde um humanóide arremessa uma bola, a Figura 4.8 exemplifica a vantagem de se poder disparar um procedimento. O controlador responsável por essa animação é constituído de duas máquinas de estados constantes. A máquina de estados responsável por arremessar a bola atua apenas nos membros superiores¹ e foi gerada automaticamente. Um procedimento implementado para que o humanóide solte a bola foi associado à sua penúltima transição. O modelo inicialmente

¹ Esses membros superiores correspondem aos atuadores 11, 12, 13 e 14 da Figura 4.2. Esta máquina de estados foi restringida apenas aos membros superiores para simplificar a sua geração, considerando que seu objetivo principal é exemplificar um procedimento sendo disparado.

segura a bola através da restrição imposta por uma junta ligando sua mão direita à bola. No instante em que a penúltima transição realiza a troca do estado ativo, o procedimento é disparado, destruindo essa junta. A bola então é arremessada. A outra máquina de estados atua no resto do corpo e possui apenas um estado, que tenta manter o corpo numa postura ereta.

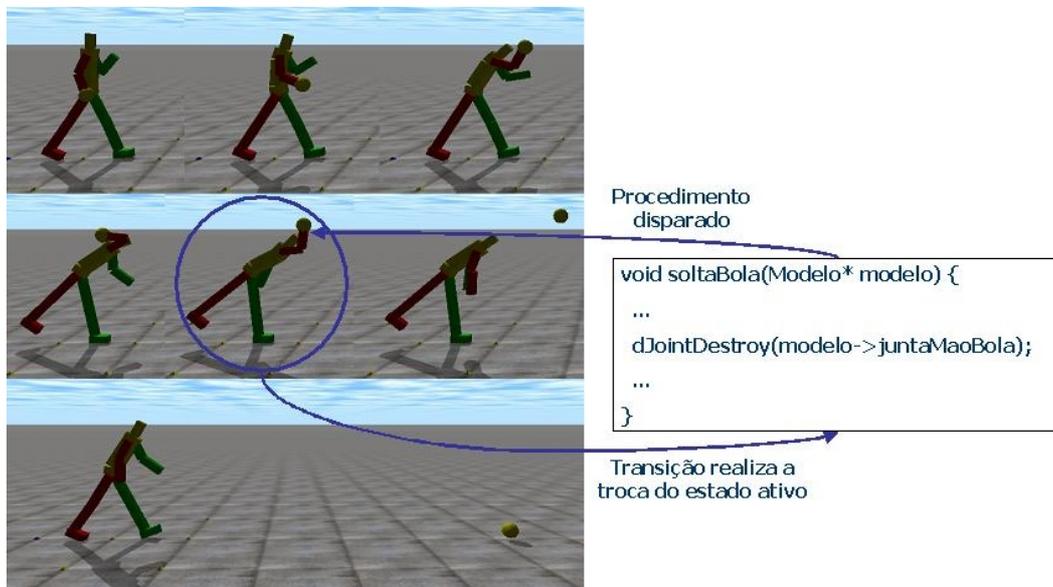


Figura 4.8: Controlador disparando um procedimento associado a uma transição.

4.6 Considerações Finais

Neste capítulo, foi proposta uma representação de controladores, que utiliza sensores e tem uma estrutura mais geral e intuitiva, disponibilizando vários tipos de parâmetros para o animador. Essa estrutura permite que, de acordo com a retroalimentação proveniente de seus sensores, várias máquinas de estados atuem simultaneamente no mesmo modelo, ou em subconjuntos de seus atuadores. Também permite que, durante seu projeto, procedimentos com instruções gerais, de acordo com o movimento desejado, sejam definidos e adequadamente associados a ela pelo animador, para serem executados automaticamente pelo controlador durante a simulação dinâmica. Além das inovações propostas na representação de controladores, neste capítulo também foi proposta uma estrutura mais geral e elaborada para os sensores utilizados, permitindo que eles sejam melhor explorados na geração automática dos controladores.

Esta representação aborda a dificuldade da escolha dos parâmetros a serem modificados automaticamente, permitindo que essa escolha possa ser realizada hierarquicamente. Também é permitido que a otimização estocástica possa ser realizada em vários estágios, onde, em cada um deles, um conjunto possivelmente diferente de parâmetros é modificado. Além disso, diferentes variações de modificação automática,

em cada estágio, podem ser utilizadas. A representação proposta facilita o projeto dos controladores, a composição de movimentos e a geração automática de controladores.

Capítulo 5

Implementação

5.1 Introdução

Neste capítulo, são discutidos os aspectos de implementação relacionados ao problema de animação de sistemas dinâmicos atuados através de controladores, em especial, os aspectos de implementação: da simulação dinâmica, da representação de controladores “máquinas de estados sensoriais” proposta e da otimização do controlador.

5.2 Simulação Dinâmica

Neste trabalho, a simulação dinâmica é realizada pelo motor dinâmico aberto, Open Dynamics Engine (ODE) (Smith, 2000). O ODE é utilizado para calcular e integrar as equações de movimento. Essas equações são derivadas a partir de um modelo de multiplicador de Lagrange baseado em velocidades proposto por Trinkle e Stewart (1996) e por Anitescu e Potra (1997). Uma grande vantagem do ODE é que as equações de movimento podem ser recompiladas durante a simulação. Assim, corpos e juntas podem ser adicionados, removidos ou alterados durante a simulação dinâmica. Essa vantagem torna o ODE mais flexível do que outros simuladores dinâmicos em que os modelos só podem ser compilados antes da simulação dinâmica.

O ODE utiliza um integrador semi-implícito de primeira ordem, onde algumas forças (forças de restrição das juntas) são calculadas como se um integrador implícito estivesse sendo utilizado e outras forças (forças externas aplicadas pelo ambiente e torques aplicados nas juntas) são calculadas como se o integrador fosse explícito. Imprecisão numérica em integradores implícitos geralmente é manifestada como uma redução de energia, ou seja, o erro causado pelo integrador tende a “amortecer” o sistema. Imprecisão numérica em integradores explícitos geralmente causa um efeito oposto, aumentando a energia do sistema. Conseqüentemente, integradores explícitos possuem uma maior tendência a instabilidades mais perceptíveis, podendo levar o sistema a se afastar ilimitadamente da solução. Como as forças aplicadas pelos controladores são tratadas pelo ODE explicitamente, o sistema pode tornar-se instável quando forças muito grandes são aplicadas. Como já mencionado, a representação “máquinas de estados sensoriais” limita os torques máximos aplicados nas juntas a fim de evitar essas instabilidades numéricas. Isso é razoável já que na realidade as forças

musculares são limitadas. Entretanto, apesar da possibilidade de instabilidades, o integrador utilizado pelo ODE é altamente estável, comparado a outros simuladores, e erros numéricos geralmente não crescem fora de controle.

O ODE também fornece um detector de colisões, que é utilizado para determinar quais corpos estão em contato em cada iteração. Os corpos dos modelos utilizados possuem formatos simples, tais como esferas e caixas. Colisões entre corpos pertencentes a um mesmo modelo são ignoradas a fim de facilitar o controle e evitar instabilidades na simulação. O detector de colisões fornecido pelo ODE é baseado no solucionador LCP de Dantzig descrito por Baraff, embora o ODE implemente uma aproximação mais rápida do modelo de fricção de Coulomb. A complexidade média deste detector de colisões para testar a interseção entre n objetos é $O(n)$ (Smith, 2000).

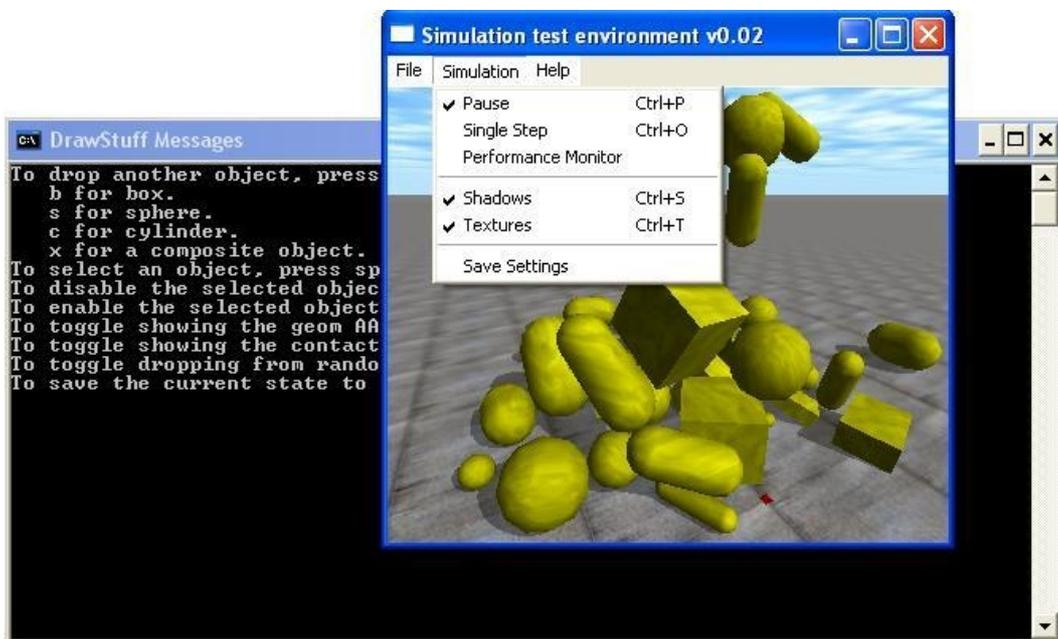


Figura 5.1: GUI do ODE compilada no Dev-C++ 4.9.8.0 no Windows XP.

O sistema de animação implementado foi programado em C++, utilizando uma interface gráfica, também disponibilizada pelo ODE (Figura 5.1), que utiliza OpenGL para renderizar os quadros da animação. A simulação dinâmica é realizada em tempo real em um PC com processador de 2.2 GHz. Como o custo computacional com o uso de controladores é determinado principalmente pelo custo da própria simulação dinâmica, as animações também são obtidas em tempo real, considerando que os controladores já tenham sido gerados. As equações de movimento são integradas utilizando um incremento de tempo fixo equivalente a 0.05s a cada iteração da simulação dinâmica.

5.3 Máquinas de Estados Sensoriais

Nesta seção, os aspectos de implementação relacionados à representação de controladores “máquinas de estados sensoriais” proposta são discutidos. Primeiramente, apresenta-se uma visão geral do seu processo de controle e, em seguida, complementam-se, com detalhes de implementação, alguns assuntos abordados anteriormente.

5.3.1 Visão Geral do Processo de Controle

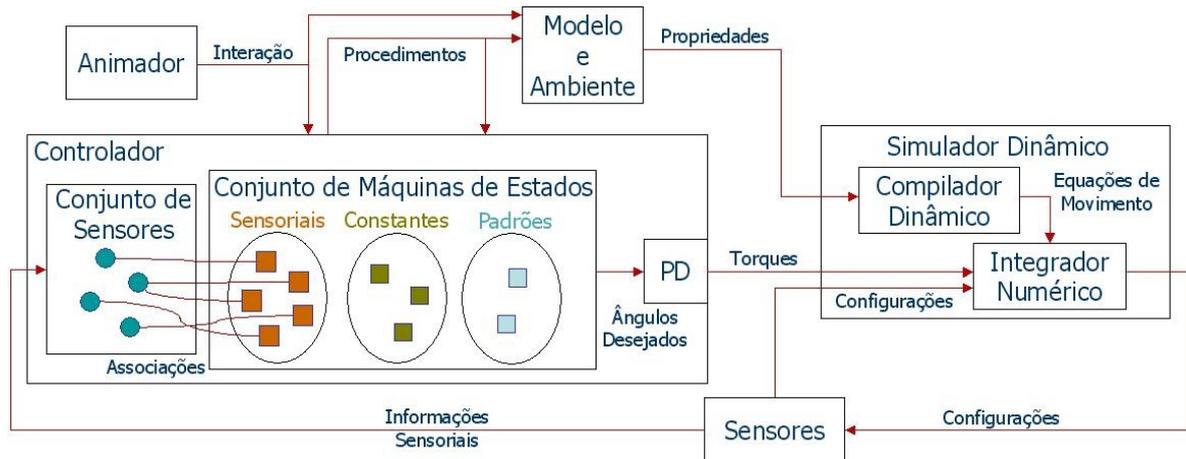


Figura 5.2: Visão geral do processo de controle, utilizando a representação “máquinas de estados sensoriais”.

Dados o modelo a ser animado e o ambiente em que a animação ocorre, a estrutura do controlador é definida de acordo com o movimento desejado, e especificada a partir de seu conjunto de máquinas de estados e de seu conjunto de sensores. A Figura 4.1, que é repetida aqui como Figura 5.2, apresenta uma visão geral do processo de controle, o qual é sintetizado da seguinte maneira:

- ▶ O compilador dinâmico fornece as equações de movimento ao integrador numérico a partir das especificações do modelo a ser animado e do ambiente de animação;
- ▶ O integrador numérico, por sua vez, utiliza essas equações e os valores dos torques gerados pelos atuadores para, a partir da configuração atual, obter a próxima configuração;
- ▶ A nova configuração é disponibilizada para os sensores e reutilizada pelo integrador numérico;
- ▶ Os sensores enviam sinais ao controlador que calcula novos torques e os envia ao integrador numérico;

- O integrador numérico processa mais um passo da simulação e uma outra configuração é obtida. E o processo se repete.

Em algumas dessas iterações, procedimentos podem ser disparados e interações com o animador podem ocorrer. Isso pode provocar alterações nas propriedades tanto do próprio controlador quanto do modelo e do ambiente.

5.3.2 Executando as Máquinas de Estados

A cada iteração, o controlador utiliza as informações fornecidas pelos sensores para determinar quais máquinas de estados estão ativas, de acordo com seus tipos (as máquinas de estados são dispostas em três vetores, um para cada tipo). Para isso, esses vetores são percorridos e todas as máquinas de estados são verificadas de acordo com o pseudo-algoritmo mostrado na Figura 5.3. Cada máquina de estados ativa e habilitada é executada. Assim, os valores dos ângulos desejados associados ao seu estado ativo são determinados e uma verificação é feita para constatar se sua próxima transição deve ser realizada. Esses valores são enviados para os controladores do tipo PD, que obtêm os torques correspondentes e calculam os torques resultantes em cada atuador através de médias ponderadas, considerando o peso da máquina de estados correspondente a cada torque. Esses torques resultantes são então utilizados pelo integrador numérico.

```

Para cada máquina de estados constante MEc
  Se (MEc está habilitada)
    Executa MEc
padrao ← verdadeiro
Para cada máquina de estados sensorial MEs
  Se (o sensor associado à MEs está ativo)
    Se (MEs está habilitada)
      Executa MEs
    padrao ← falso
Se (padrao)
  Para cada máquina de estados padrão MEp
    Se (MEp está habilitada)
      Executa MEp

```

Figura 5.3: Pseudo-algoritmo – Execução das máquinas de estados ativas e habilitadas.

5.3.3 Médias Ponderadas nos Atuadores

Um atuador pode ser afetado por mais de uma máquina de estados ao mesmo tempo, isto é, um controlador do tipo PD, que é associado a um atuador, pode receber mais de um valor de ângulo desejado numa mesma iteração da simulação dinâmica, calculando

então os torques correspondentes a todos esses valores recebidos e computando a média ponderada desses torques, para obter o torque resultante a ser aplicado pelo atuador. Os pesos utilizados nessa média são os respectivos pesos associados às máquinas de estados que enviam os sinais.

Um controlador do tipo PD recebe os valores de ângulos desejados (sinais) separadamente de cada máquina de estados, de acordo como elas vão sendo analisadas. Portanto, para cada controlador do tipo PD, são armazenados dois valores: T_{parcial} , que é o torque parcial referente à média ponderada dos torques correspondentes aos sinais já recebidos; e S_{pesos} , que é o somatório parcial dos pesos das respectivas máquinas de estados que já enviaram os sinais.

A cada iteração da simulação dinâmica, para cada sinal recebido por um controlador do tipo PD, o torque individual, correspondente a esse sinal, é calculado e limitado. Essa limitação corresponde ao torque máximo pré-definido para o atuador. O valor do torque a ser aplicado pelo atuador é atualizado, de acordo com a equação,

$$T_{\text{parcial}} = \frac{S_{\text{pesos}} * T_{\text{parcial}} + T_{\text{individual}} * P_{\text{ME}}}{S_{\text{pesos}} + P_{\text{ME}}}, \quad (5.1)$$

onde $T_{\text{individual}}$ é o torque individual correspondente ao sinal recebido já limitado e P_{ME} é o peso da máquina de estados que enviou o sinal. Além disso, o somatório parcial dos pesos também é atualizado, de acordo com a equação,

$$S_{\text{pesos}} = S_{\text{pesos}} + P_{\text{ME}}. \quad (5.2)$$

Ao final de cada iteração, esses valores T_{parcial} e S_{pesos} são zerados.

Ao se definir o valor de um ângulo desejado de uma pose, pode-se determinar que o atuador correspondente não aplique nenhum torque para que ele não tenha efeito; ou pode-se também determinar que esse atuador aplique somente torques amortecedores que procurem apenas manter o ângulo corrente. Entretanto, essa determinação é feita levando-se em conta apenas uma pose específica. Assim, outras poses, inclusive da mesma máquina de estados, podem continuar influenciando o comportamento do atuador normalmente. Desse modo, por exemplo, para que um atuador não tenha efeito durante uma determinada iteração, todos os sinais recebidos por ele, vindos de diferentes poses, devem indicar que ele não aplique nenhum torque. Outra opção é já definir um valor de ângulo desejado atrelado ao próprio atuador, fazendo parte da sua modelagem, para que o valor do torque a ser aplicado pelo atuador não dependa dos

valores de ângulos desejados definidos nas poses das máquinas de estados. Assim, o atuador passa a manter constante seu valor de ângulo desejado durante toda a simulação.

5.3.4 Interpolação entre Poses

Nas transições, interpolação linear dos ângulos desejados das poses é utilizada a fim de produzir transições mais suaves entre a pose associada ao estado atualmente ativo e a pose associada ao próximo estado ativo. Cada transição possui um *delay*, correspondendo ao tempo de duração da interpolação entre os estados que ela conecta.

No instante em que um estado se torna ativo, a pose desejada enviada aos controladores do tipo PD depende da pose associada a esse estado ativo e da pose associada ao estado anteriormente ativo, até que o *delay* de interpolação da transição que conecta esses estados tenha acabado. Somente quando esse *delay* acaba é que a pose desejada enviada aos controladores do tipo PD é exatamente a pose associada a esse estado ativo; e que a próxima transição tem permissão para ser executada.

Enquanto o *delay* da transição não acaba, a pose desejada enviada aos controladores do tipo PD é obtida através da combinação convexa das poses associadas aos estados conectados por essa transição, de acordo com a equação,

$$P_{\text{enviada}} = P_{\text{anterior}} * \left(1 - \frac{N_I}{D_I}\right) + P_{\text{ativa}} * \left(\frac{N_I}{D_I}\right), \quad (5.3)$$

onde P_{enviada} é o vetor correspondente à pose enviada aos controladores do tipo PD, P_{anterior} é o vetor correspondente à pose anteriormente ativa, P_{ativa} é o vetor correspondente à pose ativa, D_I é o *delay* de interpolação e N_I é o número de iterações desde que a pose tornou-se ativa.

Assim que uma máquina de estados começa a atuar no modelo, não há estado anteriormente ativo e um tratamento é realizado para que inicialmente a pose associada ao seu primeiro estado não seja interpolada com a pose associada ao seu último estado. Ao se reabilitar uma máquina de estados desabilitada ou ao se reativar uma máquina de estados inativa, o estado inicialmente ativo é sempre o seu primeiro estado. Para se obter uma interpolação adequada, deve-se ter o cuidado de interpolar apenas poses que correspondem às mesmas juntas dispostas numa mesma ordem, já que uma pose pode se referir a qualquer subconjunto de atuadores do modelo.

5.4 Modificação Automática

As modificações automáticas dos valores dos parâmetros dos controladores são realizadas em um módulo separado constituído por três partes: o modificador randômico, responsável por modificar diretamente os valores dos parâmetros; o avaliador, responsável por avaliar as modificações realizadas, segundo um critério de otimização; e o otimizador, responsável por realizar a busca, decidindo quais valores atribuídos aos parâmetros do controlador são os mais adequados.

5.4.1 Modificador Randômico

O modificador randômico é responsável por modificar diretamente os valores dos parâmetros do controlador. A representação “máquinas de estados sensoriais” permite que a escolha dos parâmetros que serão modificados automaticamente, em cada estágio da otimização, seja mais intuitiva. Um atributo adicionado a algumas classes componentes da representação do controlador determina se os objetos poderão ter seus parâmetros modificados automaticamente. Os parâmetros correspondentes a essas classes (máquinas de estados, estados, transições e sensores) podem ser vistos na Figura 4.6. Por exemplo, um estado contendo esse atributo com valor falso não permite que os ângulos desejados de sua pose associada sejam modificados pelo modificador randômico, ou seja, os parâmetros desse estado não fazem parte do conjunto de parâmetros escolhidos para serem modificados automaticamente. No entanto, esse atributo não possui influência sobre o movimento gerado pelo controlador e, por isso, não é considerado um parâmetro do controlador, sendo útil apenas para permitir uma escolha mais organizada e hierarquizada dos parâmetros que são modificados automaticamente. Essa escolha é hierarquizada no sentido de, por exemplo, um estado só poder ser modificado se a máquina de estados que o possuir também puder ser modificada. Em relação às máquinas de estados, independentemente dos valores desses atributos, é possível permitir, em cada estágio da otimização, que possam ser modificados: apenas os parâmetros de seus estados, apenas os parâmetros de suas transições, ou ambos.

Tal como no trabalho de van de Panne e seus co-autores (1994), a representação “máquinas de estados sensoriais” também fornece diversas variações de modificação automática, dentre os parâmetros já escolhidos para serem modificados automaticamente. O modificador randômico pode modificar todos os parâmetros ao mesmo tempo, obedecendo apenas os limites dos seus intervalos, permitindo ao

otimizador uma busca mais global; ou modificar não necessariamente todos os parâmetros ao mesmo tempo antes de ser realizada uma avaliação, podendo considerar valores iniciais e permitindo ao otimizador uma busca mais local. Em ambos os casos, o modificador randômico não utiliza informações de gradiente para realizar as modificações.

Em relação às modificações que permitem ao otimizador uma busca mais local, pode-se analisar a quantidade de parâmetros que são modificados a cada avaliação e a maneira como os valores iniciais e os intervalos de valores admissíveis dos parâmetros são considerados. Em um determinado estágio da otimização, pode ser que apenas conjuntos específicos de parâmetros sejam modificados (por exemplo, todos os ângulos desejados das poses associadas a estados específicos de diferentes máquinas de estados), ou ainda, que apenas um parâmetro seja modificado a cada avaliação. Quanto aos intervalos de valores admissíveis dos parâmetros, em um determinado estágio da otimização, os intervalos podem ser: mantidos fixos durante todo o estágio; reduzidos parcialmente; ou reduzidos iterativamente após cada avaliação. Nos casos em que os intervalos são reduzidos, a cada avaliação, apenas podem ser sorteados valores que estejam próximos aos valores atuais dos parâmetros. Quando os intervalos são reduzidos parcialmente, os seus tamanhos são mantidos fixos e eles apenas sofrem translações, de acordo com os valores atuais dos parâmetros. Já quando os intervalos são reduzidos iterativamente, além dessa translação, os seus tamanhos vão diminuindo após cada avaliação de forma que, depois da última avaliação do estágio, os intervalos consistem apenas dos próprios valores atuais dos parâmetros. A cada modificação, é realizada uma escolha randômica (sorteio) dos valores dos parâmetros dentro de seus respectivos intervalos.

A escolha de como o modificador randômico deve funcionar em cada estágio da otimização, considerando todas essas variações apresentadas, é realizada pelo animador.

5.4.2 Avaliador

O avaliador é responsável por avaliar as modificações realizadas, segundo um critério de otimização. Após cada modificação no conjunto de parâmetros escolhidos para serem modificados automaticamente, uma avaliação é realizada sobre as ações do controlador. Para isso, uma simulação dinâmica do modelo, sendo controlado pelo controlador modificado, é executada e analisada durante um número finito e previsto de iterações. Todas as simulações dinâmicas são realizadas da mesma maneira, diferindo apenas nos

valores dos parâmetros do controlador utilizado. Um critério de otimização é definido de acordo com o movimento desejado, determinando como as simulações devem ser avaliadas, e apenas um valor final (*fitness*) é retornado de cada avaliação a fim de comparar qual conjunto de valores atribuídos ao conjunto de parâmetros é o mais adequado.

5.4.3 Otimizador

O otimizador é responsável por buscar o melhor conjunto possível de valores dos parâmetros, analisando todas as modificações e avaliações de uma maneira global. O método utilizado pelo otimizador, de acordo com os valores de *fitness* retornados pelo avaliador, segundo o critério de otimização definido, guia o processo de busca, determinando a partir de quais pontos do espaço paramétrico devem ser realizadas as modificações, no intuito de construir um caminho ideal do ponto inicial ao ponto ótimo. O método utilizado neste trabalho consiste em rejeitar as modificações que resultam em pontos cujos valores de *fitness* sejam piores, sempre atualizando os parâmetros com valores que resultam em valores de *fitness* melhores do que o atual. Esse método, comparado ao método têmpera simulada, implica numa busca mais local. Entretanto, é permitida uma busca global inicialmente, em que todos os parâmetros são modificados a cada avaliação, a fim de obter melhores valores iniciais.

Neste trabalho, a otimização pode ser realizada em vários estágios. Em cada estágio, um conjunto possivelmente diferente de parâmetros é modificado. Para cada estágio deve-se determinar: o conjunto de parâmetros que será modificado, o número de modificações e avaliações realizadas neste conjunto¹, a duração destas avaliações e como o modificador randômico deve funcionar. Depois de obtidos automaticamente os valores vencedores dos parâmetros em um estágio, um outro estágio pode ser determinado diferentemente e executado, utilizando esses valores vencedores como valores iniciais. Assim, o animador pode interagir com o processo de otimização, modificando as características da otimização em cada estágio.

5.5 Considerações Finais

Este capítulo abordou tópicos necessários à implementação e à avaliação da representação “máquinas de estados sensoriais” proposta. Aspectos de implementação

¹ Na busca, não há garantias de que haja uma convergência para o ponto ótimo do espaço paramétrico. Por essa razão, ao invés de um teste de convergência, apenas um limite no número de iterações é considerado.

do sistema de animação utilizado para avaliar a representação proposta foram apresentados com um maior nível de detalhes. O capítulo seguinte apresenta os resultados dos testes realizados.

Capítulo 6

Resultados

6.1 Introdução

Neste capítulo, são apresentados os resultados dos testes de animação realizados em modelos de humanóide 2D, para avaliar a representação “máquinas de estados sensoriais” proposta. Os modelos utilizados nos testes são apresentados na Seção 6.2.

6.2 Modelos

As propriedades geométricas e físicas dos três modelos de humanóide utilizados são baseadas no modelo apresentado em (Laszlo, 1996), mas alguns ajustes foram realizados empiricamente, considerando outros parâmetros utilizados na simulação dinâmica. O modelo apresentado na Figura 6.1 foi utilizado para o teste do movimento de caminhar sobre uma superfície plana. Os modelos utilizados nos outros testes apresentam pequenas variações em relação a esse modelo original.

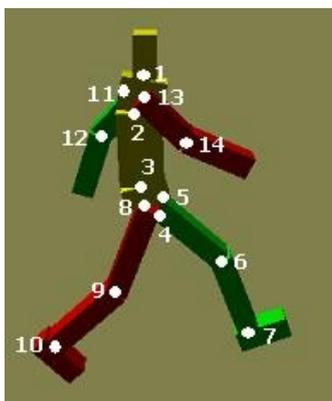


Figura 6.1: Modelo de humanóide utilizado para caminhar em uma superfície plana.

Em todos os modelos, os membros do lado esquerdo são identificados com a cor verde e os do lado direito, com a cor vermelha, a fim de facilitar a visualização da animação. O modelo da Figura 6.1 possui 14 corpos rígidos, 13 graus de liberdade internos e 14 atuadores. Dos 14 atuadores, 13 estão associados aos graus de liberdade internos e 1 está associado ao ângulo relativo entre as duas coxas. Este último atuador foi adicionado para facilitar o controle do ângulo relativo entre as duas coxas, comparados com os atuadores localizados nas juntas que unem coxa e quadril, pois não aplica torques diretamente no quadril. Como o modelo de estrutura articulada é bidimensional, os graus de liberdade nas juntas correspondem aos ângulos de rotação

em torno do eixo perpendicular ao plano do modelo. Na Figura 6.1, os pontos numerados correspondem à ordem estabelecida para os atuadores do modelo.

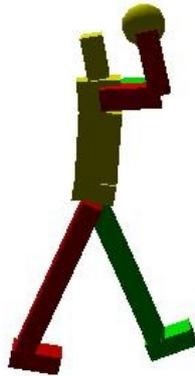


Figura 6.2: Modelo de humanóide utilizado para arremessar uma bola ao cesto.

A Figura 6.2 mostra o modelo de humanóide utilizado para arremessar uma bola ao cesto. As únicas diferenças em relação ao modelo original é que esse modelo possui um corpo rígido a mais, a bola, e que os dois braços do modelo são mantidos unidos, devido a duas juntas fixas que unem as duas “mãos” à bola. Portanto, esse modelo possui dois graus de liberdade a menos, correspondentes aos graus de liberdade de um braço. Ele possui 15 corpos rígidos e 11 graus de liberdade internos. Ao soltar a bola, o modelo torna-se igual ao original. As colisões ocorridas entre a bola e o modelo não são tratadas.

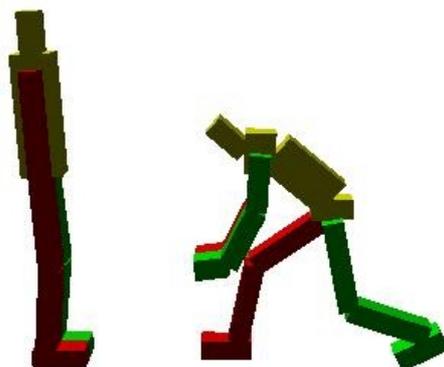


Figura 6.3: Modelos de humanóide utilizados para realizar um salto mortal para trás.

A Figura 6.3 mostra os modelos de humanóide utilizados para realizar um salto mortal para trás. O modelo de humanóide da esquerda é igual ao original, enquanto que o da direita difere do modelo original na orientação (sua frente está voltada para a esquerda) e por possuir duas mãos (dois corpos rígidos a mais) e, conseqüentemente, pulsos (dois graus de liberdade a mais). Ele possui 16 corpos rígidos, 15 graus de

liberdade internos e 16 atuadores, um para cada grau de liberdade interno e mais um que influencia diretamente no ângulo relativo entre as duas coxas, como no original.



Figura 6.4: Configuração inicial do modelo de humanóide original.

As tabelas 6.1 e 6.2 mostram propriedades do modelo de humanóide original. As localizações são referentes à configuração inicial do modelo (Figura 6.4). Os membros superiores e inferiores aparecem apenas uma vez nas tabelas e correspondem tanto aos membros direitos quanto aos membros esquerdos.

Tabela 6.1: Propriedades dos corpos rígidos do modelo de humanóide original.

Corpo Rígido	Massa (10^{-3} kg)	Localização do centro geométrico (m)			Dimensões (m)	
		x	y	z	x	y
Cabeça	5.89	0.000	1.800	0.000	0.100	0.200
Tórax	9.75	0.000	1.625	0.000	0.200	0.150
Abdômen	19.51	0.000	1.375	0.000	0.200	0.350
Quadril	16.61	0.000	1.150	0.000	0.200	0.100
Coxa	8.35	0.000	0.875	0.000	0.100	0.450
Perna	4.16	0.000	0.450	0.000	0.100	0.400
Pé	1.20	0.060	0.200	0.000	0.250	0.100
Braço	2.79	0.000	1.487	0.000	0.100	0.275
Antebraço	1.76	0.000	1.200	0.000	0.100	0.300

Tabela 6.2: Propriedades das juntas do modelo de humanóide original.

Junta	Localização (m)			k_p (10^{-3} N.m)	k_d (10^{-3} N.m.s)
	x	y	z		
Pescoço	0.000	1.700	0.000	2.25	0.075
Coluna	0.000	1.550	0.000	22.50	0.750
Cintura	0.000	1.200	0.000	22.50	0.750
Quadril	0.000	1.100	0.000	13.50	0.450
Joelho	0.000	0.650	0.000	13.50	0.450
Tornozelo	0.000	0.250	0.000	6.75	0.225
Ombro	0.000	1.625	0.000	11.25	0.375
Cotovelo	0.000	1.350	0.000	6.75	0.225

O ODE alerta que aplicar torques diretamente nas juntas do modelo pode levar a sérios problemas de estabilidade se não aplicados cuidadosamente, e oferece outra alternativa para o tratamento do amortecimento nos atuadores. Esse tratamento exige que uma velocidade desejada e um valor máximo de torque sejam especificados para cada atuador. Os corpos então são levados às velocidades desejadas em uma iteração da simulação dinâmica, considerando que os torques necessários não sejam superiores aos máximos permitidos. Esse amortecimento é implementado através de restrições. Portanto, esse procedimento foi adotado em todos os testes realizados neste capítulo. Para isso, os valores das constantes k_d (já que não são utilizadas nessa abordagem) foram definidos como zero, e utilizou-se velocidade angular desejada igual a 0 (zero) em todos os atuadores dos modelos. Os valores máximos de torque utilizados, especificados para cada atuador, foram proporcionais à maioria dos valores das constantes k_d mostrados na Tabela 6.2.

6.3 Descrição dos Testes

Nesta seção, são apresentados os resultados dos testes de animação realizados, utilizando os modelos de humanóide 2D apresentados na seção anterior. Os testes discutidos consistem em fazer o humanóide: caminhar em uma superfície plana, arremessar uma bola ao cesto, e executar um salto mortal para trás.

6.3.1 Humanóide Caminhando em uma Superfície Plana

Controlar alguns movimentos aparentemente simples, tais como o caminhar, é surpreendentemente difícil. A instabilidade natural desse movimento implica na necessidade de criar estratégias para manter o equilíbrio do modelo. Neste exemplo, a representação “máquinas de estados sensoriais” analisa a possível perda do equilíbrio do humanóide através de seus sensores, de uma maneira intuitiva. Além disso, a estrutura de sensores proposta permite que o controlador seja melhor projetado automaticamente. A estrutura do controlador gerado para o movimento de caminhar consiste de duas máquinas de estados constantes: a máquina de estados que controla o movimento do tronco e das pernas, e a máquina que controla o movimento dos braços.

A máquina de estados responsável pelo movimento do tronco e das pernas (Figura 6.5) possui seis estados, cujas poses associadas foram todas projetadas manualmente e baseadas em (Laszlo, 1996). Suas transições são todas sensoriais e alguns de seus sensores foram modificados automaticamente, através de otimização

estocástica. A idéia dessa máquina de estados é fornecer ao humanóide um ciclo padrão de passadas, o qual é explicado a seguir.

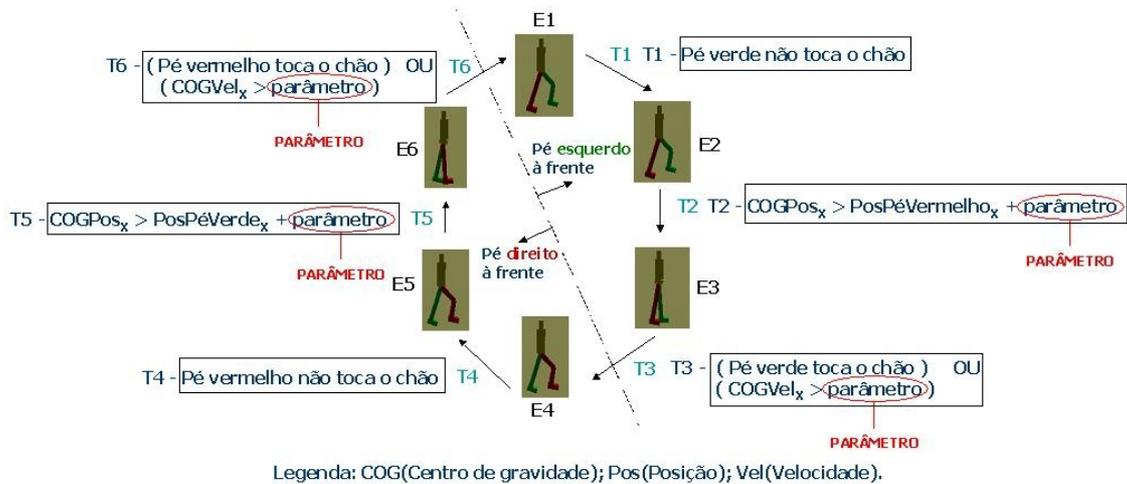


Figura 6.5: Máquina de estados constante responsável pelo movimento do tronco e das pernas de um humanóide ao caminhar.

Primeiramente, o estado E1 ativo tenta fazer com que a perna verde saia do chão e seja movida para frente da perna vermelha, iniciando a primeira passada. O estado E2 é idêntico ao estado E1 e o sensor da transição T1 simplesmente verifica se o pé verde não toca o chão. Portanto, a execução dessa primeira transição apenas garante que a perna verde já saiu do chão, e pode ser movida para frente, além de indicar que provavelmente ela estará à frente da perna vermelha quando o estado E3 se tornar ativo. Em seguida, já estando ativo o estado E2, a posição do centro de gravidade (Center of Gravity - COG) do modelo é analisada, verificando o quão à frente do pé de suporte (pé vermelho) ele está. Se o COG do modelo se localiza à frente da sua área de suporte, ele perde o equilíbrio e tende a cair para frente (Oshita & Makinouchi, 2001). O estado E3 deve então ser ativado para que a perna verde atinja logo o chão, aumentando a área de suporte do modelo e retomando o seu equilíbrio.

Quando o COG do modelo está a certa distância (determinada pelo valor do parâmetro do sensor associado) à frente do pé vermelho, a transição T2 é executada e o estado E3 é ativado. O sensor da transição T3 é responsável por verificar se o pé verde já atingiu o chão. Quando essa transição é executada, o estado E4 torna-se ativo, fazendo com que o modelo avance com o pé direito e comece outra passada. O sensor da transição T3 também verifica se a velocidade do COG do modelo está alta, acima de certa velocidade (determinada pelo valor do parâmetro do sensor), a fim de antecipar a próxima passada, já que o modelo tenderá a cair para frente mais rapidamente. Ao antecipar a passada, o modelo tende a reduzir sua velocidade. A próxima passada, a

partir do estado E4, é então realizada de forma simétrica, retornando ao estado E1 novamente. O ciclo é então repetido continuamente.

Sensores podem ser projetados com um maior número de testes, a fim de gerar um movimento mais preciso e estável e melhorar o resultado. Os sensores das transições T2, T3, T5 e T6 são projetados cada qual em função de um parâmetro que indica quando exatamente a transição deve ser executada, permitindo que o animador trabalhe em um nível mais alto. Otimização estocástica foi utilizada para modificar automaticamente esses quatro parâmetros, indicados na Figura 6.5 pelas elipses.

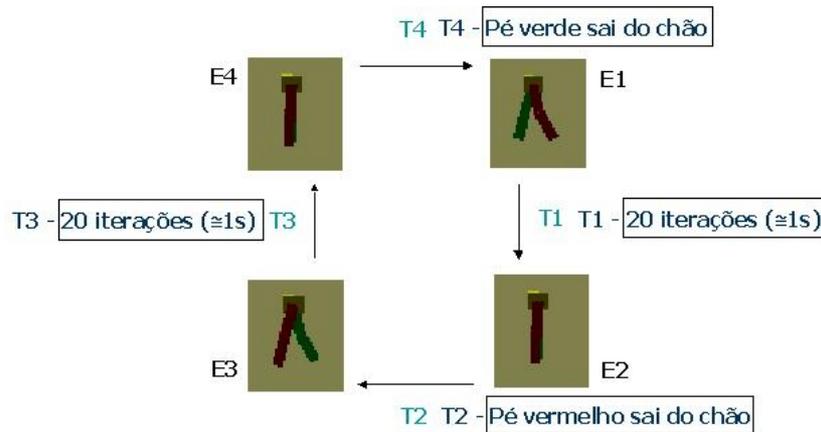


Figura 6.6: Máquina de estados constante responsável pelo movimento dos braços de um humanoide ao caminhar.

A máquina de estados responsável pelo movimento dos braços (Figura 6.6) foi toda projetada manualmente. Ela possui quatro estados, duas transições sensoriais e duas transições temporais. A idéia dessa máquina de estados é acompanhar a movimentação das pernas, ajudando o humanoide a manter o seu equilíbrio. No estado E1, o braço verde é inicialmente posicionado à frente do vermelho. A transição T1 é temporal e é executada após 20 iterações (equivalente a um segundo de simulação). Quando essa transição é executada, o estado E2 torna-se ativo, fazendo com que o modelo alinhe os seus braços. O sensor da transição T2 verifica se o pé vermelho não toca o chão, indicando que ele já saiu do chão para avançar para frente do pé verde. Quando essa transição é executada, o estado E3 torna-se ativo, fazendo com que o braço verde avance junto com a perna vermelha. O restante dessa máquina de estados, que vai do estado E3 de volta ao estado E1, também é realizado de forma simétrica. O ciclo é então também repetido continuamente.

A utilização de mais de uma máquina de estados torna os movimentos de diferentes partes do modelo mais independentes, podendo aumentar o realismo. Neste

exemplo, os braços não obedecem rigorosamente ao ritmo das pernas. Isso torna o caminhar mais natural.

A otimização estocástica da máquina de estados da Figura 6.5 foi realizada em dois estágios e a função objetivo (*fitness*) do processo de otimização foi a distância atingida pelo centro de massa do modelo. Nos dois estágios, o conjunto de parâmetros modificados consistiu apenas dos quatro parâmetros sensoriais indicados na Figura 6.5 pelas elipses. O intervalo de valores admissíveis para esses parâmetros foi de -0.25 a 0.75 . O valor inicial 0.25 foi determinado para todos eles, embora não tenha sido levado em consideração pelo modificador randômico. Também nos dois estágios, o *fitness* foi obtido ao final de 1000 iterações a cada avaliação. Entretanto, se durante as 1000 iterações o modelo caísse ou parasse, a simulação era encerrada e penalizada, retornando *fitness* igual a 0 (zero).

No primeiro estágio, todos os parâmetros foram modificados a cada avaliação, obedecendo apenas os limites dos seus intervalos, e, conseqüentemente, uma busca mais global foi realizada. Nesse estágio, foram realizadas 15 modificações e avaliações. No segundo estágio, com a utilização do conjunto vencedor de valores que foi obtido no primeiro estágio, apenas um parâmetro foi modificado a cada avaliação, realizando-se, portanto, uma busca mais local. Os intervalos de valores admissíveis para os parâmetros foram mantidos fixos durante todo o estágio. Nesse estágio, foram realizadas 30 modificações e avaliações. A Tabela 6.3 mostra a busca realizada nos dois estágios.

Tabela 6.3: Modificações automáticas nos parâmetros de um controlador utilizado para fazer um modelo de humanoíde caminhar em uma superfície plana.

Avaliações	Valor do parâmetro do sensor referente a:				<i>Fitness</i> (m)
	Transição 2 (m)	Transição 3 (m/s)	Transição 5 (m)	Transição 6 (m/s)	
Antes da otimização					
Inicial	0.250	0.250	0.250	0.250	0.000
Otimização					
Estágio 1					
Inicial	0.250	0.250	0.250	0.250	0.000
1	-0.013	0.417	0.093	0.331	8.133
2	0.363	0.264	0.134	0.291	0.000
3	-0.237	0.362	0.093	0.126	7.290
4	0.359	0.314	0.345	-0.085	0.000
5	0.044	-0.090	-0.211	0.232	8.284
6	-0.087	0.205	0.034	-0.116	8.372
7	0.069	0.672	0.644	0.249	0.000
8	-0.218	0.349	0.087	0.386	0.000
9	0.282	0.276	0.340	0.215	0.000

10	-0.228	0.663	0.299	0.423	0.000
11	0.404	0.547	0.425	-0.069	0.000
12	0.331	-0.088	0.302	0.036	0.000
13	0.061	0.505	0.018	0.597	7.660
14	0.720	-0.144	0.553	-0.145	0.000
15	-0.019	0.179	0.584	-0.232	0.000
Vencedora (6)	-0.087	0.205	0.034	-0.116	8.372
Estágio 2					
Inicial	-0.087	0.205	0.034	-0.116	8.372
1	-0.087	0.205	0.034	0.628	0.000
2	-0.087	0.205	0.522	-0.116	0.000
3	0.067	0.205	0.034	-0.116	8.693
4	0.067	0.205	0.034	0.195	8.225
5	0.067	-0.230	0.034	-0.116	8.798
6	0.067	-0.230	0.034	-0.068	8.798
7	0.209	-0.230	0.034	-0.116	0.000
8	0.067	-0.230	0.146	-0.116	0.000
9	0.327	-0.230	0.034	-0.116	0.000
10	0.069	-0.230	0.034	-0.116	8.870
11	0.069	-0.230	0.034	-0.023	8.870
12	0.704	-0.230	0.034	-0.116	0.000
13	0.069	-0.230	0.628	-0.116	0.000
14	0.069	-0.149	0.034	-0.116	8.870
15	0.069	-0.049	0.034	-0.116	8.870
16	0.069	-0.230	0.698	-0.116	0.000
17	0.661	-0.230	0.034	-0.116	0.000
18	0.069	-0.230	-0.153	-0.116	8.936
19	0.069	0.557	-0.153	-0.116	7.595
20	0.069	-0.230	0.489	-0.116	0.000
21	0.069	-0.230	0.060	-0.116	9.709
22	0.069	0.665	0.060	-0.116	8.883
23	-0.179	-0.230	0.060	-0.116	10.528
24	-0.179	-0.000	0.060	-0.116	10.528
25	-0.179	0.384	0.060	-0.116	8.946
26	-0.179	-0.230	-0.001	-0.116	0.000
27	-0.179	-0.230	0.060	0.082	10.528
28	-0.179	-0.230	0.060	0.380	8.289
29	-0.179	0.134	0.060	-0.116	0.000
30	-0.179	-0.021	0.060	-0.116	10.528
Vencedora (23)	-0.179	-0.230	0.060	-0.116	10.528
Depois da otimização					
Final	-0.179	-0.230	0.060	-0.116	10.528

A Figura 6.7 mostra a comparação do caminhar obtido antes e depois da otimização. O *fitness* obtido antes da otimização é mostrado na Figura 6.7 sem considerar as penalidades pela queda do modelo. O controlador depois de otimizado,

além de obter um melhor *fitness* na sua avaliação, fez o modelo continuar andando após as 1000 iterações avaliadas, atingindo uma distância de 45.147 m, antes de cair.

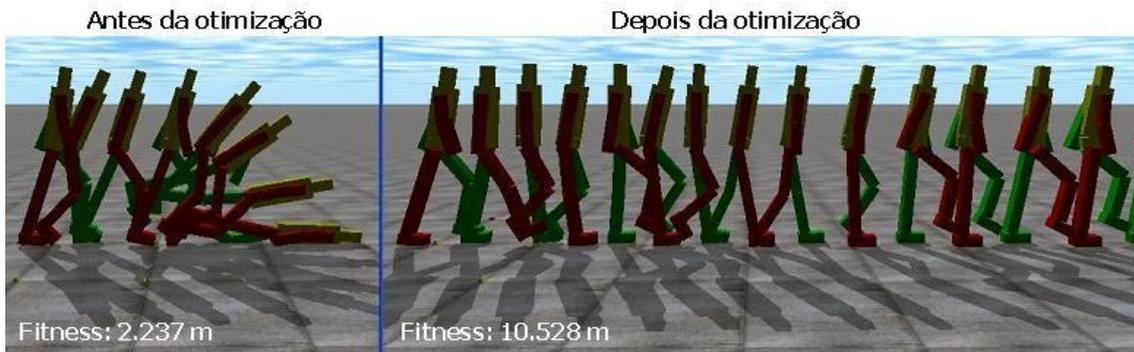


Figura 6.7: Comparação do caminhar antes e depois da otimização.

A partir desse exemplo, alguns exemplos variantes foram analisados com a finalidade de avaliar alguns aspectos importantes da representação “máquinas de estados sensoriais”. Essa representação permite tratar a simetria dos parâmetros (Van de Panne, 1996) a fim de diminuir o espaço de busca da otimização estocástica, através da utilização de procedimentos. A primeira variação é relacionada ao tratamento da simetria, onde um procedimento foi definido para que os parâmetros dos sensores associados às transições T5 e T6 recebessem os valores dos parâmetros dos sensores associados às transições T2 e T3, respectivamente. Esse procedimento foi associado ao estado E1, de tal forma que, quando iniciada a execução da máquina de estados, esse procedimento era executado. O conjunto de parâmetros modificados automaticamente foi reduzido aos parâmetros dos sensores associados às transições T1 e T2, apenas. Ao iniciar cada avaliação, a máquina de estados era executada, o procedimento também era logo executado e os parâmetros das transições T5 e T6 eram atualizadas simetricamente em relação às transições T1 e T2. Um teste de otimização semelhante ao apresentado na Tabela 6.3 foi realizado, levando-se em conta esse tratamento da simetria.

Com o espaço de busca menor, a otimização realizada com os mesmos estágios e após as 45 avaliações resultou em um controlador que retornou um *fitness* de 10.030 m, praticamente igual ao do exemplo anterior. Entretanto, após as 1000 iterações avaliadas, o modelo continuou andando até atingir uma distância de 100.739 m, antes de cair, demonstrando uma estabilidade bem maior. Além disso, espera-se que, em experimentos com um maior número de avaliações, melhores resultados sejam obtidos mais rapidamente quando a simetria dos parâmetros for tratada.

A segunda variação foi realizada para avaliar a estrutura proposta para os sensores. As transições sensoriais T2, T3, T5 e T6 da máquina de estados responsável

pelo movimento do tronco e das pernas foram substituídas por transições temporais. Como os sensores das transições sensoriais T1 e T4 não possuem nenhum parâmetro, eles foram mantidos sem alterações. Ao invés de otimizar os parâmetros dos sensores das transições, os tempos de transição é que foram otimizados. O intervalo de valores admissíveis para esses parâmetros foi de 10 a 30 iterações (tempo). O valor inicial de 20 iterações foi determinado para todos eles, embora não tenha sido levado em consideração pelo modificador randômico.

Essa otimização, sem considerar as penalidades pela queda do modelo, realizada com os mesmos estágios, após as 45 avaliações, resultou em um controlador que retornou um *fitness* de 6.332 m, menor do que o *fitness* resultante da otimização utilizando as transições sensoriais. Além disso, o modelo caiu antes mesmo das 1000 iterações avaliadas. Considerando as penalidades, todos os *fitness* obtidos nas 45 avaliações foram iguais a 0 (zero). Realizando-se um maior número de avaliações durante a otimização, o controlador resultante conseguiu fazer com que o modelo não caísse antes das 1000 iterações avaliadas. Entretanto, com as otimizações realizadas utilizando as transições temporais, o modelo não conseguiu chegar perto dos 45.147 m (distância atingida pelo controlador resultante da otimização utilizando as transições sensoriais).

Espera-se que, em experimentos com um maior número de avaliações, melhores resultados sejam obtidos mais rapidamente quando transições sensoriais forem utilizadas. Além disso, a otimização utilizando as transições sensoriais (Figura 6.7) produziu um controlador mais estável que tanto interagiu melhor com alterações do ambiente, tais como forças externas aplicadas no modelo; quanto manteve um melhor desempenho mesmo após o número de iterações avaliadas na otimização. A Figura 6.8 mostra esse controlador funcionando, mesmo quando forças externas são aplicadas ao modelo.

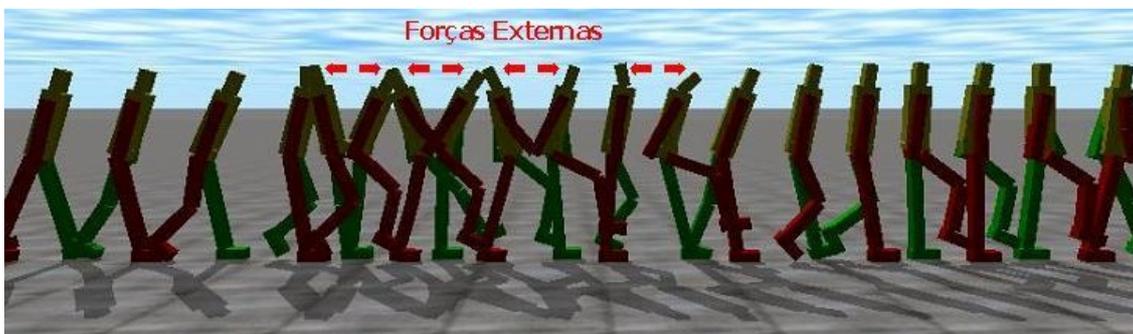


Figura 6.8: Controlador otimizado (mostrado na Figura 6.7), interagindo com pequenas forças externas sendo aplicadas na cabeça do modelo.

A terceira variação foi feita para realizar um teste simples de composição de movimentos. Duas máquinas de estados sensoriais projetadas manualmente foram adicionadas à estrutura utilizada no exemplo original. A primeira delas foi utilizada como exemplo e explicada na Seção 4.5.2. Seu sensor associado verifica se o modelo está caindo para frente. Assim que o sensor torna-se ativo, essa máquina de estados começa a atuar no modelo para que os braços sejam erguidos e protejam a cabeça durante a queda. Caso contrário, o modelo continuaria apenas tentando andar e bater a cabeça no chão, como na Figura 6.7. Essa máquina de estados possui apenas um estado, e uma pose associada em que os braços do modelo estão erguidos à frente da cabeça, semelhante à última configuração mostrada na Figura 4.7.

O sensor associado à segunda máquina de estados sensorial verifica se o modelo está caindo para trás. Ele calcula a distância horizontal do quadril em relação ao pé que está posicionado mais atrás, e é ativado quando essa distância ultrapassa um valor de tolerância. Inicialmente, as máquinas de estados responsáveis pelo caminhar atuam no modelo. Durante a caminhada, uma força externa é aplicada na parte da frente da cabeça do modelo a fim de causar a perda de equilíbrio. O modelo tende a cair para trás e, no instante em que o sensor é ativado (distância do quadril ao pé de trás supera a tolerância), a máquina de estados sensorial associada começa a atuar no modelo para que os braços sejam movimentados para trás a fim de proteger sua região anterior durante a queda. Se essa máquina não fosse ativada, o modelo continuaria apenas tentando andar e bater com força sua região anterior no chão. Essa máquina de estados também possui apenas um estado, e uma pose associada em que os braços estão posicionados atrás do modelo, semelhante à última configuração mostrada na Figura 6.9.

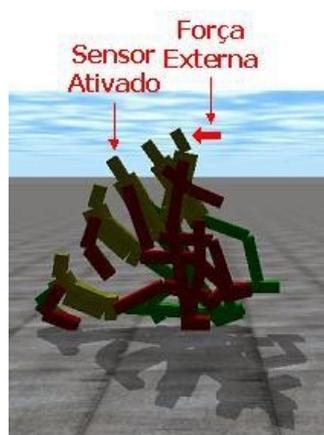


Figura 6.9: Ativação de uma máquina de estados sensorial durante a queda, para proteção do corpo com os braços.

6.3.2 Arremesso de uma Bola ao Cesto

Este teste consiste em fazer um humanóide arremessar uma bola de basquete em direção ao cesto. A estrutura do controlador gerado consiste de duas máquinas de estados constantes: uma para manter o humanóide em posição ereta, e a outra para controlar o movimento dos braços.

A máquina de estados responsável pela postura do humanóide foi desenvolvida manualmente. Ela cuida do tronco e das pernas do humanóide para mantê-lo em posição ereta. Essa máquina de estados possui apenas um estado ao qual está associada uma pose em que as pernas do modelo estão abertas a fim de facilitar seu equilíbrio (vide Figura 6.11).

A máquina de estados responsável pelo movimento dos braços (atuadores 11, 12, 13 e 14) possui três estados (Figura 6.10). O estado E3 não tem efeito no modelo e serve apenas para soltar os braços após o arremesso. Os estados E1 e E2 foram gerados automaticamente, e as poses mostradas na Figura 6.10 correspondem aos valores dos ângulos desejados utilizados como ponto inicial no processo de otimização. Os intervalos de valores admissíveis para esses parâmetros (ângulos desejados), considerando que na Figura 6.4 os ângulos dos ombros e dos cotovelos possuem valores iguais a 0.0 radiano, foram: de -0.2 a 2.6 radianos para os ombros; e de 0.0 a 2.5 radianos para os cotovelos. Os valores iniciais determinados foram: 1.5 radianos para os ombros; e 2.0 radianos para os cotovelos. Os dois braços foram mantidos unidos durante toda a simulação. Portanto, um procedimento foi associado ao estado E1 para tratar a simetria dos parâmetros dos dois braços na otimização, como no exemplo do caminhar na Seção 6.3.1.

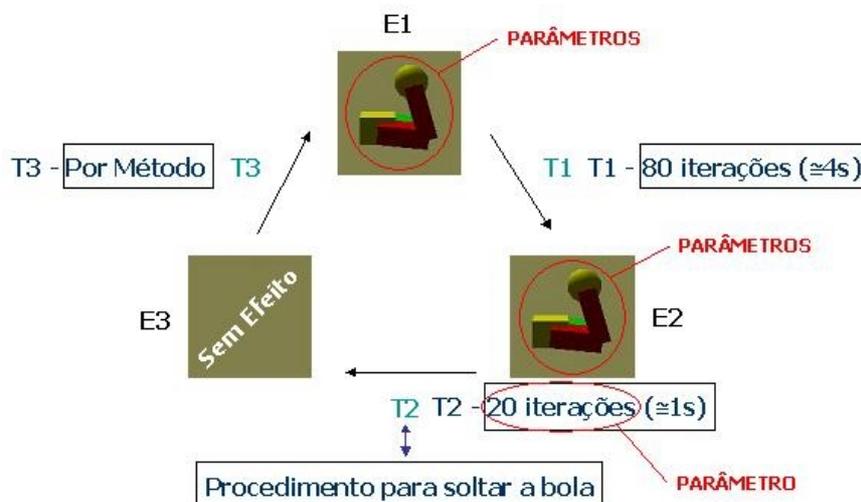


Figura 6.10: Máquina de estados constante responsável pelo movimento dos braços de um humanóide ao arremessar uma bola ao cesto.

As transições T1 e T2 são temporais, enquanto que a transição T3 foi definida apenas pela exigência da máquina de estados ser cíclica. No entanto, como o movimento é acíclico, T3 é definida como uma transição por método que nunca é executada. O tempo utilizado na transição T1, projetado manualmente, foi de 80 iterações (tempo). O tempo da transição T2 também fez parte do espaço de parâmetros do controlador que foi gerado automaticamente. O intervalo de valores admissíveis desse parâmetro foi de 0 a 40 iterações (tempo), e seu valor inicial foi de 20 iterações. A T2 está associado um procedimento responsável por soltar a bola, assim como no exemplo explicado na Seção 4.5.3 (Figura 4.8).

O processo de otimização estocástica dessa máquina de estados foi realizado em um único estágio. Os parâmetros que fazem parte do espaço de busca estão indicados através de elipses na Figura 6.10. Eles são os ângulos desejados das poses associadas aos estados E1 e E2, e do tempo da transição T2. Como a simetria entre os ângulos desejados dos braços foi tratada, foram modificados apenas dois parâmetros de cada estado, totalizando cinco elementos no conjunto de parâmetros modificados. As avaliações foram realizadas até o momento em que a bola atingia o chão. Entretanto, caso o humanóide caísse, a simulação era encerrada e penalizada, retornando *fitness* igual a 0 (zero). O critério de otimização (*fitness*) utilizado foi calculado pela função

$$\text{Fitness} = 10 + \frac{H_{\max}}{3} - \Delta_{\min}, \quad (6.1)$$

onde, durante toda a avaliação, H_{\max} é a altura máxima atingida pela bola e Δ_{\min} é a menor distância medida entre o centro da bola e o ponto determinado a 0.25 m acima do cesto. A constante igual a 10 adicionada serve apenas para que as penalidades sejam consideradas corretamente.

A cada avaliação, os parâmetros modificados eram escolhidos segundo duas alternativas: 1) apenas os parâmetros do estado E1, e 2) os parâmetros do estado E2 e o parâmetro da transição T2. Os intervalos de valores admissíveis dos parâmetros foram reduzidos parcialmente, em 50%. Em cada avaliação, o modificador randômico utiliza o valor corrente do parâmetro como centro do intervalo de sorteio que é limitado em $\pm 25\%$ do intervalo original. Assim, por exemplo, se o intervalo de valores admissíveis original tiver tamanho igual a 40 e o valor corrente do parâmetro for 20, o intervalo de valores admissíveis para sorteio é de $20 \pm 0.25 \cdot 40$, ou seja, o intervalo [10, 30]. Foram realizadas 400 modificações e avaliações.

Antes de adotar a estratégia de otimização discutida nos parágrafos anteriores, foi tentada uma outra estratégia em que todos os parâmetros do controlador faziam parte do espaço de busca a cada avaliação, respeitando os limites dos intervalos de valores admissíveis de cada parâmetro. Essa estratégia conduziu a um resultado em que o modelo arremessava a bola por baixo, e, por esse motivo, foi descartada.

A Figura 6.11 mostra a comparação do arremesso da bola ao cesto obtido antes e depois da otimização. O *fitness* obtido antes da otimização é igual a 8.079 e o *fitness* obtido depois da otimização é igual a 11.042. O controlador otimizado conseguiu realmente fazer com que o modelo acertasse o cesto, arremessando a bola por cima. Além disso, a bola atingiu uma boa altura de tal forma que a existência da tabela não foi determinante no acerto do arremesso. A Figura 6.12 mostra a trajetória que a bola percorreu depois de arremessada ao cesto, utilizando o controlador otimizado.



Figura 6.11: Comparação do arremesso de uma bola ao cesto antes e depois da otimização.

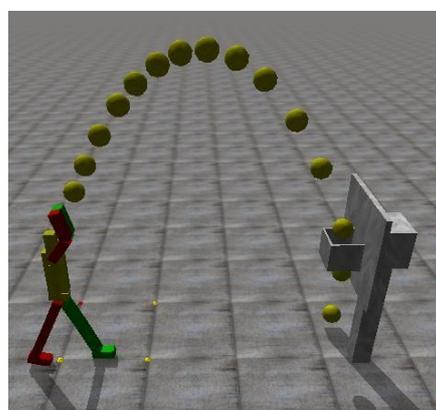


Figura 6.12: Trajetória percorrida pela bola depois de arremessada ao cesto, utilizando o controlador otimizado.

6.3.3 Salto Mortal para Trás

Este teste consiste em fazer um humanóide executar um salto mortal para trás com o auxílio de um outro humanóide que o impulsiona. A interação entre dois modelos de humanóide é testada. Como na Figura 6.3, o segundo modelo disponibiliza suas mãos para que o primeiro posicione um pé sobre elas e receba o impulso.

Os dois humanóides são tratados como submodelos de um único modelo que os engloba. Devido à generalidade da representação de controladores proposta neste trabalho, um mesmo controlador pode ser definido para atuar nos dois submodelos de forma independente. O tratamento de colisões também pode ser realizado considerando os submodelos independentes. Neste exemplo, as colisões são tratadas apenas entre corpos pertencentes a diferentes submodelos ou entre os corpos e o chão. Como nos outros exemplos, não são tratadas as colisões entre corpos pertencentes a um mesmo modelo.

A estrutura do controlador gerado consiste de três máquinas de estados constantes projetadas manualmente: duas para o primeiro submodelo e uma para o segundo submodelo. Assim como na máquina de estados responsável pelo movimento dos braços no exemplo do arremesso de bola ao cesto, as últimas transições dessas três máquinas de estados são por método e nunca são realizadas, determinando um movimento acíclico.

A máquina de estados associada ao humanóide impulsionador está mostrada na Figura 6.13. Ela possui quatro estados e quatro transições, e é responsável por fazer com que esse humanóide perceba quando o humanóide acrobata está preparado para receber o impulso, realize o impulso e assuma uma posição relaxada. Os estados E1 e E2 são idênticos, e as poses a eles associadas indicam uma configuração inicial de preparação, disponibilizando as mãos para que o acrobata posicione um pé sobre elas e possa receber o impulso. O sensor da transição T1 é o responsável por verificar se o acrobata já está preparado para receber o impulso. Essa verificação é realizada testando se um dos pés do humanóide acrobata está tocando as duas mãos do humanóide impulsionador. Se o teste for positivo, a transição T2 simplesmente funciona como uma contagem regressiva para que os dois humanóides possam começar a realizar suas tarefas em conjunto, em instantes próximos. O tempo utilizado nessa transição foi de 100 iterações. A ativação do estado E3 então faz com que um impulso necessário para ajudar o acrobata a realizar o salto mortal seja fornecido pelo impulsionador. Para isso, a pose associada a esse estado possui os braços levantados, o tronco inclinado para trás

e os joelhos menos flexionados. Em seguida, o estado E4 serve para que o humanóide assumira uma posição aparentemente relaxada, sendo ativado após 50 iterações, tempo definido para a transição T3.

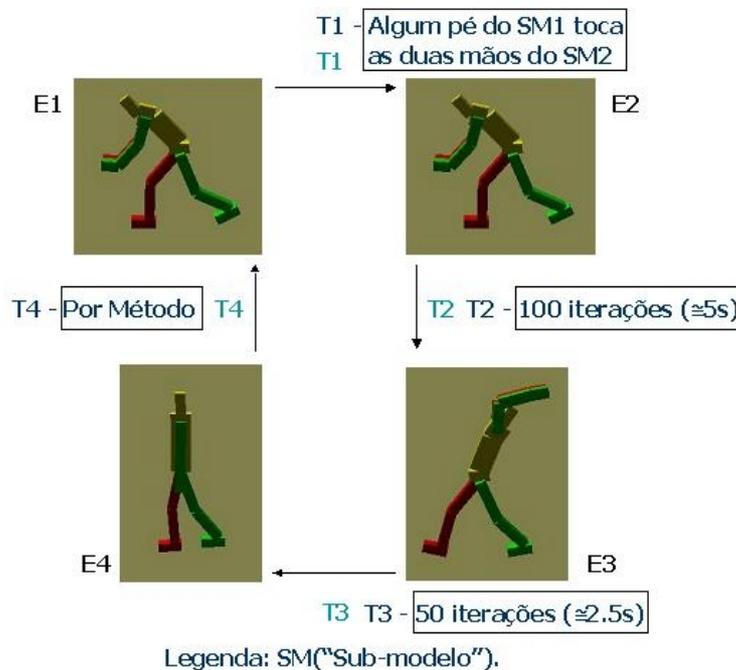


Figura 6.13: Máquina de estados constante responsável por fazer um humanóide impulsionar um outro, para que este realize um salto mortal.

A primeira máquina de estados associada ao humanóide acrobata está mostrada na Figura 6.14 e possui oito estados. Ela é responsável pelo movimento de seu tronco e de suas pernas. A pose associada ao estado E1 corresponde apenas a uma configuração inicial para o humanóide. Após 80 iterações, tempo definido para a transição T1, o estado E2 é ativado para que a perna verde do humanóide seja levantada, e para que, após 40 iterações, tempo definido para a transição T2, ela seja posicionada de tal forma que o pé verde do humanóide fique encostado sobre as duas mãos do humanóide impulsionador. Essa ação é proveniente da ativação do estado E3. O sensor da transição T3 funciona da mesma maneira que o sensor da transição T1 da máquina de estados da Figura 6.13. Ele é também responsável por verificar se os humanóides estão preparados para agirem em conjunto. Entretanto, enquanto o humanóide impulsionador apenas espera 100 iterações para agir, nessa máquina de estados, assim que esses sensores tornam-se ativos, o estado E4 é ativado e, após 85 iterações (tempo definido para a transição T4), o estado E5 é que se torna o ativo. Esses estados são responsáveis por fazerem com que o humanóide pegue primeiramente um impulso do chão. Durante essas 85 iterações (estado E4 ativo), o humanóide flexiona o joelho e o calcanhar vermelhos. Em seguida, durante as próximas 20 iterações (estado E5 ativo), tempo

definido para a transição T5, o humanóide estende o joelho e o calcanhar vermelhos e inclina o tronco para frente para que ele possa pegar o impulso do chão e se aproxime mais do humanóide impulsor.

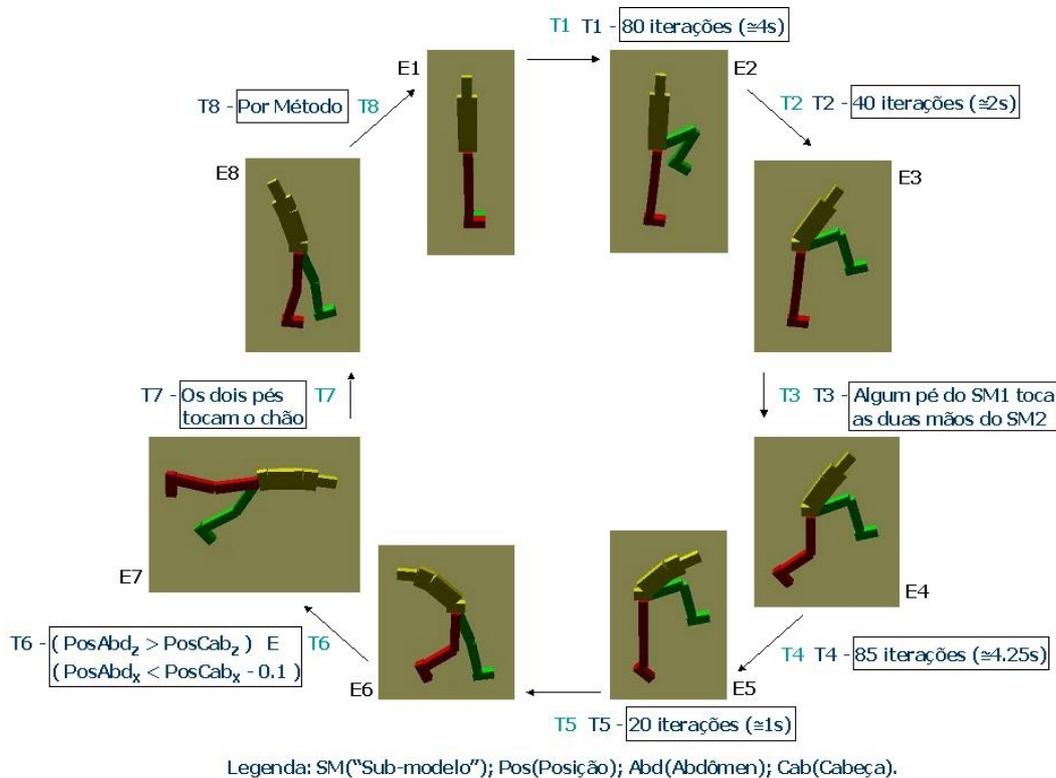


Figura 6.14: Máquina de estados constante responsável pelo movimento do tronco e das pernas de um humanóide ao executar um salto mortal, impulsionado por um outro.

Assim que a transição T5 é executada, 5 iterações após o estado E3 da máquina de estados da Figura 6.13 tornar-se ativo, o estado E6 também se torna ativo, fazendo com que o humanóide estenda a sua perna verde e aproxime uma perna da outra, para que ele seja melhor impulsionado para cima, e incline o tronco e a cabeça para trás, para que ele possa girar. A perna vermelha é mantida um pouco flexionada para tentar evitar excessivas colisões com os braços do humanóide impulsor, embora ainda haja algumas. A transição T6 e o estado E7 são os que mais influenciam o sucesso da aterrissagem do humanóide acrobata, exigindo serem projetados com um cuidado especial, devido à dificuldade de se obter o equilíbrio após algum tipo de aterrissagem (Zhao, 2004). O sensor da transição T6 analisa a posição do humanóide e é responsável por verificar se já está no momento certo para ele frear o giro. Esse sensor torna-se ativo quando o abdômen do humanóide está acima e a mais de 0.1 m à esquerda de sua cabeça. Assim que esse sensor torna-se ativo, o estado E7 também se torna ativo, fazendo com que o humanóide: incline um pouco o tronco para frente em relação à pose

associada ao estado anterior (isso ajuda a diminuir a velocidade angular); e abra as pernas, deixando os joelhos um pouco flexionados, para facilitar a obtenção do equilíbrio após a aterrissagem. Note-se que uma pose não define a orientação global ou o posicionamento global desejados. Portanto, a pose associada ao estado E7 está desenhada na forma girada apenas para que fique mais intuitiva a idéia de que o humanóide realiza uma rotação no ar. O sensor da transição T7 é responsável apenas por verificar se os dois pés do humanóide estão tocando o chão. Assim que seus dois pés tocam o chão, considera-se que ele aterrissou e o estado E8 torna-se ativo, fazendo com que ele tente assumir uma posição ereta.

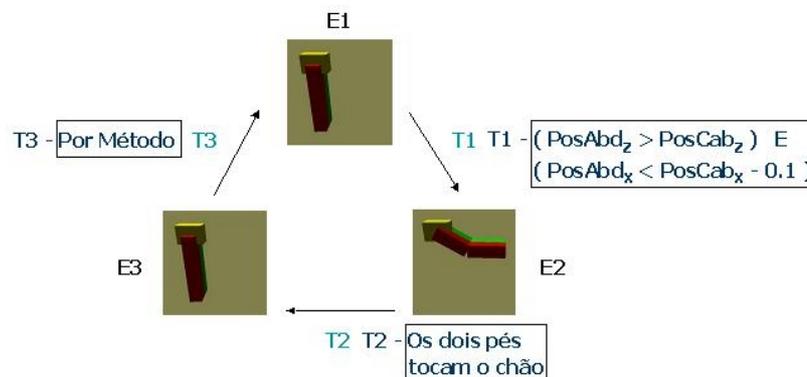


Figura 6.15: Máquina de estados constante responsável pelo movimento dos braços de um humanóide ao executar um salto mortal, impulsionado por um outro.

A segunda máquina de estados associada ao humanóide acrobata está mostrada na Figura 6.15 e possui apenas três estados e três transições. Ela é responsável pelo movimento dos braços de forma a ajudar a obter o equilíbrio do humanóide após a aterrissagem e a tornar o movimento mais natural. Os sensores das transições T1 e T2 funcionam da mesma maneira que os sensores das transições T6 e T7 da máquina de estados da Figura 6.14, respectivamente. Inicialmente, os braços do humanóide são mantidos estendidos para baixo (estado E1). Assim que a transição T1 é executada, os seus braços são posicionados à sua frente a fim de frear melhor o giro e de distribuir melhor o seu peso no momento do pouso, ajudando a obter seu equilíbrio (estado E2). Em seguida, assim que a transição T2 é executada, o estado E3, idêntico ao estado E1, torna-se ativo, fazendo com que o humanóide volte a assumir novamente uma posição relaxada para seus braços.

A Figura 6.16 mostra os quadros da animação gerada ao se aplicar o controlador aqui descrito. Já a Figura 6.17, mostra a trajetória realizada pelo humanóide acrobata,

executando um salto mortal para trás, e impulsionado pelo humanóide impulsionador, toda num mesmo quadro.

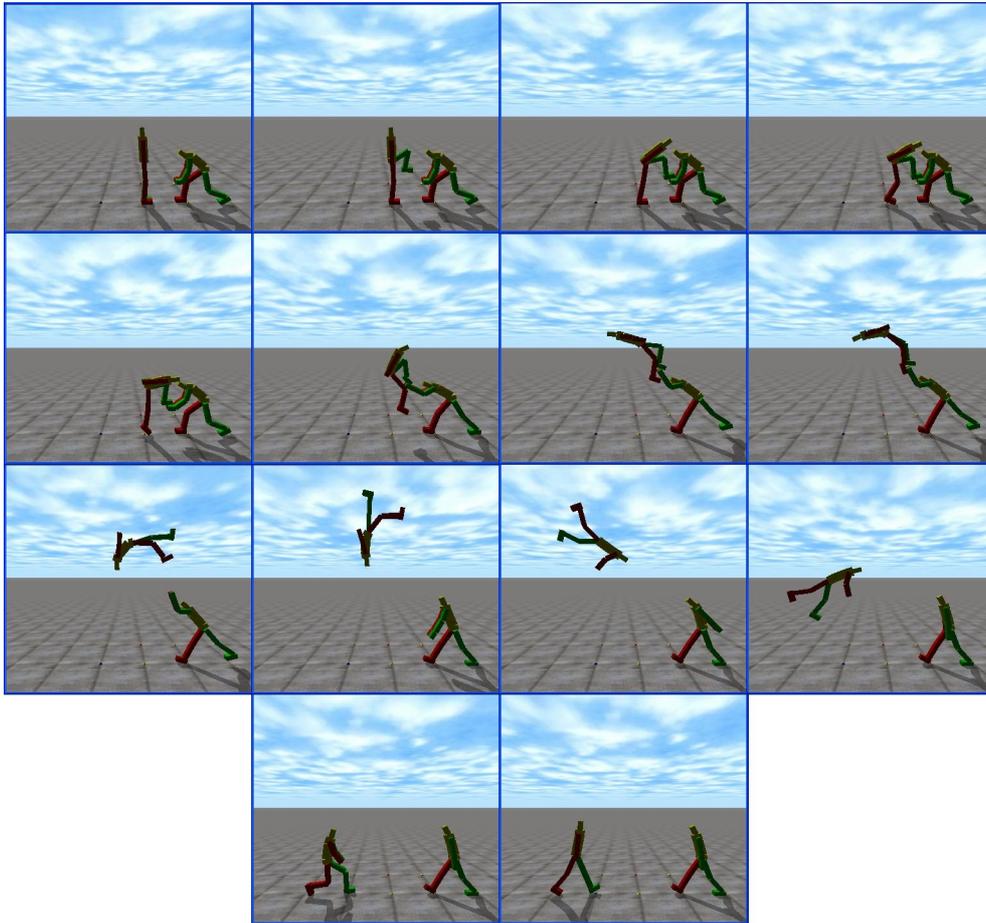


Figura 6.16: Quadros da animação de um humanóide executando um salto mortal para trás, impulsionado por um outro.

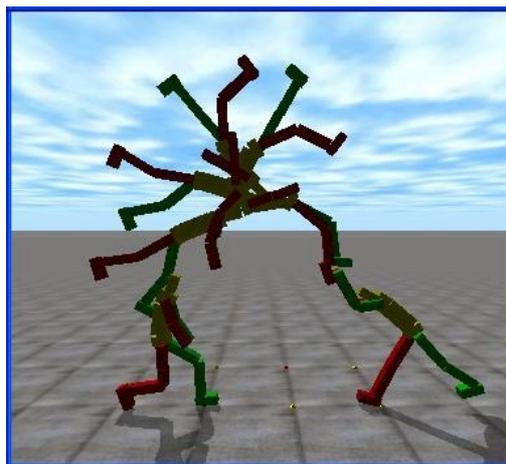


Figura 6.17: Trajetória realizada por um humanóide executando um salto mortal para trás, impulsionado por um outro.

6.4 Considerações Finais

Neste capítulo, foram apresentados os resultados dos testes de animação realizados, utilizando modelos de humanóide 2D, para avaliar a representação “máquinas de

estados sensoriais” proposta. Os testes realizados fizeram um humanóide caminhar em uma superfície plana, arremessar uma bola ao cesto e executar um salto acrobático para trás, impulsionado por um outro humanóide, explorando a interação entre modelos.

Os resultados dos testes com a representação proposta, em contraste com aquelas apresentadas na literatura, permitem considerar que:

- ▶ Ela é capaz de tratar diferentes partes do modelo com mais independência, permitindo, inclusive, que a interação entre submodelos seja realizada de uma maneira intuitiva;
- ▶ Ela permite que instruções mais gerais sejam definidas pelo animador, através de procedimentos associados a sua estrutura;
- ▶ Sua estrutura elaborada de sensores é mais adequada à geração automática dos controladores, pois permite que o animador pense no funcionamento do sensor em um nível mais alto; e
- ▶ A utilização de máquinas de estados sensoriais pode disponibilizar para o modelo diversas habilidades específicas, tornando-o mais autônomo. Essas máquinas de estados, além de reações de proteção, como nos exemplos das figuras 4.2 e 6.9, podem disponibilizar ao modelo ações, tais como levantar, sentar e pular. Os sensores associados a essas máquinas de estados são responsáveis por determinar as condições apropriadas para suas execuções (composição de movimentos).

O objetivo principal da realização de modificações automáticas nos parâmetros dos controladores foi mostrar que a otimização estocástica utilizada nessa representação:

- ▶ É facilitada pela utilização da estrutura de sensores proposta, composta por parâmetros mais expressivos;
- ▶ Facilita o projeto do controlador; e
- ▶ É capaz de obter bons resultados.

Neste capítulo, foram mostrados exemplos de otimização estocástica em que buscas rápidas foram suficientes para a obtenção de bons resultados. A busca realizada na otimização do caminhar, por exemplo, pôde ser mostrada em detalhes. Isso facilitou sua análise e compreensão. Entretanto, buscas bem mais demoradas e otimizações bem mais elaboradas, explorando as diferentes maneiras disponíveis de se modificar os

parâmetros em cada estágio, são possíveis de serem realizadas e devem trazer melhores resultados.

A busca realizada na otimização do arremesso envolveu a realização de um número maior de avaliações do que no exemplo do caminhar. Entretanto, isso não implica que o arremesso é um movimento mais difícil de se controlar. A otimização também depende de quantos e quais parâmetros são escolhidos para comporem o espaço de busca (parâmetros gerados automaticamente). No exemplo do caminhar, apenas parâmetros de transições foram gerados automaticamente, enquanto que, no arremesso, alguns ângulos desejados também foram gerados automaticamente.

Definir o critério de otimização é uma tarefa já considerada difícil. Em relação ao humanóide caminhando em uma superfície plana, nem sempre os valores dos parâmetros do controlador que fazem o modelo chegar o mais longe possível em um determinado tempo são os que conseguem levar o modelo mais longe após o tempo de avaliação. Conseguir os valores para um caminhar mais estável requer um critério de otimização mais elaborado. Já em relação ao exemplo do salto acrobático, houve dificuldade de se encontrar um critério de otimização adequado para esse movimento desejado. Algumas tentativas de modificações automáticas foram realizadas nos parâmetros do controlador, mas elas não trouxeram melhores resultados, segundo uma avaliação visual. As modificações automáticas foram realizadas principalmente em relação à aterrissagem (transição T6 e estado E7) e ao giro (estado E6). A maior dificuldade está em comparar as modificações obtidas, pois um critério de otimização mal elaborado implica em uma má comparação. Maiores investigações quanto à definição de critérios de otimização devem ser realizadas.

O projeto manual do controlador responsável por realizar o salto mortal foi bem trabalhoso. O animador precisou realizar várias avaliações, visualizando o movimento gerado na tela do computador, para conseguir que o humanóide caísse de pé e ficasse em uma posição ereta após a aterrissagem, sem cair. Entretanto, a representação proposta se mostrou capaz de controlar movimentos considerados difíceis, apesar de necessitar de um razoável esforço manual para alguns exemplos.

Capítulo 7

Conclusões

O controlador ideal deve ser capaz de gerar movimentos que se aproximem ao máximo dos requisitos de movimento estabelecidos pelo animador, e da maneira mais rápida e automática possível. É justamente a dificuldade de gerar controladores com esse grau de versatilidade o principal problema da utilização da técnica de controladores.

O projeto de um controlador é suficientemente complexo para ser desenvolvido manualmente e não menos complexo e caro computacionalmente quando desenvolvido automaticamente. Enquanto o projeto manual dos parâmetros do controlador oferece ao animador um maior controle sobre esses parâmetros, o projeto automático é realizado em um nível mais alto de abstração que libera o animador do esforço de especificar e modificar cada um desses parâmetros. Portanto, apesar do menor controle sobre os parâmetros e do maior custo computacional exigido, poder projetar controladores automaticamente é essencial para que controladores possam ser projetados de uma maneira fácil e acessível a qualquer usuário.

Este trabalho abordou o problema de controle de movimentos de personagens autônomos e a dificuldade de projetar automaticamente um repertório de habilidades específicas a serem incorporadas por esses personagens. Foi proposta uma representação de controladores com uma estrutura mais geral e mais intuitiva (disponibilizando vários tipos de parâmetros), que permite que, de acordo com os retornos dos seus sensores, várias máquinas de estados possam atuar no mesmo modelo, simultaneamente, e em subconjuntos dos seus atuadores. Essa estrutura também permite que, durante seu projeto, procedimentos com instruções gerais, de acordo com o movimento desejado, possam ser definidos e adequadamente associados a ela pelo animador, para serem executados automaticamente pelo controlador durante a simulação dinâmica. Também foi proposta uma estrutura mais geral e mais elaborada para os sensores utilizados, composta por parâmetros mais expressivos, e mais adequada à geração automática dos controladores, pois permite que o animador possa pensar no funcionamento do sensor em um nível mais alto. A representação proposta facilita o projeto dos controladores, a composição de movimentos e a geração automática de controladores. Espera-se que, além da representação proposta, as idéias presentes nela

também possam ser úteis, mesmo que isoladamente, em outras representações que venham a surgir.

A representação proposta foi avaliada, através de testes de animação. Os testes realizados consistiram em fazer um humanóide: caminhar em uma superfície plana, arremessar uma bola ao cesto, e executar um salto acrobático para trás, impulsionado por um outro humanóide, explorando a interação entre modelos. Embora os testes tenham sido realizados utilizando modelos de humanóide 2D, a representação proposta é geral, e pode ser aplicada a modelos 3D.

Uma limitação da representação proposta, também encontrada nos trabalhos relacionados, é a necessidade de uma escolha cuidadosa dos parâmetros a serem modificados automaticamente para a obtenção de bons resultados, não permitindo que controladores para modelos muito complexos sejam completamente projetados de maneira automática. Além disso, mesmo para um número de parâmetros razoável, não há garantias de que uma boa solução será encontrada. Entretanto, a representação proposta facilita essa escolha, permitindo que ela possa ser realizada hierarquicamente, e permite também que a otimização possa ser realizada em vários estágios, onde possivelmente diferentes conjuntos de parâmetros são modificados.

Uma continuação natural deste trabalho é a disponibilização de mais tipos de movimento para o modelo humanóide apresentado, tais como correr e saltar, e analisar esta representação aplicada também a modelos 3D. Outra necessidade é a construção de uma interface para facilitar o projeto do controlador, principalmente na escolha dos parâmetros a serem gerados automaticamente em cada estágio da otimização. Além disso, poder-se-ia investigar como essa escolha pode também ser automatizada, facilitando ainda mais o trabalho do animador. Por último, a interação on-line com o animador também poderia ser mais explorada, podendo ter aplicação na produção de jogos fisicamente realistas bastante interessantes.

Referências

- Anitescu, M. & Potra, F. A. (1997). Formulating Dynamic Multi-Rigid-Body Contact Problems with Friction as Solvable Linear Complementarity Problems. *Nonlinear Dynamics*, 14, 231–247.
- Armstrong, W. W. & Green, M. (1985). The Dynamics of Articulated Rigid Bodies for Purposes of Animation. *Proceedings of Graphics Interface'1985* (pp. 407-415). Montreal, Quebec, Canada.
- Auslander, J., Fukunaga, A., Partovi, H., Christensen, J., Hsu, L., Reiss, P., Shuman, A., Marks, J. & Ngo, J.T. (1995). Further experience with controller-based automatic motion synthesis for articulated figures. *ACM Transactions on Graphics*, 14(4), 311–336.
- Bekey, G. A. (2005). *Autonomous Robots: From Biological Inspiration to Implementation and Control*. MIT Press.
- Carignan, M., Yang, Y., Magnenat-Thalmann, N. & Thalmann, D. (1992). Dressing Animated Synthetic Actors with Complex Deformable Clothes. *Proceedings of ACM SIGGRAPH'1992* (pp. 99–104). Chicago, USA.
- Costa-Filho, J. C. S. (2004). Um Sistema para Animação de Estruturas Articuladas através da técnica Spacetime Constraints. Dissertação de Mestrado, Departamento de Computação, Universidade Federal do Ceará.
- Faloutsos, P. (2002). *Composable Controllers for Physics-Based Character Animation*. PhD thesis, Department of Computer Science, University of Toronto.
- Faloutsos, P., van de Panne, M. & Terzopoulos, D. (2001). Composable controllers for physics-based character animation. *Proceedings of ACM SIGGRAPH'2001* (pp. 251–260). Los Angeles, USA.
- Grzeszczuk, R., Terzopoulos, D. & Hinton, G. (1998). Neuroanimator: Fast neural network emulation and control of physics-based models. *Proceedings of ACM SIGGRAPH'1998* (pp. 9–20). Orlando, USA.
- Hodgins, J.K. & Pollard, N. S. (1997). Adapting simulated behaviors for new characters. *Proceedings of ACM SIGGRAPH'1997* (pp. 153–162). Los Angeles, USA.

- Hodgins, J. K., Wooten, W. L., Brogan, D. C. & O'Brien, J. F. (1995). Animating human athletics. *Proceedings of ACM SIGGRAPH'1995* (pp. 71–78). Los Angeles, USA.
- Laszlo, J. F. (1996). Controlling Bipedal Locomotion for Computer Animation. M. Sc. thesis, University of Toronto.
- Laszlo, J. F., van de Panne, M. & Fiume, E. (1996). Limit cycle control and its application to the animation of balancing and walking. *Proceedings of ACM SIGGRAPH'1996* (pp. 155–162). New Orleans, USA.
- Laszlo, J. F., van de Panne, M. & Fiume, E. (1997). Control of Physically-based Simulated Walking. *Proceedings of IMAGINA'1997* (pp. 231-241). Monaco.
- Laszlo, J. F., van de Panne, M. & Fiume E. (2000). Interactive control for physically-based animation. *Proceedings of ACM SIGGRAPH'2000* (pp. 201–208). New Orleans, USA.
- Magnenat-Thalmann, N. (1994). Tailoring Clothes for Virtual Actors. *Interacting with Virtual Environments*, 205–216.
- MIT Leg Laboratory. <http://www.ai.mit.edu/projects/leglab/>. Visitado em 15 de abril de 2006.
- Multon, F., France, L., Cani-Gascuel, M. & Debunne, G. (1999). Computer animation of human walking: a survey. *The Journal of Visualization and Computer Animation*, 10(1), 39–54.
- Ngo, J. T. & Marks, J. (1993a). Physically realistic motion synthesis in animation. *Evol. Comput.*, 1(3), 235–268.
- Ngo, J. T. & Marks, J. (1993b). Spacetime Constraints Revisited. *Proceedings of ACM SIGGRAPH'1993* (pp. 343–350). Anaheim, USA.
- Oshita, M. & Makinouchi, A. (2001). A Dynamic Motion Control Technique for Human-like Articulated Figures. *Computer Graphics Forum*, 20(3), 192–202.
- Pina, A., Cerezo, E. & Seron, F. (2000). Computer animation: from avatars to unrestricted autonomous actors (a survey on replication and modelling mechanisms). *Computers and Graphics Journal*, 24, 297–311.

- Raibert, M. & Hodgins, J. K. (1991). Animation of Dynamic Legged Locomotion. *Computer Graphics*, 25(4), 349-358.
- Santillana. (2006). Proyectos en red. http://www.santillana.es/proyectosEnRed/secunda/html2/cie3/06_2.htm. Visitado em 15 de abril de 2006.
- Sims, K. (1994). Evolving Virtual Creatures. *Proceedings of ACM SIGGRAPH'1994* (pp. 15–22). Orlando, USA.
- Smith, R. (2000). Open dynamics engine. <http://ode.org/>. Visitado em 15 de abril de 2006.
- Stewart, D.E. & Trinkle, J.C. (1996). An Implicit Time-Stepping Scheme for Rigid Body Dynamics with Inelastic Collisions and Coulomb Friction. *International Journal of Numerical Methods in Engineering*, 39, 2673-2691.
- Van de Panne, M. (1996). Parameterized gait synthesis. *IEEE Computer Graphics and Applications*, 16(2), 40–49.
- Van de Panne, M., Fiume, E. & Vranesic, Z. (1990). Reusable motion synthesis using state-space controllers. *Proceedings of ACM SIGGRAPH'1990* (pp. 225–234). Dallas, USA.
- Van de Panne, M. & Fiume, E. (1993). Sensor-actuator networks. *Proceedings of ACM SIGGRAPH'1993* (pp. 335–342). Anaheim, USA.
- Van de Panne, M., Kim, R. & Fiume, E. (1994). Virtual wind-up toys for animation. *Proceedings of Graphics Interface'1994* (pp. 208–215). Alberta, CAN.
- Van de Panne, M. & Lamouret, A. (1995). Guided optimization for balanced locomotion. *Proceedings of the Eurographics Workshop on Computer Animation and Simulation'1995* (pp. 165–177). Maastricht, The Netherlands.
- Van de Panne, M. & Lee, C. (2003). Ski stunt simulator: Experiments with interactive dynamics. *Proceedings of the 14th Western Computer Graphics Symposium*. Banff, CAN.
- Webciência. (2006). http://www.webciencia.com/11_29snp.htm. Visitado em 15 de abril de 2006.

- Wilhelms, J. (1986). Virya - A motion control editor for kinematic and dynamic animation. *Proceedings of Graphics Interface'1986* (pp. 141-146). Vancouver, CAN.
- Wilhelms, J. & Barsky, B. (1985). Using Dynamic Analysis for the Animation of Articulated Bodies such as Humans and Robots. *Proceedings of Graphics Interface'1985* (pp. 97-104). Montreal, CA.
- Witkin, A. & Kass, M. (1988). Spacetime constraints. *Computer Graphics*, 22(4) 159–168.
- Zhao, P. (2004). Animation Palette: An Interface for Prototyping Dynamic Aerial Motions. M. Sc. thesis, The University of British Columbia.
- Zhao, P. & van de Panne, M. (2005). User Interfaces for Interactive Control of Physics-based 3D Characters. *Proceedings of ACM SIGGRAPH 2005 Symposium on Interactive 3D Graphics and Games* (pp. 87-94). Washington D.C., U.S.A.