

**UNIVERSIDADE FEDERAL DE SANTA CATARINA  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA  
ELÉTRICA**

**CURSO DE ROBÓTICA MÓVEL**

**PROF. EDSON ROBERTO DE PIERI**

Florianópolis, Março de 2002.

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Histórico . . . . .	1
1.2	Robô Móvel: uma definição . . . . .	5
1.3	Classificação . . . . .	6
1.3.1	Segundo a Anatomia . . . . .	6
1.3.2	Segundo o Tipo de Controle . . . . .	8
1.3.3	Segundo a Funcionalidade . . . . .	9
1.4	Desafios em Robótica Móvel . . . . .	12
1.4.1	Navegação . . . . .	12
1.4.2	Reconhecimento . . . . .	12
1.4.3	Aprendizado . . . . .	12
1.4.4	Cooperação . . . . .	13
1.4.5	Cognição . . . . .	13
1.5	Exercícios . . . . .	13
<b>2</b>	<b>Sensores</b>	<b>15</b>
2.1	Definição . . . . .	17
2.2	Classificação . . . . .	17
2.3	Critérios para Avaliação de Sensores . . . . .	19
2.3.1	Sensibilidade . . . . .	19
2.3.2	Linearidade . . . . .	19
2.3.3	Faixa . . . . .	19
2.3.4	Tempo de resposta . . . . .	19
2.3.5	Precisão . . . . .	20
2.3.6	Repetibilidade . . . . .	20

2.3.7	Resolução . . . . .	20
2.3.8	Tipo de Saída . . . . .	20
2.4	Sensores Internos . . . . .	21
2.4.1	Dead Reckoning . . . . .	21
2.5	Sensores Externos . . . . .	30
2.5.1	Sensores de Proximidade . . . . .	31
2.5.2	Sensores de Distância . . . . .	39
2.5.3	Sensores de Tato . . . . .	44
2.5.4	Sensores de Deslizamento . . . . .	46
2.5.5	Sensores de Força e Torque . . . . .	47
2.6	Resumo . . . . .	49
2.7	Exercícios . . . . .	49
<b>3</b>	<b>Atuadores</b>	<b>50</b>
3.1	Introdução . . . . .	50
3.2	Tipos de Atuadores . . . . .	51
3.2.1	Pneumáticos . . . . .	52
3.2.2	Hidráulicos . . . . .	54
3.2.3	Elétricos . . . . .	54
3.2.4	Outras tecnologias . . . . .	62
3.3	Exercícios . . . . .	65
<b>4</b>	<b>Controle</b>	<b>66</b>
4.1	Cinemática do Robô . . . . .	66
4.1.1	Introdução . . . . .	66
4.1.2	Cinemática Direta . . . . .	66
4.1.3	Cinemática Inversa . . . . .	70
4.2	Dinâmica do Robô . . . . .	73
4.3	Controle Cinemático . . . . .	76
4.3.1	Funções do controle cinemático . . . . .	76
4.3.2	Tipos de trajetórias . . . . .	78
4.3.3	Geração de trajetórias cartesianas . . . . .	80
4.3.4	Interpolação de trajetórias . . . . .	80

4.3.5	Amostragem de trajetórias cartesianas . . . . .	81
4.4	Controle Dinâmico . . . . .	82
4.4.1	Controle Monoarticular . . . . .	83
4.4.2	Controle PID . . . . .	84
4.4.3	Controle PID com pré-alimentação . . . . .	84
4.4.4	Controle Adaptativo . . . . .	85
4.5	Exercícios . . . . .	86
<b>5</b>	<b>Arquitetura de Robôs Móveis</b>	<b>87</b>
5.1	Introdução . . . . .	87
5.2	Definições e Classificações . . . . .	87
5.2.1	Arquiteturas Centralizadas e Distribuídas . . . . .	88
5.2.2	Arquiteturas Reativas, Deliberativas e Híbridas . . . . .	90
5.2.3	Arquiteturas baseadas em comportamento e arquiteturas baseadas em módulos funcionais . . . . .	93
5.3	Exemplos de arquiteturas para robôs móveis . . . . .	96
5.3.1	Arquitetura de subsunção . . . . .	96
5.3.2	Arquitetura AuRA . . . . .	100
5.4	Exercícios . . . . .	102
<b>6</b>	<b>Modelagem do Ambiente</b>	<b>103</b>
6.1	Mapas baseados em células . . . . .	104
6.1.1	Modelagem por enumeração . . . . .	104
6.1.2	Modelagem por <i>Quadrees</i> e <i>Octrees</i> . . . . .	105
6.2	Modelo Topológico . . . . .	106
6.2.1	Modelo Topológico de Lugares . . . . .	106
6.2.2	Modelo Topológico de Marcas . . . . .	107
6.3	Exercícios . . . . .	107
<b>7</b>	<b>Planejamento de Trajetória</b>	<b>108</b>
7.1	Introdução . . . . .	108
7.2	O Problema Básico . . . . .	108
7.3	O Espaço de Configuração . . . . .	109
7.4	Abordagens de Planejamento . . . . .	109

7.4.1	Abordagem <b>Roadmap</b> . . . . .	109
7.4.2	Decomposição em Células . . . . .	110
7.4.3	Campo Potencial . . . . .	111
7.5	Extensões ao Problema Básico . . . . .	111
7.5.1	Múltiplos Objetos Móveis . . . . .	112
7.5.2	Restrições Cinemáticas . . . . .	114
7.5.3	Incerteza . . . . .	116
7.5.4	Objetos Movíveis . . . . .	118
7.6	Complexidade Computacional . . . . .	119
7.7	Redução da Complexidade . . . . .	120
7.7.1	Projeção no espaço de Configuração . . . . .	120
7.7.2	Fatiamento no espaço de Configuração . . . . .	121
7.7.3	Simplificação da forma do objeto . . . . .	121
7.7.4	Focalizando a Atenção em um Subconjunto do Espaço de Trabalho . . . . .	122
7.8	Relacionamento com outros problemas . . . . .	122
7.8.1	Interação com Controle de Movimento em Tempo Real . . . . .	123
7.8.2	Interação com o Sensoreamento . . . . .	124
7.8.3	Interação com o planejamento em nível de tarefa . . . . .	125
<b>8</b>	<b>Glossário</b>	<b>127</b>

# Lista de Figuras

1.1	O primeiro robô móvel: Shakey . . . . .	3
1.2	Robôs da atualidade . . . . .	5
1.3	Ciclo de interação . . . . .	6
1.4	Classificação segundo a anatomia . . . . .	6
1.5	Exemplo de robôs aquáticos . . . . .	7
1.6	Exemplo de robô com rodas . . . . .	7
1.7	Exemplos de robôs com esteiras . . . . .	8
1.8	Exemplos de robôs com pernas . . . . .	9
1.9	Classificação segundo a funcionalidade . . . . .	9
1.10	Exemplo de robô de serviço: aspirador de pó . . . . .	10
1.11	Exemplo de robô de serviço de campo: colheitadeira autônoma . . . . .	11
1.12	Exemplo de robô de campo: Dante, para exploração de vulcões . . . . .	11
1.13	Exemplo de robô pessoal: o cãozinho Aibo . . . . .	11
2.1	Diferença entre repetibilidade e acurácia. (Fonte: [6]) . . . . .	20
2.2	Sensor Potenciômetro. (Adaptado de [20]) . . . . .	22
2.3	Sensor Sincronizador.(Adaptado de [20]) . . . . .	23
2.4	Resolvedor (Adaptado de [20]) . . . . .	24
2.5	Sensor LVDT. (Adaptado de [6]) . . . . .	26
2.6	Interruptor ótico. (Adaptado de [6]) . . . . .	27
2.7	Codificador ótico incremental(Adapado de [20]) . . . . .	28
2.8	Codificador ótico absoluto. (Adaptado de [20]) . . . . .	29
2.9	Acelerometro(Adaptado de [6]) . . . . .	30
2.10	Sensor táctil passivo.(Adaptado de [20]) . . . . .	32
2.11	Sensor ativo em forma de antena. (Adaptado de [20]) . . . . .	32

2.12	Amortecedor Táctil. (Adaptado de [6]) . . . . .	33
2.13	Sensor Reed Switich magnético.(Adaptado de [20]) . . . . .	34
2.14	Sensor de efeito Hall com imã permanente (a) linhas de fluxo na ausência de um corpo ferro-magnético nas proximidades do sensor; (b) linhas de fluxo quando há um corpo ferro-magnético nas proximidades. (Adaptado de [6]) .	35
2.15	Esquema de um sensor de proximidade por micro-ondas. (Adaptado de [20])	37
2.16	Sensor ótico oposto. (Adaptado de [20]) . . . . .	38
2.17	Sensor ótico retrorefletor. (Adaptado de [20]) . . . . .	38
2.18	Sensor ótico Difuso. (Adaptado de [20]) . . . . .	39
2.19	Sistemas de triangulação determinam a distância B para o alvo colocado no ponto $P_3$ , medindo os ângulos $\phi$ e $\theta$ nos pontos $P_1$ e $P_2$ .(Adaptado de [20]) . .	40
2.20	Esquema de Triangulação Ativa.(Adaptado de [20]) . . . . .	41
2.21	Esquema de Triangulação passiva ou triangulação estereoscópica passiva utilizando câmeras de vídeo ou sensores óticos ou de micro-ondas.(Adaptado de [20]) . . . . .	42
2.22	Sensor de tato analógico.(Adaptado de [20]) . . . . .	45
2.23	Pele Artificial. (Adaptado de [6]) . . . . .	45
3.1	Grande variedade de dispositivos hidráulicos e pneumáticos . . . . .	51
3.2	Exemplo de um robô com acionamento pneumático . . . . .	52
3.3	Outro robô industrial que utiliza o ar comprimido como energia . . . . .	52
3.4	Cilindro pneumático em alumínio . . . . .	53
3.5	Outros modelos . . . . .	53
3.6	Exemplos de pistões hidráulicos . . . . .	54
3.7	Outros modelos . . . . .	54
3.8	Motor AC trifásico comum . . . . .	56
3.9	Típico motor AC monofásico . . . . .	56
3.10	Corte parcial de um motor DC . . . . .	57
3.11	Exemplos de motores DC . . . . .	58
3.12	Motor relutância variável . . . . .	60
3.13	Imã permanente . . . . .	60
3.14	Modelo híbrido . . . . .	60
3.15	Exemplo de um motor de passo . . . . .	61

3.16	Outro modelo . . . . .	61
3.17	Servomotor . . . . .	61
3.18	Fibras de um polímero eletroativo . . . . .	62
3.19	Protótipo de um robô “alado” . . . . .	63
3.20	Exemplo de um motor piezoelétrico . . . . .	63
4.1	Diagrama de relação entre cinemática direta e inversa . . . . .	67
4.2	Exemplo de um robô planar com dois graus de liberdade . . . . .	68
4.3	Matriz jacobiana direta e inversa . . . . .	73
4.4	Funcionamento do controle cinemático . . . . .	77
4.5	Atuação de um controle cinemático em um robô com dois graus de liberdade	78
4.6	Diagrama de blocos que representa o comportamento dinâmico de uma articulação . . . . .	83
4.7	Esquema de algoritmo de controle baseado em PID . . . . .	84
4.8	Esquema de algoritmo de controle baseado em PID com pré-alimentação . . .	85
5.1	Esquema genérico de uma Arquitetura Reativa . . . . .	91
5.2	Planejamento hierárquico usado em arquiteturas deliberativas . . . . .	92
5.3	Um exemplo simples da arquitetura de subsunção . . . . .	99
5.4	Arquitetura híbrida AuRA . . . . .	101
6.1	Modelagem por enumeração . . . . .	105
6.2	Decomposição do ambiente utilizando <i>quadtrees</i> . . . . .	105
6.3	Modelo topológico . . . . .	106
7.1	O caminho $\tau$ produzido pelo planejador é a entrada para o gerador de trajetória que determina as dependências de tempo dos parâmetros de configuração. A saída do gerador, as configurações desejadas $q_d$ como função do tempo, alimentam o controlador. O controlador computa a força $F$ a ser exercida por cada atuador a partir dos desvios da configuração atual $q_d$ , con- forme medida pelos sensores, relativa à $q_d$ . . . . .	124
8.1	Obtenção de um C-Obstáculo . . . . .	129

# Capítulo 1

## Introdução

Quando falamos em robôs várias imagens de filmes de ficção científica voltam à mente. Filmes como o *Jornada nas Estrelas* e *O Exterminador do Futuro* nos apresentam um fenótipo do que o desenvolvimento tecnológico visa alcançar: robôs com capacidades humanas de aprendizado e raciocínio.

Ainda estamos longe de criar máquinas autônomas, adaptáveis e inteligentes, mas avanços nas áreas de inteligência artificial e controle e o desenvolvimento de novos tipos de materiais, nos têm permitido o desenvolvimento de robôs para os mais variados fins.

### 1.1 Histórico

O desejo de construir máquinas que imitem os movimentos de seres humanos e animais acompanha as civilizações desde a antigüidade. Os egípcios construíaam homens mecânicos em forma de estátuas com articulações móveis e ajustáveis. Da Grécia antiga, tem-se a descrição de estátuas munidas de tubos falantes e a partir do século IV a.C., marionetes acionadas por sistemas de polias e pesos.

No século XVI, os então chamados *autômatos*, em geral bonecos mecânicos acionadas por sistemas de cordas que imitavam algumas atividades humanas, chamavam a atenção para a possibilidade de descrever a natureza humana. Em 1589, Bernardino Baldi, abade de Guastalla, publicou um ensaio contendo reflexões sobre a natureza e potencialidades dos autômatos [38].

No século XVII, marcados pelo pensamento dualista de Descartes com sua célebre frase *Penso, logo existo*, muitos filósofos passaram a se questionar sobre os limites do mecanicismo:

afinal, se autômatos eram capazes de descrever a natureza e os animais, seriam adequados para explicar a natureza humana?

A construção de autômatos cresceu no século XVIII, sendo que alguns se tornaram bastante conhecidos, como o pato mecânico de Jacques de Vaucanson, em 1750 e os três bonecos - um escritor, um desenhista e uma tocadora - construídos pelos irmãos Droz na Suíça, em 1733 [38].

No final do século XVIII a revolução industrial e a conseqüente adoção do modelo mecanicista de Taylor e Fayol proporcionaram a mudança definitiva das formas de produção, levando o uso das máquinas para a indústria: iniciava-se a era da produção em larga escala.

A palavra robô tem sua origem na palavra tcheca *robota*, que significa *trabalho escravo, monótono* e foi utilizada pela primeira vez em 1921, pelo dramaturgo Karel Capeck no conto de ficção *Rossum's Universal Robots*.

Em 1936, Charles Chaplin satiriza o processo de automação pesada em seu filme *Tempos Modernos*.

Em 1939, Issac Asimov, considerado o pai da robótica, popularizou o termo robô em seu livro *I Robot*, no qual estabelece as famosas três leis da robótica, que são:

- Um robô jamais deve causar dano a um ser humano nem, através de omissão, permitir que isto aconteça.
- Um robô deve sempre obedecer o ser humano, a menos que isto entre em conflito com a primeira lei.
- Um robô deve proteger-se de danos, a menos que isto entre em conflito com a primeira ou a segunda lei.

Após a invenção do transistor em 1948, os robôs passaram a ser controlados por computadores. A primeira patente para um robô industrial controlado por computador foi registrada em 1954 por George Devol. Devol criou uma memória computadorizada e um sistema de controle chamado *Universal Automation*. Mais tarde, co-fundou a companhia de robôs industriais - UNIMATION. Em 1961 o primeiro robô UNIMATE, que usava comando numérico programável, foi instalado na linha de montagem da General Motors: estava iniciada era da automação industrial.

Em 1963, surge o primeiro sistema para visão de robôs. E em 1966, o primeiro robô de pintura, cuja trajetória era treinada e armazenada.

Os robôs móveis surgiram em 1968 carregando conceitos da mecânica e da robótica fixa. A princípio, com o avanço nas áreas de sensores, processamento de imagens e inteligência artificial, dotar um robô móvel com capacidades para atuar em ambientes dinâmicos parecia ser algo simples, porém, logo percebeu-se a grande complexidade envolvida no desenvolvimento de sistemas móveis que fossem robustos e adaptáveis.

Em 1969, Nilsson [34] descreve o primeiro sistema robótico móvel que utiliza quadrees<sup>1</sup> para representar o ambiente e grafos de visibilidade para o planejamento da trajetória. Em 1972, o robô Shakey - fruto do trabalho de Nilsson na SRI International, hoje Instituto de Pesquisa de Stanford - foi introduzido como o primeiro robô móvel controlado por inteligência artificial.



Figura 1.1: O primeiro robô móvel: Shakey

Problemas relacionados com as questões de movimentos rotacionais e translacionais tornavam crítica a modelagem do ambiente. Em 1983, Lozano-Pérez [31] introduziu a idéia de uma “região de incerteza”, criada através do crescimento dos obstáculos. Assim, utilizando grafos de visibilidade para o planejamento de trajetória, o robô poderia ser tratado como um simples ponto no espaço de configuração. Este método foi o primeiro método exato aplicado ao problema do planejamento de trajetória.

Ainda em 83, Brooks [12] introduziu o método *freeway* como uma alternativa à modelagem do espaço livre e ao planejamento de trajetória, utilizando o conceito de cones generalizados.

Estes métodos, baseados em um mapa do ambiente, chamados de arquiteturas deliberativas ou planejadas, sofriam sérias restrições com relação a criação e manutenção do mapa do ambiente. Para minimizar os problemas advindos do mundo real, foram

---

<sup>1</sup>Estrutura em forma de árvore que é gerada através da decomposição de um ambiente bidimensional pelo refinamento sucessivo das células.

assumidas simplificações, como o ambiente estático e totalmente conhecido. Com estas simplificações, o problema é contornado, mas a aplicação de robôs permanece restrita a ambientes imutáveis.

Buscando generalizar as aplicações de robôs móveis e inspirando-se no comportamento de insetos, Brooks [13] introduz, em 1986, uma arquitetura reativa, denominada *arquitetura de subsunção* (do inglês *Subsumption*), na qual o robô age baseando-se na leitura de seus sensores. Esta arquitetura baseia-se na decomposição da inteligência em comportamentos individuais, gerando módulos que coexistem e cooperam para a emergência de comportamentos mais complexos. Esta arquitetura é um marco nas pesquisas de métodos reativos e ascendeu novamente a esperança de construir robôs móveis aptos a interagir em ambientes dinâmicos.

Ainda em 86, Khatib [30, 27] introduziu o método dos campos potenciais, no qual o robô transforma-se em uma partícula sob a influência de campos eletromagnéticos, constituídos pelos obstáculos e pelo ponto objetivo. Neste mesmo ano a Honda começa a trabalhar no desenvolvimento de seu robô humanoíde (figura 1.8 (b)).

Entre 87 e 89, Arkin escreveu vários artigos descrevendo uma arquitetura reativa baseada em esquemas motores, que mais tarde amadureceu em uma arquitetura híbrida, a qual denominou *AuRA - Autonomous Robot Architecture*. Em 89, Arkin [2] compara as abordagens planejada e reativa e salienta as vantagens da arquitetura híbrida.

Em 1990, Kumpel [28] apresenta um artigo descrevendo o projeto *MARIE - Mobile Autonomous Robot in an Industrial Environment*. Este projeto integra mapas geométricos e topológicos e usa um método hierárquico para navegação, onde, globalmente, utiliza mapas para encontrar o caminho e sensores para desvio local dos obstáculos.

Ainda em 90, Brooks [14] propõe alguns melhoramentos à arquitetura de subsunção e apresenta uma série de robôs, desenvolvidos no MIT, que utilizam esta arquitetura.

Dois anos mais tarde, Mataric [33] ressalta a necessidade de alguma representação do ambiente para dotar o robô de capacidades mais elaboradas do que somente a navegação aleatória e propõe um método reativo, mas que utiliza um mapa, construído através das marcas (*landmarks*) detectadas no ambiente, dentro de uma arquitetura de subsunção. O mapa é atualizado sempre que o robô detecta mudanças no ambiente.

No mesmo ano, Zelinsky [41] propõe um método simples para mapeamento do ambiente em tempo de execução, utilizando sensores de contato. Por este método, o ambiente é

mapeado em uma quadtree, onde a menor célula tem o tamanho do diâmetro do robô. O caminho é, inicialmente, dado por uma linha reta e durante a execução desta trajetória, são obtidas leituras dos sensores utilizadas para atualizar a estrutura da quadtree sobre a qual o caminho é replanejado.

Em 1994, a NASA em conjunto com a Carnegie Mellon University e Observatório de Vulcões do Alaska desenvolveram um robô hexápode chamado Dante (figura 1.12) para inspecionar vulcões ativos, em julho de 1997 o Sojourner (figura 1.2 (b)) toca o solo de Marte e em janeiro de 2000, a Carnegie Mellon apresenta um robô com quatro rodas para busca de amostras de meteoritos na Antártica (figura 1.2 (a)). O robô coleta as amostras de pedras e as classifica utilizando técnicas de inteligência artificial.



(a) Robô para busca de amostras de meteoritos na Antártica

(b) Sojourner

Figura 1.2: Robôs da atualidade

Enfim, nos últimos 10 anos, a robótica móvel desenvolveu-se muito e hoje temos os mais variados tipos de robôs, para os mais variados tipos de tarefas.

## 1.2 Robô Móvel: uma definição

Um robô móvel é um dispositivo mecânico montado sobre uma base não fixa, que age sob o controle de um sistema computacional, equipado com sensores e atuadores que o permitem interagir com o ambiente [32].

A interação com o ambiente se dá através de *ciclos de percepção-ação* que consistem em três passos fundamentais:

- Obtenção de informações acerca do ambiente através dos *sensores*;

- Processamento das *informações* obtidas e seleção de ações que serão executadas;
- Execução das ações planejadas através do acionamento dos *atuadores*.

O robô realiza diversos ciclos de percepção-ação, modificando o estado do ambiente em busca da realização da tarefa [36].

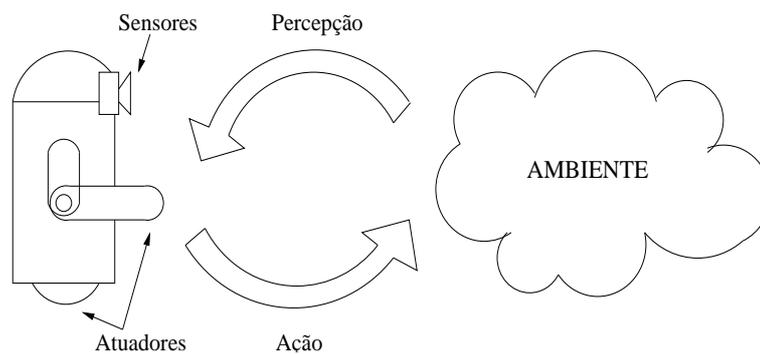


Figura 1.3: Ciclo de interação

## 1.3 Classificação

Diversas taxonomias tem sido utilizadas para classificar robôs móveis, contudo, não há, ainda, nenhuma que seja definitiva. De uma forma geral, podemos agrupar os robôs existentes de acordo com três aspectos: anatomia, tipo de controle e funcionalidade.

### 1.3.1 Segundo a Anatomia

Quanto a anatomia os robôs podem ser classificados em três grandes grupos (figura 1.4): os robôs aéreos, os aquáticos e os terrestres.

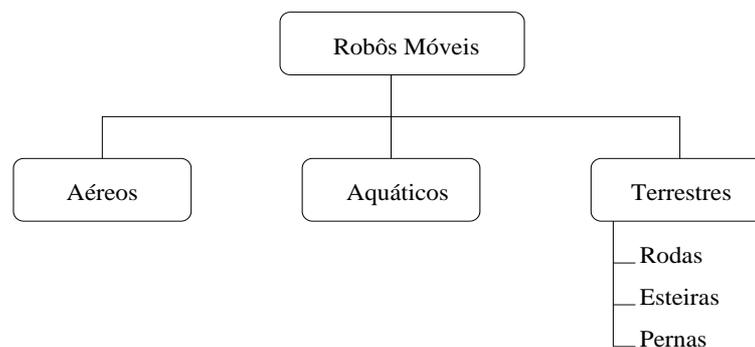


Figura 1.4: Classificação segundo a anatomia

Os robôs aéreos, geralmente são aeromodelos ou LTAs (*Lighter-Than-Air*) equipados com câmeras de vídeo e utilizados para inspeção de grandes áreas. Os aquáticos, em geral são plataformas equipadas com propulsores e balões de ar que permitem ao robô permanecer a alguns metros do fundo do mar, porém, podem também possuir patas (figura 1.5).

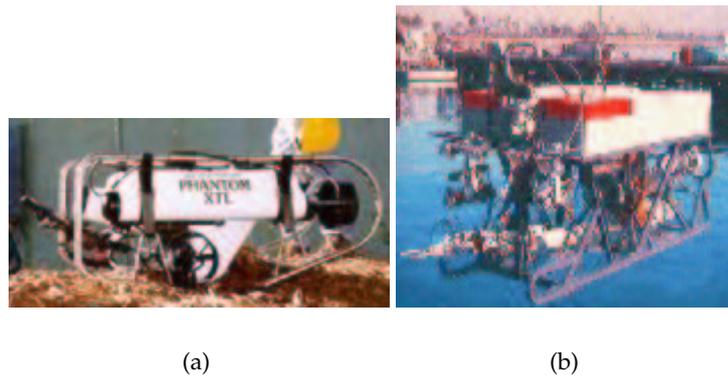


Figura 1.5: Exemplo de robôs aquáticos

Já os robôs terrestres são os mais populares e podem utilizar três tipos diferentes de atuadores: rodas, esteiras ou pernas.

- Robôs com Rodas - estes são os mais simples, pois não necessitam de um hardware tão complexo quanto os robôs com esteiras e pernas, além disso, é fácil transformar brinquedos em robôs. A principal desvantagem no uso de rodas é que, em terrenos irregulares, o desempenho pode não ser satisfatório. Em geral, a roda do robô deve possuir raio igual ou maior aos obstáculos que ele irá transpor (figura 1.6).

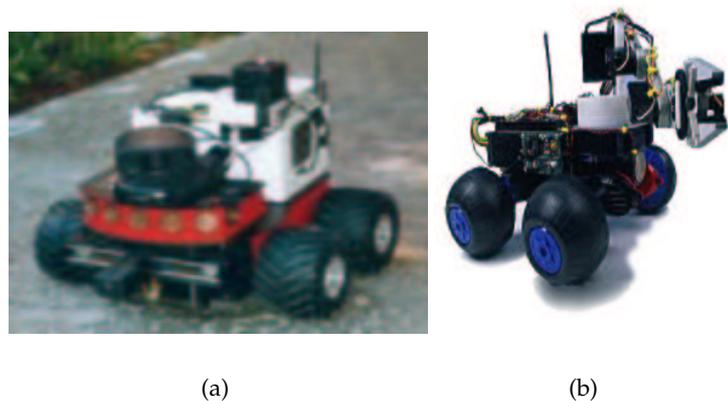


Figura 1.6: Exemplo de robô com rodas

- Robôs com Esteiras - São mais utilizados em robôs que atuam em ambientes irregulares, como por exemplo com solo fofo e pedras. A grande desvantagem está na dissipação de energia causada pelo movimento de giro da própria esteira e pelas rodas no seu interior (figura 1.7).

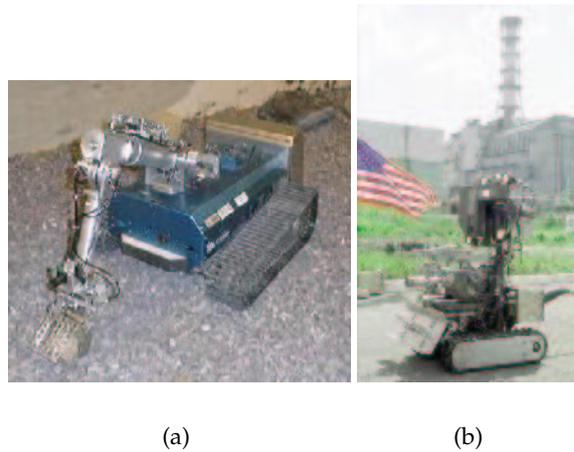


Figura 1.7: Exemplos de robôs com esteiras

- Robôs com Pernas - Este tipo de robô é utilizado em terrenos acidentados, com subidas íngremes, ou em ambientes específicos, como por exemplo, ambientes com escadas. A grande dificuldade deste tipo de robô está no desenvolvimento do projeto para controle das pernas, que têm, no mínimo, dois graus de liberdade. Há também o fator custo associado a este tipo de projeto, pois cada atuador utiliza pelo menos dois motores (figura 1.8) .

É importante ressaltar que estas são categorias difusas, uma vez que podem ser combinadas características de dois grupos na construção de um robô para uma tarefa específica.

### 1.3.2 Segundo o Tipo de Controle

Quando classificados segundo o tipo de controle, os robôs podem ser separados em três categorias:

- Teleoperados - Onde o operador realiza todos os movimentos que o robô deve fazer;
- Semi-Autônomos - O operador indica o macro comando a ser executado e o robô o faz sozinho;



(a) Robô hexápode

(b) Robô humanóide

Figura 1.8: Exemplos de robôs com pernas

- Autônomos - O robô realiza a tarefa sozinho, tomando suas próprias decisões baseando-se nos dados obtidos do ambiente.

### 1.3.3 Segundo a Funcionalidade

Ao agruparmos os robôs segundo sua funcionalidade, percebemos quatro grupos: robôs industriais, robôs de serviço, robôs de campo e robôs pessoais. Contudo, há uma sobreposição entre os três primeiros (figura 1.9), devido a diferença dos ambientes onde atuam e a necessidade de maior autonomia.

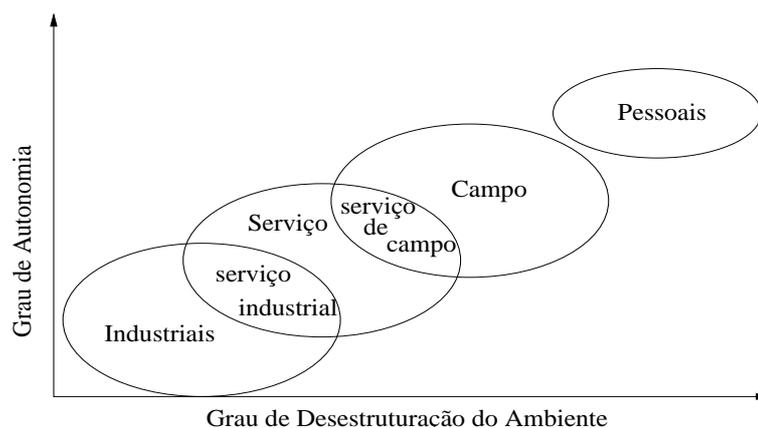


Figura 1.9: Classificação segundo a funcionalidade

- Robôs Industriais - São os robôs utilizados em linhas de produção. Estes robôs recebem tarefas determinadas *a priori* na forma de uma seqüência explícita e ações e executam este programa automaticamente. O ambiente é completamente estruturado e ajustado

para a execução da tarefa. Neste ambiente, o robô tem conhecimento exato da sua posição e da posição dos objetos. Em geral, robôs móveis industriais são plataformas móveis utilizadas para tarefas pesadas, como transporte de materiais e produtos finais em sistemas de manufatura. Esses robôs são denominados AGVs (*Automated Guided Vehicles*) e são programados para seguir linhas desenhadas no chão.

- Robôs de Serviço Industrial - Este tipo de robô possui as características de um robô de serviço, porém atua em um ambiente completamente estruturado.
- Robôs de Serviço - São os robôs móveis utilizados para serviços gerais (figura 1.10). O ambiente é estruturado e o robô possui um modelo deste ambiente, que é conhecido previamente, porém possui certa autonomia, pois processa informação sensorial, para atuar em situações imprevistas, como desviar de uma pessoa ou objeto. Estes robôs recebem macro-comandos da tarefa que devem realizar e são utilizados para tarefas de limpeza em geral (pisos, dutos de ar, metrô, etc.), em sistemas de vigilância e no transporte de materiais leves (correspondências internas, material hospitalar, etc.).



Figura 1.10: Exemplo de robô de serviço: aspirador de pó

- Robôs de Serviço de Campo - Estes robôs são robôs de serviço que atuam em ambientes externos que podem ser previamente modelados ou não. Em geral, caso haja um modelo, este é precário e há a necessidade do processamento sensorial para complementar o modelo existente. Estes robôs são utilizados na realização de tarefas agrícolas (figura 1.11) e para navegação em auto-estradas.
- Robôs de Campo - estes robôs trabalham em ambientes não estruturados, pouco conhecidos e geralmente perigosos (figura 1.12). As principais atividades destes robôs



Figura 1.11: Exemplo de robô de serviço de campo: colheitadeira autônoma

são: exploração (espacial, de cavernas, vulcões), mineração e limpeza de acidentes nucleares.



Figura 1.12: Exemplo de robô de campo: Dante, para exploração de vulcões

- Robôs Pessoais - são os robôs vendidos em prateleiras, que não desenvolvem tarefas específicas, mas interagem com os humanos e aprendem a localizar-se no ambiente (figura 1.13).



Figura 1.13: Exemplo de robô pessoal: o cãozinho Aibo

## 1.4 Desafios em Robótica Móvel

Construir um robô móvel envolve uma série de problemas. É necessário lidar com erros de posição e orientação, ruídos nas leituras dos sensores, problemas com a carga das baterias, erros de programação, etc.

Como é possível perceber, a tarefa de construir um robô com características humanas é um desafio incrivelmente difícil. Seres humanos possuem características muito difíceis de serem modeladas. Nosso sistema de visão é apurado, nossa reação a uma situação inesperada é quase que instantânea, nossos movimentos são coordenados inconscientemente para manter o equilíbrio do corpo. Nosso sistema de controle de energia opera de forma bastante otimizada. Todas estas características nos tornam “máquinas” perfeitas.

Quando desenvolvemos robôs, procuramos nos espelhar em nossas habilidades e essa busca pelo desenvolvimento de máquinas com características humanas nos apresenta uma série de desafios.

### 1.4.1 Navegação

O grande desafio da navegação de robôs móveis é criar formas de representação do ambiente. Até que ponto modelos do ambiente precisam ser refinados e livres de erros. Até que ponto podemos utilizar navegação reativa e ainda assim maximizar o desempenho do robô.

### 1.4.2 Reconhecimento

Outro grande desafio é fazer com que o robô reconheça o ambiente no qual está inserido. Isto envolve problemas com a complexidade computacional do processamento de imagem e dados, além das dificuldades encontradas ao tratar diferenças de luminosidade e ruído nos dados advindos dos sensores.

### 1.4.3 Aprendizado

Ao tornar o robô mais complexo, adicionando a ele novos sensores e atuadores, estamos também acrescentando mais complexidade ao trabalho de programação de ações do robô, além de dificultar a tarefa de calibração dos sensores e coordenação dos movimentos. O

desenvolvimento de algoritmos de aprendizado permite que o robô calibre seus sensores e desempenhe a sua tarefa de forma mais autônoma e adaptável.

#### 1.4.4 Cooperação

Uma outra linha de pesquisa argumenta a possibilidade de construir robôs pequenos e baratos que juntos possam desempenhar tarefas complexas, cooperando uns com os outros. Assim como em um formigueiro.

#### 1.4.5 Cognição

O maior de todos os desafios é explorar os limites da inteligência artificial. Até que ponto podemos criar novas formas para armazenar conhecimento e utilizar este conhecimento para a tomada de decisão dotando o robô de capacidades cognitivas de alto nível que o permitam total autonomia com relação ao meio.

### 1.5 Exercícios

1. Procure saber mais sobre o primeiro robô móvel (Shakey).
2. O que você entende por “expansão dos obstáculos”? Considere um robô com base circular.
3. Quanto a modelagem do ambiente, os métodos podem ser exatos e aproximados. O que você entende por métodos exatos e aproximados de modelagem do ambiente e quais as conseqüência da escolha de um ou outro para o planejamento da trajetória?
4. Observando o histórico descrito no início do capítulo, é possível encontrar três tipos de “arquitecturas” de robôs móveis, que dizem respeito a forma como este interage com o ambiente. Quais são estes três tipos? Busque caracterizá-los segundo algum critério e compare-os com o comportamento humano e de animais com relação ao desenvolvimento de uma tarefa.
5. Quais são os principais tipos de sensores utilizados em robótica móvel?
6. Quais são os principais tipos de atuadores utilizados em robótica móvel?

- 
7. Na sua opinião, qual a vantagem existente em construir robôs com base circular para atuar em ambientes fechados e estruturados? Ilustre uma situação onde um robô com base circular consegue cumprir sua tarefa em detrimento a um robô com base retangular.

## Capítulo 2

# Sensores

O uso tradicional da robótica tem sido as aplicações industriais, que tradicionalmente utilizam robôs fixos para manufatura. Os benefícios do uso da robótica na indústria são já bem conhecidos, para citar alguns: Melhoria na qualidade do produto, aumento da eficiência, confiabilidade e redução dos custos de produção. A introdução da mobilidade na robótica fez surgir toda uma nova gama de aplicações, além das tradicionais aplicações industriais. Essas aplicações incluem: robôs móveis para combate à incêndios, manipulação de material bélico, explosivos, material tóxico, transporte de materiais e cargas perigosas, patrulhamento e vigilância, etc.

A construção de robôs móveis, apesar de sua ampla gama de aplicações, representa um desafio tecnológico. Vários problemas precisam ser resolvidos pelos projetistas de robôs para que os robôs possam ser construídos de forma a estarem adaptados à tarefa a ser executada. As principais dificuldades encaradas pelos projetistas devem-se principalmente à imaturidade tecnológica, ainda incapaz de fornecer o suporte adequado ao seu desenvolvimento. Estas dificuldades tecnológicas podem ser resumidas em três áreas principais:[4, 20]

1. Recursos Computacionais
2. Comunicações
3. Sensores

As duas primeiras áreas têm sido bastante desenvolvidas nos últimos anos, com o desenvolvimento exponencial do poder computacional dos processadores, e com o grande desenvolvimento da tecnologia das redes de computadores, dos protocolos de comunicação, e da tecnologia de computação móvel.

A categoria dos sensores, no entanto, somente agora tem percebido um notável crescimento tecnológico. Os principais problemas enfrentados no desenvolvimento da robótica móvel surgem diretamente da necessidade de interação com objetos físicos e entidades do ambiente. Esta interação consiste principalmente na localização e identificação destas entidades e objetos enquanto o robô move-se (navega) neste meio ambiente. Este processo precisa ser realizado rapidamente, e com a máxima precisão e acurácia. No projeto de tais sensores, deve-se levar em conta as seguintes considerações:[9, 20]

- **Campo de Visão:** Deve ser amplo o bastante e com suficiente profundidade para adequar-se à sua aplicação.
- **Range Capability:** (Capacidade Alcance) O alcance mínimo de detecção, bem como o alcance máximo efetiva devem ser adequados ao uso pretendido do sensor.
- **Acurácia e Resolução:** Devem estar de acordo com as necessidades da tarefa dada.
- **Habilidade para detectar todos os objetos no ambiente:** Várias condições ambientes podem interferir e acrescentar ruído às informações captadas pelos sensores. Os mesmos devem ser capazes de minimizar tais interferências.
- **Operação em Tempo-Real:** O sensor deve ser capaz de fornecer dados em tempo real a velocidades que sejam compatíveis com a velocidade de sua plataforma receptora (robô móvel, veículo-robô, etc.)
- **Dados Concisos e de fácil interpretação:** O formato de saída de um sensor deve ser realístico do ponto de vista dos requisitos de processamento, visando reduzir ao máximo o pré-processamento da informação fornecida pelo sensor.
- **Redundância:** O sistema sensor deve sofrer uma degradação lenta e suave, e não ficar totalmente incapacitado devido à perda de um elemento sensor. Uma capacidade *multimodal* seria desejável para assegurar a detecção de todos os alvos, bem como para aumentar o nível de confiança de sua saída.
- **Simplicidade:** O sistema deve ser de baixo custo e modular, para facilitar a manutenção e permitir atualizações evolucionárias, portanto, não deve estar restrito à um tipo de hardware específico.

- **Consumo de Força:** O consumo de força deve ser mínimo, para poder ser mantida uma boa relação com os recursos limitados a bordo de robôs móveis e veículos-robôs.
- **Tamanho:** O peso e o tamanho devem ser o menor possível, de acordo com os objetivos pretendidos para o veículo ou robô.

Estas considerações não representam regras rígidas, mas o projetista deve sempre tê-las em mente antes de iniciar o projeto de um novo sistema sensor para um robô móvel.

## 2.1 Definição

Sensores são dispositivos projetados para quantificar ou detectar parâmetros especificados por meio de elementos transdutores<sup>1</sup> [24, 20, 6].

## 2.2 Classificação

Do ponto de vista da robótica, os sensores podem ser divididos em duas grandes categorias, de acordo com a sua utilização em um robô [15, 20, 6]:

1. **Sensores Internos:** São usados para medir a posição e velocidade ou aceleração das juntas ou extremidades de um robô manipulador, ou das rodas ou patas de um robô móvel.
2. **sensores Externos:** São utilizados para monitorar o próprio robô e a sua relação com o mundo ao seu redor, bem como a realização da tarefa que lhe foi destinada.

Esta divisão pode ainda ser expandida em diversas subclasses, conforme mostra a taxonomia abaixo:

1. Sensores internos
  - (a) Sensores de posição
    - i. Não-óticos
      - A. Potenciômetros
      - B. Sincronizadores

---

<sup>1</sup>Transdutores são elementos que tem a função de converter uma dada magnitude física em outra.

- C. Resolvedores
- D. Escala Indutiva Linear (Inductosyn)
- E. Transformadores Diferenciais (LVDT e RVDT)
- ii. Óticos
  - A. Interruptores Óticos
  - B. Codificadores Óticos
- (b) Sensores de Velocidade
  - i. Tacômetros
  - ii. Codificadores Óticos
- (c) Sensores de Aceleração

## 2. Sensores Externos

- (a) Sensores de Proximidade
  - i. Sensores de Contato
  - ii. Sensores de Não-contato
    - A. Sensores óticos de Proximidade
    - B. Sensores Indutivos
    - C. Sensores de Efeito Hall
    - D. Sensores Capacitivos
    - E. Sensores de Proximidade Ultra-sônicos
- (b) Sensores de Distância
  - i. Por Triangulação
  - ii. Por luz estruturada
  - iii. Por Tempo de Vôo
    - A. Laser
    - B. Pulso de Luz
    - C. Deslocamento de fase
    - D. Ultra-som (pulso-eco)

## 3. Sensores de Tato

- (a) Binários
- (b) Analógicos
- (c) Pele artificial

#### 4. Sensores de Deslizamento

- (a) Sensores de Força e Torque
  - i. Por monitoração da corrente do motor
  - ii. Usando um LVDT

## 2.3 Critérios para Avaliação de Sensores

### 2.3.1 Sensibilidade

É definida como a razão entre a taxa de mudança dos valores de saída pela mudança dos valores de entrada. Como exemplo, seja um sensor qualquer, em que um movimento de 0,001mm na entrada provoca uma mudança de 0,02V na sua saída. Neste caso dizemos que a sensibilidade do sensor é de  $\frac{0.02V}{0.001mm} = 20V/mm$

### 2.3.2 Linearidade

É a medida da constância da taxa de saída com relação à entrada. Um sensor é dito linear se a variação da entrada é constante com a variação da saída, ou seja se a relação entre ambos é dada por uma função linear.

### 2.3.3 Faixa

É a medida da diferença entre o valor mínimo e o valor máximo medido.

### 2.3.4 Tempo de resposta

É o tempo decorrido para que uma mudança nas entradas seja percebida como uma mudança estável nas saídas. Em alguns sensores a saída oscila por um certo tempo antes de alcançar um valor estável. O tempo de resposta é medido desde o início da mudança na entrada até a estabilização da saída.

### 2.3.5 Precisão

É a medida da diferença entre os valores medidos e reais.

### 2.3.6 Repetibilidade

É a medida da diferença entre duas medidas sucessivas sob as mesmas condições. Sensores podem ter alta repetibilidade, mas pouca precisão.

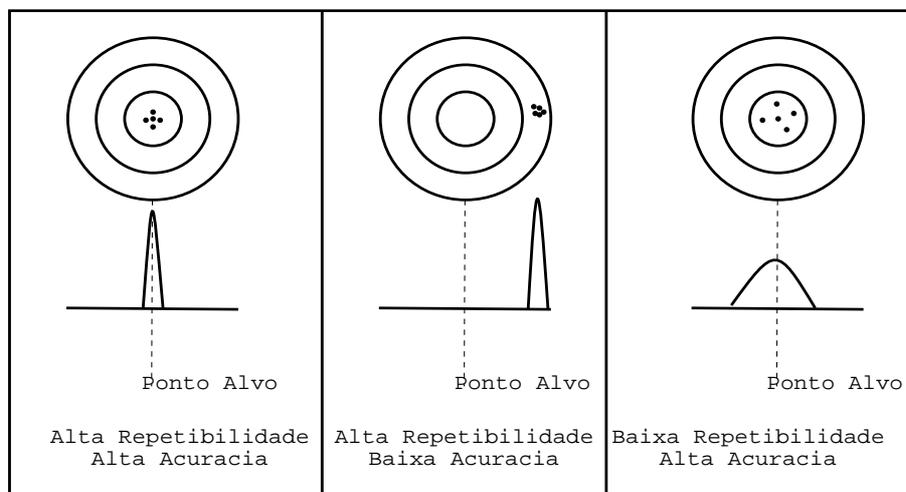


Figura 2.1: Diferença entre repetibilidade e acurácia. (Fonte: [6])

### 2.3.7 Resolução

Mede o número de medidas de valores diferentes possíveis dentro de uma faixa de mínimo e máximo.

### 2.3.8 Tipo de Saída

A saída de um sensor pode ser dada de várias formas, por exemplo, através de:

- movimento mecânico,
- tensão,
- corrente,
- pressão,
- nível hidráulico,

- intensidade luminosa, etc.

## 2.4 Sensores Internos

Os sensores internos são geralmente designados para as tarefas conhecidas como “*Dead Reckoning*” (contagem deduzida). Também são conhecidos como sensores de posição, ou sensores não óticos ou ainda como sensores de odometria. A seção seguinte vê com mais detalhes o “*Dead Reckoning*” e os sensores que são geralmente usados para esta tarefa [20].

### 2.4.1 Dead Reckoning

A expressão “*Dead Reckoning*” deriva da expressão “*Deduced Reckoning*” (Contagem Deduzida). *Dead Reckoning* é um procedimento para determinação da localização atual de um veículo através da análise das informações sobre sua velocidade e curso conhecido. Baseia-se na idéia de que se o curso é conhecido, podemos determinar a posição do veículo neste curso através da observação da velocidade de seu deslocamento, ou ainda, da observação da distância percorrida da origem até o momento. É um procedimento análogo ao que fazemos quando dirigimos de uma cidade à outra, consultando um mapa e o odômetro do carro, por exemplo.

A implementação mais simplista de *dead reckoning* é conhecida como *odometria*, onde a posição do veículo ao longo do caminho é derivada diretamente de algum tipo de odômetro embarcado. Outra forma de odometria bastante comum envolve codificadores óticos diretamente acoplados à armadura do motor ou aos eixos das rodas. Em algumas aplicações exteriores utiliza-se algumas vezes sensores magnéticos de proximidade para perceber a presença de pequenos magnetos permanentes que são acoplados ao veículo (muito usados na indústria automotiva para aferição de velocidade). Mais detalhes sobre odometria, veja [8, 11, 9, 24].

### Sensores de Odometria (ou de posição) não óticos

Os tipos mais comuns de sensores aplicados à odometria são: potenciômetros, resolvidores, e codificadores óticos. A seguir veremos cada um deles com um pouco mais de detalhe:

**Potenciômetros** São geralmente usados como sensores de posicionamento rotacional em aplicações de média acurácia e baixa velocidade que não envolvam rotação contínua (p.ex., são usados para sentir o posicionamento da cabeça de um robô, ou de suas articulações).

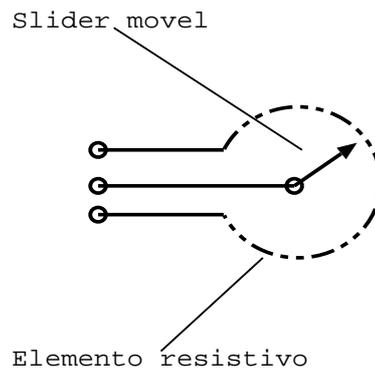


Figura 2.2: Sensor Potenciômetro. (Adaptado de [20])

Seu princípio de funcionamento é o mesmo do divisor de tensão de resistência variável: aplica-se a tensão de referência nos dois polos de entrada do potenciômetro e então lê-se a tensão de saída, a qual será alterada em função do comprimento da resistência(2.2). A tensão de saída de um potenciômetro é mostrada abaixo:

$$V_0 = V_{ref} \cdot \frac{r}{R}$$

onde:

$V_0$ = tensão de saída;  $V_{ref}$ = tensão de referência;  $r$ = resistência *wiper* terra;  $R$ = resistência total;

#### Principais Vantagens:

- baixo custo;
- interface muito simples.

#### Principais Desvantagens:

- sempre existem não linearidades nestes potenciômetros;
- Não podem ser usados em sistemas onde se requer grande precisão;
- sofre deterioração com o tempo de uso, devido ao atrito dos componentes.

### Sincronizadores e Resolvedores :

**Sincronizadores** são dispositivos eletromecânicos rotacionais usados para transmitir informação angular eletricamente, de um lugar para outro, com grande precisão. Basicamente, o sincronizador é um transformador de acoplamento variável, consistindo de um *rotor de enrolamento primário AC-excitado* muito simples e dois ou mais *enrolamentos estatores* simetricamente orientados em torno do rotor (2.3). O acoplamento magnético efetivo entre o enrolamento do rotor e os enrolamentos dos estatores envolventes varia como uma função da orientação do vetor. O acoplamento máximo ocorre quando os campos são paralelos, e o acoplamento mínimo ocorre quando o campo do rotor está ortogonalmente alinhado com respeito a um determinado enrolamento dos estatores. Assim sendo, a saída do estator forma um conjunto de sinais AC cujas respectivas magnitudes definem unicamente o ângulo do rotor em qualquer momento no tempo.

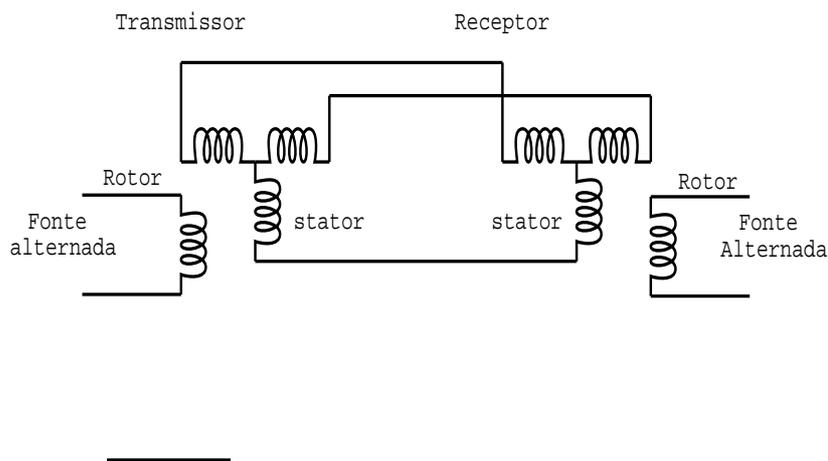


Figura 2.3: Sensor Sincronizador.(Adaptado de [20])

### Tipos de Sincronizadores

- Transmissores
- Diferenciais
- Receptores
- Transformadores de controle
- Resolvedores
- Resolvedores diferenciais

- Transformadores Lineares
- Transolvers

**Desvantagens:**

- Necessidade de adaptação para interfaceamento com equipamentos digitais.
- São de eletrônica complexa
- Geralmente ocupam grande volume
- Custo muito elevado.

**Resolvedores** são configurações especiais dos sincronizadores que desenvolvem voltagens proporcionais ao seno e ao cosseno do ângulo do rotor. São muito utilizados para dividir uma quantidade vetorial em seus componentes associados.

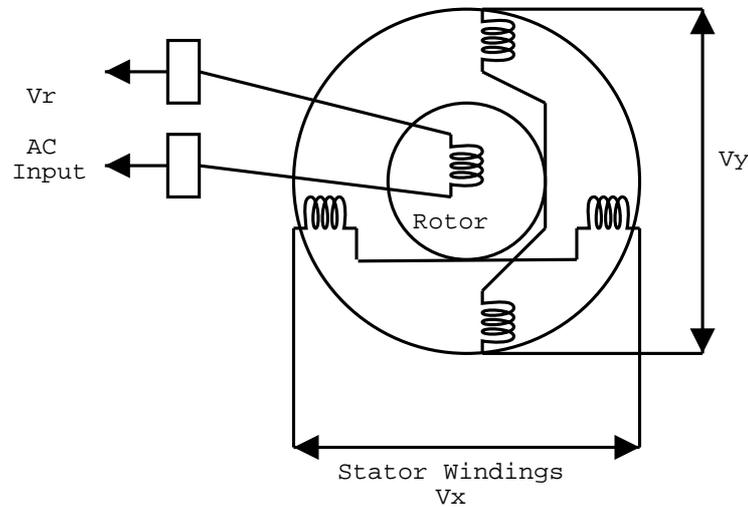


Figura 2.4: Resolvedor (Adaptado de [20])

**Vantagens:**

- É possível obter melhor resolução angular com o resolver do que com a maioria dos outros sensores de posição.

**Desvantagens:**

- Alto custo
- sofre interferência eletromagnética de outros aparelhos devido ao sinal de excitação AC.

**Escala Indutiva Linear** É bastante utilizado em máquinas de controle numérico. Possui grande precisão - 0,1 milésimos de polegada (linear) ou de  $0.00042^\circ$  É bastante similar aos resolvidores: há sempre dois componentes acoplados magneticamente, um dos quais se move em relação ao outro. Maiores detalhes veja em [6].

**Vantagens:**

- Grande precisão

**Desvantagens:**

- alto custo;
- é necessário eletrônica adicional para fazer a digitalização das saídas.

Transformadores Diferenciais LVDT<sup>2</sup> e RVDT<sup>3</sup>

O LVDT é um transdutor eletrônico que consiste de duas partes, uma móvel e outra fixa, e produz uma tensão de saída proporcional ao deslocamento da parte móvel relativo à parte fixa.

**Desvantagens:**

- Para a Robótica, tem o problema de ter que operar sobre a sua posição central, o que gera uma dificuldade de calibração, que implica em aumento de custo;
- É um dispositivo analógico
- Sua posição central varia com o tempo e a temperatura.
- Opera com ângulo de abertura de  $\pm 60^\circ$  e a maioria das juntas dos robôs se desloca mais que  $120^\circ$

---

<sup>2</sup>LVDT - Linear Variable Diferential Transformer;

<sup>3</sup>RVDT - Rotary Variable Differential Transformer.

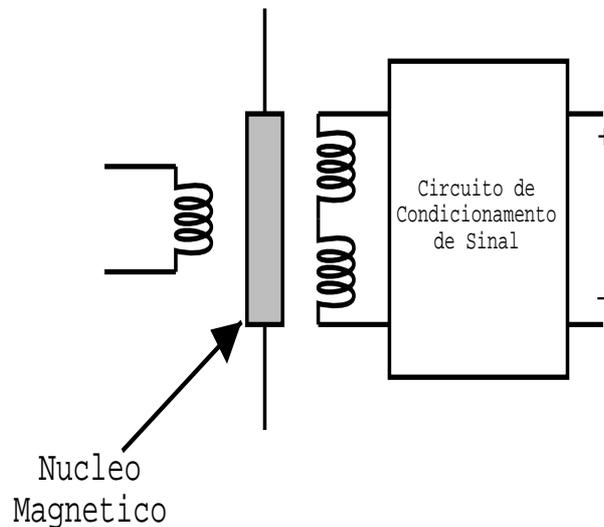


Figura 2.5: Sensor LVDT. (Adaptado de [6])

### Sensores de Odometria (ou de posição) Óticos

Possuem grandes vantagens com relação aos sensores não óticos, as principais sendo:

- sua medição não interfere no processo
- inexistência de desgaste mecânico
- alta taxa de medição
- imunidade ao ruído

A seguir relacionamos os vários tipos de sensores óticos:

**Interruptores Óticos** Em robôs do tipo ponto-a-ponto, necessita-se apenas que os pontos inicial e final sejam precisamente determinados; a informação da posição é pouco utilizada pelo sistema do robô. Nestes casos, um interruptor ótico atua com vantagens sobre as chaves de contato. Basicamente, consiste de um disco transparente, com uma região opaca colocada em algum ponto da superfície. Este tipo de sensor funciona como circuito fechado enquanto o feixe de luz atravessa a superfície transparente; e atua como circuito interrompido quando o feixe de luz alcança a parte opaca, interrompendo o feixe de luz. É utilizado para fornecer sinais de saída binários, do tipo "ligado-desligado". Sua simplicidade permite que outros pontos de interrupção sejam adicionados apenas colocando-se mais áreas opacas na superfície do disco.

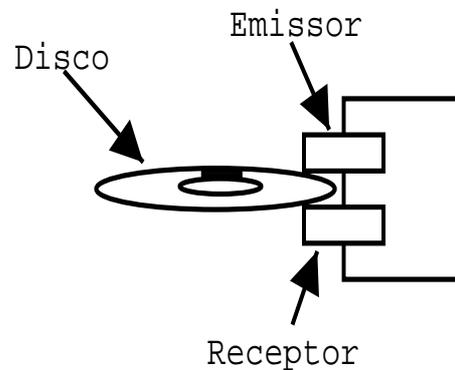


Figura 2.6: Interruptor óptico. (Adaptado de [6])

**Codificadores Óticos** Os primeiros codificadores óticos foram desenvolvidos em meados de 1940 pela *Baldwin Piano Company* para uso como “rodas de tom” (*tone wheels*) que permitiam aos órgãos elétricos imitar outros instrumentos musicais . Os dispositivos atuais basicamente embutem uma versão miniaturizada do sensor de proximidade de modo oposto (*Opposed-mode proximity sensor*). Neste esquema um canhão de luz é direcionado para um foto-detector, sendo periodicamente interrompido por um padrão codificado opaco/transparente pintado sobre a superfície de um disco rotativo intermediário fixado ao eixo de interesse. O disco rotativo pode ser construído de cromo sobre vidro, fotoplástico (*photoplast*), ou metal vazado.

#### **Vantagens:**

- Esquema de codificação direto
- Saída inerentemente digital
- Baixo custo
- Boa imunidade à ruídos

#### **Tipos básicos de codificadores óticos:**

- Incrementais
- Absolutos

**Codificadores Incrementais** Medem a velocidade rotacional e podem inferir posições relativas. O tipo mais simples é o *tacômetro codificador de canal único* 2.7, que consiste basicamente em um *interruptor inverso de luz mecânica instrumentada* que produz um certo número de pulsos de onda senoide ou quadrada para cada revolução do eixo. Quanto maior o número de pulsos, mais alta a resolução (e também o custo) da unidade.

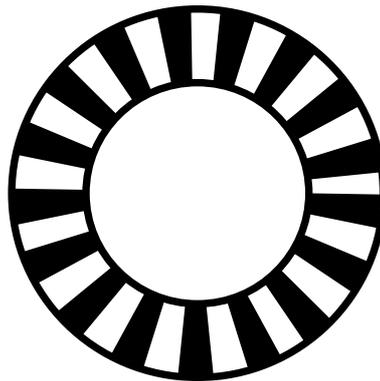


Figura 2.7: Codificador óptico incremental(Adapado de [20])

Codificadores incrementais também são bastante adequados como sensores de velocidade em sistemas de controle de média e alta velocidade, mas sofrem problemas de ruído e instabilidade em velocidades extremamente baixas devido a erros de quantização. Estes sensores, no entanto, são incapazes de detectar a direção da rotação, por isso não podem ser usados como sensores de posição. [6, 10]

**Codificadores Incrementais de Fase-Quadratura** Este tipo de sensor contorna os problemas do codificador de canal único através da adição de um segundo canal e distribuindo os detectores para que o resultado seja dois trens de pulso com uma defasagem de  $90^\circ$ . Isto permite que o decodificador eletrônico determine qual canal está conduzindo o outro, e desta maneira pode-se descobrir a direção do movimento. Para uma descrição mais detalhada veja [6, 20, 11]

**Codificadores Óticos absolutos** Sensores óticos absolutos medem diretamente a posição angular e inferem a velocidade 2.8. São tipicamente usados em aplicações rotacionais lentas que requerem informações posicionais quando a potencial perda de referência causada por interrupções de energia não podem ser toleradas.

Em sua estrutura, vários elementos detectores discretos em um array fotovoltaico são individualmente alinhados em trilhas codificadas concêntricas, como mostrado na figura

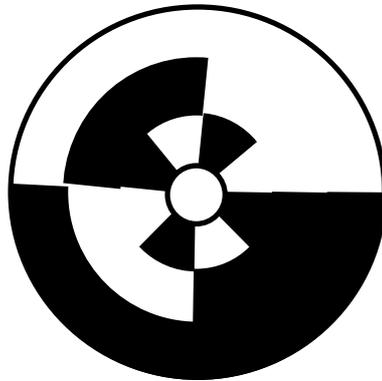


Figura 2.8: Codificador óptico absoluto. (Adaptado de [20])

2.8:

Este mecanismo passa a funcionar então como uma versão sem contato do antigo sensor de escovas. O posicionamento é então obtido pela leitura direta do código pelo array detector e a direção da rotação é obtida pela seqüência de valores lidos (crescente ou decrescente).

### **Sensores de Velocidade**

O tipo de sensor de velocidade mais comum é o tacômetro, que possui um uso bastante diversificado em diversas aplicações. Para esta finalidade ainda podemos utilizar os codificadores óticos (descritos acima) e conversores de frequência para tensão para obtermos a velocidade em valores analógicos. Para obtermos a velocidade em termos de valores digitais podemos utilizar codificadores óticos juntamente com softwares adequados.

**Tacômetros CC** Utiliza-se de um tacômetro<sup>4</sup> para medir a velocidade angular dos eixos de rotação das rodas de um robô. Os tacômetros geralmente produzem uma tensão CC que é proporcional à velocidade do eixo de rotação, quanto maior a velocidade, maior a tensão e vice versa.

### **Sensores de Aceleração**

Usam-se acelerômetros para se obter a aceleração linear. Acelerômetros são compostos de uma massa  $M$ , algum sensor de deslocamento linear, por exemplo um LVDT e um conjunto de molas com constante equivalente  $k$  2.9.

---

<sup>4</sup>Dispositivo que se utiliza da rotação de um motor CC para produzir uma tensão analógica que varia de acordo com a velocidade angular da rotação do motor. Um tacômetro CC pode ser descrito como um

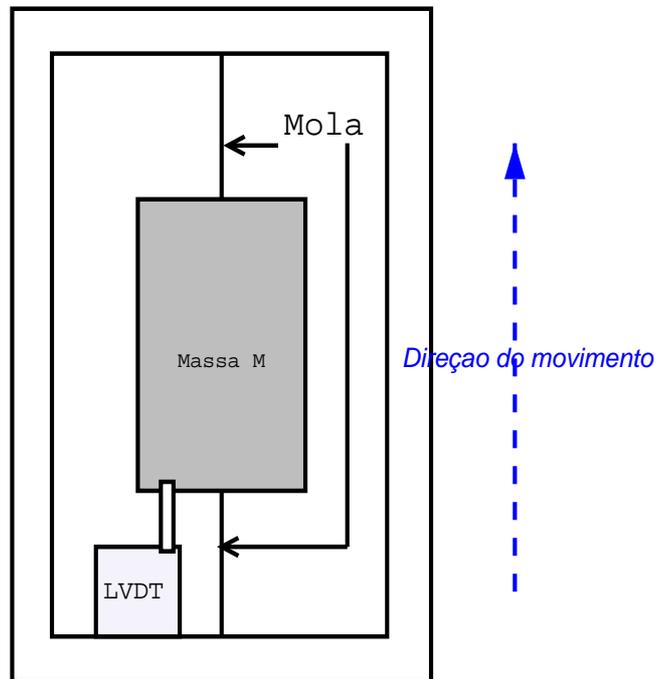


Figura 2.9: Acelerômetro(Adaptado de [6])

Usando a Lei de Newton podemos então derivar a aceleração da seguinte forma:

$$F = M \cdot a = k \cdot y \Rightarrow a = \frac{k \cdot y}{M}$$

Os acelerômetros geralmente possuem um uso limitado em Robótica devido ao fato de medirem apenas a aceleração linear. Para o caso da medida de velocidade de juntas, geralmente deriva-se o valor da aceleração diretamente do valor da velocidade angular.

## 2.5 Sensores Externos

Esta classe de sensores tem a finalidade de permitir a monitoração do robô, mas agora do ponto de vista externo, ou seja, a partir de sua relação com o mundo e com a tarefa que lhe foi determinada. São estes tipos de sensores que funcionam como os órgãos da “visão”, “audição” e “tato” dos robôs, e permitem que estes se orientem no mundo. [2, 9, 24, 20]

---

gerador que é capaz de medir a velocidade de seu eixo de rotação.

### 2.5.1 Sensores de Proximidade

Servem para indicar ao robô se ele está próximo de algum objeto do mundo. Este conceito de proximidade geralmente está relacionado à algum intervalo de distância especificado. São divididos em sensores de contato e não-contato.

#### Sensores de Contato

Sensores tácteis (ou de contato) são geralmente empregados em veículos guiados automaticamente (*Automated Guided Vehicle - AVGs*) e em robôs móveis para fornecer uma indicação de colisão de último recurso com obstruções em torno do robô. O processo de detecção deste tipo de sensor envolve contato físico direto entre o sensor e o objeto de interesse. Atualmente várias tecnologias tem sido aplicadas na construção dos chamados *esquemas sensíveis tácteis*. entre essas tecnologias de sensores, destacamos os seguintes, como mais representativos:

- Chaves de contato
- Magnéticos
- Piezoelétricos
- Capacitivos
- Fotoelétricos
- Magneto-resistivos
- Piezo-resistivos
- Ultrasônicos

Do ponto de vista da robótica móvel, estes esquemas sensores podem ser divididos em três grupos principais:

1. Sensores tácteis ou antenas
2. Amortecedores tácteis
3. *Arrays* de superfície distribuídos.

**Sensores Tácteis ou Antenas** Consistem, geralmente, de hastes de metal ligadas a terminais de chaves de contato. A figura abaixo mostra um tipo simples de sensor táctil. Podem ser divididos em duas categorias:

- **Sensores passivos:** que são geralmente fixos (2.10), e
- **Sensores ativos:** que geralmente implementam alguma estratégia mecânica de busca para tornar a orientação mais efetiva. Este processo assemelha-se muito ao movimento das antenas de um inseto (2.11). Nesta configuração de antena o ponto de contato real ao longo do sensor flexível é determinado pela medida da quantidade de rotação (ângulo  $\theta$  após o contato inicial, e o torque induzido correspondente.

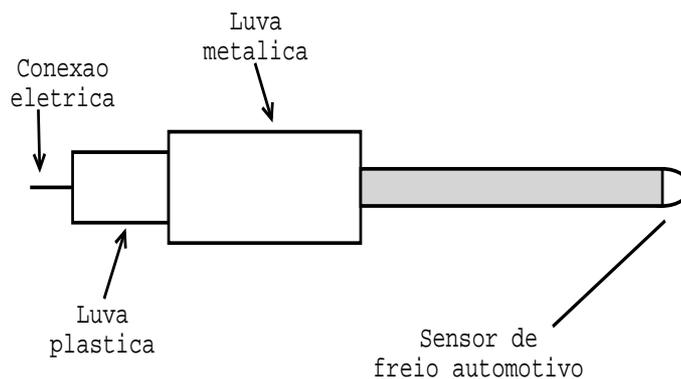


Figura 2.10: Sensor táctil passivo.(Adaptado de [20])

Atuador

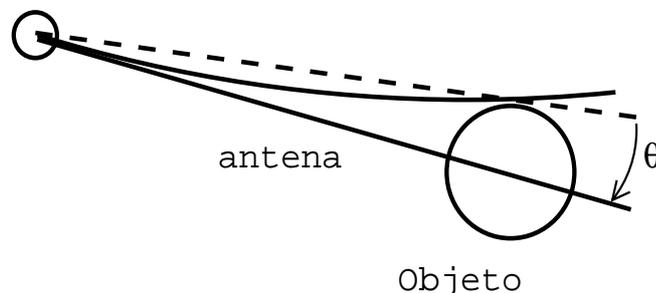


Figura 2.11: Sensor ativo em forma de antena. (Adaptado de [20])

**Amortecedores Tácteis** Consistem, geralmente, de uma placa de metal presa por uma por uma ponta e curvada para fora na outra, formando um amortecedor protetor na frente do

veículo . Esta placa está restrita a um único grau de liberdade de movimento, e quando ela sofre pressão externa provoca a ativação de alguma chave de contato. As 2.12 mostra um esquema exemplos de amortecedores tácteis.

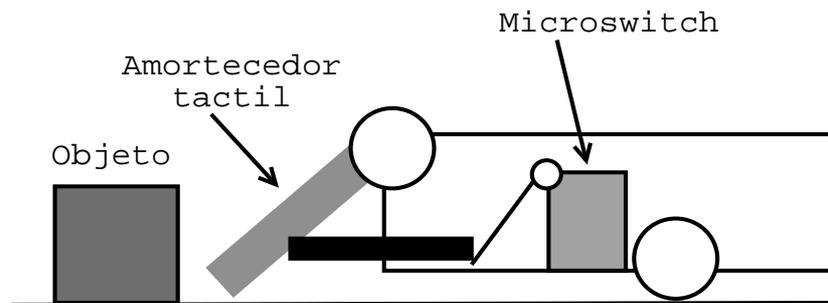


Figura 2.12: Amortecedor Tátil. (Adaptado de [6])

**Arrays de superfície distribuídos** São a nova tendência na construção de sensores tácteis. São arranjos complexos de sensores, com o objetivo de fornecer o perfil do objeto em contato com o array sensor. Além do perfil bidimensional do objeto, este array também é capaz de fornecer alguma quantificação da magnitude da força do contato.

Atualmente tem surgido um grande interesse pelo desenvolvimento de sensores contínuos, como “peles”, os quais poderiam ser incorporados diretamente ao exterior dos robôs ou veículos robóticos. Mais sobre peles artificiais na seção 2.5.3.

### **Sensores de Proximidade não contato**

São utilizados para determinar a presença de objetos próximos. Foram desenvolvidos para estender a faixa sensitiva além daquela permitida pelos sensores de contato. A sua principal vantagem é evitar o contato direto entre o robô e os objetos circundantes. Costumam ser classificados, de acordo com as suas propriedades específicas que são usadas para disparar a ação, nas seguintes categorias:

- Magnéticos
- Indutivos
- Ultrasônicos
- Microondas

- Óticos
- Capacitivos

**Sensores de Proximidade Magnéticos** Esta categoria inclui os *reed switches*, dispositivos de efeito Hall e sensores magneto-resistivos.

**Reed Switches Magnéticos** São o tipo mais simples de sensores de proximidade magnéticos. Consistem de um par de *reeds* ferromagnéticos de baixa relutância, fixados em extremidades opostas de um tubo de vidro hermeticamente fechado, dispostos de tal forma que suas pontas se sobrepõem levemente, porém sem se tocarem (2.13). As pontas dos filamentos assumem polaridades opostas quando expostas à um fluxo magnético externo, e a subsequente força atrativa força os dois *reeds* a fazerem contato, fechando um circuito elétrico que dispara a ação.

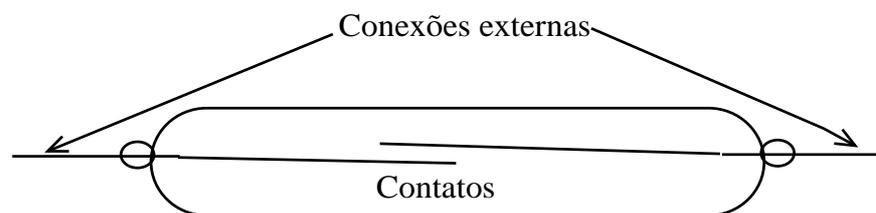


Figura 2.13: Sensor Reed Switch magnético.(Adaptado de [20])

Alguns problemas com este tipo de sensor podem ocorrer devido à trepidações, vibrações estruturais, etc. Estes sensores estão disponíveis em duas versões: *normalmente aberto* e *normalmente fechado*, e são comumente empregados como sensores de fechamento de portas e janelas em aplicações de segurança.]

**Sensores de Efeito Hall** O *Efeito Hall* foi descoberto por E. H Hall em 1879. Hall notou que uma voltagem muito pequena era gerada na direção transversal através de um condutor (2.14), gerando uma corrente na presença de um campo magnético externo, dada pela seguinte equação:

$$V_h = \frac{R_h \cdot I \cdot B}{t}$$

onde:

$V_h$  = voltagem Hall  $R_h$  = coeficiente de Hall (dependente do material)  $I$  = intensidade de corrente elétrica (em Amperes)  $B$  = densidade de fluxo magnético (perpendicular a  $I$ ) em Gauss  $t$  = espessura do elemento (em cm)

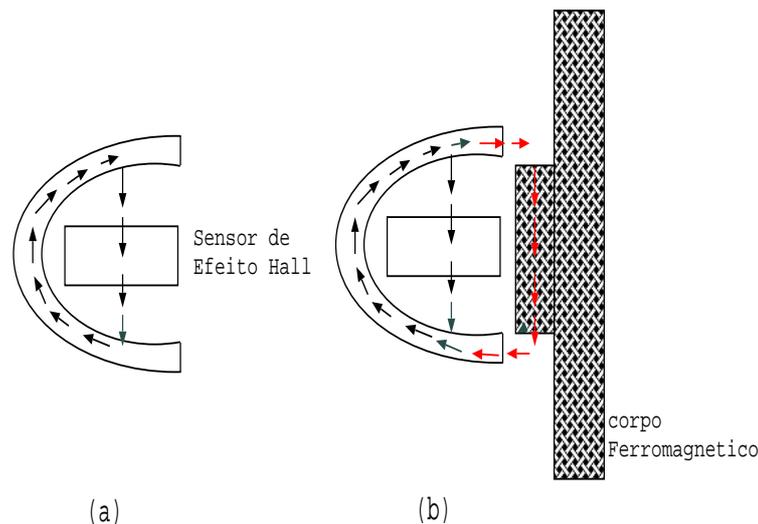


Figura 2.14: Sensor de efeito Hall com ímã permanente (a) linhas de fluxo na ausência de um corpo ferro-magnético nas proximidades do sensor; (b) linhas de fluxo quando há um corpo ferro-magnético nas proximidades. (Adaptado de [6])

**Sensores Magneto-Resistivos** Materiais *anisotrópicos* são materiais que possuem a característica de que o valor de uma dada propriedade depende da direção da medida; materiais *isotrópicos* são aqueles que sempre exibem os mesmos valores para as propriedades medidas em todas as direções. A *anisotropia* pode estar relacionada à forma do material, à sua estrutura cristalina ou ainda à sua estrutura interna. Inserir Referência (Graf, 1974)

Por exemplo, a direção da magnetização em um cristal ferro-magnético será orientada ao longo de um certo eixo cristalográfico, conhecido como *eixo fácil* (easy axis), onde *easy* refere-se à direção preferida da magnetização. (Barret et al., 1973)

A mudança na direção da magnetização em um material condutivo através da aplicação de algum campo magnético externo  $H_y$  resultará em uma mudança na resistividade  $\rho$  do material, fenômeno conhecido como *efeito magneto-resistivo*. Uma das mais comuns aplicações dos sensores magneto-resistivos em robótica é em rodas codificadoras.

Outros usos incluem: *compassing eletrônico*, medida de ângulo e posição, percepção de corrente e medida de campos magnéticos em geral. [20]

**Sensores de Proximidade Indutiva** Chaves de proximidade indutiva são os sensores mais usados atualmente para detecção de objetos de metais ferrosos e não-ferrosos a curtas distâncias. Sensores de proximidade indutiva geram um campo de rádiofreqüência oscilatório (geralmente na faixa de 100Khz a 1Mhz) ao redor de uma bobina enrolada em um núcleo de ferrite. Quando um objeto metálico entra no campo definido projetado pelo sensor, correntes são induzidas na superfície alvo. Essas correntes produzem um campo magnético secundário que interage com o campo de prova, carregando o oscilador de prova. A impedância efetiva da bobina de prova muda, resultando em um deslocamento da freqüência do oscilador (ou mudança de amplitude) que é convertida em um sinal de saída proporcional à distância percebida entre a bobina de prova e o alvo.

**Sensores de Proximidade Capacitivos** Sensores de proximidade capacitivos são muito similares aos sensores de proximidade indutiva, porém tem a capacidade de detectar também materiais dielétricos, além de objetos metálicos. Sensores capacitivos reagem à variações na capacitância elétrica entre um corpo ou lâmina de prova e o ambiente em torno do mesmo. Quando um objeto se aproxima, ocorre uma mudança geométrica no campo, e as características dielétricas dentro da região sensível causam o aumento da capacitância. Esta mudança na capacitância pode ser sentida de diversas formas:

1. Um aumento no fluxo de corrente através da lâmina de prova
2. Início de oscilação em um circuito RC
3. Diminuição na freqüência de uma oscilação *ongoing*.

**Sensores de Proximidade Ultra-Sônicos** Sensores ultra-sônicos são exemplos de sensores reflexivos, ou seja, são sensores que respondem a mudanças na quantidade de energia emitida que retorna para um detector após interagir com o objeto alvo. Sistemas ultra-sônicos típicos consistem de dois transdutores (um para transmitir e outro para receber a energia retornada), embora a velocidade relativamente baixa do som permita operar no modo transceiver, com apenas um transdutor.

O transmissor emite uma onda longitudinal na região ultra-sônica do espectro acústico (tipicamente 20 a 200 Khz) acima do limite normal do ouvido humano. A resposta recebida pelo receiver é uma função da amplitude da energia retornada, em oposição ao tempo passado antes da detecção do eco do som emitido.

Sensores ultra-sônicos são bastante úteis a distâncias maiores do que os sensores citados anteriormente, e podem detectar tanto objetos sólidos quanto líquidos. Seu funcionamento é semelhante ao de um radar: se um objeto entra no campo do sinal acústico, a energia é refletida de volta ao receiver. Como acontece com qualquer sensor reflexivo, o alcance máximo de detecção é dependente não apenas do nível de potência do sinal emitido, mas também de certas características do alvo, como área da seção reta, reflectividade e directividade. Uma vez que a amplitude do sinal recebido alcança um limiar pré-estabelecido, o sensor modifica o seu estado de saída para indicar a detecção [25, 1, 10, 35].

**Sensores de Proximidade por Micro-ondas** Sensores de micro-ondas operam a distâncias maiores ainda que os sensores ultrasônicos (tipicamente, de 5 a 150 pés ou mais). Seu funcionamento é bastante similar ao dos sensores ultrasônicos a não ser pelo fato de que estes transmitem energia eletromagnética na região das micro-ondas de rádio-freqüência (10,50GHz a 10,55GHz e 24,075GHz a 24,175GHz) (2.15). Quando a presença de um alvo reflete energia suficiente o sensor muda de estado para indicar a presença de um objeto dentro de um campo de visão.

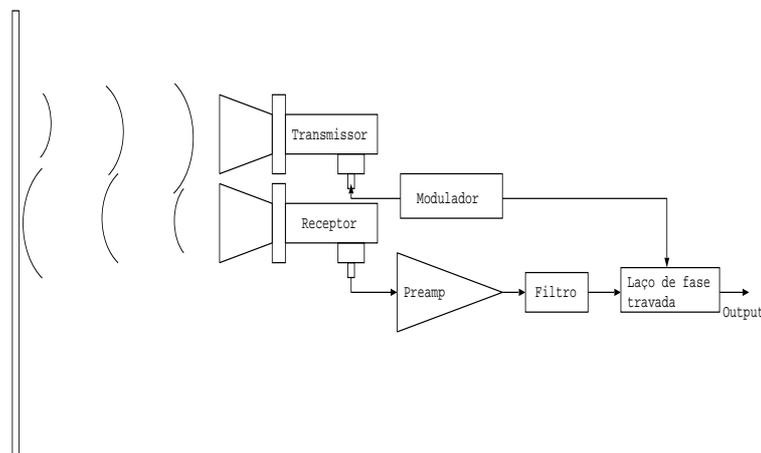


Figura 2.15: Esquema de um sensor de proximidade por micro-ondas. (Adaptado de [20])

**Sensores de proximidade óticos** Sensores óticos (ou foto-elétricos), têm vasta aplicação em sistemas industriais. Podem ser classificados em três grupos básicos:

- opostos,
- retroreflectivos e

- difusos.

Everett [20] considera que as duas primeiras categorias não são realmente sensores de proximidade no sentido estrito da terminologia. A razão desta consideração é que os dois tipos de sensores citados não são capazes de perceber qualquer tipo de objetos, percebem apenas um certo tipo de objeto no qual é instalado um receptor ou uma superfície refletora especial. Abaixo encontramos uma explicação mais detalhada sobre cada um destes sensores:

**Sensores Óticos Opostos** Conhecidos como “olhos elétricos”, surgiram nos anos 1950, e foram muito utilizados em uma grande variedade de aplicações, como abertura automática de portas, sistemas de segurança, etc. Consiste basicamente de duas unidades separadas - um emissor e um receptor - colocadas na região de interesse (emissor) e em alguma parte do robô móvel (receptor) (2.16).

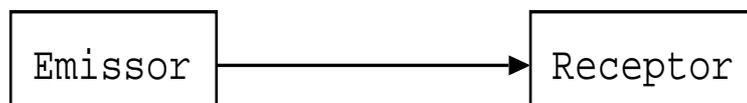


Figura 2.16: Sensor ótico oposto. (Adaptado de [20])

O emissor emite um fecho de luz (geralmente através de um LED) que é captado pelo receptor (no robô) indicando assim a presença do referido objeto (que possui o emissor). Geralmente tem pouca aplicabilidade em robótica móvel devido à sua configuração geométrica, ou seja, a necessidade de pares opostos e separados de emissor e receptor.

**Sensores Óticos Retro-reflexivos** Podem ser considerados uma evolução dos sensores óticos opostos, pois também possui um par transmissor/receptor, porém estes não estão mais colocados em partes opostas, mas sim apenas em uma única parte (no caso da robótica móvel, o robô) (2.17).

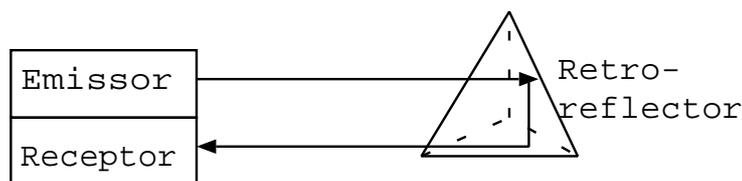


Figura 2.17: Sensor ótico retrorefletor. (Adaptado de [20])

O nome *Retro-reflexivo* deve-se ao uso de um retro-refletor (como um espelho) colocado de maneira oposta à um par transmissor/receptor. A percepção de um objeto dá-se então pela reflexão de um raio de luz emitido pelo emissor (instalado no robô) e refletido pelo retro-refletor (fixado no objeto de interesse). São usados em robótica móvel no auxílio à navegação, sendo os retro-refletores utilizados como marcações de posição que são identificadas pelo robô enquanto este se desloca no ambiente.

**Sensores óticos difusos** Operam de forma similar aos retro-reflexivos, com a diferença de que no caso dos sensores difusos não há a presença de uma contra-parte retro-reflexiva, assim, a energia retornada pelo objeto é a difusão da luz na própria superfície do objeto (2.18).

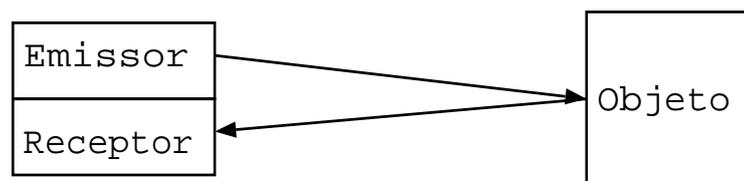


Figura 2.18: Sensor ótico Difuso. (Adaptado de [20])

### 2.5.2 Sensores de Distância

Sensores de distância são utilizados para medir a distância entre um ponto de referência (geralmente o próprio sensor) e um outro objeto localizado no espaço em volta do robô. Existem várias técnicas para medir a distância de um objeto até o sensor, cada uma aplicável a certas tecnologias de construção de sensores. As principais técnicas são [40, 1, 21]:

1. triangulação
2. Por luz estruturada
3. por tempo de voo

O sistema de medida de distância por triangulação é bastante conhecido dos seres humanos, pois é o método que nós utilizamos para determinar distâncias baseado na visão estéreo através da paralaxe visual. O sistema de medida de distância por tempo de voo também já é um velho conhecido, sendo o sistema utilizado pelos morcegos, o qual foi depois copiado

pelos seres humanos nos sonares dos navios e submarinos. Este sistema basicamente consiste em enviar uma onda sonora e então captar o seu eco, determinando a distância através da medição do tempo gasto entre a emissão do sinal e a recepção do mesmo.

### Triangulação

É um dos métodos mais simples para se medir distâncias, e é também um dos mais usados. Baseia-se na premissa trigonométrica de que, dado o comprimento de um lado e seus dois ângulos, é possível determinar os comprimentos dos outros lados e o valor dos outros ângulos através da *Lei Básica dos Senos*, que pode ser rearranjada para representar o comprimento do lado B como uma função do lado A e dos ângulos  $\theta$  e  $\phi$ :

$$B = A \frac{\sin \theta}{\sin \phi} = A \frac{\sin \theta}{\sin(\theta + \phi)}$$

onde:

B é a distância desejada do sensor até o objeto, como mostra a 2.19

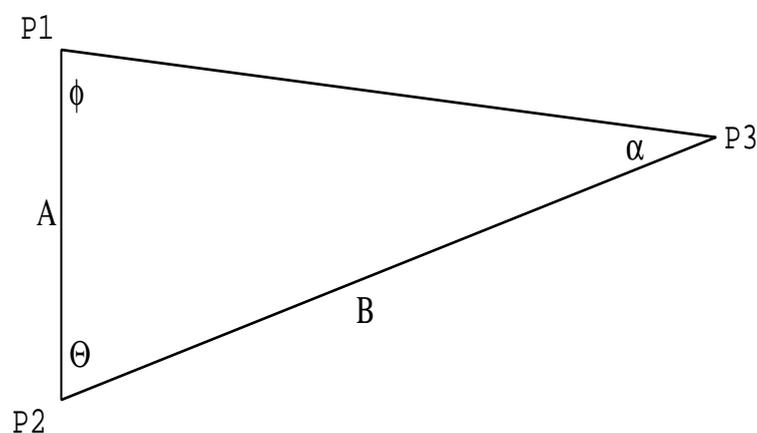


Figura 2.19: Sistemas de triangulação determinam a distância B para o alvo colocado no ponto  $P_3$ , medindo os ângulos  $\phi$  e  $\theta$  nos pontos  $P_1$  e  $P_2$ .(Adaptado de [20])

Basicamente, existem dois tipos de triangulação: a **triangulação passiva**, a qual se utiliza apenas da emissão de luz do ambiente, e a **triangulação ativa**, que utiliza uma fonte de energia para iluminar o objeto em questão. Veremos os dois esquemas a seguir.

**Triangulação ativa** Neste esquema o objeto é iluminado por um estreito fecho de luz, a qual retorna para um detector após refletir em uma pequena porção da superfície do ob-

jeto, como mostra a 2.20. Quando o detector percebe recebe o feixe de luz, pode-se obter a distância entre ele e a porção do objeto iluminado. Note que este tipo de configuração proporciona uma medida pontual. Para obtermos uma leitura de um conjunto de pontos da superfície do objeto, podemos montar o conjunto sensor sobre um sistema móvel, desta forma teremos o feixe varrendo a superfície do objeto e produzindo um conjunto de pontos do objeto cuja distância do sensor é conhecida.

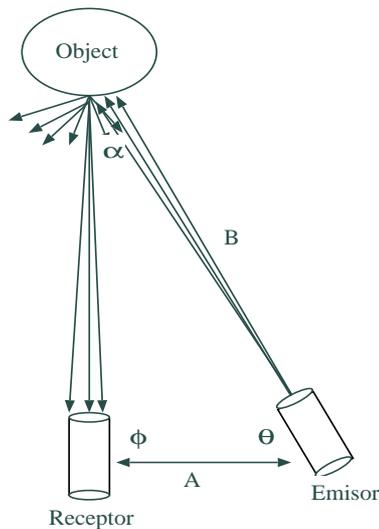


Figura 2.20: Esquema de Triangulação Ativa.(Adaptado de [20])

**Triangulação passiva** Também conhecida como Sistema Estereoscópico passivo, este sistema geralmente se utiliza de duas câmeras de vídeo arranjadas como mostra a 2.21. As câmeras são organizadas de modo a formar um triângulo com o objeto em foco. A medida dos ângulos  $\theta$  e  $\phi$  e da separação lateral (distância entre as câmeras) e a orientação conhecida permite o cálculo da distância do centro focal (ponto médio entre as duas câmeras) e o objeto alvo.

### Luz Estruturada

Esta técnica de medida de distância consiste em projetar uma luz padrão (geralmente um laser) sobre o objeto desejado e calcular a distância através da distorção provocada no padrão de luz pela reflexão da mesma na superfície do objeto. A forma mais comum é utilizar um laser projetado através de uma fenda estreita ou uma fonte de luz cilíndrica. As distorções provocadas pela reflexão da luz no objeto forma uma imagem formada por linhas, as quais

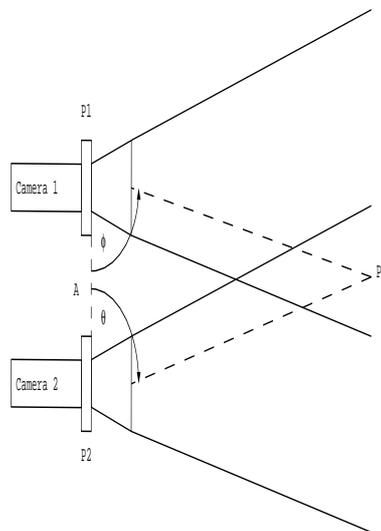


Figura 2.21: Esquema de Triangulação passiva ou triangulação estereoscópica passiva utilizando câmeras de vídeo ou sensores óticos ou de micro-ondas.(Adaptado de [20])

geram um padrão de imagem que terá de ser analisado por um computador para determinar qual o objeto que está sendo focalizado. Este processo possui um custo computacional mais elevado do que as outras técnicas, pois necessita de software mais complexo para a análise dos padrões de imagens.

### Tempo de Vôo

Sistemas de medida de distância por Tempo de Vôo (TOF - Time-Of-Flight) medem a distância do sensor ao objeto baseados no tempo requerido para que um pulso de energia (acústica ou luminosa) viaje do emissor até o objeto, seja refletida e retorne para o receptor do sensor na forma de eco. Os tipos mais comuns de energia utilizados nesta técnica são o ultra-som, rádio-freqüência, ou energia luminosa; para cada caso, os parâmetros envolvidos no cálculo da distância são: a velocidade do som no ar ( $\cong 300$  m/s), a velocidade da luz ( $3^9$  m/s). O cálculo da distância torna-se então trivial, simplesmente dado pela equação:

$$d = \frac{v \cdot t}{2}$$

onde:  $d$  = distância do sensor ao objeto;  $v$  = velocidade do feixe de energia emitido (velocidade do som ou da luz);  $t$  = tempo decorrido.

A divisão por dois torna-se necessária porque o tempo percebido é relativo ao percurso do dobro da distância, ou seja, a energia leva um tempo  $t/2$  para alcançar o objeto

e um tempo  $t/2$  para retornar como eco. Sistemas de medida de distância por tempo de vôo tem a vantagem de manter a sua acurácia de forma linear com relação ao aumento da distância, porém apresentam também algumas desvantagens:

- Pode apresentar variações na velocidade de propagação, particularmente no caso dos sistemas que se utilizam de energia acústica (ultra-som).
- Incertezas na determinação do tempo de chegada exato do pulso refletido (eco).
- Inexatidão nos circuitos de tempo usados para medir o tempo de ida e volta do eco.
- Interação das ondas incidentes com a superfície alvo.

A seguir são apresentados três esquemas para a determinação de distâncias através do tempo de vôo; os dois primeiros utilizam a emissão de um laser, e o último utiliza-se da emissão de ultra-som: **Pulsos de Luz, Deslocamento de fase e pulso-eco.**

**Pulsos de Luz (laser)** A distância do objeto é determinada medindo-se o tempo que a luz emitida leva para ir e voltar entre a fonte emissora e o receptor. A distância é obtida pela seguinte equação:

$$D = \frac{c \cdot t}{2}$$

onde :  $t$  é o tempo de vôo e  $c$  é a velocidade da luz.

Como a velocidade da luz é da ordem de 30cm/ns, a instrumentação utilizada deve ser de alta resolução, o que torna este sistema bastante caro, e portanto, utilizado somente em aplicações bastante específicas.

**Deslocamento de fase (laser)** É uma variante do método anterior, em que, em vez de pulsos de luz, emite-se um feixe contínuo de laser e obtém-se a distância através da medição do deslocamento de fase entre o feixe emitido e o feixe refletido.

**Ultra-som (Pulso-eco)** Funciona basicamente da mesma forma que a técnica de luz pulsada, com a diferença de que aqui aplica-se, em vez de um pulso de luz, um pulso de ultra-som. A distância é dada pela equação

$$D = \frac{V_s \cdot t}{2}$$

onde:  $V_s$  = Velocidade do som no ar;

Como a velocidade do som no ar é influenciada pela temperatura, e sabendo-se que  $V_s = 331,6 \text{ m/s}$  a  $0^\circ\text{C}$ , então  $V_s \approx 331,6 \sqrt{1 + \frac{T}{273}}$ , onde T é o valor da temperatura em  $^\circ\text{C}$ .

### 2.5.3 Sensores de Tato

Sensores de tato são geralmente usados em Robótica para se obter um feedback associado com o contato entre uma mão robótica de um manipulador e os objetos em seu espaço de trabalho. A importância deste tipo de sensor é devido à necessidade de se controlar a força aplicada a um objeto por um manipulador, para evitar que este objeto caia, ou seja esmagado pela mão robótica. São divididos entre analógicos e binários (ou digitais). Os binários são basicamente chaves interruptoras que informam sobre a presença ou ausência de um objeto. Os analógicos, por sua vez são capazes de fornecer sinais analógicos que são proporcionais à força aplicada sobre eles.[9]

#### Sensores Binários

Consiste geralmente de um interruptor localizado na superfície interna de cada dedo da mão do robô. Geralmente são usados vários sensores posicionados em diversos pontos da mão para se obter uma informação mais completa a respeito da posição do objeto na mão.

#### Sensores Analógicos

Neste tipo de sensor, a sua saída é proporcional à força aplicada localmente sobre o sensor. O tipo mais simples consiste de um bastão que é mecanicamente ligado a um eixo de rotação e a uma mola, como mostra a 2.22. Neste esquema, o movimento do bastão provoca uma rotação proporcional que pode ser medida através de um potenciômetro ou de um disco codificador ótico. Como a constante de elasticidade da mola é conhecida, é fácil obter-se a força para um dado deslocamento.

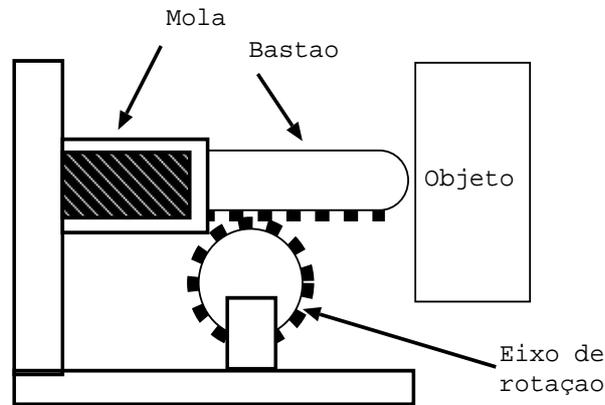


Figura 2.22: Sensor de tato analógico.(Adaptado de [20])

### Pele artificial

São matrizes de eletrodos em contato elétrico com um material elastomérico contínuo cuja resistência varia em função da compressão. Quando um objeto entra em contato com esta “pele”, ele causa deformações locais que são medidas como variações de resistência. Estas variações são convertidas em sinais elétricos que possuem amplitudes proporcionais às forças aplicadas em qualquer ponto da superfície do material. Este tipo de sensor tem, geralmente, vida limitada, pois após um certo número de operações (por volta de algumas centenas) o material começa a romper-se. Se for usado como revestimento sensor para agarrar objetos, pode ser danificado se o tal objeto possuir arestas. Devido a este inconveniente, tem de ser substituído de tempos em tempos. Além disto, este tipo de sensor costuma sofrer de **histerese**, ou seja, as resistências não retornam ao seu valor original anterior à compressão.

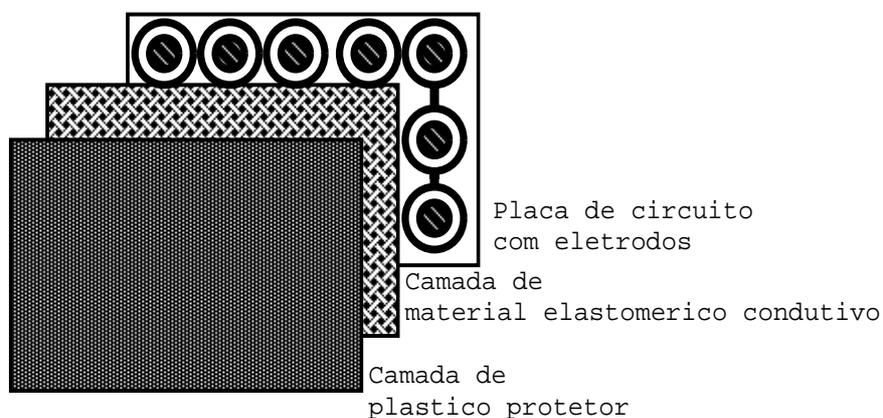


Figura 2.23: Pele Artificial. (Adaptado de [6])

Existem várias técnicas de construção de peles artificiais. Uma das mais comuns possui uma estrutura de três camadas, consistindo de material elastomérico condutivo coberto por uma camada de plástico para proteção. Estas camadas revestem uma terceira camada que é uma placa de circuito impresso com vários pares de anéis de eletrodos impressos na mesma, em forma matricial (linhas e colunas).

#### 2.5.4 Sensores de Deslizamento

Todos os sensores de tato vistos anteriormente tem a característica comum de que todos, sem exceção, medem apenas a força normal à sua superfície. Porém, em alguns casos, é necessário obter-se informação a respeito do movimento tangencial à superfície do sensor. Tal necessidade vem do fato de que, por exemplo, um robô manipulador pode precisar saber se um objeto que foi agarrado está deslizando pela sua mão para que ele ajuste a força aplicada ao objeto e evitar que o mesmo caia. [6]

Este tipo de sensor é o menos desenvolvido até o momento, porém muita pesquisa tem sido feita nesta área nos últimos anos. Uma das técnicas mais simples e que tem sido utilizada com bastante sucesso em várias aplicações é a técnica conhecida como “*Lift-and-try*”. Esta técnica utiliza a observação da corrente do motor de uma ou mais juntas do braço robótico para verificar se o objeto está ou não deslizando. Quando o robô tenta levantar um objeto desde a superfície, se a força aplicada for suficiente, a corrente do motor deveria aumentar devido ao aumento de carga. Se a força for insuficiente para erguer o objeto, o que consequentemente geraria um deslizamento, não se verifica nenhum aumento na corrente do motor. Neste caso, o robô deve ser comandado para voltar a condição de início e aumentar a força de agarramento de um valor mínimo, a seguir tentando novamente o levantamento. Este processo é repetido até que haja um aumento de corrente que indique que o objeto foi devidamente agarrado. Esta técnica apresenta alguns problemas com relação ao seu uso, principalmente:[6]

- não há garantia de que o objeto deslize e caia após ter sido agarrado;
- pode ser um procedimento muito demorado quando trata-se de agarrar objetos muito frágeis juntamente com objetos muito rígidos, pois o valor mínimo do aumento da força deve ser muito pequeno, para evitar esmagar o objeto mais frágil;
- Erros na medição da corrente (causados por picos de corrente em motores com esco-

vas) podem causar demora no processo, ou risco de esmagamento de objetos frágeis.

### 2.5.5 Sensores de Força e Torque

Existem algumas técnicas que são usadas para obter mecanismos para medição de força e torque. As mais comuns são:

- monitoração da corrente do motor;
- usando um *LVDT*
- usando *strain gages*

#### Monitoração da corrente do motor

É a mesma técnica de 'lift-and-try', porém usada para conhecer a força de agarramento aplicada à um objeto. Baseia-se no fato de que a corrente da armadura do servo-motor da garra é proporcional ao torque gerado pelo motor:

$$I_a = \frac{T}{K_t}$$

onde:

$I_a$  = Corrente da armadura

$T$  = Torque gerado

$K_T$  = Constante de torque do motor

Se for utilizado um conversor de movimento rotacional para linear (conjunto engrenagem-cremalheira), a força gerada é:

$$F = \frac{T \cdot \eta}{R}$$

onde  $\eta$  é a eficiência do conjunto engrenagem-cremalheira (usualmente cerca de 90%) e  $R$  é o raio da engrenagem. Juntando as duas equações temos:

$$F = \frac{K_T \cdot I_a \cdot \eta}{R}$$

Este esquema sensor é bastante simples de ser implementado, porém tem alguns problemas:

1. a constante de torque do motor ( $K_T$ ) e a resistência da armadura ( $R$ ) variam com a temperatura, o que causa erros de leitura a medida que o aparelho aquece-se com o uso;
2. é susceptível à ocorrência de ruídos na escova do motor e à variações na resistência da escova em motores dom escova. Uma solução bastante custosa para este problema é adotar motores sem escova. Este método não chega a ser um sensor de força na verdadeira acepção da palavra, porque a força não é realmente medida, mas inferida a partir da corrente do motor. Assim sendo, este método é considerado mais como um método para manter uma força desejada sobre um objeto. [6]

### Usando um LVDT

Este método, diferentemente do anterior, pode ser considerado um verdadeiro esquema sensor de força e torque, que pode medir precisamente posições lineares. A relação força-deslocamento para um LVDT acoplado à uma mola é: [6]

$$F = K \cdot x$$

onde  $K$  é a constante da mola e  $x$  é o deslocamento sobre uma posição de equilíbrio. Este esquema é bastante simples e muito utilizado, porém apresenta alguns problemas, que são:

1. alto custo;
2. necessidade de calibração;
3. sensibilidade à variações de temperatura
4. Variação da constante  $K$  da mola com a temperatura
5. Só é capaz de medir a força em uma direção.

## 2.6 Resumo

Neste capítulo tratamos brevemente sobre sensores, mostramos sua classificação, os vários tipos de sensores e como os mesmos são utilizados em robótica. Sensores são essenciais para o projeto de robôs, porque representam os principais mecanismos de aquisição de dados de um robô no mundo. Eles são os equivalentes aos órgãos dos sentidos nos seres humanos e em outros animais. A área de sensores está em constante desenvolvimento, e produzindo sensores cada vez mais precisos e exatos.

## 2.7 Exercícios

1. Os sensores óticos são de extrema importância, principalmente em Robótica móvel. Existem, basicamente três tipos de componentes sensores de luminosidade que são: (a) foto-resistores, (b) foto-transistores e (c) foto-diodos. Faça uma pesquisa na internet e relacione as principais diferenças entre estes três componentes, suas vantagens e desvantagens e indicando quando e onde cada um deles pode ser usado em robôs móveis.
2. Neste capítulo sobre sensores deixamos de lado uma classe de sensores conhecida como "*sensores piroelétricos*". Faça uma pesquisa sobre este tipo de sensores, relacione suas vantagens e desvantagens e relacione também algumas aplicações existentes.
3. Com o barateamento das câmeras de vídeo digital, elas tornaram-se uma boa opção para uso em robôs móveis. Quais os maiores problemas para uso destes dispositivos em robôs móveis? Que vantagens as câmeras de vídeo apresentam com relação à outros dispositivos sensores? Quais as possibilidades do uso das câmeras digitais no futuro?
4. Imagine um robô humanóide, que tenha a capacidade de correr. Obviamente, este robô precisa saber sua velocidade de deslocamento e sua aceleração. Descreva quais sensores podem ser usados para fornecer ao robô esta informação, e como estes sensores atuariam (sozinhos, em conjunto, etc)
5. Imagine um pequeno robô programado para exibir o comportamento de uma barata. Que tipo de sensores deveriam ser montados neste robô? Qual a utilidade de cada um deles?

# Capítulo 3

## Atuadores

### 3.1 Introdução

Atuadores são dispositivos responsáveis pelo movimento e articulação das partes móveis de uma máquina, seja esta uma máquina-ferramenta ou um robô. São eles que executam e colocam em prática as tarefas ordenadas pelo sistema de controle.

Os atuadores não agem só, eles fazem parte de um conjunto formado por outros dois sistemas; sistemas de transmissão e redução e os sensores de posição e velocidade.

- Sistema de transmissão e redução: são os elementos capazes de transladar e modificar o movimento gerado pelo acionamento. Ambos sistemas de transmissão e redução devem cumprir uma série de características muito restritivas devido às altas exigências de funcionamento das diversas partes do robô que devem trabalhar de forma harmoniosa.
- Sensores de posição e velocidade: são os dispositivos que captam as informações sobre posição e velocidades de cada uma das partes móveis dos robôs. Representam uma parte vital de todo sistema, uma vez que o controle irá “tomar suas decisões” baseados nos valores fornecidos por estes dispositivos.

Neste capítulo focaremos nossa atenção apenas nos atuadores de maior interesse para a robótica móvel, deixando de lado também os sistemas de transmissão e redução. O estudo de sensores já foi feito/será feito no capítulo ???.

## 3.2 Tipos de Atuadores

Os atuadores utilizados em robótica são classificados em função da energia que utilizam para funcionarem [5]. Isto é de fundamental importância pois a natureza da energia irá determinar a aplicação, as condições de funcionamento e o tipo de controle a ser utilizado. Tudo isto será decisivo no projeto e posteriormente na operação e manutenção.

Os três principais tipos de atuadores são:

- pneumáticos
- hidráulicos
- elétricos

Os acionamentos pneumáticos e hidráulicos se baseiam no uso da energia liberada por flúidos em movimentos. No primeiro caso utiliza-se ar comprimido e no segundo um flúido mais viscoso; água e óleos são um bom exemplo. Em ambos os casos, no entanto, será necessária a instalação de bombas e compressores, filtros, acumuladores, equipamentos para refrigeração, válvulas, etc. Tudo isto faz com que acionamentos pneumáticos e hidráulicos requeiram muita manutenção, representando uma grande desvantagem sobre os acionamentos elétricos.

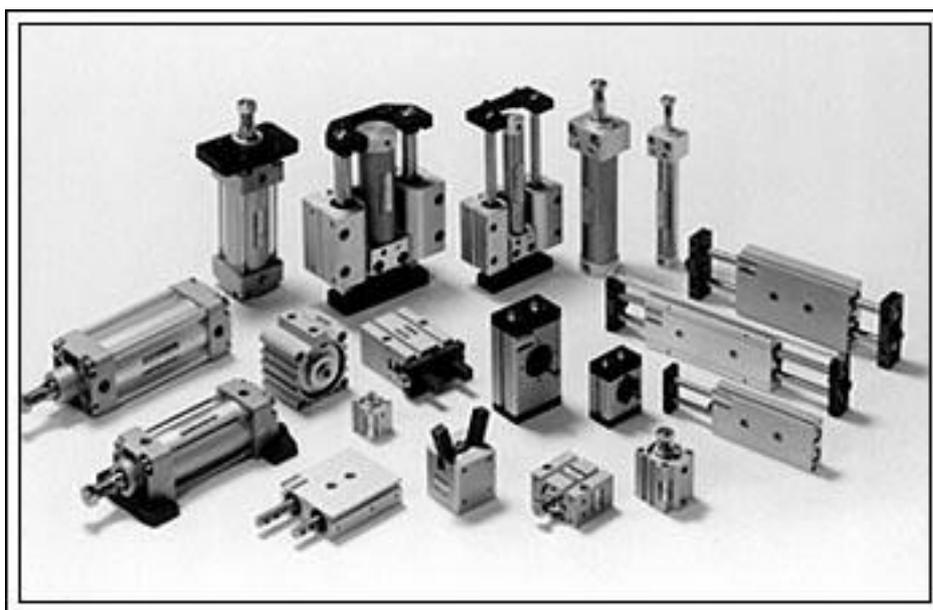


Figura 3.1: Grande variedade de dispositivos hidráulicos e pneumáticos

Atuadores baseados em motores elétricos são sem dúvida os mais utilizados em projetos de robótica, independentes do tamanho. Isto deve-se ao fato de existirem uma gama bastante grande de motores com as mais diversas características, tornando-os flexíveis o suficiente para atenderem a quase todas aplicações. As próximas seções apresentaram melhor cada um destes tipos básicos.

### 3.2.1 Pneumáticos

Como dito anteriormente, este tipo de atuador utiliza como fonte de energia a pressão do ar. Dado que o ar é um flúido altamente compressível, um controle preciso tanto da velocidade como da posição é muito difícil. Além disso, estes parâmetros dependem diretamente da carga aplicada. Se por um lado isto representa uma desvantagem, por outro faz com que este tipo de acionamento seja bastante resistente aos danos provocados por uma sobrecarga.

Em robótica utilizam-se basicamente dois tipos de acionamentos:

- cilindros pneumáticos
- motores pneumáticos
  - de aletas rotativas
  - pistão axial

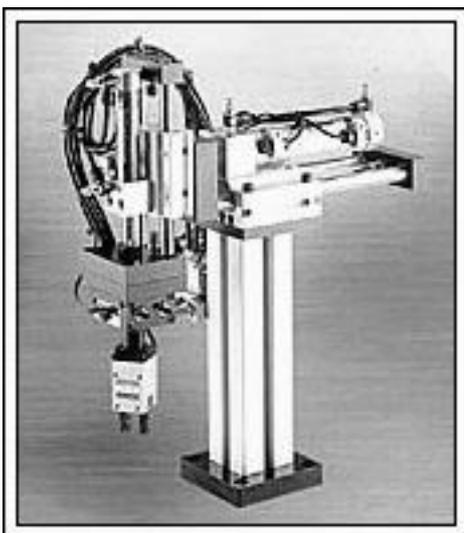


Figura 3.2: Exemplo de um robô com acionamento pneumático

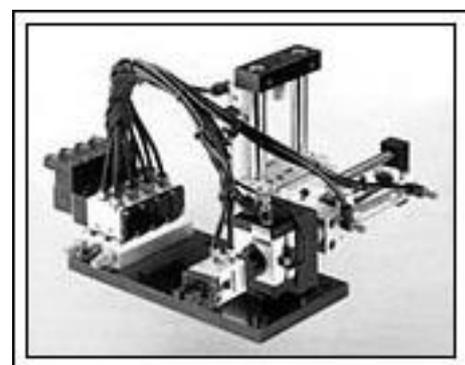


Figura 3.3: Outro robô industrial que utiliza o ar comprimido como energia

### Cilindros Pneumáticos

Os cilindros pneumáticos são formados por um câmara cujo interior abriga um êmbolo móvel. Dependendo do local de aplicação do ar, este êmbolo pode ser ejetado ou puxado. São mais usados quando se deseja posicionamentos extremos, ou seja, a câmara do cilindro totalmente rarefeita ou totalmente comprimida. Qualquer posicionamento intermediário requer o uso de detectores de posição e válvulas para controlar a distribuição de ar. Este tipo de acionamento é utilizado em robôs onde se busca movimentos rápidos, sem muita precisão e repetibilidade. Por terem um controle simples, são largamente empregados nos casos em que é suficiente o posicionamento em dois pontos distintos.



Figura 3.4: Cilindro pneumático em alumínio



Figura 3.5: Outros modelos

### Motores Pneumáticos

Os motores pneumáticos conseguem um movimento de rotação também mediante pressão do ar. Podem ser de dois tipos, os chamados de aletas rotativas e os de pistões axiais, sendo este primeiro tipo o mais utilizado.

Os motores do tipo aletas rotativas apresentam um rotor excêntrico onde estão dispostas aletas longitudinais variáveis. A entrada do ar comprimido num dos compartimentos formado por duas aletas e a carcaça, esta tende a girar até uma situação em que o compartimento tenha maior volume.

Já o modelo de pistões axiais têm um giro solidário a um tambor que se vê obrigado a girar pelas forças exercidas por vários cilindros que se apóiam sobre um plano inclinado.

Há ainda uma outra configuração que faz uso de cilindros acoplados a um sistema de pinhão-cremalheira.

### 3.2.2 Hidráulicos

São análogos aos pneumáticos, exceto que neste caso o flúido utilizado é mais denso e viscoso. A menor compressibilidade que o ar os torna mais adequados aos movimentos lentos, de maior precisão e ao manejo de grandes cargas. Apresentam estabilidade frente a cargas estáticas, ou seja, são capazes de suportar cargas (como um peso ou uma pressão exercida sobre uma superfície) sem aplicação de mais energia.

A principal desvantagens deste tipo de acionamento é a necessidade de dispositivos auxiliares. Sistemas hidráulicos necessitam de filtro de partículas, eliminadores de ar, sistemas de refrigeração, etc.

São empregados principalmente em robôs manipuladores de grande capacidade de carga.



Figura 3.6: Exemplos de pistões hidráulicos

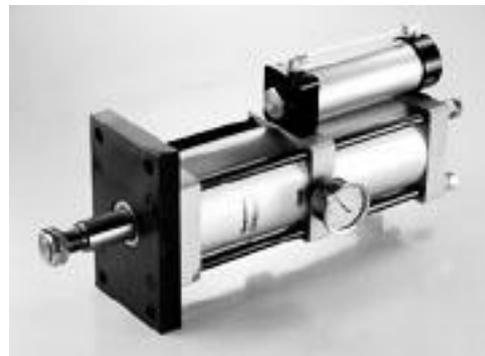


Figura 3.7: Outros modelos

Os primeiros robôs industriais construídos possuíam este tipo de acionamento, devido principalmente a uma boa relação entre peso e potência do robô.

### 3.2.3 Elétricos

Representam os mais comuns e utilizados atuadores em robótica móvel e industrial[26]. As ótimas características de controle, sensibilidade, precisão e confiabilidade fazem destes os preferidos no projeto e montagem de robôs, especialmente para fins educacionais.

### Motores Eletromagnéticos

Motores elétricos são máquinas que realizam a conversão de energia elétrica em energia mecânica.

Podem ser classificados de acordo com o número de fases de alimentação, polos do rotor, potência mecânica, frequência, tensão de alimentação, formato e material do qual são fabricados, etc. Todos estes fatores irão delinear a aplicação e as condições de funcionamento e trabalho deste equipamento.

Apesar desta grande variedade de características, motores para uso em robótica são agrupados em três categorias:

- motores de corrente alternada (AC)
  - síncronos
  - assíncronos (mais conhecidos por motores de indução)
- motores de corrente contínua (DC)
- motores de passo
- servomotores (DC)

**Corrente Alternada (AC)** São motores alimentados através de tensões alternadas senoidais. Podem ser trifásicos ou monofásicos. Motores trifásicos são mais utilizados no acionamento de robôs de grande porte no ambiente industrial, onde robustez e potência mecânica são exigidos. Já os monofásicos podem ser encontrados nas mais diversas aplicações, não possuindo um nicho específico.

São classificados em síncronos e assíncronos (ou de indução). Nos motores síncronos o indutor está situado no rotor (eixo) e é constituído por ímãs permanentes. O induzido está localizado no estator (carcaça, parte fixa) e é formado por três enrolamentos defasados  $120^\circ$  espacialmente, onde são aplicadas as tensões trifásicas defasadas também de  $120^\circ$  no tempo. A velocidade de giro do rotor é a mesma do campo girante produzido pela armadura (induzido, enrolamentos do estator), daí o nome síncrono. Esta velocidade é diretamente proporcional a frequência de alimentação e varia inversamente com o número de pares de pólos.

Os motores de indução (assíncronos) tem seu funcionamento baseado na indução de tensão no rotor. Este campo magnético produzido pelo enrolamento de campo no rotor se “acopla” ao campo girante produzido pela armadura e é “arrastado”. Assim, a velocidade de giro do rotor é um pouco menor que o campo girante da armadura. Esta diferença chama-se “escorregamento” e pode ser usada como uma medida do carregamento da máquina; quanto maior for a diferença, ou seja, maior o escorregamento, maior é a carga alimentada pelo motor. Esta diferença pode aumentar até um limite no qual a máquina perde o sincronismo e deixar de funcionar corretamente.



Figura 3.8: Motor AC trifásico comum



Figura 3.9: Típico motor AC monofásico

No passado, estes tipo de motores nunca tiveram grande aplicabilidade. Mas recentes avanços em microeletrônica e eletrônica de potência tornaram o uso destes competitivos frente aos motores de corrente contínua. Isto se deve principalmente a três fatores:

1. a construção de rotores síncronos sem escovas
2. uso de conversores estáticos que permitem variar a frequência (e com isso a velocidade) com precisão
3. emprego da microeletrônica que permite uma grande capacidade de controle

Em robótica móvel, motores AC raramente são utilizados. Isto deve-se principalmente a fonte de energia primária ser quase sempre uma bateria.

**Corrente Contínua (DC)** São motores que utilizam uma fonte de tensão contínua (DC) para funcionarem. Compõem-se de duas partes:

- o indutor, situado no estator, é responsável por criar o campo magnético de excitação

- o induzido, situado no rotor, gira devido ao campo criado pelo indutor e pelo próprio induzido, já que a corrente chega até este enrolamento através de uma estrutura chamada comutador. Este comutador é composto de contatos apoiados em escovas (geralmente de grafite) que fazem com que a polaridade da tensão se inverta após  $180^\circ$ , dando continuidade ao movimento.

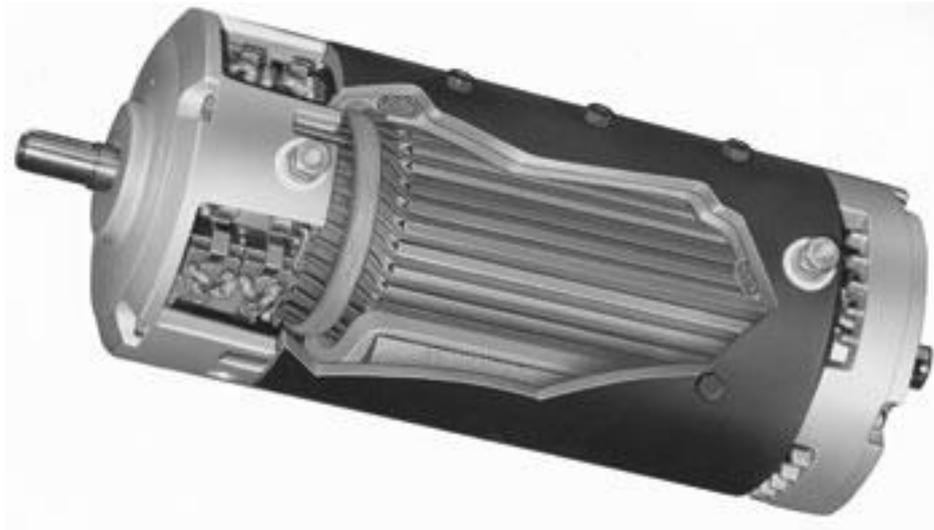


Figura 3.10: Corte parcial de um motor DC

Para que a conversão de energia ocorra neste tipo de motor, é necessário que os campos magnéticos do estator e do rotor permaneçam estáticos entre si, a conversão é maximizada quando estes campos estão em quadratura.

O aumento da velocidade pode ser feito aumentando-se a tensão do induzido (enrolamento do rotor) ou então diminuindo-se o fluxo magnético de excitação do indutor (enrolamento do estator).

A fim de melhorar o comportamento deste tipo de motor, costuma-se substituir o enrolamento do indutor por ímãs permanentes (feitos principalmente de samário-cobalto), evitando-se assim flutuações de tensão. Costuma-se também substituir o rotor bobinado por outro de metal fundido e ranhurado, diminuindo-se a inércia. Em contrapartida, esta medida costuma causar aumentos de temperatura no rotor do motor, devendo-se tomar precauções com relação a possíveis super-aquecimentos.

Estes motores são mais utilizados devido a facilidade de controle. São também os preferidos em robótica móvel; devido a fonte de energia primária ser quase sempre uma bateria.



Figura 3.11: Exemplos de motores DC

	<i>Pneumático</i>	<i>Hidráulico</i>	<i>Elétrico</i>
<b>Energia</b>	Ar comprimido (5-10 bar)	Óleo mineral (50-100 bar)	Corrente elétrica
<b>Opções</b>	Cilindros Motor de aletas Motor de pistão	Cilindros Motor de aletas Motor de pistão axial	Corrente contínua Corrente alternada Motor de passo
<b>Vantagens</b>	Baratos Rápidos Sensíveis Robustos	Rápidos Alta relação peso-potência Autolubrificantes Alta capacidade de carga Estabilidade frente à cargas estáticas	Precisos Confiáveis Fácil controle Fácil instalação Silenciosos
<b>Desvantagens</b>	Dificuldade de controle contínuo Instalação especial (filtros, compressores, etc) Ruidoso	Difícil manutenção Instalação especial (filtros, eliminadores de ar) Frequentes fugas Caros	Potência limitada

Tabela 3.1: Tabela comparativa dos diversos tipos de atuadores para robôs





Figura 3.15: Exemplo de um motor de passo



Figura 3.16: Outro modelo

desvantagens estão sua baixa potência nominal e falta de suavidade a baixas velocidades. Mas o maior inconveniente é a perda de passo devido a uma sobrecarga ou perturbação.

**Servomotores (DC)** Servomotores são motores DC que incorporam alguns dispositivos a fim de tornar seu uso mais simples. Apesar do termo *servomotor* ser usado em vários contextos diferentes, neste caso estamos nos referindo a um dispositivo DC com três terminais utilizado principalmente em controle de posição. Geralmente estes motores incluem engrenagens, limitadores de eixo, potenciômetro para a malha de realimentação de posição e pequenos circuitos integrados para o controle de posição.

São extremamente compactos e fáceis de controlar. Além disso, costumam custar mais barato que outros motores DC com engrenagens, devido a produção em massa.



Figura 3.17: Servomotor

### 3.2.4 Outras tecnologias

Representam uma nova categoria de dispositivos de acionamento. Atuadores elásticos como os chamados músculos artificiais e atuadores piezoelétricos começam a aparecer em determinadas aplicações especiais.

#### Músculos Artificiais

São polímeros eletroativos que com a ajuda de eletrodos fazem as fibras se contraírem ou expandirem num plano em resposta a um campo elétrico ou à uma força mecânica. Este mesmos polímeros podem produzir energia elétrica se forem muito deformados.

As boas características eletromecânicas aliadas à boa tolerância ao ambiente e durabilidade, sugerem aplicações como:

- microatuadores para microbombas
- válvulas
- músculos artificiais para robôs biomórficos
- robôs com asas
- geração de energia elétrica através do movimento do corpo

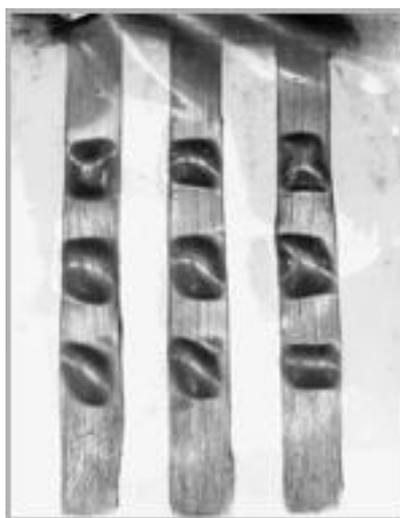


Figura 3.18: Fibras de um polímero eletroativo

Uma outra aplicação muito interessante é o uso deste tipo de tecnologia na construção de robôs com asas. A propulsão de um robô destes seria semelhante às asas de um pássaro.

Especialistas em aerodinâmica e robótica da Universidade de Toronto estão desenvolvendo pesquisas nesta área com o objetivo de projetar um robô para reconhecimento.



Figura 3.19: Protótipo de um robô “alado”

### Atuadores Piezoelétricos

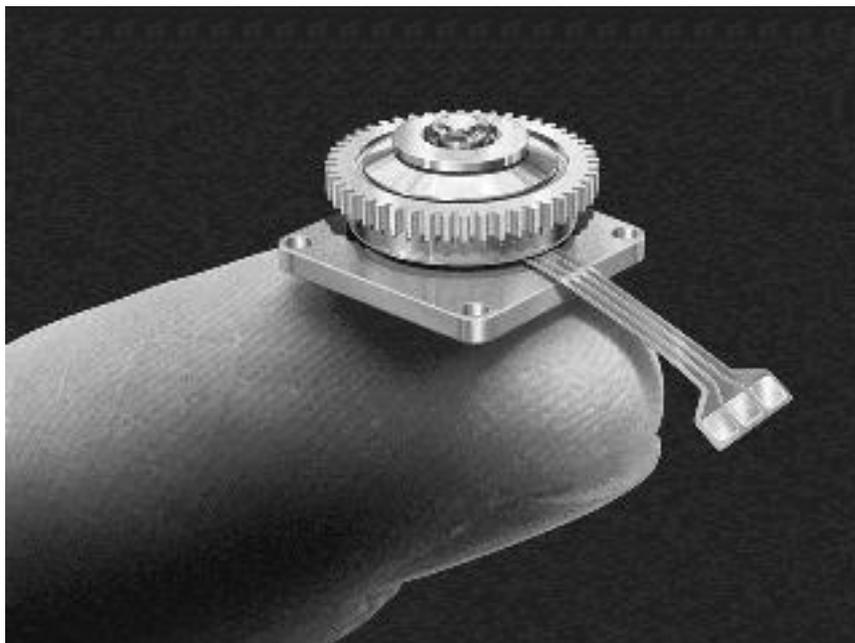


Figura 3.20: Exemplo de um motor piezoelétrico

O efeito piezoelétrico direto é aquele no qual uma cerâmica ou um cristal quando submetido a uma pressão mecânica gera uma tensão elétrica correspondente. O efeito inverso também é possível, ou seja, o material piezoelétrico quando sob efeito de um campo elétrico se deforma. As mudanças nas dimensões são função da intensidade do campo aplicado, assim como a intensidade e a polaridade da tensão produzida dependem da direção da força aplicada.

Este tipo de atuador pode ser encontrado, por exemplo, em lentes com foco automático de câmeras fotográficas, devido a excelente repetibilidade e precisão. Em robótica tem seu uso nos chamados robôs insetos.

Motores piezoelétricos, ao contrário do eletromagnéticos convencionais, giram a baixas velocidades e com grandes torques. Isto significa que eles podem ser compactos e leves, mas a fricção de acoplamento entre o rotor e o estator causa problemas de desgaste.

Principais características deste tipo de atuador:

- pequeno
- leve
- baixíssimo consumo de energia

Isto faz com que este tipo de dispositivo seja utilizado como micro-atuadores em áreas especiais como ótica, biomédica e aeroespacial.

### **3.3 Exercícios**

1. Como são classificados os atuadores utilizados em robótica ?
2. Quais os tipos de motores elétricos mais utilizados ? Por que ?
3. Em que tipo de aplicações atuadores hidráulicos são recomendados ?
4. Diferencie simples motores DC de servomotores DC.

# Capítulo 4

## Controle

### 4.1 Cinemática do Robô

#### 4.1.1 Introdução

A cinemática do robô estuda o movimento deste com respeito a um sistema de referência [5]. A cinemática se interessa pela descrição analítica do movimento espacial do robô como uma função do tempo, em particular pelas relações entre a posição e a orientação do extremo do robô (a extremidade de um braço manipulador, por exemplo).

Existem basicamente dois problemas fundamentais a serem resolvidos na cinemática do robô. O primeiro deles chamado **problema cinemático direto** consiste em determinar qual a posição e orientação do extremo final do robô com respeito a um sistema de coordenadas de referência, conhecendo-se os valores das articulações e os parâmetros geométricos dos elementos dos robôs. O segundo denominado **problema cinemático inverso** deve resolver o caminho inverso, ou seja, dada uma posição e orientação do extremo do robô, deve-se descobrir o valor das coordenadas das articulações.

Além disso, a cinemática do robô trata também de encontrar as relações entre a velocidade do movimento das articulações e dos extremos. Esta relação é dada pelo modelo diferencial, expresso através da matrix jacobiana.

#### 4.1.2 Cinemática Direta

Utiliza-se fundamentalmente álgebra vetorial e matricial para se representar e descrever a localização de um objeto no espaço tridimensional com relação a um sistema de referência

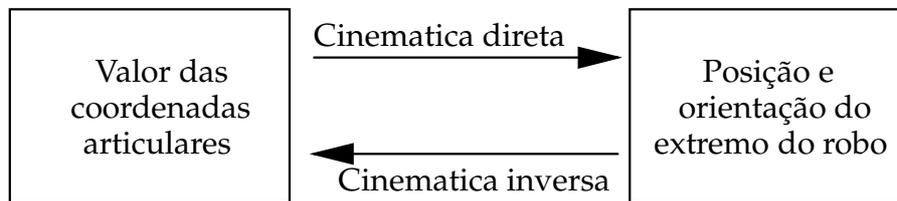


Figura 4.1: Diagrama de relação entre cinemática direta e inversa

fixo. Dado que um robô pode ser considerado uma cadeia cinemática formada por objetos rígidos unidos entre si por articulações, pode-se estabelecer um sistema de referência fixo situado na base do robô e descrever a localização de cada uma destas ligações com relação a este sistema de referência. Assim, o problema cinemático direto se reduz ao cálculo de uma matriz homogênea de transformação (chamada de  $T$ ) que relacione a posição e a orientação do extremo do robô com relação ao sistema de referência situado na base do mesmo. Esta matriz  $T$  será função das coordenadas articulares.

### Métodos de Resolução

Existem basicamente dois métodos de resolução do problema da cinemática direta.

- solução através de matrizes de transformação homogênea
- mediante o uso de quartetos

**Matriz de transformação homogênea** Chama-se de matriz de transformação homogênea a matriz que representa a posição e a orientação relativas entre as ligações consecutivas do robô. Por exemplo, a matriz  $A_1^0$  descreve a posição e orientação do sistema de referência da primeira ligação (haste) com respeito ao sistema de referência da base.  $A_2^1$  representa a matriz posição e orientação da segunda ligação com respeito à primeira.

Assim, um sistema com dois graus de liberdade teria uma matriz de transformação homogênea resultante igual ao produto de duas outras:

$$A_2^0 = A_1^0 A_2^1$$

Se considerarmos todos os graus de liberdade possíveis, teríamos uma matriz posição/orientação  $T$  igual a:

$$T = A_6^0 = A_1^0 A_2^1 A_3^2 A_4^3 A_5^4 A_6^5$$

Através de coordenadas cartesianas e ângulos de Euler podemos escrever equações que representem a posição e a orientação do extremo de um robô de seis graus de liberdade:

$$x = f_x(q_1, q_2, q_3, q_4, q_5, q_6)$$

$$y = f_y(q_1, q_2, q_3, q_4, q_5, q_6)$$

$$z = f_z(q_1, q_2, q_3, q_4, q_5, q_6)$$

$$\alpha = f_\alpha(q_1, q_2, q_3, q_4, q_5, q_6)$$

$$\beta = f_\beta(q_1, q_2, q_3, q_4, q_5, q_6)$$

$$\gamma = f_\gamma(q_1, q_2, q_3, q_4, q_5, q_6)$$

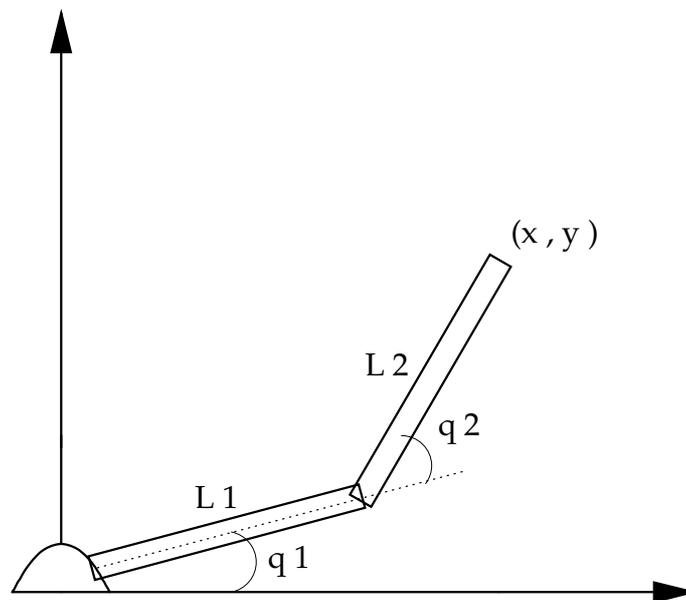


Figura 4.2: Exemplo de um robô planar com dois graus de liberdade

Para o caso ilustrado acima, simples considerações algébricas nos levam às seguintes equações:

$$x = L_1 * \cos(q_1) + L_2 * \cos(q_1 + q_2)$$

$$y = L_1 * \sen(q_1) + L_2 * \sen(q_1 + q_2)$$

Neste caso trivial, não é necessário utilizar-se da matriz homogênea. A utilização desta só se torna vantajosa com o aumento de graus de liberdade do sistema.

Embora se possa descrever a relação que existe entre os elementos contíguos através de qualquer sistema de referência ligado a cada elemento, a forma habitual que se utiliza em robótica é a representação de Denavit-Hartenberg (D-H). Denavit e Hartenberg propuseram um método matricial que permite estabelecer de maneira sistemática um sistema de coordenadas  $S_i$  ligado a cada elemento de uma cadeia articulada, podendo-se determinar a continuação das equações cinemáticas da cadeia completa.

Segundo esta representação, escolhendo-se adequadamente os sistemas de coordenadas associados a cada elemento, é possível passar de um para o outro mediante quatro transformações básicas que dependem exclusivamente das características geométricas da ligação. Estas transformações básicas consistem em uma sucessão de rotações e translações que permitem relacionar o sistema de referência do elemento  $i$  com o sistema do elemento  $i - 1$ .

As transformações citadas acima são as seguintes:

1. rotação ao redor do eixo  $z_{i-1}$  de um ângulo  $\theta_i$
2. translação ao longo de  $z_{i-1}$  de uma distância  $d_i$ ; vetor  $d_i(0, 0, d_i)$
3. translação ao longo de  $x_i$  de uma distância  $a_i$ ; vetor  $a_i(0, 0, a_i)$
4. rotação ao redor do eixo  $x_i$  de um ângulo  $\alpha_i$

$\theta_i$ ,  $a_i$ ,  $d_i$  e  $\alpha_i$  são conhecidos como parâmetros D-H da ligação  $i$ . De posse destes, obtemos as matrizes  $A$  e relacionamos assim todos a cada uma das ligações do robô.

É necessário seguir determinadas normas para o cálculo correto deste parâmetros. Não entraremos em mais detalhes nem na matemática envolvida por entendermos que este não é o objetivo deste texto. Maiores informações podem ser encontradas nas diversas referências em anexo.

**Resolução através do uso de quartetos** Este método, assim como o da matriz de transformação homogênea, é uma alternativa para representar transformações de rotação e deslizamento.

Consiste basicamente em obter uma expressão que permite conhecer as coordenadas da posição e orientação do sistema de referência associado ao extremo do robô ( $S_4$  por exemplo) com relação ao sistema de referência associado a base ( $S_0$ ). Esta relação será função dos

comprimentos de cada uma das ligações e das coordenadas articulares  $q_i$  (ângulos). Para obter a relação entre  $S_0$  e  $S_4$ ,  $S_0$  será convertido sucessivamente em  $S_1$ ,  $S_2$ ,  $S_3$  e  $S_4$ .

Terminada esta etapa, escreve-se as relações resultantes na forma vetorial e aplica-se uma série de equações e operações. Ao final de todo este processo, restará apenas uma equação que nos fornece um escalar (representando a posição) e o respectivo ângulo de rotação relativo ao sistema da base.

### 4.1.3 Cinemática Inversa

O objetivo da cinemática inversa consiste em encontrar valores para as coordenadas articulares do robô para que seu extremo se posicione e oriente segundo uma determinada localização espacial.

Assim como é possível abordar o problema cinemático direto de uma maneira sistemática a partir da utilização de matrizes de transformação homogênea e independente da configuração do robô, no problema cinemático inverso não ocorre o mesmo. Isto porque o procedimento de obtenção das equações é fortemente dependente da configuração do robô.

Existem alguns procedimentos genéricos que podem ser programados de modo que um computador pode, a partir do conhecimento da cinemática do robô (parâmetros D-H) obter uma n-upla de valores articulares que posicionem e orientem o extremo do robô. O inconveniente destes procedimentos é que se tratam de métodos numéricos iterativos, cuja velocidade de convergência e a própria convergência não estão garantidos.

Neste tipo de problema, é muito mais adequado encontrar uma solução fechada através de equações matemáticas. Este tipo de solução apresenta as seguintes vantagens sobre soluções iterativas:

- em muitas aplicações, o problema da cinemática inversa tem que ser resolvido em tempo real. Uma solução tipo iterativa não garante uma solução no momento adequado.
- ao contrário do que ocorre com o problema direto, com certa frequência a solução do problema cinemático inverso **não é única**, existindo diferentes n-uplas que posicionam o extremo do robô. Nestes casos uma solução fechada permite incluir determinadas restrições que assegurem que a solução obtida é a mais adequada entre as várias possíveis.

Apesar das dificuldades levantadas, a maior parte dos robôs possui cinemáticas relativamente simples que facilitam a resolução do problema inverso.

Por exemplo, se considerarmos somente os três primeiros graus de liberdade de muitos robôs, estes têm uma estrutura planar, isto é, os três primeiros elementos estão contidos num plano. Esta circunstância facilita a resolução do problema.

### Métodos de Resolução

São três os principais métodos de resolução de problemas de cinemática inversa.

1. Métodos geométricos
2. Matriz de transformação homogênea
3. Desacoplamento Cinemático

**Métodos geométricos** Este método é adequado para robôs com poucos graus de liberdade ou então quando se consideram apenas os primeiros graus dedicados a posicionar o extremo do robô.

Baseia-se em encontrar um número suficiente de relações geométricas entre as coordenadas articulares e as dimensões físicas de cada um dos elementos do robô. O processo de resolução é muito semelhante ao exemplo mostrado anteriormente de um robô planar de dois graus de liberdade, ou seja, são escritas equações que relacionam as variáveis  $x$ ,  $y$  e os ângulos  $q_1$ ,  $q_2$ ,  $q_n$ .

**Matriz de transformação homogênea** A princípio é possível obter o modelo cinemático inverso de um robô a partir do conhecimento de seu modelo direto. Supondo-se conhecidas as relações que expressam o valor da posição e orientação do extremo do robô em função de suas coordenadas articulares, obtemos por manipulação destas equações as relações inversas.

Apesar disto, na prática esta tarefa não é trivial, sendo em muitos casos tão complexa a ponto de ser rejeitada. Isto porque o problema cinemático direto resulta em doze equações (no caso de um robô com seis GDL). Caso estejamos interessados num número inferior de relações (cada grau de liberdade representa uma equação), existirão necessariamente certas dependências entre estas doze equações iniciais e aquelas que escolheremos para utilizar.

Este é um procedimento complicado, pois diversas manipulações matemáticas são necessárias. Resultados iguais são obtidos utilizando-se de métodos geométricos; só que com menos esforço.

**Desacoplamento cinemático** Os métodos vistos anteriormente permitem obter os valores das três primeiras variáveis articulares do robô, aquelas que posicionam seu extremo em determinadas coordenadas  $(p_x, p_y, p_z)$ ; embora se possa obter seis ao custo de uma maior complexidade.

Em geral, não basta posicionar o extremo do robô em um ponto do espaço, quase sempre é preciso também conseguir que a ferramenta na extremidade se oriente de maneira correta. Para tal, os robôs contam com outros três graus de liberdade adicionais, situados no final da cadeia cinemática e cujos eixos, geralmente, se interceptam num ponto, conhecido informalmente pela *mão do robô*. A variação destes três últimos graus de liberdade origina uma mudança na posição final do extremo do robô. O objetivo é poder orientar a ferramenta que está na *mão do robô* livremente no espaço.

Este método parte deste fato, ou seja, separando-se ambos os problemas: posição e orientação. Dada uma posição e orientação finais, estabelece-se as coordenadas *mão do robô* (interseção dos três eixos) e calcula-se os valores primeiras variáveis articulares que consigam posicionar este ponto. Depois, partindo-se dos dados de orientação obtidos anteriormente obtemos os valores do restante das variáveis articulares.

### Matriz Jacobiana

A modelagem cinemática do robô busca relações entre variáveis articulares e posição (normalmente expressas em coordenadas cartesianas) e a orientação do extremo do robô. Neste tipo de relação não se tem informações sobre as forças que atuam sobre o robô (atuadores, cargas, fricções) e que podem originar movimentos sobre o mesmo. Deve-se, portanto, conhecer além das relações entre as coordenadas articulares e seu extremo as relações entre suas respectivas derivadas.

Deste modo, o sistema de controle do robô pode estabelecer que velocidade deve imprimir a cada articulação para conseguir que o extremo do robô descreva uma trajetória temporal concreto, por exemplo, uma linha reta com velocidade constante.

Por isto, é de grande utilidade dispor de relações entre as velocidades das coordenadas

articulares e as posições e orientações da *mão do robô*. A relação entre ambos vetores de velocidade é obtida através da chamada **matriz jacobiana**.

A matriz jacobiana permite conhecer diretamente as velocidades do extremo do robô a partir dos valores de velocidades de cada articulação. Do mesmo modo, através da matriz jacobiana inversa se obtém as velocidades articulares conhecendo-se as velocidades do extremo do robô.

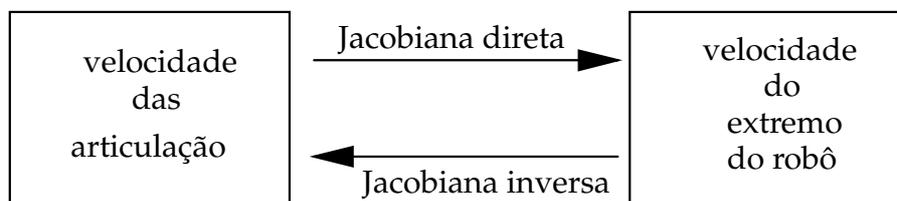


Figura 4.3: Matriz jacobiana direta e inversa

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix} = J \begin{bmatrix} \dot{q}_1 \\ \vdots \\ \dot{q}_n \end{bmatrix} \quad (4.1)$$

onde matriz jacobiana  $J$  vale:

$$J = \begin{bmatrix} \frac{\partial f_x}{\partial q_1} & \dots & \frac{\partial f_x}{\partial q_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_\gamma}{\partial q_1} & \dots & \frac{\partial f_\gamma}{\partial q_n} \end{bmatrix} \quad (4.2)$$

## 4.2 Dinâmica do Robô

A dinâmica se preocupa com as relações entre as forças que atuam no corpo e o movimento originado. Portanto, o modelo dinâmico do robô tem por objetivo conhecer a relação entre o movimento do robô e as forças aplicadas no mesmo.

Esta relação é obtida mediante o chamado modelo dinâmico, que relaciona matematicamente:

1. localização do robô definida pelas suas variáveis articulares e pelas coordenadas de localização de seu extremo, além de suas derivadas, velocidade e aceleração
2. forças aplicadas às articulações
3. dimensões do robô, como comprimento de suas hastes, massa e inércia de seus elementos

A obtenção deste modelo para mecanismos de um ou dois graus de liberdade não é excessivamente complexa mas, a medida que o número de GDL aumenta, a obtenção deste modelo dinâmico se complica enormemente. Por este motivo, nem sempre é possível obter-se um modelo dinâmico expresso de forma fechada, isto é, mediante uma série de equações, normalmente do tipo diferencial de segunda ordem, cuja integração permita conhecer que movimento surge ao aplicar umas forças e que forças se deve aplicar para se obter o movimento desejado. O modelo dinâmico deve ser resultado então de iterações mediante utilização algoritmos numéricos.

O problema da obtenção do modelo dinâmico de um robô é, portanto, um dos aspectos mais complexos da robótica, e que tem sido evitado em muitas ocasiões. Apesar disto, o modelo dinâmico é imprescindível se quisermos:

1. simular o movimento do robô
2. projeto e avaliação da estrutura do robô
3. dimensionamento dos atuadores
4. projeto e avaliação do controle dinâmico do robô

Este quarto item possui grande importância, posto que a qualidade do controle dinâmico do robô depende da precisão e velocidade de seus movimentos. A grande complexidade existente na obtenção do modelo dinâmico do robô tem motivado certas simplificações de modo que se possa utiliza-lo no projeto do controlador.

É importante notar que o modelo dinâmico completo do robô deve incluir não só a dinâmica de seus elemento (hastes), mas também o sistema de transmissão, dos atuadores e equipamentos eletrônicos de comando. Estes elementos incorporam ao modelo dinâmico novas inércias, fricções e saturações dos circuitos eletrônicos, aumentando ainda mais sua complexidade.

A seguir serão apresentados alguns métodos para se obter o modelo dinâmico do robô. Para simplificar, as hastes destes serão consideradas rígidas, sem deformações.

#### 1. Modelo dinâmico da estrutura mecânica de um robô rígido

Utilizando a formulação Lagrange-Euler

Utilizando a formulação Newton-Euler

**Modelo dinâmico da estrutura mecânica de um robô rígido** A obtenção deste modelo se baseia fundamentalmente na obtenção do equilíbrio de forças através da utilização da segunda Lei de Newton, e do seu equivalente para movimentos de rotação denominada Lei de Euler.

$$\begin{aligned}\sum F &= m\dot{v} \\ \sum T &= I\dot{\omega} + \omega(I\omega)\end{aligned}$$

Supondo-se que toda massa esteja concentrada no centro de gravidade do elemento, que não existe fricção alguma e que não exista carga.

- modelo dinâmico direto expressa a evolução temporal das coordenadas articulares do robô em função das forças sobre ele aplicadas
- modelo dinâmico inverso expressa as forças em função da evolução das coordenadas articulares e suas derivadas

No caso de um robô com cinco ou seis GDL, pode ser muito complicado obter estas equações. Como alternativa, utiliza-se a formulação Lagrangiana, que se baseia em considerações energéticas.

**Formulação de Lagrange-Euler** Este método utiliza a representação D-H baseada nas matrizes de transformação homogênea ( $A_i^{i-1}$ ) para formular o modelo dinâmico de um robô mediante equação de Lagrange. Se trata de um procedimento ineficiente do ponto de vista computacional, pois o algoritmo tem ordem de complexidade computacional  $n^4$ , ou seja, o número de operações a serem realizadas cresce com à potência 4 com o número de graus de liberdade.

Por outro lado, conduz a equações finais bem estruturadas onde os diversos pares de forças são mostrados de maneira clara.

**Formulação de Newton-Euler** Este método surge como alternativa à formulação Lagrangiana, posto que para um robô com seis GDL, sua utilização em tempo real é quase impossível.

Baseia-se na obtenção recursiva de posição, velocidade e aceleração da haste  $i$  referida a base do robô a partir da haste  $i - 1$ . Deste modo, partindo-se da ligação 1 se chega à haste  $n$ . Com estes dados, pode-se obter as forças atuantes sobre a haste  $i$  referidas a base do robô.

Este algoritmo se baseia em operações vetoriais mais eficientes que as operações matriciais da formulação Lagrangiana. Além disso, a ordem de complexidade computacional é  $n$ , ou seja, depende diretamente do número de graus de liberdade.

### 4.3 Controle Cinemático

Nas seções anteriores foi visto como levantar os modelos cinemáticos e dinâmicos de um robô. A partir destas informações é possível estabelecer as adequadas estratégias de controle do robô que redundem em uma otimização e maior qualidade de seus movimentos.

Com o controle cinemático é possível determinar quais as trajetórias que devem seguir cada articulação do robô através do tempo para alcançar os objetivos fixados pelo usuário (ponto de destino, trajetória cartesiana). Estas trajetórias foram selecionadas atendendo às restrições físicas próprias dos acionamentos e a certos critérios de qualidade de trajetória, como suavidade e precisão da mesma.

#### 4.3.1 Funções do controle cinemático

Na figura 4.4 dá para ter a idéia exata de funcionamento do controle cinemático. O controlador tem como entrada os dados procedentes do programa do robô escrito pelo usuário (ponto de destino, precisão, tipo de trajetória desenhada) e, baseado no modelo cinemático do robô, estabelece as trajetórias para cada articulação como funções no tempo. A seguir, estas trajetórias devem ser amostradas em um período  $T$ , devidamente escolhido, gerando a cada instante  $kT$  um vetor de posições articulares para o algoritmo de controle dinâmico.

Segundo [5], de maneira geral, o controle cinemático deverá realizar as seguintes funções:

1. Converter a especificação do movimento, dada no programa em uma trajetória analítica, em espaço cartesiano (evolução de cada coordenada em função do tempo).

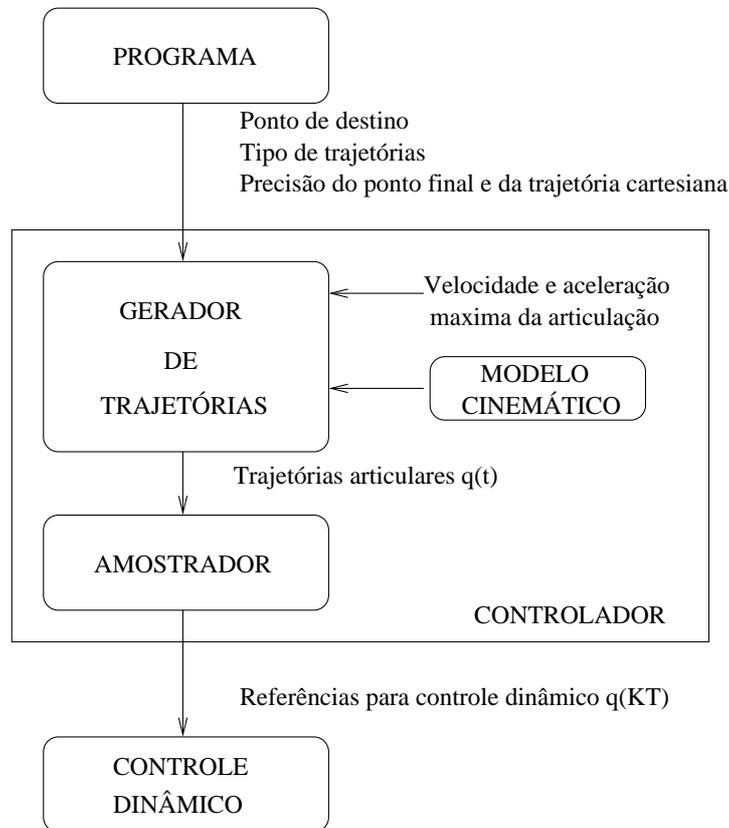


Figura 4.4: Funcionamento do controle cinemático

2. Amostrar a trajetória cartesiana obtendo um número finito de pontos da dita trajetória. Cada um destes pontos será dado um 6-upla, tipicamente  $(x, y, z, \alpha, \beta, \gamma)$ .
3. Utilizando a transformação homogênea inversa, converter cada um dos pontos em suas correspondentes coordenadas articulares. Deve se ter em conta aqui uma possível solução múltipla da transformação, assim como a possibilidade de ausência de solução e pontos singulares, de modo que se assegure a continuidade da trajetória.
4. A interpolação dos pontos articulares obtidos, gerando para cada variável articular uma expressão  $q_i(t)$  que passe ou se aproxime deles de modo que, sendo uma trajetória realizável pelos atuadores, se transforme em uma trajetória cartesiana o mais próxima possível da especificada pelo programa do usuário.
5. Amostrar a trajetória articular para gerar referências ao controle dinâmico.

A figura 4.5 mostra um possível caso onde o objetivo de um robô com dois graus de

liberdade é sair do ponto  $j^1 = (x^1, y^1)$  e alcançar o ponto  $j^4 = (x^4, y^4)$ , dentro de um tempo  $T$ . Para tanto, quatro pontos dentro da trajetória são selecionados pelo controle cinemático,  $j^1, j^2, j^3, j^4$ , que são submetidos a transformação homogênea inversa, gerando os respectivos vetores articulares  $q^1, q^2, q^3, q^4$ . Segundo algum tipo de função estes pontos são unidos de forma a gerar uma trajetória que respeite as restrições de cada acionador, tais como velocidade e aceleração máxima permitida. O resultado é uma trajetória que busca se aproximar o mais próximo possível de uma linha reta.

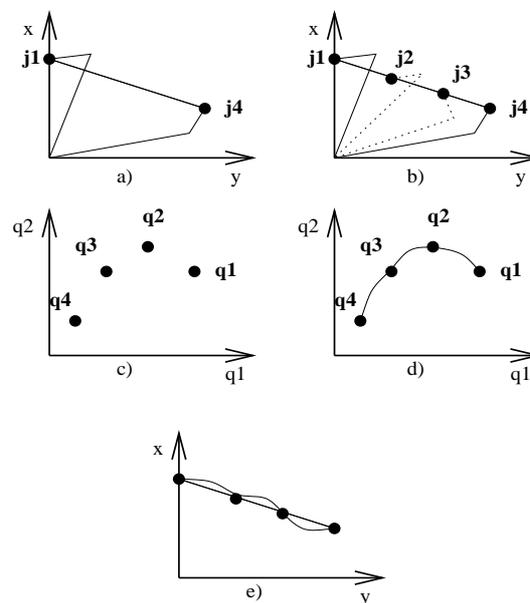


Figura 4.5: Atuação de um controle cinemático em um robô com dois graus de liberdade

Assim, dentro do projeto de um controlador cinemático para um robô, um dos pontos passíveis de decisão é quanto a que interpolador usar, de modo a unir os pontos estabelecidos pelos vetores articulares. Feito isto, ainda deve-se estabelecer os critérios para selecionar quantos e que pontos serão amostrados na trajetória no espaço cartesiano.

### 4.3.2 Tipos de trajetórias

A tarefa de mover um robô de um ponto inicial até um ponto final deve ser resolvida levando em consideração a possibilidade de terem-se infinitas trajetórias espaciais. A melhor forma pode ser escolhida segundo alguns critérios, como por exemplo, a facilidade de implementação, menor custo computacional, ou sua utilidade e aplicação em diversas áreas. Desta feita podemos encontrar robôs que disponham de trajetórias ponto a ponto,

trajetórias coordenadas ou trajetórias contínuas.

### **Trajетórias ponto a ponto**

Neste tipo de trajetória o objetivo é levar cada articulação desde o ponto inicial até o ponto final sem levar em consideração qualquer informação relativa ao estado ou relativo a evolução das demais articulações. Para isso cada articulação tenta alcançar sua meta de maneira otimizada, onde a movimentação de eixos é distingüida entre os seguintes casos:

- Movimento eixo a eixo: Neste tipo de operação um eixo move-se de cada vez. Assim a primeira articulação busca seu ponto final e só aí a segunda articulação dá início ao seu movimento. Este tipo de operação garante um menor consumo de energia, mas, em contrapartida, exige um maior tempo de ciclo.
- Movimento simultâneo de eixos: Neste caso, todas as articulações iniciam seus movimentos no mesmo instante de tempo em uma velocidade específica para cada uma delas. Uma vez que as distâncias a serem percorridas por cada eixo são diferentes, cada um finalizará seu movimento em momentos diferentes.

Devido a sua simplicidade, as trajetórias ponto a ponto dificilmente são implementadas, salvo em casos de robôs muito simples.

### **Trajетórias coordenadas**

Para que os atuadores não trabalhem no limite de suas velocidades e acelerações a todo instante, fazendo com que alguns eixos terminem seus movimentos antes de alguns mais lentos, é interessante efetuar um cálculo *a priori* garantindo que haverá uma coordenação entre os movimentos dos eixos, fazendo com que estes cheguem aos seus pontos finais no mesmo instante de tempo.

Nota-se, que neste caso, o movimento dos eixos adaptam-se àquele mais lento, de forma a não perder-se potência em outros eixos de maneira inútil.

### **Trajетórias contínuas**

Em alguns casos é interessante que a trajetória a ser seguida pelo atuador seja conhecida pelo usuário. O cálculo contínuo das trajetórias articulares garantem esta requisição de projeto.

Normalmente as trajetórias desejadas pelos usuários são em linha reta ou aquelas em forma de arco de círculo. Para tal objetivo cada articulação seguirá movimentos aparentemente caóticos com possíveis trocas de direção e velocidade sem qualquer tipo de coordenação

com as demais articulações. Ao final, o robô apresentará a trajetória desejada.

### 4.3.3 Geração de trajetórias cartesianas

Para que um usuário tenha condições de especificar o movimento desejado para o robô, o faz através da determinação das coordenadas espaciais dos pontos iniciais e finais do movimento, e eventualmente, junto com outros dados como instantes de passo, velocidades e tipos de trajetória.

Em não raras excessões, estes pontos podem estar criticamente distante um dos outros, sendo necessário a seleção de pontos intermediários suficientemente próximos para que o controle do robô consiga não só ajustar o ponto final ao especificado, mas também obedecer a trajetória indicada no programa do usuário.

Segundo [5], é preciso estabelecer um interpolador entre as localizações expressas no espaço da tarefa que dará como resultado uma expressão analítica da evolução de cada coordenada. A interpolação mais freqüente é a linear, em que cada coordenada evolui a velocidade constante desde seu valor inicial  $j^i$  até o seu valor final  $j^f$ :

$$j(t) = (j^f - j^i) \frac{t - t_i}{t_f - t_i} + j^i \quad (4.3)$$

onde  $t_i$  e  $t_f$  são os instantes de tempo em que se pretende alcançar as localizações inicial e final, respectivamente.

### 4.3.4 Interpolação de trajetórias

Já ficou definido que uma das funções do controle cinemático é a de unir uma sucessão de pontos no espaço articular de uma trajetória pela qual deseja-se que passem as articulações do robô em um instante de tempo. Seria interessante também que junto com as condições de posição-tempo, sejam levantadas as restrições de velocidade e aceleração, de maneira que se assegure a suavidade da trajetória e se limitem as velocidades e acelerações máximas. Estas restrições garantem que os atuadores estão capacitados para implementar a trajetória final.

Para isto é preciso selecionar-se algum tipo de função cujos parâmetros ou coeficientes se ajustaram ao impor-se as condições de contorno: posições, velocidades e acelerações. Na seleção desta função deve-se considerar que tanto o cálculo de seus parâmetros, como sua posterior utilização para gerar pontos ao controle dinâmico, deve fazer-se em tempo real, em que a simplicidade da função será um fator a se considerar.

Dentre as interpolações conhecidas, destacamos:

- **Interpoladores Lineares:** para o caso em que se pretende que uma das articulações do robô, passe sucessivamente pelos valores  $q^i$  nos instantes  $t^i$ , mantendo, por exemplo, a velocidade do movimento constante.
- **Interpoladores Cúbicos:** a trajetória que une os pontos pelos quais a articulação deve passar, é dado por um polinômio de grau 3 unindo os pares de pontos adjacentes. O uso de polinômios de terceiro grau permite assegurar que o polinômio passa pelos pontos e, ao mesmo tempo, permite impor os valores de velocidade para os mesmos.
- **Interpoladores em degrau:** Uma alternativa que proporciona uma solução intermediária consiste em decompor em três degraus consecutivos a trajetória que une os pontos  $q^0$  até  $q^1$ . No degrau central utiliza-se um interpolador linear, mantendo a velocidade constante, não sendo preciso imprimir aceleração alguma ao atuador. Nos degraus inicial e final, utiliza-se um polinômio de segundo grau, de modo que no degrau 1 a velocidade varie linearmente desde a velocidade da trajetória anterior até a atual, e no degrau 3 varie desde a velocidade atual até a seguinte. Tem-se assim que nos degraus inicial e final a aceleração assume valores constantes distintos de zero, enquanto no degrau intermediário a aceleração é nula.

Em [5] pode ser verificado uma apuração matemática dos interpoladores aqui destacados.

#### 4.3.5 Amostragem de trajetórias cartesianas

Depois de determinada as trajetórias a serem percorridas pelo robô através do processo de interpolação, cabe estabelecer critérios que definam uma seleção ótima de pontos da trajetória cartesiana.

As trajetórias mais habituais são a em linha reta e as trajetórias circulares. Em um primeiro momento poderia supor-se que quanto maior o número de pontos a serem selecionados, melhores resultados seriam obtidos, mas não é tão simples assim. O grande problema aqui é o custo computacional envolvido em tal processo, limitando a necessária resposta em tempo real do sistema de controle do robô, tornando evidente o problema gerado pela tomada indiscriminada de pontos.

Por outro lado, um número insuficiente de pontos, agregado a não linearidade no modelo geométrico do robô, dependendo de sua cinemática e da posição instantânea em questão, pode gerar um erro considerável na trajetória final. A solução para este problema reside no compromisso entre estes dois pontos.

#### 4.4 Controle Dinâmico

Para que as invariáveis exigências de velocidade e aceleração dos atuadores de um robô sejam atendidas, sem a deterioração da qualidade de seus movimentos algumas, investigações e avanços tecnológicos tem-se feito nesta área. Como produto desta política tem surgido materiais mais rápidos, sistemas de movimento sem folga e com baixa inércia, atuadores mais rápidos e sensores de elevada precisão.

Mesmo com estes avanços, é importante a existência de algoritmos de controle com o objetivo de melhorar ao máximo as características de velocidade e precisão. Na seção anterior viu-se que o controle cinemático encarrega-se de selecionar as trajetórias a serem cursadas, levando em conta as limitações do conjunto, e ajustando-se da melhor maneira possível as especificações do usuário. Mesmo assim, na prática, os ajustes dos movimentos do robô as especificações do usuário não serão possíveis, uma vez que as características dinâmicas do robô, como inércia, atritos e folgas, na maioria da vezes desconhecida, impedem geralmente a coincidência entre a trajetória desejada  $q_d(t)$  e a trajetória real  $q(t)$ .

É esta lacuna que o controle dinâmico tenta preencher, ou seja, é objetivo do controle dinâmico fazer com que estas duas trajetórias sejam o mais próximas possível. Para tanto o controlador faz uso do modelo dinâmico levantado e do uso de uma estratégia de controle adequada (controle PID, controle adaptativo).

Uma vez que o modelo dinâmico do robô é altamente não linear, multivariável, acoplado e de parâmetros variáveis, seu controle é extremamente complexo. Na prática algumas simplificações são aceitas, facilitando o desenho do sistema de controle, com resultados aceitáveis.

A utilização de estratégias de controle mais potentes muitas vezes não compensam, devido a fatores como custo computacional e até mesmo um elevado custo econômico. Tudo depende do tipo de problema com o qual se está tratando.

Normalmente o controle dinâmico dá-se em torno do espaço articular, ou seja, controlando as trajetórias  $q(t)$  geradas pelo controle cinemático do robô. A seguir serão apresentadas

as técnicas de controle dinâmico mais utilizadas focalizando principalmente o controle no espaço articular.

#### 4.4.1 Controle Monoarticular

Alguns fatores determinam a estratégia de controle a ser utilizada, como por exemplo, dimensões, peso e tipo de atuadores. Segundo [5] o modelo dinâmico de um robô tem o diagrama de blocos apresentado na figura 4.6.

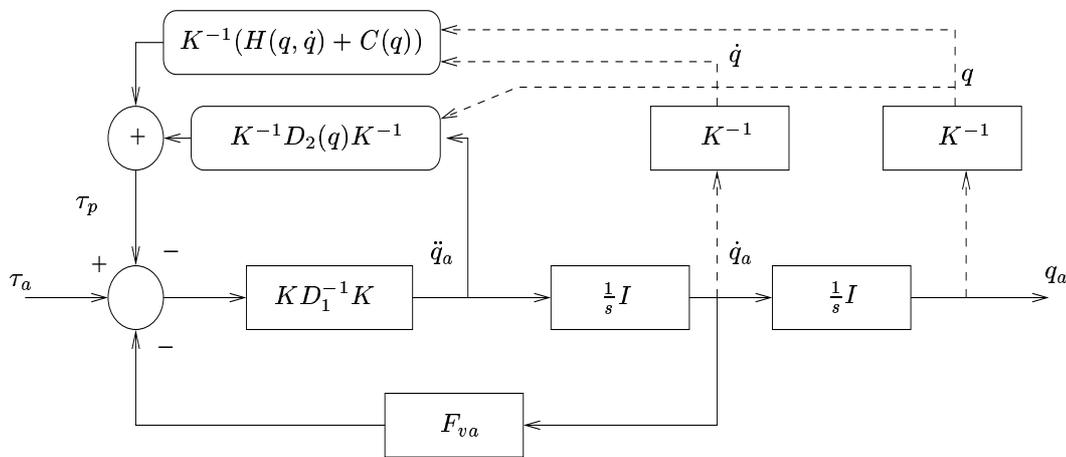


Figura 4.6: Diagrama de blocos que representa o comportamento dinâmico de uma articulação

No diagrama, podemos observar a presença de alguns fatores importantes para o modelo como:

- $F_{va}$ : uma matriz diagonal representando o atrito viscoso dos atuadores.
- $\tau_p$ : um sinal de perturbação.
- $K$ : uma matriz com os fatores de redução de cada atuador.
- $D$ : matriz de inércia.
- $H$  e  $C$ : matrizes levantadas a partir do modelo cinemático do robô.

A partir do diagrama é possível levantar as equações que descrevem o modelo dinâmico do robô, onde os blocos ovalados dependem das variáveis que chegam neles e os blocos retangulares são formados por matrizes diagonais constantes. Para uma análise matemática matricial mais profunda consulte [5].

### 4.4.2 Controle PID

Um dos controladores mais comuns na prática é o controlador PID, onde PID representa as iniciais de proporcional, integral e derivativo. Com este tipo de controle é possível anular o efeito de uma possível perturbação em regime permanente, assim como assegurar que finalmente  $q$  e  $q_d$  coincidam, sem, no entanto, conseguir que  $q$  evolua junto com  $q_d$  a todo momento.

Na figura 4.7 o controlador PID é formado pelo bloco  $R(s)$ . Para uma análise, tanto no domínio do tempo como no domínio de frequência, mais acurada consulte [29].

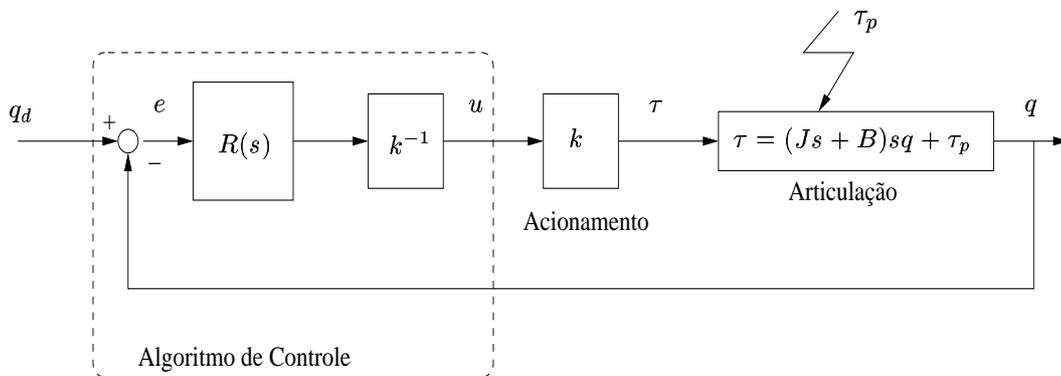


Figura 4.7: Esquema de algoritmo de controle baseado em PID

A articulação é representada de forma simplificada pela expressão  $\tau = (Js + B)sq + \tau_p$ , onde  $J$  e  $B$  são respectivamente a inércia e o atrito visto pelo atuador. Esta simplificação é possível graças a uma linearização entre o valor de entrada e o acionamento.

### 4.4.3 Controle PID com pré-alimentação

A utilização do controle PID permite anular o efeito de uma perturbação em regime permanente, assim como conseguir que a saída  $q(t)$  alcance a entrada em um tempo aceitável. Entretanto, seria interessante que uma dada trajetória  $q_d(t)$  de entrada, coincidissem com a saída o tempo todo, alcançando não somente um erro nulo em regime permanente mas também um erro nulo no período transitório do sinal.

Para conseguir tal efeito é possível utilizar um controle PID com pré-alimentação, baseado em um preciso conhecimento da articulação ( $J$  e  $B$ ). A figura 4.8 representa um esquema de controle PID com pré-alimentação, onde o sinal  $u$  é acrescentado de uma pré-alimentação

de velocidade e aceleração.

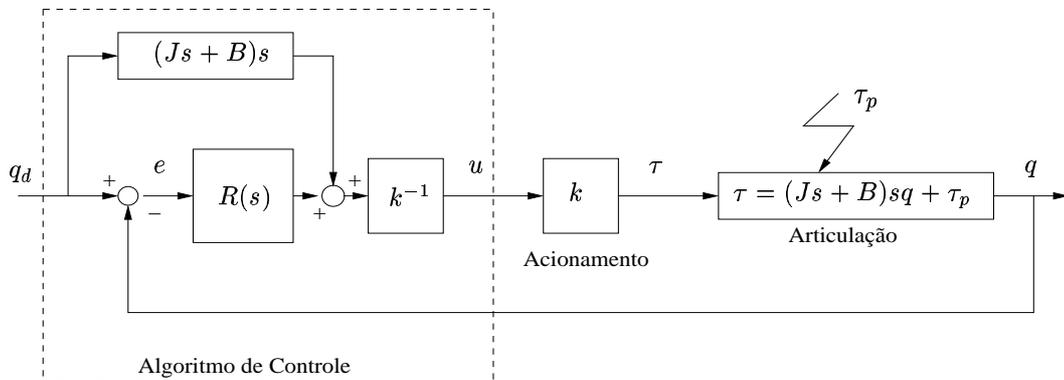


Figura 4.8: Esquema de algoritmo de controle baseado em PID com pré-alimentação

É importante ressaltar que a coincidência absoluta entre os sinais  $q(t)$  e  $q_d(t)$  diante de perturbações, só é possível caso  $J$  e  $B$  sejam perfeitamente conhecidos. Na prática, o que se tem é uma boa estimativa destes valores tendo que admitir-se certos desvios entre  $q$  e  $q_d$ .

#### 4.4.4 Controle Adaptativo

Se o modelo dinâmico do robô é perfeitamente conhecido, as técnicas de controle apresentadas anteriormente são as mais indicadas as serem utilizadas a fim de garantir que a trajetória real acompanhe a trajetória desejada. No entanto, uma vez que o modelo é prisioneiro da exigência de que é necessário o completo conhecimento dos parâmetros (inércia, atritos) esta técnica passa a apresentar uma certa limitação.

Assim surge o controle adaptativo, que encontra aplicação onde os parâmetros do modelo ou não são conhecidos ou mudam continuamente as condições de funcionamento. A idéia básica do controle adaptativo é modificar em tempo real os parâmetros que definem o regulador (um PID por exemplo) de acordo com o comportamento instantâneo do sistema.

Existem diferentes esquemas de controle que enquadram-se dentro do conceito de controle adaptativo. Em comum estes esquemas apresentam técnicas de identificação de parâmetros do modelo do sistema e algoritmos de cálculo dos parâmetros do regulador.

Um exemplo de controle adaptativo é o controle por planificação de ganhos, que pode ser considerado um caso simples de controle adaptativo em que um número determinado de reguladores estão pré-calculados para diferentes condições de funcionamento. Uma vez detectado modificações no sistema comuta-se de um controlador para outro.

---

## 4.5 Exercícios

1. O que são matrizes de transformação homogênea ?
2. Por que o problema da cinemática inversa é mais difícil de ser resolvida que a direta ?
3. O que torna formulação Lagrange-Euler inadequada computacionalmente ?
4. Identifique as funções do controle cinemático
5. Descreva possíveis aplicações para um controlador PID, e aplicações onde o controle adaptativo apresentaria melhores resultados.

## Capítulo 5

# Arquitetura de Robôs Móveis

### 5.1 Introdução

Na literatura de robôs móveis não é comum encontrar material de cunho genérico a respeito das possíveis arquiteturas existentes. Em sua maioria, este material apresenta uma única arquitetura, descrevendo suas características, vantagens e, geralmente, um estudo de caso.

Este capítulo tem por motivação cobrir esta lacuna, apresentando as mais variadas características de uma possível arquitetura por intermédio de uma classificação, que com certeza não tem nenhuma ambição de ser uma classificação definitiva. Ela apenas é apresentada no sentido de organizar as mais diferentes características de uma arquitetura.

Assim, na seção seguinte é apresentada uma proposta de classificação onde características conflitantes são comparadas, e na seção final duas das principais arquiteturas existentes e amplamente utilizadas, são descritas.

### 5.2 Definições e Classificações

É possível classificar as arquiteturas de robôs móveis segundo alguns aspectos [37], dentre eles:

- Estrutura
  - Centralizadas
  - Distribuídas

- Raciocínio
  - Reativas
  - Deliberativas
  - Híbridas
- Decomposição e encapsulamento
  - Baseadas em comportamentos
  - Baseadas em módulos funcionais

### 5.2.1 Arquiteturas Centralizadas e Distribuídas

Em robôs com uma estrutura *centralizada* as tomadas de decisões são de âmbito local. Os dados necessários para processamento, constituído das suas informações sensoriais, estão no próprio robô. Os métodos usados para o processamento da informação do robô podem ser aqueles baseados em técnicas desenvolvidas na IA simbólica, utilizando algum método de representação de conhecimento, ou até mesmo, técnicas conexionistas através do uso de redes neurais, porém não limitado a estes casos. A abordagem centralizada é o caso em que um robô age de forma isolada no ambiente. Independente das outras características de sua arquitetura, ele age de forma autônoma baseado única e exclusivamente nas informações colhidas durante sua incursão no ambiente. Não há nenhum tipo de comunicação com outras entidades do meio, e as informações por ele processadas surgem da interação direta com os objetos do ambiente, sejam eles passivos ou ativos, estáticos ou dinâmicos.

A idéia de *distribuição* vem no momento em que um robô não age de forma isolada no seu ambiente. Na verdade, ele faz parte de uma *sociedade de agentes* onde cada entidade tem seu papel a ser exercido. As idéias deste tipo de abordagem vêm da *Inteligência Artificial Distribuída* (IAD) que estuda o conhecimento e os métodos de raciocínio que podem ser necessários ou úteis para que agentes/robôs participem de sociedades de agentes/robôs [7].

A IA distribuída faz uso constante da idéia de *agente* que, no entanto, não apresenta uma definição única aceita pela comunidade. Uma possível definição é proposta por [22]:

Chama-se agente uma entidade real ou abstrata que é capaz de agir sobre ela mesma e sobre seu ambiente, que dispõe de uma representação parcial deste ambiente, que, em um universo multiagente, pode comunicar-se com outros agentes

e cujo comportamento é consequência de suas observações, de seu conhecimento e das interações com outros agentes.

Esta definição se preocupa principalmente com os mecanismos internos de tratamento de cada agente, sem estabelecer, por exemplo, o tipo de comunicação possível entre agentes. Nada é dito também sobre a granularidade dos agentes. Uma definição complementar à primeira, que ressalta o aspecto *identidade* do agente, foi proposta por [23]:

Um agente é uma entidade à qual se pode associar uma identidade única, e que é capaz de realizar cálculos formais. Um agente pode ser considerado como um meio que produz um certo número de ações a partir dos conhecimentos e mecanismos internos que lhe são próprios.

Dentro do tópico de robôs móveis, podemos interpretar estes como sendo a tal “entidade real” citada pela primeira definição ou a entidade “à qual se pode associar uma identidade única” citada na segunda definição. Assim sendo, um robô nada mais é que a implementação física de um agente inserido em um ambiente e capaz de agir no mesmo.

Existem vários motivos para distribuir sistemas inteligentes. O principal deles é que alguns domínios de aplicação são inerentemente distribuídos no espaço. Outras razões incluem:

- Melhorar a adaptabilidade, a confiabilidade e a autonomia do sistema.
- Reduzir os custos de desenvolvimento e manutenção.
- Aumentar a eficiência e a velocidade
- Permitir a integração de sistemas inteligentes existentes de maneira a aumentar a capacidade de processamento e, principalmente, a eficiência na solução de problemas.
- Permitir a integração das máquinas nas redes de atividades humanas.

Além dessas razões, deve-se ainda salientar que, para problemas realmente grandes, a única possibilidade de solução é a solução distribuída, o que coloca a idéia de distribuição como a única técnica possível quando o problema ultrapassa um certo grau de complexidade. Finalmente, o fato de a maioria das atividades inteligentes dos seres humanos envolverem mais de uma pessoa deveria ser motivo suficiente para que os aspectos sociais da inteligência fossem colocados entre os fundamentos de qualquer teoria da inteligência [7].

Por razões históricas, a idéia de distribuição na execução de tarefas se dividiu em dois enfoques: (i) *solução distribuída de problemas* (SDP) (do inglês “distributed problem solving”) e (ii) *sistemas multiagentes* (SMA) (do inglês, “multi-agent systems”) [19].

A SDP tem como foco principal o *problema*, conforme a tradição na IA simbólica, da qual este enfoque é diretamente derivado. Seus objetivos são utilizar a capacidade de processamento e a robustez oferecidas pela tecnologia de redes para atacar problemas naturalmente distribuídos ou excessivamente complexos. Para a SDP, os agentes são pré-programados para cooperar, e seus métodos visam garantir que esta cooperação ocorra de maneira coerente, robusta e eficiente. A qualidade de um sistema de SDP é medida pelo desempenho global do sistema na solução de um problema específico.

Os agentes envolvidos em SDP são programados para cooperar, dividir tarefas, comunicar-se de maneira confiável, etc. No entanto experiências em ciências sociais mostram que não é simples estabelecer tais propriedades em uma coleção de indivíduos.

O estudo das pressuposições básicas sobre agentes que garantam a possibilidade de ação cooperativa em sociedade é o foco de estudo dos SMA, isto é, neste caso o foco das pesquisas é o *agente*. O estudo de SMA é naturalmente multidisciplinar e envolve conceitos provenientes de diversas disciplinas, por exemplo, economia, teoria de jogos, ciências sociais, etologia, etc. Dentre estes conceitos, o de *agente racional*, isto é, aquele que age no sentido de maximizar seus benefícios e minimizar suas perdas, apresenta especial interesse para a área de SMA.

### 5.2.2 Arquiteturas Reativas, Deliberativas e Híbridas

Quanto ao método de raciocínio as arquiteturas dividem-se em *reativas* e *deliberativas*. Um terceiro tipo surge com a combinação das duas anteriores, os sistemas híbridos, visando explorar as vantagens dos dois mecanismos de raciocínio, e simultaneamente tentar diminuir, ou amenizar, suas desvantagens.

Entende-se por *reação* a forma como alguém ou algo comporta-se diante de um determinado estímulo, e por *reativo* aquilo que produz uma reação. Como exemplo, imagine o caso em que desapercebidamente colocamos a mão sobre um corpo extremamente quente. Uma ação de reflexo imediata faz com que a afastemos do corpo aquecido, sem que tal ação seja comandada pelo cérebro, já que não há problema a ser resolvido neste caso: uma vez que a integridade de nosso corpo está em risco uma atitude que garanta a nossa sobrevivência de-

va ser tomada. Um caso típico em que uma condição sensorial (temperatura elevada) exige uma ação imediata (o afastamento de nosso mão). Assim, a *reação*, neste caso, foi afastar a mão da superfície quente, e a temperatura elevada foi seu estímulo *reativo*.

Uma arquitetura reativa é composta por uma série de comportamentos que relacionam certas condições sensoriais a um conjunto de ações do robô, além de ser dotada de métodos que garantam a devida coordenação destes comportamentos. A figura 5.1 apresenta um descrição simples de uma arquitetura reativa. Em geral, esta arquitetura produz robôs adequados para operação em tempo real, já que a simplicidade dos comportamentos reativos favorece uma alta velocidade de processamento computacional.

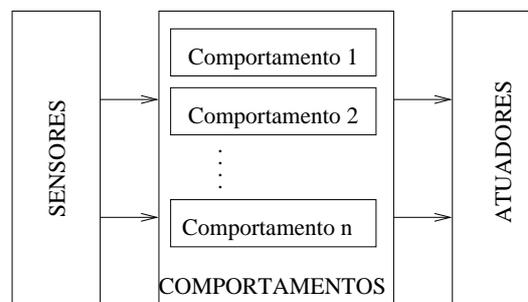


Figura 5.1: Esquema genérico de uma Arquitetura Reativa

*Planejar* (ou *deliberar*) uma tarefa consiste em determinar as ações e seqüências necessárias para conseguir sua execução no contexto de um certo modelo de mundo. Assim um robô inteligente pode tentar construir um modelo interno representando o ambiente no qual está inserido. Se um problema fosse então designado ao robô, este poderia tentar primeiro explorar o espaço de soluções obtidos a partir de seu modelo interno, gerando um plano de atividades e, então, realizar a ação física propriamente dita, através de seus atuadores. Então para que um planejamento seja ótimo é necessário um *modelo do ambiente* o mais perfeito possível. Este modelo é obtido através da fusão das informações sensoriais numa estrutura de dados global. Uma tarefa completa admite um planejamento hierárquico, onde uma tarefa se decompõe em seqüências de tarefas mais simples que por sua vez podem se decompor em outras subtarefas. Geralmente, tem-se um controle hierárquico, onde o planejamento é subdividido em módulos funcionais, tipicamente dependentes tanto de informações espaciais, quanto de restrições temporais. A figura 5.2 ilustra este modelo, onde se tem quatro níveis hierárquicos no planejador: nível de planejamento global estratégico,

de planejamento intermediário tático, de planejamento local de curto prazo e de controle de atuador. Todos acessam informações do modelo do mundo, cada um referente ao conhecimento de seu interesse. Nota-se que, do nível superior da hierarquia (referente ao planejamento global) ao inferior (referente ao controle dos atuadores), cresce a restrição de tempo na resposta e diminui o espaço físico de interesse [36].

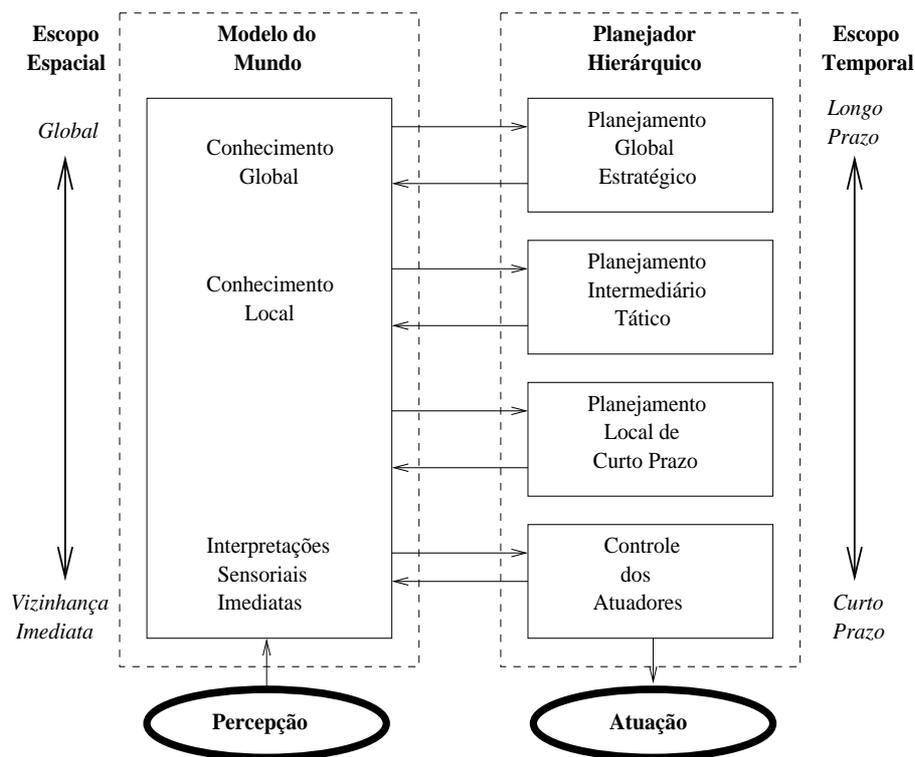


Figura 5.2: Planejamento hierárquico usado em arquiteturas deliberativas

Um último paradigma de raciocínio para robôs móveis são as arquiteturas híbridas que visam englobar as funcionalidades dos dois paradigmas anteriores ao mesmo tempo que busca minimizar os problemas apresentados pelos mesmos. As arquiteturas híbridas incorporam um elemento de planejamento sobre a definição e seleção de comportamentos reativos individuais. Assim, uma arquitetura híbrida corresponde a uma arquitetura reativa controlada por um plano de execução e de seqüenciamento de comportamentos. Através da incorporação da habilidade de raciocínio baseado em modelos internos do mundo (deliberação), estas arquiteturas permitem a *reconfiguração dinâmica* de sistemas de controle reativo [36].

A integração de deliberação e controle reativo é um problema complexo. Contudo, existe

um consenso que nenhuma estratégia é completamente satisfatória isoladamente, e ambas devem ser levadas em consideração para produzir um sistema flexível, robusto e inteligente.

Um robô deve ser capaz de responder rapidamente e de forma eficiente às mudanças dinâmicas e não modeladas que ocorrem no mundo. Se um sistema puramente deliberativo tentar modelar e pré-planejar todas as eventualidades, corre-se o risco de que o processo de planejamento nunca termine. Também não é seguro para o robô fazer suposições grosseiras sobre o mundo, que não reflitam a sua natureza dinâmica, traduzindo-as em um plano inalterável, que guiará todas as suas ações ou decisões futuras. Por outro lado, uma abordagem reativa responde eficientemente a dados sensoriais imediatos, mas é menos eficaz na integração de conhecimentos sobre o mundo.

### 5.2.3 Arquiteturas baseadas em comportamento e arquiteturas baseadas em módulos funcionais

Existe um dualismo interessante entre estas duas abordagens, onde aspectos complementares estão explícitos no projeto do sistema, e aspectos opostos emergem das interações envolvendo as partes da arquitetura. Por exemplo, em arquiteturas baseadas em módulos funcionais respostas comportamentais não estão explícitas na arquitetura, ao invés disso elas emergem do planejador com as dadas metas e o modelo de mundo particular que foi construído a partir dos dados sensoriais. Em sistemas baseados em comportamento dá-se o inverso. Assim respostas comportamentais estão explícitas no projeto e não existe qualquer meta representada explicitamente, ou seja, a meta emerge do funcionamento normal do agente.

Os contrastes entre estes dois estilos de construção de robôs e seus programas podem ser resumidos na tabela 5.1. Cada linha tenta resumir brevemente alguns aspectos do dualismo entre as duas abordagens.

Em arquiteturas baseadas em módulos funcionais nunca existe uma seqüência de ações explicitamente representada *a priori* no programa do robô. Ao invés disso, uma seqüência é produzida como resultado do raciocínio sobre o modelo do mundo e sobre as metas designadas para o sistema. Na abordagem baseada em comportamento existem muitas vezes ações específicas representadas, mas as metas específicas do robô nunca são explicitamente representadas, nem sequer existem quaisquer planos [17].

Aspecto	Baseado em Módulos	Baseado em Comportamento
Explícito	fusão de sensores modelos metas planos	filtros seleção de ação esquemas de ação comportamento
Problema	busca	problema de ligação
Saída	escolhe próxima ação	ações concorrentes
Emergente	resposta comportamental	metas e planos (aparente)

Tabela 5.1: Comparação entre estilos de construção de robôs

Da comparação apresentada acima pode-se retirar algumas definições. É possível dizer que um *comportamento* encapsula as tarefas de percepção, planejamento e execução necessárias para conseguir um aspecto específico do controle do robô. Como características gerais, destacam-se:

- Cada comportamento produz uma ação e é responsável por um aspecto particular do controle do veículo para conseguir alguma tarefa particular.
- Os comportamentos operam de forma assíncrona, e podem trabalhar em paralelo com outros comportamentos.
- Um comportamento baseia-se exclusivamente nos dados sensoriais que são relevantes para suas necessidades particulares de tomada de decisão.

No caso de uma decomposição por módulos funcionais é freqüente que se estruture o robô nos seguintes níveis:

- **Planejador:** Planeja um caminho que cumpra a missão encomendada pelo operador. A descrição do mesmo tem baixa resolução.
- **Navegador:** A partir do caminho gerado pelo planejador e com a informação do ambiente disponível, refina-se o caminho reduzindo os erros.
- **Piloto:** A partir da trajetória gerada pelo navegador e da informação disponível neste nível, geram-se seqüências de comandos de controle.
- **Controle de execução:** Executa os comandos enviados pelo piloto, transformando-os em ações que administram os atuadores.

A decomposição em comportamentos encapsula em um único módulo várias funções. Um comportamento pode caracterizar-se por:

- Receber entradas dos sensores, já processados ou não.
- Gerar saídas para os atuadores
- Resolver uma atividade concreta integrando no dito comportamento todo o necessário para tal fim.
- Resolver uma tarefa completa mesclando vários componentes simples. A mescla dos comportamentos admite muitas variações: ponderar os comportamentos, inibir comportamentos, ativá-los alternativamente, etc.

Um vez imbuido nossos robôs com múltiplos comportamentos capazes de manipularem uma vasta variedade de circunstâncias e alcançar uma certa quantidade de tarefas, estamos voltados para o problema de decidir que comportamento, ou comportamentos, deve estar ativo em um dado momento. Existem dois componentes para esta decisão: como selecionar comportamentos potencialmente corretos, e como resolver conflitos entre comportamentos. Mais especificamente, precisamos considerar os seguintes assuntos [16]:

- **Coerência:** Mesmo que muitos comportamentos possam estar ativos em um dado momento, ou possam ser constantemente ativados ou desativados, o robô deve ainda apresentar coerência com relação as suas ações e metas. Ele não deve intercalar entre comportamentos inconsistentes rapidamente, nem deve permitir que dois comportamentos estejam ativos simultaneamente se estes interferem um no outro de forma que nenhum opere satisfatoriamente.
- **Importância:** Os comportamentos que estão ativos devem ser importantes para a situação em que o robô se encontra. Por exemplo, o robô deve ativar o comportamento que recarrega suas baterias quando estas estiverem com um nível de carga baixo, e não o contrário.
- **Adequação:** Os mecanismo de seleção de comportamento devem garantir que as metas de longo prazo serão encontradas. Por exemplo, um robô limpador de chão deve realizar sua tarefa em circunstâncias normais, além de fazer todas as tarefas relacionadas necessárias para que tal meta seja concluída com sucesso.

Alguns dos mecanismos de seleção de comportamento podem ser encontrado em [16].

### 5.3 Exemplos de arquiteturas para robôs móveis

Nesta seção duas das principais arquiteturas de robôs móveis existentes são apresentadas: a arquitetura de *subsunção* considerada um arquitetura reativa baseada em comportamento; e a arquitetura *AuRA* classificada como uma arquitetura híbrida por conter tantos componentes reativos como deliberativos.

#### 5.3.1 Arquitetura de subsunção

Segundo [13], existem algumas exigências básicas para o sistema de controle de um robô móvel autônomo inteligente:

- Manipular múltiplas metas mesmo que estas sejam, em alguns momentos, conflitantes.
- O robô deverá, invariavelmente, tomar decisões sob condições adversas de seus sensores, uma vez que estes são passíveis de erro. Quanto maior o número de sensores maiores as chances de erro.
- O robô deve ser robusto, ou seja, ele deve continuar sua operação mesmo na presença de algumas falhas.
- O robô deve ser capaz de absorver as novas capacidades a ele incumbidas.

As primeiras técnicas visando um sistema de controle para robôs eram baseadas na decomposição do problema em um série de *módulos funcionais*, onde estes encapsulavam cada uma das tarefas necessárias para que o robô apresentasse o comportamento desejado. A arquitetura de subsunção é o que convencionamos chamar de arquitetura baseada em comportamentos onde a devida coordenação de um conjunto de comportamentos leva a satisfação do conjunto de metas estabelecidas para o robô. Como nos demais problemas na engenharia, a arquitetura de subsunção decompõe o problema em pedaços, resolve os subproblemas e em seguida compõe as soluções. O que difere é a forma com dá-se a decomposição.

A arquitetura de subsunção revela-se uma arquitetura reativa baseada em comportamentos, organizados em *camadas de competência*: através de uma hierarquização de comportamentos aqueles de nível mais alto correspondem a execução dos objetivos das tarefas

especificadas, enquanto as de níveis mais baixo correspondem a tarefas mais básicas, como aquelas que garantem a sobrevivência e integridade do robô. As camadas decidem quando e como agirem, sem o uso de sub-rotinas de outras camadas, no que convencionou-se chamar de *sistemas de atividades*.

Brooks, em [13], cita como exemplos de camadas de competências:

1. Evitar contatos com objetos (tanto objetos móveis como estacionários).
2. Vaguear sem objetivo definido pelo ambiente.
3. “Explorar” o mundo
4. Construir um mapa do ambiente e planejar rotas de um local para outro
5. Notar mudanças no ambiente “estático”.
6. Raciocinar sobre o mundo em termos de objetos identificáveis e executar tarefas relacionadas a tais objetos.
7. Formular e executar planos que envolvem mudanças nos estados do mundo de algum modo desejável.
8. Raciocinar sobre o comportamento dos objetos no mundo e modificar planos de acordo com tais comportamentos.

Novos níveis de competência podem ser adicionados incrementalmente, de forma evolutiva, sendo que novos comportamentos são capazes de influenciar os níveis inferiores, sem entretanto, modificar suas regras, gerando um comportamento inteligente.

Com esta arquitetura surge um novo movimento no estudo da inteligência, baseado em uma abordagem “botton up”, concentrando-se nos sistemas físicos (com inspiração na engenharia e nos sistemas biológicos) situados no mundo, buscando a execução de tarefas de forma autônoma, e adaptando-se ao ambiente ao qual está inserido.

A idéia de estruturação da inteligência baseada na coordenação de comportamentos simples surge do pensamento que afirma que a maioria das ações humanas são realizadas sem que um raciocínio complexo seja feito. “Muitas das coisas feitas em nosso dia não são planejadas, são atividades rotineiras em uma relação dinâmica com o mundo”[15].

Conforme [18], dentre as vantagens da arquitetura de subsunção destacam-se:

- Tem princípios bem definidos para o desenvolvimento de sistemas baseados em comportamento.
- Existem muitas implicações documentadas que obtiveram sucesso.
- Seu enfoque incremental, *bottom-up*, necessita que qualquer solução desenvolvida seja funcional no mundo real e com dados reais; com o mínimo de representação possível.

Além das características intrínsecas do paradigma reativo, como rápida resposta no processamento da informação e a simplicidade dos comportamentos através de um mapeamento estímulo-resposta, a arquitetura de subsunção apresenta um mecanismo de prioridade de comportamentos, permitindo que um comportamento de nível superior suprima aqueles de nível inferior, configurando uma coordenação competitiva de comportamentos com uma hierarquia pré-definida. Este mecanismo pode assumir duas formas [36]:

- **Supressão:** Neste caso, a saída produzida pelo comportamento de nível mais alto substitui aquela produzida pelo comportamento de nível mais baixo. Este último permanece ativo, mas sua ação não produz nenhum efeito, por ter sido suprimida por aquela correspondente ao comportamento prioritário.
- **Inibição:** No mecanismo de inibição, o comportamento em nível mais baixo é desativado por aquele de nível mais alto. Nesse caso, não ocorre uma substituição da ação de nível mais baixo, mas uma inibição do comportamento em si.

Cada comportamento tem uma tarefa associada a si, as de nível mais baixos sendo mais básicas, e as de níveis superiores são capazes de suprimir as de níveis inferiores. Vários subobjetivos podem coexistir e sua realização depende do nível de prioridade ao qual seu comportamento apresenta. Cada comportamento extrai da percepção do robô as informações relevantes a realização de sua tarefa, e da coordenação entre os comportamentos emerge a funcionalidade exigida pelo ambiente.

Este método em níveis, ou camadas, de competência permite naturalmente atender os requerimentos exigidos de um sistema de controle para robôs móveis apresentados no início desta seção. Isto pode ser alcançado das seguintes formas:

- Camadas individuais podem trabalhar em suas respectivas metas concorrentemente. O mecanismo de supressão media as ações que são tomadas. A vantagem aqui é que não necessário uma tomada de decisão *a priori* em busca da meta perseguida.

- De certa forma podemos ignorar o problema da fusão de sensores usando uma arquitetura de subsunção. Nem todos os sensores necessitam ser consultados em uma representação, somente aqueles cujo processamento da informação identifica-se como extremamente confiável podem ser eleitos para participarem da representação.
- A existência de múltiplos sensores e o gerenciamento inteligente destes dão robutez à arquitetura. Entretanto, existe ainda outra fonte de robustez em uma arquitetura de subsunção: níveis inferiores bem debugados continuam a funcionar mesmo quando níveis superiores são somados.
- Uma maneira óbvia de dispor de aditividade é construir cada nova camada em seu próprio processador.

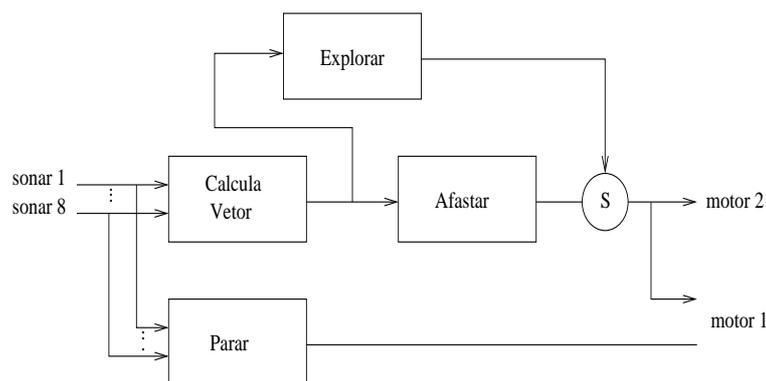


Figura 5.3: Um exemplo simples da arquitetura de subsunção

A figura 5.3 ilustra um exemplo simples de implementação da arquitetura de subsunção para um robô móvel que se move em um ambiente, desviando-se de obstáculos quando necessário, mas sem perder a direção do seu movimento [36]. O robô tem como sensores um conjunto de oito sonares distribuídos uniformemente em sua periferia, capazes de estimar a distância de obstáculos nas direções correspondentes. Um módulo de processamento da informação sensorial, *Calcula Vetor*, interpreta as leituras dos sensores como vetores e os soma, produzindo um vetor resultante que indica a direção e a magnitude relativas à posição e distância de um obstáculo imaginário, que resume estas leituras. O movimento do robô é comandado por dois motores independentes: um motor 1 para produzir velocidades iguais nas rodas, e um outro motor 2 para produzir um giro do eixo das rodas. Na camada mais baixa estão dois comportamentos básicos: *Parar*, que ao detectar, pela leitura dos sonares, a

proximidade de algum obstáculo (leitura de algum sonar maior que um valor limiar), força a parada do robô enviando um comando apropriado para o motor responsável pelo controle de velocidade das rodas, e *Afastar*, que produz (i) um giro do eixo das rodas do robô até que este fique com sua frente na direção oposta à do vetor (via comando enviado para o motor 2) e (ii) um posterior avanço proporcional à magnitude do vetor (via comando enviado para o motor 1). Numa camada mais alta, um comportamento *Explorar* combina um vetor de magnitude fixa e direção aleatória, produzindo a intervalos regulares, a um vetor de direção oposta ao daquele produzido pelo módulo *Calcula Vetor*. Desta forma, obtém-se uma nova direção para o movimento do robô, correspondente a um desvio de obstáculos mais suave, que leva em consideração uma direção preferencial de movimento (aquela estabelecida pelo comportamento *Explorar*). Este comportamento suprime (ou seja, substitui) a saída produzida pelo comportamento *Afastar*, que no entanto continua ativo: caso o comportamento *Explorar* seja desativado (possivelmente por algum outro comportamento em nível mais alto), o robô ainda terá capacidade de desviar de obstáculos satisfatoriamente.

### 5.3.2 Arquitetura AuRA

A arquitetura AuRA (*Autonomous Robot Architecture*), é um sistema híbrido surgido durante os anos 80. Esta arquitetura é dita híbrida devido a presença de dois componentes distintos: um *deliberativo* (ou também designado *planejador hierárquico*), baseado em técnicas de inteligência artificial tradicional; e um controlador reativo, baseado em uma teoria de esquemas [3]. Assim, um planejador configura o sistema reativo antes da execução, e o reconfigura para o caso de eventuais falhas. Na figura 5.4 pode ser visualizado os vários módulos que compõem a arquitetura.

É possível verificar na figura 5.4 a presença dos componentes deliberativo e reativo: um sistema hierárquico consistindo de um *planejador de missão*, *raciocinador espacial*, e um *seqüenciador de planos*, acoplados a um sistema reativo, o *controlador de esquemas*. Este módulos podem ser descritos da seguinte forma:

- **Planejador de Missão:** módulo que interage com o usuário e onde metas de alto nível e restrições são definidas a partir de uma descrição.
- **Raciocinador Espacial:** a partir de um cartógrafo, constrói um caminho, ou seja, uma seqüência de passos com o intuito de cumprir uma missão.

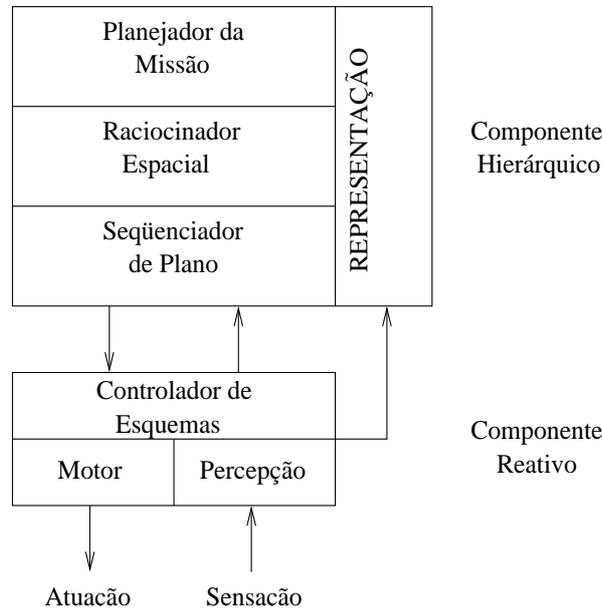


Figura 5.4: Arquitetura híbrida AuRA

- Seqüenciador de Planos: cada passo definido no componente anterior é traduzido em um comportamento reativo.
- Controlador de Esquemas: monitora e constrói os comportamentos a partir de bibliotecas de esquemas sensoriais.

Uma vez iniciada a execução reativa, o componente deliberativo não é ativo pelo menos até que se detecte uma falha em tal execução. Uma falha típica é detectada pela falta de progresso, evidenciando uma velocidade zero ou sinal gerador de *timeout*. Neste momento o planejador hierárquico é reinvocado um estágio por vez, de baixo para cima na hierarquia, até o problema ser resolvido. Primeiro, o Seqüenciador de Planos tenta refazer a rota do robô baseado nas informações que foram obtidas durante a navegação e armazenadas na memória de curto prazo. Se por algum motivo isto mostra-se insatisfatório - por exemplo, a rota está completamente bloqueada dentro deste contexto local - o Raciocinador Espacial é novamente invocado. Ele tenta gerar uma nova rota global que sobrepõe a região inteira afetada. Se ainda assim houver falha, o Planejador de Missão é reinvocado, informando o operador da dificuldade e perguntando pela reformulação ou abandono da missão.

A arquitetura AuRA é altamente modular. Os componentes da arquitetura podem ser substituídos por outros de forma direta. Outra característica é a flexibilidade que ele fornece

por intermédio da introdução de métodos adaptativos e de aprendizado. Não obstante, a AuRA cobre uma vasta quantidade de problemas, incluindo domínios como:

- Ambientes de manufatura
- Navegação tridimensional como as encontradas em domínios aéreos e aquáticos
- Navegação em ambientes abertos e fechados
- Competições de robôs
- Cenários militares
- Manipulação móvel
- Equipes de robôs

Finalmente, uma das maiores características do AuRA resulta do casamento entre dois paradigmas distintos da IA: a deliberação e a reatividade.

## 5.4 Exercícios

1. Descreva duas situações onde possam ser utilizadas uma arquitetura centralizada e uma arquitetura distribuída, aponte os motivos para tal escolha.
2. Descreva exemplos em que uma arquitetura reativa não seria suficiente para alcançar os objetivos do robô, necessitando de algum nível de deliberação.
3. Cite um possível exemplo de aplicação da Arquitetura de Subsunção e porque esta seria a melhor alternativa.
4. Dentro da arquitetura AuRA aponte os módulos reativos e deliberativos.

## Capítulo 6

# Modelagem do Ambiente

Para que o robô possa interagir com o ambiente de forma mais eficaz do que simplesmente a navegação aleatória, faz-se necessário criar alguma forma de representação deste ambiente, que possibilite ao robô coordenar suas ações no sentido de desempenhar sua tarefa [33].

Esta representação do ambiente é construída a partir das leituras obtidas pelos sensores e é utilizada para o planejamento de trajetória.

O problema da aquisição de modelos é bastante complexo e está longe de ser resolvido, pois há uma série de fatores práticos que limitam a habilidade do robô em “aprender” e “usar” modelos precisos. Alguns destes fatores são [39]:

1. Sensores - as condições de operação dos sensores podem resultar em perda, falha ou informações irrelevantes, não permitindo obter a exata localização dos obstáculos.
2. Limitação da percepção - a faixa de percepção de muitos sensores é limitada a uma zona próxima ao robô. Para adquirir informações globais, o robô precisa explorar o ambiente.
3. Ruído na leitura dos sensores - as medidas dos sensores estão corrompidas por ruídos. Geralmente, a distribuição deste ruído não é conhecida.
4. Imprecisão dos movimentos - os movimentos do robô são imprecisos, ocasionando erros de odometria freqüentes.
5. Complexidade e dinâmica - os ambientes são, em geral, complexos e dinâmicos, o que torna impossível manter modelos exatos dos mesmos.

6. Requisitos de tempo-real - os requisitos de tempo real impõem que o modelo do ambiente seja simples, acessível e fácil de manipular, entretanto, construir modelos simples implica maior grau de imprecisão. É preciso balancear precisão e simplicidade.

Apesar destas dificuldades, dois paradigmas têm se estabelecido como sendo os principais tipos de modelos utilizados em robótica móvel. São eles o modelo baseado em células e o modelo topológico. Nos modelos baseados em células, o ambiente é representado por um conjunto de células onde cada célula corresponde a uma região do ambiente. Nos modelos topológicos o ambiente é representado por meio de um grafo, onde os nós correspondem a situações distintas, lugares ou ainda a marcas (*landmarks*).

## 6.1 Mapas baseados em células

Nesta abordagem o ambiente bidimensional ou tridimensional é representado por meio de um conjunto de células. Os modelos de decomposição em células são métodos aproximados, uma vez que não representam completamente o espaço livre.

Há dois esquemas básicos e principais de decomposição do ambiente em células: enumeração da ocupação espacial e *quadtrees* ou *octrees*[37].

### 6.1.1 Modelagem por enumeração

A modelagem por enumeração consiste na criação de uma matriz de células de tamanho fixo. Este é um método aproximado, onde a resolução está diretamente relacionada com o tamanho da célula. Células grandes reduzem a complexidade, mas promovem a perda de espaço livre, enquanto que células pequenas permitem uma modelagem mais acurada e, conseqüentemente, o aumento da complexidade.

O método é bastante simples, principalmente para acessar um determinado ponto no ambiente. A cada célula é atribuída uma probabilidade de ocupação, ou ainda pode-se considerar a ocupação binária (1 para células com obstáculos e 0 para células vazias).

Os diferentes locais do ambiente são distingüidos baseando-se na posição geométrica do robô, com relação ao sistema de coordenadas global. A posição do robô é estimada incrementalmente, baseando-se na informação do sensor de odometria. A figura 6.1 apresenta um exemplo de decomposição do ambiente em células utilizando a modelagem por enumeração.

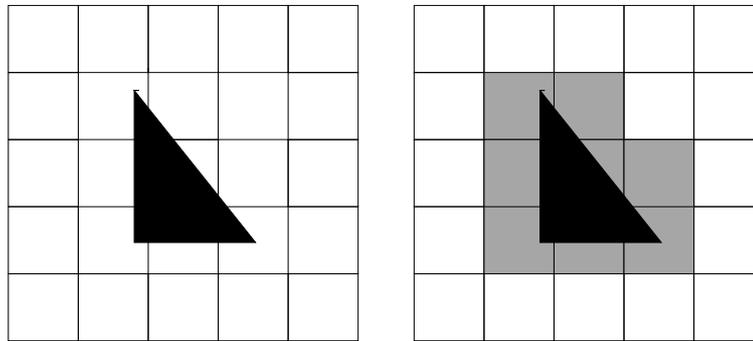
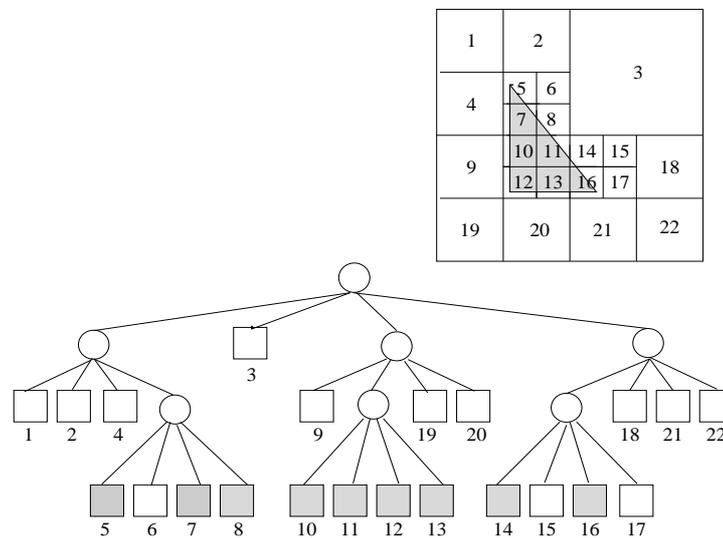


Figura 6.1: Modelagem por enumeração

### 6.1.2 Modelagem por *Quadtrees* e *Octrees*

Estes métodos são derivados do anterior, porém são mais eficientes, pois a resolução é facilmente ajustável e em uma mesma representação podem coexistir diferentes graus de resolução. As *quadtrees* e *octrees* são estruturas de dados em forma de árvore que modelam objetos em duas ou três dimensões respectivamente. A figura 6.2 apresenta a modelagem de um ambiente utilizando uma *quadtree*.

Figura 6.2: Decomposição do ambiente utilizando *quadtrees*

Neste modelo, as células recebem a atribuição de cheias, vazias ou mistas. Células cheias são aquelas totalmente cobertas por obstáculos, células vazias são aquelas cuja intersecção com a região de obstáculos é nula e mistas, quando parte da célula é ocupada pelo obstáculo.

## 6.2 Modelo Topológico

A idéia deste modelo é representar relações entre entidades. Essa representação pode ser feita por meio de grafos onde os vértices representam as entidades e as arestas, as relações. As entidades podem ser de dois tipos: lugares ou marcas. Por utilizar grafos, a representação torna-se compacta, facilitando o processamento e busca por caminhos.

Neste modelo, a localização do robô é relativa e freqüentemente é difícil determinar se dois lugares parecidos são na verdade o mesmo ou não, principalmente quando o robô chega a eles por caminhos distintos.

### 6.2.1 Modelo Topológico de Lugares

A modelagem de lugares é feita de forma hierárquica, utilizando relações de adjacência. Este tipo de modelagem é simples e facilita a manipulação do modelo. Na figura 6.3 tem-se um exemplo de ambiente e o seu modelo topológico utilizando lugares como entidades.

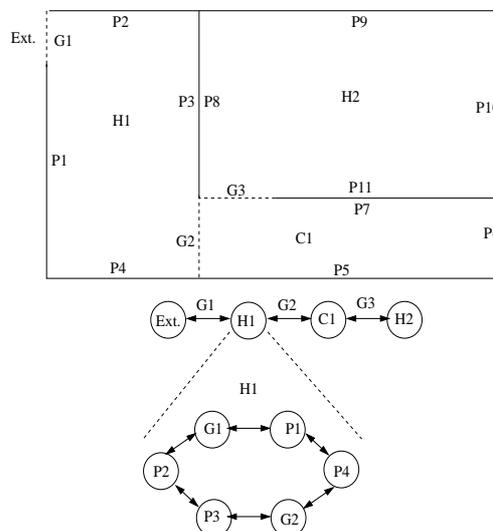


Figura 6.3: Modelo topológico

O ambiente é formado pelo meio externo (Ext.), por duas salas (H1 e H2) e um corredor (C1). A transição do meio externo (Ext.) para a sala H1 é feita através da porta G1. Estando em H1, é possível chegar ao corredor C1 utilizando a porta G2. De forma semelhante, utilizando G3 é possível chegar à sala H2.

Ao refinar o modelo, as entidades são expandidas, gerando novos grafos. Assim, pode-se representar, por exemplo, a sala H1 como sendo o conjunto formado pelas paredes P1,

P2, P3 e P4 e pelas portas G1 e G2. As arestas do grafo representam a relação de adjacência entre estas entidades.

### 6.2.2 Modelo Topológico de Marcas

Uma marca (*landmark*) pode ser qualquer objeto ou conjunto de objetos que integre o ambiente. As marcas são classificadas em dois grupos: naturais e artificiais. As marcas naturais não podem ser modificadas ou movidas (uma parede, uma porta, um edifício, uma montanha, etc.). Já as marcas artificiais podem ser movidas ou modificadas.

São 3 os tipos de objetos que podem compor uma marca:

- Objetos geométricos - estas marcas são formadas por objetos geométricos simples. Ex.: polígono, prismas, etc.
- Objetos estruturados - são marcas formadas por objetos dotados de uma estrutura bem definida e mais complexa. Ex.: sinais de tráfego, marcas específicas para localização, portas, um cruzamento, um edifício de forma e cor determinados.
- Objetos não-estruturados - estas marcas são formadas por objetos sem estrutura bem definida. Ex.: uma árvore, uma rocha, etc.

Os modelos de marcas são utilizados para localização do robô, planejamento de ações (um evento sensorial ativa um determinado comportamento) e controle do robô (marcas que afetam a velocidade do robô, como por exemplo, um sinal de trânsito). Para o reconhecimento de uma marca no ambiente são necessários os seguintes passos: detecção, identificação e extração da informação.

## 6.3 Exercícios

1. Os dois métodos apresentados aqui são métodos aproximados de modelagem do ambiente. Pesquise e descreva brevemente um método de representação que seja exato. Por que este tipo de modelo não é muito utilizado?
2. Monte uma pequena tabela listando as vantagens e desvantagens de métodos de modelagem exatos e aproximados.
3. A criação de modelos topológicos apresenta-se como uma forma simples de modelagem. Quais tipos de sensores são utilizados para detectar marcas?

## Capítulo 7

# Planejamento de Trajetória

### 7.1 Introdução

O objeto de estudo do planejamento de trajetória em robótica móvel é dar aos robôs a capacidade de planejar seus próprios movimentos, sem a necessidade de interferência direta de seres humanos. Esta é uma das áreas mais importantes da robótica móvel atualmente, pois a elaboração de um plano de movimentação é uma tarefa extremamente complexa, salvo quando o ambiente de atuação do robô é limitado e cuidadosamente controlado, situação ideal que geralmente não acontece na maioria das aplicações realísticas.

Neste capítulo abordaremos as técnicas básicas de planejamento de trajetória utilizadas atualmente em robótica móvel.

### 7.2 O Problema Básico

O problema básico do planejamento de trajetória pode ser encarado como uma simplificação na qual as questões principais são isoladas e estudadas em profundidade antes que considerarmos dificuldades adicionais. No problema básico assume-se que o robô é o único objeto móvel no *espaço de trabalho*<sup>1</sup>. Também são ignoradas as propriedades dinâmicas do robô, evitando assim as questões temporais. A movimentação também é reduzida para uma movimentação sem contato, ignorando-se as interações que podem ocorrer entre o robô e os objetos físicos no ambiente. O objetivo de todas estas simplificações é transformar o problema do planejamento de um problema “físico” para um problema “puramente

---

<sup>1</sup>do inglês

geométrico”. Também assumimos que o robô é um único objeto rígido, isto é, um objeto cujos pontos são fixos com respeito a cada um dos outros (ou seja, o robô não apresenta partes móveis, como braços, pernas, ou outros apêndices móveis). Assumimos também que os movimentos deste objeto são restringidos unicamente pelos obstáculos dispostos no espaço de trabalho.

O problema básico pode então ser resumido como o problema de, a partir de uma posição inicial e uma orientação inicial, gerar um caminho composto por uma seqüência contínua de posições e orientações do robô que evitam o contato com os outros objetos no espaço de trabalho e que termina numa posição meta pré-estabelecida.

### 7.3 O Espaço de Configuração

Uma configuração de um objeto arbitrário é uma especificação da posição de todos os pontos neste objeto, relativos à um **frame** de referência fixo. Um espaço de configuração  $C$  de um objeto  $A$  é então o conjunto de todas as configurações do objeto  $A$ .

### 7.4 Abordagens de Planejamento

Trataremos agora as principais abordagens para o planejamento de trajetória mais comumente utilizadas em robótica móvel.

#### 7.4.1 Abordagem Roadmap

Este método consiste em capturar a conectividade do espaço livre do robô em uma rede de curvas chamada **Roadmap**, no espaço de configuração  $C_{livre}$  ou em seu fechamento<sup>2</sup>  $cl(C_{livre})$ . Uma vez que o **roadmap** é construído, ele é utilizado como um conjunto de caminhos padronizados. O planejamento de trajetória resume-se então em encontrar um caminho que faça a interconexão dos pontos definidos pela configuração inicial pela configuração final desejada. Vários métodos baseados nesta abordagem foram propostos, entre eles:

- Grafos de visibilidade

---

<sup>2</sup>Do inglês *closure*

- Diagramas de Voronoi
- Redes **Freeway**
- Silhueta

Outro método *roadmap* chamado de **retração** consiste em definir uma função contínua de  $C_{livre}$  sobre um subconjunto de si mesmo (o **roadmap**) tais que as restrições desta função para este subconjunto é o mapa identidade (em topologia tal função é chamada de “**retração**”). Em um espaço de configuração bidimensional o espaço de configuração  $C_{livre}$  é tipicamente retraído em seu diagrama de Voronoi. Este diagrama é o conjunto de todas as configurações livres cuja distância mínima para a região  $C - obstaculo CB$  é alcançada com pelo menos dois pontos nos limites de  $CB$ .

A vantagem deste diagrama é que ele produz caminhos livres que tendem à maximizar a separação entre o Robô e os obstáculos. Quando os  $C - obstáculos$  são polígonos, o diagrama de Voronoi consiste de segmentos de retas e parabólicas.

#### 7.4.2 Decomposição em Células

São os métodos mais extensamente estudados. Consistem na decomposição do espaço livre do robô em regiões simples chamadas células, de tal forma que um caminho entre duas configurações em células diferentes pode ser facilmente gerado. Um grafo não-direcionado representando a relação de adjacência entre as células é então construído, e sobre este grafo a busca do caminho é realizada. Este grafo é chamado de grafo é chamado de *grafo de conectividade*. Os nós deste grafo são as células extraídas do espaço livre, e dois nós estão conectados por uma aresta se e somente se duas células correspondentes são adjacentes. O resultado da busca é uma seqüência de células chamada de *canal*. Um caminho livre pode ser então computado a partir desta seqüência.

Os métodos de decomposição em Células podem ser divididos em duas categorias:

1. Métodos exatos
2. Métodos aproximados

##### **Métodos Exatos:**

Estes métodos decompõem o espaço em células cuja união é exatamente o espaço livre.

**Métodos Aproximados:**

Os métodos aproximados produzem células de formas pré-definidas (por exemplo, forma retangular) cuja união está estritamente inclusa no espaço livre.

**7.4.3 Campo Potencial**

É uma abordagem mais direta para o planejamento de trajetória. Consiste em discretizar o espaço de configuração em um fino grid regular de configurações e realizar a busca por um caminho livre neste grid. Esta abordagem necessita de heurísticas adequadas para direcionar a busca, pois o grid, em geral, tem um tamanho consideravelmente grande. Vários tipos de heurísticas têm sido propostas, entre elas as de maior sucesso são aquelas que se tornaram conhecidas como **Campos Potenciais**.

A metáfora sugerida por esta terminologia é que o robô, representado como um ponto no espaço de configuração, é uma partícula que se move sob a influência de um potencial artificial produzido pela configuração-meta e os espaços de configuração dos objetos no espaço de trabalho. A configuração-meta gera um **potencial atrativo** que puxa o robô em sua direção, enquanto que os obstáculos geram um **potencial repulsivo**, que empurra o robô para longe deles. O gradiente negado do potencial total é tratado como uma força artificial aplicada ao robô.

Este método pode ser bastante eficiente se comparado com os métodos anteriores, porém como ele é, essencialmente, um método de otimização descendente rápido, tem a grande desvantagem de poder ficar preso em máximos locais. Uma solução para este problema é projetar as funções potenciais para que não possuam máximos locais além da configuração-meta. Outra solução seria complementar a abordagem de campo potencial básica com mecanismos de escape de máximos locais.

**7.5 Extensões ao Problema Básico**

A noção básica do problema de planejamento de trajetória faz suposições que limitam significativamente a viabilidade prática das suas soluções. Na prática é bastante difícil reduzir um problema de robótica real à uma instância do problema básico, resolver esta instância e depois adaptá-la para produzir soluções adequadas às condições do problema original. Para tentar resolver esta situação foram criadas uma série de extensões ao modelo básico.

As principais extensões estão descritas a seguir:

### 7.5.1 Múltiplos Objetos Móveis

No problema básico é suposto que os obstáculos são fixos, que o robô é a única entidade móvel no espaço de configuração e que o mesmo é constituído de um único objeto rígido. Consideraremos agora uma série de extensões que retiram estas suposições. Uma extensão consiste na inclusão de obstáculos móveis no espaço de trabalho, outra permite que vários robôs operem ao mesmo tempo no espaço de trabalho, sendo que cada um atua como um obstáculo móvel para cada um dos outros e uma terceira extensão considera robôs articulados, ou seja, robôs que são compostos por vários objetos rígidos conectados por juntas.

#### Obstáculos Móveis:

Na presença de obstáculos móveis o problema de planejamento de trajetória não pode mais ser resolvido meramente pela construção de um caminho geométrico. Uma função contínua de tempo especificando as configurações do robô a cada instante do tempo deve ser gerada em vez disso. Isto pode ser feito pela adição de uma dimensão representando o tempo ao espaço de configuração do robô. O novo espaço, denotado por  $CT$ , é chamado de espaço-tempo de configuração. Este novo espaço de configuração mapeia a movimentação dos obstáculos em regiões estáticas em um determinado instante de tempo  $t$ , de tal forma que toda seção reta através de  $CT$  no instante  $t$  é o espaço de configuração do robô no instante  $t$ . Ela corta os  $CT$  – obstáculos de acordo com os  $C$  – obstáculos correspondentes no espaço de trabalho dos obstáculos em suas localizações no instante  $t$ . O planejamento de trajetória consiste então em achar um caminho entre os  $CT$  – obstáculos em  $CT$ . Já que o tempo é irreversível, este caminho deve ter a propriedade de nunca retroceder ao longo do eixo do tempo. Para isso, os métodos de planejamento devem ser modificados para levar em conta esta especificidade relativa ao tempo.

Se não há restrições sobre a velocidade e a aceleração do robô, e se o movimento de todos os obstáculos é conhecido de antemão, é bastante simples estender alguns dos métodos básicos de planejamento para manipular este novo problema. Se tais restrições existem, elas se traduzem em restrições geométricas sobre a inclinação e a curvatura do caminho ao longo da dimensão do tempo, o que torna o planejamento bem mais difícil.

**Múltiplos Robôs:**

O planejamento de trajetória com múltiplos robôs difere do planejamento com obstáculos móveis porque o movimento de todos os robôs no ambiente deve ser planejado, enquanto que o movimento dos obstáculos móveis não está sob o controle do planejador.

Uma forma de se trabalhar com múltiplos robôs operando no mesmo espaço de trabalho é tratá-los como um único robô com múltiplos corpos. O espaço de configuração composto do robô  $A = A_1, \dots, A_p$  é  $C = C_1 \times \dots \times C_p$ , onde:

$A$  representa o robô multi-corpos;

$A_1, \dots, A_p$  representam cada uma das partes (ou corpos) de que é constituído o robô multi-corpos;

$C_1, \dots, C_p$  representam cada um dos espaços de configuração relativos a cada um dos corpos do robô;

$C$  é o produto cartesiano dos espaços de configuração  $C_i$  dos robôs individuais  $A_1, \dots, A_p$ .

Todas as configurações em  $C$  determinam uma única posição e orientação para cada robô  $A_i$  e um obstáculo  $B_j$ , ou dois robôs  $A_i$  e  $A_j$ . Os métodos básicos para planejamento de trajetória podem então ser utilizados para planejar um caminho de  $A$  em  $C$ . Esta abordagem para planejamento de trajetória multi-robô é chamada de **planejamento centralizado**. Uma dificuldade com este método é que ele pode produzir espaços de configuração de altas dimensões, pois a dimensão do espaço de configuração composto  $C$  é igual a soma das dimensões de todos os espaços de configuração individuais  $C_1$  a  $C_p$ .

Outra abordagem para o planejamento de trajetória com múltiplos robôs, chamada de **Planejamento Desacoplado**, consiste em planejar o movimento de cada robô de maneira mais ou menos independente dos outros robôs, e considerar as interações entre os caminhos em uma segunda fase do planejamento. Utilizando esta abordagem pode-se reduzir significativamente a complexidade computacional, porém compromete-se a completude do plano gerado. A figura 9 mostra um problema de planejamento onde a abordagem desacoplada falha. O problema consiste em trocar a posição de dois robôs (representados pelos discos) em um corredor estreito onde um robô não pode ultrapassar o outro, mas que, ao fim do corredor existe espaço suficiente para realizar a permutação dos dois robôs. Como o plane-

jamento desacoplado considera os robôs separadamente, este método não tem mecanismos para inferir que ambos os robôs devem primeiro mover-se para o fim do corredor.

### **Robôs Articulados:**

Um robô articulado  $A$  é constituído de vários objetos rígidos móveis  $A_1, \dots, A_p$ , chamados de *elos*, conectados por juntas (juntas de revolução ou juntas prismáticas). Cada junta restringe o movimento relativo dos dois objetos os quais ela conecta. Um exemplo típico de um robô articulado é um braço robótico.

O robô  $A$  pode ser visto como um conjunto de  $p$  objetos rígidos móveis. As restrições impostas pelas juntas sobre o movimento relativo dos  $A_i$ 's determina um subconjunto do espaço de configuração composto desses objetos, que é o espaço de configuração real de  $A$ . Toda configuração neste subconjunto determina uma posição e orientação únicas para cada uma das partes do robô. Este subconjunto é, em geral, fácil de parametrizar, por exemplo, pela associação de um ângulo ou de uma distância com cada junta. Vários métodos básicos de planejamento de trajetória podem ser estendidos para contemplar esta variação de forma bastante direta, pelo menos teoricamente.

### **7.5.2 Restrições Cinemáticas**

No problema básico supomos que o robô era um objeto de “vôo livre”<sup>3</sup>, isto é, as únicas restrições ao seu movimento eram devidas aos obstáculos. Em alguns problemas, no entanto, é necessário impor restrições cinemáticas adicionais à movimentação dos robôs. Consideramos aqui dois tipos de restrições:

1. restrições holonômicas e
2. restrições não-holonômicas.

### **Restrições Holonômicas:**

Vamos assumir que a configuração é representada por uma lista de parâmetros de cardinalidade mínima. Uma restrição de igualdade holonômica é uma relação de igualdade entre esses parâmetros, que pode ser resolvida por apenas um dos parâmetros. Tal relação reduz

---

<sup>3</sup>do inglês *free flying object*

a dimensão do espaço de configurações real do robô de uma dimensão. Um conjunto de  $k$  restrições holonômicas independentes reduz este espaço de  $k$  dimensões.

Considere, por exemplo, o caso de um objeto tridimensional  $A$  habilitado a voar livremente, mas restrito a girar ao redor de um eixo fixo relativo a  $A$ . Representamos a orientação de  $A$  por três ângulos. a restrição imposta restringe as possíveis orientações de  $A$ , e pode ser expressa como duas equações independentes entre esses ângulos. Enquanto a dimensão do espaço de configurações de um objeto “livre” é 6, o espaço de configurações de  $A$  tem dimensão 4. O caso particular onde  $A$  pode transitar livremente no eixo de orientação fixo pode também ser considerado um problema com restrições holonômicas. contudo, já que este problema é equivalente ao problema de planejar o movimento de um robô puntiforme no  $R^N$ , podemos encará-lo como um caso particular do problema básico.

Os robôs articulados nos dão outro exemplo de restrições holonômicas. Como exemplo, considere uma junta de revolução que determina duas restrições holonômicas. Assim sendo, enquanto seis parâmetros são necessários para definir a configuração de dois objetos planares de “vôo livre”, quatro parâmetros bastam para determinar a posição e a orientação de dois objetos planares conectados por uma junta de revolução.

Restrições holonômicas certamente afetam a definição do espaço de configuração dos robôs, e podem mesmo modificar a sua conectividade global. Apesar disso, as restrições holonômicas não fazem surgir nenhuma questão fundamental. A maioria dos métodos básicos de planejamento continuam aplicáveis.

### **Restrições Não-Holonômicas:**

Uma restrição de igualdade não-holonômica é uma equação não integrável envolvendo os parâmetros de configuração e suas derivadas (parâmetros de velocidade). Tais restrições não reduzem a dimensão do espaço de configurações do robô, mas reduzem a dimensão do espaço de possíveis movimentos diferenciais (isto é, o espaço das direções de velocidade) em uma dada configuração. Como exemplo, considere um carro-robô móvel andando em uma superfície plana. Este robô pode ser modelado como um objeto retangular movendo-se em  $\mathcal{W} = \mathcal{R}^e$  (ver figura 12. Sabemos, por experiência, que em um espaço vazio podemos dirigir o robô para qualquer direção, com qualquer orientação. Assim, o espaço de configuração do robô tem 3 dimensões, duas de translação e uma de rotação. Vamos representar uma configuração do robô  $A$  por  $(x, y, \theta)$ , onde  $x$  e  $y$  são as coordenadas no frame  $\mathcal{F}_{\mathcal{W}}$ , do ponto

médio  $r$  entre as duas rodas trazeiras e  $\theta \in [0, 2\pi)$  é o ângulo entre o eixo  $X$  de  $\mathcal{F}_W$  e os eixos principais de  $A$ . A qualquer instante durante um movimento (assumindo que não há deslizamento) a velocidade de  $r$  tem que apontar ao longo do eixo principal de  $A$ . Com isto, o seu movimento é restringido pela relação

$$-\sin\theta dx + \cos\theta dy = 0$$

e pode ser demonstrado que esta equação é não integrável, e assim sendo, é uma restrição de igualdade não-holonômica. Devido à esta restrição o espaço de movimentos diferenciais  $(dx, dy, d\theta)$  do robô em qualquer configuração  $(x, y, \theta)$  é um espaço bidimensional. Se o robô é um objeto “livre”, este espaço poderia ser tridimensional. O movimento instantâneo do carro robô é determinado por dois parâmetros: a velocidade linear ao longo do seu eixo principal e o ângulo de esterçamento. Contudo, quando o ângulo de esterçamento é diferente de zero, o robô modifica a sua orientação, e com ela a sua velocidade linear, permitindo que as configurações do robô se espalhem em um espaço tridimensional.

Além disso, o ângulo de esterçamento (denotado por  $\phi$ ) é, em geral, restrito a tomar valores em um intervalo  $[-\phi_{max}, +\phi_{max}]$ , com  $\phi_{max} = \frac{\pi}{2}$ . Esta restrição pode ser escrita como uma desigualdade envolvendo  $dx, dy$  e  $d\theta$ . Isto restringe o conjunto de possíveis movimentos diferenciais sem mudar sua dimensão. Chamamos também a isto de restrição de desigualdade não-holonômica.

Restrições não holonômicas restringem a geometria dos caminhos livres possíveis (ou exeqüíveis) entre duas configurações. Este tipo de restrição é bem mais difícil de se lidar do que as restrições holonômicas.

### 7.5.3 Incerteza

No problema básico é feita a suposição de que o robô pode seguir exatamente os caminhos gerados pelo planejador. Também supõe-se que a geometria dos obstáculos e a sua localização são exatamente conhecidas. Nenhum ajuste de um robô real satisfaz estas suposições, e tanto o controlador do robô quanto a geometria são imperfeitos. Felizmente, na maioria das vezes não temos que nos preocupar com estas imperfeições, porque elas são muito pequenas e tornam-se irrelevantes com relação à tolerância admitida com relação à tarefa que está sendo executada pelo robô. Porém, nem sempre este é o caso. Em alguns casos o robô pode ter pouco ou nenhum conhecimento sobre o seu próprio espaço de trabalho. Nestes casos recai sobre os sensores a tarefa de obter a informação necessária para que

o robô realize a sua tarefa. É necessário que o robô explore o seu espaço de trabalho.

Uma situação intermediária entre os dois extremos acima ocorre quando há erros no controle do robô e nos modelos geométricos iniciais, mas estes erros estão restritos à regiões limitadas. Por exemplo, as localizações reais dos obstáculos são ligeiramente diferentes daquelas no modelo do robô, mas os erros na definição dos parâmetros dessas configurações são limitados. Similarmente, o robô pode mover-se ao longo de uma direção diferente da que lhe foi ordenada, mas a direção real do movimento está contida em um cone estreito centrado na direção do movimento ordenado ao robô. Para poder tratar com estes erros de limites, supomos que o robô é equipado com sensores que podem ser usados em tempo de execução para adquirir informações adicionais. Um porém é que os sensores não são perfeitos. Um exemplo disto é que um sensor de posição não retorna a posição exata do robô. Desta forma é bastante realista assumir que os erros sensoriais estão contidos em uma região de incerteza limitada. Quando os erros no controle, sensoreamento e modelo são razoavelmente pequenos, é interessante que os planos gerados sejam tolerantes à estes erros, ou seja, que as metas sejam alcançadas confiavelmente pela antecipação de várias possíveis contingências.

O problema do planejamento de trajetória com incertezas limitadas pode ser então enunciado como a seguir: “Dada uma região  $I$  inicial e uma região meta  $G$  no espaço de configuração do robô, gerar um plano de trajetória cuja execução garanta que o robô alcance a configuração em  $G$  se ele começar em qualquer configuração (desconhecida) em  $I$ , a despeito da incerteza limitada no controle, sensoreamento ou modelo”.

Uma solução para este problema é um plano que combina comandos de movimentos e leituras de sensores que interajam em tempo de execução, visando diminuir a incerteza e guiar o robô em direção à meta.

A inclusão da incerteza limitada faz surgir novas questões importantes que não são contempladas nem pelo problema básico e nem pelas extensões anteriores. Devido à incerteza no controle, um movimento de comando pode produzir qualquer caminho dentre uma infinidade de caminhos consistentes tanto com o comando quanto com a incerteza. Para que o planejamento garanta o sucesso da execução, todos estes caminhos devem alcançar a meta. Esta, porém, não é uma condição suficiente. O plano deve também terminar *na* meta. Devido à incerteza no sensoreamento, pode não ser uma questão trivial reconhecer a confiabilidade do alcance da meta. O planejador deve assegurar que a informação suficiente

estará disponível para o controlador durante a execução (reconhecibilidade da meta).

A incerteza muitas vezes leva ao uso de comandos de movimentos baseados em sensores, cujo comportamento é menos sensível à erros do que os comandos de movimento posicionais. Comandos de movimento **Force Compliant** são um exemplo de tais comandos. Quando eles são usados, o robô pode tocar a superfície dos obstáculos e deslizar por elas (ao longo da projeção da direção comandada para o movimento) ou pode apenas parar. Em contraste, com a incerteza no controle, nenhum comando controlado por posição pode garantir alcançar  $X$ . O planejamento de comandos de movimento baseados em sensores podem requerer que se leve em consideração a física do espaço de trabalho.

#### 7.5.4 Objetos Móveis

Como dissemos anteriormente, no problema básico os objetos no espaço de trabalho à exceção apenas do próprio robô, são todos obstáculos estacionários sem movimento. A seguir estendemos o problema para introduzir objetos que se movem, mas seu movimento não está sob o controle do robô. Trataremos agora de mais uma extensão: Objetos móveis, ou seja, objetos do espaço de trabalho que o robô pode movimentar livremente, arrastando, empurrando ou agarrando.

A presença de objetos móveis requer alterações no planejador, para que sejam gerados comandos de movimento com metas intermediárias a serem alcançadas, e que além disso, não estão especificadas no problema original. Por exemplo, mesmo que não exista caminho possível para uma dada configuração meta, considerando um dado arranjo do espaço de trabalho, o robô pode criar um caminho pela movimentação dos objetos de um lugar para outro. Para fazer isto, o planejador deve selecionar estas outras locações (metas intermediárias). Assim, um plano de movimento parece com uma seqüência alternativa de movimentos de trânsito e transferência: durante um movimento de trânsito o robô move-se sozinho. Durante um movimento de transferência o robô move-se levando consigo algum dos objetos móveis. Os movimentos de trânsito e transferência residem em espaços diferentes: os movimentos de trânsito ocorrem no espaço de configuração do robô, mas diferentes arranjos produzem diferentes geometrias neste espaço. Um movimento de transferência ocorre no espaço de configuração da união do robô com os objetos móveis que o robô carrega consigo. Nesta situação o principal problema para o planejador é determinar qual espaço utilizar e quando mudar de um espaço para o outro.

A maneira mais comum para um robô dispor de um objeto móvel é agarrá-lo rigidamente enquanto move-se com ele. No caso de um robô equipado com uma garra de múltiplos dedos, agarrar um objeto já é por si só um problema difícil, pois requer que várias restrições sejam satisfeitas, inclusive:

- força de fechamento (força que as garras aplicam ao objeto);
- estabilidade;
- tolerância à erros de posição;
- minimização da força de agarro;
- possibilidade geométrica.

O planejamento do agarro também é parte dos problemas de planejamento do movimento. Também são partes do problema de planejamento de movimento: empurrar e empilhar objetos, considerando a lei da gravitação.

A introdução dos objetos móveis permite que surjam novos tipos de problemas de planejamento em que a meta mais importante não é apenas atingir uma dada configuração do robô. Também são metas a composição de arranjos, como montagens das diversas partes de um objeto ou organizar as peças de uma linha de montagem.

## 7.6 Complexidade Computacional

Instâncias do mesmo tipo de problema de planejamento de movimento podem diferir consideravelmente em seu “tamanho”, por exemplo: a dimensão do seu espaço de configuração e o número de obstáculos no espaço de trabalho. É importante quantificar a performance dos métodos de planejamento - tempo requerido para resolver um problema - e a complexidade computacional inerente dos problemas de planejamento como uma função destes parâmetros. A análise dos métodos é útil para garantir sua praticabilidade e para detectar os pontos oportunos que podem ser otimizados para aumentar a eficiência. A análise dos problemas é útil para sugerir novas maneiras de formulá-los, quando existem fortes evidências de que a formulação original possui apenas soluções de custo computacional elevado.

## 7.7 Redução da Complexidade

Como foi visto no item anterior, o tempo requerido para se resolver um problema de planejamento de trajetória aumenta rapidamente com o aumento da dimensão do espaço de configuração, com o número de restrições polinomiais sobre a movimentação do robô e o grau destas restrições. Assim, é importante que estes parâmetros sejam tão pequenos quanto possível.

Para esta finalidade, podemos aproximar o problema real por um problema simplificado, porém realístico, antes de submetê-lo ao planejamento. Se a tarefa do robô é razoavelmente bem compreendida, e se é aceitável negociar alguma generalidade em troca de uma melhor performance, uma tal simplificação é quase sempre possível. Os problemas simplificados podem ser fornecidos pelo usuário ou gerados automaticamente pelo planejador a partir da descrição da entrada original.

A seguir apresentamos algumas técnicas de aproximação dos problemas de planejamento de trajetória em outros problema mais simples. A maior parte da discussão enfoca o planejamento de caminhos livres para objetos “livres” e braços robóticos.

### 7.7.1 Projeção no espaço de Configuração

Podemos reduzir a dimensão do espaço de configuração  $C$  pela substituição do robô  $A$  pela superfície ou volume varridos por ele enquanto ele se move ao longo dos eixos independentes  $r$ . Isto corresponde a projetar o espaço de configurações  $m$  – dimensional  $C$  ao longo  $r$  de suas dimensões. Seja  $C'$  o espaço projetado. Sua dimensão é  $m - r$ . Os  $C$  – obstáculos em  $C$  são projetados dentro de  $C'$  e o planejamento da trajetória é conduzido em  $C'_{livre}$ , o espaço livre em  $C'$ .

**Exemplo:** Considere um objeto bidimensional  $A$ , que pode tanto sofrer translação quanto rotação livremente no plano. Este objeto  $A$  pode ser substituído por um disco  $A'$  contendo  $A$  (idealmente, o disco é limitado pelo círculo de espalhamento mínimo de  $A$ , isto é, o menor círculo que engloba todos os pontos de  $A$ ). Esta aproximação reduz a dimensão do espaço de configuração de 3 para 2; Mas ela é realística apenas quando  $A$  não é um objeto alongado.

### 7.7.2 Fatiamento no espaço de Configuração

Outra forma de reduzir a dimensão do espaço de configuração é considerar uma seção reta de dimensão  $m - r$  através de  $C$  e planejar um caminho livre no espaço definido por esta seção reta. Isto significa proibir o movimento de  $A$  ao longo de  $r$  eixos.

**Exemplo:** Seja  $A$  um objeto tridimensional de “voo livre”. Para planejar um caminho da configuração inicial  $q_{inicial} = (T_{inicial}, \theta_{inicial})$  para a configuração meta  $q_{meta}(T_{meta}, \theta_{meta})$ , podemos considerar a configuração intermediária  $q_{inter} = (T_{meta}, \theta_{inicial})$  na qual  $A$  tem a mesma orientação  $\theta_{inicial}$  como a sua configuração inicial e a mesma posição  $T_{meta}$ , como a configuração meta. Então o planejador pode tentar gerar primeiro um caminho de  $q_{inicial}$  para  $q_{inter}$  com uma orientação fixa de  $A$  e um segundo caminho de  $q_{inter}$  para  $q_{meta}$  com a posição fixa de  $A$ . Cada um dos dois caminhos reside em uma seção reta tridimensional do espaço de configuração.

### 7.7.3 Simplificação da forma do objeto

A complexidade do planejamento de trajetória aumenta com o número de restrições polinomiais que definem o espaço livre, e o grau máximo dessas restrições. Uma forma de simplificar estas restrições é aproximar as formas, tanto do robô quanto do número de obstáculos, pela redução do número de superfícies que definem os seus limites, ou o grau das equações dessas superfícies. As técnicas mais simples, porém eficientes consistem em aproximar os objetos por cubos ou esferas. Isto reduz o grau máximo das restrições, porém ao custo de perdas na acurácia da aproximação, que podem vir a ser grandes. Uma tentativa de diminuir estas perdas pode resultar no aumento do número de superfícies que aproximam o objeto, o que aumenta o número de restrições.

Existem dois tipos distintos de aproximações: as **limitantes** e as **limitadas**:

- **Aproximações limitantes:** consistem em substituir os objetos originais por outros que contenham completamente os originais. Se um caminho livre é gerado com esta aproximação, então este também é um caminho livre com os objetos originais (o inverso não é sempre verdade);
- **Aproximações limitadas:** consistem em substituir os objetos originais por novos objetos que estejam completamente contidos dentro dos originais. Se nenhum cami-

nho livre é gerado com esta aproximação, isto implica que não existe caminho livre com os objetos originais (novamente, o inverso nem sempre é verdade). Este tipo de aproximação é geralmente utilizado para detectar se um problema tem ou não solução.

As aproximações acima podem ser usadas da seguinte maneira: primeiramente constróem-se várias aproximações, tanto limitantes quanto limitadas, com níveis de precisão crescentes. A partir daí, o planejador atua iterativamente usando aproximações cada vez mais finas, até que um caminho livre seja gerado, ou até que esteja evidente que tal caminho não existe.

#### 7.7.4 Focalizando a Atenção em um Subconjunto do Espaço de Trabalho

As técnicas descritas anteriormente são simplificações heurísticas. Elas se baseiam no fato de que na maioria dos casos a movimentação acontece em um espaço relativamente desordenado, de tal forma que ainda restam trajetórias de solução após as várias estratégias de aproximação terem eliminado algumas. Contudo, em muitos casos existem locais isolados que são muito desordenados e/ou onde o contato com os obstáculos têm de ser feitos (por exemplo, agarramento e acoplamento de partes). As simplificações heurísticas não podem ser aplicadas à estas localidades desordenadas.

Se estas localidades desordenadas podem ser identificadas antecipadamente, pode-se dividir o problema do planejamento original em subproblemas que envolvem localidades desordenadas que requerem que somente um subconjunto restrito do espaço de trabalho seja considerado. Assim o número de restrições sobre o movimento é consideravelmente reduzido com respeito ao problema original.

Em muitos outros casos essas localidades desordenadas correspondem à situações estereotipadas (por exemplo, o movimento de um robô atravessando uma porta ou virando em um corredor). Pode ser bastante útil tentar reconhecer estas situações e tentar aplicar técnicas de planejamento específicas e mais eficientes para estes casos.

### 7.8 Relacionamento com outros problemas

Um planejador automático de trajetória pode ser utilizado de várias formas: ele pode ser incluído em um sistema de programação de robôs *off-line* [30], objetivando a simplificação da tarefa de descrição de trajetórias do robô; ele também pode ser parte de um sistema de planejamento de processos gráficos interativos (p. ex., montagem de um dispositivo

eletro-mecânico) ou ser usado para checar a viabilidade geométrica das operações planejadas (ex: furar um buraco, casamento de duas partes de uma peça, etc). De maneira mais geral, ele pode ser usado para gerar automaticamente cenas gráficas animadas. Além destas aplicações de curto prazo, a meta final do planejamento automático de trajetória é criar robôs autônomos.

Embora o problema do planejamento de trajetória seja um problema crítico na criação de robôs móveis, ele não é o único. Outros grandes problemas incluem o controle de movimentação em tempo real, o sensoriamento e o planejamento em nível de resposta. Por causa da interação potencialmente complexa entre estes elementos, o projeto de robôs verdadeiramente autônomos certamente irá requerer que todos estes problemas sejam resolvidos concorrentemente. Porém, o conhecimento atual destas questões é insuficiente para apresentar um quadro compreensível de um tal projeto. Faremos a seguir uma breve análise de algumas possíveis interações entre o planejamento de trajetória e outros problemas relacionados.

### 7.8.1 Interação com Controle de Movimento em Tempo Real

Um plano de movimentação especifica um movimento a ser executado por um robô. Por exemplo, um caminho especifica uma seqüência contínua de configurações que o robô deve atravessar em direção à configuração-meta. A tarefa básica do controlador tempo real é fazer o robô executar o plano de movimentação, isto é, o caminho gerado.

Suponhamos que o plano de movimentação é um caminho livre  $\tau$ . A tarefa básica do controlador em tempo real é fazer o robô executar o plano de movimentos, isto é, fazer o robô seguir o caminho gerado.

Vamos supor agora que o plano de movimentos é um caminho livre  $\tau$ . A tarefa do controlador é transformar  $\tau$  em forças<sup>4</sup> a serem exercidas pelos atuadores do robô. Tipicamente, esta transformação é dividida em dois passos:

- **Geração de Trajetória:** consiste em decidir sobre o perfil da velocidade ao longo do caminho. Este passo pode ser feito anteriormente à execução do movimento.
- **Monitoramento de trajetória:** consiste na computação das forças a serem exercidas pelos atuadores a cada instante, para permitir realizar o movimento desejado.

---

<sup>4</sup>Aqui a palavra *força* é utilizada em um sentido genérico, indicando uma força ou um torque, dependendo da natureza do atuador.

O monitoramento da trajetória pode usar a equação dinâmica do robô, a qual expressa que as forças aplicadas pelos atuadores é igual à resultante das várias forças que agem sobre o robô durante o movimento (forças gravitacional, atrito, inércia, centrífuga e coriolis) para computar as forças que devem ser realizadas por cada atuador. Se a equação utilizada for um modelo perfeito, não é necessário nenhum *feedback*, caso contrário o mesmo torna-se necessário. Contudo, devido à várias fontes de perturbação externas e internas, algum tipo de sensoramento se faz necessário para determinar qualquer desvio entre o estado desejado e o estado real do robô. (7.1)

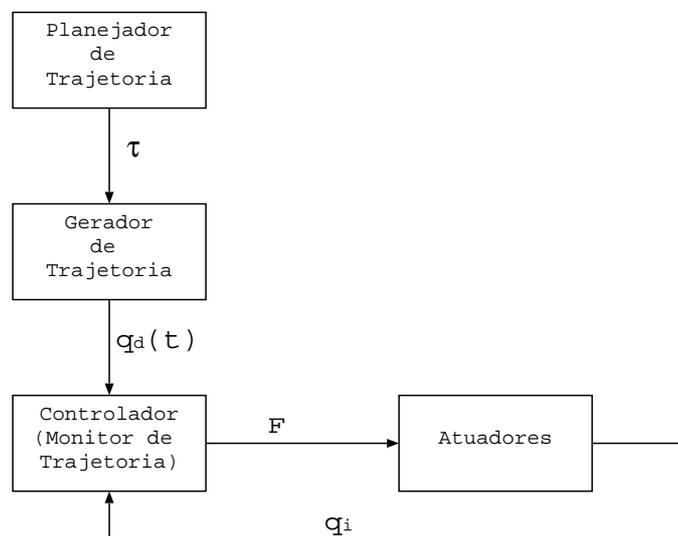


Figura 7.1: O caminho  $\tau$  produzido pelo planejador é a entrada para o gerador de trajetória que determina as dependências de tempo dos parâmetros de configuração. A saída do gerador, as configurações desejadas  $q_d$  como função do tempo, alimentam o controlador. O controlador computa a força  $F$  a ser exercida por cada atuador a partir dos desvios da configuração atual  $q_d$ , conforme medida pelos sensores, relativa à  $q_d$ .

### 7.8.2 Interação com o Sensoramento

No problema básico do planejamento de trajetória é suposto que o robô tem conhecimento completo e exato do seu ambiente. Como já foi mencionado anteriormente, esta suposição não é realística. Existem muitos problemas reais em que a geometria do espaço de trabalho não é conhecida totalmente, ou é apenas conhecida de forma parcial. Se o robô não possui nenhum conhecimento anterior sobre o seu ambiente, o planejamento é inútil. Então o robô deve confiar no sensoramento para adquirir informações enquanto move-se pelo espaço de trabalho, reagindo aos dados sensoriais. Se o robô, por exemplo, está equipado com senso-

res de proximidade, um método de campo potencial pode ser aplicado *on-line* para orientar o movimento. A configuração-meta é usada para criar o potencial atrativo, enquanto que os dados fornecidos pelos sensores de proximidade são usados para criar o potencial repulsivo. Não existe planejamento anterior e o robô pode facilmente ficar preso em um máximo local do campo potencial. O escape deste tipo de situação exige que alguma estratégia de exploração por tentativa e erro por parte do robô.

### 7.8.3 Interação com o planejamento em nível de tarefa

Uma meta do planejamento de trajetória é especificada como um arranjo espacial de objetos físicos. Alcançar esta meta pode contribuir para a realização de uma tarefa mais global. Um exemplo típico pode ser um robô de reparos cuja tarefa é diagnosticar falhas em peças de equipamentos eletromecânicos e repará-las. Encontrar uma falha típica requer que o equipamento seja inspecionado e operado, dispositivos de medida devem ser conectados e pontas de prova devem ser inseridas e removidas. Cada uma destas operações pode ser especificada por um arranjo espacial de objetos, mas decidir quais as operações devem ser executadas e qual a seqüência correta das operações - planejamento de tarefa - requer capacidades de raciocínio e planejamento de "alto nível", que são superiores àquelas possuídas por um planejador de movimentos.

Uma das áreas que mais tem contribuído para este tipo de planejamento é a **Inteligência Artificial** (IA). Os métodos de planejamento da IA têm se mostrado altamente adequados para a construção de planejadores cujas saídas são seqüências de operações, cada uma especificando um novo relacionamento espacial meta ao planejador de movimentos.

## Exercícios

1. Discuta em que aspectos o planejamento de trajetória difere de
  - (a) verificação de colisões;
  - (b) evitar colisões
2. Escreva um algoritmo para verificar se dois polígonos se chocam, dadas as coordenadas de todos os seus vértices e uma lista das arestas que interconectam pares de vértices.

- 
3. Foi dito que a representação do robô como um ponto no seu espaço de configuração transforma o problema de planejamento de trajetória de um objeto dimensionado em um problema “mais simples”. Você concorda ou discorda com este ponto de vista? Explique sua resposta.
  4. Discuta as suposições sob as quais é realístico considerar um robô com rodas como um objeto bidimensional que se move livremente em um plano.
  5. Identifique e discuta brevemente alguns problemas típicos de planejamento de trajetória para
    - (a) um robô móvel carregando objetos no ambiente de um escritório;
    - (b) um robô de uma plataforma espacial movido por foguetes e equipado com dois braços para montar plataformas orbitais.

## Capítulo 8

# Glossário

### **Robô Móvel ( $A$ )**

Um robô móvel é um dispositivo mecânico, montado sobre uma base não fixa, dotado de dispositivos para movimentação e equipado com sensores e atuadores. Possui movimentos de rotação e translação e responde à um sistema computacional. É representado como um objeto rígido, que possui um sistema Cartesiano de coordenadas  $F_A$ , cuja origem é dada por  $O_A$ .

### **Espaço de Trabalho ( $W$ )**

É o espaço físico no qual o robô se movimentará. Este espaço é modelado por um espaço euclidiano  $N$ -dimensional ( $R^N$ ) onde  $N$  é igual a 2 ou 3. A este espaço é fixado um sistema Cartesiano de coordenadas  $F_W$ , cuja origem é dada por  $O_W$ .

### **Obstáculos Fixos ( $B_i$ )**

São objetos rígidos, fixos no espaço de trabalho, que oferecem restrições ao movimento do robô.

### **Obstáculos Dinâmicos ( $B_i(t)$ )**

São objetos que podem movimentar-se pelo espaço de trabalho e alteram dinamicamente o espaço de configuração. Obstáculos móveis podem ser pessoas, outros robôs móveis ou ainda, obstáculos não conhecidos 'a priori' pelo robô.

**Configuração ( $q$ )**

A configuração de um objeto é a especificação da posição de todos os pontos deste objeto com relação a um sistema de coordenadas fixo. Assim, uma configuração  $q$  de  $A$  é a especificação da posição e orientação de  $F_A$  com relação a  $F_W$ .

**Espaço de Configuração ( $C$ )**

É a região formada por todas as possíveis configurações de  $A$  no espaço de trabalho, considerando-se as restrições impostas pelos obstáculos. Neste espaço o robô é representado por um ponto (a origem do seu sistema de coordenadas), e os obstáculos sofrem a expansão dada pela soma de uma 'região ou zona de incerteza'. O subconjunto de  $W$  ocupado por  $A$  em uma dada configuração  $q$  é denominado  $A(q)$ .

**C-Obstáculo ( $CB_i$ )**

Um C-Obstáculo é o obstáculo ( $B_i$ ) do espaço de trabalho ( $W$ ) mapeado para uma região no espaço de configuração ( $C$ ):

$$CB_i = \{q \in C / A(q) \cap B_i \neq \emptyset\}$$

O C-Obstáculo é obtido pelo obstáculo acrescido de uma 'região de incerteza'. Esta região de incerteza é construída pela origem do sistema cartesiano de coordenadas do robô ( $O_A$ ) quando este é 'passado' sobre as bordas do obstáculo ( $B$ ).

Um exemplo simples é mostrado na figura 8.1. O robô ( $A$ ) é um disco e tem a origem do seu sistema de coordenada ( $O_A$ ) fixa em seu centro. O obstáculo ( $B$ ) é um obstáculo poligonal inserido em um espaço de trabalho ( $W$ ) bidimensional ( $R^2$ ). O C-Obstáculo ( $CB$ ) é obtido pelo 'crescimento' de  $B$  isotropicamente pelo raio de  $A$ . A borda de  $CB$  é a curva formada pela origem  $O_A$  quando 'passada' por toda a borda de  $B$ .

**Região C-Obstáculo**

É a união de todos os C-Obstáculos  $CB_i$  no espaço de configuração  $C$ .

$$\bigcup_{i=1}^q CB_i$$

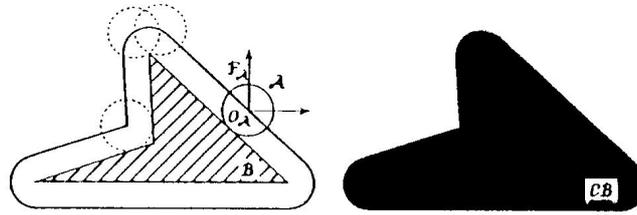


Figura 8.1: Obtenção de um C-Obstáculo

### Espaço Livre ( $C_{free}$ )

É a região no espaço de configuração livre para a atuação do robô. Também pode ser definido como a interseção vazia entre as configurações do robô com os obstáculos.

$$C_{free} = \{q \in C/A(q) \cap (\bigcup_{i=1}^q B_i) = \emptyset\}$$

### Caminho ( $\tau$ )

É o mapa contínuo das configurações do robô que o conduzem do ponto inicial ao ponto objetivo.

$$\tau : [0, 1] \rightarrow C_{free}$$

com  $\tau(0) =$  ponto inicial e  $\tau(1) =$  ponto objetivo.

# Referências Bibliográficas

- [1] AND, I. E. Object recognition with ultrasonic sensors.
- [2] ARKIN, R. Towards the unification of navigational planning and reative control. In *AAAI Spring Symposium on Robot Navigation* (1989).
- [3] ARKIN, R. C., AND BALCH, T. R. Aura: principles and practice in review. *JETAI* 9, 2-3 (1997), 175–189.
- [4] ASAMI, S. Robots in japan: Present and future. *IEEE Robotics & Automation Magazine* (June 1994), 22–26.
- [5] BARRIENTOS, A., F.PEÑÍN, L., BALAGUER, C., AND ARACIL, R. *Fundamentos de Robótica*. McGraw-Hill, 1997.
- [6] BASTOS FILHO, T. F. Oficina de robstica, 1999.
- [7] BITTENCOURT, G. Inteligência artificial distribuída. In *I Workshop de Computação do ITA* (São José dos Campos, São Paulo, Outubro, 6 1998).
- [8] BORENSTEIN, B. J. Internal correction of dead-reckoning errors with the smart encoder trailer.
- [9] BORENSTEIN, J., EVERETT, H., AND FENG, L. Where am i? sensors and methods for mobile robots positioning. University of Michigan, Apr. 1996.
- [10] BORENSTEIN, J., EVERETT, H., FENG, L., AND WEHE, D. Mobile robot positioning: Sensors and techniques, 1997.
- [11] BORENSTEIN, J., AND FENG, L. Correction of systematic odometry errors in mobile robots, 1995.

- [12] BROOKS, R. Solving the find-path problem by good representation of free-space. *IEEE Transactions on Systems, Man., and Cybernetics* 13, 3 (Mar/ Abr 1983), 190–197.
- [13] BROOKS, R. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation* 2, 1 (Mar. 1986), 14–23.
- [14] BROOKS, R. Elephants don't play chess. in *Robotics and Autonomous Systems* Vol. 6, págs. 3–15, 1990.
- [15] BROOKS, R. New approaches to robotics. in *Science*, Vol. 253, September, págs 1227–1232, 1991.
- [16] BROOKS, R. A. Integrated systems based on behaviors. *SIGART Bulletin* (August 1991), 46–50.
- [17] BROOKS, R. A. Behaviour-based humanoid robotics. In *IEEE/RSJ International Conference on Intelligent Robots and Systems: IROS 96* (November 1996).
- [18] DA SILVA BAGANTINI, D. D. Um sistema multiagente para o simulador soccerserver. Master's thesis, Universidade Federal do Rio Grande do Sul, April 2001.
- [19] DURFEE, E. H., AND ROSENSCHEIN, J. S. Distributed problem solving and multi-agent systems: Comparisons and examples. In *International Workshop on Distributed Artificial Intelligence* (May 1994).
- [20] EVERETT, H. *Sensors for Mobile Robots: Theory and Application*. A.K. Peters, 1995.
- [21] EVERETT, H., GAGE, D., GILBRETH, G., LAIRD, R., AND SMURLO, R. Real-world issues in warehouse navigation.
- [22] FERBER, J., AND GASSER, L. Intelligence artificielle distribuée. In *Tutorial Notes of the 11th Conference on Expert Systems and their Applications* (France, 1991).
- [23] GASSER, L. Boundaries, identity and aggregation: Plurality issues in multiagent systems. In *Decentralized Artificial Intelligence*, E. Werner and Y. Demazeau, Eds. Elsevier Science Publishers, 1992, pp. 199–212.
- [24] J. BORENSTEIN, L. FENG, H. E. *Navigation Mobile Robots: Systems and Techniques*. A.K. Peters, 1996.

- [25] JARVIN, R., AND BYRNE, J. Robot navigation: Touching, seeing and knowing. In *Proceedings of 1st. Australian Conf. Artificial Intelligence* (1986).
- [26] J.L. JONES, A.M. FLYNN, B. S. *Mobile Robots: Inspiration to Implementation*, second ed. A.K. Peters, 1999.
- [27] KHATIB, O. Real time obstacle avoidance for manipulators and mobile robots. *International Journal of Robotics Research*, 5(1) (1986), 90–98.
- [28] KUMPEL, D., AND SARRADILLA, F. Robot navigation strategies in a partially known environment using a space-time learning graph. In *Proceedings of the Sixth CIM - Europe Annual Conference* (1990), pp. 65–75.
- [29] KUO, B. C. *Sistemas Automáticos de Control*, third ed. CECSA, 1995.
- [30] LATOMBE, J. *Robot Motion Planning*. Kluwer Academic Publishers, Boston, MA, 1991.
- [31] LOZANO-PÉREZ, T. Spatial planning: A configuration space approach. *IEEE Transactions on Computers* C-32, 2 (Feb. 1983), 108–120.
- [32] MARCHI, J. Navegação de robôs móveis autônomos: Estudo e implementação de abordagens. Master's thesis, UFSC - Universidade Federal de Santa Catarina, 2001.
- [33] MATARIC, M. Integration of representation into goal-driven behavior-based robots. *IEEE Transactions on Robotics and Automation* 8, 3 (June 1992), 304–312.
- [34] NILSSON, N. A mobile automaton: An application of artificial intelligence techniques. In *Proceedings of the 1st. International Joint Conference on Artificial Intelligence* (1969), pp. 509–520.
- [35] OHYA, A., KOSAKA, A., AND KAK, A. Vision-based navigation by a mobile robot with obstacle avoidance using single-camera vision and ultrasonic sensing.
- [36] RIBEIRO, C., COSTA, A., AND ROMERO, R. Robôs móveis inteligentes: Princípios e técnicas. In *Anais do XXI Congresso da Sociedade Brasileira de Computação - SBC'2001* (July 2001), vol. 03, pp. 258–306.
- [37] SOAREZ, R. Curso de robots móviles. Universidad Carlos III de Madrid, Nov. 1999.
- [38] TEIXEIRA, J. *Mente, Cérebro & Cognição*. Editora Vozes, 2000.

- 
- [39] THRUN, S., AND BÜCKEN, A. Integrating grid-based and topological maps for mobile robot navigation. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence* (Aug. 1996).
- [40] WALLNER, F., AND DILLMANN, R. Real-time map refinement by use of sonar and active stereo-vision. *Robotics and Autonomous Systems*, 16 (1995), 47–56.
- [41] ZELINSKY, A. A mobile robot exploration algorithm. *IEEE Transactions on Robotics and Automation* 8, 6 (Dec. 1992), 707–717.