

## 2 Fundamentação Teórica

### 2.1 Modelo cinemático do robô móvel

A cinemática é o estudo mais básico do comportamento do sistema mecânico. Nos robôs móveis precisamos compreender este comportamento para um apropriado projeto e compreensão de como criar o software de controle. O processo para compreender o movimento do robô começa com o processo de descrição na contribuição do movimento de cada roda do robô móvel.

#### 2.1.1 Modelo cinemático e restrição

Deduzindo um modelo para um movimento completo de todo o processo, cada roda individualmente contribui ao movimento do robô, ao mesmo tempo, e estabelece restrições sobre o movimento do robô. As rodas são tratadas em conjunto com o chassi do robô, mas as forças e restrições de cada roda tem que ser expressadas em relação à referência do chassi; isto é particularmente importante na robótica móvel porque estas ações são precisamente a natureza do movimento (Sie04).

#### Representando a posição do robô

Através deste análise, modelamos o robô como um corpo rígido sobre rodas, operado em um plano horizontal. As dimensões do robô no plano são três, dois para a posição no plano e um para a orientação no eixo vertical que é ortogonal ao plano. Assume-se que existem graus de liberdade e flexibilidade adicionais devido ao eixo da roda, e as juntas que existem na roda. No entanto, neste análise considera-se o robô como corpo rígido ignorando as juntas e graus de liberdade internos do robô e de suas rodas.

Para especificar a posição do robô no plano, estabelecemos uma relação entre a referência global no plano e a referência local no robô, como se ilustra na Fig. 2.1. Os eixos  $X_I$  e  $Y_I$  definem uma base inercial arbitrária sobre o plano como uma referência global com origem  $O : X_I, Y_I$ . Para especificar a

posição do robô, escolhemos um ponto  $P$  sobre o chassi como sua posição de referência. A base  $X_R$  e  $Y_R$  define dois eixos relativos a  $P$  sobre o chassi do robô e deste modo o sistema de referência local do robô. A posição de  $P$  em relação ao sistema de referência global é especificada pelas coordenadas  $x$  e  $y$ , e a diferença angular entre as referencias global e local é dada por  $\theta$ . Agora pode-se descrever a posição do robô com o vetor de três elementos, como se ilustra na equação 2-1.

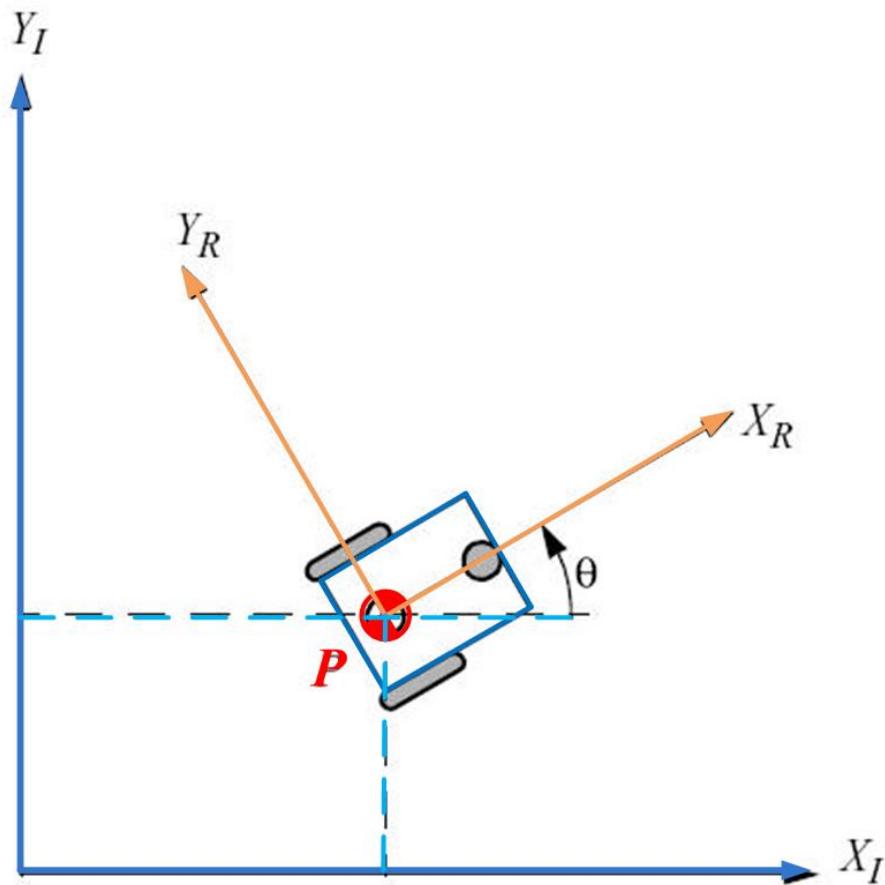


Figura 2.1: sistema de Referência global e sistema de referência local do robô

$$\xi_I = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} \quad (2-1)$$

Para descrever o movimento do robô em termos de componentes de movimento, será necessário traçar o movimento ao longo do eixo de referência global para o movimento dos eixos locais sobre o robô. O traçado é acompanhado usando a matriz de rotação ortogonal.

$$R(\theta) = \begin{pmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (2-2)$$

Esta matriz pode ser usada para traçar o movimento em relação ao sistema de referência global  $X_I, Y_I$  em termos do sistema de referência local  $X_R, Y_R$ . Esta operação é denotada por  $R(\theta)\dot{\xi}_I$  porque o cálculo depende do valor de  $\theta$ . Por exemplo, consideremos o robô da figura 2.2. Para este robô,  $\theta = \frac{\pi}{2}$  podemos calcular a matriz de rotação instantânea do robô segundo as equações 2-3 e 2-4.

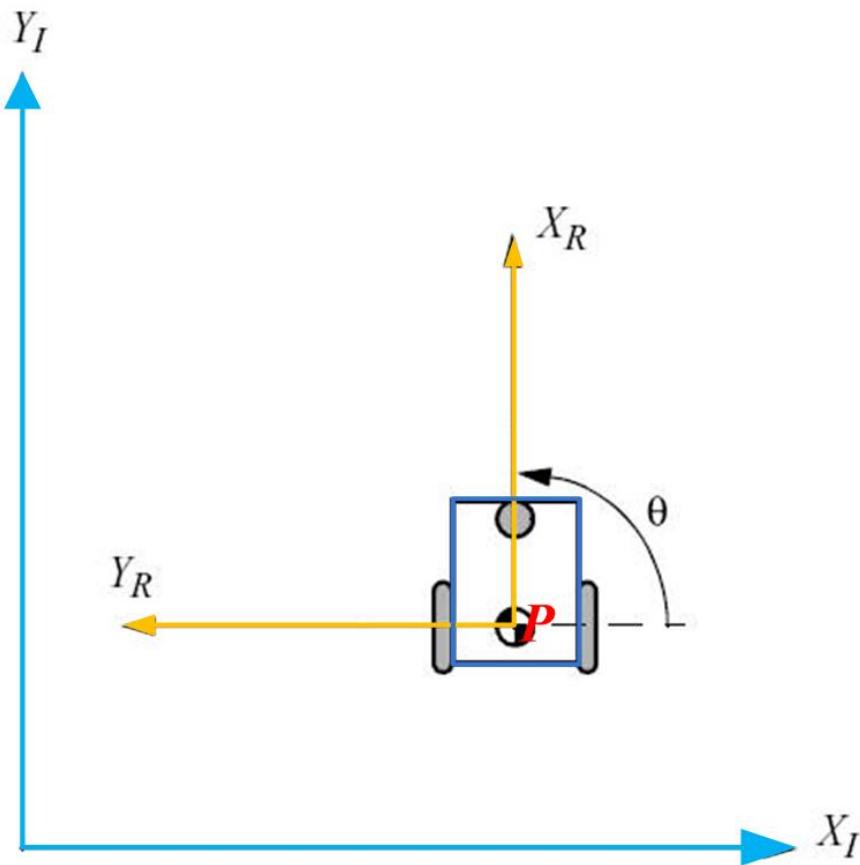


Figura 2.2: Robô móvel alinhado com os eixos do sistema de referência local

$$\dot{\xi}_R = R\left(\frac{\pi}{2}\right)\dot{\xi}_I \quad (2-3)$$

$$R\left(\frac{\pi}{2}\right) = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (2-4)$$

Dada uma velocidade  $(\dot{x}, \dot{y}, \dot{\theta})$  no sistema de referência global, podemos calcular as componentes de movimento ao longo do eixo local.

$$\dot{\xi}_R = R\left(\frac{\pi}{2}\right)\dot{\xi}_I = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \dot{y} \\ -\dot{x} \\ \dot{\theta} \end{pmatrix} \quad (2-5)$$

### Modelo cinemático direto

No caso mais simples, o mapeamento descrito pela equação 2-3 é suficiente para gerar a fórmula que captura a cinemática direta do robô móvel. Como se desloca o robô dado sua geometria e a velocidade da suas rodas?. Mais formalmente, considere-se o exemplo ilustrado na figura 2.3.

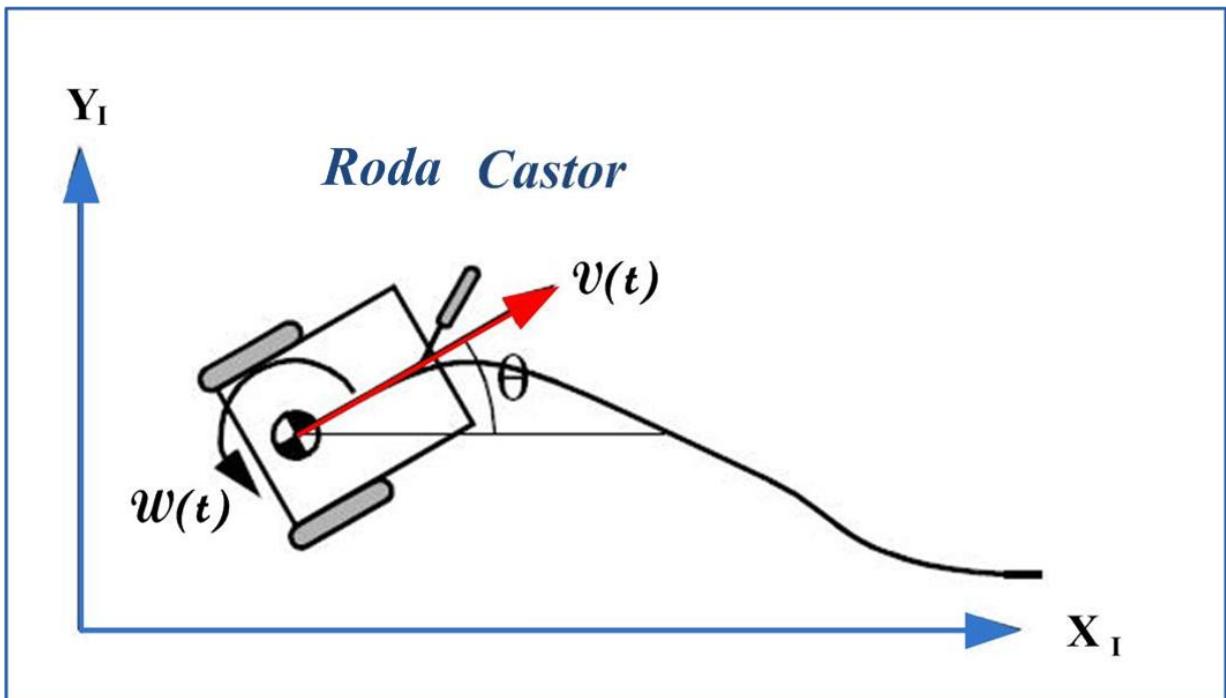


Figura 2.3: Referência global da unidade do robô

O robô possui duas rodas, cada uma com um motor próprio e independente, e cada roda tem um diâmetro  $r$ . Seja um ponto  $P$  centrado entre as duas rodas, cada uma a uma distância  $l$  de  $P$ . Dados  $r$ ,  $l$ ,  $\theta$ , e a velocidade própria de cada roda,  $\dot{\varphi}_1$  e  $\dot{\varphi}_2$ , o modelo da cinemática direta poderia predizer a velocidade de todo o robô em relação ao sistema de referência global:

$$\dot{\xi}_I = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = f(l, r, \theta, \dot{\varphi}_1, \dot{\varphi}_2) \quad (2-6)$$

Da equação 2-3 sabemos que podemos calcular o movimento do robô no sistema de referência global do movimento em relação ao sistema de referência local:  $\dot{\xi}_I = R(\theta)^{-1}\dot{\xi}_R$ . Por conseguinte, a estratégia seria primeiro calcular a contribuição de cada roda no sistema de referência local  $\dot{\xi}_R$ .

Suponha que o sistema de referência local esteja alinhado tal que o robô se desloque ao longo do eixo  $+X_R$ , como se ilustra na figura 2.1. Primeiro consideremos a contribuição da velocidade de cada roda para a velocidade de

translação em  $P$  na direção de  $+X_R$ . Ambas rodas têm que girar à mesma velocidade se quisermos que o robô avance numa só direção, a contribuição de cada roda à velocidade do ponto  $P$  é a mesma,  $\dot{x}_{r1} = r\dot{\varphi}_1$ ,  $\dot{x}_{r2} = r\dot{\varphi}_2$  e  $\dot{x}_{r1} = \dot{x}_{r2}$ . Para poder assegurar uma velocidade correta no componente  $\dot{x}_R$  de  $\dot{\xi}_R$ , fazemos com que a velocidade no ponto  $P$  seja uma velocidade ponderada das rodas. O valor de  $\dot{y}_R$  é sempre zero, porque nenhuma roda contribui com o movimento nessa direção. Finalmente, calculamos o componente rotacional  $\dot{\theta}_R$  de  $\dot{\xi}_R$ . Outra vez, a contribuição de cada roda pode ser calculada independentemente e adicionada; considere-se a roda direita (Roda1). O contínuo giro desta roda resulta num contador no sentido contrário de giro do relógio no ponto  $P$ . Lembrando que se a roda 1 gira sozinho, o robô gira com a Roda2(roda esquerda) como apoio. A velocidade de rotação  $\omega_1$  em  $P$  pode ser calculado porque a roda esta girando instantaneamente ao longe do arco de círculo de radio  $2l$ :

$$\omega_1 = \frac{r\dot{\varphi}_1}{2l} \quad (2-7)$$

A mesma equação 2-7 se aplica à roda esquerda, com a exceção que o giro resulta num giro no sentido de giro do relógio no ponto  $P$ , 2-8.

$$\omega_2 = -\frac{r\dot{\varphi}_2}{2l} \quad (2-8)$$

Combinando estas equações individuais 2-7 e 2-8, o modelado cinemático da unidade do robô é:

$$\dot{\xi}_I = R(\theta)^{-1} \begin{pmatrix} \frac{r\dot{\varphi}_1 + r\dot{\varphi}_2}{2} \\ 0 \\ \frac{r\dot{\varphi}_1}{2l} + \frac{r\dot{\varphi}_2}{2l} \end{pmatrix} \quad (2-9)$$

Esta aproximação do modelado cinemático 2-9 pode prover informação acerca do movimento do robô dado seus componentes de velocidade das rodas. De qualquer modo nós desejamos determinar o possível espaço do movimento para cada desenho de chassi do robô, para fazer isto se tem que estudar as restrições sobre o movimento do robô imposto pelas rodas.

### Restrições na cinemática da roda

O primeiro passo para encontrar o modelo cinemático do robô é expressar as restrições sobre o movimento das rodas individualmente (Sie04). O movimento individual das rodas pode mais tarde ser combinado para calcular o movimento de todo o robô. Há muitos tipos de modelos básicos das rodas com propriedades cinemática variadas, sim embargo aqui só trataremos de dos tipos que são os que mais interessam neste trabalho.

De qualquer modo, muitas suposições importantes simplificaram esta apresentação, é assumido que o plano da roda sempre é vertical e que existe um único ponto de contato entre a roda e o chão; mas lá disto é assumido que não há deslizamento no ponto de contato. Isto é, a roda experimenta somente movimento baixo condições de rolamento e rotação pura no eixo vertical através do ponto de contato.

Baixo estas suposições, apresenta-se dois restrições para cada tipo de roda; a primeira restrição reforça o conceito de contato de rolamento - que a roda deve girar quando o movimento toma lugar na direção apropriada -; A segunda restrição reforça o conceito de não deslizamento lateral - que a roda não deve deslizar-se ortogonalmente ao plano da roda.

A roda que tratamos neste trabalho é a roda standard fixa que não tem eixo vertical de rotação para cabeceio. Seu ângulo ao chassi é fixo, e isto limita seu movimento só para atrás e adiante e no plano de rotação do ponto de contato com o chão, a figura 2.4, descreve uma roda fixa standard  $A$  e indica a posição de sua postura relativo ao sistema de referência local  $X_R, Y_R$ . A posição de  $A$  esta expressado em coordenadas polares pela distância  $l$  e o ângulo  $\alpha$ . O ângulo do plano da roda relativo ao chassi esta denotado por  $\beta$ , o qual é fixa. A roda de radio  $r$ , pode girar no tempo, e então seu posição rotacional em torno de seu eixo horizontal é uma função do tempo  $t : \varphi(t)$ .

As restrições de rolamento para esta roda reforça que tudo movimento ao longe da direção do plano da roda deve ser acompanhado a razão de giro da roda tal que exista puro rolamento no ponto de contato:

$$\begin{pmatrix} \sin(\alpha + \beta) & -\cos(\alpha + \beta) & (-l) \cos \beta \end{pmatrix} R(\theta) \dot{\xi}_I - r \dot{\varphi} = 0 \quad (2-10)$$

O primeiro termo da equação 2-10 denota o movimento total ao longe do plano da roda; os três elementos do vetor sobre a esquerda representam o mapeamento de cada variável  $\dot{x}, \dot{y}, \dot{\theta}$  para as contribuições do movimento ao longe do plano da roda. Note que o termo  $R(\theta) \dot{\xi}_I$  é usado para transformar os parâmetros do movimento  $\dot{\xi}_I$  que esta no sistema de referência global  $X_I, Y_I$  dentro dos parâmetros do movimento no sistema de referência local  $X_R, Y_R$  que são mostradas na equação 2-3.

A restrição de deslizamento para esta roda reforça que o componente do movimento ortogonal da roda ao plano da roda deve ser zero, como se amostra na equação 2-11:

$$\begin{pmatrix} \cos(\alpha + \beta) & \sin(\alpha + \beta) & l \sin \beta \end{pmatrix} R(\theta) \dot{\xi}_I = 0 \quad (2-11)$$

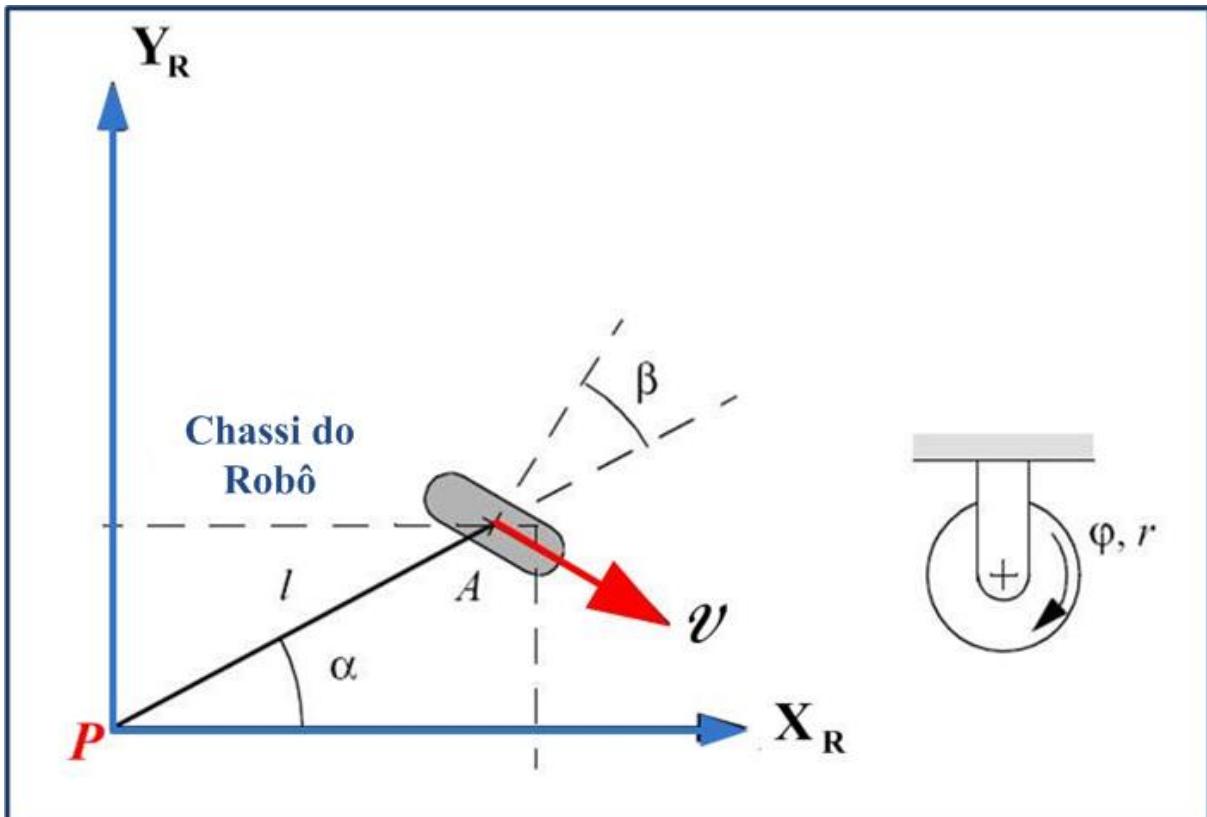


Figura 2.4: Roda standard fixa e seus parâmetros

### Restrição na cinemática do robô

Dado o robô móvel com \$m\$ rodas, pode-se calcular as restrições cinemática do chassi do robô (Sie04). Cada roda impõe restrições ao movimento do robô e então o processo é simplificado à combinação apropriada de todas as restrições cinemática aparecidas de todas as rodas.

A roda standard fixa que usamos neste trabalho tem impacto sobre a cinemática do chassi do robô e sim embargo requer considerações quando se calcula as restrições cinemática do robô. Suponha que o robô tem um total de \$N\_f\$ rodas standard e \$\beta\_f\$ seja a orientação dessas rodas fixas. Denotaremos \$\dot{\varphi}\_f(t)\$ como a velocidade angular das rodas.

Então a restrição de rolamento de todas as rodas pode agora ser colecionada numa única equação 2-12:

$$J_1(\beta_f)R(\theta)\dot{\xi}_I - J_2\dot{\varphi} = 0 \quad (2-12)$$

Nesta expressão 2-12 se tem que \$J\_2\$ é uma matriz constante diagonal \$N \times N\$ onde as entradas são os valores dos rádios \$r\$ de cada roda; \$J\_1(\beta)\$ denota uma matriz com projeção para todas as rodas no movimento. Onde na equação 2-13 \$J\_{1f}\$ é uma matriz constante das projeções de todas as rodas fixas. Isto

tem o tamanho ( $N_f \times 3$ ), com cada fila consistindo dos três termos na matriz da equação 2-10 para cada roda fixa.

$$J_1(\beta_f) = \begin{pmatrix} J_{1f} \end{pmatrix} \quad (2-13)$$

Em resumo, a equação 2-12 representa a restrição que todas as rodas standard tem ao girar em torno a seu eixo horizontal. Agora usamos a mesma técnica para colecionar as restrições de deslizamento para todas as rodas numa única expressão:

$$C_1(\beta_f)R(\theta)\dot{\xi}_I = 0 \quad (2-14)$$

$$C_1(\beta_f) = \begin{pmatrix} C_{1f} \end{pmatrix} \quad (2-15)$$

Das equações 2-14 e 2-15,  $C_{1f}$  é uma matriz de ( $N_f \times 3$ ) onde seus termos são os mesmos da equação 2-11 para todas as rodas fixas. Assim a equação 2-14 é uma restrição sobre todas as rodas fixas e que seus termos de movimento ortogonal aos planos da roda devem ser zero. Esta restrição sobre todas as rodas fixas tem um grão significado sobre a manobrabilidade do chassi do robô. Pelo tanto a equação geral para a cinemática do robô queda como amostra-se na equação 2-16.

$$\begin{pmatrix} J_1(\beta_f) \\ C_1(\beta_f) \end{pmatrix} R(\theta)\dot{\xi}_I = \begin{pmatrix} J_2\varphi \\ 0 \end{pmatrix} \quad (2-16)$$

## 2.2

### Percepção Sensorial

Uma das mais importantes tarefas de um sistema autônomo de alguma classe é adquirir conhecimento acerca do ambiente. Isto é realizado fazendo medições usando vários tipos de sensores e logo extraindo informação significativa dessas medidas.

Nesta seção se apresentam os mais comuns sensores usados em robôs móveis e logo se discute um pouco sobre a estratégia da extração da informação dos sensores (Eve95).

### Sensores para robôs móveis

Há uma ampla variedade de sensores usados em robôs móveis, alguns destes sensores são usados para medidas de simples valores como a temperatura interna da eletrônica do robô ou a velocidade angular dos motores. Outros, mais sofisticados podem ser usados para adquirir informação acerca do ambiente do robô ou sobre a medida direita da posição global do robô. Nesta seção trata sobre sensores usados para a extração acerca do ambiente do robô.

## H

Tabela 2.1: Classificação dos sensores usados nas aplicações de robôs móveis

Classificação General	Sensor y Sistema	PC ou EC	A ou P
Sensores Tacteies	ópticos	EC	A
Sensores do motor	encoders e potenciômetros	PC	P
Velocidade	doppler som	EC	A
Visão	CCD/CMOS câmeras	EC	P

Segundo (Sie04) classificam-se os sensores usando dos importantes eixos funcionais : *proprioceptive/exteroceptive* e *passive/active*.

Sensores *Proprioceptive* medem valores internos do sistema do robô; por exemplo, velocidade do motor, carga da roda, ângulo da junta do robô, voltagem da bateria.

Sensores *Exteroceptive* adquiram informação do ambiente do robô, por exemplo, medida da distância, intensidade da luz, amplitude do som.

O sensor *Passive* mede as condições ambientais da energia de entrada ao sensor, por exemplo as sondas de temperatura, microfones e *CCD* das câmaras.

O sensor *Active* pode dirigir mais interações controladas com o ambiente, eles muitas vezes realizam um desempenho superior; de qualquer modo, estes sensores introduz muito risco como a interferência entre seus sinais.

A tabela 2.1 amostra a classificação dos sensores mais usados para aplicações de robôs móveis. Os sensores mais utilizados vão ser comentados brevemente. Onde A = ativos; P = passivos; P/A = passivos e ativos; PC = proprioceptive; EC, Exteroceptive.

Os tipos de sensores da tabela 2.1 estão ordenadas em ordem ascendente de complexidade. A continuação se descreve alguns sensores brevemente.

### **Encoder ópticos**

Os encoder ópticos são os dispositivos mais populares para a medida angular da posição e velocidade para o controle do motor. Nos robôs móveis, os encoder são usados para o controle da posição e a velocidade das rodas. Eles estimam a posição do robô no sistema do robô e quando é aplicado ao problema da localização, algumas correções são requeridas para estes casos como será discutido mais adiante.

A Figura 2.5 ilustra um disco típico do sensor montado no eixo do motor para aplicações de leitura da posição e direção do motor. Quando o motor gira, o estator gera 2 quadratura de pulsos e um pulso index. A partir destes dados é possível inferir velocidades. Estes sinais são ilustradas na Figura 2.6:

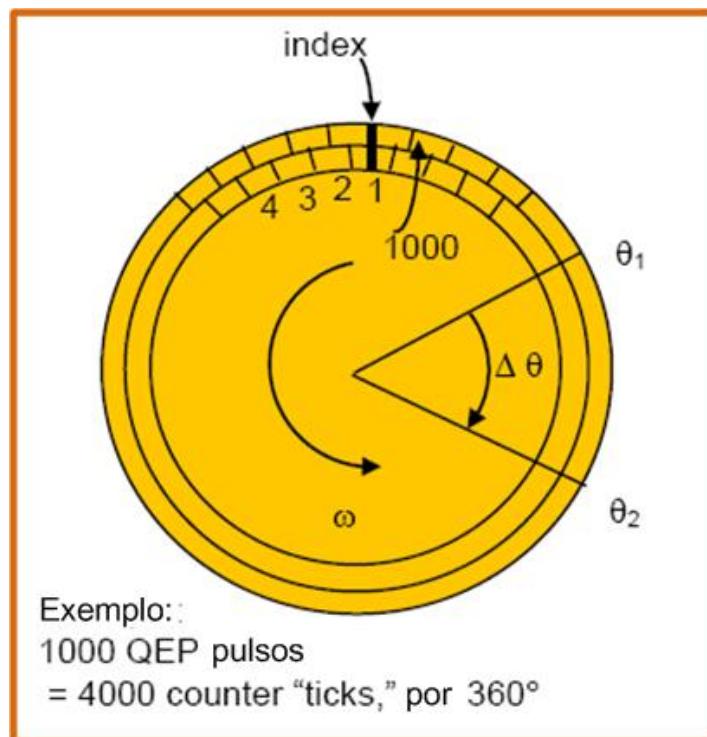


Figura 2.5: Disco do sensor de posição, usado para inferir velocidade

### Sensores ultra-som

O princípio básico de estes sensores é transmitir um paquete de pressão em forma de onda e medir o tempo que toma em refletir esta onda e retornar ao receptor. A distância  $d$  do objeto que causa a reflexão pode ser calculado baseado sobre a velocidade de propagação do som  $c$  e o tempo de vôo  $t$ .

$$d = \frac{c \times t}{2} \quad (2-17)$$

A velocidade do som no ar está dado por:

$$c = \sqrt{\gamma RT} \quad (2-18)$$

Onde :  $\gamma$  = taça da temperatura específica;  $R$  = constante do gás;  $T$  = temperatura em K;

A forma de como se propaga esta onda se ilustra na Figura 2.7

## 2.3 Localização-Posicionamento

A navegação é um dos maiores desafios que enfrenta o robô móvel, o êxito da navegação requer o êxito na percepção, na localização, na cognição e no controle de movimento, o robô tem que modular a saída do motor para conseguir a trajetória desejada.

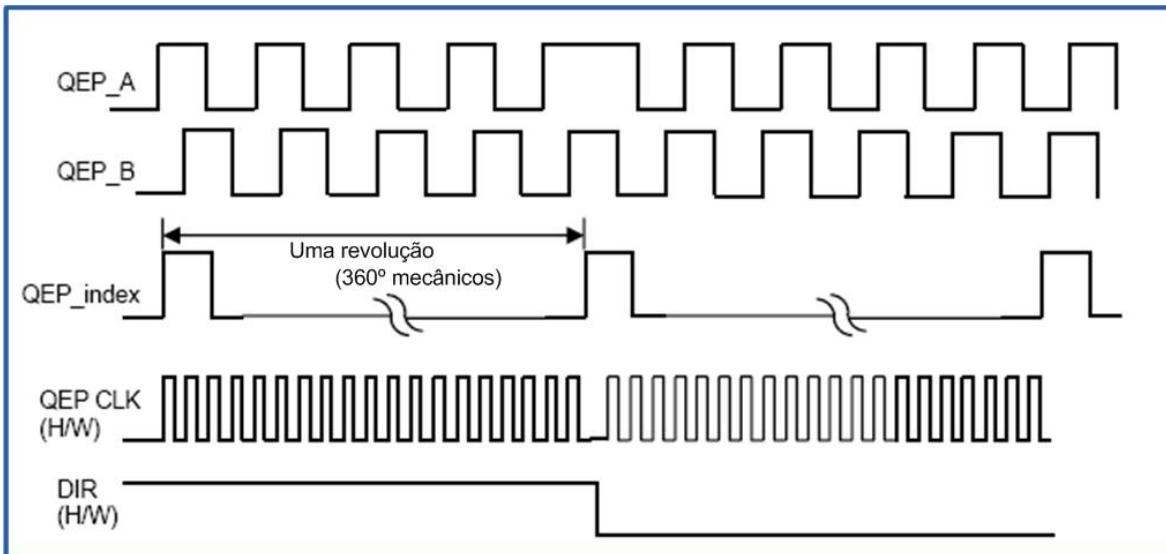


Figura 2.6: Pulsos de quadratura do encoder

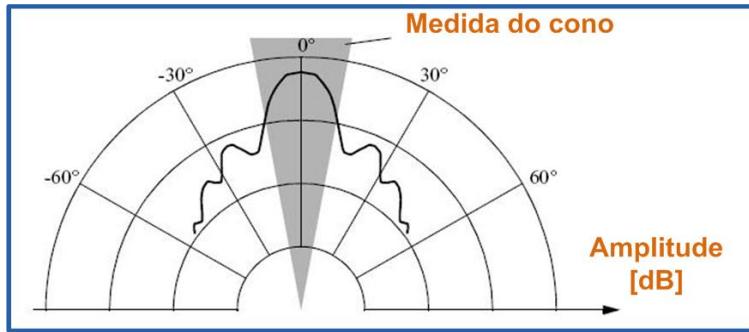


Figura 2.7: Intensidade típica da distribuição de um sensor ultra-sônico

Se poderia-se adjuntar um sensor *GPS* ao robô móvel, muitos dos problemas da localização poderiam ser obviados; este sensor poderia informar ao robô sua posição exata, dentro e fora do ambiente, logo a resposta à pergunta, Onde estou?, poderia ser disponível, infelizmente, tão sensor não é geralmente prático, primeiro por o alto custo e segundo porque a localização implica mais que o conhecimento da posição absoluta no sistema de referência terrestre. Considere-se um robô que inter-atua com humanos. Este robô poderia precisar identificar seu posição absoluta, mas seu posição relativa em relação ao humano também é igual de importante. Sua localização pode incluir a identificação de humanos usando um arranjo de sensores, logo calcular sua posição relativa ao humano.

Claramente, os sensores e atuadores do robô jogam um papel integral no processo de localização. Isto é porque da imprecisão e imperfeição dos sensores e atuadores que a localização dificulta-se.

Os sensores são fundamentalmente importantes para o processo de per-

cepção, e por conseguinte o grau ao qual estes sensores podem distinguir o estado do ambiente é crítico, o ruído nos sensores induz uma limitação sobre a consistência da leitura do sensor no mesmo ambiente. Muitas vezes a fonte do problema no ruído do sensor é que algumas características do ambiente não são capturadas pela representação do robô e são obviados.

Por exemplo o sistema sensorial humano, particularmente visual, tende a recepcionar entradas únicas em cada local único, é dizer que cada lugar se mira diferente. A capacidade deste único mapa é somente aparente quando um considera situações onde a falha se mantém; é dizer se um humano passa por um edifício não conhecido e completamente obscuro, quando seu sistema visual somente observa a obscuridade, o sistema de localização diminui rapidamente.

Nos robôs, a leitura do sensor não é único, isto é chamado *aliasing* do sensor, e é uma norma e não a exceção. Num robô com muitos sensores tem uma grande variedade de estados no ambiente por perceber e que muitos destes estados poderiam ativar o mesmo valor a estes sensores, assim o robô não pode distinguir de entre muitos estados. Outro problema de localização é o ruído do atuador, em particular uma só ação tomada por o robô pode ter muitos possíveis resultados diferentes, em resumo o atuador de um robô móvel introduz incerteza sobre o estado futuro, por conseguinte o simples fato do movimento tende a incrementar a incerteza do robô móvel.

É importante notar que desde o ponto de vista do robô, a incerteza do atuador é visto como um erro em odometria, ou a incapacidade do robô de estimar seu própria posição no tempo usando o conhecimento da sua cinemática e dinâmica. A fonte do erro geralmente amostra um modelo incompleto do ambiente, porque o robô não modela o fato de que o chão possa ser escorregadio e possa deslizar-se, todos estes fontes de erro não modeladas resultam numa imprecisão entre o movimento físico do robô, a intenção de movimento e a estimarão sensorial do movimento (Sie04).

### 2.3.1

#### **Modelo do erro para a estimarão da posição por odometria**

Geralmente a posição de um robô é representada por o vetor:

$$p = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} \quad (2-19)$$

Para um robô diferencial a posição pode ser estimada começando do conhecimento da posição e integrando o movimento (somando o incremento da distância percorrida), para um sistema discreto com um intervalo de

amostragem fixo  $\Delta t$  o incremento das distâncias de percorrido ( $\Delta x; \Delta y; \Delta\theta$ ) são:

$$\Delta x = \Delta s \cos(\theta + \frac{\Delta\theta}{2}) \quad (2-20)$$

$$\Delta y = \Delta s \sin(\theta + \frac{\Delta\theta}{2}) \quad (2-21)$$

$$\Delta\theta = \frac{\Delta s_r - \Delta s_l}{b} \quad (2-22)$$

$$\Delta s = \frac{\Delta s_r + \Delta s_l}{2} \quad (2-23)$$

onde:  $(\Delta x; \Delta y; \Delta\theta) =$  trajetória percorrida no ultimo intervalo de amostragem;  $\Delta s_r; \Delta s_l =$  distância percorrida pela roda direita e esquerda respectivamente;  $b =$  distância entre as dois rodas do robô diferencial.

Assim obtemos a posição atual  $p'$ :

$$\begin{aligned} p' &= \begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = p + \begin{pmatrix} \Delta s \cos(\theta + \frac{\Delta\theta}{2}) \\ \Delta s \sin(\theta + \frac{\Delta\theta}{2}) \\ \Delta\theta \end{pmatrix} \\ &= \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} + \begin{pmatrix} \Delta s \cos(\theta + \frac{\Delta\theta}{2}) \\ \Delta s \sin(\theta + \frac{\Delta\theta}{2}) \\ \Delta\theta \end{pmatrix} \end{aligned} \quad (2-24)$$

Usando a relação para  $(\Delta s; \Delta\theta)$  das equações 2-23 e 2-22, mais adiante obtemos a equação básica para a atual posição de odometria (para robôs de manejo diferencial).

$$\begin{aligned} p' &= f(x, y, \theta, \Delta s_r, \Delta s_l) \\ &= \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} + \begin{pmatrix} \frac{\Delta s_r + \Delta s_l}{2} \cos(\theta + \frac{\Delta s_r - \Delta s_l}{2b}) \\ \frac{\Delta s_r + \Delta s_l}{2} \sin(\theta + \frac{\Delta s_r - \Delta s_l}{2b}) \\ \frac{\Delta s_r - \Delta s_l}{b} \end{pmatrix} \end{aligned} \quad (2-25)$$

Tal como foi discutido antes, a atual posição de odometria pode dar somente uma estimarão da posição atual irregular. Devido à integração das incertezas do erro de  $p$  e o erro de movimento durante o movimento incremental ( $\Delta s_r; \Delta s_l$ ), a posição do erro baseado na integração da odometria cresce com

o tempo.

Agora se tem que estabelecer o modelo para a integração da posição  $p'$  para obter a matriz de co-variância  $\sum_{p'}$  da posição estimada. Para fazer isto, assumimos que o ponto inicial da matriz de co-variância  $\sum_p$  é conhecido; para o incremento do movimento  $(\Delta s_r; \Delta s_l)$  assumimos a seguinte matriz de co-variância  $\sum_{\Delta}$ :

$$\sum_{\Delta} = covar(\Delta s_r, \Delta s_l) = \begin{pmatrix} k_r |\Delta s_r| & 0 \\ 0 & k_l |\Delta s_l| \end{pmatrix} \quad (2-26)$$

Onde  $\Delta s_r$  y  $\Delta s_l$  são as distâncias da trajetória de cada roda, e  $k_r, k_l$  são as constantes do erro representando os parâmetros não determinísticos do controle do motor e a inter-ação da roda com o chão, como se observa na equação 2-26 temos feito as seguintes suposições:

- Os dois erros do controle individual das rodas são independentes.
- As variâncias dos erros (ambas rodas), são proporcionais ao valor absoluto das distâncias percorridas  $(\Delta s_r; \Delta s_l)$ .

Os erros do movimento são devido ao movimento impreciso por motivos de deformação das rodas, deslizamento, rugosidade do chão, erro nos encoder, etc. O valor das constantes de erro  $k_r$  y  $k_l$  dependem do robô e o ambiente e deveriam ser experimentalmente estabelecidos para melhorar o desempenho.

Se assumimos que  $p$  y  $\Delta_{rl} = (\Delta s_r; \Delta s_l)$  não são correlacionados, e a derivada da equação 2-25 é favoravelmente aproximada pela expansão de Taylor de primeiro ordem ( linearização), concluímos usando a lei de propagação do erro:

$$\sum = \nabla_p f \cdot \sum_p \nabla_p f^T + \nabla_{\Delta_{rl}} f \cdot \sum_{\Delta} \nabla_{\Delta_{rl}} f^T \quad (2-27)$$

A matriz de co-variância  $\sum_p$  é por suposto sempre permitido pela  $\sum_{p'}$  do passo anterior, e pode assim ser calculado depois de especificar o valor inicial.

Usando a equação 2-25, podemos desenvolver os Jacobianos,  $F_p = \nabla_p f$  e  $F_{\nabla_{rl}} = \nabla_{\nabla_{rl}} f$ :

$$\begin{aligned} F_p &= \nabla_p f = \nabla_p(f^T) = \left( \frac{\partial f}{\partial x} \quad \frac{\partial f}{\partial y} \quad \frac{\partial f}{\partial \theta} \right) \\ &= \left( \begin{array}{ccc} 1 & 0 & -\Delta s \sin(\theta + \frac{\Delta \theta}{2}) \\ 0 & 1 & \Delta s \cos(\theta + \frac{\Delta \theta}{2}) \\ 0 & 0 & 1 \end{array} \right) \end{aligned} \quad (2-28)$$

$$F_{\Delta_{rl}} = \begin{pmatrix} \frac{1}{2} \cos(\theta + \frac{\Delta\theta}{2}) - \frac{\Delta s}{2b} \sin(\theta + \frac{\Delta\theta}{2}) & \frac{1}{2} \cos(\theta + \frac{\Delta\theta}{2}) + \frac{\Delta s}{2b} \sin(\theta + \frac{\Delta\theta}{2}) \\ \frac{1}{2} \sin(\theta + \frac{\Delta\theta}{2}) + \frac{\Delta s}{2b} \cos(\theta + \frac{\Delta\theta}{2}) & \frac{1}{2} \sin(\theta + \frac{\Delta\theta}{2}) - \frac{\Delta s}{2b} \cos(\theta + \frac{\Delta\theta}{2}) \\ \frac{1}{b} & -\frac{1}{b} \end{pmatrix} \quad (2-29)$$

## 2.4

### Navegação de robôs móveis

A navegação baseada em sensores propicia ao robô um comportamento dependente da situação interna do veículo e do meio onde está inserido (Fre92). O sistema de percepção é o responsável pelo tratamento e envio de dados coletados por sensores, determinando o tipo de movimento a ser executado pelo veículo.

Periodicamente podem ser verificados vários parâmetros internos próprios do robô (nível de energia, presença de falhas entre outros). Esses fatores podem ser responsáveis por avarias e problemas que impeçam o perfeito desempenho do robô. Uma vez detectados, poderão ser solucionadas pelo sistema de controle.

Este ambiente também pode ser sensoriado periodicamente, possibilitando diretamente a execução de uma determinada ação com base nos dados obtidos por sensoriamento, ou a execução da sub-tarefa de mapeamento do ambiente.

Na navegação baseada em sensores, as três sub-tarefas que compõem as técnicas de controle são realizadas em tempo real de operação. Ao contrário da navegação em locais conhecidos, em que o mapa do ambiente previamente conhecido, sendo o planejamento determinado numa fase anterior, a navegação por sensores, além de exigir um sistema de percepção mais complexo, necessita de técnicas de controle de alto desempenho de execução nas tomadas de decisão, e ao nível operacional, a cada instante. A capacidade de tratar ruídos outra característica que devem apresentar os sistemas baseados em sensores, pois esses são bastante comuns quando se trata com tais tipos de dispositivos. O sistema de controle deve ser tolerante a sinais ruidosos, conseguindo diminuir a sua influência maléfica no comportamento do veículo.

Uma das vantagens propiciadas pela utilização de sensores é a possibilidade do robô móvel operar autonomamente. Tal fato possibilita a existência de veículos que possam realizar suas tarefas completamente desconectados de qualquer tipo de *hardware* de apoio, como também sem qualquer tipo de interferência externa (Ben96). Assim, um comportamento inteligente pode ser esperado dos robôs que realizam suas tarefas, baseados nas informações obti-

das do ambiente e de si próprios. O trabalho desenvolvido aborda o estudo de sistemas de controle que se enquadram nessa categoria de controle de navegação.

#### 2.4.1

##### Controle de agentes

Ao se tratar com navegação baseada em sensores, consideram-se os robôs móveis envolvidos como agentes. Um agente é um processo capaz de possuir percepção, computação e ação dentro de seu mundo, podendo ser físico ou no.

##### Arquiteturas para controle de agentes

Baseado em planos (*planner-based approach*), que é uma estratégia deliberativa, onde a arquitectura de controle é do tipo *top-down*(Mat94). É usado um modelo de mundo centralizado para verificar a informação sensorial e gerar ações. Essa abordagem permite a formulação explícita das tarefas e metas do sistema, e estimarão da qualidade da desempenho do agente.

Muitos dos métodos tradicionais utilizados para planejamento de caminho em locais conhecidos estão sendo modificados para serem utilizados no caso de navegação por sensores *planner-based*. Estudos estão sendo realizados para utilizarem-se heurísticas do tipo campo potencial e diagrama de Voronoi na navegação sensoriada (Cho95). Por outro lado, incertezas que possam vir a ocorrer no sensoriamento e mudanças do ambiente podem requerer freqüente remanejamento, sendo o seu custo muitas vezes alto para sistemas complexos. A abordagem baseada em planejamento é criticado devido a esta dificuldade de convivência com a complexidade do problema consequentemente no permitindo reações em tempo real e tratamento de ruídos.

Várias pesquisas estão sendo realizadas com o objetivo de desenvolverem-se sistemas de controle de agentes em tempo real . Uma das abordagens mais proeminentes são do tipo *bottom-up* e implementam estratégias de controle para agentes como uma coleção de pares de ação-reação pré-programados com mínimos estados. Esses sistemas não mantém modelos internos de representação do ambiente, mas simplesmente comandam a ação apropriada para um determinado conjunto de valores sensoriados. Conta-se com uma relação direta entre os valores sensoriados e a ação , associada a uma rápida realimentação do ambiente. Estratégias puramente reativas tem-se mostrado eficazes para uma variedade de problemas que podem ser definidos temporariamente, mas são inflexíveis para a execução em tempo real dividido a sua incapacidade de armazenar informação dinamicamente.

Arquiteturas híbridas enfocam um compromisso entre abordagens puramente reativas e abordagens deliberativas pelo emprego de um sistema reativo para controle de baixo nível e um planejamento para decisões de alto nível. Sistemas híbridos formam um campo de elevado potencial para pesquisa. Como exemplo, pode-se citar o planejamento reativo ou execução reativa que utiliza primitivas de alto nível para planejamento, as quais cuidam de todos os detalhes da execução (*Reactive Actions Packages*), bem como o PRS (*Procedure Reasoning System*) que é uma arquitetura para inovação de regras flexíveis (Mat94) entre outros. Esses sistemas tendem separar os módulos de controle em duas ou mais partes independentes, mas intercomunicação. Em muitos casos, processos reativos de baixo nível cuidam das ações imediatas de sobrevivência do robô, enquanto os níveis mais altos selecionam as sequências de ações.

A abordagem baseada em comportamento é uma extensão de sistemas reativos, mas que apresenta alguns aspectos da abordagem baseada em planejamento (Jor03). Embora algumas vezes seja encontrada numa forma relativamente confusa na literatura, a estratégia baseada no comportamento consideravelmente mais potente do que as abordagens puramente reativas, uma vez que não possui limitações em seus estados internos, formado por pares restritos de ação-reação. Mesmo que os sistemas baseados no comportamento possuam algumas propriedades dos sistemas reativos, casualmente contenham componentes reativos, sua computação é limitada a consultas. Esses sistemas podem usar diferentes formas de representação interna do ambiente bem como executar computação distribuída, a fim de decidir qual ação-efeito aplicar (Mat94).

#### 2.4.2

#### Técnicas de controle inteligentes

O comportamento inteligente pretendido pode ser alcançado com o uso de módulos controladores baseados em sistemas inteligentes artificiais, utilizando as arquiteturas aplicáveis a agentes (Lak98). Técnicas de controle advindas da inteligência computacional, tanto simbólica como conexionalista, estão sendo utilizadas nas três sub-tarefas de controle (mapeamento, planejamento e execução). Estes módulos recebem os dados provenientes dos sensores, podendo, a partir dos dados sensoriados: montar mapas internos de representação, fornecer ações e comportamentos a serem executados, modificar-se conforme o ambiente entre outros. Controladores que utilizam redes neurais (Lak98), sistemas fuzzy (Les04), sistemas especialistas (Ben96) e algoritmos genéticos (Koz94), estão sendo utilizados como módulos de controle de comportamento. A seguir serão vistos mais detalhadamente cada um destes módulos.

## Módulos com sistemas especialistas baseados em regras

Neste tipo de módulo de controle, as sinais provenientes dos sensores são analisados por sistemas especialistas (Vee95). Estes sistemas são formados por tabelas ou regras que tentam mapear as possíveis situações que possam vir a ser apresentadas ao robô. Leszek (Les04) propõe um sistema de interpretação que, ao receber os dados dos sensores, compara-os com uma série de regras e, baseado nestas, toma decisões. Esses sistemas são previamente definidos por um especialista que procura cercar as situações possíveis de acontecer. Desta forma, a robustez do sistema está diretamente relacionada com o número de regras. A medida que o número de regras aumenta, tornam-se mais difíceis o gerenciamento e os testes. Os sistemas baseados em regras possuem um bom desempenho para manipulações simbólicas, o que não é o caso da classificação dos dados sensoriados, onde existe a necessidade de transformação de dados numéricos em símbolos. Um outro problema é a dificuldade de, uma vez definido o sistema, expandi-lo para um número maior de regras, a fim de cobrir novas situações, a dependência de dados fidedignos também é grande neste tipo de tratamento.

## Módulo por sistemas fuzzy

Módulos de controle por sistemas *fuzzy* utilizam a lógica *fuzzy* para a interpretação dos sinais recebidos dos sensores. A lógica *fuzzy* baseada na teoria de conjuntos e na teoria das possibilidades. Na teoria de conjuntos clássicos, um elemento pertence ou não a determinado conjunto (Zad84). Na teoria dos conjuntos *fuzzy*, um elemento pode pertencer parcialmente a um determinado conjunto, ou seja, existe um determinado grau de possibilidade de o elemento pertencer ao conjunto. Isso pode ser ilustrado pelo gráfico na figura 2.8

Se fosse desejado categorizar a distância de um obstáculo ao robô da figura 2.8 em noções difusas de distância, seria possível dizer que: um obstáculo à distância de 38 metros do robô encontrasse-se a uma distância "um pouco" perto e "um pouco" média. A um obstáculo a 10 metros, estaria totalmente perto. Tal sentença traduz a noção que o ser humano tem das grandezas físicas (Zad84). A mente humana não sabe exatamente o quanto a água está fria ou quanto está morna, porém consegue tomar decisões baseadas nessa noção difusa das grandezas. A abordagem *fuzzy* bastante simbólica, por tratar com noções qualitativas do universo de tratamento.

O controle *fuzzy* está baseado na idéia difusa das grandezas. O controlador recebe os valores advindo dos sensores e, então, os fuzzyfica, ou seja, ele determina qual o grau de possibilidade da medida pertencer a cada um dos conjuntos *fuzzy* de entrada, neste caso representando os possíveis estados do

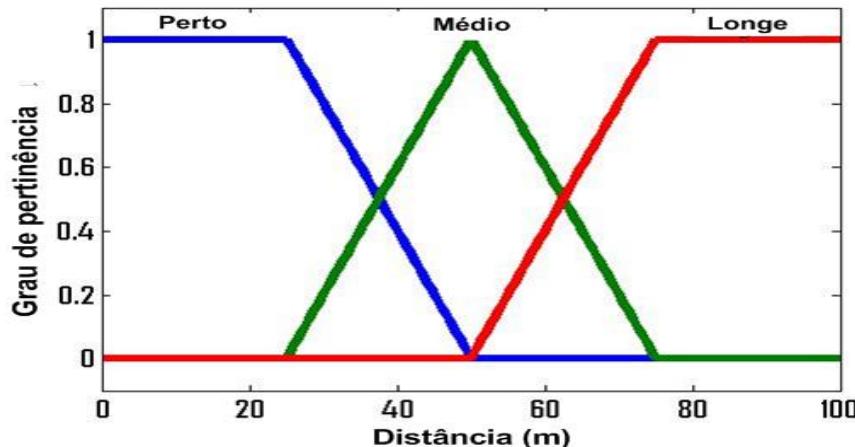


Figura 2.8: Conjuntos *fuzzy* - A variável distância apresenta três conjuntos *fuzzy* que representam os três conceitos lingüísticos: perto, médio, longe

robô em relação ao meio (por exemplo, perto de obstáculos, longe do alvo entre outros). Feito isso, consulta-se a base de regras e toma-se uma decisão. A base de regras um conjunto de proposições do tipo SE..ENTÃO... Por exemplo: SE perto ENTÃO desvie, onde a saída um comportamento do robô. Uma vez avaliadas todas as regras, realiza-se uma espécie de ponderação das mesmas, de modo a se obter um comportamento final da saída (Zad84).

Da mesma forma que o módulo de controle por sistemas especialistas, o controle por *fuzzy* composto por regras do tipo SE/ENTO, as quais são facilmente entendidas se comparadas com equações matemáticas. A vantagem que, devido às suas funções, esses sistemas permitem uma interface entre o nível simbólico (das regras) e o nível de sinal (dos sensores), realizando perfeitamente o casamento entre os dois. Ressalta-se o fato de que, nos sistemas *fuzzy*, o elemento a ser tratado pertence parcialmente a um conjunto de regras, havendo trânsito gradual entre os diversos tipos, o que não ocorre nas regras heurísticas (Les04).

Infelizmente, esse tipo de estrutura em forma de regras dificulta modificação do sistema e consequente adaptação a novas situações, que seria uma das exigências básicas para o funcionamento dos sistemas ditos autônomos. Outra dificuldade é a extração de regras dos conjuntos de dados a ser realizada por um operador humano. Em decorrência desses fatos, estuda-se sua utilização em combinação com outros sistemas mais flexíveis.

### Módulo por redes neurais artificiais

Redes neurais artificiais (RNAs) foram criadas na tentativa de obter-se um modelo que descrevesse o funcionamento do cérebro. RNAs são formadas,

em sua maioria, por elementos no lineares, altamente conectados, denominados de neurônios artificiais (Ben96). As redes neurais apresentam a habilidade de aprender com a experiência. O aprendizado se dá através da modificação do valor das conexões entre os neurônios (pesos).

Módulos de controle por RNAs utilizam-nas para a interpretação dos dados provenientes dos sensores e geração de alguma ação nos atuadores. Esta dissertação utiliza várias características que as redes neurais possuem, e que são requeridas pelos princípios de autonomia em ambientes reais. Algumas dessas propriedades são as seguintes: flexibilidade e generalização, tolerância a falhas em componentes físicos, facilidade de convivência com a micro-estrutura do robô (podendo modelar sua própria estrutura de maneira a explorar as melhores características senso-motoras do robô)(Fre92), tolerância a ruído, natureza paralela (veloz para aplicações em tempo real)(Vee95), e se no forem impostos limites arquitectura da rede (conexões e funções de transferência), ter-se- um dispositivo com grande potencial para o tratamento de estruturas temporais e mapeamentos complexos.

A utilização de redes neurais para controle de RNAs pode ser executada através do desenvolvimento de redes cujo aprendizado (e consequente modificação dos valores dos pesos) acontece de forma que a rede aprenda a tomar atitudes corretas na presença de determinadas situações sensoriais verificadas.

A rede recebe como entrada os valores providos dos sensores e fornece como saída um determinado comportamento a ser executado pelo sistema de acionamento do robô. Essas redes podem ser treinadas anteriormente (supervisionadas) ou podem dinamicamente sofrer modificações (não-supervisionado)(Fre92).

Como desvantagem do uso de redes neurais para controle de robôs está o fato de que o mapeamento das características e a classificação dos sinais, internamente na rede, não são visíveis e são de difícil entendimento. Assim, não se sabe de que forma a rede está armazenado determinado conhecimento, nem em que local isso é feito (Fre92).

Outro problema existente quanto convergência de aprendizado, o que algumas vezes pode ser muito lento, prejudicando assim o desempenho do robô. Além disso, em alguns casos, o próprio aprendizado não é garantido, o que inicializa a sua posterior utilização.

### **Support vetor machine**

"support vetor machine"(SVM)é um procedimento construtivo universal de aprendizagem baseado em "statistical learning theory". O termo universal significa que o SVM pode ser utilizado para o aprendizado de várias repre-

sentações como as redes neurais, as funções de base radial, "splines" e funções polinomiais. Atualmente é uma técnica usada com muitas vantagens sobre outros métodos de aprendizado de máquinas e com pesquisas em *SVM* baseados no algoritmo *Simultaneous Localization and Mapping (SLAM)* que permite a um robô autônomo móbil navegar num ambiente dinâmico ou estático (Jia07). Fazendo uma comparação com redes neurônicas, o *SVM* tem funções kernel que mapeiam um espaço dimensional muito grande, o espaço de busca só tem um único mínimo local, o treinamento e a classificação é extremosamente eficiente fornecendo a precisão e a robustez.

### Módulo de controle por algoritmos genéticos

Algoritmos genéticos (AGs) são algoritmos de procura baseados em mecanismos de seleção natural e genética natural (Ash06). Ao contrário das redes neurais, nas quais a busca de soluções pode ser obtida através do uso de funções matemáticas de minimização de erros (Les04), os algoritmos genéticos combinam a capacidade de sobrevivência de estruturas entre suas cadeias, onde as melhores estruturas são trocadas randomicamente, gerando novas estruturas e gerando um algoritmo de busca inovador. Os AGs eficientemente exploram informações históricas para especular e gerar novo ponto de busca com a expectativa de aumento do desempenho dos sistemas.

Módulos de controle por algoritmos genéticos podem ser utilizados individualmente onde o controlador recebe os dados provenientes dos sensores, gerando uma população de possíveis ações que serão avaliadas e reproduzidas, resultando numa ação final de execução. O algoritmo-base deste tipo de módulo pode ser considerado como um tipo de tentativa-e-erro [Fuk94], o que pode levar o sistema a encontrar um mínimo global e no local, como em outros métodos. Por esta razão, algoritmos genéticos são uma potente ferramenta de otimização que pode ser utilizados por outros módulos de controle.

### Módulo de controle híbrido

Fukada (Fuk94) faz uma comparação entre os diversos tipos de módulos de controle inteligente ver tabela 2.2 . Como se pode notar, todos os métodos apresentam diferentes características. Com o objetivo de aliarem-se as vantagens presentes em cada abordagem, está-se voltando para o uso de módulos de controle híbridos. Como exemplo, podem-se usar redes neurônicas e sistemas *fuzzy* como pré-processadores de regras heurísticas, transformando dados numéricos em conjuntos de dados simbólicos a serem manipulados.

Tabela 2.2: Comparação entre controladores: FO-Forte, ME-Médio, RA-Razoável e FR-Fraco

1	MM	AP	TR	RC	NL	OT
Locais Conhecidos	FO	FR	RA	FR	FR	FR
Redes Neurais	FR	FO	FO	FR	FO	ME
Fuzzy	ME	FR	FO	RA	FO	FR
Sistemas especialistas	RA	FR	FR	FO	RA	FR

Onde: MM - Modelamento Matemático, AP - Aprendizado, TR - Tempo Real, RC - Representação do conhecimento, NL - No Linearidade, OT - Otimização.

Outra aplicação decorre de sistemas que possuem um elevado número de parâmetros de entrada, o que torna difícil para um operador humano determinar o completo conjunto de regras de descrição do sistema. Pode-se aplicar redes neurônicas, pois essas não exigem tais procedimentos, uma vez que apenas criam relações entre pares entra/sada (Fuk94). Combinações de sistemas *fuzzy* com redes neurônicas estão sendo desenvolvidas. Redes *fuzzy* neurônicas possuem o seu conhecimento estruturado e fornecido por especialistas através de funções de pertinência, ao mesmo tempo em que essas funções são modificadas por processos de aprendizado (Koz94). Algoritmos genéticos estão sendo utilizados para otimizar topologias de redes neurais, propiciando a estas possibilidades de evolução (conformar discutido anteriormente). Também a capacidade de manipulação de símbolos apresentada pelos algoritmos genéticos pode produzir novas regras ou conhecimento para sistemas especialistas.

#### 2.4.3

#### Técnicas por Visão Computacional

Correspondência de imagens é fundamental em diversos problemas de visão computacional como reconhecimento de objetos, reconhecimento de cenas, montagem automática de mosaicos, obtenção da estrutura 3D de múltiplas imagens, correspondência estéreo e perseguição de movimentos. Uma abordagem para se trabalhar com correspondência de imagens é se usar descritores locais para se representar uma imagem. Descritores são vetores de características de uma imagem ou de determinadas regiões de uma imagem e podem ser usados para se comparar regiões em imagens diferentes. Este vetor de características é normalmente formado em imagens diferentes. Este vetor de características é normalmente formado por descritores locais ou globais. Descritores locais computados em pontos de interesse provaram ser bem sucedido em aplicações como correspondência e reconhecimento de imagens (Mik03).

Descritores são distintos, robustos à oclusão e não requerem segmentação. Existem diversas técnicas para se descrever regiões locais em uma imagem (Mik03). O mais simples descritor é um vetor com as intensidades dos pixels da imagem. A medida de correlação cruzada pode ser então usada para computar a similaridade entre duas regiões. Porém, a alta dimensionalidade de tal descritor aumenta a complexidade computacional da comparação. Então, esta técnica é principalmente usada para se encontrar correspondências ponto a ponto entre duas imagens. A vizinhança de um ponto também pode ser escalada de modo a reduzir sua dimensão. Outro descritor simples é a distribuição de intensidades de uma região representada por seu histograma.

Trabalhos recentes têm se concentrado em fazer descritores invariáveis a transformações nas imagens. Mikolajczyk e Schimid (Mik04) propuseram um detector de pontos de interesse invariável a transformações afins através da combinação de um detector invariável à escala e da técnica "*second moment of Harris corners*" (Har88). Ling e Jacobs (lin05) propuseram um sistema para a construção de descritores de intensidade locais invariáveis a deformações em geral. Lowe (Low99) propôs uma maneira rápida e eficiente de computar características invariáveis a transformações em escala, que medem a distribuição do gradiente em regiões detectadas invariáveis à escala.

### Transformação *SIFT*

*SIFT* (*Scale Invariant Feature Transform*) é uma técnica de processamento de imagens que permite a detecção e extração de descritores locais, favoravelmente invariáveis a mudanças de iluminação, ruído de imagem, rotação, escala e pequenas mudanças de perspectiva. Estes descritores podem ser utilizados para se fazer a correspondência de diferentes visões de um objeto ou cena. Descritores obtidos com a técnica *SIFT* são altamente distintos, ou seja, um determinado ponto pode ser corretamente encontrado com alta probabilidade em um banco de dados extenso com descritores para diversas imagens. Um aspecto importante da técnica *SIFT* é a geração de um número grande de descritores que conseguem cobrir densamente uma imagem quanto a escalas e localizações. A quantidade de descritores é particularmente importante para o reconhecimento de objeto, onde a capacidade de se encontrar pequenos objetos em ambientes desordenados requer ao menos 3 pontos encontrados em comum para uma identificação confiável. A obtenção de descritores *SIFT* é feita através das seguintes etapas:

- Detecção de extremos: Nesta primeira etapa é feita procura para todas escalas e localizações de uma imagem. Isto é feito utilizando-se a dife-

ença de filtros gaussianos de modo a se identificar pontos de interesse invariáveis à escala e rotação.

- Localização de pontos chave: Para cada localização em que foi detectado um extremo, um modelo detalhado é ajustado de modo a se determinar localização e escala. Pontos chaves, ou pontos de interesse, são então selecionados baseando-se em medidas de estabilidade.
- Definição de orientação: É definida a orientação de cada ponto chave através dos gradientes locais da imagem. Toda operação a partir de então será feita com relação a dados da imagem transformados em relação à orientação, escala e localização de cada ponto chave. Desta maneira se obtém invariância a estas transformações
- Descritor dos pontos chaves: Nesta etapa é feita a construção dos descritores ao se medir Gradientes locais em uma região vizinha a cada ponto de interesse. Estas medidas são então transformadas para uma representação que permite níveis significativos de distorção e mudança na iluminação.

Em tarefas de comparação de imagens e reconhecimento, descritores *SIFT* são extraídos das imagens para então poderem ser comparados.

### Transformada Hough

A transformada *hough* (?) é uma técnica de extração de características chaves numa imagem, sim embargo pode ser extensiva para a identificação de posições de formas diversas. A transformada *hough* pode ser entendida de modo genérico como uma tabela de parâmetros que descrevem um modelo, a tabela seria preenchida para cada dado de um conjunto de dados apresentados, encontrando todos os modelos possíveis que coincidissem com cada ponto e atualizando a tabela, aumentando às células referentes aos possíveis parâmetros dos modelos encontrados.

A idéia básica da transformada *hough* se ilustra na Figura 2.9, onde uma linha é um conjunto de pontos  $(u, v)$  tal que 2-30:

$$u \times \cos \theta + v \times \sin \theta = d \quad (2-30)$$

Para algum ponto  $(u, v)$ , existe uma família de parâmetros de linhas através deste ponto, dado pela equação 2-30. Cada ponto obtém um voto por cada linha na família; se há uma linha que tem muitos votos, então isso deveria ser a linha que passa através dos pontos. Com esta técnica é possível encontrar nas imagens objetos como linhas, círculos o elipses.

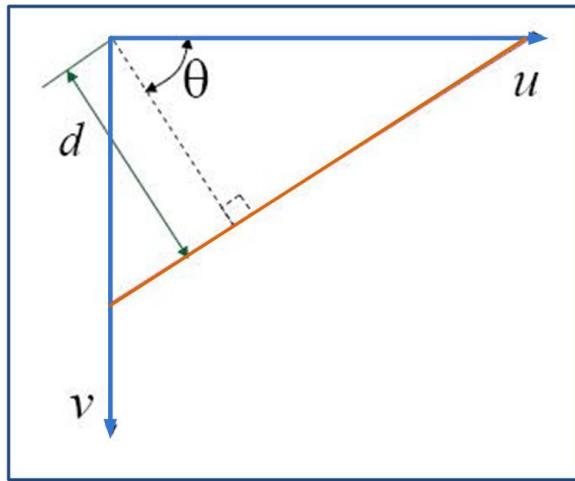


Figura 2.9: Idéia básica da Transformada Hough

### Transformada RANSAC

O algoritmo *RANSAC* (*Random Sample Consensus Algorithm*), é apresentado em (Fis81)(Lac00), como um método para estimar os parâmetros de um modelo para um conjunto de dados conhecidos mas com presença de diversos dados errôneos.

O algoritmo não é aplicado independentemente porque seu eficiência é limitada quando há um grande número de dados, não conseguindo uma boa estimação do modelo. Por isso é sugerido que se aplique anteriormente Hough, o outra técnica conseguindo gerar um conjunto mais robusto que o inicial.

*RANSAC* é um algoritmo bem simples definido como segue. Dado um modelo com parâmetros  $\vec{x}$  se deseja estimar-os. Para tal, é assumido:

- Os parâmetros podem ser estimados a partir de um número  $N$  de ítems num conjunto de dados conhecidos.
- A probabilidade de um dado selecionado aleatoriamente para ser parte de um bom modelo é dada por  $p_g$ .
- A probabilidade de que o algoritmo termine sim que se encontre um bom modelo é dada por  $p_{falla}$ .

O algoritmo é então executado através das seguintes etapas:

1.  $N$  ítems são escolhidos de modo aleatório.
2. A partir dos ítems escolhidos,  $\vec{x}$  é estimado.
3. Logo encontra-se o número de ítems que encaixam ao modelo para determinada tolerância especificada. Este número é chamado de  $K$ .

4. Caso  $K$  seja grande ou suficiente, para um limite escolhido, o algoritmo termina com sucesso.
5. O algoritmo é repetido de 1 a 4 um número  $L$  de vezes.
6. Caso o Algoritmo não tenha terminado depois de  $L$  tentativas, o algoritmo falho.

## 2.5

### Planejamento da Trajetória

O planejamento da trajetória *path planning*, já seja global ou local, consiste em encontrar uma rota segura capaz de levar ao veículo desde a posição atual até a especificada do destino. O conceito de rota segura implica o cálculo de um caminho ao menos continuo na posição, que seja livre de obstáculos. Em virtude desta rota, o gerador constituirá as referencias que entregam-se ao controle de movimento.

Existem vários métodos de planificação, todos eles se fundamentam em uma primeira fase de construção de algum tipo de grafo sobre o espaço livre, segundo a informação adquirida do entorno, para posteriormente usar um algoritmo de busca em grafos que encontre o caminho ótimo segundo à função de custo, entre eles temos os seguintes:

- Planificação baseada em grafos de visibilidade.
- Planificação baseada em diagramas de Voronoi.
- Planificação baseada em modelado do espaço livre.
- Planificação baseada na descomposição de células.
- Planificação baseada em campos potenciais.

A fundamentação teoria apresentada neste capítulo será utilizada na proposta de uma metodologia para auto localização e mapeamento de um robô móvel. no próximo capítulo, o ambiente de desenvolvimento é descrito, incluindo o robô móvel utilizado e seus sensores e atuadores.