

Capítulo 1 – Introdução à robótica móvel – Locomoção

Robôs móveis: São uma colecção de subsistemas. **Locomoção:** Como é que o robô se move no seu ambiente. É o processo que faz com que o robô se mova pela aplicação de forças. **Percepção:** Como é que o robô se apercebe do ambiente e de si próprio. **Controlo:** Como é que o robô transforma a percepção em acções. **Comunicação:** Como é que os robôs comunicam entre si e com operadores externos. **Dinâmica:** Estuda o movimento provocado por essas forças. **Cinemática:** Estuda a matemática do movimento sem ter em conta as forças que o provocam. Estuda as relações geométricas que governam o sistema e as relações entre parâmetros de controlo e o comportamento do sistema. **Rodas:** A roda gira segundo o eixo X fazendo o robô deslocar-se segundo o eixo Y. Deslizamentos laterais segundo o eixo X, fazem com que o robô se mova segundo a linha Z. **Características da roda ideal:** 1. O robô é constituído por mecanismos rígidos. 2. Não há deslizamento na direcção ortogonal do deslocamento. 3. Não há deslizamento entre a roda e o chão. 4. Cada roda do robô tem no máximo um eixo de direcção. 5. Todos os eixos de rotação são perpendiculares ao chão. **Parâmetros das rodas:** Para velocidades pequenas o modelo da roda ideal pode ser usada para determinar os modelos cinemáticos dos robôs móveis com rodas. Parâmetros das rodas: r – raio da roda; v – Velocidade linear da roda; w – velocidade angular da roda; t – velocidade direcciona da roda. **Tipos de rodas:** fixa; centrada, direcciona não centrada, omnidirecciona. **Centro de rotação instantâneo:** Para que um robô se mova como resultado do girar das suas rodas, tem que existir um ponto em relação ao qual cada roda do robô siga um movimento circular. Esse ponto chama-se *centro de rotação instantâneo* ou *centro de curvatura instantâneo* e corresponde ao ponto de intercepção dos eixos de rotação de todas as rodas. O robô pode alterar o seu CRI, alterando a orientação individual das rodas em relação aos seus eixos verticais. Existem vários mecanismos para provocar essa alteração.

Veículos não-holonómicos: Um veículo que se move no plano têm 3 graus de liberdade: uma posição (x, y) e uma orientação θ . Estes 3 elementos (x, y, θ) definem a *pose* ou *postura* do veículo. No entanto, os veículos normalmente não têm controlo total e independente sobre os três parâmetros e têm que executar manobras complexas para atingir determinada pose que dependem do ambiente e da configuração do próprio veículo. Veja-se por exemplo o caso do estacionamento de um carro. Um veículo que se pode mover em algumas direcções, mas não noutras, é um veículo *não-holonómico*.

Controlo diferencial de direcção: Usa dois motores; É fácil de manobrar; É fácil de construir; É fácil de controlar; É sensível a erros de velocidades das rodas (difícil fazer o robô andar em linha recta). **Análise cinemática:** $VD = \omega * (R + L/2)$; $VE = \omega * (R - L/2)$; $R = (L/2) * (VE + VD) / (VD + VE)$; $\omega = (VD - VE) / L$. Se $VD = VE$, R é infinito e o robô move-se em linha recta. Se $VD = -VE$, R é zero e o robô gira sobre o ponto intermédio do eixo que une as duas rodas. Para outros valores de VD e VE , o robô move-se numa trajetória circular em relação ao ponto CRI.

Triciclo: usa dois motores; não é tão fácil de manobrar; é fácil de construir; é fácil de controlar. **Análise cinemática:** A roda faz um ângulo α com a linha de direcção. O veículo roda com uma velocidade angular ω em torno do ponto CRI à distancia R do ponto intermédio do eixo que une as duas rodas. R e ω são dados por: $R = D * \tan(\pi/2 - \alpha)$; $\omega = V / (D/2 + R)$; Onde D é a distância do eixo traseiro à roda dianteira e V é a velocidade linear dessa roda.

Controlo síncrono de direcção: usa dois motores no mínimo; é fácil de manobrar; é difícil de construir; é fácil de controlar. Um sistema complexo de correias faz com que todas as rodas giram em uníssono e rodem à mesma velocidade apontando na mesma direcção. O controlo é feito escolhendo a direcção em que as rodas devem apontar e a velocidade a que devem rodar. O controlo síncrono dá ao robô um comportamento holonómico. **Análise cinemática:** O CRI está sempre no infinito e os parâmetros de controlo ω e V são controlados independentemente.

Controlo omnidirecciona: usa três motores; é tão fácil de manobrar; é difícil de construir; não é tão fácil de controlar. **Análise cinemática:** r - wheel radius F_0, F_1, F_2 - unit direction vectors; v - desired body velocity expressed in body coordinate frame; w - angular velocity; b - wheel baseline; v_0, v_1, v_2 - wheel linear velocities; $\omega_0, \omega_1, \omega_2$ - wheel angular velocities; n - wheel number; p_n - velocity of the body at a given wheel n ; $F_0 = [-1, 0]$; $F_1 = [1/2, -\sqrt{3}/2]$; $F_2 = [1/2, \sqrt{3}/2]$; $\omega_0 = (v \cdot F_0 + b \cdot \omega) / r$; $\omega_1 = (v \cdot F_1 + b \cdot \omega) / r$; $\omega_2 = (v \cdot F_2 + b \cdot \omega) / r$

Controlo Ackerman: usa dois motores; Não é tão fácil de manobrar; Não é tão fácil de construir; é fácil de controlar. **Análise cinemática:** O ângulo de direcção da roda interior dianteira é um pouco maior do que o da roda exterior dianteira, assim como a rotação da roda interior traseira é um pouco menor do que a da roda exterior traseira (para prevenir derrapagem).

Capítulo 2 – Motores

Princípio de funcionamento de um servomotor: os servomotores são motores cc que giram apenas para determinar a posição (são formado por: posição desejada, circuito eléctrico de controlo (sinal de erro), motor cc,

engrenagem redutora, sensor de posição e feedback). O controlo electrónico é feito por PWM. o sinal de controle é enviado ao servo, indica a posição desejada. este sinal é comparado com o sinal feedback que indica a posição actual medida pelo sensor de posição. É gerado um sinal de erro que é usado para ajustar a posição do motor. Um servomotor usa o seu eixo (e roda) e em sentido horário e autónomo, mediante um ângulo tipicamente de 180°.

Capítulo 3 – Sensores

Tarefas de um robô: 1. obter representações do seu estado global (o estado interno dos seus sistemas; conhecimento sobre o ambiente envolvente); 2. elaborar decisões sobre a actuação a fazer em função do seu estado global e do programa de controlo; 3. actuar sobre o sistema: activar actuadores em conformidade com determinadas leis de controlo ou de comportamento dos actuadores

Transdutores: é todo o dispositivo que transforma uma forma de energia noutra forma de energia, isto é, converte um sinal de entrada, num sinal de saída de outro tipo. Os transdutores podem ser de dois tipos, sensores e actuadores.

Actuadores: é qualquer dispositivo eléctrico, mecânico, pneumático ou hidráulico, com a função de alterar e/ou manter a posição de um elemento do robô que executa determinada acção (por exemplo os eixos das rodas). Os actuadores respondem a sinais enviados pelo sistema de controlo do robô. Os actuadores são usados pelo robô para a geração de acção, nomeadamente a geração de movimento de elementos.

Percepção: consiste na extracção de propriedades a partir da aplicação de métodos matemáticos ou outros, sobre as informações vindas do(s) sensor(es) para obter uma conclusão, avaliação de estado, ou caracterizar uma propriedade do ambiente ou do sistema.

Sensores: é qualquer dispositivo que permite ao robô aperceber-se do meio ambiente que o rodeia e do seu próprio estado de funcionamento. Genericamente convertem sinais captados no meio ambiente, em sinais eléctricos que são enviados para o sistema de controlo do robô para serem processados. Os robôs usam os sensores para aquisição de dados e para auto-controlo. Os sensores podem ser classificados em duas categorias: sensores passivos e sensores activos. Os sensores activos também chamados de transceivers, são sensores que baseiam o seu funcionamento na emissão e recepção de sinais (Ex.: sonar, IV, laser). **Categorias de sensores:** Os sensores usados em robótica são peças chave nos actuais tópicos de investigação: construção de mapas, localização, planeamento de trajectórias, detecção de obstáculos, etc. Podem ser divididos nas categorias: • posicionamento e medição de distâncias (São usados para determinar a posição de robôs móveis no meio ambiente em que se encontram; sensores de distância (sonar, IV, laser), bússola digital, GPS, etc.); • detecção de movimento (detectam tipos de movimento como translação e rotação (encoders), velocidade rotacional e longitudinal (velocímetros), aceleração (acelerómetros), etc.); • visão (usados para a vigilância do ambiente onde se encontra o robô. Detectam tipos de movimento como translação e rotação (encoders), velocidade rotacional e longitudinal (velocímetros), aceleração (acelerómetros), etc.); • inspiração biológica (simulam os sentidos dos órgãos sensoriais biológicos; olfacto (nariz electrónico), tacto (sensores tácteis, pele artificial), audição (analisador de som) e visão (visãorobótica)); • ambiente (são usados para explorar o ambiente onde se encontra o robô e detectar fenómenos físicos tais como temperatura, pressão do ar, radiação, humidade, campos magnéticos, etc).

Sensores de posicionamento

► **Bússola digital:** permite determinar a orientação absoluta em relação ao campo magnético terrestre.

• **Ligação ao BasicX-24:** possibilidade de ligação em rede. O esquema de endereçamento usado permite ligar um máximo de 65000 nós em rede; a capacidade pode ter várias task a executar ao mesmo tempo, inclui conversor analógico digital de 8 canais (correspondem às portas 13 a 20) e de 10 bits de resolução. Todos os canais podem ser usados tanto como input analógico ou digital. o input analógico deve estar entre 0v e 5v. é aconselhável que a ligação terra da fonte de voltagem que se está a medir, seja ligada à terra do BX-24.

► **IR Beacon:** quando usados aos pares permitem determinar a orientação de um deles em relação ao outro. Cada dispositivo tem 4 emissores e 4 receptores de IV estrategicamente posicionados de modo a definirem 4 direcções (N, S, E, W). Os dispositivos estão constantemente a alternar o seu funcionamento entre emissão e recepção do sinal de IV. Assim, quando um dispositivo muda a sua orientação em relação ao outro, o sensor que estava na direcção do outro dispositivo, deixa de receber o sinal de IV e o sensor que fica na direcção do outro dispositivo passa a receber o sinal. A orientação de um dispositivo em relação ao outro é assim determinada pelo sensor que estiver a receber o sinal em dado momento.

Sensores de medição de distâncias: • **Princípios mais comuns de medição:** - **Tempo de voo:** conhecida a velocidade de propagação de um feixe emitido pelo sensor e medido o tempo que o eco deste leva a retornar ao sensor depois de embater num objecto, a distância do sensor ao objecto pode ser determinada pela expressão: $d = \frac{1}{2} v \times Pt$; - **Triangulação:** Conhecido o ângulo que um feixe emitido pelo sensor faz com o eco deste que retorna ao sensor depois de embater num objecto, a distância do sensor ao objecto pode ser determinada pela expressão: $d = \tan Q \times A$. Sendo A a distância entre o emissor e o receptor do feixe. Ex.: • Infravermelhos; • Sonar (ultra-som); • Laser; • Radar.

► **Medição de distâncias com sonar:** medem a distância usando o princípio do tempo de voo. Um circuito transmissor emite sinais acústicos com uma frequência típica de 50kHz (ultra-sons). Estes sinais ao chocarem com objectos provocam ecos que são captados por um circuito receptor. A distância a que se encontra o objecto é proporcional ao tempo que decorre entre a transmissão do sinal e a recepção do eco correspondente. Este tipo de sensor acústico é mais preciso do que o sensor de IV. Existem no mercado vários exemplares de diferentes marcas.

• **Sonar:** Vantagens: • Baixo custo; • Leves e pequenos; • Baixo consumo energético. Desvantagens: • Atenuação do sinal; • Reflexão especular do sinal; • Cross talk; • Limites mínimos e máximos na medição; • Efeito das condições atmosféricas; • Baixa velocidade do som (ex.: Sonar SRF04)

► **Medição de distância com laser:** segue o mesmo princípio de funcionamento do sonar, mas usa um feixe de luz laser em vez de som ultra-sónico. Tipicamente o feixe gira num plano horizontal permitindo obter uma fatia 2D do ambiente. • **Laser:** Vantagens: • Velocidade superior (3×10^8 m/s); • Grande precisão (10 mm); • Resolução angular superior. Desvantagens: • Limitado ao plano 2D; • Consumo de energia elevado; • Custo elevado; • Materiais indetectáveis.

► **Medição de distâncias com infravermelhos:** medem a distância usando o princípio da triangulação. Um circuito transmissor emite um feixe de infravermelhos. Este feixe é reflectido por um objecto e um circuito receptor capta-o determinando o ângulo de reflexão. A distância a que se encontra o objecto é proporcional a este ângulo de reflexão. Existem no mercado vários exemplares de diferentes marcas. • **Infravermelhos:** Vantagens: • Baixo custo; • Mais rápidos do que o sonar. Desvantagens: • Menos precisos que o sonar; • Materiais indetectáveis; • Gama de distâncias limitadas. (ex.: Sharp GP2D12).

Sensores de detecção de movimento: • **Codificador absoluto:** gera um padrão linear de bits representativo de uma posição angular de um eixo; • **Codificador incremental (encoder):** gera um impulso por cada incremento de um certo valor conhecido no deslocamento angular de um eixo. Podem ser usados para determinar a posição e/ou a velocidade de eixos. Usados também para indicar o número total de voltas de eixos (odómetros). Podem ter resoluções muito altas, não necessitam de contacto mecânico, permitem repetir resultados e são fáceis de construir de forma caseira. São relativamente caros.

► **Sensores de velocidade:** a velocidade medida é normalmente a velocidade angular, isto porque a maioria dos sistemas baseiam-se em elementos de rotação. Há ainda os sistemas que, usando informação temporal de relógio, usam informação sensorial de posição e derivam a velocidade – caso dos codificadores incrementais; - **Taquímetro:** dispositivo que gera uma tensão eléctrica proporcional à velocidade de rotação. Acopla-se ao veio em rotação (apropriado para grandes velocidades de rotação, i.e., dezenas de rotações por segundo); - **Giroscópios:** dispositivo que indica a velocidade angular; fica solidário com o sistema em rotação (apropriado para baixas velocidades angulares, i.e., menores que 100 °/s).

Sensores de inspiração biológica

• **Sensores de contacto (bumpers e whiskers):** os bumpers consistem normalmente numa série de inúmeros interruptores que se abrem ou fecham por contacto e estão recobertos de um material amortecedor. É muito fácil construir este tipo de sensores de uma forma caseira, onde o limite é a nossa imaginação. São baratos, simples de usar e robustos. Os whiskers (bigodes de gato) são muitas vezes usados para navegação efectiva e são filamentos semi-rígidos que activam um contacto eléctrico quando tocam em algo.

• **Sensores de som:** Descodificador de tom

• **Sensores de cor:** Sensor de cor TCS230

• **Sensor de visão robótica (ex.: CMUCam) – Características:** A 17 frames por segundo, a CMUCam é capaz de: • Identificar a posição e tamanho de um objecto; • Medir o histograma RGB ou YUV de uma região da imagem; • Identificar e seguir automaticamente o primeiro objecto visível; • Seguir automaticamente um objecto através de um servomotor ligado directamente; • Enviar a imagem capturada através da porta série; • Enviar um bitmap com a forma do objecto detectado, através da porta série.

Sensores de ambiente

► **Sensor de radiação infravermelha:** na sua forma mais simples detectam a proximidade de um objecto através da radiação reflectiva. Podem usar triangulação para determinar a distância a que se encontra o objecto. São baratos e simples de usar. Alguns modelos usam modelação de frequência para se tornarem imunes a fontes de radiação infravermelha do meio ambiente. Fototransistor como sensor da chama da vela A chama da vela é uma fonte de radiação infravermelha. • **Modelação do sinal de infravermelhos:** normalmente a luz IV emitida pelo emissor é modelada a uma determinada frequência e o receptor apenas responde à luz recebida nessa mesma frequência. O circuito ao lado pode ser construído com o emissor de IV D1 que emita luz com um comprimento de onda de 980nm, por exemplo. O receptor T1 deverá ter um filtro óptico que deixe passar luz apenas com esse mesmo comprimento de onda e terá também um filtro electrónico que permita apenas passar luz modelada a 38kHz. Por outras palavras, o receptor só responde a luz infravermelha que pisque 38.000 vezes por segundo. Este método evita interferências provocadas por fontes externas de IV como o sol (0Hz) ou as luzes das casas (120 Hz). Os sensores IV podem dar leituras erradas devido à interferência de fontes de infravermelhos presentes no ambiente.

► **Sensor de linha branca**

► **Sensor de radiação ultravioleta (ex.: UV-Tron)**

► **Sensor piroelétrico:** sensíveis ao calor em forma de radiação infravermelha. São úteis para detectar objectos emissores de calor. São baratos, simples de usar e robustos, mas normalmente é necessário haver movimento para detecção contínua (ex.: Eltek 442-3 IR-EYE).

Circuitos úteis

► **Multiplexer:** quando o número de entradas do controlador não chegar podemos usar um multiplexer como o 74150 para conseguir mais entradas. O 74150 permite obter 16 entradas usando apenas 4 portas do controlador.

► **Window comparator:** útil para interface com sensores de output analógico, quando não se dispõem de entradas analógicas no microcontrolador. O circuito activa saídas digitais que indicam ao microcontrolador quando a saída analógica do sensor é inferior ou superior a limites pré-definidos.

Fusão sensorial: Nos robôs com tarefas mais complexas, o uso individual de sensores pode não ser suficiente e ser: • Impreciso; • Incompleto; • Erróneo. Na tentativa de resolver estes problemas, os robôs actuais estão muitas vezes equipados com vários sensores homo e heterogéneos. Estes sensores recolhem diferentes informações sensoriais sobre o ambiente envolvente e o estado interno do robô. A combinação dessas informações com o objectivo de resolver os problemas mencionados anteriormente designa-se por fusão sensorial. Vantagens da fusão sensorial: robustez; aumento da precisão; redução de ambiguidade; aumento da confiança. Desvantagens da fusão sensorial: complexidade e instabilidade; sinais heterogéneos; sincronização; ruído; poder computacional. **A fusão sensorial poderá ser feita a partir de:** sensores diferentes; posições e tempos diferentes. **A informação sensorial poderá ser:** redundante (obtida a partir da mesma posição); complementaria (obtida de posições diferentes ou de sensores diferentes); incerta (os dados têm uma resolução espacial e precisão limitadas); incompleta (não chega a ser obtida). **Tipos de fusão sensorial:** • Complementária (fusão de dados sensoriais incompletos para criar um modelo mais completo); • Competitiva (fusão de dados incertos para diminuir a incerteza); • Cooperativa (os dados de um sensor dependem dos dados de outro sensor). **Procedimento:** os dados sensoriais são transformados para uma dada representação interna. Essa representação interna é comum a todos os sensores. A fusão sensorial é realizada nessa representação interna (função: competitiva, cooperativa – conversão e triangulação, complementaria - conversão). **Métodos de fusão sensorial:** são métodos matemáticos que tem em conta a incerteza da informação sensorial: • Kalman filtering; • Métodos bayesianos; • Redes neuronais.

Capítulo 4 – Localização

Representação do ambiente: a escolha correcta da representação interna do ambiente é um dos principais factores que influencia o sucesso de uma técnica de localização. O mapa do ambiente deve ser capaz de armazenar informações suficientes para que o robô se localize, mas não pode ser muito pesado a ponto de comprometer a performance do sistema de controlo.

Existem duas formas principais de representação de ambiente utilizadas pelas técnicas de localização:

- **Mapas geométricos:** representam os objectos de acordo com a sua posição geométrica absoluta. Este tipo de mapa pode ser poligonal (os obstáculos são representados por polígonos) ou baseado em grelhas (o valor de cada célula indica a presença de um obstáculo naquela posição).

- **Mapas topológicos:** são baseados nas relações geométricas observadas entre características do ambiente, ao invés de sua posição absoluta. Esta representação toma a forma de grafos, onde os nós são os pontos de referência (ou características) e os arcos indicam a relação entre estes pontos e como o robô pode mover-se entre eles. Os mapas topológicos podem ser construídos sem a necessidade de estimar a posição exacta do robô. Esta abordagem permite a integração de grandes mapas, minimizando o problema de erros sensoriais, pois as conexões entre os nós são relativas. O problema com este tipo de representação é o fato de que é necessário que o sistema seja capaz de reconhecer os pontos de referência e diferencia-los uns dos outros.

Localização: Um posicionamento correcto e uma navegação efectiva é uma pré-condição básica para uma aplicação robótica móvel de sucesso. Para poder utilizar um mapa do ambiente para executar os planos de navegação, o sistema de controlo do robô deve possuir alguma forma de localização. Localização é um dos problemas fundamentais na robótica móvel autónoma. **Localização Local:** Quando a posição inicial do robô é conhecida, a tarefa do algoritmo de localização é manter a posição conhecida com o menor erro possível, isto é chamado de localização local. Este tipo de localização necessita de recalibração quando a estimativa do erro excede um limite preestabelecido. **Localização Global:** É a habilidade de se estimar a posição do robô sem nenhum conhecimento de sua posição inicial e a habilidade do robô em se relocalizar quando este se perder no ambiente. **Localização Robusta:** É a habilidade de se estimar a posição do robô num ambiente dinâmico onde um número significativo de observações podem não ser encontradas no mapa. Por exemplo, a aparência de um ambiente pode mudar significativamente quando os móveis são trocados de posição, portas são abertas ou fechadas, e pessoas se locomovem no ambiente. **Localização Activa:** Para aumentar a eficiência da localização pode-se optar por eliminar as ambiguidades entre as possíveis posições de forma activa. Controlando de forma activa a movimentação do robô para que ele se locomova em direcção a objectos ou lugares que possam facilitar a quebra da ambiguidade.

Classificação dos métodos de localização a partir de medidas: • **relativas (Dead-Reckoning):** a posição actual é determinada com base na última posição conhecida e na medição do movimento do robô (localização baseada em odometria; navegação inercial); • **absolutas:** a posição actual é determinada com base em informação sobre pontos de referência do mundo exterior (localização: baseada em faróis; baseada em marcas ou atributos; topológica; scan Matching; Markov); • **mistas:** a posição actual é determinada recorrendo a métodos mistos de posicionamento relativo e posicionamento absoluto.

Localização baseada em odometria: determina a localização do robô de forma incremental através da contagem do deslocamento angular das rodas. É um sistemas totalmente autocontido (não depende de partes externas ao robô). Está no entanto sujeito a erros que pela natureza incremental do método, podem crescer muito depressa e de forma descontrolada. **Erros da odometria:** - **sistemáticos:** (característicos do robô e/ou dos sensores): rodas com diâmetros diferentes e não-alinhadas; valor médio do diâmetro diferente do eixo nominal; encoder de resolução finita e tempo de amostragem discreto. - **Não-sistemáticos** (São característicos da interacção do robô com o ambiente): movimento em terreno irregular; derrapagem das rodas (solo escorregadio; grandes acelerações; forças internas - rodas de apoio; forças externas - interacção com obstáculos; rodas sem apoio).

Localização baseada em faróis: são dispositivos que permitem obter informação sobre pontos do mundo exterior cuja localização é conhecida. Os faróis enviam essa informação ao robô por via óptica ou rádio, a qual é depois usada para determinar a posição do robô através: - **Trilateração:** a posição do robô é determinada com base na distância deste a um conjunto de faróis. Existem duas configurações típicas: 3 ou mais faróis em localizações conhecidas e um receptor no robô, ou um transmissor no robô e receptores em localizações conhecidas. O sistema GPS é um exemplo de um sistema que usa trilateração. - **Triangulação:** A posição (x_0 , y_0) e a orientação G_0 do robô são determinadas a partir dos ângulos G_1 , G_2 , e G_3 , que o sinal emitido por três faróis de coordenadas conhecidas, faz com o eixo de direcção do robô.

Localização baseada em marcas: as marcas são sinais que o robô detecta e que lhe permitem determinar a sua posição usando métodos semelhantes aos usados com os faróis, e/ou comparando as marcas com mapas do mundo. (adquirir informação sensorial; detectar e segmentar as marcas; determinar correspondência entre os dados sensoriais e o mapa do mundo; calcular a posição). Detectar e determinar correspondências, são as tarefas mais complexas. Após ter sido determinada a correspondência, podemos usar triangulação para calcular a posição do robô. - **Marcas naturais:** são objectos ou características do meio ambiente cuja função específica não é o auxilio na

navegação de robôs, mas que podem ser usados como pontos de referência caso sejam detectada pelo robô. - **Marcas Artificiais:** são objectos ou marcadores especialmente desenhados para serem usados como pontos de referência para a navegação do robô. A detecção é muito mais fácil neste caso porque as marcas são especialmente desenhadas para que isso aconteça. Para além disso as marcas são conhecidas à partida o que simplifica os algoritmos de detecção e correspondência.

Localização topológica: baseada na descrição topológica do ambiente. Este método é motivado pelas evidências da ciência cognitiva que indicam que os humanos dependem inicialmente de informações topológicas, ao invés de métricas, para navegação. O modelo topológico é organizado como um grafo, com os nós representando pontos de referência (chamados "distinctive places" – DPs), e os arcos representando rotas conectadas. Conhecimento procedural sobre como detectar os DPs e de como seguir as rotas descritas pelos arcos são incorporados no modelo e informações métricas são associadas com cada nó e arco para permitir uma precisão geométrica local. O robô utiliza esse grafo para se mover de forma eficiente de um DP para outro seguindo as rotas definidas pelos arcos. O método depende muito da noção de DPs. Um DP é definido como um ponto no ambiente que é distinguível localmente de acordo com alguns critérios geométricos. Isto é definido como "assinatura", um sub-conjunto de características que são maximizadas no DP, tais como: distância entre objectos, simetria do ambiente, entre outros.

Scan Matching: processo de transladar e rodar uma varredura sensorial de distância (obtida de sensores ultrasónicos, por exemplo) de forma que uma sobreposição entre a leitura dos sensores e um mapa a priori seja encontrada. A maioria dos métodos de scan matching assumem que a estimativa da posição inicial é muito próxima da posição real do robô para poder limitar o espaço de procura.

Localização de Markov: algoritmo probabilista: Ao invés de manter uma única hipótese sobre a posição do robô, a localização Markov mantém uma distribuição de probabilidade sobre um espaço de hipóteses. A representação probabilista permite que cada uma destas hipóteses seja avaliada de uma forma matemática. Localização de Markov usa uma representação explícita e discreta para a probabilidade de todas as posições no espaço de estados. Representa o ambiente por uma grelha ou um mapa topológico com um número finito de estados possíveis (posições). Durante cada actualização, a probabilidade para cada estado (elemento) do espaço todo é actualizado. $P(A)$ é a probabilidade de que A é verdade. Por exemplo, $p(r \ t=1)$: probabilidade de que o robô esteja na posição l no tempo t. Deseja-se computar a probabilidade de cada posição individual dadas as acções e as medidas dos sensores. $P(A|B)$: probabilidade condicional de A dado que se conhece B. Por exemplo, $p(r \ t=l|ir)$: probabilidade de que o robô r esteja na posição l no tempo t dadas as entradas dos sensores ir . No início, sem conhecimento da posição inicial a distribuição de probabilidade é uniforme. O robô detecta a primeira porta: ao ver só uma porta, possui a mesma probabilidade de estar nas portas 1, 2 ou 3. O robô desloca-se. O modelo de movimento possibilita estimar uma nova distribuição de probabilidade baseada no movimento. O robô detecta a segunda porta: baseado no conhecimento a priori, a probabilidade de estar na porta 2 é dominante.

Capítulo 6 – Planeamento de Caminhos

Navegação de robôs móveis: um robô móvel enfrenta 3 problemas: Onde estou?; Para onde vou?; Como vou lá chegar?. Estes três problemas definem o processo de navegação de robôs móveis. É a capacidade do robô se movimentar dentro do seu ambiente, sendo capaz de atingir uma posição objectivo, enquanto se desvia de obstáculos que podem ser encontrados no seu caminho. A navegação de robôs móveis está relacionada com a realização de três tarefas por parte do robô: localização (a determinação da sua própria posição no ambiente); construção de mapas (a construção de uma representação do ambiente); planeamento de trajectória (a determinação de um caminho no ambiente entre a posição inicial e a posição objectivo, sem que colida com obstáculos). Quando nos referimos à navegação de robôs móveis autónomos, queremos descrever as técnicas que fornecem os meios para que um robô autónomo se mova de forma segura de um local para outro do ambiente. Encontrar um caminho, da posição actual para a posição objectivo, é um dos problemas fundamentais da robótica móvel autónoma. Um método de planeamento de trajectória deve garantir um caminho até o destino, ou indicar se o destino é inacessível.

Problemas relacionados: diversos problemas vêm desafiando os cientistas da área de robótica móvel à muitos anos (localização do robô, obstáculos móveis, imprecisão do mapa, imprecisão dos sensores, etc). Foram

desenvolvidas diversas técnicas para tentar resolver estes problemas e tornar a navegação mais robusta e capaz de actuar de forma satisfatória em ambientes complexos. - **Localização:** É o principal problema que dificulta a navegação. Sem saber a posição do robô em relação à sua representação do ambiente (mapa), pouquíssimos sistemas são capazes de controlar o robô de forma adequada. Problema bastante complexo, mas que está na base de qualquer sistema de controlo robótico. - **Obstáculos Móveis:** Os obstáculos móveis dificultam ainda mais o planeamento de trajectórias. Têm sido desenvolvidos alguns métodos que tentam resolver o problema: • Métodos de formulação espaço/tempo (baseados no conhecimento preciso da movimentação e trajectória de todos os obstáculos do ambiente. Dados os movimentos previstos de todos os obstáculos no ambiente, é criada uma área de alcance contendo todas as possíveis posições do robô no futuro. Então esta área é utilizada para guiar o robô para a posição desejada. O principal problema com essa abordagem é a quantidade de tempo necessário para construir a área de alcance) e de detecção de colisões (baseados na navegação reactiva). - **Mapeamento do ambiente:** Para executar tarefas complexas de forma eficiente num dado ambiente, os robôs móveis devem ser capazes de adquirir e/ou manter um modelo desse ambiente. Em geral o robô inicia a navegação somente com um conhecimento parcial, ou até mesmo sem nenhum conhecimento sobre o ambiente. À medida que o robô se move no ambiente novos obstáculos são detectados. Então o modelo de ambiente inicial torna-se incompleto e necessita de ser actualizado. O robô deve ter a capacidade de diferenciar obstáculos móveis de obstáculos estáticos, pois somente os obstáculos considerados fixos no ambiente poderão fazer parte do mapa. Vários factores impõem limitações práticas na habilidade do robô em adquirir e utilizar de forma eficiente um modelo de ambiente: limitações sensoriais; ruído sensorial; movimento impreciso; complexidade e dinâmica do ambiente; necessidades de tempo real. As técnicas de mapeamento podem ser divididas em duas categorias: geométricas e topológicas. As técnicas geométricas geralmente são implementados utilizando-se matrizes de ocupação que dividem a área a ser mapeada em unidades menores. Cada uma destas unidades possui um atributo que indica a probabilidade daquele espaço estar ou não ocupado. As técnicas topológicas representam o ambiente como uma colecção de pontos de referência interconectados. Tanto a posição do robô como as conexões entre os pontos de referência podem ser modelados como distribuições de probabilidade. - **Fusão sensorial:** os robôs autónomos dependem normalmente de vários sensores para obter uma visão coerente e consistente do estado actual do ambiente, mas devido a que diferentes sensores reagem de forma diferente ao mesmo estímulo, ou fornecem dados incorrectos ou inconsistentes, estes acabam por provocar incertezas na navegação. Estas discrepâncias entre os sensores devem ser tratadas de alguma forma para permitir que o robô tenha uma visão unificada do ambiente. A fusão sensorial diz respeito a esse tratamento. A maioria dos avanços nas técnicas de fusão sensorial são baseados na teoria de estimativas onde se destacam os métodos Filtro de Kalman e Estimativas Bayesianas. Apesar destes métodos terem tido sucesso em combinar dados sensoriais para determinar um subconjunto comum de estados utilizando sensores similares (ex: a localização do robô derivada de um GPS, odometria e leituras de um giroscópio), ainda não parece existir uma extensão óbvia para a fusão de sensores multi-modal. A vantagem das abordagens multi-modais é que elas podem combinar dados de sensores não similares (ex: um sensor de ruído e uma câmara de vídeo) para imitar o raciocínio humano (ex: ouvir uma carro indica uma maior probabilidade de se estar a ver um carro).

Principais técnicas de navegação: Sensorial/Reativa; roadmap (Mapa de Estrada); Matricial.

Abordagem Sensorial/Reactiva: o processo de planeamento de trajectórias do robô incorpora directamente as informações sensoriais, reflectindo o estado actual do ambiente. Isto é justamente o oposto das técnicas de navegação clássicas, onde o conhecimento total sobre a geometria do ambiente precisa ser estabelecido antes de se iniciar o processo de planeamento. A navegação baseada em sensores pode ser preferível em relação aos métodos clássicos porque: - em muitos casos o robô não possui conhecimento antecipado sobre o ambiente onde ele ira actuar; - o robô dispõe de um conhecimento parcial sobre o ambiente devido a limitação de memória; - os modelos de ambiente geralmente não são precisos; - no ambiente podem ocorrer situações inesperadas ou mudanças rápidas. ► **Algoritmo “Bug”:** inspirados no comportamento de insectos. Método apropriado para quando se tem informação limitada sobre o espaço. Apenas é conhecida a direcção até ao objectivo. O robô apenas tem percepção local para detectar obstáculos. Assume-se um espaço onde existe um número finito de obstáculos com formas finitas que podem ser intersectados por uma linha um número de vezes finito. ► **Algoritmo “Bug” 1:** considerar uma linha s-line que une a posição inicial do robô e a posição do objectivo. 1) Avançar em direcção ao objectivo segundo a s-line; 2) Rodear o obstáculo até encontrar novamente a s-line; 3) Deixar de rodear o obstáculo e continuar em direcção ao objectivo segundo a s-line; 4) Repetir até atingir o objectivo. ► **Algoritmo “Bug” 2:** 1) Avançar em direcção ao objectivo; 2) Rodear o obstáculo até poder avançar novamente em direcção ao objectivo; 3) Repetir até atingir o objectivo

• **Navegação baseada em comportamentos:** o robô executa comportamentos como “seguir a parede” ou “entrar na sala”. Estes comportamentos podem ser representados através de autómatos de estados finitos nos quais os estados correspondem aos comportamentos e os arcos correspondem aos gatilhos que indicam um novo comportamento. O estado actual indica o comportamento seguido pelo robô. Quando o robô recebe as informações sensoriais, e estas informações correspondem a um determinado gatilho que o estado actual possui, o comportamento que o gatilho indica torna-se no novo estado actual. Já que os autómatos de estados finitos são baseados em comportamentos e gatilhos, o robô não necessita de um modelo do ambiente ou informações completas sobre o estado actual do mundo.

Abordagem Roadmap: para a geração de caminhos consiste em reduzir as informações do ambiente a um grafo representativo dos possíveis caminhos existentes entre a posição actual do robô e a posição objectivo. Depois o grafo é usado para determinar uma trajectória que leve o robô da sua posição actual até ao objectivo. No geral os métodos roadmap são rápidos e simples de implementar, mas são limitados quanto à forma de representação das informações do ambiente.

► **Decomposição Celular:** a ideia básica por trás do método de decomposição celular é que um caminho entre a posição actual e o objectivo pode ser determinado subdividindo-se o espaço livre em regiões menores chamadas células. Após esta decomposição, um grafo de conectividade é construído de acordo com as adjacências entre as células, onde os nós representam as células no espaço livre, e os arcos entre os nós indicam as adjacências. Através do grafo de conectividade, um caminho contínuo, ou canal, pode ser determinado simplesmente seguindo as células livres adjacentes. A decomposição celular pode ser **exacta** - o espaço livre é decomposto em células cuja união é exactamente o espaço livre; ou pode ser **aproximada** - o espaço livre é decomposto em células de forma pré-definida (quadrados por exemplo), em que a união das células está estritamente incluída no espaço livre. •

Decomposição trapezoidal: O espaço S com a posição inicial do robô e a posição do objectivo. Partição do espaço livre em células trapezoidais e triangulares com base nos vértices dos obstáculos e na fronteira do ambiente (partição exacta). Este método garante uma partição exacta e completa: todo o espaço livre será decomposto em células. A decomposição em células é representada num grafo cujos vértices representam as células e as arestas representam a relação de adjacência entre estas. Pesquisa de um caminho de células adjacentes desde a posição inicial do robô até à posição do objectivo. Transformação da sequência de células num caminho livre de obstáculos, por exemplo, conectando os pontos médios dos segmentos de intersecção entre duas células adjacentes. A decomposição trapezoidal é exacta mas não é óptima. A obtenção do número mínimo de células é um problema NP-completo. • **Decomposição quadtree:** Cada região ocupada com uma mistura de obstáculo e espaço vazio é subdividida recursivamente em quatro quadrantes. Este método não garante uma partição exacta e completa. Cada região ocupada com uma mistura de obstáculo e espaço vazio é subdividida recursivamente em quatro quadrantes. As células rectangulares são decompostas recursivamente em células rectangulares mais pequenas. Para um dado nível de resolução apenas são usadas as células cujo interior esteja totalmente dentro do espaço livre. As células adjacentes são novamente representadas por um grafo que é usado para encontrar um caminho livre de obstáculos.

► **Grafo de Visibilidade:** é obtido gerando-se segmentos de recta entre os pares de vértices dos obstáculos. Todo o segmento de recta que estiver inteiramente na região do espaço livre é adicionada ao grafo. Para executar o planeamento de trajectória, a posição actual e o objectivo são tratados como vértices, isso gera um grafo de conectividade onde se podem utilizar algoritmos de procura para se encontrar um caminho livre. Os métodos mais simples utilizados para construir um grafo de visibilidade tem complexidade $O(n^3)$. O caminho mais curto que for encontrado no grafo de visibilidade é o caminho óptimo para o problema especificado. Mas os caminhos encontrados no grafo “tocam” os obstáculos. Isso não é aceitável num problema de navegação robótica, por esse motivo os obstáculos devem ser modificados, criando-se um espaço de configuração. Para a utilizar um método de navegação com grafo de visibilidade é necessário quer o mapa seja completo e bem definido. Não é possível navegar num ambiente que possua obstáculos móveis e a localização do robô móvel deve ser conhecida com precisão durante toda a navegação. O espaço S com a posição inicial do robô e a posição do objectivo. Construção de um grafo de visibilidade formado por todas as sequências de segmentos que unem vértices de obstáculos sem atravessar nenhum deles, incluindo a posição inicial do robô e a posição do objectivo. Determinação do caminho mais curto entre a posição inicial e a posição final, isto é, caminho com menor número de segmentos. O caminho final requer processamento adicional para se afastar dos obstáculos.

► **Diagramas de Voronoi:** é uma estrutura geométrica que representa informações de proximidade sobre uma série de pontos ou objectos. Dada uma série de objectos, o plano é particionado atribuindo para cada ponto o seu objecto mais próximo. Os pontos que não possuem um único objecto mais próximo formam o diagrama de Voronoi. Isto é, os

pontos no diagrama de Voronoi são equidistantes de dois ou mais objectos. Os diagramas de Voronoi podem ser generalizados de diferentes maneiras, dependendo da aplicação prática desejada. Podem por exemplo ser construídos para polígonos, em vez de para objectos pontuais. No caso do planeamento de trajetória, cada obstáculo é representado por polígonos côncavos ou convexos. Para encontrar o diagrama generalizado de Voronoi para esta colecção de polígonos, podemos calcular o diagrama através de uma aproximação, convertendo os obstáculos numa série de pontos. As faces dos polígonos são subdivididas numa série de pontos e depois é calculado o diagrama de Voronoi para essa colecção de pontos. Após o diagrama de Voronoi ser calculado, os segmentos do diagrama que intersectam algum obstáculo são eliminados. Uma vez que o diagrama esteja calculado para todo o conjunto de obstáculos, é necessário adicionar a posição do robô e o destino ao conjunto de pontos. Depois podemos então utilizar um algoritmo de procura para encontrar um caminho do ponto inicial até o destino, que será um subconjunto do diagrama de Voronoi. Este método gera uma rota que na sua maior parte permanece equidistante dos obstáculos, criando um caminho seguro para o robô se locomover, apesar de não ser o caminho mais curto. Este método possui os mesmos inconvenientes dos grafos de visibilidade. O mapa deve ser preciso e o ambiente não pode conter obstáculos móveis. A posição do robô precisa ser conhecida com precisão para que os caminhos possam ser seguidos.

Abordagem Matricial: representa o ambiente utilizando uma matriz de tamanho fixo. Cada célula da matriz pode conter diferentes atributos que auxiliam na navegação do robô. Cada célula, por exemplo, pode indicar a presença de um obstáculo na região correspondente do ambiente, ou indicar a probabilidade desta determinada região estar ou não ocupada por um obstáculo. As técnicas que utilizam matrizes, também chamadas de matrizes de ocupação, são simples de se construir, representar e manter em ambientes de larga escala. O uso de matrizes facilita a determinação de caminhos até o destino. No entanto as matrizes consomem muita memória e requerem muito tempo para actualizações globais. Isto ocorre pois a resolução da matriz deve ser precisa o suficiente para capturar todos os detalhes importantes do ambiente. • **Transformada de Distância:** neste método a célula do destino é marcada com uma distância de valor 0. Todas as outras posições são marcadas com um valor muito alto, ou “infinito”. O algoritmo inicia o cálculo no destino, e em cada passo ele visita todas as posições adjacentes às posições visitadas no passo anterior. O valor de distância para a posição L adjacente a posição M é actualizado da seguinte forma: Se mapa(i) estiver ocupado então $d(i) = \text{infinito}$; Senão $d(i)$ recebe o menor entre $d(i)$ ou $(d(j) + c(i,j))$; Onde $c(i, j)$ é o custo associado para se mover da posição M para a posição L. A transformada de distância se expande em torno do destino como uma onda, propagando-se em torno dos obstáculos. O caminho mais curto de qualquer posição até ao destino pode ser encontrado seguindo os vizinhos com o menor $d(i)$ até que $d(i)$ seja igual a 0. Cada vez que a posição do destino for alterada, a transformada de distância para todo o mapa deve ser recalculada. Outro problema associado com este método, é o facto de que os caminhos são gerados muito próximos dos obstáculos, não aceita obstáculos móveis e o mapa deve ser bem definido. • **Campos de Potencial:** os métodos de campos potenciais para a navegação de robôs móveis são muito populares entre os investigadores da área. Nestes métodos os obstáculos exercem uma força virtual repulsiva no robô, enquanto o objectivo aplica uma força virtual atractiva no robô. A soma de todas as forças, a força resultante R, determina a direcção e a velocidade do movimento. Uma das razões para a popularidade deste método é a sua simplicidade e elegância. Um método de campos potenciais simples pode ser implementado rapidamente e produz um resultado inicial aceitável sem a necessidade de muitos refinamentos. Obstáculos Potencial atractivo Potencial repulsivo, o campo potencial é a soma de dois potenciais (duas forças): o potencial atractivo e o potencial repulsivo. A soma dos dois potenciais define um gradiente de navegação que dirige o movimento local do robô em direcção ao objectivo. Apesar dos bons resultados fornecidos pelos métodos de campos de potencial, eles possuem uma série de problemas inerentes, independentes de uma implementação em particular: - Situações de armadilha devido a mínimos locais (comportamento cíclico); - Não conseguem passar através de obstáculos muito próximos; - Oscilação na presença de obstáculos ou em corredores estreitos. As armadilha podem ser solucionadas com o uso de heurísticas que acabam por gerar caminhos muito longo. A utilização de campos potenciais necessita que o robô seja capaz de identificar a posição, ou direcção do objectivo. A grande vantagem deste método é que ele não necessita de uma representação interna do ambiente (mapa), e permite navegar por um ambiente que possua obstáculos móveis. • **Algoritmos de procura AStar e DStar:** apesar dos algoritmos de procura AStar(A^*) e DStar(D^*) não necessitarem que o espaço de estados seja representado como uma matriz, na maioria das aplicações de navegação para robôs móveis em que se utiliza AStar ou DStar o ambiente é representado desta forma. O algoritmo AStar é indicado para ambientes estáticos e o DStar para ambientes dinâmicos.

Resumo dos Método: representação, custo computacional, mínimos locais, global, ambiente dinâmico, posição precisa. **DistBug:** n utiliza, baixo, não, não, não, não. **Nav baseada em comportamentos:** não utiliza, baixo, sim, não, sim, não. **Mapas neurais:** campo neural, alto, sim, sim, não, sim. Grafo de visibilidade: poligonal, alto, não, sim, não, sim. **Voronoi:** poligonal, alto, não, sim, não, sim. **Decomposição Celular:** poligonal e topológica, médio, não, sim, não, sim. **Transformada de distancia:** grade, médio, não, sim, não, sim. **Campos Potenciais:** grade e poligonal, médio, sim, não, sim, não. **Astar:** grade, baixo, não, sim, não, sim. **Dstar:** grade, médio, não, sim, sim, sim.

Capítulo 7 – Architecturas de Controlo

Arquitectura Delirativa: o robô usa todos os sensores e toda a informação global que possui do mundo e cria um plano de acção. O robô executa uma pesquisa através de todos os planos possíveis até encontrar uma que executa a tarefa. Este tipo de raciocínio pode ser muito demorado pelo que não é adequado a situações em que o robô tem que actuar com rapidez. No entanto, nas situações em que há tempo, estas arquitecturas são apreciáveis, pois é possível encontrar um plano que optimiza a execução de determinada tarefa.

Arquitectura baseadas em comportamentos: são inspirados na biologia e tratam modelar a maneira como os cérebros dos animais lidam com os problemas difíceis de raciocínio e acção. Estas arquitecturas são também formadas por várias partes ou camadas, mas as diferentes camadas são muito diferentes entre si. As camadas são formadas por conjuntos de comportamentos, isto é, processos que recebe importantes e enviam outputs entre si.

Arquitectura híbrida: combinar o melhor das arquitecturas deliberativas e reactivas, nesta arquitectura existe um módulo responsável por executar o planeamento a longo prazo e o outro para lidar com situações que necessitam de reacção imediata. Exemplo para controlar o robô bombeiro: (input, sensores, percepção, modulação, planeamento, execução, motores, output dos actuadores) 1º as entradas sensoriais são utilizadas para modificar a representação interna do ambiente; 2º baseada nesta representação um plano a longo prazo é elaborado, isto resulta numa série de acções que o robô deve executar para alcançar o seu objectivo; 3º esta série de acções é utilizada para comandar os actuadores do robô, isto completa o ciclo de controlo e o sistema é reiniciado para atingir novos objectivos.

Arquitectura subsumption: proporciona uma manobra de combinar, controlo distribuído em tempo real, com comportamentos disparados por informação sensorial, os comportamentos são simplesmente níveis de controlo que são executados em paralelo, sempre que determinados sensores a disparem. **Características:** a ligação directa entre os sensores e actuadores significa que a maioria dos comportamentos pode ser visto como simples reflexos, isto é, importante porque este tipo de sistemas não necessita de um modelo do mundo. Uma vez, que não há um modelo, o robô necessita de muita memória e o processamento é trivial, o que permite a implementação em microcontroladores simples. Comportamentos complexos podem ser obtidos a partir das iterações de comportamentos mais simples (ex. para o robô bombeiro: sonar - evitar; câmara – centrar robô; sensor luz – detectar vela; sensor linha – detectar linha; navegar; actuadores).

Problema da integração entre a camada reactiva e a deliberativa: as camadas diferem em termos de representação, a camada reactiva usa tipicamente uma representação muito simples do estado do ambiente e trabalha tipicamente com intervalos de tempo muito pequenos implementados em loops de feedback entre sensores e motores; a camada deliberativa usa representações complexas do mundo e trabalha com escalas de tempo muito maiores na ordem dos minutos ou horas. Para resolver este problema, geralmente requer um terceiro módulo responsável pela gestão da comunicação entre o módulo planeador e o módulo reactivo. Por este motivo estas arquitecturas são também chamadas de arquitecturas de 3 camadas (planeador, camada intermédia, sistema reactivo).

Arquitectura de microprocessador: RISC (Reduced instruction set computer); CISC (Complex instruction set computer); SISC (Specific instruction set computer).