

## Capítulo

# 4

## Planejamento de Caminho para Sistemas Robóticos Autônomos

Luís B. P. Nascimento, Vitor G. Santos, Diego S. Pereira,  
Daniel H. S. Fernandes e Pablo J. Alsina

### *Abstract*

*Path Planning is one of the most fundamental processes in autonomous robotics. It is a computational method that calculates a set of poses that a robot must perform in order to fulfil a specific task without colliding with obstacles. Currently, many path planners, libraries and virtual platforms has been developed focused in this area due to its applicability in robotics, games, virtual reality, bioinformatics, among others. In this chapter we present introductory aspects of path planning, presenting the main techniques, libraries and applications, focusing in autonomous robotics.*

### *Resumo*

*O Planejamento de Caminho é um dos processos fundamentais da robótica autônoma dado que planejadores são métodos computacionais capazes de calcular um conjunto de poses que o robô deve assumir para para realizar uma determinada tarefa, evitando colisões. Atualmente tem surgido diversos planejadores, bibliotecas e plataformas para trabalhar nessa área devido a sua aplicabilidade não só na robótica, mas também para jogos/realidade virtual, bioinformática, entre outros. Neste capítulo, aspectos introdutórios sobre planejamento de caminho são abordados, apresentando as principais técnicas, bibliotecas, e aplicações, com destaque na robótica autônoma.*

### **4.1. Introdução**

É indiscutível que o ramo da robótica vem se expandindo cada vez mais, já que dispositivos robóticos vêm sendo aplicados na execução de tarefas anteriormente realizadas pelo homem. Em fábricas de produção, por exemplo, atividades que exigem demasiada força física e um tempo hábil para serem concluídas, estão sendo realizadas com maior frequência por robôs, uma vez que máquinas possuem características como repetibilidade,

velocidade e segurança na execução de tarefas. Em outras aplicações em que dispositivos interagem diretamente com o ser humano, como em procedimentos cirúrgicos, órteses ativas e exoesqueletos, por exemplo, a precisão dos movimentos é algo imprescindível, sendo assim, a robótica é introduzida a fim de auxiliar na realização de movimentos humanos e na redução de falhas.

Dispositivos robóticos necessitam de uma programação prévia para realizar determinadas tarefas, e quando programados corretamente, realizam tarefas complexas com sucesso. Anteriormente, os robôs eram limitados a movimentar-se apenas de acordo com as situações para as quais foram designados. Atualmente, são capazes de realizar movimentos de maneira autônoma, com um mínimo de intervenção humana. O robô autônomo necessita reagir de forma coerente à mudanças dinâmicas no ambiente no qual está inserido, como por exemplo, desviar mediante a presença de obstáculos. Nesse contexto, tomando como base os problemas fundamentais da robótica (Percepção, Planejamento e Ação), um robô autônomo deve ser habilitado a fazer aquisição de informações do ambiente para estimar sua posição e se mover em segurança de uma posição para outra, sem perder sua localização ou sem colidir com possíveis obstáculos [Murphy 2000].



Figura 4.1: Fluxograma com os problemas fundamentais da robótica autônoma.

No sistema de percepção, o robô faz uso de sensores para coletar informações do ambiente no qual está inserido. Na Figura 4.2 um sistema de percepção cria um mapa 2D do ambiente na qual o robô está inserido através de sensores, ao mesmo tempo que se localiza, técnica conhecida como SLAM (Simultaneous Localization and Mapping).

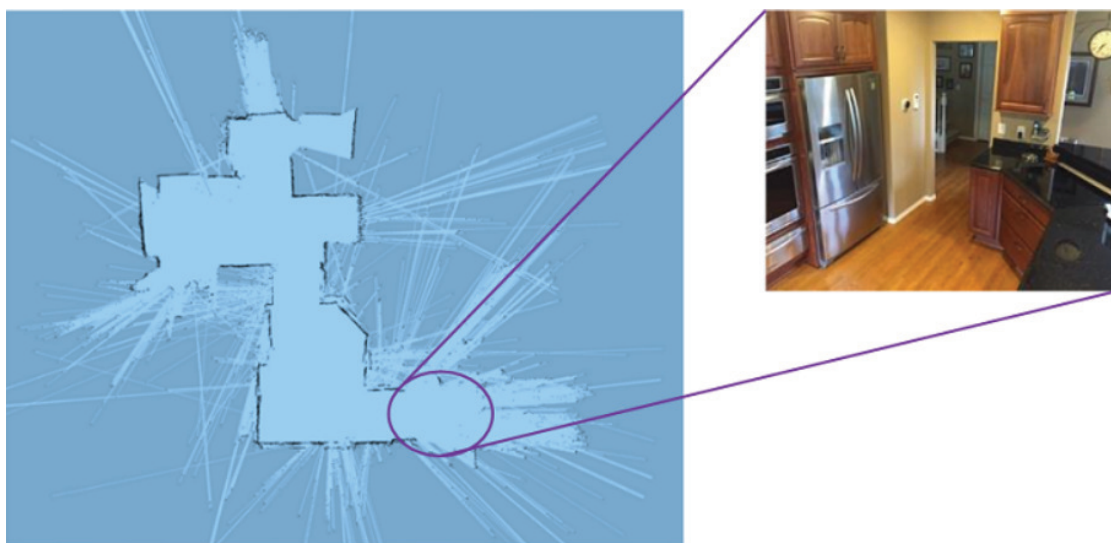


Figura 4.2: Mapeamento e localização simultânea. Na esquerda, um mapeamento em 2D de um ambiente. Na direita, o ambiente na qual o robô está inserido. Ilustração adaptada de [Sakemoto 2017]

O Planejamento de caminho (*Path planning* ou *motion planning*) é um dos principais problemas da robótica autônoma pois define uma sequência de ações a serem realizadas para que um robô, partindo de seu estado atual, possa alcançar o objetivo final, evitando colisões ao longo do percurso [Siegwart and Nourbakhsh 2004, Latombe 1991]. Um exemplo do processo de planejamento de caminho para um robô móvel é ilustrado na Figura 4.3.

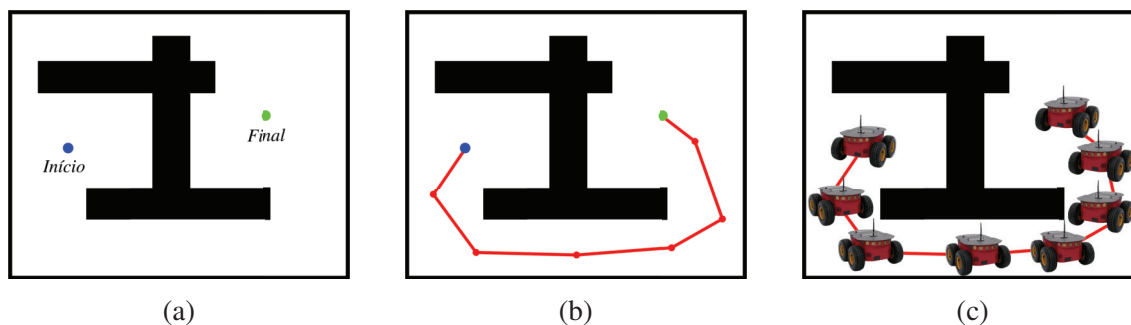


Figura 4.3: Ilustração do processo de planejamento de caminho. (a) Representação do mapa do ambiente. (b) Caminho calculado entre o ponto inicial e o ponto final. (c) Robô executando a ação planejada.

O planejamento de caminho é realizado por meio de métodos computacionais capazes de calcular um conjunto de poses e orientações que o robô deve assumir para realizar uma determinada tarefa. Na Figura 4.3 o robô deve sair de um ponto inicial (ponto azul) até um ponto final (ponto verde), evitando colisões. Na Figura 4.3(a) o mapa do ambiente é representado. Em Figura 4.3(b) mostra um caminho em vermelho que descreve as poses e orientações que o robô móvel deve assumir para sair do ponto inicial e chegar até o final. Na Figura 4.3(c), de maneira ilustrativa, o robô realiza as ações planejadas.

Planejar um caminho é uma etapa essencial na automatização de movimentos para todos os tipos de robôs, seja na transposição de obstáculos ou subir uma escada para um robô humanoide, definir uma missão para um veículo aéreo autônomo ou até mesmo uma tarefa para um robô manipulador industrial. Além disso, a cada momento vêm surgindo novos métodos, bibliotecas e plataformas para trabalhar com o planejamento devido a sua aplicabilidade não só na robótica, mas também para jogos [Edelkamp and Plaku 2014], bioinformática [Al-Bluwi et al. 2012], entre muitos outros. Nesse sentido, este capítulo apresenta uma visão introdutória sobre o problema do planejamento de caminho mostrando sua importância na robótica autônoma.

As demais seções desse documento estão descritas como segue: Na Seção 4.2 é apresentado um breve histórico sobre o planejamento de caminho, onde o importante conceito de Espaço de Configurações é introduzido e importantes definições são realizadas. Na Seção 4.3 os métodos clássicos de planejamento são apresentados. A Seção 4.4 apresenta o que são métodos baseados em amostragem, ilustrando alguns algoritmos. Na Seção 4.5 são apresentadas algumas ferramentas interessantes sobre planejamento de caminho, como OMPL e MoveIt. Finalmente, as conclusões são apresentadas na Seção 4.6, seguida dos currículos resumidos dos autores e das referências.

## 4.2. Histórico e Espaço de Configurações

Segundo [LaValle 2006], a primeira ideia de algoritmo de planejamento foi provavelmente introduzida por [Nilsson 1969], entretanto, apenas no final dos anos de 1970 as pesquisas envolvendo planejamento de caminho ganharam maior força, devido ao advento do conceito de espaço de configurações, o *C-space*, uma representação que abstrai e simplifica problemas de planejamento. O *C-space* é uma representação matemática que mapeia um problema que está especificado no espaço de trabalho (*workspace*), incluindo as características físicas do robô, para um novo espaço, onde os obstáculos possuem uma representação equivalente e o robô é representado por um ponto, com forma e dimensão irrelevante. Essa ideia de representação do robô foi inicialmente introduzida por [Udupa 1977] e melhor explorada e popularizada por [Lozano-Pérez and Wesley 1979] e [Lozano-Perez 1983]. Uma ilustração do processo de mapeamento do espaço de trabalho para o espaço de configurações é mostrado na Figura 4.4.

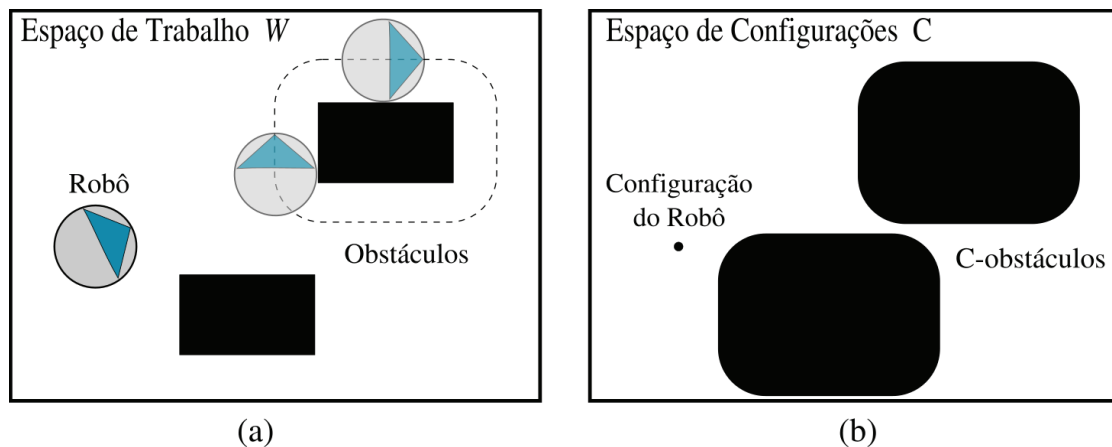


Figura 4.4: Ilustração do processo computação do espaço de configurações para um robô circular. (a) Representação do robô no espaço de trabalho. (b) Representação do robô e dos obstáculos no espaço de configuração.

A representação do mundo através do *C-space* [Lozano-Pérez and Wesley 1979] facilita o planejamento dois motivos fundamentais: O processo de chegar a colisão entre o robô e um obstáculo ocorre ao chegar se um ponto está localizada numa determinada região. Além disso, possibilita que problemas relacionados a geometria e cinemática do robô sejam resolvidos pelo mesmo algoritmo.

Na Figura 4.5 pode ser observada mais uma ilustração que aborda o espaço de configurações, agora para um robô manipulador, com duas juntas rotacionais ( $\theta_1$  e  $\theta_2$ ).

Em 4(a) robô no espaço de trabalho realiza uma tarefa de sair de uma posição e assumir outra e a na outra imagem 4(b) o mesmo cenário é utilizado, porém o robô é representado de maneira pontual (sem forma ou dimensões relevantes), mapeado para o *C-space* e realizando a mesma tarefa. A simplificação proporcionada pelo *C-space* é facilmente perceptível, principalmente quando observa-se que o planejamento no espaço de trabalho  $\mathcal{W}$  envolve toda a análise da cinemática para obter uma sequência movimentos que permita uma trajetória segura partindo de sua pose inicial até sua posição final, diferentemente do caminho resultante no espaço de configuração, representado simplesmente

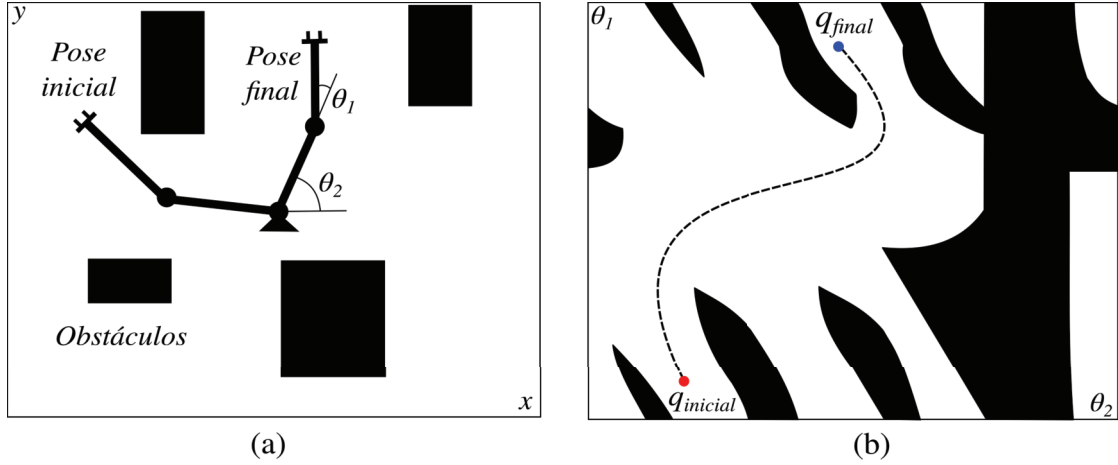


Figura 4.5: (a) Robô manipulador com duas juntas rotacionais representado no Espaço de trabalho  $\mathcal{W}$  com duas poses diferentes. Em (b) representação do caminho que o robô deve assumir para sair de  $q_{inicial}$  para  $q_{final}$ , mapeado no espaço de configuração  $C_{space}$ . Ilustração adaptada de Siegwart and Nourbakhsh (2004)

por uma sequência de pontos entre as configurações  $q_{inicial}$  até  $q_{final}$  através da região  $C_{livre}$ .

A fim de facilitar o entendimento dos conceitos abordados nesse trabalho, serão apresentadas formalmente as definições associadas à representação de um problema de planejamento no espaço de configuração. As seguintes definições foram retiradas da tese de doutorado de [Molina 2014], que apresentou uma extensa fundamentação teórica sobre planejamento:

**Definição 1 (Configuração  $q$  de um robô)** A configuração  $q$  de um robô é um conjunto de  $n$  parâmetros que especifica completamente a postura de um robô,  $\mathcal{A}$ , em seu espaço de trabalho,  $\mathcal{W}$ .

**Definição 2 (O espaço de configurações  $\mathcal{C}$ , ou  $C_{space}$ )** O espaço de configurações  $\mathcal{C}$ , ou  $C_{space}$ , é o espaço  $n$ -dimensional  $\mathcal{C}$  que contém todas as possíveis configurações  $q$  de um robô  $\mathcal{A}$  em seu espaço de trabalho  $\mathcal{W}$ .

**Definição 3 (Caminho  $\mathcal{T}$ )** Dada uma configuração inicial do robô  $\mathcal{A}$ ,  $q_{inicial}$ , e uma configuração final desse mesmo robô,  $q_{final}$ , um caminho no espaço de configurações é o mapeamento definido por:

$$\mathcal{T} : [0, 1] \rightarrow C_{space} \mid \mathcal{T}(0) = q_{inicial} \text{ e } \mathcal{T}(1) = q_{final}. \quad (1)$$

**Definição 4 (Obstáculo no  $C_{space}$  ou  $C_{obs}$ )** Seja  $\mathcal{B}$  o conjunto de todas as regiões ocupadas em  $\mathcal{W}$  e  $\mathcal{A}(q)$  a parte de  $\mathcal{W}$  ocupada pelo robô  $\mathcal{A}$  em uma dada configuração  $q$ . Então, os obstáculos no  $C_{space}$  são definidos como:

$$C_{obs} = \{q \in C_{space} \mid \mathcal{A}(q) \cap \mathcal{B} \neq \emptyset\}. \quad (2)$$

**Definição 5 (Região navegável do  $C_{space}$  ou  $C_{livre}$ )** A região navegável do  $C_{space}$ , ou  $C_{livre}$ , é o subconjunto do espaço de configurações que não pertence ao  $C_{obs}$ , ou seja:

$$C_{livre} = C_{space} \setminus C_{obs}. \quad (3)$$

**Definição 6 (Caminho  $\mathcal{T}$  livre de obstáculos)** Dada uma configuração inicial do robô  $\mathcal{A}$ ,  $q_{inicial}$ , e uma configuração final desse mesmo robô,  $q_{final}$ , um caminho livre de obstáculos no espaço de configurações é o mapeamento definido por:

$$\mathcal{T} : [0, 1] \rightarrow C_{livre} \mid \mathcal{T}(0) = q_{inicial} \text{ e } \mathcal{T}(1) = q_{final}. \quad (4)$$

A partir dessas definições, já será possível introduzir alguns métodos de planejamento de caminho presentes na literatura. Iniciaremos falando dos chamados Métodos Clássicos de planejamento, e depois comentaremos sobre alguns métodos baseados em amostragem.

### 4.3. Métodos clássicos de planejamento

Uma classificação apresentada em 1991 por Latombe [Latombe 1991], e ainda muito aceita nos dias atuais, divide as técnicas de planejamento quanto à forma de representar e interpretar o espaço de configuração do robô. A chamada *taxonomia clássica* agrupa os métodos de planejamento em três grandes categorias:

1. Métodos baseados em **Mapa de Rotas** (*Roadmaps*)
2. Métodos baseados em **Decomposição de Células** (*Cell Decomposition*)
3. Métodos baseados em **Campo Potencial** (*Potential Field*)

#### 4.3.1. Mapa de Rotas

Os métodos baseados em Mapa de Rotas, considerados uma das primeiras abordagens no planejamento de caminho, objetivam capturar a conectividade entre as diferentes regiões de  $C_{livre}$  por meio de uma rede de curvas 1-dimensional, isto é, um *mapa de rotas*. Com a estrutura mapa de rotas construída, o problema do planejamento resume-se a um processo de busca por um caminho  $\tau$  que conecte  $q_{inicial}$  e  $q_{final}$  entre os demais vértices.

Dentre as diversas abordagens aplicadas na construção de um mapa de rotas, é possível destacar o *grafo de visibilidade* (*visibility graph*) e diagrama de Voronoi (*Voronoi diagram*) como as técnicas mais populares. [Molina 2014] destaca que essas duas técnicas, individualmente, retratam duas das principais preocupações da navegação autônoma: A busca por caminhos mais curtos e a busca por caminhos mais seguros.

##### 4.3.1.1. Grafos de Visibilidade

O método de grafo de visibilidade foi inicialmente introduzido por [Nilsson 1969], sendo uma das primeiras estratégias de planejamento de caminho desenvolvidas. Esse método tem o objetivo de construir um caminho livre de obstáculos através de uma linha poligonal

que conecta o ponto inicial  $q_{inicial}$  e o ponto final  $q_{final}$  através dos vértices dos obstáculos do ambiente representados no  $C_{space}$  [Latombe 2012].

Dessa forma, um grafo de visibilidade pode ser definido como um grafo  $G$  não-dirigido, que possui as seguintes características:

- Os vértices de  $G$  correspondem a  $q_{inicial}$ ,  $q_{final}$  e os vértices de  $C_{obs}$ .
- Dois vértices de  $G$  são conectados por uma aresta se e somente o segmento de reta unindo eles for também uma aresta de  $C_{obs}$ , ou se este segmento estiver totalmente contido no espaço livre  $C_{livre}$ , com a possível exceção de suas extremidades.

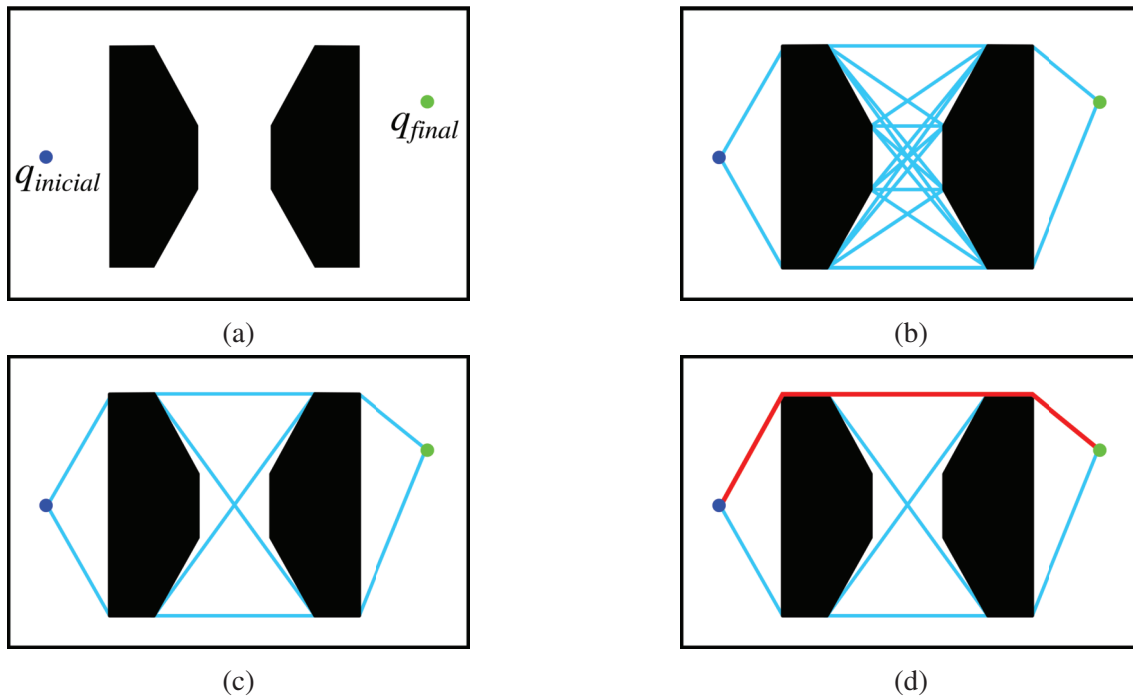


Figura 4.6: Planejamento com Grafo de Visibilidade. (a) Mapa no espaço de configuração. (b) Arestas do Grafo de visibilidade. (c) Arestas do Grafo de visibilidade reduzido. (d) Caminho livre de obstáculos encontrado.

Após a construção do grafo de visibilidade, é necessário realizar um algoritmo de busca para encontrar o caminho a ser percorrido pelo robô. Nessa etapa, geralmente procura-se pelo menor caminho possível de acordo com a distância Euclidiana no  $\mathbf{R}^2$ .

Essa metodologia da construção do grafo pode ter seu desempenho otimizado se percebermos que algumas arestas de  $G$  nunca serão utilizadas, uma vez que elas geram um maior caminho a ser percorrido. Assim, dados dois obstáculos, são úteis apenas as arestas que são tangentes a ambos os obstáculos. Esse grafo pode ser obtido pela eliminação de segmentos não-tangentes, denominamos *grafo de visibilidade reduzido*  $G'$ . Se em tal grafo existir um caminho que une  $q_{inicial}$  e  $q_{final}$ , é possível afirmar que  $G'$  conterá os segmentos de linhas que correspondem ao caminho de menor distância. [Otoni 2000].

### 4.3.1.2. Diagrama de Voronoi

No planejamento baseado no clássico diagrama de Voronoi [Dirichlet 1850], o espaço livre é mapeado por uma rede de arcos 1-dimensional, que visam maximizar a distância entre o robô e a região  $C_{obs}$ . Após o diagrama ser construído, as configurações inicial e final são conectadas aos pontos pertencentes ao mapa de rotas mais próximos à  $q_{inicial}$  e  $q_{final}$ , respectivamente. A Figura 4.7 ilustra um planejamento de caminho com diagrama de Voronoi.

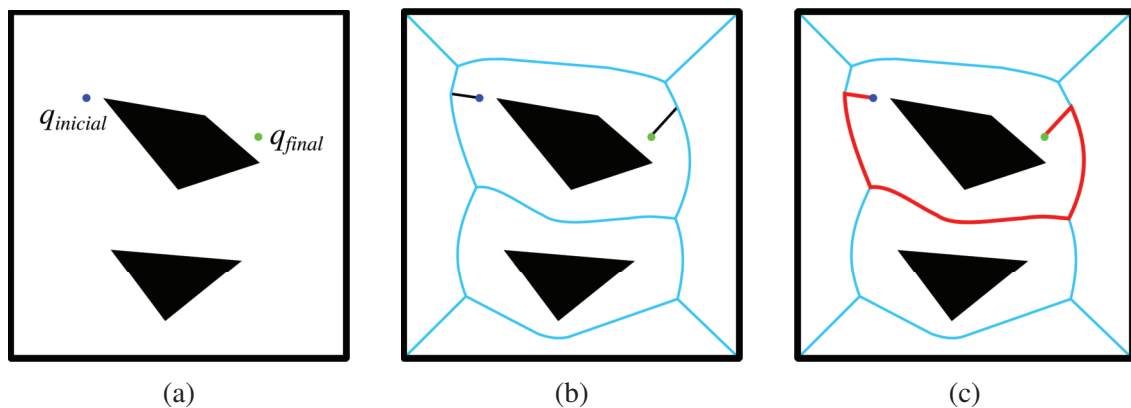


Figura 4.7: Planejamento de caminho baseado em Diagrama de Voronoi. (a) Mapa no espaço de configurações. (b) Diagrama de Voronoi calculado para os vértices do obstáculo. (c) Caminho encontrado. Ilustração adaptada de [Latombe 1991]

Para os métodos basados em mapa de rotas, uma etapa fundamental é a busca pelo menor caminho dentro do grafo, pois após a construção do grafo, o planejamento de caminho resume-se a uma busca por um caminho entre vértices e arcos de um grafo. Por meio de um algoritmo de busca é possível percorrer o grafo de maneira sistemática, explorando os seus vértices (nós) e arcos, a fim de obter o melhor caminho entre os nós representados pela configuração  $q_{inicial}$  e pela configuração  $q_{final}$ .

Conforme [Russell and Peter Norvig 1995], as estratégias de buscas se distinguem pela ordem em que os nós do grafo são examinados, sendo assim, classificados em Algoritmos de Busca não-informada ou busca cega (*Uninformed search*) e Algoritmos de busca informada ou busca heurística (*Informed search*). Nas abordagens de busca não-informadas é possível citar os algoritmos de Busca em Largura ou *Breadth-first* [Lee 1961, Even 1979], Busca em Profundidade ou *Depth-first* [Tarjan 1971], Busca de custo uniforme [Russell and Peter Norvig 1995] e o algoritmo de Dijkstra [Dijkstra 1959].

Nas abordagens de busca informada existe um componente fundamental nos algoritmos, a chamada *função heurística*, denotada por  $h(n)$ , que retorna um custo estimado entre um determinado nó e o nó objetivo, no caso, o nó que representa a configuração final. O algoritmo de Busca pela melhor escolha ou Busca gulosa (*Best-first*) [Pearl 1984] e o algoritmo A\* [Hart et al. 1968, Hart et al. 1972] são exemplos de buscas heurísticas.



### 4.3.2. Decomposição em Células

Nos métodos baseados em decomposição de células, o espaço é decomposto em um conjunto de regiões não sobrepostas, chamadas células. Na estratégia de decomposição exata em células, o espaço de configuração livre é decomposto em um conjunto de regiões convexas, construídas a partir dos limites geométricos da região de obstáculos, considerando que  $C_{obs}$  é uma região poligonal [Latombe 1991]. O método trata cada célula computada como um nó único que será armazenado num grafo, o qual representa a conectividade das regiões livres. O próximo passo é a busca pelas células que levam da configuração inicial até a configuração final. Por meio das células selecionadas, um caminho pode facilmente ser traçado seguindo os nós do grafo correspondentes às células convexas.

A Figura 4.8 ilustra uma decomposição exata de um espaço bidimensional povoado por dois obstáculos poligonais.

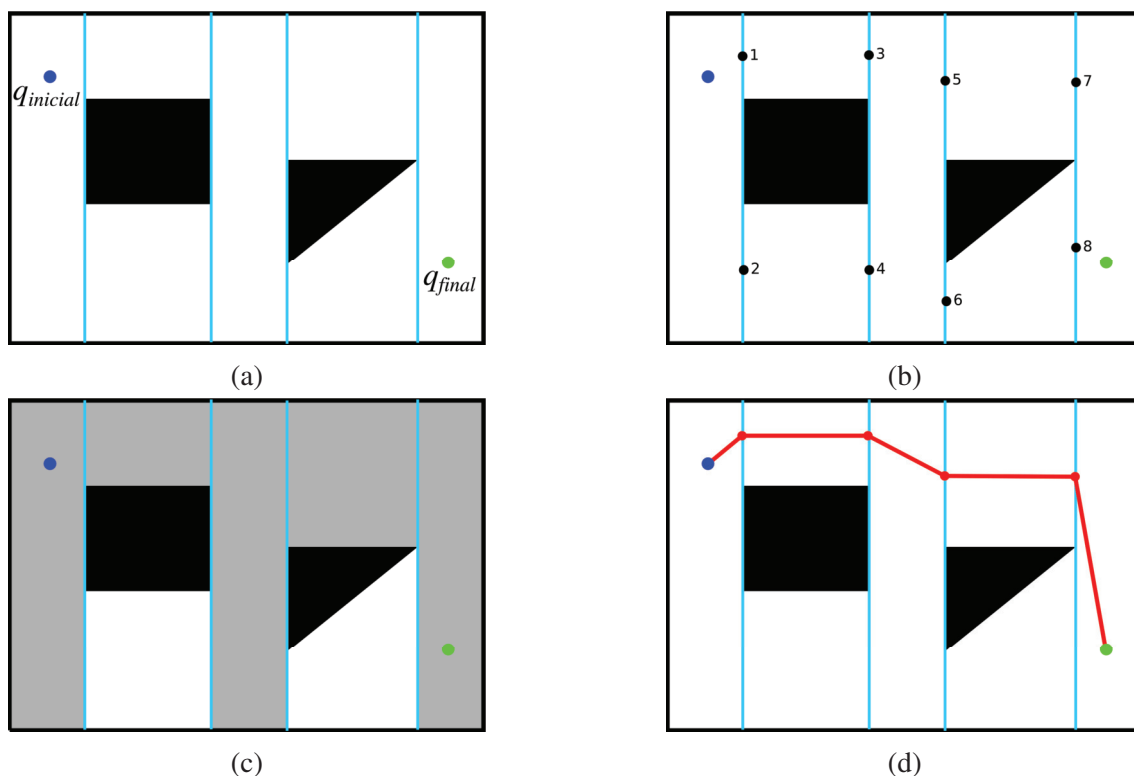


Figura 4.8: Planejamento baseado em Decomposição em Células. (a) Mapa decomposto em regiões convexas. (b) Cada célula é representada por um nó num grafo. (c) Células necessária para encontrar o menor caminho. (d) Caminho encontrado.

De modo similar aos métodos baseados em mapa de rotas, os métodos baseados em decomposição de células têm como principal objetivo a criação de um grafo não dirigido capaz de capturar a conectividade entre as diferentes regiões do espaço livre. Conforme [Molina 2014], a principal diferença entre as duas abordagens é que nos mapas de rotas se preocupa principalmente na forma como as arestas (arcos entre dois nós) são definidas, na decomposição em células a atenção é voltada na forma como os nós (células) que compõem o grafo são definidos, fazendo com que as arestas sejam definidas trivialmente a partir da simples relação de adjacência entre duas células.

### 4.3.3. Campos Potenciais

Nos métodos baseados em Campos Potenciais [Khatib 1986], sugere-se que o robô seja uma partícula movendo-se sob a influência de um campo de potencial artificial, tal que os obstáculos geram um campo potencial de repulsão contra o robô e a configuração final o robô. Dessa forma, a cada configuração, a direção do movimento do robô é determinada pela força resultante proveniente do campo potencial.

A Figura 4.9 um ambiente com três obstáculos mapeado para o espaço de configurações é discretizado em uma grade regular. A cada posição da grade, um potencial é atribuído com base em uma função potencial  $U(q)$ , definida como segue:

$$U(q) = U_{atr}(q) + U_{rep}(q). \quad (5)$$

sendo que  $U_{atr}(q)$  é a função de potencial atrativo associada à configuração final e  $U_{rep}(q)$  é a função de potencial repulsivo, associada à região de obstáculos.

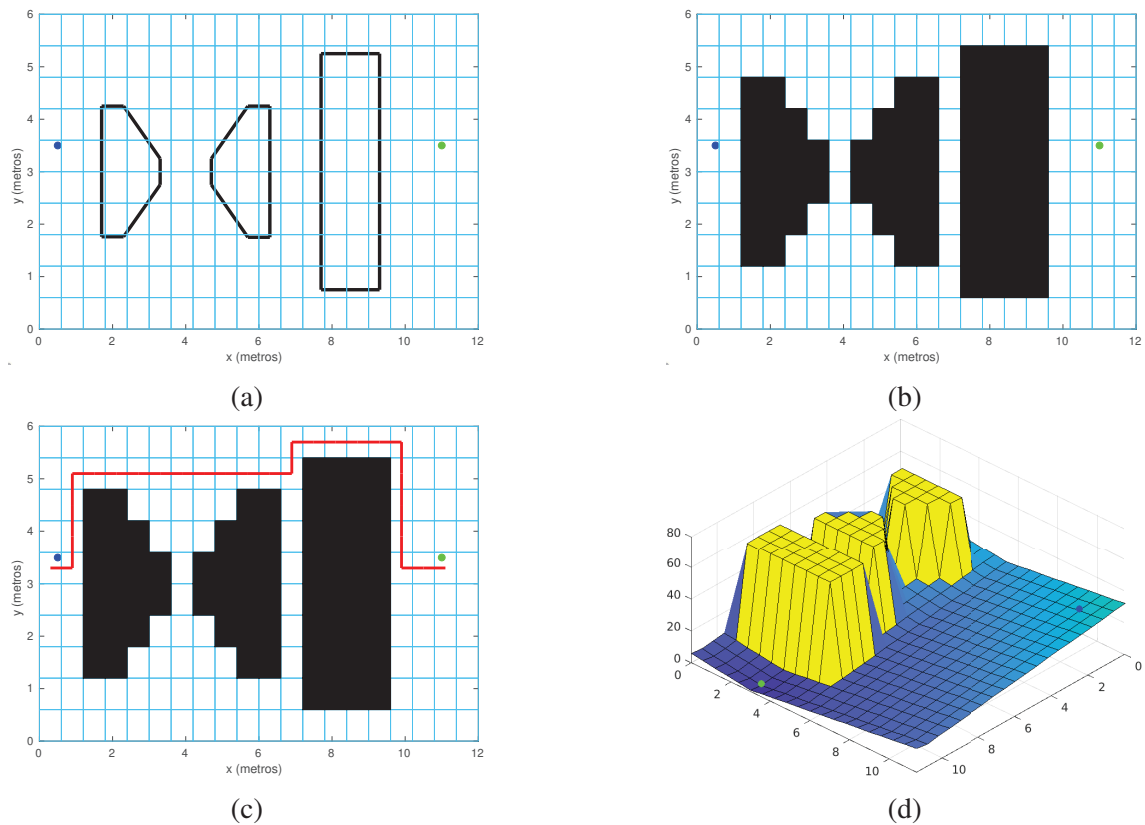


Figura 4.9: Ilustração do método Campos Potenciais. (a) Espaço de configuração com dois obstáculos representado por uma grade. (b) As regiões da grade ocupadas pelos obstáculos são consideradas como obstáculos, expandindo a região  $C_{obs}$ . (c) Caminho livre de obstáculos gradado. (d) Região representando o campo potencial gradado.

O planejamento é realizado de forma implícita onde a configuração do robô é deslocada incrementalmente na direção de uma força artificial  $F(q)$ , sendo que, as ações de controle recebidas pelo robô são geradas a partir dessa força que segue o gradiente

negativo do campo potencial, isto é:

$$F(q) = -\nabla U(q). \quad (6)$$

O método Campo Potencial funciona como um método de otimização baseado no gradiente que busca por um mínimo global, nesse caso, a configuração final. A região mostrada na Figura 4.9(d) ilustra o espaço de busca a ser minimizado.

#### 4.4. Métodos de Planejamento Baseados em Amostragem

Os métodos de Planejamento Baseados em Amostragem (*Sampling-Based Path Planning*) ou Planejamento Probabilístico são estratégias que surgiram inicialmente durante a década de 1990, capazes de resolver eficientemente problemas de planejamento em altas dimensões através de uma representação aproximada do espaço de configuração, o que permite uma redução considerável do custo computacional associado [LaValle 2006].

O planejamento probabilístico tem como principal característica a possibilidade de resolver um problema de planejamento sem a necessidade da computação explícita da região  $C_{obs}$ , diferentemente dos métodos da taxonomia clássica. Nessa estratégia, o espaço de configurações é representado de forma aproximada por meio de amostras aleatórias, normalmente utilizando uma determinada distribuição de probabilidade. Os pontos amostrados são conectados uns aos outros de maneira incremental, resultando, normalmente, num grafo que representa a região livre do  $C_{space}$ . As duas abordagens principais que representam os métodos probabilísticos, são eles o RRT [Lavelle 1998] e o PRM [Kavraki et al. 1996], comentados brevemente nas subseções seguintes:

##### 4.4.1. Árvore Aleatória de Exploração Rápida

A Árvore Aleatória de Exploração Rápida (*Rapidly-Exploring Random Tree - RRT*) é um algoritmo de planejamento de caminho probabilístico introduzido por [Lavelle 1998] como uma eficiente estratégia para realizar buscas em espaços de altas dimensões. De forma geral, o RRT constrói uma árvore que se expande por  $C_{livre}$  partindo da configuração  $q_{inicial}$  em direção à configuração  $q_{final}$ , evitando obstáculos, através da amostragem de nós aleatoriamente pelo espaço livre por meio de uma distribuição polarizada. O funcionamento do RRT básico pode ser observado na Figura 4.10.

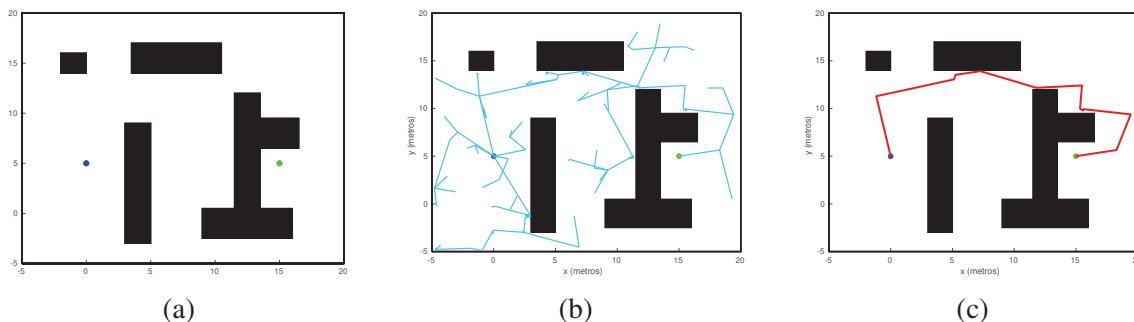


Figura 4.10: Planejamento de caminho com RRT. (a) Mapa representado no espaço de configurações. (b) Árvore aleatória formada. (c) Caminho encontrado a partir da árvore.

Nesse método, uma configuração auxiliar  $q_{aux}$  é amostrada aleatoriamente em  $C_{livre}$ . Considerando que a árvore já tenha alguns nós, é realizada uma busca para encontrar o nó  $q_{prox}$  mais próximo de  $q_{aux}$ . Se for possível conectar  $q_{prox}$  e  $q_{aux}$  através de um segmento de linha,  $q_{aux}$  se torna o novo nó da árvore. Se não for possível, é calculado um segmento de linha até a configuração livre mais próxima de  $q_{aux}$ , onde não haja colisão. Esse procedimento ocorre até que seja possível conectar as configurações inicial e final por meio da árvore gerada.

#### 4.4.2. Mapa de Rotas Probabilístico

O Mapa de Rotas Probabilístico (Probabilistic Roadmap) RPM é uma técnica de planejamento baseado em amostragem com características similares ao RRT. Introduzido por [Kavraki et al. 1996], o PRM visa representar a região  $C_{livre}$  de maneira aproximada através de um grafo unidirecional, enquanto que o RRT constrói uma árvore para o mesmo fim. A Figura 4.11 apresenta um exemplo utilizando o PRM.

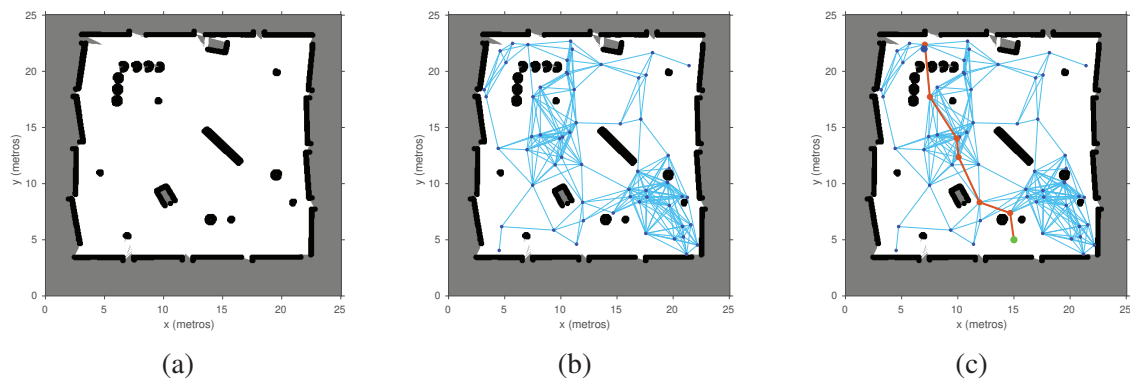


Figura 4.11: Ilustração do processo de planejamento por PFM. (a) Um mapa calculado. (b) Grafo formado no espaço livre. (c) Menor caminho extraído do grafo. Imagens de simulação obtida em [Matlab 2019]

Nesse método, um grafo é formado por meio de algumas configurações amostradas aleatoriamente pelo espaço livre. A partir de nós sementes, um grafo é montado, buscando conectar as sementes. Quanto mais pontos amostrados, maior será a probabilidade de cobertura de  $C_{livre}$  pelo grafo. Por meio de uma estratégia de busca, é possível encontrar um caminho factível pelo grafo.

#### 4.4.3. O Método da Espuma Probabilística

O método da Espuma Probabilística (*Probabilistic Foam Method* - PFM), introduzido por [Silveira and Alsina 2016], é um método de planejamento de caminho baseado em amostragem, desenvolvido com o objetivo de garantir uma região livre de obstáculos para que o robô possa realizar manobras seguras. Nesse método, uma região chamada bolha formada por um conjunto de configurações livres e limitadas por uma superfície definida como uma  $(n-1)$ -esfera. As bolhas se expandem no espaço livre de configurações sobre a superfície de outras bolhas, formando uma estrutura denominada espuma probabilística (organizada como uma árvore de busca) que é capaz de realizar uma cobertura aproximada do espaço livre.

A espuma se propaga no espaço livre a partir da configuração inicial, onde uma bolha é expandida até o obstáculo mais próximo. Um conjunto de bolhas (filhas) são expandidas na superfície da bolha inicial (bolha mãe), posicionadas de maneira aleatória, concluindo a primeira geração. Uma nova bolha mãe é selecionada dentre as filhas expandidas anteriormente, dentro de um ciclo iterativo que termina quando uma bolha engloba a configuração final (Figura 4.12(b)). Dessa maneira, por meio de uma busca partindo da bolha final (bolha que englobou  $q_{final}$ ) até a bolha inicial, seguindo o grau de parentesco das bolhas, é possível encontrar uma estrutura chamada *rosário* que formada um conjunto de bolhas sobrepostas a fim de extrair um caminho livre de obstáculos (Figura 4.12(c)).

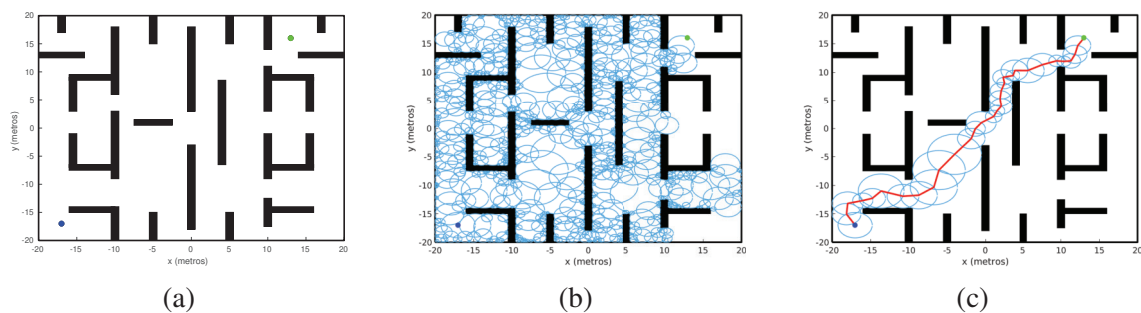


Figura 4.12: Planejamento por PFM. (a) Mapa de um labirinto. (b) Espuma probabilística formada. (c) Rosário extraído da espuma e caminho encontrado.

Esse método é ideal para aplicações em robótica assistiva onde existe a necessidade de garantir um maior grau de segurança nos movimentos. Um exemplo de aplicação pode ser observado em [Nascimento et al. 2018]b onde o PFM foi aplicado no planejamento dos movimentos de uma órtese ativa para os membros inferiores com o objetivo de transpor um obstáculo. Na Figura 4.13(a) é possível observar a região de obstáculo e uma espuma cobrindo o espaço livre. As variáveis de configuração são o ângulo da junta do quadril, o ângulo da junta do joelho e o movimento horizontal do quadril. Na Figura 4.13(b) o rosário é extraído da espuma. Na Figura 4.13(c) um caminho é calculado através do rosário e, finalmente, na Figura 4.13(d) é mostrado o movimento de transposição de obstáculo realizado para o caminho planejado.

Algumas variantes dessa técnica foram desenvolvidas como mostra o trabalho de [Nascimento et al. 2018]a onde o PFM original foi modificado, aplicando uma nova estratégia de propagação da espuma, inspirada no processo de expansão da árvore aleatória do RRT. O principal objetivo dessa estratégia foi a geração de caminhos mais curtos.

## 4.5. Ferramentas para Planejamento

Nesta seção serão apresentadas duas poderosas ferramentas para auxiliar no processo de planejamento de caminho para robôs autônomos. A primeira é uma biblioteca que contém a implementação de vários métodos de planejamento baseados em amostragem que estão no estado da arte. A segunda é uma *framework* que engloba diversas estratégias para auxiliar na programação de robôs autônomos.

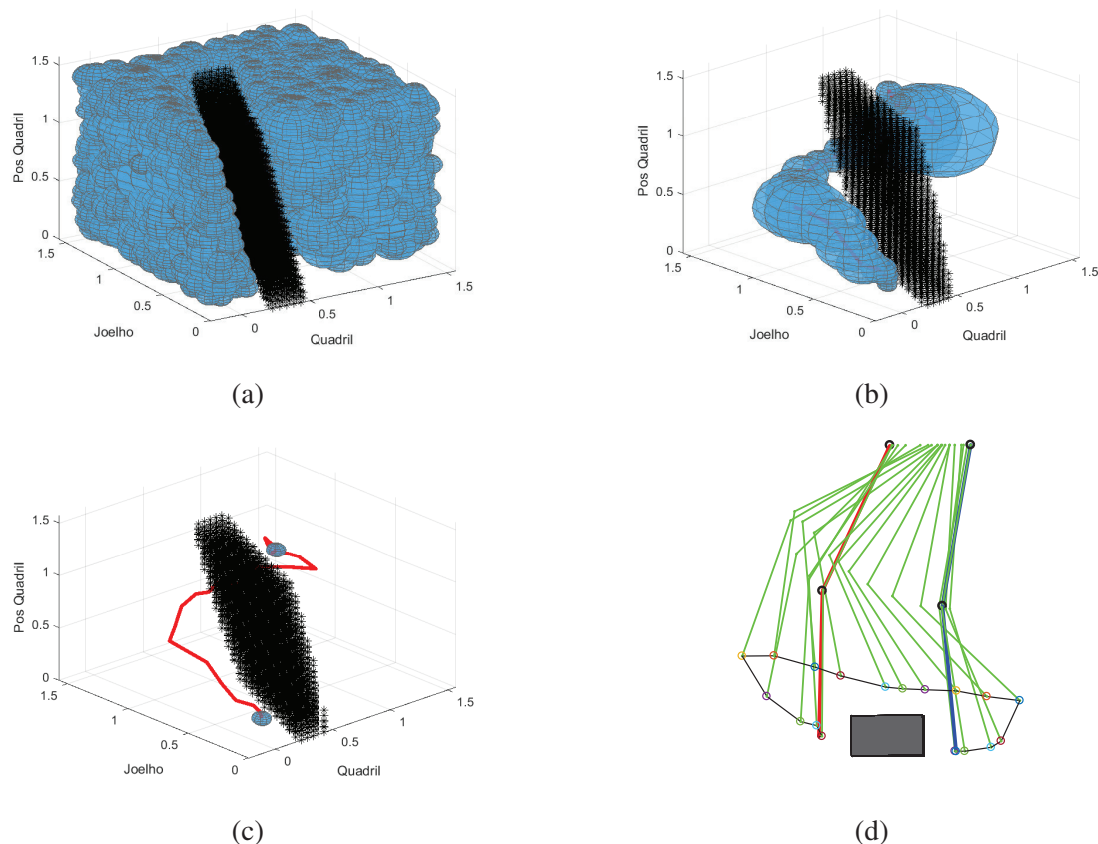


Figura 4.13: Simulação do planejamento de caminho para uma transposição de obstáculos por uma órtese ativa. (a) Espuma probabilística formada. (b) Rosário extraído da espuma. (c) Caminho encontrado. (d) Movimento da perna executado. [Nascimento et al. 2018]b

#### 4.5.1. A biblioteca OMPL

A biblioteca OMPL (*Open Motion Planning Library*) [OMPL 2019] consiste de um conjunto de métodos de planejamento de caminho baseados em amostragem que estão no estado da arte. É possível desenvolver aplicações utilizando métodos de planejamento de caminho como RRT e PRM, por exemplo, podendo ser implementados em linguagens como C++ e Python. Além disso, a biblioteca OMPL conta com classes para trabalhar com análise de colisões.

Além da biblioteca (conjunto das classes), é possível instalar uma aplicação com interface gráfica OMPL.app, onde é possível testar visualmente os métodos de planejamento com ambientes que simulam problemas clássicos de planejamento. A Figura 4.14 mostra a interface da aplicação OMPL.APP.

Nessa aplicação é possível abrir um ambiente para a simulação e um robô para realizar alguma tarefa nesse ambiente. Nas aba *Problem* é possível configurar o problema que você deseja simular. É possível selecionar o tipo de robô, definir a pose inicial e a pose final do robô dentro do ambiente. Na segunda aba *Planner* é possível selecionar qual estratégia de planejamento será utilizada, além de realizar configurações de parâmetros dos planejadores. Apenas com esses ajustes já será possível realizar uma simulação. A

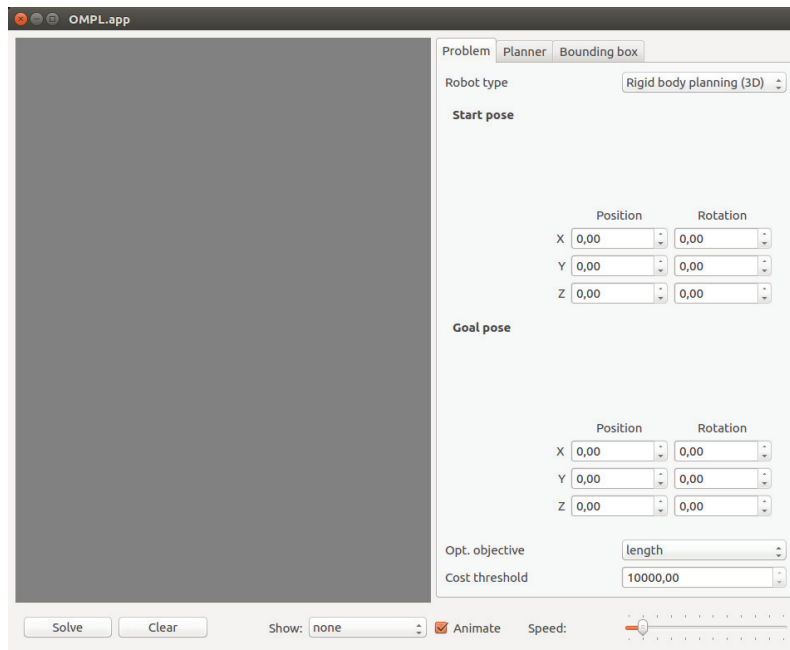


Figura 4.14: Interface gráfica inicial do OMPL.app.

Figuras 4.15-4.17 mostra uma simulação realizada com o OMPL.app para o problema clássico de planeamento do *carregador de piano*.



Figura 4.15: Ambiente de simulação iniciado. As configurações inicial e final são representadas pelos pianos.

Esse ambiente de apartamento está disponível para testes com todos os métodos de planeamento disponíveis na biblioteca. Além disso, é possível implementar um método de planeamento externo utilizando a biblioteca OMPL e testar utilizando a ferramenta OMPL.app com esse ou com outros ambientes.

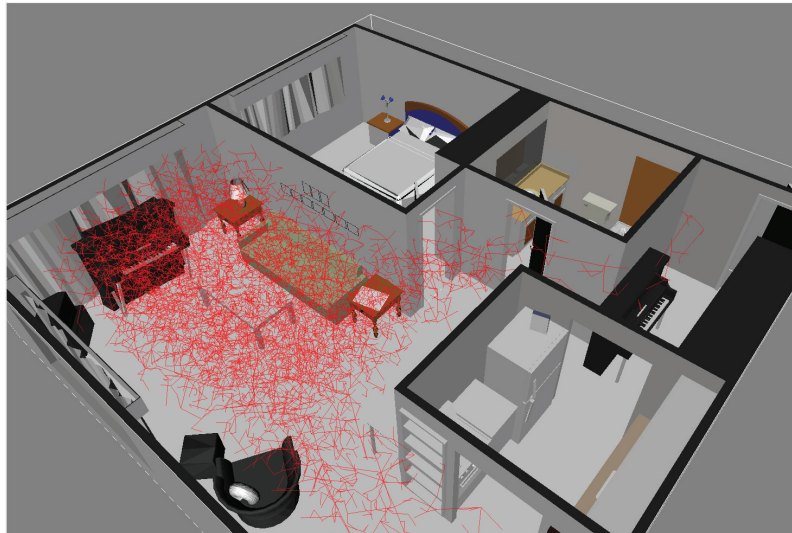


Figura 4.16: Árvore aleatória calculada partindo do piano inicial (ao fundo) para o piano final (à direita).

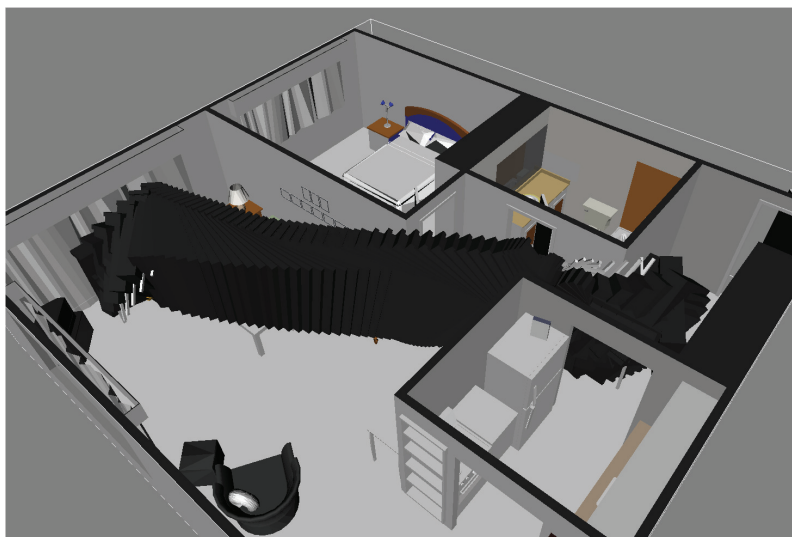


Figura 4.17: Planejamento realizado. Conjunto de configurações necessárias que o piano deve assumir para ser deslocado da configuração inicial até a configuração final.

#### 4.5.2. A *framework* MoveIt!

A *framework* MoveIt [MoveIt 2019] é uma ferramenta poderosa que engloba uma série de bibliotecas e aplicações para realizar projetos simples aos mais complexos de robótica autônoma. Essa *framework* é executada dentro do sistema ROS (*Robot Operating System*), uma completa plataforma para desenvolvimento de robôs, que fornece funcionalidades de um sistema operacional.

A *framework* MoveIt possibilita trabalhar com visão computacional, planejamento de caminho (fazendo uso de bibliotecas como OMPL), sistemas para cálculo de colisão, biblioteca para cálculo de cinemática, controle, entre outros. A Figura 4.18 mostra a



interface do sistema de visualização 3D Rviz, com a simulação de um braço robótico Panda.

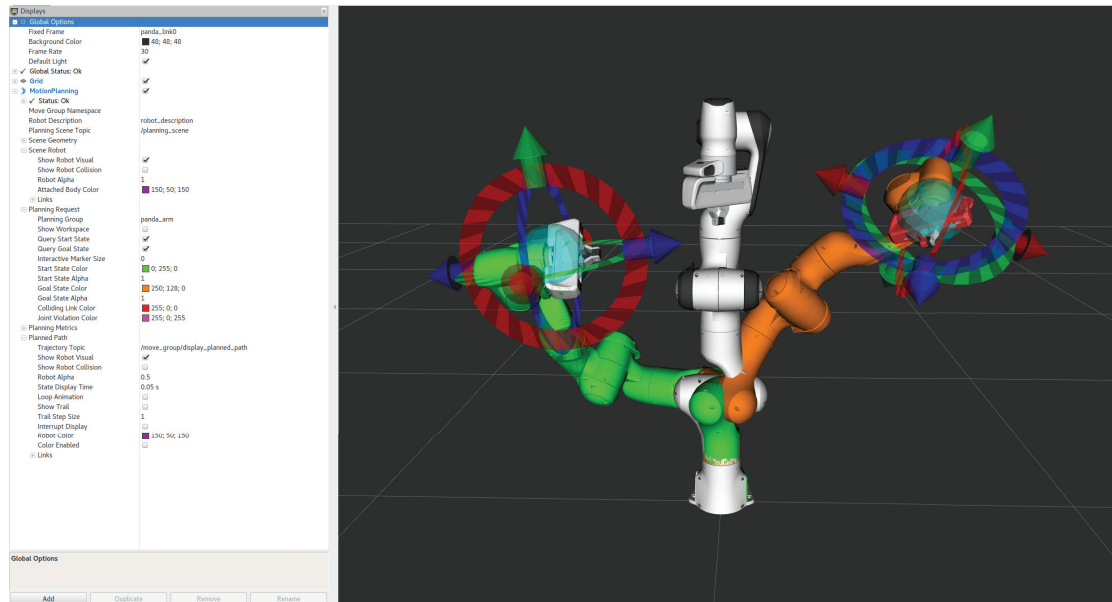


Figura 4.18: Fluxograma com os problemas fundamentais da robótica autônoma.

Na imagem, o braço robótico está sendo preparado para uma tarefa de planejamento. A imagem esverdeada do robô representa a sua configuração inicial e a imagem alaranjada do robô está representando a configuração final definida pelo usuário.

## 4.6. Conclusões

Neste documento foi apresentada uma introdução sobre o problema do planejamento de caminho para sistemas robóticos autônomos. Como mostrado, planejar caminhos livre de obstáculos é uma etapa fundamental no desenvolvimento de sistemas robóticos onde se necessita autonomia na realização de tarefas. Entretanto, é necessária uma análise do problema robótico a fim para definir uma estratégia de planejamento a ser implementada para obtenção de melhores resultados.

Para problemas mais simples, com sistemas robóticos de dois graus de liberdade (2 DoF), por exemplo, os métodos clássicos são capazes de realizar o planejamento de forma ideal. Porém, para problemas mais complexos, os métodos baseados em amostragem são os mais indicados. Além disso, é necessário analisar a natureza do problema, pois se o problema exige caminhos mais seguros, como é o caso de robôs assistivos, é indicado métodos que calculam não só caminhos factíveis, mas que também garantam regiões de livre manobrabilidade.

Finalmente, a fim de facilitar o desenvolvimento e implementação de sistemas robóticos autônomos, algumas bibliotecas e *frameworks* estão disponíveis gratuitamente para uso. Além de facilitar o desenvolvimento, essas ferramentas também ajudam no processo de ensino e aprendizagem de conceitos da robótica autônoma.

#### 4.7. Currículo dos autores

**Luís Bruno Pereira do Nascimento** possui Bacharelado em Ciência da Computação pela Universidade Estadual do Piauí (UESPI) e Mestrado em Engenharia Elétrica e de Computação pela Universidade Federal do Ceará (UFC). Atualmente é aluno de doutorado pelo Programa de Engenharia Elétrica e de Computação (PPgEEC) pela Universidade Federal do Rio Grande do Norte (UFRN). Suas áreas de interesse incluem robótica autônoma, otimização e aprendizado de máquina.

**Diego da Silva Pereira** possui graduação em Redes de Computadores pelo Instituto Federal do Rio Grande do Norte (IFRN), Mestrado em Ciência da Computação pela Universidade Estadual do Rio Grande do Norte (UERN) e está em doutoramento no Programa de Pós-graduação em Engenharia Elétrica e de Computação (PPgEEC) pela UFRN. Atualmente é professor no IFRN com pesquisas na área de redes de comunicação sem fio aplicadas à sistemas robóticos autônomos.

**Vitor Gaboardi dos Santos** possui graduação em Engenharia Elétrica na Universidade Federal do Rio Grande do Norte (UFRN) com período sanduíche na University of Kansas (USA). Atualmente é aluno de Mestrado no Programa de Pós-graduação em Engenharia Mecatrônica pela UFRN com pesquisas relacionadas a visão computacional aplicada à robótica assistiva.

**Daniel Henrique Silva Fernandes** possui Bacharelado em Engenharia Mecatrônica pela Universidade Federal do Rio Grande do Norte (UFRN). Atualmente é aluno de Mestrado no Programa de Pós-graduação em Engenharia Mecatrônica pela UFRN com pesquisas relacionadas a robótica assistiva e visão computacional.

**Pablo J. Alsina** possui graduação em Engenharia Elétrica (1987), mestrado na área de controle de motores de indução (1991) e doutorado em Engenharia Elétrica com tema em controle de manipuladores robóticos (1996), pela Universidade Federal da Paraíba. Atualmente é professor titular do Departamento de Engenharia de Computação e Automação (DCA). É professor nos Programas de Pós-Graduação em Engenharia Mecatrônica (PPGEM) e em Engenharia Elétrica e de Computação (PPgEEC) da Universidade Federal do Rio Grande do Norte (UFRN), onde é chefe do Laboratório de Robótica. Desenvolve pesquisas em robótica, nas áreas de controle, planejamento e percepção robótica, com aplicações em robótica assistiva, robótica móvel e Veículos Aéreos Não Tripulados.

#### Referências

- [Al-Bluwi et al. 2012] Al-Bluwi, I., Siméon, T., and Cortés, J. (2012). Motion planning algorithms for molecular simulations: A survey. *Computer Science Review*, 6(4):125–143.
- [Dijkstra 1959] Dijkstra, E. W. (1959). A Note on Two Problems in Connexion with Graphs. *Numer. Math.*, 1(1):269–271.
- [Dirichlet 1850] Dirichlet, G. L. (1850). Über die Reduction der positiven quadratischen Formen mit drei unbestimmten ganzen Zahlen. *Journal für die reine und angewandte Mathematik*, 40:209–227.

- [Edelkamp and Plaku 2014] Edelkamp, S. and Plaku, E. (2014). Multi-goal motion planning with physics-based game engines. In *2014 IEEE Conference on Computational Intelligence and Games*, pages 1–8.
- [Even 1979] Even, S. (1979). *Graph Algorithms*. W. H. Freeman & Co., New York, NY, USA.
- [Hart et al. 1968] Hart, P. E., Nilsson, N. J., and Raphael, B. (1968). A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107.
- [Hart et al. 1972] Hart, P. E., Nilsson, N. J., and Raphael, B. (1972). Correction to "A Formal Basis for the Heuristic Determination of Minimum Cost Paths". *SIGART Bull.*, (37):28–29.
- [Kavraki et al. 1996] Kavraki, L., Svestka, P., Latombe, J.-C., and Overmars, M. (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566–580.
- [Khatib 1986] Khatib, O. (1986). Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. *The International Journal of Robotics Research*, 5(1):90–98.
- [Latombe 1991] Latombe, J.-C. (1991). *Robot Motion Planning*. Springer US, Boston, MA.
- [Latombe 2012] Latombe, J.-C. (2012). *Robot motion planning*, volume 124. Springer Science & Business Media.
- [Lavalle 1998] Lavalle, S. M. (1998). Rapidly-exploring random trees: A new tool for path planning. Technical report, Computer Science Dept., Iowa State University.
- [LaValle 2006] LaValle, S. M. (2006). *Planning Algorithms*. Cambridge University Press, New York, NY, USA.
- [Lee 1961] Lee, C. Y. (1961). An Algorithm for Path Connections and Its Applications. *IRE Transactions on Electronic Computers*, EC-10(3):346–365.
- [Lozano-Perez 1983] Lozano-Perez, T. (1983). Spatial Planning: A Configuration Space Approach. *IEEE Transactions on Computers*, C-32(2):108–120.
- [Lozano-Pérez and Wesley 1979] Lozano-Pérez, T. and Wesley, M. A. (1979). An Algorithm for Planning Collision-free Paths Among Polyhedral Obstacles. *Commun. ACM*, 22(10):560–570.
- [Matlab 2019] Matlab (2019). Path planning in environments of different complexity. *Matlab® Robotics System Toolbox Documentation*. Online em: <https://www.mathworks.com/help/robotics/examples/path-planning-in-environments-of-difference-complexity.html>; acessado em 16-Agosto-2019.

- [Molina 2014] Molina, L. (2014). *Planejamento de Movimento Para Robôs Móveis Baseado na Condição de Horizonte Continuado (CHC)*. Doutorado, Universidade Federal de Campina Grande, Pernambuco, Brasil.
- [MoveIt 2019] MoveIt (2019). MoveIt Motion Planning Framework. Online em: <https://moveit.ros.org>; acessado em 18-Agosto-2019.
- [Murphy 2000] Murphy, R. R. (2000). *Introduction to AI Robotics*. Intelligent robotics and autonomous agents. MIT Press, Cambridge, Mass. OCLC: 247251178.
- [Nascimento et al. 2018] Nascimento, L. B. P., Pereira, D. S., Alsina, P. J., Silva, M. R., Fernandes, D. H. S., Roza, V. C. C., and Sanca, A. S. (2018). Goal-biased probabilistic foam method for robot path planning. pages 199–204, Torres Vedras, Portugal. IEEE.
- [Nascimento et al. 2018] Nascimento, L. B. P., Silva Pereira, D., Sanca Sanca, A., Silva Eugenio, K. J., Silva Fernandes, D. H., Javier Alsina, P., Valério Araujo, M., and Rabello Silva, M. (2018). Safe path planning based on probabilistic foam for a lower limb active orthosis to overcoming an obstacle. In *2018 Latin American Robotic Symposium, 2018 Brazilian Symposium on Robotics (SBR) and 2018 Workshop on Robotics in Education (WRE)*, pages 413–419.
- [Nilsson 1969] Nilsson, N. J. (1969). A Mobile Automaton: An Application of Artificial Intelligence Techniques. Technical report, Sri International Menlo Park Ca Artificial Intelligence Center.
- [OMPL 2019] OMPL (2019). The Open Motion Planning Library. *Kavraki Lab*. Online em: <https://ompl.kavrakilab.org>; acessado em 18-Agosto-2019.
- [Ottoni 2000] Ottoni, G. d. L. (2000). Planejamento de trajetórias para robôs móveis. *Projeto de Graduação em Engenharia de Computação–FURG, Rio Grande*.
- [Pearl 1984] Pearl, J. (1984). *Heuristics: Intelligent Search Strategies for Computer Problem Solving*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.
- [Russell and Peter Norvig 1995] Russell, S. J. and Peter Norvig (1995). *Artificial Intelligence: A Modern Approach*, volume 25. Prentice-Hall, Englewood Cliffs, New Jersey, 1 edition.
- [Sakemoto 2017] Sakemoto, M. (2017). Build an autonomous mobile robot with the Intel® RealSense™ Camera, ROS\*, and SAWR. *Intel® Software*. Online em: <https://software.intel.com/en-us/articles/build-an-autonomous-mobile-robot-with-the-intel-realsense-camera-ros-and-sawr>; acessado em 16-Agosto-2019.
- [Siegwart and Nourbakhsh 2004] Siegwart, R. and Nourbakhsh, I. R. (2004). *Introduction to Autonomous Mobile Robots*. Intelligent robotics and autonomous agents. MIT Press, Cambridge, Mass, 1 edition. OCLC: ocn649700153.
- [Silveira and Alsina 2016] Silveira, Y. S. and Alsina, P. J. (2016). A New Robot Path Planning Method Based on Probabilistic Foam. In *2016 XIII Latin American Robotics Symposium and IV Brazilian Robotics Symposium (LARS/SBR)*, pages 217–222.

- [Tarjan 1971] Tarjan, R. (1971). Depth-first search and linear graph algorithms. In *12th Annual Symposium on Switching and Automata Theory (Swat 1971)*, pages 114–121.
- [Udupa 1977] Udupa, S. M. (1977). *Collision Detection and Avoidance in Computer Controlled Manipulators*. Doutorado, Dept. of Electrical Engineering, California Institute of Technology, California.