



UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA MECATRÔNICA



2º RELATÓRIO: Planejamento de mapa e inclusão do espaço de configuração com obstáculos

CoppeliaSim 4.1.0 (V-REP)

Iago Lucas Batista Galvão
Jhonat Heberston Avelino de Souza
Thiago de Araújo Brito

2º Relatório: Planejamento de mapa e inclusão do espaço de configuração com obstáculos

CoppeliaSim 4.1.0 (V-REP)

Terceira entrega de três do 2º Relatório da disciplina EGM0007 - Sistemas Robóticos Autônomos do Mestrado em Engenharia Mecatrônica da Universidade Federal do Rio Grande do Norte (UFRN), referente a nota parcial da segunda unidade.

Professor(a): Pablo Javier Alsina.

Sumário

	Páginas
1 INTRODUÇÃO	4
2 OBJETIVO	4
3 METODOLOGIA	4
4 PLANEJADOR DE CAMINHOS	7
4.1 MAPA E ESPAÇO DE CONFIGURAÇÃO	7
4.2 GRAFOS DE VISIBILIDADE	11
4.3 CAMPOS DE POTENCIAIS	13
4.4 MAPA PROBABILÍSTICO DE ROTAS	16
5 CONCLUSÃO	17
REFERÊNCIAS	18

1 INTRODUÇÃO

Com o aumento do poder computacional, houve um crescimento no número de pesquisas na área de robótica e, atualmente, o pesquisador é capaz de realizar simulações com qualidade em cenários 2D/3D com várias ferramentas revolucionárias. Os simuladores possibilitam a realização de experimentos, sem a necessidade de construir o *hardware* do robô para desempenhar vários tipos de testes. Como tal desenvolvimento custa caro, esta alternativa possibilita a pesquisa em robôs mais viável e difundida.

Existem muitos *softwares* de simulação robótica disponíveis como, por exemplo, Open HPR, Gazebo, Webots, V-REP etc (1). A plataforma utilizada nesse estudo foi a *Virtual Robot Experimentation Platform* (V-REP), pois atende muitos requisitos, como estrutura de simulação versátil e escalável, arquitetura de controle distribuída, controle por *Script*, *Plugins*, ou API cliente remota entre outras funcionalidades (1), ademais, esta é uma das mais empregadas nos estudos robóticos de todo o mundo.

A versatilidade do simulador contribui para diferentes plataformas robóticas e várias aplicações, não atentando-se somente ao fato de simular robôs com resultados iniciais. Outrossim, isenta o gasto e a necessidade da aquisição do *hardware* para resultados mais realísticos e adaptáveis.

Na Primeira etapa foi criado um modelo simples de robô móvel com a modelagem cinemática, como discutida antes, também criado controladores de posição, seguidor de caminho e trajetória.

Nessa etapa, de planejamento de mapa, aonde criamos nosso espaço de configuração (robô como um ponto), incluindo os obstáculos poligonais na planta baixa e detecção de colisões, e projetar caminho a ser percorrido com o robô.

2 OBJETIVO

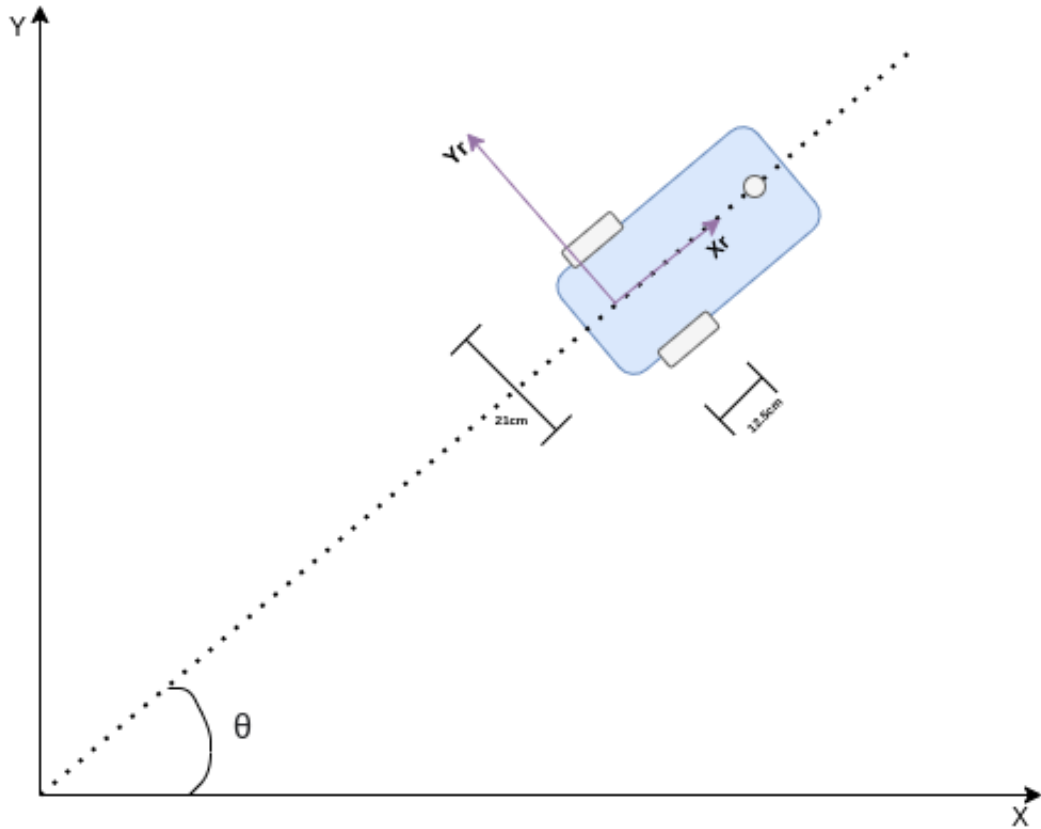
O propósito desta simulação é aplicar os conhecimentos teóricos aprendidos, desenvolvendo Planejadores de Caminhos para Robô Móvel que permitam ao mesmo executar movimentos especificados em espaço povoado de obstáculos, sem colidir com os mesmos. Nesta terceira entrega são implementados planejadores de caminhos baseados em grafos de visibilidade, campos potenciais e amostragem; os obstáculos modelados como polígonos e criado o espaço de configuração, onde é possível representar nosso robô como um ponto no mapa para detecção de colisões.

3 METODOLOGIA

Primeiramente a uma modelagem não holonômica do robô para projetar o controlador e trajeto foi calculado com os parâmetros do robô. Dessa forma, empregou-se os conceitos

aprendidos de modelagem cinemática e não holonômica de robôs, como demonstrada na Figura 1.

Figura 1. Modelagem do robô



Fonte: Autores.

Roda Direita:

- $\theta_d = -90^\circ$
- $\alpha_d = 180^\circ$
- $l_d = \frac{b}{2}$

Roda Esquerda:

- $\theta_e = 90^\circ$
- $\alpha_e = 0^\circ$
- $l_e = \frac{b}{2}$

Com base nessas informações podemos obter as equações derrapagem lateral e rolamento:

$$\begin{bmatrix} 1 & 0 & \frac{b}{2} \end{bmatrix} * ({}^iR_R(\theta))^T * q' = w_d * r_d \quad (1)$$

$$\begin{bmatrix} 1 & 0 & \frac{-b}{2} \end{bmatrix} * ({}^iR_R(\theta))^T * q' = w_e * r_e \quad (2)$$

Restrição de derrapagem lateral:

$$\begin{bmatrix} 0 & 1 & 0 \end{bmatrix} * ({}^iR_R(\theta))^T * q' = 0 \quad (3)$$

Ao agruparmos as Equações 1, 2 e 3 teremos nossa modelagem cinemática do robô:

$$\begin{bmatrix} 1 & 0 & \frac{b}{2} \\ 1 & 0 & \frac{-b}{2} \\ 0 & 1 & 0 \end{bmatrix} * ({}^iR_R(\theta))^T * q' = \begin{bmatrix} r_d & 0 \\ 0 & r_e \end{bmatrix} * \begin{bmatrix} w_d \\ w_e \end{bmatrix} \quad (4)$$

para encontrar q' :

$$\begin{bmatrix} X \\ Y \\ \theta \end{bmatrix} = \begin{bmatrix} \frac{\cos(\theta) * (r_d * w_d + r_e * w_e)}{2} \\ \frac{\sin(\theta) * (r_d * w_d + r_e * w_e)}{2} \\ \frac{(r_d * w_d - r_e * w_e)}{b} \end{bmatrix} \quad (5)$$

onde $r_d = r_e = 6,25cm$ e $b = 21cm$.

Além da posição e orientação, necessita-se do controle das velocidades linear e angular para concretizar o controlador da trajetória a ser seguida. Dessa forma podemos iniciar a modelagem dos controladores, os quais serão discutidos nos próximos tópicos.

Inicialmente utilizamos a API do V-Rep com Python para projetar os controladores, porém para essa segunda etapa que é para planejamento do mapa, teríamos que implementar funções do zero, a qual não é o principal intuito dessa disciplina, por isso optamos com mudar nossa principal ferramenta de implementação para MATLAB (MATrix LABoratory) trata-se de um software interativo de alta performance voltado para o cálculo numérico. O MATLAB integra análise numérica, cálculo com matrizes, processamento de sinais e construção de gráficos em ambiente fácil de usar onde problemas e soluções são expressos somente como eles são escritos matematicamente, ao contrário da programação tradicional, onde tem funções já implementadas a partir de tools box como *Robotics System Toolbox*.

Na primeira etapa do projeto foi dividido em três subetapas: a primeira foi a simulação do modelo cinemático, a segunda etapa foi gerado o caminho baseado em um polinômio de 3° aplicado pelo usuário e, a terceira, implementa os controladores cinemáticos do robô móvel.

Para a segunda etapa do projeto, objetivou-se o planejamento do mapa. Para isso, foram inseridos obstáculos no espaço de trabalho, os quais devem ser reconhecidos pela câmera e considerados pelo robô móvel ao planejar o caminho baseados em grafos e campos potenciais. Ademais, o caminho é mostrado na tela e o robô segue o mesmo, acionado pelo controlador seguidor de caminhos.

Por fim, na terceira etapa, foi adicionado o planejador de caminhos por amostragem, onde são criados pontos aleatórios no espaço de configuração e traçado os possíveis caminhos que não

colidem com os obstáculos.

4 PLANEJADOR DE CAMINHOS

Com as informações do modelo cinemático, é possível implementar o robô e realizar sua modelagem no *software* **V-REP**. No programa, é possível aplicar as dimensões do robô desejado no projeto junto com sua modelagem cinemática para que assim seja possível realizar os testes desejados. Os resultados desejados são mostrados em gráficos na tela. Além disso, há a possibilidade de exportar todos os dados da simulação, possibilitando seu uso onde for desejado. Para essa etapa do projeto, com o objetivo de planejar caminhos por diferentes métodos, foi necessário posicionar alguns obstáculos na plataforma de simulação para que os mesmos sejam evitados pelo robô. Com os obstáculos alocados nas posições escolhidas, foi então gerado um mapa a ser utilizado na geração dos caminhos. Para a geração do mapa, foi utilizado um sensor de visão posicionado na parte superior da plataforma com a captura virada para baixo, possibilitando assim a obtenção da imagem de todo o campo utilizado para movimentação do robô.

A planta baixa do nosso espaço junto com os obstáculos é ilustrada na Figura 2, onde em preto são os objetos a serem evitados.

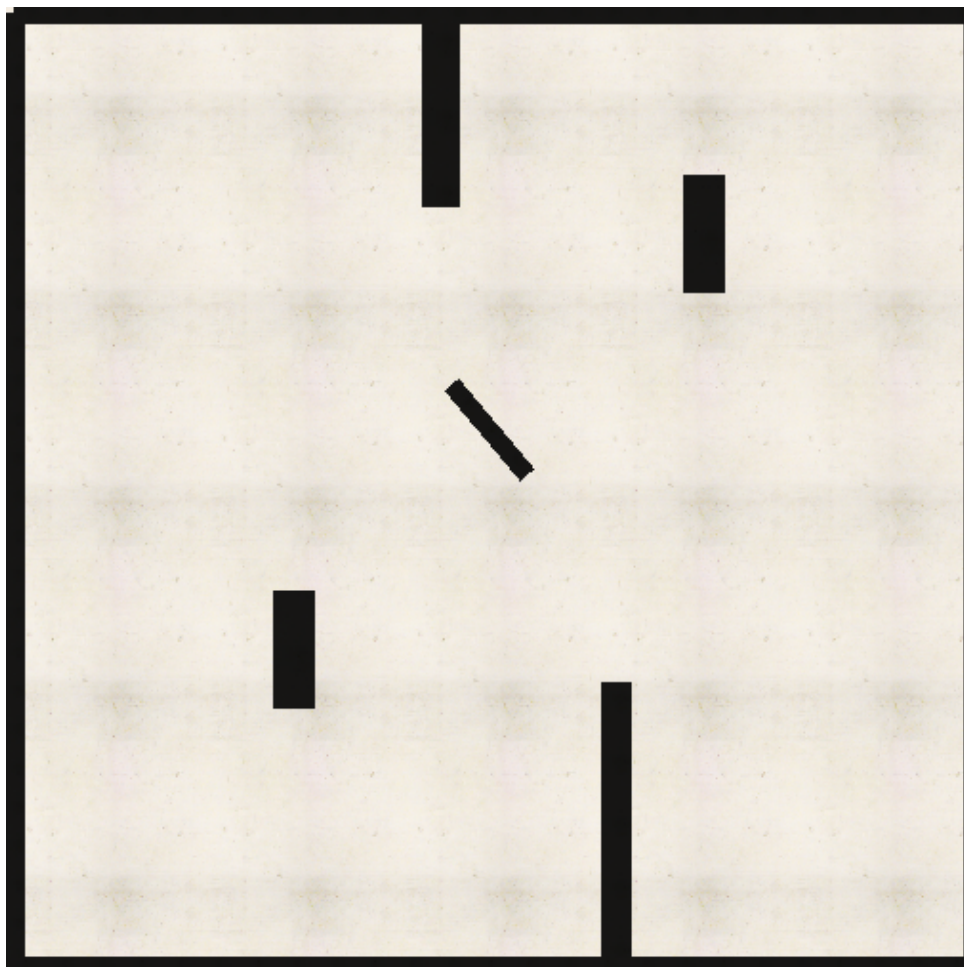
Segue o *link* do repositório no GitHub contendo todos os projetos do curso: <https://github.com/jhonathebers/robotic-systems>

4.1 MAPA E ESPAÇO DE CONFIGURAÇÃO

A imagem do mapa obtida conforme relatado no tópico anterior foi então utilizada para a geração do mapa a ser seguido pelo robô. Para que a imagem capturada possa ser utilizada como mapa, foram utilizadas funções do *software* **Matlab** para que a mesma fosse convertida em uma matriz de *pixels* com valores de acordo com a cor capturada da imagem. Com base nessa matriz, foi possível identificar os obstáculos distribuídos pela plataforma e assim gerar uma grade de ocupação binária, ou seja, o mapa binário a ser utilizado pelo robô. O mapa obtido é representado na Figura 3, onde pode ser observado que o mesmo possui dimensões de 5x5 metros com sua origem no centro (0,0) da imagem.

Após a obtenção do mapa, é então necessário a obtenção do espaço configuração que será utilizado para o deslocamento do robô. No espaço configuração onde o robô irá se deslocar, considera-se o robô como um ponto circular se deslocando entre os obstáculos. Sabendo disso, é então necessário a dimensão de raio de giro do robô para assim determinar o diâmetro do ponto a ser considerado no espaço configuração. O robô possui comprimento de *40 cm*, sendo essa sua maior dimensão. Portanto, pode-se adotar o robô como um ponto de *40 cm* de diâmetro se deslocando pelo mapa. Entretanto, a forma do robô não é modificada, a alteração é feita no próprio mapa. Para isso, o mapa tem seus obstáculos inflados, fazendo assim com que o robô não sobreponha nenhum obstáculo, ou seja, colisão. Para inflar o mapa, deve-se considerar as

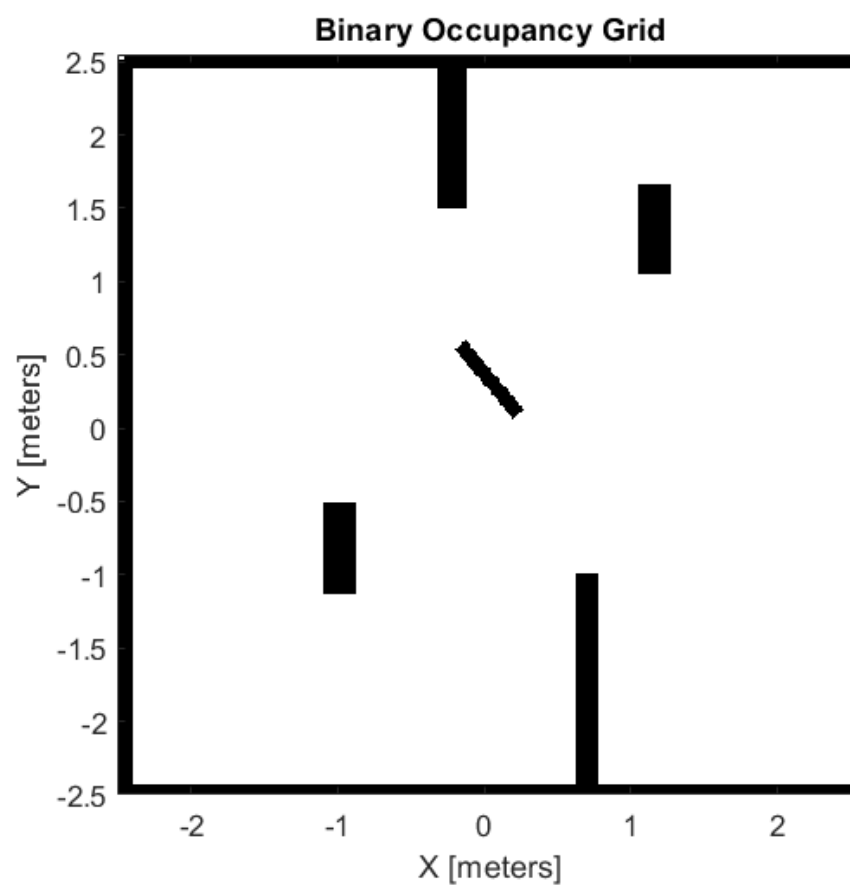
Figura 2. Mapa



Fonte: Autores.

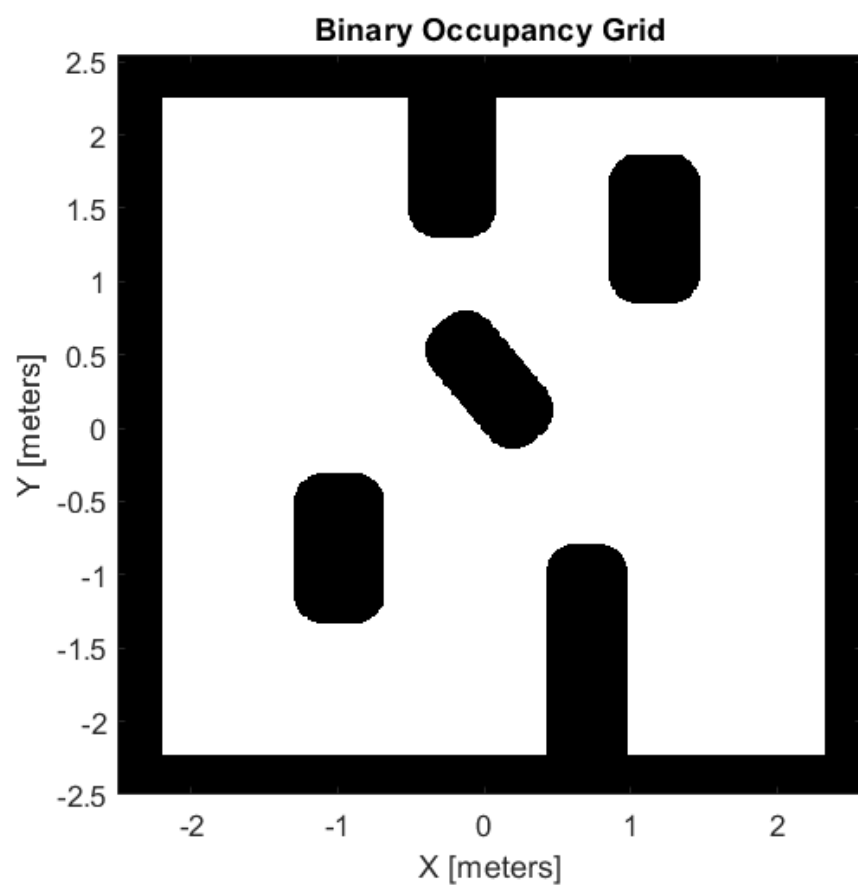
dimensões do robô, que no caso do trabalho aqui relatado foi adotado um raio de giro de *20 cm*, mas esse valor pode ser aumentado se desejar maior segurança contra colisão com os obstáculos. O espaço configuração é ilustrado na Figura 4.

Figura 3. *Mapa Binário*



Fonte: Autores.

Figura 4. *Mapa Binário Aumentado com Raio de Giro*



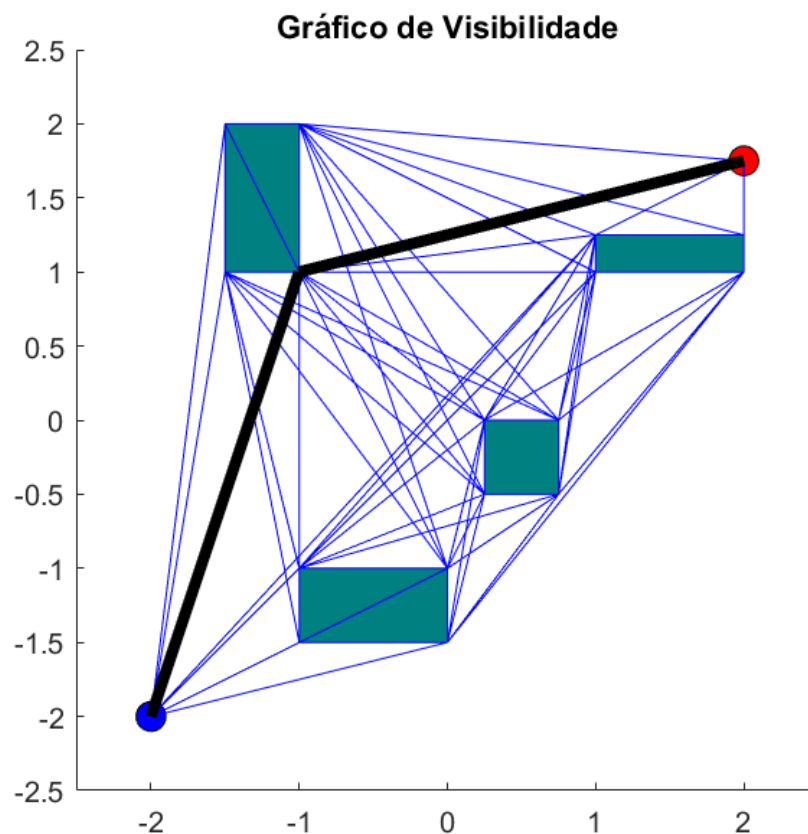
Fonte: Autores.

4.2 GRAFOS DE VISIBILIDADE

O primeiro método utilizado no projeto para geração de caminhos foi o grafos de visibilidade. Nesse método, retas são criadas a partir do ponto de origem ao ponto de destino, concatenado os vértices dos obstáculos. Ao mapear todas as retas que ligam os pontos aos vértices e estes entre si, o caminho seguido é determinado pela menor distância a ser percorrida, somando-se as retas traçadas.

Na Figura 5 está representado a imagem do planejamento por grafos, onde os círculos em verde e amarelo, representam respectivamente, os pontos inicial e final. As retas em azul, simbolizam os caminhos traçados pelos vértices e pontos; a linha em preto, ilustra o menor caminho a ser seguido, desviando dos obstáculos.

Figura 5. *Grafos de Visibilidade*



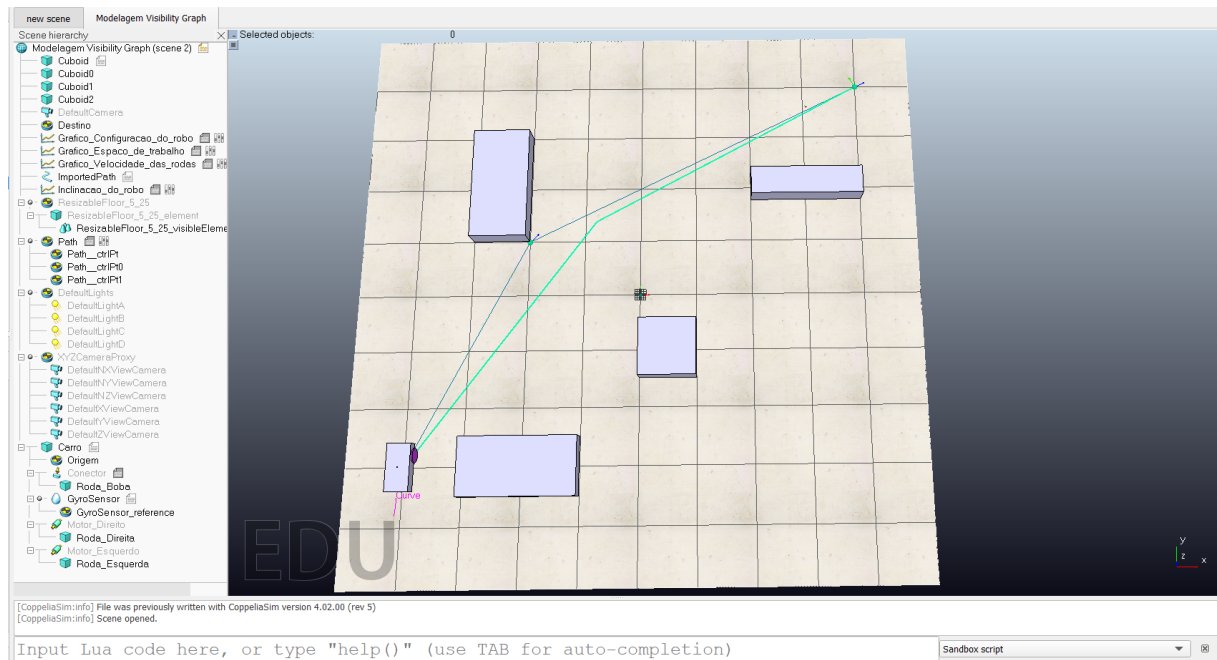
Fonte: Autores.

No final do código utilizado para geração do caminho por grafos, é possível gerar a matriz de pontos (caminho) a serem seguidos pelo robô e exportar do *MatLab*, é possível aplicar o controle de velocidades (seguidor de caminho) para simular o trajeto do robô no *V-REP*. Como foi utilizado um ponto para criação do caminho, necessitou-se de um ajuste nas dimensões dos obstáculos, equivalente ao raio de giro do robô utilizado no simulador. Com isso, os obstáculos

foram inflados em *40cm* (dobro do raio de giro para ajustar o "diâmetro") dos obstáculos, o que evita colisão do carro ao passar próximo ao vértice do obstáculo.

A Figura 6 demonstra a cena do simulador, com o caminho a ser seguido em azul e os objetos em cinzas, representam proporcionalmente os obstáculos ponderados no gerador de grafos.

Figura 6. *Grafos de Visibilidade no V-REP*

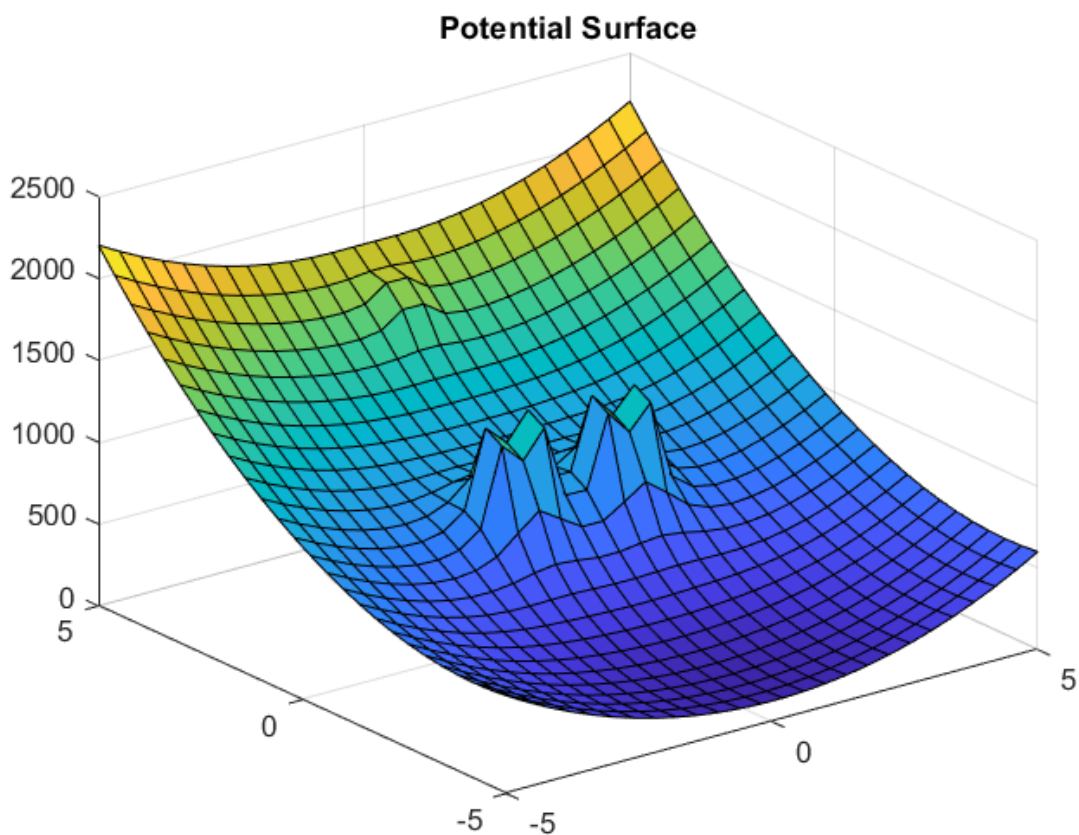


Fonte: Autores.

4.3 CAMPOS DE POTENCIAIS

Na segunda experiência, aplicou-se o método de planejamento de caminhos por campos potenciais. Neste, foram inseridos dois obstáculos idênticos, um ao lado do outro, de formato cilíndrico. O método consiste em "descer" da maior altitude para a menor (ponto inicial mais elevado do que o ponto final), evitando os obstáculos, os quais geram um campo potencial ao redor destes. Na Figura 7 é ilustrado o mapa da superfície gravitacional, onde parte em azul representa a área de posição final e os dois vales ao centro, os obstáculos cilíndricos.

Figura 7. Mapa de Superfície Gravitacional

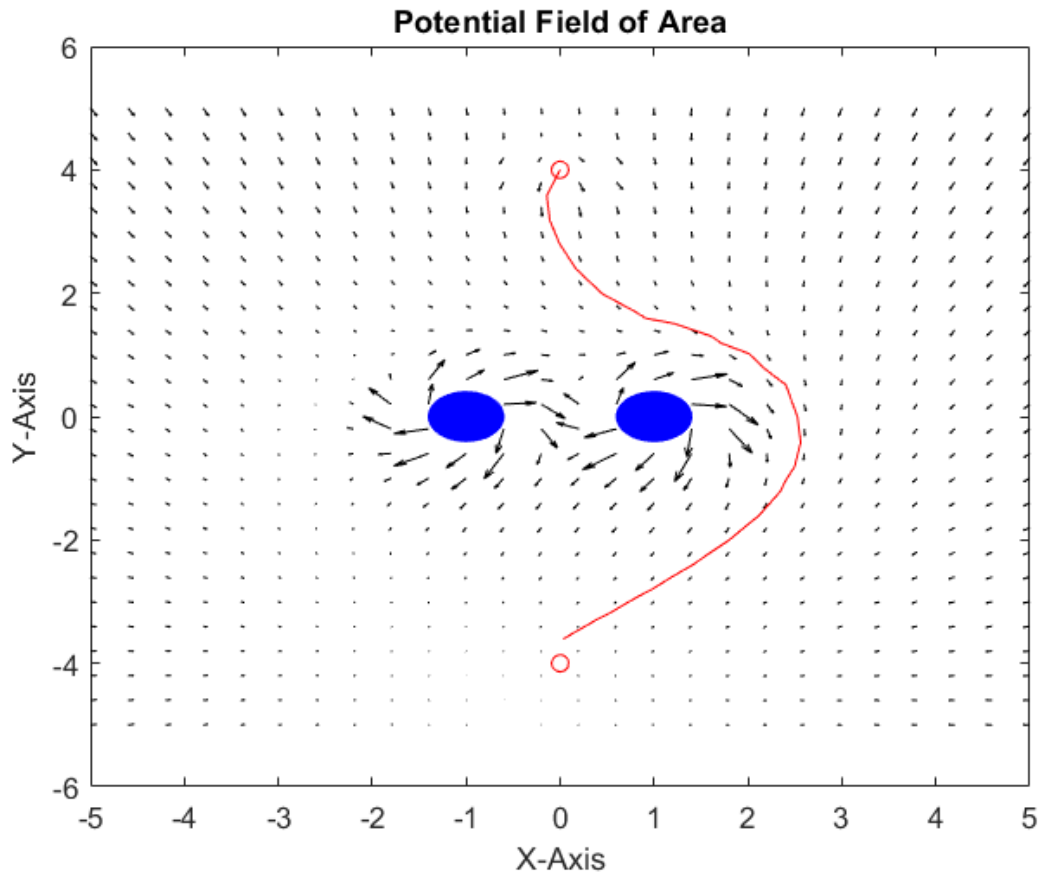


Fonte: Autores.

Assim, quanto mais próximo do objeto, maior será o vetor do campo gerado e o robô móvel deve locomover-se na menor distância necessária, desviando dos vetores de valores altos, próximos ao objeto. Dito isto, o robô percorre apenas vetores até o tamanho limitado, distanciando-se dos obstáculos de acordo com o formato destes. Na Figura 8 está representado o campo potencial com os vetores ilustrados pelas setas pretas (quanto maior a seta, maior a força do campo), os objetos cilíndricos em azul e a curva a ser seguida em vermelho.

Posteriormente, é importado para o simulador a curva a ser seguida pelo controle das velocidades linear e angular. Dessa maneira, o robô percorre o menor caminho até seu destino,

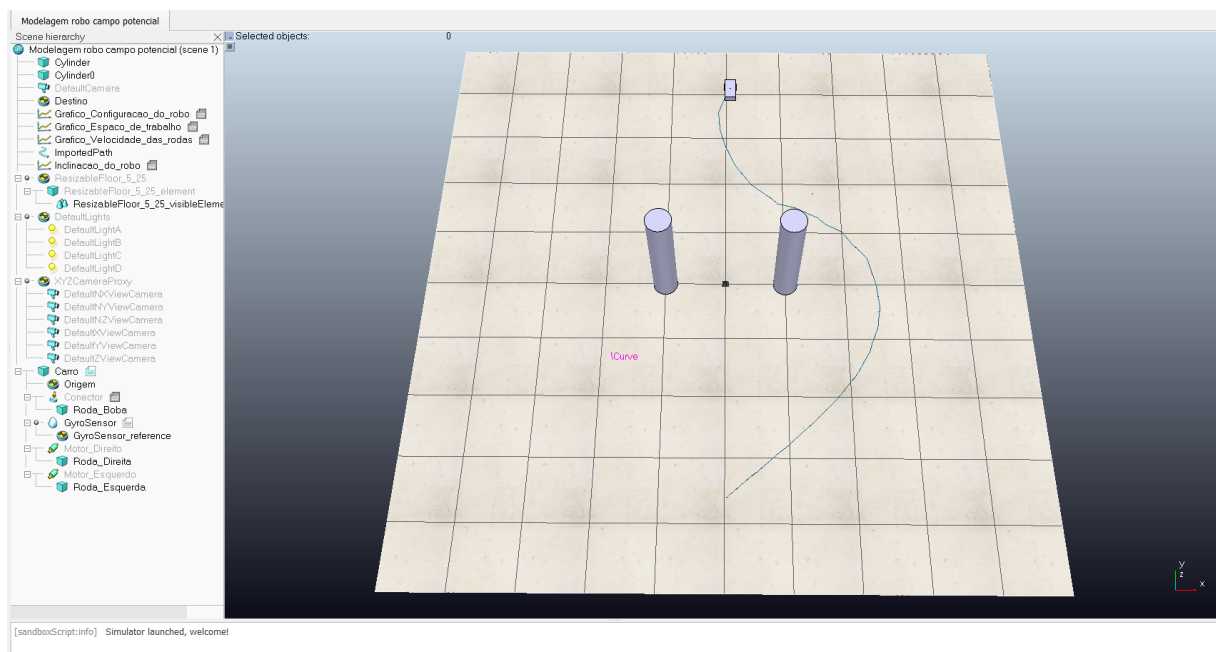
Figura 8. *Área do Campo Potencial*



Fonte: Autores.

nota-se que o mesmo não traçou a curva ao centro, pois o limite determinado pelo usuário foi alto, podendo assim, ajustar para adequar-se ao tamanho real do carro. A Figura 9 demonstra a cena do *V-REP* com o caminho traçado em azul, similar ao planejado.

Figura 9. *Campo Potencial no V-REP*



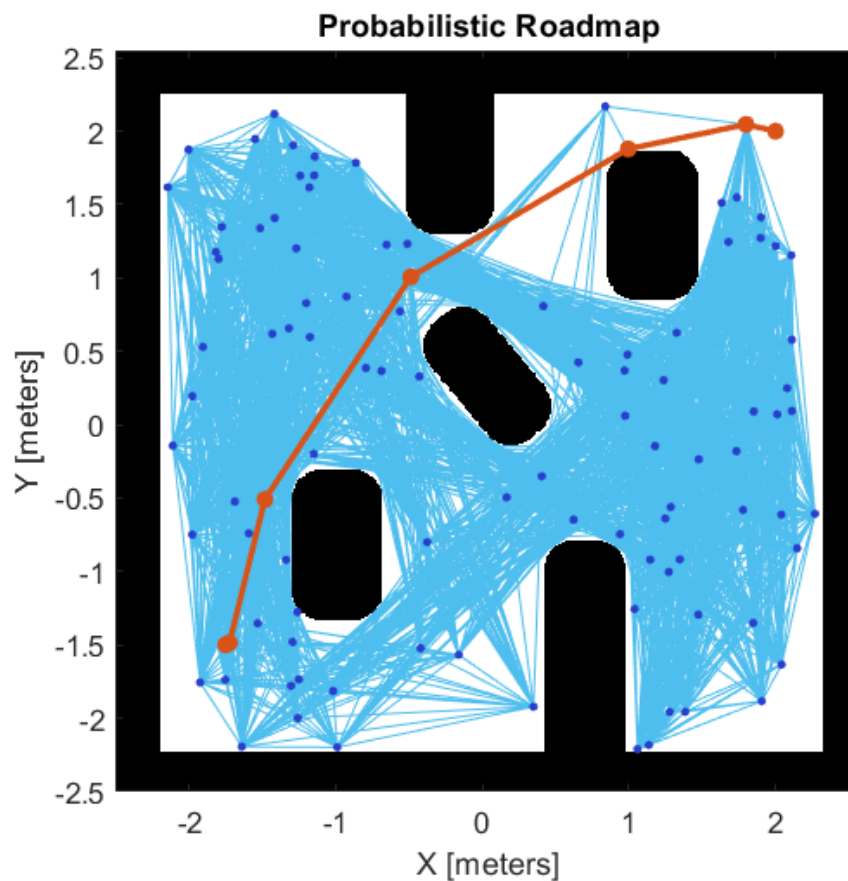
Fonte: Autores.

4.4 MAPA PROBABILÍSTICO DE ROTAS

Posteriormente, foi então obtido o caminho por mapa de rotas probabilísticas (PRM). Para encontrar o caminho entre o ponto de partida até o destino desejado foi definido a quantidade máxima de 200 nós, quantidade que pode ser alterada a depender da precisão desejada e capacidade de cálculo disponível. Uma maior quantidade de nós resulta em um caminho melhor a ser percorrido com uma menor distância. Por meio do mapa probabilístico de rotas é possível obter todas as rotas possíveis para trajetória do robô móvel no espaço de configuração adequado (parte branca da imagem binária aumentada).

Conhecendo todos os caminhos possíveis para locomoção, a menor trajetória a ser seguida pelo carro é definida e traçada no mapa. Na Figura 10, está demonstrado em azul, as possíveis rotas as quais evitam os obstáculos e, em vermelho, o caminho a ser seguido do ponto inicial (-1.7500, -1.5000) ao ponto final (2.0000, 2.0000).

Figura 10. Mapa com os Pontos de Locomoção e Trajetória

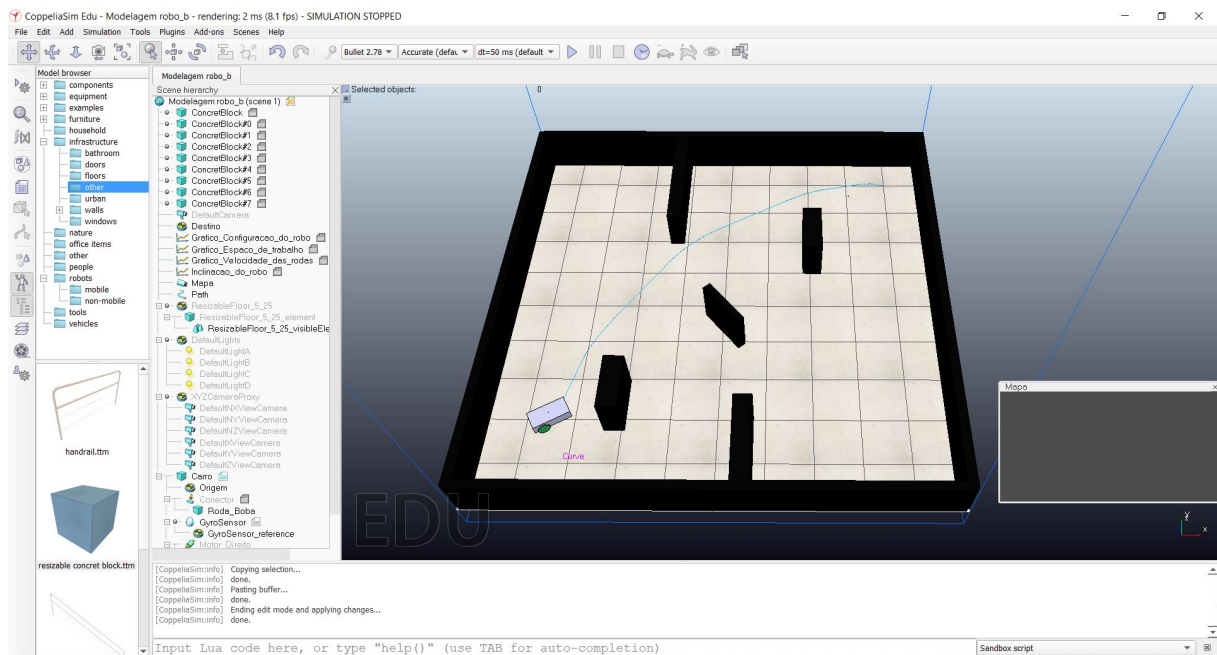


Fonte: Autores.

Finalmente, usufruindo do mapa probabilístico e o caminho a ser seguido, é possível aplicar os controladores de velocidades linear e angular para seguir o trajeto planejado e evitar quaisquer colisões. A Figura 11 ilustra a cena do robô móvel na sua posição e orientação inicial, bem como,

a trajetória a ser percorrida, onde está evita os obstáculos no simulador.

Figura 11. *Curva Plotada no Simulador*



Fonte: Autores.

5 CONCLUSÃO

Com a inovação de simuladores como este, pesquisadores de todos os níveis podem realizar suas pesquisas sem a necessidade física do dispositivo robótico. Assim, reduzindo ao máximo os possíveis erros de programação e custos, antes da aplicação no robô físico.

A partir desta, também, facilitou a compreensão da teoria mostrada em aula, principalmente, nos casos de escassez ou ausência dos materiais necessários para tal finalidade. Ademais, destaca-se a importância dessas ferramentas simulatórias, para o caso atual de pandemia, onde as aulas são remotas.

O planejamento de caminho foi implementado com sucesso e a precisão é dada pelo limite de pontos escolhidos para traçado das rotas sem colisões. Para tanto, dependendo da aplicação, a precisão pode ser maior, porém demanda proporcionalmente o aumento do gasto computacional.

Como o ajuste da distância do raio de giro foi exatamente o tamanho do robô móvel, ressalta-se que deve-se considerar erros de dimensionamento para aplicações reais, onde deve ser ajustado o aumento dessa distância para garantir a não colisão deste.

Finalmente, o experimento resultou nas proposições desejadas e este serve de base para os próximos, onde será desenvolvido um sistema de mapeamento para robô móvel que permita ao mesmo executar movimentos especificados em espaço povoado de obstáculos, sem colidir com os mesmos e sem dispor de um mapa previamente fornecido pelo operador.

REFERÊNCIAS

- [1] ROHMER, E.; SINGH, S. P. N.; FREESE, M. V-rep: A versatile and scalable robot simulation framework. In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. [S.l.: s.n.], 2013. p. 1321–1326. 4
- [2] NASCIMENTO, L. B. P. et al. Introdução ao v-rep: Uma plataforma virtual para simulação de robôs. In: *Alex Oliveira Barradas Filho; Pedro Porfirio Muniz Farias; Ricardo de Andrade Lira Rabêlo. (Org.). Minicursos da ERCEMAPI e EAComp 2019. 1ed.Porto Alegre: Sociedade Brasileira de Computação*. [S.l.: s.n.], 2019. p. 49–68.