Planejamento de Caminhos para Robôs Móveis em Ambientes Acidentados

Alexandre Souza Santos^{*†} Héctor Azpúrua^{†•}
Gustavo Pessin[†] Gustavo Medeiros Freitas^{•†}

^{*} Escola de Minas - Universidade Federal de Ouro Preto
Ouro Preto, MG, Brasil (e-mail: alexandre2525@gmail.com).

[†] Instituto Tecnológico Vale (ITV)

Ouro Preto, MG, Brasil (e-mails: hector.azpurua@itv.org,
gustavo.pessin@itv.org, gustavo.medeiros.freitas@pq.itv.org)

• Universidade Federal de Minas Gerais
Belo Horizonte, MG, Brasil

Abstract: Vale's Speleology team has a robotic platform, the EspeleoRobô, capable of moving on rough terrain while inspecting and mapping natural caves. Since the robot is a teleoperated device, its operability depends on the quality of communication with the control base. One solution to the problem is to provide autonomy for EspeleoRobô. Therefore, the present work consists on an initial step for the development of an autonomous navigation system, in which the objective is to analyze path planning techniques that optimize the robot navigation on rough terrain considering its kinematic model. We represent the environment using triangle meshes modeled as a graph and generate optimal paths with the Dijkstra's algorithm considering the following metrics as cost function: traveled distance, terrain traversability, and robot power consumption. Moreover, this paper proposes a cost function that accounts for multiple goals during the path planning, where the user can set the trade-off between them through weights. Lastly, we study the robot-terrain interaction using Pybullet e V-REP. We validate the metrics through simulations using artificial 3-D and real natural cave maps.

Resumo: A equipe de Espeleologia da Vale possui uma plataforma robótica, o EspeleoRobô, capaz de se locomover em terrenos acidentados e realizar a inspeção e mapeamento de cavidades naturais. Por se tratar de um dispositivo teleoperado, sua operacionalidade depende da qualidade de comunicação com a base de controle. Uma solução para o problema seria dar autonomia ao EspeleoRobô. Sendo assim, o presente trabalho corresponde a um passo inicial para o desenvolvimento de um sistema de navegação autônomo, no qual o objetivo é analisar técnicas de planejamento de caminho que otimizem a navegação do robô em terrenos acidentados considerando seu modelo cinemático. O ambiente é representado por malhas triangulares modeladas como um grafo, onde são gerados caminhos ótimos utilizando o algoritmo de Dijkstra, considerando as seguintes métricas como função de custo: distância percorrida, transversalidade do terreno e consumo energético do robô. Ainda, é proposta uma função de custo que considera múltiplos objetivos durante o planejamento de caminho, onde a relação entre elas é ajustada pelo operador por meio de pesos. Por fim, é realizado um estudo para analisar a interação do robô com o terreno utilizando os *softwares* Pybullet e V-REP. Para validação das métricas são feitas simulações utilizando um mapa 3-D artificial e um mapa real de cavidade natural.

Keywords: Path Planning, Mobile Robotics, Field Robotics.

Palavras-chaves: Planejamento de Caminhos, Robótica Móvel, Robótica de Campo.

1. INTRODUÇÃO

Robôs móveis vem sendo empregados para explorar ambientes desconhecidos ou com riscos para a vida humana, por exemplo exploração planetária (Schuster et al., 2019). Outro exemplo onde esses robôs estão sendo utilizados é no campo da mineração. Empresas que trabalham nessa área precisam inspecionar regiões com presença de cavidades naturais para determinar a relevância da cavidade no processo de viabilização de um projeto de mineração.

No entanto, a maioria das cavidades em regiões com presença de minério de ferro são estreitas e de difícil acesso, onde o profissional em trabalho de campo pode encontrar vários riscos como desabamentos e presença de animais selvagens e peçonhentos (Figura 1).

Figura 1. Riscos relacionados à inspeção de cavidades naturais: desabamentos e presença de animais selvagens e peçonhentos.

Com intuito de auxiliar o trabalho do espeleólogo, evitando sua presença nesses locais, a equipe de Espeleologia e Tecnologia da Diretoria de Planejamento e Desenvolvimento de Ferrosos da Vale deu início ao projeto do EspeleoRobô, adquirindo uma plataforma robótica teleoperada capaz de se locomover em terrenos acidentados e realizar a inspeção e mapeamento de cavidades (Cota et al., 2017; Freitas et al., 2018).

A transmissão de sinais entre o EspeleoRobô e a base de controle é feita via rádio, e apesar dos êxitos em diversos testes realizados em campo, foram constatados problemas recorrentes de perda de comunicação entre a base e o robô. Isso porque a comunicação via rádio apresenta limitações quando se trata de ambientes não-estruturados com obstáculos como cavidades. Essas falhas podem causar danos na estrutura do robô caso esse caia em um buraco ou até mesmo a perda do equipamento caso esse entre em uma área que impossibilite o resgate.

Uma solução para o problema seria dar autonomia ao EspeleoRobô. Dessa maneira, um sistema de instrumentação e controle embarcados tomaria as decisões para a aprimorar a exploração e garantir a integridade do equipamento. Nesse sentido, o robô conta com uma torre de mapeamento composta por câmeras de alta resolução e o sensor Velodyne LiDAR VLP-16. A modelagem tridimensional da torre de instrumentação acoplada sobre o robô e um exemplo de teste em campo estão ilustrados na Figura 2.

O problema de navegação autônoma envolve várias tarefas como localização, mapeamento, planejamento de caminho e controle da trajetória. Este trabalho tem como objetivo analisar técnicas de planejamento de caminho afim de otimizar a exploração do robô em ambientes acidentados. Para validação das técnicas são utilizados ambientes representados por meio de malhas triangulares modeladas como um grafo, onde são gerados caminhos ótimos utilizando o algoritmo Dijkstra considerando as seguintes métricas

(a)

(b)

Figura 2. (a) Modelagem tridimensional da torre de instrumentação onde as dimensões de comprimento, largura e altura são 58,8cm, 38cm e 51,8cm, respectivamente; (b) exemplo de teste em campo.

como função de custo: distância percorrida, transversalidade do terreno e consumo energético do robô. Ainda, é proposta uma função de custo que considera múltiplos objetivos durante o planejamento de caminho. A relação entre os objetivos é ajustada por meio de pesos definidos pelo operador. Dessa forma, o algoritmo de planejamento de caminho avalia a relevância de cada métrica ao encontrar trajetos ligando as posições inicial e final. Por fim, é realizado um estudo para analisar a interação do robô com o terreno. Assim, o algoritmo tenta prevenir o tombamento do robô, garantindo sua integridade durante a exploração.

O estudo aqui descrito é uma extensão do trabalho apresentado em (Santos et al., 2018), onde os autores consideram o robô como uma partícula. Nesse trabalho está sendo considerado o modelo cinemático do EspeleoRobô, o que implica modificações nos cálculos das métricas de distância percorrida, transversalidade do terreno e consumo energético do robô. Anteriormente a métrica de distância percorrida calculava apenas os deslocamentos lineares do robô; a nova métrica incorpora o giro das rodas necessário para realizar deslocamentos angulares. De maneira análoga, a nova métrica de consumo energético considera a energia que o robô gasta para rotacionar, além da energia gasta em movimentos lineares. A nova métrica de transversalidade considera o modelo cinemático do robô para avaliar a interação do mesmo com o terreno, calculando os ângulos de tombamento do dispositivo com base nas posições dos pontos de contato entre o robô e o terreno.

Este trabalho é composto por 7 seções. Na Seção 2 são expostos os conceitos teóricos como fundamentação para o estudo. Na Seção 3 é apresentada a metodologia empregada para o planejamento de caminhos. Na Seção 4 são expostas as métricas de mobilidade utilizadas. Na Seção 5 são apontados os resultados obtidos por meio de simulações. Na Seção 6 são descritas as análises das simulações realizadas. E por fim, na Seção 7 são apresentadas as conclusões do presente trabalho e as atividades futuras.

2. PLANEJAMENTO DE CAMINHO PARA ROBÔS MÓVEIS

O processo de planejamento de caminho para robôs móveis vem sendo estudado há décadas. Esse processo pode ser dividido basicamente em duas etapas. A primeira consiste na modelagem do ambiente de navegação, geralmente realizada por meio de sensores como câmeras, lasers, sonares, entre outros. A segunda consiste na utilização de

algoritmos de busca para encontrar um caminho ótimo para o robô entre as posições inicial e final. Diferentes trajetos podem ser obtidos de acordo com o critério de busca, como menor distância percorrida ou consumo energético (Giesbrecht, 2004).

2.1 Modelos para representação de ambientes naturais

Segundo Burgard et al. (2016), os principais modelos geométricos para representação de ambientes naturais são grades de elevação, grades 3-D e malhas ou *meshes*.

Os modelos de grades de elevação descrevem o terreno como uma função $h = f(x, y)$, onde x e y representam as coordenadas planares das células da grade e h a elevação correspondente. Devido às suas características de estrutura de dados simples e de fácil obtenção, mapas de elevação têm sido bastante utilizados em aplicações de robótica (Broggi et al., 2013; Fankhauser et al., 2014). Esses mapas assumem um plano como referência; no entanto, essa suposição é violada em diversas situações, inviabilizando a utilização desse tipo de modelo.

Uma solução consiste em representar o terreno através de nuvem de pontos ou grades tridimensionais. Dessa forma, todos os dados são preservados em seu formato original, não havendo restrição de geometria do ambiente. Apesar de ser uma representação mais genérica, as grades 3-D têm um custo computacional elevado, sendo inviáveis em determinadas aplicações.

Outra alternativa é representar o terreno por meio de modelos de malhas. Esses modelos podem representar qualquer combinação de superfícies de uma forma compacta. Contudo, esse tipo de representação pode encontrar dificuldades quando se trata de ambientes muito complexos (Siciliano and Khatib, 2016).

2.2 Algoritmos de busca de caminho

Uma vez que o ambiente foi representado utilizando algum dos métodos citados na Seção 2.1 também representados como grafos, é necessário buscar o melhor caminho entre as posições inicial e final definidas pelo usuário. Grafos são estruturas compostas por nós ou vértices ligados por arcos ou linhas, onde pode-se atribuir pesos aos arcos. Os principais algoritmos de busca de grafo são: busca em profundidade ou *Depth-First Search* (DFS), busca em largura ou *Breadth-First Search* (BFS), A^* , Dijkstra e D^* .

O algoritmo A^* proposto por Hart et al. (1968) é um dos mais utilizados em aplicações de robótica. Esse método de busca utiliza a função de custo $f(n) = g(n) + h(n)$ para orientar a pesquisa do caminho mais curto, sendo $g(n)$ a distância percorrida da posição inicial até a posição atual e $h(n)$ a distância entre as posições atual e final. Dessa maneira, o algoritmo escolhe sempre o sucessor com o menor valor. O algoritmo A^* apresenta algumas características que o torna tão famoso. Esse método de busca considera uma heurística admissível. Isso significa que o A^* subestima o custo do caminho ótimo $f(n)^*$, ou seja, $f(n) \leq f(n)^*$. Isso faz com que o algoritmo explore outros trajetos com custo inferior ao percurso ótimo, encerrando a busca somente quando o caminho mais curto ou ótimo é encontrado. Ainda, esse algoritmo apresenta a busca mais rápida se comparado com outras estratégias de planejamento ótimo que utilizam a mesma heurística (Giesbrecht, 2004).

O algoritmo proposto por Dijkstra (1959) é uma variação do algoritmo A^* onde $h(n) = 0$, ou seja, $f(n) = g(n)$. Dessa forma, o método não utiliza nenhuma heurística na busca do percurso ótimo, fazendo somente a expansão do caminho mais curto até que a posição final seja encontrada (Choset, 2005). O algoritmo de Dijkstra é utilizado para encontrar o caminho de custo mínimo entre vértices de um grafo com arestas de peso não negativo. Escolhido um vértice como raiz da busca, esse algoritmo calcula o custo mínimo desse vértice para todos os demais vértices do grafo. O algoritmo de Dijkstra parte de uma estimativa inicial para o custo mínimo e vai sucessivamente ajustando essa estimativa. Um vértice é dito estar fechado quando o algoritmo já tiver obtido um caminho de custo mínimo do vértice tomado como raiz da busca até ele. Caso contrário, o mesmo é dito estar aberto. Quando todos os vértices tiverem sido fechados, os valores obtidos são os custos mínimos dos caminhos que partem do vértice raiz até os demais vértices do grafo. O percurso de custo mínimo propriamente dito é obtido a partir dos vértices chamados de precedentes.

2.3 Trabalhos relacionados

No trabalho descrito em (Ishigami et al., 2007) é apresentado um algoritmo de planejamento de caminho para um *rover* planetário que leva em consideração o escorregamento das rodas do veículo. Para encontrar os melhores trajetos os autores utilizaram o algoritmo de Dijkstra considerando três índices em sua função de custo: inclinação, irregularidade do terreno e distância percorrida. Esses índices são normalizados e a relação entre eles é ajustada por meio de pesos. Dessa forma, o algoritmo de planejamento de caminho pode considerar a relevância de cada índice ao gerar soluções. Como resultado, foram avaliados dois trajetos obtidos pelo algoritmo onde manteve-se constante os pesos dos índices, porém, com restrições de inclinação de $15,0^\circ$ e $7,5^\circ$ em relação aos ângulos de rolagem (*roll*) e arfagem (*pitch*), respectivamente. Ambas soluções obtiveram resultados satisfatórios. No entanto, o caminho com menor valor de restrição de inclinação mostrou ser uma solução mais segura, apesar de ser um percurso mais longo. O desenvolvimento apresentado pelos autores é similar ao estudo descrito nesse documento. No entanto, nesse trabalho o ambiente é representado por meio de malhas triangulares ao invés de mapas de elevação. Ainda, o algoritmo proposto também considera a métrica de consumo energético do robô aos gerar caminhos soluções.

No estudo realizado por Raja et al. (2015) é proposto um algoritmo de planejamento de movimento para um *rover* com seis rodas em terrenos irregulares. Esse algoritmo utiliza o método de campos de potenciais para representar o ambiente, onde é apresentada uma função de gradiente que considera as forças de atração, repulsão, tangencial e gradiente. A força de gradiente varia de acordo com os ângulos rolagem, arfagem e guinada, que são obtidos por meio do modelo cinemático do veículo. O algoritmo considera essa força de modo a evitar que o *rover* navegue por locais onde o gradiente tenha valor elevado, resultando em um caminho mais seguro. São atribuídos pesos para os componentes da função de gradiente, que por sua vez são otimizados utilizando algoritmos genéticos. O algoritmo também avalia as velocidades das rodas do veículo para garantir a estabilidade e prevenir escorre-

gamento. São comparados percursos que utilizam quatro diferentes métricas como função de custo do algoritmo: consumo de energia, escorregamento das rodas, força de tração e distância percorrida. Por meio de simulações e testes experimentais, os autores concluíram que o método proposto é capaz de gerar melhores trajetos em terrenos irregulares se comparado com o método convencional de campos de potenciais.

Em (Jeddisaravi et al., 2016) os autores propõem um planejamento de caminho multi-objetivo para robôs móveis baseado no algoritmo A*. O trabalho tem foco na aplicação de exploração planetária, na qual o ambiente possui fendas, colinas e composições rochosas. O objetivo do algoritmo proposto é encontrar caminhos de forma a minimizar métricas de dificuldade, risco e elevação entre a posição inicial até posição final. Os caminhos devem ser os mais curtos possíveis para reduzirem o tempo de exploração e a energia consumida, e ao mesmo tempo seguro de modo a evitar problemas para o robô durante a navegação. Os autores também utilizaram pesos para ajustar a relação entre as métricas e confirmam a aplicabilidade do algoritmo por meio de simulações em dois cenários distintos.

3. METODOLOGIAS PARA PLANEJAMENTO DE CAMINHOS

A metodologia proposta neste artigo consiste na definição dos modelos de representação do ambiente e do robô, do algoritmo planejamento de caminho e das métricas utilizados. São utilizados mapas conhecidos com representação discreta baseados na geometria do ambiente, onde não são tratadas incertezas de modelagem. Assume-se que toda área do terreno é trafegável, a pose do robô com respeito ao mapa é conhecida a todo momento e os coeficientes de atrito do terreno são constantes. São avaliados percursos obtidos por meio de otimização das métricas de distância percorrida, transversalidade do terreno e consumo energético do robô utilizando o *software* Matlab. Ainda, é proposta uma função de custo que considera múltiplos objetivos durante o planejamento de caminho. A relação entre os objetivos é ajustada por meio de pesos definidos pelo operador. O algoritmo de planejamento de percurso avalia a relevância de cada métrica ao encontrar trajetos ligando as posições inicial e final. Por fim, é realizado um estudo para analisar a interação do robô com o terreno utilizando os *softwares* Pybullet e V-REP. O algoritmo tenta prevenir o tombamento do robô, garantindo sua integridade durante a exploração.

3.1 Modelo de representação do ambiente

O modelo de representação do ambiente adotado neste trabalho é o de malhas triangulares. O mapa de cavidade utilizado foi gerado a partir de uma nuvem de pontos coletada pela equipe de espeleologia da Vale em visitas em áreas com cavidades naturais. Esse mapa é relevante para o estudo pois a empresa recorrentemente precisa inspecionar áreas com características similares. Um exemplo dessa representação está ilustrado na Figura 7.

3.2 Representação do robô

Neste trabalho é considerado o modelo cinemático do EspeloRobô. O robô é um veículo do tipo *skid-steering* e é utilizado com a configuração de seis rodas. Ele é capaz de

locomover para frente e para trás, e fazer rotações no seu próprio eixo. Sendo assim, o EspeleRobô, teoricamente, é capaz de seguir qualquer caminho livre de obstáculos no solo por meio de uma sequência de movimentos lineares e angulares. O robô pode se deslocar do centro de uma malha até o centro outra com vértice em comum. As malhas são definidas como nós ou vértices, e cada nó é associado a um custo de acordo com a métrica utilizada. Os nós nos quais o robô pode se deslocar a partir um vértice de referência estão ilustradas na Figura 3.

Figura 3. (a) Representação da posição atual do robô e dos possíveis destinos; (b) grafo correspondente com os custos de acordo com a métrica utilizada.

3.3 Algoritmo de busca

De acordo com o modelo de representação do ambiente adotado, o algoritmo de Dijkstra foi escolhido para o desenvolvimento do estudo por ser ótimo e adequado para problemas de busca em grafos com arcos de pesos fixos, distintos e não-negativos. Esse algoritmo tem convergência garantida para todos os casos, enquanto o A* só é ótimo se a heurística $h(n)$ for admissível. Com relação à métrica de distância percorrida, uma heurística válida é a distância euclidiana entre as posições atual e final do robô. No entanto, não é trivial estimar as heurísticas para as métricas de transversalidade do terreno e consumo energético do robô, pois ambas são extremamente dependentes do perfil do terreno percorrido. Além disso, é possível utilizar o algoritmo de Dijkstra de maneira simples para otimizar múltiplas métricas simultaneamente. O pseudocódigo do método escolhido está descrito pelo Algoritmo 1.

Algoritmo 1: Pseudocódigo do Algoritmo de Dijkstra.

- 1: Adicionar o nó inicial na lista de nós ABERTA e calcular a função de custo $f(n) = g(n)$.
- 2: Remover da lista ABERTA o nó com o menor valor de $f(n)$ e o colocar na lista de nós FECHADA. Esse é o nó de índice n . Em caso de empate, decidir pelo nó mais próximo do destino e que possui menor distância percorrida.
- 3: **se** $n = \text{destino}$ **então**
- 4: Usar os ponteiros dos nós para obter o caminho solução.
- 5: FIM
- 6: **senão**
- 7: Determinar todos os nós sucessores (filhos) de n e calcular $f(n)$ para cada sucessor que não esteja na lista FECHADA.
- 8: Associar os valores de $f(n)$ calculados aos nós sucessores que não estejam na lista de nós ABERTA ou FECHADA, colocá-los na lista ABERTA e adicionar ponteiros apontados ao nó pai de índice n .
- 9: Associar o menor valor de $f(n)$ a cada nó sucessor já contido na lista de nós ABERTA (mínimo(novo $f(n)$, velho $f(n)$).
Obs: em caso de empate para a métrica de transversalidade do terreno, o algoritmo irá escolher o caminho mais curto e mais próximo da posição final.
- 10: Retornar ao passo 2.
- 11: **fim se**

3.4 Métrica de distância percorrida

A métrica de distância percorrida considerando a cinemática do robô incorpora o giro das rodas necessário para realizar deslocamentos lineares e angulares que o mesmo precisa fazer para se deslocar da posição inicial até a posição final. Assim, a métrica em questão é calculada da seguinte maneira:

$$Dc_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} + \frac{\alpha_{ij} \cdot 2 \cdot \pi \cdot l_5}{360}, \quad (1)$$

onde o primeiro termo é a distância euclidiana entre os nós adjacentes de índices i e j , α_{ij} equivale à rotação em graus que o robô precisa fazer entre os nós adjacentes de índices i e j , e l_5 é a metade da distância em metros entre as duas rodas centrais do EspeleRobô. A função de custo utilizando essa métrica é definida pela seguinte equação:

$$C_1(p) = \sum_{n=p} Dc_{ij}(n), \quad (2)$$

onde o percurso p consiste em uma série de nós vizinhos dado por $p = \{n_{inicial}, \dots, n_i, \dots, n_{final}\}$. O trajeto mais curto pc_{dist} será encontrado de forma a satisfazer a seguinte equação:

$$\min C_1(p) = C_1(pc_{dist}). \quad (3)$$

3.5 Métrica de transversalidade do terreno

A métrica de transversalidade considerando a cinemática do robô visa obter o caminho estável ou com menor risco de capotamento para o robô da posição inicial até a posição final, ambas definidas pelo usuário. Essa métrica é baseada no ângulo de tombamento robô, que indica qual é o menor ângulo que o dispositivo pode ser rotacionado até capotar (Papadopoulos and Rey, 1996). Os ângulos de tombamento do robô são calculados a partir da interação do robô com o terreno utilizando o simulador de física Pybullet, ilustrado na Figura 4, que retorna as posições dos pontos de contato com o terreno e centro de gravidade do robô. Esse *software* foi escolhido por ser uma ferramenta leve e capaz de gerar resultados com muita rapidez. É feita uma integração entre o Matlab e Pybullet de modo que o Matlab execute automaticamente o código escrito em Python enquanto o algoritmo de planejamento de caminho encontra a solução.

Neste trabalho é feita uma modificação no desempate do algoritmo de Dijkstra. Para a métrica de transversalidade do terreno, em caso de empate na escolha do menor custo, o algoritmo irá escolher o caminho mais curto e mais próximo da posição final. Essa modificação foi realizada para que os caminhos obtidos utilizando essa métrica em um mapa plano sem inclinação fossem os idênticos aos caminhos ótimos de distância percorrida e consumo energético do robô.

Uma vez obtidos os ângulos de tombamento do robô como descrito em (Papadopoulos and Rey, 1996), essa métrica de transversalidade é calculada da seguinte forma:

$$Tc_{ij} = \gamma_{max} - \min [\gamma(n)_{ij}] \mid n = 1, 2, \dots, 6, \quad (4)$$

onde γ_{max} é o maior ângulo de tombamento possível, isto é, quando o robô está em um plano sem inclinação, e $\gamma(n)_{ij}$ é um vetor com os seis ângulos de tombamentos do robô quando o mesmo está posicionado sobre o nó de índice i e

Figura 4. Estudo da interação do robô com o terreno utilizando o simulador Pybullet.

orientado em direção ao nó de índice j . A função de custo utilizando essa métrica é definida pela seguinte equação:

$$C_2(p) = \sum_{n=p} Tc_{ij}(n), \quad (5)$$

onde o percurso p consiste em uma série de nós vizinhos dado por $p = \{n_{inicial}, \dots, n_i, \dots, n_{final}\}$. O trajeto com menor risco de capotamento pc_{tran} será encontrado de forma a satisfazer a seguinte equação:

$$\min C_2(p) = C_2(pc_{tran}). \quad (6)$$

3.6 Métrica de consumo energético do robô

A métrica de consumo energético considerando a cinemática do robô leva em consideração as energias que o robô gasta ao se locomover em linha reta e fazer rotações no próprio eixo. Para tal, foram realizadas simulações utilizando o simulador V-REP. Nessas simulações o robô seguiu em linha reta por 50m em planos com diferentes inclinações a uma velocidade linear de 1m/s. Os coeficientes de atrito dinâmico e estático foram definidos como 0,6 e 0,9, respectivamente. Também foi realizado um ensaio para estimar o consumo médio do robô ao realizar rotações no próprio eixo. Nesse experimento o robô rotacionou 20 vezes em torno do seu próprio eixo em um plano sem inclinação e o consumo médio para rotacionar 360° foi de $E_{med} = 37.735,9W$. Para cada cenário foram realizadas três simulações e o consumo final é a média aritmética das medidas.

Para encontrar o consumo energético do robô em outras inclinações foram realizadas regressões lineares, ilustradas na Figura 5.

Figura 5. (a) Consumo energético do robô em planos com inclinações negativas; (b) consumo energético do robô em planos com inclinações positivas.

Dessa forma, a métrica de energia é calculada das seguintes maneiras:

$$Ec_{ij} = \left(\frac{E_{med}\alpha_{ij}}{360} \right) + (-475,07\theta_{ij} + 1089,3)D_{ij}, \forall \theta < 0, \quad (7)$$

$$Ec_{ij} = \left(\frac{E_{med}\alpha_{ij}}{360} \right) + (564,97\theta_{ij} + 1364,9)D_{ij}, \forall \theta \geq 0, \quad (8)$$

onde Ec_{ij} é a energia gasta para rotacionar e deslocar o robô entre os nós adjacentes de índices i e j , α_{ij} é a rotação em graus que o robô precisa fazer entre os nós adjacentes de índices i e j , θ_{ij} é o ângulo formado entre o vetor que liga os centros das malhas i e j e o plano XY, e D_{ij} é a distância euclidiana entre os nós de índices i e j . A função de custo utilizando essa métrica é definida pela seguinte equação:

$$C_3(p) = \sum_{n=p} Ec_{ij}(n), \quad (9)$$

onde o percurso p consiste em uma série de nós vizinhos dado por $p = \{n_{inicial}, \dots, n_i, \dots, n_{final}\}$. O trajeto mais econômico pc_{ener} será encontrado desta forma:

$$\min C_3(p) = C_3(pc_{ener}). \quad (10)$$

3.7 Métricas combinadas

As métricas considerando a cinemática do robô são combinadas de forma a obter um caminho que considere a relevância de cada uma. A função de custo para essa abordagem é dada pela seguinte equação:

$$C_4(p) = \sum_{n=p} (P_d Nd_{ij} Dc_{ij}(n) + P_t Nt_{ij} Tc_{ij}(n) + P_e Ne_{ij} Ec_{ij}(n)), \quad (11)$$

onde P_d , P_t e P_e são os pesos que ditam as prioridades entre os índices de distância percorrida, transversalidade do terreno e consumo energético do robô, respectivamente. Esses pesos são configurados pelo usuário de acordo com as prioridades da exploração. Por exemplo, se a prioridade da missão for o tempo, o usuário deve ajustar P_d com um valor superior em relação aos demais pesos. Em ambientes com alto risco de capotamento do robô, o usuário deve dar preferência ao peso P_t . Já em situações onde o consumo energético é crítico, o peso P_e deve ter prioridade em relação aos demais. A somatória desses pesos tem valor unitário. Nd_{ij} , Nt_{ij} e Ne_{ij} são os coeficientes de normalização, que por sua vez são obtidos de acordo com vizinhança dos nós adjacentes, onde todos os custos são divididos pelo maior custo local de acordo com cada métrica. O caminho de navegação pc_{comb} resultante será encontrado desta maneira:

$$\min C_4(p) = C_4(pc_{comb}). \quad (12)$$

4. RESULTADOS

Nesta seção são apresentados os resultados obtidos para o planejamento de caminhos em cenários distintos.

4.1 Validação do algoritmo

Inicialmente, as métricas propostas foram validadas utilizando um mapa 3-D gerado manualmente a partir de curvaturas em forma de gaussianas para simular irregularidades no terreno (Figura 6). Esse mapa foi gerado

dessa forma para validar o algoritmo de forma visual, o que poderia não ser intuitivo em um mapa de terreno acidentado. Foram utilizados diferentes pesos para obtenção dos caminhos com métricas combinadas. No entanto, o resultado que obteve os melhores valores de distância percorrida, ângulo mínimo de tombamento e energia consumida foi utilizando $P_d = 0,25$, $P_t = 0,5$ e $P_e = 0,25$. O custo total normalizado é a soma dos custos normalizados das três métricas. Os caminhos ótimos obtidos para cada métrica estão representados na Figura 6. Os valores de distância percorrida, ângulo mínimo de tombamento, energia consumida, e o custo total normalizado de cada caminho estão apresentados na Tabela 1. Para evitar que o robô navegue próximo das extremidades do mapa, são removidas as áreas próximas à borda num raio equivalente a largura do EspeleRobô. Os caminhos ótimos obtidos para cada métrica no mapa 3-D estão representados na Figura 6.

Figura 6. Caminhos ótimos obtidos utilizando o algoritmo de Dijkstra em um mapa com curvaturas gaussianas. As cores das malhas estão relacionadas com o eixo Z.

4.2 Simulações em um mapa de cavidade natural

Após concluída a etapa de validação, foram realizadas novas simulações agora em um mapa de cavidade natural. Os pesos utilizados para obter o caminho com métricas combinadas foram mantidos (i.e., $P_d = 0,25$, $P_t = 0,5$, e $P_e = 0,25$). Os caminhos ótimos para cada métrica estão ilustrados na Figura 7. Os valores de distância percorrida, mínimo ângulo de tombamento, energia consumida e custo total normalizado de cada caminho estão apresentados na Tabela 2.

Figura 7. Caminhos ótimos obtidos utilizando o algoritmo de Dijkstra em um mapa de cavidade natural. As cores das malhas estão relacionadas com o eixo Z.

Tabela 1. Valores calculados para os caminhos ótimos obtidos nas simulações realizadas em um mapa com curvaturas gaussianas considerando a cinemática do robô.

Caminho	Distância percorrida (m)	Ângulo mínimo de tombamento (°)	Energia consumida (W)	Custo total normalizado
Mais curto	9,21	5,38	48504,00	5,89
Mais estável	15,72	45,93	96795,00	5,34
Mais econômico	9,42	18,96	44316,00	4,36
Métricas combinadas	9,96	39,95	49404,00	3,52

Tabela 2. Valores calculados para os caminhos ótimos obtidos nas simulações realizadas em um mapa de cavidade natural considerando a cinemática do robô.

Caminho	Distância percorrida (m)	Ângulo mínimo de tombamento (°)	Energia consumida (W)	Custo total normalizado
Mais curto	8,33	7,05	109800,00	23,25
Mais estável	15,26	26,38	282020,00	18,17
Mais econômico	8,48	18,70	106120,00	21,00
Métricas combinadas	11,43	23,50	182710,00	16,19

4.3 Simulações no V-REP

Uma vez obtidos os caminhos ótimos para cada métrica, foram realizadas simulações no V-REP para garantir que os caminhos são factíveis pelo robô. Durante as simulações foram utilizadas diferentes velocidades lineares para o robô, onde foi constatado que o mesmo não conseguiu completar os percursos com velocidades superiores a 0,3m/s, isso devido ao escorregamento das rodas. Para que o robô conseguisse completar todos os caminhos encontrados pelo algoritmo de Dijkstra, a velocidade linear foi mantida constante em 0,3m/s. A velocidade angular varia de forma proporcional à rotação que o robô precisa fazer para seguir o caminho desejado. Os coeficientes de atrito dinâmico e estático foram mantidos em 0,6 e 0,9, respectivamente. Como resultado, o robô conseguiu percorrer todos os caminhos encontrados pelo algoritmo de Dijkstra, não encontrando dificuldade. Uma das simulações está ilustrada pela Figura 8. Os caminhos são os mesmos apresentados na Figura 7. Um vídeo com algumas das simulações pode ser encontrado no *link*: <https://www.youtube.com/watch?v=YLU5qn0-z9I&feature=youtu.be>.

Figura 8. Seguimento de caminho em uma mada de cavidade natural utilizando o simulador V-REP: posição inicial (a), posição dois (b), posição três (c) e posição final (d).

5. ANÁLISE DOS RESULTADOS

Analisando os resultados das Tabelas 1 e 2, é possível concluir que os caminhos obtidos pelo algoritmo de Dijkstra considerando as métricas propostas mostraram ser ótimos, visto que os melhores valores de distância percorrida, ângulo mínimo de tombamento e energia consumida estão relacionados aos trajetos que utilizam as respectivas métricas no planejamento de caminho. O menor custo total normalizado também está relacionado com o caminho obtido com métricas combinadas. É mais intuitivo chegar a essa conclusão de forma visual observando os resultados obtidos na etapa de validação do algoritmo, ilustrados na Figura 6. Ainda, é possível observar que os caminhos encontrados utilizando as métricas de forma combinada alcançaram valores intermediários de distância percorrida, mínimo ângulo de tombamento e energia consumida.

Analisando os resultados apresentados na Tabela 2, o caminho com métricas combinadas é que o apresenta o menor custo total normalizado. Isso se deve à tentativa do algoritmo em minimizar todos os atributos de forma simultânea, levando em consideração o peso atribuído a cada uma.

6. CONCLUSÃO

Neste trabalho é apresentado um estudo sobre técnicas de planejamento de caminho para robôs móveis em ambientes acidentados, que servirá como base para o desenvolvimento de um sistema de navegação autônomo para o EspeleoRobô. Esse sistema terá como objetivo otimizar a exploração do robô em terrenos acidentados e garantir a integridade do equipamento.

Com os resultados obtidos, foi constatado que é possível realizar efetivamente o planejamento de caminho utilizando as métricas propostas de distância percorrida, transversalidade do terreno e consumo energético considerando a cinemática do robô. A métrica de energia utiliza valores realistas obtidos por meio de simulações, enquanto a métrica de transversalidade mostrou ser eficaz visto que o robô conseguiu executar todas as trajetórias sem que houvesse capotamento. Ainda, é proposta uma função de custo que considera múltiplos objetivos durante o planejamento de caminho. Dessa maneira, o algoritmo avalia a relevância de cada métrica ao encontrar trajetos ligando as posições inicial e final.

No presente trabalho é feita uma prova de conceito para validação das metodologias propostas. Em trabalhos futuros, pretende-se realizar simulações com outros algoritmos de exploração, por exemplo o RRT*. Os códigos desenvolvidos em Matlab serão otimizados e adaptados para o sistema que controla o EspeleoRobô, *Robot Operating System* (ROS). Ainda, serão estudadas técnicas de mapeamento e localização simultânea (SLAM) e algoritmos de exploração. Dessa forma, o sistema fará com que o robô seja capaz de mapear e se localizar no terreno, e escolher de forma automática os destinos que deseja explorar. Posteriormente, serão realizados testes em campo para avaliar o desempenho do robô funcionando em modo autônomo.

AGRADECIMENTOS

Os autores gostariam de agradecer à equipe do Laboratório de Robótica do Instituto Tecnológico Vale. O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, Brasil (CAPES), Código de Financiamento 001; do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq); da Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG); e da Vale S.A.

REFERÊNCIAS

- Broggi, A., Cardarelli, E., Cattani, S., and Sabbatelli, M. (2013). Terrain mapping for off-road autonomous ground vehicles using rational b-spline surfaces and stereo vision. In *Intelligent Vehicles Symposium (IV), 2013 IEEE*, 648–653. IEEE.
- Burgard, W., Hebert, M., and Bennewitz, M. (2016). World modeling. In *Springer handbook of robotics*, 1135–1152. Springer.
- Choset, M. (2005). *Principles of robot motion: theory, algorithms, and implementation*. MIT press.
- Cota, E., Torre, M., Rocha, F., Garcia, G., Junior, A., Ramos, V., Queiroz, V., Zanini, V., Brito, G., Marques, A., Pinto, E., Nogueira, L., Freitas, G., Miola, W., Reis, M., Araújo, R., and Brandi, I. (2017). Dispositivo de monitoramento remoto de cavidades - EspeleoRobô. In *XIII Simpósio Brasileiro de Automação Inteligente*.
- Dijkstra, E. (1959). A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1), 269–271.
- Fankhauser, P., Bloesch, M., Gehring, C., Hutter, M., and Siegwart, R. (2014). Robot-centric elevation mapping with uncertainty estimates. In *Mobile Service Robotics*, 433–440. World Scientific.
- Freitas, G., Rocha, F., Torre, M., Junior, A., Ramos, V., Nogueira, L., Santos, A., Cota, E., Miola, W., Reis, M., Costa, B., Ledezma, L., Evangelista, R., Alcantra, P., Lima, R., Souza, T., Brandi, I., Araújo, R., and Gomes, M. (2018). Multi-terrain inspection robotic device and methods for configuring and guiding the same. BR Patent 2018050025.
- Giesbrecht, J. (2004). Global path planning for unmanned ground vehicles. Technical report, Defence Research and Development Canada.
- Hart, P., Nilsson, N., 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.
- Ishigami, G., Nagatani, K., and Yoshida, K. (2007). Path planning for planetary exploration rovers and its evaluation based on wheel slip dynamics. In *Robotics and Automation, 2007 IEEE International Conference on*, 2361–2366. IEEE.
- Jeddisaravi, K., Alitappeh, R., and Guimarães, F. (2016). Multi-objective mobile robot path planning based on A* search. In *Computer and Knowledge Engineering (ICCCKE), 2016 6th International Conference on*, 7–12. IEEE.
- Papadopoulos, E. and Rey, D. (1996). A new measure of tipover stability margin for mobile manipulators. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 4, 3111–3116. IEEE.
- Raja, R., Dutta, A., and Venkatesh, K. (2015). New potential field method for rough terrain path planning using genetic algorithm for a 6-wheel rover. *Robotics and Autonomous Systems*, 72, 295–306.
- Santos, A.S., Azpúrua, H.I.P., Pessin, G., and Freitas, G.M. (2018). Path planning for mobile robots on rough terrain. In *2018 Latin American Robotic Symposium, 2018 Brazilian Symposium on Robotics (SBR) and 2018 Workshop on Robotics in Education (WRE)*, 265–270. IEEE.
- Schuster, M., Brunner, S., Bussmann, K., Büttner, S., Dömel, A., Hellerer, M., Lehner, H., Lehner, P., Porges, O., Reill, J., et al. (2019). Towards autonomous planetary exploration. *Journal of Intelligent & Robotic Systems*, 93(3-4), 461–494.
- Siciliano, B. and Khatib, O. (2016). *Springer handbook of robotics*. Springer.