

NAVEGAÇÃO DE UM QUADRIRROTOR EM FLORESTAS ESPARSAS UTILIZANDO PLANEJAMENTO DE MOVIMENTO E SENSOR A LASER

HENRIQUE N. MACHADO*, GUILHERME A. S. PEREIRA*

*Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal de Minas Gerais
Av. Antônio Carlos 6627, 31270-901, Belo Horizonte, MG, Brazil

Emails: henriquenm@ufmg.br, gpereira@ufmg.br

Abstract— This paper presents and evaluates a strategy for aerial vehicles navigation in sparse forests. The strategy uses a previous published methodology that integrates vector fields, for vehicle guidance, and an asymptotically optimal sampling based motion planner, for obstacle avoidance. The main contributions of this work are the efficient implementation of laser based obstacle detectors, which is essential to allow the motion planner to run in the limited onboard hardware, and the execution of several actual experiments that show the efficiency of the method.

Keywords— Aerial vehicles, robotics, navigation, laser based obstacle detection, sparse forests.

Resumo— Esse artigo apresenta e avalia uma estratégia para navegação de veículos aéreos em florestas esparsas. A estratégia é baseada em uma metodologia previamente publicada que integra campos vetoriais, para a guiagem do veículo, e um planejador de movimento probabilístico assintoticamente ótimo, responsável pelo desvio de obstáculos não modelados. As principais contribuições desse trabalho são a implementação eficiente de detectores de obstáculos em dados de sensores a laser, o que é essencial para que o planejador seja executado em hardwares limitados a bordo do veículo, e a execução de vários experimentos reais que mostram a eficiência do método.

Palavras-chave— Veículos aéreos, robótica, navegação, detecção de obstáculos com laser, florestas esparsas.

1 Introdução

A navegação de robôs em florestas ou regiões com muitas árvores e outros tipos de vegetação é um grande desafio, devido aos vários obstáculos (ver Figura 1), como troncos, folhagens, terrenos íngremes ou alagados e detritos. Essas adversidades muitas vezes dificultam o uso de veículos terrestres, exigindo sistemas de locomoção complexos e/ou adaptativos (Freitas et al., 2010). Nessas situações, pequenos veículos aéreos autônomos ou UAVs (*Unmanned Air Vehicles*), como os quadrisrotor, aparecem como uma alternativa viável. Com este tipo de tecnologia, é possível realizar a exploração e reconhecimento do ambiente abaixo das copas das árvores, o que pode ser útil para aplicações como mapeamento de áreas; pesquisas biológicas sobre a fauna e flora; auxílio a equipes de resgate, entre outros. Apesar do terreno das florestas não influenciar na locomoção dos veículos aéreos, esses devem contar com sistemas de detecção e desvio de obstáculos, essenciais para garantir a navegação segura entre as árvores.

Existem pesquisas recentes nesta área como Cui et al. (2014) que desenvolveram um sistema de navegação autônoma de um UAV em floresta utilizando uma técnica de *Simultaneous Localization and Mapping* (SLAM) para identificar os obstáculos, sua posição no mapa e em seguida planejar uma trajetória viável livre de colisões. De forma similar Schultz et al. (2016) apresenta uma metodologia para mapeamento e navegação de um veículo aéreo em florestas, denominada SNAM (*Simultaneous Navigation and Mapping*) que utiliza



Figura 1: Foto do experimento realizado com o quadrisrotor. O robô estava se movimentando de forma autônoma e utilizando o laser planar para identificar e desviar dos obstáculos no caminho.

três lasers planares com inclinações diferentes para se obter informações do espaço 3D do ambiente. Chisholm et al. (2013) fizeram um estudo preliminar do uso de um UAV com um laser planar para pesquisas florestais. Já Shah et al. (2017) e Langelaan and Rock (2004) propuseram métodos de navegação utilizando o sensor inercial (IMU) do veículo e uma câmera frontal para mapear as árvores de uma determinada região. Além desses, diversos outros estudos recentes sobre localização (Zhen et al., 2017), planejamento de movimento (Pfeiffer et al., 2017), (Singh et al., 2017), (Oleynikova et al., 2017) e desvio automático de obstáculos (Kamel et al., 2017) poderiam ser utilizados para resolver o problema.

Considerando estas pesquisas e a importância de se navegar em florestas de forma autônoma, este trabalho apresenta uma estratégia para navegação de um UAV em florestas esparsas. Nesse contexto, considera-se como florestas esparsas aquelas onde as distâncias entre as árvores é grande o suficiente para que (i) o veículo possa voar entre árvores mantendo uma distância de segurança e (ii) a copa das árvores não impeçam completamente passagem de sinais de GPS. Levando em consideração um veículo com diâmetro de um metro, nesse trabalho é definido como esparsas regiões onde as árvores estão com um afastamento médio de 5 metros. A estratégia proposta nesse trabalho aplica a técnica de desvio automático de obstáculos proposta por Pereira et al. (2016), que foi devidamente integrada a um sistema de detecção eficiente de obstáculos utilizando sensor a laser. Uma das principais contribuições do artigo é a realização de experimentos reais (ver Figura 1) utilizando um quadricóptero equipado com um laser planar. Nos experimentos mostrados no artigo o veículo percorre uma curva fechada planar (altura constante) dentro de uma floresta esparsa desconhecida a uma velocidade média de 1 m/s.

A próxima seção descreve a metodologia proposta. Os resultados experimentais são apresentados na Seção 3 enquanto as conclusões e trabalhos futuros são apresentados na Seção 4.

2 Metodologia

A metodologia proposta neste trabalho pode ser dividida em três etapas. A primeira consiste na detecção dos obstáculos e tratamento dos dados do sensor de distância a laser, a fim de se obter informações adequadas para serem utilizadas pelo planejador de movimento. Em seguida, é utilizado o planejador de trajetória que integra campos vetoriais e *Rapidly-exploring Random Tree Star* (RRT*). Finalmente é utilizado um controlador de velocidades para que o UAV siga corretamente o caminho calculado. Essas operações são executadas em *loop* com o período de t_{loop} segundos. Cada uma dessas etapas são detalhadas a seguir.

2.1 Detecção obstáculos e verificação de colisão

Para a detecção dos obstáculos é necessário equipar o robô com um sensor de distância adequado. Nesse trabalho utilizou-se um sensor de distância a laser (LIDAR) planar paralelo ao plano da base do UAV. Essa configuração restringe a operação do veículo ao plano de detecção do sensor que é considerado paralelo ao solo (altura constante) devido aos pequenos ângulos de inclinação do veículo. Considera-se também que está disponível a posição e a orientação do veículo em rela-

ção ao mundo, obtido através da combinação de GPS, IMU, barômetro e magnetômetros. Com essas informações, por meio de transformações de referências, os pontos obtidos pelo sensor do laser são representados no referencial fixo do mundo em três dimensões.

Uma vez que a informação do sensor esteja representada no referencial fixo do mundo, procura-se definir analiticamente as regiões dos obstáculos. Isto é de fundamental importância para o sistema, pois os testes de colisões para criação da árvore RRT* correspondem à etapa de maior custo computacional, podendo dispendir 99% de todo o tempo de processamento do algoritmo (Bialkowski et al., 2011). Como trata-se de um sistema embarcado com limitações de peso, espaço e potência elétrica, a eficiência do algoritmo é determinante para o sucesso da implementação.

Uma solução para a representação dos obstáculos e criação de um mapa local analítico seria crescer, para cada ponto do laser, uma área circular com o raio do veículo mais uma distância de segurança. No entanto, considerando que o laser possa ler 1041 pontos, seria necessário realizar esse mesmo número de operações para cada teste de colisão da árvore de RRT*. Como se trata do cálculo de maior custo do sistema, a velocidade de execução do algoritmo seria diretamente proporcional a esse número. Isso geraria um custo computacional elevado para se utilizar em um sistema embarcado no quadricóptero, inviabilizando uma implementação real.

Portanto propõe-se algumas considerações no algoritmo de detecção de obstáculos para reduzir o custo computacional (ver Figura 2).

1. A região de busca do algoritmo RRT* é limitado a um círculo de raio R_{rtt} centrado na posição do veículo. Essa seria uma distância suficiente para o robô planejar o caminho sem ter que testar colisões em todo espaço de trabalho.
2. São definidas distâncias mínima d_{min} e máxima d_{max} em que são considerados os pontos de medição do laser. O restante dos pontos são descartados.
3. Dentro da região de interesse, definida no item anterior, toda sequência de no máximo n pontos são agrupadas. Cada grupo de pontos é representado por um círculo cujo diâmetro é a distância entre o primeiro e último ponto da sequência. A Figura 2 ilustra essa abordagem na qual cada grupo é representado por um círculo de cor marrom.

Observe-se que para o problema em questão, a maioria dos obstáculos são troncos de árvores, que são objetos aproximadamente cilíndricos. Portanto, a consideração feita no item 3 é razoável. No caso de objetos não circulares, como barrancos

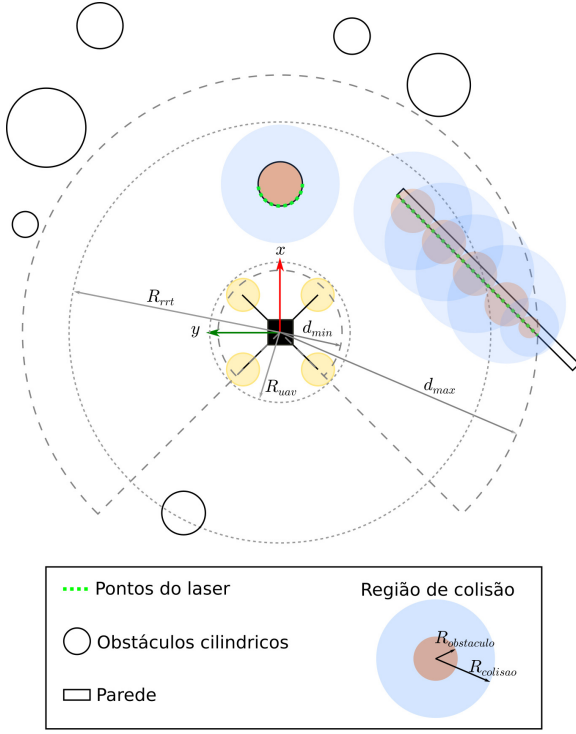


Figura 2: Representação do UAV ao centro e 8 obstáculos (7 árvores cilíndricas e uma parede). O laser mede a distância para todos os obstáculos, exceto um atrás do veículo, devido à limitação de seu campo de visão (270°). Apenas os obstáculos dentro da área entre $d_{\min}$ e $d_{\max}$ são utilizados para definir a região de colisão. Os obstáculos mais distantes, fora do raio R_{rrt} de busca do RRT*, não interferem no planejamento do caminho.

ou paredes, a limitação do valor de n pontos cria uma sequência de algumas circunferências. Nesse caso, a limitação do número de pontos (nesse artigo usa-se $n = 50$, escolhido empiricamente) é importante, pois um círculo muito grande poderia cobrir grandes áreas livres e impedir o movimento do veículo.

Todas as considerações feitas reduzem o número de operações de detecção de colisões para aproximadamente o número de obstáculos próximos ao robô (ver Figura 3). Em uma floresta esparsa esse número é de algumas unidades. No pior caso, que raramente aconteceria em uma floresta esparsa, se $n = 50$ e o número de pontos do laser é 1041, o número de obstáculos seria 21. Além disso, cada verificação de colisão se reduz a testar se uma configuração está ou não dentro de uma circunferência. Com isso tem-se ganhos expressivos de velocidade de processamento, viabilizando a implementação em tempo de execução, mesmo em hardwares limitados.

Portanto a região de colisão é definida por circunferências com o raio dado por $R_{colisao} = R_{obstaculo} + R_{uav} + R_{seguranca}$, onde: (i) $R_{obstaculo}$ é o raio estimado do objeto pelo algoritmo de trata-

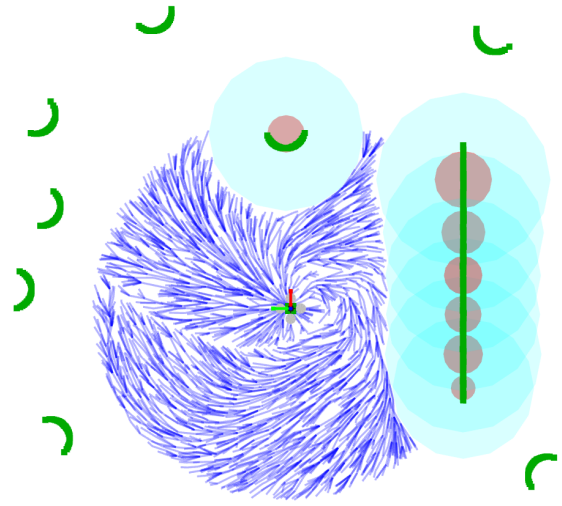


Figura 3: Visualização dos dados de simulação em um ambiente com vários obstáculos e um UAV com seu referencial ao centro. Os pontos medidos pelo laser planar estão em verde. Se estes pontos estão dentro da região de interesse, é estimado o obstáculo (cilindro marrom) e sua região de colisão (cilindro azul claro). Observe-se que a árvore do RRT* (em azul) não cresce na região de colisão. Nesse exemplo, há também uma parede ao lado direito do UAV que é subdividida em uma sequência de cilindros.

mento do laser, (ii) R_{uav} é o raio da circunferência que engloba o veículo e (iii) $R_{seguranca}$ é uma distância de segurança para evitar que o robô passe muito próximo aos obstáculos. Com a região de colisão devidamente definida é realizado planejamento do movimento descrito a seguir.

2.2 Planejador de movimento

A navegação utilizada nesse artigo é baseada no método proposto por Pereira et al. (2016). Esse método utiliza como planejador global um campo vetorial artificial. Esse campo é projetado com conhecimento parcial do ambiente e tem como meta o cumprimento da tarefa a ser executada pelo veículo. No caso específico desse trabalho, desejas-se que o veículo passe periodicamente por regiões da floresta. Assim, foi utilizado o campo vetorial para seguimento de curvas fechadas bidimensionais proposto em Gonçalves et al. (2010). Mais especificamente, foi utilizado o campo mostrado na Figura 4, que tem como curva alvo a função $\alpha(x,y) = x^4 + y^4 - 1000 = 0$.

Como a posição das árvores em uma floresta não é conhecida *a priori*, um planejador local é utilizado em conjunto com o campo vetorial para evitar colisões. Esse planejador local roda a cada atualização do mapa local obtido com o sensor a laser. Como proposto em Pereira et al. (2016), o planejador local utilizado nesse trabalho é o planejador probabilístico RRT*. Esse planejador en-

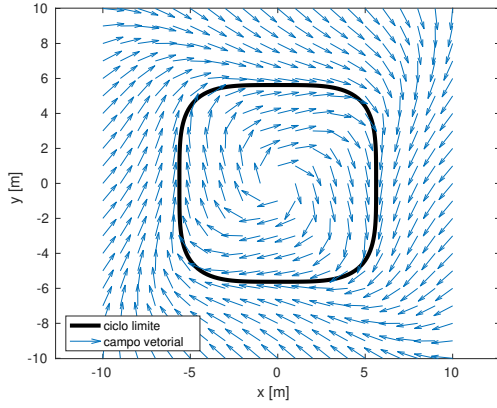


Figura 4: Campo vetorial utilizado neste trabalho. Ao seguir o campo o UAV converge para o ciclo limite.

contra um caminho livre de colisões que minimiza um funcional de custo especificado. O funcional utilizado nesse trabalho representa o quão próximo da integral do campo vetorial está o caminho. Assim, o custo é mínimo se o veículo segue perfeitamente o campo vetorial. Como muitas vezes as árvores impedem que o campo seja seguido exatamente, o planejador então encontra o caminho seguro que mais se aproxima do caminho ideal. Assim, o robô desvia dos obstáculos e continua a seguir a trajetória de interesse.

É importante mencionar que o RRT* está sendo utilizado como um planejador local, e a busca por melhores caminhos ocorre somente dentro de uma circunferência centrada na posição atual do robô, conforme proposto na subseção anterior. Como trata-se de um planejador probabilístico, esse fato pode causar o cálculo de caminhos descontínuos ou com homotopias diferentes, o que quer dizer, por exemplo, que o caminho calculado no instante t comanda o veículo a passar pelo lado direito de uma árvore e o caminho do instante $t + t_{\text{loop}}$ indica que o melhor caminho é pelo lado esquerdo. Situações como essa podem causar grandes problemas durante o voo, fazendo inclusive com que o veículo pare por indecisão.

Para evitar esse tipo de comportamento, calcula-se o caminho no tempo t e logo em seguida integra-se (simula-se) esse caminho por t_{loop} segundos encontrando-se o ponto p_{end} , que seria o ponto do caminho onde o veículo estaria após t_{loop} segundos. Enquanto o veículo real segue o caminho em direção a p_{end} , uma nova instância do planejador é então iniciada contendo p_{end} como início do caminho. Uma vez que o caminho é calculado, ele deve ser seguido pelo veículo com o uso de um controlador de caminhos, como mostrado a seguir.

2.3 Controlador de caminhos

Como resultado do planejador de movimento obtém-se um caminho definido como uma sequência de pontos em $\mathbb{R}^3$. Como assume-se operação em um plano, a coordenada z desse caminho é constante e indica a altura de voo desejado. Para o veículo seguir o caminho, é definido um campo vetorial de velocidades que possui uma componente atrativa, calculada como sendo proporcional ao menor vetor que liga o centro do veículo ao caminho, e uma componente paralela ao caminho, responsável por sua velocidade longitudinal.

Apesar do caminho não definir uma orientação para o veículo, a utilização de um controlador de orientação é importante para manter o sensor a laser sempre apontando para a direção de voo do veículo. Como o laser planar cobre um ângulo de 270° ou menos, sem o controle de orientação um obstáculo na frente do robô poderia não ser detectado. Portanto, nesse trabalho utiliza-se um controlador proporcional que reduz o erro do ângulo entre o vetor da velocidade medida e a frente do veículo.

A próxima seção apresenta resultados experimentais e simulações da metodologia proposta.

3 Experimentos

Essa seção apresenta resultados experimentais que ilustram e avaliam a metodologia proposta. A próxima subseção descreve o sistema utilizado nesses experimentos.

3.1 Sistema experimental

Os experimentos apresentados nesse artigo foram realizados com um quadricóptero marca DJI modelo Matrice 100 equipado com os seguintes equipamentos:

1. Sensor de distância à laser planar marca Hokuyo modelo 30LX-EW com 270° de campo de visão e 30 metros de alcance;
2. GPS e magnetômetro;
3. Computador de baixo nível e IMU, *DJI N1 flight controller* responsável pelo controle de voo do veículo;
4. Computador de alto nível Odroid XU4 contendo oito núcleos de até 2 GHz, arquitetura ARM e 2GB de memória RAM;
5. Câmera frontal Pentax Optio W80, utilizada apenas para gravar vídeos dos experimentos.

Uma foto do veículo é mostrada na Figura 5. O computador de alto nível está conectado a entrada UART do UAV utilizando um conversor USB/TTL. Por meio desse canal de comunicação, é possível obter informações do estado estimado

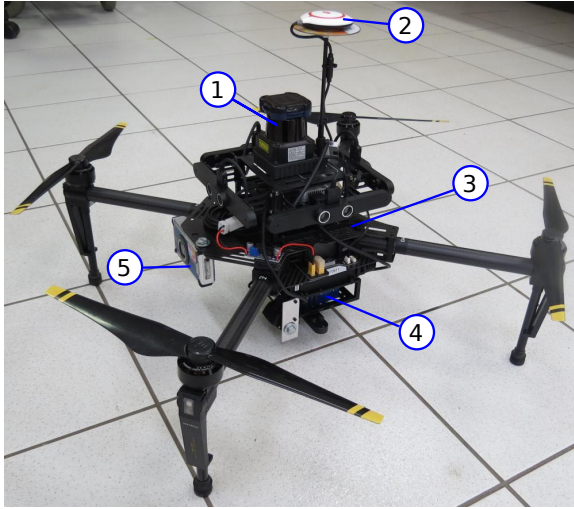


Figura 5: Equipamento utilizado nos experimentos reais. Quadricóptero com (1) laser planar, (2) GPS, (3) controlador de voo, (4) computador e (5) câmera frontal.

do robô e enviar comandos de velocidades. O sistema de navegação, percepção e controle do veículo mostrado nas seções anteriores foi executado no computador de alto nível Odroid, que utiliza o sistema operacional Ubuntu Mate 16.04 e o ROS (*Robot Operating System*) Kinect. O sistema de desvio de obstáculos foi desenvolvido com o auxílio da biblioteca OMPL (*Open Motion Planning Library*) e a comunicação com o robô utilizou o SDK da fabricante do veículo.

Considerando o veículo utilizado, tem-se os seguintes parâmetros para o método proposto: tempo de cálculo do RRT* $t_{loop} = 1$ s; $R_{rrt} = 5$ m; velocidade do UAV $v_{uav} = 1$ m/s; distâncias mínima e máxima dos pontos do laser $d_{min} = 0,4$ m e $d_{max} = 6,0$ m; raio do veículo $R_{uav} = 0,5$ m; e raio de segurança $R_{seguranca} = 2,0$ m. Com esses parâmetros foram realizados testes em simulação e ambiente real como mostrado a seguir.

3.2 Simulações

Para testar e validar a metodologia antes de realizar os experimentos no veículo real, foram realizados testes no ambiente de simulação Gazebo. A Figura 6 mostra um ambiente onde as árvores foram simuladas por cilindros. A Figura 7 ilustra o funcionamento do método nesse ambiente por meio de 4 instantes de tempo. Nessas figuras, a curva representando a integral do campo vetorial é mostrada em preto. Essa curva é inscrita em um quadrado de 11 m de lado. Na ausência de obstáculos, essa integral representa o caminho desejado para o veículo. A árvore RRT* calculada dentro do círculo de busca é mostrada em azul. Observe na Figura 7(a) que o veículo ainda não detectou nenhum obstáculo em seu raio de busca e portanto o caminho calculado, mostrado

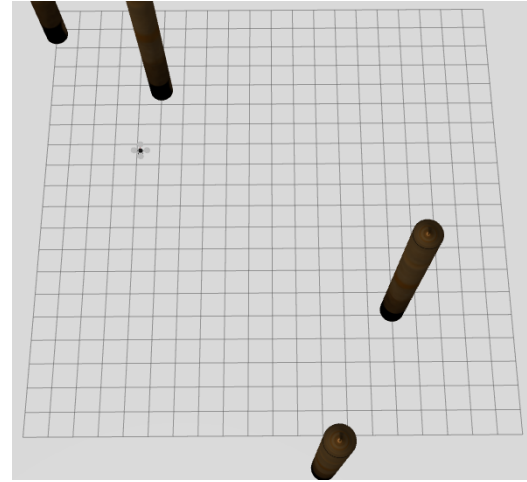


Figura 6: Cenário de simulação de um UAV no Gazebo com alguns cilindros representando os troncos das árvores.

em vermelho, o guia para a curva preta. Em (b) há uma interseção entre o raio do obstáculo e o raio de busca, causando uma deformação na árvore e consequentemente um caminho que diverge do obstáculo. Na sequência, mostrada em (c) e (d), observa-se que o veículo desvia do obstáculo detectado e volta a seguir o caminho especificado pelo campo vetorial.

3.3 Resultados em ambiente real

Com o veículo real, foram realizados experimentos no campus da UFMG em uma região com várias árvores suficientemente afastadas para não interferirem no sinal do GPS e com espaços que permitiam a navegação do veículo. A Figura 8 mostra a visualização dos dados de um experimento real típico. Assim como na simulação, a trajetória do robô converge para a integral do campo vetorial. Quando o laser detecta obstáculos no caminho o desvio é executado. Após desviar do obstáculo o robô volta a seguir o campo vetorial até convergir para o caminho planejado. Por se tratar de um caminho fechado o UAV continua seguindo a trajetória por várias voltas seguidas, até o usuário intervir e retomar o controle do veículo. As trajetórias do veículo em um experimento completo é mostrado na Figura 9. Um vídeo com os experimentos pode ser encontrado em: <https://goo.gl/mqas68>.

4 Conclusões e Trabalhos Futuros

Esse artigo apresentou uma abordagem para navegação de robôs aéreos em florestas esparsas. Os resultados apresentados mostram que a abordagem é promissora e pode ser utilizada no futuro na realização de tarefas de monitoramento, inspeção e busca em matas e florestas. Para que essas tarefas sejam plenamente realizadas em florestas mais

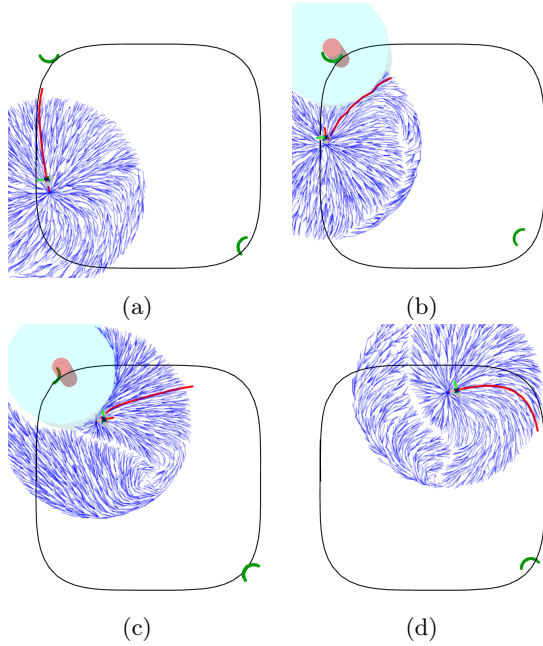


Figura 7: Simulação da metodologia no ambiente mostrado na Figura 6. A linha preta representa a curva a ser seguida, e é especificada pelo campo vetorial. A partir da posição estimada do robô, a árvore RRT* (em azul) cresce até um limite circular de raio R_{rrt} . Os pontos medidos pelo laser planar estão em verde. Em (b) estes pontos estão dentro da região de interesse. Nesse caso é estimado um obstáculo (cilindro marrom) e sua região de colisão (cilindro azul claro). O planejador então retorna o caminho (em vermelho) fora da região de colisão. Em (c) o veículo segue a trajetória e refaz os cálculos com os novos dados dos sensores. Após desviar do obstáculo o UAV volta a seguir o planejador global em (d).

fechadas, onde o maior problema são as constantes falhas do sinal de GPS, trabalha-se atualmente no desenvolvimento de um sistema de localização baseado em SLAM que utiliza as árvores para auxiliar na localização nos momentos em que o GPS é perdido. Trabalhos futuros também incluem um aumento da velocidade de translação do veículo, para aumentar a área coberta durante o voo. Para tanto, buscar-se-á uma eficiência ainda maior dos algoritmos de navegação, o que garantirá que se desvie de obstáculos de forma eficiente mesmo a altas velocidades. Outras propostas de continuidade são a substituição do laser planar por um laser 3D ou uma câmera RGB-D, o que permitirá o planejamento de movimento em $\mathbb{R}^3$.

Agradecimentos

Os autores agradecem a ajuda dos colegas do CORO, Vinícius Mariano, Antônio Chiella e Adriano Rezende. Esse trabalho foi parcialmente financiado pela CAPES, FAPEMIG, processo TEC-

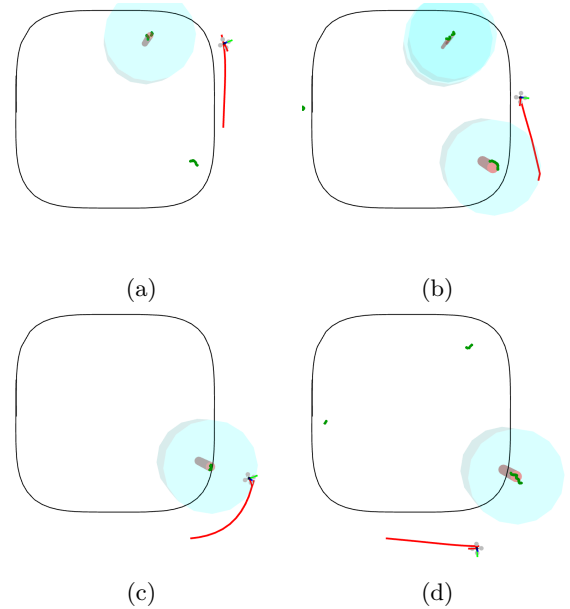


Figura 8: Resultados de um experimento real. A figura mostra os mesmos dados mostrados na simulação, exceto a árvore RRT*, cujos dados não foram gravados para não comprometer o desempenho de funcionamento do sistema. Observe que o veículo, representado por seu referencial, se desloca próximo a algumas árvores e desvia com sucesso das regiões de colisão.

APQ-00850-15 e CNPq, processo 465755/2014-3 (INCT). Henrique Machado e Guilherme Pereira são bolsistas do CNPq.

Referências

- Bialkowski, J., Karaman, S. and Frazzoli, E. (2011). Massively parallelizing the rrt and the rrt, *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3513–3518.
- Chisholm, R. A., Cui, J., Y.Lum, S. K. and Chen, B. M. (2013). Uav lidar for below-canopy forest surveys, *Journal of Unmanned Vehicle Systems* **01**(01): 61–68.
- Cui, J. Q., Lai, S., Dong, X., Liu, P., Chen, B. M. and Lee, T. H. (2014). Autonomous navigation of uav in forest, *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 726–733.
- Freitas, G., Gleizer, G., Lizarralde, F., Hsu, L. and dos Reis, N. R. S. (2010). Kinematic reconfigurability control for an environmental mobile robot operating in the amazon rain forest, *Journal of Field Robotics* **27**(2): 197–216.
- Gonçalves, V. M., Pimenta, L. C., Maia, C. A., Dutra, B. C. and Pereira, G. A. (2010). Vec-

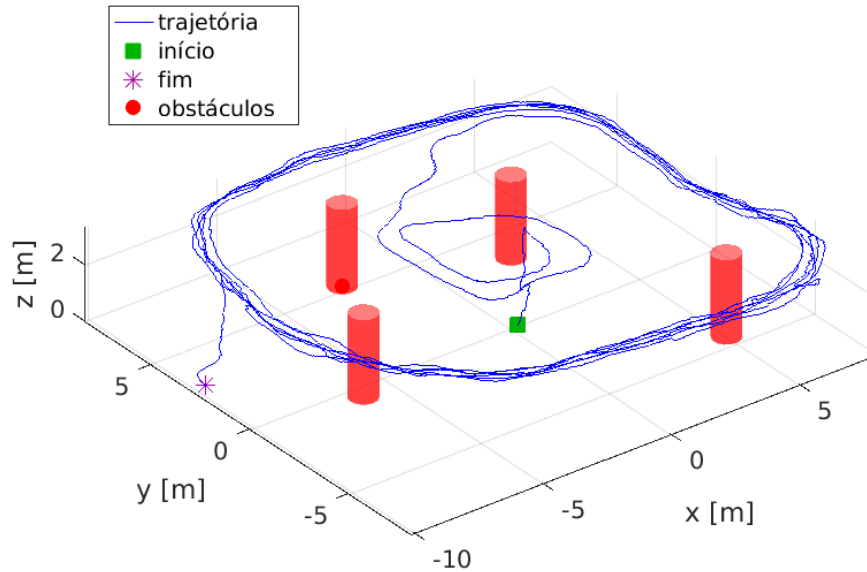


Figura 9: Trajetória percorrida pelo UAV no experimento em ambiente real. Os cilindros representam as árvores detectadas durante o movimento. Nesse experimento o veículo se deslocou automaticamente a partir da origem do sistema de coordenadas e, após várias voltas, pousou manualmente no ponto marcado com um asterisco.

tor fields for robot navigation along time-varying curves in n-dimensions, *IEEE Transactions on Robotics* **26**(4): 647–659.

Kamel, M., Alonso-Mora, J., Siegwart, R. and Nieto, J. (2017). Robust collision avoidance for multiple micro aerial vehicles using nonlinear model predictive control, *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 236–243.

Langelaan, J. and Rock, S. (2004). Navigation of small uavs operating in forests, *AIAA Guidance, Navigation, and Control Conference and Exhibit*.

Oleynikova, H., Taylor, Z., Fehr, M., Siegwart, R. and Nieto, J. (2017). Voxblox: Incremental 3d euclidean signed distance fields for on-board mav planning, *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1366–1373.

Pereira, G. A. S., Choudhury, S. and Scherer, S. (2016). A framework for optimal repairing of vector field-based motion plans, *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 261–266.

Pfeiffer, M., Schaeuble, M., Nieto, J., Siegwart, R. and Cadena, C. (2017). From perception to decision: A data-driven approach

to end-to-end motion planning for autonomous ground robots, *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1527–1533.

Schultz, A., Gilabert, R., Bharadwaj, A., de Haag, M. U. and Zhu, Z. (2016). A navigation and mapping method for uas during under-the-canopy forest operations, *2016 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, pp. 739–746.

Shah, U., Khawad, R. and Krishna, K. M. (2017). Detecting, localizing, and recognizing trees with a monocular mav: Towards preventing deforestation, *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1982–1987.

Singh, S., Majumdar, A., Slotine, J. J. and Pavone, M. (2017). Robust online motion planning via contraction theory and convex optimization, *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5883–5890.

Zhen, W., Zeng, S. and Soberer, S. (2017). Robust localization and localizability estimation with a rotating laser scanner, *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6240–6245.