

DESVIO DE OBSTÁCULOS POR ROBÔS SEMIAUTÔNOMOS USANDO  
PLANEJAMENTO DE CAMINHOS

ELIAS J. R. FREITAS\*, HENRIQUE A. B. PASSOS†, 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, Brasil**†Departamento de Pós-Graduação em Engenharia Elétrica - Universidade Federal de Minas Gerais*

Emails: eliasjof@ufmg.br, habp@ufmg.br, gpereira@ufmg.br

**Abstract**— This paper deals with the problem of obstacle avoidance by semi-autonomous mobile robots. These robots may not have a map of the environment or a global localization system, since its mission is controlled by a human operator. In such situations, deliberative path planning-based approaches for obstacle avoidance, which usually requires localization, are difficult to be applied. In this paper, we propose a planning based strategy for obstacle avoidance that relies on local maps and a series of local reference frames. In this approach, global localization is not required and simple odometry is sufficient to make the robot to avoid obstacles. Different from reactive strategies, usually used by semi-autonomous robots, the avoidance strategy using planning is intelligent, choosing a good local path to be followed by the robot. In this paper, the approach is illustrated using a mobile robot moving in a corridor. We show that the robot is able to efficiently avoid obstacles at its maximum speed.

**Keywords**— service robots, safe navigation, obstacle avoidance.

**Resumo**— Esse artigo lida com o problema de desvio de obstáculos por robôs móveis semiautônomos. Esses robôs podem não ter um mapa do ambiente ou um sistema de localização global, uma vez que sua missão é controlada por um operador humano. Nessas situações, métodos deliberativos baseados em planejamento de caminhos para desvio de obstáculos, que geralmente necessitam de localização, são difíceis de serem aplicados. Nesse artigo é proposta uma estratégia para desvio de obstáculos que se vale de mapas locais e uma série de referenciais locais. Nessa estratégia, localização global não é necessária, tornando a odometria do robô suficiente para fazê-lo desviar de obstáculos. Diferentemente de estratégias reativas, normalmente utilizadas por robôs semiautônomos, a estratégia de desvio usando planejamento é inteligente, escolhendo um bom caminho local para ser seguido pelo robô. Neste artigo, a estratégia proposta é ilustrada em um robô móvel se locomovendo em um corredor. Os resultados mostram que o robô, em sua velocidade máxima, é capaz de desviar eficientemente de obstáculos.

**Palavras-chave**— robôs de serviço, navegação segura, desvio de obstáculos.

## 1 Introdução

Com o desenvolvimento da robótica, robôs móveis de serviço tem se aproximado dos seres humanos, estando cada vez mais presentes em seus ambientes de trabalho, estudo e lazer (Dayoub et al., 2015). Uma peculiaridade de alguns desses robôs é a possibilidade deles serem diretamente comandados pelas pessoas, como é o caso de cadeiras de rodas inteligentes (Simpson et al., 2008), robôs que devem seguir as pessoas pelo ambiente (Müller et al., 2008) ou sistemas de assistência à direção de automóveis (Skog e Handel, 2009). O presente artigo está principalmente interessado nesse tipo de robô, onde o controle da missão é de responsabilidade do ser humano, estando o robô responsável por tarefas específicas, em especial a segurança da missão. Esses robôs semiautônomos se diferem dos demais robôs de serviço em pelo menos três pontos: (1) eles não precisam conhecer um mapa global do ambiente em que se deslocam; (2) eles não precisam conhecer o objetivo ou alvo final; e (3) eles não precisam se localizar no ambiente. Apesar dessas diferenças, esses robôs certamente ainda devem ser capazes de evitar colisões com obstáculos encontrados em seu caminho, preservando a si próprio e aos seus usuários. A proposta de uma estratégia para desvio inteligente de obs-

táculos por robôs semiautônomos é o alvo deste artigo.

Desde o início da robótica, estratégias para desvio de obstáculos tem sido propostas (Moravec, 1980; Khatib, 1986). Estas estratégias podem ser classificadas como reativas, onde o robô reage instantaneamente à presença de obstáculos recém detectados sem se preocupar com as consequências de sua reação, ou deliberativas, onde o robô, com o uso de mapas do ambiente, planeja um caminho que não só desvia dos obstáculos, mas também considera a sua missão principal.

As estratégias reativas são muito eficientes e não requerem localização global ou modelos complexos para os obstáculos e para o ambiente (Arkin, 1998). Assim, são muito utilizadas em robôs semiautônomos, como os considerados nesse artigo (Chik et al., 2016; Baklouti et al., 2017). No entanto, como estas estratégias não consideram o futuro, muitas vezes o robô pode ser levado a situações indesejadas que podem fazê-lo parar ou seguir trajetórias muito longas.

Estratégias deliberativas, por outro lado, planejam ou replanejam caminhos que consideram, não somente o obstáculo detectado, mas também o horizonte futuro de movimento do robô. Essa abordagem pode gerar trajetórias ótimas e segu-

ras. Um trabalho recente muito interessante na área é (Yang et al., 2016), que planeja a trajetória do robô em um ambiente com pessoas, considerando explicitamente as incertezas associadas à velocidade e posição dessas pessoas. Ao contrário das estratégias reativas, essa estratégia e as demais estratégias deliberativas dependem de mapas do ambiente e principalmente da localização global do robô, necessária para planejar e para seguir o caminho planejado (Choset, 2005). Esses requisitos, que também são necessários em estratégias híbridas que combinam técnicas deliberativas e reativas, podem ser difíceis de serem obtidos por alguns robôs semiautônomos, como as cadeiras de rodas se locomovendo em ambientes internos desconhecidos (Baklouti et al., 2017).

Visando aproveitar as vantagens dos métodos deliberativos, esse artigo propõe uma estratégia de desvio de obstáculos que utiliza planejadores de caminho globais em mapas locais do ambiente. Para não depender de localização global, a estratégia proposta utiliza uma sequência de referenciais locais com tempo de vida curto. Esses referenciais são utilizados pelo robô para planejar e seguir o último caminho criado. Transformações de coordenadas com baixa incerteza entre os referenciais locais são obtidas por meio da odometria, devido à proximidade geométrica entre esses referenciais. Se planejadores completos forem utilizados, a estratégia proposta garante desvio de obstáculos estáticos e, dentro de uma região do campo de visão do robô determinada por sua velocidade, velocidade do obstáculo e tempo de cálculo do planejador, também permite o desvio de obstáculos dinâmicos. Pelo conhecimento dos autores, a proposta do artigo, com caráter prático e tecnológico, é inovadora, uma vez que não foi encontrado outro artigo na literatura que usa estratégia similar para desvio de obstáculos. Os trabalhos anteriores que usam planejamento de movimento requerem a localização métrica do robô em relação a um referencial global fixo no ambiente (Minguez et al., 2016).

Para ilustrar a estratégia proposta, apresentada na Seção 2, utilizou-se um robô planar trabalhando em um corredor. Conforme será descrito na Seção 3, para o planejamento do caminho nesse ambiente, utilizou-se um planejador local assintoticamente ótimo, RRT\* (*Optimal Rapidly-exploring Random Trees*) (Karaman et al., 2011), cuja função de custo depende de um campo vetorial que codifica a direção de movimento especificada por um usuário humano, conforme proposto em (Pereira et al., 2016). Resultados dos experimentos são mostrados na Seção 4. Conclusões e trabalhos futuros são discutidos na Seção 5.

## 2 Metodologia

A metodologia proposta neste artigo assume a existência de um planejador de caminhos completo com tempo máximo de execução, ou seja, caso existam caminhos possíveis, o planejador deve for-

necer um caminho a cada intervalo de tempo,  $t_s$ . Essa característica é comum, por exemplo, em planejadores *anytime* (Karaman et al., 2011), que calculam um caminho rapidamente e o otimizam caso tenham mais tempo disponível.

Para detecção dos obstáculos e obtenção de um mapa local, a metodologia também assume a existência de um sensor de distância instalado no robô, como um laser (LADAR) ou um Kinect. Por fim, para garantir a localização do robô em relação a um referencial local, assume-se que o robô é dotado de um sistema de odometria, que pode ser composto por encoders, unidade de medição inercial ou até mesmo câmeras para odometria visual. A metodologia proposta é explicada em detalhes a seguir em forma de um algoritmo computacional.

### 2.1 Algoritmo proposto

A estratégia de desvio de obstáculos proposta nesse artigo pode ser resumizada no Algoritmo 1.

---

#### Algoritmo 1: Desvio de obstáculos.

---

```

1:  $\mathbf{k} = 0$ ;  $\mathbf{v}_r = 0.0$ ;  $\tau_0 = \{\}$ ;
2:  $L_0 = \text{DeterminaReferencial}()$ ;
3:  $\text{mapa}_0 = \text{ObtemMapa}(L_0)$ ;
4:  $\mathbf{k} \leftarrow \mathbf{k} + 1$ ;
5: enquanto VERDADE faça
6:    $L_k = \text{DeterminaReferencial}()$ ;
7:    $\tau_c = \text{TransformaCaminho}(\tau_{k-1}, L_k)$ ;
8:    $\text{mapa}_k = \text{TransformaMapa}(\text{mapa}_{k-1}, L_k)$ ;
9:    $\mathbf{o}_c = \text{Integra}(\tau_c, \mathbf{p}_r, t_s, \mathbf{v}_r)$ ;
10:  faça em paralelo:
11:     $\tau_s = \text{Planejador}(\text{mapa}_k, \mathbf{o}_c)$ ;
12:     $\mathbf{v}_r = \text{Controlador}(\tau_c)$ ;
13:  fim paralelo
14:   $\tau_k = \text{ConcatenaCaminho}(\tau_c, \tau_s, \mathbf{p}_r, \mathbf{o}_c)$ ;
15:   $\text{mapa}_k = \text{ObtemMapa}(L_k)$ ;
16:   $\mathbf{k} \leftarrow \mathbf{k} + 1$ ;
17: fim enquanto

```

---

Considerando que o tempo pode ser discretizado em um período  $t_s$ , correspondente ao tempo de cálculo do planejador escolhido, a metodologia proposta em um tempo  $t_k$ ,  $k = 1, 2, \dots$ , inicialmente determina um referencial local  $\{L_k\}$ , o qual será a referência durante o intervalo de tempo  $t_k - t_{k-1} = t_s$  (linha 6 do Algoritmo 1).

Após a determinação de  $\{L_k\}$ , transforma-se o caminho calculado pelo planejador e o mapa local obtidos em relação ao referencial  $\{L_{k-1}\}$  (calculado em  $t_{k-1}$ ) para  $\{L_k\}$  (linhas 7 e 8). Após essa transformação, cujo tempo de cálculo é muito pequeno em relação a  $t_s$  e pode ser desprezado, uma nova instância do planejador é lançada (linha 11). Como parâmetros, esse planejador utiliza o mapa local que acaba de ser transformado para  $\{L_k\}$  (linha 8) e o ponto de origem do novo caminho,  $\mathbf{o}_c$ . Nesse ponto, o leitor pode estranhar a ausência de um ponto alvo como entrada do planejador. De

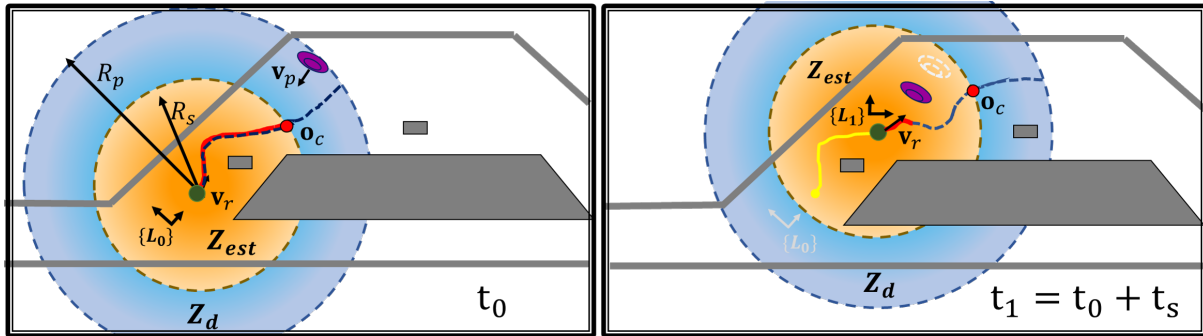


Figura 1: Estratégia de desvio de obstáculos proposta em dois instantes de tempo. O robô é representado como uma circunferência verde, possui uma velocidade  $\mathbf{v}_r$  e está dentro da Zona Estática ( $Z_{est}$ ). Em  $t_0$ , o robô começa a seguir o caminho representado pela linha tracejada em azul escuro. Esse caminho foi fornecido pelo planejador no referencial local  $L_0$ . O trecho desse caminho, sinalizado em vermelho, representa o caminho comprometido (*committed path*). Enquanto o robô segue esse caminho, desviando dos obstáculos que estão representados pela cor cinza, o planejador está calculando um novo caminho, tendo como origem o ponto  $\mathbf{o}_c$ . No instante  $t_1$ , o robô já percorreu o caminho em amarelo e o planejador fornece o novo caminho em azul, no novo referencial local  $L_1$ . Esse novo caminho considera, inclusive, uma pessoa em movimento representada pela cor lilás. O trecho do caminho comprometido que não foi cumprido até  $t_1$ , em vermelho, é concatenado com o novo caminho, formando o caminho completo,  $\tau_1$ . O novo ponto de origem determinado nesse instante é indicado por um círculo vermelho.

fato, como um robô semiautônomo nem sempre conhece seu destino, caso esse ponto seja necessário para o planejador em questão, ele pode ser escolhido arbitrariamente e sem perda de generalidade como um ponto à frente do robô e no limite do campo de visão de seus sensores.

O ponto de origem,  $\mathbf{o}_c$ , é obtido integrando-se (ou simulando-se) o caminho anterior (calculado em relação a  $\{L_{k-1}\}$  e transformado para  $\{L_k\}$ ), durante um tempo  $t_s$  a partir da posição real do robô,  $\mathbf{p}_r$ , representada no referencial atual (linha 9). Para essa integração é considerada a velocidade média do robô,  $\mathbf{v}_r$ . A ideia por trás de se escolher um ponto do caminho anterior como sendo o início do próximo caminho é manter a continuidade do caminho do robô. Com essa estratégia, evita-se, por exemplo, situações de caminhos pertencentes a classes homotópicas diferentes. Observe que essa estratégia define um trecho do caminho anterior a ser seguido enquanto o novo caminho é calculado. Esse trecho do caminho é chamado na literatura de caminho comprometido (*committed path*) (Karaman et al., 2011).

Enquanto é solucionado o novo problema de planejamento, ou seja, durante o intervalo  $t_s$ , um controlador de caminhos fornece os devidos comandos de velocidade para manter o robô seguindo o caminho comprometido (linha 12). Como esse caminho é enviado para o controlador já no referencial  $\{L_k\}$ , não há necessidade de transformação do caminho pelo controlador, bastando-se usar como realimentação a posição do robô em relação ao referencial atual ( $\{L_k\}$ ).

Uma vez que o planejador finaliza o seu cálculo, o caminho comprometido ainda não seguido pelo robô é concatenado com o caminho recém

calculado (linha 14), um novo mapa com as últimas informações dos sensores é obtido (linha 15) e o processo se repete fazendo a transformação do novo caminho e do novo mapa para o recém criado referencial  $\{L_{k+1}\}$ .

Note que para que todo o processo funcione adequadamente, sempre é necessário se conhecer a transformação entre o referencial anterior e o referencial atual. Como  $t_s$  é normalmente um tempo muito pequeno (na ordem de 1 a 2s), uma boa odometria é geralmente suficiente para calcular as transformações de coordenadas necessárias com baixa incerteza.

A Figura 1 ilustra a estratégia de desvio de obstáculos proposta neste trabalho. Pode-se observar em  $t = t_0$  o caminho planejado no referencial atual,  $\{L_0\}$ , que levou em consideração os obstáculos dentro do campo de visão do robô no instante anterior a  $t_0$ . Também está sinalizado de vermelho nesse caminho o ponto de partida que o planejador utilizará para fornecer o novo caminho mostrado em  $t_1 = t_0 + t_s$ . Nesse instante de tempo, é mostrado o novo referencial local atual,  $\{L_1\}$  e o caminho comprometido. A seguir é realizada uma análise do algoritmo proposto.

## 2.2 Análise

Uma maneira de analisar comportamento e a eficiência da metodologia proposta é avaliar o comportamento do robô com respeito às possíveis configurações dos obstáculos durante o intervalo de planejamento. Para facilitar esta análise são criadas duas zonas circulares centradas na posição do robô no tempo  $t_k$ ,  $k = 1, 2, \dots$ , mostradas na Figura 1: a Zona Dinâmica ( $Z_d$ ) e a Zona Estática ( $Z_{est}$ ).

A Zona Dinâmica é delimitada externamente

pelo raio de planejamento  $R_p$  e internamente pelo raio da Zona Estática  $R_s$ . O valor de  $R_p$  é, de fato, um parâmetro do planejador, que pode ser escolhido, por exemplo, em função da capacidade computacional (regiões maiores demandam mais cálculos), velocidade e número de obstáculos (um horizonte de planejamento maior pode resultar em trajetórias melhores) e campo de visão dos sensores (é necessário um conhecimento dos obstáculos dentro da região de planejamento). Independentemente da escolha de  $R_p$ , a estratégia proposta nesse artigo garantirá que o robô não entrará na Zona Dinâmica no intervalo de planejamento de trajetórias,  $t_s$ , e por isso, a ausência de colisões com obstáculos estáticos e dinâmicos nessa região será assegurada. Apesar do robô não se mover para a Zona Dinâmica, é interessante notar que o caminho planejado já prevê a existência de obstáculos nessa região se esses forem detectados pelos sensores no instante inicial do planejamento.

A Zona Estática, definida pelo raio  $R_s < R_p$ , é uma zona crítica na qual durante o intervalo  $t_s$  os obstáculos são considerados estáticos. Para obstáculos estáticos, a estratégia proposta garante a ausência de colisão nessa zona. O valor de  $R_s$ , representa uma relação de compromisso entre a velocidade do robô e o tempo de planejamento. Para o pior caso, quando um robô segue uma linha reta com velocidade máxima  $v_{max}$ , define-se:

$$R_s = v_{max} \cdot t_s, \quad (1)$$

ou seja, se a velocidade do robô e/ou o tempo de planejamento aumentarem, a Zona Estática torna-se maior. Como nesta zona o robô segue um plano calculado  $t_s$  segundos atrás (caminho comprometido), esse plano somente considera obstáculos que não se moveram nesse período. Assim, para evitar colisão com obstáculos dinâmicos nessa zona, a navegação deverá também apresentar um caráter reativo, ou seja, se um obstáculo for detectado muito próximo ao robô, este deverá parar imediatamente antes que ocorra uma colisão. Caso o tempo de solução do planejamento tendesse a zero ( $t_s \rightarrow 0$ ), o tamanho da região estática também tenderia a zero ( $R_s \rightarrow 0$ ) e, assim, a navegação poderia ser totalmente deliberativa e com desvio de obstáculos garantida, mesmo na existência de objetos móveis no ambiente.

Pode-se então concluir que o desempenho do algoritmo proposto depende de alguns fatores. Apesar de ser sempre possível desviar de obstáculos estáticos com segurança (desde que esses sejam percebidos pelos sensores do robô), o desvio de obstáculos dinâmicos depende do tempo de planejamento e da velocidade do robô. O desejável é então diminuir ao máximo o tempo de planejamento (por meio de algoritmos mais eficientes ou aumento do poder computacional) de forma a permitir que o robô se desloque com uma velocidade mais alta e mesmo assim seja capaz de desviar de

obstáculos dinâmicos sem parar o seu movimento.

### 3 Estudo de caso

Para exemplificar o algoritmo proposto em um robô real, utilizou-se o robô de serviço MARIA, cuja base móvel é um Pioneer3-AT equipado com sensores de contato (*bumpers*) e um sensor a laser SICK-LMS 100, com campo de visão de 270° e frequência de amostragem de 50Hz. O ambiente dos experimentos foi um dos corredores da Escola de Engenharia da UFMG de largura 2,35m. A estratégia proposta foi implementada em C++, utilizando o ROS (*Robot Operating System*) juntamente com a biblioteca OMPL (*Open Motion Planning Library*). O processamento foi realizado por um notebook com processador i7@2.4GHz rodando Linux Ubuntu 14.04.

#### 3.1 Planejador de caminhos

O objetivo do planejador de caminhos é gerar caminhos que levem o robô a seguir o corredor até o final, mantendo-se à uma distância desejada da parede da direita (mão convencional) e desviando de possíveis obstáculos, que não são conhecidos a priori. A direção em que o robô deve se mover é especificada pelo usuário, uma vez que o robô é semiautônomo e não conhece o objetivo final da tarefa. Como critério de parada para o algoritmo, foi usado um detector de cruzamento de corredores. Dessa forma, o robô tem autonomia e segue o corredor na direção especificada até que não seja detectado nenhum cruzamento, ou seja, até o fim do corredor.

Na implementação atual do método proposto, um mapa local do tipo *Octomap Occupancy Grid* (Hornung et al., 2013) é fornecido ao planejador. Esse mapa é criado utilizando os dados do laser momentos antes do início do planejamento, conforme proposto no Algoritmo 1.

Para os resultados desse artigo, foi escolhido um planejador probabilisticamente completo, assintoticamente ótimo e *anytime*, baseado no RRT\*. Esse planejador é descrito em (Pereira et al., 2016). A grande vantagem desse método para a estratégia proposta é o fato de ele não necessitar de um ponto alvo, já que esse ponto não é conhecido pelo robô semiautônomo. No lugar do alvo, para ditar o objetivo do caminho, esse método utiliza em seu funcional de custo um campo vetorial, que deve ser seguido pelo caminho na ausência de obstáculos. Para o caso em questão, foi especificado um campo vetorial bi-dimensional formado por uma componente longitudinal (paralela às paredes do corredor) constante e uma componente transversal proporcional à distância entre a posição real do robô e a linha situada a 0,6m da parede da direita. Essa distância foi escolhida de forma que o robô ficasse o mais próximo da parede, levando em consideração sua dimensão.



### 3.2 Localização e determinação dos referenciais locais

A localização local do robô é realizada por meio de odometria, calculada pelo sistema interno do robô a partir da combinação de dados dos *encoders* conectados às suas rodas e de um girômetro que mede sua velocidade angular. A cada intervalo de tempo, essa odometria é transformada para o referencial corrente. Esse referencial é atualizado a cada  $t_s$  segundos com base na orientação relativa entre o robô e a parede do corredor. Nesse problema em específico, a orientação do referencial foi escolhida como sendo a orientação da parede para facilitar o cálculo do campo vetorial, que tem uma das componentes paralela ao corredor.

Nos experimentos mostrados nesse artigo, a detecção das paredes do corredor foi realizada utilizando um método baseado na Transformada de Hough Probabilística (Galamhos et al., 1999). Os dados do laser foram convertidos em uma imagem binária antes da aplicação do método. Dessa forma, por meio da transformada de Hough pode-se detectar as retas mais significativas na imagem, que representam as paredes do corredor. Tendo-se obtido os parâmetros das retas, é possível determinar a distância do robô às paredes e a sua orientação com respeito às mesmas. Obtidas essas informações, define-se a orientação do referencial local atual,  $\{L_k\}$ , sempre paralela à parede e no centro do corredor. É importante mencionar que, conhecendo-se a largura do corredor, é possível determinar esse referencial mesmo que apenas uma das paredes seja detectada. Resultados com o sistema descrito nessa seção são mostrados a seguir.

## 4 Resultados

A Figura 2 mostra quatro instantes de execução da estratégia proposta. Durante o experimento, o robô foi configurado para se mover com sua velocidade máxima, 0,75 m/s. O tempo de cálculo do planejador,  $t_s$ , foi de 2,0 s e o raio de planejamento,  $R_p$ , foi de 6,0 m. Nessa figura, o mapa, conforme percebido pelo sensor laser do robô é mostrado em azul, a árvore RRT\* em vermelho e o caminho a ser seguido em verde. As Figuras 2(a) e (b) mostram instantes distintos do seguimento do caminho comprometido definido em  $t = 0$  s, cujo limite é o ponto vermelho. Durante o seguimento desse caminho, um novo caminho está sendo calculado pelo planejador. Em  $t = 2$  s, o cálculo desse caminho é finalizado e um novo caminho comprometido é definido e seguido (Figuras 2(c) e (d)). Um vídeo com os experimentos mostrados nessa seção pode ser encontrado em: <http://coro.cpdee.ufmg.br/movies/sbai2017>.

A Figura 3 mostra uma foto do robô MARIA durante os experimentos. Note que nesses experimentos não foi considerado o desvio de pessoas em movimento. Foram utilizados apenas cones de segurança como obstáculos ao movimento do robô.

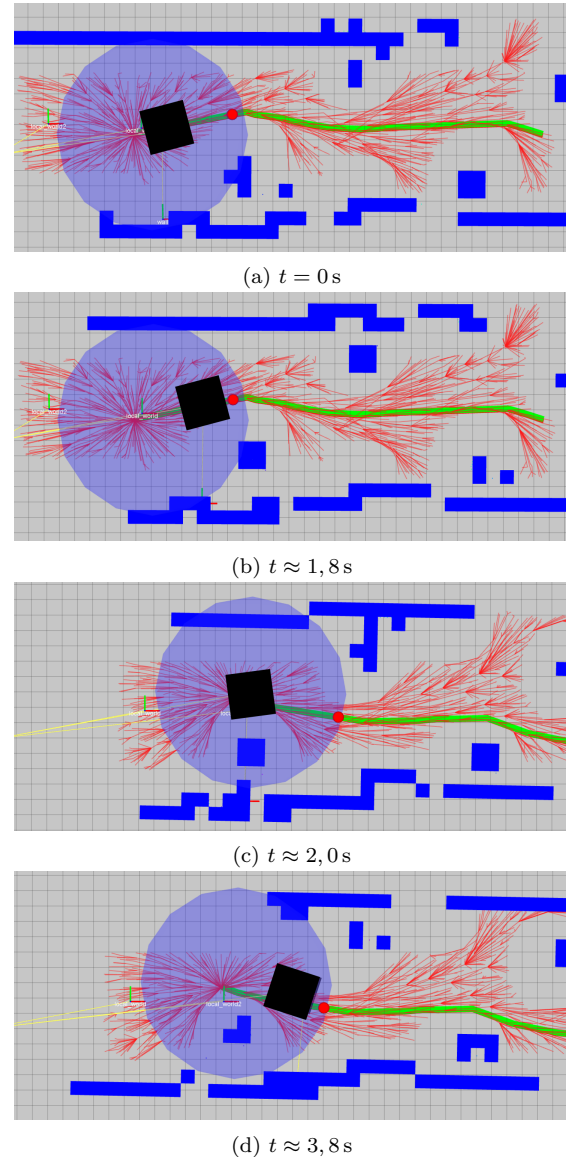


Figura 2: Dados experimentais obtidos com o robô se movendo em um corredor e desviando de obstáculos estáticos desconhecidos. O robô é representado por um retângulo preto e o caminho a ser seguido é indicado em verde. A árvore gerada pelo planejador baseado no RRT\* é mostrada em vermelho e o ponto de origem para o próximo caminho (ponto final do caminho comprometido) está em vermelho. A parede e os obstáculos são representados por quadrados azuis. O círculo em azul representa a Zona Estática, atualizada a cada intervalo de tempo de planejamento, definido como 2,0 s. (a) e (b) mostram instantes de seguimento do caminho comprometido definido em  $t = 0$  s. Durante o movimento do robô um novo caminho é calculado. Em  $t = 2,0$  s, esse novo caminho é usado para definir um novo caminho comprometido, cujo seguimento é mostrado em (c) e (d).



Figura 3: Foto do experimento da Figura 2.

## 5 Conclusão e Trabalhos Futuros

Esse artigo apresentou uma estratégia de desvio de obstáculos baseado em planejadores de movimento que não requer localização global. A ideia da estratégia proposta é manter um referencial local somente pelo tempo necessário para que o robô siga o último caminho calculado pelo planejador e para que um novo caminho seja calculado. Após esse tempo, um novo referencial local é definido e o caminho previamente calculado é transformado para o novo referencial. Como os referenciais locais estão muito próximos entre si, é possível definir uma transformação de baixa incerteza entre eles somente usando a odometria do robô. A estratégia proposta pode ser aplicada diretamente a robôs semiautônomos, que devem seguir trajetórias seguras mas não possuem um mapa do ambiente ou um sistema de localização global.

Uma análise do algoritmo proposto mostra que a estratégia é capaz de fazer o robô desviar de obstáculos estáticos e, dependendo da relação entre a velocidade do robô e o tempo de cálculo do caminho, também de obstáculos dinâmicos. Visando aumentar a chance de desvio de obstáculos dinâmicos, inclusive pessoas, trabalhos futuros considerarão um aumento da eficiência dos planejadores. Isso pode ser obtido, por exemplo, com a implementação de algumas de suas funções, como a detecção de obstáculos, em GPU ou algum hardware dedicado. Trabalhos futuros também incluem o teste da metodologia em veículos com dinâmicas diferentes, como uma cadeira de rodas inteligente e um veículo aéreo do tipo quadricóptero.

## Agradecimentos

O presente trabalho foi realizado com o apoio financeiro da FAPEMIG, do CNPQ e da CAPES - Brasil. Os autores agradecem o financiamento do INCT-SAC – Sistemas Autônomos Cooperativos (proc. CNPq 465755/2014-3)

## Referências

- Arkin, R. C. (1998). *Behavior-based robotics*, MIT press.
- Baklouti, E., Amor, N. B. e Jallouli, M. (2017). Reactive control architecture for mobile robot autonomous navigation, *Robotics and Autonomous Systems* **89**: 9 –14.
- Chik, S., Yeong, C., Su, E., Lim, T., Subramaniam, Y. e Chin, P. (2016). A review of social-aware navigation frameworks for service robot in dynamic human environments, *J. Telecom., Electronic and Computer Engineering* **8**(11): 41–50.
- Choset, H. M. (2005). *Principles of robot motion: theory, algorithms, and implementation*, MIT press.
- Dayoub, F., Morris, T. e Corke, P. (2015). Rubbing shoulders with mobile service robots, *IEEE Access* **3**: 333–342.
- Galamhos, C., Matas, J. e Kittler, J. (1999). Progressive probabilistic hough transform for line detection, *IEEE Conference on Computer Vision and Pattern Recognition*, p. 560 Vol. 1.
- Hornung, A., Wurm, K. M., Bennewitz, M., Stachniss, C. e Burgard, W. (2013). OctoMap: An efficient probabilistic 3D mapping framework based on octrees, *Autonomous Robots* **34**(3): 189–206. Software available at <http://octomap.github.com>.
- Karaman, S., Walter, M. R., Perez, A., Frazzoli, E. e Teller, S. (2011). Anytime motion planning using the RRT\*, *IEEE International Conference on Robotics and Automation*, pp. 1478–1483.
- Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots, *The international journal of robotics research* **5**(1): 90–98.
- Minguez, J., Lamiriaux, F. e Laumond, J.-P. (2016). *Motion Planning and Obstacle Avoidance*, Springer International Publishing, pp. 1177–1202.
- Moravec, H. P. (1980). Obstacle avoidance and navigation in the real world by a seeing robot rover., *Technical report*, DTIC Document.
- Müller, J., Stachniss, C., Arras, K. e Burgard, W. (2008). Socially inspired motion planning for mobile robots in populated environments, *Proc. of International Conference on Cognitive Systems*.
- Pereira, G. A., Choudhury, S. e Scherer, S. (2016). A framework for optimal repairing of vector field-based motion plans, *International Conference on Unmanned Aircraft Systems*, pp. 261–266.
- Simpson, R. C., LoPresti, E. F. e Cooper, R. A. (2008). How many people would benefit from a smart wheelchair?, *Journal of rehabilitation research and development* **45**(1): 53.
- Skog, I. e Handel, P. (2009). In-car positioning and navigation technologies—a survey, *IEEE Transactions on Intelligent Transportation Systems* **10**(1): 4–21.
- Yang, H., Lim, J. e eui Yoon, S. (2016). Anytime RRBT for handling uncertainty and dynamic objects, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4786–4793.