

UMA ESTRATÉGIA PARA NAVEGAÇÃO DE ROBÔS DE SERVIÇO SEMIAUTÔNOMOS USANDO INFORMAÇÃO LOCAL E PLANEJADORES PROBABILÍSTICOS

Data de aceite: 27/08/2020

Data de submissão: 03/08/2020

Elias José de Rezende Freitas

Instituto Federal de Educação, Ciência e
Tecnologia de Minas Gerais
Itabirito – Minas Gerais
<http://lattes.cnpq.br/4716860363006932>

Guilherme Augusto Silva Pereira

West Virginia University
Morgantown – West Virginia – EUA
<http://lattes.cnpq.br/1726523803257092>

RESUMO: Este trabalho propõe uma estratégia para a navegação de robôs semiautônomos baseada apenas em informações locais, obtidas pelos sensores instalados no robô e um planejador probabilístico que gera caminhos a serem seguidos localmente por ele, garantindo assim o desvio de obstáculos. A missão que o robô deve cumprir é definida pelo usuário por meio de uma sequência de comandos simples, como siga em frente e vire à direita, sendo esse comando codificado por campos vetoriais artificiais, que são utilizados pelo funcional de custo a ser otimizado pelo planejador. Os resultados obtidos demonstram que os robôs foram capazes de executar a missão definida pelo usuário, desviando eficientemente e de forma planejada de obstáculos estáticos e, para algumas situações bem definidas, dinâmicos, incluindo pessoas se locomovendo no ambiente durante a missão.

PALAVRAS-CHAVE: Robôs de serviço; Navegação; Desvio de obstáculos;

Planejadores probabilísticos.

SEMIAUTONOMISTS USING LOCAL INFORMATION AND PROBABILISTIC PLANNERS

ABSTRACT: This paper proposes a strategy for navigation of semi-autonomous service robots based on local information obtained by the sensors installed on the robot and on a probabilistic planner, which calculates paths to be followed locally by the robot, thus ensuring obstacle avoidance. The robot's mission is defined by a user through a sequence of simple commands, such as “go ahead” and “turn right”. These commands are encoded by artificial vector fields, which are used by the cost functional to be optimized by the planner. The results show that the robots were able to perform the user-defined mission efficiently, avoiding static obstacles and, for some well-defined situations, also dynamic obstacles, including people moving in the environment during the mission.

KEYWORDS: Service robots; Navigation; Obstacle avoidance; Probabilistic planners.

1 | INTRODUÇÃO

O desenvolvimento e a utilização de robôs têm crescido cada vez mais no mercado mundial. Eles já fazem parte da sociedade, estão presentes nas indústrias, nas ruas e dentro das casas (Dayoub, Morris e Corke, 2015). Dentre os vários robôs, existem aqueles que são destinados para fins

não-industriais, chamados de robôs de serviço.

Muitos desses robôs são ainda semiautônomos, por dependerem ou interagirem de alguma maneira com o ser humano. Alguns exemplos de robôs de serviço semiautônomos são: as cadeiras de rodas inteligentes (Simpson, LoPresti e Cooper, 2008), os robôs que devem seguir pessoas (Müller et al., 2008), os guias de museus (Dayoub, Morris e Corke, 2015) e robôs exploradores espaciais (Goldberg, Maimone e Matthies, 2002). Esses robôs se diferem dos demais robôs em pelo menos três aspectos: (i) eles não precisam conhecer o mapa global do ambiente em que se deslocam; (ii) eles não precisam conhecer o objetivo ou alvo final; e (iii) eles não precisam se localizar globalmente no ambiente. Contudo, esses robôs ainda devem ser capazes de cumprir a missão desejada pelo usuário de maneira segura, não colidindo com obstáculos (objetos, outros robôs, pessoas). O grande desafio é garantir isso em um ambiente desconhecido e dinâmico.

Para motivar este trabalho, considere a situação em que um robô, por exemplo, uma cadeira de rodas inteligente, entra em um prédio pela primeira vez, tendo que levar um usuário até uma determinada sala. Nessa situação, não se tem informações a priori do local (ambiente desconhecido), existem pessoas se movimentando (ambiente dinâmico) e, pelo fato de se ter um ambiente interno, não há sinal de GPS para facilitar a localização do robô. Assim, o conhecimento do usuário do local ou uma indicação fornecida por outra pessoa (um porteiro, por exemplo) deveria fornecer as informações necessárias para que robô autonomamente chegasse até a sala. Seria interessante que essas informações fossem fornecidas de maneira similar às orientações comumente dadas pelas pessoas, como: siga o corredor até o final, vire à direita e entre pela terceira porta. Repare que uma pessoa teria facilidade para seguir tais comandos, mesmo em um ambiente desconhecido e sem nenhuma localização global.

Assim, este trabalho visa apresentar e avaliar uma estratégia de navegação que permita que o robô haja como um ser humano e seja capaz de seguir instruções simples, mesmo sem que ele tenha uma localização global e o mapa do ambiente. Para tanto, o robô se baseará nos dados de detecção, por exemplo, de corredores, de interseções e de portas, obtidos com seus sensores.

A principal contribuição deste trabalho é então o desenvolvimento de uma estratégia de navegação a ser aplicada, principalmente, aos robôs de serviço semiautônomos.

Essa estratégia destaca-se das demais pelas seguintes características: (i) Mapeamento local: não há necessidade do conhecimento do mapa do ambiente; (ii) Replanejamento a cada intervalo de tempo de planejamento: com base no mapa local fornecido e a cada intervalo de planejamento, um planejador de movimento irá fornecer um caminho completo a ser seguido localmente pelo robô, garantindo que o robô desvie de obstáculos existentes no ambiente e cumpra a tarefa estabelecida por um usuário de forma planejada; (iii) Tarefas codificadas por meio de um campo vetorial artificial: o campo vetorial especificado para uma tarefa, por exemplo, “siga pelo corredor”, é incorporado diretamente

ao planejador de movimento, permitindo adicionar um comportamento ou uma regra social à navegação do robô; (iv) Criação de referenciais de curta duração: a cada intervalo de planejamento um novo referencial é criado e válido apenas nesse curto intervalo de tempo, de forma que apenas a odometria relativa ao novo referencial seja suficiente para que o robô siga o caminho fornecido pelo planejador, evitando as incertezas da odometria.

A próxima seção apresenta uma revisão bibliográfica sobre estratégias de navegação. A Seção 3 é dedicada a descrever a metodologia proposta em detalhes. A Seção 4 mostra a configuração experimental usada para validar a estratégia deste trabalho e a Seção 5 apresenta os resultados experimentais que ilustram o trabalho, utilizando robôs reais. Finalmente, conclusões e trabalhos futuros são discutidos na Seção 6.

2 | REVISÃO BIBLIOGRÁFICA

Para que um robô possa se locomover em um ambiente é necessário combinar métodos de localização e/ou de mapeamento e/ou de planejamento de movimento. Existem uma grande variedade de métodos (Choset et al., 2005) e a maneira na qual eles são combinados define uma determinada estratégia de navegação capaz de executar a missão designada ao robô e capaz de atender às restrições impostas tanto pelo ambiente, seja ele conhecido, parcialmente conhecido ou desconhecido, quanto pelo robô (Alves et al., 2011). Com o intuito de facilitar a compreensão dessas combinações, pode-se agrupar as estratégias de navegação em pelo menos três grupos: (i) Navegação local reativa, (ii) Navegação global com o conhecimento a priori do mapa do ambiente e (iii) Navegação local com mapeamento.

A navegação local reativa é baseada em uma arquitetura reativa de hardware e de software, introduzida no final da década de 80, que busca permitir uma resposta mais rápida às mudanças do ambiente (Brooks 1986). Dessa forma, o robô reage instantaneamente à presença de obstáculos recém detectados sem se preocupar com as consequências de sua reação. Os principais benefícios que facilitam a sua utilização em robôs semiautônomos, principalmente, daqueles que são comandados diretamente por seres humanos (Chik et al., 2016) (Baklouti, Amor e Jallouli, 2017), é o fato de serem muito eficientes e não necessitarem de localização global nem de modelos complexos para os obstáculos e para o ambiente. No entanto, uma desvantagem é que nessa estratégia não é levado em consideração o estado futuro do robô, já que não há um planejamento de movimento, podendo levar o robô a condições indesejadas, fazendo-o parar constantemente ou seguir trajetórias muito longas.

Já a navegação global com o conhecimento a priori de um mapa tem sido a mais utilizada, pois a maioria dos trabalhos assumem que o ambiente em que o robô será inserido é pré-definido, sendo possível obter um mapa prévio desse ambiente. Nessa estratégia, métodos de planejamento de movimento, como os métodos probabilísticos (Karaman et al., 2011), são utilizados para deliberar o movimento, de tal forma que os obstáculos estáticos

considerados no modelo serão evitados no cumprimento de sua missão. Exemplo desse tipo de estratégia pode ser encontrado em robôs que navegam em museus (Burgard et al. 1999) (Thrun et al. 1999), em escritórios (Marder-Eppstein et al., 2010) e em prédios (Araujo, Caminhas e Pereira, 2015). Para a obtenção do mapa desses locais, alguns trabalhos utilizam, em uma etapa anterior à navegação, as técnicas de Localização e Mapeamento Simultâneos (SLAM) (Dayoub, Morris e Corke, 2015) (Schwesinger et al., 2016) (Araujo, Caminhas e Pereira, 2015). Note que nem sempre é possível e simples para alguns robôs semiautônomos realizar esse procedimento, como, por exemplo, uma cadeira de roda inteligente se locomovendo em um ambiente interno desconhecido (Baklouti, Amor e Jallouli, 2017).

Por último, a navegação local com mapeamento é uma estratégia que a partir de um mapa obtido em tempo de execução permite que um planejador de movimento forneça caminhos a serem seguidos localmente e que são válidos no campo de visão dos sensores. Essa estratégia é encontrada, principalmente, em robôs exploradores (Stückler et al., 2015), sendo também considerada neste trabalho. O mapeamento dessa estratégia pode ser sem memória, se utilizar diretamente os dados dos sensores, ou com memória se durante a navegação o mapa é construído, usando, por exemplo, técnicas de SLAM. Esse tipo de estratégia combina tanto características reativas, já que não possui um mapa prévio do ambiente, quanto características deliberativas, por realizar um constante replanejamento local, levando em consideração obstáculos em um horizonte futuro.

3 | METODOLOGIA

A metodologia proposta neste trabalho foi estruturada conforme a arquitetura apresentada na Figura 1, considerando que a missão definida pelo usuário pode ser dividida em uma sequência de comandos simples, como virar à esquerda, virar à direita, seguir em frente e retornar.

Em síntese, esses comandos simples podem ser codificados por meio de um campo vetorial que será inserido no cálculo de um funcional de custo utilizado por um planejador de movimento.

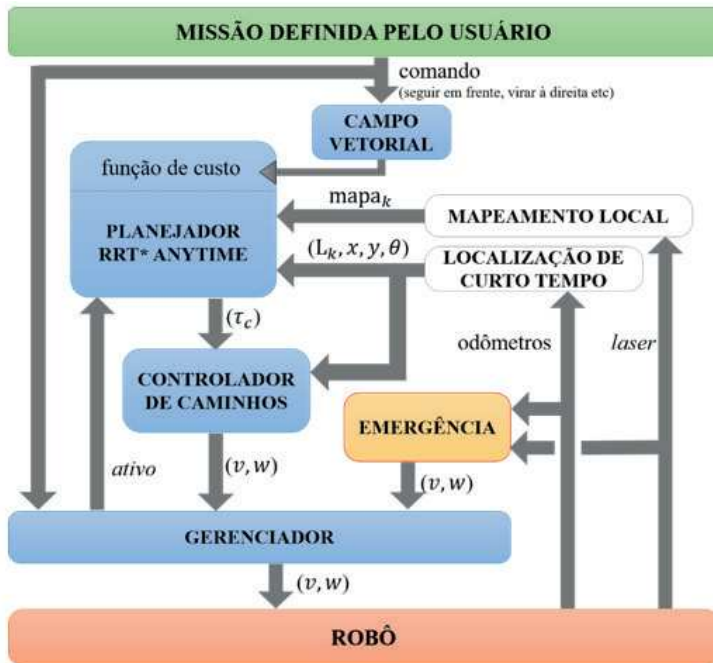


Figura 1 – Arquitetura de navegação proposta neste trabalho.

O caminho τ_c fornecido pelo planejador é utilizado por um controlador de caminho que calculará as devidas velocidades (v, ω) que o robô deve alcançar para seguir o caminho desejado. O Gerenciador da missão é o responsável por enviar as velocidades calculadas para o robô, além de ser capaz de executar alguns dos comandos simples mencionados anteriormente, e de parar o robô em caso de emergência.

Note que o planejador de movimento se baseia em um mapa local (mapa_k), obtido em tempo de execução a partir dos dados de um sensor à laser, e na localização do robô (x, y, θ) , ambos representados no mesmo referencial L_k , válido por um curto intervalo de tempo pré-determinado. Sendo assim, pode-se classificar essa estratégia de navegação como uma navegação local com mapeamento sem memória.

De forma a clarear a sequência de execução da estratégia deste trabalho, é apresentado o Algoritmo 1. Considera-se que o tempo possa ser discretizado em intervalos t_k , para $k = 1, 2, \dots$, onde o período $\Delta t = t_k - t_{k-1}$ é correspondente ao tempo de cálculo do planejador escolhido (tempo de planejamento t_s). Note que uma sequência de referenciais locais L_k será gerada na execução desse algoritmo, sendo que a cada período de planejamento um referencial local é usado como referencial fixo (linha 6 do Algoritmo 1).

Algoritmo 1: Estratégia de Navegação.

```
1   $k = 0$  ;  $v = 0.0$  ;  $\omega = 0.0$  ;  $\mathbf{v}_r = (v, \omega)$  ;  $\tau_0 \leftarrow \emptyset$  ;  $t_s =$  tempo de
   planejamento desejado ;
2   $L_0 =$  GeraReferencial();
3   $\text{mapa}_0 =$  ObtemMapa( $L_0$ );
4   $k \leftarrow k + 1$ ;
5  Enquanto ativo faça
6     $L_k =$  GeraReferencial();
7     $\tau_c =$  TransformaCaminho( $\tau_{k-1}, L_k$ );
8     $\text{mapa}_k =$  TransformaMapa( $\text{mapa}_{k-1}, L_k$ );
9     $\mathbf{o}_c =$  Integra( $\tau_c, \mathbf{p}_r, t_s, \mathbf{v}_r$ );
10   Faça em paralelo enquanto  $t \leq t_s$ :
11      $\tau_s =$  Planeja( $\text{mapa}_k, \mathbf{o}_c, t_s$ );
12      $\mathbf{v}_r =$  SegueCaminho( $\tau_c$ );
13   fim-paralelo
14    $\tau_k =$  ConcatenaCaminho( $\tau_c, \tau_s, \mathbf{p}_r, \mathbf{o}_c$ );
15    $\text{mapa}_k =$  ObtemMapa( $L_k$ );
16    $k \leftarrow k + 1$ ;
17 fim-enquanto
```

O laço de execução é realizado enquanto um sinal de ativo enviado pelo módulo gerenciador for verdadeiro. Os primeiros procedimentos do laço são gerar um novo referencial $\{L_k\}$ e em sequência transformar o caminho fornecido pelo planejador (τ_{k-1}) e o mapa local gerado a partir dos dados dos sensores (mapa_{k-1}), ambos obtidos em relação ao referencial $\{L_{k-1}\}$, para o novo referencial $\{L_k\}$ (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 . Pode-se estranhar a ausência de um ponto alvo como entrada do planejador. De 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 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 escolha, também se evita situações de caminhos pertencentes a classes homotópicas diferentes.

Observe que essa estratégia define um trecho do caminho anterior (entre a posição atual do robô \mathbf{p}_r e \mathbf{o}_c) 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 (linha 11), 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). Repare que esse caminho já está no referencial $\{L_k\}$, não havendo necessidade do Controlador de caminhos transformá-lo, utilizando apenas como realimentação a posição e orientação do robô em relação ao referencial atual ($\{L_k\}$). Passado o tempo previsto de planejamento t_s , quando o planejador retorna o novo caminho, 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 $\{L_{k+1}\}$.

É interessante ressaltar que, para que todo esse processo funcione adequadamente, sempre é necessário conhecer a transformação entre o referencial anterior e o referencial atual. Como t_s é, normalmente, um tempo muito pequeno (na ordem de 1s a 2s), uma boa odometria é geralmente suficiente para calcular as transformações de coordenadas necessárias com baixa incerteza, como poderá ser visto na Seção 5.

4 | IMPLEMENTAÇÃO

A estratégia proposta foi implementada em C++ e em Python, utilizando o ROS (Quigley et al., 2009) juntamente com a biblioteca OMPL (Şucan, Moll e Kavraki, 2012). A simulação foi realizada no StageROS e testada em dois robôs semiautônomos, vistos na Figura 2: (i) robô de serviço MARIA (*Manipulator Robot for Interaction and Assistance*) e (ii) uma cadeira de rodas em desenvolvimento no laboratório CORO/UFMG.

Os experimentos realizados exploram a aplicação na qual o robô navega em um ambiente desconhecido, não possuindo o mapa do ambiente. O cenário escolhido foi um prédio constituído de longos corredores de largura de 2,35 m, os quais havia alguns pilares que o estreitavam, continha alguns móveis e/ou objetos de decoração, mas que era essencialmente vazio, exceto pela presença de pessoas. Para testar o funcionamento da estratégia deste trabalho, foram também acrescentados outros obstáculos.

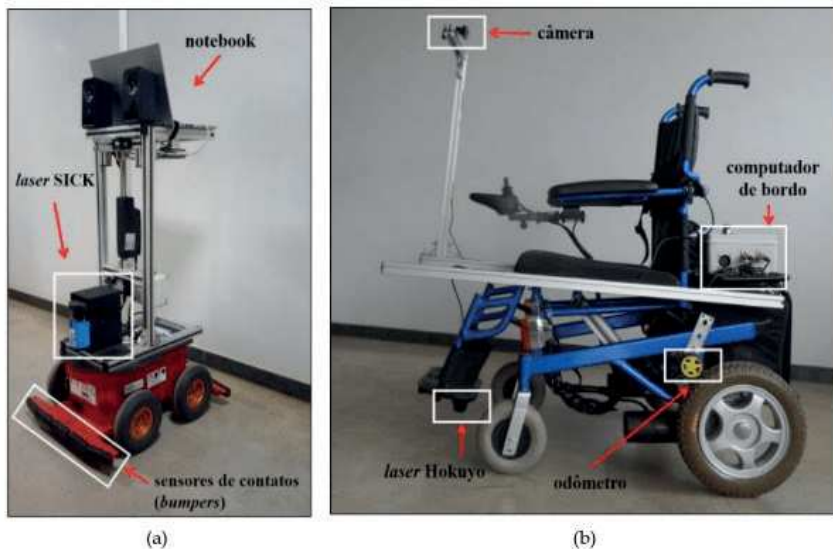


Figura 2 – Robôs de serviço semiautônomos utilizados nos experimentos reais. (a) Robô de serviço MARIA. (b) Cadeira de rodas inteligente em desenvolvimento no laboratório CORO.

5 | RESULTADOS

Nesta seção são apresentados os resultados de alguns dos experimentos realizados com o robô de serviço MARIA e com uma cadeira de rodas inteligente, sendo utilizado os mesmos parâmetros das simulações descritas em (Pereira e Freitas, 2020).

Primeiramente, o seguinte experimento foi realizado utilizando o robô MARIA: um usuário definiu a seguinte sequência de comandos: siga, vire à direita. Esses comandos foram repetidos onze vezes, de tal maneira que o robô completasse três voltas no ambiente.

A Figura 3(a) apresenta a odometria coletada durante a execução desse experimento, cujo vídeo completo pode ser encontrado em: https://youtu.be/loS_TxmtQMI. O robô foi capaz de desviar dos obstáculos existentes no ambiente, incluindo alguns cones que foram acrescentados para dificultar o cumprimento da missão.

Após a execução do experimento, observou-se que o robô terminou sua missão muito próximo do ponto de partida, o que não pôde ser observado pela odometria do robô, que apresenta um erro de integração razoavelmente grande. Mesmo com esse erro, o robô obteve sucesso na sua missão, evidenciando a importância dos referenciais de curta duração propostos na metodologia. Nesse experimento, cada referencial foi válido por apenas 1,0s, durante o qual a odometria apresenta resultados bons.

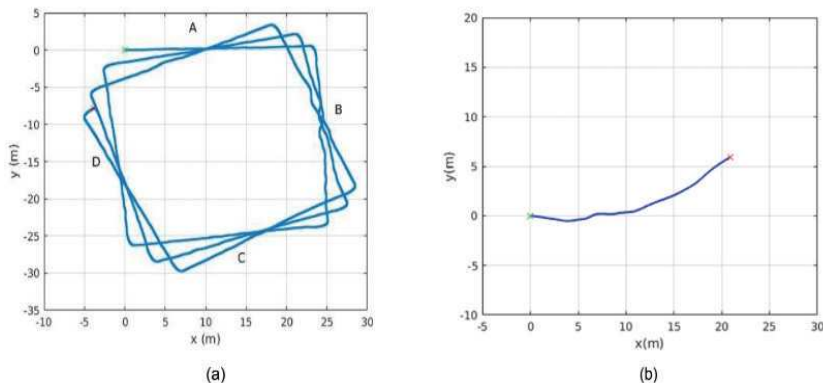


Figura 3 – Odometria coletada dos robôs: (a) MARIA durante a execução de três voltas (aprox. 300m) em um ambiente composto por quatro corredores identificados por A,B,C,D. No corredor B foram colocados obstáculos fixos (cones) no caminho do robô. (b) Cadeira de rodas inteligente ao percorrer o corredor A (aprox. 25m). Os robôs iniciam sua trajetória no ponto em verde (corredor A), finalizando no ponto em vermelho.

De forma muito similar, o segundo experimento foi realizado com a cadeira de rodas inteligente, sendo que o comando fornecido pelo usuário foi apenas siga. Dessa forma, a cadeira percorreu o corredor inteiro, mantendo-se próxima a parede à sua direita, até encontrar a primeira interseção com outro corredor. Durante esse trajeto, ela desviou com sucesso dos obstáculos estáticos presentes no ambiente. A Figura 3(b) mostra os dados da odometria durante a missão. Observe que o erro acumulado da odometria é muito maior que o do robô MARIA, chegando a quase 5m no eixo y ao fim do corredor. Novamente, mesmo com esse erro, a missão não foi comprometida devido aos referenciais de curta duração.

Para estimar como a incerteza da localização fornecida pela odometria desses robôs variou durante os experimentos, tomou-se como valor médio a própria odometria do robô e propagou-se as incertezas associadas às velocidades linear e angular dos robôs, obtidas experimentalmente, utilizando apenas a etapa de previsão do Filtro de Kalman Estendido (EKF).

Nas Figuras 4(a) e 5(a) são mostradas elipses para dois desvios padrão, o que representa um grau de confiança de aproximadamente 95 %. Observe que as áreas das elipses crescem a partir da posição inicial do robô, significando que há um aumento gradativo da incerteza da localização do robô. Esse aumento ao passar do tempo pode ser visualizado nas Figuras 4(b) e Figura 5(b). Apesar dessa incerteza sempre crescente, para a metodologia proposta, apenas a incerteza durante o tempo de planejamento t_s é, de fato, percebida pelo sistema. Como a única localização usada é relativa a um referencial criado recentemente, o erro acumulado é limitado a um valor muito pequeno, conforme mostrado nas Figuras 4(c) e 5(c), em que praticamente não se nota o crescimento das áreas das elipses e de maneira similar nas Figuras 4(d) e 5(d).

Com o intuito de também avaliar o comportamento do sistema na presença de obstáculos dinâmicos, uma pessoa moveu-se em direção ao robô, enquanto esse executava o comando siga. O robô MARIA foi configurado para uma $v_{\max} = 0,9$ m/s e o tempo de planejamento foi reduzido para $t_s = 0,45$ s. A redução do tempo de planejamento foi necessária para garantir o desvio de uma pessoa andando em uma velocidade normal de aproximadamente 1,0m/s.

Também esse experimento foi realizado utilizando a cadeira de rodas, porém dada a dificuldade de se controlar a velocidade dessa cadeira com um tempo de resposta pequeno, seja pela sua dinâmica seja pelo fato de não ser possível acionar diretamente os motores da mesma, foi configurada uma velocidade menor, $v_{\max} = 0,3$ m/s, e a pessoa caminhou com uma velocidade em torno de 0,25 m/s. Repetiu-se dez vezes esse experimento e em todos os testes os robôs foram capazes de desviar da pessoa. O vídeo completo pode ser visto em <https://youtu.be/DONxBJJ5OT8> (robô MARIA) e em <https://youtu.be/DilcPul18kc> (cadeira de rodas inteligente).

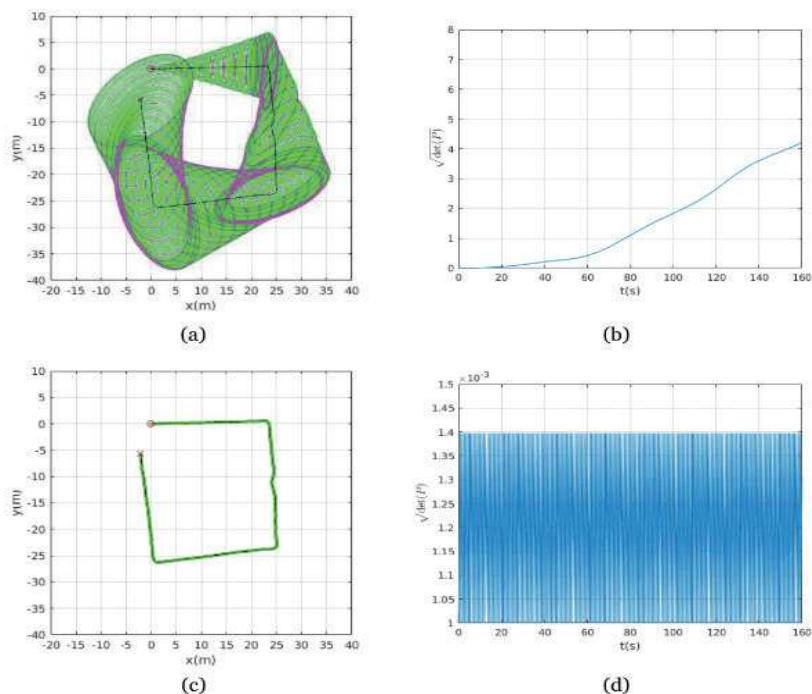


Figura 4 – Estimativa da covariância da odometria durante a primeira volta da missão, utilizando o robô MARIA. Em (a) é apresentada a estimativa da incerteza da posição do robô, representada pelas elipses, usando dois desvios padrão, ao longo dos dados da odometria em preto. Em (b) é apresentado a variação da estimativa total da incerteza de localização em relação ao tempo. Em (c) e (d) é ilustrado como a estratégia de navegação proposta neste trabalho percebe a incerteza de localização para um tempo de planejamento = 1,0s.

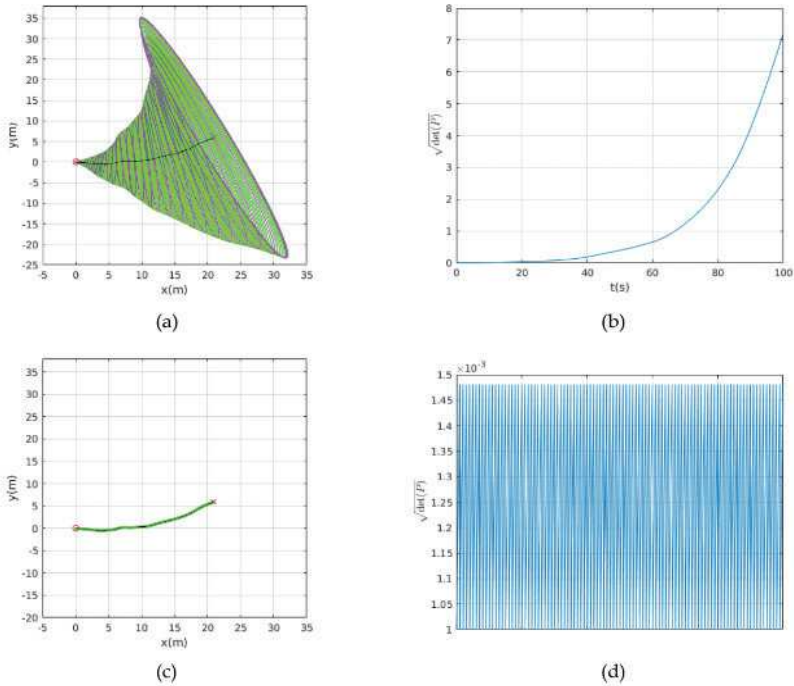


Figura 5 – Estimativa da covariância da odometria durante a navegação da cadeira de rodas inteligente por um corredor. Em (a) é apresentada a estimativa da incerteza da posição da cadeira de rodas, representada pelas elipses, usando dois desvios padrão, ao longo dos dados da odometria em preto. Em (b) é apresentada a variação da estimativa total da incerteza de localização em relação ao tempo. Em (c) e (d) é ilustrado como a estratégia de navegação proposta neste trabalho percebe a incerteza de localização para um tempo de planejamento $t_s = 1,0$ s.

6 | CONCLUSÃO E TRABALHOS FUTUROS

Neste trabalho propôs-se uma estratégia de navegação para robôs de serviço semiautônomos baseada em um planejador probabilístico que não requer a localização global do robô. Essa estratégia é destinada, principalmente, aos robôs que devem navegar com segurança, mas não possuem o mapa do ambiente nem um sistema de localização global. A ideia básica da estratégia é manter um referencial local válido somente durante o tempo necessário para que o robô siga parte do último caminho fornecido pelo planejador enquanto é computado um novo caminho que considere possíveis mudanças no espaço de trabalho do robô. Após o novo caminho ser calculado, um novo referencial é definido e o caminho é transformado para o novo referencial. O princípio da estratégia é válido já que esses dois referenciais, o anterior e o atual, estão muito próximos e, portanto, é possível definir uma transformação de coordenadas entre eles usando apenas a odometria.

Como o nosso método foi projetado para ser usado em robôs semiautônomos, foi

proposta a codificação de comandos simples do usuário por meio de campos vetoriais. Isso permitiu o uso de trabalhos anteriores que integram planejadores ótimos e campos de vetoriais para determinar o planejador de movimento utilizado neste trabalho, como apresentado em (Pereira, Choudhury e Scherer, 2016). Por meio dos experimentos realizados, verificou-se que o robô foi capaz de cumprir a missão dada pelo usuário, seguindo o mais próximo possível o campo vetorial e evitando obstáculos desconhecidos de forma planejada.

Analisando o algoritmo proposto, observou-se que um robô que executa essa estratégia é capaz de evitar obstáculos estáticos e, dependendo da relação entre a velocidade do robô, a velocidade do obstáculo e o tempo de planejamento, é também capaz de evitar obstáculos dinâmicos, incluindo pessoas, como mostrado em nossos experimentos. Portanto, trabalhos futuros pretendem aumentar a eficiência dos planejadores, por meio, por exemplo, da implementação de algumas funções em GPU ou em outro hardware dedicado. A previsão do movimento do obstáculo também é algo que poderia aprimorar este trabalho.

Uma desvantagem de nossa abordagem, que não foi crítica para nossos experimentos, mas que seria relevante se o robô pudesse se mover mais rápido, é a ausência de restrições diferenciais e/ou restrições dinâmicas no problema de otimização. Isso significa que o planejador atual pode encontrar um caminho que não pode ser seguido pelo robô, a menos que ele reduza drasticamente sua velocidade ou, até mesmo, tenha que parar. Além disso, sem restrições dinâmicas, a metodologia pode ser inviável para o uso em cadeiras de rodas inteligentes, um dos robôs semiautônomos mais comuns, já que o usuário poderia sofrer acelerações bruscas durante o movimento do robô. Uma possível solução seria utilizar um planejador de trajetória ao invés de um planejador de caminhos ou, como uma aproximação, utilizar os caminhos de Dubins para estender a árvore gerada pelo RRT*, como mostrado em (Karaman e Frazzoli, 2010).

REFERÊNCIAS

ALVES, S. F. R. et al. **Conceptual bases of robot navigation modeling, control and applications**. In: BARRERA, A. (Ed.). *Advances in Robot Navigation*. Rijeka: InTech, 2011. cap. 1.

ARAUJO, A. R.; CAMINHAS, D. D.; PEREIRA, G. A. **An architecture for navigation of service robots in human-populated office-like environments**. *IFAC-PapersOnLine*, v. 48, n. 19, p. 189 – 194, 2015. ISSN 2405-8963.

BAKLOUTI, E.; AMOR, N. B.; JALLOULI, M. **Reactive control architecture for mobile robot autonomous navigation**. *Robotics and Autonomous Systems*, v.89,p.9–14, 2017.

BROOKS, R. **A robust layered control system for a mobile robot**. *IEEE Journal on Robotics and Automation*, IEEE, v. 2, n. 1, p. 14–23, 1986.

BURGARD, W. et al. **Experiences with an interactive museum tour-guide robot**. *Artificial*

Intelligence, v. 114, n. 1-2, p. 3–55, 1999.

CHIK, S. et al. **A review of social-aware navigation frameworks for service robot in dynamic human environments.** *Journal of Telecommunication, Electronic and Computer Engineering*, v. 8, n. 11, p. 41–50, 2016. ISSN 2180-1843.

CHOSSET, H. et al. **Principles of Robot Motion: Theory, Algorithms, and Implementations** (*Intelligent Robotics and Autonomous Agents series*). Cambridge: The MIT Press, 2005. ISBN 978-0262033275.

CORKE, P. **Robotics, Vision and Control.** Berlin: Springer Berlin Heidelberg, 2011. ISBN 978-3642201431. Disponível em: <<https://doi.org/10.1007/978-3-642-20144-8>>.

DAYOUB, F.; MORRIS, T.; CORKE, P. **Rubbing shoulders with mobile service robots.** *IEEE Access*, v. 3, p. 333–342, 2015.

OLDBERG, S.; MAIMONE, M.; MATTHIES, L. **Stereo vision and rover navigation software for planetary exploration.** In: IEEE. *Proceedings of the IEEE Aerospace Conference*. 2002. v. 5, p. 2025–2036.

KARAMAN, S.; FRAZZOLI, E. **Optimal kinodynamic motion planning using incremental sampling-based methods.** In: IEEE. *Decision and Control (CDC), 2010 49th IEEE Conference on*. [S.l.], 2010. p. 7681–7687.

KARAMAN, S.; FRAZZOLI, E. **Sampling-based algorithms for optimal motion planning.** *The International Journal of Robotics Research*, Sage Publications, v. 30, n. 7, p. 846–894, 2011.

KARAMAN, S. et al. **Anytime motion planning using the RRT*.** In: *Proceedings of the IEEE International Conference on Robotics and Automation*. IEEE, 2011. p. 1478–1483. 2011.

MARDER-EPPSTEIN, E. et al. **The office marathon: Robust navigation in an indoor office environment.** In: *Proceedings of the IEEE International Conference on Robotics and Automation*. IEEE, 2010. p. 300–307.

MÜLLER, J. et al. **Socially inspired motion planning for mobile robots in populated environments.** In: *Proceedings of the International Conference on Cognitive Systems*. Springer, 2008. p. 85–90.

PEREIRA, G. A.; FREITAS, E. J. **Navigation of semi-autonomous service robots using local information and anytime motion planners.** *Robotica*, Cambridge University Press, p. 1–19.

PEREIRA, G. A. S.; CHOUDHURY, S.; SCHERER, S. **Kinodynamic Motion Planning on Vector Fields using RRT*.** Pittsburgh, PA, July 2016.

QUIGLEY, M. et al. **ROS: an open-source robot operating system.** In: *ICRA Workshop on Open Source Software*. [S.l.: s.n.], 2009.

SCHWESINGER, D. et al. **A smart wheelchair ecosystem for autonomous navigation in urban environments.** *Autonomous Robots*, Springer Nature, v. 41, n. 3, p. 519–538, 2016.

SIMPSON, R. C.; LOPRESTI, E. F.; COOPER, R. A. **How many people would benefit from a smart wheelchair?** *Journal of rehabilitation research and development*, REHABILITATION RESEARCH & DEVELOPMENT SERVICE, v. 45, n. 1, p. 53, 2008.

STÜCKLER, J. et al. Nimbro explorer: **Semiautonomous exploration and mobile manipulation in rough terrain.** *Journal of Field Robotics*, v. 33, n. 4, p. 411–430, 2015.

ŞUCAN, I. A.; MOLL, M.; KAVRAKI, L. E. **The Open Motion Planning Library.** *IEEE Robotics & Automation Magazine*, v. 19, n. 4, p. 72–82, 2012.

THRUN, S. et al. **MINERVA: a second-generation museum tour guide robot.** In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 1999. v. 3, p. 1999–2005.