

Technical Report

Frontier Exploration with Occupancy Grids and Roadmaps

Alysson Ribeiro da Silva
 Computer Science Graduate Program
 Federal University of Minas Gerais
 Belo Horizonte, Brazil

Abstract—Nesse projeto é apresentado a implementação e implantação de um controlador híbrido, deliberativo e reativo, para robôs autônomos explorarem um ambiente desconhecido. O método de exploração de fronteiras proposto por Yamauchi foi utilizado. A proposta foi avaliada utilizando um robô `pioneer3dx` no `CoppeliaSim`, onde o robô foi capaz de navegar, explorar, e mapear de forma autônoma.

Index Terms—Frontier Exploration, Exploration, Path-planning



1 INTRODUCTION

A percepção, planejamento, atuação, e controle são essenciais para garantir que um robô móvel autônomo seja capaz de executar tarefas, como a navegação e exploração, em um ambiente desconhecido [1–5]. Um dos grandes problemas relacionado aos mesmos é falta de certeza sobre o ambiente, sensores, e atuadores. Além disso, a alta complexidade de integração das 4 grandes áreas citadas torna difícil a tarefa de criar um controlador, reativo, deliberativo, ou híbrido, que seja robusto e tolerante a falhas para exploração e navegação.

Alguns dos problemas citados podem ser resolvidos com técnicas probabilísticas para mapeamento como as `occupancy grids`, controladores dinâmicos e cinemáticos, planejadores de caminhos como as `Rapidly-exploring random trees` e `Roadmaps`, controladores híbridos, reativos, e deliberativos que visam de alguma forma explorar vizinhanças ou fronteiras não exploradas. Com isso em mente, o objetivo deste projeto é propor, implementar, implantar, validar, e avaliar um controlador híbrido. A proposta visa a utilização de `occupancy grids` para representar o ambiente incerto e desconhecido, visa extrair representações do ambiente para deliberação e navegação através de exploração de fronteiras com auxílio de `Roadmaps`, e visa incorporar tolerância contra colisões indesejadas.

Nesse projeto foram implantadas, adaptadas, propostas as seguintes características:

- Criação e manutenção de `occupancy grid`.
- Criação e manutenção de `c-space` para deliberação e planejamento de caminhos.
- Proposta e criação de filtros para detecção de fronteiras utilizando `occupancy grids` e `c-space`.
- Utilização de método simples para seleção de fronteiras por proximidade.
- Criação e modificação/adequação do `A*` para planejamento de `roadmaps`.
- Navegação por sub-goals por comportamento deliberativo.

- Tolerância a colisões por comportamento reativo.
- Utilização, implantação, adequação de controladores cinemáticos e dinâmicos.
- Criação de mapas em tempo real.
- Visualização de caminhos em tempo real.
- Comportamento de retorno para base.

Para avaliar o desempenho das propostas foram realizados dois experimentos principais. O primeiro experimento avalia o desempenho do controlador ao navegar e explorar um ambiente desconhecido de nível fácil. Por outro lado, o segundo experimento avalia o comportamento do controlador nos mesmos aspectos, porém em um ambiente mais difícil. Além disso, são feitas algumas observações sobre o comportamento dos sensores e atuadores.

Ao fim deste documento são apresentadas instruções para execução da simulação utilizando o código fonte disponibilizado em conjunto ao `CoppeliaSim`.

2 METHOD

O projeto contempla a implementação e integração de `occupancy grids` para exploração de fronteiras com auxílio do `c-space`, `roadmaps`, heurística `A*`, e controlador cinemático. Como ilustrado pela Figura 1, o comportamento geral do robô é guiado por um controlador híbrido composto por 3 etapas. A primeira etapa é responsável pela geração, atualização, e manutenção do `occupancy grid`. Já a segunda etapa é responsável por gerar, filtrar, e atualizar um `c-space` correspondente. Por fim, a última etapa visa selecionar qual comportamento é mais apropriado para explorar o `c-space` gerado. Durante a escolha do comportamento, o controlador híbrido pode decidir entre escolher a fronteira mais próxima para explorar navegação, ativar um comportamento reativo para recuperar de configurações que possam causar uma colisão, ou ativar um comportamento de fim de exploração para que o robô possa voltar para a posição inicial ou para a base.

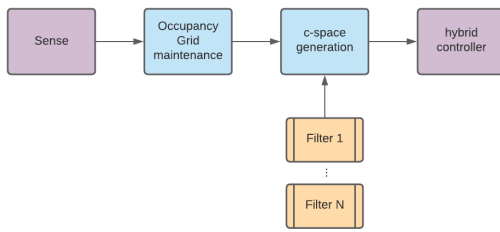


Fig. 1. Diagrama geral sobre a ordem de processamento, geração, atualização da occupancy grid e c-space.

2.1 Comportamento Híbrido do Robô

O controlador proposto é composto com uma camada reativa e outra deliberativa. O mesmo é ilustrado pela Figura 3, onde os estados representados pela cor verde são executados somente uma única vez durante a simulação e os estados representados pela cor laranja são executados diversas vezes em uma mesma simulação.

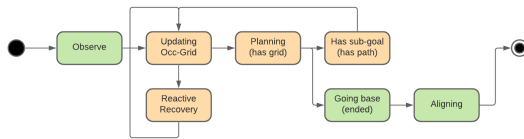


Fig. 2. Máquina de estados do controlador do robô.

Todos os comportamentos são descritos as seguir.

- **Observe** - Quando o robô está nesse estado, o mesmo executará um giro ao redor do próprio eixo para verificar fronteiras ao seu redor. O objetivo desse comportamento é permitir um mapeamento prévio dos arredores sem comprometer o robô ou ambiente.
- **Updating Occ-Grid** - Esse estado permite o robô atualizar a occupancy grid. O mesmo é executado por um número de vezes definido a-priori. A intuição por trás dessas execuções é permitir que as células da occupancy grid converjam para evitar ruídos na criação e filtragem do c-space.
- **Planning** - No estado de planejamento, o robô executará os algoritmos descritos nas Seções ?? e ?? para gerar um conjunto de sub-objetivos até a fronteira mais próxima.
 - Caso não existam mais caminhos ou fronteiras o robô deve ir para o estado **Going base** que enviará o mesmo de volta para a posição inicial ou para a base.
 - Caso um caminho seja encontrado, então o robô irá para o estado **Has sub-goal** e deverá iniciar a navegação e exploração até esse sub-objetivo.
- **Has sub-goal** - Esse estado executa o caminho gerado pelo controlador através de sua matriz de controle. Ao alcançar o último sub-goal o robô irá para o estado **Updating Occ-Grid** para atualizar a occupancy grid, gerar um novo c-space, e planejar uma nova exploração segura.

- **Going base** - Nesse estado, o robô irá calcular um caminho no c-space que o leve de volta para sua posição inicial no começo da sua interação com o ambiente. Ela simboliza o retorno para a base ou para o seu possível operador.
- **Aligning** - Após retornar para a base, esse estado é responsável por alinhar o robô com sua configuração q inicial e obtida no tempo t_0 .
- **Reactive** - Durante a sua locomoção, o estado **Reactive** é ativado caso o robô encontre uma situação perigosa. Uma situação perigosa é aquela que pode causar danos ao robô, ambiente, objetos, pessoas. Uma vez nesse estado, o robô tentará efetuar manobras para evitar acidentes. Após efetuadas, o robô volta para sua rotina no estado **Updating Occ-Grid**.

2.2 Experimental Configuration

Foram efetuados dois testes para avaliar a capacidade da proposta em navegar e explorar nos ambientes desconhecidos. O primeiro teste visa avaliar a proposta em um mapa simples. Já o segundo visa avaliar a mesma em um mapa um pouco maior e complexo. Além disso, também foi avaliado o impacto do tamanho da célula do occupancy grid e c-space para a atuação do controlador.

Para efetuar os testes foi utilizado o robô pioneer3dx equipado com um sensor laser Hokuyo URG 04LX simulado no CoppeliaSim. O sensor laser utiliza a API do CoppeliaSim para enviar os dados referentes a distância e ângulo de cada feixe. Dois mapas foram criados para validar o controlador e ambos são ilustrados pela Figura



Fig. 3. Mapas utilizados para efetuar os testes. A esquerda é ilustrado o mapa fácil. Já a direita, é ilustrado o mapa difícil.

2.3 Parâmetros

O tamanho máximo do ambiente de simulação foi fixado em 40 metros. A occupancy grid foi validada com tamanhos 320x320 e 640x640 para a configuração difícil. Os obstáculos são aumentados na geração do c-space em um fator de 5. Os mapas gerados e sua visualização possuem tamanho 640x640. Para atualização da occupancy grid no estado **Updating** são utilizadas 10 iterações. A probabilidade inicial de ocupação para cada célula da occupancy grid é igual a 50%. Caminhos impossíveis são removidos assim que encontrados durante a etapa de deliberação utilizando o c-space.

Para controlar o robô foi utilizado o controlador proposto por Desai. O comportamento de seguir sub-objetivos navega em velocidade constante entre objetivos igual a 0.2 metros por segundo. Ao estar próximo do último elemento,

a velocidade linear é configurada para um fator proporcional ao erro de posição. O tempo de atuação do robô no estado **Observe** é igual a 6 segundos. Ao perseguir uma fronteira, o robô executa apenas 90% do caminho gerado pelo A* para evitar colisões. Já o comportamento reativo é ativado ao se encontrar a menos de 1 metro de distância de obstáculos. Foi configurado um ângulo de recuperação de 30 graus a uma velocidade linear de 0.5 metros por segundo. Os sensores laser foram configurados com ruído aleatório entre

2.4 Configuração Fácil

Os experimentos efetuados no teste da configuração fácil demonstrou total eficiência do robô. Não houve comportamentos inesperados. Também não foram observados distúrbios no controlador reativo. O robô foi capaz de explorar o mapa simples, eliminando todas as fronteiras e também voltando a posição e pose inicial. A Figura 5 ilustra o caminho, c-space, e occupancy grid gerados para a configuração fácil. Os resultados referentes a esse teste podem ser visualizados em <https://youtu.be/QbH0F4hxNMM>.

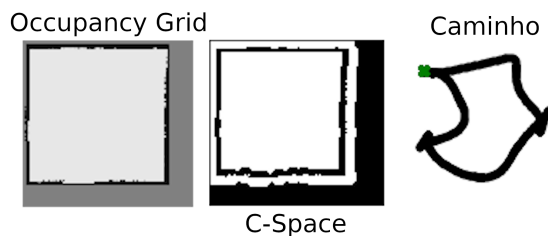


Fig. 4. Occupancy grid e c-space gerados e caminho percorrido pelo robô no mapa fácil.

2.5 Configuração Difícil

Durante os testes no mapa difícil foi observado que o robô foi capaz de navegar sem interferência do operador. Na Figura 5 são apresentados mapas gerados para a configuração difícil. A geração do mesmo pode ser visualizada através do vídeo em <https://youtu.be/wpQRa3EiAGg>.

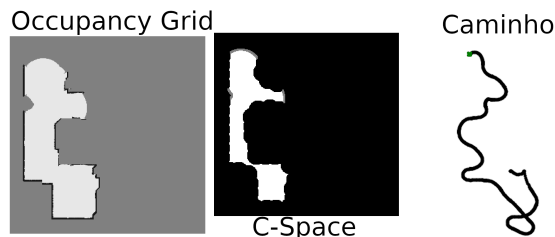


Fig. 5. Occupancy grid e c-space gerados e caminho percorrido pelo robô do mapa difícil.

Por consequência do comportamento de recuperação não ser deliberativo o robô mostrou dificuldade para se recuperar em lugares apertados. Em ambientes abertos e com espaço suficiente para navegação, não houve alterações

de comportamento e o mesmo foi capaz de navegar normalmente. Em alguns momentos, foi observado uma demora para deliberar sobre o caminho a ser percorrido. Apesar dos comportamentos observados, o robô não se comportou de forma condizente em ambientes apertados, o que pode indicar uma possível falta de ajuste de parâmetros para tunar o mesmo durante a construção e filtragem do c-space. Não foram observadas alterações para a construção da occupancy grid.

É importante notar que próximo ao fim da exploração para esse teste foi observado um comportamento inesperado do controlador do Dsai ao tentar se locomover na direção contrária a orientação atual do robô ouveram distúrbios na mesma. Porém, esse comportamento não afetou a construção do mapa e utilização dos representações pelo algoritmo.

2.6 Diferentes Tamanhos de Célula

Os diferentes tamanhos de células não influenciaram o comportamento geral do controlador. Foi observado que quando as células são maiores elas reduzem o tempo de exploração e podem auxiliar na construção eficiente do c-space. Além disso, também foi observado que a navegação tende a ficar mais suave com células maiores e mais precisa com células menores. Não foram observados problemas ligados aos ruídos adicionados na leitura do laser. A Figura 6 ilustra a diferença de resolução entre os tamanhos 320x320 (a esquerda) e 640x640 (a direita) em uma escala de 40 metros.

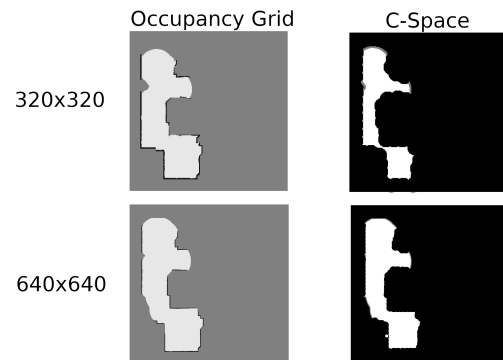


Fig. 6. Occupancy grid e c-space gerados para os tamanhos 320x320 e 640x640 em uma escala de 40 metros.

2.7 Conclusion

Nesse projeto é proposto, a implementação, implantação, de um controlador híbrido, deliberativo e reativo, para robôs autônomos utilizando occupancy grids para representar o ambiente incerto e desconhecido, representações do ambiente para deliberação e navegação através de exploração de fronteiras com auxílio de Roadmaps e c-spaces, e visa incorporar tolerância contra colisões indesejadas.

A proposta foi avaliada utilizando um robô pioneer3-dx no CoppeliaSim. Foram efetuados dois testes, um com mapa fácil e outro com mapa difícil. Para ambos os mapas o robô foi capaz de navegar, explorar, e mapear de forma autônoma. Como trabalhos futuros, pretende-se melhorar o

comportamento de recuperação reativo para evitar dificuldades em ambientes apertados. Pretende-se também melhorar os métodos para exploração e filtragem das fronteiras. Por fim, irão ser incorporados métodos para localização probabilísticos com fusão sensorial.

REFERENCES

- [1] R. Mendonça, M. M. Marques, F. Marques, A. Lourenço, E. Pinto, P. Santana, F. Coito, V. Lobo, and J. Barata, "A cooperative multi-robot team for the surveillance of shipwreck survivors at sea," in *OCEANS 2016 MTS/IEEE Monterey*, 2016, pp. 1–6.
- [2] F. Matsuno and S. Tadokoro, "Rescue robots and systems in japan," in *2004 IEEE International Conference on Robotics and Biomimetics*, 2004, pp. 12–20.
- [3] H. Yoshitake, R. Kamoshida, and Y. Nagashima, "New automated guided vehicle system using real-time holonic scheduling for warehouse picking," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1045–1052, 2019.
- [4] E. Bischoff, F. Meyer, J. Inga, and S. Hohmann, "Multi-robot task allocation and scheduling considering cooperative tasks and precedence constraints," 2020.
- [5] K. Petersen, R. Nagpal, and J. Werfel, "Termes: An autonomous robotic system for three-dimensional collective construction," 06 2011.