

Sistema Multi-Robôs para Cobertura Eficiente de Ambientes Domésticos

Bernardo Borges e Daniele Diniz

I. INTRODUÇÃO

A. Contextualização

Uma aplicação relevante na área de robótica é a dos robôs aspiradores, que têm a função de percorrer o ambiente doméstico de forma autônoma, recolhendo detritos ao longo do caminho. Este projeto abordará o problema de cobertura no contexto de sistemas multi-robôs, com o objetivo de dividir a tarefa entre dois robôs. Um dos principais desafios nesta área é garantir que os robôs executem seu trajeto de forma eficiente, evitando a movimentação aleatória pelo espaço, como observado em robôs aspiradores convencionais, conforme demonstrado no vídeo da Neato Robotics.

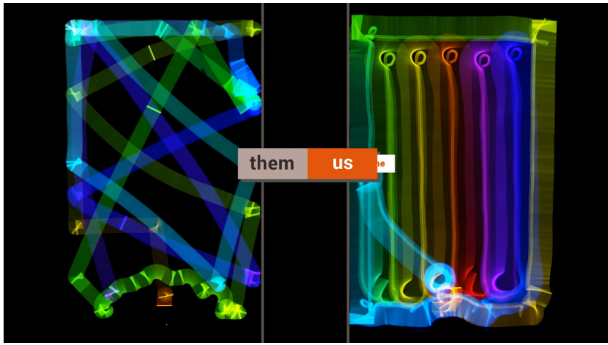


Fig. 1. Comparativo de robôs aspiradores sem e com planejamento de caminhos

Para superar esses desafios, utilizaremos nosso conhecimento em mapeamento, controle e algoritmos para desenvolver um sistema de multi-robôs que abordará o problema de cobertura do ambiente de forma eficaz, completa e, adicionalmente, enfrentaremos o desafio de implementar a colaboração entre os robôs. A implementação será realizada utilizando a linguagem Python e o simulador CoppeliaSim.

B. Motivação

Abordar este problema é crucial para aumentar a eficiência energética e reduzir a pegada de carbono dos robôs aspiradores domésticos. Um planejamento adequado permite que esses robôs executem suas tarefas de maneira mais rápida e eficiente, contribuindo significativamente para a sustentabilidade ambiental. As principais aplicações incluem a limpeza autônoma em residências, escritórios e outros espaços fechados onde a manutenção regular é necessária.

C. Descrição do Problema

O problema específico abordado é a ineficiência dos trajetos dos robôs aspiradores domésticos devido à ausência de um planejamento deliberativo de cobertura. Os desafios

incluem a criação de um mapa da residência, a discretização do ambiente em uma grade, e a implementação de algoritmos de cobertura que permitam a coordenação eficiente de múltiplos robôs. O objetivo é desenvolver uma solução que possibilite aos robôs realizar trajetos otimizados e colaborativos, minimizando o tempo e a energia consumidos para limpar o ambiente completamente. Em nossa implementação, consideraremos um ambiente com dois robôs que devem colaborar na limpeza, de forma paralela. Além disso, vamos trabalhar partindo de um mapa pré-existente do ambiente, que teria sido obtido em uma fase anterior, no caso de uma implementação real.

II. TRABALHOS RELACIONADOS

Implementações similares ao nosso projeto incluem os robôs da Neato Robotics. Esses robôs utilizam técnicas de mapeamento e navegação com base em visão computacional (bitvision) e empregam modelos treinados com florestas aleatórias para melhorar a eficiência da navegação e da cobertura do ambiente.

Diversos estudos abordam o problema de cobertura em sistemas multi-robôs. Um exemplo relevante é o artigo "Exact and Heuristic Multi-Robot Dubins Coverage Path Planning for Known Environments" publicado na revista Sensors. Este estudo apresenta métodos exatos e heurísticos para o planejamento de trajetórias de cobertura para múltiplos robôs em ambientes conhecidos. A pesquisa destaca a importância de criar trajetórias eficientes que minimizem o tempo e os recursos necessários para a cobertura completa do ambiente, focando em trajetórias do tipo Dubins, que são especialmente úteis para robôs com restrições de movimentação como os robôs aspiradores.

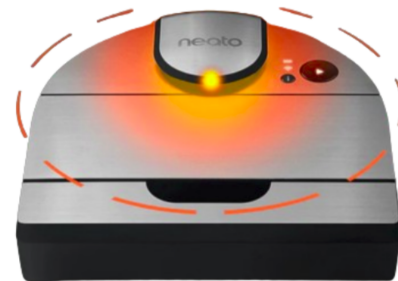


Fig. 2. Robô aspirador da empresa Neato Robotics

III. METODOLOGIA

Pensando em todos os problemas apresentados, trazemos uma solução que realizará um **mapeamento assertivo** para obter uma cobertura eficiente, realizando a delimitação de áreas de atuação dos robôs, o que evitará problemas de colisão entre eles.

A. Premissas

Para nosso projeto realizamos algumas suposições:

- O mapa do ambiente é conhecido;
- Os robôs conseguem se comunicar;
- O ambiente será **discretizado**, em células ocupadas(1) e desocupadas (0);

B. Separação de Tarefas

Considerando os múltiplos robôs, iremos atribuir regiões disjuntas para atuação. Para isso utilizaremos o algoritmo **Flood Fill**, que partirá de dois extremos do mapa e irá pintá-lo de *vermelho* e *azul* para as duas regiões.

```
def create_job_division():
    s0 = find_first_free_cell()
    sf = find_last_free_cell()
    queue = [s0, sf]
    while not queue.empty():
        (i, j, c) = queue.pop()
        if ColorGraph[i][j] != UNSET:
            continue
        ColorGraph[i][j] = c
        for (di, dj) in connect4Moves:
            if not valid_cell(di, dj):
                continue
            q.append((di, dj, c))
```

Após esse algoritmo, conseguimos as regiões para cada robô:

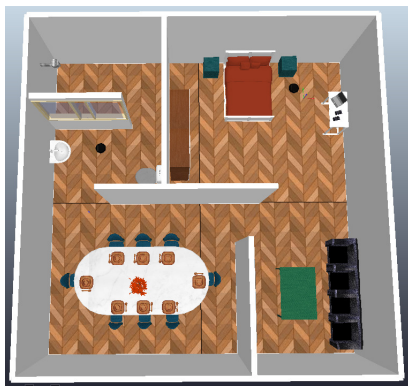


Fig. 3. Imagem aérea da cena "home"

C. Robôs e Controle

Para esse projeto usamos o robo "kobuki", que é direcional. O controle utilizado é "de Luca Oriolo" e utilizamos a locomoção inspirada em campos potênciasiais para criar uma força de atração do robô em relação aos objetivos que vamos atualizando iterativamente.

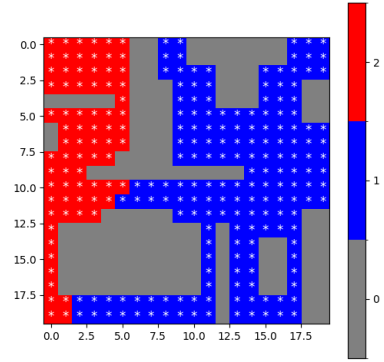


Fig. 4. Divisão das regiões em vermelho e azul

D. Navegação

Para a navegação em alto nível implementamos uma lógica que visita células mais próxima primeiro, ainda na lógica BFS:

```
for robot in robots:
    if robot.cells_left == 0: continue
    if robot.has_goal:
        robot.move_to_goal()
        continue
    [ri, rj] = robot.current_location
    for (dj, di) in moves:
        if robot.is_task_cell(ni, nj):
            robot.add_new_goal([ni, nj])
            break
    if robot.has_goal():
        robot.move_to_goal()
        continue
    # BFS to find nearest cell to clean
    path = robot.find_nearest_task_cell()
    if path == None: continue
    robot.add_new_goals(path)
    robot.move_to_goal()
```

Como podemos ver, primeiro tentamos limpar células na cruz de 4-conexão. Caso não haja uma célula próxima realizamos uma BFS no grid, que passa por células pela cor atribuída e seguimos esse caminho até o objetivo.

IV. RESULTADOS

Conseguimos realizar nossos testes com sucesso em duas cenas distintas, a primeira se chama "diningRoom" e possui uma grande mesa ao centro e bastantes células a serem cobertas. A segunda cena é "home", um apartamento com quartos, sala e banheiro, que exige organização e precisão para limpeza

- Em *DiningRoom* conseguimos coordenar os robôs para limpeza completa em 8 minutos e 21 segundos.
- Em *Home* os robôs terminaram a limpeza em 7:38.

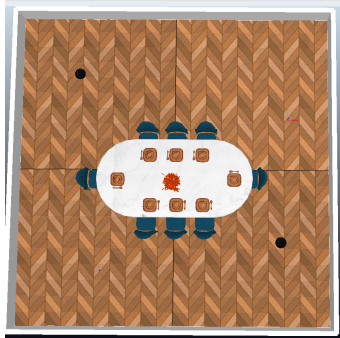


Fig. 5. Imagem aérea da cena "diningRoom"

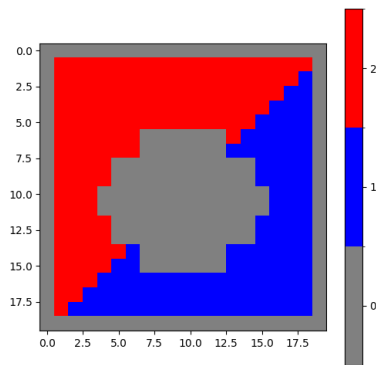


Fig. 6. Células limpas em *DiningRoom*

V. CONCLUSÃO

Com esse trabalho conseguimos entender os problemas e soluções aplicadas na coordenação de robôs, ainda que com apenas 2 elementos. Os desafios no ambiente de simulação foram de localização, atribuição de tarefas, controle e organização de código, já que há mais peças em movimento simultaneamente que deve ser pensadas.

Próximos passos possíveis para estender esse projeto serão:

- Transferência de tarefas, em um momento em que um robô terminou de limpar sua área, poderá redividir a tarefa para não permanecer ocioso;
- Atualização Automática do Mapa, para cenários mais dinâmicos, como a mudança de móveis;
- Suporte para Diversos Robôs, aumentando a frota dinamicamente dividindo a tarefa para mais robôs

REFERENCES

- [1] H. Liu, "Exact and Heuristic Multi-Robot Dubins Coverage Path Planning for Known Environments," *Sensors*, vol. 23, no. 5, p. 2560, Feb. 2023. [Online]. Available: <https://doi.org/10.3390/s23052560>.