

**SIMULAÇÃO DE ROBÔ MÓVEL DE DESCONTAMINAÇÃO DE AMBIENTE HOSPITALAR.**Matheus V. S. Ferreira<sup>1</sup>, Walter B. V. Filho<sup>2</sup>

1. Estudante da Faculdade de Tecnologia da Universidade de Brasília (FT-UnB)
2. Professor da Faculdade de Tecnologia da UnB - Departamento de Engenharia Mecânica/Orientador

**Resumo**

Após um estudo de métodos de descontaminação para identificar o mais adequado de ser automatizado na limpeza de ambientes hospitalares, foi escolhida a descontaminação por lâmpadas de radiação ultravioleta tipo C para ser aplicada no desenvolvimento de um robô de desinfecção guiado por um algoritmo de navegação autônoma. Buscando desenvolver um sistema de navegação robusto e eficiente, a metodologia focou na criação de um simulador que emulasse as condições reais de um ambiente, tornando os parâmetros do robô (potência das lâmpadas, tempo de exposição, etc.) como variáveis do sistema. O algoritmo desenvolvido foi inspirado no Algoritmo de Trémaux para resolução de labirintos com suporte da estrutura de dados de árvores para organização da memória. Os resultados demonstraram altas taxas de limpeza em nove mapas distintos que simulam ambientes reais, com média de 89,55% de áreas totalmente limpas.

**Palavras-chave:** UV-C; navegação; Python.

**Trabalho selecionado para a JNIC:** UnB.

**Introdução**

A pandemia da COVID-19 é o principal desafio enfrentado por cientistas do mundo nos últimos anos. A Organização Mundial da Saúde (OMS) recomenda o uso de máscaras e a higienização das mãos com água e sabão ou álcool 70%, no entanto a descontaminação de ambientes possui logística própria que deve ser adequada às condições do lugar a ser descontaminado.

Com isso, buscou-se selecionar um método para descontaminação de ambientes fechados, e como a maioria é nociva à pele humana, este projeto aplica conceitos de robótica móvel para a limpeza mais efetiva e segura de um quarto hospitalar. Foi feita uma revisão bibliográfica das técnicas mais difundidas de higienização de hospitais, seguindo recomendações feitas por órgãos como a OMS, o Centro de Controle e Prevenção de Doenças (CDC, em inglês) e a ANVISA.

Primeiramente foram estudados os produtos químicos, que constituem importante atuação na higienização de diversos ambientes, havendo produtos domésticos, mas também os especializados para hospitais, alguns dos quais inclusos em processos de esterilização. Dois exemplos são o hipoclorito de sódio, que na água sanitária lidera os compostos clorados utilizados para a desinfecção de superfícies (CQC, 2020), e o Peróxido de Hidrogênio Vaporizado. Como ambos são difíceis de manipular para exposição em área e têm uma composição tóxica, eles foram descartados para uso nesse projeto.

Entre as técnicas não-químicas estudadas destaca-se a desinfecção por radiação ultravioleta tipo C (UV-C). As lâmpadas UV-C têm extenso histórico de desinfecção no tratamento de água pela sua radiação possuir um ponto crítico notoriamente germicida no comprimento de onda de 253.7 nm (Juchem et al., 1998). Mesmo o método também ter contrapartidas no seu uso, não podendo entrar em contato prolongado com a pele humana, o projeto foi prosseguido com essa opção por ter ferramentas de controle mais simples e precisar de menos preparo prévio do ambiente.

O desafio é o desenvolvimento de um robô que possa ser empregado rotineiramente, assim capaz de detectar pessoas e desligar o mecanismo de limpeza se preciso, navegando de forma independente por ambientes. Com ênfase na segurança, o foco é desenvolver simulações computacionais que recriem essa dinâmica e possam detalhar e diversificar o projeto real.

**Metodologia**

Para desenvolver um projeto em torno das lâmpadas UV-C é preciso conhecer seus parâmetros. A potência da lâmpada  $P$  é inversamente proporcional ao tempo de exposição  $t$  necessário para um vírus ser inativado, tendo cada estirpe patogênica uma dosagem  $D$  indicando quanto tempo a uma certa intensidade de luz uma superfície deve ser submetida para eliminação de 99,99% dos microrganismos. Para a SARS-CoV-2,  $D = 16,5 \text{ mJ/cm}^2$  (Biasin et al., 2021). Logo, tem-se as seguintes relações:

$$D = \frac{P}{4\pi R^2} t \quad (1)$$

$$t = \frac{66\pi R^2}{P} \quad (2)$$

onde  $R$  é a distância de uma superfície para o eixo central da lâmpada (Arguelles, 2020).

Para verificar o funcionamento do algoritmo sem precisar de um modelo físico foi feita a simulação da navegação por meio da biblioteca Pygame, em Python, onde o mapa de uma sala foi representado na forma de grade/matriz binária em que seus elementos representam uma área ocupada (em preto) ou não (em branco). O robô representado na Figura 1 se desloca apenas se a região à sua volta possuir área livre, estas sendo detectadas escaneando o entorno da posição atual, gerando um vetor as distâncias para os objetos mais próximos (em vermelho) que pode ser transformado em uma matriz 6x6 com o entorno do robô simplificado, conforme a Figura 1(b) na posição (3,9).

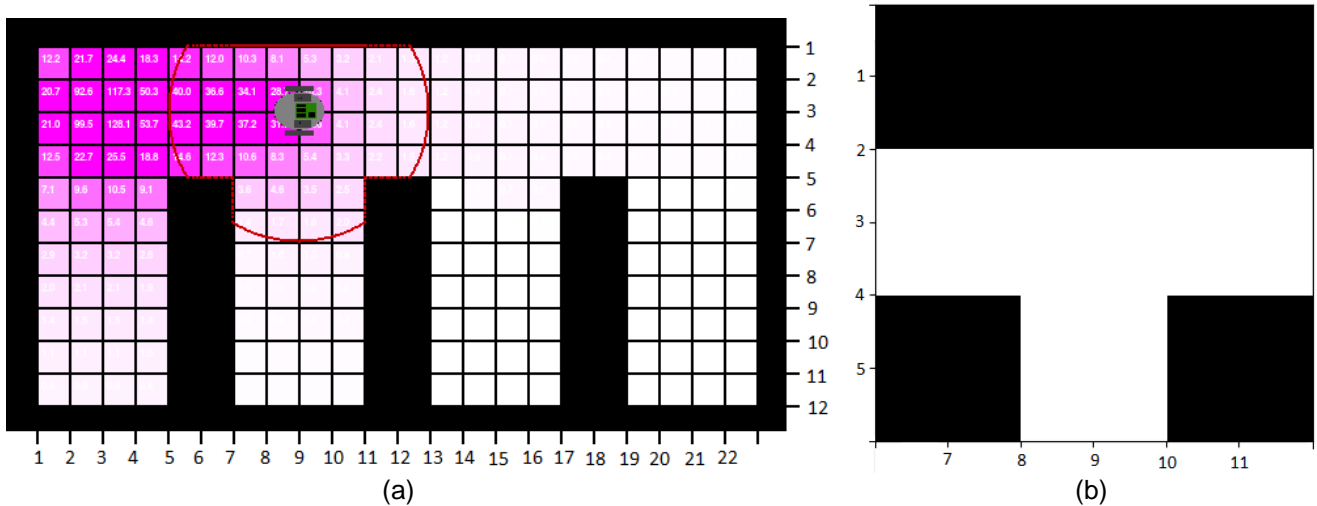


Figura 1 – (a) Simulação do robô em um mapa; (b) visualização da leitura dos sensores em (9, 3).

A navegação do robô é inspirada no algoritmo de Trémaux para solução de labirintos. Dados o mapa na forma de grade e a posição inicial, o algoritmo ordena o robô se mover na orientação atual até que não possa mais, ou seja, até que os sensores detectem um objeto à frente. Ao longo do caminho, cada ponto de mudança de direção e encruzilhadas (onde há mais de uma nova direção viável) é registrado para conhecer do mapa e poder explorá-lo mais futuramente. O algoritmo é encerrado quando não há mais espaços não-percorridos na memória (Gonzalez-Arjona et al., 2013).

Os registros de posição são armazenados na estrutura de árvore de dados, em que um novo registro passa a ser raiz dos seguintes e nó do anterior. A Figura 2 descreve a movimentação do robô ao longo do mapa, partindo da posição inicial em (3, 3) e seguindo a direita até (9, 3), onde é criado o primeiro nó de encruzilhada. O mesmo processo é repetido em (15,3), então em (21, 3) é registrado um nó de mudança de direção pelo robô ter que se movimentar para baixo. Ao chegar em (21,10), a única opção viável é retornar, o que fará o algoritmo olhar para sua árvore de dados e procurar o nó mais próximo com direções não-percorridas, no caso, em (15,3). Indo para (15,10), o algoritmo revisitará novamente a árvore e repetirá o processo acima até que de volta a posição inicial não haverá mais caminhos desconhecidos.

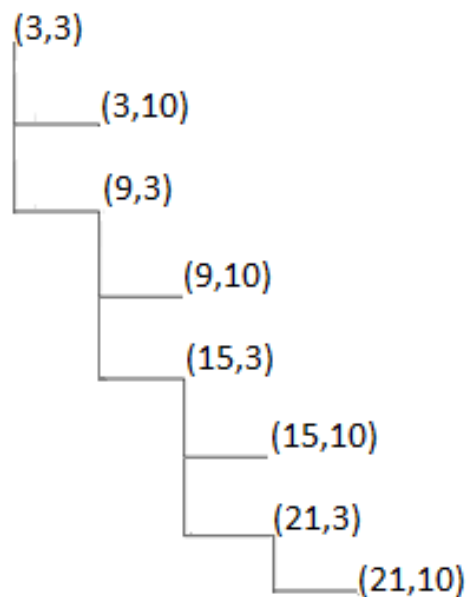


Figura 2 – Árvore de dados de posição.

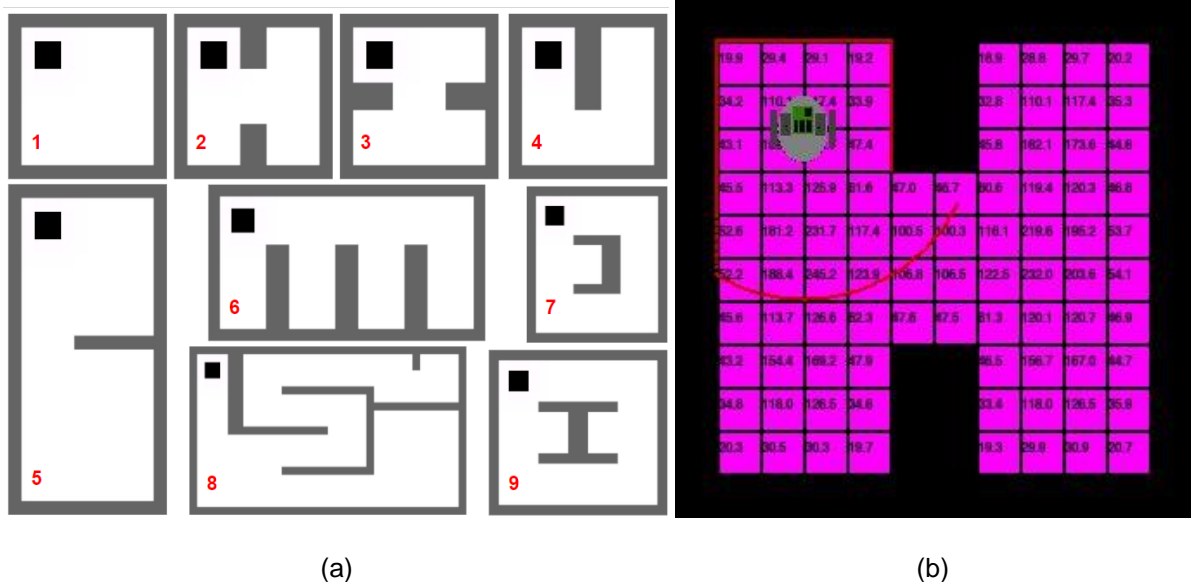


Figura 3. (a) Mapas da simulação computacional; (b) mapa 2 após navegação.

## Resultados e Discussão

Em todos os mapas da Figura 3(a) foram feitas simulações onde, ao navegar, o robô emite radiação UV-C na área do entorno, assim desenvolvendo uma forma de mensurar a eficiência da varredura. Na Figura 3(b), dosagem nos blocos de área é calculada conforme a Equação (1), considerando a potência total das lâmpadas de  $P = 60W$  (com possível atenuação de até 90% sofrida pelas mesmas) e a distância de cada bloco para o centro do robô. Os nós são a posição onde o robô espera mais prolongadamente a emissão, permanecendo um tempo de exposição de 60 segundos, tendo em vista que se há um objeto entre o ponto emissor e um bloco, a dosagem emitida naquele instante não é contabilizada para ele. No final, portanto, pelos os dados obtidos foram calculados a dosagem média, a porcentagem de áreas totalmente limpas (que receberam dosagem acima de  $D = 16,5 \text{ mJ/cm}^2$ ) e o tempo estimado na simulação (Tabela 1).

Tabela 1. Eficiência de limpeza por mapa.

Mapas	1	2	3	4	5	6	7	8	9
Área livre ( $m^2$ )	6,25	5,50	5,50	5,50	13,38	12,50	10,25	35,56	12,50
Dosagem média ( $mJ/cm^2$ )	64,87	50,58	50,69	34,13	71,65	45,31	46,49	65,66	44,36
Área totalmente limpa (%)	100	90,9	90,9	70,5	100	85,0	87,8	93,13	88,5
Tempo total (minutos)	8:36	6:36	6:36	4:36	19:30	13:32	10:50	48:44	12:36

## Conclusões

O robô será montado visando testar seu modelo de navegação e sua precisão de leitura. Pela simulação, verificou-se que o sistema computacional criado é, em pelo menos nove mapas distintos, capaz de: definir pontos por onde já passou, procurar novas rotas na sua memória, além de encerrar a operação quando não as encontrar após limpeza completa. Considera-se o resultado ótimo em quase todos os mapas, sempre chegando próximo ou ultrapassando 90% de limpeza, com exceção do mapa 4. Isso se deve a tal possuir poucos nós de acordo com as condições impostas, fazendo o robô parar poucas vezes para emitir radiação mais prolongadamente.

Pelos resultados obtidos no que foi simulado, o projeto pode ser montado fisicamente com eficácia e testado em ambientes reais onde se utilizará marcadores de luz UV-C que indicam a dosagem emitida em um certo local, essa devendo ser superior à dosagem necessária para eliminar uma porcentagem alta de vírus SARS-CoV-2 e outros microrganismos.

## Referências bibliográficas

Conselho Federal de Química - CFQ (2020). Perguntas e Respostas: Água Sanitária. Disponível em <[020-05-04\\_cartilha-perguntas-e-respostas-CFQ-V2-baixa-3.pdf](https://www.cfq.org.br/020-05-04_cartilha-perguntas-e-respostas-CFQ-V2-baixa-3.pdf)>.

Juchem, P.P., Hochberg, J., Winogron, A., Ardenghy, M., English, R. (1998). Riscos à Saúde da Radiação

Ultravioleta. Rev. Bras. Cir. Plást.1998;13(2):31-60

Biasin, M., Bianco, A., Pareschi, G., Cavalleri, A., Cavatorta, C., Fenizia, C., Galli, P., Lessio, L., Lualdi, M., Tombetti, E., Ambrosi, A., Redaelli, E., Saulle, I., Trabattoni, D., Zanutta, A., & Clerici, M. (2021). UV-C irradiation is highly effective in inactivating SARS-CoV-2 replication. *Scientific reports*, 11(1), 6260. <https://doi.org/10.1038/s41598-021-85425-w>

Arguelles, P. (2020). Estimating UV-C Sterilization Dosage for COVID-19 Pandemic Mitigation Efforts. <https://doi.org/10.13140/RG.2.2.12837.65761>.

Gonzalez-Arjona, D. Sanchez, A., López-Colino, F., De Castro, A., Garrido, J. (2013). Simplified Occupancy Grid Indoor Mapping Optimized for Low-Cost Robots. *ISPRS Int. J. Geo-Inf.* 2013, 2, 959-977. <https://doi.org/10.3390/ijgi2040959>