

PROJETO DE UM ROBÔ MÓVEL PARA DESCONTAMINAÇÃO HOSPITALAR

Matheus Virgílio da Silva Ferreira, matheusvirgiliomcp@gmail.com¹
Walter de Britto Vidal Filho, wbritto@unb.br¹

¹Departamento de Engenharia Mecânica, Universidade de Brasília, Campus Universitário Darcy Ribeiro, Brasília-DF

Resumo: Ambientes hospitalares concentram grande número de contaminações microbiológicas. Após estudar diferentes métodos de descontaminação para identificar o mais adequado a ser automatizado na limpeza de ambientes hospitalares, selecionou-se um para desenvolvimento de um robô de desinfecção guiado por algoritmo de navegação autônoma. Verificada uma série de métodos, escolheu-se a descontaminação por lâmpadas de radiação ultravioleta tipo C e, a partir disso, foram feitas simulações computacionais para medir sua eficiência com um sistema de navegação inspirado no algoritmo de Trémaux para resolução de labirintos. A simulação tem como variáveis os parâmetros do projeto do robô (dimensões, velocidade, potência das lâmpadas e tempo de emissão de radiação UV), estipulados com a metodologia assumida de definição de problema e decomposição funcional. No que se refere a sua estrutura, foram projetados os sistemas mecânico e eletroeletrônico visando portabilidade e baixo custo, seguindo o estipulado na tabela de requisitos. Os resultados demonstraram altas taxas de limpeza em nove mapas distintos que simulam ambientes reais, com percentual de área totalmente limpas acima 95% em todos os ambientes testados. Foi também testada uma aplicação de scan-matching para correção da trajetória odométrica, a qual forneceu estimativas com erro relativo médio inferior a 5% para pequenas variações, sendo utilizado poucos recursos computacionais.

Palavras-chave: UV-C, simulação, PSO, navegação

1. INTRODUÇÃO

A pandemia da COVID-19 foi o maior desafio enfrentado por cientistas mundiais nos últimos anos, e ainda que os principais esforços tenham sido para a produção de vacinas, de medicamentos e de tecnologias para as Unidades de Terapia Intensiva, o combate à pandemia também foi feito por métodos de higiene e de prevenção, como as máscaras e o álcool 70%. Com o surgimento de novas variantes da SARS-CoV-2, além de outros patógenos emergentes que ameaçaram a saúde pública, a evolução da logística de higienização dos ambientes hospitalares se mostrou uma tarefa importante. Nos primeiros dois meses de pandemia no Brasil, em 2020, mais de 100 médicos, além de pelo menos 143 enfermeiros, perderam a vida contaminados em ofício (ANAMT, 2020), chamando atenção para o desconhecimento dos riscos dos vetores de contaminação, a utilização inadequada de EPI's e falhas na descontaminação dos ambientes, as quais contribuíram para as mortes desses profissionais de saúde.

Os principais meios de contaminação seriam pela inalação do patógeno no ar e pelo levar de mãos contaminadas aos olhos, boca e nariz. A descontaminação do ambiente hospitalar era feita por funcionários de limpeza, contudo, não se mostrou suficiente, considerando que as partículas mantinham-se por determinado período no ar, e muitos desses ambientes não contavam com ventilação natural ou sistema de exaustão ou filtração de ar. Ambientes hospitalares projetados com sistemas de ar-condicionado tem a contaminação do ar como um problema crítico, para o qual vem sendo adotado o uso de filtros especiais no sistema de condicionamento de ar, o que tem mitigado a situação. Com o intuito de descontaminar os ambientes hospitalares de forma mais efetiva e reduzir o risco, foram criados sistemas robotizados de descontaminação (Fernandes, 2021; Guettari, 2021). No presente trabalho, aborda-se o projeto e desenvolvimento de um robô para descontaminação hospitalar. A principal motivação é o desenvolvimento de tecnologia nacional a um custo acessível para auxiliar o combate à pandemia.

2. REVISÃO TEÓRICA

2.1. Estudos dos Métodos de Desinfecção.

Diversos produtos químicos são utilizados na higienização de ambientes hospitalares, havendo produtos especializados para este setor. Um exemplo disso é a Água Sanitária, altamente utilizada na desinfecção de superfícies,

cujas maior parte da composição é hipoclorito de sódio (2,0-2,5% de NaClO), o qual, após diluição em água, forma o ácido hipocloroso, que é o responsável pela ação germicida (CFQ, 2020).

Enquanto o hipoclorito tem aplicação limitada devido a sua toxicidade, o ácido hipocloroso é um importante agente no combate de patógenos pela sua segurança e baixo custo. Capaz de oxidar matéria orgânica realizando cloração nos vírus, essa substância pode ser aplicada na purificação de ambientes (Block & Rowan, 2020). Há muitos produtos para desinfecção de ambientes no mercado, sendo importante diferenciar a efetividade desses em relação aos métodos mais profissionais aplicados dentro de hospitais, como processos de esterilização. Um objeto é considerado estéril quando a probabilidade de sobrevivência dos microrganismos contaminantes nele é menor do que 1:1000.000, implicando na eliminação de esporos bacterianos e impossibilitando a detecção da cultura do microrganismo na amostra posteriormente (Secretaria de Saúde do Estado do Espírito Santo, 2009).

Alguns produtos de esterilização foram estudados no início da pandemia para descontaminação e reuso de máscaras N-95, incluindo o Peróxido de Hidrogênio Vaporizado (VHP, em inglês) e o Óxido de Etileno (Fischer et al., 2020), dos quais o VHP se destaca pela aplicação em ambientes com alto risco infeccioso. É produzido pela vaporização ativa de uma solução aquosa de peróxido de hidrogênio (30-35% de H₂O₂) em placa quente e, por se tratar de uma substância instável e bastante absorvente, sua concentração deve ser constantemente monitorada por sensores também responsáveis por captar vazamentos. Essa etapa é essencial por conta de o VHP ser extremamente nocivo a seres humanos, sendo necessário que uma sala pós-higienizada receba ventilação até a concentração de H₂O₂ ser reduzida para uma faixa segura.

A inativação de microrganismos por calor é uma forma de descontaminação. Sabe-se que microrganismos são sensíveis a uma exposição contínua e prolongada ao calor, podendo levar a sua inativação. Estudos de desinfecção de máscaras N95 por calor mostraram que para uma redução de micróbios na base 5 log (99,999%) são necessários até trinta minutos de exposição da amostra a 56°C (Liao et al., 2020). Esses valores variam por estirpe do vírus e bactérias, sendo que a mesma redução em alguns tipos de SARS-CoV-2 é feita em menos tempo (Abraham et al., 2020). Contudo o uso de vapores ou de ar aquecido para descontaminar grandes superfícies necessita considerável gasto de energia.

Outro método de inativação de vírus muito conhecido é o uso da radiação por luz ultravioleta (UV) (Juchem et al., 1998). A luz UV faz parte do espectro eletromagnético, estando entre a luz visível e os raios-X (no comprimento de onda entre 400 e 200 nm), sendo as UV-A e UV-B conhecidas por efeitos danosos causados pelo sol à pele humana. Por sua vez, a UV-C está na menor faixa (280-200 nm) e possui em si um intervalo notoriamente germicida entre 285 e 260 nm (Juchem et al., 1998), sendo seu ponto de melhor ação o comprimento de onda de 253.7 nm (Fischer et al., 2020).

Definida a melhor frequência para atuação, as lâmpadas UV-C variam entre si pelas suas potências. A potência da lâmpada P é inversamente proporcional ao tempo de exposição t necessário para um vírus ser inativado. Cada estirpe patogênica possui uma dosagem D indicando quanto tempo, a uma certa intensidade de luz, o microrganismo deve ser submetido para sua inativação. A SARS-CoV-2 possui $D = 16,5 \text{ mJ/cm}^2$ (Biasin et al., 2021), representando a energia necessária para a eliminação de 99,99% dos vírus em uma superfície de 1 cm^2 . Essa relação pode ser vista na equação a seguir:

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

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

A UV-C também requer algumas precauções quanto ao seu uso. Uma exposição humana prolongada pode causar danos oculares na forma de irritação conjuntival, enquanto efeitos mais graves incluem fotoceratite e conjuntivite (Juchem et al., 1998). Devido as vantagens desta tecnologia de desinfecção, ela foi uma das primeiras utilizadas no combate à pandemia em 2020, sendo usada sobre um veículo de movimentação manual, pela empresa ARCI (Sarada et al., 2020). Em seguida esta tecnologia começou a ser automatizada através de robôs móveis (Guettari et al., 2021, Fernandes, 2021).

3. METODOLOGIA DE PROJETO

A atividade de projetar um robô está no campo dos sistemas mecatrônicos, envolvendo conhecimentos de áreas distintas da engenharia, como sistemas mecânicos, eletroeletrônicos e computacionais. A metodologia tradicional costuma dividir o projeto em subprojetos, como os sistemas propostos, tratando cada um como um sistema estanque. Segundo Pahl et al. (2005), identificam-se as seguintes fases: abstração, análise e síntese. Na abstração, busca-se o foco do problema, descartando fatores que não afetam o mesmo, enquanto, na análise, estudam-se informações para obtenção da solução de cada problema, as quais devem ser concluídas na etapa de síntese. Alguns autores (Madureira, 2015; Back et al, 2008) costumam também dividir um projeto em projeto conceitual e projeto detalhado, ou ainda, em estudo de viabilidade, em pré-projeto e projeto detalhado. Resumindo, todo projeto tem uma fase inicial de determinação do problema, seguida pelas fases de desenvolvimento do conceito, detalhamento e, enfim, validação. Seguindo essa estrutura, então, monta-se um organograma dos subprojetos, como visto na Figura 1, e é predefinido que o início do projeto será pelo sistema mecânico.

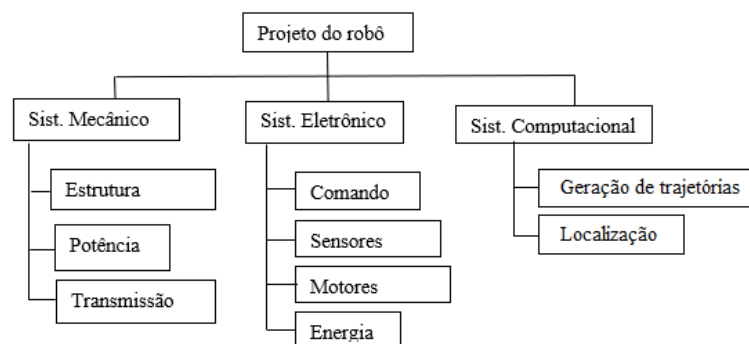


Figura 1. Organograma do projeto.

A partir da conclusão do projeto do primeiro sistema, obtém-se parâmetros para especificação do sistema eletroeletrônico, como a potência mecânica necessária, o torque, e a velocidade, pelos quais se calcula a potência elétrica e permite especificar os motores a serem escolhidos. A escolha dos motores também impacta nas placas de controle utilizadas, uma vez que devem suportar a máxima corrente exigida pelos mesmos. Nota-se que nem toda parte eletroeletrônica é codependente da mecânica, pois também é composta por sensores utilizados na odometria e localização do robô. Esses variam pela especificação necessária para tarefa, o que se traduz na precisão da trajetória. Desta forma, existem as especificações gerais do projeto e, além delas, há também especificidades de cada subprojeto que podem depender das interações entre os subsistemas. No caso do sistema computacional, que contém o software de comando, a escolha da linguagem não é diretamente ligada aos dispositivos eletrônicos, contudo os drivers e a UCP devem se comunicar de forma a garantir o melhor funcionamento da navegação sugerida no código.

3.1. Definição do Problema de Projeto

Partindo de uma necessidade inicial motivadora e seguindo o método, busca-se definir o problema a ser resolvido no projeto. Um primeiro passo é analisar as necessidades existentes e converter em parâmetros e requisitos. A sistemática para determinação começa na revisão de literatura, ou revisão do estado da arte. Neste último, pode-se fazer uma análise de mercado para buscar a existência de produtos similares e avaliar suas características. Várias empresas surgiram após a crise pandêmica da covid19 para produzir robôs com esta finalidade de descontaminação. Analisou-se cerca de dez robôs disponíveis no mercado, chegando-se à Tabela 1.

Tabela 1. Requisitos

	Variação observada	Valor adotado
Peso	43 – 140kg	$M = 25kg$
Comprimento	36-66 cm	50 cm
Largura	50-93 cm	50 cm
Altura	135-171 cm	150 cm
Velocidade máxima	0,45-1,5 m/s	$V = 0,5 m/s$
Inclinação máxima	$\theta = 5^\circ$	
Tempo de operação	3 - 8 horas	4 horas

A Tabela 1 mostra a variação observada e os valores de requisitos adotados como parâmetros de projeto. O principal método de desinfecção dos robôs vistos empregava luz UV-C, mas alguns utilizavam, de forma auxiliar, pulverização de ativos químicos.

3.2. Projeto Conceitual

Nesta fase se estabelece uma solução conceitual para cada subsistema norteado pelos requisitos estabelecidos e através de técnicas como a decomposição funcional. O conceito para desenvolver o sistema mecânico começa pela escolha do método de descontaminação. Aqui poderiam ser empregados reservatórios com líquidos desinfetantes, entretanto esses atuam apenas em superfícies, como piso e paredes, desconsiderando o ar como vetor de contaminação importante. O

uso da radiação UVC, por sua vez, consegue atingir partículas em suspensão no ar, necessita de menos preparo prévio do ambiente e é o principal meio observado na análise mercadológica, por isso foi o método selecionado.

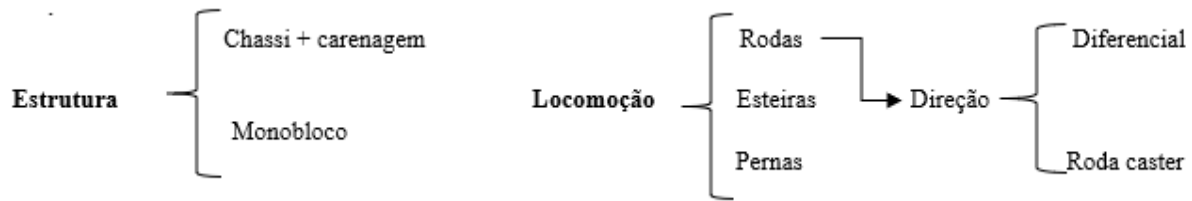


Figura 2. **Decomposição funcional**

O conceito de solução é baseado em um robô com estrutura com chassi e locomoção por rodas. Escolheu-se o controle de direção diferencial, ou seja, uma roda tratora de cada lado que pode ser acionada independentemente, permitindo um giro sobre o eixo de simetria. Isso favorece manobras em ambientes com pouco espaço livre. Uma pequena roda livre na frente e outra atrás favorece a estabilidade. O espaço interno do robô é amplo para facilitar a manutenção e fixação das placas eletrônicas de comando e potência.

3.3. Projeto do Sistema Mecânico

O projeto mecânico corresponde aos cálculos dos esforços estáticos e dinâmicos, desenhos detalhados da estrutura e dos elementos de fixação e especificação do acoplamento, dos materiais e do motoredutor. Em geral, a estrutura de robôs de limpeza UV-C pode ser dividida em duas: base móvel e conjunto de lâmpadas. A base móvel deve ter um formato adequado para poder atravessar passagens mais estreitas e ter altura suficiente para as lâmpadas chegarem acima de superfícies como camas e mesas e emitir radiação sobre elas. Espera-se também estabilidade, devendo resistir a pequenas colisões sem tombar. Na figura 3 observa-se o desenho resultante do projeto.

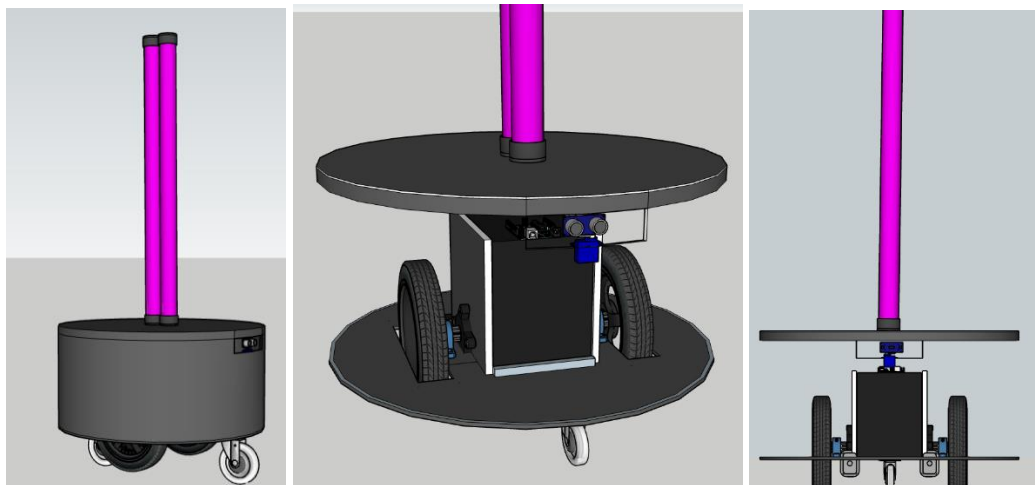


Figura 3. **Estrutura do corpo**

Para especificar os motores, deve-se calcular o torque necessário para acionamento do robô. Partindo da inclinação máxima que deseja vencer em uma rampa (fig.4), monta-se a eq. (2), onde T é o torque total, N corresponde à quantidade de motores atuando, V_e é a tensão elétrica e $e\%$ é a eficiência percentil. Por fim, chega-se ao torque necessário de cada motor (T_{motor}). Por sua vez, ω independe do diagrama de forças e é relação apenas da velocidade V e do raio das rodas r . Outras propriedades importantes de se obter são potência P , corrente máxima I e carga total c , que limitam o número de opções de escolha do motor uma vez que a sobrecarga dele pode prejudicar a fonte de tensão e demais componentes conectados. Todas essas relações estão demonstradas na figura e equações a seguir:

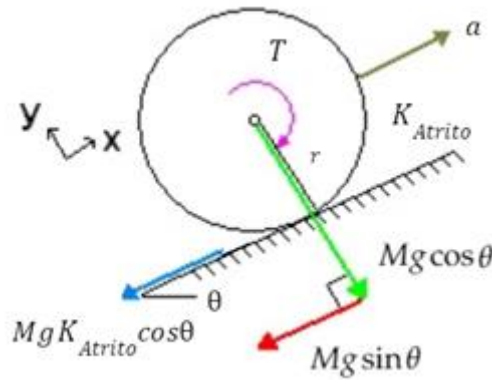


Figura 4. Cálculo do torque

$$Ma = \frac{T}{r} - Mg \sin \theta \quad (2)$$

$$T_{motor} = \frac{T}{N} = \frac{Mr(a + g \sin \theta)}{N} \quad (3)$$

$$\omega = \frac{v}{r} \quad (4)$$

$$P = \frac{T_{motor} \omega}{e} \quad (5)$$

$$I = \frac{P}{V_e} \quad (6)$$

$$c = I * (\text{tempo de operação}) \quad (7)$$

Definidos a massa M , a velocidade V e a inclinação máxima θ (Tabela 1), levando em conta ainda uma roda de raio $R = 10 \text{ cm}$, calculado pelas equações (3), (4), (5) e (6), obtém-se um torque 21,48 kgf.cm. Usando o catálogo de fabricante, chega-se ao motor JC/LC-578VA de 12V. Observa-se que o motor tem, na máxima potência, torque máximo de 55 kgf.cm, mais a velocidade angular cai com aumento do torque. Usando o torque calculado no gráfico do datasheet do motor, chega-se aos parâmetros da Tabela 2, onde eles podem ser comparados com os requisitos. A corrente de *stall* é de 24A, sendo relevante na escolha da ponte H que controla os sentidos dos motores.

Tabela 2. Propriedades do motor: requisitos x JC/LC-578VA.

	Torque máximo (kgf.cm)	Velocidade angular máxima (RPM)	Potência total (W)	Corrente máxima (A)
Requisitos	17,27	47,75	32,26	2,69
JC/LC-578VA	93,00	78,45 ^(*)	53,01 ^(*)	4,42

*Calculadas pelas informações do datasheet dado o torque de **17,27 kgf.cm** e eficiência de **26,25%**.

3.4. Projeto do Sistema Eletroeletrônico

O sistema eletroeletrônico compreende o circuito de potência, que controla e alimenta as lâmpadas UV-C e os motores, e a placa eletrônica digital de comando, representada pelo microcontrolador ou microprocessador. Escolheu-se separar os sistemas de acionamento e comando por optoacopladores para facilitar a manutenção futura e proteger o circuito de baixa potência. Os requisitos para o projeto de alta potência começam pela carga máxima dos motores, o que, pela equação (7), é a aplicação máxima de corrente ao longo do tempo de operação (Tabela 1), resultando numa medida de aproximadamente 22Ah. Observando modelos comerciais, é comum ver baterias de 12V com até 60Ah, isso faz com que o restante do sistema possa usar no máximo 38Ah. Esse cálculo é importante para a escolha das lâmpadas UV-C que, por si, também consomem bastante energia. Opta-se então por utilizar duas lâmpadas 12V UV-C de 30W, cuja potência total e eficiência de 10% (Arguelles, 2020) consegue, segundo a equação (1), desinfetar o raio de um metro em cerca de seis minutos. O consumo delas ao longo do tempo de operação seria 20Ah, dentro do limite imposto pelos motores.

O restante do subprojeto eletrônico é a especificação de dispositivos que podem ser adquiridos, não compensando fabricar placas de potência robustas. Nesse caso, foi selecionado um módulo de potência duplo BTS7960, capaz de suportar 43A em cada saída. Os sensores são especificados baseados nos requisitos das variáveis que se deseja controlar. Observou-se a necessidade de encoders para odometria e sensores de ultrassom e laser (mais especificamente, o LiDAR) para identificar obstáculos e suas distâncias.

3.5. Projeto do Sistema Computacional

3.5.1 Navegação do Robô

A navegação do robô é inspirada no algoritmo de Trémaux para solução de labirintos. Nele, dados o mapa na forma de grade e a posição inicial, o algoritmo ordena o robô a manter a orientação atual até que não possa mais seguir por ela, 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 entendimento do mapa e poder explorar futuramente. O algoritmo é encerrado quando não há mais espaços não-percorridos na memória (Gonzalez-Arjona et al., 2013). Trata-se de um algoritmo muito funcional para ambientes estreitos, onde as direções a serem tomadas por um robô são limitadas.

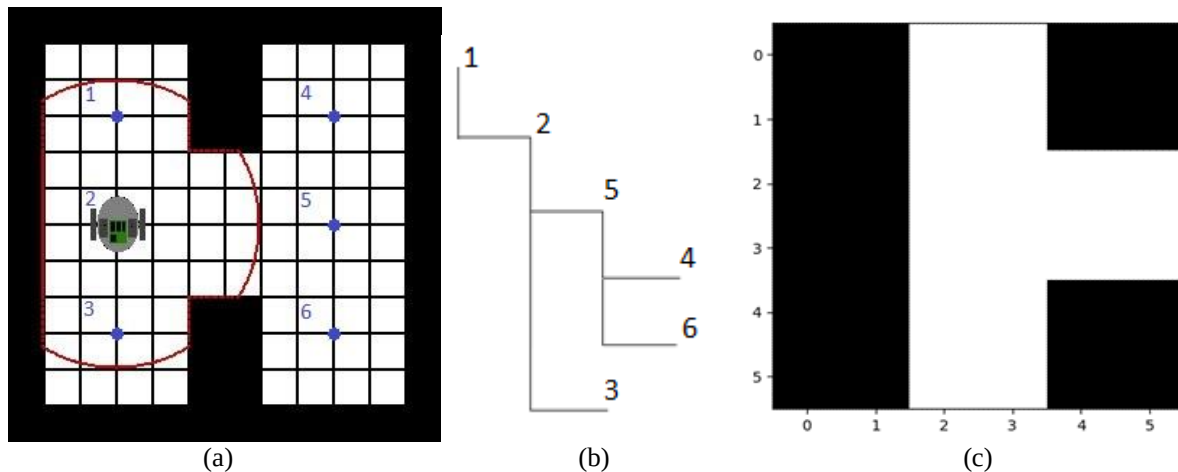


Figura 5. Simulação de navegação com árvore de dados de posição: (a) robô interagindo com o mapa; (b) árvore de dados gerada; (c) mapeamento feito a partir da posição do robô.

Para funcionar neste projeto, o algoritmo foi adaptado para registrar apenas posições que, além de possuir mais de uma direção nova para seguir (sem contar o sentido de onde veio), tenham direções diferentes do último registro. Todas as posições fins-de-trajetória, ou aqui denominados *dead-ends*, onde a única viabilidade é retornar, também são registrados. Sintetizando, os pontos a serem registrados passam por uma das seguintes condições:

- 1ª condição: mudança de direção, ou seja, quando na iteração anterior se encontrava seguindo norte e precisa virar para oeste, por exemplo;
- Caso contrário, 2ª condição: ser uma encruzilhada de direções novas e diferentes da interação anterior, onde a partir da posição atual haja a possibilidade de seguir dois ou mais caminhos distintos (sem contar com direções que darão em posições repetidas);
- Caso contrário, 3ª condição: ser um *dead-end*, geralmente fins de corredores ou posições a partir da qual seguir acabará em uma posição já percorrida.
- Caso contrário, 4ª condição: a cada um metro deslocado, caso não tenham sido registradas posições anteriormente, registra-se a posição atual.

Um ponto importante do algoritmo é que, nos fins-de-trajetória, o robô saiba para onde ir em seguida. Considerando a Figura 4(a), a navegação começa na posição 1 (que seria registrada por ser a posição inicial) e vai a 2 (registrada pela 1ª condição) e posteriormente até 3, um *dead-end* (3ª condição). A partir disso, espera-se que o robô retorne para 2 para então explorar na direção da posição 5. Esse retorno é possível pois os registros são feitos na forma de árvore de dados, com cada novo registro associado ao anterior como se fosse nó de uma raiz de dados. Ou seja, as posições 3 e 5 são nós de 2, pela trajetória que chegou a eles ter partido dela inicialmente, enquanto 2 é nó de 1. A chegada em um *dead-end* implica em voltar na árvore de dados procurando posições inexploradas. Continuando a navegação, 5 é registrado similarmente a 2, indo posteriormente a 4 e 6 (ambos *dead-ends*), que passam a ser seus nós. O fim do código ocorre quando em 6 a árvore é analisada, como disposta na Figura 4(b), verificando de volta em cada nó que não há novos caminhos a explorar.

A estrutura de dados de árvore - implementada em Python pela biblioteca Treelib - é útil para o registro de posições, pois ela sistematiza como os dados são guardados e permite traçar trajetórias de volta e verificar posições repetidas, impedindo que o robô se perca, além de determinar os melhores pontos com maior vastidão de área em contato para emissão de radiação UV-C. Cada nó contém os seguintes dados em si:

- A posição relativa ao ponto de origem (o ponto onde a navegação foi iniciada).

- As direções - direita, esquerda, cima e/ou baixo - para onde é possível se mover a partir da sua posição atual. São definidas após uma varredura feita pelos sensores de distância. As direções já percorridas são registradas de forma diferente das inéditas.
- A trajetória, isso é, a ordem de movimentos realizados para chegar de um nó até outro. Por isso, limita-se a movimentação vertical e horizontal por distâncias fixas, sendo assim mais fácil visualizar as rotas. Para possibilitar que o robô ultrapasse passagens angulosas que necessitam de movimento diagonal, foi estabelecido que o deslocamento fosse fixo e pequeno, simplificando o sensoriamento desses caminhos.
- O intervalo de tempo nas posições, uma vez que os nós serão os pontos onde ele parará para emitir radiação de forma mais prolongada. Com esse valor de tempo, pode-se calcular a dosagem total de UV-C emitida.

3.5.1. Simulação do Sistema Computacional

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. A grade é uma matriz binária em que seus elementos representam uma área ocupada ("1", em preto) ou não ("0", em branco). Foram desenhados nove mapas com diferentes dimensões, mas, em todos, a menor fração de área tem 25 cm x 25 cm. O robô, representado com dimensões 50 cm x 50 cm na Figura 4(a), se desloca 25 cm de cada vez em uma direção, mas apenas se a região à sua volta possuir área livre. As áreas livres são detectadas escaneando o raio de 1 metro da posição atual, detectando o objeto mais próximo para cada ângulo inteiro em 360 graus (desenhado em vermelho), gerando assim um vetor com 360 distâncias. Esse vetor pode ser transformado em uma matriz 6x6 com o entorno do robô simplificado, conforme a Figura 4(c), que mostra caminhos possíveis acima, abaixo e à direita a partir do centro da imagem. Ao se movimentar, a simulação do robô preenche sua árvore de posições.

3.5.2. Aplicação de Scan-matching

Como a simulação a princípio não considera possíveis erros sistemáticos e aleatórios nas distâncias percorridas, isto pode causar inconsistência entre os dados de posição registrados na memória e a posição real, uma vez desenvolvido o protótipo. Pelo princípio do algoritmo de Trémaux, é sempre necessário retornar a espaços já percorridos para explorar novos, o que pode ser problemático na implementação. Mesmo retornando para o que, na sua memória, é o local do seu nó, devido ao acúmulo de erro ao longo da trajetória, o robô pode estar deslocado em relação à primeira passagem. Seja a *pose* na primeira medição (x, y, θ) , espera-se que no retorno seja visto algo como $(x + \Delta x, y + \Delta y, \theta + \Delta \theta)$, defasado em módulo e orientação. Um novo deslocamento partindo dessa *pose* errada cria a possibilidade de ampliar o erro para os nós seguintes e fazer com que ocorra uma colisão.

Para fazer a correção, então, buscou-se na simulação formas de medir o erro a partir das leituras feitas pelos sensores, técnica conhecida como *scan-matching*, ou comparação de escaneamentos. O método de execução adotado foi a aplicação do algoritmo de otimização matemática por enxame de partículas (Bouraine et al., 2020), ou Particle Swarm Optimization (PSO), para minimizar a diferença das matrizes de mapeamento geradas pelos sensores de distância, sendo a varredura da *pose* em um segundo momento M_1 comparada com a feita na primeira vez naquela posição, que passa a ser a referência M_0 .

$$v_i(k) = wv_i(k-1) + c_1r_1[Z_i - z_i(k-1)] + c_2r_2[Z_s - z_i(k-1)] \quad (8)$$

$$z_i(k) = z_i(k-1) + v_i(k) \quad (9)$$

Provocando deslocamentos $z_i = (\Delta x, \Delta y, \Delta \theta)$ em M_1 , com cada diferente deslocamento associado à uma partícula i do enxame, busca-se transformar a matriz M_1 no mais próximo possível de M_0 , retornando o valor de z_i com menor erro na função custo de minimização. As partículas tendem a se deslocar ao longo de K interações, a cada são atualizados o melhor resultado individual de uma partícula (Z_i) e o melhor resultado histórico entre todas elas (Z_s), buscando encontrar um valor ótimo para todas convergirem. O deslocamento individual v_i é medido na equação (8), levando em conta sua interação anterior com um dado coeficiente de inércia w , e em função de Z_i e Z_s que, ponderados pelos coeficientes cognitivo (c_1) e social (c_2), respectivamente, indicam o quanto cada partícula tende em direção a sua própria experiência e o quanto é levada pela experiência do conjunto (Voit, 2010). r_1 e r_2 são números aleatórios no intervalo $[0, 1]$. Na equação (9), z_i é atualizada para medir novamente a função custo.

Na simulação computacional desse recurso, utilizaram-se dez partículas com os coeficientes $c_1 = c_2 = 4,05$, ao longo de cem iterações. A função custo de minimização escolhida foi a *Dice Loss* que é obtida dividindo a intersecção pela união de dois conjuntos, comparando as imagens pixel a pixel. Na Figura 5 são mostrados os escaneamentos gerados na forma de matriz 200x200, onde uma posição é deslocada artificialmente e então corrigida com o resultado obtido pelo PSO.

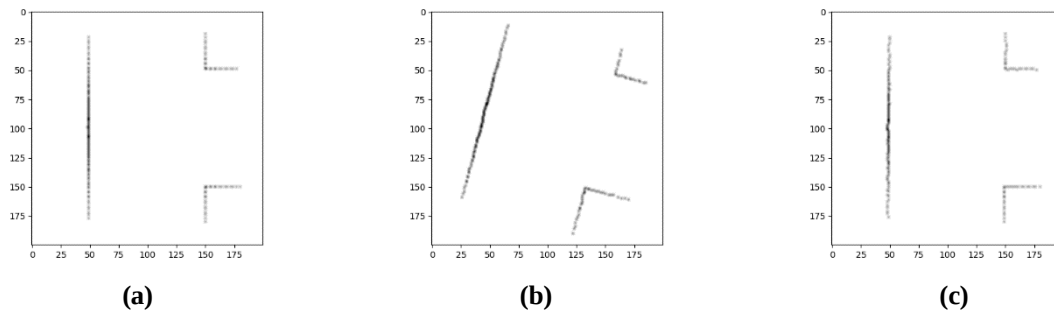


Figura 6. Mapeamento: (a) de referência M_0 ; (b) da segunda passagem M_1 ; (c) M_1 modificada com PSO.

4. RESULTADOS DA SIMULAÇÃO

Em todos os mapas da Figura 7(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. A 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. Conforme estabelecido, os nós são a posição onde o robô espera mais prolongadamente a emissão, permanecendo um tempo de exposição de 60 segundos. Na Figura 7(b) é possível observar a dosagem calculada em tempo real, tendo em vista que, na simulação, 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 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 3).

No teste do algoritmo PSO, o mapa 6 foi escolhido de base, incluindo as posições dos nós, conforme a Figura 5(a), mais o centro do mapa. O *scan-matching* é repetido seis vezes, em nove simulações distintas por posição, totalizando 378 medidas, de onde foram tirados os erros relativos médios, o valor da função custo e o tempo de realização. Além disso, foram testadas diferentes modalidades de alcance, ou seja, diferentes intervalos de valores que (x, y, θ) poderiam assumir (Tabela 4). Ressalta-se que (x, y) se referem a deslocamentos em centímetros e θ a variação em graus.

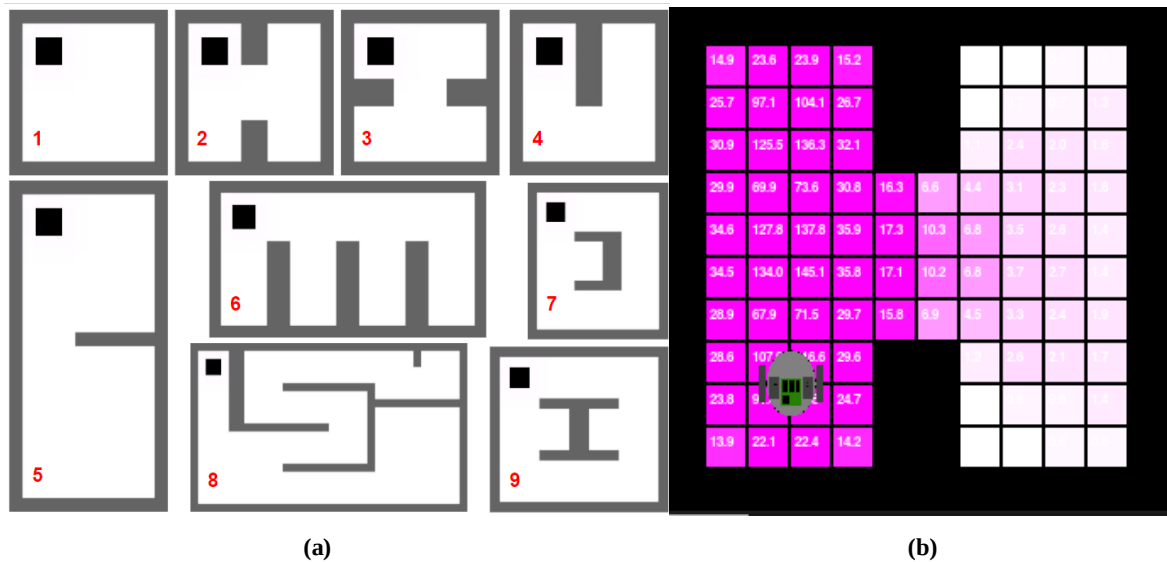


Figura 7. Mapas em simulação computacional: (a) todos os mapas utilizados; (b) Mapa 2 com limpeza em execução.

Tabela 3. 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	12,5	10,25	35,56	12,5
Dosagem média por área (mJ/cm^2)	99,40	86,28	87,42	69,40	113,68	86,57	73,94	113,59	87,73

Área totalmente limpa (%)	100	100	100	95,46	100	100	100	100	100
Tempo total (minutos)	13:47	9:57	9:57	7:57	31:11	21:23	17:31	76:15	20:57

Tabela 4. Entrada x saída do algoritmo PSO e desempenho.

Intervalo	Erro Δx	Erro Δy	Erro $\Delta \theta$	Erro médio	Função Custo	Tempo (segundos)
$ x , y , \theta < 10$	1,04%	5,31%	1,04%	2,19%	0,27	7,63
$10 \leq x , y , \theta < 20$	1,87%	11,73%	0,25%	4,62%	0,30	13,58
$20 \leq x , y , \theta < 30$	6,28%	20,41%	4,51%	10,40%	0,35	8,56
$ x , y , \theta < 30$	0,81%	6,43%	0,60%	2,62%	0,31	8,11
Média	2,50%	10,97%	1,60%	4,96%	0,31	9,47

5. CONCLUSÃO

Com o projeto desenvolvido, desenho mecânico e circuito eletroeletrônico esquematizados, 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 todos os mapas, sempre ultrapassando 95% de limpeza (Tabela 3), ainda que em um tempo que pode ser otimizado. Uma solução para isso seria, chegando em um nó e conhecendo a distância para o registro anterior, o robô emitir UV-C naquele ponto apenas o tempo certo para completar o restante da dosagem necessária, diminuindo a dosagem média por área, mas aumentando a eficiência.

A aplicação do algoritmo de *scan-matching* teve rendimento acima do esperado, considerando o baixo custo computacional e a simplicidade na utilização de PSO. Pela Tabela 4, verifica-se que os deslocamentos na horizontal Δx podem ser corrigidos com confiança superior a 90%, em todos os cenários até 30 cm, enquanto que os na vertical possuem maior instabilidade a partir de 10 cm. A variação angular $\Delta \theta$ é a medida mais confiável, com erro sempre inferior a 5%, indicando a facilidade do algoritmo em corrigi-la. O uso de dez partículas em cem iterações do PSO diminui o tempo de execução visando a operabilidade no robô, porém podem ser parâmetros aumentados para melhorar os resultados supracitados.

Pelo obtido na simulação, o projeto pode ser montado fisicamente com eficácia testada 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.

6. AGRADECIMENTOS

O presente trabalho teve apoio técnico do Grupo de Automação e Controle (GRACO) – UnB e da estudante da Universidade de Brasília, Patrícia Soares Micheletto. É dado a esses os devidos agradecimentos.

7. REFERÊNCIAS

- Abraham, J.P., Plourde, B.D., e Cheng, L., 2020. “Using heat to kill SARS-CoV-2”. *Reviews in medical virology*, 30(5), e2115. <https://doi.org/10.1002/rmv.2115>.
- ANAMT, 2020. “Brasil ultrapassa a marca de cem médicos mortos por Covid-19.” Disponível em: <<https://www.anamt.org.br/porta1/2020/05/21/brasil-ultrapassa-a-marca-de-cem-medicos-mortos-por-covid-19-dois-por-dia/>>. Acesso em junho 2020.
- Arguelles, P., 2020. “Estimating UV-C Sterilization Dosage for COVID-19 Pandemic Mitigation Efforts.”. doi: 10.13140/RG.2.2.12837.65761.
- Back, N., Ogliari, A., Dias A. e da Silva, J. C., 2008. *Projeto integrado de produtos: Planejamento, Concepção E Modelagem*, Ed. Manole, São Paulo.
- 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., and 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>.
- Block, M. S., e Rowan, B.G., 2020. “Hypochlorous Acid: A Review”. *Journal of oral and maxillofacial surgery*, 78(9), 1461–1466. <https://doi.org/10.1016/j.joms.2020.06.029>.
- Bouraine, S., Bougouffa, A. e Azouaoui, O., 2020. "NDT-PSO, a New NDT based SLAM Approach using Particle Swarm Optimization”, In *16th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pp. 321-326, doi: 10.1109/ICARCV50220.2020.9305519.
- Conselho Federal de Química (CFQ), 2020. “Cartilha: Perguntas e Respostas: Água Sanitária”. Disponível em: <http://cfq.org.br/noticia/perguntas-e-respostas-agua-sanitaria-2/>. Acesso em: dezembro 2020.
- Fernandes, C., 2021. “Zhello, o robô português que vai garantir a desinfecção de espaços públicos”. Disponível em: <https://www.natgeo.pt/ciencia/2021/11/zhello-o-robo-portugues-que-vai-garantir-a-desinfecao-de-espacos-publicos>. Acesso em janeiro 2022.
- Fischer, R.J., Morris, D.H., Van Doremalen, N., Sarchette, S., Matson, M. J., Bushmaker, T., Yinda, C. K., Seifert, S. N., Gamble, A., Williamson, B. N., Judson, S. D., de Wit, E., Lloyd-Smith, J. O., and Munster, V. J., 2020. “Assessment of N95 respirator decontamination and re-use for SARS-CoV-2”. *MedRxiv: the preprint server for health sciences*, Disponível em: <<https://doi.org/10.1101/2020.04.11.20062018>>. Acesso em janeiro 2022.
- Guettari, M., Gharbi, I. e Hamza, S., 2021. “UVC disinfection robot”. *Environ Sci Pollut Res*. Vol., 28, p. 40394–40399.
- 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*. Vol.2, p. 959-977. doi: <https://doi.org/10.3390/ijgi2040959>
- Juchem, P.P., Hochberg, J., Winogron, A., Ardenghy, M., English, R., 1998. “Riscos à Saúde da Radiação Ultravioleta.” *Rev. Bras. Cir. Plást.* Vol. 13(2), p. 31-60.
- Liao, L., Xiao, W., Zhao, M., Yu, X., Wang, H., Wang, Q., Chu, S., & Cui, Y., 2020. “Can N95 Respirators Be Reused after Disinfection? How Many Times?” *ACS nano*. Vol. 14(5), p. 6348–6356. doi: <https://doi.org/10.1021/acsnano.0c03597>
- Madureira, O.M. (2015). *Metodologia do Projeto- Planejamento, Execução e Gerenciamento*, Ed. Blucher, São Paulo.
- Pahl, G., Beitz, W., Feldhusen, J., and Grote, K. 2005, *Projeto na Engenharia: Fundamentos do Desenvolvimento Eficaz de Produtos - Métodos e Aplicações*, Ed. Blucher, São Paulo, 411p.
- Sarada, B.V., Vijay, R., Johnson, R., Rao, T. N., and Padmanabham, G. 2020. “Fight Against COVID-19: ARCI’s Technologies for Disinfection”. *Transactions of the Indian National Academy of Engineering*. Vol. 5(2), p. 349–354. doi: <https://doi.org/10.1007/s41403-020-00153-3>
- Secretaria de Estado da Saúde do Espírito Santo, 2009. *Guia de referência para limpeza, desinfecção e esterilização de artigos em Serviços de Saúde*, 1ª edição.
- Voit, J., 2010. “Otimização por enxame de partículas com congregação passiva seletiva.” In *Ph. D. dissertation*, Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brasil.

8. RESPONSABILIDADE AUTORAL

Os autores são os únicos responsáveis pelo conteúdo deste trabalho.

ROBOT DESIGN FOR HOSPITAL DISINFECTION

Matheus Virgílio da Silva Ferreira, matheusvirgiliomcp@gmail.com¹
Walter de Britto Vidal Filho, wbritto@unb.br¹

¹Mechanical Engineering Department, University of Brasília, University Camp Darcy Ribeiro, Brasília-DF

Abstract. After studying different decontamination methods to identify the most suitable one to be automated in the cleaning of hospital environments, this study intends to apply it in the development of a disinfection robot guided by an autonomous navigation algorithm. After verifying a series of methods, decontamination by ultraviolet radiation type C lamps was chosen and, from that, computer simulations were made to measure its efficiency with a navigation system inspired by the Trémaux maze-solving algorithm. The simulation has as variables the robot design parameters (such as dimensions, speed, lamp power and UV emission time), stipulated with the assumed methodology of problem definition and functional decomposition. With regard to its structure, the mechanical and electronic systems were designed aiming at portability and low cost, following the stipulated in the requirements table. The results showed high cleaning rates in nine different maps that simulate real environments, with a percentage of totally clean areas above 95% in all of them. A scan-matching application was also tested to correct the odometric trajectory, which provided estimates with an average relative error inferior to 5% for small variations, using few computational resources.

Keywords: UV-C, simulation, PSO, navigation