

# Uma abordagem para desvio de obstáculos na assistência ao controle de um quadricóptero em tempo real

Bruno S. Giovaninia, Luis Claudio Batista da Silva<sup>a,b</sup>, Paulo Fernando Ferreira Rosa<sup>\*a</sup>

<sup>a</sup> Instituto Militar de Engenharia (IME)

Praça General Tibúrcio, 80, 22290-270, Praia Vermelha, Rio de Janeiro, RJ, Brasil

<sup>b</sup>Centro Federal de Educação Tecnológica Celso Suckow da Fonseca (CEFET/RJ)

Av. Governador Roberto Silveira, 1900, Prado, Nova Friburgo, RJ, Brasil

\*rpaulo@ime.eb.br

**RESUMO:** O uso de quadricópteros para missões civis e militares vem aumentando e os desafios envolvidos no seu controle, principalmente em ambientes fechados e restritos, atraem a atenção dos pesquisadores. Desta forma, uma abordagem automática para desvio de obstáculos ganha importância, dado os riscos de acidentes envolvidos no seu controle. Este trabalho apresenta uma abordagem para evitar obstáculos automaticamente, assistindo o controle manual de um quadricóptero em ambiente restrito. O método é baseado na constante estimativa da trajetória futura da aeronave, considerando sua dinâmica, status e controle, enquanto uma malha de ocupação, representando seus arredores, é construída a partir das distâncias medidas por sonares embarcados. Estas informações são usadas para determinar colisões iminentes e, em caso positivo, substituir o controle do usuário por um controle automático PID, mantendo sua localização. Três experimentos foram realizados em um simulador, no qual foi possível validar a solução proposta, evitando obstáculos com segurança.

**PALAVRAS-CHAVE:** Desvio de obstáculos. Ambientes fechados. Controle PID. Quadricóptero.

**ABSTRACT:** The use of quadrotors to civilian and military missions has been augmented and the challenges involved on controlling it, mainly in indoors and restricted environments have attracted researchers. Thus, an automatic collision avoidance approach is of utmost importance, given the complexity of control and the risk of accidents involved in their use. This paper presents an approach for automatic obstacle avoidance assisting a manually controlled quadrotor. The method is based on constant estimation of the quadrotor's future path considering its dynamics, status, control while a grid representation of the nearby environment is built from distances measured by on-board sonars. All information is used to determine imminent collisions and, if so, overrides the operator's control by an automatic PID control, keeping its localization. Three experiments were executed on a simulator where the solution was evaluated and showed that the proposed methodology fulfilled its purpose by avoiding obstacles safely.

**KEYWORDS:** Obstacle avoidance. Indoor Environments. PID control. Quadrotor.

## 1. INTRODUÇÃO

Aeronaves remotamente pilotadas do tipo asas rotativas (multirotores) tem sido amplamente aplicadas em tarefas de defesa, tais como vigilância, mapeamento, busca e salvamento, reconhecimento, aquisição de alvos, dentre outros. Entretanto, muitos são os desafios relacionados para o uso seguro destes veículos em ambientes com obstáculos, o que tem atraído a atenção de pesquisadores de diversas áreas da robótica.

Para o objetivo deste trabalho, foi empregada uma plataforma de um quadricóptero, cuja configuração é dotada de propulsão por quatro hélices conectadas a quatro motores elétricos controlados eletronicamente. Dentre suas características, destacam-se a habilidade de realizar decolagem e aterrissagem vertical, pairar e executar manobras rápidas em diferentes condições de ambiente. Porém, controlar um quadricóptero não é uma tarefa simples quando se faz necessário voar próximo de obstáculos. Mesmo pilotos treinados podem passar por situações onde é difícil garantir um voo seguro, como, por exemplo, quando há perda de contato visual. Desta forma, um método automático para evitar colisões para assistir a operação desta plataforma é requerido.

O objetivo desse trabalho é implementar o desvio de obstáculos de um quadricóptero de forma automática, permitindo que sua operação mantenha o foco na missão global e evitar, assim, possíveis acidentes. Especificamente, propõe-se um método que consiste em evitar colisões estimando constantemente a trajetória futura do veículo, com base na sua dinâmica, no seu estado atual, no comando de controle corrente enviado pelo operador

e na distância para os obstáculos (medida através de um conjunto de sonares embarcados). Desta forma, facilita-se o controle de voo contribuindo para a segurança na pilotagem da aeronave em ambientes restritos.

A contribuição deste trabalho consiste na elaboração de um algoritmo para assistir a operação do quadricóptero evitando obstáculos baseada em localização e mapeamento simultâneos, do inglês *Simultaneous Localization and Mapping* (SLAM), com a localização global do veículo sendo calculada continuamente durante o voo através de seus dados de navegação, enquanto três sonares embarcados, posicionados e apontados para frente do veículo, medem as distâncias relativas para os obstáculos ao seu redor. Um mapeamento do ambiente é realizado através do *Octomap*, uma implementação de uma malha de um mapa de ocupação, do inglês, *Occupancy Grid Mapping* [1]. Simultaneamente, a trajetória futura é estimada baseada na atitude e velocidade atual do quadricóptero. Além disso, o algoritmo constantemente testa se uma colisão contra obstáculos pode ocorrer. Isto é realizado basicamente comparando a trajetória futura com a posição de possíveis obstáculos em um horizonte de tempo previamente definido. Se a colisão é iminente, o comando do operador é substituído por um controle Proporcional-Integral-Derivativo (PID) automático, que tem o objetivo de manter o veículo na sua posição por um segundo e devolver o controle ao operador, em seguida. Caso contrário, o método proposto não substitui o comando do operador. Existem vários passos na implementação do método. Uma visão geral da arquitetura proposta pode ser verificada na Fig. 1.

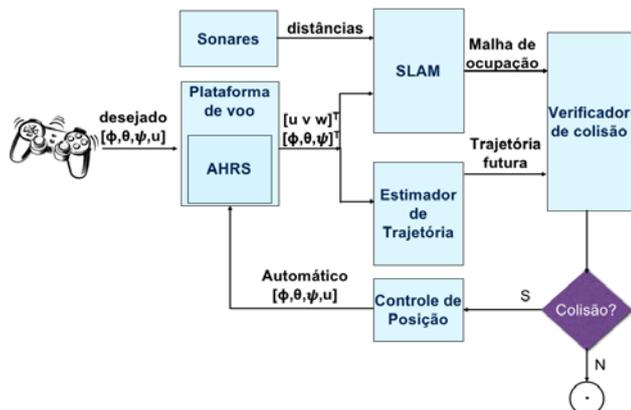


Fig. 1 – Uma visão geral da arquitetura proposta.

Primeiramente, o controle desejado pelo operador  $[\phi, \theta, \psi, u]$ , denotado pela rolagem ( $\phi$ ), arfagem ( $\theta$ ), guinada ( $\psi$ ) e intensidade ( $u$ ), é enviado ao quadricóptero, o qual fornece sua atitude através do sistema de referência de direção e atitude da aeronave, do inglês, *Attitude and Reference System* (AHRS), e sua velocidade no próprio eixo  $[u \ v \ w]^T$ , utilizando algoritmos de visão [2]. Tais informações são usadas, tanto pelo estimador de trajetória para calcular a trajetória futura, quanto pelo módulo SLAM, que utiliza estas informações em conjunto com as distâncias obtidas dos sonares para determinar a malha de ocupação que representa o ambiente. A trajetória futura e a malha de ocupação são confrontadas através de um módulo de verificação de colisão, o qual verifica a existência de uma colisão iminente. Se não há colisão iminente, o sistema não interfere e o controle permanece com o operador. Caso contrário, um controlador de posição determina o controle automático PID necessário para manter a posição do quadricóptero e passa para a aeronave, substituindo o controle do operador pelo tempo de 1s, evitando a colisão.

A solução foi concebida e preparada para embarque em um quadricóptero real através de um mini-PC (Raspberry Pi®), que são computadores que possuem pequenas dimensões e baixa capacidade de processamento computacional. Além disso ela foi implementada e testada em um simulador chamado TUM *Simulator* [3], um simulador de quadricóptero desenvolvido pela Universidade Técnica de Munique, utilizando o sistema operacional robótico ROS, do inglês *Robot Operating System*, e a plataforma de simulação Gazebo. Em ambiente simulado, a solução obteve sucesso em evitar os obstáculos em três experimentos elaborados, em que o quadricóptero foi pilotado diretamente contra um obstáculo, passou por um corredor que se torna cada vez mais estreito e percorreu um labirinto projetado, cumprindo seu objetivo de forma efetiva.

Este trabalho está dividido, além desta introdução, da seguinte maneira: a seção 2 apresenta os trabalhos acadêmicos de maior relevância para o desenvolvimento deste trabalho. A seção 3 aborda as etapas, metodologia utilizada e o desenvolvimento do trabalho. A seção 4 apresenta os resultados obtidos a partir dos experimentos realizados no simulador. A última seção traz as conclusões deste artigo.

## 2. TRABALHOS RELACIONADOS

Nos últimos anos, veículos aéreos não tripulados receberam uma maior atenção da comunidade de robótica. Muitos autores focaram na modelagem e no controle destes veículos

[4], com destaque para os quadricópteros [5].

Nesse contexto, um tutorial foi desenvolvido por [6], realizando estudos abrangentes sobre a configuração da plataforma, metodologias de modelagem, modelagem compreensiva não linear, os efeitos aerodinâmicos, identificação e simulação de um quadricóptero. O autor cita em seu trabalho que o modelo é a fundação da caracterização do veículo e o primeiro passo para o controle e simulação, definindo as relações entre as entradas e saídas do problema. Porém, por ser considerado um sistema complexo não linear de difícil formulação matemática e grande complexidade, sistemas de identificação, que envolvem a análise de percepções físicas e das entradas e saídas para a derivação do modelo, são bastante utilizados. Além disso, simplificações são normalmente assumidas como razoáveis e essenciais para a modelagem, tais como: (a) a estrutura do veículo sendo supostamente rígida e simétrica e (b) o centro de gravidade e a origem do referencial fixado ao corpo do veículo coincidirem.

Sistemas de identificação foram usados por [7] através da criação de um controlador PID para obter a estabilidade de um quadricóptero baseado em resposta rápida. Por outro lado, [8] propôs uma rede neural adaptativa para estabilizar o quadricóptero levando em consideração erros de modelagem e distúrbios do vento.

A partir de um modelo conhecido, diversas são as aplicações em estudo. Para a maioria, como navegação autônoma, tarefas multiagentes e desvio de obstáculos, a estimação da localização do veículo é considerada um grande desafio [9]. Quando se trata de ambientes fechados, a visão computacional, que se utiliza de câmeras e processamento de imagem para estimação, é uma das linhas de pesquisa sobre o problema [10]. Outra linha utiliza-se de sensores exteroceptivos para tal tarefa, como sonares [11] e laser [12]. Já para ambientes externos, torna-se possível a utilização do Sistema de Posicionamento Global (GPS), e alguns trabalhos que utilizam essa tecnologia foram desenvolvidos, como [13] e [14].

Quando se fala especificamente sobre desvio de obstáculos em tempo real, esse campo é bastante explorado em robótica. Para resolver essa questão, várias são as abordagens que vêm sendo adotadas. Dentre elas, os métodos baseados em campos potenciais artificiais, onde sensores de distância são usados e suas medidas tratadas como vetores de repulsão de forças. Utilizando essa abordagem, [15] implementou um método para prevenção de colisão reativa rápida para operação segura nas intermediações de estruturas como prédios e vegetação. Para isso, foram usados sensores lasers 3D e câmeras estéreo para detecção visual de obstáculos. O método de janelas dinâmicas, que incorpora a dinâmica do quadricóptero ao problema através da redução do espaço de busca das possíveis velocidades alcançáveis em um curto período de tempo (janela dinâmica), foi implementado por [16] em um robô de direção diferencial. Em sua abordagem, foram considerados os obstáculos próximos à trajetória utilizando uma representação de histograma para estimar a probabilidade de colisão. Outros métodos também podem ser citados, como o método de obstáculos com velocidade, que define um conjunto de velocidades possíveis que resultaria em uma colisão entre o robô e um obstáculo se movendo em uma certa velocidade [17][18]; o método de estados de colisões inevitáveis, que considera um estado em que uma colisão eventualmente irá ocorrer, independente da trajetória a ser estabelecida, que leva em conta a dinâmica tanto do robô quanto dos obstáculos, fixos ou móveis [19]; dentre outros.

[12] também desenvolveu um quadricóptero totalmente autônomo em ambiente *indoor*, utilizando para o desvio de obstáculos as medidas do sensor de varredura a laser em miniatura Hokuyo-URG® embarcado.

Em [20], foi realizado um trabalho com objetivo similar ao aqui proposto, porém somente com o foco no desvio do obstáculo, considerando o ambiente, a localização relativa do veículo e a posição dos obstáculos já previamente conhecidos. Para determinar o estado do veículo, foi utilizado um Filtro de Kalman, que obtém informações de localização de um equipamento instalado no ambiente para rastreamento do veículo chamado OptiTrack, além de informações de velocidade, atitude e velocidade angular obtidas do próprio quadricóptero. Já [11], por sua vez, também desenvolveu um método para assistir a operação de um quadricóptero com desvio automático de obstáculos. Um mapa dos arredores do veículo é construído usando FastSLAM, uma variação do problema SLAM que utiliza filtro de partículas, em que cada partícula apresenta o estado do problema com seus respectivos pesos definidos a partir da similaridade entre o estado estimado e o estado observado e que necessita de capacidade computacional elevada, tornando-se proibitivo para mini-PCs embarcados.

Diferente das abordagens dos trabalhos previamente mencionados, a contribuição deste trabalho vem da concepção de um algoritmo portátil e flexível para o desvio de obstáculos em tempo real embarcado, onde a capacidade computacional limitada de mini-PCs é considerada, além de ser uma solução onde nenhum conhecimento prévio do ambiente é necessário.

### 3. ABORDAGEM PARA DESVIOS DE OBSTÁCULOS

Considerando um quadricóptero como um robô com uma dinâmica não linear genérica em um espaço arbitrário,

$\mathcal{X} \subset \mathbb{R}^n$  sendo seu espaço de estados e  $\mathcal{U} \subset \mathbb{R}^l$  o espaço de entrada de controle, em tempo contínuo, a dinâmica do robô é uma função em  $f \in \mathcal{X} \times \mathcal{U} \times \mathbb{R} \rightarrow \mathbb{R}^n$ , dada por:

$$\dot{\mathbf{x}} = f(\mathbf{x}(t), \mathbf{u}(t), t) \quad (1)$$

onde  $t$  é o tempo,  $\mathbf{x}(t)$  é o estado do robô em  $t$  e  $\mathbf{u}(t)$  é a entrada de operação no tempo  $t$ . Portanto, dado um estado inicial  $\mathbf{x}_0 = \mathbf{x}(0)$  e uma constante de entrada de operação  $\mathbf{u}$ , o estado do robô em um instante  $t > 0$  é dado por:

$$\mathbf{x} = g(\mathbf{x}, \mathbf{u}, t) \quad (2)$$

onde  $g \in \mathcal{X} \times \mathcal{U} \times \mathbb{R} \rightarrow \mathcal{X}_T$  representa a solução da Eq.1.

Porém, quando se considera um ambiente repleto de obstáculos, o espaço de estados do robô se restringe às posições não ocupadas por eles. Define-se então  $\mathcal{O} \subset \mathbb{R}^3$ , como a subárea do espaço de movimentos possíveis do robô ocupada por obstáculos e as regiões que estão escondidas por eles quando vistas pelo estado corrente do robô. Além disso,  $\mathcal{A}(\mathbf{x})$  como a subárea ocupada pelo robô no estado  $\mathbf{x} \in \mathcal{X}$ . Assim, o problema passa a ser definido como identificar uma variação necessária  $\Delta \mathbf{u} \in \mathcal{U}$  sobre o controle de operação  $\mathbf{u}$ , dado o atual estado  $\mathbf{x}$  do robô, a fim de que se mantenha a sua posição em qualquer momento até se alcançar um período previamente definido  $t \in \mathbb{R}$  conforme a relação:

determinar:  $\Delta \mathbf{u}$

$$\text{Sujeito a: } \forall t \in [0, t], \mathcal{A}(\mathbf{g}(\mathbf{x}, \mathbf{u} + \Delta \mathbf{u}, t)) \cap \mathcal{O} = \emptyset \quad (3)$$

É importante notar que, caso a entrada de operação obedea às restrições da Eq. 3, ou seja, corresponda a um deslocamento seguro,  $\Delta \mathbf{u} = 0$  e nenhuma alteração do controle será realizada.

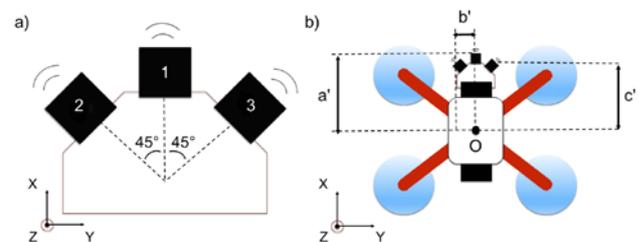
Para resolver este problema, uma abordagem é proposta para evitar que o quadricóptero colida. Algumas premissas e simplificações foram consideradas para a concepção deste trabalho:

- Ambiente *indoor* totalmente desconhecido: nenhuma informação prévia ou algum tipo de marcação no ambiente é considerado. Toda e qualquer informação sobre os arredores do veículo é obtida em tempo real durante o sobrevoo. Além disso, a utilização de Sistemas de Posicionamento Global, da sigla em inglês GPS, é proibitiva dada a característica *indoor* do ambiente.
- Natureza estática dos obstáculos: foi definido, como premissa, que os obstáculos presentes no ambiente não se movem e estão ali posicionados desde o início do sobrevoo.
- Quadricóptero modelado como uma esfera: o veículo é considerado uma região esférica quando comparado com a região ocupada pelo obstáculo. Uma modelagem precisa do formato do veículo foi dispensada dada a complexidade gerada na verificação da sobreposição entre a região ocupada pelo veículo e a região ocupada pelo obstáculo.
- Dados de navegação precisos: a utilização de um quadricóptero capaz de fornecer informações de sua velocidade com uma precisão suficiente para que problemas de cálculo de sua localização não prejudiquem o mapeamento do ambiente em tempo real.

Diversas etapas constituem a arquitetura, conforme foi ilustrado na Fig. 1. Os tópicos que suportam estas etapas serão descritos a seguir.

#### 3.1 Medidas de distância: sonares

Nesta etapa, as informações são lidas dos sensores em tempo real e traduzidas para distâncias a serem consumidas pela etapa de SLAM para mapeamento do ambiente. A configuração dos sensores embarcados é mostrada na Fig. 2.



**Fig. 2– Posição relativa dos sensores em relação ao corpo do quadricóptero. (a) Ângulo de rotação em Z. (b) Deslocamento em relação ao centro de massa.**

Três sonares foram transladados nos eixos  $X$  e  $Y$  e rotacionados no eixo  $Z$  em graus, apontando para a frente do quadricóptero, conforme pode ser visualizado na Fig. 2, onde “ $a$ ”, “ $b$ ” e “ $c$ ” denotam as distâncias transladadas dos sonares em relação ao centro de massa do veículo nos eixos de referência. Também, um filtro de medida máxima, do inglês *maxrange*, é aplicado para eliminar dados espúrios e melhorar a confiança das medidas passadas ao algoritmo.

#### 3.2 Plataforma de voo: quadricóptero

Quadrópteros são veículos aéreos não tripulados (VANT) de pequeno porte, do tipo asa rotativa, classificados como micro-VANT ou mini-VANT, de acordo com seu tamanho e capacidade de carga. Possui capacidade de decolagem e aterrissagem vertical, do inglês *Vertical Take-off and Landing* (VTOL), impulsionado por quatro motores elétricos independentes posicionados em suas extremidades e controlados eletronicamente para estabilização autônoma do voo [22]. Além disso, é considerado um sistema não-linear, fortemente acoplado com 6-DOF (sigla do inglês para Graus de Liberdade), onde três movimentos são lineares ( $x, y, z$ ) e os outros três angulares ( $\phi, \theta, \psi$ ). Porém, possui apenas quatro atuadores, sendo considerado então um sistema subatuado, com dois movimentos lineares ( $x, y$ ) dependentes dos movimentos angulares ( $\phi, \theta$ ). As forças e momentos atuando no quadróptero são produzidos pelas hélices ligadas aos rotores que, combinados, geram um impulso principal e os torques de rolagem ( $\phi$ ), arfagem ( $\theta$ ) e guinada ( $\psi$ ) [23].

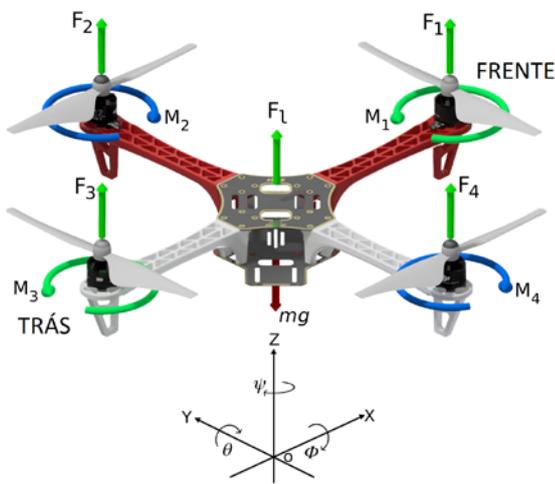


Fig. 3– Forças  $F_i$  e momentos  $M_i$  de um quadróptero.

A Fig. 3 apresenta o esquema com as principais forças e momentos gerados. Os motores  $M_i$  produzem as forças  $F_i$  (para  $i = 1, \dots, 4$ ). Os motores trabalham em pares de forma que o motor  $M_1$  (frente) e o motor  $M_3$  rodam no sentido anti-horário, enquanto que  $M_2$  e  $M_4$  giram no sentido horário, de forma a balancear o torque total do sistema.  $F_l$  representa a soma dos impulsos de cada motor. O referencial inercial é denotado por  $(x, y, z)$  e  $(\phi, \theta, \psi)$  representam os movimentos de rolagem, arfagem e guinada, respectivamente. A força peso é representada por  $mg$ .

### 3.3 Estimador de trajetória

A dinâmica do quadróptero é a base para a estimativa de trajetória. Considerando que o quadróptero utilizado fornece sua velocidade e atitude (rolagem, arfagem e guinada) em tempo real, as fontes destas informações são desprezadas e somente a sua cinemática é considerada. As transformações relacionadas ao estudo da cinemática envolve uma translação e sucessivas rotações desde o referencial inercial  $\mathcal{F}_i$  até o referencial fixado ao corpo do quadróptero  $\mathcal{F}_b$ . A Fig. 4 mostra as transformações passo a passo desde  $\mathcal{F}_i$ , passando por três referenciais intermediários, incluindo uma translação de  $\mathcal{F}_i$  para  $\mathcal{F}_v$  e três rotações  $\phi, \theta, \psi$  para a sequência  $\mathcal{F}_v - \mathcal{F}_{v1} - \mathcal{F}_{v2} - \mathcal{F}_b$  [24].

Dessa forma, sendo  $R_v^b(\phi, \theta, \psi) = R_{v2}^b(\phi)R_{v1}^{v2}(\theta)R_v^{v1}$  a matriz

de rotação que representa a transformação de  $\mathcal{F}_v$  para  $\mathcal{F}_b$ , a localização corrente  $x$ , a velocidade  $v$  e a atitude fornecida  $\theta$  pelo quadróptero, a trajetória é calculada através da integração no tempo, com as grandezas  $x$  e  $\theta$  usadas como condições iniciais para estabelecimento da trajetória e  $v$  é considerada constante durante todo seu cálculo. Esta premissa é razoável, dado que o tempo de predição, que determina o tamanho da trajetória futura, é pequeno com valor próximo de  $\Delta t$ . Um passo da predição da trajetória é mostrado na Eq. 4, onde  $dt$  é uma constante representando o intervalo de duração entre dois passos da trajetória,  $x_0$  é a condição de localização inicial da trajetória e  $x_t$  são os passos calculados.

$$x_{t+1} = x_t + R_v^b(\theta)v * dt \quad (4)$$

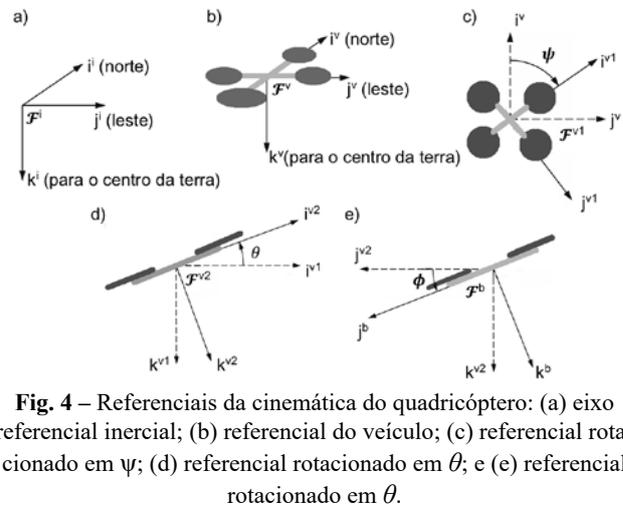


Fig. 4 – Referenciais da cinemática do quadróptero: (a) eixo referencial inercial; (b) referencial do veículo; (c) referencial rotacionado em  $\psi$ ; (d) referencial rotacionado em  $\theta$ ; e (e) referencial rotacionado em  $\phi$ .

### 3.4 Localização e Mapeamento Simultâneos

Nosso enfoque é considerar um ambiente totalmente desconhecido, sob incertezas das medidas de distâncias obtidas através dos sonares. O objetivo do SLAM é estimar a localização de um robô e o mapeamento dos seus arredores simultaneamente utilizando uma abordagem probabilística. Seja  $x_{1:t}$  a variável que denota o caminho do veículo e  $z_{1:t}$  todas as medidas e  $u_{1:t}$  todas as entradas de controle, onde  $1:t$  representa o intervalo de tempo do instante  $1$  até o instante  $t$ . A propriedade de independência condicional do problema SLAM [1] permite decompor o SLAM em um problema de localização e um problema de mapeamento independentes (Eq. 5).

$$p(x_{1:t}, m | z_{1:t}, u_{1:t}) = p(x_{1:t} | z_{1:t}, u_{1:t})p(m | z_{1:t}, x_{1:t}) \quad (5)$$

Desta forma, tratamos a localização  $x$  de forma independente, obtendo-a diretamente a partir da velocidade  $v$  e atitude  $\theta$  adquiridas do quadróptero conforme pode ser observado na Eq. 6 e, posteriormente, o mapeamento é realizado através de uma abordagem probabilística. Uma malha de ocupação (*occupancy grid map*) foi utilizada para tratar este problema, gerando mapas consistentes a partir de medidas ruidosas e incertas, dada a localização do robô.

$$x = R_v^b(\theta)v * dt \quad (6)$$

Foi adotado o *Octomap* para gerar modelos volumétricos 3D do ambiente, baseado em *Octrees* e estimação probabilística de ocupação. Como entrada, *Octomap* recebe medidas obtidas dos sonares embarcados e a resolução desejada e re-

torna uma malha de ocupação representando o ambiente dos arredores do quadricóptero durante o sobrevoo. Esta explicitamente representa tanto as áreas ocupadas, como também as áreas livres e desconhecidas. Sua principal propriedade é a atualizar probabilisticamente a malha de ocupação em tempo real, mantendo o consumo de memória baixo [21]. A Fig. 5 apresenta um cenário (a) e o mapeamento gerado pelo *Octomap* (b) após sobrevoo em ambiente simulado.

### 3.5 Detecção de colisão

Uma possível colisão é verificada baseada na interseção entre a região ocupada pelo obstáculo e pelo quadricóptero. Aqui, a região do quadricóptero é simplificada como uma esfera e a região dos obstáculos inclui áreas desconhecidas e a área acrescida formada a partir do raio da esfera.

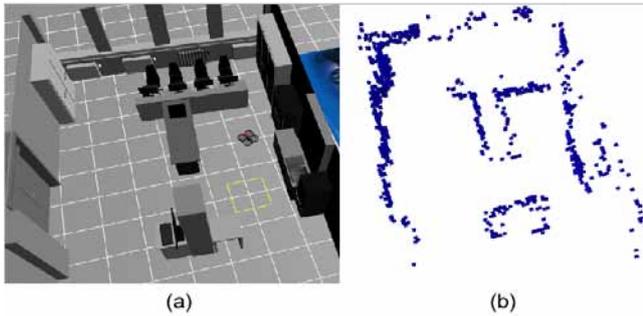


Fig. 5– (a) Ambiente simulado. (b) Mapeamento *Octomap*.

Durante o sobrevoo, uma área de verificação por colisões é delimitada a partir das possíveis posições alcançáveis em um intervalo  $t \in [0, \tau]$  enquanto a trajetória futura é constantemente calculada. Ambas são baseadas na dinâmica e estado do quadricóptero, incluindo sua velocidade e localização.

Dada a localização  $x$  do quadricóptero, a área de verificação de colisão é delimitada conforme metodologia a seguir. Inicialmente, as distâncias alcançáveis a partir da sua posição  $x$  denotada por  $\Delta_x$ , que é definida por  $\Delta_x = v_{max} * \tau$  onde  $v_{max}$  é um valor fixo configurado representando a velocidade máxima do veículo e  $\tau$  é o tempo de predição também configurado. Em seguida, todas as células da malha dentro do intervalo  $[x - \frac{\Delta_x}{2}, x + \frac{\Delta_x}{2}]$  são percorridas e, para cada uma, pergunta-se para o *Octomap* se ela está ocupada. Caso ocupada, o processo continua em duas etapas. Primeiramente, evita-se que colisões sejam verificadas contra células que não estão no caminho do quadricóptero. Por exemplo, quando o quadricóptero se movimenta paralelamente a uma parede que, por mais que a célula esteja suficientemente próxima do quadricóptero, não será alcançada. Para isso, dado o vetor distância  $d_i$  entre a localização atual do quadricóptero  $x_i$  e a posição  $x_c$  da célula expresso por  $d_i = x_c - x_i$ , é verificado se o ângulo  $\alpha$  entre  $d_i$  e o vetor velocidade do quadricóptero  $v_i$  está no intervalo  $-45 < \alpha < 45$ , conforme é mostrado na Eq. 7. Em caso positivo, a abordagem clássica de *C-Obstacles* de Latombe [25] é utilizada, expandindo-se a região ocupada pela célula pela região de possíveis configurações ocupadas pelo quadricóptero que colidem com a célula. Esta área expandida é, então, testada para uma possível interseção com cada passo da trajetória futura calculada. Em caso positivo, uma colisão é iminente e um controle automático será realizado substituindo o controle do operador. Caso contrário, o operador segue controlando normalmente o quadricóptero

sem intervenção do sistema anticolisão.

$$\cos \alpha = \frac{\langle d, v \rangle}{|d||v|} \quad (7)$$

A Fig. 6 representa a verificação de colisão em um cenário de sobrevoo do quadricóptero sendo comandado pelo operador contra um obstáculo.

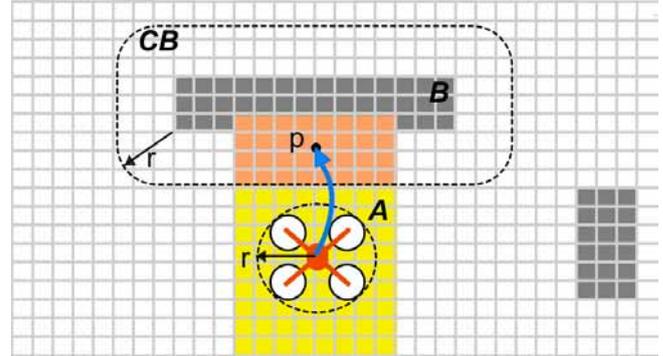


Fig. 6 – Uma ilustração da verificação de colisão durante o voo do quadricóptero. A interseção (em laranja) entre  $\Delta_x$  e a região de obstáculo (CB) representa as futuras possíveis posições com colisão iminente.

O ambiente é mapeado através de uma malha de ocupação sendo a região ocupada pelo quadricóptero dada por  $\mathcal{A}$  e a região ocupada pelo obstáculo  $\cdot$  representa o *C-Obstacle* de  $\cdot$ . A área em amarelo define a área  $\Delta_x$  e a área em laranja mostra o conjunto de células de  $\Delta_x$  que intersectam  $CB$ . Qualquer trajetória que leve à área laranja, como a trajetória representada pela seta e termina no ponto  $p$  indica uma colisão iminente.

### 3.6 Controle automático

Dada uma colisão iminente, um controle automático é realizado para levar o veículo para a última localização segura conhecida. Este local foi convenicionado como exatamente o local em que o veículo se encontra no momento que a trajetória de colisão é calculada. Além disso, a orientação também é mantida para uma melhor experiência do operador e a duração do controle automático é de  $\tau$ , podendo ser configurável.

A abordagem utilizada neste passo utiliza-se de quatro controladores PIDs combinados para controlar a localização do quadricóptero ( $x, y, z$ ) e sua orientação ( $\psi$ ) como realizado em [26]. O esquema de controle é mostrado na Fig. 7. Os erros ( $e_x(t), e_y(t), e_z(t), e_\psi(t)$ ) são calculados a partir da diferença entre a posição segura desejada e os valores retornados a partir do AHRS do quadricóptero e do estimador de localização global. Então, os erros são mapeados para comandos de controle, restringidos por valores aceitáveis. O controle é calculado na mesma frequência que o quadricóptero envia dados de navegação (aproximadamente 200Hz). As constantes de controle proporcional, integral e derivativa  $k_p, k_i$  e  $k_d$ , para cada controle, foram estimadas empiricamente

## 4. RESULTADOS EXPERIMENTAIS

Nesta seção, serão apresentados os resultados obtidos através dos experimentos executados para avaliar o algoritmo desenvolvido. Primeiramente, foi analisado o comportamento do Verificador de Colisão, variando-se o tempo de predição. A seguir, três experimentos completos foram reali-

zados para avaliar o comportamento da solução em três diferentes situações: (a) o quadricóptero foi pilotado diretamente contra um obstáculo, (b) passou por um corredor que se torna cada vez mais estreito e (c) percorreu um labirinto projetado.

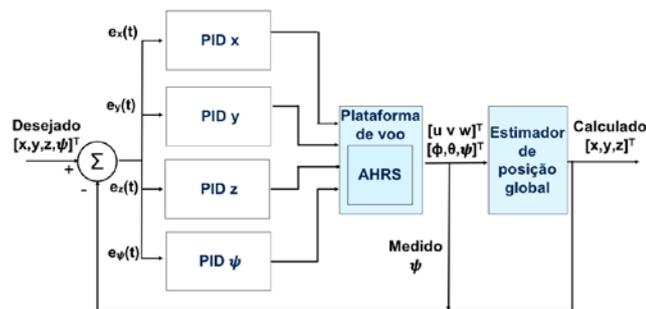


Fig. 7– Esquema do controle PID do controle automático para evitar a colisão.

#### 4.1 Tempo de predição

O tempo de predição  $\tau$  tem influência no comportamento da solução na determinação da área de verificação de colisão. Quanto menor é  $\tau$ , menor será a área de verificação de colisão e mais perto do obstáculo o quadricóptero irá parar. A Fig. 8(a-d) ilustra os volumes das áreas de verificações geradas para diferentes valores de  $\tau$  (0,5s, 1,0s, 1,5s e 2,0s, respectivamente). A partir da velocidade máxima de  $v_{max} = 1,5m/s$ , obtida através de testes empíricos no simulador, os lados do volume no plano são definidos a partir da fórmula, enquanto que o lado do eixo z é fixo com valor igual a 1 m. Desta forma, os valores de para os mesmos valores de  $\tau$  são, 0,75m, 1,5, 2,25m e 3m, respectivamente.

O experimento considera o quadricóptero sendo pilotado contra uma parede conforme ilustrado na Fig. 9. Para cada  $\tau$  (0,5s, 1,0s, 1,5s e 2,0s), o movimento foi executado e as distâncias mínimas para o obstáculo foram calculadas.

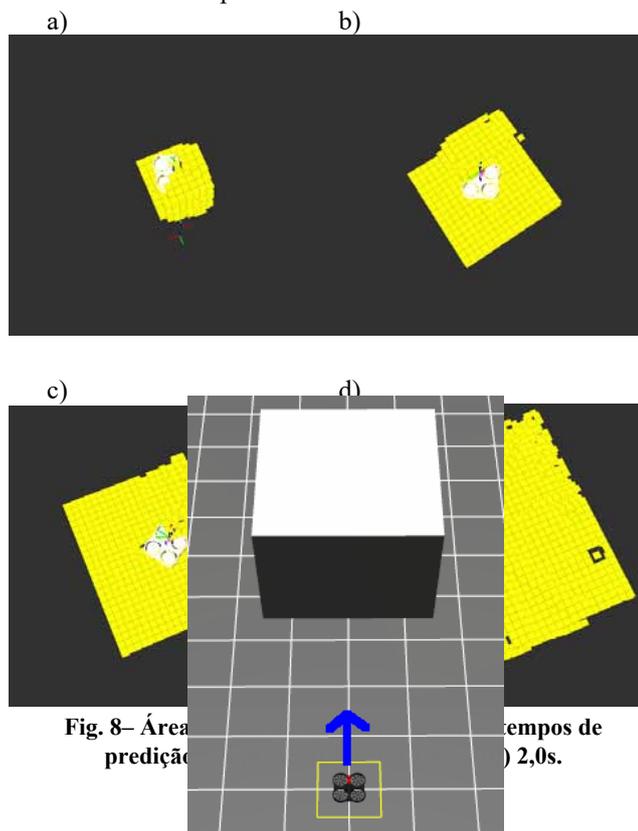


Fig. 8– Área de verificação de colisão para diferentes tempos de predição (a) 0,5s, (b) 1,0s, (c) 1,5s e (d) 2,0s.

Fig. 9– Cenário do quadricóptero pilotado pelo operador contra uma parede.

Tabela 1: Distância de parada do quadricóptero para o obstáculo considerando diferentes tempos de predição.

(s)	Distância para o obstáculo (m)
0,5	0,163925
1,0	0,493201
1,5	0,768959
2,0	1,223990

Tabela 2: Parâmetros de configuração utilizados para os experimentos completos.

PARÂMETRO	VALOR
Resolução da célula Octomap	0,1 m
Tempo de predição ( $\tau$ ) / Tempo para colisão limite	0,5 s
Raio da esfera que representa o quadricóptero	0,3 m

A Tabela 1 ilustra os resultados do experimento. Foi considerada uma margem de segurança, prevenindo possíveis imprecisões dos sensores. Logo,  $\tau = 0,5s$  apresentou um resultado satisfatório, com o veículo parando a uma distância de 0,16 da parede e foi configurado para utilização nos demais testes.

#### 4.2 Experimentos completos

Nesta seção, serão apresentados três experimentos que avaliam a solução em todas as etapas da arquitetura utilizada. Um resumo dos parâmetros de configuração assumidos para o ambiente simulado é listado na Tabela 2. A resolução da célula na malha de ocupação Octomap é de 0,2. O tempo de predição ( $\tau$ ) também determina o tempo para colisão limite. Esse limite vai identificar, baseado na distância para a célula da malha de ocupação e na velocidade do quadricóptero, se existirá ou não uma colisão futura e acionará o controle automático, em caso positivo. Desta forma, ambos os parâmetros são representados pelo mesmo valor de 0,5s. Por último, o raio de 0,3m reflete a distância entre o centro de massa e a extremidade mais distante do quadricóptero modelado no simulador.

##### 4.2.1 Experimento I

Neste experimento, foi utilizado o mesmo movimento realizado no experimento anterior, onde o quadricóptero é pilotado diretamente contra uma parede, conforme ilustrado na Fig. 9. Porém, o intuito aqui é avaliar o acionamento do controle automático quando o veículo se aproxima do obstáculo. A Fig. 10 apresenta três gráficos que ilustram o comportamento do algoritmo durante o tempo da trajetória em questão. É possível verificar o momento em que o controle automático entrou em ação na Fig. 10a. O tempo para colidir é calculado e seu resultado é igual ou superior ao tempo para colisão limite (0,5s) durante quase todo o trajeto. A série em verde (eixo esquerdo) apresenta este comportamento, sendo que só é apresentado um valor diferente do tempo de colisão limite, quando o valor é inferior a este e, nesse momento, o controle automático entra em ação. Desta forma, depois de aproximadamente 26,25s, o tempo para colisão apresenta um valor de 0,11s e o controle automático

é ativado, saindo do valor 0 para o valor 1. Este comportamento é apresentado na série em laranja (eixo direito). O comando automático, por sua vez, permanece ativo durante 1s e retorna o comando para o operador, reassumindo o valor 0, a partir de então.

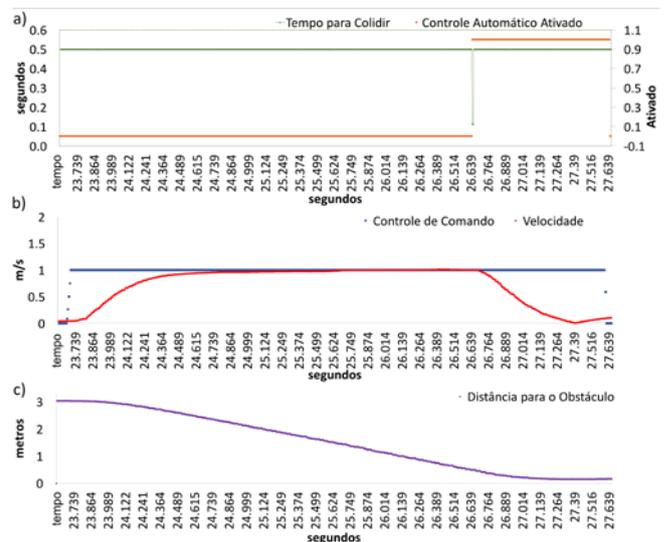
A Fig. 10b, por sua vez, ilustra a comparação entre a velocidade gerada pelo comando de controle do operador (série em azul) e a velocidade efetiva do quadricóptero (série em vermelho). Os valores apresentados no eixo vertical à esquerda do gráfico são referentes às normas dos respectivos vetores velocidade. No início, o quadricóptero sai do repouso e é acelerado até sua velocidade máxima de  $1m/s$  no instante 25s, a partir do comando de controle do operador. No instante 26,65s, é possível notar que, apesar do comando de controle manter sua intensidade e velocidade gerada, a velocidade efetiva começa a cair, não respondendo aos comandos de controle. Nota-se que, confrontando com o gráfico da Fig. 10a, é o momento em que o controle automático entrou em ação, confirmando que este sobrepôs o comando do operador e provocou a parada do quadricóptero. Por fim, a distância para a parede, durante o trajeto, é mostrada na Fig. 10c. A distância decresce com o tempo até o instante 27,15, quando, a partir de então, a distância permanece constante com um valor de aproximadamente 0,16m.

#### 4.2.2 Experimento II

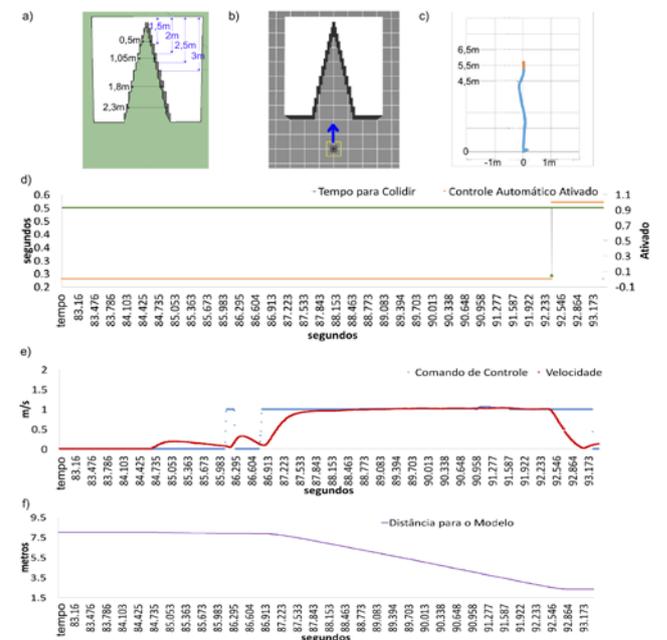
Um segundo experimento foi realizado, avaliando qual a largura mínima de um corredor para que o quadricóptero passe sem ativar o controle automático. Foi projetado, na ferramenta Google SketchUp®, um ambiente onde a largura do corredor vai diminuindo conforme o quadricóptero se aproxima do seu fim. O quadricóptero foi então diretamente pilotado através deste corredor, a fim de verificar a profundidade atingida pelo quadricóptero até o controle automático ser ativado e, conseqüentemente, descobrir a largura mínima. A Fig. 11 ilustra o ambiente simulado. O projeto, com as medidas das referidas larguras, é apresentado na Fig. 11a, enquanto que o movimento executado é mostrado, através de uma seta azul, no ambiente simulado na Fig. 11b.

O deslocamento realizado, de uma vista superior, a partir do comando de controle do operador, é mostrado na Fig. 11c em azul. Em vermelho, é mostrado o trecho do trajeto onde o comando de controle foi sobreposto pelo controle automático.

Gráficos similares aos gráficos apresentados no experimento anterior, que confrontam o tempo para colisão e ativação do controle automático e o comando de controle e a velocidade do quadricóptero, são apresentados nas Fig. 11d e Fig. 11e, respectivamente. Na Fig. 11e, é possível observar que o quadricóptero sai do repouso e sofre, na sequência, uma ligeira variação das velocidades entre os instantes 84s e 86,5s. Esta variação é realizada sob controle manual a fim de manobrá-lo para a execução da trajetória desejada. Isto pode ser verificado confrontando o mesmo instante de tempo com a ligeira variação da trajetória na origem do deslocamento, na Fig. 11c.

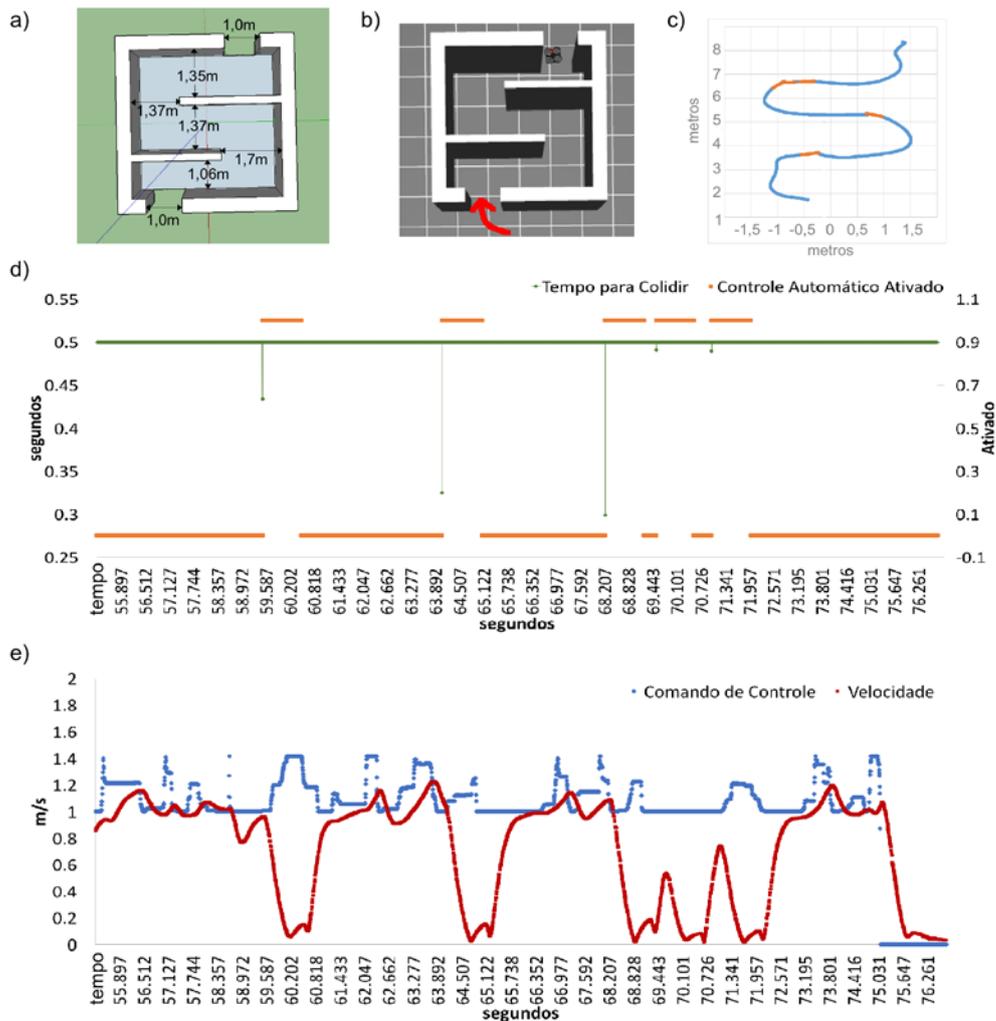


**Fig. 10– Comportamento da solução quando o quadricóptero é pilotado diretamente contra uma parede. (a) Tempo para colidir (em verde) e acionamento do controle automático (em laranja) no tempo. (b) Comando de controle (em azul) e velocidade efetiva (em vermelho) no tempo. (c) Distância para o obstáculo.**



**Fig. 11– Comportamento da solução enquanto o quadricóptero entra em um corredor cada vez mais estreito. (a) Medidas do corredor. (b) Caminho executado. (c) Trajetória percorrida. (d) Tempo para colidir (em verde) e acionamento do controle automático (em laranja) no tempo. (e) Comando de controle (em azul) e velocidade efetiva (em vermelho) no tempo. (f) Distância para o fundo do corredor.**

Um gráfico de distância também é apresentado na Fig. 11f. Porém, é medida a distância do quadricóptero para o fundo do corredor projetado. As distâncias das partes dos corredores de diferentes larguras para o fundo do corredor podem ser visualizadas em azul na Fig. 11a. Na Fig. 11f, é possível observar o decréscimo da distância durante a trajetória até o momento em que o quadricóptero permanece a uma distância constante de 2,31m do fundo do corredor, com o controle automático entrando em ação no instante 92,29s e parando o quadricóptero no instante .



**Fig. 12**– Estado da solução enquanto o quadricóptero percorre o labirinto. (a) Medidas entre paredes. (b) Caminho executado. (c) Trajetória percorrida. (d) Tempo para colidir (em verde) e acionamento do controle automático (em laranja) no tempo. (e) Comando de controle (em azul) e velocidade efetiva (em vermelho) no tempo.

A partir destes resultados, confrontando a Fig. 11f com a Fig. 11a, é possível concluir que o quadricóptero é capaz de percorrer um corredor de 1,05m de largura sem que o controle automático entre em ação. Considerando que o quadricóptero é representado por uma esfera de 0,3m de raio e que ele passa no centro do corredor, uma margem de 0,225m de cada lado foi necessária.

### 4.2.3 Experimento III

Um trajeto completo foi testado neste experimento. A ideia básica é avaliar o auxílio do controle automático durante um deslocamento com diversas mudanças de direções onde colisões são possíveis. Um labirinto foi modelado na ferramenta Google SketchUp® e é exibido, em conjunto com as medidas das distâncias entre suas paredes, na Fig. 12a. A Fig. 12b apresenta o labirinto no simulador, com a seta em vermelho representando o início da trajetória que o quadricóptero percorreu. A trajetória realizada, comandada pelo operador, é mostrada em azul na Fig. 12c. Em vermelho, estão os trechos onde controle automático foi ativado.

Neste experimento, também são apresentados gráficos que confrontam o tempo para colisão e ativação do controle automático e o comando de controle e a velocidade do quadricóptero, através das Fig. 5.13d e Fig. 5.13e, respectiva-

mente. O controle automático entrou em ação em cinco diferentes intervalos (59,5s – 60,5s, 64,1s – 65,1s, 68,2 – 69,2s, 69,6s – 70,6s e 70,9s – 71,9s), como pode ser verificado na série em laranja da Fig. 12d. Isto ocorreu devido às ocorrências de tempos para colisão inferiores à 0,58, fato explicitado na série em verde do mesmo gráfico.

A Fig. 12e, diferente dos outros experimentos, apresenta algumas variações no comando de controle (série em azul), dada a necessidade de mudança das direções durante a trajetória. Além disso, é possível verificar que a velocidade do quadricóptero (em vermelho) decaiu durante os intervalos, 59,5s – 60,5s, 64,1s – 65,1s, 68,2s – 69,2s, 69,6s – 70,6s e 70,9s – 71,9s independentemente do comando de controle do operador. Este fato confirma o acionamento do controle automático.

**Tabela 3:** Distância mínima para células nos intervalos em que o controle automático foi ativado.

Intervalo(s)	Distância mínima para o obstáculo (m)
59,5s – 60,5s	0,417316
64,1s – 65,1s	0,341586
68,2s – 69,2s	0,325506
69,6s – 70,6s	0,201657
70,9s – 71,9s	0,335225

Por fim, faltava verificar que o controle automático não permitiu a colisão do quadricóptero durante toda a trajetória. Para isso, a distância para a célula mais próxima foi calculada nos momentos em que o controle automático entrou em ação. Este valor é a distância mínima calculada para determinar o tempo de colisão. A Tabela 3 apresenta os cinco intervalos em que o controle automático estava ativo e suas respectivas distâncias mínimas calculadas. Considerando toda a trajetória, é possível observar que a distância mínima para as paredes do labirinto foi de 0,20m para o intervalo 69,6s – 70,6s, o que confirma a não existência de colisões no percurso.

A análise aqui apresentada demonstra que a solução foi capaz de evitar a colisão enquanto o quadricóptero percorre um labirinto. Observando a Fig. 12c, é possível verificar que os intervalos onde o controle automático foi ativado são instantes de correção de trajetória que, normalmente, são mais propícios a colisões.

## 5. CONCLUSÃO

Neste trabalho, foi elaborada uma solução embarcada para assistência ao controle manual de um quadricóptero, sendo capaz de impedir a colisão do veículo contra obstáculos em tempo real, considerando um ambiente totalmente desconhecido. Uma arquitetura foi proposta, baseada no cálculo simultâneo da localização e mapeamento dos arredores do quadricóptero (SLAM), durante o sobrevoo. A trajetória futura do veículo é constantemente estimada enquanto a malha de ocupação é construída a partir de três sonares embarcados apontados para a frente do quadricóptero, que obtêm as distâncias para os obstáculos dos seus arredores. O *Octomap* foi utilizado para constituir a malha de ocupação utilizando uma abordagem probabilística, endereçando a incerteza relacionada à imprecisão das medidas obtidas dos sonares. Além disso, o *Octomap* apresenta características que permitem sua utilização em computadores com capacidade de processamento limitada, como permitir diferentes resoluções da *Octree*, compactação de armazenamento e rapidez de atualização. Um controle automático PID foi elaborado para sobrepor o controle do operador pelo intervalo de tempo de e manter a posição do quadricóptero, evitando a colisão iminente.

Os três experimentos realizados mostraram que a arquitetura proposta obteve sucesso em evitar colisão do quadricóptero contra obstáculos em ambiente simulado e em tempo real. A utilização do *Robot Operating System* (ROS) em conjunto com o Gazebo e TUM *Simulator* proveram uma infraestrutura importante para a construção e validação da solução proposta, através da simulação do quadricóptero e diversos ambientes *indoor*, com diferentes cenários e obstáculos, para navegação do quadricóptero.

O algoritmo desenvolvido é configurável. Através da variação dos parâmetros de configuração ilustrados na Tabela 2, a solução pode ter seu comportamento modificado, permitindo o quadricóptero sobrevoar mais próximo ou distante dos obstáculos, de acordo com o nível de conservadorismo desejado. Além disso, o algoritmo apresenta um tempo para a verificação de colisão menor que 0,005s, valor correspondente a uma frequência de 200Hz, o que afirma sua capacidade de processamento em tempo

real e a baixa necessidade de recursos computacionais. Desta forma, a solução atendeu aos requisitos propostos em simulação, evitando a colisão com os obstáculos.

Como trabalho futuro, a solução será embarcada em um quadricóptero real controlado por um mini-PC (Raspberry PI®) e novos experimentos serão realizados. Também, serão adicionados sensores de medição de distância ópticos (LIDAR, sigla para o termo em inglês *Light Detection and Ranging*).

## REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Thrun, S.; Burgard, W.; Fox, D.; Probabilistic robotics, MIT Press, 2005.
- [2] Bristeau, P.-J.; Callou, F.; Vissiere, D.; Petit, N. et al; The navigation and control technology inside the Ar.Drone micro UAV; 18th IFAC World Congress, volume 18, págs. 1477–1484, 2011.
- [3] Sturm, J.; Gazebo Simulator for the Parrot AR.Drone quadcopter, 2012. [http://wiki.ros.org/tum\\_simulator](http://wiki.ros.org/tum_simulator), acessado em agosto de 2016.
- [4] Ross, S.; Melik-Barkhudarov, N.; Shankar, K. S.; Wendel, A.; Dey, D.; Bagnell, J. A.; Hebert, M.; Learning monocular reactive UAV control in cluttered natural environments; Robotics and Automation (ICRA), 2013 IEEE International Conference on, págs. 1765–1772, IEEE, 2013.
- [5] Ryll, M.; Bulthoff, H. H.; Giordano, P. R.; A novel overactuated quadrotor unmanned aerial vehicle: modeling, control, and experimental validation; Control Systems Technology, IEEE Transactions on, 23(2):540–556, 2015.
- [6] Zhang, X.; Li, X.; Wang, K.; Lu, Y.; A Survey of Modelling and Identification of Quadrotor Robot; Abstract and Applied Analysis, volume 2014, Hindawi Publishing Corporation, 2014.
- [7] Salih, A. L.; Moghavvemi, M.; Mohamed, H. A.; Gaeid, K. S.; Flight PID controller design for a UAV quadrotor; Scientific Research Essays, 5(23): 3660–3667, 2010.
- [8] Nicol, C.; Macnab, C.; Ramirez-Serrano; A Robust neural network control of a quadrotor helicopter; Proceedings of the Canadian Conference on Electrical and Computer Engineering, págs. 1233–1237, 2008.
- [9] Ito, K.; Han, J.; Ohya, A.; Localization using uniaxial laser range finder and IMU for MAV; System Integration (SI), 2014 IEEE/SICE International Symposium on, págs. 712–717, IEEE, 2014.
- [10] Shen, S.; Mulgaonkar, Y.; Michael, N.; Kumar, V.; Vision-based state estimation for autonomous rotorcraft MAVs in complex environments; Robotics and Automation (ICRA), 2013 IEEE International Conference on, págs. 1758–1764, IEEE, 2013.
- [11] Mendes, J.; Ventura, R.; Assisted teleoperation of quadcopters using obstacle avoidance; Journal of Automation Mobile Robotics and Intelligent Systems, 7(1):54–58, 2013.
- [12] Grzonka, S.; Grisetti, G.; Burgard, W.; A fully autonomous indoor quadrotor; Robotics, IEEE Transactions on, 28(1):90–100, 2012.
- [13] Kerns, A. J.; Shepard, D. P.; Bhatti, J. A.; Humphreys, T. E.; Unmanned aircraft capture and control via GPS spoofing; Journal of Field Robotics, 31 (4):617–636, 2014.
- [14] Roberts, A.; Tayebi, A.; A new position regulation strategy for VTOL UAVs using IMU and GPS measurements; Automatica, 49(2):434–440, 2013.
- [15] Nieuwenhuisen, M.; Droschel, D.; Schneider, J.; Holz, D.; Labe, T.; Behnke, S.; Multimodal obstacle detection and collision avoidance for micro aerial vehicles; Mobile Robots (ECMR), 2013 European Conference on, págs. 7–12. IEEE, 2013.
- [16] Saranritichai, P.; Niparnan, N.; Sudsang, A.; Robust local obstacle avoidance for mobile robot based on Dynamic Window approach; Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2013 10th International Conference on, págs. 1–4, IEEE, 2013.
- [17] Claes, D.; Hennes, D.; Tuyls, K.; Meeussen, W.; Collision avoidance under bounded localization uncertainty; Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on, págs. 1192–1198. IEEE, 2012.

- [18] Van Den Berg, J.; Wilkie, D.; Guy, S. J.; Niethammer, M.; Manocha, D.; LQG-obstacles: feedback control with collision avoidance for mobile robots with motion and sensing uncertainty; *Robotics and Automation (ICRA)*, 2012 IEEE International Conference on, págs. 346–353, IEEE, 2012.
- [19] Fraichard, T.; Asama, H.; Inevitable collision states - a step toward safer robots?; *Advanced Robotics*, 18(10):1001–1024, 2004.
- [20] Israelsen, J.; Beall, M.; Bareiss, D.; Stuart, D.; Keeney, E.; Van Den Berg, J.; Automatic collision avoidance for manually tele-operated unmanned aerial vehicles; *Robotics and Automation (ICRA)*, 2014 IEEE International Conference on, págs. 6638–6643, 2014.
- [21] Hornung, A.; Wurm, K. M.; Bennewitz, M.; Stachniss, C.; Burgard, W.; OctoMap: an efficient probabilistic 3D mapping framework based on octrees; *Autonomous Robots*, 34(3):189–206, 2013.
- [22] Salih, A. L.; Moghavvemi, M.; Mohamed, H. A.; Gaeid, K. S.; Flight PID controller design for a UAV quadrotor; *Scientific Research and Essays*, 5(23): 3660–3667, 2010.
- [23] Carrillo, L. R. G.; López, A. E. D.; Lozano, R.; Pégard, C.; *Quadrotorcraft control: vision-based hovering and navigation*; Springer Science & Business Media, 2012.
- [24] Beard, R. W.; McLain, T. W.; *Small unmanned aircraft: theory and practice*; Princeton University Press, Princeton, 2012.
- [25] Latombe, J.-C.; *Robot motion planning*, volume 124, Springer Science & Business Media, 1991.
- [26] Lugo, J. J.; Zell, A.; Framework for autonomous on-board navigation with the AR. Drone; *Journal of Intelligent & Robotic Systems*, 73(1-4):401–412, 2014.