

Desenvolvendo Simulações de Uso de Drones em Cidades Inteligentes

Lucas Baptista de Moraes
Dep. de Engenharia de Telecomunicações
Universidade Federal Fluminense
Niterói, Brasil
lucasbaptista@midia.com.uff.br

Hullysses Sabino
Instituto de Geociências
Universidade Federal Fluminense
Niterói, Brasil
hscontact21@gmail.com

Bruno Hilario
Instituto de Computação
Universidade Federal Fluminense
Niterói, Brasil
brhilario@gmail.com

Natalia Castro Fernandes
Dep. de Engenharia de Telecomunicações
Universidade Federal Fluminense
Niterói, Brasil
nataliacf@id.uff.br

Fernanda G. O. Passos
Dep. de Engenharia de Telecomunicações
Universidade Federal Fluminense
Niterói, Brasil
fernanda@midia.com.uff.br

Resumo—Dada a dificuldade de planejamento de missões para voos de drones nas cidades, devido ao ambiente e aos recursos limitados da aeronave, torna-se necessária a utilização de tecnologias de ambiente de simulação para treinamento e preparo para missões de campo. Com isso, este trabalho tem como objetivo descrever e analisar uma metodologia de desenvolvimento de simulação de voo de drones para o planejamento de missões de campo com um cenário verossímil comparado à realidade em termos de tempo e distância. Ao utilizar os simuladores, foi possível obter um tempo médio de 97,62% de precisão nos testes do ambiente virtual em relação aos testes de campo, apresentando uma verossimilhança alta obtida através dos ajustes realizados entre o Modelo Digital de Superfície e calibração dos sistemas de coordenadas utilizado.

Index Terms—Simuladores de voo de drones, Gazebo, SITL, ArduPilot, Cidades Inteligentes

I. INTRODUÇÃO

Um veículo aéreo não tripulado (VANT), mais conhecido como *drone*, é um equipamento que pode ser operado por pessoas remotamente ou voar autonomamente [1]. Tais veículos podem ser utilizados em diversas áreas e aplicações no contexto de cidades inteligentes e integradas, como: serviços emergenciais e gerenciamento de desastres, monitoramento de tráfego e público, computação de borda e extensão de conectividade de redes [2].

Drones comerciais, e também profissionais, possuem recursos limitados [3] e estão suscetíveis a possíveis acidentes [4], principalmente se operados remotamente. Por isso, é importante que cada missão, independente de sua aplicação, seja bem planejada, em termos de distância a ser percorrida, pontos de paradas chave, tempo de voo, e outras questões importantes para a operação. Portanto, é necessário que existam metodologias e ferramentas que envolvam o planejamento de missões, que simulem e forneçam dados o mais próximo da realidade, evitando possíveis não-conformidades.

Este trabalho é apoiado pela parceria entre Fundação Euclides Cunha (FEC) e Prefeitura de Niterói.

Este trabalho analisa o desenvolvimento de modelos de simulação para o planejamento de missões e operações que utilizem drones. Seu objetivo é prover um ambiente simulado o mais próximo o possível da realidade de uma cidade, em termos de escala e dimensão, com base em ferramentas de simulação de voo de drones já existentes. Tais simulações podem auxiliar na implementação de VANTs no contexto de cidades inteligentes, cada vez mais importante conforme o avanço das tecnologias de borda e da quantidade de dispositivos crescentes conectados à rede.

Analisou-se os principais simuladores existentes e suas operações básicas, bem como o processo de criação e modelagem de um ambiente 3D da região em que se encontra a Universidade Federal Fluminense (UFF). Foram realizados testes com drones reais na mesma área, permitindo avaliar a verossimilhança do ambiente simulado em relação ao ambiente de campo. Com isso, foi possível observar em que pontos a simulação é uma abordagem segura para estudos preliminares e preparação para voos de campo, que são custosos e propícios à acidentes, caso mal planejados.

O presente artigo está organizado como descrito a seguir. Na Seção II, são apresentados os principais trabalhos relacionados. Na sequência, a Seção III descreve os principais simuladores de voos de drones, enquanto que a Seção IV demonstra a metodologia utilizada na análise. A Seção V discute os resultados e a Seção VI enumera os desafios encontrados ao longo do desenvolvimento do estudo. Por fim, a Seção VII conclui o artigo.

II. TRABALHOS RELACIONADOS

O uso de drones e outros robôs em ambientes de simulação tem sido amplamente estudado na literatura, onde são encontradas diversas abordagens, como: compreender o entendimento do uso de sensores, operações básicas de missões e comunicação entre enxames de drone [5]–[9].

Bernardeschi *et al.* [10] utilizaram o simulador Gazebo e seus modelos de VANTs virtuais com *Robot Operating System* (ROS) — uma coleção de bibliotecas e *frameworks* para desenvolvimento e programação de robôs — com o objetivo de coordená-los cooperativamente através de comunicações sem fio entre drones. Os autores implementaram um código, através da biblioteca Python Dronekit, onde o algoritmo inicia a instância simulada de cada aeronave e sua localização é transmitida para os VANTs mais próximos na esquerda e também na direita. Com isso, a movimentação de cada aeronave é coordenada através de parâmetros pré-estabelecidos pelos autores, e a distância entre cada drone é calculada através da fórmula trigonométrica de Harvesine.

Abbyasov *et al.* [11] utilizaram também o Gazebo, entretanto com uma perspectiva de criar cenários semelhantes ao de terrenos reais, para que os testes com diferentes tipos de robôs em solo possam ser realizados em condições próximas do que seria na vida real. Neste caso, foi desenvolvida uma ferramenta de criação de mundos tridimensionais através de uma imagem de escala cinzenta bidimensional, com um fator considerado aceitável de tempo-real em diversos pontos do ambiente gerado. A métrica de desempenho fator de tempo-real é a razão entre o tempo de execução real de uma determinada tarefa em relação ao seu tempo de simulação. Por exemplo, se uma tarefa de 1 segundo em tempo-real leva 4 segundos para ser executada em tempo de simulação, logo o fator de tempo-real é equivalente à 0,25.

Sánchez *et al.* [12] desenvolveram um dataset sintético público utilizando o Gazebo para o movimento de robôs em solo em diferentes ambientes naturais. Os dados gerados incluem uma visão 3D disponibilizada como arquivos ROS, os quais podem ser usados para estudos com aprendizado supervisionado e *benchmarkings*.

Farley *et al.* fizeram um estudo comparando resultados simulados para robôs de solo utilizando os simuladores CoppeliaSim, Gazebo, MORSE e Webots [13]. Os autores comparam o resultado dessa simulação com testes práticos, realizados dentro do ambiente do laboratório. Cabe destaque que esse teste ocorre em um ambiente controlado e pequeno, com modelagem simples. Os autores focaram na análise da acurácia do movimento do robô, obtendo como resultados que o CoppeliaSim e o Gazebo tiveram um melhor desempenho com relação aos resultados reais.

Apesar de diversos estudos tentarem replicar condições de um cenário real e também discorrerem sobre possíveis sensores e comunicação entre drones, não foi encontrado na literatura um trabalho que apresentasse uma metodologia de simulação de VANTs em ambientes mais realistas, onde o ambiente virtual seja idêntico ou próximo o suficiente da arquitetura do cenário, como pequenas porções de uma localidade, como uma cidade ou bairro, e que possibilite o planejamento de missões com VANTs não-simulados.

III. SIMULADORES

Muitos aspectos podem ser estudados referente às características e funções que um drone pode desempenhar, como:

dinâmica de voo, avaliações físicas (*e.g.*, peso e tamanho), redes e comunicação, e voo automático através de aprendizado de máquina. Dada a complexidade de cada aspecto, existe uma vasta quantidade de simuladores que endereçam mais especificamente cada uma dessas características [14]. Esta seção tem como objetivo listar alguns dos simuladores mais utilizados na literatura, com suas funcionalidades brevemente descritas.

A. ArduPilot e Software in the Loop

O ArduPilot [15] é um *software* livre utilizado em placas controladoras de veículos aéreos. Ele pode ser utilizado em diversos tipos de aeronaves não tripuladas, como helicópteros, multicópteros e aeronaves de asa-fixa. O ArduPilot é capaz de ser simulado através do *Software in the Loop* (SITL) [16], sem ter a necessidade de nenhum *hardware* de controle específico. O SITL permite habilitar a conexão com *softwares* de estação-base para o controle do drone, como o MAVProxy, possibilitando obter dados de estado do drone (telemetria), enviar comandos de voo e criar as missões necessárias em cenários virtuais simples. O SITL possui também integração com diversos outros simuladores (*e.g.*, Gazebo, *AirSim*, *Real-Flight*, *MORSE* e *Webots*), o que permite gerar cenários mais avançados.

Com o ArduPilot simulado através do SITL, é possível realizar diversas configurações do voo de drones, como gerar arquivos de histórico, alterar velocidade do veículo, definir modos de voos (*e.g.* guiado, aterrissagem) e criar missões autônomas, *i.e.*, sem intervenção humana.

B. Gazebo

O Gazebo [17] é um simulador multifuncional e permite que o usuário possa testar cenários realísticos em relação à dinâmica e física de robôs e veículos, e também possibilita a configuração e alteração do cenário da simulação, adaptando-o conforme for necessário. O Gazebo é um simulador amplamente discutido e utilizado pela comunidade científica, e possui também diversas bibliotecas e integrações com outros softwares que enriquecem a simulação, como o próprio SITL [10]. Sua interface permite criar, de modo simples, novos robôs, sensores, objetos de diversas formas e tamanhos.

A arquitetura do Gazebo começa pelo Mundo, uma coleção de modelos, sensores e objetos arbitrários para criação de um cenário. Um Modelo, por sua vez, pode ser um simples objeto, como uma caixa, ou um robô com diversos graus de complexidade, contendo: corpo (dimensões e geometria do objeto), sensores (responsáveis por inúmeras funcionalidades como desvio de obstáculo, câmeras para foto e vídeo, câmeras de profundidade) e articulações (responsável por criar dinamismo entre os objetos que compõe um corpo) [18].

C. AirSim

O *AirSim* [19] [20] foi criado pela *Microsoft Research* com o objetivo de ser uma plataforma de simulação para pesquisa e experimentos de inteligência artificial e *deep learning*. Em

termos gráficos e visuais, é um simulador completo, com ambientes criados em *softwares* utilizados para desenvolvimentos de jogos de nova geração, como *Unity* e *Unreal Engine*. O drone pode ser controlado através de *scripts* criados em diversas linguagens de programação, como C#, Python, C++. Este simulador também possui integração com o SITL.

Sua arquitetura é composta pelo modelo de meio-ambiente, modelo do veículo, mecanismos de física, modelos de sensores e uma interface de renderização. Seu ambiente pode ter parâmetros como: gravidade, campo magnético, pressão e densidade do ar. Sua renderização visual realística traz detalhes importantes para simulação, como posicionamento de luz, sombras de objetos e prédios, sendo um dos melhores simuladores nesse quesito [21].

D. RealFlight

O *RealFlight* [22] é um dos simuladores mais populares entre pilotos de aeronaves em geral, não apenas de drones, devido à sua alta confiabilidade [14]. Contendo diversos modelos de drones reais em seu ambiente simulado, além de ambientes diversos para voos livres, importante ferramenta para operadores em treinamento, e desafios para aprimoramento de técnicas voo.

IV. METODOLOGIA DE DESENVOLVIMENTO

Neste trabalho, foca-se em desenvolver um ambiente simulado na cidade de Niterói, região metropolitana do estado do Rio de Janeiro, onde localiza-se a Universidade Federal Fluminense (UFF). Para fins acadêmicos, foi decidido realizar a simulação do *campus* da Praia Vermelha da UFF e seus arredores, que abrange a região de Niterói chamada de Praias da Baía.

Para análise do ambiente virtual desenvolvido, foi necessário realizar voos com um drone em um ambiente real. Os voos experimentais foram realizados na mesma região e com drone real de configurações suficientemente similares ao virtual.

A. Ambiente Virtual

Dado o objetivo do estudo, foram escolhidos os simuladores SITL e Gazebo para serem utilizados, uma vez que oferecem as funções básicas para o planejamento de missões, como definição de *waypoints*, física de colisão, ajustes de velocidade da aeronave e vento, sem a necessidade de entrar em questões mais complexas como de inteligência artificial e outras funções que simuladores citados na Seção III oferecem.

O primeiro passo da execução do ambiente virtual é carregar o cenário da simulação, com o modelo de mundo desenvolvido na Seção IV-A1, no Gazebo. Em seguida, inicia-se a execução do SITL, que ficará responsável por inicializar a simulação da controladora de voo do drone. Por fim, um terminal MAVProxy — estação-base de controle — é carregado, onde todas as comunicações com o drone e controles de voos serão realizados. A Figura 1 ilustra esta relação entre os simuladores e seus principais componentes.

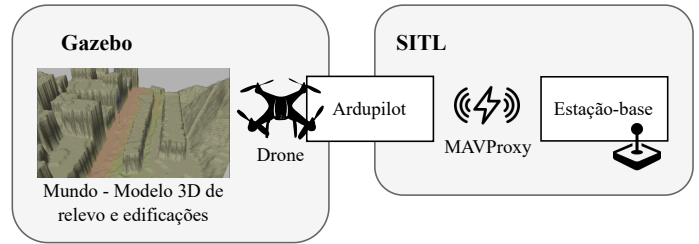


Figura 1: Diagrama do ambiente simulado proposto com Gazebo e SITL.

Ambos os *softwares* (Gazebo e SITL) são utilizados simultaneamente, sendo configurados através de um *plugin*, que para cada modelo de VANT simulado no SITL, uma instância de *plugin* é carregada no processo do Gazebo. Por sua vez, tal instância utiliza interfaces de programação (API) internas para obter dados de estado do veículo, que serão enviados para um processo externo do ArduPilot — simulado pelo SITL — através de um protocolo de rede UDP próprio customizado, chamado de *Flight Dynamics Model* (FDM). Estes dados, por sua vez, são enviados pelo processo externo para uma estação-base de controle, através do protocolo MAVLink. Dessa forma, todas as ações e comandos executados na estação-base serão refletidos simultaneamente no Gazebo.

Com os simuladores devidamente escolhidos, foi realizado o desenvolvimento de um modelo digital de superfície da região selecionada para estudo, através do uso de ferramentas de modelagem de curvas de nível (incluindo edificações) e georreferenciamento. Em seguida, foram necessários alguns ajustes e configurações de sistemas de coordenadas e orientação do drone no SITL em relação ao ambiente virtual criado.

1) *Desenvolvendo Ambiente Simulado*: O *software* QGIS [23], na versão 3.10, foi o sistema de informação geográfica de código livre e aberto utilizado neste trabalho para o desenvolvimento da modelagem tridimensional (3D) da região predefinida da cidade de Niterói. Para diminuir o escopo de estudo, os dados foram recortados para apenas a região de interesse, correspondente à Praia Vermelha onde encontra-se a Escola de Engenharia da UFF. A partir deste recorte, foi gerado um *raster* no formato TIFF (*Tagged Image File Format*) que contém informações de altitude do terreno com alta resolução espacial de meio metro — sem considerar outros aspectos como vegetação, construções e edifícios.

Uma vez que é importante considerar o voo de drones a altitudes baixas nas missões da cidade, foi identificado que a geometria das edificações seria relevante para tornar a modelagem mais realista. Por este motivo, foi considerado também o *shapefile* dos lotes das construções do município de Niterói disponíveis no portal *online* do Sistema de Gestão da Geoinformação da Prefeitura de Niterói (SIGeo) [24]. As construções dentro do *campus* da UFF foram delimitadas a partir do processo de vetorização utilizando como base as imagens do referido *Google Satellite* [25]. Como no *shapefile* dos lotes não havia informação da altimetria das construções,

foram inseridas neste arquivo 8 alturas (5, 10, 15, 20, 25, 30, 36, 45 m) em cada feição a partir do QGIS. Por exemplo, os lotes referentes aos quiosques da orla da baía da região receberam valores de 5 m; lotes referentes às casas foram editados e receberam valores baixos, enquanto que os prédios receberam valores mais altos entre as alturas consideradas. Os prédios dos institutos de ensino do *campus* da Praia Vermelha da UFF receberam a altura de 15 metros, considerando 5 andares com pé direito de 3 metros cada um. O *shapefile* das edificações foi convertido para o formato *TIFF*.

Para unir as camadas de curvas de nível do relevo e dos lotes das edificações produzidas conforme descrito nos últimos parágrafos, foi utilizada a ferramenta Mesclar do QGIS. Assim, foi produzido um Modelo Digital de Superfície (MDS), em formato *TIFF*, que considera a superfície geográfica natural (relevo) acrescido de outros elementos, no caso, edificações. O resultado final deste modelo 3D da região selecionada pode ser visto na Figura 2. Note, entretanto, que este modelo é aproximado, não correspondendo fielmente às medidas reais dos edifícios. Este modelo será carregado pelo Gazebo.

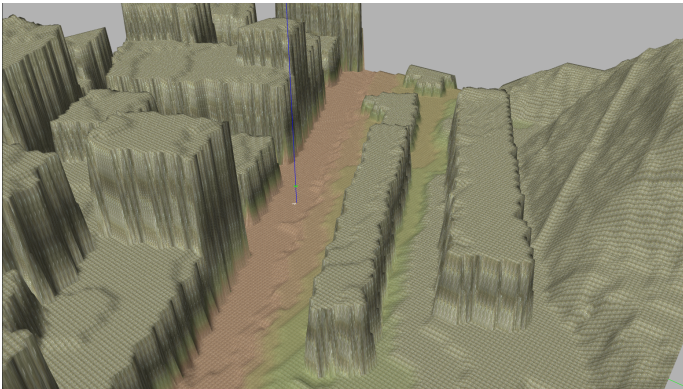


Figura 2: Modelo 3D dos arredores da Escola da Engenharia na UFF para o ambiente simulado utilizado no Gazebo.

2) *Configuração e Preparação dos Simuladores*: As configurações e instalações necessárias para todos os *softwares* de simulação foram realizadas em um computador com sistema operacional Linux Ubuntu 18.04 LTS, composto por um processador AMD Ryzen 7 2700 e 16 GB de memória principal. O primeiro passo realizado foi instalar o sistema do ArduPilot, em sua versão mais recente, simulado através do SITL. Em seguida, foi instalado o Gazebo versão 9 e o *plugin* para a integração com o ArduPilot, também em sua versão mais recente, detalhado anteriormente.

Após todos os *softwares* serem adequadamente instalados, é necessário realizar a criação do mundo que será simulado graficamente através do Gazebo, utilizando o arquivo *.tiff*, gerado através do modelo que apresenta as curvas de nível de relevo e edificações (com valores aproximados) da região, como descrito na Seção IV-A1.

Outro ponto importante de se destacar é a localização do drone referente ao SITL durante sua inicialização. Ao

ser executado, deve-se informar ao SITL manualmente as coordenadas exatas do centroide referente ao modelo em *TIFF* e as coordenadas do local onde o drone será posicionado. Sendo assim, o drone simulado pelo SITL estará exatamente posicionado conforme na localização do Gazebo. Outro ajuste relevante é a mudança de *tag* de visualização de coordenadas, que consiste em um parâmetro de configuração de modelo de mundo, uma vez que o Gazebo utiliza uma visão no sentido East-North-Up (ENU), enquanto o SITL utiliza o North-East-Down (NED). Portanto, é necessário rotacionar o parâmetro *yaw* (em torno do eixo Z do sistema cartesiano de três coordenadas) em 90 graus radianos nos parâmetros de configuração do ArduPilot, obtendo assim um alinhamento correto entre o visualizado no Gazebo e o drone simulado no SITL.

Vale ressaltar que o modelo de drone utilizado pelo Gazebo é o 3DR Iris, amplamente utilizado para estudos e pesquisas acadêmicas. É possível visualizar sua versão simulada na Figura 3. A aeronave escolhida para simulação não é igual ao drone utilizado no ambiente real, descrito posteriormente na Seção IV-B, pela falta de tal modelo disponível nas bibliotecas. Entretanto, entende-se que o 3DR Iris é semelhante o suficiente.

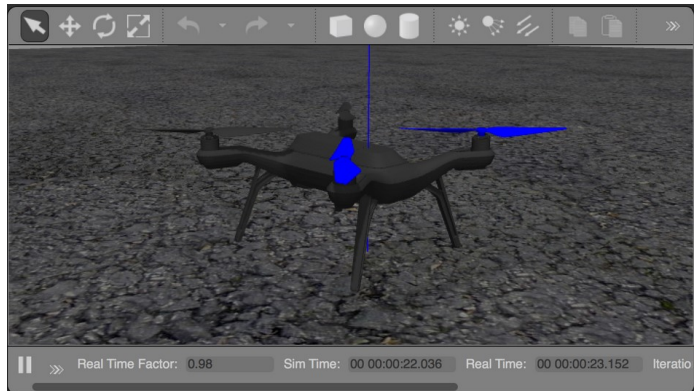


Figura 3: Modelo simulado no Gazebo do drone 3DR Iris.

Outro fator importante para o ambiente simulado, é o fato da missão ter sido criada através da estação-base de controle MAVProxy, como normalmente é feito pelos usuários de drones. Através dela, todos os *waypoints* podem ser definidos. Um exemplo será descrito a seguir, na Seção V.

B. Ambiente Real

Os testes de campo foram realizados nos arredores da Escola de Engenharia da UFF, mais especificamente ao redor do bloco, em um dia ensolarado de temperatura média de 28°C , com uma velocidade de vento baixa.

O drone utilizado é o modelo DJI Air 2S, da empresa *DJI Innovations* (DJI), uma aeronave leve de 595 g, que possui 4 hélices, com uma autonomia de voo de aproximadamente 31 minutos e uma distância máxima, referente ao piloto, de 18,5 km na horizontal e 5 km de altura. É possível visualizar o VANT na Figura 5. Este drone é bem semelhante

ao simulado no ambiente virtual e apresentado na Figura 3, ao qual a aeronave real possui 400 g, 4 hélices, e com tempo de voo aproximado de 16 a 22 minutos.

O aplicativo Litchi [26] (versão 4.25.0-g) foi utilizado para o planejamento da missão, uma vez que o aplicativo padrão da DJI, DJI Fly, não possui a funcionalidade de marcar *waypoints* nativamente, impossibilitando a criação de um voo automático. O Litchi pode ser utilizado em celulares Android e iOS, e fornece uma versão *web*, chamada de *Mission Hub*, que possui integração com os aplicativos móveis e permite também a criação de missões, que podem ser exportadas para diversos tipos de arquivos, como *.csv* e *.xlsx*.

V. RESULTADOS E ANÁLISE

Para os experimentos realizados neste artigo, foram preparados dois ambientes de testes: o virtual e o real. Seu objetivo é entender o quão verossímil é o ambiente simulado de voo de drones proposto em relação a uma missão real em termos de tempo, distância e bateria da aeronave. Estas propriedades são alguns dos pontos-chaves para que o planejamento de missões através de um ambiente virtual seja confiável.

A. Missão de Voo

Para a missão de voo, em ambos os ambientes, foram gerados 13 *waypoints* (pontos de parada) idênticos para o virtual e real, todos em uma altura de 35 metros. É possível ver os pontos de parada na Figura 4. Uma vez criada a missão, o drone irá voar automaticamente para cada um dos pontos, sem a necessidade de intervenção humana. O objetivo é analisar o tempo decorrido durante um percurso de voo do drone no ambiente simulado em comparação ao ambiente real.

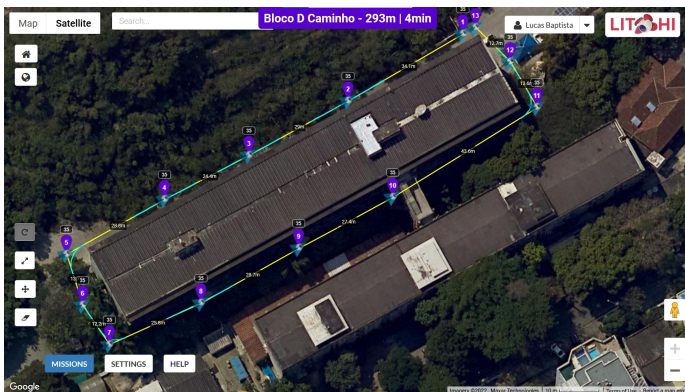


Figura 4: Editor de missões (*Mission Hub*) do aplicativo Litchi, com os 13 *waypoints* utilizados nos testes.

Para comparações, foram escolhidos 3 cenários diferentes de velocidade do drone, nos valores de 1,5, 2,0 e 2,5 metros por segundo. Foram realizadas 6 repetições de execução das missões no ambiente real (em um mesmo dia) para cada uma das velocidades, a fim de estabelecer uma média e desvio padrão e intervalo de confiança. Já para o ambiente simulado, não foram encontradas variações temporais, por se tratar de um ambiente determinístico, portanto foi levada em consideração apenas uma execução. A Tabela I apresenta as médias, com



Figura 5: Drone DJI Air 2S, modelo de drone utilizado ao longo dos testes de campo¹.

seus respectivos intervalos de confiança de 95%, do tempo total de execução para o ambiente real (campo) para chegar em cada um dos 13 *waypoints* (valor acumulado). Ela também apresenta os tempos medidos para cada *waypoint* utilizando o ambiente simulado.

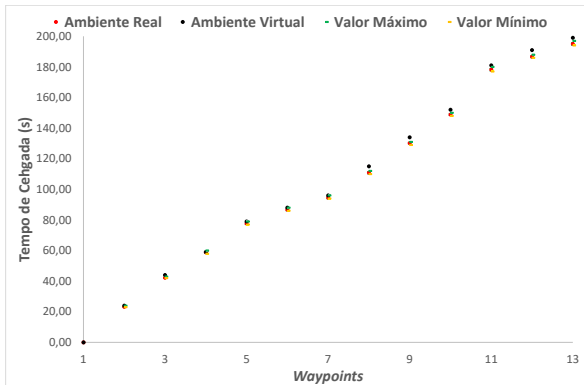
Tabela I: Visão geral de tempo de chegada em cada *waypoint* nos ambiente real e virtual (campo e simulador).

Waypoint	Tempo de Voo (s)					
	1.5 m/s		2.0 m/s		2.5 m/s	
	Campo	Sim.	Campo	Sim.	Campo	Sim.
1	0	0	0	0	0	0
2	23,17 (±0, 33)	24	17,00 (±0, 00)	18	13,83 (±0, 33)	14
3	42,17 (±0, 33)	44	31,67 (±0, 41)	32	25,33 (±0, 41)	26
4	58,83 (±0, 60)	59	43,83 (±0, 33)	45	35,33 (±0, 41)	36
5	77,67 (±0, 33)	79	58,00 (±0, 00)	59	46,67 (±0, 33)	47
6	86,67 (±0, 65)	88	64,83 (±0, 33)	66	52,17 (±0, 33)	53
7	94,50 (±0, 67)	96	70,67 (±0, 41)	73	57,17 (±0, 33)	59
8	110,83 (±0, 79)	115	83,00 (±0, 00)	86	67,33 (±0, 41)	69
9	130,17 (±0, 60)	134	97,17 (±0, 33)	101	78,83 (±0, 33)	81
10	148,83 (±0, 60)	152	111,17 (±0, 33)	114	89,83 (±0, 33)	92
11	178,17 (±0, 79)	181	133,17 (±0, 33)	136	107,33 (±0, 41)	109
12	186,67 (±0, 65)	191	139,50 (±0, 67)	144	112,67 (±0, 41)	116
13	195,00 (±0, 88)	199	145,83 (±0, 60)	150	118,17 (±0, 33)	121

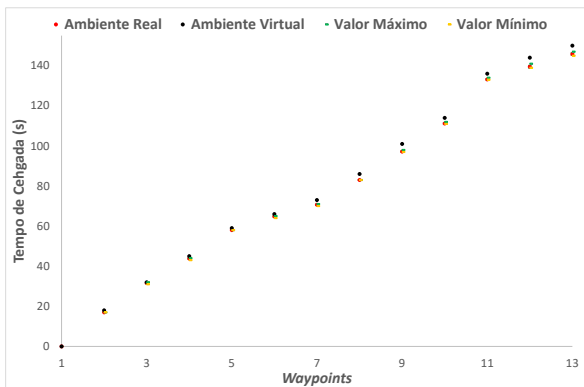
É possível visualizar nas Figuras 6a, 6b e 6c, todos os pontos medidos e seus valores máximos e mínimos, para os testes de ambiente real. O intervalo de confiança também foi inserido como barra de erro, mas seu valor é tão pequeno referente às amostras, que mostra-se imperceptível nos gráficos. É possível notar que todos os pontos, em cada cenário, são bem próximos, apontando grandes semelhanças temporais em relação aos dois ambientes, independente do aumento de velocidade.

De acordo com a Figura 7, é possível ter um maior detalhamento entre as diferenças de tempo de chegada de cada

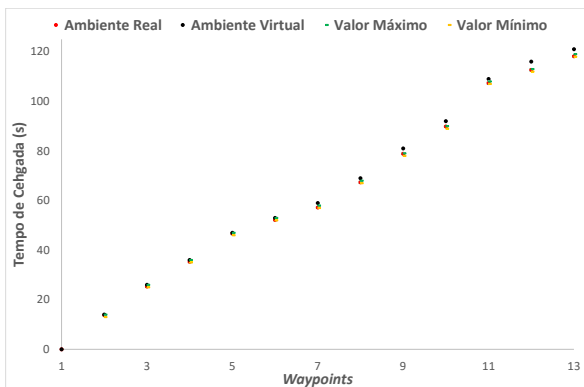
¹Fonte: <https://www.dji.com/br/air-2s>



(a) Velocidade = 1,5 m/s.



(b) Velocidade = 2,0 m/s.



(c) Velocidade = 2,5 m/s.

Figura 6: Gráfico de pontos com valores de tempo de chegada em cada *waypoint*, contendo valores máximos e mínimos.

waypoint sucessivo, ou seja, o tempo entre pontos de chegada (1 e 2, 2 e 3, e assim por diante). Os dados que compõe esta figura reforçam, mais uma vez, a baixa diferença entre o tempo real e simulado, independente do cenário.

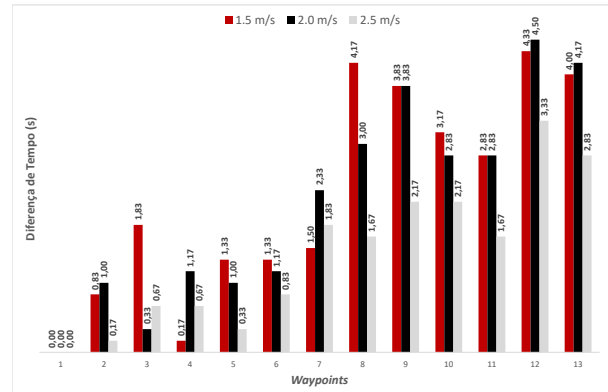


Figura 7: Diferença de tempo de chegada entre ambiente virtual e real em cada *waypoint* para os três cenários de velocidade.

Em todos os cenários de velocidade, temos um tempo médio de execução total em campo de 97,62% preciso em relação ao simulador, sendo a maior diferença de 4,17 segundos no cenário de velocidade 2.0 m/s, uma diferença aceitável em termos de missões que duram minutos.

B. Descarga de Bateria

Outro aspecto importante para o planejamento de missões, além do tempo de voo, é a bateria da aeronave e o seu desempenho ao longo do voo. Inicialmente foram feitos testes desse desempenho no SITL, levando em consideração três cenários. O primeiro cenário é referente a inicialização do simulador e do drone, mantendo-o parado com os motores desligados. O segundo cenário visa registrar o desempenho da bateria ao longo da decolagem do drone, onde o mesmo ao decolar encontra-se parado e planando. O terceiro cenário tem o propósito de avaliar o desempenho da bateria ao longo do voo do drone desenvolvendo uma missão, ou seja, movimentando-se entre *waypoints*.

É possível ter uma visão comparativa da energia e corrente consumidas nas Figuras 8 e 9, para todos os três cenários. Em todas as três curvas apresentadas, é possível verificar uma tendência linear crescente de consumo de corrente e energia ao longo do tempo, mesmo com variações de cenários onde os esforços realizados pelo drone são completamente diferentes. Vide o primeiro cenário em que o veículo encontra-se parado e desligado e, ainda assim, consome bateria no mesmo ritmo que o VANT no terceiro cenário de movimentação entre pontos. Tais resultados são inesperados e não são realísticos em termos de voos em ambiente de energia consumida ao longo do tempo nos três cenários de estado do drone. O

padrão observado leva em consideração apenas uma relação entre consumo de energia baseado no tempo, não levando em consideração ações, velocidade, condições de estado (drone ligado ou desligado), entre outros pontos. Portanto, apenas com estes resultados no simulador, já é possível notar que o modelo nativo de consumo de energia não é realístico, o que representa um problema para as simulações.

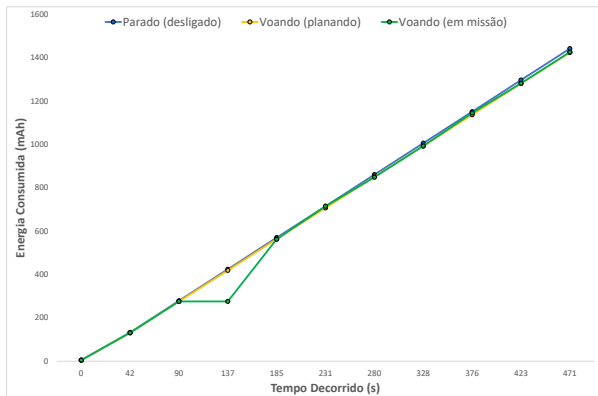


Figura 8: Energia consumida ao longo do tempo nos três cenários de estado do drone.

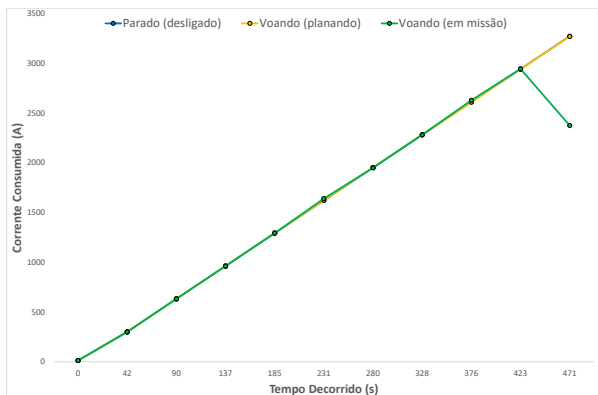


Figura 9: Corrente consumida ao longo do tempo nos três cenários de estado do drone.

VI. DESAFIOS

Devido à integração de diversas ferramentas utilizadas ao longo deste trabalho e da dificuldade de simular um ambiente real de modo fiel para aplicação da metodologia proposta, alguns desafios foram encontrados no desenvolvimento do estudo, como:

- 1) Escalabilidade do Modelo 3D: Devido à quantidade de recursos computacionais necessários para simular

uma grande área com precisão, torna-se fundamental encontrar métodos de obter uma maior eficiência no processo de modelagem 3D e construção do cenário no simulador.

- 2) Obstáculos e Vegetações: O processo de modelagem 3D descrito na Seção IV-A1 entrega com precisão suficiente cenários arquitetônicos (*e.g.* prédios, casas). Entretanto, para cenários com vegetação e obstáculos menores, como postes e placas, não foi possível encontrar um método adequado até o momento.
- 3) Modelo de Bateria Realístico: Os resultados dos testes descritos na Seção V-B apresentam um cenário irreal de consumo de energia, sendo imprescindível a utilização de um modelo de descarga de bateria compatível com a realidade. Tal modelo não foi encontrado disponível nos simuladores Gazebo ou SITL.
- 4) Variações Ambientais: Os testes foram realizados em um cenário de condições climáticas boas, com temperatura e velocidade de vento ideais. Portanto, em termos de comparação, é importante estudar outros cenários com variação destes parâmetros e ver seus impactos na metodologia apresentada.

VII. CONCLUSÃO

Este trabalho apresenta uma metodologia de desenvolvimento de um ambiente virtual e uma análise de dois fatores: o voo no ambiente modelado com as configurações básicas dos simuladores utilizados e o consumo de energia da bateria do drone. O objetivo principal é definir um cenário de verossimilhança entre o ambiente simulado e o real na cidade de Niterói. Na análise, foi considerado um escopo reduzido da cidade, estudando apenas a região onde encontra-se a Escola de Engenharia da Universidade Federal Fluminense.

Nos cenários apresentados com diferentes velocidades de voo do drone, foi possível obter um tempo médio de 97,62% de precisão nos testes de campo em relação ao simulador, com a maior diferença de tempos totais sendo de apenas 4,17 segundos, em cenários que variam de 118 a 195 segundos. Estes resultados mostram que o modelo de drone utilizado e o modelo de ambiente 3D proposto são compatíveis com a realidade e podem ser utilizados para planejar missões na cidade, se a região for expandida para outras áreas além da região de Praias da Baía.

Para o quesito desempenho da bateria, o simulador SITL não se apresenta como uma fonte confiável para planejamento de missões, uma vez que leva em consideração apenas o consumo ao longo do tempo, e não quais ações foram feitas ao longo deste tempo. Deste modo, conclui-se que é necessário incluir um modelo de consumo de energia de bateria mais realista na simulação, uma vez que o modelo nativo é bastante ingênuo.

Portanto, a metodologia de planejamento de missões de voo em cenários reais apresentada neste trabalho mostra-se uma via confiável para distâncias e tempo de voo esperado, sendo uma ferramenta importante para aplicações que envolvam o uso de VANTs. Ela mostra uma base promissora para desempenhar missões de drones na cidade de Niterói, com interfaces

abertas para incluir outros aspectos relevantes relacionadas ao ambiente, comunicação e uso de outros sensores.

Em trabalhos futuros serão realizados estudos dos pontos citados na Seção VI, aprimorando o conhecimento nas ferramentas utilizadas ao longo do trabalho. Além de incluir a utilização de sensores na aeronave (*e.g.* infravermelho) para criação de outros cenários de testes e também avaliar a comunicação entre drones e com a estação-base de controle, por ser um ponto crítico na utilização deste veículo – remotamente ou autonomamente – buscando entender alguns quesitos, como: interferências no processo, segurança e confiabilidade, e perda de sinal com a estação-base.

AGRADECIMENTOS

Agradecemos à Fundação Euclides Cunha e a Prefeitura Municipal de Niterói pelo apoio ao projeto e incentivo à ciência e tecnologia na cidade. E a toda a equipe do Projeto DroNIt e ao suporte do Laboratório Midiacom da Universidade Federal Fluminense.

REFERÊNCIAS

- [1] M. A. Khan, B. A. Alvi, A. Safi, I. U. Khan, Drones for good in smart cities: a review, in: International conference on electrical, electronics, computers, communication, mechanical and computing (EECCMC), Vol. 28, 2018.
- [2] S. H. Alsamhi, O. Ma, M. S. Ansari, F. A. Almalki, Survey on collaborative smart drones and Internet of Things for improving smartness of smart cities, *IEEE Access* 7 (2019) 128125–128152. doi:10.1109/ACCESS.2019.2934998.
- [3] C. H. Choi, H. J. Jang, S. G. Lim, H. C. Lim, S. H. Cho, I. Gaponov, Automatic wireless drone charging station creating essential environment for continuous drone operation, in: 2016 International Conference on Control, Automation and Information Sciences (ICCAIS), 2016, pp. 132–136. doi:10.1109/ICCAIS.2016.7822448.
- [4] G. Wild, J. Murray, G. Baxter, Exploring civil drone accidents and incidents to help prevent potential air disasters, *Aerospace* 3 (3) (2016) 22.
- [5] L. Rodrigues, A. Riker, M. Ribeiro, C. Both, F. Sousa, W. Moreira, K. Cardoso, A. Oliveira-Jr, Flight planning optimization of multiple uavs for internet of things, *Sensors* 21 (22). doi:10.3390/s21227735.
- [6] H. Bian, Q. Tan, S. Zhong, X. Zhang, Assessment of uam and drone noise impact on the environment based on virtual flights, *Aerospace Science and Technology* 118 (2021) 106996. doi:https://doi.org/10.1016/j.ast.2021.106996.
- [7] B. Siemiatkowska, W. Stecz, A framework for planning and execution of drone swarm missions in a hostile environment, *Sensors* 21 (12). doi:10.3390/s21124150.
- [8] Z. Shaikhanov, A. Boubrima, E. W. Knightly, Falcon: A networked drone system for sensing, localizing, and approaching rf targets, *IEEE Internet of Things Journal* 9 (12) (2022) 9843–9857. doi:10.1109/JIOT.2022.3152380.
- [9] J. Sandino, F. Maire, P. Caccetta, C. Sanderson, F. Gonzalez, Drone-based autonomous motion planning system for outdoor environments under object detection uncertainty, *Remote Sensing* 13 (21). doi:10.3390/rs13214481.
- [10] C. Bernardeschi, A. Fagiolini, M. Palmieri, G. Scrima, F. Sofia, ROS/Gazebo based simulation of co-operative UAVs, in: International Conference on Modelling and Simulation for Autonomous Systems, Springer, 2018, pp. 321–334.
- [11] B. Abbyasov, R. Lavrenov, A. Zakiev, K. Yakovlev, M. Svinin, E. Magid, Automatic tool for Gazebo world construction: from a grayscale image to a 3D solid model, in: 2020 IEEE International Conference on Robotics and Automation (ICRA), 2020, pp. 7226–7232. doi:10.1109/ICRA40945.2020.9196621.
- [12] M. Sánchez, J. Morales, J. L. Martínez, J. J. Fernández-Lozano, A. García-Cerezo, Automatically annotated dataset of a ground mobile robot in natural environments via gazebo simulations, *Sensors* 22 (15). doi:10.3390/s22155599.
- [13] A. Farley, J. Wang, J. A. Marshall, How to pick a mobile robot simulator: A quantitative comparison of CoppeliaSim, Gazebo, MORSE and Webots with a focus on accuracy of motion, *Simulation Modelling Practice and Theory* 120 (2022) 102629. doi:https://doi.org/10.1016/j.simpat.2022.102629.
- [14] A. Mairaj, A. I. Baba, A. Y. Javaid, Application specific drone simulators: Recent advances and challenges, *Simulation Modelling Practice and Theory* 94 (2019) 100–117.
- [15] ArduPilot Team, ArduPilot - versatile, trusted, open, <https://www.ardupilot.org/>, acessado em 03 de agosto de 2022.
- [16] S. Baldi, D. Sun, X. Xia, G. Zhou, D. Liu, ArduPilot-based adaptive autopilot: architecture and software-in-the-loop experiments, *IEEE Transactions on Aerospace and Electronic Systems*.
- [17] Gazebo Development Team, Gazebo, <https://gazebo.org/>, acessado em 04 de agosto de 2022.
- [18] N. Koenig, A. Howard, Design and use paradigms for gazebo, an open-source multi-robot simulator, in: 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566), Vol. 3, IEEE, 2004, pp. 2149–2154.
- [19] S.-M. Tang, C.-H. Hsu, Z. Tian, X. Su, An aerodynamic, computer vision, and network simulator for networked drone applications, in: *MobiCom '21: Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, ACM, 2021, pp. 831–833.
- [20] AirSim Development Team, Airsim project, <https://github.com/microsoft/AirSim>, acessado em 03 de agosto de 2022. (2022).
- [21] S. Shah, D. Dey, C. Lovett, A. Kapoor, Airsim: High-fidelity visual and physical simulation for autonomous vehicles, in: *Field and service robotics*, Springer, 2018, pp. 621–635.
- [22] RealFlight Development Team, Realflight, <https://www.realflight.com/>, acessado em 04 de agosto de 2022.
- [23] QGIS Development Team, QGIS geographic information system, <https://www.qgis.org>, acessado em 15 de junho de 2022. (2022).
- [24] Prefeitura de Niterói, SIGeo - sistema de gestão da geoinformação, <https://www.sigeo.niteroi.rj.gov.br/>, acessado em 01 de julho de 2022.
- [25] Google Development Team, Google earth, <https://www.google.com.br/earth/>, acessado em 13 de agosto de 2022.
- [26] Litchi Development Team, Litchi for dji drones, <https://flylitchi.com/>, acessado em 03 de agosto de 2022.