

**DINCON 2017**CONFERÊNCIA BRASILEIRA DE DINÂMICA, CONTROLE E APLICAÇÕES
30 de outubro a 01 de novembro de 2017 – São José do Rio Preto/SP

Integração de Visão Computacional ao controle de VANTs de baixo custo avaliada através de simulação em tempo real

André Rossi Kuroswiski¹

Departamento de Eletrônica Aplicada, ITA, São José dos Campos, SP,

Neusa Maria Franco de Oliveira²

Departamento de Eletrônica Aplicada, ITA, São José dos Campos, SP

Elcio Hideiti Shiguemori³

Instituto de Estudos Avançados, DCTA, São José dos Campos, SP

Resumo. A solução utilizada para a integração de algoritmos de visão computacional ao sistema de controle de voo de VANTs pode afetar diretamente desenvolvimento destas aplicações, tanto pelo custo, quanto pela complexidade envolvida. Este trabalho apresenta uma proposta simples de integração, baseada em soluções de baixo custo, que permite criar uma plataforma flexível para testes simulados e reais. Para avaliar este sistema foi proposto um algoritmo de guiamento baseado em reconhecimento de marcos e funções custo, que realiza desvios necessários em uma rota para que a aeronave evite o sobrevoos de regiões não desejadas, identificadas pela visão computacional. Os resultados dos testes em simulação, nos quais os desvios foram realizados com sucesso, confirmaram viabilidade da integração e a eficácia do algoritmo proposto.

Palavras-chave. Visão Computacional, VANT, Simulação, Guiamento

1 Introdução

A utilização de algoritmos de visão computacional em sistemas de controle e guiamento de Veículos Aéreos não Tripulados (VANT) tem se mostrado uma solução promissora para a expansão das capacidades destas aeronaves, sendo atualmente um tema bastante explorado por diversos grupos de pesquisa em todo o mundo [8]. O desenvolvimento destes algoritmos para a utilização em VANTs apresenta uma característica peculiar, que é a necessidade da integração do processamento de imagens ao sistema de controle destas aeronaves, o que depende diretamente das características do veículo utilizado, principalmente quanto ao sistema de controle de voo e processamento embarcados.

¹ andrekuros@gmail.com,

² neusa@ita.br

³ elcio@ieav.cta.br

Existem soluções variadas no mercado, como controladores de voo que já integram o processamento de imagens e soluções em que o processamento pode ser realizado externamente [12], a escolha pode depender da capacidade de processamento demandada, custo, flexibilidade ou mesmo disponibilidade no mercado, podendo cada caso exigir soluções variadas para se integrar a visão computacional ao sistema de controle. Outro ponto fundamental nestas soluções é a possibilidade de se realizar simulações que permitam um desenvolvimento e avaliação em bancada, como por exemplo em [3, 4, 7], onde simulações são usadas para avaliar fatores específicos como o controle das aeronaves ou algoritmos de tomada de decisão embarcados.

Assim, a escolha das plataformas para o desenvolvimento e utilização de algoritmos de visão computacional integrados ao controle de VANTs, apresenta-se como um ponto chave para pesquisadores desta área, podendo ter um impacto significativo no andamento do projeto. Portanto, este trabalho visa apresentar uma solução viável, flexível e de baixo custo de integração, que permite a realização de simulações em tempo real de todo o sistema, incluindo o processamento de imagens integrado ao controle, com resultados que demonstram a efetividade da mesma através da implementação de um algoritmo de guiamento baseado em visão computacional.

2 Propósito

Este trabalho tem como objetivo propor uma plataforma embarcada que opere como uma solução simples e de baixo custo para a integração de algoritmos de visão computacional ao sistema de controle de um VANT, utilizando componentes de fácil acesso no mercado, que permita uma avaliação prática, em tempo real, de seu funcionamento em um simulador, de modo a tornar mais acessível o desenvolvimento de aplicações complexas antes da realização de voos reais.

Paralelamente, com intuito de avaliar a integração apresentada, é proposto, e testado nesta plataforma, um algoritmo de guiamento baseado em Reconhecimento Marcos [11], que realiza desvios em uma rota de navegação autônoma para evitar o sobrevoo de regiões não desejadas.

3 Plataforma Proposta

A plataforma proposta neste trabalho baseia-se em componentes disponíveis no mercado, já bem estabelecidos, que utilizem hardwares e softwares abertos, de modo a se ter um sistema de baixo custo e simples prototipagem, que possua recursos suficientes para o desenvolvimento e a avaliação da integração de algoritmos de visão computacional ao sistema de controle de VANTs.

Assim, entre diversas opções de controladores de voo disponíveis [6, 13], optou-se por utilizar uma plataforma compatível com as do projeto Pixhawk [9] com o sistema Ardupilot [1] embarcado, por apresentar um bom desempenho por um custo relativamente baixo e uma grande comunidade de desenvolvimento[12]. Ao utilizar o Ardupilot, pode-se ter acesso ao controle de voo através do protocolo MAVLink [10], um protocolo otimizado para VANTs que permite, não só ler dados de voo, como também enviar comandos variados para a operação da aeronave. Para realizar esta comunicação optou-se pela utilização da biblioteca DroneKit [7], que apresenta diversos recursos já implementados, facilitando a criação da ponte entre o controle do VANT e o algoritmo de processamento de imagens. Por fim, conforme apresentado na Figura 1a, propõe-se a utilização de um computador Raspberry Pi para realizar a captura, o processamento de imagens e a integração com o controlador de voo, finalizando um sistema muito compacto e de baixo custo para ser embarcado em uma aeronave de pequeno porte.

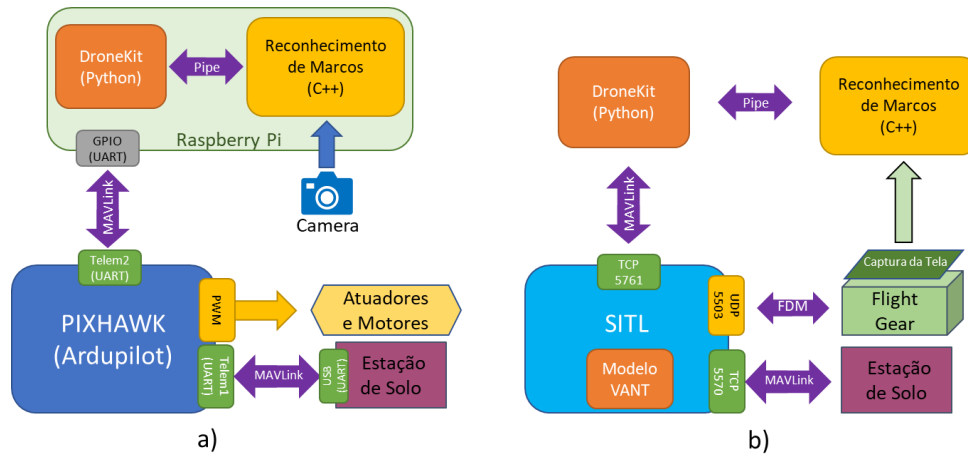


Figura 1 - Diagrama de operação do sistema a ser embarcado (a) e da plataforma de simulação utilizada (b)

Quanto a simulação, a própria plataforma ArduPilot dispõe de uma ferramenta de Software in the Loop (SITL) [1], já bem desenvolvida que, além de simular o modelo de diversas aeronaves e o funcionamento completo do controlador de voo, ainda possui saídas otimizadas para a visualização em simuladores de voo, entre elas a plataforma aberta de simulação Flight Gear, que foi utilizada no projeto. Assim, enquanto o SITL simula a dinâmica e o controle da aeronave, os resultados apresentados no Flight Gear são capturados e utilizados como entrada para o algoritmo de visão computacional, simulando uma câmera embarcada no VANT. Na Figura 1b pode ser observado o diagrama de funcionamento da plataforma de simulação, que opera de maneira muito similar à plataforma a ser embarcada.

4 Algoritmo de Guiamento

Para a avaliação da plataforma proposta foi desenvolvido um algoritmo que possibilitasse explorar as características do sistema, utilizando a visão computacional para identificar marcos específicos no terreno e realizando desvios em uma rota de navegação autônoma em função destas posições. Para isso foi utilizado um algoritmo de Reconhecimento de Marcos conforme apresentado em [11], associado a um método de guiamento baseado minimização de custos, similar ao proposto em [4], de modo a gerar sinais de comando coerentes para o controlador de voo, de modo que a aeronave evitasse o sobrevoo de áreas identificadas como não desejadas.

4.1 Reconhecimento de Marcos

O algoritmo de visão computacional utilizado é baseado em diferentes técnicas de processamento de imagens e inteligência artificial para realizar a identificação de marcos pré-selecionados. Apresentando boa acurácia na identificação a um custo computacional relativamente baixo, este algoritmo mostra-se uma solução viável para ser embarcado em aeronaves autônomas de baixo custo [11].

Para utilizar este algoritmo, imagens de uma região ou objeto que se deseja identificar, devem ser carregados previamente, assim, em tempo real, a imagem capturada pela aeronave é processada e, caso o marco seja encontrado, sua posição é indicada, inicialmente em pixels, mas que pode ser

convertida para coordenadas caso se conheça a posição atual da aeronave e as características da câmera utilizada.

No algoritmo de guiamento implementado, o Reconhecimento de Marcos é utilizado para indicar a presença de áreas que não devem ser sobrevoadas, para isso, imagens que caracterizam estas regiões, ou objetos, são inseridas previamente no sistema e, durante o voo, quando identificadas, suas posições estimadas passam a representar o ponto central de uma região a ser evitada.

4.2 Função de Custo

A técnica utilizada para se realizar os desvios desejados durante a rota foi baseada em funções de custo que definem pesos para cada posição analisada, indicando posições mais “desejadas” para a rota da aeronave através dos menores valores encontrados.

Neste trabalho foram utilizadas funções logarítmicas para definir os custos relativos aos marcos a serem evitados, deste modo pode-se definir distâncias mínimas para a qual o obstáculo passa a ter sua influência considerada, assim como a intensidade com que a aeronave deve evitar o mesmo após entrar na região de influência. Estes valores estão definidos respectivamente por R , raio de influência, e G_M , ganho do marco, na equação 4.1, na qual d representa a distância para o centro do marco.

$$C_M(d) = \begin{cases} G_M \cdot \ln\left(\frac{R}{d}\right), & 0 < d < R \\ 0, & d > R \end{cases} \quad (4.1)$$

Para completar o cálculo do custo total associado a cada ponto, foi adicionado o custo relativo ao desvio da aeronave em relação à rota desejada, devendo indicar menores custos para o deslocamento mais direto em relação ao próximo ponto de navegação. Para este valor foi utilizado a distância euclidiana entre a posição analisada e a coordenada do ponto a ser alcançado, deste modo, dado a posição da aeronave, sem a presença de obstáculos a rota se daria na menor distância possível. Com isso pode ser obtido o custo total relativo a cada posição a ser analisada, conforme apresentado na equação 4.2, onde P_{Alvo} é a posição do próximo ponto de navegação, P é a posição a ser analisada, n é o número de marcos considerados, G_{Alvo} é o ganho do alvo e P_M é a posição destes marcos.

$$C_{Total}(P) = G_{Alvo} |P - P_{Alvo}| + \sum_{i=1}^n C_M(|P - P_{M_i}|) \quad (4.2)$$

4.3 Funcionamento do Algoritmo

O algoritmo de guiamento foi implementado para operar em paralelo aos recursos já existentes no Ardupilot, assim planejou-se sua utilização em conjunto com o modo de navegação autônoma nativo deste controlador de voo, que permite, entre diversas funções, planejar uma navegação para cumprir diversos pontos de controle pré-definidos. Deste modo, o algoritmo desenvolvido só entra em operação no momento em que um marco é identificado, realizando o desvio necessário e devolvendo o controle da aeronave para o modo nativo de guiamento após ultrapassada a zona de influência a ser evitada.

Quando um novo marco é identificado, o próximo ponto de controle já planejado pelo modo navegação do Ardupilot é tomando como P_{Alvo} , e então o algoritmo assume o controle da aeronave em alto nível, enviando comandos que atuam diretamente no vetor de velocidade da aeronave,

calculados de modo garantir um deslocamento para as regiões de menor custo. Para encontrar a direção ideal de voo são realizadas buscas locais em duas fases: primeiro calcula-se o custo em pontos localizados a 2 segundos de voo na velocidade atual, em 12 direções igualmente distribuídas entre 0° e 330° em relação ao deslocamento da aeronave. Escolhe-se então a direção de voo equivalente ao ponto de menor custo, para a qual é realizada uma nova busca em 15 novas direções distribuídas igualmente em torno da direção considerada, de modo a se obter uma resolução de 2° na segunda busca. Por fim, calculando o custo equivalente em cada direção, encontra-se um novo mínimo que indicará a direção a ser voada. O processo é então repetido continuamente, garantindo que a aeronave busque os mínimos locais de modo a evitar o sobrevoio de regiões não desejadas, identificadas pelos marcos encontrados.

Após ultrapassar a região de influência do marco, o algoritmo interrompe sua atuação no controle da aeronave, e reativa o modo de navegação autônomo nativo do controlador de voo, que prossegue na navegação planejada até a identificação de novas referências.

5 Resultados e Discussões

Para avaliar o funcionamento do sistema foram realizados voos simulados de um quadricóptero sobre o cenário de um aeroporto no Flight Gear. Para estes voos foi imposto que o VANT não deveria sobrevoar aeronaves estacionadas, devendo identificá-las através do algoritmo de reconhecimento de marcos, realizando os desvios quando necessário. Neste cenário, foi selecionado para testes um pátio de estacionamento no qual até 8 aeronaves de 2 modelos diferentes podem estar estacionadas de maneiras variadas a cada execução do Flight Gear.

Inicialmente foi realizado um sobrevoio a 100 metros de altura sobre os dois modelos de aeronaves, através do qual extraiu-se as imagens utilizadas como referência para o algoritmo de reconhecimento de marcos. Em seguida testou-se a eficiência deste algoritmo ao processar as imagens do simulador, realizando perfis de voo em diferentes alturas (de 20 a 250m) e proas de aproximação (de 0° a 330° com passos de 30°), tendo sido obtido uma identificação correta dos alvos em 93% dos quadros processados. Na Figura 2 pode ser observado um exemplo dos dois modelos de aeronaves sendo identificados, assim como as imagens de cada que foram utilizadas como referência.

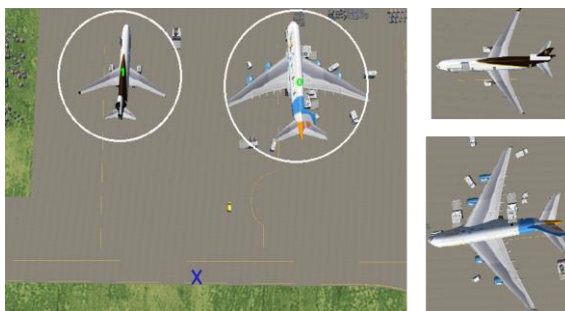


Figura 2 - Visualização das aeronaves reconhecidas durante o voo simulado (esq.) e imagens utilizadas como referência (dir.)

Confirmado o funcionamento do algoritmo de reconhecimento de marcos, foram realizados dois perfis de voo para a avaliação do algoritmo de desvio, o primeiro no sentido longitudinal e o segundo perpendicular ao pátio de estacionamento. Na Figura 3a está apresentado um dos resultados obtidos para o primeiro perfil de voo, onde pode-se observar através da linha magenta,

correspondente ao traçado de voo executado, que os desvios necessários foram realizados de modo a evitar o sobrevoos das aeronaves estacionadas. Vale ressaltar as aeronaves estacionadas no simulador não estão presentes na imagem de satélite apresentada na figura, portanto, nas posições equivalentes, foram adicionadas silhuetas de aeronaves para facilitar a visualização. Na Figura 3b pode ser observado o resultado de um dos voos no perfil perpendicular ao pátio de estacionamento, onde a identificação de duas aeronaves levou a um desvio bastante significativo em função dos parâmetros utilizados inicialmente para o algoritmo. Após um ajuste apenas em G_{Alvo} , aumentando de 1 para 3, a aeronave foi capaz cruzar entre as aeronaves sem sobrevoar nenhuma das duas.



Figura 3 - Resultados da navegação simulada sobre o pátio de estacionamento

Pelos resultados apresentados foi possível confirmar a viabilidade da integração proposta entre a visão computacional e o controlador de voo de um VANT de baixo custo, assim como a viabilidade da utilização do SITL do Ardupilot, associado as imagens capturadas do Flight Gear para se desenvolver e testar algoritmos de guiamento baseado em processamento de imagens. Soluções similares podem ser encontradas por exemplo em [3, 4, 7], no entanto, em nenhuma delas foi implementado a simulação completa, com o processamento de imagens operando em tempo real, conjuntamente com a simulação da dinâmica da aeronave, como é possível no sistema proposto.

O algoritmo de guiamento utilizado para se evitar o sobrevoos de regiões não desejadas, inicialmente proposta a fim de permitir a avaliação de diversos recursos da plataforma implementada, cumpriu o objetivo proposto, mostram-se como uma solução inicial efetiva para o problema.

6 Conclusões

Os sistemas propostos, tanto para simulação, quanto para a solução de integração embarcada entre visão computacional e o piloto automático de uma aeronave, podem ser implementados sem muitas dificuldades devido ao grau de desenvolvimento das plataformas utilizadas, tornando-se uma solução prática e de baixo custo para a prototipagem eficiente de algoritmos de visão computacional para VANTs. Certamente a utilização de componentes de código aberto mantido por comunidades virtuais, como é o caso do Ardupilot, pode representar um óbice para o sistema

pela falta de certificação e suporte dos mesmos, o que, no entanto, é contrabalanceado pelo baixo custo e grande número de usuários para o compartilhamento de experiências.

Como próximos passos pretende-se realizar voos reais em condições similares, de modo a se confirmar a fidelidade da simulação realizada, avaliando-se, além do desempenho dinâmico da aeronave, o funcionamento do algoritmo de visão computacional em um ambiente bem mais complexo e variável e o impacto destes nos resultados do algoritmo de guiamento nestas condições.

Referências

- [1] Ardupilot Development Team, ArduPilot Autopilot Suite, (2016). Disponível em: <http://ardupilot.org/ardupilot/>. [Acessado: 12-jul-(2017)].
- [2] A. Bittar, H. V. Figueiredo, P. A. Guimaraes, e A. C. Mendes, Guidance software-in-the-loop simulation using x-plane and simulink for UAVs, International Conference on Unmanned Aircraft Systems, ICUAS 2014, 993–1002, (2014), DOI:10.1109/ICUAS.2014.6842350.
- [3] A. Bittar, N. M. F. De Oliveira, e H. V. De Figueiredo, Hardware-in-the-loop simulation with X-plane of attitude control of a SUAV exploring atmospheric conditions, J. Intell. Robot. Syst. Theory Appl., vol. 73, 271–287, (2014), DOI:10.1007/s10846-013-9905-8.
- [4] D. E. Breen, Cost minimization for animated geometric models in computer graphics, J. Vis. Comput. Animat., vol. 8, 201–220, (1997).
- [5] G. Cai, J. Dias, e L. Seneviratne, A Survey of Small-Scale Unmanned Aerial Vehicles: Recent Advances and Future Development Trends, Unmanned Syst., vol. 2, 175–199, (2014), DOI:10.1142/S2301385014300017.
- [6] H. Choi, M. Geeves, B. Alsalam, e F. Gonzalez, Open source computer-vision based guidance system for UAVs on-board decision making, IEEE Aerospace Conference Proceedings, 1-5, (2016), DOI:10.1109/AERO.2016.7500600.
- [7] DRONEKIT, Introducing DroneKit-Python, 3D ROBOTICS, (2015). Disponível em: <http://dronekit.io/>. [Acessado: 31-maio-(2017)].
- [8] C. Kanellakis e G. Nikolakopoulos, Survey on Computer Vision for UAVs: Current Developments and Trends, J. Intell. Robot. Syst. Theory Appl., vol. 87, 141–168, (2017), DOI:10.1007/s10846-017-0483-z.
- [9] L. Meier, P. Tanskanen, F. Fraundorfer, e M. Pollefeys, the Pixhawk Open-Source Computer Vision Framework for Mavs, Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci., vol. 38, 13–18, (2012), DOI:10.5194/isprsarchives-XXXVIII-1-C22-13-2011.
- [10] L. Meier, MAVLink Micro Air Vehicle Communication Protocol, (2009). Disponível em: <http://qgroundcontrol.org/mavlink/start>. [Acessado: 30-maio-(2017)].
- [11] P. F. F. Silva Filho, Automatic Landmark Recognition in Aerial Images for Autonomous Navigation System of Unmanned Aerial Vehicles, ITA, (2016).
- [12] Z. Yang, F. Lin, e B. M. Chen, Survey of autopilot for multi-rotor unmanned aerial vehicles, Conference of the IEEE Industrial Electronics Society, 6122–6127, (2016), DOI:10.1109/IECON.2016.7793820.