

DESENVOLVIMENTO DE UMA INTERFACE DE CONTROLE E PROGRAMAÇÃO DE UM DRONE AUTÔNOMO

Caique de S. F. Duarte (IC)¹, Alexandre C. B. Ramos(PQ)¹

¹ Universidade Federal de Itajubá

Palavras-chave: Vants, Python, Robot operating system, Competição, Interface gráfica

Introdução

A agricultura é um dos maiores e mais importantes indústrias do mundo, decisões sobre como, quando e o que cultivar continuam altamente desotimizadas e imprecisas hoje em dia. Existe uma oportunidade de melhorar drasticamente o processo de produção de comida, utilizando dados visuais e aprendizado de máquina. (TOEWS, ROB. A Wave Of Billion-Dollar Computer Vision Startups Is Coming.)

No século XXI, com o crescimento e principalmente a adoção comercial de conceitos como os apresentados por Toews (2021), tecnologias como: drones de plantio, pulverização e inspeção do solo, processamento digital de imagens por satélites e dispositivos com múltiplos sensores e interfaces de comunicação como 5G expandem cada vez mais as oportunidades para inovação.

Neste contexto, a competição IMAV destina uma de suas provas ao processamento de imagens e dados sensoriais com drones autônomos. Na edição de 2018, por exemplo, uma das provas exigia que o drone seguisse uma corda de ponta a ponta de forma totalmente autônoma.

Propomos então o desenvolvimento de uma interface gráfica para integração dos códigos das provas, além de códigos para garantir a segurança do drone e facilidades que trazem vantagem competitiva à equipe Black Bee Drones, que representou o Brasil nesta edição.

Metodologia

O projeto deu início com o planejamento da estrutura do código, que por fim ficou dividida em 3 arquivos de código-fonte `interface.py`, `bebop.py` e `interface.ui`. Inicialmente foi usada a biblioteca Pygubu para a criação da interface em XML, que gerou o arquivo `interface.ui` com todos os botões, textos e imagens, ainda sem valor ou funcionalidade.

Na outra ponta do sistema o código `bebop.py` faz o papel de driver, enviando e recebendo comandos do Bebop através de tópicos ROS. Portanto comandos como de

movimento nos eixos XYZ, pouso e decolagem, e informações como altitude, velocidade e posição são tratados neste módulo.

Intermediando a interface XML e os comandos do Bebop está o código `interface.py`, que recebe informações do Bebop e apresenta ao usuário, e recebe comandos do usuário e envia ao Bebop, além de detectar e tratar comandos recebidos pelo teclado e controle.

O arquivo `interface.py` foi desenvolvido de forma que missões autônomas também programadas em ROS, independentemente da linguagem possam ser facilmente integradas à interface.

A figura abaixo ilustra a arquitetura do sistema:

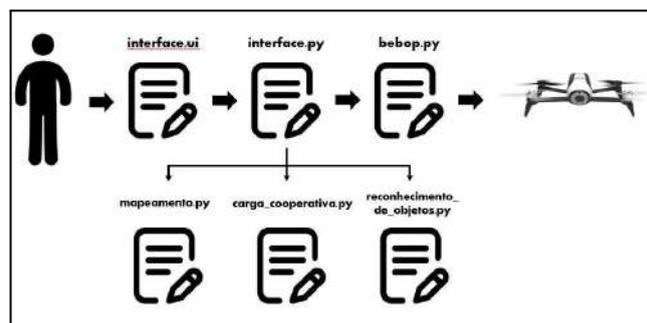


Figura 1 - Arquitetura do sistema

Fonte: Figura do autor, 2018

Como por exemplo o algoritmo que segue a corda, mencionado na introdução deste artigo, representado por um botão que, a partir do momento que é ativo, abaixa a câmera e começa a detecção e o alinhamento com a corda.

Além desta integração dos códigos, o sistema também possibilita o controle do drone através do teclado e de um *joystick*, de forma que o piloto consiga sobrescrever as ações executadas pelo código a qualquer momento em caso de emergência, garantindo a segurança na realização da prova.

A usabilidade trazida pela interface gráfica, por exemplo, na troca de controle entre o código e o teclado e na ativação de código por botão trouxeram vantagens competitivas na competição em momentos como novas tentativas dentro da janela de tempo de 10 minutos e a

transição entre uma tarefa e outra.

Resultados e discussão

A plataforma ficou funcional a tempo para a participação da equipe na edição 2018 da competição e foi utilizada para a realização das provas. O resultado final está representado abaixo.

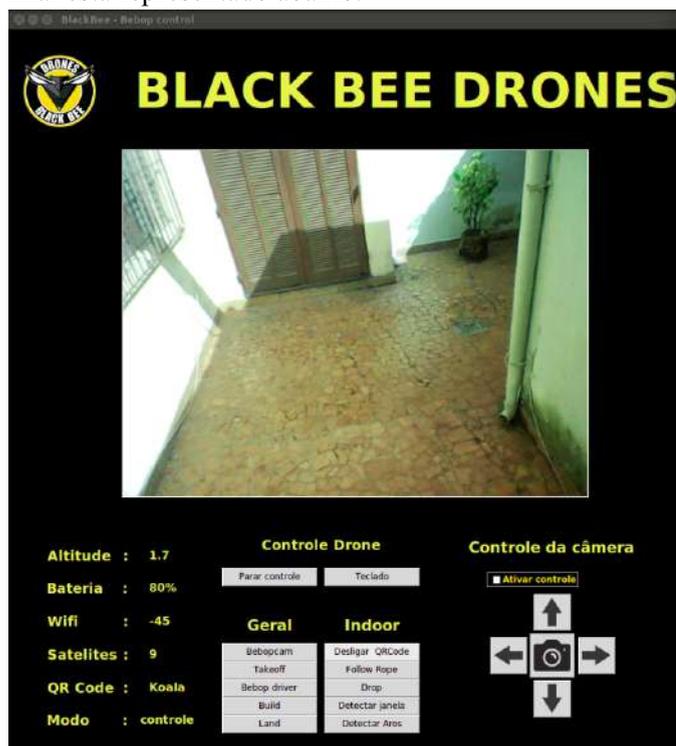


Figura 2 - Interface gráfica desenvolvida
Fonte: Figura do autor, 2018

A escolha da ferramenta ROS para o desenvolvimento da interface possibilitou que algoritmos complexos como os de visão computacional escritos em C++ se comunicassem diretamente com a interface desenvolvida em Python, trazendo uma gama quase ilimitada de possibilidades de missões que poderiam ser realizadas.



Figura 3 – Uso da plataforma na competição
Fonte: Black Bee Drones, 2018

A plataforma desenvolvida como objeto deste estudo foi utilizada nas edições 2018 e 2019 da competição, que aconteceram na Austrália e Espanha, respectivamente. Diversas modalidades e provas foram realizadas com seu auxílio, como carga cooperativa por drones, *drone parade* (voo sincronizado com drones), seguir a corda e passar por uma janela (apresentada na figura 2).

Conclusões

A plataforma atendeu aos objetivos estipulados inicialmente, impactando na execução das provas da competição pela equipe Black Bee Drones. Apesar disso, diversas melhorias podem ser inseridas, tanto na estrutura e organização do código, permitindo seu crescimento, quanto nas funcionalidades que o sistema oferece, melhorando cada vez mais o desempenho da equipe não só na competição mas durante o ano, no desenvolvimento e teste das missões.

A equipe obteve o primeiro lugar na modalidade de carga cooperativa na edição de 2018, sendo a única a conseguir realizar a prova.



Figura 4 - Imagem do certificado de primeiro lugar na competição 2018 em Melbourne - Austrália

Agradecimento

Ao professor Alexandre Carlos Brandão Ramos, por seu apoio durante a pesquisa e também pelos momentos de motivação, meus agradecimentos. Seu caráter crítico refletiu de forma positiva no projeto do começo ao fim. À toda a equipe Black Bee Drones, que me proporcionou conhecimento e estrutura para que este trabalho fosse realizado, além do companheirismo essencial em muitos momentos do projeto. E por fim, à FAPEMIG pelo apoio financeiro durante o projeto.

Referências

LENTIN, J. ROS Robotic Projects. Birmingham, United Kingdom: Packt Publishing. 2017.
STANFORD ARTIFICIAL INTELLIGENCE LABORATORY et al., 2018. Robotic Operating System, Available at: Disponível em: <<https://www.ros.org>>. Acesso em: 23 set. 2022.
LUNDH, F. 1999. An introduction to tkinter. URL: www.pythonware.com/library/tkinter/introduction/index.htm
VEGA, I e MARTINEZ-CARRANZA, J. IMAV 2018 - Towards Micro Aerial Manipulation Using a Computational Compensation Strategy. 2018.