

## SISTEMA COMPUTACIONAL BASE PARA NAVEGAÇÃO E CONTROLE DE VEÍCULOS AÉREOS NÃO TRIPULADOS

Lucas Schulze<sup>1</sup>, Douglas Wildgrube Bertol<sup>2</sup>

<sup>1</sup> Acadêmico do Curso de Engenharia Elétrica - CCT

<sup>2</sup> Orientador, Departamento de Engenharia Elétrica - CCT – douglas.bertol@udesc.br

Palavras-chave: Quadrotor. MPC. HIL.

A crescente utilização civil de veículos aéreos não tripulados (VANTs) nos últimos anos, em tarefas como missões de resgate, serviços de entrega e fotografia, exigem uma rapidez cada vez maior no desenvolvimento desses novos produtos. Uma das barreiras existentes é a dificuldade em implementar na prática algoritmos e controladores desenvolvidos e validados em simulação, pois muitas vezes os ambientes de simulação não consideram fenômenos físicos que ocorrem na situação real e/ou a incompatibilidade entre o ambiente de simulação e o hardware utilizado, problemas que exigem a adequação e reescrita de parte ou de todo o código para utilização no VANT real.

Nesse contexto, o trabalho propõe e valida um *framework* para o desenvolvimento e implementação prática de controladores desenvolvidos em ambiente de simulação.

Inicialmente foi realizado um estudo de algumas ferramentas de hardware e software para desenvolvimento de robôs. Foram realizados estudos e experimentos com a placa Raspberry Pi, o controlador de voo Pixhawk, com a plataforma de desenvolvimento de software para robótica *Robot Operation System* (ROS) e com o simulador Gazebo.

A partir do modelo matemático e da análise física de um quadrotor, foi desenvolvido um controlador preditivo baseado em modelo, do inglês *model predictive controller* (MPC), para o controle de altitude no ambiente de desenvolvimento Simulink. A escolha pelo MPC e não algum outro controlador clássico, como o PID (Proporcional Integral Derivativo), se deu pela possibilidade de considerar as limitações do VANT real no controlador e pela crescente utilização deles em sistemas ditos de dinâmica rápida.

Após os testes dentro do ambiente Simulink, que considera apenas o modelo matemático e não fatores como vento, imperfeições e assimetria no modelo real. Utilizando ferramentas de geração de código do próprio Simulink, foi implementado o controlador em um *plug-in* do Gazebo que utiliza o ROS para envio de informações da situação do VANT e recebe a posição de referência. Toda a simulação do Gazebo acontece em um Raspberry Pi, utilizando a técnica de *Hardware-in-the-loop* (HIL), sendo possível então avaliar o comportamento do controlador e do hardware em uma situação muito próxima da real.

No trabalho foram variados parâmetros do controlador MPC, como os horizontes de predição e controle, e comparados os resultados entre a simulação do modelo matemático em software e a simulação em HIL no Gazebo, analisando também o custo computacional no Raspberry Pi. Apesar da comparação ser realizada entre duas simulações, o Gazebo considera fenômenos físicos, como colisão e inércia, e a utilização da plataforma ROS permite a reutilização

de quase todo o código, sem necessidade de grandes mudanças. O que aumenta a fidelidade e representatividade da situação simulada com a situação real.

O aumento nos horizontes de predição e controle diminuíram o erro do VANT em relação a referência tanto no Simulink como no Gazebo, mas aumentaram o custo computacional no Raspberry Pi, o que era esperado e demonstra a efetividade quanto ao seguimento de referência do controlador implementado no *framework* proposto. O aumento no custo computacional acaba comprometendo o desempenho do controlador quando comparado aos requisitos de tempo real do sistema.

Pela geração de código a partir do Simulink e a forma como o *plug-in* do Gazebo foi desenvolvido, é possível alterar os parâmetros do controlador sem necessidade de alterar qualquer linha de código. Seria possível ainda alterar todo o controlador, utilizando um PID por exemplo, sem necessidade de ajustes no código. Demonstrando assim a efetividade do *framework* desenvolvido, que permite além de analisar o erro do seguimento de referencia, também o custo computacional do hardware utilizado na situação prática.