

ANÁLISE COMPARATIVA ENTRE MICROSOFT FLIGHT SIMULATOR E FLIGHTGEAR FLIGHT SIMULATOR EM TESTES HARDWARE-IN-THE-LOOP

LUIZ F. A. CANTONI*, ARMANDO A. NETO*, LUIZ CHAIMOWICZ*, MARIO F. M. CAMPOS*

*Universidade Federal de Minas Gerais
Departamento de Ciência da Computação
Laboratório de Visão Computacional e Robótica (VeRLab)
Belo Horizonte, MG, Brasil

Emails: {lcantoni, aaneto, chaimo, mario}@dcc.ufmg.br

Abstract— In this paper we present a comparative relationship between two flight simulators, the Microsoft Flight Simulator X[®] and the Flightgear Flight Simulator, taking them as basis to the implementation of Hardware-in-the-loop systems. These systems represent important tools in the project and control of industrial processes, once they provide a decrease in time and cost of those. In this work we consider the use of such simulators for the study and project of unmanned aerial vehicles, also known as *UAVs*. The main objective is to compare both simulation softwares in order to verify which of them present more advantages in relation to the other, especially considering the quality of the dynamic simulation of the actual aerial vehicles. A qualitative analysis is proposed, based on the experience of the authors with both simulators, and the best of them is presented in the end of the paper.

Keywords— Aerial robotics, HWIL systems, *UAVs*, simulation.

Resumo— Neste artigo é estabelecida uma relação comparativa entre dois simuladores de voo, o *Microsoft Flight Simulator X[®]* e o *Flightgear Flight Simulator*, enquanto bases para a construção de sistemas *Hardware-in-the-loop (HWIL)*. Esses sistemas constituem importantes ferramentas de projeto e controle de processos industriais, uma vez que proporcionam redução no tempo e custo envolvidos. Neste trabalho considera-se a utilização de tais simuladores para o estudo e projeto de veículos aéreos autônomos não-tripulados, também chamados de VAANTs. O principal objetivo é contrapor as duas ferramentas de simulação citadas afim de verificar qual delas apresenta mais vantagens em relação a outra, principalmente em relação a qualidade da simulação dinâmica de veículos aéreos reais. Uma análise qualitativa é proposta com base na experiência dos autores em relação a cada uma das ferramentas utilizadas, e a melhor delas é exibida ao final do texto.

Palavras-chave— Robótica aérea, Sistemas *HWIL*, VAANTs, simulação

1 Introdução

Sistemas *Hardware-in-the-Loop (HWIL)* são ferramentas de grande importância no planejamento e construção de projetos de engenharia. Inicialmente utilizados nos setores aeronáuticos e aeroespaciais, atualmente a utilização de testes *HWIL* estende-se às diversas áreas de pesquisa, auxiliando a minimizar o tempo e os custos envolvidos na elaboração de sistemas complexos. Na Robótica, assim como em outras áreas da Computação, tais sistemas auxiliam o estudo do comportamento de novas técnicas, testando partes de um sistema complexo de maneira modular, antes de incorporá-las às plantas reais (neste caso os sistemas robóticos), onde os efeitos são bem mais críticos.

As bases da elaboração de tais sistemas são a modelagem e a simulação. A modelagem permite reproduzir o comportamento dos sistemas reais por meio de representações matemáticas tão complexas quanto necessário, gerando uma réplica mais simples e barata dos processos estudados. A simulação permite que esses modelos sejam utilizados enquanto sistemas reais para a realização de testes em caráter experimental a custos mais baixos, e com maior rapidez.

Já os Veículos Aéreos Autônomos Não-Tripulados (VAANTs), também mais conhecidos

pela sigla *UAVs* (do inglês, *Unmanned Aerial Vehicles* ou *Uninhabited Aerial Vehicles*), podem ser definidos como aeronaves não-tripuladas que executam tarefas (como o próprio voo, por exemplo) de forma parcial ou completamente autônoma. Para isso, são projetados sistemas de controle e algoritmos capazes de prover autonomia a esse veículos.

Nos últimos anos, observou-se um crescimento considerável na utilização desses veículos. Um dos principais fatores que estimulou esse crescimento foram os avanços tecnológicos ocorridos em diversas áreas como a miniaturização de componentes eletrônicos, o desenvolvimento de sensores mais precisos, a introdução de câmeras de vídeo cada vez menores e com maior resolução, a popularização dos sistemas de GPS (*Global Position System*) de baixo custo e o avanço da comunicação *wireless*, entre outras características.

Além de preservar a vida humana (retirando o piloto da aeronave), o interesse por esse tipo de veículo está muitas vezes associado aos baixos custos de desenvolvimento e operação se comparados aos de aeronaves maiores e tripuladas. Entretanto, pelo fato de não possuir um piloto a bordo, surgem diversos desafios para fazer com que o voo (ou pelo menos uma parte dele) seja realizado de forma autônoma. São necessários sistemas eletrônicos embarcados (como piloto automático e sensores) e

algoritmos de controle e navegação que consigam manter a estabilidade e a trajetória do voo.

Nesse sentido, um requisito fundamental para a maioria dos projetos é realizar simulações para validar esses algoritmos antes de aplicá-los aos voos reais. Para isso, muitos pesquisadores combinam a técnica de *HWIL* com simuladores *COTS* (*commercial-of-the-shelf*). Essa combinação permite redução significativa no tempo de desenvolvimento, pois os simuladores de voo prevêem um ambiente que permite utilizá-los na validação de sistemas embarcados reais.

Neste trabalho apresenta-se uma comparação entre os simuladores de voo *Microsoft Flight Simulator X*[®] e *Flightgear Flight Simulator*. Esse último já foi largamente estudado e aplicado para tal fim, já que fornece uma plataforma gratuita e de qualidade, e especialmente por que possui código aberto permitindo a alteração conforme a necessidade dos pesquisadores. Já o primeiro é menos aplicado a fins científicos e possui certas vantagens se considerado.

2 Trabalhos relacionados

Ajustar um piloto automático a partir de voos reais pode consumir muitas horas de trabalho, levar a resultados insatisfatórios e a situações perigosas. Por esse motivo, muitos pesquisadores lançam mão de técnicas como *HWIL* (Adiprawita et al., 2007; How et al., 2004; Sorton and Hammaker, 2005). Obviamente, essa técnica não pode substituir os experimentos reais, embora seja muito útil nas fases iniciais. Uma vez tendo sido realizados os testes com a arquitetura *HWIL*, espera-se que o mesmo controlador possa ser utilizado diretamente sobre o veículo aéreo real.

Existem diversos trabalhos na literatura que tratam do tema sobre ambientes de simulação para veículos aéreos. Em (Göktogian et al., 2003), por exemplo, os autores discutem a arquitetura de um sistema *HWIL* de tempo real para *UAVs* onde, em especial, é apresentado um arcabouço de comunicação multi-sistemas. São estabelecidos vários mecanismos de teste e validação para missões, que permitem o estudo em um nível mais alto de autonomia para aeronaves não-tripuladas. Apesar de seu estágio inicial de desenvolvimento, a utilização desse módulo já fornece uma ferramenta bastante interessante para o estudo de sistemas com múltiplos veículos.

Além da característica multi-agente, importante para estabelecer etapas do planejamento de missões, outro fator importante a ser considerado no estágio de simulação é o caráter de visualização gráfica do ambiente. Alguns trabalhos, como (Castillo-Effen et al., 2005), por exemplo, apresentam plataformas com ambientes que permitem a inspeção visual do comportamento dos veículos, como trajetórias realizadas, acelerações laterais in-

desejáveis, entre outras. Vale destacar também a possibilidade de aplicação de técnicas de navegação baseada em imagens, além de outras estratégias fundamentadas na área de Visão Computacional.

Aspectos mais afins à simulação *HWIL* são abordados em (Gholkar et al., 2004), com um trabalho mais voltado para os testes de *hardware* de um *Micro-UAV* (*MUAV*), utilizando modelagem de sensores, simulação de dinâmica de voo e um sistema *Linux* de tempo real para controle. Os testes simulados permitiram a correção de várias situações que poderiam ser críticas em condições reais de voo, justificando plenamente o uso necessário da simulação.

Em (Astuti, 2007) foi utilizada a técnica de *HWIL* para validar algoritmos de planejamento de trajetória e controle de altitude para um *UAV* cuja missão é realizar amostragens químicas do gás da coluna de erupção de vulcões. Foi utilizado o simulador de voo *X-Plane*[®] para enviar ao piloto automático real informações sobre o veículo (velocidade, altitude, atitude e posição). O piloto automático então, envia os sinais de controle para o simulador com o objetivo de alterar as superfícies móveis do avião (aileron, profundor e leme) e assim de forma autônoma, controlar a trajetória de voo da aeronave.

Já em (Sorton and Hammaker, 2005) os autores criam um sistema *HWIL* de baixo custo, utilizando apenas um simulador de voo de código-aberto e uma ferramenta de cálculos matemáticos (*MatLab*[®]). O simulador em questão é o *Flight-Gear Flight Simulator* (*FgFS*), um pacote computacional desenvolvido principalmente para fins acadêmicos, mas que tem se tornado um dos simuladores de voo mais utilizados no mundo. Algumas de suas vantagens, além de seu código-aberto, são a qualidade da visualização gráfica, os diversos protocolos de comunicação que suportam inúmeras máquinas para simulação multi-agente e modelagem dinâmica bastante realista.

Outro exemplo de utilização do *FgFS* pode ser encontrado em (Alves Neto and Campos, 2008), onde os autores exploram a utilização do simulador para testar o sistema embarcado de um *VAANT* real construído na Universidade Federal de Minas Gerais. O resultado apresentado gerou uma arquitetura simples para o controle de trajetória de veículos aéreos e mostrou que é possível utilizar esse simulador para a navegação de veículos baseada em Visão Computacional.

Todos os trabalhos citados, e outros relativos ao mesmo assunto, apresentam implementações de sistemas *HWIL* utilizando diversas ferramentas de simulação com as mais variadas características. Os que utilizam simuladores de voo gráficos, como o *FgFS* apresentam como principais vantagens o realismo na modelagem visual do ambiente, a qualidade nas dinâmicas de voo implementadas, e a grande variedade de veículos aéreos simulados (ae-

ronaves de asa-fixa, de asa rotativa, veículos mais leves que o ar, entre outras). Por isso, este trabalho visa realizar uma comparação entre ambientes de simulação que oferecem todas estas características, mas que particularmente possuem vantagens e desvantagens na sua utilização.

3 Metodologia

A metodologia empregada neste trabalho foi basicamente dividida em duas partes:

- estudo para a operação do *MsFS*[®] enquanto ferramenta de simulação para o sistema *HWIL*;
- análise comparativa entre o *MsFS*[®] e o *FgFS*.

Na primeira parte será abordado o arcabouço utilizado para transformar o *MsFS*[®] em uma ferramenta de simulação passível de ser controlada externamente. Em outras palavras, atua-se de modo que seja possível controlar aeronaves dentro desse simulador por meio de algoritmos de controle sendo executados em outras máquinas que não a da simulação. Na segunda etapa são estabelecidos alguns parâmetros de comparação entre os dois simuladores de voo, no intuito de verificar qual o mais adequado para utilização em sistemas *HWIL*. Inicialmente, porém, é apresentado um breve histórico desses simuladores de modo a introduzir e a estabelecer comparações primárias entre as ferramentas.

3.1 Simuladores de voo

Os simuladores de voo são sistemas que tentam imprimir da forma mais realista possível a experiência de voar uma aeronave. No contexto deste trabalho, abordam-se os simuladores *COTS* (*Commercial-of-the-shelf*) com foco na aviação civil, que possuem, como principal objetivo o entretenimento, mas que devido ao nível de realismo a que chegaram, são também utilizados para treinamento de pilotos e controladores de voo no mundo real.

As principais vantagens apresentadas por esses simuladores em relação ao projeto e controle de veículos aéreos são:

- modelagem dinâmica bastante próxima ao comportamento real das aeronaves;
- ambiente de simulação e visualização tridimensional;
- possibilidade de criação de novos veículos, cenários e estratégias de planejamento;
- realismo na simulação de efeitos atmosféricos como rajadas de vento, neblina e térmicas.

3.2 Microsoft Flight Simulator X[®]

Com mais de 25 anos de história, o *Microsoft Flight Simulator*[®] é um dos simuladores de voo mais famosos do mercado. O seu tipo de licença é comercial e custa por volta de R\$140,00. Atualmente encontra-se na versão 10 (também chamada de *MsFS X*[®]) e possui uma grande comunidade de pilotos e controladores de voo que o utilizam.

Através da sua *API* (*Application Programming Interface*) é possível desenvolver programas que executam tanto dentro como fora do simulador. Além disso, é possível inserir/alterar aviões, painéis de instrumentos, cenários, informações meteorológicas, entre outras características. As possibilidades de customização e a qualidade gráfica podem ser apontadas como os principais motivos do sucesso do simulador. A *API* nativa do *MsFS X*[®] é conhecida como *SimConnect*, e é através dela que se torna possível o intercâmbio de dados entre o simulador e outros programas sendo executados externamente.

A dinâmica de voo das aeronaves é baseada em uma abordagem paramétrica. Um conjunto de tabelas aerodinâmicas determina como o avião se comporta sob certas condições. Outras *API's* desenvolvidas por terceiros, são a *JSimConnect* (livre, em Java) e a *FSUIPC* (comercial). Esta última está disponível desde a versão 6 do *MsFS*[®] e ambas possuem a mesma finalidade daquela citada anteriormente.

3.3 Flightgear Flight Simulator

O *FgFS* (Olson, 2007) é um simulador de voo de código-aberto com diversas funcionalidades. A mais importante delas é a possibilidade de se estabelecer canais de comunicação via rede de instâncias do *FgFS* com outros processos, rodando em diferentes máquinas. É possível, por exemplo, estabelecer o controle de navegação de aeronaves do simulador a partir de diferentes estratégias de controle programadas no computador de bordo de aeronaves reais. Outra vantagem, assim como no caso do *MsFS*[®], é a existência de diversos modelos de aeronaves, ambientes e condições atmosféricas construídas dentro do simulador, o que permite criar situações bastante semelhantes àsquelas existentes no mundo real.

A Figura 1, retirada de (Alves Neto and Campos, 2008), apresenta um exemplo de arquitetura de comunicação em que foi estabelecida a implementação de um sistema *HWIL* entre esse simulador de voo, um computador de bordo de uma aeronave real, e um sistema servo-visual de processamento de imagens, que na prática, representava um computador de capacidade moderada.

O *FgFS* fornece sinais adquiridos dos sensores da aeronave para o módulo de controle, via interface de rede. Em outro canal, do mesmo protocolo, o computador de bordo envia sinais de

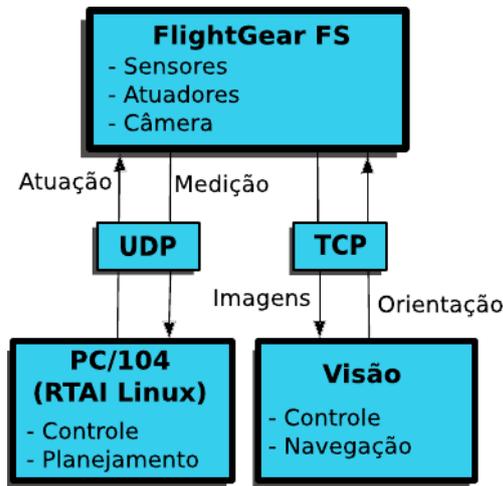


Figura 1: Estrutura de comunicação entre um computador de bordo e o *FgFS*.

controle para o veículo instanciado no simulador. Esses sinais são determinados em função de alguma missão específica, elaborada no módulo de planejamento. Paralelamente, o simulador envia imagens de uma câmera virtual, através de outro canal de comunicação, para serem processadas no módulo servo-visual. Outras informações de controle fazem o caminho reverso, guiando a aeronave conforme as tarefas determinadas.

3.4 Utilização do MsFS[®] em HWIL

Nas versões anteriores ao *MsFS X*[®] (2004, 2002, 2000, 98 e 95) é utilizado um mecanismo baseado em *IPC* (*Inter-process communication*) para recuperar/alterar dados no simulador. Basicamente, os módulos são *DLLs* (*Dynamic-link library*) que obedecem a uma determinada interface sendo executadas dentro do espaço de memória do simulador. Para facilitar esse processo, foram desenvolvidas (por terceiros e para fins comerciais) as bibliotecas *FSUIPC* e *WideFS* que tem como objetivo principal permitir o desenvolvimento de customizações para esse simulador, além de possibilitar a execução dos módulos criados em outra máquina, que não seja a que está executando o simulador. A *WideFS* se comunica através do protocolo de rede *TCP/IP*.

Já na versão *X*, a *Microsoft* desenvolveu uma nova *API* para acesso ao simulador, chamada *Sim-Connect*. Uma das mudanças principais foi a forma de acesso ao simulador. Para esse caso, a comunicação é realizada por meio do protocolo *TCP/IP*. Essa *API* funciona como uma interface de comunicação entre o simulador de voo e, por exemplo, um computador de bordo no qual se deseje executar algoritmos de controle para as aeronaves virtuais.

Na Figura 2 é possível observar como é estabelecido o sistema *HWIL* no caso da utilização do *MsFS X*[®] como simulador. Fazendo um para-

lelo entre essa arquitetura e aquela apresentada na Figura 1 percebe-se que toda a comunicação acontece em duas vias, uma por onde o computador de controle recebe informações advindas do simulador (como sensores da aeronave, condições de voo, imagens) e outra pela qual são enviadas informações ao simulador para controle da aeronave.

3.5 Análise comparativa

Para realizar a análise comparativa entre os dois simuladores, optou-se por uma abordagem qualitativa. Foram inicialmente selecionadas algumas características desejáveis de simulação para a implementação do sistema *HWIL*. Essas características descrevem cada simulador de maneira mais genérica, sem se aprofundar em questões como estrutura do código ou formas de alterações do mesmo.

A cada uma dessas características foi atribuída um conceito variando entre “baixo” e “alto” que exprime a qualidade de cada simulador naquele ponto específico. Esses conceitos foram definidos em um consenso entre os autores, em função das experiências adquiridas durante vários experimentos.

O custo financeiro foi um dos critérios avaliados embora seja considerado como uma característica de menor importância. Mais importante porém é a facilidade de utilização do sistema que representa, entre outras coisas, a complexidade de criar a comunicação entre computador de controle e o simulador. A portabilidade nos diz se o simulador é ou não multi-plataforma (*Windows, Unix, MacOS*), enquanto a qualidade gráfica da simulação traz benefícios tanto no monitoramento do processo quanto no processamento de imagens para navegação.

Outros critérios de suma importância são a frequência de simulação, necessária para implementar sistemas de controle mais estáveis; a possibilidade de construção de aeronaves no simulador, que não sejam aquelas pré-existentes; o custo reduzido na simulação de múltiplos veículos para missões de coordenação; a extração de imagens para missões baseadas em processamento visual; e, por fim, a documentação de utilização do software. Na sequência são apresentados e discutidos os resultados finais obtidos com essa abordagem.

O item simulação multi-veículos representa a capacidade que o simulador possui em simular e controlar dois ou mais veículos ao mesmo tempo. O *MsFS X*[®] permite esse controle, porém, com limitações significativas. O vento, por exemplo, é aplicado apenas no avião principal da simulação. A solução para ambos os simuladores é executar várias instâncias em computadores distintos e fazê-las se comunicar via rede.

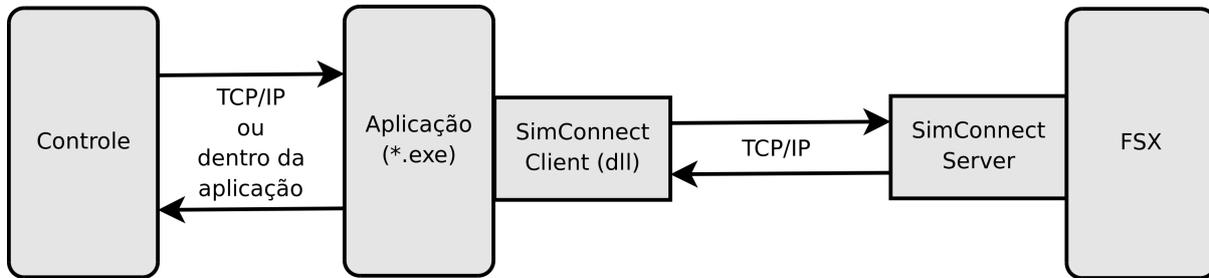


Figura 2: Arquitetura de comunicação do sistema *HWIL* utilizando o simulador *MsFS*[®].

4 Resultados

De acordo com as características de análise estabelecidas na seção anterior foi composta uma tabela onde foram explicitados os valores atribuídos para cada um dos simuladores. Esses valores foram escolhidos de forma empírica e subjetiva, em termos das tecnologias existentes até a presente data. O resultado final pode ser visto na Tabela 1.

De acordo com os critérios estabelecidos, constatou-se que o *FgFS* ofereceu mais vantagens do que o *MsFS*[®]. Entretanto, de uma maneira menos genérica, pode-se considerar que a escolha do simulador depende muito da aplicação.

Apenas para fins de comparação entre as duas arquiteturas *HWIL* implementadas, foram estabelecidos dois experimentos para controle de navegação utilizando uma máquina externa. Para isso foram utilizadas as representações de cada simulador para a aeronave *Piper J3-Cub*, ambas sujeitas ao mesmo algoritmo de controle proporcional, integral e derivativo (PID).

Primeiramente foi realizado um teste de controle de altitude. Tomando-se como base a velocidade de voo de aproximadamente 70 nós e uma altitude inicial de 1500 pés, aplicou-se um sinal de referencial de 2000 pés na entrada do controlador. Conforme é possível observar na Figura 3, o comportamento dinâmico dos dois modelos do mesmo avião foram bem semelhantes quando submetidos ao mesmo controlador, embora o *overshoot* e o tempo de assentamento no caso do *MsFS X*[®] tenham sido um pouco maior. Essa diferença se deve à frequência de simulação, à dinâmica de voo e aos parâmetros de cada modelo dos diferentes simuladores.

No outro experimento a mesma aeronave foi utilizada para comparação de resposta ao controle de orientação. Tomando como referência a mesma velocidade de 70 nós, a uma altitude 1500 pés, e orientação lateral (guinada) de 280 graus, utilizou-se outro controlador PID para orientar a aeronave no espaço em relação ao Norte magnético da Terra (zero grau). Foi estabelecido um comando de referência de 100 graus na entrada do controlador, cuja resposta pode ser visualizada na Figura 4.

Aqui é possível notar que os comportamentos dos dois simuladores frente ao mesmo controlador

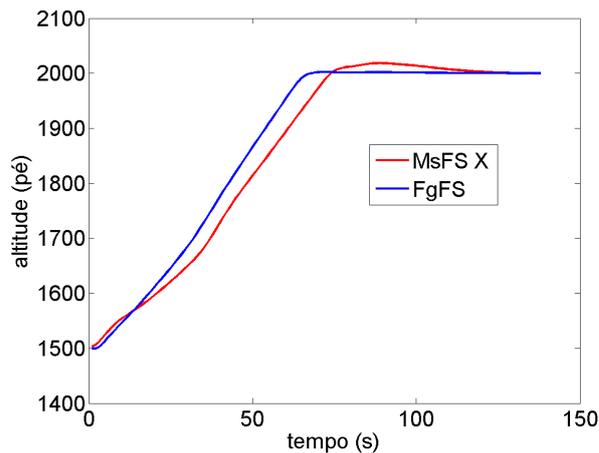


Figura 3: Controle de altitude utilizando o modelo *Piper J3-Cub* dos dois simuladores e um controlador do tipo PID.

foram muito semelhantes, mostrando que ambos poderiam ser utilizados para esse tipo de teste com essa aeronave.

5 Conclusões e Trabalhos Futuros

Este trabalho apresentou uma comparação entre duas ferramentas de simulação, o *MsFS X*[®] e *FgFS*, relativa a implementação de um sistema *HWIL*. Dados os vários critérios de avaliação descritos anteriormente constatou-se que o *Flightgear Flight Simulator* constitui uma ferramenta mais vantajosa, uma vez que possui código aberto, é gratuita mas, principalmente, permite simulações dinâmicas com altas frequências.

Constatou-se experimentalmente que a frequência de simulação do *MsFS X*[®] está diretamente ligada à taxa de atualização gráfica da tela, como um jogo em que a cada iteração do laço principal todas essas tarefas (e outras mais) são realizadas. Isso é prejudicial a medida que incorpora ao *MsFS X*[®] a necessidade de uma máquina poderosa para a simulação (já que a sua qualidade gráfica é superior). Já no *FgFS*, essas duas tarefas acontecem separadamente, de modo que é possível aumentar consideravelmente a frequência de simulação aerodinâmica (para

Tabela 1: Comparação entre os simuladores *MsFS X*[®] e *FgFS*.

| Característica | <i>FgFS</i> | <i>MsFS</i> [®] |
|--|-------------|--------------------------|
| Custo da ferramenta | Baixo | Médio |
| Facilidade de utilização do simulador para <i>HWIL</i> | Alta | Baixa |
| Portabilidade | Alta | Baixa |
| Qualidade gráfica da simulação | Baixa | Alta |
| Frequência de simulação | Alta | Média |
| Construção de aeronaves | Baixa | Média |
| Simulação multi-veículos | Baixa | Baixa |
| Extração de imagens para controle servo-visual | Baixa | Baixa |
| Documentação | Alta | Média |

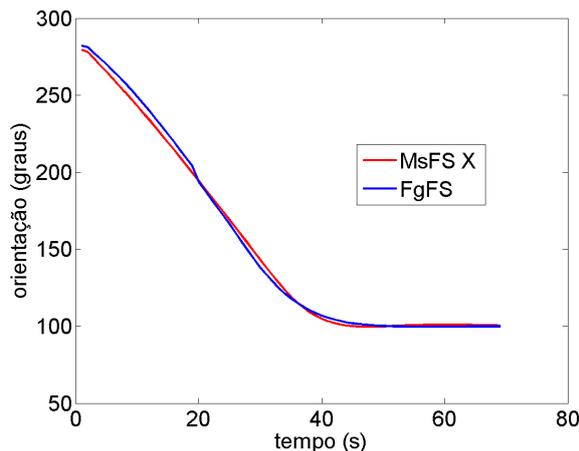


Figura 4: Controle de orientação (guinada) utilizando o modelo *Piper J3-Cub* dos dois simuladores e um controlador do tipo PID.

centenas de *hertz*) em detrimento da atualização gráfica.

Nos próximos trabalhos espera-se ampliar o número de simuladores considerados, incorporando inclusive ferramentas não dedicadas a treinamento de pilotos. Espera-se poder ampliar os resultados obtidos, utilizando uma metodologia com direcionamento mais técnico e quantitativo e, assim, tentar identificar quais as melhores ferramentas para cada tipo de missão.

Agradecimentos

Os autores agradecem o apoio financeiro oferecido pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (Capes) e da Fundação de Amparo a Pesquisa de Minas Gerais (Fapemig).

Referências

Adiprawita, W., Ahmad, A. S. and Sembiring, J. (2007). Hardware in the loop simulation for simple low cost autonomous uav (unmanned aerial vehicle) autopilot system research and development, *International Conference*

on Electrical Engineering and Informatics (ICEEI'07).

Alves Neto, A. and Campos, M. F. M. (2008). Implementação de um sistema hardware-in-the-loop para veículos aéreos autônomos não-tripulados, *Congresso Brasileiro de Automática (CBA'08)*, Juiz de Fora, MG, Brazil.

Astuti, G. (2007). *Unmanned Aerial Vehicles: State of the Art and the Road to Autonomy, Intelligent Systems, Control, and Automation: Science and Engineering*, Vol. 33, Springer, chapter Hardware in the Loop Tuning for a Volcanic Gas Sampling UAV, pp. 485–505.

Castillo-Effen, M., Alvis, W. and Castillo, C. (2005). Modeling and visualization of multiple autonomous heterogeneous vehicles, *U.S. Navy Gran*, Vol. 3, pp. 2001–2007.

Gholkar, A., Isaacs, A. and Arya, H. (2004). Hardware-in-loop simulator for mini aerial vehicle, *Technical Report 400076*, Department of Aerospace Engineering, IIT Bombay, Powai, Mumbai, India.

Göktogian, A. H., Nettleton, E., Ridley, M. and Sukkarieh, S. (2003). Real time multi-uav simulator, *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Vol. 2, Taipei, Taiwan, pp. 2720–2726.

How, J., King, E. and Kuwata, Y. (2004). Flight demonstrations of cooperative control for uav teams, *Proceedings of the 2004 AIAA Unmanned Unlimited Technical Conference, Workshop and Exhibit*.

Olson, C. (2007). Flightgear flight simulator. <http://www.flightgear.org> (Acesso em 12 de Agosto de 2007).

Sorton, E. F. and Hammaker, S. (2005). Simulated flight testing of an autonomous unmanned aerial vehicle using flightgear, *American Institute of Aeronautics and Astronautics, Institute for Scientific Research, Inc.*, Fairmont, West Virginia.