

SISTEMA *HARDWARE IN THE LOOP* PARA TESTE DE DETERMINAÇÃO DE ATITUDE DE UM PILOTO AUTOMÁTICO

Lorena Alves dos Santos

Instituto Tecnológico de Aeronáutica
Praça Marechal Eduardo Gomes, 50 - Vila das Acácias.
São José dos Campos, SP - 12228-900.
lorena@ita.br

Neusa Maria Franco de Oliveira

Instituto Tecnológico de Aeronáutica
Praça Marechal Eduardo Gomes, 50 - Vila das Acácias.
São José dos Campos, SP - 12228-900.
neusa@ita.br

Resumo

O trabalho propõe o desenvolvimento de um sistema *Hardware in the Loop* para um piloto automático, de modo que ele mantenha a atitude da aeronave simulada. O piloto automático deverá estar sobre uma plataforma gímbal que simula os movimentos de *pitch* e *roll* da aeronave. Para realização dos movimentos da aeronave no simulador, dados de navegação são obtidos do simulador e do piloto automático. O simulador de voo executa o movimento da aeronave e envia os dados de atitude para plataforma realizar os movimentos do simulador, gerando um novo dado de atitude no piloto automático.

Palavras-Chaves: *Hardware in the Loop*, Piloto Automático, Simulador de vôo, Veículo Aéreo não tripulado.

Abstract

This article considers the development of a system the hardware In The Loop of an autopilot who keeps the attitude of a simulated aircraft. The autopilot is located on a platform gimballed that simulates the movements of pitch and roll of the aircraft. For accomplishment of the movements of the aircraft in the simulator, navigation data will be gotten of the proper simulator and the autopilot. The flight simulator executes the movement of the aircraft and sends given of attitude for platform gimballed to carry through the movements received from the flight simulator, generating new given of new data of attitude in the autopilot.

Keywords: Hardware in the Loop, Autopilot, Flight Simulator, Unnamed Aerial Vehicle.

1. INTRODUÇÃO

Veículo Aéreo Não Tripulado (VANT) é um tipo de aeronave em que não há necessidade de um piloto embarcado para sua guagem (2). Essas aeronaves podem ser controladas a distância, porém o uso de controle autônomo nestas aeronaves modernas tem sido bastante empregado, especialmente em casos que envolvem o risco de vidas humanas (7).

Um VANT autônomo necessita de um Piloto Automático (PA) para controle da aeronave em uma rota pré-estabelecida sem intervenção humana (11).

Os pilotos automáticos utilizam Unidades de Medidas Inerciais (UMI) para obter informações de atitude da aeronave. A determinação da atitude de uma aeronave durante o vôo não é uma tarefa simples e o erro na sua determinação pode causar a queda da aeronave. Assim, o desenvolvimento de um sistema HIL para realização destes testes antes de um vôo experimental é de grande validade para reduzir os riscos de queda do aeromodelo.

A simulação HIL é uma técnica que permite realizar uma representação rápida do modelo virtual de um sistema, em um protótipo físico através de sistemas embarcados, com o intuito de permitir a realização de testes mais próximos com a realidade (5). O conceito de HIL é bastante útil em indústrias automotivas e espaciais, já que sistemas complexos são difíceis de modelar. Este conceito proporciona uma grande confiabilidade aos resultados e também uma diminuição do tempo no desenvolvimento de projetos, sem precisar de um elevado orçamento ou protótipos elaborados para realização dos testes (3).

Em (4), cujo objetivo é reduzir erros da UMI, são apresentados dois sistemas independentes. O primeiro gera sinais necessários para executar os motores de passo e um segundo sistema que coleta dados do piloto automático. No primeiro sistema utiliza-se um microcontrolador que envia sinais para um *Motor Driver*, que fornece sinais para os motores de passo da plataforma gimbal. O segundo sistema manda um pacote de dados do piloto automático para uma estação de solo, em seguida a estação de solo manda para um computador de base os dados de atitude do piloto automático para que o gimbal realize os movimentos obtidos através do piloto automático. O computador de base está ligado aos *encoders* da plataforma gimbal. Com o desenvolvimento nesta plataforma, o desempenho da UMI neste trabalho obteve uma redução de 38% dos erros de *roll* e 36% dos erros de *pitch*.

Em (9) é apresentada a construção de um sistema *hardware in the loop* para análise de seus movimentos que, consiste em unidades independentes: Simulador 6-DOF (Matlab/Simulink), onde a dinâmica da aeronave é representada em seis graus de liberdade, simulador de vôo, Piloto automático sobre a plataforma gimbal e uma estação de solo. Na primeira unidade tem-se a dinâmica da aeronave e definição do controle: automático ou via rádio controle. Esta unidade manda os estados da aeronave para a plataforma gimbal, através de uma comunicação serial. Assim, o piloto automático, que está sobre a plataforma gimbal, envia os dados via *wireless* para estação de solo obter o relatório e monitorar o piloto automático. Os dados são, também, enviados via serial para o sistema 6-DOF, o qual agora se comunica com o simulador de vôo através de um protocolo UDP para visualização do movimento de atitude da aeronave.

O piloto automático utilizado no trabalho é de baixo custo e, portanto, a unidade de medida inercial se torna sensível e não precisa suficiente para a navegação levando a um desvio da rota estabelecido pela missão do VANT. Para melhorar a precisão, é necessário quantificar os erros que a UMI gera (4). Neste contexto, torna-se interessante desenvolver um

sistema para *Hardware in the loop*, com o objetivo de proporcionar testes para o sistema de navegação do piloto automático, analisando os movimentos de atitude obtidos pela UMI empregada. O sistema será composto de duas unidades independentes. A primeira será um piloto automático localizado sobre uma plataforma gimbal de dois eixos, fornecendo dados de atitude via serial e o simulador de voo, X-Plane, fornecendo dados de navegação via UDP. A segunda consiste na comunicação dos dados obtidos do X-Plane com a plataforma gimbal, para que a plataforma gimbal realize os movimentos de atitude recebidos pelo X-Plane.

2. SISTEMA DE INTERESSE

Neste trabalho, chamamos de bloco de navegação o sistema responsável por disponibilizar informações sobre a posição velocidade, atitude e direção de um veículo com relação a um referencial, utilizando informações fornecidas por acelerômetros, giroscópios, sensores de pressão e GPS. Em (12), encontra-se uma abordagem maior do sistema de navegação desenvolvido.

Para testar o bloco de navegação desenvolvido do Pegasus Autopilot que é um piloto automático para aeronaves não tripuladas de asa fixa ainda em fase de desenvolvimento, idealizou-se uma plataforma HIL que fosse capaz de obter e simular dados de atitude. A plataforma de simulação é composta por um microcontrolador ARM Cortex M3 conectado a unidade de medida inercial, representando o bloco de navegação do piloto automático, o simulador de voo X-Plane, um sistema de eixos móveis composto por servomotores o qual foi chamado de gimbal, sobre este gimbal está o piloto automático (Figura 1), e um software intermediador desenvolvido na plataforma Labview para realizar a comunicação entre os hardwares e o simulador presente no ambiente necessário para realização da simulação.

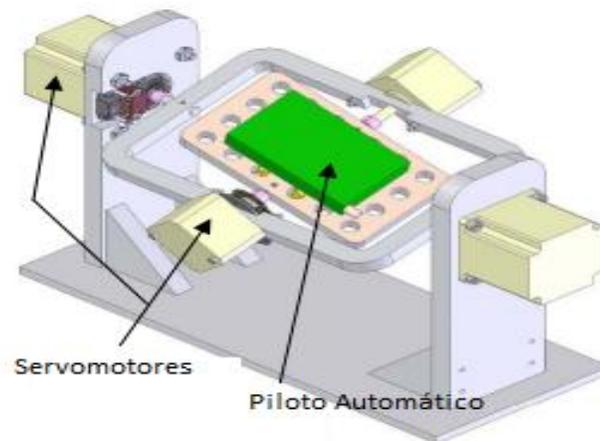


Figura 1. Sistema de eixos móveis, gimbal. Fonte: Adaptada de (3).

No sistema de interesse, o X-Plane simulará uma aeronave em voo, tendo atitude variável ao longo do tempo. As informações de atitude recebidas do X-Plane devem ser

enviadas ao sistema de eixos moveis, gimbal, de forma que este gimbal obtenha a mesma atitude da aeronave em vôo no X-Plane.

Para isso é necessário converter as informações de atitude recebidas do X-Plane, disponibilizada em graus, para um sinal PWM, que é o sinal de comando dos servos motores que irão atuar no gimbal, alterando os seus eixos e em consequência alterando também a atitude do piloto automático que estará sobre a plataforma gimbal.

O sistema de interesse descrito é apresentado na figura 2, os dados do piloto automático obtidos no Labview servem apenas como uma interface que os dados de pitch e roll possam ser visualizados pelo usuário. Os dados obtidos do X-Plane além de serem mostrados na interface do Labview, eles também são processados e enviados aos servos motores da plataforma gimbal.

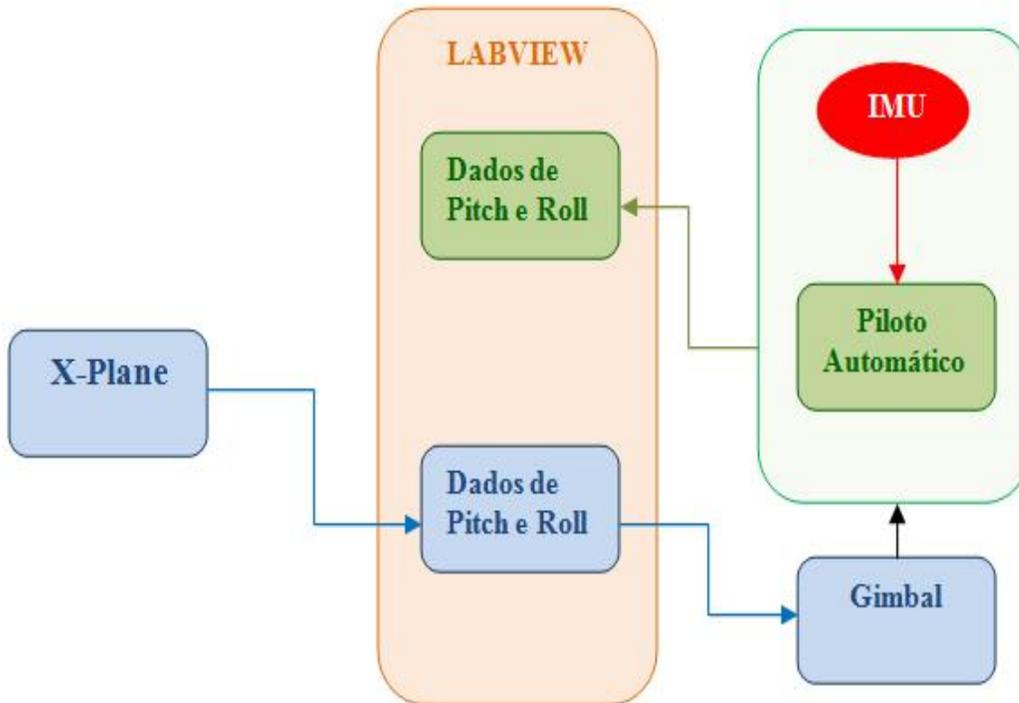


Figura 2. Sistema de interesse. Fonte: próprio autor.

3. SISTEMA DESENVOLVIDO

No sistema desenvolvido o X-Plane se comunica com o meio externo enviando um pacote de dados através do protocolo UDP. Outros métodos de comunicação do X-Plane com o meio externo podem ser visto em (8). A comunicação através do UDP é uma forma de comunicação rápida, porém sem garantia de recebimento de dado. Apesar deste problema, as simulações de controle de aeronaves não têm sofrido com esse problema (6). O fato de ter

escolhido este meio de comunicação para obter dados de vôo do X-Plane se deve a capacidade de enviar e receber dados para outros dispositivos facilitando o desenvolvimento.

Além de configurar qual o tipo de comunicação com o meio externo, é necessário ajustar a quantidade de pacotes que serão enviados a cada segundo. A taxa de envio de pacotes utilizados no trabalho é de 20 pacotes por segundos, garantindo uma atualização dos dados a cada 50 milissegundos. A capacidade de transmissão do X-Plane é de até 99.9 pacotes de dados por segundos.

Outro fator configurado no X-Plane é qual o dado que será enviado no pacote para o meio externo. O pacote de dados para um único identificador é composto de uma sequência de 41 bytes. O X-Plane utiliza variáveis no formato ponto flutuante de precisão simples para quase todas as informações enviadas através do seu barramento de comunicação em rede. Isto significa que os dados são codificados utilizando-se conjunto de 4 bytes (10). A organização do pacote de dados de entrada e saída do X-Plane e seu funcionamento total podem ser visto com uma abordagem maior em (10) e (14).

No software intermediador, desenvolvido no Labview, são tratados os dados de atitude recebido do X-Plane, os mesmos dados de atitude que serão enviados para o microcontrolador arduíno para controle dos servos motores, e por fim os dados de atitude lidos do piloto automático. A seguir são descritas as fases de desenvolvimento realizadas no software intermediador. A estrutura do software intermediador é apresentada na figura 3.

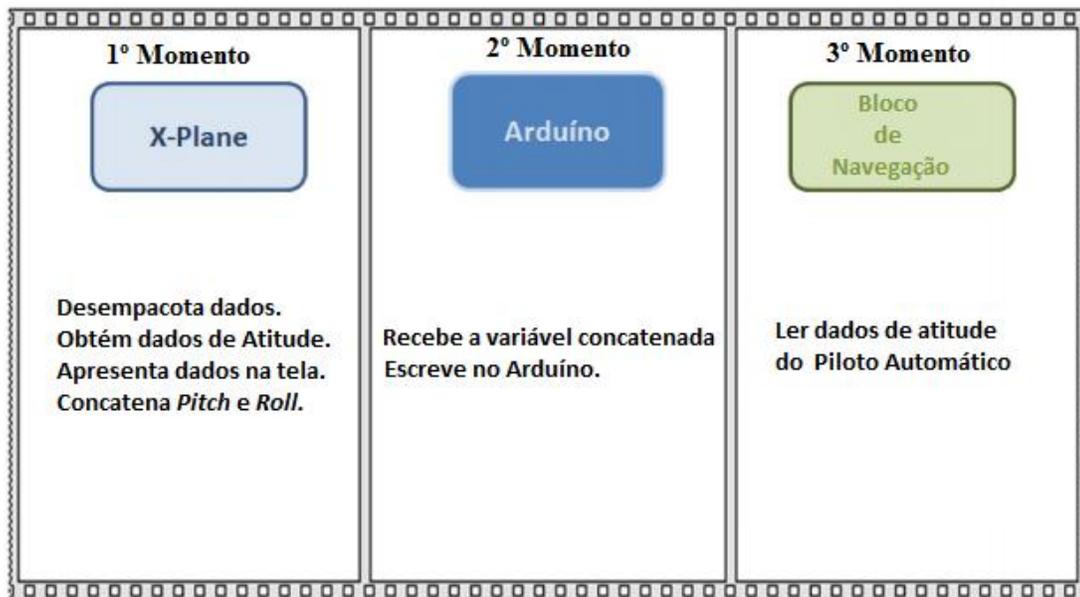


Figura 3. Estrutura do software intermediador. Fonte: próprio autor.

No primeiro momento, o pacote de dados recebido do X-Plane é desempacotado, obtendo os dados de *pitch* e *roll* da aeronave naquele instante de vôo. São apresentados na interface os valores de atitude, em seguida concatenam-se os dados para que sejam enviados ao arduíno.

No segundo momento, o dado concatenado anteriormente é escrito via comunicação serial no arduino. Após a escrita, a execução do sistema passa a ser executada no arduino,

O sistema desenvolvido para controle dos servos motores está embarcado no microcontrolador Arduino Uno. Primeiramente, são feitas as configurações de portas e pinos que serão utilizados. Em seguida é realizada a leitura dos dados vindos do Labview. Os dados são desmembrados, e em seguida convertidos para o tipo inteiro, para que o sistema reconheça o valor em grau. Uma condição também foi criada para que existam apenas dados de *pitch* e *roll* positivos por fim os valores dos ângulos são enviados para os servos motores para que eles possam realizar os movimentos na plataforma gimbal de acordo com os dados recebidos do X-Plane. A figura 4 apresenta o trecho principal do código realizado para controle dos servos. Pode-se observar que é um desenvolvimento simples, tornando-se o principal motivo de escolha para utilizar o microcontrolador arduino para este tipo de sistema.

```
void loop()
{
  char SuaString2[16] = "xxxxxxxxxxxxxxx";
  if (Serial.available()>0)
  {
    for (int i=0; i<16;i++)
    {
      SuaString[i]=Serial.read();
    }
  }
  for (int j=0; j<16;j++)
  {
    SuaString2[j]=SuaString[j];
  }
  //desconcatenar string
  pitch_s= strtok(SuaString2,";");
  roll_s= strtok(NULL, ";");
  //Converter para inteiro
  NumeroP = atoi(pitch_s);
  NumeroR = atoi(roll_s);
  if (NumeroP <0 ) { NumeroP= NumeroP * (-1);}
  if (NumeroR <0 ) { NumeroR= NumeroR * (-1);}
  //controle de servos
  myservo2.write(NumeroP);
  myservo.write(NumeroR);
  delay(100);
}
```

Figura 4. Trecho do código embarcado no arduino para controle dos servomotores. Fonte: próprio autor.

No terceiro momento, dados de atitude do bloco de navegação são lidos via serial e apresentados na interface desenvolvida.

4. DISCUSSÕES E RESULTADOS

Não foi possível obter os resultados das leituras de atitude informadas pelo bloco de navegação embarcado na plataforma gimbal, este assumindo a atitude fornecida pela X-Plane, Este descrito na seção do sistema de interesse. Isto não foi possível devido à indisponibilidade da plataforma gimbal em tempo hábil.

Daí, o teste do sistema desenvolvido foi realizado em duas etapas, apresentados na figura 5 e descritos abaixo:

1. O sistema recebe informações de atitude do X-Plane e a envia para os servos motores.
O teste já foi realizado e à medida que a aeronave varia sua atitude (*pitch* e *roll*) verifica-se que os servos variam suas posições acompanhando as informações recebidas.
2. O sistema recebe as informações de atitude lidas do sistema de determinação de atitude do bloco de navegação. Estas informações são apresentadas na tela de interface.

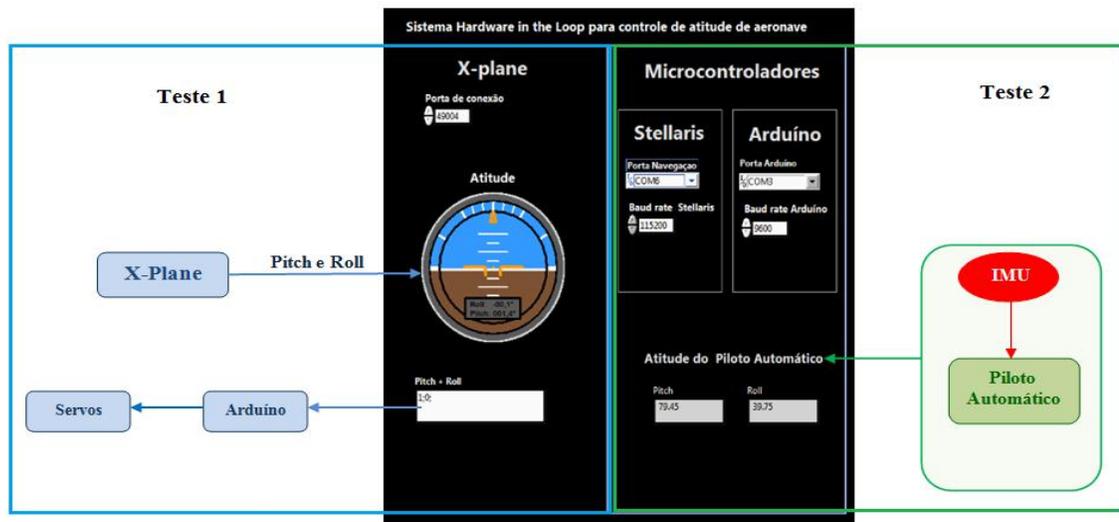


Figura 5. Diagrama de blocos apresentando os testes realizados. Fonte: próprio autor.

A figura 6 apresenta o esquema real de como foram realizados os testes. Em 1, é apresentado o X-Plane. Em 2 apresenta-se a interface desenvolvida no Labview mostrando as atitudes obtidas do piloto automático e do X-Plane. Em 3, mostra a Unidade de medida inercial conectada ao microcontrolador que está embarcado o piloto automático desenvolvido; Em 4, o microcontrolador arduíno que está recebendo dados do labview e enviando os graus de atitude recebidos para os servomotores apresentados em 5.



Figura 6. Esquema real da simulação. Fonte: própria do autor.

5. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] ALI, A; BUTT, W. A. Validation of an Autopilot through Hardware in the loop Simulation. National Engineering and Scientific Commission, Islamabad.
- [2] BARNHART, R. K et al. Introduction to Unmanned Aircraft Systems. ed. Boca Raton, FL, USA: CRC Press, 1ª ed. 2011.
- [3] CARRILLO, G, A, C. Estratégias para a correção dos efeitos de atraso de sistemas Hardware In the Loop (HIL). Dissertação (Mestrado) – Programa de Pós Graduação em Engenharia Mecânica – Universidade Estadual de Campinas, Campinas, SP, Brasil, 2012.
- [4] CARTER, L; ALI K. Hardware in the loop Simulator for IMU calibration. Department of Computer Engineering, Jackson State University, USA.
- [5] DEL SIGNORE, M. J.; KROVI, V. Virtual prototyping and hardware-in-the-loop testing for musculoskeletal system analysis, Proceedings of the IEEE international conference Mechatronics and automation, Ontario, Canada .pp. 394-399, 2005.
- [6] GARCIA R,D; BARNES L. Multi-UAV Simulator Utilizing X-Plane, IEEE J. Intelligent & Robotic Systems, vol. 57, pp. 393-406. 2009.
- [7] GARCIA, R.D.; BARNES, L. Multi-UAV- Simulator utilizing X-Plane. Journal of Intelligent and Robotic Systems, Springer, Dordrecht, Netherlands, v. 57 p.393-406, 2010.

- [8] JENIE, Y.I; INDRIYANTO, T. X-Plane-Simulink Simulation of a Pitch-Holding Automatic Control System for Boeing 747, Presented in Indonesian-Taiwan Workshop, Bandung, Indonesia. 2006.
- [9] JUNG, D; TSIOTRAS, P. Modeling and Hardware in the loop simulation for a small unmanned Aerial Vehicle. Georgia Institute of Technology, Atlanta, GA, 30332-0150.
- [10] LEWIS, J.R. X-Plane, UDP, and visual basic, for X-Plane version 9. Disponível em <http://www.jefflewis.net/XPlaneUDP_9.html>.
- [11] MCLEAN, D. Automatic Flight Control System. New York, NY, USA: Prentice Hall Corporation, 1st. ed. 1936.
- [12] PEREIRA, O M. Sistema de Navegação e Determinação de atitude inercial para aeronaves não tripuladas com base em sensores de baixo custo. Dissertação (Mestrado) – Divisão de dispositivos eletrônicos – Instituto Tecnológico de Aeronáutica, São José dos Campos, SP, Brasil, 2013.
- [13] POLINI, L; PARNENZINI V; INNOCENTI M. Distributed Real –time Hardware and Man in the loop Simulativo for the ICARO II Unmanned Systems Autopilot. Department of Electrical and Computer Engineering. University of Pisa, Via Diotisalvi, Pisa, Italy.
- [14] RIBEIRO, L.R. Plataforma de Testes para Sistemas de Piloto Automático Utilizando Matlab/Simulink e Simulador de vôo X-Plane. Dissertação (Mestrado) – Divisão de dispositivos eletrônicos – Instituto Tecnológico de Aeronáutica, São José dos Campos, SP, Brasil. 2011.
- [15] SUN, Y; JU,J; LIANG, Y. Using Virtual Instruments to Develop an Actuator-based Hardware –in-the-Loop Simulation Test-bed for Autopilot of Unmanned Aerial Vehicle. Department of Mechanical Engineering, Cheng Shiu University,. Department of Aeronautics and Astronautics, Chinese Air force Academy, Kaohsiung Count.
- [16] TITERTON, D. H; WESTON, J. L. Strapdown Inertial Navigation technology. London, UK: IEE Radar, Sonar, Navigation and Avionics Series 5, 1997.