Article

Controlador Fuzzy Para Simulador de Voo X-Plane 11

Gabriel Oliveira [1], Rafael Almaleh [2] e Vinicius Camargo [3]

[1] gabriel.luis@ufrgs.br
[2] hess.rafael95@gmail.com
[3] camargo@ufrgs.br

Data Início: 20/06/2020; Data Final: 10/07/2020;

Resumo: Neste trabalho foi desenvolvido um controlador fuzzy para o simulador de voo X-Plane 11. Após período inicial de familiarização com o simulador, foram obtidos dados de navegação compostos pelos sensores do ambiente virtual (altitude do avião) e as decisões de atuação do usuário (alteração no pitch para subir ou descer altitude). Estes dados foram gravados através de um datalogger, e utilizados, através do fuzzy c-means e outras técnicas, para gerar regras para um controlador Fuzzy. Por fim, tais regras foram testadas no simulador para avaliar se o controlador era capaz de manter a altitude do sistema, mesmo com perturbações externas, e concluiu-se que o controlador obteve sucesso em seu objetivo.

Abstract: In this project, a fuzzy controller for the X-Plane 11 flight simulator was developed. After an initial familiarization period with the simulator, navigation data composed by the sensors of the virtual environment (altitude of the plane) and the user decisions were obtained (change in pitch to increase or decrease altitude). This data was recorded using a datalogger, and then rules for a Fuzzy controller were generated, using fuzzy c-means and other techniques. Finally, these rules were tested in the simulator to verify if the controller was able to maintain the system's altitude, even with external disturbances, and it was concluded that the controller was successful in its objective.

Keywords: controlador fuzzy, simulador de voo, X-Plane, fuzzy, c-means

---

1. Introdução

Bla bla bla Bla bla bla

Bla bla bla Bla bla bla

2. Metodologia experimental

2.1. O simulador de voo X-Plane 11

O simulador de voo X-Plane 11 é uma plataforma de alta fidelidade em relação a modelos de voo, permitindo, também, uma grande interação e personalização dos utilizadores. A sua fidelidade de modelo de voo é reconhecida pela FAA, órgão máximo da aviação civil americanai. Além de apresentar um modelo de voo fidedigno, o X-Plane é a plataforma ideal para conseguir um interfaceamento simples e rápido com outras plataformas. Essa característica é extremamente desejável no contexto deste trabalho, dado o fato de que os autores tinham o propósito de realizar o sensoriamento e atuação nas condições de voo e controles, respectivamente.

O software X-Plane é desenvolvido pela LaminarResearch. A versão 11, a mais atual, foi lançada em 2016. O modelo de voo difere dos concorrentes por implementar a teoria de elementos laminaresii. Esse modelo é similar ao cálculo de elementos finitos para forças aerodinâmicas, calculando as forças

e momentos atuantes na aeronave pelo cálculo das partes individuais que compõem a aeronave. Outros simuladores utilizam tabelas preenchidas com dados empíricos, o que pode se traduzir em um comportamento fiel para aeronaves bem conhecidas e avaliadas, mas que não serve para novas aeronaves desenvolvidas ou desconhecidas – como aeronaves antigas e protótipos, por exemplo. Essas diferenças podem ser sumarizadas em um caso bem simples: uma cadeira poderia voar num simulador tradicional se fossem colocados dados fictícios em uma tabela de forças, porém jamais voaria no X-Plane dado o fato de que a estrutura não é aerodinâmica.

O X-Plane funciona com datarefs e commands. Um dataref nada mais é que uma variável do sistema. Essas variáveis compreendem desde o valor atual da frequência do rádio VHF do avião, até a altitude da aeronave. Além disso, existem outras variáveis que servem como um identificador do avião, como a distância da porta principal até o centro de gravidade. Em resumo, os datarefs servem para caracterizar aeronaves e também registrar o seu estado e os de seus sistemas. Os commands, como diz seu nome, nada mais são que comandos que podem ser enviados para o simulador. Dentre eles existem comandos operacionais, como mudar a câmera, pausar a simulação; e há também comandos no sentido aeronáutico, como abaixar flaps e recolher trem de pouso.

Existem ocasiões em que os datarefs e os commands se confundem. Para a situação do trem de pouso, existem os comandos baixar e levantar trem; e existe o dataref com a posição do trem. Dessa forma, nota-se que um comando representa uma ação, com os datarefs sendo o estado resultante. Nada disso impede que possa se realizar a escrita em certos datarefs. Caso uma falha no trem ocorresse, poderia-se injetar um valor correspondente ao meio do caminho da sua excursão. Para os comandos de voo, pode-se escrever na variável correspondente à posição do joystick nos seus 3 eixos, correspondendo ao leme, aileron e estabilizador horizontal. O leme atua na guinada, o aileron na rolagem e o estabilizador na arfagem, conforme Figura 1.

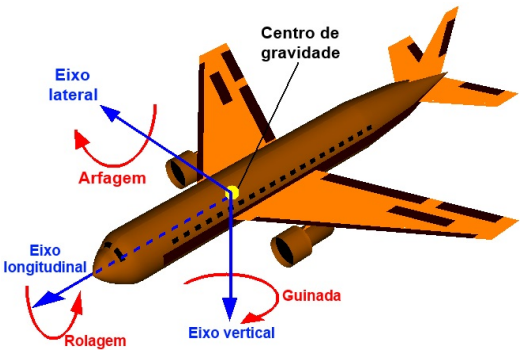

Figure 1. Eixos e momentos presentes em uma aeronave.
Fonte: http://canalpiloto.com.br/teorias-rotativas-05

Para realizar o controle do ângulo de arfagem, deve-se atuar no joystick para frente e para trás, variando o estabilizador. De modo a realizar esse comando, os autores seguiram a abordagem de escritura no dataref da arfagem do joystick via mensagens UDP, dado o caráter simples, eficaz e rápido da implementação. O dataref em questão é o sim/controls/joystick/yoke$_p itch_r atio, variando de -1 a 1, onde valores negativos indicam nariz para baixo e valores positivos nariz para cima$.

2.2. Aquisição de dados para o controlador

Os dados de navegação foram obtidos através da utilização da biblioteca socket. Após a definição dos datarefs e commands a serem enviados e recebidos pelo X-Plane, foi iniciado uma sequência de voos de treino, nos quais submeteu-se o avião a estímulos externos que alterariam seu ângulo de ataque, seja pela definição manual no dataref correspondente ou pela simulação de eventos imprevisíveis (como por exemplo, uma tempestade). Foram então coletados a posição atual do pitch da aeronave e a ação do usuário que corresponderia à normalização do sistema para o compensamento do ângulo de

ataque. A derivada do erro foi obtida utilizando a diferença entre os erros subsequentes. Os dados obtidos seguem a forma apresentada na Tabela 1.

Table 1. Exemplo de dados capturados do simulador X-Plane 11

ErroPitch	dErroPitch	CmdPitch
-28,74195	-3,30541	0,30054
-30,51777	-1,77582	0,30645
-34,32018	-3,80241	0,42801
-38,45888	-4,13870	0,43243
-42,94374	-4,48486	0,44581

2.3. Treinamento do Controlador

Através da utilização da biblioteca *skfuzzy*, foi desenvolvido um sistema de controle baseado nos dados coletados. O método *skfuzzy.cluster.cmeans* foi utilizado para criar os clusters do sistema, relacionando o erro e a derivada do erro do ângulo de ataque, com a variação da posição do *stick* controlador do *pitch*. Os clusters obtidos estão presentes na figura 2.

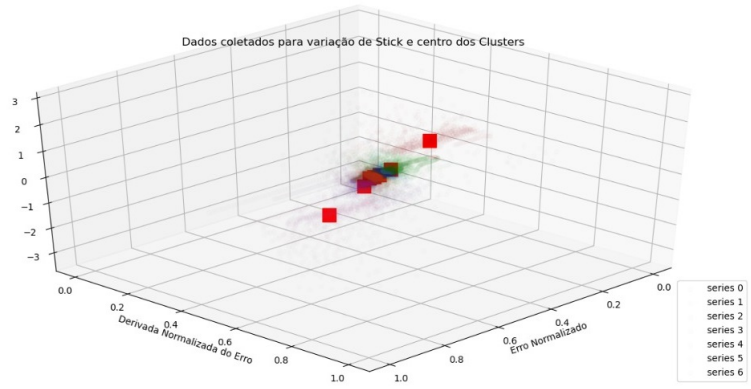

Figure 2. Clusters obtidos no sistema.

A fim de criar uma transição mais suave no sistema, foram utilizadas funções de pertinência sigmóide e gaussiana criadas pelos autores, substituindo as funções tradicionais trapezoidal e triangular presentes na biblioteca *skfuzzy*. Os centros dos clusters foram projetados e ordenados em ordem crescente no eixo Z, e tais pontos dos centros e serviram de base para criar as funções de pertinência, sendo utilizados como o ponto máximo ou o ponto de descida/subida nos extremos, conforme cada caso. Isso está demonstrado na figura 3.

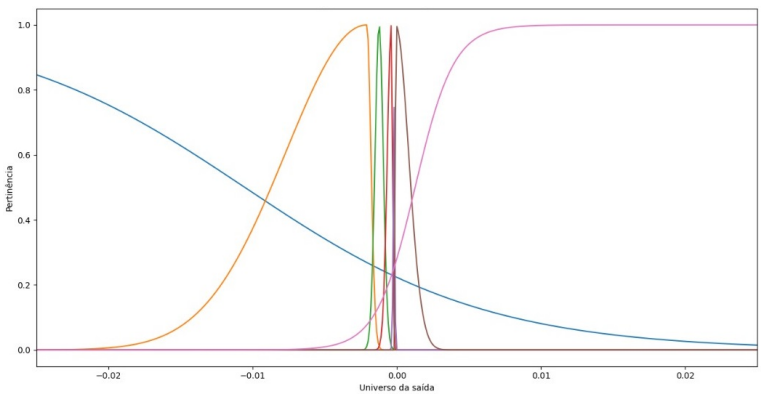

Figure 3. Função de pertinência para a sáida do sistema.

O conjunto dos pontos das combinações das entradas e a saída do sistema resultante está mostrado na superfície de saída presente na figura 4. A superfície resultante foi considerada satisfatória, pois pode-se notar que quando ambos o erro e a derivada do erro são altos, o stick de saída é posicionado com um valor negativo alto. O oposto é verificado para valores baixos de erro e derivada do erro.

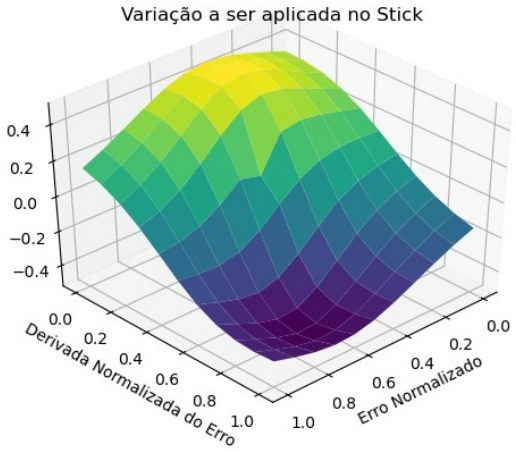

Figure 4. Saídas do sistema de controle resultante

3. Resultados obtidos com o Controlador Fuzzy

Utilizando uma conexão socket com o simulador, os dados de navegação foram transmitidos em tempo real para o controlador. Para cada dado, o controlador verifica a pertinência do dado em relação a cada cluster anteriormente definido, e define a saída defuzzificada do sistema a ser enviada para a entrada do simulador.

Para o teste inicial, propôs-se manter um ângulo de ataque fixo em 10 graus. Em seguida, o operador inseriu duas perturbações no sistema: Primeiramente, o ângulo de ataque foi aumentado para aproximadamente 20 graus, e, após a mormalização do sistema, o ângulo de ataque foi dimínuído para aproximadamente -20 graus. Nota-se, através da figura 5, que o sistema retornou com sucesso à posição inicialmente fixada para o ângulo de ataque em ambos os casos, conforme esperado.

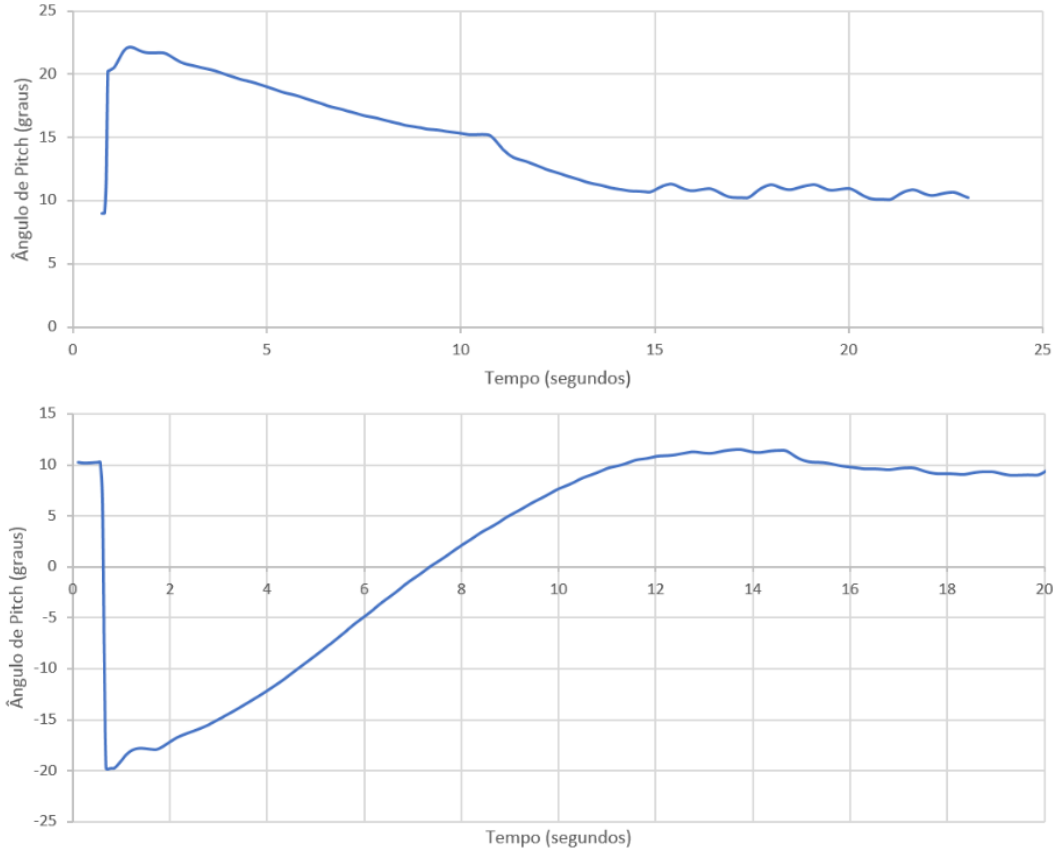

Figure 5. (**a**)Atuação do controlador para uma perturbação positiva no ângulo de ataque.(**b**)Atuação do controlador para uma perturbação negativa no ângulo de ataque.

4. Comparação dos resultados com Controlador Proporcional

Para fins de validação do controlador fuzzy, foi feita a comparação dos resultados do mesmo utilizando um controlador proporcional. Este controlador foi idealizado utilizando a fórmula 1, de forma a escalonar o erro do ângulo, que varia de 0 a 1, para o universo de saída da posição do stick do ângulo de ataque, que varia de -1 a 1. Os resultados obtidos estão presentes na figura 6.

$$stickpos = -1 * (2 * Errodongulodeataque - 1) \quad (1)$$

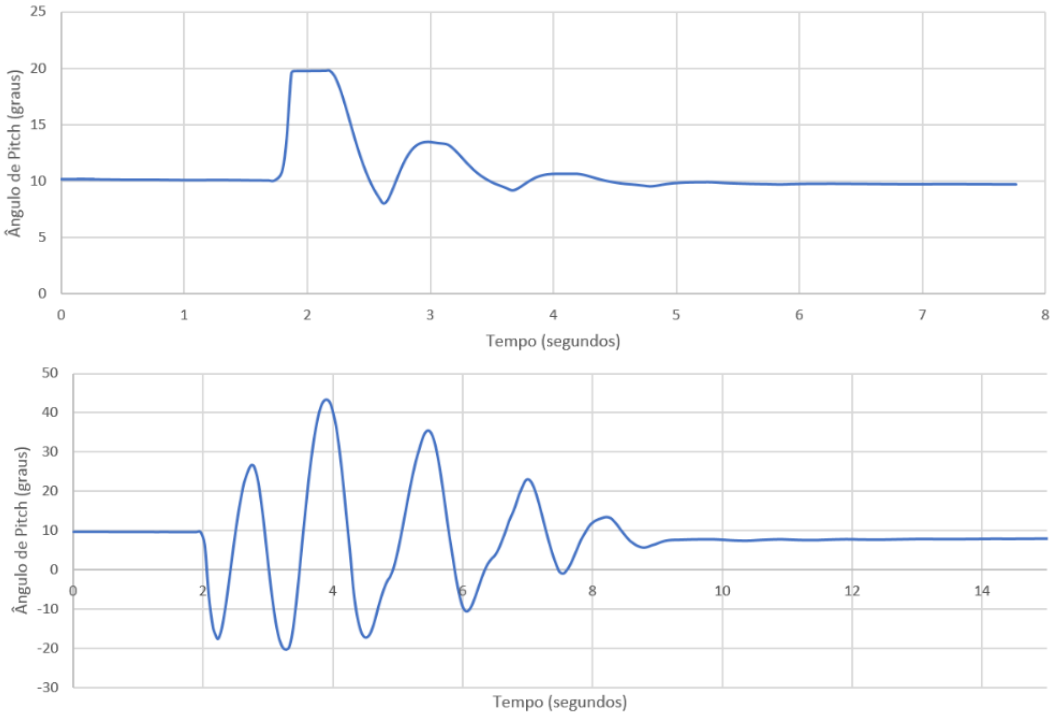

Figure 6. (**a**)Atuação do controlador para uma perturbação positiva no ângulo de ataque.(**b**)Atuação do controlador para uma perturbação negativa no ângulo de ataque.

Através da comparação entre o controlador fuzzy e o controlador proporcional, conclui-se que o controlador fuzzy possui uma resposta mais suave e previsível. Nota-se que o controlador proporcional obteve resultados mais rápidos do que o controlador fuzzy, porém apresentou um *overshoot* indesejado para valores negativos de ângulo de ataque. Isso ocorre pois a aeronave tem uma tendência natural a subir, e isso, combinado com o comando de colocar o stick pra cima, potencializa o comando de subida através de uma realimentação positiva.

5. Conclusão

BLABLABLABLABLABLABLABLABLABLABLABLABLABLABLABALBALBALBALBALBALBALBA

References

1. Author1, T. The title of the cited article. *Journal Abbreviation* **2008**, *10*, 142–149.
2. Author2, L. The title of the cited contribution. In *The Book Title*; Editor1, F., Editor2, A., Eds.; Publishing House: City, Country, 2007; pp. 32–58.