

# 1 Trabalhos Relacionados

Na literatura, diversos são os exemplos de sistemas não lineares para os quais se desenvolvem diferentes tipos de controladores. Dentre estes tipos, encontramos os PD (Proporcional-Derivativo), PID (Proporcional-Integral-Derivativo), LQ (Linear-Quadrático), Fuzzy e ANFIS dentre outros sendo que este último tem se mostrado bastante presente para tentar atuar sobre diferentes tipos de sistemas.

A grande quantidade de materiais propondo a implementação de controladores neuro-fuzzy se deve, em parte, ao fato de geralmente, para efetuar tal controle, não ser necessário se fazer uma modelagem matemática aprofundada do sistema que, em diversos casos, é de grande complexidade e, por isso, demandaria um grande esforço, sendo que em muitos casos os resultados não seriam tão bons quanto os desejados, além de oferecer ainda uma boa robustez ao controle.

Além do próprio controle de estabilidade de quadricópteros, podem-se encontrar exemplos de diversos sistemas que implementam controladores neuro-fuzzy. É o caso da implementação de um controlador para *see and avoid*<sup>1</sup> (referente ao desvio de ocasionais obstáculos) de um *drone*, como se pode encontrar em [Olivares-Mendez et al. \(2012\)](#). Já [Guo et al. \(2003\)](#), propõem a implementação de um controlador neuro-fuzzy para controlar a estabilidade sobre o balanço de um barco. Os autores mostraram, ainda, que o controlador neuro-fuzzy obteve melhor resultado do que um controlador PID, chegando à conclusão de que controladores neuro-fuzzy são promissores para efetuar tal controle.

A grande gama de opções de controle sobre quadricópteros pode ser dividida em dois grupos principais: o controle *tradicional* e o controle *inteligente*. O primeiro diz respeito ao uso de técnicas consolidadas para controle baseado na atuação sobre eles a partir do conhecimento de sua modelagem, ao passo que o segundo se refere às técnicas de controle que envolvem componentes de Inteligência Computacional, agregando, por exemplo, representação de estados através de variáveis linguísticas e capacidade de aprendizagem.

## 1.1 Controle Tradicional de Quadricópteros

Esta seção aborda diferentes propostas de controladores tradicionais para quadricópteros visando a contextualização do estado da arte para depois se poder comparar alguns resultados das técnicas tradicionais e inteligentes de controle para esse sistema.

---

<sup>1</sup> Ver e evitar (tradução nossa)

Em (RAZINKOVA et al., 2014), foi proposto um controlador PD para a descrição adequada das trajetórias definidas de um quadricóptero *Indoor*. Para permitir um funcionamento adequado *Outdoor*, foi integrado ao controlador PD um controle adaptativo, adicionando termos ao controlador convencional, de forma a permitir o ajuste correto da trajetória do quadricóptero quando submetido a distúrbios nos eixos X e Y (plano horizontal). Os resultados mostraram que o controlador adaptativo melhorou consideravelmente o controle de trajetória do *drone*, reduzindo em 64% o erro de posicionamento para os casos de trajetória em linha reta ao longo do plano XY, e em 72% para os casos de trajetória circular sobre o plano XY. Segundo os autores, este resultado representa uma melhora significativa, uma vez que o controlador resultante possui uma arquitetura simples e não requer computação extensiva, o que é indesejado para qualquer controle de sistema, especialmente para UAVs, devido à sua limitação de bateria.

Em (MUSTAPA et al., 2014) é proposto um controlador PID para efetuar o controle visando a estabilidade da altitude de um quadricóptero. Neste trabalho, foi utilizado um modelo matemático desenvolvido previamente para descrever o comportamento do sistema. Para ajustar os parâmetros do controlador PID, foi necessário realizar um experimento para determinar o momento de inércia do *drone*. Uma vez levantados os valores necessários, um controlador PID foi ajustado e se mostrou eficiente. A simulação do sistema foi realizada no Matlab® Simulink®.

Khatoon et al. (2014) também propuseram um controlador PID para controlar a estabilidade em altitude de um *drone*, mantendo a posição no eixo XY constante, mesmo esta altitude sendo uma variável muito sensível a mudanças em outros parâmetros. No trabalho, é mostrado que um controlador PID sozinho é capaz de exercer tal controle com robustez. A escolha deste controlador se deu graças à sua robustez e facilidade de modelagem. Entretanto, é também apontado pelos autores que, apesar da simplicidade para se modelar um controlador PID, isto requer uma modelagem do sistema como um todo, o que não é fácil, devido à sua estrutura complexa, suas dinâmicas não lineares e sua natureza subatuada. O sistema modelado para o *drone* mostrou ser altamente instável, justificando a necessidade de um controlador. A partir de extensivas simulações no Matlab® Simulink®, o sistema desenvolvido se mostrou bem sucedido, implementando de fato um controle robusto de altitude para um helicóptero quadricóptero.

Além desses controladores que aplicam técnicas tradicionais, são também vários os exemplos de trabalhos que exploram o controle inteligente de quadricópteros.

## 1.2 Controle Inteligente de Quadricópteros

Além das técnicas de controle tradicionais, encontram-se também na literatura diversas abordagens para a construção de controladores de estabilidade para *drones*

utilizando técnicas da Inteligência Computacional. As propostas vão do uso de algoritmos de otimização (e.g. algoritmos genéticos, PSO) ao uso de sistemas neuro-fuzzy expandido para o espaço dos números complexos.

Em (REZAZADEH et al., 2013), é proposto um controlador neuro-fuzzy (ANFIS) para o controle estabilizando a altitude de um quadricóptero. O módulo *fuzzy* acoplado a um controlador PID é usado para controlar a atitude. Os controladores *fuzzy* atualizam os ganhos do PID *online* para produzir uma resposta apropriada. A saída é calculada diretamente pelo ajuste dos pesos das entradas de acordo com as regras de inferência *fuzzy*. Estas regras, que são a base de conhecimento, são determinadas por um algoritmo computacional baseado em redes neurais (REZAZADEH et al., 2013). O controlador proposto foi comparado a um simples controlador PID e, a partir de simulações, foi mostrado que se obteve melhor resposta, por exemplo, tendo em vista o tempo de assentamento e a sobrelevação. Outro resultado obtido foi de que o controlador ANFIS é capaz de estabilizar o sistema quando exposto a perturbações externas.

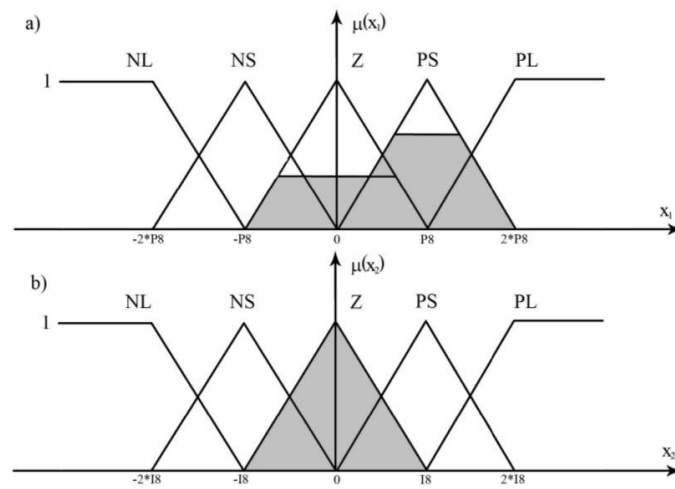
Em (MAHFOUZ et al., 2013), foi proposto um controlador parecido. Neste trabalho, também foi sugerido um controlador ANFIS que, juntamente a um algoritmo genético, ajusta os pesos de um controlador PID. Mais uma vez, o controlador desenvolvido obteve sucesso, sendo apto a garantir a estabilidade de um drone mesmo quando sujeito a condições externas. No trabalho, foi mostrado que a sobrelevação do controlador ANFIS é menor do que a do PID. Além disto, o erro do estado estável no caso do ANFIS é zero, enquanto o erro do estado estável no caso do controlador PID não o é. Uma conclusão muito importante a que se chega é que, do ponto de vista de robustez, os dois controladores são efetivos; do ponto de vista de performance, o ANFIS é melhor que o PID.

Em (MAJ; BUTKIEWICZ, 2013) é proposto um controle de estabilidade para um quadricóptero usando lógica fuzzy. Neste trabalho, três controladores fuzzy são usados: um para cada um dos dois eixos horizontais e mais um controlador geral para todo o sistema. Com isto, o erro angular e a aceleração são as variáveis linguísticas. Em ambos os casos, são usados cinco termos linguísticos: muito negativo (NL), pouco negativo (NS), zero (Z), pouco positivo (PS) e muito positivo (PL). Os conjuntos NS, Z e PS são triangulares enquanto os conjuntos NL e PL são trapezoidais, como mostra a Figura 1.

O bloco de fuzzificação calcula o grau de confiança dos conjuntos de entrada e um segundo bloco executa a inferência, que calcula o grau de confiança dos conjuntos de saída. As operações MIN e MAX são usadas como operadores AND e OR respectivamente para os conjuntos fuzzy. O conjunto de regras utilizadas são mostradas no Quadro 1 e os conjuntos fuzzy antes da defuzzificação são mostrados na Figura 2.

Os autores obtiveram sucesso no controle de estabilidade do *drone*, mas somente utilizando um bloco integrador. Apesar de os reguladores e motores serem do mesmo

Figura 1 – Conjuntos fuzzy usados em (MAJ; BUTKIEWICZ, 2013); a) erro de posição; b) erro de giro



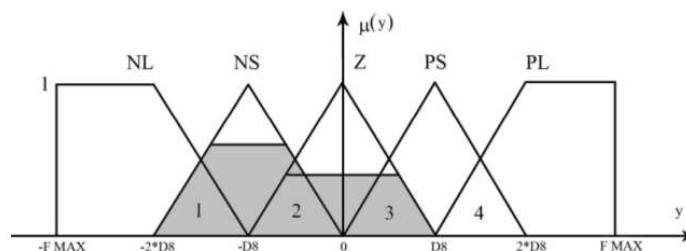
Fonte: (MAJ; BUTKIEWICZ, 2013)

Quadro 1 – Regras de inferência fuzzy usadas em (MAJ; BUTKIEWICZ, 2013)

Erro Angular	Erro de Giro				
	NL	NS	Z	PS	PL
NL	PL	PL	PL	PS	Z
NS	PL	PL	PS	Z	NS
Z	PL	PS	Z	NS	NL
PS	PS	Z	NS	NL	NL
PL	Z	NS	NL	NL	NL

Fonte: Adaptado de Maj e Butkiewicz (2013)

Figura 2 – Conjuntos fuzzy antes da defuzzificação usados em (MAJ; BUTKIEWICZ, 2013)



Fonte: (MAJ; BUTKIEWICZ, 2013)

tipo, algumas características deles são levemente diferentes. Como consequência, os motores não giram todos na mesma velocidade causando uma inclinação do quadrotor. Esta inclinação pode ser corrigida com o uso de um bloco integrador. Desta forma, foi observado que um controlador puramente fuzzy pode não ser suficiente para lidar com

esta situação mas que, acoplando um bloco integrador, o sistema certamente pode ser estabilizado.

Coza e Macnab (2006) propuseram um controlador fuzzy adaptativo para controlar a estabilidade de um quadricóptero *Outdoor*. A escolha deste tipo de controlador se deu graças a inconvenientes apresentados por outros métodos. Controle adaptativo pode ser apropriado mas requer conhecimento aprofundado do tipo de não linearidades nas dinâmicas. Um SMC (*Sliding Mode Control*<sup>2</sup>) pode ser apropriado mas pode causar vibração indesejada no sinal de controle. Controles por RNAs podem ser apropriados, mas requerem intenso esforço computacional para operar.

A abordagem utilizada em (COZA; MACNAB, 2006) se baseia na ideia de que diferentes conjuntos de centros de decodificação são capazes de aproximar uniformemente a mesma função não linear. Desta forma, é usado treino supervisionado sobre os centros alternados: o erro do estado treina o centro de controle e a diferença na saída treina os centros alternados. A partir de simulações, mostrou-se que o resultado obtido foi um controle estável, computacionalmente eficiente e teoricamente robusto a perturbações. Entretanto, em uma das situações de simulações de perturbações por vento, o controlador teve de sacrificar a boa performance computacional para prevenir o desvio de centro. Ainda assim, foi mostrado que um modelo fuzzy adaptativo usando atualização de controle e centro pode ser usado para cada eixo de rotação: X, Y e Z. Para tanto, apenas quatro funções Gaussianas de pertinência fuzzy foram usadas para cada entrada de controle.

Em (RABHI et al., 2011) uma outra abordagem é utilizada. Neste caso, os autores propuseram um controlador Fuzzy TSK (Takagi-Sugeno-Kang) para controlar a estabilidade de um quadrotor. O controlador foi modelado usando ferramentas matemáticas, mais especificamente a LMI (*Linear Matrix Inequalities*<sup>3</sup>) e modelado empregando PDC (*Parallel Distributed Compensation*<sup>4</sup>). O objetivo foi projetar um controlador fuzzy realimentado para garantir a estabilidade do sistema com robustez. Simulações mostram que o controlador projetado garante, de fato, a estabilidade global do sistema em malha fechada.

Sheikhpour e Shouraki (2013) seguiram a mesma linha de Rabhi et al. (2011) e propuseram um controlador fuzzy TSK para estabilização de altura de um quadrotor. Mais uma vez, a técnica PDC foi utilizada para projetar o controlador de realimentação. Neste trabalho, entretanto, o controle de estabilidade levou em conta especificações de performance como taxa de decaimento e restrições na entrada. Como ambas condições podem ser representadas por LMIs, elas também foram usadas neste caso, sendo mais

<sup>2</sup> Controle de Deslizamento de Modo, tradução nossa

<sup>3</sup> Desigualdades Matriciais Lineares

<sup>4</sup> Compensação Paralela Distribuída

complexas por lidar com as restrições dadas. Foi mostrado que simultaneamente à resolução das LMIs, foi projetado não apenas um controlador fuzzy estável mas também com inclusão de resposta de velocidade desejada e restrições na altitude de entrada no controlador. Os resultados obtidos em simulações mostram a viabilidade desta abordagem.

Niroumand et al. (2013) propuseram uma forma alternativa de controlador também usando a lógica fuzzy. Na abordagem utilizada, primeiramente foi derivado um modelo dinâmico não linear de um quadrotor e então foi desenvolvido o controlador híbrido usando métodos de controle tradicionais e inteligentes para estabilização de dinâmicas rotacionais. A técnica IBS (*Integral Backstepping*) é um poderoso método de controle tradicional amplamente utilizado para tal tipo de sistema mas encontrar o coeficiente apropriado do algoritmo é um trabalho crítico. Neste caso, esse problema foi resolvido usando um método de controle fuzzy. Foi mostrado nos resultados que o método IBS possui domínio de atração mais amplo em comparação a métodos de controle lineares como o PID e uma melhor convergência devido a condições iniciais rígidas. Foi mostrado ainda que ambos controladores IBS e Fuzzy IBS foram capazes de controlar o sistema de forma apropriada. Entretanto, o controlador FIBS (*Fuzzy Integral Backstepping*) obteve resultados levemente superiores ao IBS além de ter apresentado benefícios como uma melhor rejeição a perturbações e melhor robustez.

Já em (BASRI et al., 2014), foi proposto um controlador FSBC (*Fuzzy Supervisory Backstepping Controller*<sup>5</sup>). O controlador projetado consiste em controlador de *backstepping* que pode selecionar automaticamente seus parâmetros *on line* por um mecanismo de supervisão fuzzy. O critério de estabilidade para a estabilização do quadrotor é provada pelo teorema de Lyapunov. Extensivas simulações mostram a eficácia desta abordagem, apontando que uma alta precisão no transiente e no tempo de controle foram alcançados. Além disto, os resultados indicam que a técnica proposta pode estabilizar quadrotores com melhor performance se comparado a técnicas de estabilização lineares.

Em (GAO et al., 2014b), é proposto um outro tipo de controlador híbrido. Neste caso, o controle é feito a partir de um sistema composto por dois controladores independentes: um controlador *Backstepping* e um segundo controlador Fuzzy PID, que alia a lógica fuzzy a um controlador PID. Quando o quadrotor voa em boas condições, o controle de estabilidade é assumido pelo controlador *Backstepping*. Quando o quadrotor encontra vento ou outro fator de perturbação durante o voo, o controlador Fuzzy PID é usado para estabilizá-lo.

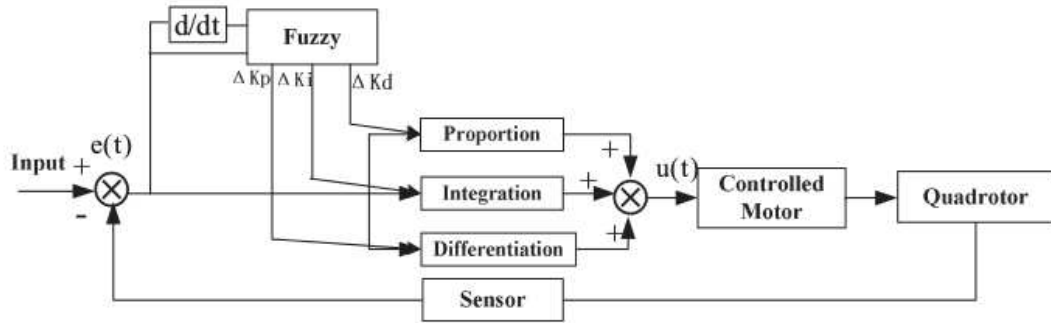
O controlador Fuzzy PID é um controlador PID que emprega o Sistema de Inteferência Fuzzy (FIS) para ajustar os parâmetros  $K_p$ ,  $K_i$ ,  $K_d$ , ganhos proporcional, integral e diferencial respectivamente. A Figura 3 mostra o diagrama de blocos representando o

<sup>5</sup> Controlador por Backstepping Fuzzy Supervisionado, tradução nossa



controlador Fuzzy PID.

Figura 3 – Diagrama de blocos do controlador híbrido usando em (GAO et al., 2014b)



Fonte: Gao et al. (2014b)

Em relação à estrutura fuzzy, há duas entradas para a inferência fuzzy: o erro e a derivada do erro. São três saídas, cada uma sendo um parâmetro do controlador PID  $\Delta k_p$ ,  $\Delta k_i$  e  $\Delta k_d$ . O universo de discurso foi normalizado e dividido em sete subconjuntos fuzzy. Os termos linguísticos foram definidos da seguinte forma: NB, negativo grande; NM, negativo médio; NS, negativo pequeno; ZO, aproximadamente zero; PS, positivo pequeno; PM, positivo médio; e PB, positivo grande. Os conjuntos fuzzy para cada variável de entrada consistem de sete variáveis linguísticas:  $e = \{NB, NM, NS, ZO, PS, PM, PB\}$ ,  $e_c = \{NB, NM, NS, ZO, PS, PM, PB\}$ . As variáveis linguísticas das saídas são atribuídas da seguinte forma:  $\Delta k_p = \{ZO, PS, PM, PB\}$ ,  $\Delta k_i = \{ZO, PS, PM, PB\}$ ,  $\Delta k_d = \{ZO, PS, PM, PB\}$ . As regras de inferência Fuzzy para as variáveis de saída  $\Delta k_p$ ,  $\Delta k_i$  e  $\Delta k_d$  são mostradas nos Quadros 2, 3 e 4 respectivamente.

Quadro 2 – Regras de inferência fuzzy usadas em (GAO et al., 2014b) para a variável  $\Delta K_p$

	$\Delta E_c$						
E	NB	NN	NS	ZO	PS	PM	PB
NB	PB	PM	ZO	ZO	ZO	PM	PB
NM	PB	PM	ZO	ZO	ZO	PM	PB
NS	PB	PB	PS	ZO	PS	PB	PB
ZO	PB	PB	PS	ZO	PS	PB	PB
PS	PB	PB	PS	ZO	PS	PB	PB
PM	PB	PM	ZO	ZO	ZO	PM	PB
PB	PB	PM	ZO	ZO	ZO	PM	PB

Fonte: Adaptado de Gao et al. (2014b)

O controlador proposto alcançou sucesso nos testes realizados, mostrando permitir completar a trajetória sem que perturbações afetem a precisão de controle. O Sistema de Inferência Fuzzy se mostrou eficiente para ajustar os ganhos do controlador PID de forma a possibilitar o controle eficiente do quadrotor.

Quadro 3 – Regras de inferência fuzzy usadas em (GAO et al., 2014b) para a variável  $\Delta K_i$

	$\Delta E_c$						
E	NB	NN	NS	ZO	PS	PM	PB
NB	ZO	ZO	ZO	PS	ZO	ZO	ZO
NM	ZO	ZO	ZO	PS	ZO	ZO	ZO
NS	ZO	ZO	PS	PS	PS	ZO	ZO
ZO	ZO	ZO	PS	PS	PS	ZO	ZO
PS	ZO	ZO	PS	PS	PS	ZO	ZO
PM	ZO	ZO	ZO	PS	ZO	ZO	ZO
PB	ZO	ZO	ZO	PS	ZO	ZO	ZO

Fonte: Adaptado de Gao et al. (2014b)

Quadro 4 – Regras de inferência fuzzy usadas em (GAO et al., 2014b) para a variável  $\Delta K_d$

	$\Delta E_c$						
E	NB	NN	NS	ZO	PS	PM	PB
NB	PB	PB	PM	PS	PM	PB	PB
NM	PB	PB	PM	PS	PM	PB	PB
NS	PB	PM	PS	ZO	PS	PM	PB
ZO	PB	PM	PS	ZO	PS	PM	PB
PS	PB	PM	PS	ZO	PS	PM	PB
PM	PB	PB	PM	ZO	PM	PB	PB
PB	PB	PB	PM	PS	PM	PB	PB

Fonte: Adaptado de Gao et al. (2014b)

Em outro trabalho, os mesmos autores implementaram um outro controlador híbrido. Em (GAO et al., 2014a), foi proposto um controlador Fuzzy PD. Além disso, neste trabalho, são comparadas as performances de três controladores diferentes: PID, Fuzzy PD e *Backstepping*. Mais uma vez, aqui o controlador fuzzy tem como função ajustar os parâmetros do controlador PD:  $\Delta K_p$  e  $\Delta K_d$ .

A Figura 4 mostra o diagrama de blocos do controlador neste caso. Como se pode ver, o diagrama mostra o controlador Fuzzy PD e um controlador PID.

Em relação à estrutura fuzzy, mais uma vez aqui há duas entradas para a inferência: o erro, sua derivada e os conjuntos fuzzy para cada uma delas consistem de sete variáveis linguísticas:  $e = \{NB, NM, NS, ZO, PS, PM, PB\}$ ,  $e_c = \{NB, NM, NS, ZO, PS, PM, PB\}$ . A variável de saída do sistema Fuzzy PD é apenas uma:  $U_{fuzzy} = \{ZO, PS, PM, PB\}$ . O Quadro 5 mostra as regras definidas para o sistema de inferência fuzzy.

O controlador Fuzzy PD é adotado para garantir uma supressão rápida e com sobrelevação quando o desvio é grande. Já o controlador PID é adotado para eliminar o estado estacionário quando o desvio é pequeno.





para ajuste de um IBC. O método é usado para minimizar o erro quadrático (SE) de uma função de custo que quantifica a performance de todo o sistema. Resultados de numerosas simulações mostram a validação e as boas performances alcançadas pelo método proposto.

Em ([BOUBERTAKH et al., 2013](#)) é proposta uma ideia parecida: utilizar o algoritmo PSO para ajustar os pesos de quatro controladores PD, cada qual responsável por controlar um rotor. A ideia é usar o PSO para minimizar o SE da função de custo que quantifica o sistema, assim como visto em ([YACEF et al., 2013](#)). As simulações feitas mostraram a eficácia do controlador proposto.

Como se pode ver, as técnicas inteligentes têm sido largamente utilizadas para propor diferentes tipos de controladores para quadrotores. Como já dito anteriormente, isso deve às inúmeras vantagens citadas que elas trazem. Algumas dessas técnicas foram utilizadas neste trabalho, como é mostrado a seguir.

## Referências

- BASRI, M.; HUSAIN, A.; DANAPALASINGAM, K. Fuzzy supervisory backstepping controller for stabilization of quadrotor unmanned aerial vehicle. In: **Intelligent and Advanced Systems (ICIAS), 2014 5th International Conference on**. [S.l.: s.n.], 2014. p. 1–5. Citado na página 6.
- BOUBERTAKH, H.; BENCHAREF, S.; LABIOD, S. Pso-based pid control design for the stabilization of a quadrotor. In: **Systems and Control (ICSC), 2013 3rd International Conference on**. [S.l.: s.n.], 2013. p. 514–517. Citado na página 10.
- COZA, C.; MACNAB, C. A new robust adaptive-fuzzy control method applied to quadrotor helicopter stabilization. In: **Fuzzy Information Processing Society, 2006. NAFIPS 2006. Annual meeting of the North American**. [S.l.: s.n.], 2006. p. 454–458. Citado na página 5.
- GAO, Q.; YUE, F.; HU, D. Research of precision flight control for quadrotor uav. In: **Guidance, Navigation and Control Conference (CGNCC), 2014 IEEE Chinese**. [S.l.: s.n.], 2014. p. 2369–2374. Citado 2 vezes nas páginas 8 e 9.
- GAO, Q.; YUE, F.; HU, D. Research of stability augmentation hybrid controller for quadrotor uav. In: **Control and Decision Conference (2014 CCDC), The 26th Chinese**. [S.l.: s.n.], 2014. p. 5224–5229. Citado 3 vezes nas páginas 6, 7 e 8.
- GUO, C.; SIMAAN, M.; SUN, Z. Neuro-fuzzy intelligent controller for ship roll motion stabilization. In: **Intelligent Control. 2003 IEEE International Symposium on**. [S.l.: s.n.], 2003. p. 182–187. ISSN 2158-9860. Citado na página 1.
- KHATOON, S.; SHAHID, M.; IBRAHEEM; CHAUDHARY, H. Dynamic modeling and stabilization of quadrotor using pid controller. In: **Advances in Computing, Communications and Informatics (ICACCI), 2014 International Conference on**. [S.l.: s.n.], 2014. p. 746–750. Citado na página 2.
- MAHFOUZ, M.; ASHRY, M.; ELNASHAR, G. Design and Control of Quad-Rotor Helicopters Based on Adaptive Neuro-Fuzzy Inference System. v. 2, n. 12, p. 479–485, 2013. ISSN 2278-0181. Citado na página 3.
- MAJ, W.; BUTKIEWICZ, B. Flying n-copter with fuzzy logic control. In: **Signal Processing Symposium (SPS), 2013**. [S.l.: s.n.], 2013. p. 1–6. Citado 2 vezes nas páginas 3 e 4.
- MUSTAPA, Z.; SAAT, S.; HUSIN, S.; ABAS, N. Altitude controller design for multi-copter uav. In: **Computer, Communications, and Control Technology (I4CT), 2014 International Conference on**. [S.l.: s.n.], 2014. p. 382–387. Citado na página 2.
- NIROUMAND, F.; FAKHARIAN, A.; SEYEDSAJADI, M. Fuzzy integral backstepping control approach in attitude stabilization of a quadrotor uav. In: **Fuzzy Systems (IFSC), 2013 13th Iranian Conference on**. [S.l.: s.n.], 2013. p. 1–6. Citado na página 6.

OLIVARES-MENDEZ, M.; CAMPOY, P.; MELLADO-BATALLER, I.; MEJIAS, L. See-and-avoid quadcopter using fuzzy control optimized by cross-entropy. In: **Fuzzy Systems (FUZZ-IEEE), 2012 IEEE International Conference on**. [S.l.: s.n.], 2012. p. 1–7. ISSN 1098-7584. Citado na página [1](#).

RABHI, A.; CHADLI, M.; PEGARD, C. Robust fuzzy control for stabilization of a quadrotor. In: **Advanced Robotics (ICAR), 2011 15th International Conference on**. [S.l.: s.n.], 2011. p. 471–475. Citado na página [5](#).

RAZINKOVA, A.; GAPONOV, I.; CHO, H.-C. Adaptive control over quadcopter uav under disturbances. In: **Control, Automation and Systems (ICCAS), 2014 14th International Conference on**. [S.l.: s.n.], 2014. p. 386–390. ISSN 2093-7121. Citado na página [2](#).

REZAZADEH, S.; ARDESTANI, M.; SADEGHI, P. Optimal attitude control of a quadrotor uav using adaptive neuro-fuzzy inference system (anfis). In: **Control, Instrumentation, and Automation (ICCIA), 2013 3rd International Conference on**. [S.l.: s.n.], 2013. p. 219–223. Citado na página [3](#).

SHEIKHPOUR, S.; SHOURAKI, S. B. A model-based fuzzy controller using the parallel distributed compensation method for quadrotor attitude stabilization. In: **Electrical Engineering (ICEE), 2013 21st Iranian Conference on**. [S.l.: s.n.], 2013. p. 1–6. Citado na página [5](#).

YACEF, F.; BOUHALI, O.; HAMERLAIN, M.; REZOUG, A. Pso optimization of integral backstepping controller for quadrotor attitude stabilization. In: **Systems and Control (ICSC), 2013 3rd International Conference on**. [S.l.: s.n.], 2013. p. 462–466. Citado 2 vezes nas páginas [9](#) e [10](#).

## Apêndices

## APÊNDICE A – Código para Criação de Modelo Neuro-Fuzzy para Altitude e Definição de Dados para Treinamento

```

1      % le arquivo fis referente ao controle de altitude
2      fismat = readfis('fis_altitude.fis');
3
4      % define numero de casos a serem avaliados (treinamento + teste)
5      n = 300;
6      % define conjunto de n entradas aleatorias para o sistema fuzzy
7      % respeitando o range de cada entrada
8      input = zeros(n,2);
9      for i=1:n
10         z_value = rand * 2 - 1;
11         z_dot_value = rand * 10 - 5;
12         input(i,:) = [ z_value z_dot_value ];
13     end
14
15     % avalia resposta fuzzy para cada entrada
16     output= evalfis(input,fismat);
17
18     % define data como vetor relacionando cada conjunto de entradas ...
19     % a saida
20     % - obtida pelo sistema fuzzy
21     data = [];
22     for i=1:n
23         data(i,:) = [ input(i,:) output(i) ];
24     end
25
26     % define que 2/3 dos dados obtidos serao usados para treinamento
27     % e 1/3 sera usado para teste da rede
28     train = data(1:2*n/3,:);    % dados para treinamento
29     test = data(2*n/3+1:n,:);  % dados para validacao do sistema ...
30     treinado
31
32     % gera modelo fuzzy Sugeno a partir do Mamdani modelado
33     sugFIS = mam2sug(fismat);
34     % salva modelo Sugeno em disco com o nome fis_altitude_neuro.fis
35     writefis(sugFIS, 'fis_altitude_neuro.fis');

```

## APÊNDICE B – Código para Criação de Modelo Neuro-Fuzzy para Atitude e Definição de Dados para Treinamento

```

1      % le arquivo fis referente ao controle de atitude
2      fismat = readfis('fis_atitude.fis');
3
4      % define numero de casos a serem avaliados (treinamento + teste)
5      n = 300;
6      % define conjunto de n entradas aleatorias para o sistema fuzzy
7      % respeitando o range de cada entrada
8      input = zeros(n,2);
9      for i=1:n
10         phi_value = rand * 4 - 2;
11         phi_dot_value = rand * 3 - 1.5;
12         input(i,:) = [ phi_value phi_dot_value ];
13     end
14
15     % avalia resposta fuzzy para cada entrada
16     output= evalfis(input,fismat);
17
18     % define data como vetor relacionando cada conjunto de entradas ...
19     % a saída
20     % obtida pelo sistema fuzzy
21     data = [];
22     for i=1:n
23         data(i,:) = [ input(i,:) output(i) ];
24     end
25
26     % define que 2/3 dos dados obtidos serao usados para treinamento
27     % e 1/3 sera usado para teste da rede
28     train = data(1:2*n/3,:);    % dados para treinamento
29     test = data(2*n/3+1:n,:);  % dados para validação do sistema ...
30     % treinado
31
32     % gera modelo fuzzy Sugeno a partir do Mamdani modelado
33     sugFIS = mam2sug(fismat);
34     % salva modelo Sugeno em disco com o nome fis_atitude_neuro.fis
35     writefis(sugFIS, 'fis_atitude_neuro.fis');

```