

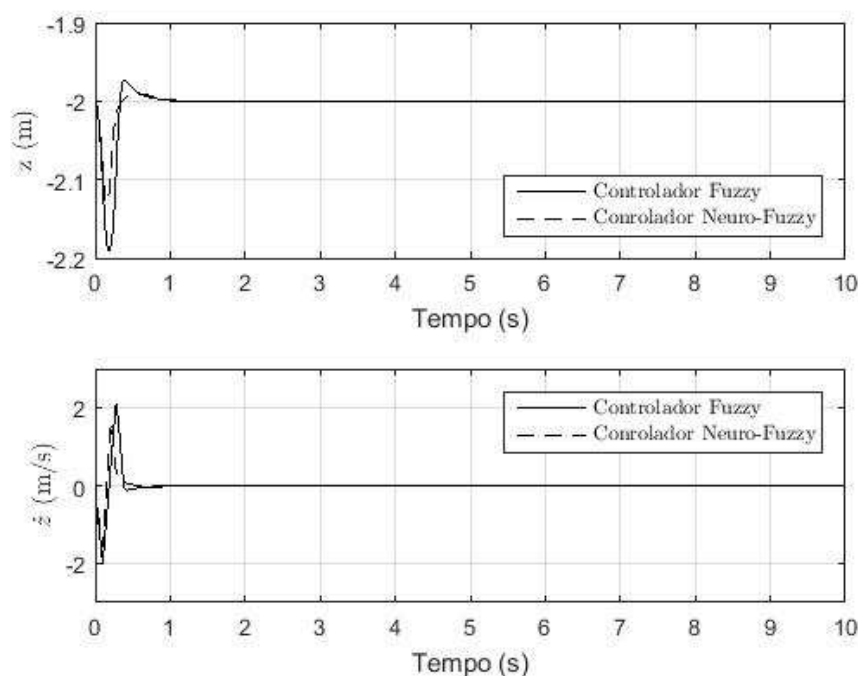
Sumário

| | |
|---|-----------|
| 1 – Resultados | 1 |
| 1.1 Análise Consolidada dos Resultados | 13 |
| 2 – Conclusão | 15 |
| 2.1 Trabalhos futuros | 16 |
| Referências | 17 |
| | |
| Apêndices | 18 |
| | |
| APÊNDICE A – Código para Criação de Modelo Neuro-Fuzzy para Altitude e Definição de Dados para Treinamento | 19 |
| | |
| APÊNDICE B – Código para Criação de Modelo Neuro-Fuzzy para Atitude e Definição de Dados para Treinamento | 20 |

1 Resultados

A Figura 1 mostra a posição no eixo vertical (z) do *drone*, bem como sua variação no sistema em que atua o controlador *fuzzy* projetado. Como se pode ver, o distúrbio foi devidamente controlado, fazendo com que o quadricóptero retornasse à posição inicial $z = -2$ m e também ao repouso¹ representado por $\dot{z} = 0$ m/s. Nesta figura, entretanto, não fica tão clara a diferença de desempenho dos controladores *fuzzy* e *neuro-fuzzy*, aspecto que pode ser claramente verificado na Figura 2. Como se pode ver, tanto para z quanto para \dot{z} , o *neuro-fuzzy* apresenta desempenho melhor. No controle sobre a posição z , o controlador *neuro-fuzzy* apresentou redução do tempo de convergência em 29%, e da variação do sistema em 31%, além de eliminar a sobrelevação apresentada pelo *fuzzy*. Já sobre a velocidade \dot{z} , apresentou uma redução no tempo de convergência de 29% além melhorar a variação do sistema em 23%. A partir destes resultados, verifica-se que o controlador *neuro-fuzzy* fez com que o distúrbio fosse melhor absorvido e que sua correção ocorresse mais rapidamente.

Figura 1 – Comparação da resposta das saídas z e \dot{z} no controle de altitude *fuzzy* e *neuro-fuzzy* para o sistema com massa $m = 2$ kg



Já a Figura 3 mostra a ação de ambos os controladores. O controlador *neuro-fuzzy* apresentou melhor resultado quanto ao gasto energético, apresentando um gasto 14% menor do que o *fuzzy*.

¹ i.e. velocidade nula

Figura 2 – Comparação em mais detalhes da resposta das saídas z e \dot{z} no controle de altitude fuzzy e neuro-fuzzy para o sistema com massa $m = 2$ kg

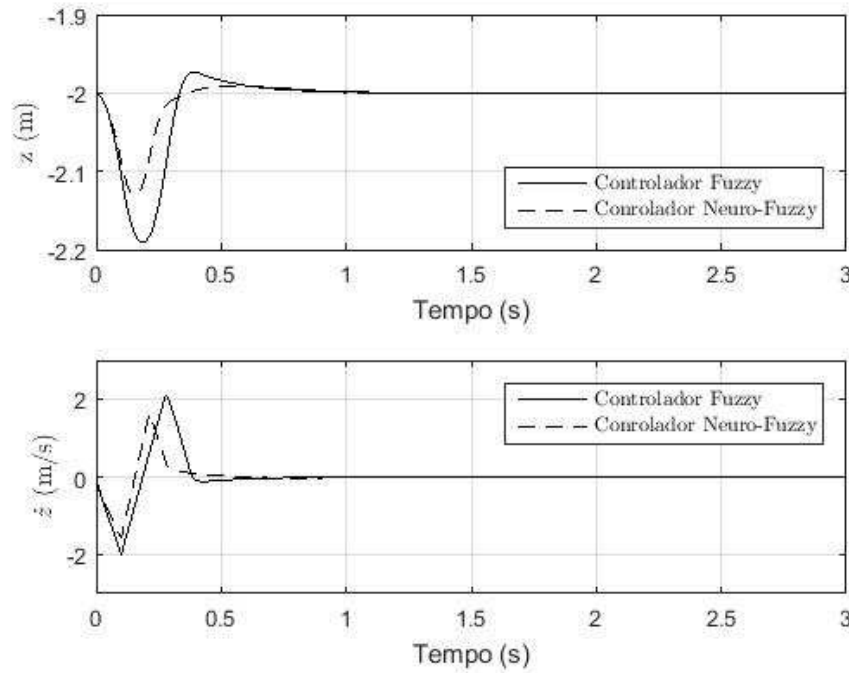
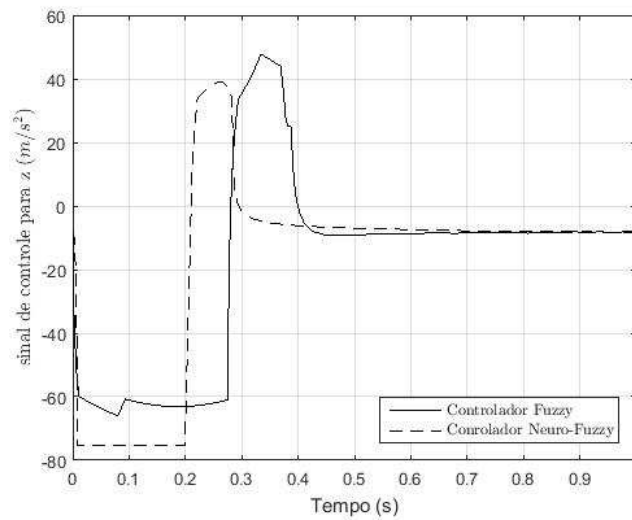


Figura 3 – Comparação da ação dos controladores *fuzzy* e *neuro-fuzzy* na estabilização em altitude do sistema com massa $m = 2$ kg



Já as Figuras 4 e 5 mostram a estabilidade de atitude em torno dos eixos x e y (i.e em relação ao plano horizontal XY), representados por θ e ϕ respectivamente. Como se pode ver, ambos os estados são devidamente controlados e, com isto, o *drone* volta à estabilidade horizontal, com ângulos e velocidades angulares nulas no estado permanente.

A partir das Figuras 6 e 7, que mostram as respostas obtidas em mais detalhes,

Figura 4 – Comparação da resposta das saídas ϕ e $\dot{\phi}$ no controle de atitude *fuzzy* e neuro-*fuzzy* para o sistema com massa $m = 2$ kg

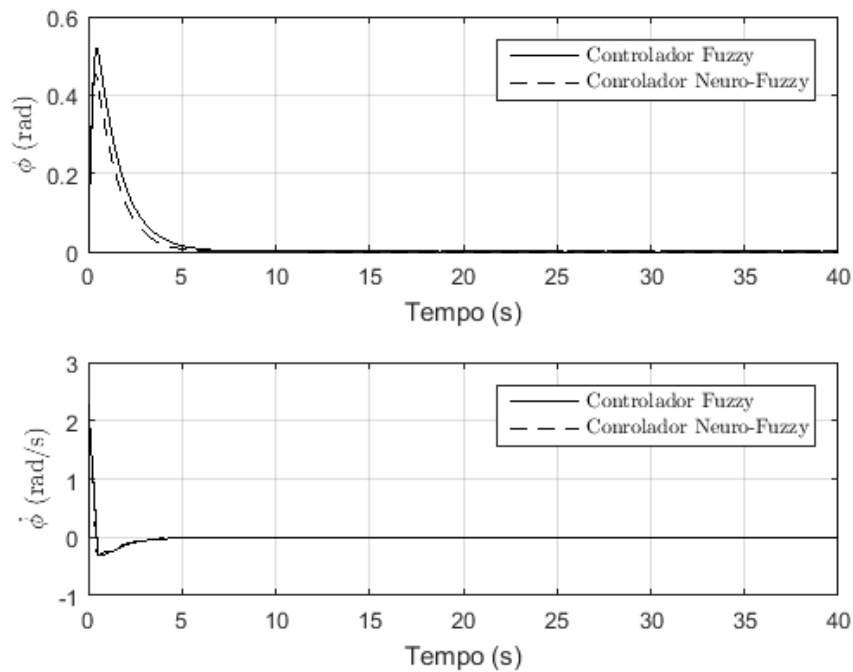
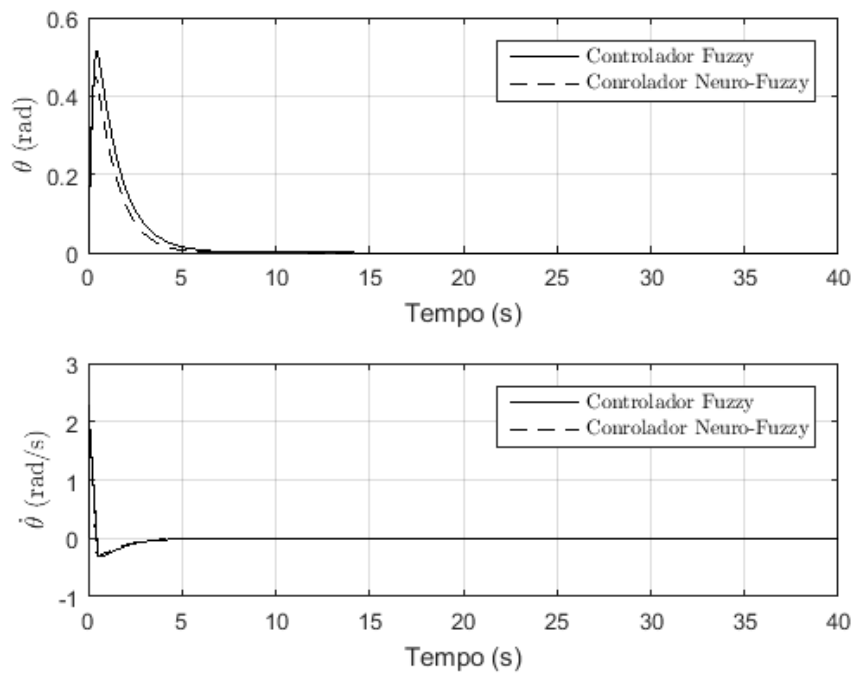


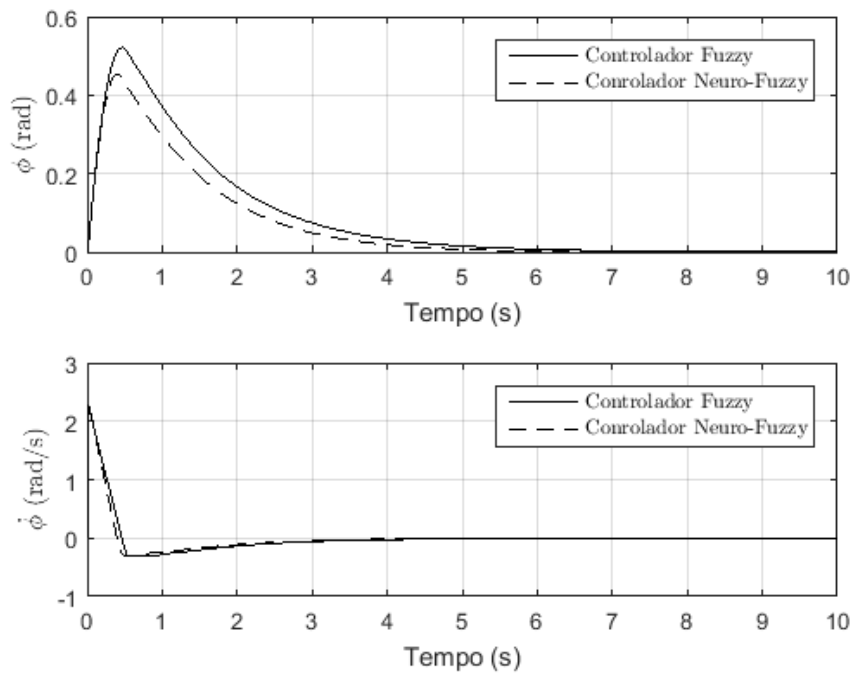
Figura 5 – Comparação da resposta das saídas θ e $\dot{\theta}$ no controle de atitude *fuzzy* e neuro-*fuzzy* para o sistema com massa $m = 2$ kg



pode-se ver que, mais uma vez o controlador neuro-*fuzzy* mais uma vez teve desempenho superior ao *fuzzy*, fazendo com que os ângulos ϕ e θ convergissem 2% mais rápido, além de reduzir suas variações a 13%. Com relação às velocidades angulares

$\dot{\phi}$ e $\dot{\theta}$, foi capaz de reduzir o tempo de convergência em 3%, não afetando a variação nem a sobrelevação apresentada pelo sistema quando estabilizado pelo controlador *fuzzy*. Desta forma, verifica-se que o controle neuro-fuzzy levou o sistema a uma menor variação, representando que o ângulo máximo de inclinação alcançado pelo *drone* é menor e corrigido mais rapidamente.

Figura 6 – Comparação em mais detalhes da resposta das saídas ϕ e $\dot{\phi}$ no controle de atitude *fuzzy* e neuro-*fuzzy* para o sistema com massa $m = 2$ kg



As Figuras 8 e 9 mostra a ação de ambos os controladores no processo de estabilização da atitude do *drone*. Como se pode ver, ambas as respostas são bastante parecidas, devido ao fato de o sistema ser praticamente simétrico em relação aos seus eixos x e y. Em ambos os casos, o controlador neuro-*fuzzy* apresentou resultado inferior ao *fuzzy*, consumindo 12% mais energia do que este.

Além da verificação da eficiência dos controladores atuando sobre o sistema para o qual foram projetados, eles foram testados num sistema em que um dos parâmetros foi acrescido de mais de 100 %, com a massa passando de 2,3 kg para 5 kg.

A resposta dos controladores *fuzzy* e neuro-*fuzzy* para altitude do *drone* nessas circunstâncias são mostradas nas Figuras 10 e 11, sendo que esta segunda é apenas uma forma melhor de comparar a ação dos dois controladores. Como se pode perceber, ambos os controladores levaram à estabilização do sistema, sendo que desta vez cada um obteve desempenho melhor sob determinados aspectos. No controle da posição vertical z , o neuro-fuzzy apresentou tempo de convergência 57% maior, em parte causado por uma sobrelevação, que não foi apresentada pelo *fuzzy*. Em contrapartida, o

Figura 7 – Comparação em mais detalhes da resposta das saídas θ e $\dot{\theta}$ no controle de atitude *fuzzy* e *neuro-fuzzy* para o sistema com massa $m = 2$ kg

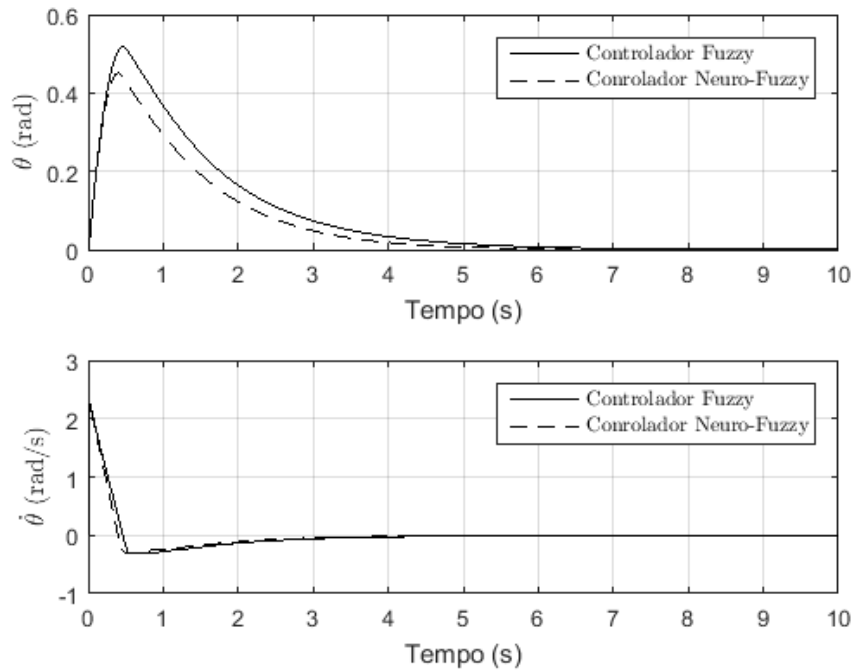
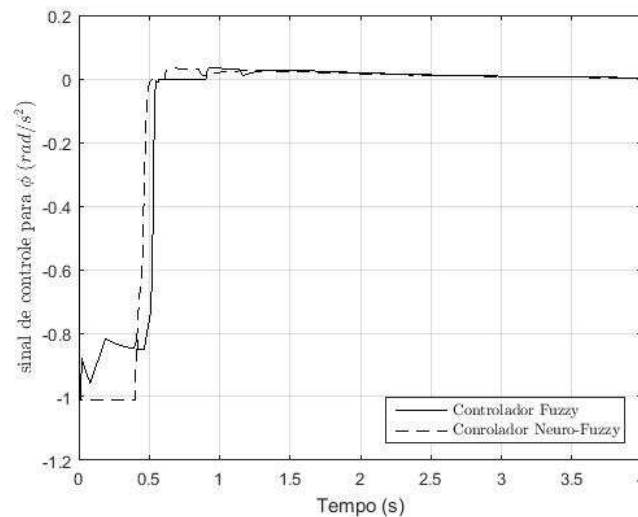


Figura 8 – Comparação da ação dos controladores *fuzzy* e *neuro-fuzzy* na estabilização em atitude do sistema com massa $m = 2$ kg para variável ϕ



neuro-fuzzy apresentou uma menor variação, a reduzindo em 20% se comparado ao *fuzzy*. Com relação à velocidade \dot{z} , o controlador *neuro-fuzzy* apresentou aumento de 23% no tempo de convergência, mas melhorou o sistema nos quesitos variação e sobrelevação, as reduzindo em 16% e 33%, respectivamente. Esses resultados apontam que o quadricóptero, quando submetido ao controle *neuro-fuzzy*, apresentou movimentos mais suaves até ter sua altitude estabilizada, apesar de ter sido necessário mais tempo para que ela ocorresse.

Figura 9 – Comparação da ação dos controladores *fuzzy* e *neuro-fuzzy* na estabilização em atitude do sistema com massa $m = 2$ kg para variável θ

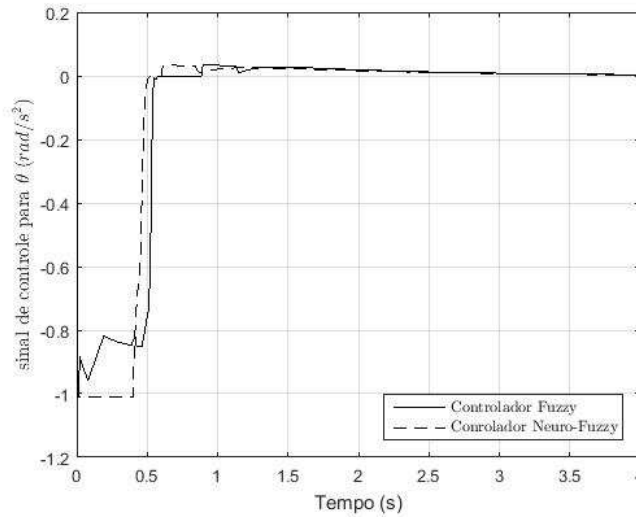
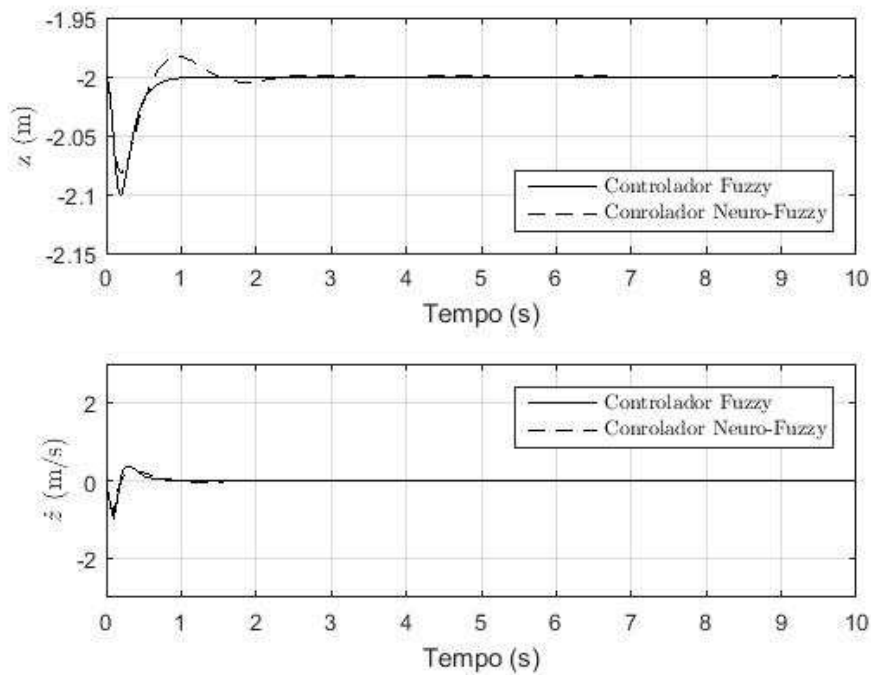


Figura 10 – Comparação da resposta das saídas z e \dot{z} no controle de altitude *fuzzy* e *neuro-fuzzy* para o sistema com massa $m = 5$ kg



A Figura 12 mostra a ação de ambos os controladores sendo que o *neuro-fuzzy* apresentou um resultado ligeiramente superior ao *fuzzy* com relação ao gasto energético, consumindo 7% menos.

As Figuras 13 e 14 mostram os resultados obtidos pelos controladores de atitude no sistema com massa $m = 5$ kg. Percebe-se que mais uma vez o sistema convergiu ao seu estado de estabilidade com os ângulos nulos e velocidades angulares também

Figura 11 – Comparação em mais detalhes da resposta das saídas z e \dot{z} no controle de altitude *fuzzy* e *neuro-fuzzy* para o sistema com massa $m = 5$ kg

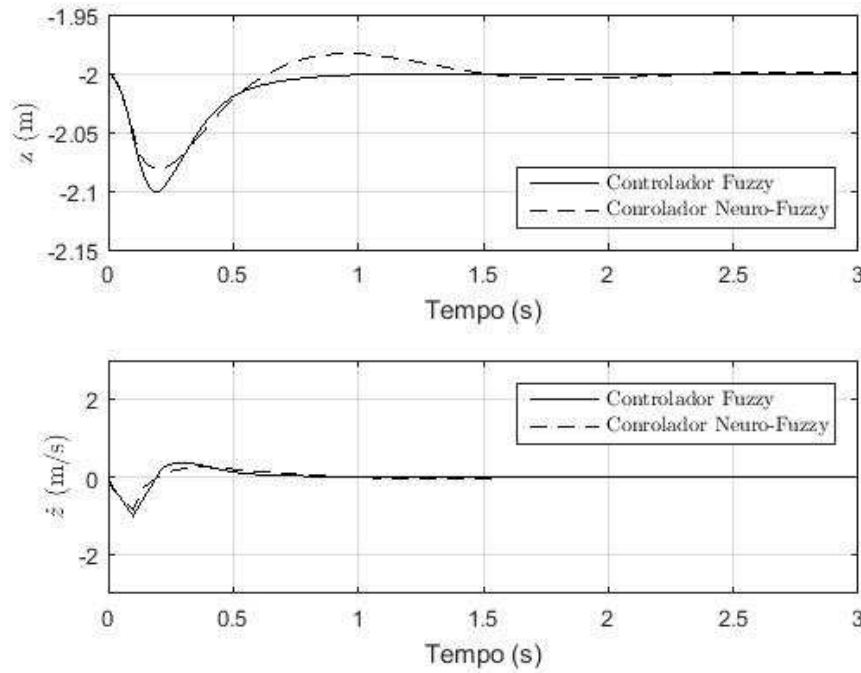
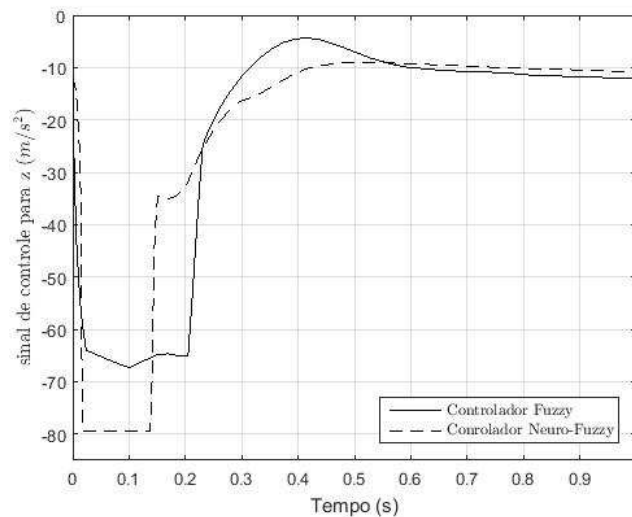


Figura 12 – Comparação da ação dos controladores *fuzzy* e *neuro-fuzzy* na estabilização em altitude do sistema com massa $m = 5$ kg



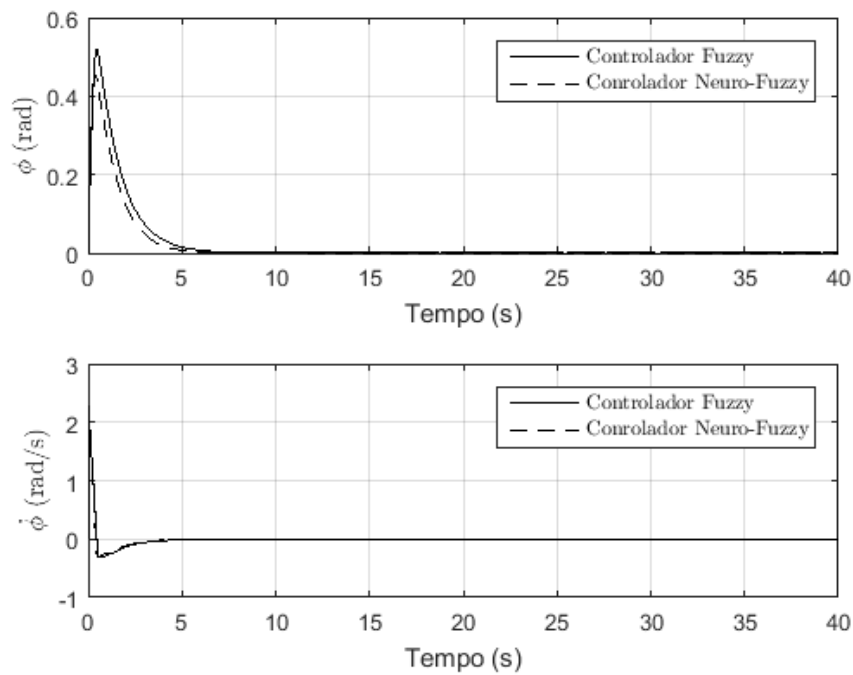
nulas, representando que o quadricóptero, após a ação de controle, tanto *fuzzy* quanto *neuro-fuzzy*, fica estável e com orientação plana².

As Figuras 15 e 16 mostram em mais detalhes as repostas obtidas pelos controladores sobre a atitude do sistema com massa $m = 5$ kg. A partir delas, nota-se que mais uma vez o controlador *neuro-fuzzy* apresentou desempenho levemente superior

² i.e paralela ao plano XY

ao *fuzzy*. Sobre os ângulos ϕ e θ , a convergência ocorreu 3% mais rapidamente e a variação apresentada reduziu 14%. Já sobre as velocidades angulares $\dot{\phi}$ e $\dot{\theta}$, a redução de tempo de convergência com o *neuro-fuzzy* foi de 2%, mantendo a mesma sobrelevação e variação oferecidas pelo *fuzzy*. Com isto, mais uma vez o controle *neuro-fuzzy* fez com que o ângulo máximo de inclinação do *drone* fosse inferior ao alcançado pelo sistema controlado pelo *fuzzy*, e além de reduzir o tempo necessário para sua estabilização definitiva.

Figura 13 – Comparação da resposta das saídas ϕ e $\dot{\phi}$ no controle de atitude *fuzzy* e *neuro-fuzzy* para o sistema com massa $m = 5$ kg



Por fim, as Figuras 17 e 18 mostram a ação de ambos os controladores no processo de estabilização da atitude do *drone*. Assim como para o sistema com massa $m = 2$ kg, em ambos os casos o controlador *neuro-fuzzy* apresentou resultado inferior ao *fuzzy*. Desta vez, entretanto, o consumo foi 10% maior.

Já no teste de robustez a ruídos de medição dos controladores desenvolvidos, a Figura 19 mostra a resposta do sistema ao ruído de medição representado pela Figura 20. A mesma resposta é mostrada na Figura 21, porém com maiores detalhes, permitindo uma melhor comparação do sistema controlado pelos diferentes controladores.

Como se pode ver, o controlador *neuro-fuzzy* obteve uma resposta melhor se comparado ao *fuzzy*, apresentando uma redução em 39 % na variação do sistema, além de uma convergência 13 % mais rápida. Além disso, nota-se que, com o controlador *neuro-fuzzy*, o sistema ficou mais estável, apresentando menores variações, apresentando assim, uma melhor resposta ao sistema sujeito a ruídos.

Figura 14 – Comparação da resposta das saídas θ e $\dot{\theta}$ no controle de atitude *fuzzy* e *neuro-fuzzy* para o sistema com massa $m = 5$ kg

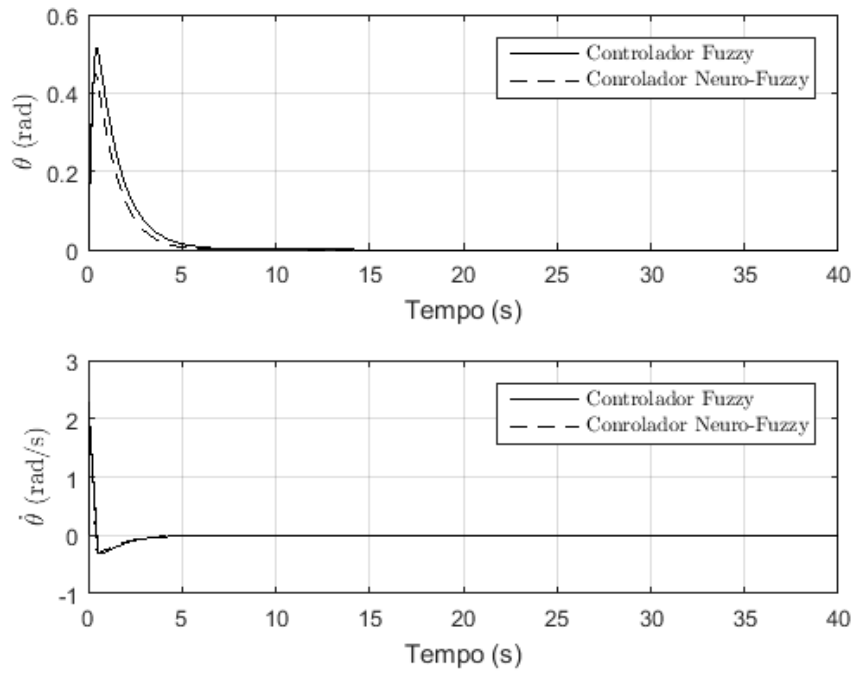
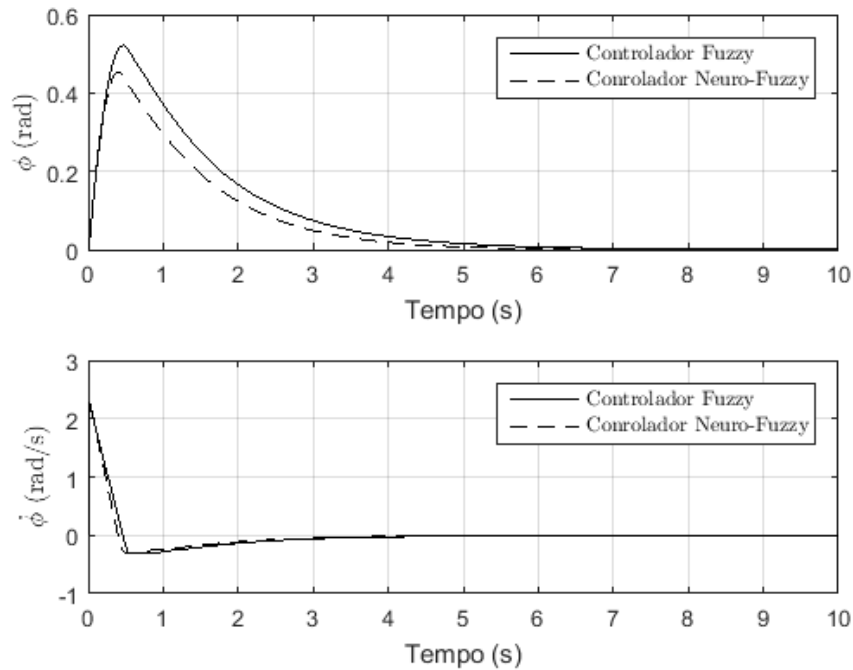


Figura 15 – Comparação da resposta das saídas ϕ e $\dot{\phi}$ no controle de atitude *fuzzy* e *neuro-fuzzy* para o sistema com massa $m = 5$ kg



Por fim, a Figura 22 mostra a resposta dos dois controladores ao longo tempo. Como se pode ver, o controlador *neuro-fuzzy* apresentou eficiência energética 33 % superior ao *fuzzy* até o momento de convergência. Além disso, percebe-se que a ação do

Figura 16 – Comparação da resposta das saídas θ e $\dot{\theta}$ no controle de atitude *fuzzy* e neuro-*fuzzy* para o sistema com massa $m = 5$ kg

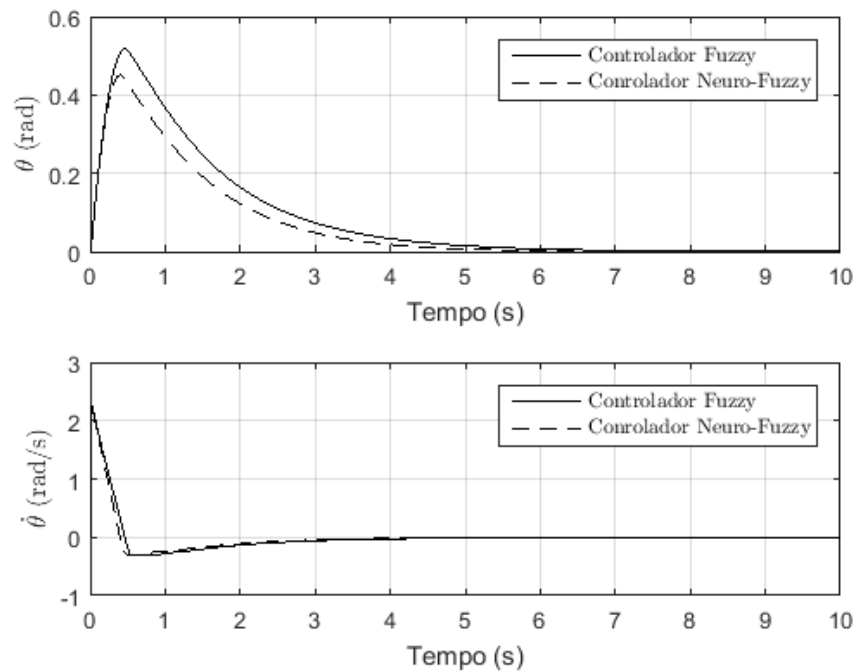
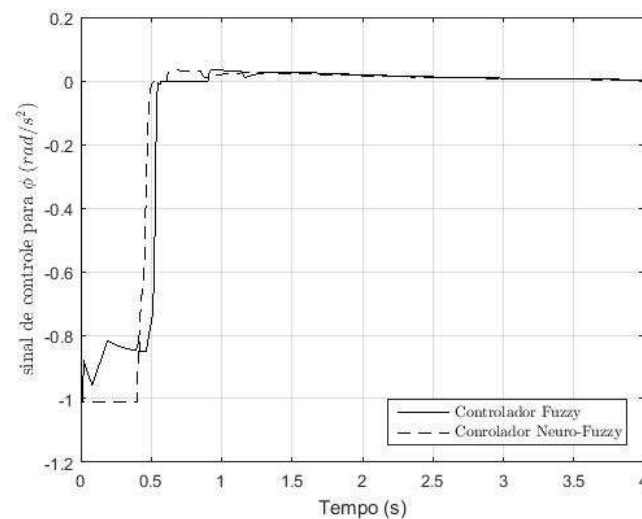


Figura 17 – Comparação da ação dos controladores *fuzzy* e neuro-*fuzzy* na estabilização em atitude do sistema com massa $m = 5$ kg para variável ϕ



controlador neuro-*fuzzy* é muito mais sutil na absorção dos ruídos, levando assim a um grande ganho de desempenho energético a longo prazo.

Figura 18 – Comparação da ação dos controladores *fuzzy* e *neuro-fuzzy* na estabilização em atitude do sistema com massa $m = 2$ kg para variável θ

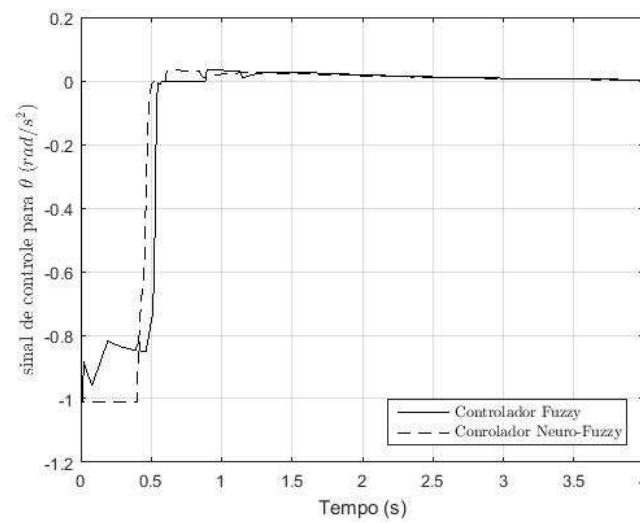


Figura 19 – Comparação da resposta das saídas z e \dot{z} no controle de altitude *fuzzy* e *neuro-fuzzy* para o sistema com massa $m = 2$ kg sujeito a ruídos de medição da variável z

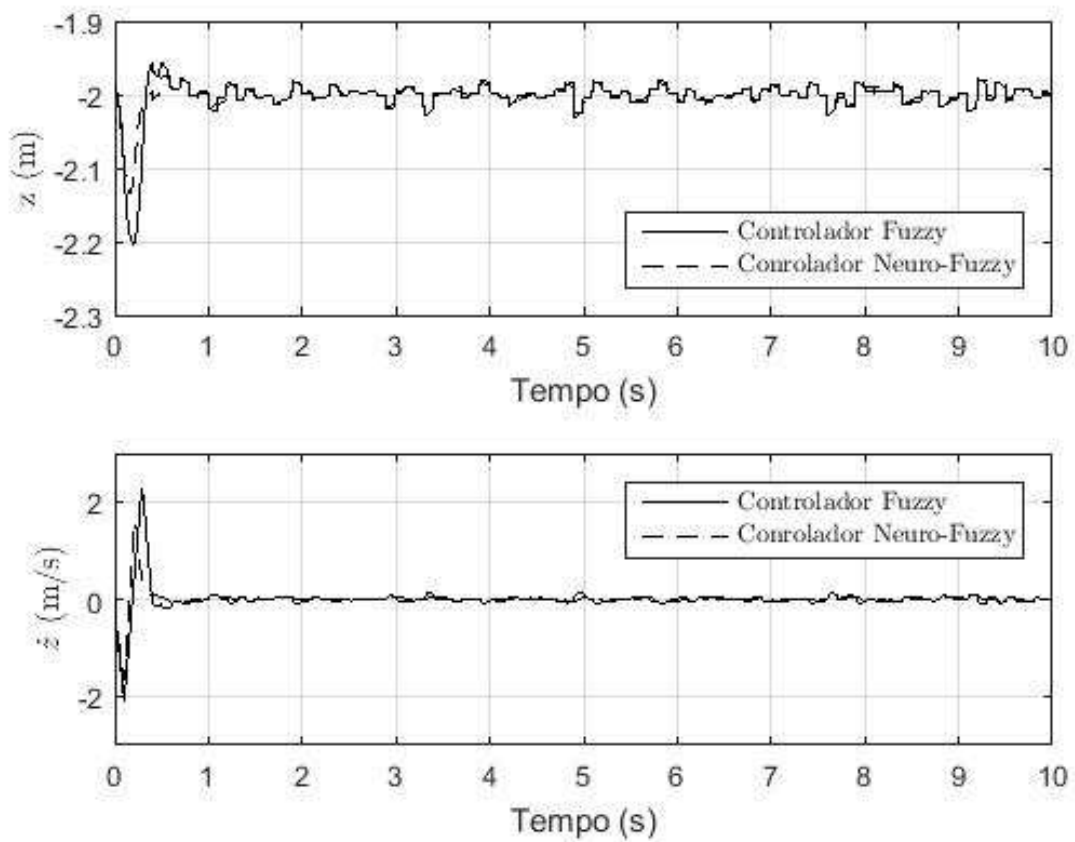


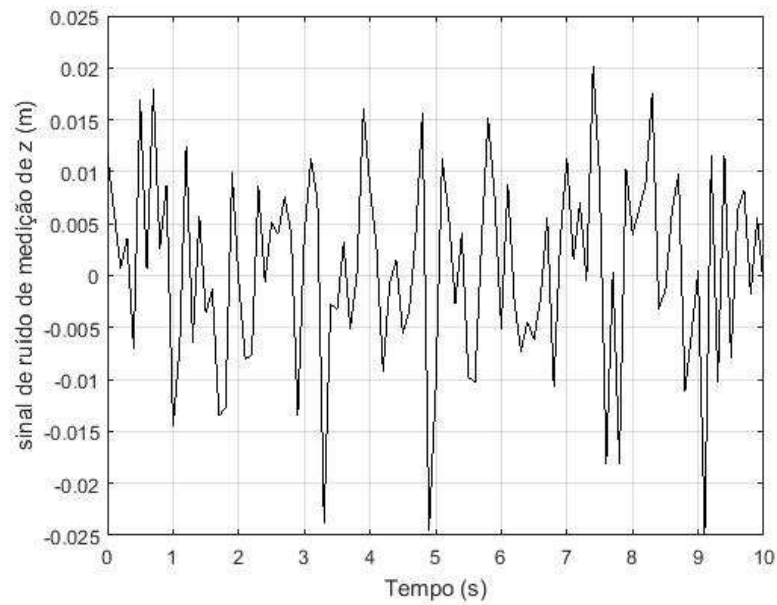
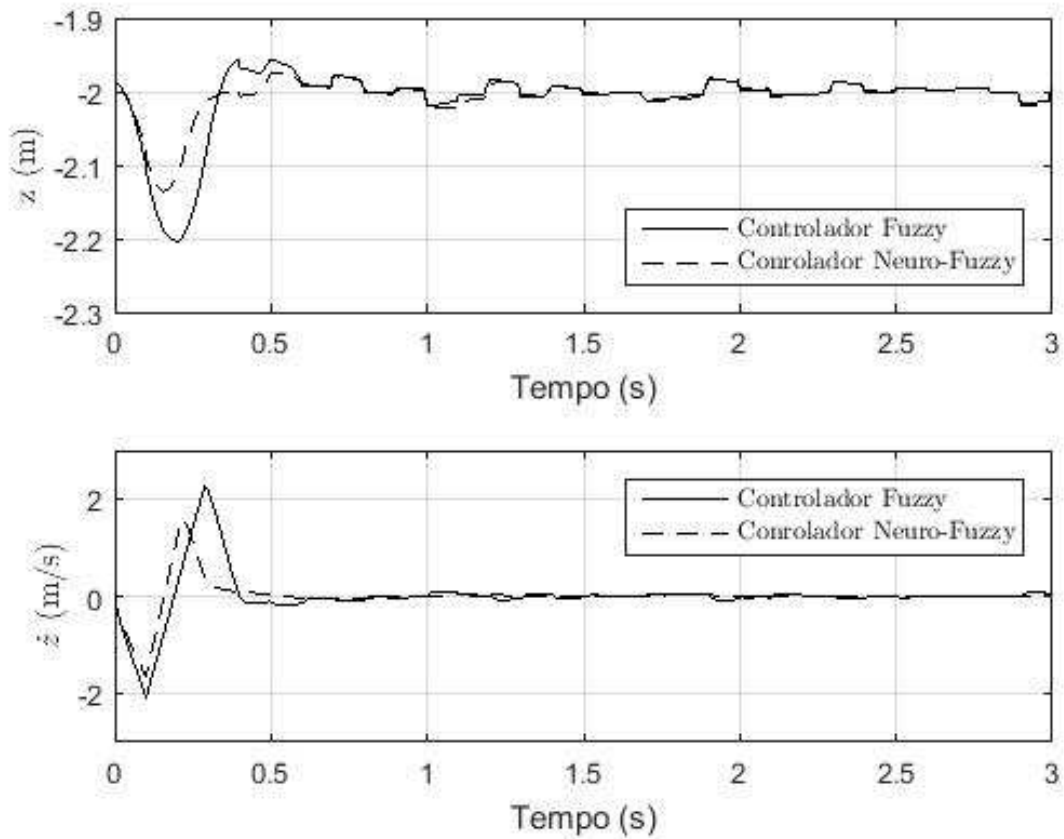
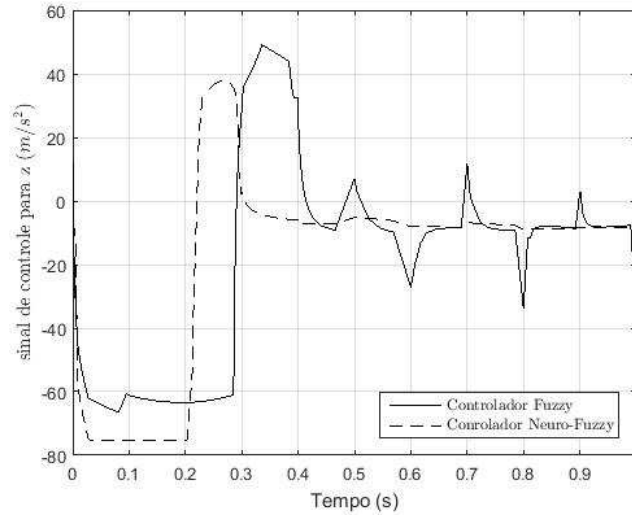
Figura 20 – Sinal de ruído de medição sobre o valor real da variável z Figura 21 – Comparação em mais detalhes da resposta das saídas z e \dot{z} no controle de altitude *fuzzy* e *neuro-fuzzy* para o sistema com massa $m = 2$ kg sujeito a ruídos de medição da variável z 

Figura 22 – Comparação da ação dos controladores *fuzzy* e *neuro-fuzzy* na estabilização em altitude do sistema com massa $m = 2$ kg sujeito a ruídos de medição da variável z



1.1 Análise Consolidada dos Resultados

Como se pode ver, os controladores de atitude e altitude tanto *fuzzy* quanto *neuro-fuzzy* foram eficientes levando à estabilização do sistema em todos os casos testados, inclusive na situação em que a massa do sistema foi aumentada em mais de 100 %. Isso indica que estes controladores podem ser utilizados, por exemplo, em situações em que o drone precisaria transportar uma carga que tenha sua massa ou até mesmo uma superior.

Em quase todos os casos, nota-se um comportamento do controlador *neuro-fuzzy* superior ao do *fuzzy*, o que já era esperado tendo em vista que o primeiro alia o poder do segundo às vantagens das RNAs, fazendo com que, a partir de um treinamento supervisionado utilizando o próprio modelo *fuzzy*, possa se construir um controle mais abrangente e com resposta melhorada. Além disto, verificou-se uma melhora no consumo energético no controle de altitude ao utilizar o controlador *neuro-fuzzy*. No controle de atitude, entretanto, o controlador *neuro-fuzzy* apresentou pior eficiência energética.

Já no experimento envolvendo ruídos de medição, o controlador *neuro-fuzzy* de altitude se mostrou muito superior ao *fuzzy*, apresentando menor variação e menor tempo de convergência além de uma eficiência energética muito melhor, atuando de forma muito sutil para as correções dos ruídos incluídos no sistema ao passo que o *fuzzy* gasta muito mais energia para fazer cada uma dessas correções.

Estes resultados mostram que, de fato, as técnicas de Inteligência Computacional podem ser aplicadas para projetar controladores eficientes e robustos para atuar sobre

sistemas multivariável e que, além disto, o poder de treinamento dos ANFISs realmente é capaz de fazer com que o desempenho de controladores neuro-fuzzy seja melhorado se comparado ao daqueles puramente fuzzy, tanto com relação à qualidade de resposta quanto à eficiência energética.

A sumarização dos resultados é feita nos Quadros 1, 2 e 3.

Quadro 1 – Desempenho do controlador neuro-fuzzy comparado ao fuzzy no controle sobre o sistema com massa $m = 2$ kg

| Aspecto | Resultado comparativo ao fuzzy |
|--|--------------------------------|
| Tempo de convergência de z | 29 % melhor |
| Variação de z | 23 % melhor |
| Tempo de convergência de ϕ e θ | 2 % melhor |
| Variação de ϕ e θ | 13 % melhor |
| Consumo energético para controle em altitude | 14 % melhor |
| Consumo energético para controle em atitude | 14 % pior |

Quadro 2 – Desempenho do controlador neuro-fuzzy comparado ao fuzzy no controle sobre o sistema com massa $m = 5$ kg

| Aspecto | Resultado comparativo ao fuzzy |
|--|--------------------------------|
| Tempo de convergência de z | 57 % pior |
| Variação de z | 20 % melhor |
| Tempo de convergência de ϕ e θ | 3 % melhor |
| Variação de ϕ e θ | 14 % melhor |
| Consumo energético para controle em altitude | 7 % melhor |
| Consumo energético para controle em atitude | 10 % pior |

Quadro 3 – Desempenho do controlador neuro-fuzzy comparado ao fuzzy no controle sobre o sistema com massa $m = 2$ kg submetido a ruídos de medição da variável z

| Aspecto | Resultado comparativo ao fuzzy |
|--|--------------------------------|
| Tempo de convergência de z | 13 % melhor |
| Variação de z | 39 % melhor |
| Consumo energético para controle em altitude | 33 % melhor |

2 Conclusão

Ao longo deste trabalho, discorreu-se sobre o crescente uso de estratégias da Inteligência Computacional para implementar controladores de sistemas não lineares. Além disto, como foi mostrado pelos experimentos computacionais realizados, o uso de controladores devidamente projetados é fundamental para fazer com que esses sistemas instáveis atuem da forma planejada e possam ser estabilizados.

No contexto deste trabalho, o sistema controlado é um quadricóptero e as variáveis são referentes à sua atitude e altitude, representando portanto um controle multivariável. As alternativas propostas como controladores foram o *fuzzy* e o *neuro-fuzzy*. A opção pelo primeiro se deveu ao fato de ele permitir a modelagem a partir de variáveis e termos linguísticos, além de acrescentar robustez ao sistema. Já a opção pelo segundo, *neuro-fuzzy*, foi devido ao fato de este agregar as características de RNAs aos sistemas *fuzzy*, possuindo um poder de aprendizado capaz de melhorar sua performance.

De fato, os resultados mostram que o controlador *neuro-fuzzy* realmente obteve melhor desempenho. No controle de altitude, reduziu o tempo de convergência em 29% e a variação do sistema em 31%, além de eliminar a sobrelevação apresentada pelo *fuzzy*. O controle *neuro-fuzzy* de atitude também apresentou melhoras, apesar de não tão significativas quanto essas: reduziu o tempo de convergência em 2% e a variação do sistema em 13%.

Além disto, num teste para verificar a robustez dos controladores, a massa do sistema foi acrescida em 117%, passando de 2 kg para 5 kg. Neste novo contexto, o controlador de atitude *neuro-fuzzy* mais uma vez foi superior ao *fuzzy*, reduzindo o tempo de convergência em 3% e da variação em 14%. Já no controle de altitude, o único fator melhorado pelo *neuro-fuzzy* foi a variação do sistema, sendo reduzida em 20%, ao passo que seu tempo de convergência cresceu 57%, além de ter sido inserida uma sobrelevação na resposta.

No experimento computacional incluindo ruídos de medição, ambos os controladores se mostraram capazes de lidar com eles. Neste ponto, o controlador *neuro-fuzzy* apresentou desempenho bastante superior, levando a uma convergência mais rápida e com menor variação além de consumir menos energia.

Apesar das diferenças de desempenho, os controladores *fuzzy* e *neuro-fuzzy* tanto para atitude quanto para altitude do sistema foram capazes de estabilizá-lo, mesmo quando submetido a uma variação substancial de parâmetros, que foi representada pelo aumento da massa em mais de 100%.

Desta forma, mostra-se que se podem usar técnicas de IC para controlar, de forma eficiente, sistemas não lineares complexos e, além disso, que controladores neuro-*fuzzy* podem ser utilizados para melhorar o desempenho de controladores *fuzzy* apesar de, em algumas situações, piorar a resposta se comparado a estes.

2.1 Trabalhos futuros

Os resultados obtidos nesta dissertação abrem espaço para diferentes frentes de trabalho, tais como:

- Implementação dos controladores propostos sobre um sistema físico ;
- Investigação mais completa sobre a resposta dos controladores desenvolvidos quando atuando sobre sistemas sujeitos a diferentes tipos de ruído, tais como de medição e de atuação;
- Comparar os resultados obtidos pelos controladores desenvolvidos aos obtidos por outros controladores que implementam técnicas de IC e outros que seguem técnicas tradicionais de controle.

Referências

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');

```