

Desenvolvimento de um controlador *fuzzy* para quadrotores

Diego Câmara Sales, Adriano Frutuoso, André Laneira dos Santos

Laboratório de Automação Industrial e Robótica, Departamento de Eletrônica, Telecomunicações e Computação
Universidade Federal do Amazonas

Manaus, Amazonas

Emails: diego@ifam.edu.br, Adriano.frutuoso@gmail.com, adnrehls@hotmail.com

Resumo — Hoje em dia os helicópteros do tipo quadrotor estão se tornando populares no desenvolvimento de pesquisas em veículos aéreos não tripulados, VANT, pela sua capacidade de pouso vertical e de planar em locais de difícil acesso. O controle da maioria destes veículos é baseado em modelos dinâmicos físicos ou experimentais, podendo apresentar um alto custo devido ao tempo de desenvolvimento do projeto. Neste artigo é apresentada o desenvolvimento de um controlador *fuzzy*, que dispensa a necessidade de um modelo matemático para o projeto do controlador de estabilidade, visto que o mesmo é baseado em regras linguísticas. O controle *fuzzy* de um quadrotor tem como vantagem a diminuição do tempo de elaboração e custo do projeto.

Palavras Chave — Desenvolvimento; Controle Fuzzy; VANT; Custo de projeto;

I. INTRODUÇÃO

Os VANTs tornaram-se num objeto de estudo e desenvolvimento depois de terem sido provada as suas capacidades para enfrentarem situações em que por alguma razão o objetivo da missão não fosse a mais aconselhada para uma aeronave de pilotagem convencional, tendo atualmente variadas configurações e capacidades, de acordo com as missões propostas. Entre essas missões, destacam-se situações de reconhecimento e vigilância, onde podem desempenhar um papel importante, na observação de fenômenos meteorológicos e geológicos, onde não é possível a presença humana, campos e fazendas onde existe a necessidade de pulverização de produtos químicos em áreas precisas[2], em laboratórios e universidades onde são usados para fins de investigação científica[3].

Estes veículos aéreos não tripulados têm propulsão própria utilizando forças aerodinâmicas que são responsáveis por sua sustentação e não possuem cabine de pilotagem, pois podem ser controlados à distância ou podem possuir algoritmos sofisticados de voo que não requerem a intervenção humana[4].

O quadrotor é um VANT de asa rotativa composto de quatro rotores localizados nas extremidades de uma estrutura cruzada. O voo do quadrotor pode ser controlado alterando as velocidades de cada rotor. As características construtivas do quadrotor proporcionam vantagens sobre outros VANTs, tais como: a capacidade de pairar no ar, e realizar pequenos movimentos precisos[1].

O controle do quadrotor apresenta dificuldades na definição das características mecânicas e no modelo dinâmico real em virtude de sua não linearidade, controle de multivariáveis, fazendo com que o controle do quadrotor seja um desafio.

O objetivo deste trabalho é apresentar o desenvolvimento de um controlador *fuzzy* capaz de estabilizar o sistema sem a necessidade de ter um modelo dinâmico do quadrotor, e utilizando a menor quantidade de regras linguísticas com o intuito de viabilizar a implementação do controlador em sistemas embarcados de baixa capacidade computacional. Vários estudos foram realizados aplicando-se técnicas de controle *fuzzy*. Alguns trabalhos correlatos foram estudados dentre eles destacam-se: controlador proporcional derivativo sintonizado por lógica *fuzzy* [4] onde o autor implementa o controle do tipo fuzzy PD de um quadrotor efetuando simulações computacionais para comprovar a sua eficácia demonstrando dificuldades na implementação prática do controlador, controlador do tipo PID-like *fuzzy*[5] onde apresenta dificuldades na definição dos ganhos do controlador. Controle de um grau de liberdade utilizando controlador *fuzzy* [6] apresenta o uso de um controlador *fuzzy* com a ausência de ganhos do tipo proporcional e derivativo com resultados experimentais satisfatórios demonstrando a possibilidade de implementar este tipo de controle em quadrotores.

II. QUADROTOR

O quadrotor é um VANT de asa rotativa, conforme ilustrado na Figura 1. Nesta seção será abordada a descrição do seu funcionamento com o intuito de definir as entradas e saídas do sistema buscando efetuar o controle *fuzzy* mediante a observação do seu comportamento.

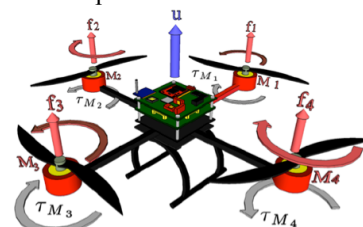


Figura 1. VANT do tipo quadrotor

A. Dinâmica do Sistema

As formulações matemáticas da Equação 1 descrevem o comportamento dinâmico do quadrotor. Neste modelo é possível executar simulações computacionais e análises detalhadas propondo métodos de controle de suas variáveis, assim como analisar o comportamento da planta a ser controlada.

O modelo inclui a resultante dos efeitos giroscópios da rotação do corpo rígido no espaço e a rotação dos quatro propulsores. Além disso, pode ressaltar que o modelo apresenta um comportamento não linear [1].

O quadrotor tem capacidade de rotacionar em três eixos. A rotação é realizada ajustando-se a velocidade de cada rotor em relação aos outros três. A diferença de velocidade entre os pares de motores faz com que ocorra rotação nos eixos de rolagem, roll (ϕ), e arfagem, pitch (θ), e guinada, yaw (ψ), conforme ilustrado na Figura 2.

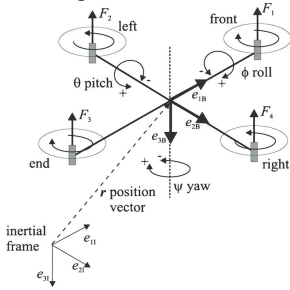


Figura 2. Eixos de rotação, ângulos e forças do Quadrotor. [1]

O modelo apresentado em [1] inclui os efeitos giroscópios resultantes da rotação do corpo rígido no espaço e os quatro propulsores podem ser representados:

Tabela 1. Símbolos e suas definições.

Símbolo	Definição	Símbolo	Definição
ϕ	Ângulo Roll	x	Posição X
θ	Ângulo Pitch	y	Posição Y
ψ	Ângulo Yaw	z	Posição Z
I_x, I_y, I_z	Inércias do Corpo	d	Fator de Arrasto
J_r	Inercia Rotacional	m	Massa do Corpo
Ω_i	Velocidade do rotor	l	Comprimento
b	Fator de Rotação	u_1, u_2, u_3	Sinais de Controle
g	gravidade		

$$\begin{cases}
 \ddot{\phi} = \dot{\phi} \cdot \dot{\psi} \frac{I_y - I_z}{I_x} - \frac{J_r}{I_x} \cdot \dot{\theta} \cdot \Omega + \frac{l}{I_x} \cdot u_2 \\
 \ddot{\theta} = \dot{\phi} \cdot \dot{\psi} \frac{I_z - I_x}{I_y} + \frac{J_r}{I_y} \cdot \dot{\theta} \cdot \Omega + \frac{l}{I_y} \cdot u_3 \\
 \ddot{\psi} = \dot{\phi} \cdot \dot{\theta} \frac{I_x - I_y}{I_z} + \frac{1}{I_z} \cdot u_4 \\
 \ddot{x} = (\cos\phi \cdot \sin\theta \cdot \cos\psi + \sin\phi \cdot \sin\psi) \cdot \frac{1}{m} \cdot u_1 \\
 \ddot{y} = (\cos\phi \cdot \sin\theta \cdot \cos\psi - \sin\phi \cdot \cos\psi) \cdot \frac{1}{m} \cdot u_1 \\
 \ddot{z} = -g + (\cos\phi \cdot \cos\theta) \cdot \frac{1}{m} \cdot u_1
 \end{cases} \quad (1)$$

O sinal de controle do sistema são apresentadas por u_1, u_2, u_3, u_4 conforme apresentado em (2).

$$\begin{cases}
 u_1 = b \cdot (\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\
 u_2 = b \cdot (\Omega_4^2 - \Omega_2^2) \\
 u_3 = b \cdot (\Omega_3^2 - \Omega_1^2) \\
 u_4 = b \cdot (\Omega_2^2 + \Omega_4^2 - \Omega_3^2 - \Omega_1^2) \\
 \Omega = \Omega_1 + \Omega_2 - \Omega_3 - \Omega_4
 \end{cases} \quad (2)$$

O sinal de controle u_1 é responsável pela decolagem do quadrotor. Já o sinal de controle u_2 e u_3 são responsáveis pelo movimento de rolagem e arfagem, respectivamente. E o sinal de controle u_4 é responsável pelo movimento de guinada. Percebe-se que todos os sinais de controle dependem diretamente da velocidade dos motores, Ω_i .

B. Unidade de Medição Inercial

A unidade de medição inercial, ou IMU (*Inertial Measurement Unit*), é uma unidade responsável por executar a medição de velocidade angular e orientação de um objeto utilizando um giroscópio de 3 eixos, um acelerômetro de 3 eixos em conjunto com um magnetômetro de 3 eixos. Utilizando a interface de comunicação I²C é possível acessar as medidas independentes de rotação, aceleração e magnetismo ou podem ser utilizadas para calcular a orientação absoluta do sensor o que permite o controle de atitude (posição angular) de veículos aéreos. O controle do quadrotor é realizado a partir das medidas de uma IMU. Portanto sua confiabilidade e precisão são características desejáveis para o sistema. [9]

As IMUs de baixo custo fazem com que seja um desafio desenvolver algoritmos de pós processamento capazes de suprimir ruídos mecânicos ocasionados por vibrações geradas pela atuação dos motores fixos na carenagem do quadrotor, sendo necessária a implementação de filtros digitais, o que acarreta em um aumento de processamento computacional.

O componente MPU6050, Figura 3, é um sensor capaz de estimar os ângulos de atitude (*roll*, *pitch* e *yaw*) e medir a velocidade angular através de giroscópios. Além disso, possui um filtro digital passa baixa configurável que disponibiliza a leitura dos ângulos de atitude com baixa interferência a ruídos, fazendo com que o sistema de controle não necessite efetuar o tratamento dos dados diminuindo assim o processamento dos dados do microcontrolador.



Figura 3. IMU de baixo custo MPU 6050

O componente também possui internamente um processador digital de movimento ou DMP (*Digital Motion Processor*), que é capaz de processar um algoritmo complexo da fusão de movimento de seis eixos executando os cálculos de posição, movimentação e filtragem sendo necessário efetuar a configuração dos registradores. A sequência de configuração da DPM é apresentada a seguir:

Habilitar o Filtro Passa Baixa de 10Hz.

Definir o valor de offset dos eixos X, Y e Z do giroscópio.

Definir a escala do giroscópio em +/-1000 graus por segundo.

Definir a taxa de atualização da leitura do sensor em 50 Hz

Receber os dados dos ângulos de rotação e posição.

A utilização do filtro digital da DMP apresenta um atraso na leitura dos ângulos de atitude de 13ms, desta forma foi definido a taxa de atualização da leitura do sensor a cada 50Hz. A configuração do filtro digital deve ser feita de forma a sincronizar a taxa de atualização de uma nova leitura e o tempo que a DPM leva para executa a filtragem, fazendo com que a atuação do controle não seja prejudicada.

Os ângulos de atitude estimados provenientes da MPU devem ser estabilizados após um período de tempo de leitura, encontrando assim os valores de referencia atual que devem ser utilizados para definir a posição Zero do sistema. O desenvolvimento de uma estrutura de testes é necessário para proporcionar ao sistema os ângulos de referência de forma confiável, o método utilizado para definir a referência dos eixos foi definido através da criação de um programa capaz de efetuar a leitura dos dados do sensor até ocorrer a estabilidade, após isso é definido o valor offset para que o ângulo de referência inicial do sistema seja zero.

C. Unidade de Controle (Sistema Embarcado)

A plataforma utilizada para efetuar o controle do sistema é o Arduino due, que é baseada no microcontrolador ARM Cortex M3 AT91SAM3X8E de 32 bits. Possui 54 pinos digitais sendo 12 saídas PWM configuráveis em 8 e 12 bits de resolução, 12 entradas analógicas, 2 conversores Digital para Analógico, 4 interfaces serial, interface SPI, interface I2C e uma frequência de processamento de 84MHz.

Este microcontrolador possui 512KBytes de memória Flash e uma memória RAM de 96KBytes (sendo dois bancos: 64KB e 32KB). Possui uma interface de programação USB e linguagem de programação baseada em C da plataforma Arduino. Foi escolhida esta plataforma porque possibilita configurar a resolução de saída do controlador, sendo efetuado testes de comportamento do sistema com a resolução de 8 e 12 bits.

D. Controlador de Velocidade e Atuadores

Os atuadores do sistema são responsáveis pela movimentação do quadrotor, o atuador utilizado é o de corrente contínua do tipo sem escova, BLDC - *Brushless Direct Current*, aplicado em aeromodelismo sendo capaz de desenvolver o empuxo necessário para atuar no controle e estabilização. Os motores são acionados através de um controlador de velocidade eletrônico que efetua o controle de velocidade e acionamento dos motores.

O Controlador Eletrônico de velocidade, ESC, recebe os dados da modulação por largura de pulso, PWM, e controla a atuação dos motores alimentando os polos na sequência correta para o seu devido funcionamento, a frequência de atualização do motor é de 50Hz limitada pelo ESC, com isso o sistema de controle deve ser executado em um período máximo de 20ms para que possa atuar no motor ciclicamente, a Figura 5 apresenta o sistema proposto para executar o controle e ajustes do Offset dos motores.

Os motores utilizados são idênticos mas possuem características mecânicas e elétricas sutilmente diferentes, sendo necessário fazer o balanceamento ou definição de um valor de offset para efetuar a compensação entre estes.

Uma das formas de efetuar a verificação da rotação do motor é utilizando um sensor de rotações por minuto, *rpm*, podendo assim ajustar o offset de cada motor. O sensor de *rpm* utilizado é o modelo Hobbywing brushless RPM que efetua a detecção da alteração de voltagem do controlador de velocidade do motor podendo medir velocidades entre 1000 a 300000 RPM. A interpretação do sinal deste sensor é feita relacionando a quantidade de pólos do motor sendo necessário efetuar uma relação entre pólos e *rpm*.

A leitura proveniente do sensor RPM é feita através da quantidade de polos do motor e a contagem dos pulsos de uma rotação completa, sendo possível estimar a relação entre o valor de PWM e RPM.

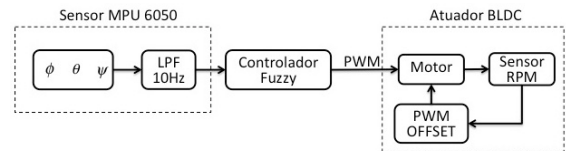


Figura 5. Sistema de controle proposto

A geração do empuxo é criada a partir da rotação do motor em conjunto com uma hélice fixada em seu eixo de rotação. A escolha da hélice deve ser feita mediante a estrutura física do quadrotor seguindo as características descritas pelo fabricante sendo necessário efetuar o balanceamento das hélices para que os empuxos gerados pelos motores sejam aproximadamente iguais, podendo efetuar o controle de acordo com a estratégia a ser implementada.

E. Plataforma de Testes

O hardware utilizado para implementação do sistema de controle de estabilidade é baseado em componentes de baixo custo utilizados em aeromodelismo. A comunicação entre o quadrotor e o computador foi feita com a utilização de duas placas Xbee S1 onde uma envia os dados do sistema e a outra recebe os dados no computador podendo ser visualizado através da comunicação serial. A Figura 10 apresenta a plataforma de testes utilizada em laboratório para desenvolvimento da pesquisa no sistema.



Figura 9. Plataforma de Testes.

A forma escolhida para executar os testes de estabilidade do sistema deve deixar a estrutura física livre de tal forma que possa efetuar movimentos em torno do seu eixo sem que ocorra interferências provenientes da estrutura adotada. O método mais barato e com baixa interferência é com a utilização de barbantes presos em todos os braços da estrutura do quadrotor.

III. CONTROLE DE ESTABILIDADE INTELIGENTE

O controle de estabilidade utilizando logica *fuzzy* tem como entradas os sinais de atitude roll, pitch e yaw que são responsáveis pela estabilidade do sistema. O controle destes ângulos é feito de forma independente, sendo necessário a implementação de três controladores *fuzzy* onde suas entradas são o erro do ângulo e a leitura da aceleração do giroscópio de cada eixo para garantir a rápida atuação do sistema mediante a movimentações em um curto período de tempo.

A altura é controlada utilizando um valor de referência definido como offset de altura, a figura 6 apresenta o controlador proposto com suas entradas e saídas.

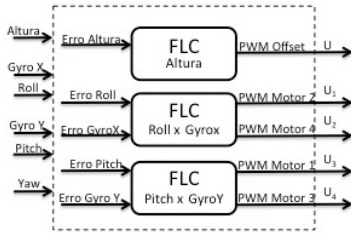


Figura 6. Proposta do Controlador Lógico *Fuzzy*

A saída do controlador é o sinal de controle aplicado diretamente aos controladores de velocidade que acionam os motores, através da análise de funcionamento pode ser afirma que os sinais de saída dependem do angulo yaw e dos pares de motores M_1, M_3 e M_2, M_4 assim como um valor de offset para a altura desejada de acordo com a Fórmula 3.

$$\begin{aligned} PWM_{motor1} &= PWM_{offset} + PWM_{roll} - PWM_{yaw} \\ PWM_{motor2} &= PWM_{offset} + PWM_{pitch} + PWM_{yaw} \\ PWM_{motor3} &= PWM_{offset} - PWM_{roll} - PWM_{yaw} \\ PWM_{motor4} &= PWM_{offset} - PWM_{pitch} + PWM_{yaw} \end{aligned} \quad (3)$$

O controlador *fuzzy* é composto seguindo quatro elementos, conforme apresentado em sequência.[7]

Módulo de fuzzificação: Modela matematicamente a informação das variáveis de entrada por meio de conjuntos *fuzzy* conhecidos como Funções de Pertinência.

Módulo da base de regras : Constitui as regras utilizadas para definir a performance e comportamento do sistema, informando as variáveis de entrada e saída assim como as suas classificações linguísticas.

Módulo de inferência: Define quais são os conectivos lógicos usados para estabelecer a relação *fuzzy* que modela a base de regras utilizada.

Módulo de defuzzificação: Converte a conclusão do modulo de inferência das variáveis de saída *fuzzy* para um valor numérico.

A implementação do controle segue as seguintes características:

Módulo de inferência do tipo Mandami, três ou mais funções de Pertinência do tipo Triangulares e Trapezoidais. Regras *Fuzzy* seguindo a forma se(erro angulo) e (erro giroscópio) então(PWM_{motor_x}) recebe defuzzificação. Método de defuzzificação do tipo Centro de Gravidade(Centroid) e Operadores e(AND) como mínimo.

A. Definição das funções de pertinência: Entradas e Saídas

A definição das funções de pertinência é uma fase importante no desenvolvimento dos controladores *fuzzy* pois as características do controlador são definidas apartir da quantidade de funções utilizadas e sua posição em relação as outras classes. Alguns procedimentos são efetuados de forma empírica pelo método da tentativa e erro, o que pode ser bastante lento dependendo da complexidade do sistema que se proponha a controlar. Estudos conhecidos na literatura apresentam métodos que podem obter melhores resultados em relação ao método empírico como SARSA, Reinforcement Learning , Q-learning dentre outros.[6]

A definição das funções de pertinência utilizadas para a implementação das regras dos controladores foram baseadas na respostas das funções de transferência do sistema dinâmico do quadrotor baseado nos estudos apresentados no artigo de [9] que implementou testes de forma simulada do comportamento de um controlador do tipo *fuzzy* aplicado a diversos tipos de plantas.

As funções de pertinência descrevem o comportamento do sistema mediante a classificação de suas variáveis de controle, na Figura 7 é apresentada as funções de pertinência aplicadas ao angulo roll e ao giroscópio, onde são divididos em três classes Negativo, Zero e Positivo com o formato triangular e trapezoidal.

A classificação Negativo e Positivo apresenta a largura de atuação do controlador na faixa de ângulos negativos e positivos. A classe Zero apresenta os valores em que o controlador deve buscar sua estabilidade de acordo com a definição do método centroide, onde o grau de pertinência da classe Zero é 1 no angulo zero.

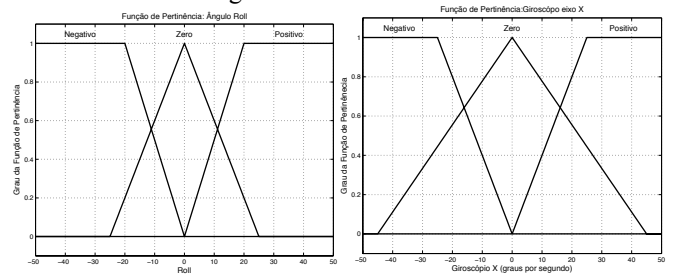


Figura 7. Funções de Pertinência do Ângulo de Entrada Roll e giroscópio

A função de pertinência da saída do sistema de controle é feita em uma faixa de [0,50] do sinal de PWM, conforme Figura 8. A alteração da faixa de operação do controlador para faixas de valores maiores afeta diretamente no comportamento do controlador causando alguma oscilação.[10]

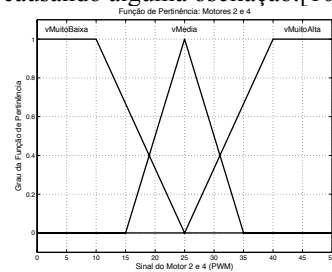


Figura 8. Funções de Pertinência das Saídas.

B. Implicação Fuzzy

As regras linguísticas do controlador *fuzzy* descrevem o comportamento do sistema mediante combinações das entradas dos ângulos a serem controlados através de conectivos lógicos. A base de regras representa as ações a serem tomadas mediante as possíveis entradas, sendo necessário o conhecimento especializado do comportamento da dinâmica do sistema. Foi utilizado inicialmente 9 regras linguísticas aplicadas pela combinação das variáveis de entrada, conforme Tabela 2.

Tabela 2. Começo de uma Tabela de uma Única Coluna.

Motor 2 e Motor 1				Motor 4 e Motor 3			
ϕ/ψ	N	Z	P	θ/ψ	N	Z	P
N	VA	VA	VM	N	VM	VB	VB
Z	VA	VM	VB	Z	VA	VM	VB
P	VM	VB	VB	P	VA	VA	VM

Note que a abreviação das entradas são: “N” para “Negativo”, “Z” para “Zero”, “P” para “Positivo” e as regras geradas para as saídas do sistema são: “VA” para “Velocidade Alta”, “VM” para “Velocidade Média”, “VB” para “Velocidade Baixa”.

IV. IMPLEMENTAÇÃO DO CONTROLE EM SISTEMA EMBARCADO

A implementação do controle em sistema embarcado foi feita mediante a utilização da biblioteca eFLL (Embedded Fuzzy Logic Library) que é escrita em C++/C possibilitando a implementação em sistemas embarcados.

O uso desta biblioteca apresenta limitações em sistemas microcontrolados de 8 bits sendo indicado o uso em 32 bits e por este motivo optamos utilizar a plataforma Arduino Due conforme apresentado anteriormente.

A. Configuração do Controlador fuzzy

A configuração do controlador deve ser feita da seguinte forma:

Definir as funções de pertinência das entradas e saídas.

Relacionar as funções de pertinência seguindo a seguinte lógica: funções de pertinência da entrada com as três entradas dos sensores e funções de pertinência da saída com os quatro motores do sistema.

Criar as regras que antecedem a ação do controlador *fuzzy* definindo todas as possibilidades de leitura do sensor, exemplo: roll sendo negativo, zero e positivo, giroscópio x sendo negativo, zero e positivo.

Criar as regras de consequência das entradas do controlador *fuzzy* com seus conectivos, exemplo: roll negativo e giroscópio x negativo. Relacionar as regras de consequência com as saídas a serem atuadas.

Criação das regras de como o controlador deve se comportar relacionando regras que antecedem ao controlador e suas consequências. Após a configuração do controlador, deve ser indicado as entradas e saídas do sistema relacionando com

as variáveis provenientes dos sensores e dos motores para que possa ser feito a fuzzificação e defuzzificação.

O controle de estabilidade do ângulo de atitude roll pode ser observado na Figura 10 onde são apresentados os gráficos da velocidade angular, ângulo roll e as saídas dos motores 2 e 4 do sistema implementados na plataforma arduino.

B. Resultados Experimentais

A resposta do sistema é apresentando a seguir onde foi definido o ângulo de referência setpoint em 0 graus com resolução do PWM em 8bits e 50% da potência máxima aplicada aos motores do sistema, simulando assim a decolagem do quadrotor. A Figura 9 apresenta o sinal de controle do giroscópio X onde observa-se a oscilação do sistema na faixa entre [-20,20] graus por segundo demonstrando a atuação do controlador.

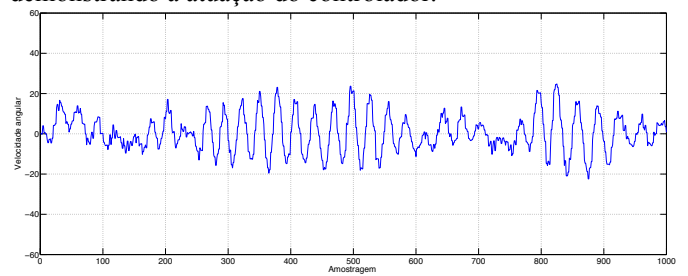


figura 9. Velocidade Angular do Giroscópio X

O controle do ângulo de atitude roll demonstra uma variação entre -5 e 6 graus podendo obter melhores resultados com a utilização de métodos de ajuste das funções de pertinência do sistema. A taxa de amostragem utilizada é de 50hz, demonstrando a ação do controlador em um tempo de 20 segundos.

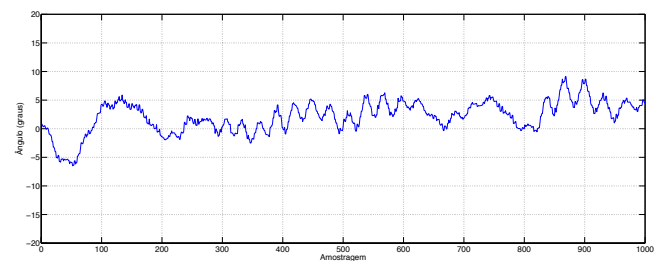


Figura 10. Ângulo Roll

O sistema embarcado da plataforma de testes possibilitou a execução de testes com a resolução do sinal PWM em 8 e 12 bits. Desta forma pode ser visto a ação do controlador com estas resoluções, onde na Figura 11 a cor azul representa o sinal de controle com 8 bits e o sinal vermelho a resolução de 12 bits.

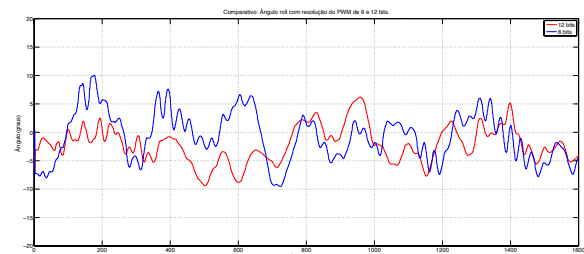


Figura 11- Comparação de resultados com resoluções de 8 e 12 bits do PWM.

O comportamento do controle pode ser visto na figura 12. A utilização do sinal PWM em 12 bits demonstrou uma grande melhoria em relação ao de 8 bits, pois a taxa de variação do ângulo de controle com o sinal de PWM em 12 bits demonstrou ser melhor aproximadamente 64% em relação a resolução de 8 bits apesar de ser menos eficiente ao buscar o setpoint em 0 graus. Este comportamento se dá mediante o aumento da quantidade de velocidades que o motor pode atuar, com 8 bits temos 256 possibilidades de velocidade contra 4096 da resolução de 12 bits, apresentando maior suavidade na resposta do sistema.

O controle do ângulo de atitude roll demonstra uma variação entre -5 e 6 graus onde a taxa de amostragem utilizada é de 50hz sendo executado 32 segundos de testes na Figura 12. Na Figura 13 pode ser observado o comportamento do controlador *fuzzy* roll ao ser aplicado uma perturbação simulando uma colisão com um obstáculo ou outros tipos de distúrbios, onde a atuação do controlador demonstra ser satisfatória com um tempo de acomodação menor que 100 amostras, 1.8 segundos.

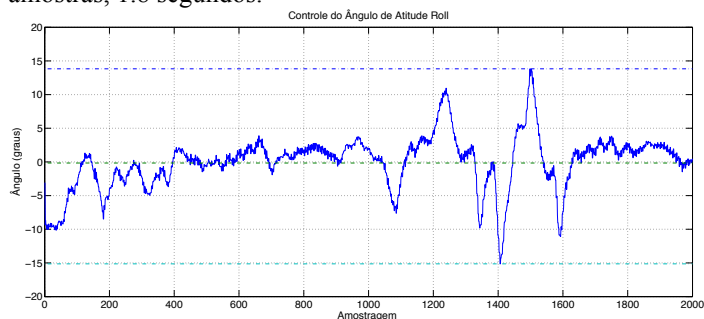


Figura 13. Controle do Ângulo de atitude roll e giroscópio do eixo X, 12 bits.

A escolha da quantidade de regras linguísticas utilizadas foram baseadas no trabalho[11] onde é apresentando que, o numero de funções de pertinência influencia na suavidade da resposta e precisão do sistema de controle nebuloso. No entanto, um número elevado de funções de pertinência ocasiona um aumento no número de regras para o controlador, tornando complicada a sua sintonia, com somente nove regras é possível resolver satisfatoriamente a maioria dos problemas de controle, como pode ser visto no desenvolvimento deste trabalho.

V. CONCLUSÕES

O controle de estabilidade do quadrotor apresenta grandes desafios na compreensão das características físicas e elétricas dos componentes além do seu comportamento não linear acarretando em métodos de controle mais robustos e maior capacidade de processamento. A grande maioria dos trabalhos executados descrevem o controle das seis variáveis de forma independente, neste trabalho foi apresentado um método de simplificação dos controladores utilizando apenas três controladores inteligentes *fuzzy* com o uso de três funções de pertinência demonstrando resultados satisfatórios na estabilização do sistema e mediante a perturbações.

O sistema embarcado utilizado para controlar o quadrotor possibilitou a configuração da resolução do sinal de controle

PWM que pode trabalhar entre 8 e 12 bits, experimentos comprovaram a melhoria no comportamento do controlador *fuzzy* com 12 bits deixando o sistema com uma resposta mais suave e linear e também melhor resposta a perturbações em comparação com 8 bits.

O controlador Embarcado apresentou algumas dificuldades em relação ao tempo de resposta do sistema pois a atualização dos motores depende da frequência de trabalho do ESC escolhido, que é de 50Hz tendo um período de atualização de 20ms na ação dos motores afetando o controle ao ter de atuar rapidamente em uma variação de angulação em um curto período de tempo. Os trabalhos futuros a serem desenvolvidos buscam melhorar o tempo de resposta do sistema utilizando um ESC de 400Hz fazendo com que o sistema possa efetuar o controle com um menor tempo de amostragem.

REFERENCIAS

- [1] S. Bouabdalla, Design and control of quadrotors with application to autonomous flying. PhD thesis, EPFL, 2007.
- [2] R. Sugiura, T. Fukagawa, N. Noguchi, K. Ishii, Y. Shibata, K. Toriyama, Field information system using an agricultural helicopter towards precision farming. Proceedings of 2003 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, 2 1073 1078, July 2003.
- [3] Syed Ali Raza and Wail Gueaieb. Intelligent Flight Control of an Autonomous Quadrotor, Motion Control, Federico Casolo (Ed.), 2010.
- [4] S. Costa, "Controle e Simulação de um Quadrirotor convencional", em Dissertação para obtenção do Grau de Mestre em Engenharia Aeroespacial. IST- Portugal, 2008.
- [5] M. Santos, V. López, F. Morata, "Intelligent Fuzzy Controller of a Quadrotor" em IEEE International. Conference on Intelligent Systems and Knowledge Engineering, Hangzhou, 2010.
- [6] S. Zangenehpour, B. Dehghan, B. Asaei, "Controlling a One Degree of Freedom Arm, Using Fuzzy Controller", em International Conference on Information, Networking and Automation (ICINA), 2010.
- [7] M. Amendola, A. Souza, L. Barros, Manual do uso da teoria dos conjuntos fuzzy no Matlab, Apostila IMECC e UNICAMP, 2005.
- [8] K. M. Passino and S. Yurkovich, livro Fuzzy Control. Menlo Park CA: Addison Wesley Longman, 1998.
- [9] <http://invensense.com/mems/gyro/mpu6050.html>
- [10] K. Gowrishankar, V. Elancherathan, "Adaptive Fuzzy Controller to Control Turbine Speed", Ubiquitous Computing and Communication Journal, Puducherry, India.
- [11] J. C. Netto, Controladores Nebulosos Aplicados a Processos Industriais: Estudo Comparativo de Sintonia. Dissertação de Mestrado, Universidade Federal de Juiz de Fora, 2005.