

ACQUISITION AND PROCESSING FOR CONTROL OF A QUADCOPTER
**Anderson Weber Schenkel, Mauro Fonseca Rodrigues, Nadine Lambrecht Szambelan,
 Robson Lopes, Sandi da Costa Gehm**

Universidade Regional do Noroeste do Estado do Rio Grande do Sul - UNIJUÍ

Departamento de Ciências Exatas e Engenharias – Ijuí - RS

anderson.wschenkel@hotmail.com, mauro.rodrigues@gmail.com, nadine.sz@hotmail.com,
robsonletric@outlook.com, sandigehm@yahoo.com.br

Abstract. The purpose of this work is make the acquisition and processing for control of a quadcopter. One modeling Will be presented, followed by the system architecture, acquisition and signal processing and quadcontrol. Closing phases of the Project, experimental testes were carried out in order to analyse the ir behavior.

Keyword: *Quadcopter, Brushless, Controler.*

1. INTRODUÇÃO

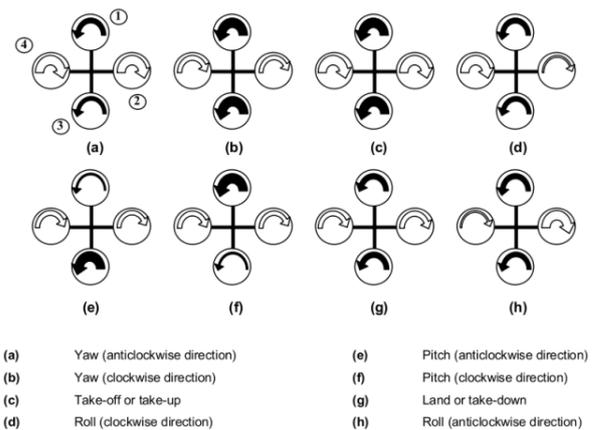
O avanço tecnológico de sistemas computacionais inteligentes tem sido de tal forma amplamente conhecido em diferentes áreas, onde se notabilizou também um aumento de utilização e desenvolvimento de UAVs (*Unmanned Aerial Vehicles*) para diversas aplicações como, por exemplo, exploração de zonas de difícil acesso e em aplicações militares como combate aéreo, transporte e estabilização e voo estacionário.

2. PRINCIPIO DE FUNCIONAMENTO

O quadricóptero é uma aeronave impulsionada por quatro motores que apresentam uma configuração em cruz, simetricamente distribuídos ao longo das extremidades das estruturas, orientados verticalmente, com capacidade de aterrissagem e descolagem vertical. Dois dos rotores giram no sentido dos ponteiros do relógio (CW) e os outros dois no sentido contrário (CCW), de forma que a velocidade angular da aeronave seja igual a 0.

Variando as velocidades dos motores é possível movimentar a aeronave. Na figura 1 estão representadas as diferentes movimentações possíveis, assim como as combinações das velocidades dos motores para executar esses movimentos.

Figura 1 - Movimentos do quadricóptero



Fonte: Moudgalya, 2007.

2.1 Modelo Cinemático

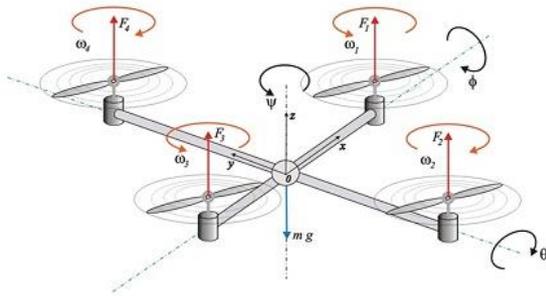
Para analisar o quadricóptero serão utilizados os ângulos de inclinação (*pitch*), rotação (*roll*) e guinada (*yaw*), para obter a matriz de rotação R que será utilizada para relacionar com o referencial da terra através da seguinte equação [1]:

$$r^n = \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} + R \begin{bmatrix} 0 \\ 0 \\ F \end{bmatrix} m^{-1} \quad (1)$$

Sendo que g é a aceleração gravitacional e F representa a soma das forças que são geradas pelos motores. A velocidade angular do quadricóptero é 0, se as velocidades das hélices forem iguais, mas para se conseguirem se movimentar é necessário criar uma aceleração angular igual a:

$$l \begin{bmatrix} p' \\ q' \\ r' \end{bmatrix} = \begin{bmatrix} l(f_2 - f_4) \\ l(f_1 - f_3) \\ k(f_1 - f_2 + f_3 - f_4) \end{bmatrix} - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times l \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (2)$$

Figura 2 - Modelo do quadricóptero.



Fonte: Amorim e Duarte, 2014

3. SENSORES E ATUADORES

3.1 Motores

O quadricóptero disponível no laboratório utiliza quatro motores do tipo *brushless* *turnigy aerodrive*. Os motores *brushless* são mais silenciosos, mais confiáveis e têm uma vida útil maior que os motores de corrente contínua normais.

As características do motor são apresentadas na tabela 1:

| | |
|-----------------------|------|
| Kv (rpm/v) | 1050 |
| Operating Current (A) | 4~12 |
| Max Current (A) | 15 |
| Max Voltage (A) | 11 |

Tabela 1–Características do Motor Turnigy Aerodrive.

3.2 Controladores

Os motores *brushless* necessitam de controladores próprios (ESC) para fornecer a alimentação aos motores. A ambientação é controlada através do sinal PWM que o arduino envia para o ESC. O quadricóptero do laboratório utiliza quatro controladores *turnigy plush* de 25A conforme figura 3, com as seguintes características:

Cont Current: 25A
 Burst Current: 35A
 BEC Mode: Linear
 BEC : 5v / 2A
 Lipo Cells: 2-4
 NiMH : 5-12
 Weight: 22g
 Size: 24x45x11mm
 Supported motor speed for all TURNIGY ESCs is;
 (Maximum): 210000 RPM (2 poles), 70000 RPM (6 poles), 35000 RPM (12 poles)

Figura 3 - Eletronic SpeedControl (ESC).



Fonte: Beginners Guide to Connecting Your RC Plane Electronic Parts

3.3 Plataforma de montagem

A plataforma escolhida para servir como base para a montagem deste trabalho foi o Arduino Uno representado na figura 3. Esta placa funciona a 16 MHz e suporta ligação I2C.

3.4 Sensores MPU6050

É necessário um sensor que permita obter algum tipo de informação relativamente à localização espacial do quadricóptero. O acelerômetro é um sensor que, para um objeto estático, indica o valor da aceleração linear em três eixos e portanto a sua direção. Como ele aponta sempre para o centro da Terra, consegue-se saber sempre a inclinação do acelerômetro com alguma precisão, se o objeto não estiver se movimentando. Numa translação de um objeto, estamos criando uma força e, portanto alterando a sua aceleração. Outro problema dos acelerômetros é a sua sensibilidade a vibrações. Isto faz com que o acelerômetro, por si só, não seja suficiente para a localização espacial do quadricóptero. Se for adicionado um giroscópio, que mede a velocidade angular se torna possível contrariar este problema.

4. AQUISIÇÃO E PROCESSAMENTO DE SINAIS

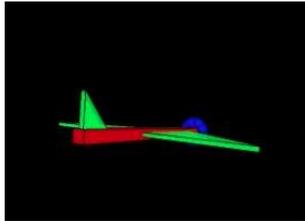
4.1 Testes

De forma a comprovar o sistema de aquisição implementado antes do seu uso no quadricóptero, foram realizados testes com o sistema em repouso e o seu movimento 3-D espacial. Foi utilizada a ferramenta em Java

Processing que nos permitiu a simulação visual 3-D/espacial do sensor e, deste modo, testar visualmente.

Foi possível observar um tempo de calibração necessário para o *Yaw* estabilizar, enquanto que para o *Roll* e para o *Pitch* não houve necessidade e correspondiam com algum sucesso ao esperado.

Figura 4: Movimento UAV em simulação



Fonte: Software Java

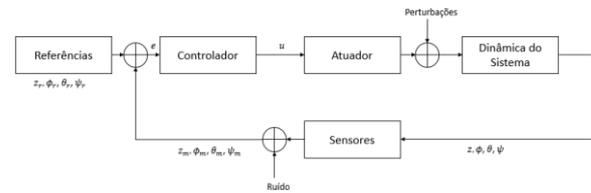
Posteriormente, foram realizados testes ao sensor montado no quadricóptero e observou-se alguma oscilação nos valores obtidos, os quais não foram significativos para o resultado esperado.

4.2 Processamento e Controle

Como referido anteriormente, o controle foi realizado em plataforma Arduino. Procedeu-se procurar controlar o sistema recorrendo a PID ou a uma implementação *fuzzy* [2]. Estas abordagens não se revelaram propícias e acabou por não levar essas implementações do papel para a prática, limitando-se a um conjunto de funções em malha aberta que implementava as movimentações básicas.

Foi analisado o filtro complementar. Este oferece uma razoável filtragem dos sinais obtidos dos sensores. Por aplicação de um filtro passa baixo digital ao sinal do giroscópio e um filtro passa alto ao sinal do acelerômetro podemos obter sinais razoáveis e dignos de comparação com métodos mais elaborados como o filtro de Kalman [3].

Figura 5 - Diagrama de blocos do modelo.



Fonte: Elaboração do autor

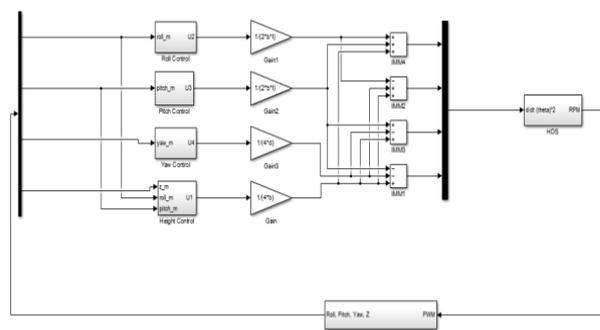
Tem de ser utilizado em ciclo infinito. Em cada interação é atualizado o valor dos ângulos de *pitch* e *roll* pelos novos valores obtidos pelo giroscópio. A seguir é comparado com os valores obtidos pelo acelerômetro, assegurando que o valor não se desvia num curto espaço temporal provocado por perturbações externas [4].

$$\hat{\text{Ângulo}} = 0.98 * (\hat{\text{ângulo}} + \hat{\text{ângulogiroscópio}} * dt) + 0.02 * \hat{\text{ânguloacelerômetro}}$$

Foi analisado o filtro de Kalman. Este utiliza um modelo dinâmico do sistema em causa de forma de estimar os estados do sistema a partir de medições efetuadas ao longo do tempo, por exemplo do giroscópio e acelerômetro. Como estas medições contêm ruídos, o filtro tenta reduzir a sua presença através de estimações do estado do sistema.

Escolhendo um valor para referência do ângulo de *Yaw*, é feita a diferença entre o valor medido e o valor atual, procedendo ao cálculo da tensão que irá corrigir o desvio encontrado.

Figura 6- Esquema modelo da simulação do sistema



Fonte: Elaboração do autor

O objetivo é apresentado em cima, onde existem quatro PID's para controle dos ângulos e da altura em função de uma referência [5]. Seguidamente são feitas operações para obter as RPM para cada motor. Por fim, foram implementadas as seguintes funcionalidades:

- Movimentação básica: para cima, para baixo, esquerda, direita, frente e, para trás;
- Movimentação peculiar: girar no sentido horário, girar contador no sentido anti horário. *Rotate*;
- Armamento dos motores - Biblioteca *Servo* e função *writemilliseconds*, passando como argumento o valor das rotações por minuto pretendidas. A cada primeira utilização é necessário passar as rotações por minuto (RPM) máximas, ouvir um bip, passar as RPM mínimas, ouvir outro bip, e só aí fica concluída a armação. A partir deste momento, os motores estão prontos a arrancar a RPM entre os valores colocados na armação.

Ainda era possível a escolha manual das RPM e a visualização dos ângulos no terminal do Arduino [6].

5. CONSIDERAÇÕES FINAIS

O grande tema deste trabalho era a projeção e desenvolvimento de um trabalho no âmbito da aquisição e processamento de sinal. Inicialmente foi efetuado o estudo da arte do hardware disponível. Foi previsto uma dificuldade em obter a localização espacial do quadricóptero, e esta apenas foi conseguida de forma relativa através de sensores que apresentaram uma boa estabilidade. A nível de controle, o projeto ficou um pouco aquém do desejado. No entanto, a análise efetuada, só por si, foi extremamente educativa, aliando o controle por aproximações lineares para um PID a um controle não linear por *Fuzzy Logic*.

Efetuada os testes, comprovou-se que o Quadricóptero não apresentava o necessário para um voo autónomo e seguro, pois apresentava alguma flexibilidade nos braços de alumínio permitindo alguma

rotação dos motores, não existindo equilíbrio.

REFERÊNCIAS

- [1] Duarte, A., Cerqueira, J., Amorim, F. (2014) *EEC0149 Relatório de Estudantes: Localização espacial de um objeto xyz*, MIEEC, FEUP;
- [2] M. Santos, V. López e F. Morata (2010) *Intelligent Fuzzy Controller of a Quadrotor*, IEEE Int. Conf. On Intelligent Systems and Knowledge Engineering, Hangzhou, China [November 2010]
- [3] José Sousa (2011) *Simulação e Desenvolvimento de um Veículo Aéreo Autónomo de Quatro Rotores*, Dept. Engenharia Electrotécnica e de Computadores, Faculdade de Engenharia da Universidade do Porto, Porto, Portugal.
- [4] Moreira, Eduardo J., Ferreira, Filipe O., Oliveira, João B. (2013) *EEC0149 Relatório de Estudantes: Controlo do Roll, Pitch e Yaw de um Quadrotor*, MIEEC, FEUP;
- [5] Moudgalya, Kannan M. (2007) *Digital Control*, WILEY;
- [6] Sturm, Jürgen, Cremers, Daniel e Kerl, Christian (2014), *AUTONAVx Autonomous Navigation for Flying Robots*, TUMx;