

UNIVERSIDADE FEDERAL DE OURO PRETO  
INSTITUTO DE CIENCIAS EXATAS E APLICADAS  
DEPARTAMENTO DE COMPUTAÇÃO E SISTEMAS

JONATHAS ROCHA DOS SANTOS

**MODELAGEM E IDENTIFICAÇÃO DE PARÂMETRO DE UM VEÍCULO AÉREO  
NÃO-TRIPULADO (VANT)**

João Monlevade

2018

JONATHAS ROCHA DOS SANTOS

**MODELAGEM E IDENTIFICAÇÃO DE PARÂMETRO DE UM VEÍCULO AÉREO  
NÃO-TRIPULADO(VANT)**

Monografia apresentada ao curso Engenharia da computação do Instituto de Ciências Exatas e Aplicadas, da Universidade Federal de Ouro Preto, como requisito parcial para aprovação na Disciplina “Trabalho de Conclusão de Curso II”.

Orientador: Edgard Gregory Torres Saravia

João Monlevade

2018

R672m Rocha, Jonathas.  
Modelagem e identificação de parâmetro de um veículo aéreo não-tripulado (VANT) [manuscrito] / Jonathas Rocha. - 2018.

45f.:

Orientador: Prof. MSc. Edgard Gregory Torres.

Monografia (Graduação). Universidade Federal de Ouro Preto. Instituto de Ciências Exatas e Aplicadas. Departamento de Computação e Sistemas de Informação.

1. Análise de sistemas . 2. Modelagem . 3. Modelos matemáticos. I. Gregory Torres, Edgard. II. Universidade Federal de Ouro Preto. III. Título.

CDU: 004.41

Catálogo: [ficha.sisbin@ufop.edu.br](mailto:ficha.sisbin@ufop.edu.br)



MINISTÉRIO DA EDUCAÇÃO  
UNIVERSIDADE FEDERAL DE OURO PRETO  
REITORIA  
INSTITUTO DE CIÊNCIAS EXATAS E APLICADAS  
DEPARTAMENTO DE ENGENHARIA ELETRICA

**FOLHA DE APROVAÇÃO**

Jonathas Rocha dos Santos

**Modelagem e identificação de parâmetro de um veículo aéreo Não-Tripulado(VANT)**

Membros da banca

Edgard Gregory Torres Saravia - Mestrado - UFOP  
Alexandre Magno de Souza - Mestrado - UFOP  
Tatiana Alves Costa - Doutorado - UFOP

Versão final

Aprovado em 17 de julho de 2018

De acordo

Edgard Gregory Torres Saravia



Documento assinado eletronicamente por **Edgard Gregory Torres Saravia**, COORDENADOR DO CURSO DE ENGENHARIA ELETRICA, em 21/03/2020, às 16:34, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site [http://sei.ufop.br/sei/controlador\\_externo.php?acao=documento\\_conferir&id\\_orgao\\_acesso\\_externo=0](http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0), informando o código verificador **0045005** e o código CRC **253F30EF**.

**Referência:** Caso responda este documento, indicar expressamente o Processo nº 23109.002731/2020-39

SEI nº 0045005

R. Diogo de Vasconcelos, 122, - Bairro Pilar Ouro Preto/MG, CEP 35400-000  
Telefone: - www.ufop.br

## **DEDICATÓRIA**

A Deus, pois, sem ele não teria forças para essa longa jornada, a meus pais, pela capacidade de acreditar em mim

## **AGRADECIMENTOS**

Aos meus pais, pela paciência e compreensão que tiveram fazendo com que chegasse a vencer nos momentos de dificuldade em diversas situações, por todo apoio e conselhos, o meu muito obrigado.

Aos professores do curso de Engenharia da Computação, por todos os ensinamentos passados que contribuíram para minha formação profissional e pessoal.

Ao Professor orientador Edgard Gregory Torres Saravia, pela dedicação, companheirismo e profissionalismo durante toda a execução deste trabalho. A meus amigos, namorada e todos que fizeram parte direta ou indiretamente da minha vida nesta longa caminhada, longos anos de muita luta e persistência

O medo de realizar a tarefa consome mais  
tempo e energia que a própria tarefa

(Autor desconhecido)

## RESUMO

O estudo envolvendo quadricóptero vem ganhando destaque nesses últimos anos, pesquisas envolvendo a aplicação do objeto em tarefas repetitivas estão sendo empregadas nos mais diversos fins, nesse contexto o presente trabalho visa levantar um análogo matemático para o quadrimotor do tipo Ar.drone 1.0. Análogo matemático é uma peça chave para empregar a autonomia do veículo, para chegar nesse ideal, é apresentado uma breve revisão bibliográfica da sua operação e seus aspectos de construção, a partir disso, entra no âmbito da identificação do sistema e mostra algumas alternativas de modelos matemático sugeridos por outras fontes bibliográficas. Mediante esse histórico, é feito um modelo ARX (autoregressivo com entradas exógenas) onde não exige grande conhecimento acerca do equacionamento do sistema. Com isso, foi feito, a excitação de cada entrada do Ar.Drone e analisado o comportamento de suas respectivas saída com o software MATLAB e a toolbox de identificação de sistema. Como resultado da identificação, é sugerido modelos ARX, para as dinâmica de arfagem, rolagem, guinada e altitude a partir das entradas de excitação já considerando os efeitos do controlador interno.

**Palavras-chave:** quadrimotor, modelagem e identificação de sistemas, arx, ardrone

## **ABSTRACT**

The study involving quadricoptero has been gaining prominence in recent years, researches involving the application of the object in repetitive tasks are being used in the most diverse purposes, in this context the present work aims to raise a mathematical analogue for the four-dimensional type Ar.drone 1.0, mathematical analogue is a key piece to employ the autonomy of these vehicles. To reach this ideal, a brief bibliographic review of its operation and its construction aspects is presented. It goes on to identify the system and shows some alternatives of mathematical models suggested by other bibliographic sources. Through this history, an ARX (autoregressive with exogenous inputs) model is made where it does not require great knowledge about the system equation. With this, the excitation of each ArDrone input was made and the behavior of its respective output analyzed with the MATLAB software and the system identification toolbox. As a result of the identification, ARX models are suggested for the dynamics of pitch, roll, yaw and altitude from the excitation inputs already considering the effects of the internal controller.

**Keywords:** quadcopter, modeling and identification of systems, ardrone

## LISTA DE FIGURAS

Figura 1 - <b>Ar.Drone com involucro de proteção à esquerda e sem a direita</b> .....	19
Figura 2 - <b>As cores da proteção forma uma etiqueta tag no AR.Drone</b> .....	22
Figura 3 - <b>Imagem capturada pela câmera frontal utilizando a biblioteca AR.Drone SDK 2.0 do AR.Drone 1.0</b> .....	22
Figura 4 - <b>Modelo simplificado do drone</b> .....	28
Figura 5 – <b>altitude</b> .....	28
<b>Figura 6 – rolagem</b> .....	29
<b>Figura 7 – arfagem</b> .....	29
Figura 8 – <b>guinada</b> .....	30
Figura 9 – <b>Resultado da dinâmica de rolagem</b> .....	37
Figura 10 - <b>Resultado da validação arfagem</b> .....	38
Figura 11 – <b>Resultado da identificação da dinâmica de rolagem</b> .....	39
Figura 12 - <b>Resultado da validação da dinâmica de rolagem</b> .....	40
Figura 13 – <b>Resultado da identificação da guinada</b> .....	41
Figura 14 - <b>Resultado da identificação da dinâmica de altitude</b> .....	42

## LISTA DE TABELAS

Tabela 1 - Aplicações de drones em várias áreas .....	15
Tabela 2 - Comparativo entre AR.Drone 1.0 e AR.Drone 2.0 .....	19
Tabela 3 - Números das portas de comunicação de AR.Drone .....	21
Tabela 4 - Principais comandos AT.....	24
Tabela 5 - Estrutura da mensagem enviado pelo canal NAVDATA.....	25
Tabela 6 -Valores de entrada .....	35
Tabela 7 - Valores de entrada para validação do modelo.....	35

## LISTA DE ABREVIATURAS

VANT	-	VEÍCULO AEREO NÃO TRIPULADO
VTOL	-	VERTICAL TAKE-OFF AND LANDING
DECEA	-	DEPARTAMENTO DE CONTROLE DO ESPAÇO AÉREO
RTA	-	ROADS AND TRANSPORT AUTHORITY
ARX	-	AUTOREGRESSIVE WITH EXOGENOUS INPUT
MATLAB	-	MATRIX LABORATORY
API	-	APPLICATION PROGRAMMING INTERFACE
EUA	-	ESTADOS UNIDOS DA AMÉRICA
ESSID	-	EXTENDED SERVICE SET IDENTIFIER

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO.....</b>	<b>14</b>
1.1	DEFINIÇÃO DO PROBLEMA.....	16
1.2	JUSTIFICATIVA.....	17
1.3	OBJETIVO GERAIS E ESPECÍFICOS.....	17
1.4	ESTRUTURA DO TRABALHO.....	17
<b>2</b>	<b>REVISÃO BIBLIOGRAFICA.....</b>	<b>18</b>
<b>3</b>	<b>FUNDAMENTAÇÃO TEORICA.....</b>	<b>19</b>
3.1	HARDWARE.....	19
3.2	SOFTWARE.....	20
3.3	COMUNICAÇÃO.....	20
3.3.1	COMANDO AT.....	23
3.3.2	COMANDO DE NAVEGAÇÃO (NAVDATA).....	24
3.3.3	CONECTANDO AO DRONE.....	25
3.4	MOVIMENTOS.....	27
3.4.1	ELEVAÇÃO.....	28
3.4.2	ROLAGEM.....	28
3.4.3	ARFAGEM.....	29
3.4.4	GUINADA.....	29
3.5	IDENTIFICAÇÃO.....	30
3.5.1	MODELO ARX.....	31
<b>4</b>	<b>METODOLOGIA.....</b>	<b>33</b>
4.1	MODELAGEM.....	33
4.2	IDENTIFICAÇÃO DE SISTEMAS.....	34
4.3	MATÉRIAS E METÓDOS.....	35
4.4	ENSAIO.....	35

<b>5</b>	<b>RESULTADOS E CONTRIBUIÇÕES.....</b>	<b>37</b>
5.1	CONCLUSÃO .....	42
<b>6</b>	<b>REFERÊNCIAS .....</b>	<b>44</b>

# 1 INTRODUÇÃO

Aeronaves são sem dúvida um meio de transporte essencial para a sociedade, o fato de conseguirem deslocar-se em linha reta de um ponto de origem ao destino possibilitou reduções significativas de tempo uma vez que os meios terrestres enfrentam vários obstáculos como, relevo no terreno, barreiras naturais entre outros. Além dessas características especiais, aliado a inserção de sistema de controle embarcados as aeronaves possibilitaram a execução das mais variadas tarefas, desde a pilotos automáticos, tarefas de monitoramento a levantamento topográficos entre outros.

Para alcançar certo grau de independência antes é precisa conhecer o comportamento desses veículos considerando-o como um sistema conhecendo a medida como evoluem no tempo e comportam nos mais variados fatores ambientais.

Com modelos dinâmicos de aeronaves é possível projetar simuladores de voo e controladores para pilotagem automática ou assistida, analisar sua aerodinâmica, projetar missões, etc (Sharma, 2009). As grandes empresas americanas vem mostrando grandes interesses em aeronaves independentes do tipos quadricoptero, em destaque a *Uber*, *Amazon* e *Google*, essas gigantes do setor de tecnologia tem noticiado grandes avanços na aplicação de drones para resolver tarefas do cotidiano, A *Uber*, (BBC, 2010) planeja testar carros voadores sobre demanda até 2020, *Amazon* e *Google*, estão começando a testar um sistema de entrega com drones sobre algumas cidades americanas e um desejo de desenvolver seus próprios sistema de controle de trafego aéreo de baixa e alta velocidade (Gordon, 2018) a Tabela 1 mostra outras tarefas que os quadrimotores podem ser aplicados, desde a áreas publicitárias até a indústria.

Logo, visto a evidência em torno do tem esse trabalho dedica-se a propor um modelo matemático a um aeronaves do tipo quadrimotor para descrever a sua dinâmica de voo.

Tabela 1 - Aplicações de drones em várias áreas

Fotografia Aérea e filmagem	Fotos publicitárias
	Fotos de imóveis
	Cobertura aérea de eventos ou notícias
	Fotos arqueológicas
Agricultura	Mapeamento infravermelho
	Mapeamento de áreas para aplicação de adubo
	Monitoramento de desmatamento
	Monitoramento de poluentes e poluição
	Exploração geológica
	Meteorologia
	Avaliação ambiental e de dano ambiental
Industria	Inspeção de locais de difícil acesso
	Inspeção de tubulações e dutos
	Inspeção de linhas de transmissão
	Acompanhamento de construções
	Mapeamento de níveis de terreno

Fonte: M. Johnnatan Adaptado

## 1.1 DEFINIÇÃO DO PROBLEMA

No início do trabalho surgiu as perguntas tais como, qual o drone deve ser utilizado? pois hoje no mercado há inúmeros modelos de drones a escolha de qual aderir muitas vezes se torna difícil. Quadricopteros comerciais de companhias consagradas como Ascending Technologies <sup>1</sup> ou Microdrone <sup>2</sup> são frequentemente utilizadas devido a robustez, embora sejam caros e seus *firmware* não são customizáveis, outra alternativa é a construção de um veículo de acordo com a necessidade, no entanto, essa alternativa é altamente dispendiosa de tempo, custo e exige conhecimento sobre a montagem. Para contornar o problema de construção do hardware e conseqüentemente diminuir o tempo de desenvolvimento, algumas pesquisas utilizam placas de controladores de voos com código aberto como por exemplo o Arducopter <sup>3</sup>, trabalho como dos autores (Lim, Park, Lee, & Kim, 2012) fornece uma visão geral das características de construção de quadrimotores utilizando esse controlador. Entretanto, mesmo reutilizando todo o hardware, o processo de construção ainda se mantém duradouro, peças precisam ser importadas pois no mercado nacional tem um preço elevado.

Após uma revisão bibliográfica foi notado a utilização de um quadrimotor francês em alguns trabalhos acadêmicos, com destaque ao texto do autor (Krajnik T. V., 2011) onde é sugerido um modelo de fácil aquisição com enorme potencial e facilidade de uso. Logo chegou-se à conclusão de utilizar o quadricoptero Ar.drone 1.0, da fabricante Francesa Parrot, suas principais vantagens são: o baixo custo, fácil disponibilidade de peças de reposição, câmera frontal e inferior e já dispor de interface de comunicação sem fio para a leitura/escrita das principais variáveis de voo. A principal desvantagem é a baixa autonomia da bateria, durando em média cerca de dez minutos o que não invalida a sua utilização devido ao curto período de tempo das manobras que serão executadas (cerca de 30s) durante os testes.

Após a escolha do quadrimotor o presente trabalho se dedica a estudar uma alternativa de baixo custo para modelagem dinâmica de quadrimotores, trabalhos como (Sousa, 2015) utiliza o Ar.Drone acoplado ao Matlab <sup>4</sup> para aquisição dos dados do veículo e análise de dados, mas, como desvantagem a comunicação fica presa a plataforma, outros trabalhos como (Santana, 2011) e (Lopes, 2013) ocultam o processo de coleta de dados dificultando a reprodução do trabalho para estudo. Logo, o trabalho atual apresenta uma alternativa de

---

<sup>1</sup> Alemanha - <http://www.asctec.de/>

<sup>2</sup> Estados Unidos - <https://www.microdrones.com/en/>

<sup>3</sup> Estados Unidos - <http://ardupilot.org/copter/>

<sup>4</sup> Matlab - Software de Alta performance voltado para o cálculo numérico.

modelagem com a aquisição de dados utilizando um código escrito na linguagem de programação C++ que pode ser facilmente adaptado e importados em outros projetos.

## 1.2 JUSTIFICATIVA

Modelos matemáticos são peças chaves para sistema de controle automático, fornecem informações para prever seu comportamento mediante algumas entradas afetadas ou não por fatores do ambiente, como vento, peso. São úteis em uma gama de aplicações, como, jogos, simuladores para treinamento e até em avaliação de desempenho, sem sequer expor o veículo a fatores reais.

O quadricoptero Ar.drone 1.0 foi escolhido para pesquisa devido a seu baixo custo de aquisição de peças de reposição, além de contar com a facilidade de leitura/escrita dos parâmetros de voo através de uma conexão *ad-hoc wifi*. O autor (Krajnik T. V., 2011) mostra como o AR.drone pode ser inserido na pesquisa e ensino. Nesse mesmo artigo são feitos exemplos de tarefas básica onde o veículo pode ser aplicado como estabilização de posição, seguimento de objetos e navegação autônoma, com destaque especial o autor fornece uma alternativa para contornar a complexidade da obtenção do modelo .

## 1.3 OBJETIVO GERAIS E ESPECÍFICOS

O objetivo desse trabalho é a obtenção de um modelo dinâmico do quadrimotor Ar.drone 1.0 e seus parâmetros. Será excitado cada entrada individual e analisado a sua resposta a cada mudança.

O tipo de identificação de sistema utilizado será a caixa preta ARX, onde pouco se conhece sobre as leis físicas que regem o sistema. Além disso, será discutido os aspectos de leitura de dados entre o quadricoptero e o computador de mesa.

## 1.4 ESTRUTURA DO TRABALHO

O presente trabalho está estruturado em quatro etapas, revisão bibliográfica, descrição do sistema, metodologia e resultados, Na revisão bibliográfica, é feito um breve resumo do estado da arte envolvendo Ar.Drone e identificação de sistemas, no capítulo seguinte, são apresentados os detalhes da dinâmica de voo dos quadrimotores, logo em seguida são apresentados os aspectos de construção do veículo, tanto em hardware, como software.

Na metodologia, é entrado a fundo nas particularidades dos modelos pesquisados. Finalizando nos resultados, onde são mostradas a experiência de todo o trabalho.

## 2 REVISÃO BIBLIOGRAFICA

Com o intuito de montar uma base teórica em torno do tema, foi realizado uma busca nas publicações mais relevantes envolvendo, identificação de sistema, modelagem matemática e validação utilizando o quadrimotor Ar.Drone. Estudos como o do autor (Santana, 2011), faz o comparativo entre uma modelagem caixa branca com uma caixa preta, é apresentado boas referências de onde encontrar os parâmetros para elaboração de um modelo. Como principal resultado é obtido um modelo ARX com respostas que tendem a seguir dinâmica real do aeromodelo.

Mais tarde o autor através do artigo (Santana, 2011) apresenta o resultado do trabalho da modelagem anterior, onde é posto dois veículos para voar em formação um líder e o outro seguidor.

Há também outros trabalhos correlatos, como o de (Lopes, 2013) onde é avaliado dois modelos, um caixa branca apresentado por outros autores e um modelo caixa preta ARX. Nesse trabalho os autores verificam a possibilidade de modelagem caixa preta tratando as entradas do sistema desacoplado (Excitação de uma entrada, mantendo as outras constantes). É feita a análise respostas dos dois modelos executando manobras desacopladas e unidas para compara com as respostas do veículo acoplado. Os resultados da modelagem apresenta uma resposta que segue a tendência do movimento do Ar.Drone exceto para a taxa de elevação vertical onde ambos os modelos tiveram dificuldade em representar a resposta real.

No trabalho (Bristeau, 2011), expõe a tecnologia de navegação e controle embutida dentro do AR.Drone, os autores discutem a precisão, a estabilidade e a robustez da estimação de estado utilizado. Além disso, os detalhes que envolvem sistema de navegação são expostos, detalhes como, sensores inerciais, sonar, técnicas de visão e contabilização e suas respectivas calibrações tornando referência importantíssima para entender os aspectos internos do veículo. Logo mais é apresentados os aspectos de hardware e software presentes no AR.Drone.

### 3 FUNDAMENTAÇÃO TEORICA

#### 3.1 HARDWARE

O Ar.Drone está disponível em duas versões 1.0 e 2.0, compartilham a mesma estrutura física e divergem em aprimoramentos de hardware e software. Ambos são compostos por quatro motores sem escovas de alta performance presos na extremidade da arte de fibras de carbono e dispõem de um invólucro para proteção mostrado na Figura 1 usados para voos internos e uma cesta de polipropileno expandido para diminuir as vibrações do motor. Tabela 2 ilustra as principais diferenças entre os dois modelos.

**Figura 1** - Ar.Drone com involucro de proteção à esquerda e sem a direita



Fonte: piskorshi (2012)

**Tabela 2** - Comparativo entre AR.Drone 1.0 e AR.Drone 2.0

Características	AR.Drone 1.0	AR.Drone 2.0
Câmera horizontal	640 x 480 (VGA)	1280 x 720 (720p HD)
Câmera vertical	176 x 144 (QCIF)	320 x 240 (QVGA)
Acelerômetro	3 eixos	3 eixos
Giroscópio	3 eixos	3 eixos
Sensor de pressão	Não	Sim
Ultrassom	Sim	Sim
Magnetômetro	Não	3 eixos
Kernel Linux	2.6.27	2.6.32
Processador	ARM 32B 468MHZ	ARM Cortex A8 32b 1GHz
RAM	128Mb	1GB
Wifi	b/g	b/g/n

Fonte: Piskorshi (2012)

Como visto na Tabela 2, há incrementos significativos de hardware do AR.Drone 1.0 para 2.0. A câmera horizontal e vertical ganhou em qualidade e resolução, também foi adicionado sensor de pressão para realizar medidas mais precisas de altitude e magnetômetro para melhor medir a orientação no espaço. O kernel Linux foi atualizado, e o destaque foi para o processador que foi aumentado a velocidade em 2,13 vezes e a memória de armazenamento aumentada em 7,81 vezes. Mas adiante serão mostrados os aspectos de software.

### 3.2 SOFTWARE

Como foi destacado no comparativo acima o sistema interno executa uma distribuição GNU/Linux baseada em BusyBox com kernel 2.6.27 para o AR.drone 1.0 e 2.6.33 para o AR.drone 2.0, a placa de controle interna possui funções de detecção de colisão ao identificar um obstáculo é bloqueado imediatamente as hélices do veículos para evitar colisões repetitivas, além disso, a plataforma fornece ao operador a execução de manobras assistidas, como decolagem que é feita automaticamente basta o operador enviar o comando de decolagem ao veículo, então o AR.drone sairá do solo e permanecerá a altura pré-estabelecida metro a espera de novos comando de voo. Há também, assistência a pousos e modo emergência.

### 3.3 COMUNICAÇÃO

Nesta seção será discutido os aspectos de comunicação entre o Ar.drone e um cliente, os detalhes de comunicação a estrutura das mensagens e bibliotecas para conexão.

O AR.Drone pode ser controlado por qualquer dispositivo que suporte conexão *Sem fio* em modo *Ad-hoc* a comunicação feita entre o drone e algum dispositivo é realizado nos seguintes passos:

1. Ao ligar o veículo é criado um sinal WIFI com um ESSID <sup>5</sup>geralmente chamado ardrone\_XXX.
2. O usuário se conecta através da rede ESSID (Extended Service Set Identifier), identificador da rede.
  - O dispositivo cliente requisita um endereço IP ao servidor DHCP.

---

<sup>5</sup> ESSID - Extended Service Set Identifier (identificador

- O servidor DHCP do AR.drone concede ao cliente um IP definido pela regra:
    - o O Endereço IP do servidor mais um, para versões inferiores a 1.1.3.
    - o O Endereço IP do servidor mais um entre 1 a 4 (para AR.Drone 2.0 e AR.Drone 1.0 depois da versão 1.1.3).
3. Então, o dispositivo cliente poderá começar enviando requisições ao endereço IP do Ar.Drone e suas portas.

Há quatro canais de comunicação no AR.Drone, que possibilita o envio e recebimento dos parâmetros de voo, cada um deste canais estão associados a uma porta UDP/TCP diferente. A Tabela 3 mostra os canais de comunicação e suas respectivas portas. O primeiro canal é o canal de comando, através dele é feito requisições para pouso e decolagem, mudança de configuração do controlador, calibração de sensores entre outros. Além do mais, o canal é utilizado para enviar ao drone parâmetro desejados de ângulos de Euler veja Figura 4.

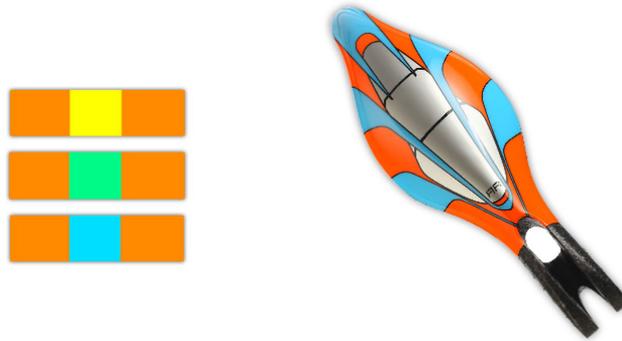
**Tabela 3 - Números das portas de comunicação de AR.Drone**

Porta	Tipo de comunicação	Nome
5554	UDP	Canal de navegação (Navdata) enviado do drone para o cliente
5555	TCP/IP	Canal de fluxo de enviado do drone para o cliente
5556	UDP	Canal de comando AT e configuração
5559	TCP/IP	Transferência de dados críticos e informação do sistema

Fonte: Piskorshi (2012)

O canal “Navdata” fornece o estado da aeronave, dados pre-processados dos sensores indicando se o drone está voando, calibrando os sensores entre outros são enviados do drone ao cliente em uma conexão UDP sob a porta 5554. O canal é frequentemente utilizado para leitura dos ângulos de Euler da aeronave, altitude e o estado da bateria, também pode ser obtido a distância entre outros Ar.drone’s através da etiqueta de cores ilustrado na Figura 2.

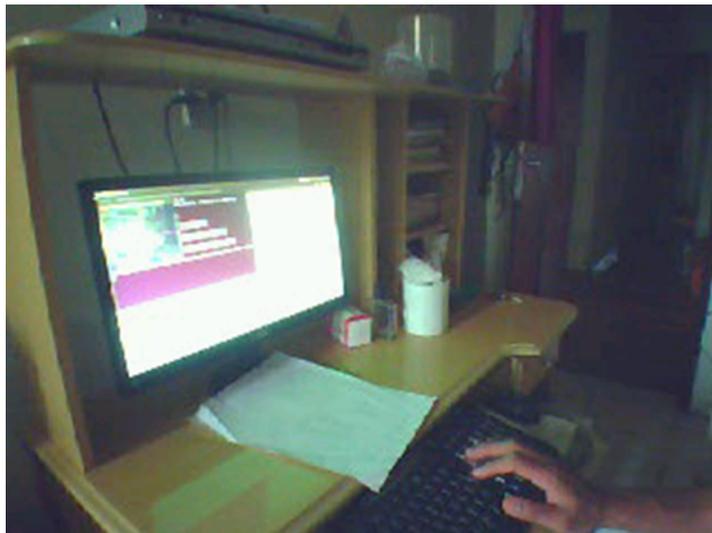
**Figura 2** - As cores da proteção forma uma etiqueta tag no AR.Drone



Fonte: Piskorshi (2012)

O terceiro canal de fluxo de imagens, fornece imagens capturadas pela câmera inferior e frontal sobre uma conexão TCP/IP enviada do drone para o cliente. As imagens das câmeras são compactadas e escaladas para reduzir o tamanho e acelerar a transferência com. Como desvantagem não é possível receber as imagens das câmeras inferior e superior simultaneamente apenas uma por vez.

**Figura 3** - Imagem capturada pela câmera frontal utilizando a biblioteca AR.Drone SDK 2.0 do AR.Drone 1.0



Fonte: Autor (2018)

O quarto canal de comunicação sob a porta 5559 requer uma conexão TCP e transfere dados críticos do drone, esse canal é usado para recuperar dados de configuração importantes sem depender de acesso concorrente entre os outros canais. Em seguida será apresentado com detalhes os aspectos das mensagens de configuração e navegação (NAVDAT) dos canais.

### 3.3.1 Comando AT

O canal de comando AT espera uma cadeia de caracteres de 8-bits ASCII finalizada com os caracteres <CR>, e uma nova linha como delimitador. Um comando básico sempre inicia com três caracteres **AT\*** seguido pelo nome do comando, o símbolo igual, o número sequencial e uma lista de valores separados por vírgula delimitado por <CR>.

Um único pacote UDP pode conter um ou mais comandos, separados por uma nova linha. Abaixo segue um exemplo de comandos:

```
AT*PCMD_MAG=21625,1,0,0,0,0,0,0<CR>AT*REF=21626,290717696<CR>
```

Como explicado os comandos PCMD\_MAG e REF ilustrado acima são exemplos de comandos AT, com sequenciais 21625 e 21626.

Para evitar que o drone processe dados antigos, um número sequencial é associado a cada comando AT enviado, assim não executará nenhuma sequencial repetido. Esse número de sequência é redefinido para 1 toda vez que o cliente desconecta da porta UDP 5556(normalmente a conexão é desfeita quando não há nenhum comando durante mais de 2 minutos).

**Tabela 4 - Principais comandos AT**

Comando AT	Argumento	Descrição
AT*REF	Input	Decolagem, pouso modo emergência
AT*PCMD	Flag, rolagem, arfagem, elevação, guinada, velocidade vertical, velocidade angular da guinada	Movimenta o drone
AT*PCMD_MAG	Flag, rolagem, arfagem, elevação, guinada, velocidade vertical, velocidade angular da guinada	Movimenta o drone
AT*FTRIM		Altera a referência de plano horizontal a altura atual do drone
AT*CONFIG	Chave, valor	Configura os parâmetros do AR.Drone 2.0
AT*CALIB	Número do dispositivo	Pedi para o drone calibrar o acelerômetro

Fonte: Piskorshi (2012)

### 3.3.2 Comando de navegação (navdata)

O drone fornece ao cliente dois principais fluxo de dados, dados de navegação e de vídeo. O dados de navegação são enviados periodicamente ao cliente (< 5ms) informações do drone(ângulo, altitude, câmera, velocidade, detecção de tag entre outros), o detalhes do segundo canal será ocultado devido à fuga da proposta do trabalho para mais informações veja (piskorshi, Brulez, & Eline, 2012). Está seção mostra como receber os pacotes navdata e decodificá-los.

O canal navdata as informações são enviadas pela porta UDP 5554 e transmitidas no formato binário. Consiste em vários blocos de seções de dados chamados opções. Cada opção contém um cabeçalho (4 bytes) identificando o tipo de informação contida na mensagem, um inteiro de 16 bits que armazena o tamanho do bloco e várias informações armazenadas com inteiros de 32 bits, números de ponto flutuante de precisão única de 32 bits

ou matrizes. A Tabela 5 mostra a estrutura do pacote para mais detalhes veja (piskorshi, Brulez, & Eline, 2012) .

**Tabela 5** - Estrutura da mensagem enviado pelo canal NAVDATA

Header 0x55667788	Drone state	Sequence number	Vision flag	Option 1			...	Checksum block		
				id	size	data		cks id	size	cks data
32-bit int.	32-bit int.	32-bit int.	32-bit int.	16-bit int.	16-bit int.	...	...	16-bit int.	16-bit int.	32-bit int.

Fonte: Piskorshi (2012)

### 3.3.3 CONECTANDO AO DRONE

Nas seções anteriores foram apresentados os aspectos de comunicação do drone, olhando por alto é visto a complexidade em criar uma conexão com o aparelho devido a necessidade de lidar com programação concorrente e sockets trabalhos com o, visto que, seria necessário controlar o envio dos comandos AT, os comandos navdata e o recebimentos de vídeo. Uma em cada threads. Por sorte o fabricante já disponibiliza a biblioteca chamada AR.Drone 2.0, essa biblioteca possui um conjunto de ferramentas que facilmente controla o drone, em particular ela fornece, uma thread de gerenciamento de comandos AT, que coleta comandos enviados por todas as outras threads e enviá-os de maneira ordenada com números sequenciais corretos. Uma thread de gerenciamento de navdata que recebe automaticamente o fluxo navdata, decodifica e fornece ao aplicativo cliente, dados de navegação prontos para uso por meio de chamadas de retorno, thread de gerenciamento de vídeo, que recebe automaticamente o gerenciamento de vídeo e gravação para o AR.Drone 1.0 e para a versão 2.0 dispõem de uma thread exclusiva para gravação das imagens em HD visto na Tabela 2 a inclusão de câmera não versão 2.0.

Para conexão foi desenvolvida uma aplicação em C++ utilizando a biblioteca AR.Drone 2.0 divulgado pelo trabalho (Krajnik T. V., 2011). Trata-se de uma API em C/C++ que implementa os protocolos de comunicação do drone. Assim torna-se fácil a implementação de aplicações e controle de comunicação do drone.

Para a aplicação foi apenas alterada poucas linhas de código do projeto original, uma função chamada input foi criada. e outra para salvar o estado do aeromodelo em um arquivo de texto.

A função input é responsável por executar a manobra do veículo no instante de tempo desejado.

### Algoritmo 1 - Código função input

```

void input(int angle,long start_time)
{
    if((get_time() - start_time) > 35000 )
    {
        break_input = true;
        return;
    }
    if((get_time()- start_time) >= 0 && (get_time()- start_time) <= 30000 )
        *inputs[angle] = 16500.0;
    if((get_time()- start_time) >= 30000 && (get_time()- start_time) <= 90000 )
        *input[angle] = -16500.0;
    if((get_time()- start_time) >= 90000 && (get_time()- start_time) <= 13000 )
        *input[angle] = 16500.0;
    if((get_time()- start_time) >= 13000 && (get_time()- start_time) <= 15500 )
        *input[angle] = 8250.0;
    if((get_time()- start_time) >= 15500 && (get_time()- start_time) <= 200000 )
        *input[angle] = 16500.0;
    if((get_time()- start_time) >= 20000 && (get_time()- start_time) <= 245000 )
        *input[angle] = 8250.0;
    if((get_time()- start_time) >= 245000 && (get_time()- start_time) <= 30000 )
        *input[angle] = 16500.0;
    if((get_time()- start_time) >= 30000 && (get_time()- start_time) <= 35000 )
        *input[angle] = -8250.0;
}

```

Na função main há uma estrutura de repetição sem fim que fica à espera de comandos do teclado, foi inserido o Algoritmo 2 para salvar os valores de voo retornadas pelo canal navdata.

### Algoritmo 2 - Escreve arquivo com os dados de voo

```

while (stop == false){
    fprintf(stdout,"Angles %.2f|%.2f|%.2f|%.2f|%.2f|%.2f|%.2f|%.2f\n",
        helidata.phi,
        helidata.psi,
        helidata.theta,
        pitch,
        roll,
        yaw,
        height,
        helidata.battery
    );
}

```

Portanto, com as alterações completas e o código recompilado basta seguir os passos abaixo

- Ligar o drone
- Conectar ao sinal sem fio

- Executar aplicativo (Imediatamente as luzes passam de vermelhas para verdes)
- Pressionar “Q” para decolar (O arquivo de log é criado no diretório atual)
- Esperar o drone levantar vôo
- Pressionar a tecla “M” (O drone executará a manobra)
  - Pressionar a tecla “A” para pousar.

Após a execução da manobra o aplicativo escreve o arquivo de log seguindo a estrutura abaixo:

```
-310.00|-3541.00|1230.00|211.00|0.00|-0.00|0.00|
```

Onde:

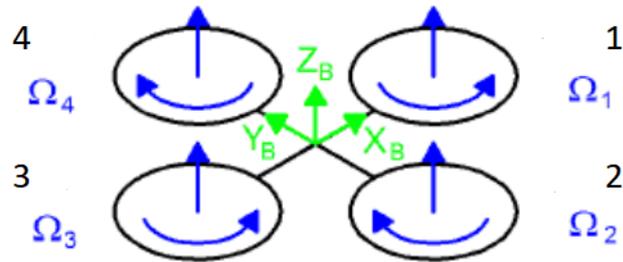
```
%.2f|%.2f|%.2f|%.2f|%.2f|%.2f|%.2f|
```

- Velocidade angular arfagem (rad/s),
- Velocidade angula guinada(rad/s),
- Arfagem (m/s),
- Rolagem (m/s),
- Guinada (m/s),
- Altura (m/s),
- Tensão da bateria;

### 3.4 MOVIMENTOS

Os quadrimotores possuem um grau de liberdade a mais que os aviões, são rolagem, arfagem, guinada e elevação, esses movimentos são feitos através da manipulação da velocidade angular de cada hélice. Na Figura 4, é mostrado o modelo simplificado do drone. Em cada extremidade 1,2,3 e 4 é preso um motor com velocidade angular  $\Omega_n$ . O par de hélices da frente 2,3,1 e 4 (giram em sentidos diferentes para compensar o efeito do torque produzido pelos motores. Assim a força produzida por cada um dos motores supera a gravidade e levantar vôo. Mas adiantes é apresentado como o aeromodelo pode realizar os quatros movimentos descritos acima.

**Figura 4** - Modelo simplificado do drone

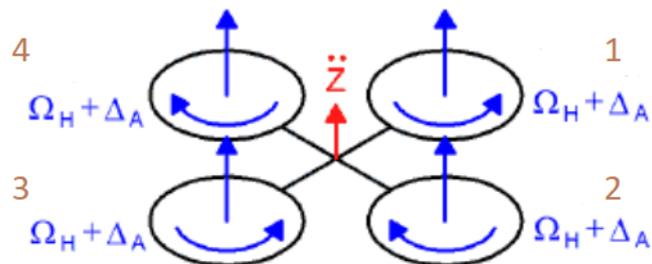


Fonte: Adaptado de Piskorshi (2012)

### 3.4.1 ELEVAÇÃO

Quando se deseja aumentar ou diminuir a altitude do quadricoptero o controlador interno transmite aos motores um incremento ou decremento de velocidade  $\Delta_A$  a cada um dos motores. Quando ocorre essa mudança, o veículo eleva ou declina sua altitude em cima do eixo  $Z_B$  conforme a Figura 5, pois haverá uma soma da força de propulsão para cima.

**Figura 5** – altitude

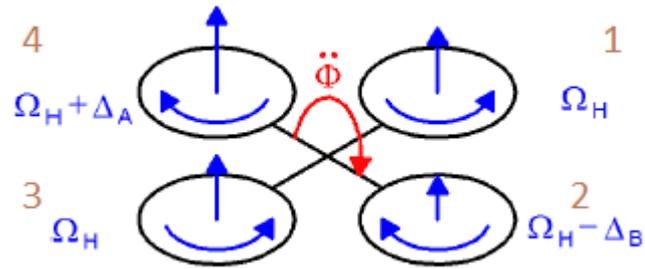


Fonte: Adaptado de Piskorshi (2012)

### 3.4.2 ROLAGEM

Quando se deseja rotacionar o drone sobre o eixo  $X_B$  o controlador interno altera a velocidade das hélices das extremidades 1 e 3 por um fator  $\Delta_A$  e  $-\Delta_B$ . Com essa configuração, o drone rotaciona em torno do eixo  $X_B$  como detalha a Figura 6.

Figura 6 – rolagem

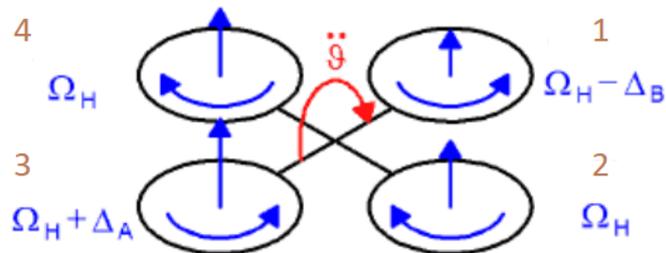


Fonte: Adaptado de Piskorshi (2012)

### 3.4.3 ARFAGEM

Parecido com o movimento rolagem, a dinâmica de arfagem é um movimento obtido ao variar as velocidades das hélices das extremidades 1 e 3, localizadas sobre o eixo  $X_B$ , ao adicionar a velocidade das hélices por um fator  $\Delta_A$  e  $-\Delta_B$  a plataforma sofre uma rotação sobre o eixo  $Y_B$  mantendo os outros ângulos constantes.

Figura 7 – arfagem

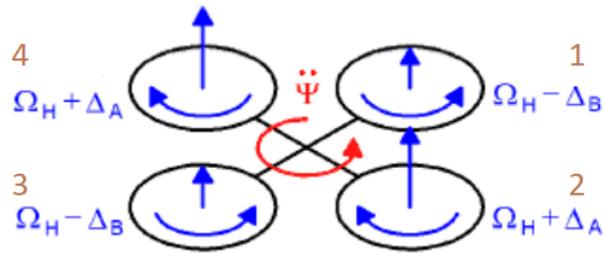


Fonte: Adaptado de Piskorshi (2012)

### 3.4.4 GUINADA

Esta dinâmica rotaciona o quadrimotor em relação ao eixo  $Z_B$ , sendo possível manter o quadrimotor pairado e girando em cima do eixo  $Z_B$ . A dinâmica é feita aumentando e diminuindo gradativamente os pares de hélices 1, 2, 3 e 4.

**Figura 8 – guinada**



Fonte: Adaptado de Piskorshi (2012)

Vale destacar que todas essas dinâmicas são todas implementadas pelo controlador interno do AR.Drone o operador apenas enviara qual a dinâmica a executar.

### 3.5 IDENTIFICAÇÃO

Esta seção tem como objetivo apresentar uma introdução à identificação de sistemas dinâmicos, os passos para obtenção e as principais técnicas.

Segundo (Aguirre, 2007), Modelagem matemática é a área de conhecimento que estuda maneira de construir e implementar modelos (matemáticos) de sistemas reais. Há várias formas de classificar as técnicas de modelagem, Modelagem caixa branca, modelagem caixa preta e modelagem caixa cinza.

A modelagem caixa branca é obtida a partir das leis físicas, logo é necessário conhecer bem o sistema em estudo, a modelagem caixa cinza onde é pouco ou nenhum conhecimento prévio do sistema é necessário. Por fim, a modelagem caixa preta é o tipo de técnica onde não há nenhum conhecimento prévio do sistema. Para outros autores a classificação passa a se chamar identificação completa ou parcial, análogo a modelagem caixa preta e cinza.

Em geral há três passos para identificação de sistemas:

- Especificar e parametrizar modelos matemáticos que representem o sistema;
- Identificar os parâmetros do modelo escolhido que melhor se relacionam ao sistema;
- Validar o modelo para verificar se o modelo escolhido corresponde as expectativas finais;

A obtenção dos parâmetros pode ser calculada pelo método de mínimos quadrados para sistemas lineares. Para ajuste de modelos não lineares podem ser utilizados métodos baseados no gradiente ou algoritmos genéticos.

(Ogata, 1993), definiu o modelo matemático de sistemas dinâmicos como sendo o conjunto de equações que representam a dinâmica de um sistema precisamente bem. O modelo matemático é o primeiro passo para a representação de modelos dinâmicos, sendo considerada a parte mais importa para identificação.

Existem diversos modelos que podem representar um sistema de maneira diferentes, dependendo da perspectiva a ser considerada. Alguns dos modelos utilizados, para modelar sistemas lineares são os modelos auto regressivo, como o modelo auto regressivo com entradas exógenas (ARX) e modelo auto regressivo com média móvel e entradas exógenas (ARMAX), modelos de varáveis de estados e funções de transferência

### 3.5.1 MODELO ARX

Dado o sistema com perturbação, como sendo  $U(t)$  a entrada,  $e(t)$  a perturbação e  $y(t)$  a saída tem-se:

A equação que representa o sistema é dada por

$$y(t) = G(q)u(t) + H(q)e(t)$$

sendo,

$$G(q) = \sum_{k=1}^{\infty} g(k)q^{-k} \quad e \quad H(q) = 1 + \sum_{k=1}^{\infty} g(k)q^{-k}$$

Este é o modelo de um sistema invariante no tempo, onde  $G$  e  $H$  são as funções transferência do sistema,  $e(t)$  a perturbação e  $q^{-k}$  é um operador de atraso. As funções de transferência possuem parâmetros que devem ser identificados. O vetor de parâmetros é definido por  $\theta$  e desta forma pode-se escrever a equação acima como sendo:

$$y(t) = G(q, \theta)u(t) + H(q, \theta)e(t)$$

Definindo-se o preditor um passo a frente para  $y$  a equação acima fica:

$$\hat{y}(t|\theta) = H^{-1}(q, \theta)G(q, \theta)u(t) + [1 - H^{-1}(q, \theta)]y(t)$$

Onde  $\hat{y}(t|\theta)$  representa o valor predito da variável  $y$  no instante  $t$ . A predição utiliza o vetor de parâmetros  $\theta$ .

O modo mais simples de representar a relação entre a entrada e a saída para um sistema é dada pela equação linear em diferenças:

$$y(t) + a_1 y(t-1) + \dots + a_{n_a} y(t-n_a) = b_1 u(t-1) + \dots + b_{n_b} u(t-n_b) + e(t)$$

O modelo representado pela equação acima é chamado de equação para o modelo com erro. Os parâmetros para este caso são:

$$\theta = [a_1 a_2 \dots a_{n_a} b_1 \dots b_{n_b}]^T$$

Definindo

$$A(q) = 1 + a_1 q^{-1} + \dots + a_{n_a} q^{-n_a}$$

E

$$B(q) = b_1 q^{-1} + \dots + b_{n_b} q^{-n_b}$$

Tem-se equação corresponde a equação com:

$$G(q, \theta) = \frac{B(q)}{A(q)}, H(q, \theta) = \frac{1}{A(q)}$$

O modelo acima é chamado de modelo ARX, onde  $A(Q)/y(t)$  é parte regressiva e  $B(q)u(t)$  é a entrada exógena.

## 4 METODOLOGIA

### 4.1 MODELAGEM

Já é conhecida na literatura o modelo dinâmico de um AR.drone, apresentado por (Kim, 2010)

$$\begin{cases} m\ddot{x} = (\cos \psi \sin \phi + \cos \psi \cos \phi \sin \theta)u_1 \\ m\ddot{y} = (-\cos \psi \sin \phi + \sin \psi \cos \phi \sin \theta)u_1 \\ m\ddot{z} = (\cos \phi \cos \theta)u_1 - mg \\ I_{xx}\ddot{\phi} = u_2 - (I_{zz} - I_{yy})\theta\psi \\ I_{yy}\ddot{\theta} = u_3 - (I_{xx} - I_{zz})\phi\psi \\ I_{zz}\ddot{\psi} = u_4 \end{cases}$$

O modelo é expresso no sistema de coordenadas do corpo, sendo o centro de massa o ponto de interesse, onde  $m$  representa a massa,  $g$  a aceleração da gravidade e  $I_{zz}$ ,  $I_{yy}$  e  $I_{xx}$  são os momentos de inércia. Os parâmetro  $u_1, \dots, u_4$ , são parâmetros que correspondem as forças de torque gerado pelas hélices. Segundo (Bristeau, 2011), o AR.drone considera em seu firmware um modelo similar ao acima. Entretanto, para obter alguns parâmetros, a interface de comunicação do veículo não permite acessar as tensões elétricas aplicadas nos motores, logo, impossibilita a utilização do modelo.

O firmware do Ar.drone disponibiliza o conjunto de variáveis de voo, onde,  $\phi$ ,  $\theta$  e  $\psi$  que representam os ângulos de rolagem, arfagem e guinada do veículo expresso no sistema de coordenadas global,  $z$  a altitude do veículo, o firmware já trata a influência das inclinações  $\phi$  e  $\theta$  sobre  $z$ .  $v_x$  e  $v_y$  representam a velocidade linear do veículo em relação aos eixos  $X_b$  e  $Y_b$ . Além disso, os sinais de controle podem ser enviados ao drone para realizar alguma ação desejada, os parâmetros desse sinais é uma 4-upla,  $[u_z, u_\psi, u_\phi, u_\theta] \in [-1,0, +1,0]$  sendo,  $u_z$  um comando de velocidade linear sobre o eixo  $z$ ,  $u_\psi$  envia um comando de inclinação sobre o eixo  $Z_b$ ,  $u_\phi$ , envia um comando de inclinação sobre o eixo  $X_b$  e  $u_\theta$  envia uma comando de inclinação do veículo em relação ao eixo  $Y_b$ .

Como o controlador interno é capaz de configurar, manter os ângulos desejados e a velocidade em torno de cada eixo, não é necessário lidar com a complexidade do modelo acima. Alguns trabalhos como (Krajnik T. V., 2011), (Engel, Sturm, & Cremers, 2012)e (Santana, 2011), levam em consideração a resposta do AR.Drone aos sinais de controle submetidos aos efeitos do controlador interno chegando ao seguinte modelo:

$$\begin{cases} \dot{v}_x = K_1 u_{v_x} - K_2 u_{v_x} \\ \dot{v}_y = K_3 u_{v_y} - K_4 u_{v_y} \\ \ddot{z} = K_5 u_{u_z} - K_6 \dot{z} \\ \ddot{\psi} = K_7 u_{\psi} - K_8 \dot{\psi} \end{cases}$$

Onde  $\dot{v}_x$  representa a aceleração linear sobre o  $X_B$ ,  $\dot{v}_y$ , representa a aceleração linear sobre o eixo  $Y_B$ ,  $\ddot{z}$ , é a 'aceleração linear sobre o eixo  $z$  no sistema de coordenadas global e  $\ddot{\psi}$ , representa a aceleração angular no  $z$  no sistema de coordenadas global.  $K_1, \dots, K_8$  são os parâmetros a serem identificados, visto que, variam ligeiramente para diferentes drones.

## 4.2 IDENTIFICAÇÃO DE SISTEMAS

Há várias formas de identificar um sistema e obter seu análogo matemático, em consequência, não existem apenas um modelo que possa descrever o sistema, há uma família com naturezas e performances diferentes que podem representá-los.

Algumas considerações são levantadas durante a identificação, que podem, simplificar o processo de obtenção do modelo tal como, linearidade, invariância no tempo, estáticos ou dinâmicos, discretos ou contínuos entre outros.

Um desses modelos de identificação é o auto regressivo com entrada externa, nesse trabalho será utilizado este modelo, estimando seus parâmetros através do método de mínimos quadrados já embutido no toolbox de identificação do MATLAB.

O interesse é encontrar os parâmetros para a equação do modelo auto regressivo com entradas exógenas, onde a sua estrutura é:

$$y(k) + a_1 y(k-1) + \dots + a_{n_a} y(k-n_a) = b_1 u(k-n_k) + \dots + b_{n_b} u(k-n_b-n_k+1) + e(k)$$

$y(k)$  é a saída no tempo,  $n_a$ , números de polos,  $n_b$ , número de zeros  $n_k$  é o número de amostras de entrada que ocorreram antes que a entrada afeta a saída. Ou tempo morto.

Uma forma diferente e compacta de escrever a equação acima é:

$$A(q)y(t) = B(q)u(t-n_k) + e(t)$$

Onde  $q$  é o operador de atraso e :

$$A(q) = 1 + a_1q^{-1} + \dots + n_{n_a}q^{-n_a}$$

$$B(q) = b_1 + b_2q^{-1} + \dots + b_{n_b}q^{-n_b+1}$$

Para avaliar a qualidade do modelo, a *toolbox* de identificação de sistema possui a função de custo FIT descrita pela seguinte equação, para avaliar a qualidade do ajuste:

$$FIT = 1 - \frac{\|y - \hat{y}\|}{\|y - \bar{y}\|} * 100$$

Onde ,  $y$  é a saída experimental medida,  $\hat{y}$  é a saída do modelo e  $\bar{y}$  é o valor esperado

### 4.3 MATÉRIAS E METÔDOS

Para a aquisição dos dados foi seguido o passo a passo descrito na seção 3.3.3.

Após a obtenção foi utilizado o software MATLAB e o *toolbox* de identificação de sistemas.

### 4.4 ENSAIO

Foi realizado o experimento em ambiente fechado executando os passo da , Onde cada entrada do AR.Drone ,  $u_z$ ,  $u_{\dot{\psi}}$ ,  $u_{\phi}$  e  $u_{\theta}$  receberam os sinais em malha aberta descritos Tabela 6 e Tabela 7. Quando uma entrada sofria uma excitação as outras eram mantidas iguais a zero, simulando um movimento desacoplado do drone.

Os dados da Tabela 6 foram utilizados na obtenção do modelo ARX, enquanto a Tabela 7, foi a validação.

Tabela 6 -Valores de entrada

Tempo[s]	0-2,5	2,5-6	6-11	11,0-15	15-20	20-25
Entrada	0,5	-0,5	0,5	-0,25	0,1	-0,1

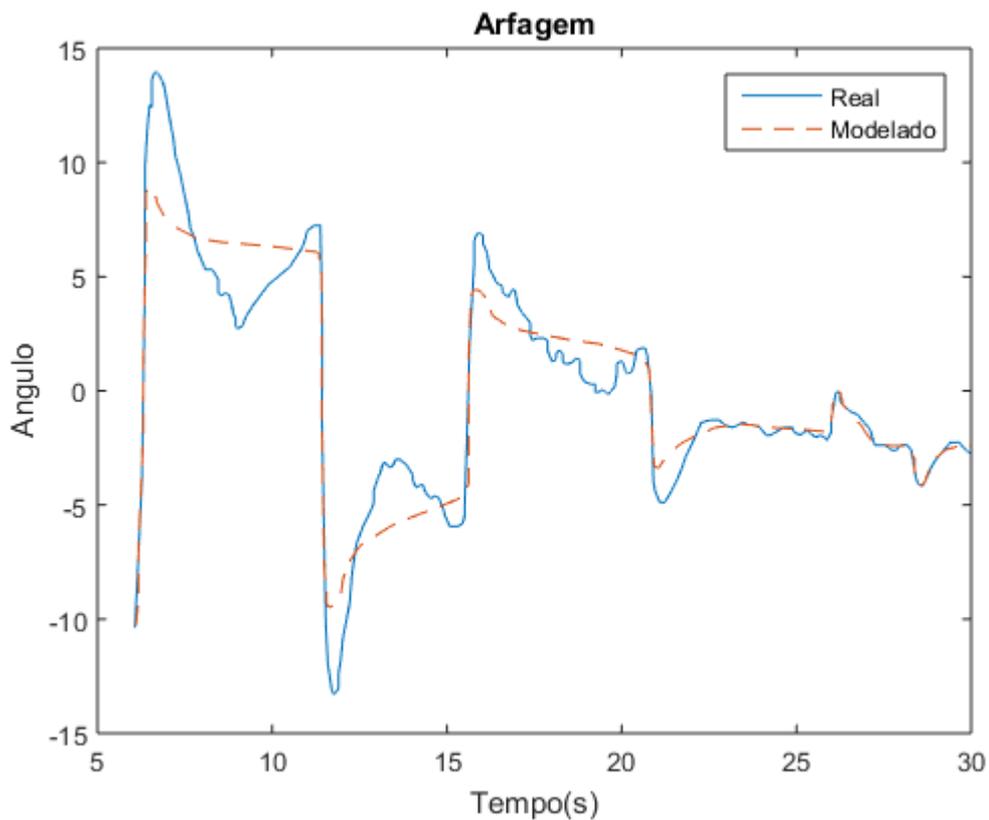
Tabela 7 - Valores de entrada para validação do modelo

Tempo(s)	0-3	3-5	5-9	9-15,5	15,5-20	20-25
Entrada	0,5	-0,25	0,5	-0,5	0,1	0,5

## 5 RESULTADOS E CONTRIBUIÇÕES

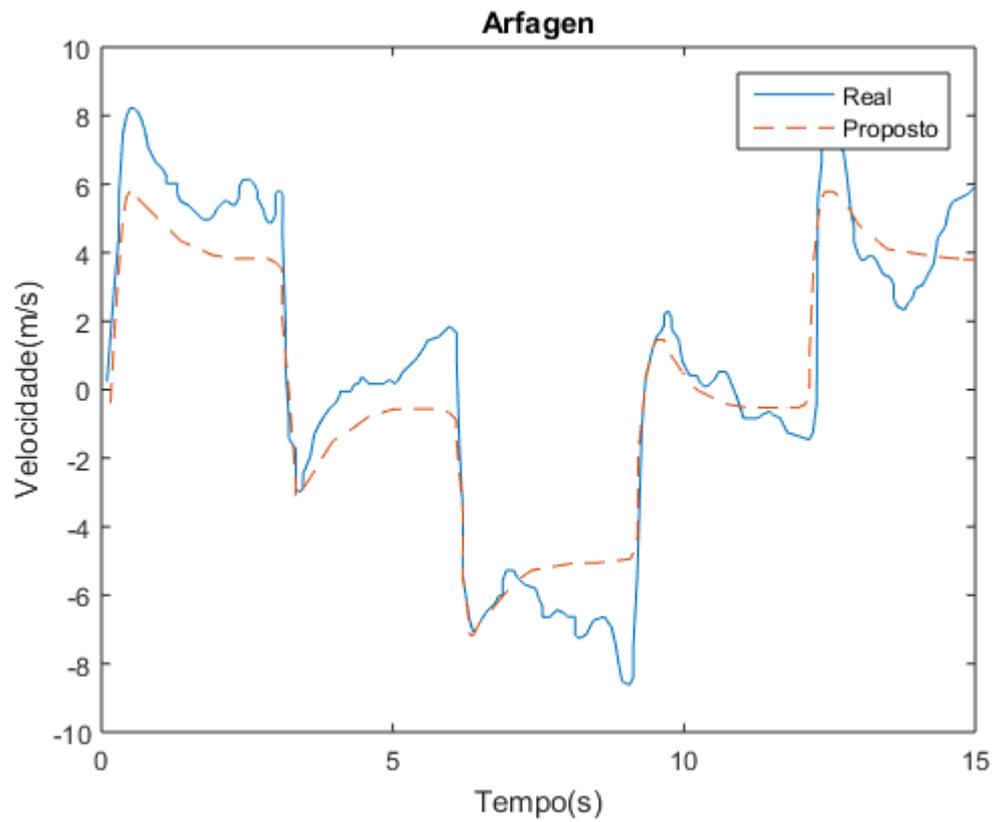
Após o processamento dos dados e a estimação dos parâmetros foi obtido os resultados, primeiramente pela dinâmica de arfagem, ao estimar os valores pelos mínimos quadrados o melhor ajuste da curva foi para os parâmetros,  $n_a = 4, n_b = 8$  e  $n_k = 1$ , onde o FIT (função de custo) foi igual a 70.85%.

**Figura 9** – Resultado da dinâmica de rolagem

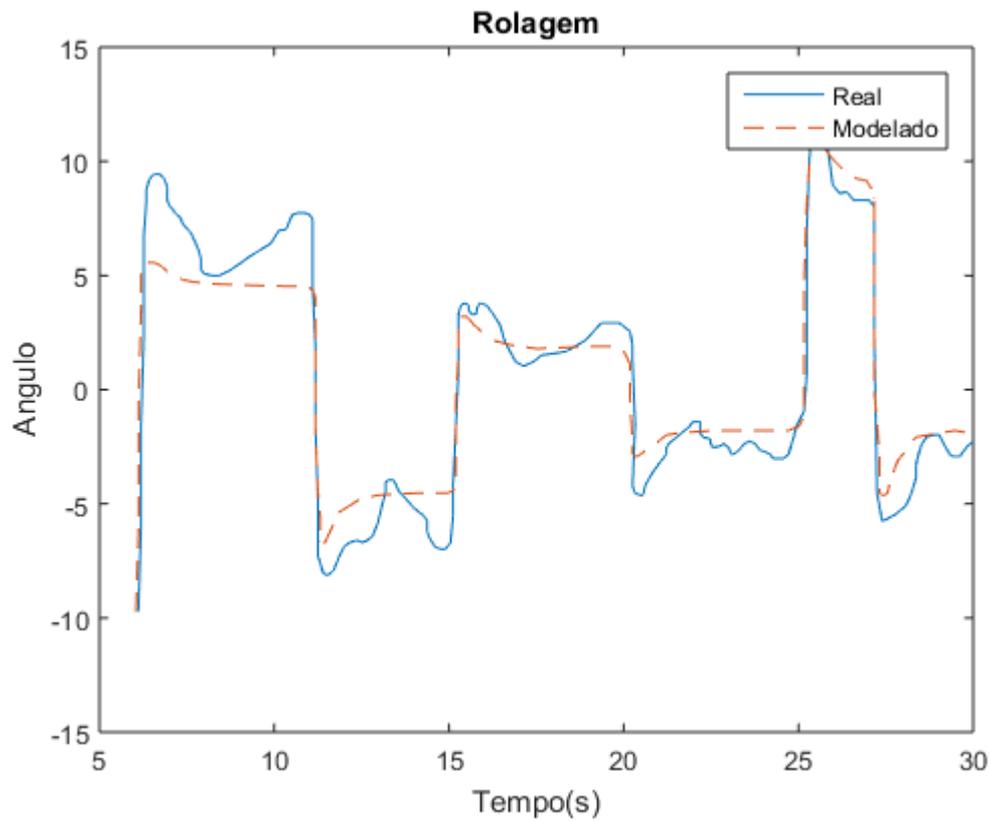


A Figura 9 apresenta a comparação entre a resposta real e modelada. É visto que o que o modelo ARX proposto acompanha a tendência da dinâmica, mesmo nas mudanças brusca de amplitude, entretanto, para amplitudes mais elevadas o modelo proposto não acompanha a amplitude real do sistema e seus vales durante a estabilização.

A validação do modelo é mostrada na Figura 10 onde foi obtido o FIT igual a 69.39%.

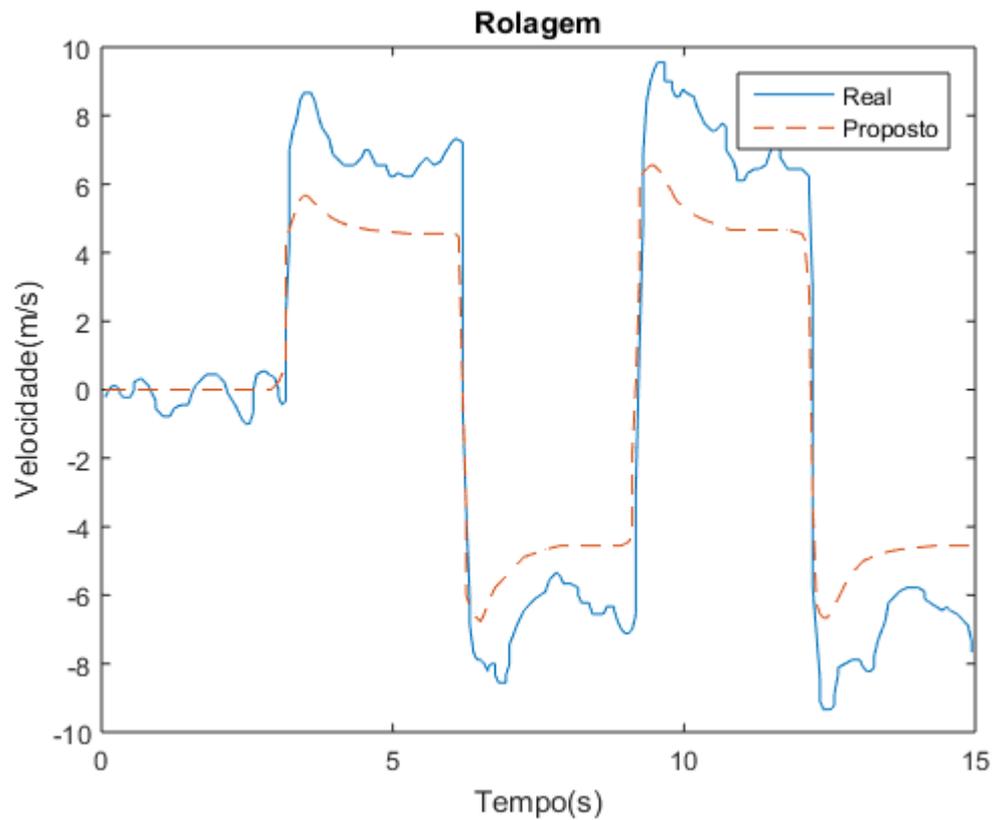
**Figura 10** - Resultado da validação arfagem

Na excitação desacoplada da dinâmica de rolagem(roll), o melhor ajuste obtido, foi com os valores dos parâmetros,  $n_a = 2, n_b = 14$  e  $n_k = 1$  com FIT igual 72.51%, superior ao da arfagem. Suas respostas são ilustradas na Figura 11.

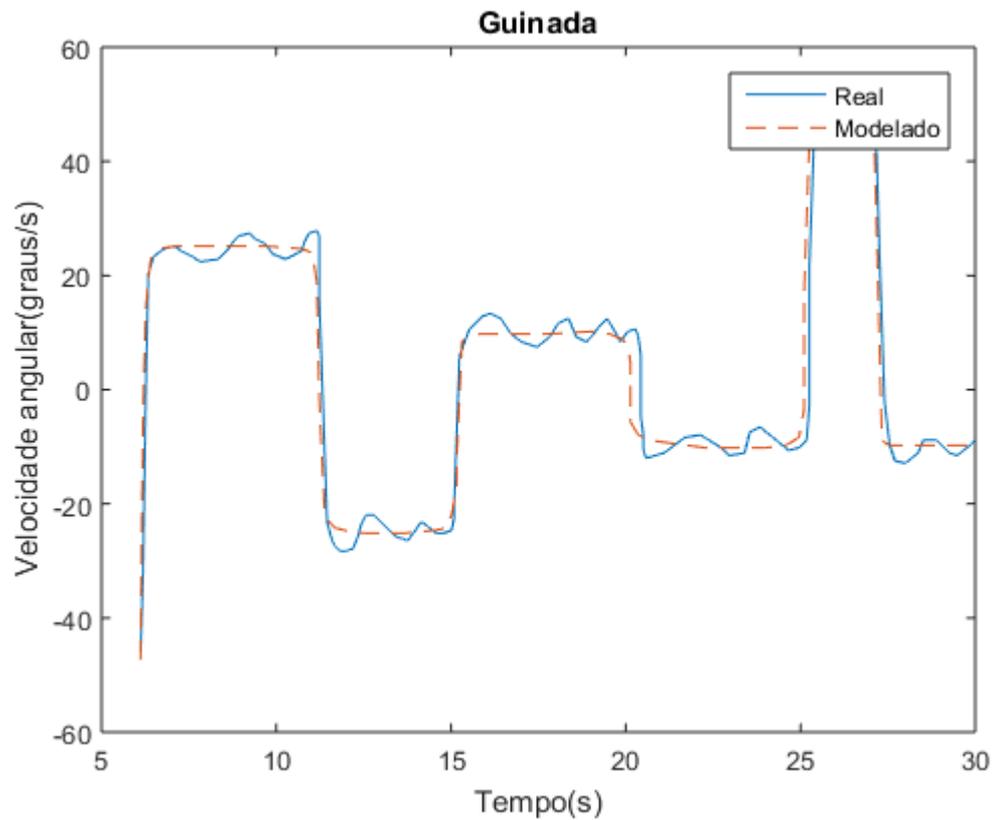
**Figura 11** – Resultado da identificação da dinâmica de rolagem

Nesta dinâmica acontece o mesmo efeito para a arfagem, o modelo proposto não acompanha a amplitude máxima do sistema e descreve melhor em amplitudes menores. O resultado da validação é ilustrado na Figura 12 o FIT obtido foi 67.07%.

**Figura 12** - Resultado da validação da dinâmica de rolagem

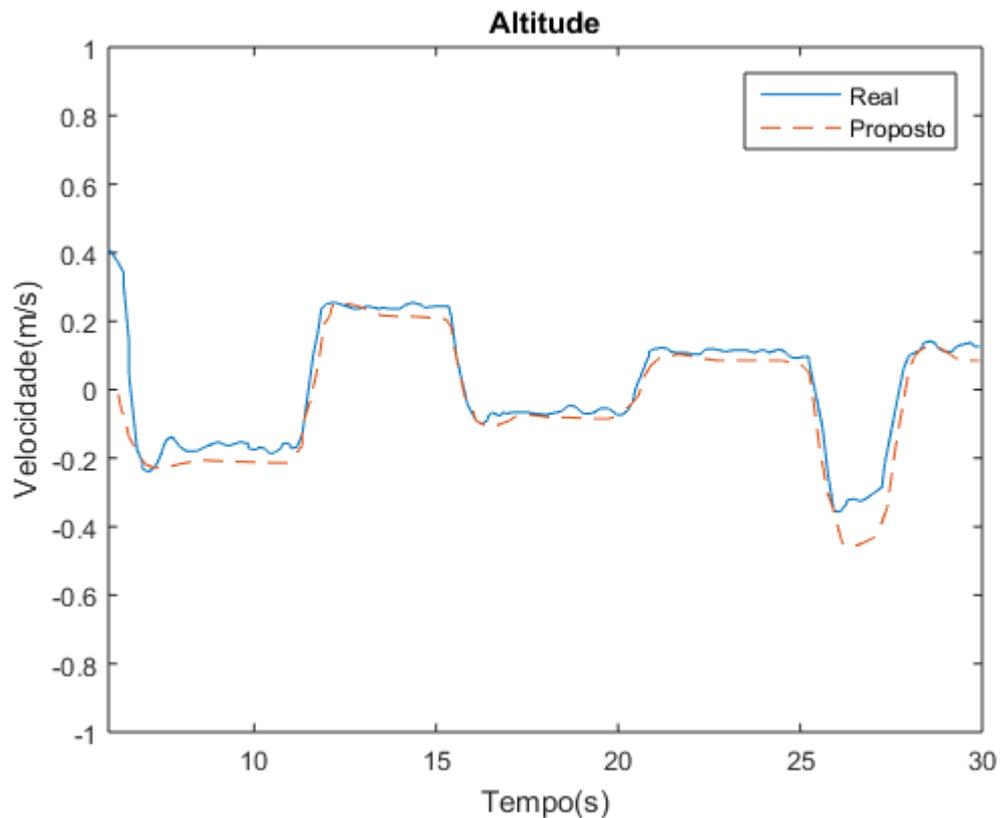


Já a dinâmica da guinada onde o sinal de entrada é uma variação da velocidade e saída velocidade angular. Neste caso o ajuste da curva foi igual a 90.62% e parâmetros  $n_a = 1$ ,  $n_b = 9$  e  $n_k = 2$ . É visto na Figura 13 que o modelo proposto acompanha as mudanças do sinais, inclusive em amplitude do sinal diferente das dinâmicas anteriores.

**Figura 13** – Resultado da identificação da guinada

Por fim, a dinâmica de saída é a altitude, como sinal de entrada é a variação da velocidade e saída a taxa de elevação vertical. A identificação resultou em um FIT igual a 72.92% e parâmetros do modelo ARX iguais a  $n_a = 9$ ,  $n_b = 1$  e  $n_k = 6$ , abaixo é mostrada a resposta e comparação entre o modelo real e proposto.

**Figura 14** - Resultado da identificação da dinâmica de altitude



## 5.1 CONCLUSÃO

A identificação do sistema através do modelo ARX demonstrou que, com parâmetros obtidos pelo método dos mínimos quadrados, conseguiu descrever bem as tendências do movimentos do AR.Drone, todavia, o modelo proposto para dinâmicas de arfagem e rolagem, tendem a não acompanhar as respostas dinâmicas mais elevadas. Analisando também a dinâmica de altitude, há fatores que contribuíram para a existência de erros, como as inclinações das dinâmicas de rolagem e arfagem no cálculo da altitude pelo sonar (sensor interno para cálculo de distância) introduzindo erros valor.

Após a comparação dos resultados com outros trabalhos, foi visto que houve resultados melhores com a utilização de uma coeficientes da função de transferência de dois polos e um zero, usando o algoritmo dos mínimos quadrados para estimar os parâmetros, mostrando a possibilidade de utilização da simplificação do modelo.

Conclui-se que estimar o estado do veículo através da entrada, ao invés de lidar com a dinâmica complexa do sistema é um boa forma de reduzir o tempo de identificação, visto que há parâmetros não disponibilizados pelo software.

## 6 Referências

- Aguirre, L. A. (2007). *Introdução à identificação de sistemas*. Belo Horizonte: UFMG.
- BBC. (26 de Abril de 2010). *Uber plans to test on-demand flying cars network*. Fonte: [bbc.com: http://www.bbc.com/news/business-39715061](http://www.bbc.com/news/business-39715061)
- Bristeau, e. a. (2011). The navigation and control technology inside the ar drone micro uav. *18th IFAC World Congress*, 1477-1484.
- DECEA. (10 de Março de 2017). *Publicações DECEA*. Fonte: DECEA: <https://publicacoes.decea.gov.br/?i=publicacao&id=4510&refresh=2B1EA259-1A4C-42A5-8BB6C4C604509BD9>
- Engel, J., Sturm, J., & Cremers, D. (2012). Camera-based navigation of a low-cost quadrocopter, *Proceedings of the 2012. IEEE/RSJ International Conference on Intelligent Robots And Systems*, pp. 2815-2821.
- Gertler, J. . (3 de Janeiro de 2002). *www.loc.gov/crsinfo/*. Fonte: Congressional Research Service: <https://fas.org/sgp/crs/natsec/R42136.pdf>
- Gordon, J. (10 de Março de 2018). *Amazon and Google to begin testing delivery drones above several U.S. cities WITHIN MONTHS using their own air traffic control system*. Fonte: [dailymail.com: http://www.dailymail.co.uk/news/article-5485375/Amazon-Google-developing-air-traffic-control.html](http://www.dailymail.co.uk/news/article-5485375/Amazon-Google-developing-air-traffic-control.html)
- Hall, K. (2016). The emergence of lethal. *Security Dialogue*, 47(3), 223-238.
- Kim, J. K. (2010). Accurate modeling and robust hovering control for a quadrotor vtol aircraft. *Journal of intelligent & Robotic Systems*, 57, pp. 1-4.
- Kindervater H, K. (2016). The emergence of lethal surveillance: Watching and killing in the history of drone technology. *Security Dialogue*, 43(3), 223-238.
- Krajník, T. (2011). *Krajník*. Fonte: Department of Cybernetics Prague: <http://labe.felk.cvut.cz/~tkrajnik/ardrone>

- Krajnik, T. V. (2011). Ar-drone as a platform for robotic research and education. *Communications in Computer and Information Science*, 161, pp. 172-186.
- Lim, H., Park, J., Lee, D., & Kim, H. J. (2012). Build your own quadrotor open-source projects on unmanned aerial vehicles. *IEEE Robot. Automat. Mag*, 19(3), 33-45.
- Lopes, L. L. (2013). Modelagem e validação de um quadrimotor Ardrone Parrot. *Modelagem e validação de um quadrimotor Ardrone Parrot*.
- Ogata, K. (1993). *Engenharia de Controle Moderno*. Rio de Janeiro: Prentice Hall do Brasil.
- piskorshi, S., Brulez, N., & Eline, P. D. (2012). *Ar.Drone Developer Guide Parrot. SDK Version 2.0*.
- Santana, L. V. (2011). Modelagem, estimação de estados e controle de formação líder-seguidor usando o quadrimotor ar.drone. *Anais do XX Congresso Brasileiro de Automática*.
- Sharma, R. S. (2009). Adaptation of modelling and identification: an aeronautical engineering. *Identification and Control*, 6, pp. 104-109.
- Sousa, J. D. (2015). *Identificação de um veículo aéreo não tripulado*. Brasília.
- Wakefield, J. (28 de Outubro de 2017). *Dubai tests drone taxi servisse*. Fonte: BBC: <http://www.bbc.com/news/business-39715061>