

# Modelagem de Alto Nível e Controle do Quadrimotor *Bebop 2* \*

Harrison Neves Marciano \* Anthony Oliveira Pinto \*  
 Vinicius Pacheco Bacheti \* Mauro Sérgio Mafra Moreira \*  
 Alexandre Santos Brandão \*\* Mário Sarcinelli Filho \*

\* Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal do Espírito Santo, ES (e-mail: hnmarciano@gmail.com, anthonymiglio@gmail.com, vinicius.bacheti@gmail.com, mauromafra@gmail.com, mario.sarcinelli@ufes.br).

\*\* Departamento de Engenharia Elétrica, Universidade Federal de Viçosa, MG (e-mail: alexandre.brandao@ufv.br)

**Abstract:** This paper deals with a simplified dynamic model for the Parrot *Bebop 2* quadrotor, whose parameters are identified through a least-mean square error procedure. Moreover, such a model is used to design high-level controllers to autonomously guide a single vehicle or a formation of two of them, in trajectory-tracking tasks. The controllers thus designed are used in experiments of autonomous navigation, whose results validate the proposed model and the designed controllers.

**Resumo:** Este artigo trata de um modelo dinâmico simplificado para o quadrimotor Parrot *Bebop 2*, cujos parâmetros são identificados através de um procedimento de mínimos quadrados. Além disso, esse modelo é usado para projetar controladores de alto nível para guiar autonomamente um único veículo ou uma formação de dois deles, em tarefas de seguimento de trajetória. Os controladores assim projetados são utilizados em experimentos de navegação autônoma, cujos resultados validam o modelo proposto e os controladores projetados.

**Keywords:** UAVs; Flying robots; Guidance navigation and control; Autonomous robotic systems; Motion control systems.

**Palavras-chaves:** VANTs; Robôs aéreos; Guiagem navegação e controle; Sistemas robóticos autônomos; Sistemas de controle de movimento.

## 1. INTRODUÇÃO

O quadrimotor *Bebop 2*, projetado e fabricado pela Parrot Drones SAS<sup>1</sup>, é um micro veículo aéreo - MVA (do inglês *Micro Aerial Vehicle - MAV*) com um piloto automático incorporado, que propicia um bom controle de atitude. Assim, depois que o usuário liga o quadrimotor ele permanece em voo pairado 1 m acima do solo, com os ângulos de arfagem, rolagem e guinada iguais a zero, aguardando comandos de movimento. Esse procedimento é incluído no piloto automático do veículo, que também monitora o nível da bateria e ativa um procedimento nativo de volta para casa, quando a bateria fica abaixo de um limite ou a conexão com o controle remoto é perdida (o veículo é controlado através de um link Wi-Fi).

Além desses recursos, comuns em MVAs semelhantes, a qualidade das imagens fornecidas pelo *Bebop 2* se destaca, inclusive por usar lente olho de peixe, de forma que ele é uma boa opção para várias aplicações, como inspeção de infra-estruturas (Jordan et al., 2018) ou plantações (Corti et al., 2019), ou outras aplicações que demandem análise visual (Hood et al., 2017; Amorim et al., 2019).

O *firmware* a bordo do *Bebop 2* entrega ao usuário as variáveis  $[z \ v_x \ v_y \ \theta \ \phi \ \psi]^T$ , que correspondem à altitude do veículo em relação ao solo abaixo dele (o solo é considerado uma superfície plana e o *firmware* já trata ângulos de rotação e inclinação não nulos), as velocidades de translação ao longo dos eixos  $x^b$  e  $y^b$ , e os ângulos de Euler  $\phi$ ,  $\theta$  e  $\psi$  (os ângulos de rolagem, arfagem e guinada, que correspondem à atitude do veículo), referidos ao sistema de coordenadas globais  $x^w y^w z^w$  (ver os sistemas de coordenadas na Figura 1).

Para fazer o drone navegar, após a decolagem pelo procedimento nativo de seu *firmware*, os comandos  $u_\theta$ , referente ao ângulo de arfagem,  $u_\phi$ , referente ao ângulo de rolagem,  $u_z$ ,

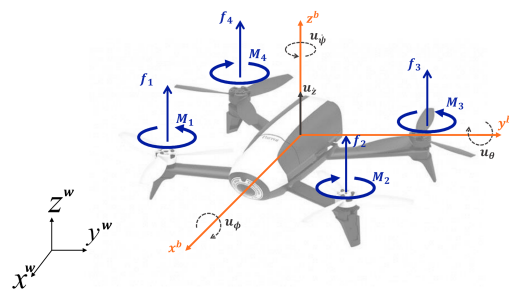


Figura 1. Sistemas de coordenadas associados ao quadrimotor *Bebop 2*.

\* Trabalho financiado pelo CNPq - Conselho Nacional de Desenvolvimento Científico e Tecnológico, e FAPES - Fundação de Amparo à Pesquisa e Inovação do Espírito Santo.

<sup>1</sup> <https://www.parrot.com/us/drones/parrot-bebop-2-power-pack-fpv>

referente à velocidade linear ao longo do eixo  $z^b$ , e  $u_\psi$ , referente à velocidade angular em torno do eixo  $z^b$  (velocidade de guinada), devem ser enviados ao veículo. Note-se que todos esses comandos são valores numéricos limitados ao intervalo  $[-1, 1]$ , de acordo com o fabricante do *Bebop 2*. Tais comandos podem ser gerados por um *joystick* operado por um piloto humano, ou sintetizados por um algoritmo computacional, como é o caso neste artigo. Nos dois casos, um canal Wi-Fi fornece o link para enviar comandos ao drone e receber dados relativos ao seu estado atual.

Comandos de movimento gerados por computador devem ser enviados ao *Bebop 2* por meio do ROS (do inglês *Robot Operating System*<sup>2</sup>), que atua como uma interface de transmissão de mensagens, realizando a comunicação inter processos. Os processos, neste artigo, são o sistema de controle da navegação do drone, que é executado em um computador externo (uma estação terrestre), e o código nativo em execução no processador a bordo do veículo (processador *dual core* com placa GPU *quad-core* e capacidade de armazenamento de 8 GB de memória *flash*), que trata os comandos recebidos e os dados sensoriais a serem enviados.

O *firmware* a bordo do *Bebop 2* também permite configurar limites para as velocidades e as inclinações máximas do veículo. O piloto automático inclui controladores de baixo nível responsáveis por estabilizar a atitude do drone (Santana et al., 2016). Para aplicações que não exigem manobras agressivas, como inspeção agrícola ou de infraestruturas, por exemplo, pode-se limitar os valores dos ângulos  $\theta$  e  $\phi$  de arfagem e rolagem a valores pequenos ( $5^\circ$ , neste trabalho), como foi feito em (Santana et al., 2016) para o quadrimotor *AR.Drone*. Portanto, pode-se considerar que tais ângulos são muito próximos de zero, não influenciando na matriz de rotação que relaciona os dois sistemas de coordenadas  $^b$  e  $^w$ , não influenciando na geração de forças e torques associados ao movimento do corpo rígido. Essa suposição significa assumir que o *Bebop 2* desenvolve velocidades lineares nos eixos  $x^b$ ,  $y^b$  e  $z^b$ , além de poder girar em torno do eixo  $z^b$ , sempre preservando os ângulos de arfagem e rolagem em zero (sem inclinar-se). Além disso, a dinâmica associada a  $\theta$  e  $\phi$  é muito mais rápida que aquela associada à translação do veículo, de modo que se pode supor que os comandos  $u_\phi$  e  $u_\theta$  causam instantaneamente velocidades ao longo dos eixos  $y^b$  e  $x^b$ . Assim, os comandos enviados ao drone podem ser vistos como comandos de velocidade  $u_{v_x}$ , para translação no eixo  $x^b$ ,  $u_{v_y}$ , para translação no eixo  $y^b$ ,  $u_{v_z}$ , para translação no eixo  $z^b$ , e  $u_{v_\psi}$ , para rotação em torno do eixo  $z^b$ , que compõem o vetor  $\mathbf{U} = [u_{v_x} \ u_{v_y} \ u_{v_z} \ u_{v_\psi}]^T$  de comandos. Quanto aos dados de interesse coletados pelo *firmware* do veículo, eles se tornam  $[x \ y \ z \ \psi]^T$ , em coordenadas globais, além de  $[\dot{x} \ \dot{y} \ \dot{z} \ \dot{\psi}]^T$ , estes nas coordenadas do veículo. Portanto o quadrimotor pode ser tratado, com as simplificações para  $\theta$  e  $\phi$ , como um sistema completamente atuado de ordem quatro.

Na sequência se proporá um modelo simplificado para representar a dinâmica do *Bebop 2*, assumindo que o mesmo está voando com inclinações limitadas a  $5^\circ$ . Esse modelo inclui alguns parâmetros dinâmicos, que serão estimados, e o modelo assim obtido será usado para projetar controladores de alto nível para esse VANT. Para concluir o trabalho, os controladores projetados serão testados em voo, o que permitirá concluir que

é viável utilizar o modelo simplificado proposto. Para abordar esses tópicos, o artigo é dividido em poucas seções, daqui em diante, começando com a Seção 2, na qual o modelo proposto para o VANT *Bebop 2* é discutido. Na sequência, a Seção 3 discute a estimação dos parâmetros do modelo proposto e valida o modelo obtido através de dados gerados pelo modelo e dados obtidos em voos com o veículo. Continuando, a Seção 4 usa o modelo proposto para projetar um controlador para um *Bebop 2*, enquanto a Seção 5 trata do controle da navegação de uma formação de dois *Bebop 2*, para mostrar que o modelo proposto, com os parâmetros estimados, é adequado para projetar controladores de alto nível para guiar o drone. Finalmente, a Seção 6 destaca as principais conclusões do trabalho.

## 2. O MODELO PROPOSTO

Tendo em conta as simplificações mencionadas na seção anterior, se propõe o modelo dinâmico simplificado

$$\ddot{\mathbf{X}} = \mathbf{f}_1 \mathbf{U} - \mathbf{f}_2 \dot{\mathbf{X}} \quad (1)$$

para o quadrimotor *Bebop 2*, no qual  $\mathbf{U} = [u_{v_x} \ u_{v_y} \ u_{v_z} \ u_{v_\psi}]^T$  é o vetor de comandos para as velocidades lineares em  $x$ ,  $y$  e  $z$  e a velocidade angular ao redor do eixo  $z$ , valores esses no intervalo  $[-1, 1]$ , enquanto que  $\dot{\mathbf{X}} = [\dot{x} \ \dot{y} \ \dot{z} \ \dot{\psi}]^T$  é o vetor de velocidades instantâneas e  $\ddot{\mathbf{X}} = [\ddot{x} \ \ddot{y} \ \ddot{z} \ \ddot{\psi}]^T$  é o vetor de acelerações instantâneas. Já  $\mathbf{f}_1$  e  $\mathbf{f}_2$  são matrizes diagonais contendo os parâmetros do modelo, que serão estimados na Seção 3, sendo  $\mathbf{f}_1 = \text{diag}(k_1 \ k_3 \ k_5 \ k_7)$  e  $\mathbf{f}_2 = \text{diag}(k_2 \ k_4 \ k_6 \ k_8)$ .

Tal modelo é o mesmo proposto em (Santana et al., 2014, 2016) para o quadrimotor *AR.Drone*, e também utilizado em (Santos et al., 2017a,b, 2019) para controlar o *AR.Drone*. A razão para propor o mesmo modelo dinâmico é que os pilotos automáticos do *AR.Drone* e do *Bebop 2* permitem limitar os ângulos de arfagem e rolagem, base para a simplificação do modelo.

Todas as variáveis em (1) são referenciadas ao sistema de coordenadas do veículo (eixos  $x^b y^b z^b$  na Figura 1). Para considerar as velocidades  $\dot{\mathbf{X}} = [\dot{x} \ \dot{y} \ \dot{z} \ \dot{\psi}]^T$  referenciadas ao sistema global de coordenadas (eixos  $x^w y^w z^w$ ), a matriz de rotação

$$\mathbf{R} = \begin{bmatrix} \cos \psi & -\sin \psi & 0 & 0 \\ \sin \psi & \cos \psi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

deve ser multiplicada pelas matrizes  $\mathbf{f}_1$  e  $\mathbf{f}_2$ , resultando em

$$\ddot{\mathbf{X}} = \mathbf{R} \mathbf{f}_1 \mathbf{U} - \mathbf{R} \mathbf{f}_2 \dot{\mathbf{X}} = \mathbf{F}_1 \mathbf{U} - \mathbf{F}_2 \dot{\mathbf{X}}, \quad (3)$$

que é o mesmo modelo, agora em coordenadas globais, detalhado como

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} k_1 \cos \psi & -k_3 \sin \psi & 0 & 0 \\ k_1 \sin \psi & k_3 \cos \psi & 0 & 0 \\ 0 & 0 & k_5 & 0 \\ 0 & 0 & 0 & k_7 \end{bmatrix} \begin{bmatrix} u_{v_x} \\ u_{v_y} \\ u_{v_z} \\ u_{v_\psi} \end{bmatrix} - \begin{bmatrix} k_2 \cos \psi & -k_4 \sin \psi & 0 & 0 \\ k_2 \sin \psi & k_4 \cos \psi & 0 & 0 \\ 0 & 0 & k_6 & 0 \\ 0 & 0 & 0 & k_8 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\psi} \end{bmatrix}. \quad (4)$$

Observe-se que como os ângulos de arfagem e rolagem são próximos de zero o corpo do veículo permanece sempre horizontal, independentemente do seu movimento. ou seja, os eixos  $z^b$  e  $z^w$  são sempre paralelos. Logo, não há projeções de  $\dot{z}$  e  $\dot{\psi}$  sobre os eixos  $x^w$  e  $y^w$ , e por isto a matriz  $\mathbf{R}$  resume-se à simplificação mostrada em (2).

<sup>2</sup> <https://www.ros.org/>

### 3. ESTIMANDO OS PARÂMETROS DO MODELO

Os parâmetros do modelo dinâmico do quadrimotor *Bebop 2*, que são os elementos de  $\mathbf{f}_1$  e  $\mathbf{f}_2$  (ver (1)), foram estimados conforme o procedimento descrito em (Santos et al., 2017b). Porém, aqui se utilizou uma configuração com quatro câmaras do sistema de captura de movimento *OptiTrack*<sup>3</sup> para obter a posição e orientação do quadrimotor em voo.

#### 3.1 Determinação dos Parâmetros

Aplicou-se um sinal de excitação conhecido a cada grau de liberdade do quadrimotor, e mediram-se as posições e orientações do veículo em resposta a tal sinal através do sistema *OptiTrack*. Amostras do sinal de excitação foram aplicadas a cada grau de liberdade do veículo, excitando-se os outros três com zero, e uma sequência de posições e orientações foi coletada. Diferenciando-se numericamente tais dados se obtiveram as velocidades e acelerações (ver (4)).

A técnica de mínimos quadrados foi utilizada, considerando-se a regressão (ver (Santos et al., 2017a))

$$\mathbf{U} = \begin{bmatrix} g_{11} & g_{12} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & g_{23} & g_{24} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \ddot{z} & \dot{z} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \ddot{\psi} & \dot{\psi} \end{bmatrix} \boldsymbol{\theta} = \mathbf{G}\boldsymbol{\theta}, \quad (5)$$

onde

$$g_{11} = \ddot{x} \cos \psi + \dot{y} \sin \psi, \quad g_{12} = \dot{x} \cos \psi + \dot{y} \sin \psi, \\ g_{23} = \ddot{x} \cos \psi + \dot{y} \sin \psi, \quad g_{24} = \dot{x} \cos \psi + \dot{y} \sin \psi,$$

e

$$\boldsymbol{\theta} = [\theta_1 \quad \theta_2 \quad \theta_3 \quad \theta_4 \quad \theta_5 \quad \theta_6 \quad \theta_7 \quad \theta_8]^T \\ = \left[ \frac{1}{k_1} \quad \frac{k_2}{k_1} \quad \frac{1}{k_3} \quad \frac{k_4}{k_3} \quad \frac{1}{k_5} \quad \frac{k_6}{k_5} \quad \frac{1}{k_7} \quad \frac{k_8}{k_7} \right]^T. \quad (6)$$

Note-se que as posições  $x$ ,  $y$  e  $z$  e a orientação  $\psi$  foram obtidas usando o sistema *OptiTrack*. Logo, elas se referem ao sistema global de coordenadas, assim como as velocidades e acelerações obtidas por diferenciação numérica.

Para cada entrada de controle do quadrimotor, isto é, para cada uma das variáveis  $u_{v_x}$ ,  $u_{v_y}$ ,  $u_{v_z}$  e  $u_{v_\psi}$ , o sinal de excitação

$$u_d(t) = \frac{0,4}{4,5} [3 \sin(0,2\pi t) + \sin(0,6\pi t) + 0,5 \sin(\pi t)], \quad (7)$$

amostrado com 200 ms, foi aplicado durante 120 s, mantendo-se as outras três entradas fixas em zero. Em outras palavras, enquanto  $u_{v_x}$  recebia os valores em (7)  $u_{v_y} = 0$ ,  $u_{v_z} = 0$ , e  $u_{v_\psi} = 0$ . Ou seja,  $u_d(t)$  significa  $u_{v_x}(t)$ ,  $u_{v_y}(t)$ ,  $u_{v_z}(t)$  e  $u_{v_\psi}(t)$ , em cada um dos voos realizados para a coleta de dados.

Portanto, usando (5) e (6), e considerando o sinal de entrada em (7) e o conjunto de medidas de posição/orientação fornecido pelo sistema *OptiTrack*, se estimaram os elementos das matrizes  $\mathbf{f}_1 = \text{diag}(k_1, k_3, k_5, k_7)$  e  $\mathbf{f}_2 = \text{diag}(k_2, k_4, k_6, k_8)$  em (1). Para o voo com o sinal de excitação em (7) aplicado à entrada de controle  $u_{v_x}$ , com  $u_{v_y} = 0$ ,  $u_{v_z} = 0$  e  $u_{v_\psi} = 0$ , o gráfico da Figura 2 mostra a convergência dos valores dos parâmetros  $\theta_1$  e  $\theta_2$  de (6), dos quais os parâmetros  $k_1$  e  $k_2$  são obtidos. Repetindo-se o procedimento para as demais entradas,  $u_{v_y}$ ,  $u_{v_z}$  e  $u_{v_\psi}$ , os demais elementos do vetor  $\boldsymbol{\theta}$  são obtidos, e a partir deles os parâmetros  $k_1, \dots, k_8$  são obtidos.

<sup>3</sup> <https://optitrack.com/motion-capture-robotics/>

Assim sendo, as matrizes  $\mathbf{f}_1$  e  $\mathbf{f}_2$  obtidas são

$$\mathbf{f}_1 = \begin{bmatrix} 0,8417 & 0 & 0 & 0 \\ 0 & 0,8354 & 0 & 0 \\ 0 & 0 & 3,966 & 0 \\ 0 & 0 & 0 & 9,8524 \end{bmatrix} \\ \mathbf{f}_2 = \begin{bmatrix} 0,18227 & 0 & 0 & 0 \\ 0 & 0,17095 & 0 & 0 \\ 0 & 0 & 4,001 & 0 \\ 0 & 0 & 0 & 4,7295 \end{bmatrix} \quad (8)$$

Note-se, porém, que os valores dos parâmetros em (8) são válidos para a configuração adotada, que corresponde aos limites 5° para  $\theta$  e  $\phi$ , 1[m/s] para a velocidade linear no eixo  $z^b$  do sistema de coordenadas do veículo, e 100[°/s] para a velocidade angular ao redor desse mesmo eixo. Se tais limites forem alterados os parâmetros  $\theta_1, \dots, \theta_8$  certamente variarão. Para poder utilizar tais valores dos referidos parâmetros é que os mesmos limites acima mencionados foram mantidos para os voos discutidos adiante, para os quais se projetaram controladores utilizando o modelo com os parâmetros estimados nesta subseção.

#### 3.2 Validação do Modelo

Para validar o modelo obtido, o sinal senoidal composto

$$u_v(t) = 0,2[\sin(0,2\pi t) + \sin(0,4\pi t)], \quad (9)$$

diferente daquele usado para estimar os parâmetros do modelo, foi aplicado a cada entrada de controle, mantendo-se as demais entradas de controle em zero, de forma similar ao que foi feito para estimar os parâmetros  $k_1, \dots, k_8$ . A amostragem também foi em 200 ms, e a medição durou entre 70 e 80 s. A resposta correspondente foi medida, para todas as saídas de interesse, novamente utilizando o sistema *OptiTrack*. As velocidades correspondentes, obtidas a partir do modelo estimado e a partir das medidas de posição e orientação do quadrimotor, quando excitado pelo sinal de teste em (9), são apresentadas na Figura 3, para cada grau de liberdade (simulação se refere a dados coletados a partir do modelo estimado, e experimento se refere a medidas tomadas com o veículo em voo).

Como a figura mostra, o modelo estimado aproxima muito bem o comportamento real do quadrimotor *Bebop 2*. A qualidade de tal modelo também foi avaliada através de sua utilização no projeto de controladores de voo para um único *Bebop 2* e para uma formação de dois *Bebop 2*, em tarefas de seguimento de trajetória (ver Seções 4 e 5).

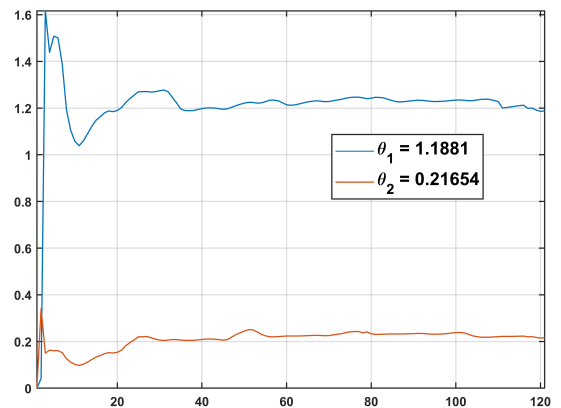


Figura 2. Convergência dos parâmetros  $\theta_1$  e  $\theta_2$  de (6).

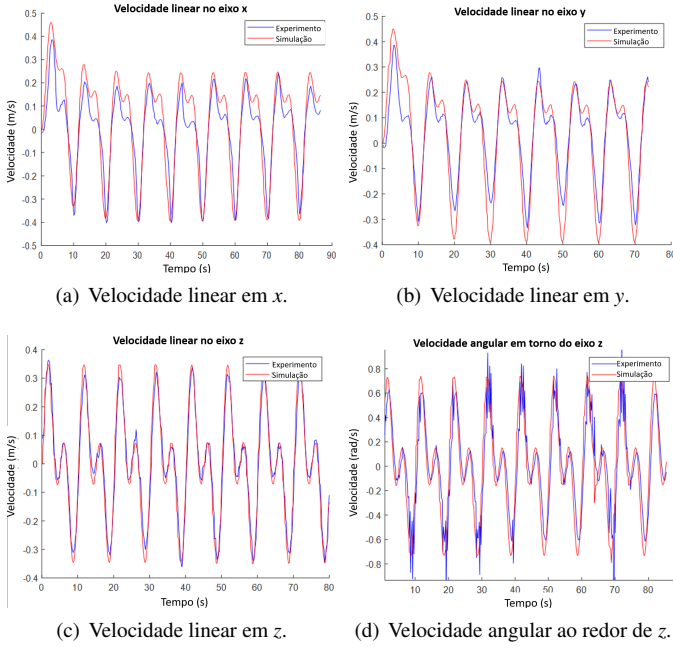


Figura 3. Velocidades lineares em  $x^b$ ,  $y^b$  e  $z^b$  e velocidade angular em torno do eixo  $z^b$ , obtidas do modelo e do veículo real.

#### 4. CONTROLE DO QUADRIMOTOR *BEBOP 2*

O modelo proposto na Seção 2, com os parâmetros estimados na Seção 3, será, daqui em diante, usado para projetar um controlador para guiar a navegação de um *Bebop 2*, como foi feito em (Santana et al., 2016) para o quadrimotor *AR.Drone*.

Assim é que a lei de controle

$$\mathbf{U}_d = \mathbf{F}_1^{-1} \left[ \ddot{\mathbf{X}}_d + \mathbf{K}_d \dot{\mathbf{X}} + \mathbf{K}_p \mathbf{X} + \mathbf{F}_2 \dot{\mathbf{X}} \right], \quad (10)$$

baseada na técnica de dinâmica inversa, é proposta, a qual tanto pode ser utilizada para tarefas de posicionamento quanto seguimento de trajetória. Em tal lei de controle  $\mathbf{X}_d$  representa a trajetória a ser seguida, sendo, portanto, uma função do tempo. Note-se que a velocidade da trajetória a ser seguida, dada por  $\dot{\mathbf{X}}_d$ , também é uma função do tempo. Para o caso de uma tarefa de posicionamento (ou regulação do erro de posição),  $\mathbf{X}_d$  é um valor constante, portanto com velocidade  $\dot{\mathbf{X}}_d$  nula. Por sua vez,  $\mathbf{X}$  é a posição instantânea do veículo ao longo do tempo. De tais valores se obtém  $\mathbf{X} = \mathbf{X}_d - \mathbf{X}$  e  $\dot{\mathbf{X}} = \dot{\mathbf{X}}_d - \dot{\mathbf{X}}$ , que são os erros de posição e de velocidade. Quanto às matrizes  $\mathbf{K}_p$  e  $\mathbf{K}_d$ , elas são matrizes de ganho  $4 \times 4$ , diagonais e positivas definidas.

Introduzindo tal lei de controle em (4) resulta que

$$\ddot{\mathbf{X}} + \mathbf{K}_d \dot{\mathbf{X}} + \mathbf{K}_p \mathbf{X} = \mathbf{0}, \quad (11)$$

que é a equação de malha fechada do sistema, na qual  $\ddot{\mathbf{X}} = \ddot{\mathbf{X}}_d - \ddot{\mathbf{X}}$ . Em termos de estabilidade do sistema quando se fecha a malha com o controlador proposto, o erro de posição  $\mathbf{X}$ , cuja dinâmica é dada por (11), converge assintoticamente a zero, de acordo com a teoria de Lyapunov combinada com o Teorema de La Salle (Vidyasagar, 2002), assim como sua derivada  $\dot{\mathbf{X}}$ .

O controlador assim projetado é utilizado, na sequência, para guiar um quadrimotor *Bebop 2* em uma tarefa de seguimento de trajetória, num ambiente exterior. O experimento é montado de acordo com a ilustração da Figura 4. São utilizados dois com-

putadores portáteis, um deles com uma extensão LINUX sob a qual roda uma instalação ROS. O segundo deles roda o código MATLAB correspondente ao sistema de controle, sob o sistema operacional WINDOWS 10. O *joystick* é usado pelo operador para sobrepor-se ao controlador automático e tomar o controle manual do drone, parando o experimento em uma situação de insegurança. Completando o sistema tem-se os satélites do GPS e um roteador, cuja responsabilidade é estabelecer o canal Wi-Fi que conecta o *Bebop 2* e o computador que roda a instalação ROS e o drone, para envio de comandos e recepção dos dados correspondentes ao estado atual da aeronave.

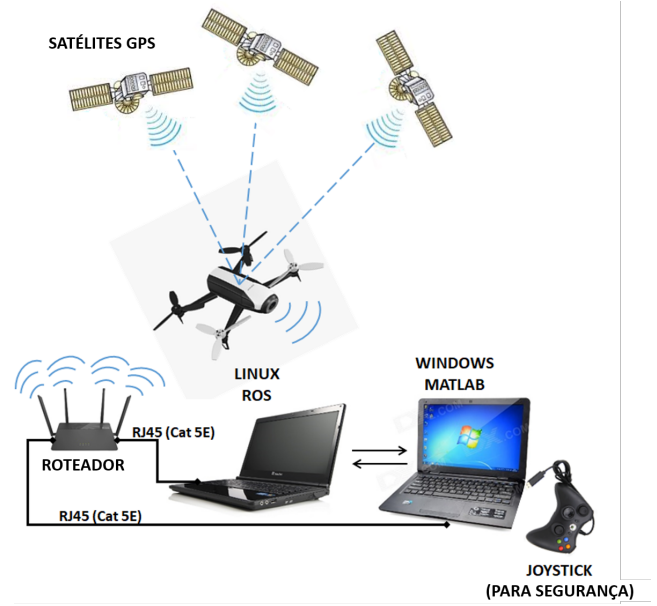


Figura 4. Sistema utilizado para rodar o experimento de voo externo com um só drone.

A estrutura da Figura 4 é então utilizada para guiar o *Bebop 2* para seguir uma trajetória na forma da lemniscata de Bernoulli (uma curva similar a um oito), cuja descrição é

$$\mathbf{X}_d = [x = 5 \cos(\omega t) \text{ m} \quad y = 5 \sin(2\omega t) \text{ m} \quad z = 8 \text{ m}]^T,$$

com  $\omega = \frac{2\pi}{60} \text{ rad/s}$ , e para a qual

$$\dot{\mathbf{X}}_d = [\dot{x} = -5\omega \sin(\omega t) \quad \dot{y} = 10\omega \cos(2\omega t) \quad \dot{z} = 0]^T.$$

A informação sobre o estado atual do VANT durante seu voo é fornecida por seus sensores internos, que incluem uma Unidade de Medição Inercial - UMI (do inglês Inertial Measurement Unit - IMU), um módulo receptor de GPS, uma câmera de vídeo apontada para baixo, um sensor ultrassônico e um sensor barométrico. O *firmware* do *Bebop 2* trata os dados fornecidos por todos esses sensores, usando um algoritmo proprietário de fusão de dados, e entrega a posição e orientação do VANT, a saber  $[x \ y \ z \ \psi]^T$ , que são os elementos do vetor  $\mathbf{X}$  de (1) e (4). A frequência de amostragem, então, é  $5 \text{ Hz}$ , como definido pelo fabricante do quadrimotor.

A posição e orientação inicial do robô é sua posição e orientação *default* após decolar, ou seja,  $\mathbf{X}_0 = [0 \text{ m} \ 0 \text{ m} \ 1 \text{ m} \ 0 \text{ rad}]^T$ , valores que correspondem ao instante em que o controlador automático assume o comando do veículo e o experimento se inicia. Por fim, os valores adotados para as matrizes de ganho  $\mathbf{K}_p$  e  $\mathbf{K}_d$  foram, respectivamente,  $\text{diag}(1, 1, 3, 1)$  and  $\text{diag}(1, 5, 1, 5, 2, 2)$ .

As Figuras 5 e 6 mostram o percurso do *Bebop 2* quando seguindo a trajetória lemniscata de Bernoulli e os correspondentes erros de posição e orientação. Como se pode ver em tais figuras, o drone foi capaz de seguir a trajetória proposta bastante bem, a despeito de rajadas de vento (vale lembrar que o voo é em ambiente exterior), validando o modelo que foi utilizado no projeto do controlador (deve-se ter em conta que a técnica de dinâmica inversa exige um bom modelo para o projeto do controlador).

## 5. CONTROLE DE UMA FORMAÇÃO DE DOIS *BEBOP 2*

Para reforçar a validação do modelo proposto em (4) para o *Bebop 2*, ele será utilizado nesta seção para projetar um controlador para guiar uma formação contendo dois de tal quadrimotor. Trata-se de uma formação em linha homogênea, que corresponde a uma estrutura virtual, exatamente a linha imaginária que conecta os dois veículos.

A formulação matemática do problema é a mesma encontrada em (Rabelo et al., 2018) e (Moreira et al., 2019), com a diferença que aqui são considerados dois VANTs (formação homogênea), e não um VANT e um veículo terrestre (formação

heterogênea). Em ambos os casos se definem dois espaços de variáveis, a saber o espaço da formação e o espaço dos robôs. O espaço da formação é caracterizado pelas chamadas variáveis da formação, a saber,

$$\mathbf{q} = [x_f \ y_f \ z_f \ \rho_f \ \alpha_f \ \beta_f]^T$$

(ver Figura 7), que correspondem à posição  $(x_f, y_f, z_f)$  da extremidade da linha virtual que coincide com o primeiro *Bebop 2*, a distância  $\rho_f$  entre os dois quadrimotores, o ângulo  $\alpha_f$  entre a projeção da linha virtual que une os dois drones no plano  $x^w y^w$  e o eixo  $x^w$ , e o ângulo  $\beta_f$  formado pela linha virtual que une os dois quadrimotores e o plano  $x^w y^w$ . Já o espaço dos robôs é caracterizado pelas variáveis

$$\mathbf{x} = [x_1 \ y_1 \ z_1 \ x_2 \ y_2 \ z_2]^T,$$

que correspondem às suas posições.

A passagem de um espaço ao outro se dá por meio de duas transformações: a) a transformação direta,  $\mathbf{x} \rightarrow \mathbf{q}$ , que permite obter o estado atual da formação a partir das posições medidas dos dois VANTs, e b) a transformação inversa  $\mathbf{q} \rightarrow \mathbf{x}$ , que define como os VANTs devem se posicionar para obter o estado desejado da formação. Especificamente, a transformação direta é dada por

$$\mathbf{q} = \begin{bmatrix} x_f = x_1 \\ y_f = y_1 \\ z_f = z_1 \\ \rho_f = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \\ \alpha_f = \arctan\left(\frac{y_2 - y_1}{x_2 - x_1}\right) \\ \beta_f = \arctan\left(\frac{z_2 - z_1}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}}\right) \end{bmatrix} = f(\mathbf{x}). \quad (12)$$

A Figura 8 ilustra o sistema de controle projetado para conduzir a formação de dois *Bebop 2*. Ali se pode notar que há um controlador da formação, cuja função é definir como a formação deve variar ao longo do tempo (ele gera as variáveis  $\dot{\mathbf{q}}_r$ ) para alcançar o estado desejado  $\mathbf{q}_d$ . A partir de  $\dot{\mathbf{q}}_r$  se obtém as velocidades de referência  $\dot{\mathbf{x}}_r$  para os VANTs, através da relação  $\dot{\mathbf{x}}_r = \mathbf{J}^{-1} \dot{\mathbf{q}}_r$ . Observe-se que  $\dot{\mathbf{x}}_r$  está em coordenadas globais, pois as variáveis da formação estão em coordenadas globais. Quanto à matriz Jacobiana  $\mathbf{J}$ , ela é dada por  $\mathbf{q} = \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} \dot{\mathbf{x}} = \mathbf{J} \dot{\mathbf{x}}$ .

Então, as velocidades de referência  $\dot{\mathbf{x}}_r$  dos VANTs são separadas em dois grupos,  $\dot{\mathbf{x}}_{r1}$  e  $\dot{\mathbf{x}}_{r2}$ , um para cada VANT. Tais velocidades ainda estão em coordenadas globais, e são transfor-

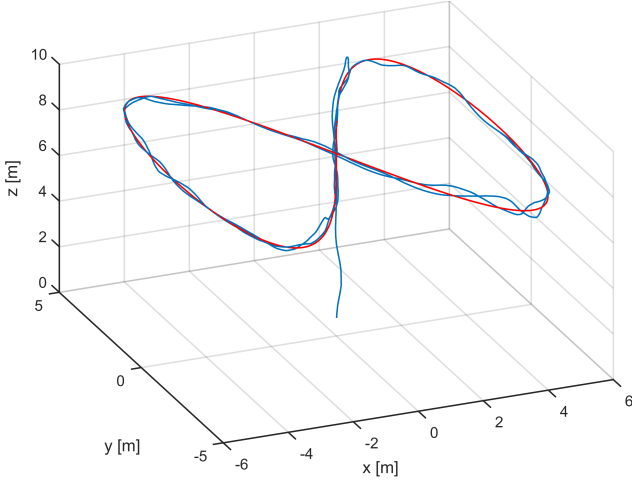


Figura 5. Trajetória do quadrimotor ao seguir a trajetória lemniscata de Bernoulli.

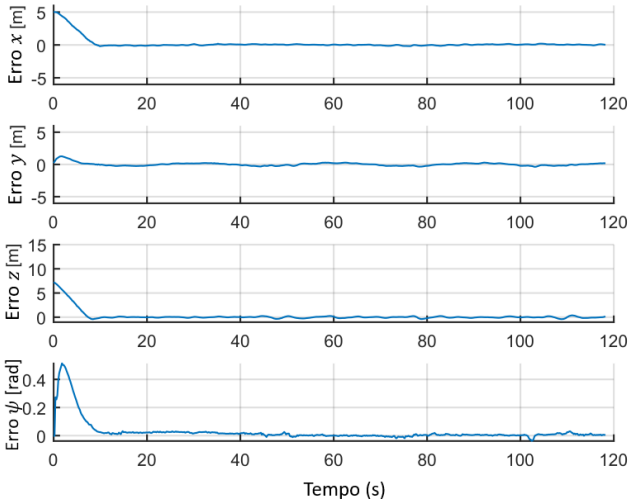


Figura 6. Erros de posição e orientação para o quadrimotor *Bebop 2* seguindo a trajetória lemniscata de Bernoulli.

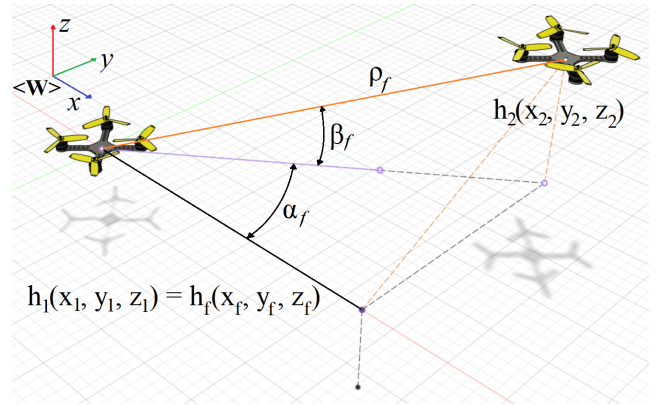


Figura 7. Representação de uma formação de dois VANTs.

madras para os sistemas de coordenadas de cada VANT quando multiplicadas pela matriz de cinemática inversa  $\mathbf{R}^{-1}$  (ver (2)) dos robôs, gerando-se os comandos de velocidade  $\mathbf{v}_{1d}$  e  $\mathbf{v}_{2d}$  nas coordenadas dos veículos. Tais comandos de velocidade são então enviados para um compensador dinâmico (os blocos CD na Figura 8), como proposto em (Santos et al., 2019) para o *AR.Drone 2.0*. A função de cada um desses blocos é reduzir os erros de seguimento de velocidade, compensando a dinâmica dos veículos, não considerada até aqui. Um detalhe significativo é que tais compensadores podem não ser usados, se a dinâmica dos veículos puder ser desconsiderada. Quando eles são utilizados, suas saídas,  $\mathbf{u}_{1d}$  e  $\mathbf{u}_{2d}$ , são os comandos de velocidade efetivamente enviados para os dois VANTs, como mostra a figura. Note-se ainda que a entrada e a saída de cada compensador dinâmico são velocidades referenciadas ao sistema de coordenadas do veículo, enquanto que as posições  $\mathbf{x}_1$  e  $\mathbf{x}_2$  dos veículos são referenciadas ao sistema global.

No que se refere ao controlador da formação, sua ação é fazer a diferença entre o estado desejado da formação e seu estado atual ir para zero, ou seja, fazer com que  $\tilde{\mathbf{q}} = \mathbf{q}_d - \mathbf{q} \rightarrow \mathbf{0}$  quando  $t \rightarrow \infty$ . Para assegurar isso, a lei de controle

$$\dot{\mathbf{q}}_r = \dot{\mathbf{q}}_d + \mathbf{L}_1 \tanh(\mathbf{L}_2 \tilde{\mathbf{q}}) \quad (13)$$

é adotada, na qual  $\mathbf{L}_1$  e  $\mathbf{L}_2$  são matrizes constantes diagonais e positivas definidas. Já a função  $\tanh(\cdot)$  é usada como um saturador suave, para evitar a saturação dos atuadores dos robôs no caso de erros de formação  $\tilde{\mathbf{q}}$  muito grandes. Quanto a  $\dot{\mathbf{q}}_d$ , trata-se da primeira derivada temporal de  $\mathbf{q}_d$ , e é zero para tarefas de posicionamento (quando  $\mathbf{q}_d$  é uma constante) ou uma função do tempo, para tarefas de seguimento de trajetória (quando  $\mathbf{q}_d$  é função do tempo). Portanto, o controlador da Figura 8 é válido tanto para posicionamento quanto para seguimento de trajetória. Note-se, por fim, que tal controlador é assintoticamente estável, conforme (Rabelo et al., 2018).

Quanto ao compensador dinâmico associado a cada VANT, os blocos CD na Figura 8, eles são projetados conforme proposto em (Santos et al., 2019) para o quadrimotor *AR.Drone 2.0*. Para projetar tais compensadores o modelo em (1) é reescrito como

$$\mathbf{U} = \mathbf{f}_1^{-1} \ddot{\mathbf{X}} + \mathbf{f}_1^{-1} \mathbf{f}_2 \dot{\mathbf{X}} = \mathbf{D} \ddot{\mathbf{X}} + \boldsymbol{\eta}, \quad (14)$$

ou

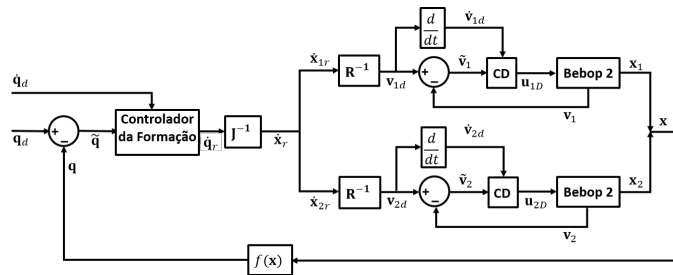


Figura 8. Estrutura do controlador adotado para guiar uma formação de dois VANTs.

$$\underbrace{\begin{bmatrix} u_{v_x} \\ u_{v_y} \\ u_{v_z} \\ u_{v_\psi} \end{bmatrix}}_{\mathbf{U}} = \underbrace{\begin{bmatrix} \frac{1}{k_1} & 0 & 0 & 0 \\ 0 & \frac{1}{k_3} & 0 & 0 \\ 0 & 0 & \frac{1}{k_5} & 0 \\ 0 & 0 & 0 & \frac{1}{k_7} \end{bmatrix}}_{\mathbf{D}} \underbrace{\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\psi} \end{bmatrix}}_{\dot{\mathbf{X}}} + \underbrace{\begin{bmatrix} \dot{x} & 0 & 0 & 0 \\ 0 & \dot{y} & 0 & 0 \\ 0 & 0 & \dot{z} & 0 \\ 0 & 0 & 0 & \dot{\psi} \end{bmatrix}}_{\boldsymbol{\eta}} \underbrace{\begin{bmatrix} k_2 \\ k_4 \\ k_6 \\ k_8 \\ k_7 \end{bmatrix}}_{\boldsymbol{\eta}}, \quad (15)$$

onde os parâmetros  $k_i, i = 1, \dots, 8$  são os elementos das matrizes  $\mathbf{f}_1$  e  $\mathbf{f}_2$  estimados na Seção 3.

A lei de controle

$$\mathbf{u}_D = \mathbf{D} \boldsymbol{\sigma} + \boldsymbol{\eta}, \quad (16)$$

originalmente proposta em (Santos et al., 2019) para o VANT *AR.Drone 2.0*, baseada na técnica de dinâmica inversa, é proposta para cada *Bebop 2* da formação. Nela

$$\boldsymbol{\sigma} = [\sigma_1 \ \sigma_2 \ \sigma_3 \ \sigma_4]^T = \dot{\mathbf{v}}_d + \mathbf{K}_D \tilde{\mathbf{v}},$$

onde  $\mathbf{K}_D$  é uma matriz de ganhos diagonal e positiva definida,  $\mathbf{v}_d$  é a velocidade desejada para o VANT,  $\mathbf{v}$  é sua velocidade atual, e  $\tilde{\mathbf{v}} = \mathbf{v}_d - \mathbf{v}$  é o erro de velocidade a ser reduzido pelo compensador dinâmico. Para obter  $\dot{\mathbf{v}}_d$  se toma a derivada, numericamente falando, a velocidade desejada  $\mathbf{v}_d$ , proveniente do controlador da formação (ver Figura 8). Note-se que a malha de controle correspondente ao compensador dinâmico é assintoticamente estável, conforme (Santos et al., 2019).

Na sequência um experimento de seguimento de trajetória com a formação de dois *Bebop 2* é mostrado, utilizando-se o esquema de controle da Figura 8 e a estrutura da Figura 4, agora com dois *Bebop 2*. O objetivo é mostrar que os erros nas variáveis de formação são próximos de zero, validando assim o modelo usado no projeto dos compensadores dinâmicos.

A pose inicial do VANT 1 é  $\mathbf{X}_1 = [0 \ m \ 0 \ m \ 1 \ m \ 0 \ rad]^T$ , e a do VANT 2 é  $\mathbf{X}_2 = [0 \ m \ 5 \ m \ 1 \ m \ 0 \ rad]^T$ . Já a formação desejada é descrita como  $\mathbf{q}_d = [(2 + 3 \cos \frac{2\pi}{20} t) \ m \ (5 + 3 \cos \frac{2\pi}{20} t) \ m \ 18 \ m \ 0 \ rad \ \frac{\pi}{6} \ rad]^T$ . A posição do primeiro VANT, que caracteriza a posição da formação, varia com o tempo, de acordo com uma trajetória circular. Quanto a  $\dot{\mathbf{q}}_d$ , tal valor é obtido por derivação de  $\mathbf{q}_d$ . Já as matrizes  $\mathbf{L}_1$  e  $\mathbf{L}_2$  do controlador da formação são dadas por  $\mathbf{L}_1 = \text{diag}(0, 5, 0, 5, 0, 5, 0, 5, 0, 5, 0, 6)$  e  $\mathbf{L}_2 = \text{diag}(1, 1, 1, 1, 1, 1)$ , enquanto a matriz  $\mathbf{K}_D$  dos compensadores dinâmicos é dada por  $\mathbf{K}_D = \text{diag}(0, 8, 0, 8, 0, 8, 0, 8)$ .

Os resultados deste experimento podem ser visualizados na Figura 9, que mostra vistas 3D e XZ dos percursos dos robôs, evidenciando as duas trajetórias circulares seguidas pelos dois *Bebop 2*, e 10, que mostra a convergência a zero dos erros nas variáveis da formação. Como se pode ver, a tarefa de seguimento de trajetória foi efetivamente cumprida pela formação, com erros pequenos, conforme esperado, validando assim o modelo usado no projeto do sistema de controle.

## 6. CONCLUSÃO

O objetivo deste artigo, mostrar que o modelo simplificado adotado para o quadrimotor *AR.Drone* também é válido, em sua estrutura, para o quadrimotor *Bebop 2* foi plenamente atingido. Tal hipótese inicial foi assumida porque ambos quadrimotores têm piloto automático a bordo, tal que as suposições que levaram ao modelo simplificado com quatro graus de liberdade

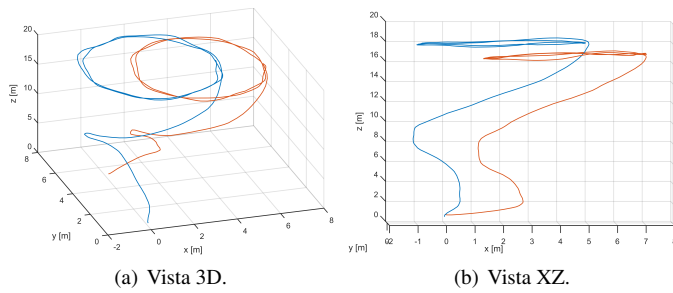


Figura 9. Percursos seguidos pelos dois *Bebop 2* durante o seguimento da trajetória circular.

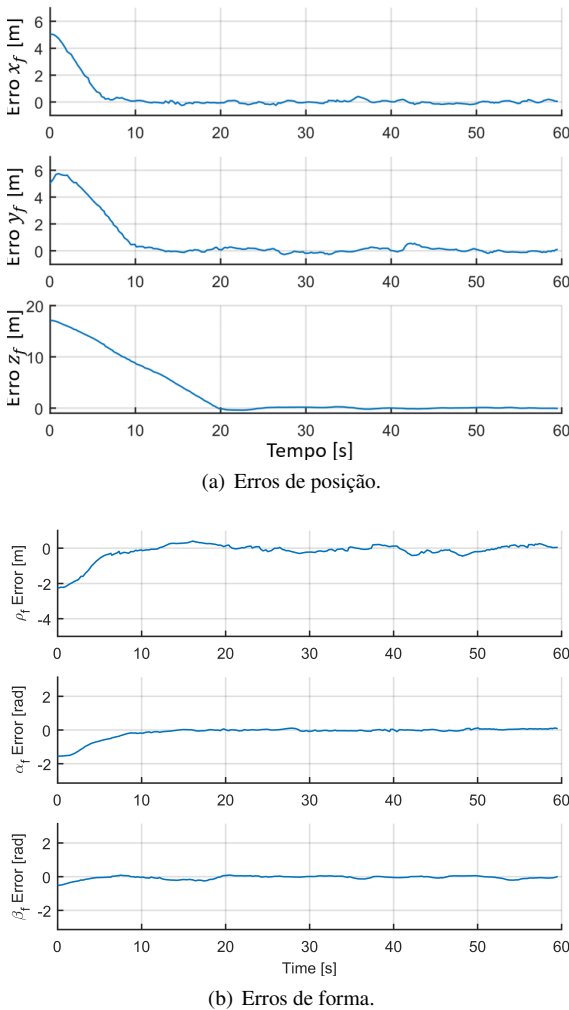


Figura 10. Erros nas variáveis da formação de dois *Bebop 2* durante o seguimento da trajetória circular.

e completamente do *AR.Drone* também são válidas para o *Bebop 2*. A diferença é somente no que se refere aos valores dos oito parâmetros do modelo, cujos valores foram aqui estimados para o *Bebop 2*.

Na sequência do trabalho o modelo proposto, com os parâmetros estimados, foi adotado para projetar controladores para guiar de forma autônoma o quadrimotor *Bebop 2* em tarefas de seguimento de trajetória, primeiro considerando apenas um quadrimotor *Bebop 2* e depois para uma formação de dois *Bebop 2*, em ambientes exteriores, nos dois casos. Os resultados obtidos alguns deles apresentados e analisados no texto, per-

mitem dizer que os controladores projetados usando o modelo simplificado proposto tiveram desempenho muito bom. Portanto, a conclusão é que o modelo proposto para o quadrimotor *Bebop 2* representa bem o comportamento dinâmico do veículo. A única restrição é que por causa da limitação nos ângulos de arfagem e rolagem, que foram limitados a valores menores que  $5^\circ$ , manobras agressivas não são permitidas.

Como conclusão final, então, para cumprir missões de posicionamento ou seguimento de trajetória que não exijam manobras agressivas o modelo simplificado aqui proposto é uma boa aproximação para o comportamento do quadrimotor *Bebop 2* em voo, confirmando-se a veracidade da hipótese inicial desse artigo.

## REFERÊNCIAS

- Amorim, L.A., Vassallo, R.F., and Sarcinelli-Filho, M. (2019). Building mosaics using images autonomously acquired by a uav. In *2019 Int Conf on Unmanned Aircraft Systems (ICUAS)*, 306–312. Atlanta, GA, USA.
- Corti, M., Cavalli, D., Cabassi, G., Vigoni, A., Degano, L., and Marino Gallina, P. (2019). Application of a low-cost camera on a uav to estimate maize nitrogen-related variables. *Precision Agriculture*, 20(4), 675–696.
- Hood, S., Benson, K., Hamod, P., Madison, D., O’Kane, J.M., and Rekleitis, I. (2017). Bird’s eye view: Cooperative exploration by ugv and uav. In *2017 Int Conf on Unmanned Aircraft Systems (ICUAS)*, 247–255. Mimi, FL, USA.
- Jordan, S., Moore, J., Hovet, S., Box, J., Perry, J., Kirsche, K., Lewis, D., and Tse, Z.T.H. (2018). State-of-the-art technologies for uav inspections. *IET Radar, Sonar Navigation*, 12(2), 151–164.
- Moreira, M.S.M., Brandão, A.S., and Sarcinelli-Filho, M. (2019). Null space based formation control for a uav landing on a ugv. In *2019 Int Conf on Unmanned Aircraft Systems (ICUAS)*, 1389–1397. Atlanta, GA, USA.
- Rabelo, M.F.S., Brandão, A.S., and Sarcinelli-Filho, M. (2018). Centralized control for an heterogeneous line formation using virtual structure approach. In *2018 Latin American Robotics Symposium (LARS)*, 135–140. João Pessoa, PB, Brazil.
- Santana, L.V., Brando, A.S., Sarcinelli-Filho, M., and Carelli, R. (2014). A trajectory tracking and 3d positioning controller for the ar.drone quadrotor. In *2014 Int Conf on Unmanned Aircraft Systems (ICUAS)*, 756–767. IEEE, Orlando, FL, USA.
- Santana, L.V., Brandão, A.S., and Sarcinelli-Filho, M. (2016). Navigation and cooperative control using the ar.drone quadrotor. *Journal of Intelligent & Robotic Systems*, 84(1), 327–350.
- Santos, M.C.P., Rosales, C.D., Sarcinelli-Filho, M., and Carelli, R. (2017a). A novel null-space-based uav trajectory tracking controller with collision avoidance. *IEEE/ASME Transactions on Mechatronics*, 22(6), 2543–2553.
- Santos, M.C.P., Rosales, C.D., Sarapura, J.A., Sarcinelli-Filho, M., and Carelli, R. (2019). An adaptive dynamic controller for quadrotor to perform trajectory tracking tasks. *Journal of Intelligent & Robotic Systems*, 93(1), 5–16.
- Santos, M.C., Santana, L.V., Brandão, A.S., Sarcinelli-Filho, M., and Carelli, R. (2017b). Indoor low-cost localization system for controlling aerial robots. *Control Engineering Practice*, 61, 93 – 111.
- Vidyasagar, M. (2002). *Nonlinear Systems Analysis*. Society for Industrial and Applied Mathematics, second edition.