

SISTEMA PARA ESTIMAÇÃO E CONTROLE DA POSIÇÃO 3D DE UM QUADRIMOTOR EM AMBIENTES INTERNOS

LUCAS VAGO SANTANA*, ALEXANDRE SANTOS BRANDÃO†, MARIO SARCINELLI FILHO‡, RICARDO CARELLI§

**Coordenadoria de Automação Industrial, Instituto Federal do Espírito Santo, Linhares, ES, Brasil*

†*Núcleo de Especialização em Robótica - NERO, Departamento de Engenharia Elétrica - DEL, Universidade Federal de Viçosa - UFV, Viçosa - MG, Brasil*

‡*Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal do Espírito Santo, Vitória, ES, Brasil*

§*INAUT - Instituto de Automática, Universidad Nacional de San Juan, San Juan, Argentina*

Emails: lucas@ifes.edu.br, alexandre.brandao@ufv.br, mario.sarcinelli@ufes.br, rcarelli@inaut.unsj.edu.ar

Abstract— This paper presents a method for the estimation and control of the 3D position of a quadrotor in indoor environments, capable to regulate an automatic hover flight over a reference. The proposed approach applies the extended Kalman filter - EKF - as framework to fuse the available information from the sensors onboard the quadrotor and a PD controller to regulate the flight. Experimental results are presented to demonstrate effectiveness of the proposed method.

Keywords— UAV, Visual-Inertial Navigation, Position Control, Autonomous Hovering

Resumo— Este artigo apresenta um método para estimação e controle de posição 3D de um quadrimotor em ambientes internos, capaz de regular automaticamente um voo pairado sobre um referencial. A abordagem proposta aplica o Filtro de Kalman Estendido - EKF - como mecanismo de fusão das informações obtidas dos sensores embarcadas no quadrimotor e um controlador PD para regular o voo. Resultados experimentais são apresentados para comprovar a eficácia do método proposto.

Palavras-chave— VANT, Navegação Inercial-Visual, Controle de Posição, Voo Pairado Automático

1 Introdução

O estudo de técnicas de controle aplicadas a veículos aéreos não tripulados (VANTs) tem sido um assunto bastante explorado no meio acadêmico, com resultados significativos já publicados (Mellinger and Kumar, 2011; Müller et al., 2011; Weiss et al., 2012).

Uma parte expressiva desses trabalhos usa como plataforma experimental veículos aéreos de pás rotativas, provavelmente por sua versatilidade de manobras em voo. Entretanto, estes resultados ficam limitados, em se tratando de aplicações reais, pois se baseiam no uso de um sistema de visão computacional com câmeras externas ao veículo para realizar sua localização. Assim sendo, alguns estudos mais recentes têm se dedicado a investigar meios de possibilitar a navegação autônoma de um VANT, utilizando apenas sensores embarcados na própria aeronave, em ambientes onde seja ou não possível utilizar dados de GPS.

Dentre os muitos trabalhos envolvendo controle de VANTs de pás rotativas utilizando sistemas de visão externos, podem ser citados (Müller et al., 2011) e (Mellinger and Kumar, 2011). Já dentre aqueles que utilizam somente os sensores embarcados nos veículos podem ser citados (Krajnik et al., 2011), (Dijkshoorn and Visser, 2011), (Engel et al., 2012) e (Weiss, 2013).

Uma breve descrição daqueles trabalhos que inspiraram a pesquisa que resultou neste artigo é apresentada na sequência.

Em (Krajnik et al., 2011) é apresentada uma série de experimentos realizados com base em controle servo-visual de posição. Um modelo para localização e navegação automática do quadrimotor AR.Drone 1.0 é proposto, comparando sua resposta ao controle com a de sistemas lineares. A estimação e o controle da posição são feitos através de transformações geométricas aplicadas aos dados coletados por um sistema de visão. Nota-se aqui a necessidade constante da visão de um alvo no controle, porém mesmo assim os resultados são muito interessantes.

Por outro lado, em (Dijkshoorn and Visser, 2011), (Engel et al., 2012) e (Weiss, 2013) métodos que oferecem maior robustez são discutidos. Estes trabalhos provavelmente representam o estado da arte na estimação e controle de posição para VANTs, por meio de sensores inerciais e visão computacional. Todos exploram o filtro de Kalman estendido (do inglês *Extended Kalman Filter - EKF*), como método para fusão dos dados originados dos diversos sensores.

Em (Dijkshoorn and Visser, 2011), em particular, as informações visuais são obtidas por extração de características das imagens através do emprego do algoritmo SURF (Bay et al., 2008),

o qual, a partir de uma correlação entre imagens, transforma tais características em métricas, que por sua vez são integradas às demais leituras sensoriais através de um EKF.

Já em (Engel et al., 2012) e (Weiss, 2013), a técnica empregada na extração de dados visuais é o PTAM (Klein and Murray, 2009), que, apesar de desenvolvido para realidade aumentada, calcula e devolve dados de posição e orientação da câmera de vídeo em um ambiente rico em características na imagem. Mais uma vez, ambos exploram o EKF para integrar os dados visuais com os dados inerciais.

Neste trabalho o que se propõe é uma técnica inspirada nestes exemplos de sucesso, com o uso de um algoritmo de extração de dados visuais e sua incorporação às demais leituras sensoriais por meio de um EKF. A técnica aplicada, discutida na Seção 3.3, apesar de ser mais simples, consegue manter a robustez necessária ao controle de posição em um voo pairado. Além disso, oferece a oportunidade de testar o EKF como meio de fusão de dados inerciais e visuais sem a necessidade de aplicar algoritmos caixa preta, como seria o caso do SURF e do PTAM.

2 A plataforma AR.Drone 2.0

A plataforma experimental escolhida no desenvolvimento deste trabalho é o quadrimotor da fabricante Parrot, modelo AR.Drone 2.0, que pode ser visto na Figura 1, juntamente com o sistema de coordenadas adotado neste trabalho.



Figura 1: AR.Drone 2.0 e os sistemas de coordenadas adotados ($\{w\}$ é o sistema de coordenadas global, e $\{b\}$ é o sistema de coordenadas do veículo).

Trata-se de um veículo aéreo autônomo comercializado como brinquedo de última geração, originalmente projetado para ser controlado por *smartphones* e *tablets* via rede *Wi-Fi*. Dessa forma, o AR.Drone permite seu controle a partir de qualquer dispositivo compatível com este formato de comunicação. É importante destacar que a plataforma oferece a facilidade de compra no mercado, a um custo reduzido ¹, se comparado a outras semelhantes. Além disso, a Parrot fornece gratuitamente um conjunto de ferramentas

¹Atualmente em torno de trezentos dólares americanos - USD 300,00

de *software*, que facilita o desenvolvimento de algoritmos de controle para o AR.Drone. Detalhes adicionais podem ser encontrados em (Piskorski et al., 2012).

2.1 Dados sensoriais

O AR.Drone 2.0 vem equipado de fábrica com acelerômetros, giroscópios, magnetômetros, duas câmeras de vídeo e um computador de bordo que gerencia estes sensores e a rede de comunicação sem fio do veículo. O *firmware* instalado no VANT é capaz de realizar de forma automática os procedimentos de decolagem, aterrissagem e a estabilização de voo, além de também responder a comandos de movimento externos, quando recebe algum. Assim, como resultado dos algoritmos internos, fica disponível ao desenvolvedor o conjunto de variáveis $\mathbf{q} = [z \ v_x \ v_y \ \phi \ \theta \ \psi]$, onde

- z representa a altitude do veículo em relação ao solo abaixo dele;
- v_x e v_y representam as velocidades lineares desenvolvidas, nos referenciais x_b e y_b ;
- ϕ , θ e ψ representam os ângulos de inclinação apresentados pelo AR.Drone, todos referenciados no sistema de coordenadas global.

A descrição de tais informações, bem como o acesso a elas, são discutidos em (Piskorski et al., 2012). Em (Bristeau et al., 2011) são analisadas as técnicas aplicadas internamente no *firmware* para obtenção das mesmas. Portanto, recomenda-se ao leitor interessado verificar estes documentos em busca de maiores esclarecimentos a respeito da tecnologia envolvida no funcionamento interno do AR.Drone.

2.2 Controle de movimento do AR.Drone

Os comandos de movimento para o AR.Drone são enviados através de um protocolo específico em sua rede *Wi-Fi*. Neste protocolo, os sinais de comando são normalizados, de sorte que cada elemento do vetor de sinais de controle $\mathbf{u} = [u_z, u_\psi, u_\phi, u_\theta] \in [-1.0, +1.0]$, sendo que

- u_z representa um comando de velocidade de deslocamento linear sobre o eixo z ;
- u_ψ representa um comando de velocidade angular em torno do eixo z ;
- u_ϕ representa um comando de inclinação em relação a x_w ;
- u_θ representa um comando de inclinação em relação a y_w .

A exemplo dos relatos em (Krajnik et al., 2011) e (Sa et al., 2013), é possível assumir que

a resposta executada pelo VANT a partir da aplicação desses sinais se dá através de uma relação linear. Tais sinais (\mathbf{u}) são transformados em referências para os controladores internos do AR.Drone. Então, em termos numéricos isto significa que há um limite estabelecido internamente para velocidade linear na direção z , que é $v_{z_{max}} = 0.7$ m/s. Ao receber o comando $u_z = 0.5$, o AR.Drone executará uma velocidade $v_z = 0.35$ m/s. O mesmo raciocínio pode ser tomado para os demais sinais de controle.

Apesar de todas estas características, o AR.Drone não possui a capacidade de realizar um voo pairado completamente autônomo. Os procedimentos de decolagem e aterrissagem acontecem de forma autônoma, porém uma vez no ar o VANT começa a "deslizar", deslocando-se da sua posição inicial. Tal efeito é conhecido na literatura específica como *drifting*, e é decorrente dos erros do sistema de medição à medida que há uma evolução temporal. Assim, para realizar o controle proposto se faz necessário estimar e corrigir a variação da posição através de um sistema de controle em malha fechada, assegurando a realização de voo pairado sobre uma referência dada.

3 Estimação da posição 3D via EKF

O filtro de Kalman estendido já há alguns anos vem sendo aplicado como ferramenta para estimação de estados e no auxílio ao controle de robôs (Thrun et al., 2005). O filtro exige um modelo de transição e observação dos estados no formato

$$\begin{aligned}\mathbf{x}_k &= f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) + \mathbf{w}_{k-1} \\ \mathbf{z}_k &= h(\mathbf{x}_k) + \mathbf{v}_k\end{aligned}\quad (1)$$

onde f e h são funções diferenciáveis não lineares, e \mathbf{w} e \mathbf{v} representam ruídos Gaussianos de média zero. Já o índice k indica o instante de tempo discreto.

Para sua implementação primeiramente é preciso definir os estados que serão estimados pelo filtro, neste caso

$$\mathbf{x}_k = [x \ y \ z \ \dot{x} \ \dot{y} \ \dot{z} \ v_x \ v_y \ \phi \ \theta \ \psi \ \dot{\psi}]^T, \quad (2)$$

onde x , y , z , \dot{x} , \dot{y} e \dot{z} são as posições, em (m), e as velocidades lineares, em (m/s), do AR.Drone em relação ao referencial global. Já v_x e v_y são as velocidades lineares desenvolvidas pelo veículo no seu próprio referencial, em (m/s). Finalmente, ϕ , θ e ψ são as posições angulares, em (rad), e $\dot{\psi}$ é a velocidade angular em torno do eixo z , em (rad/s).

3.1 Modelo de predição dos estados

O modelo de predição f utilizado na estrutura do filtro define como um estado evolui de um instante

de tempo até outro. As equações adotadas foram as seguintes:

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ z_{k+1} \\ \dot{x}_{k+1} \\ \dot{y}_{k+1} \\ \dot{z}_{k+1} \\ v_{x_{k+1}} \\ v_{y_{k+1}} \\ \phi_{k+1} \\ \theta_{k+1} \\ \psi_{k+1} \\ \dot{\psi}_{k+1} \end{bmatrix} = \begin{bmatrix} x_k + \delta t \cdot \dot{x}_k \\ y_k + \delta t \cdot \dot{y}_k \\ z_k + \delta t \cdot \dot{z}_k \\ c_{\psi_k} \cdot v_{x_k} - s_{\psi_k} \cdot v_{y_k} \\ s_{\psi_k} \cdot v_{x_k} + c_{\psi_k} \cdot v_{y_k} \\ \dot{z} + \delta t \cdot \ddot{z}(\mathbf{x}, \mathbf{u}) \\ v_{x_k} + \delta t \cdot \dot{v}_{x_k}(\mathbf{x}) \\ v_{y_k} + \delta t \cdot \dot{v}_{y_k}(\mathbf{x}) \\ \phi_k + \delta t \cdot \dot{\phi}_k(\mathbf{x}, \mathbf{u}) \\ \theta_k + \delta t \cdot \dot{\theta}_k(\mathbf{x}, \mathbf{u}) \\ \psi_k + \delta t \cdot \dot{\psi}_k \\ \psi_k + \delta t \cdot \dot{\psi}_k(\mathbf{x}, \mathbf{u}) \end{bmatrix}, \quad (3)$$

onde δt , representa o intervalo de amostragem, e c_{ψ} e s_{ψ} representam $\cos(\psi)$ e $\sin(\psi)$, respectivamente. Destaca-se que tais modelos foram obtidos observando o comportamento do AR.Drone em voo e levando em consideração os relatos de (Krajnik et al., 2011), (Engel et al., 2012) e (Sa et al., 2013).

Devido à natureza subatuada do quadrimotor (Brandão, 2013), assume-se que as acelerações horizontais \dot{v}_x e \dot{v}_y são geradas indiretamente pelos estados \mathbf{x} . Para isto, considera-se a força de propulsão (f_{prop}) resultante dos quatro rotores constante, de forma que \dot{v}_x e \dot{v}_y são modeladas como função apenas da inclinação do veículo e da força de arrasto do ar (f_{arr}), aqui considerada proporcional à velocidade naquela direção, isto é

$$\begin{aligned}\dot{v}_x &= f_{prop_x} - f_{arr_x} \\ \dot{v}_y &= f_{prop_y} - f_{arr_y},\end{aligned}\quad (4)$$

ou

$$\begin{aligned}\dot{v}_x &= K_1(s_{\psi}s_{\phi}c_{\theta} + c_{\psi}s_{\theta}) - K_2v_x \\ \dot{v}_y &= K_3(-c_{\psi}s_{\phi}c_{\theta} + s_{\psi}s_{\theta}) - K_4v_y,\end{aligned}\quad (5)$$

com as constantes K_1 , K_2 , K_3 e K_4 representando coeficientes de proporcionalidade identificados experimentalmente.

De modo análogo, a influência dos sinais de controle $\mathbf{u} = [u_z, u_{\dot{\psi}}, u_{\phi}, u_{\theta}]$ é incorporada no modelo de predição no cálculo de $\ddot{z}(\mathbf{x}, \mathbf{u})$, $\ddot{\psi}(\mathbf{x}, \mathbf{u})$, $\dot{\phi}(\mathbf{x}, \mathbf{u})$ e $\dot{\theta}(\mathbf{x}, \mathbf{u})$, através dos sistemas lineares

$$\begin{aligned}\ddot{z}(\mathbf{x}, \mathbf{u}) &= K_5u_z - K_6\dot{z} \\ \ddot{\psi}(\mathbf{x}, \mathbf{u}) &= K_7u_{\dot{\psi}} - K_8\dot{\psi} \\ \dot{\phi}(\mathbf{x}, \mathbf{u}) &= K_9u_{\phi} - K_{10}\phi \\ \dot{\theta}(\mathbf{x}, \mathbf{u}) &= K_{11}u_{\theta} - K_{12}\theta,\end{aligned}\quad (6)$$

onde as constantes de K_5 a K_{12} também representam coeficientes de proporcionalidade e devem ser levantadas experimentalmente.

É importante ressaltar que em nenhum momento se afirma que este modelo esteja completo. Porém, apesar das considerações simplificadoras os resultados que ele apresenta são suficientes para realizar o controle de posição em um voo pairado, como se pode ver na sequência do artigo.

3.2 Modelo de observação dos estados

Para completar o filtro é preciso estabelecer o modelo de observação dos estados. Neste artigo, o método empregado foi estabelecer dois modelos distintos: \mathbf{h}_1 , para ser utilizado quando não existir informação visual disponível, e \mathbf{h}_2 para quando houver, sendo

$$\mathbf{h}_1 = \begin{bmatrix} z \\ v_x \\ v_y \\ \phi \\ \theta \\ \psi \end{bmatrix} \quad e \quad \mathbf{h}_2 = \begin{bmatrix} x_w \\ y_w \\ z \\ v_x \\ v_y \\ \phi \\ \theta \\ \psi \end{bmatrix}, \quad (7)$$

onde z , v_x , v_y , ϕ , θ e ψ são tratadas como observações diretas das variáveis, fornecidas pelo sistema de navegação do AR.Drone, enquanto x_w e y_w representam as coordenadas globais do veículo nestas direções, e são obtidas conforme descrito na Seção 3.3.

Dessa forma, quando há informações visuais disponíveis, é possível utilizá-las para corrigir a incerteza de posição. Note-se, no entanto, que mesmo que isto não aconteça a posição continua a ser estimada, porém ao custo de uma incerteza maior.

3.3 Recuperação de informações visuais

As coordenadas x_w e y_w representam o ponto onde o AR.Drone se encontra posicionado, em relação a um alvo na origem do sistema de coordenadas global. Claro que esta estimativa inclui incertezas e ruídos, mas fornece dados para atualização dos estados no EKF. A Figura 2 ilustra a relação entre os sistemas de coordenadas, assim como mostra o alvo.

No cálculo dos valores de x_w e y_w a técnica aplicada faz uso de algoritmos presentes na biblioteca OpenCV, de acesso público. Como referência em seu uso, foi realizado um estudo em (Bradski and Kaehler, 2008) onde se encontrou a possibilidade de realizar uma filtragem de cor associada a uma busca de centróide. Assim, o que se faz é capturar uma imagem da câmera e buscar nela elementos de coloração verde. Quando detectada uma área com quantidade mínima de pixels desta cor, as coordenadas do centróide são calculadas através do algoritmo momentos invariantes de Hu



Figura 2: Relação entre os sistemas de coordenadas da imagem, do AR.Drone e global.

(Hu, 1962). De posse das coordenadas do centróide, aplica-se a equação

$$\begin{bmatrix} x_w \\ y_w \end{bmatrix} = z \cdot \begin{bmatrix} c_\psi & -s_\psi \\ s_\psi & c_\psi \end{bmatrix} \cdot \left(\begin{bmatrix} 0 & \frac{1}{fs_y} \\ \frac{1}{fs_x} & 0 \end{bmatrix} \cdot \begin{bmatrix} u - c_u \\ v - c_v \end{bmatrix} + \begin{bmatrix} s_\theta \\ -s_\phi \end{bmatrix} \right) \quad (8)$$

de onde se obtém a posição da câmera em relação ao alvo. Em tal equação u e v são as coordenadas do centróide na imagem, enquanto c_u , c_v , fs_x e fs_y são parâmetros intrínsecos da câmera, obtidos a partir de calibração.

4 Controle da posição

Para o controle de posição a técnica aplicada foi um controlador PD, que compensa rotações de ψ . Assim, dada uma referência no espaço tridimensional $\mathbf{P}_d = [x_d \ y_d \ z_d \ \psi_d]$ junto com a saída do filtro, o controle é realizado como

$$\begin{aligned} u_z &= K_{p1}(z_d - z) - K_{d1}\dot{z} \\ u_\psi &= K_{p2}(\psi_d - \psi) - K_{d2}\dot{\psi} \\ u'_\phi &= K_{p3}(y_d - y) - K_{d3}\dot{y} \\ u'_\theta &= K_{p4}(x_d - x) - K_{d4}\dot{x} \\ u_\phi &= u'_\phi s_\psi - u'_\theta c_\psi \\ u_\theta &= u'_\phi c_\psi + u'_\theta s_\psi. \end{aligned}$$

Este controlador PD se mostrou suficiente na regulação das variáveis, alcançando a convergência dos erros a zero. Isto se deve à natureza da resposta do AR.Drone aos sinais de controle. Primeiramente, u_z e u_ψ já representam um comando de velocidade. Portanto, se entende este comportamento como um de natureza integral, que associado ao controlador tende a eliminar os erros em regime. Já os sinais u_ϕ e u_θ , se bem analisados, também o fazem, pois representam indiretamente ganhos de velocidade nos sentidos y_b e x_b , respectivamente. Isto é, u_ϕ provoca uma velocidade $-v_y$

e u_θ provoca uma velocidade $+v_x$. Chama-se a atenção do leitor para o comando u_ϕ , cuja reação é contrária ao ganho de velocidade na direção y_b , por isso a presença do sinal negativo.

5 Resultados experimentais

Todos os resultados apresentam o controle de posição e orientação executado pelo método descrito na Seção 4. O comparativo presente nas imagens foi feito entre a saída de posição e orientação do filtro, $\mathbf{P}_{ekf} = [x_f \ y_f \ z_f \ \psi_f]$, e o resultado obtido através de odometria, definido como $\mathbf{P}_{odm} = [x_o \ y_o \ z \ \psi]$, onde x_o e y_o são obtidos integrando e rotacionando as velocidades v_x e v_y , enquanto z e ψ são os sinais coletados diretamente dos sensores do AR.Drone.

O experimento 1, teve um objetivo simples que era permanecer ancorado sobre o referencial, com $\mathbf{P}_d = [0.0 \ 0.0 \ 1.5 \ 0.0]$. Note-se, pela Figura 3(a), que sem a correção visual fornecida pelo sistema a odometria tende a fornecer valores cada vez mais errados, ao longo do tempo. Isto se deve ao efeito de *drifting*, onde os erros numéricos acumulados na integração dos sinais de velocidade crescem com o tempo.

O experimento 2, ilustrado na Figura 3(b), foi realizado com o mesmo objetivo de controle do experimento 1. A diferença é que são aplicadas perturbações de posição durante o voo na ordem de 0.5m, deslocando o AR.Drone do referencial de voo pairado. É interessante destacar, mesmo sem análise matemática neste sentido, a robustez do sistema, pois o mesmo se recupera após perder o referencial do campo de visão, retomando o controle do voo ancorado.

O experimento 3, da Figura 3(c) é mais completo. Seu objetivo foi alcançar a posição desejada $\mathbf{P}_d = [0.0 \ 0.0 \ 1.2 \ -\pi/4]$. Durante a execução do experimento as luzes do ambiente foram apagadas e acesas. Em seguida o AR.Drone foi submetido a perturbações de posição. Neste experimento é importante destacar a compensação da rotação no controlador PD, foi o que possibilitou a realização do voo em um referencial $\psi_d \neq 0$.

Finalmente, para uma melhor compreensão desta seção, o leitor é convidado a assistir os vídeos que originaram os resultados aqui apresentados, nos links

<http://youtu.be/HQeNkbyDoY8>,
<http://youtu.be/HY0vbKORHhw> e
http://youtu.be/_DwUfg7TTTU.

6 Conclusões

O método aplicado acrescenta robustez ao problema de pairar sobre um objeto, pois não se baseia somente em uma técnica de estimação, seja ela visual ou inercial, mas sim na fusão das duas

modalidades. Tal procedimento permite ao sistema a capacidade de perder momentaneamente o objeto de referência no campo de visão da aeronave, e recuperar a informação posteriormente.

Conforme visto na Seção 5, o método de estimação pode ser usado no controle de posição tridimensional sobre um referencial, mesmo sob orientações ψ_d diferentes de zero. Futuramente, pretende-se aplicá-lo em um sistema de cooperação descentralizado líder-seguidor, onde, utilizando apenas seus próprios sensores, o VANT será capaz de obter os dados de posição relativa a um robô terrestre, por exemplo, bem como estimar a velocidade de tal robô (seu líder), o que é essencial neste tipo de problema.

Seguramente algumas limitações práticas requerem tratamentos que provavelmente conduzirão a melhores resultados no controle. Dentre elas, é importante destacar a necessidade de resolver atrasos de comunicação, sejam eles originados pela rede *Wi-Fi* ou inerentes aos sistema na notória diferença de atualização entre os dados inerciais e visuais no AR.Drone. Independente disso, de uma forma geral o método aqui exposto se mostrou eficaz no cumprimento da tarefa proposta.

Agradecimentos

Os autores agradecem ao CNPq pelo apoio financeiro para a realização desta pesquisa (processo 473185/2012-1). Eles também agradecem à UFES, ao IFES, à UFV e à FAPEMIG, por viabilizarem sua participação neste trabalho. Dr. Sarcinelli Filho também agradece o apoio financeiro adicional da FAPES - Fundação de Amparo à Pesquisa do Espírito Santo ao projeto.

Referências

- Bay, H., Ess, A., Tuytelaars, T. and Gool, L. V. (2008). Speeded-up robust features (surf), *Computer Vision and Image Understanding* **110**(3): 346 – 359.
- Bradski, G. R. and Kaehler, A. (2008). *Learning OpenCV - computer vision with the OpenCV library: software that sees*, O'Reilly Media, Inc., Sebastopol, CA, USA.
- Brandão, A. S. (2013). *Projeto de Controladores Não Lineares para Voo Autônomo de Veículos Aéreos de Pás Rotativas*, Tese de doutorado, Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal do Espírito Santo, Vitória - ES.
- Bristeau, P.-J., Callou, F., Vissiere, D. and Petit, N. (2011). The navigation and control technology inside the ar.drone micro uav, *Proceedings of the 18th IFAC World Congress*, Vol. 18, Milão, Itália, pp. 1477–1484.

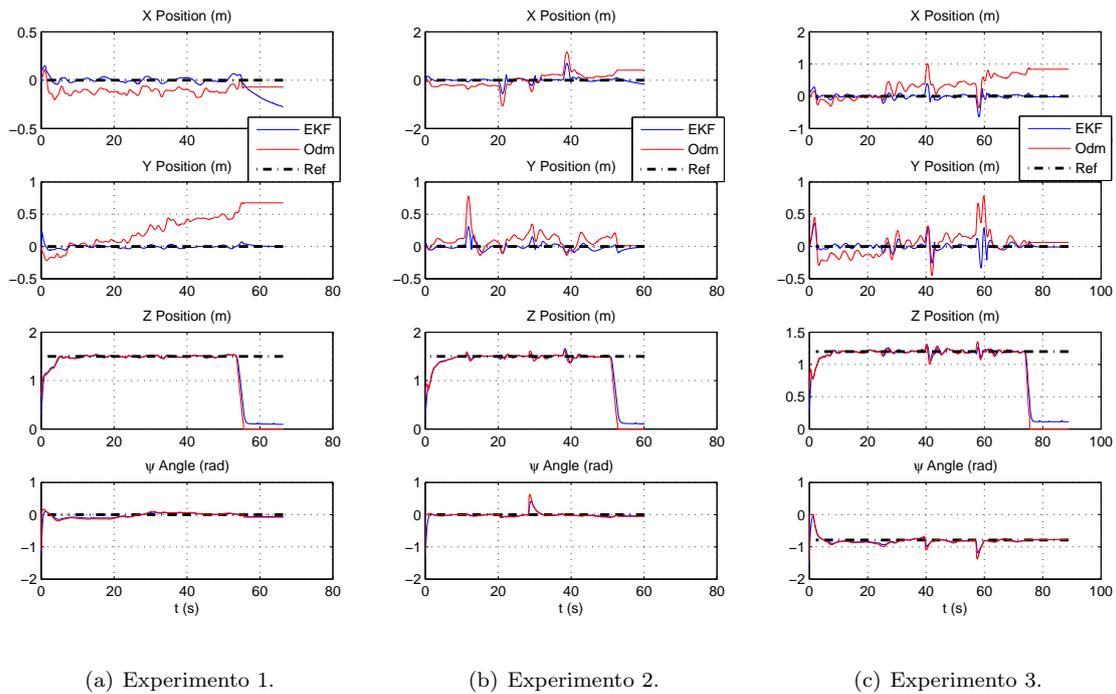


Figura 3: Resultados experimentais.

- Dijkshoorn, N. and Visser, A. (2011). Integrating sensor and motion models to localize an autonomous ar.drone, *International Journal of Micro Air Vehicles* **3**(4): 183–200.
- Engel, J., Sturm, J. and Cremers, D. (2012). Camera-based navigation of a low-cost quadcopter, *Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vilamoura-Algarve, Portugal, pp. 2815–2821.
- Hu, M.-K. (1962). Visual pattern recognition by moment invariants, *IRE Transactions on Information Theory* **8**(2): 179–187.
- Klein, G. and Murray, D. (2009). Parallel tracking and mapping on a camera phone, *Proceedings of the Eighth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'09)*, IEEE Computer Society, Washington, DC, USA, pp. 83–86.
- Krajník, T., Vonasek, V., Fiser, D. and Faigl, J. (2011). Ar-drone as a platform for robotic research and education, in D. Obdrzalek and A. Gottscheber (eds), *Research and Education in Robotics - EUROBOT 2011*, Vol. 161 of *Communications in Computer and Information Science*, Springer, pp. 172–186.
- Mellinger, D. and Kumar, V. (2011). Minimum snap trajectory generation and control for quadrotors, *Proceedings of the 2011 IEEE International Conference on Robotics and Automation*, Xangai, China, pp. 2520–2525.
- Müller, M. W., Lupashin, S. and D’Andrea, R. (2011). Quadcopter ball juggling, *Proceedings of the 24th IEEE/RSJ International Conference on Intelligent Robot Systems*, San Francisco, USA, pp. 5113–5120.
- Piskorski, S., Brulez, N., Eline, P. and DHaeyer, F. (2012). *AR.Drone Developer Guide*, Parrot. SDK Version 2.0.
- Sa, I., He, H., Huynh, V. and Corke, P. (2013). Monocular Vision based Autonomous Navigation for a Cost-Effective Open-Source MAVs in GPS-denied Environments, *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Wollongong, Australia.
- Thrun, S., Burgard, W. and Fox, D. (2005). *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*, The MIT Press.
- Weiss, S., Achtelik, M. W., Chli, M. and Siegwart, R. (2012). Versatile distributed pose estimation and sensor self-calibration for an autonomous mav, *Proceedings of the 2012 IEEE International Conference on Robotics and Automation*, St. Paul, MN, USA, pp. 31–38.
- Weiss, S. M. (2013). *Vision Based Navigation for Micro Helicopters*, Tese de doutorado, Department of Electrical Engineering and Information Technology, Eidgenössische Technische Hochschule (ETH), Zurich, Switzerland.