

# Um algoritmo de navegação não linear 3D para veículos aéreos não tripulados

Guilherme V. Pelizer<sup>1</sup>, Natássya B. F. Silva<sup>1</sup>, Kalinka R. L. J. C. Branco<sup>1</sup>

<sup>1</sup>Instituto de Ciências Matemáticas e de Computação – Universidade de São Paulo (USP)

Caixa Postal 668 – 13.560-970 – São Carlos – SP – Brasil

`guilherme.pelizer@usp.br`, `{ays,kalinka}@icmc.usp.br`

**Abstract.** *Unmanned Aerial Vehicle automatic pilot allows them to work autonomously by keeping the aircraft in predefined flight conditions and by executing navigation tasks, like path following. A non-linear algorithm, named Non-Linear Guidance Law (NLGL), uses a non-linear control law to provide path following for 2D scenarios. However, is important to consider the complete motion in the three dimension space for aircraft. Therefore, this paper proposes a new non-linear guidance algorithm, based on NLGL, to provide path following in the 3D scenario. The simulation results, implemented with the aircraft kinematic motion, show that the new algorithm performs properly when the parameters are correctly set.*

**Resumo.** *O piloto automático dos Veículos Aéreos Não Tripulados (VANTs) é o que permite que essas aeronaves funcionem de modo autônomo, mantendo-as em condições de voo preestabelecidas e executando tarefas de navegação, como o seguimento de trajetórias. Existe atualmente um algoritmo de navegação não linear, chamado Non-Linear Guidance Law (NLGL), que utiliza uma lei de controle não linear para o seguimento de trajetórias, porém para cenários 2D. Mas no caso de aeronaves, é interessante considerar seu movimento completo nas três dimensões espaciais. Logo, nesse artigo é proposto um novo algoritmo de navegação não linear, baseado no NLGL, para o seguimento de trajetórias no cenário 3D. Os resultados das simulações, implementadas com o movimento cinemático de aeronaves, demonstram o bom funcionamento desse algoritmo quando os parâmetros são setados corretamente.*

## 1. Introdução

Veículos Aéreos Não Tripulados (VANTs) são aeronaves que não possuem um piloto humano a bordo e que são capazes de voar sendo pilotadas remotamente ou por meio de um sistema autônomo [1]. VANTs podem ser usados em diversas aplicações, como monitoramento ambiental, de estruturas e de plantações, mapeamento de vias, detecção de incêndios, procura e resgate e rastreamento [Austin 2010], [Zhang e Kovacs 2012].

O funcionamento de um VANT é baseado em um conjunto de hardware e software que permite que o voo e a missão sejam pré-programados antes da decolagem. O sistema autônomo, também conhecido como piloto automático, é o que permite que esses sistemas funcionem de modo autônomo. Esse sistema é responsável por manter as

condições de voo e pela execução de tarefas de navegação, como o seguimento de trajetórias [Valavanis 2007]. Esse conjunto de hardware e software pode ser dividido em duas partes: o sistema de controle e o sistema de navegação.

O piloto automático recebe informações dos sensores da aeronave a respeito de sua posição espacial, velocidade e atitude. Com esses dados, o sistema de navegação gera instruções de qual estado deve ser seguido para que o veículo continue na trajetória programada. Essas instruções são então usadas pelo sistema de controle que envia comandos aos atuadores (superfícies de controle e aceleração do motor) para seguir a trajetória [Elkaim et al. 2015].

Uma das principais tarefas do sistema de navegação é o seguimento de trajetórias. Existem diversos algoritmos que tentam resolver essa tarefa para VANTs, como o *Carrot-Chasing* [Sujit et al. 2014], o *Pure Pursuit and LOS Guidance* (PLOS) [Kothari et al. 2010] e o *Vector Field* (VF) [Nelson et al. 2007]. Porém, esses algoritmos não levam em consideração as não linearidades das equações de movimento das aeronaves.

Uma solução para esse problema é o algoritmo de navegação não linear, ou *Non-linear Guidance Law* (NLGL) [Park et al. 2007], que apresenta uma lei não linear para o seguimento de trajetórias para aeronaves no cenário 2D, utilizando também o conceito de pontos virtuais.

O trabalho de Sujit et al. [Sujit et al. 2014] apresenta uma comparação desse algoritmo com outros algoritmos presentes na literatura para o cenário 2D para VANTs e ressaltam a boa performance do NLGL. Porém, o grande diferencial de aeronaves não tripuladas é a possibilidade de movimentar-se nas três dimensões espaciais, mas para isso é necessário estender os algoritmos de seguimento de trajetórias para o cenário 3D.

O objetivo desse artigo é apresentar o desenvolvimento de um algoritmo de navegação não linear, baseado no NLGL, que permite o seguimento de trajetórias no cenário 3D. Para comprovar o funcionamento do novo algoritmo foram realizadas simulações que consideram o modelo cinemático 3D de uma aeronave, variando-se os parâmetros do algoritmo para também fornecer um entendimento a respeito destes. Também são apresentados os resultados das simulações do NLGL para o cenário 2D para efeito de comparação.

O restante do artigo está organizado da seguinte forma: a Seção 2 descreve o funcionamento do NLGL para o cenário 2D; enquanto que a Seção 3 apresenta o desenvolvimento do novo algoritmo de navegação não linear para o cenário 3D; as Seções 4 e 5, descrevem, respectivamente, os resultados das simulações dos algoritmos para os cenários 2D e 3D; finalmente, a Seção 6 apresenta a conclusão.

## **2. Algoritmo de Navegação Não Linear 2D**

O algoritmo *Non-linear Guidance Law* (NLGL) utiliza o conceito de pontos virtuais, que são pontos inseridos na trajetória para que a aeronave o siga. Com isso, a aeronave se aproxima da trajetória e, eventualmente, acaba seguindo-a corretamente.

O ponto virtual  $\mathbf{p}_{pv}=(x_{pv}, y_{pv})$  é calculado de acordo com a intersecção entre a trajetória retilínea e um círculo de raio  $R$ . A lei de controle para a obtenção da taxa de variação do ângulo de *heading*,  $r_d$ , é dada por (1) [Park et al. 2007].

$$r_d = \frac{2v^2 \text{sen}(\eta)}{R} \quad (1)$$

sendo  $\eta$  o ângulo que direciona a aeronave até a trajetória e  $v$  a velocidade da aeronave.

O algoritmo completo pode ser visto no Algoritmo 1, no qual  $\mathbf{W}_i$  e  $\mathbf{W}_{i+1}$  são, respectivamente, os *waypoints* ultrapassado e próximo que definem a reta da trajetória,  $\mathbf{p}=(x, y)$  a posição da aeronave naquele instante e  $\psi$  o ângulo de *heading*.

**Algoritmo 1: Algoritmo NLGL**

1. Inicializar:  $\mathbf{W}_i=(x_i, y_i)$ ,  $\mathbf{W}_{i+1}=(x_{i+1}, y_{i+1})$ ,  $\mathbf{p}=(x, y)$ ,  $\psi$ ,  $R$ ,  $v$ .
2.  $\alpha = \text{atan2}(y_{i+1} - y_i, x_{i+1} - x_i)$
3. Determinar  $\mathbf{p}_{pv}=(x_{pv}, y_{pv})$ , que é a intersecção entre a circunferência de raio  $R$  e a reta entre os pontos  $\mathbf{W}_i$  e  $\mathbf{W}_{i+1}$ .
4.  $\beta = \text{atan2}(y_{pv} - y, x_{pv} - x)$
5.  $\eta = \psi - \beta$
6.  $r_d = \frac{2v^2 \text{sen}(\eta)}{R}$

### 3. Algoritmo de Navegação Não Linear 3D

O algoritmo de navegação não linear 3D foi criado com base em uma analogia com o algoritmo NLGL apresentado na Seção 2. Essa analogia é realizada após a rotação  $\mathbf{C}(\alpha)$  (2) no plano horizontal da trajetória, sendo  $\alpha$  o ângulo de inclinação da reta que define a trajetória [Breivik e Fossen 2005].

$$\mathbf{C}(\alpha) = \begin{pmatrix} \cos(\alpha) & -\text{sen}(\alpha) & 0 \\ \text{sen}(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (2)$$

Logo, nesse novo algoritmo são calculados dois pontos virtuais, o  $\mathbf{p}_{pv,y}=(x_{pv,y}, y_{pv,y}, 0)$ , calculado na intersecção entre a trajetória retilínea e um círculo de raio  $R_y$  definido no plano horizontal, e o ponto  $\mathbf{p}_{pv,z}=(x_{pv,z}, 0, z_{pv,z})$ , calculado na intersecção entre a trajetória retilínea após a rotação e um círculo de raio  $R_z$  definido no plano vertical. Em seguida são aplicadas as leis de controle dadas por (3) e (4) para se obter a taxa de variação do ângulo *heading* e a taxa de variação do ângulo *pitch*.

$$r_d = \frac{2v^2 \text{sen}(\eta_{\text{heading}})}{R_y} \quad (3)$$

$$q_d = \frac{2v^2 \text{sen}(\eta_{pitch})}{R_z} \quad (4)$$

sendo  $\eta_{heading}$  o ângulo que direciona a aeronave até a trajetória no plano horizontal,  $\eta_{pitch}$  o ângulo que direciona a aeronave até a trajetória no plano vertical, e  $v$  a velocidade da aeronave.

O algoritmo completo é apresentado no Algoritmo 2, no qual  $W_i$  e  $W_{i+1}$  são, respectivamente, os *waypoints* ultrapassado e próximo que definem a reta da trajetória e  $p=(x, y, z)$  a posição da aeronave naquele instante.

#### Algoritmo 2: Algoritmo de navegação não linear 3D

1. Inicializar:  $W_i=(x_i, y_i, z_i)$  ,  $W_{i+1}=(x_{i+1}, y_{i+1}, z_{i+1})$  ,  $p=(x, y, z)$  ,  $\psi$  ,  $\theta$  ,  $R_y$  ,  $R_z$  ,  $v$  .
2.  $\alpha = \text{atan2}(y_{i+1} - y_i, x_{i+1} - x_i)$
3. Determinar  $p_{pv,y}=(x_{pv,y}, y_{pv,y}, 0)$  , que é a intersecção entre a circunferência de raio  $R_y$  e a reta entre os pontos  $(x_i, y_i, 0)$  e  $(x_{i+1}, y_{i+1}, 0)$  .
4.  $W_i^R = C(\alpha)W_i$
5.  $W_{i+1}^R = C(\alpha)W_{i+1}$
6. Determinar  $p_{pv,z}=(x_{pv,z}, 0, z_{pv,z})$  , que é a intersecção entre a circunferência de raio  $R_z$  e a reta entre os pontos  $(x_i^R, 0, z_i^R)$  e  $(x_{i+1}^R, 0, z_{i+1}^R)$  .
7.  $\beta_{heading} = \text{atan2}(y_{pv,y} - y, x_{pv,y} - x)$
8.  $\beta_{pitch} = \text{atan2}(z - z_{pv,z}, x_{pv,z} - x)$
9.  $\eta_{heading} = \psi - \beta_{heading}$
10.  $\eta_{pitch} = \theta - \beta_{pitch}$
11.  $r_d = \frac{2v^2 \text{sen}(\eta_{heading})}{R_y}$
12.  $q_d = \frac{2v^2 \text{sen}(\eta_{pitch})}{R_z}$

## 4. Resultados da simulação 2D

A comprovação do funcionamento do algoritmo NLGL apresentado na Seção 2 foi realizada por meio de simulações no Matlab utilizando a modelagem cinemática 2D do movimento de uma aeronave, representada por (5)-(7) [Sujit et al. 2014].

$$\dot{x} = v \cos(\psi) \quad (5)$$

$$\dot{y} = v \sin(\psi) \quad (6)$$

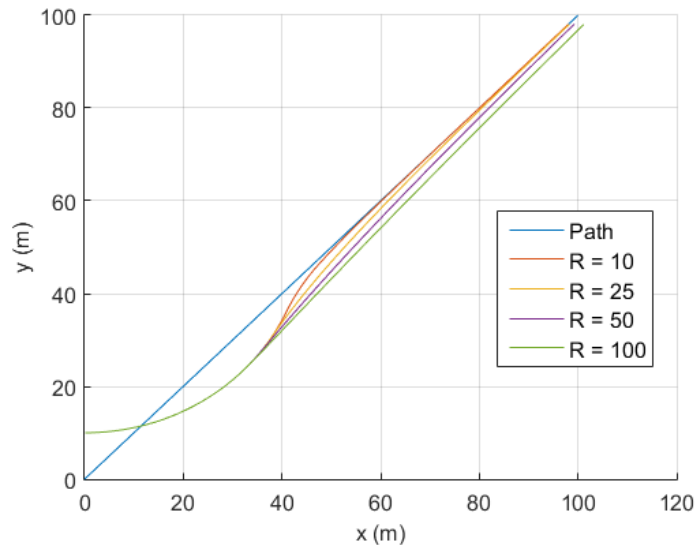
$$r = \dot{\psi} \quad (7)$$

onde a posição da aeronave é dada por  $p=(x,y)$ ,  $v$  é a intensidade da velocidade considerada constante,  $r$  é a taxa de variação do ângulo *heading* e  $\psi$  é o ângulo *heading*.

Como a taxa de variação do ângulo *heading* é fornecida pelo algoritmo de navegação de seguimento de trajetória a cada iteração, é possível calcular  $\psi$  realizando sua integração. Em seguida, calcula-se as velocidades dado o valor do *heading* e obtém-se as posições pela suas integrações.

Durante toda a simulação a intensidade da velocidade  $v$  é mantida constante e igual a 15 m/s. Também é considerada a limitação física da aeronave para realizar curvas com  $|r| < r_{max}$ , sendo  $r_{max} = 0.33$  rad/s [Sujit et al. 2014].

As simulações foram realizadas variando-se o parâmetro  $R$  para observar o seu efeito no seguimento da trajetória, conforme pode ser visto na Figura 1.



**Figura 1. Trajetórias 2D das simulações do algoritmo NLGL variando o parâmetro  $R$ .**

É possível observar que o algoritmo funciona corretamente e que quanto maior o valor de  $R$  mais lentamente a aeronave converge para a trajetória, porém para os valores de  $R$  muito baixos não é possível calcular a intersecção do círculo com a reta que define a trajetória e o algoritmo deixa de funcionar. Nesse caso, o melhor valor para o parâmetro  $R$  é 10 metros, já que é o que mais se aproxima com a trajetória desejada.

## 5. Resultados da simulação 3D

Já a comprovação do novo algoritmo de navegação não linear 3D apresentado na Seção 3 foi realizada utilizando a modelagem cinemática 3D do movimento da aeronave, representada por (8)-(12) [Titterton e Weston 2004], também implementada no Matlab.

$$\dot{x} = v \cos(\psi) \cos(\theta) \quad (8)$$

$$\dot{y} = v \operatorname{sen}(\psi) \cos(\theta) \quad (9)$$

$$\dot{z} = -v \operatorname{sen}(\theta) \quad (10)$$

$$r = \dot{\psi} \quad (11)$$

$$q = \dot{\theta} \quad (12)$$

onde a posição da aeronave é dada por  $p=(x,y,z)$ ,  $v$  é a intensidade da velocidade considerada constante,  $r$  é a taxa de variação do ângulo *heading*,  $q$  é a taxa de variação do ângulo *pitch*,  $\psi$  é o ângulo *heading* e  $\theta$  é o ângulo *pitch*.

Nas simulações para esse modelo também é considerado que as taxas de variação dos ângulos *heading* e *pitch* são fornecidas pelo algoritmo de navegação não linear 3D e a partir das suas integrações são obtidos os ângulos  $\psi$  e  $\theta$ . Em seguida são calculadas as velocidades para cada eixo e a partir das suas integrações são obtidas as posições da aeronave a cada instante.

Durante toda a simulação a intensidade da velocidade  $v$  é mantida constante e igual a 15 m/s. Também é considerada a limitação física da aeronave para realizar curvas tanto em relação ao *heading* quanto ao *pitch*, com  $|r| < r_{max}$  e  $|q| < q_{max}$ , sendo  $r_{max} = 0.33$  rad/s e  $q_{max} = 0.19$  rad/s [Sujit et al. 2014].

As simulações foram realizadas variando-se o parâmetro  $R_y$  e  $R_z$  para observar os seus efeitos no seguimento da trajetória. A Figura 2 mostra as trajetórias resultantes para diferentes valores de  $R_y$  e fixando-se  $R_z = 20$  m e a Figura 3 mostra as trajetórias resultantes para diferentes valores de  $R_z$  e fixando-se  $R_y = 20$  m.

É possível observar que o algoritmo funciona corretamente e que o comportamento da mudança é o mesmo que acontece na análise do NLGL para o cenário 2D. Quanto maior o valor de  $R_y$  ou  $R_z$  mais lentamente a aeronave converge para a trajetória, porém para os valores muito baixos não é possível calcular a intersecção dos círculos com a reta que define a trajetória e o algoritmo deixa de funcionar. Nesse caso, o melhor valor para os parâmetros  $R_y$  e  $R_z$  são 20 metros, já que é o caso em que a trajetória resultante mais se aproxima da trajetória desejada.

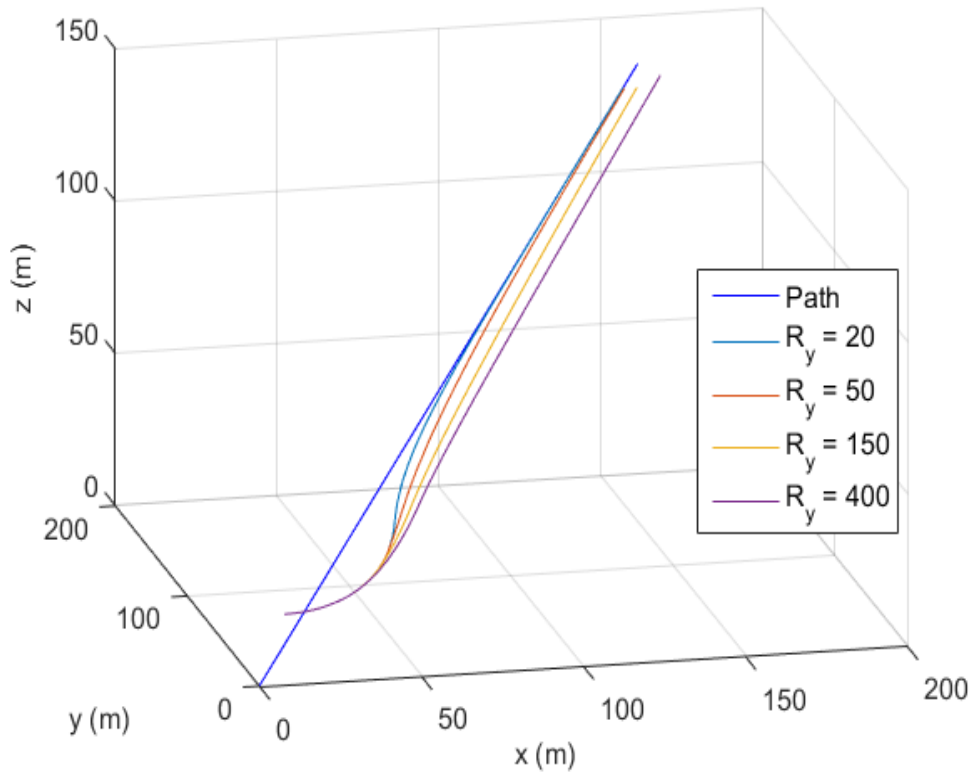


Figura 2. Trajetórias 3D das simulações do algoritmo de navegação não linear 3D variando o parâmetro  $R_y$  e  $R_z=20$  m.

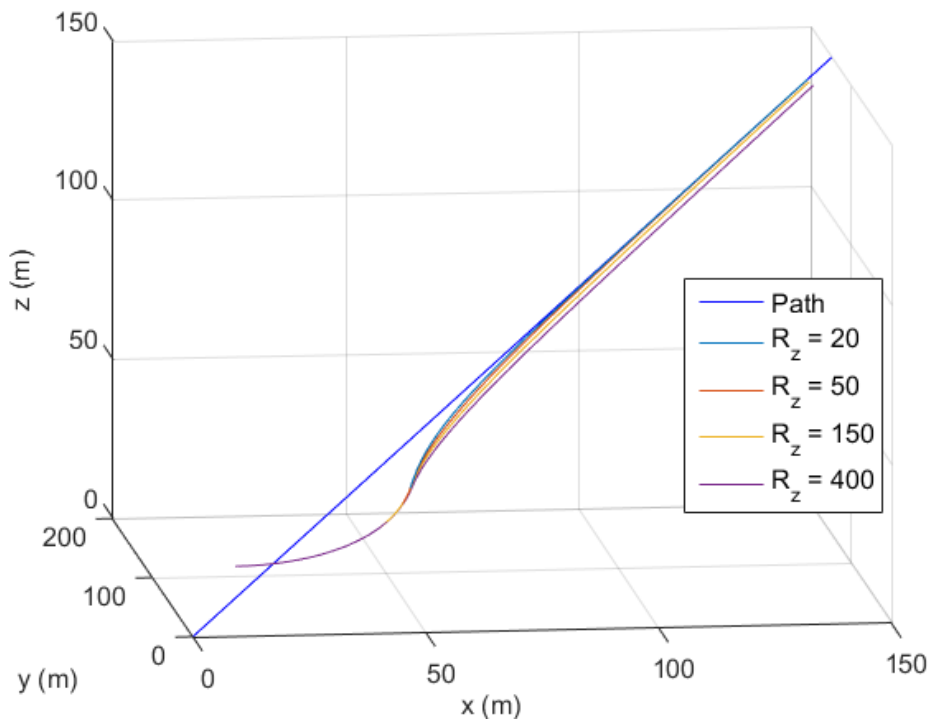


Figura 3. Trajetórias 3D das simulações do algoritmo de navegação não linear 3D variando o parâmetro  $R_z$  e  $R_y=20$  m.

## 6. Conclusão

Foram apresentados os resultados de simulações para dois algoritmos de navegação para o seguimento de trajetórias. O *Non-linear Guidance Law* (NLGL) que utiliza uma lei de controle não linear e permite que uma aeronave siga uma trajetória 2D corretamente e um novo algoritmo de navegação não linear 3D, baseado no primeiro, mas que permite que a aeronave siga uma trajetória 3D. A grande vantagem do uso do algoritmo de navegação não linear 3D é contemplar todas as possibilidades de voo de uma aeronave, que pode se mover em três coordenadas espaciais.

Também foram comparados os comportamentos dos algoritmos para diferentes valores dos seus parâmetros, demonstrando que quanto maior for o raio do círculo usado para determinar-se o ponto virtual, mais lentamente o algoritmo convergirá. Porém, existe também uma limitação inferior para o valor desse raio, já que se for muito pequeno não será possível encontrar uma intersecção com a trajetória. No caso do NLGL o melhor valor, entre os avaliados, para o parâmetro  $R$  é de 10 metros, enquanto que no caso do novo algoritmo de navegação não linear 3D, o melhor valor, entre os avaliados, para ambos os parâmetros  $R_y$  e  $R_z$  é de 20 metros.

Como trabalho futuro pretende-se reproduzir os experimentos em ambientes mais realísticos, como plataformas de *Hardware-in-the-Loop* com o auxílio de um simulador de voos.

## Agradecimentos

Os autores gostariam de agradecer à FAPESP pelo apoio financeiro concedido pelos processos 2012/13641-1 e 2015/21249-2.

## Referências

- Austin, R. (2010) “Unmanned Aircraft Systems: UAVS Design, Development and Deployment” Aerospace Series, Wiley.
- Breivik, M. e Fossen, T. I. (2005) “Principles of Guidance-Based Path Following in 2D and 3D”, In: 44th IEEE Conference on Decision and Control, 2005 European Control Conference. CDC-ECC '05., pp. 627-634. doi: 10.1109/CDC.2005.1582226
- Elkaim, G. H., Lie, F. A. P. e Gebre-Egziabher, D. (2015) “Principles of Guidance, Navigation, and Control of UAVs”, In: Handbook of Unmanned Aerial Vehicles, Editado por Kimon P. Valavanis e George J. Vachtsevanos, Springer Netherlands, p. 347-380. doi: 10.1007/978-90-481-9707-1\_56
- Kothari, M., Postlethwaite, I. e Gu, D. W. (2010) “A suboptimal path planning algorithm using rapidly-exploring random trees”, In: International Journal of Aerospace Innovations, vol. 2, no. 1, pp. 93–104.
- Nelson, D. R., Barber, D. B., McLain, T. W. e Beard, R. W. (2007) “Vector field path following for miniature air vehicles”, In: IEEE Transactions on Robotics, vol. 23, no. 3, pp. 519–529, 2007.



- Park, S., Deyst, J. e How, J. P. (2007) “Performance and Lyapunov Stability of a Nonlinear Path Following Guidance Method”, In: *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 6 , pp. 1718-1728. doi: 10.2514/1.28957
- Sujit, P. B., Saripalli, S. e Sousa, J. B. (2014) “Unmanned Aerial Vehicle Path Following: A Survey and Analysis of Algorithms for Fixed-Wing Unmanned Aerial Vehicless”, In: *IEEE Control Systems*, vol. 34, no. 1, pp. 42-59. doi: 10.1109/MCS.2013.2287568
- Titterton, D. e Weston, J. L. (2004) “Strapdown Inertial Navigation Technology” IET.
- Valavanis, K. P. (2007) “Advances in Unmanned Aerial Vehicles: State of the Art and the Road to Autonomy”, Springer Publishing Company, Incorporated, 2007.
- Zhang, C. e Kovacs, J. M. (2012) “The application of small unmanned aerial systems for precision agriculture: a review” In: *Precision Agriculture*, vol. 13, no. 6, p. 693–712. Springer US. doi: 10.1007/s11119-012-9274-5