

DESENVOLVIMENTO DE UM SIMULADOR DE QUADRICÓPTEROS EM PLATAFORMA MATLAB[®]

Rafael M. J. A. Souza, Gabriela V. Lima, Josué S. Morais, Aniel S. Morais

Laboratório de Automação, Servomecanismos e Controle - Faculdade de Engenharia Elétrica, Universidade Federal de Uberlândia - Caixa Postal 38400-089, Uberlândia - MG, Brasil

E-mails: rafael.mjias@gmail.com, gabriela.vieira.lima@hotmail.com, josue@eletrica.ufu.br, aniel@eletrica.ufu.br

Resumo – Nos últimos anos, é notável a crescente utilização de veículos aéreos não tripulados (VANTs) em diversas aplicações, fato corroborado pelo grande número de artigos científicos publicados na área de modelagem e controle desses veículos. Dentre os inúmeros tipos de VANTs, se destacam os modelos de helicópteros com múltiplos rotores, dentre eles os quadricópteros (4 rotores), hexarotores (6 rotores) e os octarotores (8 rotores). O presente trabalho propõe o desenvolvimento de um ambiente de simulação de um quadricóptero previamente modelado para ser empregado em ensaios de diferentes tipos de controle antes da validação no modelo físico, o que permite maior agilidade nos ensaios e diminui o risco de acidentes com o veículo real. A simulação será desenvolvida em ambiente Matlab[®] e possibilitará a criação de cenário de simulação com diversas interferências ambientais de acordo com a necessidade do usuário.

Palavras-Chave – Quadricóptero, Simulação, Matlab.

DEVELOPMENT OF A QUADROTOR SIMULATOR USING MATLAB

Abstract - In the last years, it's possible to note the increase of the use of unmanned aerial vehicles (UAVs) in various applications, a fact corroborated by the large number of scientific articles published in the field of modeling and control of these vehicles. Among several types of UAVs, the models of helicopters with multiple rotors are between those most used, these can be Quadrotors models (with 4 rotors), Hexarotors models (with 6 rotors) and Octarotors (with 8 rotors). This paper proposes the development of a simulation environment of a previously modeled Quadrotor to be used in testing of different types of control before the validation using the physical model, which allows greater flexibility in testing and reduces the risk of accidents with the actual Quadrotor vehicle. The simulation will be performed using Matlab[®] and will be able to set a simulation scenario with several environmental interferences according to the user need.

Keywords - Quadrotor, Simulation, Matlab.

I. INTRODUÇÃO

A partir da década de 2000, notou-se um aumento repentino no número de estudos publicados na área de robôs aéreos autônomos [6]. Os trabalhos publicados nessa área sugerem desde a viabilidade da utilização de micro robôs [7], até o controle retroalimentado por duas câmeras [8], estudo do *design* e controle em ambientes fechados [9], e utilização de controle multi agente para aeronaves com peso superior a 1 kg [10].

Dentre as subcategorias dos VANTs, os helicópteros com quatro hélices ou quadricópteros merecem destaque por apresentar maior adaptabilidade a diferentes tarefas e maior simplicidade na construção, além de permitirem pouso e decolagem na vertical e não necessitar de controle de inclinação da hélice [4]. Na Figura 1 é possível visualizar a estrutura do quadricóptero.



Fig. 1. Estrutura de um VANT, do tipo quadricóptero.

No entanto para um bom funcionamento deste sistema, faz-se necessário desenvolver estratégias de controle que atuem no controle de posição do veículo e na rejeição de ruídos do ambiente.

Desta forma, este trabalho tem por objetivo desenvolver um ambiente de simulação para testes de estratégias de controle. Com o intuito de evitar danos ao veículo aéreo e agilizar os ensaios de diferentes técnicas de controle, foi proposto o desenvolvimento de um simulador capaz de reproduzir o comportamento de um quadricóptero, em diferentes situações ambientais, envolvendo distúrbios externos como: chuva, vento e obstáculos na trajetória do robô.



XII CEEL – ISSN 2178-8308
13 a 17 de Outubro de 2014
Universidade Federal de Uberlândia – UFU
Uberlândia – Minas Gerais – Brasil

II. DESENVOLVIMENTO

O projeto proposto será implementado na plataforma Matlab®, que será capaz de se comunicar com o ambiente de programação Simulink® e com o editor de ambientes 3D V-Realm Builder.

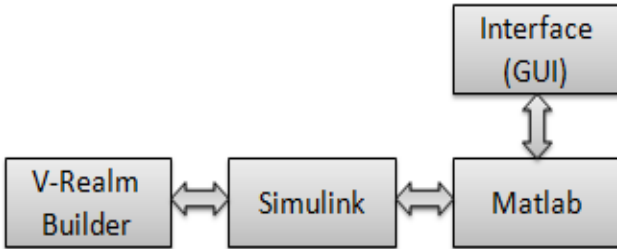


Fig. 2. Diagrama esquemático dos módulos do simulador.

Na Figura 2 é possível visualizar o diagrama esquemático do projeto. A interação entre o usuário e o software se dará através do bloco “Interface (GUI)”, o qual será responsável por receber as variáveis de configuração dos ensaios e exibir posteriormente os valores das variáveis de saída e de estado do sistema. O bloco responsável por receber, processar e enviar as variáveis para a interface é o bloco “Simulink”, no qual a simulação efetivamente será feita. No entanto, como forma de melhor funcionamento do programa essa troca de variáveis será mediada através do bloco “Matlab”. Por fim,

tem-se o bloco “V-Realm Builder” que se responsabilizará simplesmente por, a partir das variáveis de estado processadas pelo Simulink, gerar um ambiente visual tridimensional no qual será possível obter visualmente o comportamento do sistema em questão.

A. Interface (GUI)

A interface gráfica de usuário (GUI) irá interagir com o ambiente de simulação, através da comunicação com a plataforma Matlab®, no qual estarão disponíveis todas as variáveis de projeto oriundas do Simulink® e do V-Realm Builder.

Esta interface foi desenvolvida visando à simplicidade e velocidade na aquisição de dados dos ensaios. Além disso, será possível programar diferentes cenários de testes e visualizar em tempo real o comportamento do veículo aéreo, conforme observado na Figura 3.

A interface desenvolvida possui quatro grandes áreas para melhor visualização das informações. O campo situado à esquerda, nomeado como “Cenários”, contém informações como: duração de ensaio, tipo de cenário e presença de ventos e/ou chuva. Além disso, é possível selecionar a direção e a intensidade dos ventos, bem como o instante em que os distúrbios climáticos irão atuar no quadricóptero. Ainda nesse campo, o botão “Iniciar” na cor verde é responsável pela partida do ensaio após a escolha de todos os parâmetros da simulação.

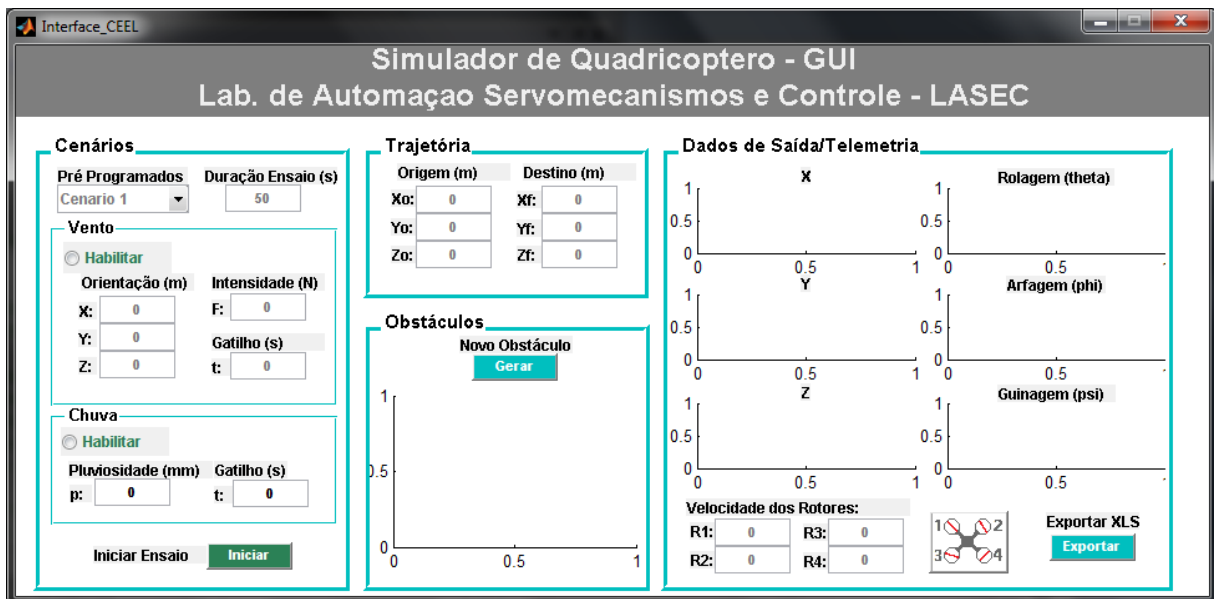


Fig. 3. Interface gráfica do simulador.

No centro da interface, em sua parte superior situa-se a área “Trajetória”, na qual se deve inserir as coordenadas de origem (X_0 , Y_0 , Z_0) e de destino do veículo aéreo (X_f , Y_f , Z_f). Caso o ensaio seja apenas para testar estratégias de controle para rejeição de distúrbios externos faz-se necessário deixar todos os seis parâmetros iguais a 0.

Na parte central e inferior da interface, na guia “Obstáculos”, estão presentes um botão e um gráfico. O gráfico diz respeito à área de manobra do quadricóptero entre os dois pontos escolhidos na guia “Trajetória”. Ao

clique no botão “Gerar”, o usuário poderá desenhar o obstáculo no gráfico de modo a inseri-lo no ambiente de simulação.

Por fim, na parte direita da interface gráfica, a guia “Dados de Saída/Telemetria” contém os campos que indicam os valores das variáveis de saída ao longo do tempo, sendo: Orientação (rolagem θ), arfagem (Φ), guinada (Ψ) e Posição (X , Y , Z). Também é possível visualizar o valor instantâneo da velocidade angular dos quatro rotores do veículo (ω_1 , ω_2 , ω_3 , ω_4). Todos esses

dados poderão ser salvos em uma planilha de Excel ao se clicar no botão “Exportar” na parte inferior desta guia.

B. Matlab®

O bloco “Matlab®” se situa no contexto do projeto como uma plataforma, na qual todas as variáveis de projeto são compartilhadas entre a GUI e o bloco Simulink®.

No ambiente de programação Simulink® não é possível projetar uma interface de usuário para interagir com a simulação. Desta forma, faz-se necessário a criação da interface em Matlab® para que então ocorra a interação entre essas duas plataformas e se estabeleça o fluxo de dados desejado.

No início de cada um dos ensaios, o usuário define os parâmetros de simulação através da interface gráfica, e então estes parâmetros são enviados para o Simulink® através do ambiente Matlab®, da mesma forma que, os dados de saída da simulação, tais como: a posição e orientação do quadricóptero, as velocidades de cada rotor e o tempo do ensaio, são disponibilizadas na GUI através da interface entre o Simulink® e o Matlab®.

C. Simulink®

O bloco Simulink® se posiciona como parte central do projeto. Ele realiza todas as operações matemáticas e modifica as variáveis de projeto, disponibilizando-as para os blocos da GUI e do V-Realm Builder.

É neste bloco que serão implementados o modelo dinâmico do quadricóptero e o modelo dos ruídos ambientais, bem como a estratégia de controle a ser testada.

Na figura 4 é possível observar o diagrama de blocos que exprime todos os subsistemas programados em Simulink®, bem como a relação deles com os outros módulos do simulador.

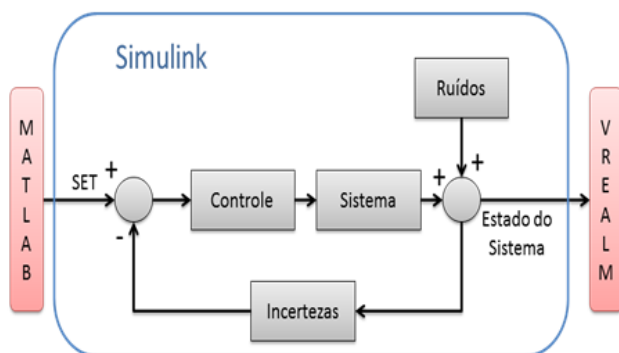


Fig. 4. Subsistemas do módulo “Simulink”.

1) Reprodução do comportamento do quadricóptero

O subsistema mais importante a ser implementado no simulador é a modelagem dinâmica do quadricóptero, uma vez que, a partir desta modelagem será possível estimar o comportamento de um quadricóptero real.

O modelo matemático a ser utilizado no simulador é baseado nas equações físicas de movimento e deriva do formalismo de Euler-Lagrange para sistemas dinâmicos, conforme mostrado nas equações 1 e 2 [1], [2].

$$\begin{cases} \ddot{X} = \frac{1}{m} (\cos \psi \sin \theta \cos \phi + \sin \psi \sin \theta) U + \frac{\delta_{\xi_x}}{m} \\ \ddot{Y} = \frac{1}{m} (\sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi) U + \frac{\delta_{\xi_y}}{m} \\ \ddot{Z} = -g + \frac{1}{m} (\cos \theta \cos \phi) U + \frac{\delta_{\xi_z}}{m} \end{cases} \quad (1)$$

Onde:

\ddot{X} - Aceleração no eixo X.

\ddot{Y} - Aceleração no eixo Y.

\ddot{Z} - Aceleração no eixo Z.

m - Massa do quadricóptero.

ψ - Ângulo de guinada.

θ - Ângulo de rolagem.

ϕ - Ângulo de arfagem.

U - Força de impulsão gerada pelos quatro rotores.

g - Aceleração gravitacional.

δ_{ξ} - Vetor de distúrbios.

$$\ddot{n} = J(n)^{-1} (\tau_n - C(n, \dot{n}) \dot{n}) \quad (2)$$

Onde:

$J(n)$ - Matriz de inércia rotacional.

τ_n - Vetor de momentos de guinada, rolagem e arfagem.

n - Vetor posição dos ângulos de Euler (ψ, θ, ϕ).

\dot{n} - Vetor velocidade dos ângulos de Euler (ψ, θ, ϕ).

\ddot{n} - Vetor aceleração ângulos de Euler (ψ, θ, ϕ).

$C(n, \dot{n})$ - Matriz rotacional de Coriolis e forças centrípetas.

A equação 1 representa a dinâmica de translação da aeronave, que relaciona a variação em cada um dos eixos de coordenadas pertencentes a \mathbb{R}^3 de acordo com os ângulos de Euler e as características físicas do VANT.

Já a segunda equação se encontra na forma matricial compacta e representa o comportamento de rotação da aeronave. Esta equação depende do momento e da inércia em cada eixo de rotação, variáveis que dependem das características físicas do veículo.

A partir destas duas equações é possível obter as seis variáveis de controle do projeto, que correspondem às três coordenadas de posição e as três coordenadas de orientação ($X, Y, Z, \psi, \phi, \theta$) do quadricóptero.

Após o cálculo destas variáveis, o bloco do sistema é capaz de enviá-las para o bloco do V-Realm Builder e simular visualmente a interação da aeronave.

2) Geração de ruídos e mudanças ambientais

Uma das inovações desse trabalho é o acréscimo de ruídos ambientais à simulação do VANT.

A importância deste submódulo diz respeito à possibilidade de testes com diferentes tipos de controladores em um ambiente externo.

Na literatura existem diversos trabalhos que desenvolvem a modelagem e o controle de quadricópteros com restrições quanto à utilização em ambientes com condições desfavoráveis, como rajadas de vento e chuva [4], [5]. Desta forma, a idéia de acrescentar estas perturbações no simulador advém da necessidade de

utilização de veículos não tripulados autônomos em diversas aplicações sujeitas a fenômenos climáticos, como mapeamento de plantações, entrega de encomendas, videomonitoramento entre outras.

Para a geração de rajadas de vento será utilizado o bloco “Discrete Wind Gust Model” do Simulink®, no qual os parâmetros de entrada são as coordenadas de direção ao longo de cada um dos três eixos cartesianos e a intensidade da rajada de vento, todos escolhidos através da GUI desenvolvida.

Já a geração da chuva ocorrerá a partir de estudos a respeito do comportamento dos veículos aéreos sob esta condição climática. Planeja-se obter um modelo matemático capaz de reproduzir os efeitos de chuva no simulador.

Na Figura 5 está mostrado o diagrama de blocos do sistema inteiro, o qual, por comparação com o diagrama presente na Figura 4 é possível situar cada subsistema em um bloco construído no Simulink®.

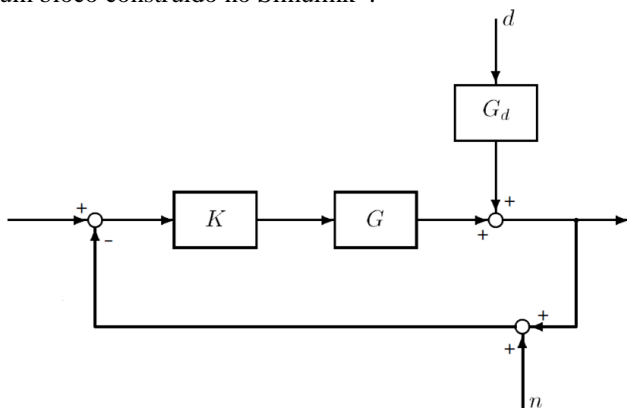


Fig. 5. Diagrama de blocos do sistema.

De acordo com a Figura 5, o bloco “K” corresponde ao bloco de controle que se situa na entrada do bloco do sistema “G”. À frente se tem o bloco dos ruídos (vento e chuva) indicado pela letra “ G_d ”. Esse bloco recebe o sinal de ruído “d” e através de uma modelagem matemática que, a princípio será tomada pronta do bloco “Discrete Wind Gust Model”, realiza a operação de vento e chuva desejados.

3) Gerador de Incertezas

A segunda inovação desse trabalho consiste no acréscimo de um bloco que representa as incertezas das medições e dinâmicas não modeladas.

Como incerteza de medição, considera-se o erro das medidas de dimensões e massa do veículo, assimetrias, além do erro de instrumentação das medidas dos sensores do veículo. Já as incertezas sobre os efeitos não modelados contemplam variação do arraste das hélices com a temperatura, efeito solo, fenômenos climáticos transientes, alteração da carga da bateria do veículo, efeitos da mudança de temperatura, entre outros [1].

Como forma de simplificação, o usuário poderá definir no programa a taxa de incertezas, variando numericamente de 0 a 100% em relação aos parâmetros do veículo.

A importância da inclusão das incertezas no projeto é para que se comprove nos ensaios a robustez do controle utilizado na aeronave, o que implicará em maior confiabilidade no momento de implementação do controle testado no veículo real.

Na Figura 5 é possível notar a localização das incertezas no sistema, a qual é representada pela letra “n” e é acrescida após o sinal receber as perturbações de vento e chuva.

4) Implementação do controle a ser testado

O bloco fundamental da simulação é representado pelo bloco do controle, uma vez que a principal motivação do desenvolvimento deste simulador é a possibilidade de criar um sistema capaz de realizar ensaios para certificar a eficácia e robustez do controle projetado.

Na literatura, são apresentadas diversas estratégias de controle para aeronaves, mas algumas técnicas são mais indicadas para cada situação.

Nos quadricópteros, existem dois aspectos a serem controlados: estabilidade da aeronave mediante os distúrbios externos, e o controle de trajetória.

O controle de estabilidade da aeronave é realizado a partir da aquisição dos dados dos sensores acelerômetros e giroscópios e dos valores de *set point* gerados. O algoritmo mais indicado para fornecer robustez a aeronave é o método H_∞ linear e não linear [1].

O segundo aspecto a ser controlado é a trajetória do VANT. Uma vez definido o ponto de chegada, o veículo deverá ser capaz de traçar uma rota até ele, fazendo os desvios necessários. O controle de trajetória novamente relaciona os dados gerados pelo *set point* e as posições oriundas dos sensores. Os métodos mais indicados para este tipo de controle são: PID, *Backstepping*, Controle Preditivo e H_∞ linear e não linear [1], [4], [5].

D. V-Realm Builder

O bloco “V-Realm Builder” é responsável pelo mecanismo de animação tridimensional do simulador.

Uma vez que o ensaio é configurado pela GUI e a simulação realizada pelos subsistemas via Simulink®, o comportamento visual do quadricóptero é processado pelo V-Realm Builder®.

A interface entre o Simulink e o V-Realm é realizada de forma nativa através do bloco “VR Sinc” presente no Simulink. Por esse bloco é possível editar cada componente em três dimensões no Builder e disponibilizar as variáveis desejadas, como translação e rotação dos corpos, para processamento e modificação nos blocos do Simulink®.

Na Figura 6 é possível visualizar o modelo primitivo do quadricóptero desenhado no V-Realm incluso em um ambiente 3D.



Fig. 6. Modelo tridimensional de um quadricóptero.

Planeja-se desenhar diferentes ambientes com obstáculos, com o intuito de enriquecer a simulação a partir dos testes com o controlador de trajetória, detecção de obstáculos e robustez.

III. CONCLUSÕES

Após a implementação desse simulador, espera-se obter um instrumento útil para testes de diferentes tipos de controle em veículos quadricópteros. O simulador será bastante útil na medida em que se diminuem os riscos de avarias na aeronave e permite a realização de ensaios mais ágeis, objetivos e independente do estado climático do ambiente.

Todas essas vantagens implicam em redução dos custos de cada ensaio e permite a existência de ensaios mesmo em situações ambientais indesejáveis.

A. Trabalhos Posteriores

Como forma de aprimorar a qualidade dos testes na simulação, planeja-se realizar ensaios com um quadricóptero real e obter dados sobre as respostas do veículo dadas diversas situações iniciais. Esse experimento serviria como uma ferramenta para aumentar a confiabilidade do simulador e certificar de que o comportamento do mesmo seja o quanto mais condizente com a realidade possível.

Outra melhora prevista no simulador seria o aumento da customização dos ensaios a partir da possibilidade de se alterar as propriedades do quadricóptero pela GUI. Dentre as características a serem alteradas, planeja-se disponibilizar mudanças na massa, no momento de inércia em cada um dos eixos do veículo, no número de rotores (o que incluiria na simulação a possibilidade de se utilizar hexacópteros e octacópteros), na potência dos motores, nas dimensões das hastes do veículo, entre outras propriedades.

Por fim, após a implementação de todas essas melhorias, pretende-se transcrever o código de Matlab®

para a linguagem C++ visando à possibilidade de patente sobre o simulador.

AGRADECIMENTOS

Os autores deste projeto agradecem a CAPES pelo apoio, bem como ao suporte dos professores e alunos do Laboratório de Automação, Servomecanismos e Controle (LASEC) pela disposição em oferecer ajuda sempre que necessário.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] G. V. Raffo, "Robust Control Strategies for a Quadrotor Helicopter - An Underactuated Mechanical System". Tese de Doutorado, Universidad de Sevilla - Escuela Técnica Superior de Ingeniería, 2011.
- [2] A. S. Brandão, I. H. B. Pizetta, M. Sarcinelli Filho and R. Carelli, "Modelagem e Controle Não Linear Subatuado de um Quad-rotor - Parte 1", XIX Congresso Brasileiro de Automática, 2012.
- [3] A. Rodic, G. Mester, "The Modeling and Simulation of an Autonomous Quad-Rotor Microcopter in Virtual Outdoor Scenario", *Acta Polytechnica Hungarica*, Vol 8, No. 4, 2011.
- [4] R. C. Sá, "Construção, Modelagem Dinâmica e Controle PID para Estabilidade de um Veículo Aéreo Não Tripulado do tipo Quadrirotor", Tese de Mestrado, Universidade Federal do Ceará, Fortaleza, 2012.
- [5] S. Bouabdallah, and R. Siegwart, "Full Control of a Quadrotor", *In Proc. IEEE Int. Conf. on Intelligent Robots and Systems*, San Diego, CA, 2007.
- [6] S. Bouabdallah, "Design and Control of Quadrotors with Application to Autonomous Flying". Tese de Doutorado, D, École Polytechnique Fédérale de Lausanne, 2007.
- [7] G. Fay, "Derivation of the aerodynamic forces for the mesicopter simulation," Stanford University, USA, 2001.
- [8] E. Altunç, "Vision based control of unmanned aerial vehicles with applications to an autonomous four rotor helicopter, quadrotor," Tese de Doutorado, University of Pennsylvania, 2003.
- [9] P. Castillo et al., "Modelling and Control of Mini-Flying Machines". Springer.
- [10] N. Guenard et al., "Control laws for the tele operation of an unmanned aerial vehicle known as an x4-flyer," in Proc. (IEEE) International Conference on Intelligent Robots (IROS'06), Beijing, China, 2006.