The Thirty-Third International
FLAIRS Conference (FLAIRS-33)

Recreating Bat Behavior on Quad-Rotor UAVs—A Simulation Approach

M. Hassan Tanveer,[*] Antony Thomas,[†] Xiaowei Wu,[*],
Rolf Müller,[‡] Pratap Tokekar,[§] Hongxiao Zhu[*]
[*]Department of Statistics, Virginia Tech, USA
[†]DIBRIS, University of Genoa, Italy
[‡]Department of Mechanical Engineering, Virginia Tech, USA
[§]Department of Computer Science, University of Maryland, USA

Abstract

We develop an effective computer model to simulate sensing environments that consist of natural trees. The simulated environments are random and contain full geometry of the tree foliage. While this simulated model can be used as a general platform for studying the sensing mechanism of different flying species, our ultimate goal is to build bat-inspired Quad-rotor UAVs—UAVs that can recreate bat's flying behavior (e.g., obstacle avoidance, path planning) in dense vegetation. To this end, we also introduce an foliage echo simulator that can produce simulated echoes by mimicking bat's biosonar. In our current model, a few realistic model choices or assumptions are made. First, in order to create natural looking trees, the branching structures of trees are modeled by L-systems, whereas the detailed geometry of branches, subbranches and leaves is created by randomizing a reference tree in a CAD object file. Additionally, the foliage echo simulator is simplified so that no shading effect is considered. We demonstrate our developed model by simulating real-world scenarios with multiple trees and compute the corresponding impulse responses along a Quad-rotor trajectory.

1 Introduction

Many environments, such as dense vegetation and narrow caves, are not easily accessible by human beings. Unmanned Aerial Vehicles (UAVs) provide cost-effective alternatives to human beings for a large variety of tasks in such environments, including search, rescue, surveillance, and land inspection. In recent years, impressive progress has been made in UAVs, leading to revolutions in the aerodynamic structure, mechanical transmission, actuator, computer control, etc. Despite these advances, existing technology in UAVs is still limited as most systems can only operate in clear, open space (Dey et al. 2011) or in fields with sparsely distributed tree obstacles (Barry, Florence, and Tedrake), and most existing approaches for localization and planning fail in the presence of large number of obstacles. Moreover, sensors used in these systems are often bulky which hinders efficient navigation (Abdallah 2019). It is highly desirable to build safe and efficient UAV systems that do not fail under different real-world conditions.

Among many directions in technological innovation, bio-inspired technology provides a promising solution that may break the performance boundary in UAVs. Mammals, insects and other organisms often exhibit advanced capabilities and features that would be desirable for UAVs. They can rapidly pick out salient features buried in large amounts of data, and adapt themselves to the dynamics of their environments. Adopting prototypes that emulate the characteristics and functions found in living creatures may enable robots to maneuver more efficiently without the aid of approaches such as simultaneous mapping and localization (SLAM), GPS or inertial units. In recent years, bio-inspired approaches have already given rise to robots that operate in water (Yao, Song, and Jiang 2011), air (Duan and Qiao 2014) and on land (Zhou and Bi 2012) and, in some cases, transiting in various media. For UAVs in particular, "Microbot" has been created in 2002 by The California Institute of Technology (Bogue 2015), which achieves independent fly by imitating the morphological properties of versatile bat wings. In 2011, AeroVironment successfully developed the "Hummingbird" by mimicking hummingbirds (Coleman et al. 2015). The Hummingbird is trained and equipped to continue flying itself with its own supply of energy. The flapping wings can effectively control its attitude angles. Besides these examples, there are several other conventional designs developed, including Robird (Folkertsma et al. 2017), DelFly (De Croon et al. 2016), and Bat Bot (Ramezani et al. 2015).

In this research, we consider using the echolocation system of bats as a biological model for the study of highly parsimonious biosonar sensors for UAVs. Millions of years' biological development provides bats numerous incredible skills to navigate freely in complex, unstructured environments. Relying on miniature sonar systems with a few transducers—a nose (or mouth) and two ears, bats achieve much better navigation performance than engineered systems. Specifically, a echolocating bat emits brief ultrasonic pulses through mouth or nostrils, and use the returning echoes to navigate (Griffin 1958). Based on bats' biosonar, we aim to develop a bat-inspired sonar sensing and navigation paradigm for quad-rotor UAVs. To achieve this, we adopt a data-driven approach that integrates large-scale simulations with statistical learning to gain insights and replicate bats' abilities.

Results presented in this paper are based on our initial efforts in recreating the sensory world of bats via computer simulation. We develop an effective computer model to simulate

Copyright © 2020, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

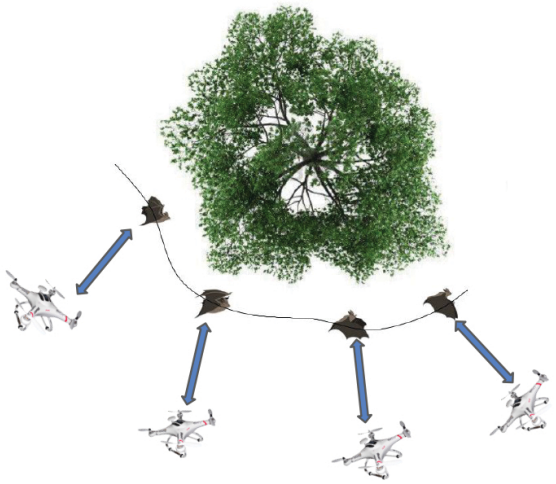


Figure 1: A bat navigating around a tree. We mimic the highly developed bio-sonar system in bats by simulating sonar and leaf beampatterns and validate it through different experiments. A Quad-rotor using this sonar is visualized.

sensing environments that consist of natural looking trees. The simulated environments are random and contain full geometry of the tree foliage. While this model can be used as a general platform for studying the sensing mechanism of different flying species, our ultimate goal is to build bat-inspired Quad-rotor UAVs—UAVs that can recreate bat's flying behavior (e.g., obstacle avoidance, path planning) in dense vegetation. To this end, we also introduce an foliage echo simulator that can produce simulated echoes by mimicking bat's biosonar. In Figure 1, we demonstrate how a bat is mimicked by a Quad-rotor while navigating across a tree. In our current model, a few realistic model choices or assumptions are made. First, in order to create natural looking trees, the branching structures of trees are modeled by L-systems, whereas the detailed geometry of branches, sub-branches and leaves is created by randomizing a reference tree in a CAD object file. Additionally, the foliage echo simulator is simplified so that no shading effect is considered. We demonstrate our developed model by simulating real-world scenarios with multiple trees and compute the corresponding impulse responses along a Quad-rotor trajectory.
The rest of this paper is organized as follows. In Section 2, we describe the method of simulating a sensing environment with multiple natural looking trees and the theory behind the foliage echo simulator. We elaborate experimental results and analyses in Section 3. Finally, in Section 4, a general conclusion and the direction towards future work are given.

2 Material and Methods

We develop a computational framework that consists of two simulators, one for the simulation of sensing environment which produces random trees with necessary geometry (e.g., leave locations, size and orientations etc.), another for the simulation of foliage echoes which produces sonar impulses by mimicking the biosonar system of bats. In this Section, we elucidate the main methodology used in these simulators.

We simulate the topology of each individual tree by combining Lindenmayer systems (L-systems) with modified CAD implemented object files. An L-system is a graphical model commonly used to describe the growth pattern of plants (Prusinkiewicz and Lindenmayer 1996). It defines the branching pattern of a plant through recursively applying certain production rules on a string of symbols. Each symbol in the string defines a structural component (e.g., branch, terminal). Each recursive iteration creates an additional level of growth of the string. The final string represents the branching structure of the grown tree. While L-system is commonly used to produce branching structures (Shlyakhter et al. 2001), we found that it is not sufficient for generating natural looking trees because of over-simplified assumptions. For example, tree models based on L-systems often model branches as straight lines while ignoring the natural curvatures of the branches. Furthermore, most of the L-system models often rely on a few parameters to control the lengths, thickness and angles of branches. Although probability distributions can be introduced to randomize these parameters, they are often not enough to characterize all features of a particular tree species. For these reasons, we choose to adopt L-system to generate the first level branching locations at the trunk. To generate the branches and sub-branches, we modify reference trees from CAD developed object files by randomizing the branch curvatures, lengths, and sub-branch locations. This produces random trees that look more realistic. In Figure 2, we demonstrate the plot of a tree simulated by L-system with the first level branching structure only.
Information about the branches, sub-branches and leaves of the simulated trees is stored as an organizational structure of building systems. This is associated with 3D CAD drawing that includes faces and vertices modelled as meshes. This provides a complete 3D tree with planarity for each branch. The planarity makes it easy to visualize the tree with short computing time based on available data (i.e: Polygon, vertices, textures etc), thereby offering a convenient way to effectively imagine scenarios with other trees in forests. The L-system does not really follow drawing standards (i.e: with the geometric information). Hence, in order to make branches and sub-branches, we should follow certain rules using 3D CAD tools that abides by the tree geometry (see Figure 3).
Based on the simulator of a random tree, we are able to generate a community that consists of random number of trees. We determine the number of trees and the locations of these trees in a 2-D region by sampling from Inhomogeneous Poisson process (IPP). Let $D \subset \mathbb{R}^2$ denote the 2-D region on which the community of trees will be built. The random locations (i.e., (x, y) coordinates) of the trees are denoted by $S = \{s_i\}_{1 \leq i \leq n}$. We assume that S follows an IPP with intensity function $\lambda(s) : D \rightarrow \mathbb{R}^+$, where $\lambda(s)$ is a parameter to be specified by user which describes how dense the trees are at every location. Small values of $\lambda(s)$ indicate sparse regions whereas high values indicate dense regions. The number of trees, n, follows a Poisson

distribution $\int_D \lambda(s)ds$. To simulate S given n, we adopt a thinning approach (Lewis and Shedler 1979).

For the simulation of foliage echoes, we follow the approach of (Ming et al. 2017). Here, we briefly summarize the method. In the current model, the leaves are simplified as circular disks. The simulated foliage echoes are stored as time-domain (discrete) signals. Let $Y = \{y_1, \ldots, y_n\}$ denote one time-domain signal to be simulated. Let $Y^* = \{y_1^*, \ldots, y_{n'}^*\}$ denote the Fourier transform of Y in the frequency domain. To obtain Y, we first compute Y^* and apply inverse fast Fourier transform. It is assumed that y_k^* is nonzero in the frequency ranges between 60 to 80 kHz, which corresponds to the strongest harmonic in the biosonar impulses of the *Rhinolophus ferrumequinum bat* (Andrews and Andrews 2003). According to acoustic laws of sound reflection (Bowman, Senior, and Uslenghi 1987a), each Fourier component y_k^* is the superposition of all the reflecting echoes from the reflecting facets within the main lobe of the sonar. It takes the form

$$y_k^* = \sum_{i=1}^{m} A_{ki}\cos(\phi_{ki}) + j\sum_{i=1}^{m} A_{ki}\sin(\phi_{ki}), \qquad (1)$$

where m denotes the number of reflecting facets within the main lobe of the sonar, A_{ki} is the amplitude at frequency f_k (which is the frequency corresponding to y_k^*) for the i-th facet, ϕ_{ki} is a phase delay parameter at f_k for the i-th facet. The term A_{ki} can be computed by

$$A_{ki} = S(az_i, el_i, f_k, r_i)L_i(\beta_i, a_i, f_k)\frac{\lambda_k}{2\pi r_i^2}, \qquad (2)$$

where $S(az_i, el_i, f_k, r_i)$ represents the sonar beampattern with az_i and el_i being the azimuth and elevation angles of the line that connects the sonar with the i-th reflecting facet, r_i is the distance between the sonar and the i-th reflecting facet, $L_i(\beta_i, a_i, f_k)$ is the beampattern of the reflecting facet with β_i and a_i being the incident angle and of the i-th reflecting facet respectively. The sonar beampattern has the general form

$$S(\cdot) = A_1\exp\{-(a(x-x_0)^2 + 2b(x-x_0)(y-y_0) + c(y-y_0)^2 t)\} \qquad (3)$$

where A_1 is the amplitude, a, b, c are the parameters of Gaussian function. The value of a, b, c are determined by empirical data. The leaf beampattern can be approximated by cosine function of the form

$$L_i(\cdot) = A\left(c\left(f_k, a_i\right)\cdot\cos\left(Bc\left(f_k, a_i\right)\cdot\beta_i\right)\right) \qquad (4)$$

where $c = 2\pi a_i f_k/v$, with v being the speed of sound and A, B are functions of c. A detailed description of (3) and (4) is beyond the scope of this paper and we refer the interested readers to (Bowman, Senior, and Uslenghi 1987b; Adelman, Gumerov, and Duraiswami 2014).

The simulation of sensing environments and foliage echoes provide us rich amount of sensing data under various sensing tasks. Within the simulated setup, we can design a Quad-rotor that mimics the bat behaviors for tasks such as insects prey. The dynamics of Quad-rotor UAV can be taken from, e.g., (Bouabdallah 2007). We employ the control, that has been recently developed to stabilize UAV while flying (Tanveer, Recchiuto, and Sgorbissa 2019).

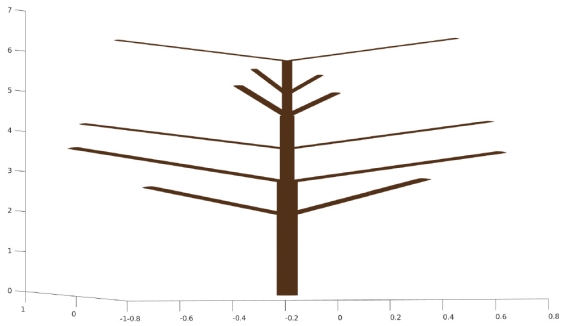


Figure 2: L-system for branch generation on the trunk.

3 Results and Discussion

We performed a pilot study by designing a simple sensing scene that involves multiple trees. These trees are constructed by combing an L-system with CAD developed object files as described in the previous section. When visualizing the trees, leaves were approximated using the mid points of the triangular meshes used to model leaves in CAD.

We conduct several simulations in the MATLAB environment to demonstrate the performance of model. The performance are evaluated on an Intel® Core™ i7-3632QM under Ubuntu 16.04 LTS. The simulation with multiple trees has been done on a ten-core server computer. Tree locations in the environment are determined by sampling from an IPP model. The trees are different from each other in terms of of branches angles, sizes and leaves distribution. Moreover, the initial branching pattern follows that of an L-system. In each simulation, we construct a tree (or trees) and analyze the impulse responses from simulated sonar echoes. The impulse responses are computed at different sonar locations in the environment to mimic a flying Quad-rotor. For example, we have computed impulses at regular intervals along a circular path around a tree and impulses for a trajectory directly towards the tree. The beam width of the sonar main lobe is chosen to be 10, 20, and 50 degrees.

Figure 4 presents a tree when the Quad-rotor navigates through the tree and observes no leaf in the sensor main lobe, hence no output is generated. Figure 5 demonstrates the situation when the sonar encounters leaves and branches, which results in impulse as shown in Figure 6. Figure 7 presents two trees when the Quad-rotor navigates through the tree in a circular path. It encounters leaves and branches at four instances. The impulse responses of the four sample points are shown in Figure 8. In addition, Figure 10 shows two and three trees with multiple sonar sample points.

To analyze the computational complexity, we compute the total computation time while the Quad-rotor completes an entire trajectory. Under one setup, we increase the number of locations along the trajectory where the impulse responses

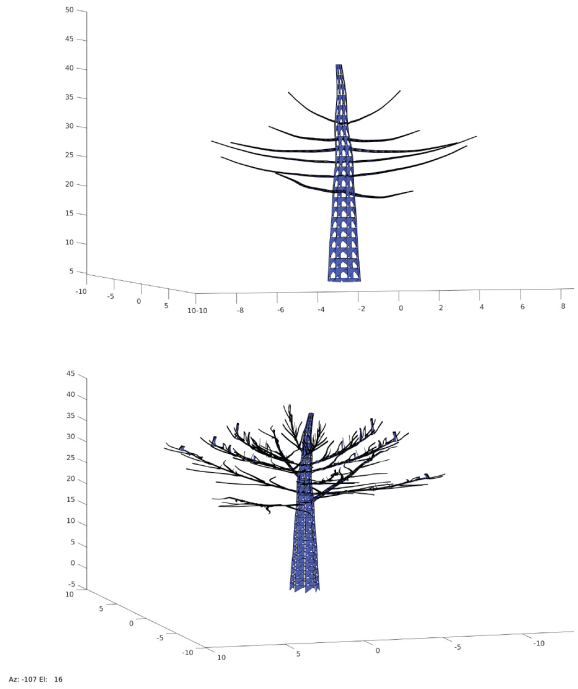


Figure 3: L-system and object file fusion for generating tree branches and sub branches

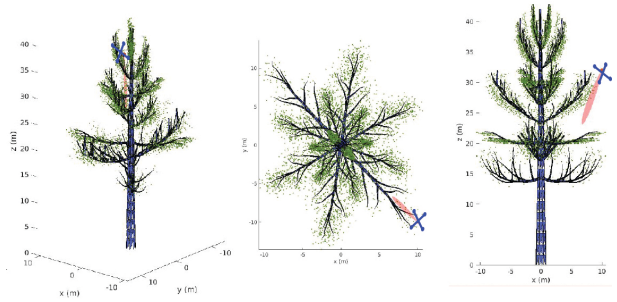


Figure 4: Quad-rotor navigating with no leaves encountering in main lobe of sensor

are computed. Under another setup, we increase the number of trees (T) in the environment. The computation time for different scenarios are shown in Table 1. We observe that, increasing the number of computation points has a direct effect on the computation time, which is quite intuitive. It is interesting to note that for a circular trajectory around one tree (T = 1) with radius of 6.2, it takes only around one second on average to compute 15 impulse responses at regular intervals of 24 degrees. For more than one tree, we set the centre of the circular trajectory to be the mean position of the tree locations. We observe that, increasing the number of trees varies the leaf densities and hence has a direct effect on the computation time. For T = 5, it takes only about 3 seconds on average to compute impulses at 15 points along the trajectory. Overall, our model performs fairly well in real-time.

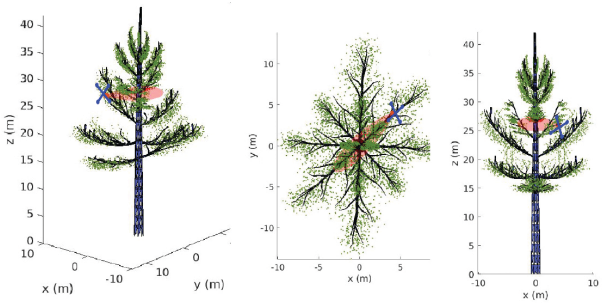


Figure 5: Quad-rotor navigating with leaves encountering in main lobe of sensor

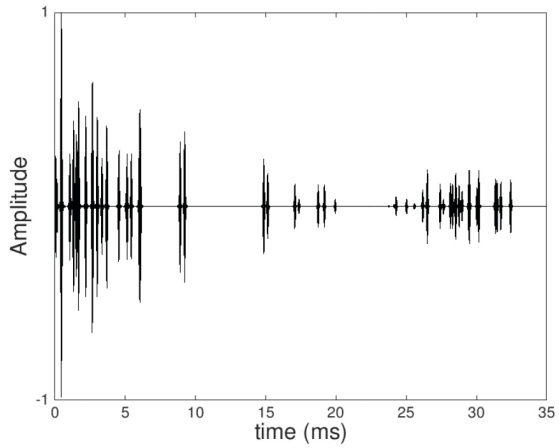


Figure 6: Impulse response of main lobe

4 Conclusion

In this research we explore how to recreate bat behaviors on Quad-rotor UAVs in dynamic environments, thereby transforming nature into bio-technology. In particular, we propose a computational approach to simulate the sensing environments and to simulate foliage echoes during different sensing scenarios.

In this preliminary study, we mainly focused on model development and experimental validation in a simulated/known environmental setting. The impulse responses can be further analyzed using state-of-the-art artificial intelligence and machine learning methods to predict different parameters like leaf density, orientation, density. This is a promising direction since it enables navigating in unknown environments. Currently, the trajectories followed by the sonar are predefined and we only analyze the impulse generated at different time instances along the trajectory. Immediate next step is to extend it to an active navigation scenario in which an optimal path can be calculated. Another interesting future direction is to extend the framework towards task and motion planning in large knowledge-intensive domains, as recently done in (Lo, Zhang, and Stone 2018; Thomas, Mastrogiovanni, and Baglietto 2019). We also plan to model the shading effect between leaves, for example by using an adjusted attenuation function. In order to deal with

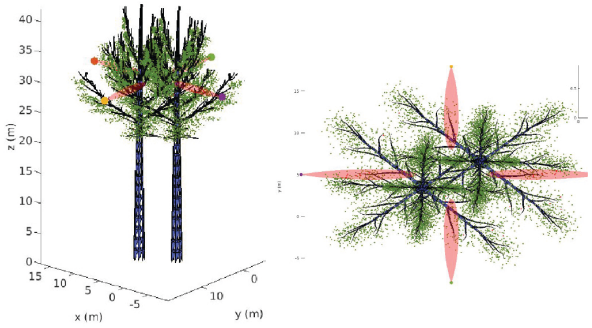


Figure 7: Quad-rotor navigating across 2 trees with leaves encountering in main lobe of sensor

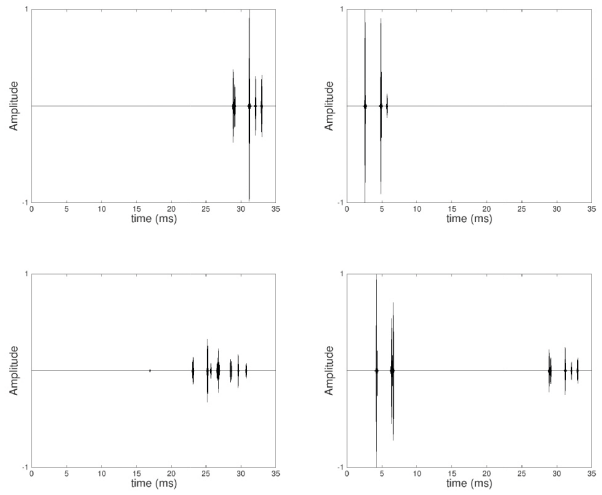


Figure 8: Impulse response of main lobe at 4 different locations along a circular trajectory.

real world uncertainties, we also plan to integrate our model with Inverse perspective mapping (IPM) approach. This can be done by mounting a camera on the UAV in order to obtain a bird's eye view (Tanveer, Sgorbissa, and Thomas 2020).

5 Acknowledgments

This research is supported in part by National Science Foundation (NSF) grant #1762577.

References

Abdallah, R. 2019. *Reliability approaches in networked systems: Application on Unmanned Aerial Vehicles*. Ph.D. Dissertation.

Adelman, R.; Gumerov, N. A.; and Duraiswami, R. 2014. Software for computing the spheroidal wave functions using arbitrary precision arithmetic. *arXiv preprint arXiv:1408.0074*.

Andrews, M. M., and Andrews, P. T. 2003. Ultrasound social calls made by greater horseshoe bats (rhinolophus ferrumequinum) in a nursery roost. *Acta Chiropterologica* 5(2):221–235.

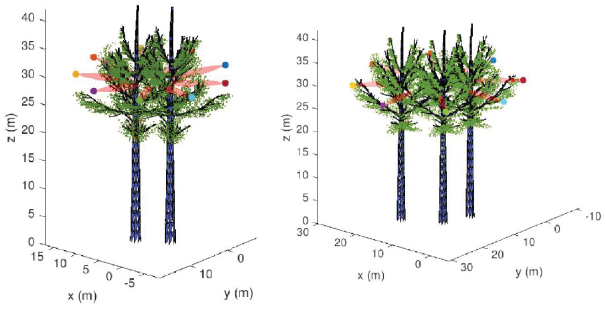


Figure 9: Quad-rotor navigating across different number of trees with leaves encountering in main lobe of sensor

Computation points	Impulse computation time (s)				
	$\mathtt{T}=1$	$\mathtt{T}=2$	$\mathtt{T}=3$	$\mathtt{T}=4$	$\mathtt{T}=5$
1	0.73	0.77	0.86	0.93	1.07
5	0.88	1.00	1.19	1.33	1.52
10	0.97	1.21	1.42	1.72	2.11
15	1.03	1.68	1.92	2.54	3.03

Table 1: Total impulse computation for an entire circular trajectory when the impulses are computed at $1, 5, 10$ and 15 locations along the trajectory. $\mathtt{T} = \{1,\ldots,5\}$ represent different number of trees in the environment. The center of the circular trajectory is the mean location of the trees in the environment.

Barry, A. J.; Florence, P. R.; and Tedrake, R. High-speed autonomous obstacle avoidance with pushbroom stereo. *J. Field. Robot.* 1–17.

Bogue, R. 2015. Miniature and microrobots: a review of recent developments. *Industrial Robot: An International Journal* 42(2):98–102.

Bouabdallah, S. 2007. Design and control of quadrotors with application to autonomous flying. Technical report, Epfl.

Bowman, J.; Senior, T.; and Uslenghi, P. 1987a. *Electromagnetic and acoustic scattering by simple shapes*. New-York: Hemisphere Publishing Corporation.

Bowman, J. J.; Senior, T. B.; and Uslenghi, P. L. 1987b. Electromagnetic and acoustic scattering by simple shapes (revised edition). *New York, Hemisphere Publishing Corp., 1987, 747 p. No individual items are abstracted in this volume.*

Coleman, D.; Benedict, M.; Hrishikeshavan, V.; and Chopra, I. 2015. Design, development and flight-testing of a robotic hummingbird. In *AHS 71st annual forum*, 5–7.

De Croon, G.; Perçin, M.; Remes, B.; Ruijsink, R.; and De Wagter, C. 2016. *The DelFly*. Springer.

Dey, D.; Geyer, C. M.; Singh, S.; and Digioia, M. 2011. A cascaded method to detect aircraft in video imagery. 30(12):1527–1540.

Duan, H., and Qiao, P. 2014. Pigeon-inspired optimization: a new swarm intelligence optimizer for air robot path plan-

ning. *International Journal of Intelligent Computing and Cybernetics* 7(1):24–37.

Folkertsma, G. A.; Straatman, W.; Nijenhuis, N.; Venner, C. H.; and Stramigioli, S. 2017. Robird: a robotic bird of prey. *IEEE robotics & automation magazine* 24(3):22–29.

Griffin, D. R. 1958. *Listening in the Dark*. Yale Univ. Press.

Lewis, P. A. W., and Shedler, G. S. 1979. Simulation of nonhomogeneous poisson processes with degree-two exponential polynomial rate function. *Oper. Res.* 27(5):1026–1040.

Lo, S.-Y.; Zhang, S.; and Stone, P. 2018. Petlon: planning efficiently for task-level-optimal navigation. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, 220–228. International Foundation for Autonomous Agents and Multiagent Systems.

Ming, C.; Gupta, A. K.; Lu, R.; Zhu, H.; and Müller, R. 2017. A computational model for biosonar echoes from foliage. *PLOS ONE* 12(8):1–18.

Prusinkiewicz, P., and Lindenmayer, A. 1996. *The Algorithmic Beauty of Plants*. New York, NY, USA: Springer-Verlag New York, Inc.

Ramezani, A.; Shi, X.; Chung, S.-J.; and Hutchinson, S. 2015. Bat bot (b2), an articulated-winged bat robot. In *poster presentation. International Conference of Robotics and Automation, May 26th-30th, Seattle, WA*. Citeseer.

Shlyakhter, I.; Rozenoer, M.; Dorsey, J.; and Teller, S. 2001. Reconstructing 3d tree models from instrumented photographs. *IEEE Computer Graphics and Applications* 21(3):53–61.

Tanveer, M. H.; Recchiuto, C. T.; and Sgorbissa, A. 2019. Analysis of path following and obstacle avoidance for multiple wheeled robots in a shared workspace. *Robotica* 37(1):80–108.

Tanveer, M. H.; Sgorbissa, A.; and Thomas, A. 2020. An ipm approach to multi-robot cooperative localization: Pepper humanoid and wheeled robots in a shared space. In Gusikhin, O., and Madani, K., eds., *Informatics in Control, Automation and Robotics*, 429–447. Cham: Springer International Publishing.

Thomas, A.; Mastrogiovanni, F.; and Baglietto, M. 2019. Task-motion planning for navigation in belief space. *arXiv preprint arXiv:1910.11683*.

Yao, X.; Song, Y.; and Jiang, L. 2011. Applications of bio-inspired special wettable surfaces. *Advanced Materials* 23(6):719–734.

Zhou, X., and Bi, S. 2012. A survey of bio-inspired compliant legged robot designs. *Bioinspiration & biomimetics* 7(4):041001.