

Bio-inspired source seeking and obstacle avoidance on a palm-sized drone

Nishant Elkunchwar¹ (nishante@uw.edu), Vikram Iyer², Melanie Anderson¹, Krishna Balasubramanian¹, Jessica Noe¹, Yash Talwekar¹, and Sawyer Fuller¹

Abstract—The small form factor of palm sized unmanned aerial vehicles (UAVs) combined with their ability to freely maneuver in 3D space with holonomic trajectories and carry custom sensors makes them an ideal platform for autonomous source seeking in challenging environments. Equipped with the appropriate sensor, a small UAV could autonomously navigate towards light or heat sources such as forest fires or locate a radio-frequency (RF) transmitter attached to anything from a package in a warehouse to an animal tagged with a radio tracker. Leveraging small UAVs for this task however requires addressing their size weight, power, and computational constraints. While prior source seeking robots have used search strategies that require extensive training, such as reinforcement learning, we instead look to biology and employ a simple ‘run and tumble’ gradient following algorithm inspired by bacterial chemotaxis. The result is a computationally inexpensive approach requiring as little as 30 instructions/second, allowing this strategy to scale down to millimeter scale robots with small microcontrollers. Using insights from simulation, we report a success rate of 91% in real-time demonstrations of our UAV navigating towards a fire or light source while avoiding obstacles. Measurements from a small Bluetooth transmitter indicate it also produces a compatible gradient at ranges of 50–100 m. We conclude by discussing how this technique could scale down to sub-cm microrobots seeking RF power sources.

Index Terms—Biologically-inspired robots, Aerial Systems: Applications, micro/nano robots

I. INTRODUCTION

Unmanned Aerial Vehicles (UAVs) are versatile, with applications ranging from rescue operations [1] to toxic algal bloom detection [2]. In addition to surveying an area by systematically traversing it, UAVs augmented with sensing capabilities could intelligently and autonomously navigate towards points of interest to locate a signal source. Consider the concrete problem of using a UAV to find a forest fire: fires, especially large ones, produce significant amounts of heat and light, therefore creating a gradient of these quantities that can be measured at a distance. A drone with a temperature or light sensor could follow this gradient to autonomously locate the fire. Similarly, radiofrequency (RF) emissions can be detected at long distances. Recent advancements in miniaturization

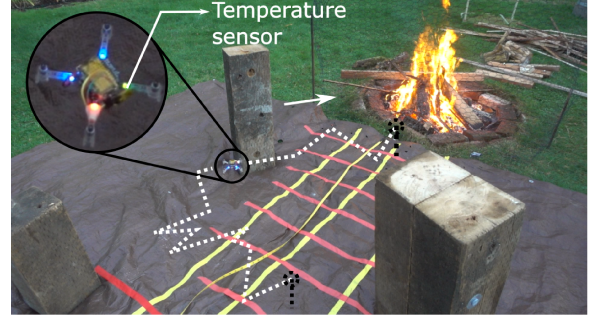


Fig. 1: Our palm-sized drone uses a biology-inspired algorithm while avoiding obstacles to reach a source such as a fire, RF emission, or light source.

of radio tags [3] present the potential to turn any object into a signal source that could be measured at a distance. This includes locating trackers attached to keys, packages in warehouses, avalanche beacons [4], or even radio tracking of invasive species like giant hornets in a forest [5], [6].

Realizing these applications, however, requires addressing a number of practical and technical challenges. First, many of these application scenarios such as those inside a building or in a forest require operating in a cluttered airspace full of obstacles. This becomes difficult for large UAVs as their size prevents them from maneuvering in these environments. Recently developed palm-sized drones [7] are ideal for these scenarios as they are small enough to navigate around obstacles, or fly over them when needed by leveraging their ability to maneuver in 3D space which overcomes a significant limitation of ground-based vehicles.

Scaling down to smaller drones, however, introduces a number of technical challenges as they pose significant size, weight, and power (SWaP) constraints. In addition to limits on the sensors, and flight time, the SWaP constraints require many of these platforms to use small microcontrollers with limited computational capabilities. Therefore, it is imperative to find a computationally efficient and robust means to search for sources and avoid obstacles in complex environments.

If we look to nature however, we observe that even the simplest organisms can perform source seeking behaviors. For example light source attraction through positive phototaxis is a well-known phenomenon found in a broad spectrum of nocturnal organisms [8]; similarly chemotaxis, or the biasing of movement towards environments that contain higher concentrations of beneficial chemicals, is observed in

Authors would like to thank Air Force Office of Scientific Research (AFOSR), grant no. FA9550-14-1-0398 by The Air Force Center of Excellence on Nature-Inspired Flight Technologies and Ideas (NIFTI) for funding the equipment for this research

¹Department of Mechanical Engineering, University of Washington, Seattle, WA, USA

²Department of Electrical and Computer Engineering, University of Washington, Seattle, WA, USA

even unicellular organisms such as bacteria [9]. In this work we leverage these insights to utilize a simple, bio-inspired algorithm and design an end-to-end system that guides a palm-sized drone towards a signal source while avoiding obstacles.

While recent work [10] has demonstrated successful light source seeking using reinforcement learning, this approach has a number of shortcomings for our target applications in complex and unknown environments. This approach cannot be generalized because the control policy generated is limited by the training data; the robot cannot seek a source in the center of a room as it never experienced that scenario. Similarly, scenarios without obstacles produced roundabout trajectories due to the robot’s tendency to avoid the center of the room and it was not able to navigate around closely spaced obstacles. Moreover, this algorithm has a high computational overhead, making it difficult to scale to SWaP limited platforms.

We instead examine less complex stochastic methods. Prior work has explored the idea of stochastic source seeking [11]–[14]; while these works provide mathematical guarantees for convergence, they have not been implemented on real hardware and do not incorporate obstacle avoidance which is essential in cluttered environments. Other works have also analyzed bio-inspired source seeking algorithms [15]–[17], but are also limited to simulation. In contrast, we seek to implement a source seeking system that can run in real-time with a palm-sized UAV in different environments.

While others have attempted tasks like RF source localization [18], the use of ground robots limits it to a nonholonomic action space. Additionally, source seeking using multiple robots has been performed in [19], [20], but to optimize for SWaP constraints we focus on a single robot design which does not require other infrastructure or communication.

To develop our bio-inspired search algorithm we look to chemical sensing efforts using mobile robots which has been actively researched since the 1990s, with a focus on gas source localization. A number of bio-inspired algorithms have been developed, mimicking the search procedures of animals like the silkworm moth *Bombyx mori*; however, these algorithms are designed for gas sensing in turbulent airflow [21]. For non-dynamic sources such as light, temperature, and radio signals, we can explore an even simpler class of strategies that rely solely on concentration gradients, such as iterative chemo-tropotaxis and Braitenberg vehicle strategies. These two approaches are feasible from a computational complexity perspective, but they require at least two sensors. To achieve a truly minimal approach which can scale to even smaller robots, we adopt the E. Coli run-and-tumble algorithm which can be implemented with a single sensor [21].

In this paper we present an end-to-end implementation and demonstrate that this simple ‘run and tumble’ algorithm works robustly. The system successfully locates 91% of the tested heat and light sources including a fire in real time while avoiding obstacles as shown in Fig. 1 and video provided in the supplemental material. Additionally, we develop a simulation framework and take measurements of an RF

TABLE I: Parameter values for algorithm 1

Parameter	Simulation	Light seeking	Heat seeking
stop_threshold	10^4	800 lux	13°C
fwd_velocity	90 pixels/sec	0.1 m/s	0.2 m/s
ao_time	100 ms	2 s	0.5 s
ao_angle	0.1°	20°	20°
run_time	10 ms	1 s	1 s
obst_threshold	40 pixels	0.5 m	0.35 m

source versus distance to demonstrate the potential for further applications.

The rest of the paper is organized as follows: Section II presents the algorithm for the source seeking along with detailed analysis of computational complexity. Section III presents the system components required for the study, followed by hardware experiments on the drone in Section IV. Section V concludes the results and discusses scalability.

II. ALGORITHM TESTING IN SIMULATION

We began designing and testing the algorithm in simulation using the Python library PyGame [22]. This allowed for refining of the algorithm while avoiding damage and wear on the hardware. The simulated environment consisted of five randomly placed circular obstacles of random size between 40 and 120 pixels in an arena measuring 700×700 pixels. We use a standard inverse square law model to model light propagation from a point source at the center. The inverse square law model was qualitatively verified in the real world (section IV). Simulated robot motion commands were designed to correspond to commands in the drone’s control library. This facilitated the direct transfer of the algorithm from simulation to robot.

The laser distance sensors were also simulated to give distances to obstacles in the front, right, back and left directions of the robot. To avoid the robot leaving the arena, the screen boundaries were also considered obstacles for the simulated distance sensors. We implemented a behavior-based algorithm for source seeking. The robot performed three behaviors:

- **run**: move forward with a constant velocity,
- **tumble**: turn left or right to a random angle ranging from 0° to 180° , and
- **avoid-obstacle**: move directly away from the closest detected obstacle along the direction of the distance sensor, and change heading slightly.

A simple finite state machine described in Algorithm 1 was designed [23] to switch between these behaviors. We implemented a tumble instead of strafing in a random direction for source seeking because the sensor needs to be pointing towards the source in applications like temperature seeking. In the case of obstacle avoidance, we simply strafed away from the obstacle, bypassing the need for a tumble. The robot declared reaching the light source when the intensity exceeded a threshold value set empirically based on the light source. The robot also successfully navigated narrow

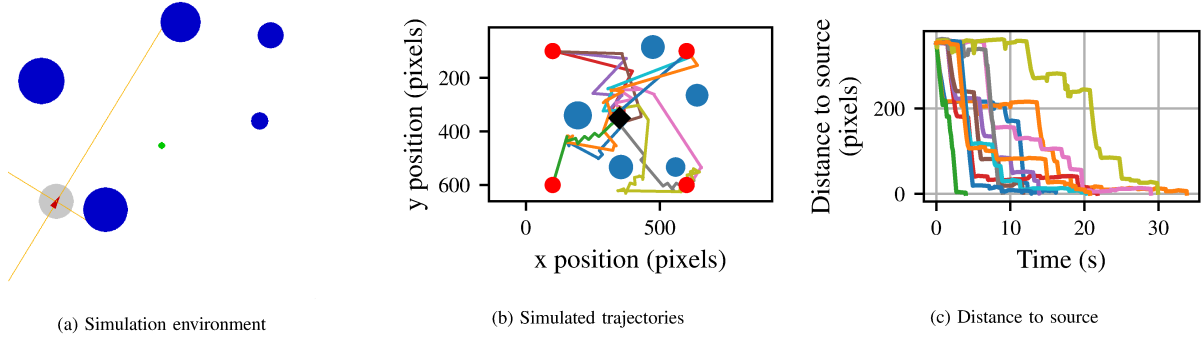


Fig. 2: (a) Screenshot of the simulation environment. The blue circles are randomly placed obstacles, (b) Multiple trajectories (different colors) with different starting points (red dots) at the same distance from the source (black diamond), (c) Variation of distance of the robot from the light source for the trajectories in fig. 2b with time.

Algorithm 1: Source seeking algorithm with obstacle avoidance. Parameter values are as given in table I

Data: Sensor readings (obtained asynchronously in experiment and synchronously in simulation)

Result: Control commands for source seeking

```

1 Function avoid-obstacle():
2   | move away from closest obstacle for ao_time;
3   | turn away from closest obstacle by ao_angle;
4   | start moving forward with fwd_velocity;
5 Function run():
6   | move forward with fwd_velocity for
   |   run_time;
7 Function tumble():
8   | if random(0,1) ≤ 0.5 then
9   |   | turn right by random(0, 180) degrees;
10  | else
11  |   | turn left by random(0, 180) degrees;
12  |   | start moving forward with fwd_velocity;
13 while signal_strength < stop_threshold do
14   | if min(distance_sensors) < obst_threshold
   |   | then
15   |   |   | avoid-obstacle();
16   | else
17   |   | if signal_strength > prev_strength then
18   |   |   | run();
19   |   |   | if source is temperature then
20   |   |   |   | stop for 3 sec
21   |   | else
22   |   |   | tumble();
23 declare source and land;

```

passages between closely placed obstacles. The simulation assumes:

- both sensor readings are noiseless
- the light intensity follows an inverse squared model
- the simulated laser distance sensors had full range for the entire arena
- the obstacles did not cast any shadows



(a) Arena for light source seeking



(b) Arena for temperature source seeking

Fig. 3: Experiment setup

Despite the assumptions made during development of the algorithm in simulation, physical experiments demonstrated that the algorithm can perform robustly with real sources in the presence of noise. It was also shown that for light, heat, and bluetooth sources, the signal strength distribution was qualitatively a monotonically decreasing function with some noise.

Fig. 2 shows trajectories of the algorithm executing the search in a given environment from varying initial conditions. The distance to the obstacle on average decreased with time in all trajectories we simulated. Intermittent increases in distance arise when a random tumble results in a trajectory away from the source, or when an obstacle avoidance maneuver interrupts a path with decreasing distance. However, these intermittent increases in distance can be decreased by choosing a small time interval between successive iterations. In practice, this is limited by the robot's dynamics and an appropriate time was chosen in experiments. These results suggest that this algorithm is able to robustly perform source seeking.

Table II shows that increasing the threshold distance in-

obst_threshold	average seek time for 10 runs
20 pixels	6.54 s
40 pixels	11.10 s
80 pixels	13.74 s
100 pixels	22.88 s

TABLE II: Seek times in simulation with different threshold distances for triggering obstacle avoidance behavior as per algorithm 1

creases the seek time. However, in practice, a very small threshold may lead to collisions with obstacles. The chosen value depends on the robot size, velocity and latency of the algorithm loop. In simulation, the latency could be made very small to yield a very efficient algorithm. In experiments, the light and temperature sensors provided a measurement every 120 ms and 200 ms respectively, and the robot's dynamics preclude immediate changes in direction. Accordingly, we chose a threshold of 50 cm for light and 35 cm for temperature seeking to ensure that the robot was able to consistently avoid obstacles. We note the values in table I were chosen such that the algorithm works in a variety of scenarios and do not require tuning using a-priori knowledge about the signal distribution or environment. A turn command was also added to the `avoid-obstacle()` behavior since without this command, the robot could get stuck on obstacles in the front.

Next we analyze the required computation. At every time increment of 500 ms, first the moving average filter is updated with 6 instructions: subtract the oldest value in the queue from the running total, increment a memory pointer, add n to queue pointer (for an n -length moving average window), store new reading at the top of the queue, add new reading to the running total, and divide by the number of samples. Then the algorithm performs a comparison for source declaration, as well as four distance threshold comparisons and another field strength comparison for run and tumble. A single control command is issued if the action determined by the mentioned comparisons is `run()`. Actions corresponding to `tumble()` and `avoid-obstacle()` require two and three control commands, respectively. The `run()` and `avoid-obstacle()` routines also need one timer comparison each. This gives a total of 14 to 16 instructions per cycle, or 28 to 32 per second.

III. SYSTEM COMPONENTS

A. Sensors

A BH1750FVI light intensity sensor (3 mm x 1.6 mm) was used to measure light intensity values. It consists of a photoresistor and an on-board 16-bit ADC integrated in a single chip that outputs digital intensity values. The sensor has a range of 1-65535 lux and was used in 1 lux resolution mode at 8 Hz. This sensor was mounted on the top of the drone facing up assuming a downward shining light source. For temperature sensing, a TI HDC2010 sensor (1.49mm x 1.49 mm) was used which can measure temperatures ranging from -40°C to 125°C with a 14 bit resolution at 5 Hz. The sensor was positioned facing the front of the drone. Both sensors use a standard $I^2\text{C}$ interface. To measure Bluetooth

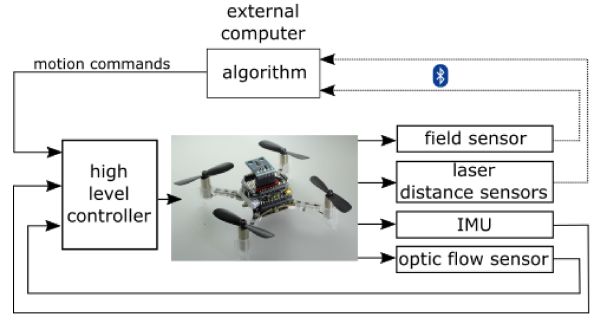


Fig. 4: Block diagram showing interaction of system components. The external computer sent control commands as per algorithm 1

signal strength, an evaluation module for the nRF52840 was attached to the drone and programmed to output its signal strength information (RSSI) over a UART link.

The Bitcraze multiranger deck was used to detect obstacles. It consists of five VL53L1X time of flight laser range finders on board with a range of 4 meters and a 50 Hz frequency. The sensors provide distances to lateral obstacles in four perpendicular directions as well as to overhead obstacles.

The Bitcraze optic flow deck was also used for navigation which consists of a PMW3901 optic flow sensor with a VL53L1X time of flight sensor. It provides the horizontal velocities and vertical position of the drone which can be fused with the on-board IMU to obtain 3D position.

B. Crazyflie quadcopter

A commercially available palm-sized drone platform called Crazyflie 2.1 by Bitcraze was used as the platform for implementing this algorithm. The robot weighs 27 g and measures 112 mm across. It is equipped with a Cortex-M4 microcontroller for onboard processing. The default firmware [24] was modified to include custom drivers for the light and temperature sensors to obtain sensor measurements.

The drone is configured to send light intensity or temperature measurements, multiranger distances and its 3D position to an external computer using its onboard Bluetooth radio. Based on these sensor values, a Python program computed control commands to be sent back to the drone to be executed on-board (fig. 4).

IV. EXPERIMENTS

Fig. 3 shows the arenas used for physical experiments. It measured approximately 8 m² and consisted of a single source and 2 to 3 obstacles in the robot's path.

The light intensity distribution was characterized by manually flying the drone in a closely-spaced trajectory at a 0.3 m height and recording the intensity values (Fig. 5a (a)). The light intensity qualitatively follows an inverse square distribution with distance from the source. Fig. 5 (b) shows that the the temperature around a fire, measured using the same procedure, follows a similar distribution, with more noise. This field characterisation was used to determine the threshold value used to declare the source had been found. The top plot in Fig. 5b shows that the temperature sensor takes a few seconds to stabilise, so a stop command

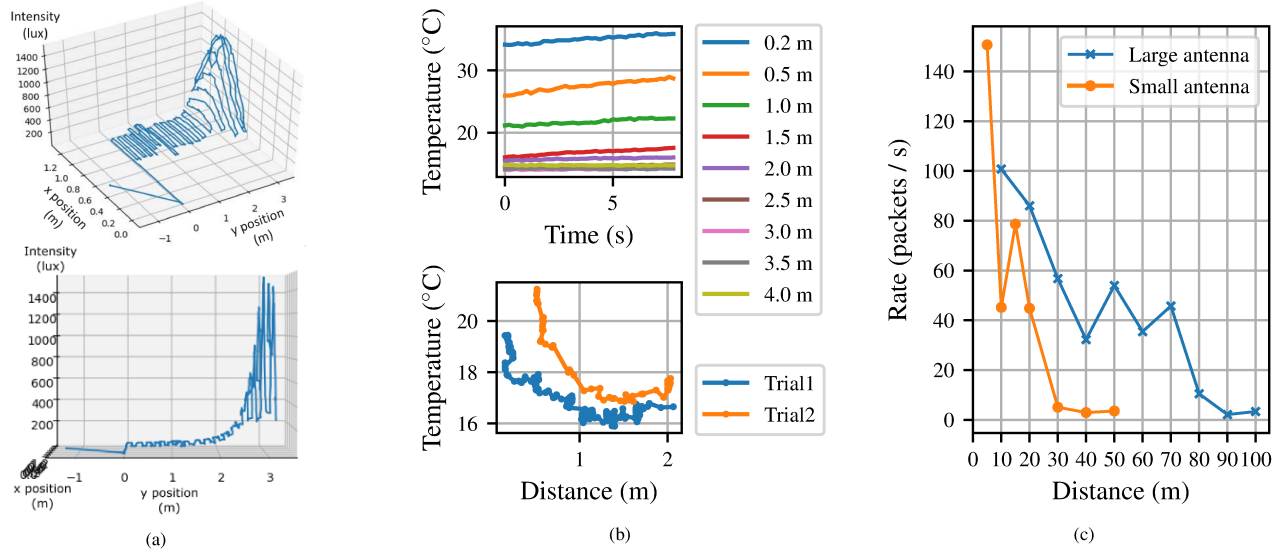


Fig. 5: (a) Light intensity distribution of an indoor light source using drone's position estimate. (top) with 2D position (bottom) along a straight line path towards the source, (b) Temperature distribution for a fire using ground truth distance (top), and drone position estimate (bottom), (c) Bluetooth source packet detection rate distribution with ground truth distance

was added after every `run()` command during fire-seeking trials. Fig. 5b(c) shows measurements from two Bluetooth transmitters programmed to send packets periodically. The large antenna has a smartphone form factor; the small antenna has a light-weight insect-scale form factor [3]. We performed measurements in an open field with the transmitter and receiver each at a height of 1 m. We observed a large variance in Received Signal Strength Indicator (RSSI) caused by multipath fading. While a strategy of averaging combined with circular or side-to-side motion patterns could be used to mitigate this, we found that packet detection rate produced a gradient similar to those we observed for temperature and light.

The same algorithm as designed in simulation was implemented on the drone with parameter values given in table I. The sensor measurements were obtained asynchronously at an external computer. All communication between the drone and the external computer was done via the on-board Bluetooth link. Fig. 4 shows the system components. To reduce sensor noise, field intensity readings were smoothed using a 10-element moving average filter for the light source. For the temperature source, the top plot in fig. 5b shows that the temperature sensor takes some time to stabilise. To address this, we added a stop command for 3 s after every run command and increased the moving average window to 30 samples. The robot moved in a 2D plane 0.3 m above ground.

The robot successfully localised the light source in 4 out of 4 trials and localised the fire in 6 out of 7 trials (one crash with an obstacle), giving a success rate of 100% and 85.71% respectively. Fig. 6 and Fig. 7 show the trajectories of the drone during the successful trials. The mean time to localize the light and temperature sources was 1 min 58 s and 1 min 13 s respectively. A video of these trials is available in the supplemental material and the code will be made available upon publication.

V. CONCLUSION AND SCALING DISCUSSION

In this work we demonstrated that a biology-inspired search strategy can be used to locate a signal source using an aerial robot as small as the palm of a human hand. The system only utilizes feedback from sensors carried on-board the drone to navigate towards the source and avoid obstacles. This work is an important step toward practical application of such SWaP constrained robots toward source seeking operations such as fire localization.

Next we discuss how these results are applicable to more severely SWaP-constrained robots that are significantly smaller, measuring a few centimeters or less. The simulation software allows us to explore other signal propagation models. Such distributions could incorporate noise models or empirical measurements from physical environments of interest. This allows for rapid experimentation to inform the design of future source following robots and tuning of parameters such as the velocity, run time, or tumble angle.

More importantly, our bio-inspired algorithm has two key advantages that allow it to scale down to insect-scale robots. First, it only requires a single sensor, so the robot can be physically very small. A single phototransistor (≈ 1 mg, 1 mm) can measure light [25]. Our fire seeking experiments used a small (3 mg, 1.49x1.49 mm) temperature sensor that has been used for insect-scale applications [3], [26]. Alternative sensors such as biological antennae for gas sensing could be used as well [27].

Second, the algorithm requires minimal computation, approximately 30 instructions per second. This constitutes a negligible fraction of the flight control system on our robot platform. Moreover, multiple insect-scale robot platforms have recently taken steps toward power autonomy [28]–[30]. As these efforts continue to progress, we anticipate a growing need for high level control algorithms that can allow these robots to perform useful tasks. Our bio-inspired run and

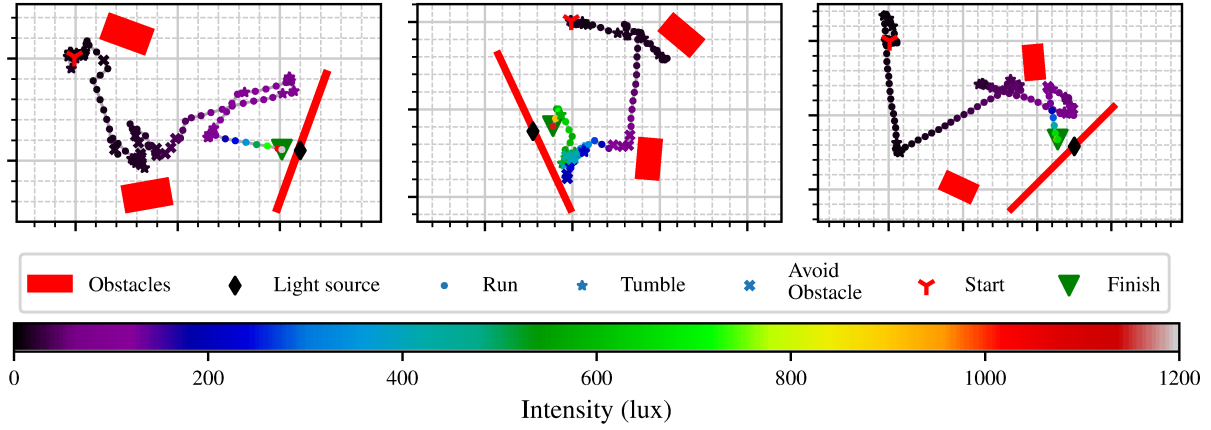


Fig. 6: Drone path for light source seeking trials. The flight path was estimated using the drone’s internal state estimator. Obstacle and source positions are approximate. Major gridlines are every 1 m and minor gridlines are every 20 cm.

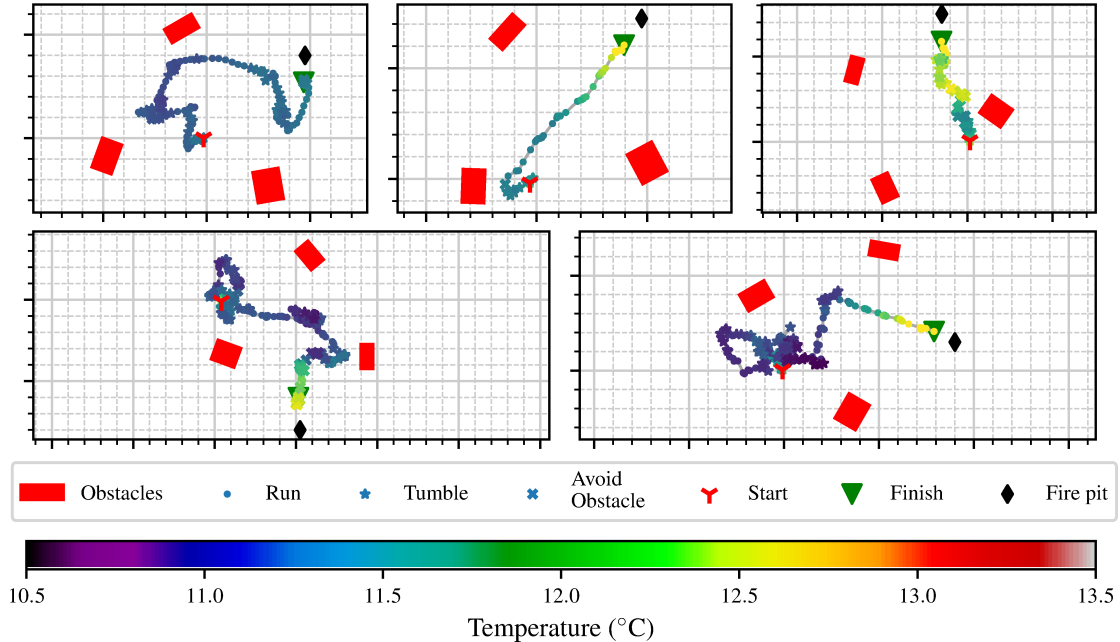


Fig. 7: Drone path for heat source seeking trials (the source is a fire). The flight path was estimated using the drone’s internal state estimator. Obstacle and source positions are approximate. Major gridlines are every 1 m and minor gridlines are every 20 cm.

tumble algorithm is an excellent candidate for this task as it can easily be implemented on the small microcontrollers used in these robots.

We emphasize that while building a further miniaturized robot is beyond the scope of this paper, the exact same sensor hardware (temperature/light sensors) and algorithm could be directly deployed on smaller robots. This technique could also help address the problem of powering small robots. For example, passive diode-detector circuits as shown in [26] can be used to measure the amplitude of an RF signal. If this were combined with the source seeking technique described in this work, it could allow a robot to come into close proximity to a RF power source such as a WiFi router and harvest emissions to recharge itself [31].

Future work could also improve on the algorithm itself.

The current implementation restricts the run motion to 2D, but this could be expanded into 3D space as well. The algorithm can be made more efficient by obtaining directional information by fusing information from other sensors. While we focus here on a minimalist implementation, robots with cameras or multiple sensors placed apart could de-randomize the tumble direction and improve source seeking times.

ACKNOWLEDGMENTS

The authors thank Yogesh Chukewad for insightful discussions during the preparation of this manuscript as well as Johannes James for assistance with the fire seeking experiments.

REFERENCES

- [1] Y. Karaca, M. Cicek, O. Tatli, A. Sahin, S. Pasli, M. F. Beser, and S. Turedi, "The potential use of unmanned aircraft systems (drones) in mountain search and rescue operations," *The American journal of emergency medicine*, vol. 36, no. 4, pp. 583–588, 2018.
- [2] C. Kislik, I. Dronova, and M. Kelly, "Uavs in support of algal bloom research: a review of current applications and future opportunities," *Drones*, vol. 2, no. 4, p. 35, 2018.
- [3] V. Iyer, M. Kim, S. Xue, A. Wang, and S. Gollakota, "Airdropping sensor networks from drones and insects," in *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, pp. 1–14, 2020.
- [4] K. Grasegger, G. Strapazzon, E. Procter, H. Brugger, and I. Soteras, "Avalanche survival after rescue with the recco rescue system: a case report," *Wilderness & Environmental Medicine*, vol. 27, no. 2, pp. 282–286, 2016.
- [5] P. J. Kennedy, S. M. Ford, J. Poidatz, D. Thiéry, and J. L. Osborne, "Searching for nests of the invasive asian hornet (*vespa velutina*) using radio-telemetry," *Communications biology*, vol. 1, no. 1, pp. 1–8, 2018.
- [6] K. Schlosser, "Uw researcher put tiny tracking technology on giant hornets to help state deal with murderous pest," <https://www.geekwire.com/2020/uw-researcher-put-tiny-tracking-technology-giant-hornets-help-state-deal-murderous-pest/>, 2020.
- [7] W. Giernacki, M. Skwierczyński, W. Witwicki, P. Wroński, and P. Kozierski, "Crazyflie 2.0 quadrotor as a platform for research and education in robotics and control engineering," in *2017 22nd International Conference on Methods and Models in Automation and Robotics (MMAR)*, pp. 37–42, IEEE, 2017.
- [8] A. C. Owens and S. M. Lewis, "The impact of artificial light at night on nocturnal insects: A review and synthesis," *Ecology and evolution*, vol. 8, no. 22, pp. 11337–11358, 2018.
- [9] G. H. Wadhams and J. P. Armitage, "Making sense of it all: bacterial chemotaxis," *Nature reviews Molecular cell biology*, vol. 5, no. 12, pp. 1024–1037, 2004.
- [10] B. P. Duisterhof, S. Krishnan, J. J. Cruz, C. R. Banbury, W. Fu, A. Faust, G. C. de Croon, and V. Janapa Reddi, "Learning to seek: Deep reinforcement learning for phototaxis of a nano drone in an obstacle field," *arXiv*, pp. arXiv–1909, 2019.
- [11] S.-i. Azuma, M. S. Sakar, and G. J. Pappas, "Stochastic source seeking by mobile robots," *IEEE Transactions on Automatic Control*, vol. 57, no. 9, pp. 2308–2321, 2012.
- [12] E. Ramirez-Llanos and S. Martinez, "Stochastic source seeking for mobile robots in obstacle environments via the spsa method," *IEEE Transactions on Automatic Control*, vol. 64, no. 4, pp. 1732–1739, 2018.
- [13] S.-J. Liu and M. Krstic, "Stochastic source seeking for nonholonomic unicycle," *Automatica*, vol. 46, no. 9, pp. 1443–1453, 2010.
- [14] J. Cochran and M. Krstic, "Nonholonomic source seeking with tuning of angular velocity," *IEEE Transactions on Automatic Control*, vol. 54, no. 4, pp. 717–731, 2009.
- [15] I. Rañó, M. Khamassi, and K. Wong-Lin, "A drift diffusion model of biological source seeking for mobile robots," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3525–3531, IEEE, 2017.
- [16] R. Zou, V. Kalivarapu, E. Winer, J. Oliver, and S. Bhattacharya, "Particle swarm optimization-based source seeking," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 3, pp. 865–875, 2015.
- [17] J. R. Bourne, E. R. Pardyjak, and K. K. Leang, "Coordinated bayesian-based bioinspired plume source term estimation and source seeking for mobile robots," *IEEE Transactions on Robotics*, vol. 35, no. 4, pp. 967–986, 2019.
- [18] N. Atanasov, J. Le Ny, N. Michael, and G. J. Pappas, "Stochastic source seeking in complex environments," in *2012 IEEE International Conference on Robotics and Automation*, pp. 3013–3018, IEEE, 2012.
- [19] N. A. Atanasov, J. Le Ny, and G. J. Pappas, "Distributed algorithms for stochastic source seeking with mobile robot networks," *Journal of Dynamic Systems, Measurement, and Control*, vol. 137, no. 3, 2015.
- [20] S. Li, R. Kong, and Y. Guo, "Cooperative distributed source seeking by multiple robots: Algorithms and experiments," *IEEE/ASME Transactions on mechatronics*, vol. 19, no. 6, pp. 1810–1820, 2014.
- [21] A. J. Lilienthal, A. Loutfi, and T. Duckett, "Airborne chemical sensing with mobile robots," *Sensors*, vol. 6, no. 11, pp. 1616–1678, 2006.
- [22] P. Shinnars, "Pygame." <http://pygame.org/>, 2011.
- [23] "GitHub repository for crazyflie-run-and-tumble." Available: <https://github.com/thecountoftuscany/crazyflie-run-and-tumble>. 2020.
- [24] "GitHub repository for crazyflie-firmware." Available: <https://github.com/bitcraze/crazyflie-firmware/>. 2020.
- [25] S. B. Fuller, M. Karpelson, A. Censi, K. Y. Ma, and R. J. Wood, "Controlling free flight of a robotic fly using an onboard vision sensor inspired by insect ocelli," *Journal of The Royal Society Interface*, vol. 11, no. 97, p. 20140281, 2014.
- [26] V. Iyer, R. Nandakumar, A. Wang, S. B. Fuller, and S. Gollakota, "Living iot: A flying wireless platform on live insects," in *The 25th Annual International Conference on Mobile Computing and Networking*, pp. 1–15, 2019.
- [27] M. J. Anderson, J. G. Sullivan, J. L. Talley, K. M. Brink, S. B. Fuller, and T. L. Daniel, "The "smellicopter," a bio-hybrid odor localizing nano air vehicle," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6077–6082, IEEE, 2019.
- [28] J. James, V. Iyer, Y. Chukewad, S. Gollakota, and S. B. Fuller, "Liftoff of a 190 mg laser-powered aerial vehicle: The lightest wireless robot to fly," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–8, IEEE, 2018.
- [29] A. T. Baisch, C. Heimlich, M. Karpelson, and R. J. Wood, "Hamr3: An autonomous 1.7 g ambulatory robot," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5073–5079, IEEE, 2011.
- [30] N. T. Jafferis, E. F. Helbling, M. Karpelson, and R. J. Wood, "Untethered flight of an insect-sized flapping-wing microscale aerial vehicle," *Nature*, vol. 570, no. 7762, pp. 491–495, 2019.
- [31] V. Talla, B. Kellogg, B. Ransford, S. Naderiparizi, S. Gollakota, and J. R. Smith, "Powering the next billion devices with wi-fi," in *Proceedings of the 11th ACM Conference on Emerging Networking Experiments and Technologies*, pp. 1–13, 2015.