

TinySense: A Lighter Weight and More Power-efficient Avionics System for Flying Insect-scale Robots

Zhitao Yu^{*,1,3}, Joshua Tran^{*,2}, Claire Li², Aaron Weber¹, Yash P.Talwekar¹, Sawyer Fuller^{1,2}

Abstract—In this paper, we introduce advances in the sensor suite of an autonomous flying insect robot (FIR) weighing less than a gram. FIRs, because of their small weight and size, offer unparalleled advantages in terms of material cost and scalability. However, their size introduces considerable control challenges, notably high-speed dynamics, restricted power, and limited payload capacity. While there have been advancements in developing lightweight sensors, often drawing inspiration from biological systems, no sub-gram aircraft has been able to attain sustained hover without relying on feedback from external sensing such as a motion capture system. The lightest vehicle capable of sustained hovering—the first level of “sensor autonomy”—is the much larger 28 g Crazyflie. Previous work reported a reduction in size of that vehicle’s avionics suite to 187 mg and 21 mW. Here, we report a further reduction in mass and power to only 78.4 mg and 15 mW. We replaced the laser rangefinder with a lighter and more efficient pressure sensor, and built a smaller optic flow sensor around a global-shutter imaging chip. A Kalman Filter (KF) fuses these measurements to estimate the state variables that are needed to control hover: pitch angle, translational velocity, and altitude. Our system achieved performance comparable to that of the Crazyflie’s estimator while in flight, with root mean squared errors of 1.573 deg, 0.186 m/s, and 0.136 m, respectively, relative to motion capture.

I. INTRODUCTION

Insect-scale robots have the potential to be deployed in the thousands or millions to perform “fast, cheap, and out of control” space missions, collective assembly tasks, or hazard detection, owing to their small size and low materials cost. Recent incarnations of sub-gram aircraft have demonstrated controlled flight [1], [2], as well as electromagnetically-mediated power delivery [3], [4], [5]. But sensor autonomy, that is, the ability to hover without external feedback, has not yet been demonstrated below 10 g, never mind 1 g. A central question is “What are the minimum sensor suite and computation resources needed for the task of flight control?” Of particular concern when designing such a control architecture is how the physics of scale affects sensing and control. The paper [6] serves as a step toward a missing but complementary analysis to previous work that has largely focused on scaling effects on actuation and mechanics in small robots [7], [8], [9], [10].

This research was partially supported by the National Science Foundation through awards ECCS-2054850 and CNS-2235207

*Authors contributed equally.

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

²Paul G. Allen School of Computer Science, University of Washington, Seattle, WA, USA

³Department of Applied Mathematics, University of Washington, Seattle, WA, USA

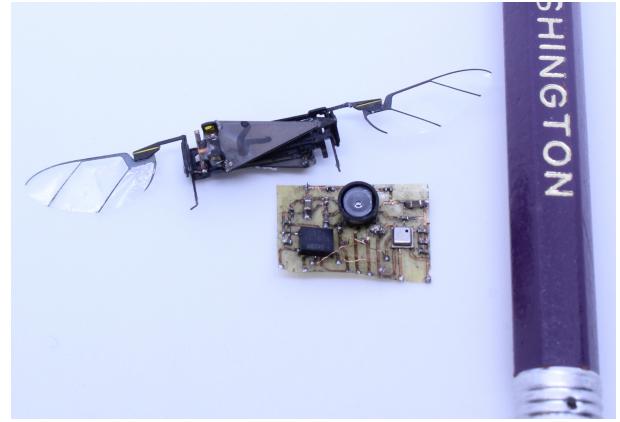


Fig. 1: The TinySense sensor suite shown next to a U. Washington Robofly and a standard pencil for scale. The width of the sensor suite board is approximately 12 mm and its mass is 78.4 mg.

Insect-scale robots have extreme size, speed, weight, and power (SSWaP) constraints. “Sensor autonomy” including stable hovering, represents the first level of autonomy in the hierarchy proposed in [11], and is the foundation upon which higher level capability is built. We impose a requirement that sensing and computation take no more than 10% of the power required to stay in the air; any more than this and flight time is severely impacted. This factor holds for important examples of autonomous drones, such as the 1.5 kg system in [12] and the 30 g system in [13]. This suggests a power budget of ~ 20 mW for a 150 mg Robofly (Figure 1) that consumes 100 mW to fly (about 200 mW after losses from a 50% efficient boost converter are factored in).

Large drones can carry larger sensor payloads such as light-based ranging systems (LIDAR) and the global positioning system (GPS). However, GPS does not work well in indoor environments [14] and small robots are not able to hold emissive sensors due to constraints in size, speed, weight, and power consumption (SSWaP). Recent work has introduced a 187 mg avionics suite that consumes 21 mW [15], considered gyroscope-free avionics [6], and has investigated the effect of wing-induced vibrations [16].

In this paper, we introduce a new avionics system for autonomous hovering flight that is even better tailored in mass and energy consumption for an insect-scale robot (Fig. 3) than previous work [15]. To do so, we made two key refinements to lower power and mass. First, we replaced the power-hungry laser rangefinder with a pressure sensor. In recent years, these have attained sufficient precision to be used for the purpose of altitude estimation and control, even

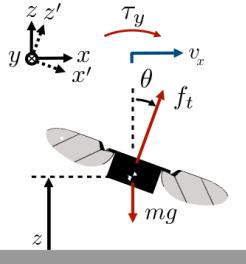


Fig. 2: Forces and state variables of a small hovering Flying Insect Robot (FIR).

for insect-scale robots. Second, we replaced the optic flow sensor with a global shutter camera and computed optic flow on a microcontroller, resulting in a reduction in weight from 97 mg to 44 mg. As in [15], this new sensor suite is able to estimate the crucial variables that are needed for controlled flight: pitch angle, translational velocity, and altitude.

Our contributions, in order of importance, are:

- 1) A reduction in weight of the sensor suite to 78.4 mg which is less than half of the previous lightest sensor suite [15].
- 2) Demonstration of accurate estimates of the state variables pitch angle, translational velocity, and altitude using data collected wirelessly from this sensor suite while mounted on a drone.
- 3) Optic flow estimation on a microcontroller.

Subsequent sections elaborate on the dynamics, sensor configurations and measurement models, and validate our findings with data acquired wirelessly from our sensor suite, which is then analyzed on a desktop computer.

II. METHODOLOGY

A. Dynamics

The basic dynamics of most small hovering devices are unstable in pitch angle and position. This was true for the first robotic flies [17] and EHD thruster actuated robots [18], as well as larger aircraft such as the tailless Delfly Nimble [19]. Techniques to stabilize them passively through modification of wing configuration [20] or using air dampers [21] are either unproven or result in undesirable steady-state oscillations, respectively. Our approach in this work is to first use linearized dynamics and observation models from which we can construct a Kalman observer of these unstable states so that they can be controlled.

Fig. 2 depicts forces acting on a flapping-wing robot such as a gnat robot or the Robofly, but the proposed model can apply to almost any small flying aircraft, including EHD-based thrusters [22]. Near hover, the vehicle's dynamics are well approximated by a linear state-space model [23], [24], [21] of the form $\dot{q} = Aq + Bu + Gw$, where w is the process noise, assumed to be zero-mean Gaussian white noise with known covariance. Process noise w is included because it is a necessary consideration for the design of the estimator.

To simplify the analysis, we restrict our attention to motion in the x - z plane; the dynamics in the y - z plane are very similar. We choose a minimal state vector $q = [\theta, v_x, z]^T$, where

θ is the pitch angle (rad), v_x is the vehicle's translational velocity in the world x -direction (m/s), and z is its vertical position or altitude (m) as illustrated in Fig. 2. When the nonlinear Euler-Lagrange equation dynamics are linearized by taking the Jacobian at $\theta = 0$, thrust force $f_t = mg$ (N), and $z = z_d$, where z_d is the desired altitude (m), the dynamics matrices are given by:

$$A = \begin{bmatrix} 0 & 0 & 0 \\ g & -\frac{b}{m} & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad (1)$$

where m is the vehicle's mass (kg), and b is the vehicle's translational drag proportionality constant (Ns/m) [24]. Note that because we are concerned in this paper about only estimation, we ignore inputs from the motor, and instead use the measurements of the gyroscope $u = [\omega_m]$ (rad/s) as the only input. This is discussed further in Section V.

III. SENSOR SUITE AND MEASUREMENTS

Next we consider the sensor suite and state estimator. Our interest is in sensors that are either available commercially off-the-shelf, nearly so, or can be fabricated using the same tools that have been successful building insect robots such as the Robofly [25] and Robobee [26], [1]. A further constraint is that they require extremely low power for both the sensor itself and subsequent transduction and processing. This largely eliminates sensors that emit power, such as radar, sonar, depth cameras that emit structured light, and scanning laser rangefinders [27], but permits passive sensing such as vision. The global positioning system is largely denied indoors and does not provide enough resolution anyway (1–10 m), and, further, requires significant power for signal processing. A lightweight and low-power sensing modality that can provide the necessary information to hover is “optic flow,” which is defined as the velocity of the image (in rad/s) across the camera imager [28], [29], [30]. In tandem with other sensors, optic flow can be used to estimate the distance to obstacles as the imager translates through space [31]. Methods for estimating optic flow include calculating luminance gradients locally [32] (Lucas-Kanade), or image-wide [33], as well as iterative search for matching blocks of pixels [34]. Distances can also be estimated with a second camera and stereo vision using line matching [35], but this requires a second camera and a long “baseline” between cameras for accuracy, which adds excessive weight.

Given these constraints, we propose the following set of onboard sensors: An optical sensor, a pressure sensor, and a gyroscope. We designed a customized microcircuit with the three sensors. (Fig. 3)

A. Camera and optic flow

The system employs an optical sensor, the PAG7920LT (PixArt Imaging, Inc., Taiwan), to capture 160×120 QQVGA images with a 120° field of view at 100 Hz over a 32 MHz SPI connection, with which the microcontroller computes optic flow via the Lucas-Kanade Method. The reason we select this camera is that, to our knowledge, it is the only

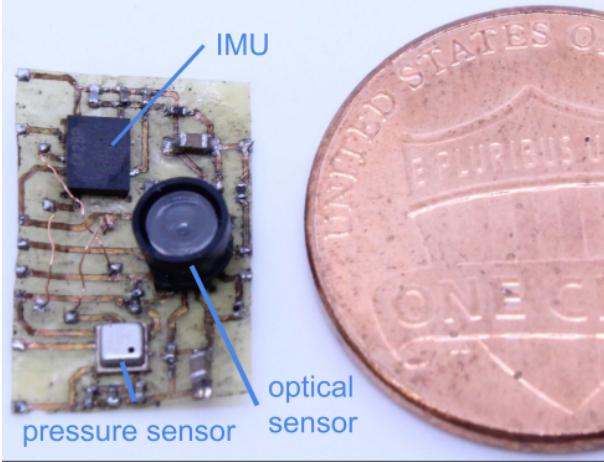


Fig. 3: Sensor suite (PAG7920LT optical sensor, ICM42688-P IMU, BMP388 pressure sensor) next to US 1 cent coin.

global shutter camera that provides an SPI interface to transmit the image data. Other global shutter cameras require a special interface called MIPI (mobile industry processor interface), which is not available for any microcontroller. The majority of cameras employ a “rolling shutter” in which pixel rows are captured at sequential periods of time. A global shutter camera, by contrast, captures that all pixels at the same time, which ensures an accurate optic flow measurement.

To minimize the data transmitted wirelessly over Bluetooth from our TinySense sensor suite, it is necessary to perform all optic flow calculations onboard. We wrote a Lucas-Kanade algorithm using the Arduino development environment in C++ to run on NRF52-based Adafruit Express board, chosen for its low size and weight.

The Lucas-Kanade algorithm is performed by calculating the following quantities:

$$J = \begin{bmatrix} I_x(p_1) & I_y(p_1) \\ I_x(p_2) & I_y(p_2) \\ \dots & \dots \\ I_x(p_n) & I_y(p_n) \end{bmatrix}, \mathbf{v} = \begin{bmatrix} V_x \\ V_y \end{bmatrix}, b = \begin{bmatrix} -I_t(p_1) \\ -I_t(p_2) \\ \dots \\ -I_t(p_n) \end{bmatrix},$$

where p_1, p_2, \dots, p_n are the pixels in the image, \mathbf{v} is the image flow in pixels per frame, and $I_x(p_i), I_y(p_i), I_t(p_i)$ are the image’s horizontal, vertical, and temporal (time) partial derivatives respectively, at pixel p_i . Then, the optic flow is estimated (in pixels/frame) by taking the pseudoinverse $\mathbf{v} = (J^T J)^{-1} J^T b$.

To minimize computation for the limited microcontroller resources, we reduced the image resolution by a factor of 4 (producing a 40×30 resolution) by skipping pixels.

The two most expensive operations involve computing $J^T J$ and $J^T b$. A 40×30 image produces an J matrix with $(40-2) \times (30-2) = 1064$ rows and 2 columns and a b matrix with 1064 rows and 1 column. Multiplying two matrices of size $k \times l$ and $l \times m$ requires at worst kml multiplications. Thus, computing $J^T J$ and $J^T b$ require 4256 and 2128 multiplications, respectively. We neglect the additional operations required to compute their product (2 multiplications) and the

2×2 matrix inverse (6 multiplications and one divide), giving a total operations count of 6384.

We additionally implemented a form of sparse optic flow in which flow vectors were computed for twelve 10×10 patches, which are averaged to produce a final estimate. A 10×10 image produces an J matrix with $(10-2) \times (10-2) = 64$ rows and 2 columns and a b matrix with 64 rows and 1 column. Thus, computing $J^T J$ and $J^T b$ require approximately 256 and 128 multiplications, respectively. With twelve matrices of size 10×10 , the total number of multiplication is $12 \times (256 + 128) = 4608$. This is a reduction of approximately 28%.

To convert the optic flow from pixels/frame to angular velocity (rad/s), we found the calibration constant $\alpha = -0.027$ (rad/pixel) by rotating the TinySense by hand and performing a linear regression against its gyroscope readings.

B. Pressure sensor

Altitude measurements were conducted by a Bosch BMP388 pressure sensor. This sensor estimates altitude as a proportion of the difference between the air pressure and sea level. We set it for $2 \times$ pressure oversampling and an IIR filter coefficient of 3 to maximize resolution. To compensate for variations in local pressure causing inaccurate altitude measurements, the sensor was assigned a bias equal to the average of the first 25 measurements where the sensor system was grounded; this bias was subtracted from every reading.

C. Gyroscope

For the onboard inertial measurement unit (IMU), we chose the TDK ICM42688-P because it is a recent model that is available in the smallest commercially-available package size of 2.5×3 mm. It includes both a 3-axis gyroscope and a 3-axis accelerometer. In this work, we used only the gyroscope, which measures angular velocity through the detection of Coriolis forces within an electromechanical resonator. We configured the sensor to have a ± 2000 degrees/s full-scale (FS) range and enabled the sensor-provided low-noise mode to improve the sensitivity and minimize noise. During initial experiments, we noticed that the gyroscope’s signal was noisier than expected and differed significantly from the Crazyflie’s gyroscope, even after applying a digital low-pass filter. In handheld, non-flying experiments, however, the gyroscope proved accurate. We hypothesized that vibrations from the spinning propellers, which were at a higher frequency than the 100 Hz sampling rate, were introducing aliasing. We enabled the sensor’s anti-aliasing low-pass filter with a cutoff frequency of 42 Hz, which is well below the Nyquist frequency associated with the 100 Hz sampling rate. We found that this setting significantly reduced the gyroscope’s noise and error.

IV. STATE ESTIMATION

To implement a Kalman Filter, we must first ascertain whether our system is observable at equilibrium. To show this, we linearize the observation model for the task of hovering. Note that the gyroscope provides a low-noise

TABLE I: Comparison between TinySense and prior work

	component	size (mm)	mass (mg)	power (mW)
prior work [15]	IMU	2.5×3×0.91	14	3
	rangefinder	4.9×2.5×1.56	16	5
	optical flow	5×5×3.08	97	12
	board/discretes	–	60	–
	total	–	187	21
TinySense	IMU	2.5×3×0.91	14	3
	pressure	2.0×2.0×0.75	14	2.6
	camera chip	2.2×2.7×6.9	24	10
	lens	3.8×3.8×1.95	20	–
	board/discretes	–	26.4	–
	total	–	78.4	15.6

measurement of angular velocity ω . In cases like this in which the sensor provides more precise information than can be derived knowing the system's actuated inputs (e.g. torque applied by the aircraft's rotors or wings), it is common practice to reformulate the Kalman Filter to use the sensor's readings as "input" u , and to ignore actual system input. This has an additional advantage that one less state variable is required for each such measurement. Our system includes gyroscope readings as one of its inputs and hence angular velocity ω does not appear among the state variables. Our observation model for the suite of sensors is:

$$\mathbf{y} = \begin{bmatrix} \Omega_m + n_o \\ z_m + n_p \end{bmatrix}, \quad (2)$$

where Ω_m is the optic flow measurements (rad/s) from optic flow sensor, z_m is the measurement (m) from the pressure sensor, and n_o and n_p are the noise terms for the two sensors. We assume noise terms are zero-mean Gaussian white noise, a standard assumption in sensor noise modeling to simplify the analysis. The measurement model for the optic flow in the x direction, as measured by the optic flow camera is a nonlinear function of the state [6]:

$$\Omega_m = \omega_m - \frac{v_x}{z}, \quad (3)$$

where ω_m is the angular velocity measurement (rad/s) from gyroscope. The optic flow measurement depends on z . We can take the Jacobian linearization at a desired altitude z_d :

$$\Omega_m(z) = \Omega_m(z_d) + O(z) = \omega_m - \frac{v_x}{z_d} + O(z). \quad (4)$$

With this linearization, our measurement model can be cast into the Kalman Filter model, $\mathbf{y} = C\mathbf{q} + D\mathbf{u} + \mathbf{n}$, where

$$C = \begin{bmatrix} 0 & -\frac{1}{z_d} & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad D = \begin{bmatrix} 1 \\ 0 \end{bmatrix}. \quad (5)$$

The observability matrix $[C; CA]$ is full rank, meaning that a Kalman Filter will be able to estimate the system's state using the measurements given in Eq. (5).

The Kalman Filter observer estimates $\hat{\mathbf{q}}$ by numerically integrating the system:

$$\dot{\hat{\mathbf{q}}} = A\hat{\mathbf{q}} + B\mathbf{u} + K(\mathbf{y} - C\hat{\mathbf{q}} - D\mathbf{u}), \quad (6)$$

where K is the Kalman gain.

We calculated optic flow variance R_{n1} to be 0.017 by subtracting its value from that predicted by mocap using Eq. (3) during a constant-velocity horizontal translational

flight. We calculated the variance of the pressure sensor R_{n2} to be 0.0055 by taking measurements while holding it at a constant altitude for 60 seconds:

$$R_n = \text{diag}(R_{n1}, R_{n2}) = \text{diag}(0.017, 0.0055).$$

To estimate the process noise covariance matrix Q_n , we employed a hybrid optimization approach, combining grid search for coarse parameter initialization, gradient descent for fine-tuning, and manual adjustment to minimize the RMSE between TinySense state estimates and mocap ground truth:

$$Q_n = \text{diag}(0.0124^2, 0.001^2, 0.22^2).$$

For the process noise (disturbance) matrix G , we assumed that the noise enters the system as white noise disturbance adding to the derivatives of the three states θ , v_x , and z . Based on these, the Kalman gain K was computed using the `lqe` command in python-control [36]:

$$G = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad K = \begin{bmatrix} 0.095 & 0 \\ -1.32 & 0 \\ 0 & 3 \end{bmatrix}.$$

We implemented the Kalman Filter offline in Python by iteratively computing the state estimates $\hat{\mathbf{q}}_{i+1}$ according to $\hat{\mathbf{q}}_{i+1} = \hat{\mathbf{q}}_i + \dot{\hat{\mathbf{q}}}_i \cdot dt$.

V. EXPERIMENTS

A. Robot platform

We performed experiments using Crazyflie 2.0, a palm-sized quadrotor by Bitcraze (Sweden), in wind-free conditions in an indoor environment. In addition to its built-in gyroscope, the Crazyflie was equipped with the Flow deck v2 consisting of a downward-facing optic flow sensor and a laser rangefinder; it uses an Extended Kalman Filter to estimate its state in flight [37].

B. Motion capture system

To provide a ground-truth comparison for the sensors, we used a twelve-camera OptiTrack Flex 13 motion capture arena, with spatial accuracy ± 0.20 mm. The Crazyflie drone was fitted with four retro-reflective motion capture markers and its position and orientation were recorded at 120 Hz. Velocity was estimated by taking the numerical derivative of the position data in post-processing.

C. Experiment

We connected our sensor suite to an NRF52840 Express (Adafruit), a board which supports data transmission through UART and 2.4 GHz Bluetooth Low Energy compatibility. We mounted the board as close to the geometrical center of the Crazyflie as possible to maintain a reasonable center of mass and minimize vibration interference to the IMU from the propellers. Power was supplied from the Crazyflie's 250 mAh Lithium Polymer (LiPo) battery.

Each trial consisted of commanding the robot to rise to an altitude of 1 m, translate forward 1 m at 1 m/s, and then descend back to the ground. We ran this experiment three times.

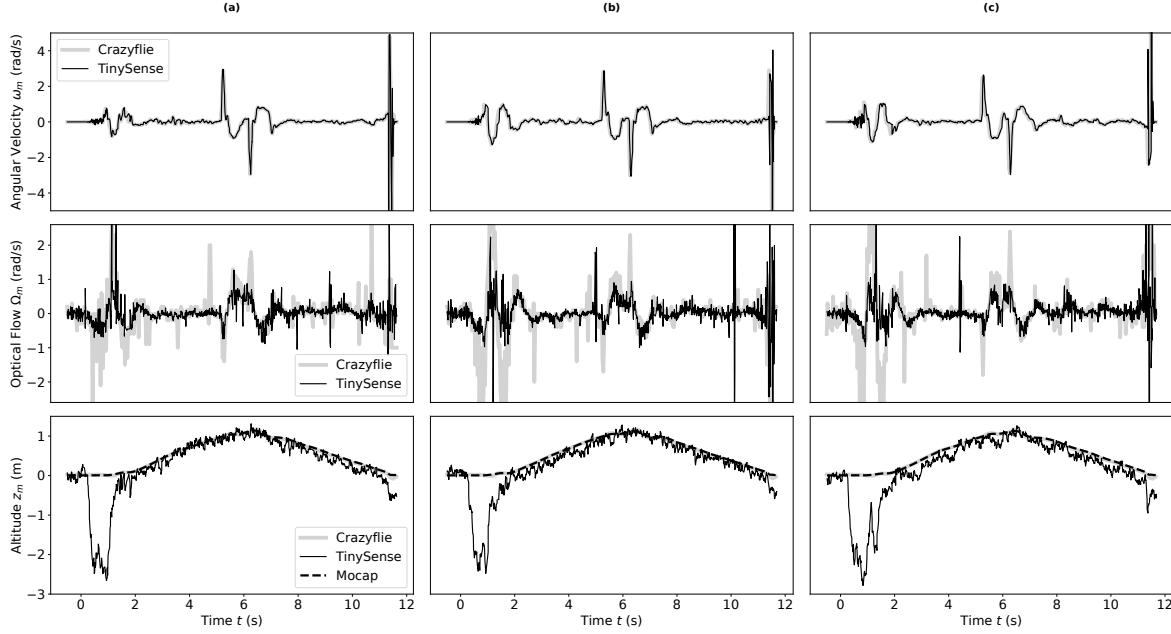


Fig. 4: Sensor measurements from the onboard gyroscope, optic flow sensor, and pressure sensor of TinySense are compared with those from the Crazyflie for three different flight experiments (a), (b), and (c); trajectories are described in the main text and corresponding state estimates in Fig. 5. In the altitude plots, altitude is measured by the Crazyflie’s laser rangefinder, while the TinySense uses a pressure sensor; mocap data is also included for comparison.

We collected measurements from the TinySense’s sensors on a laptop as they were transmitted over Bluetooth at 100 Hz as binary data. Simultaneously, a second computer captured motion capture data at 120 Hz and used a Python-based script to capture sensor measurements and state estimates transmitted from the Crazyflie over Bluetooth at its maximum rate of 30 Hz. Each receiving device independently recorded a UTC timestamp, which was used to synchronize the data. The sensor measurements from the TinySense compared with the Crazyflie, and the motion capture (mocap) system’s measurement for altitude are shown in Fig. 4.

D. Computational load

Table II provides an estimate of the power consumption of the optic flow algorithm, by calculating the number of clock cycles needed for each operation. Our estimate includes the matrix operations described in Section III-A as well as additional housekeeping operations, and is based on ARM Cortex-M4 instruction timings for addition, subtraction, multiplication, and memory read/write operations; integer divisions require 12 cycles at most and floating-point divisions require 14 cycles. Given that in our current system all sensors update at a rate of 100 Hz, we estimated that the total number of needed cycles per second is ~ 25.0 Mhz. Operating the nRF52840 with an efficiency of $52 \mu\text{A}/\text{MHz}$ and a battery-powered voltage of 4.2 V, we estimate the power usage of the optic flow algorithm to be 5.5 mW.

E. Results and discussion

Fig. 5 compares our sensor suite’s state estimate to that of the Crazyflie’s built-in estimator. States derived from the mocap system are considered to be ground truth. For the state

estimator, time $t = 0$ corresponds to when the gyroscope reading becomes significantly different from zero, indicating liftoff. Fig. 4 shows that a pressure increase occurs when the propellers first turn on, which can be observed as a drop in measured altitude. To mitigate unwanted effects of this in the Kalman Filter, we ignored (set to zero) altitude measurements z_m during the first 1.8 s after liftoff. Nevertheless, once it is in the air, the altitude measurement matches quite closely with the laser rangefinder on the Crazyflie, though with slightly more noise.

Our system’s estimate compares well with the Crazyflie’s, despite receiving gyroscope readings at only 100 Hz, which is ten times slower [37]. The key metric of pitch angle RMSE is approximately equal (Table III). Large errors in translational velocity v_x appear around $t = 5$ and $t = 6$. These occur after sudden changes in attitude, during which angular velocity exceeds 3 rad/s (Fig. 4). At these moments, we observed that TinySense optic flow readings are consistently lower than those of the Crazyflie. Lucas-Kanade cannot detect motion accurately if the displacement between frames is greater than a half pixel between image frames [38]. This limits optic flow measurement to $\Omega_m < 0.5\alpha F = 1.4$ rad/s, where $\alpha=0.027$ is the angular increment between pixels and $F=100$ is the frame rate (Hz). We believe that the lateral velocity estimate diverges during these fast maneuvers because it depends on erroneous optic flow readings.

VI. CONCLUSION

In the present study, we introduce a new sensor suite with dramatically reduced mass, and associated optic flow and state estimation software that is compatible with the computational constraints of an onboard microcontroller small

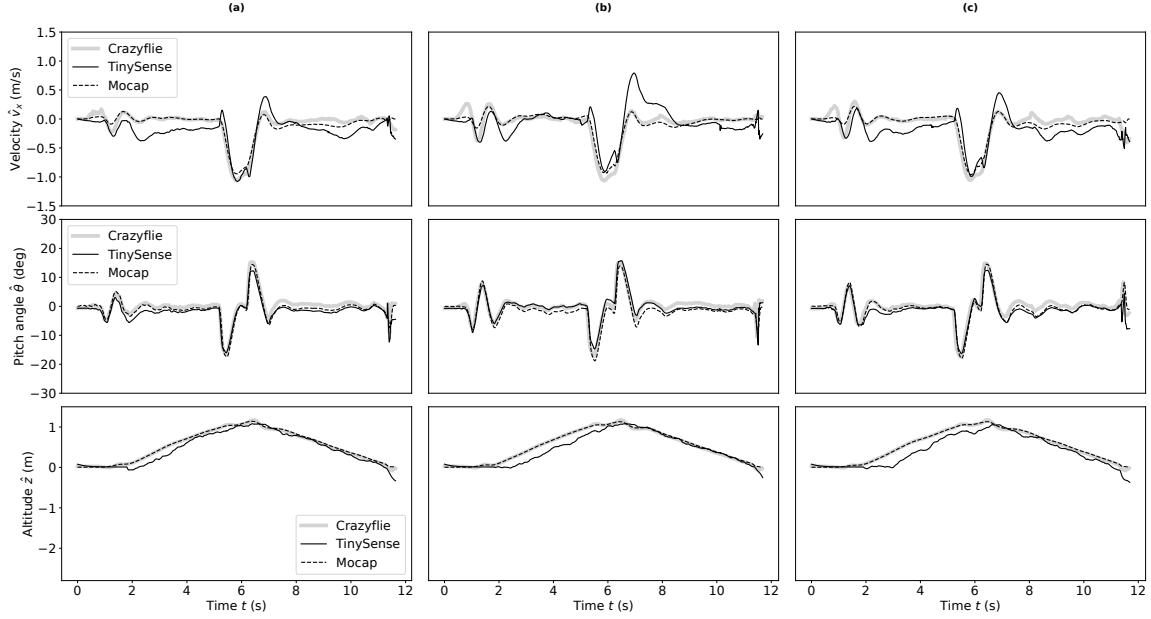


Fig. 5: Comparison between state estimates from TinySense’s Kalman Filter, Crazyfly, and mocap. We conducted three flight experiments (a), (b), and (c), with the Kalman filter estimates starting at time zero. Pressure sensor measurements were ignored during the first 1.8 s because pressure changes due to the ground effect at takeoff confound readings.

TABLE II: Estimate of optic flow algorithm power use

update frequency (Hz)	occurrences/update	task	single-cycle operations	int division operations	floating-point division operations	total cycles/update	total cycles/s (Mhz)	estimated power (mW)
100	1	skip frame	8493	0	8	249781	25.0	5.5
	12	LK OF	18434	128	2			
	1	copy data	1200	0	0			

TABLE III: Root Mean Squared Error (RMSE) of estimates relative to mocap system

	velocity v_x (m/s)	pitch angle θ (deg)	altitude z (m)
Crazyfly	0.075 ± 0.009	1.619 ± 0.267	0.021 ± 0.001
TinySense	0.186 ± 0.015	1.573 ± 0.166	0.136 ± 0.026

enough to fly aboard flying insect robots (FIRs) weighing less than a gram. Our proposed avionics suite diverges in two important ways from the 28 g Crazyfly helicopter, the lightest vehicle yet to perform controlled hovering, and previous work [15]. First, we replaced a power-hungry laser rangefinder with a much more efficient (and slightly less ideal) and lighter pressure sensor. Second, we replaced an off-the-shelf optic flow sensor, which is available only in relatively heavy packages, with a custom camera with a global shutter and a custom-written Lucas-Kanade-based optic flow algorithm running onboard a 10 mg microcontroller small enough to fly onboard an FIR. We estimate the compute power usage is about 5 mW. Our system uses a linear Kalman Filter to estimate pitch angle, translational velocity, and altitude. This can be readily adapted to a nonlinear extended Kalman Filter for more generality [15].

The cumulative weight of this sensor suite is only 78.4 mg, rather less than the 187 mg of previous work [15]. With the addition of a 30 mg microcontroller and crystal, the overall system remains well within the 252 mg payload

limit of the 143 mg robotic platform reported in [20]. Our system demonstrates average RMSE values of 0.186 m/s for velocity, 1.573 deg for pitch angle, and 0.136 m for altitude when compared to the motion capture system. In comparison, the Crazyfly platform achieves RMSE values of 0.075 m/s, 1.619 deg, and 0.021 m for the same states during similar flight maneuvers. This shows that our system performs closely to the state-of-the-art Crazyfly in these scenarios. The pressure sensor showed negligible drift over a 60-second period, providing a usable altitude estimate as long as its output during takeoff is ignored. Future improvements include mitigating the lateral velocity errors that occur during rapid rotations that exceed the limit of Lucas-Kanade estimation (Section V-E) by increasing the frame rate or image blurring. Additionally, our patch-based optic flow estimation (Section III-A) may allow outliers to be rejected based on their eigenvalues to eliminate low-contrast patches, reducing noise [38].

Future work will integrate TinySense along with a microcontroller and power hardware onto insect-scale robots such as the Robofly [39] to perform visual navigation [40]. The Robofly performs faster accelerations than the Crazyfly and is a different vibration environment, but is otherwise similar. Previous work shows vibrations do not confound gyroscope readings [41], [16]. Algorithmic changes in optic flow may be needed to accommodate vibration, but our choice of a global shutter camera will mitigate tearing artifacts.

REFERENCES

- [1] K. Y. Ma, P. Chirarattananon, S. B. Fuller, and R. J. Wood, "Controlled flight of a biologically inspired, insect-scale robot," *Science*, vol. 340, no. 6132, pp. 603–607, 2013.
- [2] S. Kim, Y.-H. Hsiao, Y. Chen, J. Mao, and Y. Chen, "Firefly: An insect-scale aerial robot powered by electroluminescent soft artificial muscles," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6950–6957, 2022.
- [3] J. M. James, V. Iyer, Y. M. Chukewad, S. Gollakota, and S. B. Fuller, "Liftoff of a 190 mg laser-powered aerial vehicle: The lightest wireless robot to fly," in *Robotics and Automation (ICRA), IEEE Int. Conf.*, 2018, pp. 1–8.
- [4] 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, pp. 491–495, 2019.
- [5] T. Ozaki, N. Ohta, T. Jimbo, and K. Hamaguchi, "A wireless radiofrequency-powered insect-scale flapping-wing aerial vehicle," *Nature Electronics*, vol. 4, no. 11, pp. 845–852, 2021.
- [6] S. Fuller, Z. Yu, and Y. P. Talwekar, "A gyroscope-free visual-inertial flight control and wind sensing system for 10-mg robots," *Science Robotics*, vol. 7, no. 72, p. eabq8184, 2022.
- [7] W. S. N. Trimmer, "Microbots and micromechanical systems," *Sensors and Actuators*, vol. 19, pp. 267–287, 1989.
- [8] G. Caprari, T. Estier, and R. Siegwart, "Fascination of down scaling—alice the sugar cube robot," *Journal of micromechatronics*, vol. 1, no. 3, pp. 177–189, 2001.
- [9] J. J. Abbott, Z. Nagy, F. Beyeler, and B. J. Nelson, "Robotics in the small, part i: microrobotics," *IEEE Robotics & Automation Magazine*, vol. 14, no. 2, pp. 92–103, 2007.
- [10] I. Paprotny and S. Bergbreiter, "Small-scale robotics: An introduction," in *workshop at the IEEE International Conference on Robotics and Automation*. Springer, 2013, pp. 1–15.
- [11] D. Floreano and R. J. Wood, "Science, technology and the future of small autonomous drones," *Nature*, vol. 521, no. 7553, pp. 460–466, 2015.
- [12] H. D. Escobar-Alvarez, N. Johnson, T. Hebble, K. Klingebiel, S. A. P. Quintero, J. Regenstein, and N. A. Browning, "R-advance: Rapid adaptive prediction for vision-based autonomous navigation, control, and evasion," *Journal of Field Robotics*, vol. 35, no. 1, pp. 91–100, 2018. [Online]. Available: <http://dx.doi.org/10.1002/rob.21744>
- [13] D. Palossi, A. Loquercio, F. Conti, E. Flamand, D. Scaramuzza, and L. Benini, "A 64mw dnn-based visual navigation engine for autonomous nano-drones," *IEEE Internet of Things Journal*, 2019.
- [14] M. B. Kjærgaard, H. Blunck, T. Godsk, T. Toftkjær, D. L. Christensen, and K. Grønbæk, "Indoor positioning using gps revisited," in *Pervasive Computing: 8th International Conference, Pervasive 2010, Helsinki, Finland, May 17–20, 2010. Proceedings 8*. Springer, 2010, pp. 38–56.
- [15] Y. P. Talwekar, A. Adie, V. Iyer, and S. B. Fuller, "Towards sensor autonomy in sub-gram flying insect robots: A lightweight and power-efficient avionics system," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 9675–9681.
- [16] A. Naveen, J. Morris, C. Chan, D. Mhrous, E. F. Helbling, N.-S. P. Hyun, G. Hills, and R. J. Wood, "Hardware-in-the-loop for characterization of embedded state estimation for flying microrobots," *arXiv preprint arXiv:2411.06382*, 2024.
- [17] N. O. Pérez-Arancibia, K. Y. Ma, K. C. Galloway, J. D. Greenberg, and R. J. Wood, "First controlled vertical flight of a biologically inspired microrobot," *Bioinspiration and Biomimetics*, vol. 6, no. 3, p. 036009, Sep 2011. [Online]. Available: <http://dx.doi.org/10.1088/1748-3182/6/3/036009>
- [18] D. S. Drew and K. S. Pister, "First takeoff of a flying microrobot with no moving parts," in *Manipulation, Automation and Robotics at Small Scales (MARSS), 2017 International Conference on*, 2017, pp. 1–5.
- [19] M. Karásek, F. T. Muijres, C. De Wagter, B. D. Remes, and G. C. de Croon, "A tailless aerial robotic flapper reveals that flies use torque coupling in rapid banked turns," *Science*, vol. 361, no. 6407, pp. 1089–1094, 2018.
- [20] S. B. Fuller, "Four wings: An insect-sized aerial robot with steering ability and payload capacity for autonomy," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 570–577, 2019.
- [21] S. B. Fuller, Z. E. Teoh, P. Chirarattananon, N. O. Pérez-Arancibia, J. Greenberg, and R. J. Wood, "Stabilizing air dampers for hovering aerial robotics: design, insect-scale flight tests, and scaling," *Autonomous Robots*, February 2017. [Online]. Available: <http://dx.doi.org/10.1007/s10514-017-9623-3>
- [22] H. K. Hari Prasad, R. S. Vaddi, Y. M. Chukewad, E. Dedic, I. Novoselov, and S. B. Fuller, "A laser-microfabricated electrohydrodynamic thruster for centimeter-scale aerial robots," *PloS one*, vol. 15, no. 4, p. e0231362, 2020.
- [23] Z. E. Teoh, S. B. Fuller, P. C. Chirarattananon, N. O. Pérez-Arancibia, J. D. Greenberg, and R. J. Wood, "A hovering flapping-wing micro-robot with altitude control and passive upright stability," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ Int. Conf.*, Vilamoura, Algarve, Portugal, 7–12 October 2012, pp. 3209–3216.
- [24] 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.
- [25] Y. M. Chukewad, J. M. James, A. Singh, and S. B. Fuller, "Robofly: An insect-sized robot with simplified fabrication that is capable of flight, ground, and water surface locomotion," *IEEE Transactions on Robotics*, 2021, (in press).
- [26] R. J. Wood, "The first takeoff of a biologically inspired at-scale robotic insect," *IEEE Trans. Robotics*, vol. 24, no. 2, pp. 341–347, 2008.
- [27] A. Beyeler, "Vision-based control of near-obstacle flight," Ph.D. dissertation, ECOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE, 2009.
- [28] G. K. Taylor and H. G. Krapp, "Sensory systems and flight stability: What do insects measure and why?" in *Insect Mechanics and Control*, ser. Advances in Insect Physiology, J. Casas and S. Simpson, Eds. New York: Academic Press, 2007, vol. 34, pp. 231–316. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0065280607340058>
- [29] J. Serres and F. Ruffier, "Optic flow-based robotics," 2016.
- [30] J. R. Serres and F. Ruffier, "Optic flow-based collision-free strategies: From insects to robots," *Arthropod structure & development*, vol. 46, no. 5, pp. 703–717, 2017.
- [31] J. J. Koenderink and A. J. van Doorn, "Facts on optic flow," *Biological Cybernetics*, vol. 56, pp. 247–254, 1987.
- [32] B. D. Lucas, T. Kanade, *et al.*, "An iterative image registration technique with an application to stereo vision." in *IJCAI*, vol. 81, 1981, pp. 674–679.
- [33] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, pp. 185–203, 1981.
- [34] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.
- [35] C. D. Wagter, S. Tijmons, B. Remes, and G. de Croon, "Autonomous flight of a 20-gram flapping wing mav with a 4-gram onboard stereo vision system," in *Robotics and Automation (ICRA), 2014 IEEE Int. Conf.*, Hong Kong, China: IEEE, 2–5 June 2014.
- [36] S. Fuller, B. Greiner, J. Moore, R. Murray, R. van Paassen, and R. Yorke, "The python control systems library (python-control)," in *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE, 2021, pp. 4875–4881.
- [37] M. W. Mueller, M. Hamer, and R. D'Andrea, "Fusing ultra-wideband range measurements with accelerometers and rate gyroscopes for quadrocopter state estimation," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 1730–1736.
- [38] J. L. Barron, D. J. Fleet, S. S. Beauchemin, and T. Burkitt, "Performance of optical flow techniques," in *Proceedings 1992 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 1992, pp. 236–237.
- [39] 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)*. IEEE, 2018, pp. 3587–3594.
- [40] Z. Yu, G. Zardini, A. Censi, and S. Fuller, "Visual confined-space navigation using an efficient learned bilinear optic flow approximation for insect-scale robots," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 4250–4256.
- [41] S. B. Fuller, E. F. Helbling, P. Chirarattananon, and R. J. Wood, "Using a MEMS gyroscope to stabilize the attitude of a fly-sized hovering robots," in *Micro Air Vehicle (IMAV), 2014 Int. Conf.*, Delft, The Netherlands, August 12–15 2014.