Optical Flow and IMU Fusion for Drone Horizontal Velocity Control

DJEDJIGA BELFADEL
JOHN CAIN
DAVID HAESSIG
CHERIF CHIBANE

This paper presents a vision-aided, horizontal velocity control system for unmanned aerial vehicles (UAVs). Angular velocity data provided by an Inertial Measurement Unit (IMU) is combined with pixel displacement data given by an optical flow sensor to produce a measure of horizontal velocity in two dimensions. Robust velocity control is achieved without reliance on the Global Positioning System (GPS). To validate the proposed approach, a series of dynamic simulations and hardware experiments were conducted. The hardware tests were designed to characterize the output of the optical flow sensor as a function of horizontal velocity and altitude. This data was then used to formulate an optical flow sensor mathematical model that, to our knowledge, is new and has not appeared in any prior optical flow sensor-related literature. This model is shown to produce output data that is consistent with the experimentally measured sensor data gathered under equivalent conditions. In addition, this model has been used to create the optical flow sensor model employed when simulation testing a UAV's controlled dynamic motion. Simulation results demonstrate that this method successfully achieves accurate and reliable control of a UAV's velocity in the horizontal plane.

Manuscript received March 28, 2024; revised June 19, 2024; released for publication January 10, 2025

Refereeing of this contribution was handled by Mattia Brambilla.

D. Belfadel and J. Cain are with the Department of Electrical Engineering, Fairfield University, Fairfield, CT 06824 USA (e-mail: dbelfadel@fairfield.edu; john.cain@student.fairfield.edu).

D. Haessig and C. Chibane are with the AuresTech Inc., Bridgewater, NJ 08807 USA (email: dave@aurestech.com; cherif.chibane@aurestech.com).

1557-6418/2024/$17.00 © 2024 JAIF

I. INTRODUCTION

Localization is the process of determining an object's position and possibly also its orientation in space. It is related to navigation and is a process that is critical in aerospace and robotics applications [1]. A similar and less stringent process is that of location stabilization, which involves the control of an object's velocity and stabilization of its location, but does not also require knowledge of the object's location. The process targeted by the work presented in this paper is that of location stabilization.

To perform localization and location stabilization, robots typically estimate their velocity and position by numerically integrating acceleration data provided by an inertial measurement unit (IMU) [2], [3]. This IMU data contains, however, measurement noise and sensor bias errors that can significantly degrade the accuracy of these estimates [4]. The numerical integration processes involved in generating the estimates can also introduce additional error, which further degrades accuracy [5]. Euler integration, for example, causes truncation errors similar to Taylor series approximations [6]. These and other sources of error can lead to the deterioration of the velocity and position signals required for accurate localization and stabilization. Over time, these errors can accumulate, resulting in significant inaccuracies that affect the system's performance [7]. The impact of these drift errors can be mitigated by adding sensors that are able to observe the drift. A wide range of sensors can be used for this purpose, one being the optical flow sensor (OFS).

Cameras in unmanned aerial vehicles (UAVs) can aid in localization through object recognition and tracking, utilizing landmarks. Smart cameras and applicable software libraries will calculate motion from frame changes and use digital signal processing to detect changes in shadow levels in each pixel [8]. These approaches, however, are computationally demanding, and issues like unknown obstacles or lighting changes can cause data inaccuracies. To address these, SLAM (simultaneous localization and mapping) uses multiple cameras and Light Detection and Ranging (LIDAR) sensors to handle environmental lighting changes [9]. SLAM combines multiple sensors to generate a 3D map of the surroundings [10], offering accurate position estimation but at a high computational cost. Self-maneuvering UAVs must also be capable of autonomous landing. Several researchers, such as [8] and [11], have investigated the use of cameras and vision-aided systems for autonomous landing by fusing data from cameras, OFS, and IMUs without the need for altimeters like LIDAR or ultrasonic sensors. These studies demonstrate that vision-based systems can accurately measure both altitude and horizontal velocity, which is expected to improve landing performance. In contrast, our work does not involve a camera and instead relies on an altimeter, showing that horizontal velocity can be measured effectively using only the OFS, IMU, and altitude sensor. Additionally, this sensor configuration requires significantly fewer computational resources than vision-based systems. OFS provide a low cost and computationally tractable approach for drone location stabilization, platform navigation [12], [13], and to perform position and velocity control, as done in this paper and in [14]. Their effectiveness as a sensor for short-term stabilization has been established [15], but their applicability for long-term positioning is still under active research. Studies assessing the potential of OFS for position estimation, particularly when fused with other sensor systems, are found in [13] and [16].

This paper presents a new mathematical model for an OFS (the Truth Model) and compares this model to experimentally measured sensor data captured over a range of operating conditions that would be typical when the OFS is mounted on a drone, i.e., limited altitude, limited velocity, and limited roll and pitch tilt angles, i.e., less than 25°. This new OFS model is shown herein to provide good results for several test cases that fall within this operating envelope.

This paper also presents a horizontal velocity control system design for rotary wing UAVs based on the new OFS model. This design involves a velocity sensor equation that combines the output of the OFS with angular rate data given by an IMU, and from these produces a measure of the drone platform horizontal translational velocity in two dimensions. A simulation study is performed to assess the impact that OFS pixel quanization has on control system performance. The velocity sensor and control design presented herein assumes that platform's altitude above the ground is known, and available as an input to the velocity sensor. Altitude data is assumed to be provided, for example, by a separate device such as a sonar or radar altimeter. Two single-axis models are applied to generate the 2D output from a single two-axis OFS.

To assess the OFS's response to changes in the physical environment, we constructed a linear rail setup to observe the OFS's output at various sensor velocities and altitudes. Experiments were also conducted to determine the sensor's resilience to environmental fluctuations, varying light levels, and surface textures. These test results are reported herein, and the results are used to validate the new OFS model proposed herein.

A Simulink model of the physical plant (i.e., a fixed-wing UAV), the proposed velocity sensor, the OFS, and the Control Law were constructed to evaluate this system's performance. It is a single-axis, continuous-time dynamic model of a UAV's horizontal motion and includes an IMU model, a hybrid (continuous-time, discrete-time) model of an OFS, and a discrete-time proportional–integral (PI) controller. Performance has been assessed in three scenarios: a step response in commanded velocity, a response to a wind gust, and the tracking of very slow velocity commands. Each scenario demonstrates an important drone control quality that is facilitated by the OFS. The results demonstrate (a) the rapid response rise time of approximately 1 s, (b) the detection and rejection of location perturbations caused by wind, and (c) the control of velocity at very low velocity setpoint levels. These results indicate that the control design approach proposed here is an effective method for controlling a drone's horizontal velocity.

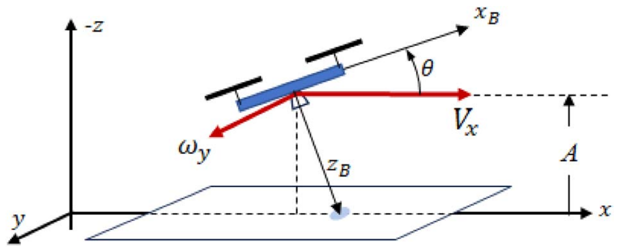


Figure 1. Scene velocity as a function of platform linear and angular velocities.

Section II outlines the problem formulation and fundamental OFS equations, detailing the OFS Truth Model and Velocity Sensor Model for control. Section III, Experimental Setup, covers the linear velocity rail system, IMU, and PMW3901 OFS. Section IV, Simulated and Experimental OFS Results and Comparison, compares experimental results with simulations, including constant velocity tests, oscillatory motion, and additional testing. Section V discusses the control design, plant dynamics, and simulation results, including step response, wind gust response, and slow movement. Finally, Section VI presents conclusions.

II. PROBLEM FORMULATION AND SENSOR MODEL

The problem under consideration is illustrated in Fig. 1, which portrays a drone equipped with an OFS assembly at an altitude A, moving with horizontal velocity V_x, and rotating with angular velocity ω_y about the y axis. The platform's translational motion is limited to motion along the x axis, and the pitch angle θ is assumed to be small, i.e., less than 25° in magnitude. The sensor's field of view (FOV) is centered at the location where the z_B and x axes intersect. The platform body coordinate frame contains axes x_B and z_B (y_B is not shown).

The pinhole model of the image motion observed by an OFS, given in [15], is often used to mathematically characterize the output of this device. This model defines the instantaneous velocity of the image falling on the image plane at a pixel located at (x, y) on the image plane with focal distance f. This pixel velocity field, also called the optical flow field, is represented symbolically as $P(x, y, f) = \{\delta_x, \delta_y\}$, where δ_x and δ_y are the components of the optical flow velocity vector on the image plane at location (x, y). The detailed derivation of these linear, continuous-time velocities is provided in [15] and expressed as:

$$\delta_x = \frac{V_z x - V_x f}{A} - \omega_y f + \omega_z y + \frac{\omega_x xy - \omega_y x^2}{f}, \qquad (1)$$

$$\delta_y = \frac{V_z y - V_y f}{A} + \omega_x f - \omega_z x + \frac{\omega_x y^2 - \omega_y xy}{f}, \quad (2)$$

where A is the distance from the OFS to the ground, i.e., the altitude. As noted in [15], however, an OFS produces a single pair of values at every sample time and not the velocities being observed at each pixel over the field of the image plane. An OFS device and processing algorithm generate, in a sense, the mean of the optical flow field in a central neighborhood of pixels around the center point at pixel $P(0, 0, f)$. Thus, it is sensible to compute the output of an OFS device, as noted in [15], using the velocity values given by (δ_x, δ_y) at the point $P(0, 0, f)$. After substituting this point into equations (1) and (2), we find:

$$\delta_x = -\frac{V_x f}{A} - \omega_y f, \quad (3)$$

$$\delta_y = -\frac{V_y f}{A} + \omega_x f, \quad (4)$$

Noting that δ_x and δ_y are linear velocities of a pixel at an orthogonal distance f, we can divide equations (3) and (4) through by f and use angular velocities $\Omega_x = \frac{\delta_x}{f}$ and $\Omega_y = \frac{\delta_y}{f}$:

$$\Omega_x = -\frac{V_x}{A} - \omega_y, \quad (5)$$

$$\Omega_y = -\frac{V_y}{A} + \omega_x. \quad (6)$$

This allows us to work in terms of the angular velocity of the OFS body, a more natural and physically observable quantity associated with OFS motion rather than the linear translational optical flow present on the image plane. Here, equation (6) applies to Fig. 1, while equation (5) pertains to translational motion along the y axis and is not pictured. The problem formulation for that case is analogous to that of the x axis but involves the assembly's velocity in the y direction and roll angular velocity about the x axis. These are the equations we carry forward into the development of the hybrid continuous-time and discrete-time model of the OFS output. Reference [15] also includes a discrete-time OFS model. A key difference between that model and the one being presented here is that this new model incorporates the finite resolution and quantization associated with the pixels. Additionally, our model computes the quantized angular displacement observed by the sensor and the change in image angle from one image frame to the next. Thus, it does not quantize the instantaneous angular velocity appearing near the center of the image frame, but quantizes the integrated, continuous-time velocity (i.e., accumulated angular displacement up to the time when the most recent image frame is captured).

A. OFS Truth Model

The model created to represent the OFS output is one that involves the integral of the angular rate of the scene. Since both equations (5) and (6) are very similar, we will continue with only equation (5), dropping the subscripts for clarity. Equation (5) is then:

$$\Omega = -\omega - \frac{V}{A}, \quad (7)$$

where Ω is the scene angular velocity, ω is the angular velocity of the sensor body, V is the sensor translational velocity, and A is the sensor distance (or altitude) from the scene being observed (i.e., the ground). Integrating this to produce an angle:

$$\phi(t) = \int_{t_0}^{t} \left(-\omega - \frac{V}{A}\right) dt, \quad (8)$$

where $\omega, V,$ and A are functions of the continuous time variable t.

To produce the sensor output, we integrate $\phi(t)$ to the frame time kT at frame index k, quantizing to the quantization step size Q, and generate the difference in the quantized result:

$$\phi(kT) = \int_{t_0}^{kT} \left(-\omega - \frac{V}{A}\right) dt, \quad (9)$$

$$X_k = \text{floor}\left(\frac{\phi_k}{Q}\right), \quad (10)$$

$$\Delta X_k = X_k - X_{k-1}. \quad (11)$$

This value, ΔX_k, is what comes out of the sensor to represent motion of the scene in one direction. It is an integer value and has units of counts. There is an equivalent equations for ΔY_k, representing motion in the other orthogonal direction.

B. Sensor Model for Conversion to Linear Velocity

To convert from the sensor output's change in scene angle ΔX_k to linear velocity, we integrate the scene angle over a single frame time period:

$$\Delta\phi_k = \int_{(k-1)T}^{kT} \left(-\omega - \frac{V}{A}\right) dt. \quad (12)$$

To generate this integrated result, we assume that the inputs are constant over the very short time period T:

$$\Delta\phi_k = -\omega_k T - \frac{V_k}{A_k} T. \quad (13)$$

The angular rate ω_k is the pitch angular velocity and is provided by the IMU. The altitude A_k is assumed to be known, provided by an altitude sensor, which can be a barometric altimeter's measure of the distance above the take-off altitude, assuming that the ground is relatively flat. Equation (11) is converted from units of counts to units of radians (angle) using the quantization step size:

$$\Delta\phi_k = \Delta X_k Q. \quad (14)$$

Table I
Linear Rail Test Bed Hardware Component

Component type	Specifications
PMW3901MB (2-axis optical flow sensor)	Measurement range: 80 mm-infinity, 4-Wire SPI @ 2 MHz
HC-SR04 (Ultrasonic ranging sensor)	Measurement range: 2 cm–400 cm, eight 40 kHz signal transmit and detect.
2× Atmega328P (Arduino)	Low power AVR 8-bit microcontroller, 32K bytes of progammable flash memory, and support of SPI/I2C communication.
DRV8825 (stepper motor driver IC)	Rated for 2.5 A max current, and a resolution of 32 microsteps.
ZD-0029 R REIFENG (limit switch)	Connect VCC (ramps of +); Black line: Connect GND(ramps of -); Green line: Connect SIGNAL(ramps in s).
Nema 17 stepper motor	Bipolar stepper, Step Angle: 1.8deg, with a holding torque of 45 Ncm (63.74oz.in)
AC/DC power supply	110/220 V AC in, adjustable up to 12 V 30 A out.

Combining equations (13) and (14) and solving for velocity V_k:

$$V_k = \left[-\omega_k - \Delta X_k \left(\frac{Q}{T}\right)\right] A_k. \tag{15}$$

This is the equation that will be used to produce a velocity measurement that will serve as the feedback signal for velocity control.

The OFS is an imaging device that generates pixel flow measurements from the scene in view. By comparing pixel content between successive frames of an image, the direction and speed of motion of the image are determined within the sensor. Consider the case in which the scene is moving by several pixels in distance from one frame to the next, with a frame capture time of 20 ms (i.e., 50 Hz). The delta-angle output produced by the sensor is an integer count value representing the angular rotation measured between the two frame images, quantized to an integer representing the pixel size. The fractional part of the motion is removed from the output by quantization. When the motion is slow enough so that it takes more than 20 ms to move a signal pixel, then the output data toggles between 0 and 1 with a duty cycle that depends on velocity. If motion is very slow so that it takes several frames, e.g., 10, to move by a single pixel, then the output will sit at zero (0) counts until the scene finally reaches a distance of one pixel, and at that sample time it will output a change of 1 pixel, then it will drop back to outputting zeros while moving to the next pixel quantum distance.

The OFS's quantization of angular position has an advantage over that of a sensor that measured angular velocity; the OFS does not introduce an accumulating error due to quantization. A velocity sensor, on the other hand, could experience an angular rate that remains within one quantum, and that motion would never be evident in the sensor's output. The OFS instead is sensing angular displacement, or the integral of the angular rate, and detecting the change in that angle from frame to frame, so that even the smallest angular rate input is eventually detected when that motion reaches an angular displacement of one pixel. The sensor outputs a count value of 1 when the distance accumulated exceeds another quantization step size. When the sensor is moving slowly and therefore the scene is moving slowly over the focal plane, the sensor produces a stream of 0's punctuated with a periodic value of 1 count that occurs when the distance traversed reaches a distance equaling one pixel.

III. EXPERIMENTAL SETUP

A. Linear Velocity Rail System

We evaluated the PMW3901 sensor's ability to adapt to environmental changes in light intensity, surface texture, and altitude using a linear rail system with 80/20 aluminum extrusion bars, powered by a Nema 17 stepper motor and a DRV8825 stepper driver.

The PMW3901 collects 2D displacement data, and an HC-SR04 ultrasonic sensor is used to measure altitude. This setup is most effective for distances under 5 m. A powered rail sliding mechanism replicates linear flight along a single axis using 80/20 aluminum extrusion rods and a stepper motor-powered belt drive. Two Atmega328P (Arduino) microcontrollers handle data collection and slider motion. The stepper motor, driven by a DRV8825 driver IC and powered by a 12V, 30A power supply, determines the slider position using a limit switch for homing the slider and tracking individual stepper motor steps. The setup can be placed on sawhorses to simulate varying altitudes. The hardware components of the linear rail test bed are detailed in Table I, while the system layout is depicted in Fig. 2.

B. Inertial Measurement Unit

An IMU measures the linear accelerations and angular rotational rates that occur along a coordinate frame that is attached to the IMU body. This body frame rotates with the IMU itself. Since the IMU senses 3 axes of rate and acceleration, it is referred to as a 6-axis sensor. A model of the IMU can be found in [17]. This IMU was not integrated with the experimental test setup but will be used in hardware testing that will involve both

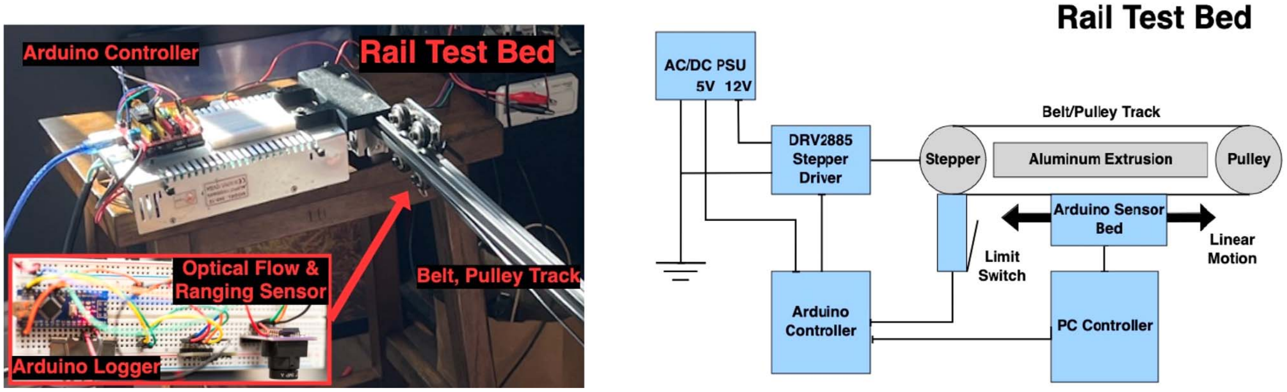


Figure 2. Linear rail system with stepper-driven belt drive.

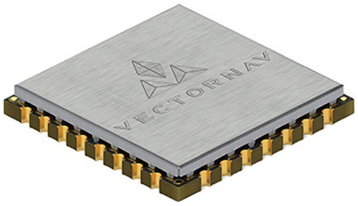


Figure 3. The VN-100 IMU/AHRS by VectorNav technologies.

the OFS and IMU. It is described here, therefore, for reference.

1) VN-100 IMU/AHRS Sensor Specifications: The VN-100, manufactured by VectorNav Technologies, shown in Fig. 3, is an example of an IMU that is commonly used in applications such as UAVs, drones, and robotics. Tables II and III provide details of its performance specifications and attitude/heave capabilities. The VN-100's compact size and low power consumption make it an attractive option for use in small or portable devices where space and power are limited. Linear accelerations measured by the IMU must be transformed from the IMU body to a navigation frame, corrected for gravity, and integrated to produce platform velocity and position. As noted, IMU sensor noise causes the dead-reckoning navigation solution to deviate from the correct solution with a random walk type of error that grows with time and is unbounded. To constraint the growth of these drift errors, an aided inertial navigation algorithm will utilize measurement data provided by additional sensors to improve or aid the solution produced by the IMU alone. Sensor data may include, for example, optical flow, measurement of platform ground speed, air speed, altitude above ground, range to a specific location, etc.

Table III
The VN-100 IMU Attitude and Heave Performance

Specification	Value
Range (heading/yaw, roll)	±180°
Range (pitch)	±90°
Heading (magnetic)	2.0° RMS
Pitch/roll (static)	0.5° RMS
Pitch/roll (dynamic)	1.0° RMS
Heave accuracy	5% or 5 cm
Delayed heave accuracy	2% or 2 cm
Angular resolution	0.001°

C. OFS Model

The Pimoroni PMW3901 OFS, shown in Fig. 4, is an example of an embedded light-tracking optical sensor. It captures a collection of pixel displacement values known as an optical flow field, and then averages this field over a region at the center of the sensor's Field of View, producing what is designated a pixel vector with x and y components. The Pimoroni sensor data sheet indicates that the sensor has a Field of View (FOV) of 42°, (±21°). Detailed specifications of the PMW3901 Optical Flow Sensor are provided in Table IV. From a series of tests at varying velocities during which the sensor pixel vector outputs were captured, it was observed that the PMW3901 must move a minimum of 0.0015 radians to register a new output with a change of 1 pixel count. The flow vectors produced within the sensor in either the x or

Table II
Sensor Performance Specifications

Specification	Accelerometer	Gyroscope	Magnetometer	Barometer
Range	±16 g	±2000°/s	±2.5 Gauss	10–1200 mbar
In-run bias stability	<0.04 mg	<10°/h	-	-
Noise density	0.14 mg/$\sqrt{\text{Hz}}$	0.0035°/s/$\sqrt{\text{Hz}}$	140 μGauss/$\sqrt{\text{Hz}}$	-
Bandwidth	260 Hz	256 Hz	200 Hz	200 Hz
Cross-axis sensitivity	±0.05°	<0.05°	±0.05°	-

Figure 4. The Pimoroni PMW3901 optical flow sensor.

y direction must separately average to this quantization level to produce a new pixel count of 1. A key feature of the sensor is that it is not measuring velocity but is measuring a change in the location of the scene.

IV. SIMULATED AND EXPERIMENTAL OFS RESULTS AND COMPARISON

A. Constant Velocity Tests at Fixed Altitudes

This section contains the results of the experiments performed to assess the PMW3901 sensor's ability to provide useful velocity and displacement measurement data at various speeds and over a range of fixed altitudes. This evaluation was primarily focused on understanding the sensor's performance when altitude is known, with the aim of gauging the reliability of its raw data output. Tests were conducted on the linear rail system described above. In this first set of tests, the sensor was moved at a constant velocity from one end of the rail system to the other. Sensor output data was captured for a brief period before motion began and ended after motion stopped at the other end. Hardware tests and simulations were run at four velocities: 0.05, 0.1, 0.2, and 0.3 m/sec, all with a sampling frequency of 50 Hz and all at a distance of 0.66 m between the OFS and the ground scene. These results are given in Fig. 5.

For an input velocity setpoint of 0.05 m/s on the rail system, the OFS model produces an output having a mean value of 1.016 counts/sample when motion is occurring (4.7–15.4 s). In all four cases, the mean "counts/sample" values shown on each plot were computed over the time periods of OFS motion. Multiplication by the sample rate of 50 samples/s produces the sensor's Output Rate in counts/s, the third column of Table V. For example: 1.016 counts/sample × 50 samples/s = 50.8 counts/s.

Table IV
Specifications of the PMW3901 Optical Flow Sensor

Specification	Value
Field of view (FOV)	42° (±21°)
Minimum radian change for output	≈ 0.0015 radians
Frame time	20 ms (50 Hz)
Quantization level for output	1 pixel count
Motion detection type	Angular displacement

Table V
Sensor Raw Output Parameters and Derived Quantization Step Size

Velocity (m/s)	True angular rate Ω (radians/s)	Output rate (counts/s)	Quant step—Q (radians/count)
0.05	0.076	50.8	0.001501
0.1	0.152	102.5	0.001483
0.2	0.303	199	0.001522
0.3	0.455	294	0.001548

To generate the quantization step size Q, we use the relation:

$$Q \cdot (\text{Output Rate}) = \Omega$$
$$\frac{\text{radians}}{\text{count}} \cdot \frac{\text{counts}}{\text{s}} = \frac{\text{radians}}{\text{s}}. \tag{16}$$

The true angular rate Ω is computed with equation (5) with $\omega_y = 0$ (the OFS was not rotating):

$$\Omega_x = -\frac{V_x}{A}. \tag{17}$$

For an altitude of 0.66 m and the velocity setpoint values of column 1, negated since the rail system velocities were in the negative direction, one computes the quantization step size given in column 4, which differ slightly due to random error present in the test setup.

The mean Quantization Step Q for the PMW3901 computed by averaging these values is thus 0.0015 radians/count or 0.086 deg/count. This value has been used in the optical sensor model as well as in the computation of platform velocity from sensor output data. The jumpiness in this data is due in part to the pixel quantization; however, quantization error does not account for all of the "noise" that is observed. It is conjectured that this jumpiness represents real velocity perturbations caused by stepper motor velocity jitter and imperfections in the test setup. In addition, there may be additive measurement error contributed by the sensor itself. Our suspicion, however, is that the hashiness in the measured OFS output data is caused primarily by actual velocity perturbations. The OFS output can be used to measure translation displacement by summing the ΔX_k values. This was done for all 4 velocity test cases, with the results shown here in Fig. 6.

These summed outputs reach values of 526, 521, 506, and 504 counts for the four velocities of 0.5, 1.0, 2.0, and 3.0, respectively. Converting from counts to distance using the formula $QA\left(\sum \Delta X_k\right)$ produces a set of distance measurements ranging from 0.52 to 0.05 m, a difference of 4%.

B. Comparison to Simulation

The OFS Truth Model developed in Section II and defined in equations (9)–(11) is shown here in Fig. 7 as a Simulink model in block diagram form.

The results produced by this model are compared to the experimental results plotted in Fig. 5 first by driv-

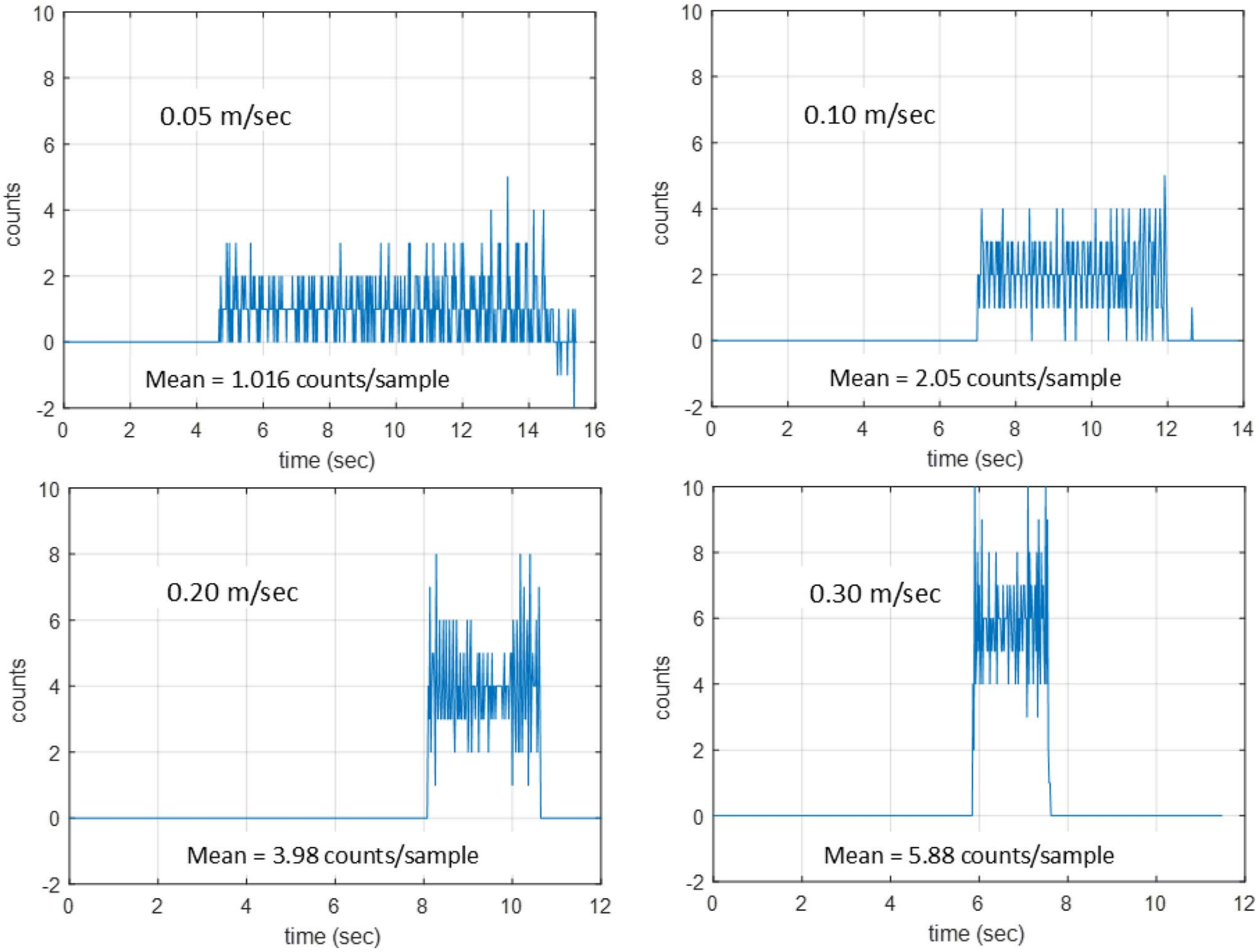


Figure 5. Optical flow sensor output on one axis during rail testing with approximately constant linear velocities.

ing the model with a purely constant velocity of 0.5 m/s, and second by driving the model with a velocity profile derived from the experimentally captured ΔX_k samples, creating a piecewise constant, continuous-time velocity signal as $\Delta X_k/\Delta T$. The results of the simulations are given in Fig. 8, with the constant velocity results on the left, and the experimentally measured velocity data on the right.

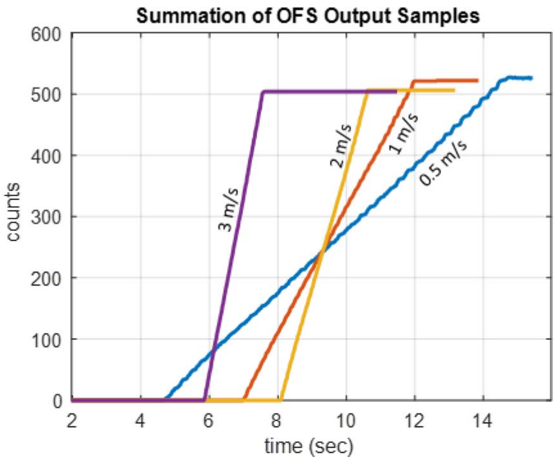


Figure 6. Running sum (discrete integration) of OFS output samples for four velocity setpoints.

For the constant velocity case, the OFS models' output is very regular and periodic, giving an output of 1 count most often, separated by an occasional output of 0. For the case on the right, in which the velocity is derived from the experimental data, the OFS model produces an output that precisely matches the output of the actual physical OFS device (see Fig. 5). This indicates that the model faithfully represents what is happening within the OFS in this scenario, i.e., constant velocity, low altitude, and zero angular velocity conditions. It should be noted that the new model produces results that match the output of the OFS when driving it with velocity time histories derived from the data of the other "constant" velocity cases. Since the simulated OFS outputs matched the experimental data of Fig. 5, they are not replotted here.

C. Test Results with Oscillatory Input Motion and Varying Altitude

Dependence of Sensor Output Flow on Altitude: Tests were performed to highlight the impact of altitude on the magnitude of flow vectors and the level of noise present

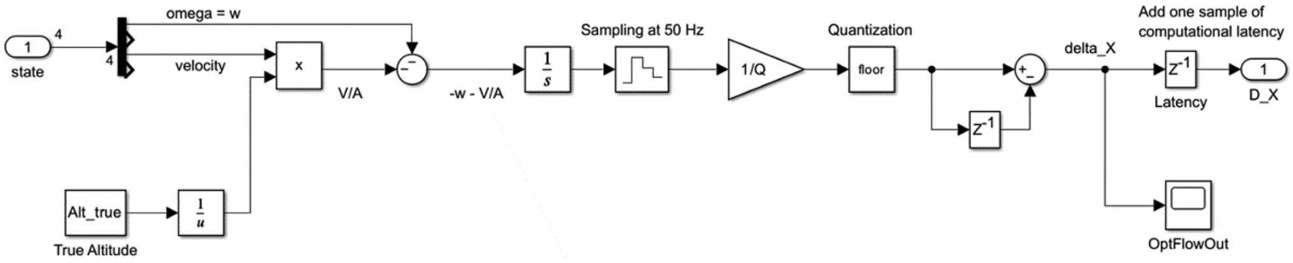


Figure 7. The OFS truth model.

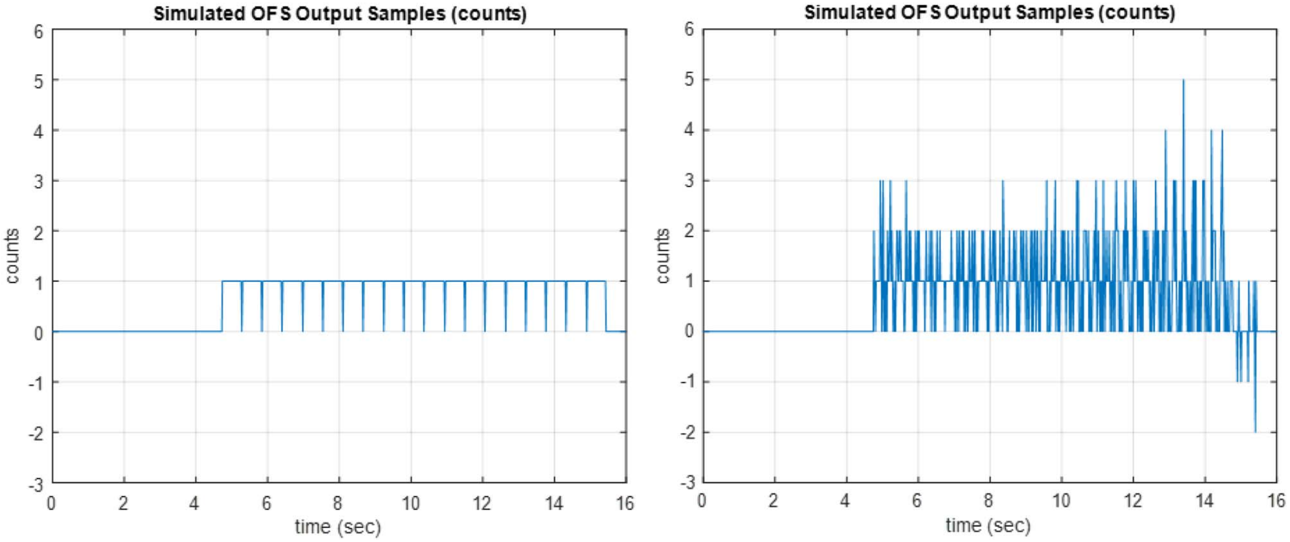
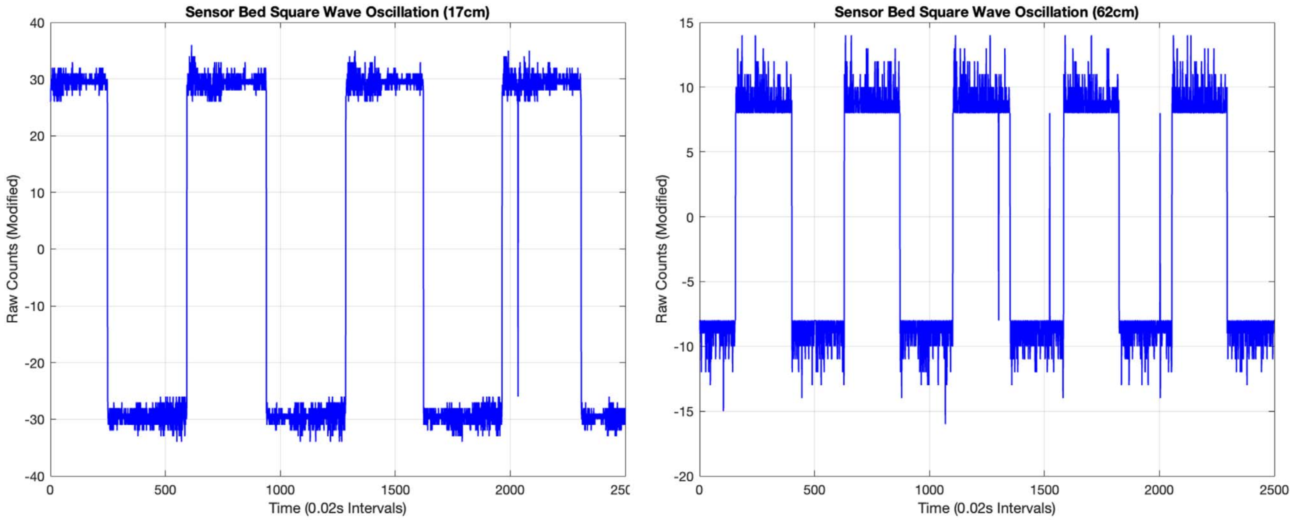


Figure 8. Comparison of OFS model outputs for constant and variable velocity, experimentally measured velocity data, velocity setpoint of 0.5 m/s.

in the data. These observations underscored the significance of altitude adjustments in accurately computing platform velocity. In this test, the rail system was moving in a triangular manner with directions of travel reversals separated by periods of constant velocity magnitudes of 0.02 m/s. The sample frequency was 25 Hz, so in a single sample, the slider moved 0.8 mm. At a distance of 17 cm above the ground, this equates to an angular change of 47 mrad ($8e^{-4}/17e^{-3}$). Given the quantization size $Q = 0.0015$ rad/count, this leads to an estimated ΔX_k optical flow count [see equation (11)] of $0.047/0.0015 = 31.4$ counts/sample. As noted in



(a) (b)

Figure 9. Sensor output data during periodic, constant velocity triangular wave motion at two altitudes, 17 and 62.5 cm.

Fig. 9a, the output is nominally equal to this expected output level. When elevated to 62.5 cm, the angular change becomes 12.8 mrad, and the output nominally 8.5 counts/sample. This is clearly reflected in the plot of Fig. 9(b).

D. Additional Characterization Tests

To better understand the impact of quantization and altitude, we expanded our experimental study to include other environmental variables to verify if consistent velocity readings are maintained at a fixed altitude.

1) Light Level Tests: Given the OFS's reliance on shadow changes for velocity determination, we examined how varying light conditions affect displacement accuracy. Two tests evaluated the PMW3901MB sensor's performance under different lighting: one at dusk and another in direct sunlight, both at 0.66 m altitude. The observed displacements were 0.59 m and 0.57 m, respectively, demonstrating the sensor's ability to produce comparable displacement readings across light levels.

2) Surface Quality Tests: The sensor's response to surface texture was also assessed, given the significance of landmarks and texture in displacement detection. Two scenarios were compared: one with a checkerboard-taped grid surface and another over a wooden floor. The displacement estimations were 0.61 m for the textured surface and 0.58 m for the nontextured, with the actual displacement set at 0.6 m. Interestingly, flow counts varied significantly between surfaces, with nontextured surfaces eliciting larger vector magnitudes due to fewer but more pronounced feature changes.

V. HORIZONTAL HOLD CONTROL DESIGN AND SIMULATION TESTING

A. Plant Dynamics

The OFS model developed above has been used in two ways. First, to create a Sensor Truth Model for use in simulation of the system involving the sensor. Secondly, to produce an algorithm for converting the OFS output signals to translational velocity measurements. These measurements in the x- and y-directions are used in generating the control signals that act to drive the drone velocity to zero, thereby holding the drone steady, or to some non-zero setpoint velocity provided by the pilot. A single-axis controller is developed and applied separately to each axis. One controls the translational motion of the drone in the Fwd-Rev direction, and the other controls the Left-Right translational motion.

To produce a dynamic model of the drone along a single axis, we assume that the platform has two degrees of freedom: translation along one axis (x-axis) and rotational about an axis that is perpendicular to the x-axis and horizontal (i.e. the pitch axis). A diagram of the drone is given below in Fig. 10. We assume the drone is not accelerating vertically, and therefore the weight of the drone is equal to the component of thrust in the vertical direction (F_V). The thrust of the props (F_T) has vertical component

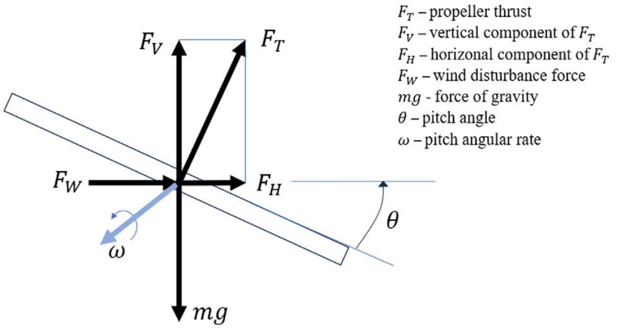

F_T – propeller thrust
F_V – vertical component of F_T
F_H – horizonal component of F_T
F_W – wind disturbance force
mg - force of gravity
θ – pitch angle
ω – pitch angular rate
F_V
F_T
F_W
F_H
θ
ω
mg

Figure 10. Free-body diagram of drone—single-axis horizontal motion.

$$F_V = F_T \cos(\theta) \tag{18}$$

and horizontal component

$$F_H = -F_T \sin(\theta). \tag{19}$$

The horizontal component of thrust causes the drone to accelerate. The vertical causes it to remain at a fixed altitude.

Since

$$F_V = mg, \tag{20}$$

we have:

$$F_T = \frac{mg}{\cos(\theta)}, \tag{21}$$

$$\begin{aligned} F_H &= -F_T \sin(\theta) \\ &= -\left(\frac{mg}{\cos(\theta)}\right)\sin(\theta), \\ &= -mg\tan(\theta) \end{aligned} \tag{22}$$

Using Newton's second law, the sum of the forces acting on a body along an axis equals the change in momentum along that axis

$$\sum F = m\ddot{x} \tag{23}$$

leads to:

$$m\ddot{x} = F_H + F_W \tag{24}$$

$$= -mg\tan(\theta) + F_W, \tag{25}$$

where F_W is the disturbance force due, for example, to wind gusts.

The dynamic angular response of the platform is assumed to be governed by a pitch control system having a second-order response with a rise time of 0.3 s. The corresponding second-order system is:

$$\frac{\theta}{u} = \frac{\Omega_p^2}{s^2 + 2\zeta\Omega_p s + \Omega_p^2} \tag{26}$$

with $\Omega_p = 5.4$ rad/s and $\zeta = 0.71$. In this model, the input u is the commanded pitch angle, or the pitch angle setpoint.

The following state vector is defined:

$$x = \begin{bmatrix} \omega & \theta & v & x \end{bmatrix}' = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \end{bmatrix}', \tag{27}$$

which leads to the following state space model:

$$\begin{aligned} \dot{x}_1 &= \Omega_p^2 \left(u - x_2\right) - 2\zeta\Omega_p \dot{x}_1 \\ \dot{x}_2 &= x_1 \\ \dot{x}_3 &= -g\tan\left(x_2\right) + F_W \\ \dot{x}_4 &= x_{3,} \end{aligned} \tag{28}$$

where F_W is the force of the wind and u is the control input.

B. Simulink Simulation Model and Control Design

A model of the single-axis translational motion of a drone platform, the OFS, IMU, and Proportional-Integral-Derivative (PID) controller was developed, with the top level block diagram shown in Fig. 11.

A velocity setpoint, shown at left, is the signal injected by the drone pilot. This command is compared to the velocity achieved to produce an error signal, which enters a PI controller producing a commanded pitch angle u same as in equation (28) into the plant.

The OFS block content is provided in Fig. 7. The IMU model is pictured in Fig. 14.

The coefficients and parameters required to simulate this model and reproduce the results presented in this paper are detailed in Table VI. A PI controller was designed with gains specifically chosen to achieve a 1 s step response rise time and sufficient integral action to mitigate the effects of constant wind disturbances. The PI controller coefficients, along with other essential parameters, including the UAV's rotational dynamic coefficients for an attitude-stabilized quadcopter, are provided in Table VI. These parameters were used in the simulations model of Figs. 7 and 11–15, and simulation results of Figs. 16 and 17.

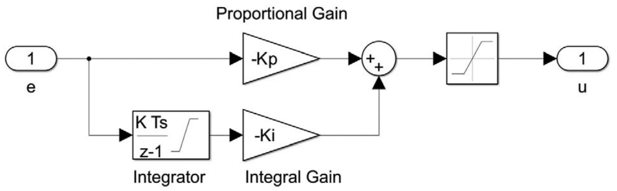


Figure 12. Controller with proportional and integral paths, and pitch angle command saturation.

C. Simulation Test Results

Simulation tests were performed to assess the effectiveness of a horizontal velocity control system for a drone, incorporating the OFS and an IMU. The results were generated using the single-axis dynamic model discussed earlier, a model that includes a continuous-time fourth-order representation of plant dynamics and discrete-time models for the OFS, IMU, and the digitally implemented PI controller. Both sensors operate at a frequency of 50 Hz, and each sensor model includes a computational latency of one sample period, 20 ms. There are no other impairments or noise sources added to the sensor outputs.

D. Step Response in Velocity

The step response test is conducted to ensure that the system responds to changes in velocity setpoints with an appropriate speed of response sufficient to meet a drone pilot's expectations for responsiveness yet avoiding jerky or erratic behavior. The results of this simulation test are presented in Fig. 16. At the 5 s mark, a step change command is initiated by the pilot, commanding a velocity change from 0 to 2 m/s. This is shown in Fig. 16(a) where we see that the velocity increases from 0 to 2 m/s with a rising time of less than 1 s and a setting time of about 2 s. The true velocity exceeds the setpoint with an acceptable level of overshoot, settling at 2 m/s as desired. To effect this change, the controller caused the platform's pitch angle to drop to −20° as shown in Fig. 16(b). The drone then pitches forward and accelerates along the positive x-axis. During this transient response, the OFS and IMU generated outputs are depicted in Fig. 16(c) and (d), respectively.

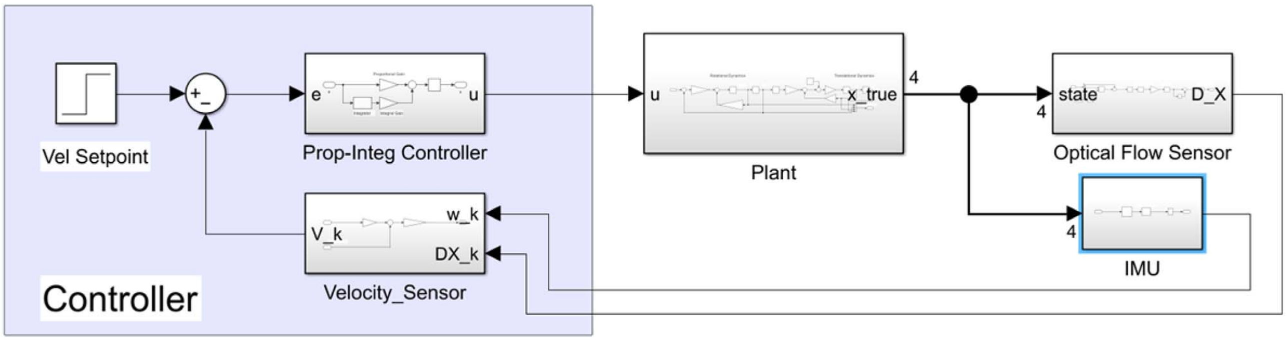
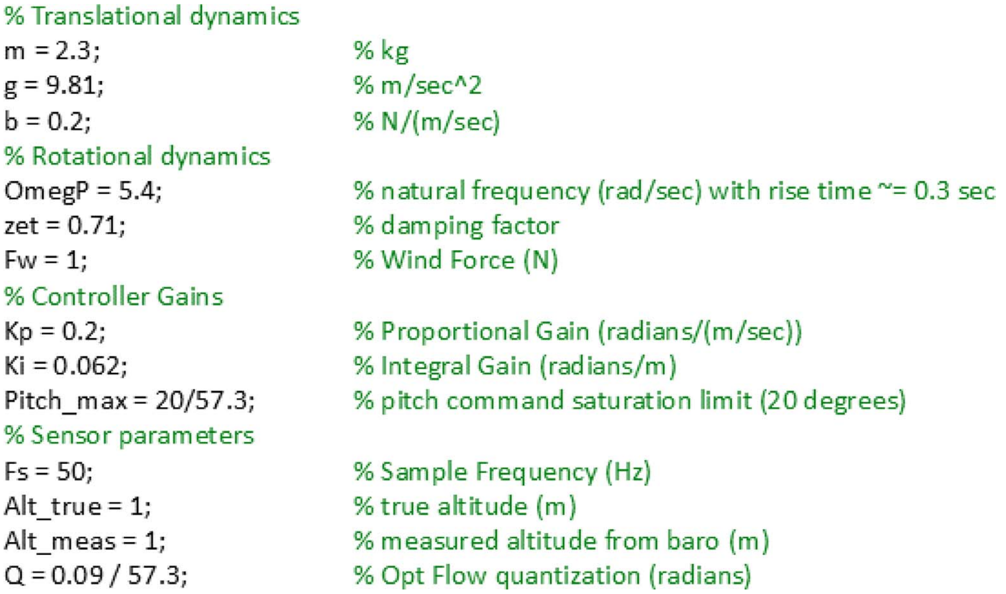


Figure 11. Control system model with single-axis drone (plant) and sensors.

Table VI
Simulation Model Parameters



```
% Translational dynamics
m = 2.3;                    % kg
g = 9.81;                   % m/sec^2
b = 0.2;                    % N/(m/sec)
% Rotational dynamics
OmegP = 5.4;                % natural frequency (rad/sec) with rise time ~= 0.3 sec
zet = 0.71;                 % damping factor
Fw = 1;                     % Wind Force (N)
% Controller Gains
Kp = 0.2;                   % Proportional Gain (radians/(m/sec))
Ki = 0.062;                 % Integral Gain (radians/m)
Pitch_max = 20/57.3;        % pitch command saturation limit (20 degrees)
% Sensor parameters
Fs = 50;                    % Sample Frequency (Hz)
Alt_true = 1;               % true altitude (m)
Alt_meas = 1;               % measured altitude from baro (m)
Q = 0.09 / 57.3;            % Opt Flow quantization (radians)
```

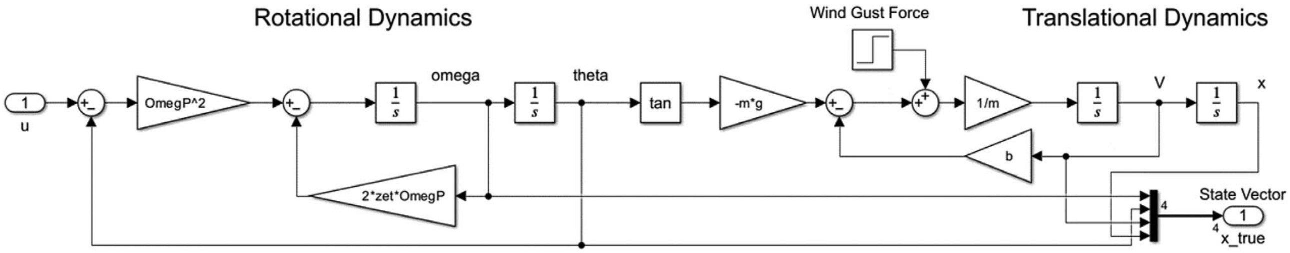


Figure 13. Plant model based on drone dynamics.

E. Response to a Wind Gust

The purpose of the wind gust test is to verify that the controlled system's response to a wind gust is acceptably small. The results of this simulation test are as follows: Commencing at 5 s, a headwind force of 4 Newtons is applied, representing a significant force relative to the drone's weight of 22 Newtons. The headwind drives the drone backward and the velocity sensed is negative as shown in Fig. 17(a). In response to this horizontal velocity disturbance, the controller initiates a 10° downward pitch, as illustrated in Fig. 17(b), to counteract the wind. The drone is driven backwards a distance of about 2.7 m at which point it comes to a halt with zero velocity. The integral part of the PI controller has developed a 10° pitch angle to hold the drone against the headwind. The pilot can command the drone to follow a positive velocity setpoint if they wish to move the drone back to its original location. The OFS output during this time is shown in Fig. 17(d).

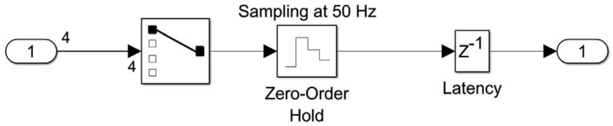


Figure 14. Inertial measurement unit Simulink block diagram.

F. Very Slow Movement

The very slow movement test is designated to verify that the highly quantized output from the OFS is adequate to enable the velocity controller to induce extremely low-velocity horizontal motion. The results of this simulation test are illustrated in Fig. 18. The velocity-commanded setpoint transitions from 0 to 1 cm/s at 2 s [see Fig. 18(a)]. At this speed, the drone moves at a barely perceptible pace. The resulting OFS output is a pulse stream with a very low duty cycle, as shown in Fig. 19, with pulses that alternate between 0 and −1 at a rate of approximately 7 Hz. Noting that we are sampling the sensor's output at a rate of 50 Hz, this stream is producing one count over a period of 7 or 8 samples. The results show that the controller achieves the commanded 1 cm/s velocity as noted in Fig. 18(a). Although the pulsing introduces some jumpiness in the control signal (pitch angle command), the plant's dynamics effectively filter out this "noise," resulting in a smooth achieved veloc-

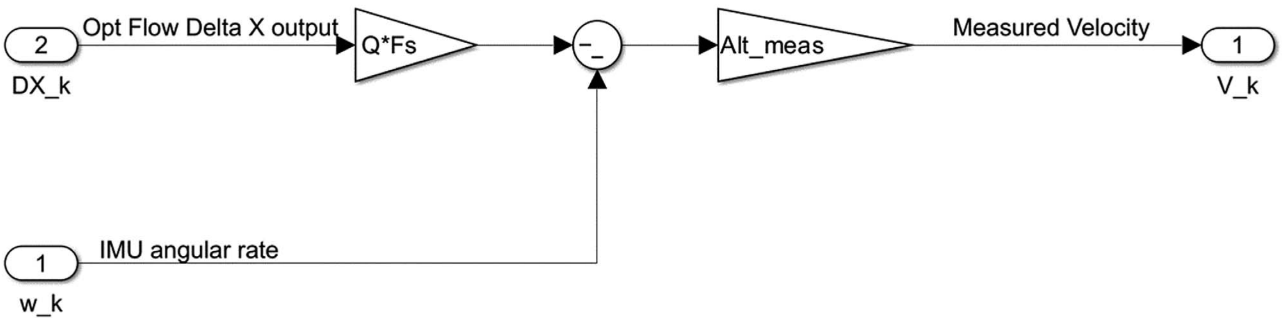


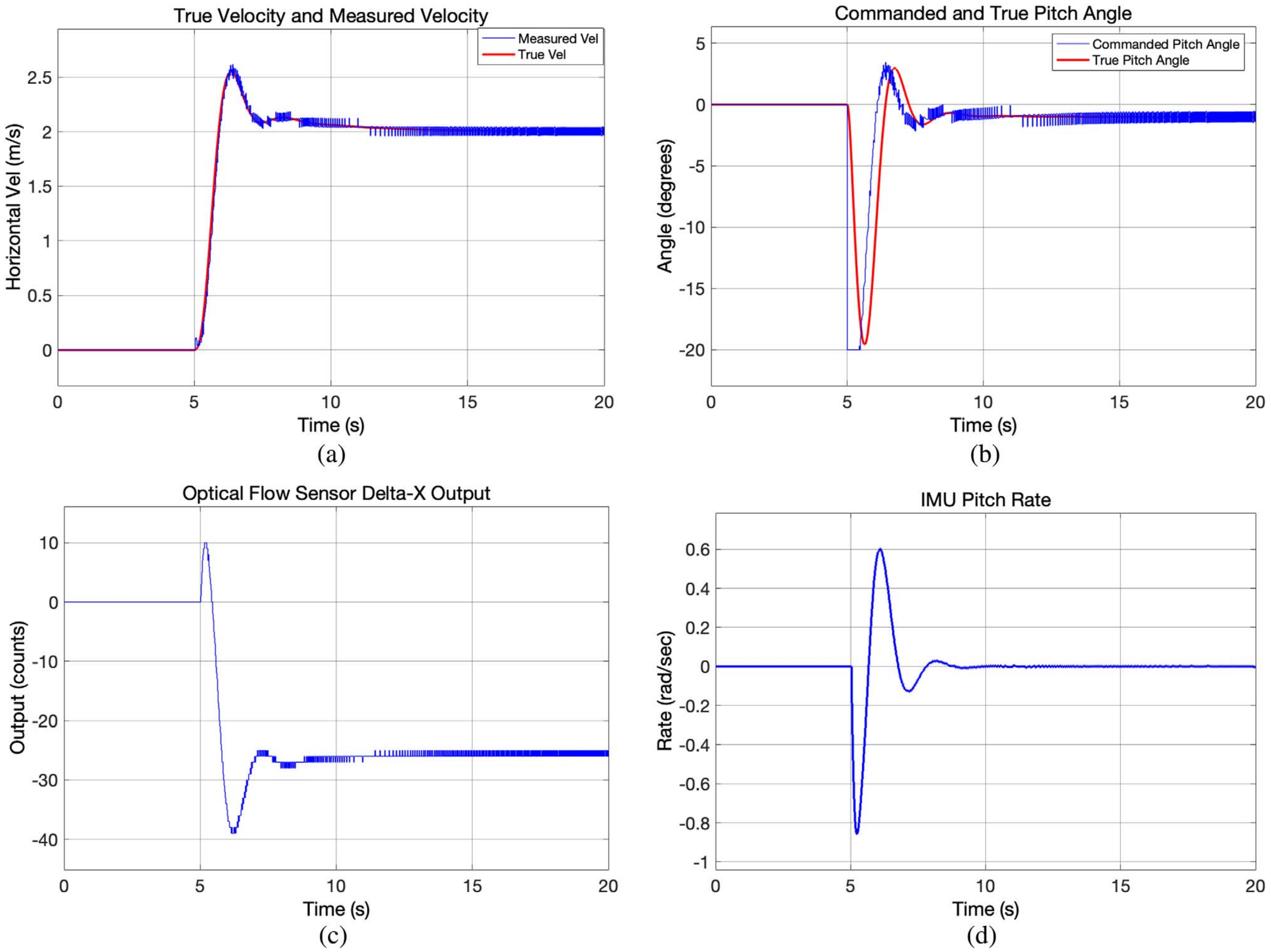
Figure 15. Velocity sensor Simulink block diagram.

ity. The IMU's Pitch Rate output during this period is presented in Fig. 18(d). The successful completion of this test indicates that the OFS's output is indeed adequate for the velocity controller to induce and regulate extremely low-velocity horizontal motion, thereby affirming its suitability for tasks that demand high levels of precision and control.

Zooming into the OFS output in Fig. 19, we see that it does not produce a non-zero output for 0.7 s after the commanded step change at 2 s. During this period, the sensor has not yet experienced sufficient motion to generate an output. Subsequently, once the output initiates, the sensor produces a pulsed signal stream with a low - duty cycle. The average output is effectively a fraction of one count.

VI. CONCLUSIONS AND FUTURE WORK

The ability to effectively process and integrate data from a diverse set of sensors is important in UAV navigation, localization, and control, and can improve system operational performance and safety. In this paper, a new OFS model was presented, a model that captures the impact of pixel quantization and uses platform angular



(a) (b) (c) (d)

Figure 16. Step response in velocity.

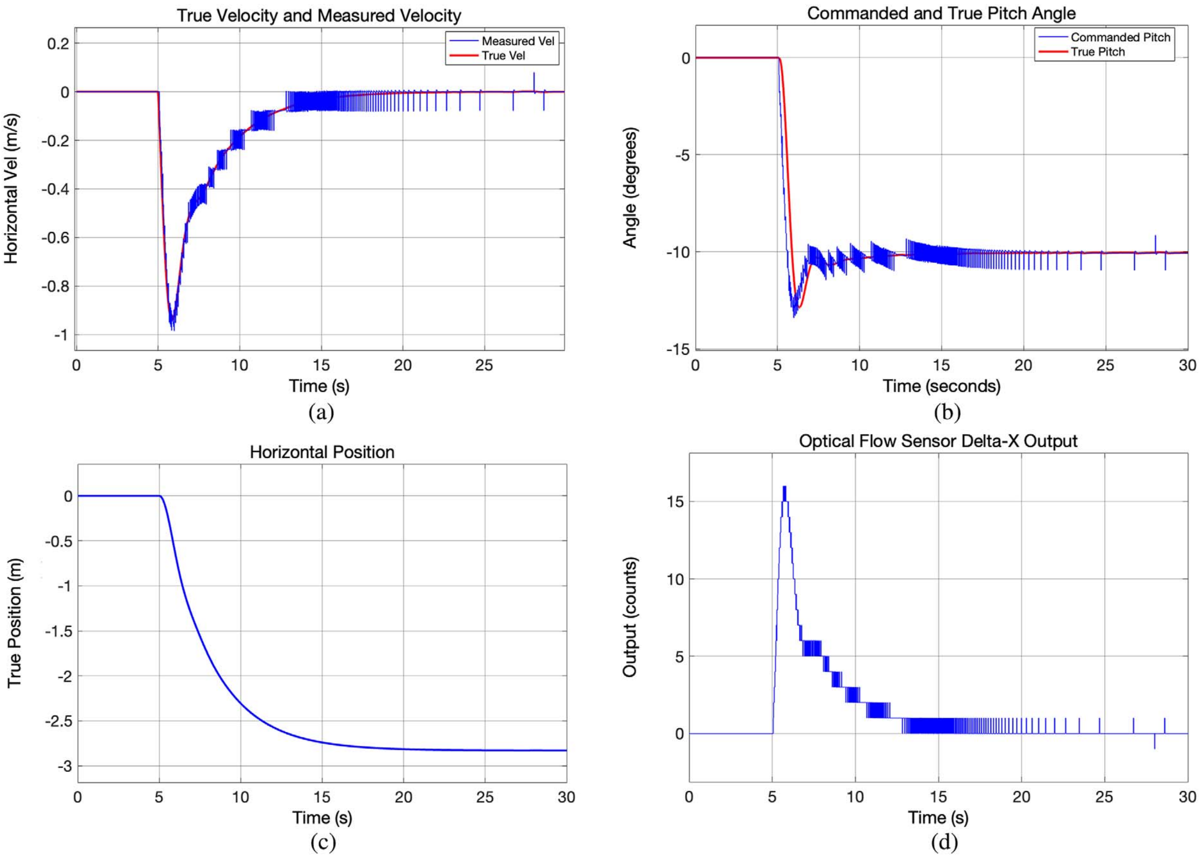


Figure 17. Response to wind gust.

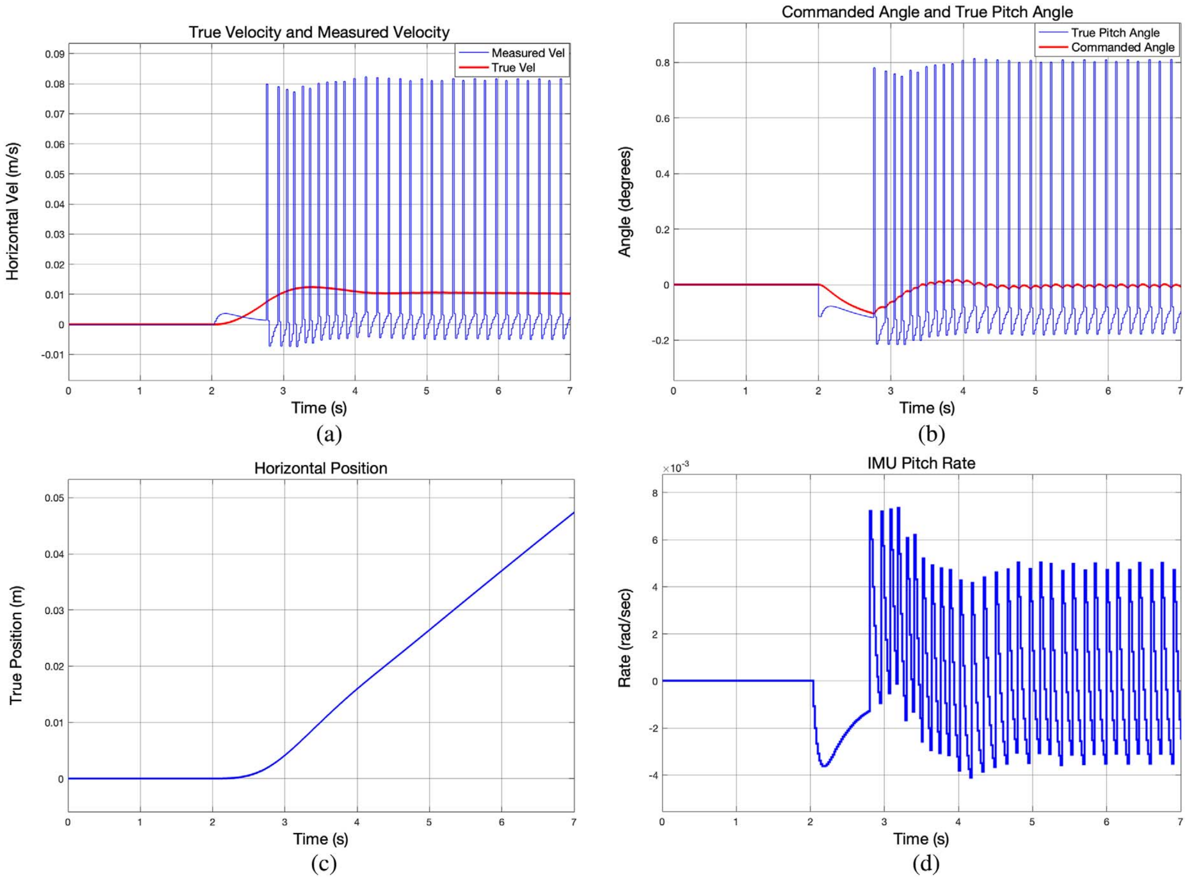


Figure 18. Very slow commanded velocity of 1 cm/s (0.01 m/s).

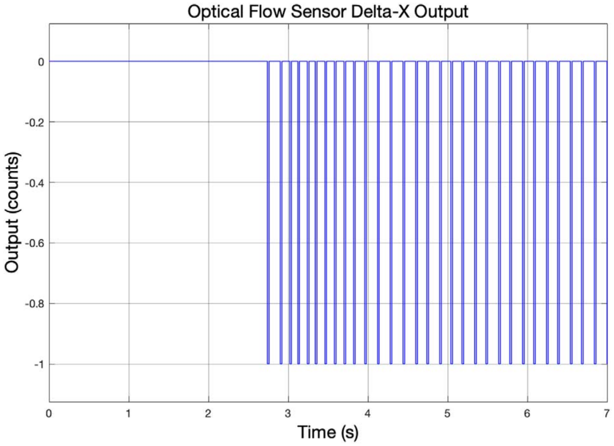


Figure 19. Optical flow sensor output during very slow commanded velocity of 10 cm/s.

rate and linear velocity to generate the OFS output. This model accurately reflects a key feature of the sensor—the fact that it is measuring angular displacement rather than angular velocity. How this impacts platform horizontal velocity control is discussed. From this new model, a velocity measurement equation was derived that fuses platform IMU and OFS data. The combination of data from these sensors has been shown to enable the generation of accurate velocity estimates in two dimensions, a feature particularly beneficial in GPS-compromised environments. Using this velocity measurement as feedback, a control system for horizontally stabilizing drone location has been proposed. Future work will include the assessment of IMU bias and random noise error on horizontal hold control system performance.

REFERENCES

[1] S. Huang and G. Dissanayake
"Robot localization: An introduction,"
in *Wiley Encyclopedia of Electrical and Electronics Engineering,* J. G. Webster, Ed. Hoboken, NJ, USA: Wiley, 2016, pp. 1–3.

[2] T. D. Barfoot
State Estimation for Robotics: Second Edition, Introduction. Cambridge, U.K.: Cambridge Univ. Press, 2024, pp. 1–4.

[3] Q. Yuan and I.-M Chen
"Localization and velocity tracking of human via 3 IMU sensors,"
Sensors Actuators A: Phys., vol. 212, pp. 25–33, 2014.

[4] S. M. Joshi
"Adaptive control in the presence of simultaneous sensor bias and actuator failures," National Aeronautics and Space Administration, Langley Research Center, Hampton, VA, NASA Tech. Rep. NASA/TM-2012-217231, 2012.

[5] B. Wang
"Reduced integration time improves accuracy in dead reckoning navigation systems,"
Analog Dialogue, vol. 47, no. 7, pp. 722–723, July 2013.

[6] M. L. Abell and J. P. Braselton
Differential Equations with Mathematica, 5th ed. San Diego, CA: Academic Press, 2022.

[7] H. B. Gilbert, O. Celik, and M. K. O'Malley
"Long-term double integration of acceleration for position sensing and frequency domain system identification," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatron.*, 2010, pp. 453–458 .

[8] Y. Nabavi, D. Asadi, and K. Ahmadi
"Image-based UAV position and velocity estimation using a monocular camera,"
Control Eng. Pract., vol. 134, no. 3, 2023.

[9] F. Santoso, M. A. Garratt, and S. G. Anavatti
"Visual–inertial navigation systems for aerial robotics: Sensor fusion and technology,"
IEEE Trans. Automat. Sci. Eng., vol. 14, no. 1, pp. 260–275, Jan. 2017.

[10] MathWorks
"What is SLAM (Simultaneous Localization and Mapping)—MATLAB," Accessed: April 7, 2024. [Online]. Available: https://www.mathworks.com/discovery/slam.html

[11] S. Lange, N. Sünderhauf, and P. Protzel
"Autonomous landing for a multirotor UAV using vision," in *Proc. SIMPAR 2008 Int. Conf. Simulation Simul., Model. Program. Auton. Robots*, 2008, pp. 482–491 .

[12] Z. Y. Chen, W. C. Chen, X. M. Liu, and C. Song
"Fault-tolerant optical flow sensor/SINS integrated navigation scheme for MAV in a GPS-denied environment,"
J. Sensors, vol. 2018, pp. 9678505, 2018.

[13] F. Kendoul, I. Fantoni, and K. Nonami
"Optic flow-based vision system for autonomous 3D localization and control of small aerial vehicles,"
Robot. Auton. Syst., vol. 57, no. 6–7, pp. 591–602, 2009.

[14] T. G. Kim, D. M. Kim, S. K. Kim, Y. S. Kim, and S. H. Han
"Improved optical sensor fusion in UAV navigation using feature point threshold filter,"
Int. J. Aeronautical Space Sci., vol. 23, no. 1, pp. 157–168, 2022.

[15] X. Li, C. X. He, and Y. M. Tang
"Improved modeling and fast in-field calibration of optical flow sensor for unmanned aerial vehicle position estimation,"
Measurement, vol. 225, 2023, Art. no. 114066.

[16] S. Lee and J.-B. Song
"Robust mobile robot localization using optical flow sensors and encoders," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2004.

[17] H. Didari, F. Lotfi, H. Taghirad, and S. Germi
"Position estimation for drones based on visual SLAM and IMU in GPS-denied environment," in *Proc. 7th Int. Conf. Robot. Mechatron.*, 2019, pp. 120–124.

Djedjiga Belfadel received the Ph.D. degree in electrical and computer engineering from the University of Connecticut, Storrs, CT, USA, in 2015. She is currently an Associate Professor and the ECE Graduate Program Director in the Department of Electrical and Biomedical Engineering Department, Fairfield University, Fairfield, CT, USA. Her research traverses a wide breadth of estimation theory, with a specific emphasis on practical applications such as drone navigation and target tracking. Her scholarly contributions extend to space-based infrared (IR)/electro-optical (EO) sensors, signal and image processing, machine learning, and big data. Alongside her research, she is committed to enhancing engineering education and boosting the representation of women and underrepresented groups within the engineering sector.

John Cain received the M.S. degree in Electrical and Computer Engineering in 2024 and the B.S. degree in Computer Science in 2022, both from Fairfield University, Fairfield, CT, USA, where he worked as a graduate research assistant. He is currently a firmware engineer at SmartSense. His research interests include sensor systems/communication and embedded electronics.

David Haessig received the B.S. and M.S. degrees in mechanical engineering from Lehigh University, Bethlehem, PA, USA and the Ph.D. degree in electrical engineering from NJIT, Newark, NJ, USA. With over 40 years of experience in communication, control, and navigation systems, he is currently the Chief Scientist at AuresTech Inc., Bridgewater, NJ USA. Previously, at BAE Systems, he was a Technical Director II, leading significant projects like the Joint Tactical Radio System's Anti-Jam Wideband Networking Waveform and developing UAV SATCOM control systems. He also worked on submarine control systems at General Dynamics and advanced wireless technologies at Lucent Technologies, where he earned a Bell Labs Gold Award. He is a registered Professional Engineer in New Jersey, an IEEE Life Senior member, and an Adjunct Professor at NJIT. He holds 13 patents and has 27 technical publications.

Cherif Chibane is the Founder and Chief Technology Officer at AuresTech Inc., Bridgewater, NJ, USA, where he leads R&D in configurable computing for wireless communications. He has also served as a Senior Research Scientist at MIT Lincoln Laboratory and as a Science and Technology PM at Draper Laboratory, focusing on advanced communication systems for NASA and DARPA. At BAE Systems, he led the development of System On Chip technology and UAV navigation solutions. Cherif holds an MSEE from Fairleigh Dickinson University. His expertise includes RF systems, control systems, and embedded systems development.