2011 IEEE/RSJ International Conference on
Intelligent Robots and Systems
September 25-30, 2011. San Francisco, CA, USA

Hardware in the Loop for Optical Flow Sensing in a Robotic Bee

Pierre-Emile Duhamel[1,2], Judson Porter[1], Benjamin Finio[1,2],
Geoffrey Barrows[3], David Brooks[1], Gu-Yeon Wei[1], and Robert Wood[1,2]

***Abstract*— The design of autonomous robots involves the development of many complex, interdependent components, including the mechanical body and its associated actuators, sensors, and algorithms to handle sensor processing, control, and high-level task planning. For the design of a robotic bee (RoboBee) it is necessary to optimize across the design space for minimum weight and power consumption to increase flight time; however, the design space of a single component is large, the interconnectedness and tradeoffs across components must be considered, and interdisciplinary collaborations cause different component design timelines.**

In this work, we show how the development of a *hardware in the loop* (HWIL) system for a flapping wing microrobot can simplify and accelerate evaluation of a large number of design choices. Specifically, we explore the design space of the visual system including sensor hardware and associated optical flow processing. We demonstrate the utility of the HWIL system in exposing trends on system performance for optical flow algorithm, field of view, sensor resolution, and frame rate.

I. INTRODUCTION

Highly interdisciplinary research to develop a colony of bio-inspired microrobotic bees, or RoboBees, is divided into three broad areas: body, brain, and colony. The body group investigates the design and manufacturing of an approximately 500 mg flapping-wing MAV including considerations of aerodynamics [1], artificial wings [2], actuation of wing stroke [3], control [4], passive stability [5], and power electronics [6]. The brain group is developing power efficient computational hardware and architectures [7] to enable high performance autonomous RoboBee algorithms. The colony group investigates algorithms for collective and emergent behavior of a group of RoboBees, despite minimal programming and communication between individuals [8].

The development of effective autonomous RoboBees presents a broad range of challenges and difficulties. The scale imposes stringent mass (<500 mg) and power (<350 mW) constraints on all components of the system to allow flight of the robot, and any small inefficiency will impact flight time [6]. The robotic system must be optimized for low mass and power consumption within acceptable bounds on performance. However, each individual component is complex with many design choices to consider. To further complicate the optimization problem, the interconnectedness between components necessitates that the significant number of parameters defining each component not be considered in isolation. Due to the interdisciplinary nature of this project, each component will be at a different phase of development, increasing the difficulty of integration and full system testing. The RoboBee will also be targeted at a wide range of applications, where a design point optimized for one application may not be ideal for a different application.

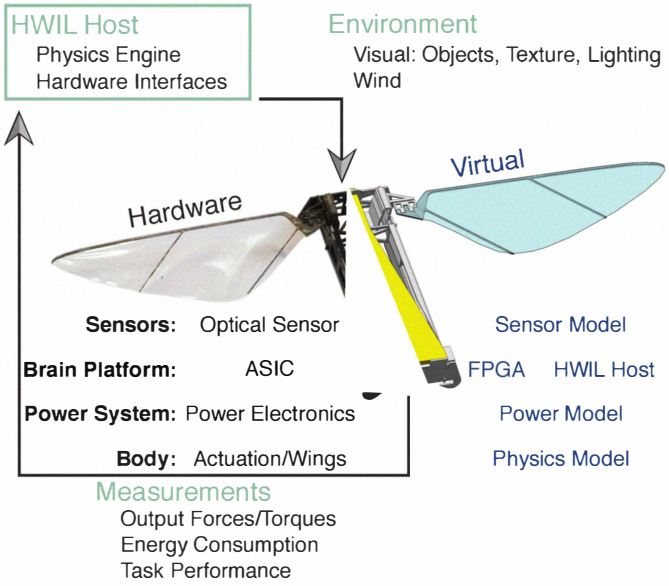

Fig. 1. System Overview

To overcome these challenges, we have developed a *hardware in the loop* (HWIL) system. This system is modular to allow simultaneous development and evaluation of individual components and the entire RoboBee. Fig. 1 depicts the high-level vision of the HWIL system. The system provides a virtualized environment for the RoboBee to operate in, modular implementations of each RoboBee component (hardware or software), and the ability to obtain any measurements of interest. At the start of the project, each component is implemented with a high-level functional software model, represented to the right of Fig. 1. As the project progresses, the high-level component models can be replaced with actual hardware implementations, shown to the left of Fig. 1.

This work was supported by the National Science Foundation (NSF) Expeditions in Computing Award #: CCF-0926148. P.E. Duhamel was supported by a Graduate Research Fellowship from the NSF. B. Finio was supported by the Department of Defense (DoD) through the National Defense Science & Engineering Graduate Fellowship (NDSEG) Program. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

The authors are with the [1]School of Engineering and Applied Sciences, Harvard University, Cambridge, MA; [2]Wyss Institute, Harvard University, Cambridge, MA; and [3]Centeye Inc., Washington, D.C. pduhamel@fas.harvard.edu

This system meets our challenging design needs by exposing design parameters of all components for rapid, full-system testing and optimization across the integrated RoboBee design space. To alleviate different hardware design timelines the RoboBee can exist in the HWIL system as any combination of components where some may be software and others hardware. The system allows measurement of all metrics of interest to evaluate performance of the RoboBee, and the virtualized environment is reconfigurable, such that the RoboBee system can be optimized for widely varying environmental conditions. The HWIL system can incorporate cost models of power consumption and component weight to allow evaluation of tradeoffs between performance and cost (weight, power) at both the individual component and full system levels. The system allows both flexibility in evaluating the entire system performance, and targeted exploration of the design space of individual components. This work introduces the RoboBee HWIL system.

To demonstrate effectiveness of the HWIL system, we explore the design space of a visual sensor and associated optical flow processing components. Optical flow can broadly be described as the apparent motion of objects through an observer's field of view, given relative motion between the viewer and the objects, Fig. 2. Previous work has shown that flying insects use optical flow for various tasks, including velocity control, obstacle avoidance, and landing [9]. Given its use by real honeybees, it is reasonable to evaluate optical flow for exteroceptive feedback for a RoboBee. The design space for an optical flow sensor is significantly large as there is a wide range of optical flow algorithms, as well as dependence on characteristics of the visual sensor, such as resolution, field of view, and frame rate.

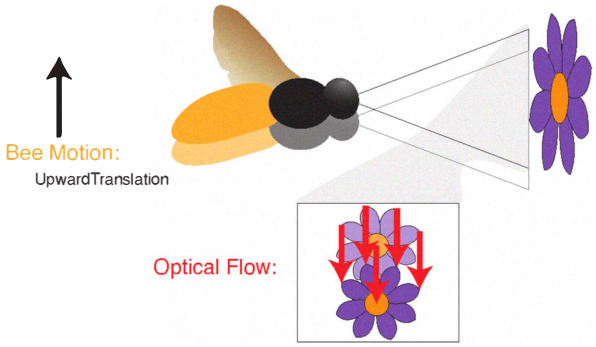

Fig. 2. Here as the bee moves upwards, the per pixel optical flow vectors tend in the downward direction. This vector field can be used to estimate vertical/lateral motions.

The use of simulators, software in the loop, and hardware in the loop systems for development and testing of UAVs is a well established technique [10]–[14]. Many of these systems are developed to test new algorithms for existing platforms [11], or do not modularize all components of the system. Our work is unique in that it allows for both software models and actual hardware implementations of all components of the system. Optical flow is commonly used for controlling robots and UAVs [15]–[20]. These robotic systems typically employ several cameras with optical flow processing using either biologically inspired EMD-type algorithms [16], [19], or more conventional techniques such as the Lucas-Kanade algorithm [20]. Rather than just evaluating the performance of a robot with one optical flow configuration, we use our system to evaluate a range of algorithms and configurations. Other research has compared the performance of different optical flow algorithms [20]–[22]. These works however either do not consider the algorithms in the context of a robotic system [21], [22], or do not consider the impact of the sensor design on the choice of algorithm [20]. Additionally, our flapping-wing micro air vehicle platform is orders of magnitude smaller than the fixed-wing planes [15], helicopters [17], quadcopters [18] and ornithopters [16] used in other works. Flight at the scale of an insect presents stricter power and weight constraints than larger vehicles.

The rest of this work is organized as follows. We first detail the current implementation of each HWIL system component in Section II, and validate some of our software models in Section III. Then we demonstrate in Section IV how the HWIL system allows us to explore the design space for an optical flow sensor and its utility in exposing component level design space trends that impact whole system performance. Section V concludes the paper.

II. SYSTEM OVERVIEW: OPTICAL FLOW TEST CASE

The goal of the RoboBee HWIL system is to allow deep and thorough evaluation of each system component at all stages of the development cycle. Here, we report on the current status of each component in the context of an optical flow sensor characterization. The HWIL system is composed of a virtual environment, visual sensors, optical flow processing, control, and a body physics simulation.

For this design space exploration, we consider the RoboBee as depicted in Fig. 3. The RoboBee is limited to motions in the X, Z world plane by actuating a lift force F_{axb} and yaw torque τ_{psi}. A visual sensor is attached to the center of the body facing the z_b direction. The use of 2D planar motion is not a limitation of the HWIL system as the environment, sensors, control, and physics engine are all capable of 3D motions. By limiting the motion of the RoboBee to 2D we can narrow the relevant design parameters. Optical flow based control in 3D expands the design space to include the number, arrangement, and orientation of sensors on the body; more complicated optical flow motion interpretation algorithms; and more complicated control strategies to combine sensory data. The use of 2D allows us to focus on the properties of a single optical flow sensor, which provides a starting point for future explorations of the broader 3D case.

A. Environment

The HWIL system provides a realistic environment for the RoboBee to operate in by generating simulated inputs for all sensors as well as physical environmental disturbances. The virtual environment currently consists of visualizations of the

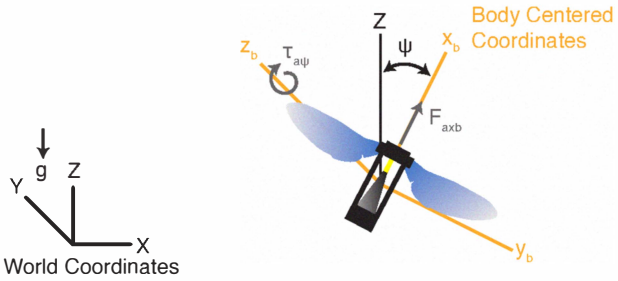

Fig. 3. The vehicle can actuate force F_{axb} aligned with x_b and torque $\tau_{a\psi}$ about z_b in the body centered coordinate system (x_b, y_b, z_b). Motion is constrained to the plane formed by the world-fixed XZ axes with translational degrees of freedon x_b and y_b and rotational degree of freedom ψ between the Z and x_b axes.

virtual world for visual sensor input and wind disturbance models, and both elements can be adjusted to simulate various operating environments.

The software design of the visual environment is similar to a game engine; however, we implemented the environment directly on top of the OpenGL[1] image rendering library rather than using an off-the-shelf game engine. This allows greater flexibility and better integration with the rest of the HWIL system. The world is bounded with a skybox and the ground plane is generated from a height map. Models of flowers or other objects can be imported in the 3DS format. A sample rendering of the environment is shown in Fig. 4.

Fig. 4. Example of virtual 3D visual environment rendering.

The wind model subsystem allows for arbitrary $1 - \cos$ profiles for a discrete gust model [23]. Here we use an impulse and step wind profiles as seen in Fig. 5. The wind force is then determined by $F = \frac{1}{2}\rho C A v^2$, where $\rho = 1.2\,\mathrm{kg/m^3}, C = 1.05, A = 1\,\mathrm{cm}^2$ are the density of air at sea level, the bluff body drag coefficient for a cube (a rough approximation of the RoboBee body shape), and the wetted area of the RoboBee body respectively.

B. Vision Sensor

The HWIL system allows for visual processing of the environment by both hardware sensors and software sensor models.

[1]Open Graphics Library: http://www.opengl.org/

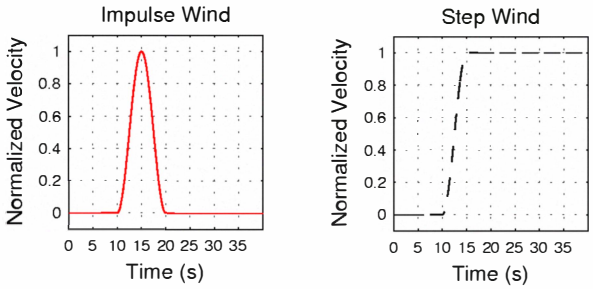

Fig. 5. Wind Model: From left to right, impulse wind in solid red and step wind in dotted black.

1) Centeye FireflySmall: The hardware vision sensor used is the FireflySmall vision chip from Centeye, Inc. [24]. The FireflySmall chip is a variable acuity image sensor chip designed for embedded vision and robotic applications. This chip has a 128×256 array of logarithmic response pixels in a $2.4\,\mathrm{mm} \times 4.7\,\mathrm{mm}$ focal plane, with a $18.3\,\mu\mathrm{m}$ pixel pitch. The focal plane includes a binning network allowing M $\times$ N blocks of superpixels to be formed by shorting together all pixel circuits within such a superpixel block, with M and N each selectable from 1, 2, 4, and 8. Other peripherals include on-chip bias generators, an 8-bit ADC, a voltage regulator for powering analog circuits, and a parallel interface designed to enable operation from a microcontroller or a DSP. Fig. 6 shows the FireflySmall sensor die.

For the current study, the chip was operated in log-response mode, where the voltage output is a logarithmic function of light incident on a pixel. This feature has the advantage of compressing a wide range of light intensities into a smaller voltage range, and allows the sensor to be operated without the use of precise timing to read the pixel array. The reduced complexity of operating the chip results in lower weight and computational burden on the RoboBee.

When using the FireflySmall sensor in hardware, it is aimed at a LCD monitor displaying the rendered visual environment. A Sunex DSL240A lens is mounted on the sensor as seen in Fig. 6. We assume that the Sunex lens effectively copies the image from the LCD onto the sensor surface when focused on the LCD and aligned such that the edges of the rendering are at the edges of the active sensor area. Therefore, the field of view of the lens is determined by the environmental rendering displayed on the LCD, which is modeled as a pinhole lens in OpenGL, rather than the Sunex lens. The pinhole lens model is more relevant as small, lightweight optics such as a printed pinhole lens [25] are more desirable to the RoboBee platform than traditional optics.

2) Virtual Sensor Model for Centeye FireflySmall: The ability to use software models of hardware sensors in the HWIL system allows for expanded explorations of sensor parameters by offering quicker, easier adjustment of lens and sensor hardware configurations as well as exploration of parameter choices outside of current hardware capabilities. The current study was completed on a single workstation where interfacing to RoboBee hardware is possible. An accurate

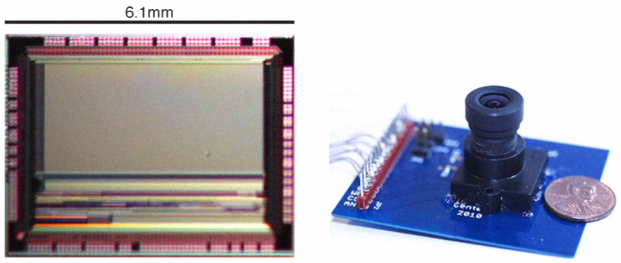

Fig. 6. Images of the FireflySmall sensor. The image on the left shows the sensor bare die, while the image on the right shows the sensor attached to a breakout board and with more conventional lens mounted.

software vision sensor model could allow the same simulations to be implemented on a cluster for larger parameter studies. The virtual software sensor model includes the same parameters as considered when using the hardware sensor including resolution, field of view for a pinhole lens, and the intensity response of the sensor. We do not currently consider secondary effects such as non-fixed-pattern noise or various modes of optical distortion. Since the sensor is virtualized, both resolution and field of view of the pinhole lens are determined by the environmental rendering parameters. An intensity response calibration transforms rendered intensities to sensor intensity. A comparison between the performance of the hardware sensor and the virtual sensor model is shown in Section III.

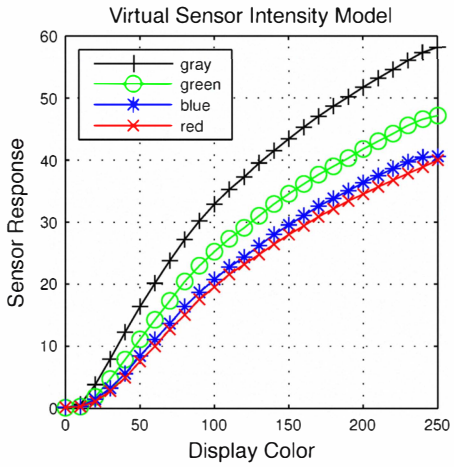

Fig. 7. Average measured response of the sensor to shades of gray, red, blue, and green. The color of the line corresponds to the color being measured.

To develop the intensity model of the FireflySmall sensor, we positioned the sensor in front of the LCD monitor used for displaying the simulated scene and measured the intensity response across a range of colors displayed on the LCD. The sensor measures one channel intensity; however, we calibrated the sensor across different display colors to ensure any effects of sensing the color image in monochrome are accounted for. For example, texture that may be apparent when viewed in color may be masked when viewed in monochrome by two colors having the same monochrome intensity. A calibration routine displayed approximately 17,000 RGB colors on the monitor and recorded the average brightness response across the sensor for each color. A nearest-neighbor interpolation routine generated the rest of the color space. The step size of the measurements was chosen such that the difference in response between steps was less than 1, ensuring that no accuracy was lost due to the use of nearest-neighbor interpolation. This interpolation resulted in a lookup table with a sensor response value for each of the approximately 16 million colors in the 8-bit RGB color space. This lookup table was then used to convert the RGB values of the rendered environment into an approximation of the sensor response looking at the monitor displaying that same image. Fig. 7 shows the response of the sensor to the range of gray, red, green, and blue values.

C. Optical Flow

There are a wide range of algorithms available for calculating the optical flow field, resulting in a range of computational complexities and effectiveness for control. When considering different optical flow algorithms, we are interested, primarily, in the tradeoff between power and performance. For this purpose we consider the widely used Lucas-Kanade algorithm, along with the computationally simpler, biologically-inspired Image Interpolation algorithm. We evaluate and compare the performance of both of these algorithms in Section IV.

1) Lucas-Kanade: The Lucas-Kanade (LK) [26] algorithm is one of the most commonly used optical flow algorithms for computer vision applications. The main assumption in this algorithm is that the optical flow in a small neighborhood of pixels is the same. With this assumption, a least squares search is performed to find the estimated velocity for each pixel. The components of the per pixel optical flow vectors are averaged to generate one optical flow vector for the whole sensor. In the HWIL system, we use the OpenCV [27] implementation of LK optical flow.

2) Image Interpolation: In contrast to the higher computational complexity of LK, we also investigate a simplified version of the Image Interpolation Algorithm proposed by Srinivasan [28]. This algorithm was modified to compute optical flow with subpixel precision. For I1 and I2 being two sequential images, the algorithm computes optical flow by constructing shifted versions of I1 in each direction vertically and horizontally, and determining the linear mixture of these shifted versions of I1 that best match I2 using a 2-norm metric. The linear mixture amounts form the optical flow measurement where the vector $\widehat{xy}$ contains the resulting optical flow components.

D. RoboBee Body Physics Model

The nonlinear body physics model is responsible for updating the position of the RoboBee within the environment. It takes four forces as inputs: the body reaction force from the controller, gravity, the wind disturbance force from the environment, and an aerodynamic damping force. All forces are expressed in the body centered coordinate frame and the

vertical (1), lateral (2), and yaw (3) equations of motion become:

$$F_{axb} + F_{gxb} + \frac{1}{2}\rho CA \left(sgn(v_{wxb})||v_{wxb}||^2 - sgn(\dot{x}_b)||\dot{x}_b||^2\right) = m\ddot{x}_b \quad (1)$$

$$F_{gyb} + \frac{1}{2}\rho CA \left(sgn(v_{wyb})||v_{wyb}||^2 - sgn(\dot{y}_b)||\dot{y}_b||^2\right) = m\ddot{y}_b \quad (2)$$

$$\tau_{a\psi} = I_{zz}\ddot{\psi} \quad (3)$$

In the first term of equation (1), the simulated RoboBee body can actuate a lift force F_{axb}, while lateral reaction force is always zero. In the yaw equation of motion (3), $\tau_{a\psi}$ is the RoboBee body actuated yaw torques with the moment of inertia $I_{zz} = 1.31\,\mathrm{g\,mm^2}$ measured from a SolidWorks model of the RoboBee body. The vehicle cannot exert a force in the lateral y_b direction, which is faithful to the current RoboBee actuated degrees of freedom. Thus, horizontal motion is achieved by tilting the body (i.e. $\psi \neq 0$), which creates a horizontal component of the lift force [29]. The F_{gxb} term in equation (1) and F_{gyb} term in equation (2) are the force of gravity transformed into the body centered coordinate system. The terms proportional to v_{wxb} or v_{wyb} are the force of the external wind and the terms proportional to $\dot{x}_b$ or $\dot{y}_b$ are the damping force due to moving the body through air. The force and torque equations of motion are then used to update the state of the RoboBee body at each time step using Euler's method.

E. Control

The controller uses optical flow derived body state approximations as input to reject environmental disturbances. The equations of motion described previously are linearized about stable hover, i.e. $x = \dot{x} = y = \dot{y} \equiv 0$, $F = mg$ and $\tau = 0$, since the controller is used for optical flow based hover in Section IV. The linearized system is used to design a state feedback controller with gain matrix K using the pole-placement MATLAB routine `place`, assuming perfect state information (i.e. the state vector $\bar{x} = [x, \dot{x}, y, \dot{y}, \psi, \dot{\psi}]$).

In the HWIL system, state information $\dot{x}$ and $\dot{y}$ are the optical flow measurements, and x and y positions are found by integration of optical flow. The optical flow measurements are in units of pixels/second. Before each trial, the RoboBee is paned at 1 m/s in the horizontal and vertical direction and optical flow is recorded. The optical flow is averaged to determine a gain factor G to interpret optical flow measurements as velocities in m/s. Since we are not studying the controller in this work, the gain factor G acts to normalize optical flow signals across trials such that the same controller and K matrix are used for all trials. Rotational state information ψ and $\dot{\psi}$ directly measures body orientation state. This measurement could be accomplished many ways on the RoboBee such as an ocelli [30], inclinometer, or more complex optical flow methods. The tradeoffs between these different hardware are worthy of future study with the HWIL system. Thus an estimate of the state $\hat{x}$ is sent to the controller block K to determine the control vector $\hat{u} = [F, \tau]$ (Fig. 8). A nonlinear saturation block limits the force and torque to realistic bounds for the vehicle, $0 < F < 2F_g$ and $-10^{-6}Nm < \tau < 10^{-6}Nm$, which are consistent with measurements of actual RoboBee performance [31], [32].

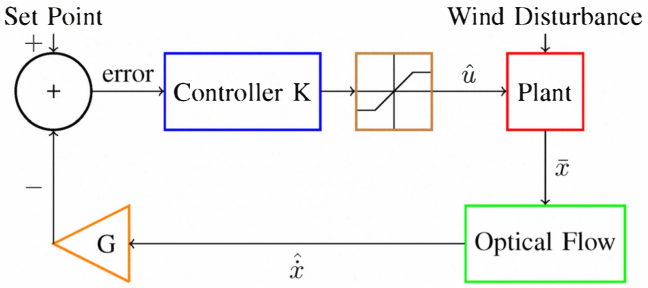

Fig. 8. Block diagram of the controller. An estimate $\hat{x}$ of the state vector $\bar{x}$ is determined by the optical flow sensor. A gain G relates optical flow measurements in pixels/sec to m/s and comparison to the set point determines error which the controller uses to determine body-frame control force and torque $\hat{u}$ with realistic upper and lower bounds placed on force and torque values. The plant inputs are $\hat{u}$ and the environmental wind disturbance.

III. SYSTEM VALIDATION

As described in Section II, the HWIL system supports a hardware vision sensor and a software model based on the sensor. In this section, we compare the performance of the sensor hardware and model, in the context of our test case, to validate and justify our ability to swap between them.

To examine the accuracy of the sensor model, an image from the simulated environment processed by the sensor model was compared with output of the sensor viewing the same image displayed on the LCD monitor. Fig. 9 shows a histogram of the per-pixel differences between the model and the sensor viewing the same image. As this figure demonstrates, the correlation is generally good. The differences that were recorded are likely due to warping from the single element lens and temporal noise sources (non-fixed-pattern noise). While the accuracy could be improved by accounting for these variables, the acceptable performance at this point justifies the assumption that intensity response is the most important characteristic of the sensor.

Now we compare use of the sensor hardware and model in the overall HWIL system, where translational optical flow is input to the controller as an approximate measure of the system state. For a single set of configuration parameters representing the middle parameters of each variable (Algorithm: LK, Resolution: 64 × 64 px, FOV: 90°, Frequency: 100 Hz, Wind Model: Impulse), the system was run with the hardware sensor and sensor model in two separate runs. The trajectory and velocity comparisons are displayed in Fig. 10. Comparing $\dot{X}$ and $\dot{Z}$ of the RoboBee over time, the reaction to the wind is similar, indicating good correlation between hardware and model. Any small differences in the

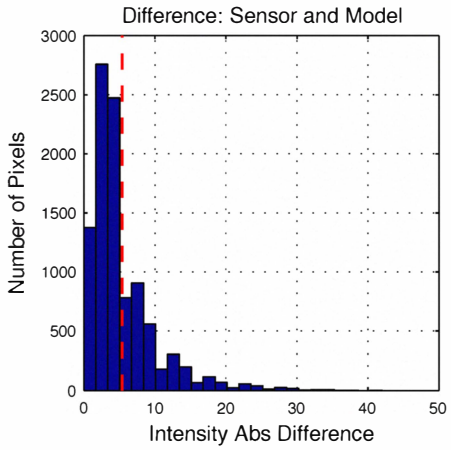

Fig. 9. Histogram showing differences between sensor and model. Dotted line indicates average difference. Lower differences indicate better correlation between model and actual sensor readings.

velocity tend to compound over time in the trajectory, and as we are using velocity control, it is a poor representation of performance.

IV. HWIL DESIGN SPACE EXPLORATION

In this section, we demonstrate the value of our HWIL system for exploring large design spaces. The design space of an optical flow sensor for stable hover of a RoboBee is large, but a subset of the overall RoboBee design space. Stable hover represents one of the more challenging behaviors for a MAV. The results described here point to general trends in the optical flow sensor design space, but the final RoboBee design is by no means fixed to the results of this study. We expect that exploring other aspects of the RoboBee design space will further refine the trends of this study. We aim to demonstrate that the HWIL system is valuable in exposing trends in large design spaces, and that the system can scale to explore increasingly larger portions of the RoboBee design space as more components are integrated into the system.

An optical flow sensor has many design parameters that affect computational complexity and ultimately power consumption, including the optical flow algorithm, sensor resolution, and sample rate of the optical flow sensor. We leverage the software sensor model to probe a greater range of design points. For each combination of parameters in Table I, a simulation was run for 120 seconds of virtual time and trajectory data similar to that of Fig. 10 was recorded. For each of the 240 simulations, the maximum and convergence time (Fig. 10) were found for $\dot{X}$, which represents the world velocity response in the direction of wind disturbance. Convergence was defined as the first sample where $|\dot{X}|$ stays under a threshold (0.1 m/s) for the following interval (0.5 s) of samples. Since the controller attempts to eliminate velocity caused by the wind disturbance, decreases in either metric are interpreted as a more successful system configuration.

Fig. 11 summarizes the parameter sweep results. The upper blue box plot summarizes the $\dot{X}$ convergence time metric

TABLE I
CONFIGURATION PARAMETER SWEEP

Parameter	Values Swept
Optical Flow Algorithms	II, LK
Square Sensor Resolution (Px)	32,64,128,256
Lens Field of View (Deg)	70,90,110
Loop Frequency (Hz)	40,60,80,100,120
Wind Direction	X
Wind Models	Impulse, Step
Max Wind Velocity (m/s)	3

and lower green box plot summarize the maximum $\dot{X}$ metric in the same manner. Each plot has a fixed algorithm and wind model. Within each plot, each bar fixes one variable, either resolution, field of view, or sensor frequency. The points forming the bar vary the remaining two parameters. The height of an individual box indicates the influence on system performance of that variable. A smaller box indicates that variable is the main driver of performance relative to the other two variables. Box plots across the range of one variable can be used to determine if there is a trend for varying one of resolution, field of view, or sensor frequency independent of the other two parameters. Then by comparing trends across box plots, the effect of optical flow algorithm and wind model on the identified trend can be determined.

Increasing resolution generally resulted in decreasing max $\dot{X}$ and $\dot{X}$ convergence time independent of other parameters. This trend can be explained by higher sensor resolution offering the optical flow algorithm more smooth changes in intensity. For the instance of II and step wind, max $\dot{X}$ showed an inverse parabola trend. The cause of this trend is worthy of further study with the HWIL system.

As the frequency of optical flow sensing (i.e. the image sensor frame rate) increases, both performance metrics show a slight decreasing trend independent of other parameters. This trend can be explained by higher sensor frequency offering better estimates of changes in body velocity.

Increasing field of view resulted in an increase in both performance metrics. This represents a non-obvious result, as generally wider field of view is desired for improved robotic control [19]. While we are still investigating the root cause of this behavior, we suspect this is due to increased sensitivity to rotation at larger fields of view affecting the translational measurement of optical flow. This result points to the utility of the HWIL system as this trend represents a counterintuitive result worthy of further study.

While LK generally has better performance compared to II, the difference is slight, especially considering the much larger computational complexity of LK. This is likely due to the averaging step applied to LK to generate a single optical flow vector. Since LK generates a dense optical flow field (one optical flow vector per pixel), the optical flow field is averaged to generate a single vector as input to the controller. In the future, the dense optical flow field from LK could be used to estimate other components of the body state, such as yaw, rather than rely on a separate sensor.

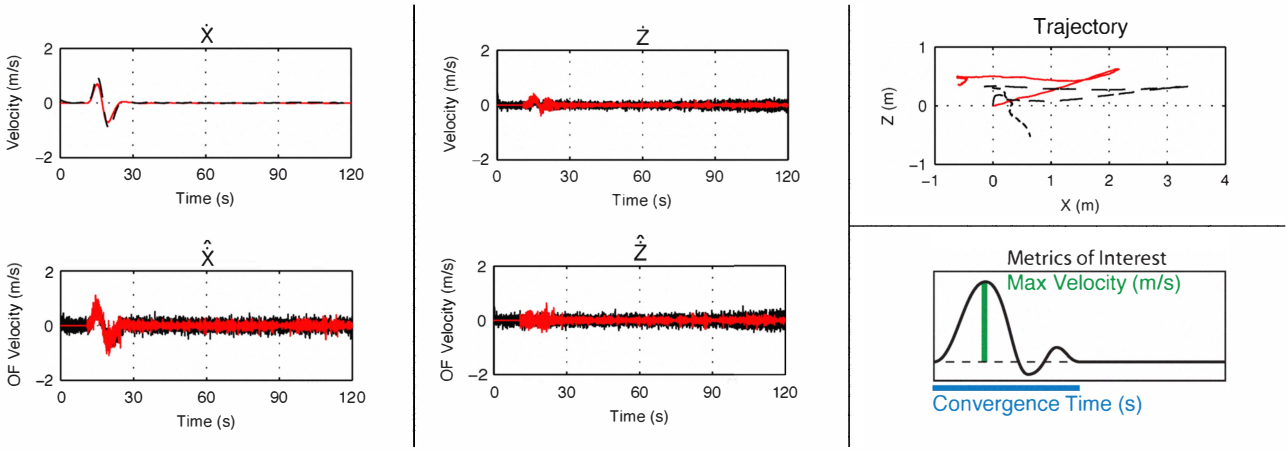

Fig. 10. Comparison of hardware sensor (dashed black) to software model (solid red) for one data point (Algorithm: LK, Resolution: 64 × 64 px, FOV: 90°, Frequency: 100 Hz, Wind Model: Impulse). Measurements are absolute world state: $\dot{X}$, $\dot{Z}$ and world state estimated by optical flow sensing: $\hat{\dot{X}}$, $\hat{\dot{Z}}$. Top Right: Trajectory in world. Bottom Right: Metrics of interest measured from $\dot{X}$ for Fig. 11.

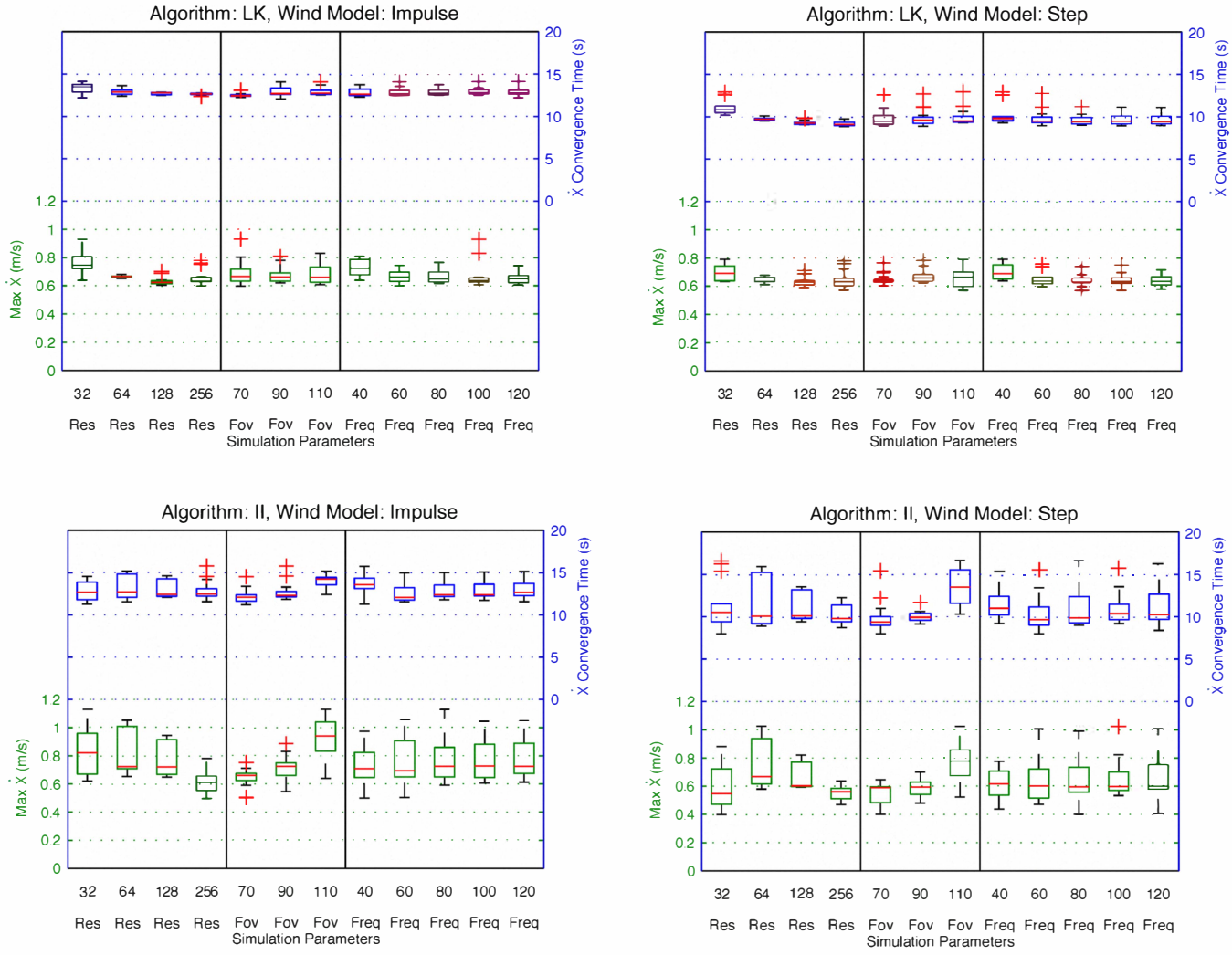

Fig. 11. Summary of parameter sweep. Upper Blue Box Plot: $\dot{X}$ convergence time metric. Lower Green Box Plot: Max $\dot{X}$ metric. Each plot has a fixed algorithm (top two plots are LK, bottom two plots are II) and wind model (left plots are impulse, right plots are step). Within a plot, three simulation parameters are evaluated: resolution, field of view, and sensor frequency. One bar is formed by all trials having the algorithm and wind model of the plot and the fixed simulation parameter of the x axis, while the remaining two simulation parameters vary.

There are diminishing returns in performance beyond a resolution of approximately 64 × 64 pixels and beyond a sensor frequency of 60 Hz. Although these will likely not represent the final operating points of the RoboBee, they do point to general trends in the impact of these variables. As the HWIL system is refined and more components are added, these will represent good starting points for continued refinement of these findings. The value of our HWIL system is that for a complex robotic system, where small changes in a vast number of parameters could significantly impact whole system performance, the HWIL system allows for trend exploration and whole robotic system optimizations.

V. CONCLUSION

Development of a robotic bee involves considerations of how a vast number of parameters impact whole system performance, weight, and power consumption. In this work, we propose our HWIL system as a framework to understand tradeoffs in the design parameter space within each component as well as for the system as a whole. The modularity of the system allows for different levels of hardware and virtual component realism, which promotes interdisciplinary collaboration. The HWIL grand vision was illustrated by an optical flow sensor design test case where impact of design parameters and trends in performance were exposed. We expect that building more detailed models of all system components and running significantly larger design space explorations will offer unique insights into MAV power and performance tradeoffs.

REFERENCES

[1] J. Whitney and R. Wood, "Aeromechanics of passive rotation in flapping flight," *J. Fluid Mechanics*, vol. 660, pp. 197–220, 2010.

[2] H. Tanaka and R. Wood, "Fabrication of corrugated artificial insect wings using laser micromachined molds," *J. Micromech. Microeng.*, vol. 20:7, 2010.

[3] B. Finio, J. Whitney, and R. Wood, "Stroke plane deviation for a microrobotic fly," in *IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, 2010.

[4] N. Perez-Arancibia, J. Whitney, and R. Wood, "Lift force control of a flapping-wing microrobot," in *American Controls Conf., San Francisco, CA*, June 2011.

[5] P. Sreetharan and R. Wood, "Passive torque regulation in an underactuated flapping wing robotic insect," in *Robotics: Science and Systems*, vol. 20:7, 2010.

[6] M. Karpelson, J. Whitney, G.-Y. Wei, and R. Wood, "Design and fabrication of ultralight high-voltage power circuits for flapping-wing robotic insects," in *Applied Power Electronics Conf., Fort Worth, TX*, Mar. 2011, pp. 2070–2077.

[7] M. Lyons, M. Hempstead, G. Wei, and D. Brooks, "The accelerator store framework for high-performance, low-power accelerator-based systems," *Computer Architecture Letters*, vol. 9, no. 2, pp. 53–56, 2010.

[8] K. Dantu, B. Kate, J. Waterman, P. Bailis, and M. Welsh, "Programming micro-aerial vehicle swarms with karma," in *to appear in the 9th Annual Conference on Embedded Networked Sensor Systems, Seattle, WA*, 2011.

[9] M. Srinivasan, S. Zhang, J. Chahl, S. G., and M. Garratt, "An overview of insect-inspired guidance for application in ground and airborne platforms," *Proc. Instn Mech. Engrs, Part G: J. of Aerospace Engineering*, vol. 218, no. 6, pp. 375–388, 2004.

[10] S. Han, A. D. Straw, M. H. Dickinson, and R. M. Murray, "A real-time helicopter testbed for insect-inspired visual flight control," in *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, May 2009, pp. 3055–3060.

[11] D. Jung, J. Ratti, and P. Tsiotras, "Real-time implementation and validation of a new hierarchical path planning scheme of UAVs via hardware-in-the-loop simulation," *Journal of Intelligent and Robotic Systems*, vol. 54, pp. 163–181, 2009.

[12] G. Cai, B. Chen, T. Lee, and M. Dong, "Design and implementation of a hardware-in-the-loop simulation system for small-scale UAV helicopters," in *Automation and Logistics, 2008. ICAL 2008. IEEE International Conference on*, Sept. 2008, pp. 29–34.

[13] L. Schenato, X. Deng, W. Wu, and S. Sastry, "Virtual insect flight simulator (VIFS): a software testbed for insect flight," in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, vol. 4, 2001, pp. 3885–3892.

[14] E. Johnson and S. Mishra, "Flight simulation for the development of an experimental UAV," in *Proceedings of the AIAA modeling and simulation technologies conference*, 2002.

[15] J.-C. Zufferey and D. Floreano, "Fly-inspired visual steering of an ultralight indoor aircraft," *Robotics, IEEE Transactions on*, vol. 22, no. 1, pp. 137–146, Feb. 2006.

[16] F. Bermudez and R. Fearing, "Optical flow on a flapping wing robot," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, 2009, pp. 5027–5032.

[17] M. Srinivasan, S. Zhang, J. Chahl, G. Stange, and M. Garratt, "An overview of insect-inspired guidance for application in ground and airborne platforms," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 218, no. 6, pp. 375–388, Jan. 2004.

[18] S. Zingg, D. Scaramuzza, S. Weiss, and R. Siegwart, "MAV navigation through indoor corridors using optical flow," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, May 2010, pp. 3361–3368.

[19] M. Reiser and M. Dickinson, "A test bed for insect-inspired robotic control," *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 361, no. 1811, pp. 2267–2285, 2003.

[20] C. McCarthy and N. Barnes, "Performance of optical flow techniques for indoor navigation with a mobile robot," *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, vol. 5, pp. 5093–5098, 2004.

[21] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. Black, and R. Szeliski, "A database and evaluation methodology for optical flow," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, Oct. 2007, pp. 1–8.

[22] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, "Performance of optical flow techniques," *International Journal of Computer Vision*, vol. 12, pp. 43–77, 1994.

[23] *Flying Qualities of Piloted Airplanes: MIL-F-8785C*, U.S. Military, 1980.

[24] (2011) Centeye website. [Online]. Available: http://centeye.com/

[25] G. Barrows, "Low profile camera and vision sensor," US Patent Application No. US 12/710073., Publication No. US 2011/0026141, February 2011.

[26] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *International joint conference on artificial intelligence*, vol. 3, 1981, pp. 674–679.

[27] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[28] M. Srinivasan, "An image-interpolation technique for the computation of optic flow and egomotion," *Biological Cybernetics*, vol. 71, no. 5, pp. 401–415, 1994.

[29] B. M. Finio and R. J. Wood, "Distributed power and control actuation in the thoracic mechanics of a robotic insect," *Bioinspiration & Biomimetics*, vol. 5, no. 4, 2010.

[30] W. Wu, L. Schenato, R. Wood, and R. Fearing, "Biomimetic sensor suite for flight control of a micromechanical flying insect: design and experimental results," in *IEEE International Conference on Robotics and Automation*, 2003, pp. 1146–1151.

[31] R. Wood, "Design, fabrication, and analysis, of a 3dof, 3cm flapping-wing MAV," in *IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, 2007.

[32] B. Finio, K. Galloway, and R. Wood, "An ultra-high precision, high bandwidth torque sensor for microrobotics applications," in *IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, 2011.