Visual Confined-Space Navigation Using an Efficient Learned Bilinear Optic Flow Approximation for Insect-scale Robots

Zhitao Yu[1], Gioele Zardini[2], Andrea Censi[2], Sawyer Fuller[1,3]

***Abstract*— Visual navigation for insect-scale robots is very challenging because in such a small scale, the size, weight, and power (SWaP) constraints do not appear to permit visual navigation techniques such as SLAM (Simultaneous Localization and Mapping) because they are likely to be too power-hungry. We propose to use a biology-inspired approach, which we term the bilinear optic flow approximation, that is more computationally efficient. We build on previous work that has shown that the bilinear approximation can be used for visual servoing. Here, we show that a bilinear approximator can be learned that is able to stabilize the heading of a robot while performing continuous forward motion in a corridor-shaped environment. This is a necessary capability for confined-space navigation that insect-sized robots are likely to perform. In this work, we describe the underlining methodology of the method and built a 2D visual simulation environment and omnidirectional camera model to validate our results.**

I. INTRODUCTION

Compared with the large robots, insect-scale robots have advantage of lower cost because the material cost is dramatically reduced and also they can perform many tasks better because of the small size such as locating the leaks of the gas of interest in dense piping infrastructure, space exploration with much lower cost, and replacing fixed-in-place air sensors in urban environments. However, the reason why the insect-scale robots are still in research stage rather than industry deployed is because as the scale reduces dramatically, the sensor feedback quality and energy available for computation also reduces drastically, which makes the realization of insect-scale robots challenging to have the similar performance as the larger robots. For example, larger drones can be equipped with several sensors such as Lidars, thermal sensors and the global positioning system (GPS), but for insect-scale robots, most of those sensors are either too power-hungry or too heavy. Also, especially for indoor environment, the GPS signal cannot be accurate enough [1]. One of our contributions is to solve the challenge for insect-scale robots to do navigation given size, weight, and power (SWaP) constraints.

Visual navigation is one aspect of visual servoing, which uses the information from vision sensors to control robot motion. Visual information has been used as an important input to robotic systems to do navigation [2], manipulation [3], and human-robot interaction tasks [4]. However, most of the algorithms are for large robots without tight SWaP constraints. In nature, the fly is good at visual navigation under those constraints. Flies have omnidirectional eyes that can perceive visual information and then do navigation in surroundings [5]. Bio-inspired by flies, we do visual navigation with insect-scale robot with an omnidirectional camera using an optic flow estimation. Here, we are interested in a corridor following task (Fig. 2) because it is the essential behavior required for navigating between obstacles in a cluttered environment. A reflexive corridor following controller could operate in real-time, freeing a high-level controller to pursue long range goals such as search or path planning.

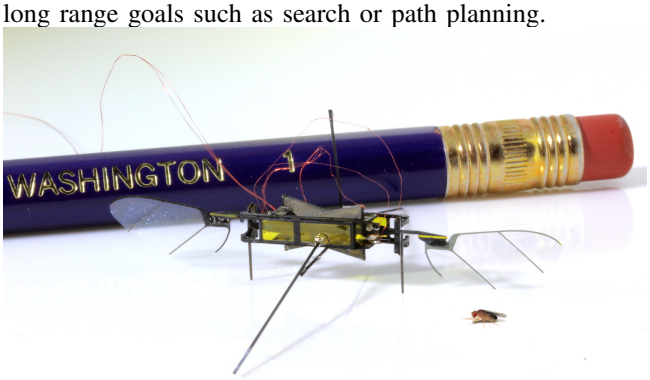

Fig. 1. The 143 mg U. Washington Robofly (pencil in background for scale) weighs a bit more than a honeybee (101 mg). We aim to perform visual flight control with the very low compute power available on robots of this size and below.

Our proposed control architecture takes inspiration from fruit flies. Rather than using state estimation derived from a stored map (SLAM), we rely on optic flow, a measure of the velocity of motion of visual scenery as the robot or animal moves through it. Optic flow requires less computation and provides a crude measure of distances to obstacles [6]. This provides enough information to carry out many tasks, such as source seeking [7] and navigating through indoor and outdoor environments [8]. Lucas-Kanade [9] is a simple and effective [10] method to estimate optic flow from raw pixels. To reduce computation power still further for application on a gnat robot [11], in this work we employ an insect-inspired simplification. Known as the Hassenstein-Reichard correlator [12], it entails only multiplying derivatives of pixel readings and predicts a number of aspects of insect behavior [13].

We assume our robot must be controlled by a computer carried onboard, rather than a remote computer to which data is transmitted wirelessly, for three important reasons. First and most importantly, many anticipated applications

*This research was partially supported by the National Science Foundation (award number FRR-2054850).

[1]Department of Mechanical Engineering, University of Washington, Seattle, WA, USA {zhitaoyu, minster}@uw.edu

[2]Institute for Dynamic Systems and Control, ETH Zürich, Zürich, Switzerland {gzardini, acensi}@ethz.ch

[3]Paul G. Allen School of Computer Science, University of Washington, Seattle WA 98195, USA

for small flying robots are indoors or in confined spaces, where a consistent wireless channel is not available. Second, even with a reliable channel, wireless transmission consumes excessive power. For example, consider the 71 mg Robofly (Fig. 1), which weighs less than a honeybee and requires 60 mW to fly [14]. Transmitting low-resolution (160×120 pixel) video at 1–5 frames per second over low power Bluetooth consumes 4–18 mW [15]. Even at this low frame rate, which is too low for many control tasks, wireless transmission represents an excessive fraction of the power needed to simply stay in the air. And matters become far worse as scale diminishes to that of a gnat. Third, an important advantage to small robots is their potential to be deployed in large numbers; this advantage is largely nullified if they are forced share a crowded communication channel and an over-burdened server.

The platform we have in mind for the ensuing analysis and simulation is extremely small and power-constrained flying robots with the size of insects. Such robots, whether actuated by wings or other means such as electrohydrodynamic thrust, are dynamic and unstable as a general rule [16], [17], [18]. For this work, we build on parallel work, under review [19], that introduces an ultra-lightweight sensor suite consisting of an accelerometer and a downward-facing optic flow camera to stabilize these unstable hover dynamics. The sensor suite provides an estimate, and therefore the ability for the robot to control, lateral velocity. This represents the first level of control autonomy, "sensor autonomy," proposed in the hierarchy introduced in [20]. Here we are concerned with the next level up, in which the robotic agent senses and responds to the external environment. This is known as "reactive autonomy."

We assume therefore that the task to be solved here is to provide an outer-loop controller that provides control inputs in the form of desired lateral and rotational velocity values to the inner loop hovering controller. We also assume that the inner-loop controller is capable of controlling the heading velocity $\dot{\phi}$. On a larger drone equipped with a full inertial measurement unit (IMU), this would be done using a feedback control loop in which the angular velocity was measured using a MEMS gyroscope. However, we are targeting extremely small aircraft, which may not be compatible with the power draw (tens of mW) and mass (tens of mg) of current MEMS gyroscope technology. We assume that the $\dot{\phi}$ is instead measured using the downward-facing optic flow camera, trained on rotating imagery in a relatively straightforward extension of the work derived here.

Previous work has introduced a "biologically plausible" means to use motor babbling to learn a simple, bilinear model that approximates optic flow patterns observed by omnidirectional cameras during self motion purely by observing pixel readings and their derivatives. This approach has been used to stabilize attitude [21] as well as pose [22] in simulation and on physical robots [23], also known as "visual servoing."

Here, our contribution is to show that a similar approach can learn how to stabilize a state of continuous motion down the length of a confined or cluttered corridor-shaped environment.

II. RELATED WORK

Some robot navigation works use the sensors like RGB-D camera [24] or Lidars [25]. In the environment where GPS cannot work well, a state estimation with the Lidar and IMU was proposed [26]. However, for insect-scale robots, those sensors are too heavy and power consumption is too high.

Most works in visual navigation are to create a map on surrounding environment using sensors, and simultaneously robots can localize itself in the map anytime. This is the process called Simultaneous Localization and Mapping (SLAM) [27][28]. A SLAM implementation for generating 3D map for unknown environment is introduced in [29]. However, SLAM is a heavy computation process and also require a significant memory, thus for insect-scale robots, it is not ideal to use SLAM to do navigation.

Recently, deep learning approach emerges in robot navigation tasks. Dorbala *et al.* [30] proposed a deep learning approach using a convolutional neural network (CNN) to do corridor following task with wheeled robots. Saxena *et al.* [31] presented an end-to-end deep learning based approach for visual servoing by training CNN on colored images. However, deep learning approach only works when there are sufficient data for training and if the distribution of the test dataset is different from the training dataset, the deep learning approach will not work well. Because deep learning architecture usually has lots of parameters, it may consume significant computation power.

Given the SWaP constraints, for insect-scale robots, researchers utilized bio-inspired approaches. Fuller *et al.* [32] showed that insect-inspired visual autocorrelation can navigate a hovercraft robot through a corridor successfully. Censi *et al.* [21] proposed a bilinearly estimation approach using optic flow to stabilize visual attitude. To best of our knowledge, There is no research on navigating an insect-scale robot through a corridor, which is the main task of our paper.

III. METHODOLOGY

In this section, we will introduce the kinematics and autocorrelation-based visual control.

A. Kinematics and control

We developed a kinematics model for the corridor following task. The dynamics read:

$$\begin{cases} \dot{x} = u_x \cos\phi - u_y \sin\phi, \\ \dot{y} = u_x \sin\phi + u_y \cos\phi, \\ \dot{\phi} = u_\phi. \end{cases} \tag{1}$$

,where $u = [u_x, u_y, u_\phi]$ is the control input for the velocity in body frame. And $\dot{x}, \dot{y}$ and $\dot{\phi}$ are the velocity in world frame. To control the dynamics, we use a proportional controller:

$$u_\phi = K_{p\phi}(0 - \hat{\phi}). \tag{2}$$

We are mainly interested in geting an estimation $\hat{\phi}$.

B. Bilinear optic flow approximation

We would like to get estimation of $\hat{\phi}$ using pure optic flow information. We have the precise relation for how luminance intensity varies with time due to camera motion given by Eq. (3) [33]:

$$\dot{\boldsymbol{l}} = (\boldsymbol{s} \times \nabla_{\boldsymbol{s}} \boldsymbol{l}) \cdot \boldsymbol{\omega} + \mu(\boldsymbol{s}, \boldsymbol{p}) \nabla_{\boldsymbol{s}} \boldsymbol{l} \cdot \boldsymbol{v}, \tag{3}$$

where $\mu(\boldsymbol{s}, \boldsymbol{p})$ is the reciprocal of the distance to the visual element, known as the nearness. $\boldsymbol{s}$ is a continuous index ranging over the "sensel space" $\mathcal{S}$, which is a unit sphere in this case denoted as $\mathbb{S}^2$. Nearness depends on the geometry of the environment and the pose $\boldsymbol{p}$ (position and orientation) of the robot.

To simplify control, we consider a Taylor series approximation of Eq. (3), which is nonlinear, in terms only of $\boldsymbol{l}$ and $\boldsymbol{u}$, that is, the luminance readings and the control inputs. The intent is to approximate the "average" behavior of Eq. (3) across the typical distributions encountered in our environment of interest. These include, in particular, the shape of the nearness function μ, the image contrast, and the pose of the vehicle. The first terms of the Taylor series are given by

$$\dot{\boldsymbol{l}} \approx A + B\boldsymbol{u} + C\boldsymbol{l} + \boldsymbol{u}M\boldsymbol{l}, \tag{4}$$

where $\boldsymbol{u} = [\boldsymbol{\omega}, \boldsymbol{v}]^\intercal \in \mathbb{R}^p$ is the linear and angular velocity vectors of the robot (p = 6 for the 3D case; for 2D planar motion, p = 3). Assuming $\boldsymbol{l}$ must be discretized by, for example, a digital imaging surface, then $\boldsymbol{l} \in \mathbb{R}^n$, and $\nabla_s \boldsymbol{l}$ is replaced with its discrete 1D approximation $\frac{dl}{d\gamma} \approx \frac{l_{k+1} - l_{k-1}}{2\delta\gamma}$, which can be represented as a matrix operator $D\boldsymbol{l}$, so that $M \in \mathbb{R}^{p \times n \times n}$ is the product of some matrix times D. It can be shown that the first three terms, in terms of A, B, and C, are zero, leaving only the last term, which is bilinear in $\boldsymbol{l}$ and $\boldsymbol{u}$. The bilinear approximation can be written in terms of 2D matrices as

$$\dot{\boldsymbol{l}} = \sum_{i=1}^{p} M_i \boldsymbol{l} u_i, \tag{5}$$

where the M_i are p different $n \times n$ matrices.

If the M_i's in Eq. 5 above can be found, then they can be used in the to estimate the state of motion $\boldsymbol{u}$. The least-squares solution is

$$\hat{\boldsymbol{u}}_i = -\left((M_i \boldsymbol{l})^\intercal M_i \boldsymbol{l}\right)^{-1} \boldsymbol{l}^\intercal M_i \dot{\boldsymbol{l}}. \tag{6}$$

This can be compared to the Lucas-Kanade method of optic flow estimation [9], but where the estimation is of a state geometrical motion rather than simple translational motion relative to a flat surface. It was shown in [34] and subsequent work that it is possible use a simpler estimator that eliminates the matrix inversion step, leaving the bilinear term

$$\hat{\boldsymbol{u}}_i = -c\boldsymbol{l}^\intercal M_i \dot{\boldsymbol{l}}, \tag{7}$$

where c is inverse of the average image contrast: $c = (\mathbb{E}_m\{||\nabla \boldsymbol{l}||_2^2\})^{-1}$. Eq. (7), when approximated using discrete pixels and when the derivative is computed at discrete time increments, is mathematically identical to a Hassenstein-Reichardt correlator in which the lag is a pure delay [34].

Previous work [34], [22], [23] showed that, with mild conditions on the shape of μ, it is possible to derive a controller that can locally drive the robot to a pose in which its luminance readings match those of a goal "snapshot" $\boldsymbol{g}$ using bilinear estimator. This is known as visual servoing.

For the case of 2D planar motion, for example, $\boldsymbol{u} = [\dot{\phi}, \dot{x}, \dot{y}]^\intercal$. To perform visual servoing, that is, driving the pose $\boldsymbol{p}$ to become coincident with a goal pose $\boldsymbol{p}_g$, a "goal" luminance snapshot, $\boldsymbol{l} = \boldsymbol{g}$ must be available. The proportional controller for a kinematic system (defined only by velocities) is simply $\boldsymbol{u}_i = \boldsymbol{l}^\intercal M_i(\boldsymbol{g} - \boldsymbol{l})$. This approach can also stabilize an inertial, dynamic system driven by forces and torques, where $\boldsymbol{u} = [\boldsymbol{\tau},\ \boldsymbol{f}] \in \mathbb{R}^6$ (3D) or $\in \mathbb{R}^3$ (2D planar motion). A stabilizing proportional-derivative (PD) visual servoing controller has the form $\boldsymbol{u}_i = \boldsymbol{l}^\intercal M_i(\boldsymbol{g} - \boldsymbol{l} - k_p \dot{\boldsymbol{l}})$, where k_p is the proportional feedback gain [22]. In other words, the matrices M_i capture luminance correlations. That is, they encode how luminance readings change with time, and simultaneously and as a result, also the direction of the error gradient between two displaced images if the displacement is small.

C. Learning the M matrices

We learn the matrices by detecting correlations between pixel readings, their derivatives, and randomly-chosen motor inputs through "motor babbling." We have $\hat{\boldsymbol{u}} = c\boldsymbol{l}^\intercal M \boldsymbol{l}$ Then [21] shows that $\hat{\boldsymbol{u}}$ is an unbiased estimator of $\boldsymbol{u}$.

We would like to learn a matrix M to minimize the expected estimation error $\mathbb{E}\|\dot{\boldsymbol{l}}^\intercal M_i \boldsymbol{l} - \boldsymbol{u}\|_2^2$, for each component of $\boldsymbol{u}$, we can learn M_i according to the error function:

$$\mathbb{E}\{(\dot{\boldsymbol{l}}^\intercal M_i \boldsymbol{l} - u_i)^2\}. \tag{8}$$

Given a particular training sample $(\boldsymbol{u}, \dot{\boldsymbol{l}}, \boldsymbol{l})$, M_i can be updated according to a stochastic gradient descent (SGD), which consists in minimizing Eq. (9) only with respect to the particular sample. The update rule is

$$\dot{M}_i = \lambda(\dot{\boldsymbol{l}}^\intercal M_i \boldsymbol{l} - \boldsymbol{u}_i)\dot{\boldsymbol{l}}\boldsymbol{l}^\intercal, \tag{9}$$

where λ is the learning rate. Through repeated exposures to randomly-chosen inputs ("motor babbling"), this increases the value of elements of M_i in which $u_i, \boldsymbol{l}$, and $\dot{\boldsymbol{l}}$ are correlated. It is instructive to think of the product $\boldsymbol{l}_j \dot{\boldsymbol{l}}_k$ as the "optic flow": if the luminance is increasing in time at location j and it is high at a nearby location k, then this corresponds to positive optic flow, for example.

D. Estimating ϕ

We now expand the space of quantities to be estimated to include not just velocities, but other states that are relevant to the task of corridor following, that is, stabilizing those

variables that the inner-loop hovering controller (see Introduction) is not able to estimate on its own. These consist of y, the lateral position of the robot, and ϕ, the heading angle of the robot relative to the axis of a corridor-shaped environment along which it is traveling. Our treatment builds on [32], which indicated that an autocorrelation weighting function exists that can provide an estimate of ϕ for use in control. That work used a frequency-domain analysis of an autocorrelation scheme that only considered nearest-neighbor pixels. Here we provide a more general solution by training the weighting function in an *abinitio* fashion. This has the benefit that statistics of the scenery as well as the camera calibration itself are all learned at once.

We assume that the environment geometry resembles a narrow corridor, and that the vehicle is equipped with an additional omnidirectional camera. The bilinear model will be used to estimate components of the robot's state that cannot be observed with the sensor suite. We would like to estimate the robot orientation ϕ by using the bilinear optic flow estimate $\hat{\phi}$.

We use a Taylor expansion as before, this time as a three-term expansion in $\boldsymbol{u}$, $\boldsymbol{p}$, and $\boldsymbol{l}$, and have

$$\dot{\boldsymbol{l}} = M_\omega \boldsymbol{l}\omega + (M_\phi \boldsymbol{l}\phi + M_y \boldsymbol{l}y)^\intercal \boldsymbol{v}. \tag{10}$$

We linearize at $\boldsymbol{u_0} = [\omega, v_x, v_y] = [0, 0.1m/s, 0]$ and $y = 0$, then we have

$$\dot{\boldsymbol{l}} = M_\phi \boldsymbol{l}\phi v_x, \tag{11}$$

where v_x can be incorporated into M_ϕ, then we can have the estimation $\hat{\phi} = \boldsymbol{l}^\intercal M_\phi \dot{\boldsymbol{l}}$, where M_ϕ can be learned using the method in last section.

IV. Experiments

A. Set up the scene

A diagram of the robot, its corridor-shaped environment, and the camera model is shown in Fig. 2. For all simulation tasks, we extracted a single horizontal line of pixels from a photograph from an outdoor scene and projected these onto both of the corridor walls.

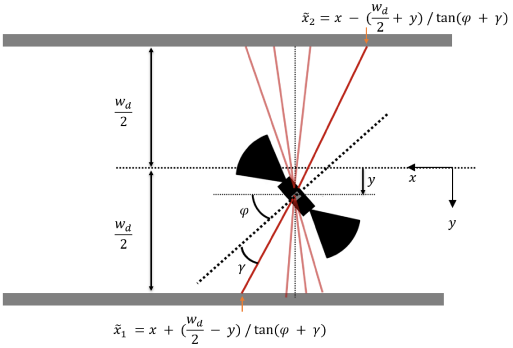

Fig. 2. A diagram of the robot in its environment (top view). The robot is assumed to move continuously in the positive x-direction ($\dot{x} > 0$. The red lines indicate directions of pixel readings taken by its omnidirectional camera. These are comparable to the ray casting algorithm used in computer graphics nomenclature. The solid dashed line is the heading angle ϕ of the robot. $\tilde{x}_1$ is the position of the intersection for the left wall, $\phi + \gamma < \pi$ and $\tilde{x}_2$ is the position of the intersection for the right wall, $2\pi > \phi + \gamma > \pi$.

The parameters are:

1) w_d: distance between walls = 0.1 m.
2) i_s: The image size is set to 1 m.
3) q : State of the robots. $q = [\phi, \dot{\phi}, x, \dot{x}, y, \dot{y}]$
4) γ : The angle of each ray relative to the heading direction.

The implementation procedure is briefly described as below:

1) The angle for the ray according to the x-axis is $\phi + \gamma$.
2) If $\phi + \gamma < \pi$, then the ray intersects the left wall, and intersection position on the wall is given by

$$\tilde{x}_1 = x + (w_d/2 - y)/\tan(\phi + \gamma). \tag{12}$$

If $2\pi > \phi + \gamma > \pi$, then the wall intersection position is

$$\tilde{x}_2 = x - (w_d/2 + y)/\tan(\phi + \gamma). \tag{13}$$

3) Then we can use $\tilde{x}$ to get the luminance readings of each ray (corresponding pixel value on the image).
4) We also need to calculate the distance from the robot to the position of the intersection of the ray and the wall . If $\phi + \gamma < \pi$, then we have the distance d is given by

$$d = \sqrt{\frac{w_d/2 - y}{\sin(\phi + \gamma)}}. \tag{14}$$

If $2\pi > \phi + \gamma > \pi$, then we have the distance d calculated by Eq. (15).

$$d = \sqrt{\left|\frac{w_d/2 + y}{\sin(\phi + \gamma)}\right|}. \tag{15}$$

5) We also need to know the direction of each ray to calculate $\nabla_s l$. For each ray, the direction is $(\cos(\phi + \gamma), \sin(\phi + \gamma))$.

B. Validating the camera model

We constructed a simulation environment in Python. To confirm that our simulation was correctly performing the necessary numerical operations, we confirmed that its outputs satisfied Eq. (3) in 2D. We describe briefly this process. The parameters are:

1) $\dot{\boldsymbol{l}}$: The time derivative of the luminance readings.
2) $\boldsymbol{s}$: The direction, which can be represented as a vector on the unit sphere. In general, $\boldsymbol{s} \in S$ is the sensel direction and S is the sensel space.
3) $\nabla_s \boldsymbol{l}$: The spatial derivative of the luminance readings.
4) $\boldsymbol{\omega}$: The angular velocity.
5) $\mu(\boldsymbol{s}, \boldsymbol{p})$: The reciprocal of the distance to the visual element, also called nearness.
6) $\boldsymbol{v}$: The translational velocity of the robot.

The left hand side of Eq. (3) can be calculated by:

$$LHS = \dot{\boldsymbol{l}} \approx \frac{\boldsymbol{l}(t - 2dt) - \boldsymbol{l}(t)}{2dt}. \tag{16}$$

As we know the right hand side of Eq. (3) is :

$$RHS = (\boldsymbol{s} \times \nabla_{\boldsymbol{s}} \boldsymbol{l}) \cdot \boldsymbol{\omega} + \mu(\boldsymbol{s}, \boldsymbol{p}) \nabla_{\boldsymbol{s}} \boldsymbol{l} \cdot \boldsymbol{v}. \tag{17}$$

The first term of Eq. (17) can be calculated as

$$(\boldsymbol{s} \times \nabla_{\boldsymbol{s}} \boldsymbol{l}) \cdot \boldsymbol{\omega} = \Big[(\cos(\phi+\gamma), \sin(\phi+\gamma)) \times \\ (\frac{l_{2,i+1} - l_{2,i-1}}{2d\gamma} \cdot (-\sin(\phi+\gamma), \cos(\phi+\gamma)) \Big] \cdot \dot{\phi}. \tag{18}$$

Note that we have used the two-dimensional version of the cross product "×", which results in a scalar. This can be shown by promoting the 2D vectors to 3D according to $[x_1, y_1, 0] \times [x_2, y_2, 0]$, which produces a vector that only has a nonzero value along its z direction; the output of the 2D cross product is the value along the z direction. The second term of Eq. (17) can be calculated as

$$\mu(\boldsymbol{s}, \boldsymbol{p}) \nabla_{\boldsymbol{s}} \boldsymbol{l} \cdot \boldsymbol{v} = \mu(\boldsymbol{s}, \boldsymbol{p}) \frac{l_{2,i+1} - l_{2,i-1}}{2d\gamma} \cdot \\ \Big[(-\sin(\phi+\gamma), \cos(\phi+\gamma)) \cdot (v_x, v_y) \Big]. \tag{19}$$

We select time step $dt = 1/200$ s and pick the ray index 5 (the 6th ray) and then propagate the time for 50 dt to see whether the LHS is equal to RHS on those 50 time points. We set $y = 0.05$ m, $\dot{y} = 0.05$ m/s and set $\phi = 5°$, $\dot{\phi} = 100°/s$ and $x = 0.25$ m, $\dot{x} = 0.1$ m/s. The validation result is shown in Fig. 3. We can see they match well.

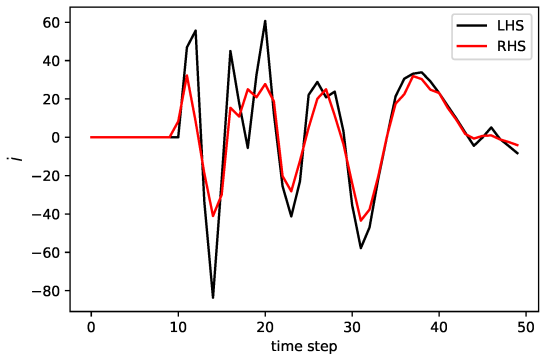

Fig. 3. Validation of the camera model. We have RHS $= (\boldsymbol{s} \times \nabla_{\boldsymbol{s}} \boldsymbol{l}) \cdot \boldsymbol{\omega} + \mu(\boldsymbol{s}, \boldsymbol{p}) \nabla_{\boldsymbol{s}} \boldsymbol{l} \cdot \boldsymbol{v}$. And we have LHS $= \dot{l}$. We can see the RHS can follow LHS well.

C. Learning M_ϕ

Because we only need to estimate ϕ, thus we only need to learn the correlator matrix $M_\phi \in \mathbb{R}^{n \times n}$. We can rewrite learning rule Eq. (9) to

$$\dot{M}_\phi = \lambda \phi \dot{\boldsymbol{l}} \boldsymbol{l}^\intercal. \tag{20}$$

We use a motion babbling strategy to develop the tranining dataset. For each iteration, we random sample x, y, ϕ and $\dot{\phi}$ in a certain range and use the sampled values to do ray casting to get $\dot{l}$ and l. Then we can calculate $\dot{M}_\phi$ to update M_ϕ by

$$M_\phi = M_\phi + \dot{M}_\phi dt. \tag{21}$$

We set $\lambda = 1, dt = \frac{1}{200}$ s, and we sample x around $x = 0.5$ m with range 0.3 m, y around the middle of the corridor $y = 0$ m with range 0.03 m, ϕ around heading forward $\phi = 0°$ with range 30° and $\dot{\phi} = 0$ with range 114.6°/s. We trained 20000 iterations to make M_ϕ asymptotically approach a form, up to a constant, that has the appearance shown for example in Fig. 4. The learned M_ϕ matrix is shown in Fig. 4. The shape is a skew-symmetric matrix matching the derivation in [21].

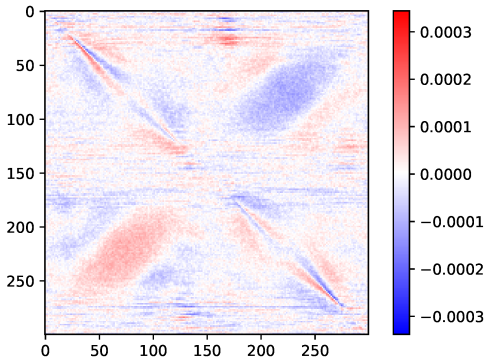

Fig. 4. The learned M_ϕ matrix. According to the derivation in [21], it should be a skew-symmetric matrix. We can see our simulation matches the derivation.

D. Control simulation (stablizing ϕ)

We constructed a simulation of the dynamics, Eq. (2), using fixed-step Euler Integration, to explore the stabilization behavior. For the simulation experiment, we set the initial condition $x_0 = 0.2$ m, $y_0 = 0.0$ m, $\dot{x}_0 = 0.1$ m/s, $\dot{y}_0 = 0.0$ m/s, $\dot{\phi}_0 = 0$ and we set different initial conditions for ϕ with a range of initial angles to get different trajectories to see whether ϕ can converge to zero. In this simulation, we only control $\dot{\phi}$ according to Eq. (2). We set the simulation time period is 4 s, the time step is $\frac{1}{200}$ s. $K_{p\phi} = 5$.

Fig. 5 shows that our approach is able to steer ϕ to a within approximately 5° of zero when initiated across a range of initial conditions. Then we plot the ϕ vs time in Fig. 5, we can see all trajectories converge to somewhere around $\phi = 0$

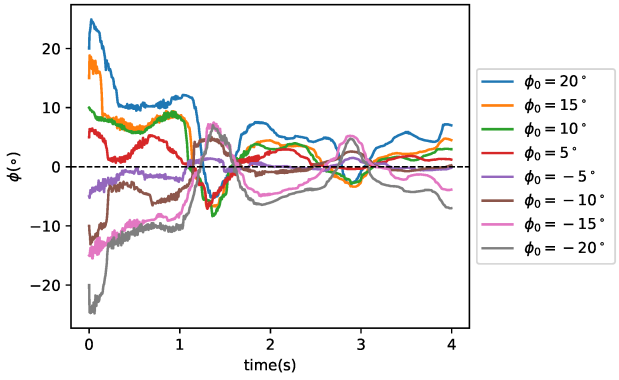

Fig. 5. ϕ vs time. There are six trajectories with different initial conditions shown in the right bar.

To show that our optic flow estimation can estimate ϕ well, we plot ϕ and $\hat{\phi}$ in the same plot for two trajectories ($\phi_0 = 5°$ and $\phi_0 = 15°$) shown in Fig. 6 and Fig. 7.

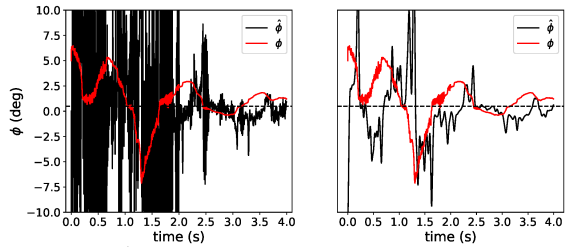

Fig. 6. ϕ and $\hat{\phi}$ for the trajectory with initial condition $\phi_0 = 5°$. The left figure is plotting ϕ and raw $\hat{\phi}$. The right figure is plotting ϕ and Gaussian filtered $\hat{\phi}$ to remove spurious noise.

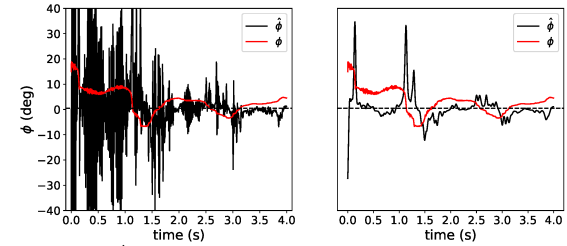

Fig. 7. ϕ and $\hat{\phi}$ for the trajectory with initial condition $\phi_0 = 15°$. The left figure is plotting ϕ and raw $\hat{\phi}$. The right figure is plotting ϕ and Gaussian filtered $\hat{\phi}$ to remove spurious noise.

The Fig. 8 shows that the robot can navigate through the corridor successfully given the initial heading angle $\phi_0 = 15°$.

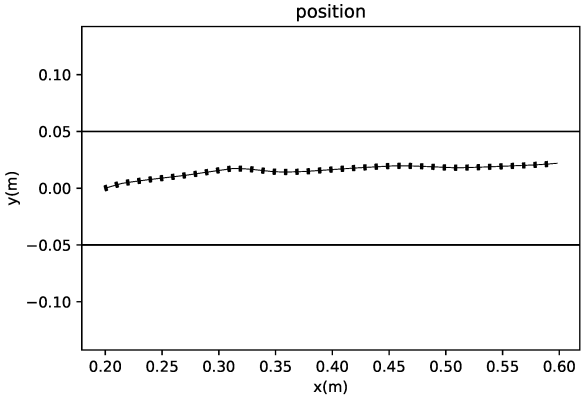

Fig. 8. An example trajectory of the simulated robot between the corridor walls shows that it is able to align itself with the long axis of the corridor.

V. Conclusion and Future Work

Our paper has shown that using purely the optic flow estimation, we could stabilize the heading ϕ of a simulated insect-scale robot as it moves through a corridor. The bilinear optic flow approximation method we used has the advantage that it is end-to-end: a process of motor babbling allows it to learn pixel correlations regardless of their orientation, so it therefore requires no camera calibration. On a physical system, the necessary M matrices can be learned by moving the robot around while recording state using a motion capture arena or temporary on-board sensors such as tiny laser rangefinders [35]. The approach is also therefore easy to generalize to 3D flight control.

Our approach builds on an assumption that it is operating as an "outer-loop" controller that sends desired translational and rotational velocities to a fast "inner loop" that takes care of stabilizing the unstable dynamics of such a robot and regulating flight speeds. How the hovering controller is able to operate under the extreme SWaP constraints of an insect-sized robot are described in greater detail in [19]. We believe our approach is suited to tiny robots because it operates entirely using multiply and add operations, which are relatively fast and efficient computational operations. The learned M matrices are skew-symmetric and largely sparse, so when used in practice we anticipate that near-zero entries will be zeroed out, and only half of the matrix needs to be stored in memory. Further optimizations may be possible by neglecting entries that correspond to pixel readings that are far apart. We anticipate that this will allow our method to do navigation in real time using the constrained capabilities of an onboard, general purpose microcontroller.

The learned M_ϕ appears to be performing the same computational operation as the ϕ estimator derived in [32], which itself was inspired by [36]. All three use a similar method to determine the location of the centroid of the optic flow pattern in the lateral directions. As the robot moves down the corridor, it is subject to an optic flow pattern that is highest in the lateral directions (directly to the left and right when facing forward) and zero straight ahead and behind. Therefore the centroid of this peak represents an indication of the direction one is facing when moving forward. Locating the centroid entails multiplying a function $f(x)$ by x and integrating ($\int x f(x) dx$). A weighting gradation, ranging from positive to zero to negative can be observed along the diagonal of M_ϕ, where the zero is at the lateral ray directions (ray elements 75 and 225 in Fig. 4).

Future work will aim to expand the the capabilities to also stabilize the lateral position y. One promising approach is to use deeper network than a bilinear model, such as using higher order terms. We also note that there is a confound in that both $\dot{\phi}$ and y produce the same average pattern of optic flow [32]. We anticipate that it may be possible to eliminate this effect by taking advantage of the downward-facing camera that is needed for stable hover to estimate, and somehow eliminate the confounding effect of $\dot{\phi}$. An important strength of the present work is that it requires almost no effort to extend it to 3D motion control. To do so we will construct a 3D graphical simulator, and later apply the results to a physical robot fly prototype we have created in our laboratory [37], [38].

References

[1] A. Beyeler, J.-C. Zufferey, and D. Floreano, "Vision-based control of near-obstacle flight," *Autonomous robots*, vol. 27, no. 3, pp. 201–219, 2009.

[2] Q. Bateux, E. Marchand, J. Leitner, F. Chaumette, and P. Corke, "Training deep neural networks for visual servoing," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 3307–3314.

[3] A. Al-Shanoon and H. Lang, "Robotic manipulation based on 3-d visual servoing and deep neural networks," *Robotics and Autonomous Systems*, p. 104041, 2022.

[4] L. Shi, C. Copot, and S. Vanlanduit, "A bayesian deep neural network for safe visual servoing in human–robot interaction," *Frontiers in Robotics and AI*, vol. 8, 2021.

[5] G. K. Taylor and H. G. Krapp, "Sensory systems and flight stability: what do insects measure and why?" *Advances in insect physiology*, vol. 34, pp. 231–316, 2007.

[6] J. J. Koenderink, "Optic flow," *Vision Research*, vol. 26, no. 1, pp. 161–180, 1986.

[7] M. J. Anderson, J. G. Sullivan, T. Horiuchi, S. B. Fuller, and T. L. Daniel, "A bio-hybrid odor-guided autonomous palm-sized air vehicle," (in press).

[8] H. D. Escobar-Alvarez, N. Johnson, T. Hebble, K. Klingebiel, S. A. P. Quintero, J. Regenstein, and N. A. Browning, "R-advance: Rapid adaptive prediction for vision-based autonomous navigation, control, and evasion," *Journal of Field Robotics*, vol. 35, no. 1, pp. 91–100, 2018. [Online]. Available: http://dx.doi.org/10.1002/rob.21744

[9] B. D. Lucas, T. Kanade, *et al.*, "An iterative image registration technique with an application to stereo vision." in *IJCAI*, vol. 81, 1981, pp. 674–679.

[10] J. L. Barron, D. J. Fleet, S. S. Beauchemin, and T. Burkitt, "Performance of optical flow techniques," in *Proceedings 1992 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 1992, pp. 236–237.

[11] A. M. Flynn, "Gnat robots (and how they will change robotics)," 1987.

[12] B. Hassenstein and W. Reichardt, "Systemtheoretische analyse der zeit-, reihenfolgen- und vorzeichenauswertung bei der bewegungsperzeption des russelkafers chlorophanus," *Zeitschrift Fur Naturforschung*, vol. 11b, pp. 513–524, 1956.

[13] E. Buchner, *Photoreception and Vision in Invertebrates*. Plenum, 1984, ch. Behavioral Analysis of Spatial Vision in Insects, pp. 561–621.

[14] Y. M. Chukewad, J. M. James, A. Singh, and S. B. Fuller, "Robofly: An insect-sized robot with simplified fabrication that is capable of flight, ground, and water surface locomotion," *IEEE Transactions on Robotics*, 2021, (in press).

[15] V. Iyer, A. Najafi, J. James, S. Fuller, and S. Gollakota, "Wireless steerable vision for live insects and insect-scale robots," *Science Robotics*, vol. 5, no. 44, 2020. [Online]. Available: https://robotics.sciencemag.org/content/5/44/eabb0839

[16] N. O. Pérez-Arancibia, K. Y. Ma, K. C. Galloway, J. D. Greenberg, and R. J. Wood, "First controlled vertical flight of a biologically inspired microrobot," *Bioinspiration and Biomimetics*, vol. 6, no. 3, p. 036009, Sep 2011. [Online]. Available: http://dx.doi.org/10.1088/1748-3182/6/3/036009

[17] S. B. Fuller, M. Karpelson, A. Censi, K. Y. Ma, and R. J. Wood, "Controlling free flight of a robotic fly using an onboard vision sensor inspired by insect ocelli," *J. Royal Society Interface*, vol. 11, no. 97, August 2014. [Online]. Available: http://rsif.royalsocietypublishing.org/content/11/97/20140281.abstract

[18] D. S. Drew, B. Kilberg, and K. S. Pister, "Future mesh-networked pico air vehicles," in *Unmanned Aircraft Systems (ICUAS), 2017 International Conference on*, 2017, pp. 1075–1082.

[19] S. B. Fuller, Z. Yu, and Y. Talwekar, "Visual flight control and wind gust rejection in 1 mg gnat robots," 2022, under review.

[20] D. Floreano and R. J. Wood, "Science, technology and the future of small autonomous drones," *Nature*, vol. 521, no. 7553, pp. 460–466, 2015.

[21] A. Censi, S. Han, S. B. Fuller, and R. M. Murray, "A bio-plausible design for visual attitude stabilization," in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*. IEEE, 2009, pp. 3513–3520.

[22] S. Han, A. Censi, A. D. Straw, and R. M. Murray, "A bio-plausible design for visual pose stabilization," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. San Francisco, CA: IEEE, 25–30 September 2010, pp. 5679–5686.

[23] A. Censi and R. M. Murray, "Bootstrapping bilinear models of simple vehicles," *The International Journal of Robotics Research*, vol. 34, no. 8, pp. 1087–1113, 2015.

[24] A. S. Huang, A. Bachrach, P. Henry, M. Krainin, D. Maturana, D. Fox, and N. Roy, "Visual odometry and mapping for autonomous flight using an rgb-d camera," in *Robotics Research*. Springer, 2017, pp. 235–252.

[25] S. Ramasamy, R. Sabatini, A. Gardi, and J. Liu, "Lidar obstacle warning and avoidance system for unmanned aerial vehicle sense-and-avoid," *Aerospace Science and Technology*, vol. 55, pp. 344–358, 2016.

[26] A. Bry, A. Bachrach, and N. Roy, "State estimation for aggressive flight in gps-denied environments using onboard sensing," in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 1–8.

[27] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in *European conference on computer vision*. Springer, 2014, pp. 834–849.

[28] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *2007 6th IEEE and ACM international symposium on mixed and augmented reality*. IEEE, 2007, pp. 225–234.

[29] A. Bachrach, R. He, and N. Roy, "Autonomous flight in unknown indoor environments," *International Journal of Micro Air Vehicles*, vol. 1, no. 4, pp. 217–228, 2009.

[30] V. S. Dorbala, A. A. Hafez, and C. Jawahar, "A deep learning approach for robust corridor following," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 3712–3718.

[31] A. Saxena, H. Pandya, G. Kumar, A. Gaud, and K. M. Krishna, "Exploring convolutional networks for end-to-end visual servoing," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 3817–3823.

[32] S. B. Fuller and R. M. Murray, "A hovercraft robot that uses insect-inspired visual autocorrelation for motion control in a corridor," in *Robotics and Biomimetics (ROBIO), 2011 IEEE Int. Conf.*, Phuket, Thailand, 7–11 December 2011, pp. 1474–1481.

[33] A. Censi and R. M. Murray, "Bootstrapping sensorimotor cascades: a group-theoretic perspective," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2011, pp. 2056–2063.

[34] A. Censi, S. Han, S. B. Fuller, and R. M. Murray, "A bio-plausible design for visual attitude stabilization," in *Decision and Control (CDC), 2009 IEEE Int. Conf.* Shanghai, China: IEEE, 16–19 December 2009, pp. 3513–3520.

[35] E. F. Helbling, S. B. Fuller, and R. J. Wood, "Pitch and yaw control of a robotic insect using an onboard magnetometer," in *Robotics and Automation (ICRA), 2014 IEEE Int. Conf.*, 2014, pp. 5516–5522.

[36] J. S. Humbert and A. M. Hyslop, "Bioinspired visuomotor convergence," *IEEE Transactions on Robotics*, vol. 26, pp. 121–130, 2010.

[37] Y. M. Chukewad and S. B. Fuller, "Yaw control of a hovering flapping-wing aerial vehicle with a passive wing hinge," (in press).

[38] D. Dhingra, Y. M. Chukewad, and S. Fuller, "A device for rapid, automated trimming of insect-sized flying robots," *IEEE Robotics and Automation Letters*, 2020.