

A bio-plausible design for visual attitude stabilization

Andrea Censi, Shuo Han, Sawyer B. Fuller, Richard M. Murray

Abstract—We consider the problem of attitude stabilization using exclusively visual sensory input, and we look for a solution which can satisfy the constraints of a “bio-plausible” computation. We obtain a PD controller which is a bilinear form of the goal image, and the current and delayed visual input. Moreover, this controller can be learned using classic neural networks algorithms. The structure of the resulting computation, derived from general principles by imposing a bilinear computation, has striking resemblances with existing models for visual information processing in insects (Reichardt Correlators and lobula plate tangential cells). We validate the algorithms using faithful simulations of the fruit fly visual input.

I. INTRODUCTION

Biology was the original inspiration for the field which is now known as control theory [1], and still surpasses human engineering in the complexity, efficiency, robustness, and inherent evolvability of solutions. Many long-standing problems, such as artificial intelligence and “true” computer vision, seem still out of reach from traditional design paradigms. In the case of human intelligence, the deluded engineer can justify his failure by invoking the fact that the brain is composed of about 10^{11} neurons and 10^{15} synapses, vastly exceeding the theoretical computational power of current artificial systems. But this excuse does not hold for simple animals such as flying insects. Using only 10^5 neurons, they are able to perform feats of control we are not yet able to explain: when a fly is startled, it leaps in uncontrolled fashion, tumbling through the air; yet within a few wing strokes, the fly stabilizes and flies away [2].

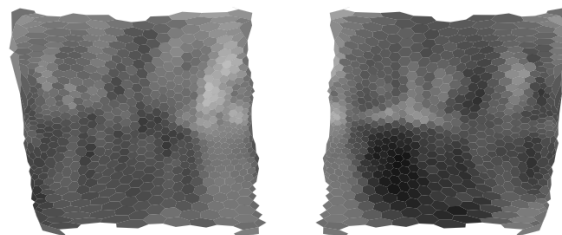
As engineers, we admire these behaviors because they are realized under severe computational constraints. In real time, a fly’s brain must convert an enormous amount of sensor information, most of it being noisy and sparse visual input (Fig. 1), into a limited number of motor commands. This is realized using distributed and noisy computing elements (neurons) which have a bit rate on the order of 1 kbps or lower [3]. This noisy, distributed, slow control system is able to control the very fast flight dynamics: wing flapping occurs at over 200 Hz, and when chasing another fly during mating, a blowfly can make course corrections in as little as 30 ms after its target has changed course [4]. Moreover, while most of the system has been designed through evolution, there is also an element of adaptation (“learning”), as evidenced by the fact that flies can recover from wing damage.

Analyzing the solutions developed under these constraints may provide insight into some of engineering challenges

The authors are with the Division of Engineering and Applied Science at the California Institute of Technology. Address: MC 107-81, 1200 E. California Blvd., 91125, Pasadena, CA (USA). E-mail: {andrea,murray}@cds.caltech.edu, {minster,hanshuo}@caltech.edu



(a) Outdoor naturalistic scene (image source: [6])



(b) Simulated fly’s perception of the scene

Fig. 1. An example of simulated insect vision. The visual scene (a) is perceived by the fruit fly as in (b), according to the current physiological models used in biology. The two eyes sample a field of view that comprises nearly the entire visual sphere.

of the moment. Information-rich robotic systems such as autonomous vehicles gather gigabytes of sensory input per second and must quickly react. Foreseeing the eventual demise of Moore’s law, the predominantly serial nature of current processors must give way to parallel and even intentionally noisy computation [5]; learning might be an answer to the problem of designing systems too complex for a human mind to grasp.

Biology, on the other hand, can benefit from a reverse-engineering of natural systems, because it helps transitioning from a purely qualitative to a quantitative analysis, which allows a better understanding of underlying biological limitations, design principles, and mechanisms for adaptation. Classical successes of quantitative modelling include the case of visual steering [7] and pitching [8] in insects. Currently, the focus is in tying these disparate findings together into a unified framework for flight stabilization and maneuvering, using a control-theoretic language [9].

The engineer who wants to get her feet wet in biology can take a number of approaches [10]. Usually, *bio-mimetic* refers to the direct mimicry of known biologic processes; *bio-inspired* refers to a design loosely based on a general idea from biology (e.g., genetic algorithms); in between these, we use the term *bio-plausible* to refer to an implementation that adheres to known biological constraints, but does not yet have conclusive evidence, nor has been disproved, by experiments. This kind of work has the potential to both inspire novel engineering methods and provide testable, quantitative hypotheses to biologists.

In this work, we propose a *bio-plausible* solution for the

task of visual attitude stabilization, a core requirement for flying insects. We report a control law that stabilizes visual input to a memorized “goal” image. The controller is bio-plausible in that we limit computations to parallel operations, the controller receives as input only luminance readings gathered directly from visual sensors, and computations are learned using a supervisory signal. We are especially interested in modelling the visual processing faithfully to biology; for simplicity, we assume ideal “torque” commands as the controller output, while, in reality, torques are applied by modulating the wing motion (still, biologically plausible wing kinematics have been proposed that could give a hovering fly full actuation according to fluid dynamics [9]). Evidence for bio-plausibility comes from the fact that the resulting visual processing is loosely equivalent to existing models for the visual processing in the fly.

Formally, we consider a second-order system on $SO(3)$:

$$\begin{cases} \dot{\mathbf{r}} &= \mathbf{r} \boldsymbol{\omega}_{\times} \\ \mathbb{I} \dot{\boldsymbol{\omega}} &= (\mathbb{I} \boldsymbol{\omega}) \times \boldsymbol{\omega} + \boldsymbol{\tau} \end{cases} \quad (1)$$

Here $\mathbf{r} \in SO(3)$ is the body attitude with respect to a fixed world frame, $\boldsymbol{\omega} \in \mathbb{R}^3$ is the angular velocity in the body frame, $\boldsymbol{\omega}_{\times}$ is the corresponding skew-symmetric matrix $[0 - \omega_3 \ \omega_2; \omega_3 \ 0 - \omega_1; -\omega_2 \ \omega_1 \ 0]$, \mathbb{I} is the angular inertia matrix, and $\boldsymbol{\tau}$ is the input torque. The only available sensory input for control is the visual input. At each time t , we assume to know $y(s, t)$, which is a function defined on the unit sphere ($s \in \mathbb{S}^2$). Given an environment, or “map”, $m : \mathbb{S}^2 \rightarrow \mathbb{R}$, the visual input y is given by

$$y(s, t) \triangleq m(\mathbf{r}(t)s)$$

We assume to know a “goal” image $g : \mathbb{S}^2 \rightarrow \mathbb{R}$, which is a snapshot of y taken at a certain goal configuration \mathbf{r}^g ; $g(s) = m(\mathbf{r}^g s)$. We do not assume to know either \mathbf{r} , $\boldsymbol{\omega}$, or \mathbf{r}^g ; the only input data are the current image y and the goal image g . We solve the following problem:

Problem 1: (Visual attitude stabilization) Choose the input torque $\boldsymbol{\tau}$ such that $y \rightarrow g$.

The paper is organized as follows. Section II discusses related work in the engineering literature. Section III concerns the estimation of $\boldsymbol{\omega}$ from the visual input. We show that the least-squares estimate of $\boldsymbol{\omega}$ is given by the nonlinear function of y : $\hat{\boldsymbol{\omega}}_{LS} = \langle (Sy)(Sy)^* \rangle^{-1} \langle \dot{y}(Sy) \rangle$, where S is a certain linear differential operator to be introduced, and $\langle \cdot \rangle$ is integration over the sphere. Then, we show that such estimator can be “chopped” to the skew-symmetric bilinear function in y and \dot{y} : $\hat{\boldsymbol{\omega}}_{BL} = c \langle \dot{y}(Sy) \rangle$. This estimator is not as precise as $\hat{\boldsymbol{\omega}}_{LS}$ but is still useful as a damping term in the control law. Moreover, we show how a smoothed approximation to S can be learned from the data, if a reference signal $\boldsymbol{\omega}$ is available in the training stage.

Section IV considers the problem of attitude stabilization. We show that a PD controller for the system can be written as two bilinear forms, in g/y and \dot{y}/y and is arguably the simplest solution for this particular task. Then, we show that the information learned for estimation can be used in the

TABLE I

SYMBOLS USED IN THIS PAPER

In the following, let V be a generic vector space, $C^1(A, B)$ the set of differentiable functions from A to B (with $B = \mathbb{R}$ if omitted), $L^1(A, B)$ be the set of Lebesgue-integrable functions, $\mathfrak{X}(\mathbb{S}^2)$ the set of vector fields on \mathbb{S}^2 .

Symbol	Type	Meaning
\mathbf{r}	$SO(3)$	attitude (unknown)
\mathbf{r}^g	$SO(3)$	goal attitude (unknown)
$\boldsymbol{\tau}$	\mathbb{R}^3	torque (control input)
\mathbb{I}	$\mathbb{R}^{3 \times 3}$	angular inertia matrix
$\boldsymbol{\omega}$	$TSO(3) \simeq \mathbb{R}^3$	angular velocity in body frame
m	$C^1(\mathbb{S}^2, \mathbb{R})$	environment map
$y(\cdot, t)$	$C^1(\mathbb{S}^2, \mathbb{R})$	visual input ($= m \circ \mathbf{r}$)
y	\mathbb{R}^n	sampled y
g	$C^1(\mathbb{S}^2, \mathbb{R})$	goal image ($= m \circ \mathbf{r}^g$)
$\langle f \rangle$	$L^1(\mathbb{S}^2, V) \rightarrow V$	integral of f over \mathbb{S}^2 ; see (5)
S	$C^1(\mathbb{S}^2) \rightarrow \mathfrak{X}(\mathbb{S}^2)$	See Def. 1
$\mathcal{I}[\boldsymbol{\omega}]$	$\mathbb{R}^{3 \times 3}$	FIM for $\boldsymbol{\omega}$; see (6)
P	$\mathbb{S}^2 \times \mathbb{S}^2 \rightarrow \mathbb{R}$	covariance of y
\star	$L^1(\mathbb{S}^2, V) \times C^1([0, 2\pi]) \rightarrow C^1(\mathbb{S}^2, V)$	spherical smoothing
M	$C^1(\mathbb{S}^2) \rightarrow \mathfrak{X}(\mathbb{S}^2)$	learned linear form used in $\hat{\boldsymbol{\omega}}_{LBL}$
\mathbf{M}	$\mathbb{R}^{n \times n \times 3}$	discretized M

controller as well. Section V discusses the “bio-plausibility” of these algorithms and the relation with Elementary Motion Detectors (EMDs) [7], models used in biology to explain the local visual processing in flying insects, as well as with the global processing in the tangential cells [11]. Section VI presents experiments using simulated visual input for a fruit fly.

II. RELATED WORK

This paper considers the problem of visual stabilization from many different angles: estimation, control, machine learning and biology. The single aspects have been explored in isolation in many other works; for reasons of space, we only give representative examples. The first use in engineering of EMD-based computations has been for steering a mobile wheeled robot [12]. In [13], EMDs and optic flow computations were used to trigger a saccade operation for a flying helicopter. In [14], matched filters are learned for both attitude stabilization and velocity regulation, but no proof of stability is given. In previous work in our group [15], [16], a control-theoretic stability proof is given, but the input is pure optic flow, which in general is not easy to estimate, and there is no learning involved. In general, it is easy to use EMD and/or optic flow computation as primitives for realizing a particular behavior, but it is difficult to obtain formal proofs of correctness, especially if learning is involved. On the other hand, these works showed behaviors more complicated than attitude stabilization. “Traditional” control laws for stabilization of insect flights were proposed in [17] using not vision, but the (more easily tractable) information from the ocelli and the halteres.

We now set aside the bio-inspired literature, and briefly consider related work for attitude stabilization as a purely engineering problem. If the state is available for feedback, a PD controller can be easily derived [18]. The problem is that extracting the state from the visual input is not trivial.

Attitude stabilization can be seen as a particular case of *visual servoing* [19], which is usually formulated by using features (or “landmarks”, or “fiducial points”); this is not applicable to the kind of visual input we consider (Fig. 1a). Recently, new methods have been proposed that do not rely on features [20], [21], [22]. The resulting algorithms have many of the desirable characteristics of the gradient flow that we use. However, these works do not concern either learning of the control law, or motion estimation.

III. ESTIMATION OF ANGULAR VELOCITY

In this section we will investigate how to estimate ω , the angular velocity in body frame, from the visual input y alone. The immediate motivation is to construct a damping term for the control law. In this context, we are not interested in a perfect recovery, but in the best that can be achieved under the constraints that we discussed; in particular, by constraining the computation to be bilinear in y and \dot{y} .

A. Least-squares estimation of ω

Estimating ω from the visual input is a particular case of estimating the optic flow. Because under purely rotational motion all the optic flow is coherent, we can derive a closed form formula for ω . We now state some preliminaries.

Lemma 1: The relation among y , \dot{y} and ω is

$$\dot{y}(s, t) = (s \times \nabla_s y(s, t))^* \omega(t). \quad (2)$$

(Proof omitted). The operator $y \mapsto s \times \nabla_s y(s)$ turns out to be very useful, and is worthy of the following abbreviation.

Definition 1: Define $S : C^1(\mathbb{S}^2, \mathbb{R}) \rightarrow \mathfrak{X}(\mathbb{S}^2)$ as

$$(Sy)(s) \triangleq s \times \nabla_s y(s).$$

Equation (2) can now be written as

$$\dot{y} = (Sy)^* \omega. \quad (3)$$

In practice, both y and \dot{y} are corrupted by noise. By elementary signal processing considerations, we can assume that the noise on y is negligible with respect to the noise on \dot{y} . Assuming a statistical model of the kind $\dot{y} = (Sy)^* \omega + \epsilon$, with uniform independent noise, the least-squares estimate of ω is

$$\hat{\omega}_{LS} \triangleq \langle (Sy)(Sy)^* \rangle^{-1} \langle \dot{y}(Sy) \rangle. \quad (4)$$

The notation $\langle f \rangle$ stands for the integral of f over the sphere with respect to the unique rotational-invariant measure dS :

$$\langle f \rangle \triangleq \int_{\mathbb{S}^2} f dS. \quad (5)$$

Equation (4) requires the inversion of what can be identified as the Fisher Information Matrix [23] for ω :

$$\mathcal{I}[\omega] \triangleq \langle (Sy)(Sy)^* \rangle. \quad (6)$$

If such matrix is singular, then there is not enough contrast in all directions to disambiguate ω exactly. In that case, using the pseudo inverse gives the solution with minimum norm.

B. Bilinearly constrained estimation of ω

The least-squares estimator $\hat{\omega}_{LS}$ is a complicated non-linear function of y . In this context, we consider what is possible to do with less computation. It is evident that no useful *linear* (in y and \dot{y}) estimator of ω can be found. The next simplest thing is looking for a *bilinear* form in y and \dot{y} . One approach would be to just chop off the $(\mathcal{I}[\omega])^{-1}$ factor from (4) and substitute it with a simple constant $c > 0$, thus obtaining

$$\hat{\omega}_{BL} \triangleq c \langle \dot{y}(Sy) \rangle. \quad (7)$$

This, indeed, is the best bilinear estimator of ω , in the sense specified by the following proposition.

Proposition 1: The estimator (7) is the unique bilinear estimator such that, for all ω , $\mathbb{E}_{m,r} \{ \hat{\omega}_{BL} \} = \omega$. Moreover, the constant c is the inverse of the average image contrast: $c = (\mathbb{E}_m \{ \|\nabla y\|_2^2 \})^{-1}$.

Proof: Suppose we are looking for a generic bilinear estimator for ω , of the form $\hat{\omega}_{BL} = \langle (My)\dot{y} \rangle$ for some linear operator $M : C^1(\mathbb{S}^2, \mathbb{R}) \rightarrow \mathfrak{X}(\mathbb{S}^2)$. Imposing $\mathbb{E}_{m,r} \{ \langle (My)\dot{y} \rangle \} = \omega$ leads to $\mathbb{E}_{m,r} \{ \langle (My)(Sy)^* \omega \rangle \} = \omega$ and hence $\mathbb{E}_{m,r} \{ \langle (My)(Sy)^* \rangle \} = \text{Id}$. By substitution, the solution to this linear equation in M is $My = \mathbb{E}_{m,r} \{ \langle (Sy)(Sy)^* \rangle^{-1} Sy \}$. The factor $\mathbb{E}_{m,r} \{ \langle (Sy)(Sy)^* \rangle \}$ is the average Fisher Information Matrix over all possible environments and orientations. To evaluate this term, we must make some assumptions about the environments. If we assume that r is uniformly distributed on $\text{SO}(3)$, then by symmetry considerations $\mathbb{E}_{m,r} \{ \langle (Sy)(Sy)^* \rangle \}$ must be a multiple of the identity matrix. In particular, we would have $\mathbb{E}_{m,r} \{ \langle (Sy)(Sy)^* \rangle \} = \mathbb{E}_m \{ \|\nabla y\|_2^2 \} \mathbf{I}$ and therefore the optimal c in (7) is $c = (\mathbb{E}_m \{ \|\nabla y\|_2^2 \})^{-1}$. ■

We now investigate the properties of the estimator $\hat{\omega}_{BL}$. It is apparent that $\hat{\omega}_{BL}$ is *arbitrarily inaccurate* in terms of scale, without more constraints on the environment. Suppose that $\hat{\omega}_{BL} = \omega$ in a certain environment y . If we consider an environment which is twice as bright, by letting $y' = 2y$, then the resulting $\hat{\omega}_{BL}$ will be off of a factor of 4 in the new environment. This is the cost to pay for chopping off the normalizing factor $\langle (Sy)(Sy)^* \rangle^{-1}$ from (4). Fortunately, we can easily see that, while the scale can be arbitrarily wrong, the direction of $\hat{\omega}_{BL}$ cannot be more than 90° off from ω .

Proposition 2: $\omega^* \hat{\omega}_{BL} \geq 0$.

Proof: $\omega^* \hat{\omega}_{BL} = \omega^* \langle \dot{y}(Sy) \rangle = \omega^* \langle (Sy)(Sy)^* \omega \rangle \geq 0$ ■

This property allows to use $\hat{\omega}_{BL}$ in a damping control law.

Lastly, we show that $\hat{\omega}_{BL}$ is actually a *skew-symmetric* bilinear form of y and \dot{y} .

Lemma 2: Let $a, b \in C^1(\mathbb{S}^2)$. Then the bilinear form $(a, b) \mapsto \langle a(Sb) \rangle$ is skew-symmetric: $\langle a(Sb) \rangle = -\langle b(Sa) \rangle$.

Proof: Given $a, b \in C^1(\mathbb{S}^2)$, consider the function $\langle a(b \circ r) \rangle = \langle (a \circ r^{-1})b \rangle$. Compute the gradient with respect to r at $r = \text{Id}$ for both sides to obtain the thesis. ■

C. Learnable, bilinearly constrained estimation of ω

The preceding section showed how to compute analytically the bilinear estimator $\hat{\omega}_{BL}$ to use in the damping term in the

control law. This section shows how an analogous estimator can be learned from the data, if the reference signal ω is available during training. For the purpose of analysis, it is convenient to stick in this section to considering y as a continuous function defined on the sphere. In Section VI, we will give the formulas again for the discrete case and we will discuss the implementation details.

We look for a linear operator $M : C^1(\mathbb{S}^2, \mathbb{R}) \rightarrow \mathfrak{X}(\mathbb{S}^2)$ that minimizes the expected estimation error $\mathbb{E}_{m,r,\omega} \left\{ \|\langle \dot{y}(My) \rangle - \omega\|_2^2 \right\}$. Fortunately, we can consider the three components of ω separately, and write for each of the three “slices” of M the error function

$$\mathbb{E}_{m,r,\omega} \left\{ (\langle \dot{y}(M^k y) \rangle - \omega^k)^2 \right\}, \quad k = 1, 2, 3. \quad (8)$$

Given a particular training sample (ω, \dot{y}, y) , M^k can be updated according to a *stochastic gradient descent* rule, which consists in minimizing (8) only with respect to the particular sample. The update rule is

$$\dot{M}^k = \alpha (\langle \dot{y}(M^k y) \rangle - \omega_i) \dot{y} y^*, \quad k = 1, 2, 3. \quad (9)$$

The factor $\alpha > 0$ is the learning rate. The term $(\langle \dot{y}(M^k y) \rangle - \omega^k) \in \mathbb{R}$ is a global error correction term. The structure of the learning rule is essentially a perceptron, with the flavor of a Hebbian rule in the product $\dot{y} y^*$.

D. Properties of the learning rule

The following is a statistical analysis of the learning rule. More precisely, we consider the estimator

$$\hat{\omega}_{\text{LBL}} \triangleq \langle \dot{y}(\mathbb{E}\{M\}y) \rangle,$$

where $\mathbb{E}\{M\}$ is the expected value of the learned M over sufficiently many random training samples (“LBL” stands for Learned BiLinear). In practice, as detailed in the experiments section, we observe that the learning rule produces skew-symmetric operators (up to noise and oscillation). At the moment, we can prove this mathematically only in the two-dimensional case ($y : \mathbb{S}^1 \rightarrow \mathbb{R}$, $r \in \text{SO}(2)$, $\omega \in \mathbb{R}$).

Proposition 3: In the two-dimensional case, the expected value of the operator M is proportional to SP , where P is the covariance of y : $\mathbb{E}\{M\} \propto \text{SP}$.

Proof: Given the update rule (9), we can write for M :

$$M(T) = \alpha \int_0^T (\langle \dot{y}(t)(M(t)y(t)) \rangle - \omega(t)) \dot{y}(t)y(t)^* dt$$

If we assume that the training samples were exhaustive, we can substitute the integral with the expectation:

$$\mathbb{E}\{M\} \propto \mathbb{E}_{y,\omega} \{ (\langle \dot{y}(M(t)y) \rangle - \omega) \dot{y} y^* \},$$

and by expanding and neglecting the scalar factors, we obtain

$$\begin{aligned} \mathbb{E}\{M\} &\propto \mathbb{E}_{y,\omega} \{ ((\omega \text{S}y)^* M(t)y - \omega) (\omega \text{S}y) y^* \} \\ &= \mathbb{E}_\omega \{ \omega^2 \} \mathbb{E}_y \{ ((\text{S}y)^* M(t)y - 1) \text{S}y y^* \} \\ &\propto \mathbb{E}_y \{ \text{S}y y^* \} = \text{S}(\bar{y}^2 \mathbf{1}^* + P) = \text{SP}. \end{aligned}$$

In the last passage, \bar{y} is the average of y and $\mathbf{1}$ represents the constant function of the sphere, for which $\text{S}\mathbf{1} = \mathbf{0}$. ■

We conjecture that this result holds also in the 3D case, as we find in the experiments (compare Fig. 3), and continue with this assumption. The operator SP is essentially a smoothed version of S . To see this, consider that, if y is assumed to be a uniform isotropic spatial process, then the covariance between $y(s_1, t)$ and $y(s_2, t)$ is a function of the geodesic distance $d(s_1, s_2)$. Thus $P(s_1, s_2) = p(d(s_1, s_2))$ for some function p . When P is used as a linear operator, the result is akin to a smoothing operator. Formally, define the smoothing operation on the sphere in the following way:

Definition 2: Given a function $f : L^1(\mathbb{S}^2, V)$ and a kernel $\varphi \in C^1([0, \pi])$, define the smoothed function $\varphi \star f$ as:

$$(\varphi \star f)(s) = \int_{\mathbb{S}^2} \varphi(d(s, u)) f(u) du.$$

With this notation, if $P(s_1, s_2) = p(d(s_1, s_2))$, $\text{SP}y$ can be written as $\text{S}(p \star y)$, thereby evidencing the smoothing properties of SP . To prove that $\mathbb{E}\{M\}$ can be used for estimation, and to prove the equivalent of Proposition 2 for $\hat{\omega}_{\text{LBL}}$, we must introduce a new hypothesis on the covariance P .

Condition 1: (Factorization property of P) The covariance operator P can be decomposed as $P = Q^2$ where Q has a spatial dependent kernel ($Q(s_1, s_2) = q(d(s_1, s_2))$ for some kernel q).

Examples of covariance kernels that allow this factorization are the Gaussians and exponentials (a Gaussian can be written as the convolution of two Gaussians, and likewise for exponentials) — see [24] for more information about spatial processes and kernels. This is a condition on the statistical properties of the environment. Naturalistic images, which are usually assumed to have $1/f$ power spectra and exponential covariance kernel [25], satisfy this condition.

Proposition 4: If Condition 1 holds, then $\mathbb{E}\{M\}$ is skew-symmetric and $\omega^* \hat{\omega}_{\text{LBL}} \geq 0$.

Proof: In this proof we use the following commutation properties of \star (the proof of which is omitted): 1) $\langle g(\varphi \star f) \rangle = \langle (\varphi \star g)f \rangle$; and 2) $\text{S}(\varphi \star f) = \varphi \star (\text{S}f)$. Suppose that P satisfies Condition 1. To show that $\langle a, b \rangle \rightarrow \langle a(\text{SP}b) \rangle$ is skew-symmetric, note that $\langle a(\text{SP}b) \rangle =$

$$\langle a(\text{S}Q^2b) \rangle = \langle a(\text{S}(q \star q \star b)) \rangle = \langle (q \star a)\text{S}(q \star b) \rangle.$$

From this, skew-symmetry follows from skew-symmetry of S . To show that $\omega^* \hat{\omega}_{\text{LBL}} \geq 0$, note that from the previous equation, $\omega^* \hat{\omega}_{\text{LBL}} = \omega^* \langle \text{S}(q \star y) (\text{S}(q \star y))^* \rangle \omega \geq 0$. ■

Note that the “ideal” bilinear form is skew-symmetric (Proposition 2) and that the expected value of the learned operator $\mathbb{E}\{M\}$ is skew-symmetric as well (Proposition 2). Given these results, it makes sense to limit the search space to skew-symmetric operators, by modifying the learning rule (9) to

$$\dot{M}^k = \alpha (\langle \dot{y}(M^k y) \rangle - \omega^k) (\dot{y} y^* - y \dot{y}^*), \quad k = 1, 2, 3. \quad (10)$$

IV. VISUAL ATTITUDE STABILIZATION

We now consider the problem of attitude stabilization. Without loss of generality, choose the world reference frame such that the goal attitude r^g is the identity matrix. One

possible candidate for an error measure to be minimized is $J(\mathbf{r}) = \frac{1}{2} \|g - y\|_2^2$. It is easy to see that the gradient of J can be expressed through the operator S .

Proposition 5: The gradient flow that minimizes $\frac{1}{2} \|g - y\|_2^2$ is $\boldsymbol{\omega} = \langle g(Sy) \rangle$.

Proof: Recalling that $y = g \circ \mathbf{r}$, and the fact that rotations are unitary operators, we can write that

$$J(\mathbf{r}) = \frac{1}{2} \|g - g \circ \mathbf{r}\|^2 = -\langle gy \rangle + \text{const.}$$

Given (3), the time derivative of J is

$$\dot{J}(\mathbf{r}) = -\langle g\dot{y} \rangle = -\langle g(Sy)^* \boldsymbol{\omega} \rangle = -\langle g(Sy) \rangle^* \boldsymbol{\omega}. \quad (11)$$

and thus $\boldsymbol{\omega} = \langle g(Sy) \rangle$ minimizes $J(\mathbf{r})$. ■

From this, it follows that $\boldsymbol{\omega} = \langle g(Sy) \rangle$ would stabilize y to g for a first-order system. For a system with inertia, we must include a damping term. Using the results developed in Section III, we know that $\langle \dot{y}(Sy) \rangle$ can be used as a damping term.

Proposition 6: Let $k_p, k_d > 0$. If $\langle (Sy)(Sy)^* \rangle|_{\mathbf{r}=\mathbf{r}^g} > 0$, the control law

$$\tau = k_p \langle g(Sy) \rangle - k_d \langle \dot{y}(Sy) \rangle$$

makes $\mathbf{r} = \mathbf{r}^g$, $\boldsymbol{\omega} = \mathbf{0}$ locally asymptotically stable.

Proof: Consider the candidate Lyapunov function

$$\begin{aligned} V(\mathbf{r}, \boldsymbol{\omega}) &= k_p \frac{1}{2} \|y - g\|_2^2 + \frac{1}{2} \boldsymbol{\omega}^* \mathbb{I} \boldsymbol{\omega} \\ &= -k_p \langle gy \rangle + \frac{1}{2} \boldsymbol{\omega}^* \mathbb{I} \boldsymbol{\omega} + \text{const.} \end{aligned}$$

Note that $V(\mathbf{r}, \boldsymbol{\omega}) \geq 0$ and $V(\mathbf{r}^g, \mathbf{0}) = 0$. Moreover, if $\mathcal{I}[\boldsymbol{\omega}] = \langle (Sy)(Sy)^* \rangle > 0$, then $V(\mathbf{r}, \boldsymbol{\omega})$ is *locally positive definite*: $(\mathbf{r}^g, \mathbf{0})$ is an isolated minimum. This can be checked by showing that $\mathcal{I}[\boldsymbol{\omega}]$ is the Riemannian Hessian of $\|y - g\|_2^2$ at $\mathbf{r} = \mathbf{r}^g$ [26]. We can prove that $\dot{V} \leq 0$ as follows:

$$\begin{aligned} \dot{V}(\mathbf{r}, \boldsymbol{\omega}) &= -k_p \langle g\dot{y} \rangle + \boldsymbol{\omega}^* \mathbb{I} \dot{\boldsymbol{\omega}} \\ &= -k_p \langle g(Sy) \rangle^* \boldsymbol{\omega} + \boldsymbol{\omega}^* ((\mathbb{I}\boldsymbol{\omega}) \times \boldsymbol{\omega} + \tau) \\ &= -k_p \langle g(Sy) \rangle^* \boldsymbol{\omega} + \boldsymbol{\omega}^* (k_p \langle g(Sy) \rangle - k_d \langle \dot{y}(Sy) \rangle) \\ &= -k_d \boldsymbol{\omega}^* (\langle \dot{y}(Sy) \rangle) = -k_d \boldsymbol{\omega}^* \langle (Sy)(Sy)^* \rangle \boldsymbol{\omega} \leq 0 \end{aligned}$$

By Lyapunov's theorem, this implies that the equilibrium is locally stable. By LaSalle's invariance principle, the orbits of the system will eventually stay in the largest invariant set such that $\dot{V} = 0$. Because $\mathcal{I}[\boldsymbol{\omega}]|_{\mathbf{r}=\mathbf{r}^g} > 0$, then $\mathcal{I}[\boldsymbol{\omega}] > 0$ also in a neighbourhood of \mathbf{r}^g . Therefore, $\dot{V} \rightarrow 0$ implies that $\boldsymbol{\omega} \rightarrow \mathbf{0}$. Thus the system approaches an equilibrium point $(\mathbf{r}, \mathbf{0})$. From $\boldsymbol{\omega} \rightarrow \mathbf{0}$ it follows that $\dot{\boldsymbol{\omega}} \rightarrow \mathbf{0}$ and thus $\tau \rightarrow 0$. This implies the gradient of $\langle yg \rangle$ vanishes. Because $\mathcal{I}[\boldsymbol{\omega}] > 0$ is the Hessian, in a neighbourhood of \mathbf{r}^g the gradient vanishes only at $\mathbf{r} = \mathbf{r}^g$, which is then an asymptotically stable equilibrium. ■

The control law used in Proposition 6 has some drawbacks. The technical disadvantage is that S is a differential operator, hence it can be used only on smooth functions y , and it must be approximated if the function y is sampled at a finite number of points. The other problem is that the domain of convergence might be quite small. Consider an environment which is composed of by a dark background and a few very small, very bright spots. In that case, the gradient of

$\|y - g\|$ is zero almost anywhere, and so it cannot be used in the control law. These two problems can be mitigated by smoothing g and y . Instead of $\|g - y\|_2^2$, we consider the error function $\|(\varphi \star g) - (\varphi \star y)\|_2^2$ for some smoothing kernel φ . This is equivalent to a smoothed environment $m' = \varphi \star m$. Therefore, the proof of Proposition 6 could be adapted for a control law such as

$$\tau = k_p \langle (\varphi \star g) S(\varphi \star y) \rangle - k_d \langle (\varphi \star \dot{y}) S(\varphi \star y) \rangle. \quad (12)$$

By Proposition 3, the learned M is a smoothed version of S , and therefore using M in place of S is equivalent to using a control law in the form of (12). The following proposition establishes that we can use M not only as part of a *learnable estimator* of $\boldsymbol{\omega}$, but also as part of a *learnable controller* of \mathbf{r} that solves the problem of attitude stabilization.

Proposition 7: If Condition 1 holds, the control law

$$\tau = k_p \langle g(\mathbb{E}\{M\}y) \rangle - k_d \langle \dot{y}(\mathbb{E}\{M\}y) \rangle \quad (13)$$

makes $\mathbf{r} = \mathbf{r}^g$, $\boldsymbol{\omega} = \mathbf{0}$ locally asymptotically stable if $\langle (S(q \star y) S(q \star y)^*) \rangle|_{\mathbf{r}=\mathbf{r}^g} > 0$.

Proof: Omitted; adapt the proof of Proposition 6 using the considerations about $\mathbb{E}\{M\}$ in the proof of Proposition 4. ■

V. BIO-PLAUSIBILITY

This section investigates the bio-plausibility of the proposed control law. We briefly review basic information about the fly. Then we discuss the traditional models used for characterizing its visual processing, with particular regard to the estimation of angular velocity. Finally we compare our proposed algorithm to those models.

The fruit fly: The fruit fly (*Drosophila melanogaster*) is a small (2.5 mm) dipteran used as a model organism in many branches of biology. Its notable agility comes from the combination of large eyes, relatively large brains for their size, and halteres, which are a pair of vestigial wings that act as vibratory gyroscopes [27]. The fly has two compound eyes, which comprise 1398 small lenses (*ommatidia*) that sample the image across most of the visual sphere (Fig. 1b) [9]. Each is able to detect luminance intensity with a certain angular span $\Delta\phi$ (typically 4.5–6 deg) by focusing light via a tiny lens onto underlying photodetectors. Because of optical diffraction, the luminance sensitivity profile can be modeled as a Gaussian with $\sigma^2 \simeq (1.1\Delta\phi)^2$. The temporal response can be modeled as a low-pass filter with a time constant of 20 ms [28]. After local processing, the visual information is passed to a special class of neurons known as the *lobula plate tangential cells* (LPTCs), of which there are only a few, numbering in the tens. These cells integrate optic flow patterns across large areas of the visual field and are thought to encode information pertaining to self-motion in the fly such as rotation or translation [11]. They synapse onto downstream neurons that lead to motor commands.

Elementary Motion Detectors: The fly is thought to extract information about its state of motion in large part by observing patterns of optic flow across the retina. Behavioral

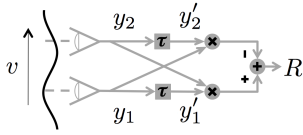


Fig. 2. Diagram of an EMD, the fly’s “elementary motion detector”. A pair of visual sensors, such as the ommatidia of the fly, observe a moving luminance signal (shown here as a sinusoid). Opposing delay (τ) and correlate (\times) operations compare nearby visual sensors to give a response corresponding to the direction of visual motion.

experiments on turning response in tethered insects inside rotating drums have suggested a correlator-type model for visual motion detection (Fig. 2) [7]. In its simplest form, the correlator is known as an “elementary motion detector” (EMD) and consists of opposed delay-and-multiply operations operating on the output of a pair of nearby visual sensors.

While EMDs were thought to act as estimators for the rate of visual motion across the eye, their performance is quite poor in comparison to computational approaches such as the gradient method [29]. Formally, the optic flow $\mathbf{v}(s) \in \mathfrak{X}(\mathbb{S}^2)$ is the apparent motion of objects on the retina due to the egomotion of the sensor. In the case of purely rotational motion, the optic flow at the retinal point s satisfies $\mathbf{v}(s) = \boldsymbol{\omega} \times s$. To report the classical analysis, we restrict ourselves to the 2-dimensional case ($\mathbf{r} \in \mathbb{S}^1$, $\boldsymbol{\omega} \in \mathbb{R}$, $\mathbf{v}(s) \in \mathbb{R}$), as common in neurobiology. In that case $\mathbf{v}(s) = \boldsymbol{\omega}$, which simplifies the analysis (or complicates it, according to the point of view).

Consider the response of an EMD like that in Fig. 2 to a moving sinusoid. Thus the first sensor senses $y_1 = A \sin(2\pi f v t)$ and the second $y_2 = A \sin(2\pi f(v t + \Delta\phi))$, where f is the spatial frequency (cycles/rad) of the sinusoid projected onto the eye and $\Delta\phi$ is the angular displacement between the two sensors. If the delay block is a first-order low-pass filter with time constant τ with output y' satisfying $\tau y' + y = y$, with $\tau \approx 35$ ms in flies, it can be shown that the steady-state response is [7]

$$R = \frac{1}{2\pi\tau} A^2 \frac{2\pi f v}{(2\pi f v)^2 + 1/(2\pi\tau)^2} \sin(2\pi f \Delta\phi). \quad (14)$$

Because $f/\tau \ll 1$, the response is nearly linear in v (and thus $\boldsymbol{\omega}$) for low values, though it incurs aliasing as v is increased. Equation (14) shows the two main limitations of EMDs as optic flow/angular velocity estimators: i) the response is dependent on the spatial frequency f to the same degree as v ; and ii) the response is contrast-dependent because it is proportional to A^2 . Because the system is nonlinear, the superposition effect does not hold, and this analysis does not hold for arbitrary signals other than sinusoids. It has been shown that if the input to the EMD is a moving naturalistic image, the noise in the response may be orders of magnitude larger than the averaged signal \bar{R} [25]. The contrast sensitivity problem — intrinsic in the bilinear computation — can be mitigated by employing saturating nonlinearities [25].

Nevertheless, numerous behavioral and electrophysiological studies have demonstrated insect behavior consistent with the use of EMDs, along with all of their shortcomings, for visual motion detection and control [30]. A striking example is the behavior exhibited when a flying insect is placed at the center of a drum that is rotated about a vertical axis. When different spatial frequencies are presented on the drum, the torque applied by the insect to reduce visual slip is not dependent on the drum’s rotation rate, but rather on the temporal frequency experienced by the fly, as predicted by the EMD model.

Bio-plausibility: We now examine the bio-plausibility of our algorithm. The control law makes no questionable assumptions about biology and, in principle, could be implemented using neurons. The computations constitute a direct sensory-motor cascade that is one-shot parallel, without internal state, assuring a rapid computation by neurons. We use only luminance readings from visual sensors, rather than idealized measures such as optic flow [16].

Moreover, the computations bear striking resemblance the local computations of EMDs plus a global computation in the LPTCs, as follows. In comparing EMDs and our bilinear estimator $\hat{\omega}_{\text{LBL}}$, the first formal difference is that $\hat{\omega}_{\text{LBL}}$ uses the pure derivative \dot{y} and the EMD uses the low-pass filtered version $y'(t)$. The difference is only superficial. For temporal frequencies ($2\pi f v$) below the cut-off frequency $1/\tau$ of the filter, one can use the approximation $\dot{y}(t) \simeq \frac{1}{\tau}(y(t) - y'(t))$. Substituting this into our bilinear estimator, we obtain that $\langle \dot{y}(S y) \rangle \simeq \frac{1}{\tau} \langle (y - y')(S y) \rangle = -\frac{1}{\tau} \langle y'(S y) \rangle$, where we used the fact $\langle y(S y) \rangle = 0$ (Lemma 2). Thus the pure derivative \dot{y} and the filtered y' can be used interchangeably in the estimator.

Switching to the discretized form of S, note that the response $R = y_2 y'_1 - y_1 y'_2$ of an EMD can be written as $R = y'^T \mathbf{M} y$, with $\mathbf{M} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$ being a skew-symmetric matrix. Because $\hat{\omega}_{\text{LBL}}$ is also a *skew-symmetric* bi-linear form of y and \dot{y} (or y'), we can write $\hat{\omega}_{\text{LBL}}$ as a linear combination of a series of EMDs. Looking back at the biology, we can map this further computation to global LPTCs. The factorization of the computation in a local *nonlinear* computation and a global *linear* weighting can be considered energetically efficient as it reduces the neural cabling.

Thus our algorithm is compatible with the current models of visual information processing in the fly. Moreover, the theory developed in the previous sections offers some additional insights. i) Our analysis concerns the full 3D rotation case, which has not been considered in the biology literature. In this context, Propositions 2 and 4 could be interpreted as the “performance guarantee” of an EMD computation in the 3D case. ii) While the EMD computation is thought to be a fixed neural mechanism, the LPTCs neural weight must come from adaptation. We have found a means to derive the optimal weighting for EMDs and LPTCs (tangential cells) without any pre-existing condition such as foreknowledge of the orientations of the various EMD pairs. Learning requires only a supervisory signal $\boldsymbol{\omega}$ which could come from the halteres. iii) Finally, we described a way in which the weights

that are learned for *estimation* can be used for *control* as well (see the use of $\mathbb{E}\{\mathbf{M}\}$ in the control law (13)).

For reasons of space, we did not comment in detail about the fly’s visual attitude *control*, because such analysis should also concern ocelli and halteres, which, like vision, measure rotation rates. Initial analysis reveals that a control law such as $\tau = \langle g(\mathbf{M}y) \rangle$ is compatible with the yaw torque profiles obtained in the experiments with tethered flies that tend to orient themselves toward vertical stripes [31].

VI. SIMULATIONS

We used fsee [32] to simulate the visual input of the the fruit fly *Drosophila*. Such software has already been used to model fly flight control and implements realistic spatial blurring and temporal filtering according to current knowledge in visual physiology [9]. The software, given a 3D virtual world description, produces monochromatic luminance readings ($y(s, t) \in [0, 1]$). We choose as our environment model a cylindrical arena with panoramas of naturalistic scenes on its wall, in addition to a background showing the sky and clouds. This reproduces the statistical properties of the actual input experienced by a fly. Fig. 1b shows a typical rendering of the visual stimulus. In this paper, we are mainly interested in the processing of visual input, therefore we use a simple unconstrained second-order system on $\text{SO}(3)$ to represent the fly dynamics. More accurate models of the flight dynamics have been studied in [9].

Learning algorithm: We now rewrite using finite-dimensional notation the learning rule explained in Section III-C in the case of an infinite-dimensional operator (for which analysis was simpler). Let $\underline{y} \in \mathbb{R}^n$, $\dot{\underline{y}} \in \mathbb{R}^n$, be the visual input sampled at the direction of the n eyes: $\underline{y}_i \triangleq y(s_i, t)$ for some direction s_i . Because the learning rule is adaptive, we do not need to know the actual directions s_i . The operator \mathbf{M} becomes a finite-dimensional tensor $\mathbf{M} \in \mathbb{R}^{n \times n \times 3}$. We consider each “slice” $\mathbf{M}^k \in \mathbb{R}^{n \times n}$, for $k = 1, 2, 3$, separately. With this notation, our learned bilinear estimator is $\hat{\omega}_{\text{LBL}}^k \triangleq \dot{\underline{y}}^T \mathbf{M}^k \underline{y}$ for $k = 1, 2, 3$. Given a tuple $(\omega, \dot{\underline{y}}, \underline{y})$, the learning rule (10) is discretized as $\mathbf{M}^k \leftarrow \mathbf{M}^k + \alpha(\omega^k - \dot{\underline{y}}^T \mathbf{M}^k \underline{y})(\dot{\underline{y}} \underline{y}^T - \underline{y} \dot{\underline{y}}^T)$, where α is the learning rate. If α is properly chosen and tuned accordingly during the learning stage, \mathbf{M}^k will converge to a local minimum of the quadratic cost function $\sum_n (\omega^k - \dot{\underline{y}}^T \mathbf{M}^k \underline{y})^2$ [33]. To obtain proper convergence, it is important that the learning tuples be representative of all the possible inputs.

Learning simulations: A characteristic feature of the fly’s visual system is that the low-level computation, as discussed in Section V, is *local*, concerning only first- and second-neighbors. Hence we constrain $\mathbf{M}_{ij} = \mathbf{0}$ if $d(s_i, s_j) > 30^\circ$. This makes the \mathbf{M} matrices very sparse, with about 73,000 non-zero elements out of about 2 million. Because through the algorithm we enforce skew-symmetry, the effective number of entries is half of that.

Another way to motivate the sparsity is that, by Proposition 3, \mathbf{M} is supposed to approximate SP , which has a sparse pattern. The covariance operator \mathbf{P} is sparse because

distant parts of the environment are uncorrelated, as shown in Fig. 3b; \mathbf{S} , a local differentiation operator, preserves the sparseness; thus SP is sparse. Incidentally, pruning the matrix also helps with the numerical convergence, because otherwise estimating all the 2 million entries would be prone to overfitting.

In the 2D case, the tensor \mathbf{M} can be easily visualized (Fig. 3a) and it clearly appears to be the derivative of a smoothing operator. In the 3D case, the 3 matrices \mathbf{M}^k cannot be easily visualized because they are large and sparse. Therefore, we consider only the important statistics by plotting $\|\mathbf{M}_{ij}\|$ against $d(s_i, s_j)$ in Fig. 3d, for different values of simulated spatial blurring. As predicted by Proposition 3, the learned \mathbf{M} is shaped by the derivative of the covariance kernels (Fig. 3b and 3c).

Attitude stabilization experiments : We conducted simulations to evaluate the domain of convergence for the control law in Proposition 7 (we only proved local results). In the simulations, we set the inertia matrix \mathbb{I} equal to the identity matrix. Fig. 4b shows an example of convergent behavior. The system starts with an initial error of about 30° . Fig. 4c shows the corresponding input torque. While the torque does not evolve monotonically due to inaccurate estimate of the velocity, it eventually goes to zero as the system converges, which is guaranteed by Proposition 4. Fig. 4d plots the percentage of convergence, averaged over 350 random initial conditions, against the initial displacement. Runs that did not converge typically settled on a local minimum. The percentage of convergence drops significantly after 30° ; this is speculated to arise from the locality of \mathbf{M} , which only allows ommatidia connections within 30° .

VII. CONCLUSIONS

In this paper, we considered the problem of purely visual attitude stabilization. We applied a rigorous control-theoretic geometric analysis to a bio-plausible learnable algorithm. The algorithm derived from general principles, by constraining the computation to be skew-symmetric, has striking similarities to Elementary Motion Detectors (EMDs) and lobula plate tangential cells (LPTCs) in the fly. These results encourage us to think that such an approach can be successful on two fronts: I) rigorous design of control laws to be implemented on novel computational substrates; and II) reverse engineering of complex animal behavior.

Acknowledgements. This work was supported in part by the US Army Institute for Collaborative Biotechnology (ICB) and the Boeing Corporation. The authors wish to thank Andrew Straw for his assistance and encouragement.

REFERENCES

- [1] N. Wiener, *Cybernetics, or control and communication in the animal and the machine*. The MIT Press, 1948.
- [2] G. Card and M. H. Dickinson, “Performance trade-offs in the flight initiation of *drosophila*,” *The Journal of Experimental Biology*, vol. 211, pp. 341–353, 2008.
- [3] S. B. Laughlin, *The Computing Neurons*, ch. The Reliability of single neurons and circuit design – a case study, pp. 322–336. New York: Addison-Wesley, 1989.
- [4] M. F. Land and T. S. Collett, “Chasing behavior of houseflies,” *Journal of Comparative Physiology A*, vol. 89, pp. 331–357, 1974.

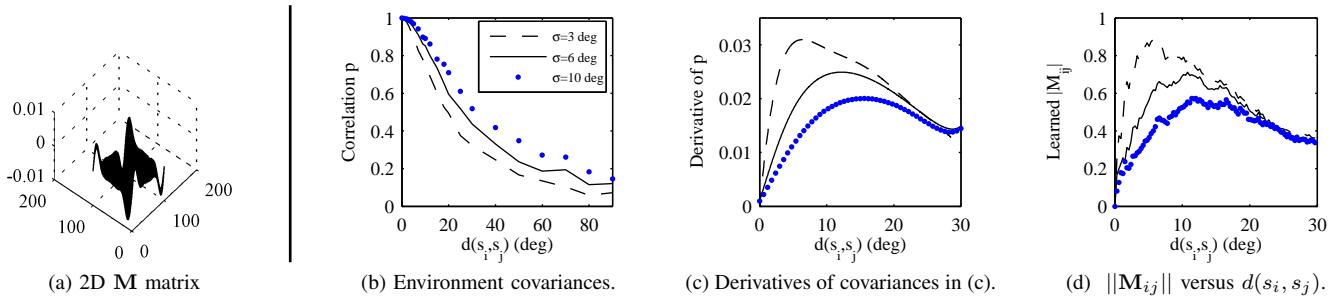


Fig. 3. Results of the learning algorithm. (a) In 2D, it is possible to display the tensor \mathbf{M} as an $n \times n$ matrix, where n is the number of ommatidia. One can see the resulting skew-symmetric matrix has the shape of a derivative of a Gaussian, as predicted by Prop. 3. (b), (c), (d): In 3D, the $n \times n \times 3$ tensor \mathbf{M} cannot be easily displayed. In (b), we first compute the environment covariance kernel, for the same naturalistic image, simulating different values of the spatial blurring. We plot the average correlation of $y(s_i)$ and $y(s_j)$ as a function of $d(s_i, s_j)$. (c) shows the derivatives, computed analytically from an 8th-order polynomial fit of (b). In (d), one can see that $\|\mathbf{M}_{ij}\|$ has a very good match with the plots in (c), thereby validating Prop. 3.

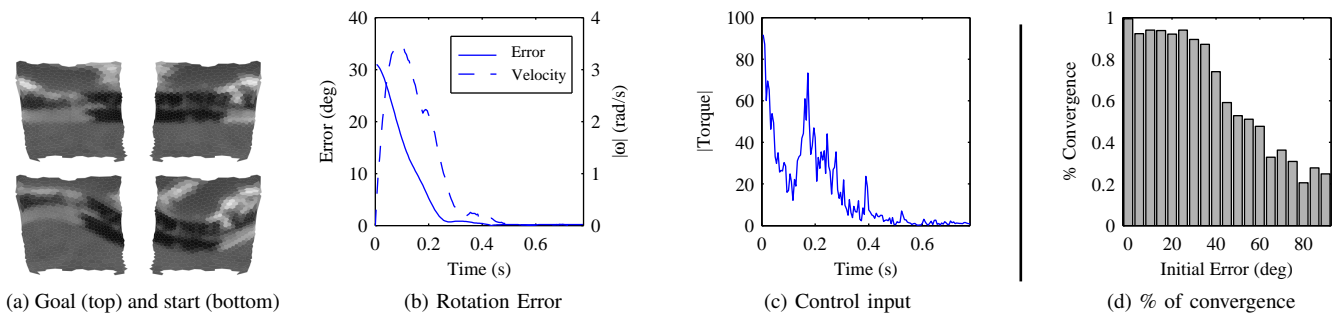


Fig. 4. Results of attitude stabilization. (a), (b), (c) show the results of one particular stabilization experiment. In (a), goal image and initial image are depicted. (b) shows the current error, using the metric on $SO(3)$ defined as $d(\mathbf{r}_1, \mathbf{r}_2) = \arccos((\text{trace}(\mathbf{r}_1^{-1} \mathbf{r}_2) - 1)/2) \in [0, \pi]$, along with $\|\omega\|$. (c) shows the input torque. In (d), the graph displays the percentage of convergence as a function of the initial error. The percentage of convergence drops significantly after 30° , which is the maximum distance we allow for ommatidia connections.

[5] L. N. Chakrapani, P. Korkmaz, B. E. S. Akgul, and K. V. Palem, "Probabilistic system-on-a-chip architectures," *ACM Transactions on Design Automation of Electronic Systems*, vol. 12, p. 28, August 2007.

[6] A. D. Straw, T. Rainsford, and D. C. O'Carroll, "Contrast sensitivity of insect motion detectors to natural images," *J. of Vision*, vol. 8, 2008.

[7] W. Reichardt, *Principles of Sensory Communication*, ch. Autocorrelation, a principle for the evaluation of sensory information by the central nervous system., pp. 303–317. John Wiley & Sons, New York, 1961.

[8] C. T. David, "Visual control of the partition of flight force between lift and thrust in free-flying drosophila," *Nature*, vol. 313, 1985.

[9] W. B. Dickson, A. D. Straw, and M. H. Dickinson, "Integrative model of *Drosophila* flight," *AIAA Journal*, vol. 46, pp. 2150–2164, 2008.

[10] C. M. Dario Floreano, *Bio-Inspired Artificial Intelligence: Theories, Methods, and Technologies*. The MIT Press, 2008.

[11] H. G. Krapp and R. Hengstenberg, "Estimation of self-motion by optic flow processing in single visual interneurons," *Nature*, vol. 384, pp. 463–466, 1996.

[12] N. Franceschini, J. M. Pichon, C. Blanes, and J. M. Brady, "From insect vision to robot vision," *Philosophical Transactions: Biological Sciences*, vol. 337, pp. 283–294, 1991.

[13] J. Zufferey and D. Floreano, "Fly-inspired visual steering of an ultralight indoor aircraft," *IEEE Trans. on Robotics*, vol. 22, 2006.

[14] T. R. Neumann and H. H. Bulthoff, "Behavior-oriented vision for biomimetic flight control," in *Proc. of EPSRC/BBSRC Int. Workshop on Biologically Inspired Robotics*, pp. 196–203, 2002.

[15] M. Reiser, J. Humbert, M. Dunlop, D. Del Vecchio, R. Murray, and M. Dickinson, "Vision as a compensatory mechanism for disturbance rejection in upwind flight," in *American Control Conference*, 2004.

[16] J. Humbert, R. Murray, and M. Dickinson, "A control-oriented analysis of bio-inspired visuomotor convergence," in *Decision and Control, 44th IEEE Conference on*, pp. 245–250, 12–15 Dec. 2005.

[17] E. G. Domenico Campolo, Luca Schenato, "Multimodal sensor fusion for attitude control of micromechanical flying insects: a geometric approach," *IEEE Trans. on Robotics*, 2007.

[18] F. Bullo and R. M. Murray, "Proportional derivative (PD) control on the Euclidean group," Tech. Report Caltech/CDS 1995–010. <http://resolver.caltech.edu/CaltechCDS:1995.CIT-CDS-95-010>.

[19] F. Chaumette and S. Hutchinson, *Handbook of robotics*, ch. Visual Servoing and Visual Tracking, pp. 563–583. 2008.

[20] G. S. Chirikjian, P. T. Kim, J.-Y. Koo, and C. H. Lee, "Rotational matching problems," *International Journal of Computational Intelligence and Applications*, vol. 4, pp. 401–416, 2004.

[21] F. Chaumette, "Image moments: a general and useful set of features for visual servoing," *IEEE Trans. on Robotics*, 2004.

[22] V. Kallem, M. Dewan, J. Swensen, G. Hager, and N. Cowan, "Kernel-based visual servoing," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1975–1980, 2007.

[23] H. L. V. Trees and K. L. Bell, *Bayesian Bounds for Parameter Estimation and Nonlinear Filtering/Tracking*. Wiley-IEEE Press, 2007.

[24] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[25] R. O. Dror, D. C. O'Carroll, and S. B. Laughlin, "Accuracy of velocity estimation by Reichardt Correlators," *Journal of the Optical Society of America A*, vol. 18, pp. 241–252, February 2001.

[26] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization algorithms on matrix manifolds*. Princeton University Press, 2008.

[27] M. H. Dickinson, "Haltere-mediated equilibrium reflexes of the fruit fly, *drosophila melanogaster*," *Philosophical Transactions of the Royal Society of London B*, vol. 354, pp. 903–916, 1999.

[28] M. Juusola and R. C. Hardie, "Light adaptation in drosophila photoreceptors: I. response dynamics and signaling efficiency at 25°C ," *Journal of General Physiology*, vol. 117, no. 1, pp. 3–25, 2000.

[29] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, pp. 185–203, 1981.

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

[31] W. Reichardt and T. Poggio, "Visual control of orientation behavior in the fly," *Quarterly Reviews in Biophysics*, vol. 9, no. 3, 1976.

[32] http://www.dickinson.caltech.edu/Research/Grand_Unified_Fly.

[33] S. Haykin, *Neural Networks: A Comprehensive Foundation*. Prentice-Hall, 1999.