2008 IEEE/RSJ International Conference on Intelligent Robots and Systems
Acropolis Convention Center
Nice, France, Sept, 22-26, 2008

The Cyborg Fly: A biorobotic platform to investigate dynamic coupling effects between a fruit fly and a robot

Chauncey F. Graetzel, Vasco Medici, Nicola Rohrseitz, Bradley J. Nelson and Steven N. Fry

Abstract— **The robust and efficient flight control of insects provide a powerful model system for autonomous microrobots. Conversely, robots offer a robust experimental platform on which to test biological hypotheses. This interaction of biology and robotics is an exciting but challenging task, because the vast disparities between both can lead to inaccurate or even misleading conclusions. In this paper, we present a biorobotic platform that can arbitrarily define the dynamic couplings between a fruit fly and a robot. The platform is used to explore the stability and emergent properties of the biorobotic couple. The fruit fly's wing kinematics are measured in real time and used to drive an autonomous robot. In turn, the robot's sensory information is transformed back into visual feedback to the fly. Using different case studies, we explore how the choice of feedback influences the success of the biorobotic device. We discuss the meaning of such feedback in view of biomimetic implementations.**

I. INTRODUCTION

Flies, like many other flying insects, achieve exceedingly robust flight control despite their inherent instability and limited neural resources. Studies of biological flight control aim at gaining a deeper understanding of the biological basis of locomotor behavior, which can serve as the basis for biomimetic design principles for robots [1]–[3].

While robotics can benefit from biological research in this way, robotic platforms can conversely enhance biological understanding by serving as testbeds to explore and validate hypothesis of sensorimotor pathways [4], [5]. The freedom in the design of the robotic device allows to create arbitrary experimental situations that can be used in a repeatable way to explore ”cases” that would be unpractical or impossible to test on the organism itself. Such an approach may also lead to the identification of emergent behaviors that would be difficult to identify from the observation of the organism's behavior alone [6].

Though such biorobotic implementations are intriguing, the transfer between a target biological system and its robotic counterpart is non-trivial and prone to misconceptions [7], [8]. The transfer starts by modeling the biological process. The choice of the model's complexity level is crucial, because a too general model will lose meaningfulness and a too complex model may be impossible to realize in an artificial system.

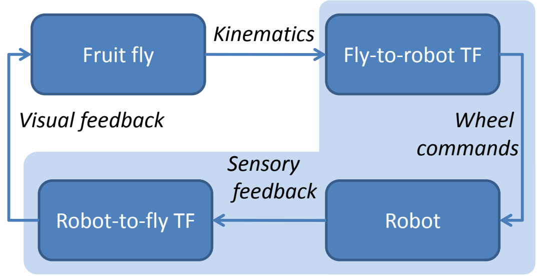


Fig. 1. Fly-in-the-loop: the fruit fly's normal sensory feedback loop is replaced with a non-invasive biorobotic coupling (grey box).

The model is then implemented in a robotic system. Because biological systems have different building blocks and operate at different spatiotemporal scales than engineered devices, this important step always involves a certain *level of abstraction*. One of the most important and difficult effort is to properly take into account the differences in physical implementation. For instance, the underlying functional principle of an insect's compound eye might be successfully reproduced in an artificial system using very different building blocks (e.g. a CCD pixel instead of an ommatidium), as long as the extracted optic flow is the same.

Another big difference is the spatiotemporal scaling: biological organisms are often orders of magnitude smaller and faster than their artificial counterparts. Even if we were able to recreate an exact but scaled copy of a biological process, the optimality of the biological system might still be lost through the scaling.

Finally, once the implementation has taken place, we verify that the artificial system does indeed reproduce the target biological behavior. The comparison can be made using different criteria, i.e. this step involves another level of abstraction. A too lenient criteria might fail to notice strong drawbacks of the implemented system.

In conclusion, the process of coupling biological and robotic systems contains several pitfalls. There is the risk that the biological model is taken out of context and becomes therefore meaningless. The success of a biologically inspired robot to reproduce the behavior of the target biological system is not sufficient to make conclusions about the validity of the couplings that have been used in between. These misconceptions may lead to false 'optimality' claims for biomimetic robots or even to misleading scientific claims in the case of robotic-augmented biology.

To investigate this biology/robotic coupling without mak-

This work is supported by the Swiss National Science Foundation grant #205320-116353

Vasco Medici, Nicola Rohrseitz and Steven Fry are with the Department of Physics, Swiss Federal Institute of Technology of Zürich, 8092 Zürich, Switzerland {vasco,nicola,steven}@ini.uzh.ch

Chauncey Graetzel and Bradley Nelson are with the Department of Mechanical Engineering,Swiss Federal Institute of Technology of Zürich, 8092 Zürich, Switzerland {cgraetzel,bnelson}@ethz.ch

978-1-4244-2058-2/08/$25.00 ©2008 IEEE.

ing assumptions, one approach is to actually take the biological control system (the insect), and let the organism control the artificial system itself. The organism becomes part of the control loop, in a special kind of non-invasive "cyborg" system (See Fig.1). A recent example of such a system was presented by Hertz *et al*, where a cockroach drove a mobile robot by walking on a modified computer mouse ball [9].

In this paper, we present a non-invasive 'Cyborg' system as a biorobotic platform to explore the emergent behaviors resulting from the coupling of a tethered fly and a wheeled robot (see Fig.1).

The relevance of this work can be separated into three aspects that address the biorobotic issues stated above. First, the platform allows a direct interaction with the fly's sensorimotor pathway, giving a model-free paradigm to gain a functional understanding of the processes at play. Second, through the use of flexible transfer functions, the platform allows a vast exploration of the spatial and temporal couplings in between the robotic and biological system. Finally, the platform represents a clear artificially-closed-loop paradigm, with a visual, interpretable, output state - the robot's behavior - and therefore helps clarify this complex concept.

Through the understanding of biorobotic couplings, such cyborg systems may contribute to medicine, in cases where a prosthetic replaces a disabled biological subsystem while being controlled by the patient's brain. To this mean, researchers in the field of neuroprostethics have been studying techniques to have animals directly control robotic devices [10]–[13].

II. THE CYBORG SYSTEM

The concept of the Cyborg system is shown in Fig.2 and pictures of the system are represented in Fig.3. In the Cyborg Fly, the intended corrective flight maneuvers of a tethered fly (Fig.2.A) are measured with a high speed camera (2.B). These data are the input to a user-definable transfer function (2.C) that generates motor commands for a wheeled robot (2.D). As the robot moves through a cluttered environment, it returns sensory information (visual and range data) to a second transfer function (2.E) that generates an image for the flight arena (2.F). The fly responds to the visual stimulus, closing the loop. The individual components are described in detail below.

A. Fruit fly

We tethered wild-type fruit flies (*Drosophila melanogaster*) to a tungsten rod using standard procedures [14].

B. High speed camera: the digital wing beat analyzer

We measure the fly's reaction from changes in wing kinematics using a custom designed high speed vision system. The $\sim$ 230 Hz wing beat frequency of fruit flies puts stringent constraints on the acquisition and processing of the images. Our vision system robustly extracts the wing position in real time at 7000 Hz (3500 Hz per wing) and with minimal latencies (maximum 150 µs). The system feeds this data into an extended Kalman filter. The Kalman filter's state vector $\boldsymbol{x}$ provides a real-time estimation of wing beat frequency f, amplitude A, phase ϕ and mean angular position m, which can be transformed into motor commands to the robot. More details about the digital wing beat analyzer can be found under [15] and [16].

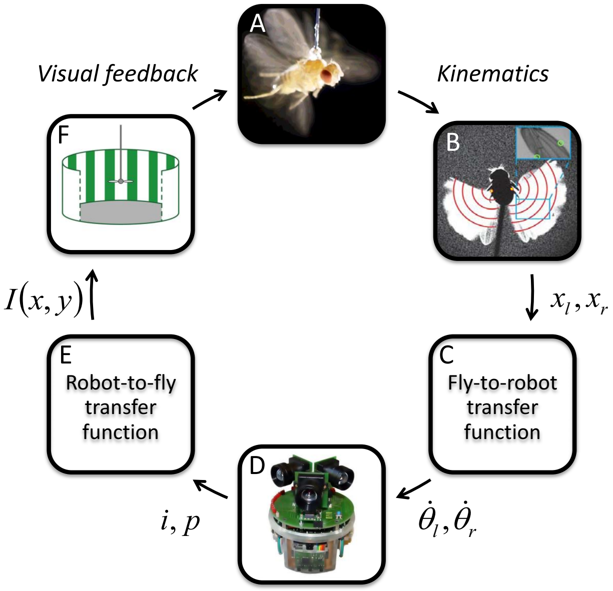

A
Visual feedback
Kinematics
B
F
$I(x,y)$
x_l, x_r
E
Robot-to-fly transfer function
C
Fly-to-robot transfer function
D
i, p
$\dot{\theta}_l, \dot{\theta}_r$

Fig. 2. The Cyborg Fly is composed of six processes. Each process is described in detail below.

C. Fly-to-robot transfer function

The fly-to-robot transfer function transforms the four Kalman parameters extracted by the high speed vision system into velocity commands to the wheels of the robot as follows:

$$\begin{pmatrix}\dot{\theta}_l \\ \dot{\theta}_r\end{pmatrix} = \mathrm{TF}_{\text{fly-to-robot}}(\boldsymbol{x_l}, \boldsymbol{x_r}) \quad (1)$$

$$= \mathrm{TF}_{\text{fly-to-robot}}(A_l, A_r, m_l, m_r, \phi_l, \phi_r, f) \quad (2)$$

where $\dot{\theta}_l$ and $\dot{\theta}_r$ are the angular speed commands to the left and right wheels of the robot [°/s], $\boldsymbol{x} = \begin{bmatrix}A & m & \phi & f\end{bmatrix}^T \in \mathbb{R}^4$ is the state vector of the left and right wing's Kalman filters as described in the previous paragraph. Because flies cannot separately modulate their left and right wing beat frequencies such that $f_l = f_r = f$ in (2). Note that each term in (1) and (2) are intrinsically time dependent and (1) may be a function of state vectors at different discrete times $(\boldsymbol{x}(t), \boldsymbol{x}(t-1), ..., \boldsymbol{x}(t-n))$.

The choice of (1) is completely unconstrained. Fruit flies, however, present a small repertoire of robust control responses: Yaw torque is controlled by varying the difference of stroke amplitude $(A_l - A_r)$ [17]. Lift is controlled by increasing the mean speed of both wings $(A \cdot f)$ [18]. Thrust is controlled by a modulation of both pitch (induced by changes in the mean stroke position m) and total force magnitude (proportional to $A \cdot f$) [19].

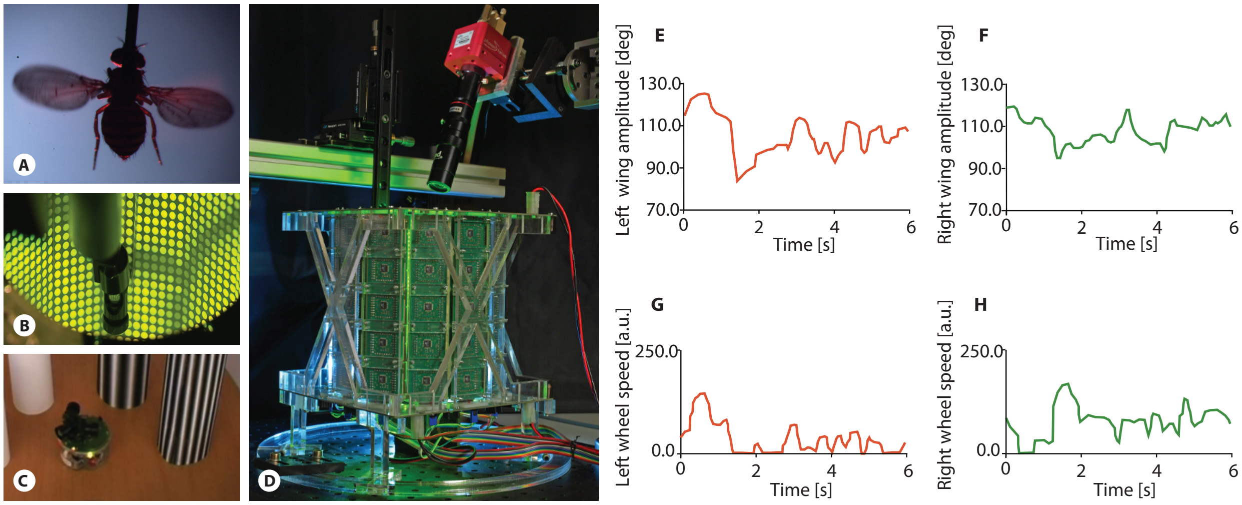


Fig. 3. Experimental setup and example of data. A: tethered fruit fly viewed from high speed camera. B: Inside view of flight arena C: E-puck robot inside maze D: Visual flight simulator with high speed camera E: Left wing amplitude F: Right wing amplitude G: Left wheel speed H: Right wheel speed. The fly-to-robot transfer function for this example is the one formulated in Equ. 3. Initially, the robot is turning right, but then a sudden decrease in the fly's left wing amplitude makes it turn left.

As a note of caution, tethered flight is subject to some quantifiable artifacts [19], because the tether disrupts the normal sensory feedback to the fly. In other words, a tethered fly's responses provide meaningful control signals, but one should be careful when generalizing these results to free flight.

Given this knowledge, a first logical approach is to use a control output that mimics the control of flight. Two general classes of transfer functions can be built that should result in a robot mimicking the fly's intended maneuvers:

- Yaw response of the robot is coupled to the yaw response of the fly:

$$\begin{pmatrix} \dot{\theta}_l \\ \dot{\theta}_r \end{pmatrix} = G_1 \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} A_l \\ A_r \end{bmatrix} + H_1 \qquad (3)$$

  where G_1 [s^{-1}] is the gain from stroke amplitude difference (proportional to the fly's turning torque) to wheel velocity difference (robot wheel velocity), and H_1 is a set speed constant.
- Forward velocity of the robot is coupled to the lift/thrust response of the fly:

$$\begin{pmatrix} \dot{\theta}_l \\ \dot{\theta}_r \end{pmatrix} = G_2 \cdot f \cdot \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} A_l \\ A_r \end{bmatrix} + H_2 \qquad (4)$$

  where G_2 is the unit-less gain from wing velocity (proportional to lift force) to forward robot velocity, and H_2 is an added speed constant.

D. Robot

We implemented the experiments on an e-puck (www.e-puck.org) robot equipped with an array of three linear cameras (102 pixels) each and 8 proximity sensors. The control of the robot wheels was performed at 50 Hz while the readout of the linear camera and proximity sensory was performed at 10 and 20 Hz, respectively. All communications with the robot were performed through a Bluetooth wireless interface. The robot control is programmed using microcontroller-specific C code.

E. Robot-to-fly transfer function

The visual image and/or the proximity sensor from the robot is employed to generate an image for the flight arena of the fly:

$$I(x, y) = \mathrm{TF}_{\text{robot-to-fly}}(\boldsymbol{i}, \boldsymbol{p}) \qquad (5)$$

where $I(x, y)$ represents the image to be shown on the flight arena. $\boldsymbol{i} = \left(i_1, i_2, ..., i_{318}\right) \in \mathbb{R}^{318}$ is the linear image from the three cameras of the robot and $\boldsymbol{p} = \left[p_1, p_2, ..., p_8\right] \in \mathbb{R}^8$ is the output of the eight proximity sensors.

Again, there are an infinite number of transfer functions that can be implemented. In this case, we focused on stimuli that are known to create robust yaw and lift responses in flies. Table I gives a list of the ones we used.

F. LED visual flight simulator

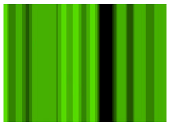
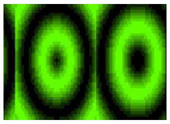
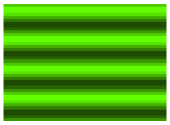
The LED visual flight simulator consisted in a 5x6 circular array of LED panels, each containing 8x8 LEDs. The individual panels were based on the design by Reiser *et al* [20], and were controlled via a National Instruments CompactRIO device (www.ni.com/compactrio/) through an I2C communication protocol. The flight simulator was refreshed at 30-400Hz, depending on the choice of the robot-to-fly transfer function.

G. Overall system control

The processes were controlled through a LabVIEW interface (National Instruments, www.ni.com) that provides a simple mean to manage the information flow and give the

TABLE I

DIFFERENT ROBOT-TO-FLY TRANSFER FUNCTIONS THAT WERE USED

Robot-to-fly transfer function type	Description	Image
A. Undistorted robot view	The fly directly sees the output of the linear cameras.	
B. Nearest object defines azimuth and speed of expansion/contraction	The center of expansion is set in the same direction as the nearest object. The distance to the object defines the speed of expansion. A center of contraction is generated at the exact opposite of the expansion.	
C. Brightest object defines stripe position	The azimuth of the brightest object in the linear camera image defines the azimuth of the bright stripe in the flight arena.	
D. Object distance drives sinusoidal grating vertical speed	If the nearest object is to the right (left), the pattern will go up (down, respectively). The closer the object, the faster the pattern motion.	

user the possibility to vary the transfer function parameters (see Fig.4). The high speed camera system ran on a separate computer and sent the Kalman state vector via UDP packets to the LabVIEW computer. The LabVIEW program read the UDP input and transformed it into motor commands by employing the user-defined fly-to-robot transfer function. In parallel, the LabVIEW program logged the sensory information from the robot and transformed it into a 2D image (via the user-defined robot-to-fly transfer function) that was sent to the CompactRIO device through Ethernet. The CompactRIO device transformed the image into I2C commands and sent them to the individual panels.

The hardware latency of the whole cyborg system is under 50 ms, mainly limited by the necessity to stream the robot's sensory information to the computer via Bluetooth.

III. EXPERIMENTS

A. *Naturalistic feedback*

In a first set of experiments, we used the most direct, natural, form of feedback. The robot's movements were matched to the predicted free-flight movements of the fly. This was done by combining Eq. (3) and (4) to have the robot turn proportionally to the difference in wing beat amplitude and advance at a speed proportional to the mean wing speed. Experimental data is shown in Fig.3.E-H. The fly was shown a live view from the robot's cameras (Table I.A). These feedbacks were similar to the feedback used in the cockroach experiments by Hertz *et al* [9].

This would seem naively to be the best choice of feedback, because it represents the most realistic mapping of the environment onto the animal, and vice versa. Interestingly, the system did not perform robustly in these circumstances. The robot would crash into obstacles, not showing the desired behavior.

B. *Amplified naturalistic feedback*

To overcome these issues, we enhanced the visual feedback to the fly while keeping the same fly-to-robot transfer function: We generated expansion/contraction stimuli based on the mean distance of surrounding objects: as an object got closer, the expansion would accelerate (Table I.B). In this amplified situation, the Cyborg system showed robust obstacle avoidance behavior (see ”CyborgFly.mp4” video in supplementary material). After adjusting the transfer function gains, the robot drove through the arena for several minutes while avoiding walls and obstacles.

C. *Inverted response feedback*

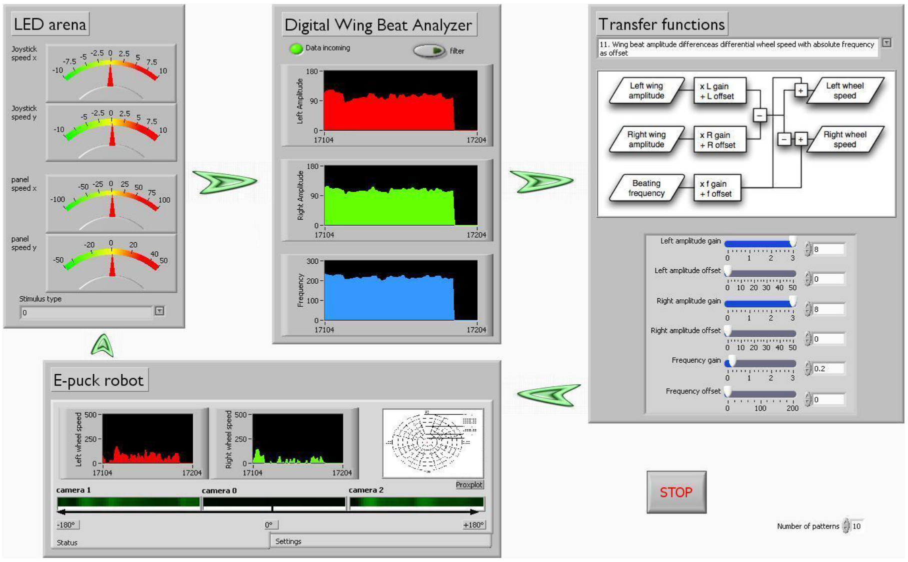
Flies are known to generate a robust stripe fixation behavior. We used this tracking behavior to generate obstacle avoidance in the robot. To this end, we placed the bright stripe pattern (Table I.C) in the opposite direction of the closest object. The tracking behavior of the fly was therefore coupled with the inverse behavior in the robot: obstacle avoidance.

D. *Decoupled response feedback*

Taking this artificial coupling even further, we coupled completely unrealistic behaviors together. For example, we used the lift response of the fly to control the turning of the robot:

$$\begin{pmatrix}\dot{\theta}_l\\ \dot{\theta}_r\end{pmatrix} = G_3 \cdot \begin{bmatrix}1 & 1\\ -1 & -1\end{bmatrix}\begin{bmatrix}f \cdot A_l - H_3\\ f \cdot A_r - H_4\end{bmatrix} + H_5 \tag{6}$$

To adjust the feedback accordingly, we used the mean distance of obstacles on each side of the robot to control the speed of ascent/descent of a stripe pattern (Table I.D). These experiments succeeded equally well in generating stable obstacle-avoidance behavior.



Stimulus type 0

11. Wing beat amplitude differenceas differential wheel speed with absolute frequency as offset

Left amplitude gain 8

Left amplitude offset 0

Right amplitude gain 8

Right amplitude offset 0

Frequency gain 0.2

Frequency offset 0

Number of patterns 10

Fig. 4. LabVIEW user interface: the user can visualize the fly's response (center) choose a transfer function type and set its gains (right side). The robot actuator and sensor data are logged at the bottom and transferred to the panel system via a second transfer function (left).

In summary, all of these experiments show that the emergent behavior of the system produces many non-intuitive results. Obviously, it is not so that a direct representation of the mapping leads to the expected, stable, behavior, because of the large differences in terms of temporal and spatial scaling. Secondly, if an experimental paradigm does result in similar behavior, it doesn't necessarily represent a truly biomimetic implementation.

IV. DISCUSSION & CONCLUSION

This work set out to experimentally explore the dynamic couplings present in biorobotic implementations. In doing so, our platform provides a common framework for the disparate fields of biology and robotics to understand and evaluate how they can mutually benefit each other. As Webb shows [7], the transfer of a target biological process into a robotic one is a multi-step operation that can be summed up as: *validating a model of a biological system through a robotic implementation*.

In the case where the robotic implementation is used to better understand a biological process, this validation is clearly necessary to verify that the robotic model can really be used as a replacement for the biological one. The robotic platform can then be employed to make new predictions about the target system in a whole new set of experimental conditions. In the case where the robotic implementation is used to perform the same function as the biological one (biomimetics), the validation is necessary to make sure that the target system's behavior is indeed reproduced.

As we have underlined before, this validation is not straightforward and is prone to misconceptions. For instance, if the artificial system's behavior matches the target system's behavior, it does not necessarily mean that the hypothesis is correct. Conversely, if the artificial system's behavior does *not* match the target system's behavior, it does not either mean that the hypothesis is false [7]! Let us illustrate this concept using the Cyborg Fly.

In our first set of experiments, we showed how a naturalistic feedback does not necessarily lead to a useful response. In the experiments, the transfer functions were chosen so that the robot would perform the fly's intended movements and the fly was given a direct visual feedback from the robot. Nonetheless, the robot failed to perform in any useful way.

The cause of this apparent contradiction lies in the way the biological and robotic systems are abstracted: we implicitly hypothesize that the robot reproduces the fly's behavior. For this, we use models of insect flight control (Eq. (3) & (4)) and transcribe them into commands that generate similar types of responses in the robot. However, the fly and the robot systems are very different and their responses cannot be so easily compared. The fly turns 90° within 50 ms, the robot takes at least one hundred times longer. The fly uses its wings to advance through a fluid volume while the robot rolls on a flat two-dimensional surface. The claim that "the robot reproduces the fly's behavior" has not taken into account neither the vast spatiotemporal differences nor the differences in implementation that was necessary to compare these disparate systems. In these first experiments, the fly probably saw a quasi-static image that generated very little optical flow. With the lack of meaningful visual feedback, a robust closed-loop behavior can not be expected.

It is interesting to note that we are fast at finding a critical cause to the failure of our system. However, had the system worked, e.g. due to some artifact, we would have very likely not seen or even intentionally ignored the differences cited above, on the basis that the system was functional. This first set of experiments therefore shows how important it is to correctly make abstraction of the spatiotemporal and implementation differences, and to critically analyze the outcome of a biorobotic implementation.

In the second set of experiments, we showed how we can overcome these differences by altering the model: we link the distance of an object to the expansion speed of a stimulus. By doing so, we compensate for the discrepancy in time constants between the fly and the robot by introducing a derivative operator (position is transformed into speed). This results in a functional closed-loop behavior, where the fly-driven robot is able to avoid obstacles. The underlying hypothesis, however, has changed. We cannot claim anymore to have a naturalistic feedback as we have added an amplification term. This change of hypothesis is often neglected in biorobotic approaches, where the model modifications are seen as small adjustments that help the system run smoothly. As these modifications accumulate, their combined effect can become more important than the biological model itself, strongly undermining the ”bio-inspired” aspect.

For the biomimetic case, this results in a non-biomimetic engineered device that is simply ill-named. This does not have a very dramatic effect, except if the optimality of the biological system is (falsely) used as an argument during the design phase. For the biorobotic case, this can lead to false scientific claims and is therefore quite dangerous, especially if the model modifications are not openly declared and are, as a consequence, very hard to find.

Our third and fourth experiments showed that it is not sufficient for a robotic system to perform just like its biological counterpart to claim that the underlying processes are the same. To illustrate this, we used a partially (III-C) or completely (III-D) decoupled processes to generate stable closed-loop turning behavior (Eq. 6). In our case, we have done it intentionally. Unfortunately, such effects are often unforeseen artifacts of the experimental paradigm, and can lead to false claims. For instance, the turning behavior of fruit flies in tethered setups is strongly affected by the lack of gyroscopic feedback. It is therefore dangerous to make conclusions on free-flight turning based on tethered experiments.

The simple ”success/failure” classification done here has the advantage of providing clear, unambiguous, results for our different case studies. Our future work will nonetheless complement this qualitative assessment, and characterize quantitatively the performance of the Cyborg system.

In summary, the biorobotic platform presented here offers a unique way to analyze the couplings between biology and robotics. Our work has demonstrated how the choice of coupling between the biology and robotic processes often lead to unexpected properties. From these observations, the guidelines to a meaningful pursuit of biorobotic approaches can be drawn. Such understanding is crucial to biomimetic implementations and robotic-augmented biology. It may also potentially benefit the medical field of brain-robot interfaces.

V. ACKNOWLEDGEMENTS

We thank Jean-Christophe Zufferey and Dario Floreano (www.lis.epfl.ch) for valuable discussions and for providing the e-puck robots, Urs Keller and Gilles Caprari for helping program the e-puck robots, Chris Rogers for helping program the panels, Jan Bartussek for helping run experiments, Mathias Moser for helping with the flight arena, and the attendees of the ”Cyborg Fly” workshop at the ”Flying Insects and Robots” symposium for their feedback.

REFERENCES

[1] X. Y. Deng, L. Schenato, and S. S. Sastry, “Flapping flight for biomimetic robotic insects: Part II - Flight control design,” *IEEE Transactions on Robotics*, vol. 22, no. 4, pp. 789–803, 2006.

[2] K.-H. Jeong, J. Kim, and L. P. Lee, “Biologically inspired artificial compound eyes,” *Science*, vol. 312, no. 5773, pp. 557–561, 2006.

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

[4] R. F. S. J. Franceschini, N., “A bio-inspired flying robot sheds light on insect piloting abilities,” *Curr Biol*, vol. 17, no. 4, pp. 329–335, 2007.

[5] J. Halloy, G. Sempo, G. Caprari, C. Rivault, M. Asadpour, F. Tache, I. Said, V. Durier, S. Canonge, J. M. Ame, C. Detrain, N. Correll, A. Martinoli, F. Mondada, R. Siegwart, and J. L. Deneubourg, “Social integration of robots into groups of cockroaches to control self-organized choices,” *Science*, vol. 318, no. 5853, pp. 1155–1158, 2007.

[6] A. Abbott, “Biological robotics: Working out the bugs,” *Nature*, vol. 445, no. 7125, pp. 250–253, 2007.

[7] B. Webb, “Validating biorobotic models,” *Journal of Neural Engineering*, vol. 3, no. 3, pp. R25–R35, 2006.

[8] E. Datteri and G. Tamburrini, “Biorobotic experiments for the discovery of biological mechanisms,” *Philosophy of Science*, vol. 74, no. 3, pp. 409–430, 2007.

[9] G. Hertz. (2008) Cockroach controlled mobile robot. [Online]. Available: http://www.conceptlab.com/roachbot/

[10] M. Kositsky, A. Karniel, S. Alford, F. K. M., and M.-I. F. A., “Dynamical dimension of a hybrid neurorobotic system,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 11, no. 2, pp. 155–159, 2003.

[11] H. Sato, C. Berry, B. Casey, G. Lavella, Y. Yao, J. VandenBrooks, and M. Maharbiz, “A cyborg beetle: Insect flight control through an implantable, tetherless microsystem,” in *MEMS 2008*, Tucson, AZ, USA, 2008.

[12] J. Wessberg, C. R. Stambaugh, J. D. Kralik, P. D. Beck, M. Laubach, J. K. Chapin, J. Kim, J. Biggs, M. A. Srinivasan, and M. A. L. Nicolelis, “Real-time prediction of hand trajectory by ensembles of cortical neurons in primates,” *Nature*, vol. 408, no. 6810, pp. 361–365, 2000.

[13] A. Abbott, “Neuroprosthetics: In search of the sixth sense,” *Nature*, vol. 442, no. 7099, pp. 125–127, 2006.

[14] L. F. Tammero and M. H. Dickinson, “Collision-avoidance and landing responses are mediated by separate pathways in the fruit fly, Drosophila melanogaster,” *Journal of Experimental Biology*, vol. 205, no. 18, pp. 2785–2798, 2002.

[15] C. F. Graetzel, S. N. Fry, and B. J. Nelson, “A 6000 hz computer vision system for real-time wing beat analysis of Drosophila,” in *The First IEEE / RASEMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, Pisa, 2006, pp. 278–283.

[16] C. F. Graetzel, B. J. Nelson, and S. N. Fry, “A dynamic region-of-interest vision tracking system applied to the real-time wing kinematic analysis of tethered Drosophila,” *(submitted)*, 2008.

[17] S. N. Fry, R. Sayaman, and M. H. Dickinson, “The aerodynamics of free-flight maneuvers in Drosophila,” *Science*, vol. 300, no. 5618, pp. 495–498, 2003.

[18] F. O. Lehmann and M. H. Dickinson, “The changes in power requirements and muscle efficiency during elevated force production in the fruit fly Drosophila melanogaster,” *Journal of Experimental Biology*, vol. 200, no. 7, pp. 1133–1143, 1997.

[19] S. N. Fry, R. Sayaman, and M. H. Dickinson, “The aerodynamics of hovering flight in Drosophila,” *Journal of Experimental Biology*, vol. 208, no. 12, pp. 2303–2318, 2005.

[20] M. B. Reiser and M. H. Dickinson, “A modular display system for insect behavioral neuroscience,” *Journal of Neuroscience Methods*, vol. 167, no. 2, pp. 127–139, 2008.