MEMS & high speed vision development and application to reverse-engineer Drosophila flight control

Author(s):
Graetzel, Chauncey

Publication Date:
2008

Permanent Link:
https://doi.org/10.3929/ethz-a-005742422

Rights / License:
In Copyright - Non-Commercial Use Permitted

This page was generated automatically upon download from the ETH Zurich Research Collection. For more information please consult the Terms of use.
MEMS & High Speed Vision: Development and Application to Reverse-engineer *Drosophila* Flight Control

A dissertation submitted to the

ETH ZURICH

for the degree of

DOCTOR OF SCIENCES

presented by

Chauncey Graetzel

Dipl. Ing. Microtech., EPFL
Born on September 29, 1980
Citizen of St-Sulpice, Vaud

Accepted on the recommendation of

Prof. Dr. Bradley Nelson examiner
Dr. Steven Fry, co-examiner
Prof. Dr. Xinyan Deng, co-examiner

2008
I would like to thank my advisers, Prof. Dr. Bradley Nelson and Dr. Steven Fry, for the support they provided throughout my PhD work. Multidisciplinary research between two labs is a challenging task, and I am grateful for the open-mindedness they showed throughout the PhD. I also thank Prof. Xinyan Deng and Prof. Raffaello D’Andrea for accepting to serve on my thesis committee.

I thank all the members of the Institute of Robotics and Intelligent Systems for accompanying me all along this journey, especially Simon Muntwyler and Felix Beyeler for the help they provided on the MEMS sensors. I am grateful to my colleagues at the Fly group, Jan Bartussek, Vasco Medici, Nicola Rohrseitz and Alexander Attinger, for their valuable feedback and discussions.

This work would not have been the same without the contributions of the students that worked under my supervision. I thank Daniel Kistler, Urs Keller, Philippe Goffin, Baran Eren, Christian Gehring, Claude Holenstein and Michael Bütkofer for their great Bachelor thesis. I thank Vinzenz Schönfelder and Mathias Moser for the commitment they showed during their Master thesis. Mathias was further involved with the project for an extra year following his thesis and I am extremely grateful for this time he spent, which turned out to be highly rewarding for all of us.

I also thank Prof. Chris Rogers for the large amount of energy he injected in the project during his 2006-2007 sabbatical at ETHZ and his continued interest in the project since.

Last but not least, I wish to thank my family, especially Michael, Carole, Aimie, Lili and above all my wife, Jenny, for the support they showed throughout these years. I am lucky to be surrounded by such great people.
# CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.2.3</td>
<td>Existing solutions</td>
<td>30</td>
</tr>
<tr>
<td>4.2.4</td>
<td>Design requirements</td>
<td>31</td>
</tr>
<tr>
<td>4.2.5</td>
<td>Implementation</td>
<td>33</td>
</tr>
<tr>
<td>4.2.6</td>
<td>Calibration</td>
<td>34</td>
</tr>
<tr>
<td>4.2.7</td>
<td>Real-time data acquisition</td>
<td>36</td>
</tr>
<tr>
<td>4.2.8</td>
<td>Results</td>
<td>42</td>
</tr>
<tr>
<td>4.2.9</td>
<td>Discussion</td>
<td>45</td>
</tr>
<tr>
<td>4.3</td>
<td>Visual flight simulator</td>
<td>48</td>
</tr>
<tr>
<td>4.3.1</td>
<td>Other insect visual displays</td>
<td>48</td>
</tr>
<tr>
<td>4.3.2</td>
<td>Design requirements</td>
<td>49</td>
</tr>
<tr>
<td>4.3.3</td>
<td>Implementation</td>
<td>49</td>
</tr>
<tr>
<td>5</td>
<td>Reverse-engineering biological flight</td>
<td>55</td>
</tr>
<tr>
<td>5.1</td>
<td>Biomechanics of tethered flight</td>
<td>56</td>
</tr>
<tr>
<td>5.1.1</td>
<td>Results</td>
<td>59</td>
</tr>
<tr>
<td>5.2</td>
<td>Lift control</td>
<td>60</td>
</tr>
<tr>
<td>5.2.1</td>
<td>Methods</td>
<td>60</td>
</tr>
<tr>
<td>5.2.2</td>
<td>Experiments</td>
<td>62</td>
</tr>
<tr>
<td>5.2.3</td>
<td>Results</td>
<td>63</td>
</tr>
<tr>
<td>5.2.4</td>
<td>Discussion</td>
<td>66</td>
</tr>
<tr>
<td>5.2.5</td>
<td>Conclusion</td>
<td>68</td>
</tr>
<tr>
<td>5.3</td>
<td>Thrust control</td>
<td>69</td>
</tr>
<tr>
<td>5.3.1</td>
<td>Methods</td>
<td>69</td>
</tr>
<tr>
<td>5.3.2</td>
<td>Results</td>
<td>70</td>
</tr>
<tr>
<td>5.3.3</td>
<td>Discussion and conclusion</td>
<td>70</td>
</tr>
<tr>
<td>6</td>
<td>Closing remarks</td>
<td>73</td>
</tr>
<tr>
<td>6.1</td>
<td>Contributions</td>
<td>73</td>
</tr>
<tr>
<td>6.2</td>
<td>Outlook</td>
<td>74</td>
</tr>
<tr>
<td>A</td>
<td>MEMS calibration</td>
<td>77</td>
</tr>
<tr>
<td>A.1</td>
<td>Static calibration of three-axis sensors</td>
<td>77</td>
</tr>
<tr>
<td>A.1.1</td>
<td>Calibration procedure</td>
<td>77</td>
</tr>
<tr>
<td>A.1.2</td>
<td>Extraction of calibration matrix</td>
<td>79</td>
</tr>
<tr>
<td>A.2</td>
<td>Dynamic calibration</td>
<td>81</td>
</tr>
<tr>
<td>A.2.1</td>
<td>Introduction</td>
<td>81</td>
</tr>
<tr>
<td>A.2.2</td>
<td>Theoretical Modeling</td>
<td>82</td>
</tr>
<tr>
<td>A.2.3</td>
<td>Experimental identification</td>
<td>86</td>
</tr>
</tbody>
</table>
## CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.2.4 Dynamic calibration implementation</td>
<td>93</td>
</tr>
<tr>
<td>A.2.5 Validation of the dynamic calibration</td>
<td>95</td>
</tr>
<tr>
<td>A.2.6 Results</td>
<td>95</td>
</tr>
<tr>
<td>A.2.7 As of when is a dynamic calibration necessary?</td>
<td>96</td>
</tr>
<tr>
<td>B QSS model of lift generation</td>
<td>99</td>
</tr>
<tr>
<td>B.1 Model variables and constants</td>
<td>99</td>
</tr>
<tr>
<td>B.2 Inertial forces</td>
<td>100</td>
</tr>
<tr>
<td>B.3 Aerodynamic forces</td>
<td>101</td>
</tr>
<tr>
<td>C Exploration of bio/robot coupling</td>
<td>103</td>
</tr>
<tr>
<td>C.1 The Cyborg system</td>
<td>105</td>
</tr>
<tr>
<td>C.1.1 Fruit fly</td>
<td>106</td>
</tr>
<tr>
<td>C.1.2 Digital wing beat analyzer</td>
<td>106</td>
</tr>
<tr>
<td>C.1.3 Fly-to-robot transfer function</td>
<td>108</td>
</tr>
<tr>
<td>C.1.4 Robot</td>
<td>109</td>
</tr>
<tr>
<td>C.1.5 Robot-to-fly transfer function</td>
<td>109</td>
</tr>
<tr>
<td>C.1.6 LED visual flight simulator</td>
<td>110</td>
</tr>
<tr>
<td>C.1.7 Overall system control</td>
<td>110</td>
</tr>
<tr>
<td>C.2 Experiments</td>
<td>111</td>
</tr>
<tr>
<td>C.2.1 Naturalistic feedback</td>
<td>111</td>
</tr>
<tr>
<td>C.2.2 Amplified naturalistic feedback</td>
<td>112</td>
</tr>
<tr>
<td>C.2.3 Inverted response feedback</td>
<td>112</td>
</tr>
<tr>
<td>C.2.4 Decoupled response feedback</td>
<td>113</td>
</tr>
<tr>
<td>C.3 Discussion &amp; conclusion</td>
<td>113</td>
</tr>
<tr>
<td>D Press coverage</td>
<td>117</td>
</tr>
<tr>
<td>References</td>
<td>120</td>
</tr>
<tr>
<td>Curriculum Vitae</td>
<td>133</td>
</tr>
</tbody>
</table>
Abstract

In this thesis, I developed novel robotic tools and applied them to reverse-engineer insect flight biomechanics and flight control. The robotic tools are not only novel technologies that provide access to previously unmeasurable data, but also analytical tools that provide a new way to interpret the data.

The model organism I studied was the fruit fly, *Drosophila melanogaster*. Fruit flies achieve an awe-inspiring control of their inherently unstable flight using only limited neural resources. They therefore represent an important model for the understanding of sensorimotor pathways and have attracted the attention of engineers developing micro robotic devices at similar Reynolds numbers.

From a technological perspective, the spatial and temporal scales of flight control in *Drosophila* are challenging. This thesis therefore resulted in product-level tools that pushed technological limits further to meet the demands of the biological application.

Concretely, three new tools were developed: MEMS micro force sensors, high speed vision systems, and visual flight arenas. The MEMS sensors provide a leap in the resolution (< 1 μN), bandwidth (> 1 kHz) and automation of biological force measurements. The high speed vision implementation provides a new tool to extract wing kinematics in real time at sampling rates above 6 kHz, which is several times faster than common vision-based tracking devices. Finally, the flight arenas were developed to stimulate the fly with precise visual patterns at high sampling rates.

Together, these three tools form a highly automated biorobotic platform that allows the exploration of open and closed-loop paradigms in flight control research. Furthermore, the developed methods have been extended to several complementary projects, proving their general value.

To apply these tools and study the sensorimotor pathways of *Drosophila*, I employed a system’s level approach. This ”black box” strategy requires both a precise control of the inputs and detailed knowledge of the outputs of the analyzed system. Tethered approaches, where the fly is mechanically held in place, ideally provide these
desired conditions because the sensory input is well defined and the response can be measured in greater detail than in free flight approaches.

The approach was applied to four main studies of flight control: 1) I experimentally validated a well-established aerodynamic model of flapping flight by comparing it for the first time with instantaneous flight forces. 2) I studied lift control by measuring its frequency response and characterizing it with system identification tools. 3) I analyzed speed control by concurrently measuring pitch, thrust and lift responses. The results were compared to free-flight experiments. 4) I dynamically coupled a mobile robot with the behavior of the fly and explored the effects of various coupling strategies.

In conclusion, these applications provide an overall advancement in the understanding of the biophysics and neural control mechanisms of biological flapping flight. From an engineering perspective, the technologies developed for these experiments have expanded the measurement possibilities at low Reynolds numbers and high temporal frequencies.
Dans cette thèse, de nouvelles microtechniques robotiques sont développées et appliquées à l’analyse de la biomécanique et du contrôle de vol d’insectes.

Ces techniques robotiques ne sont pas seulement des nouvelles technologies donnant accès à des mesures jusqu’alors inatteignables, mais aussi des techniques d’analyse qui permettent de caractériser l’organisme biologique de façon nouvelle.

Le modèle biologique étudié est la *Drosophila melanogaster*. Celle-ci réalise un contrôle précis de son vol malgré des conditions instables et des ressources neuronales limitées. Pour ces raisons, la *Drosophila* est un modèle important pour la compréhension du contrôle de vol biologique et a attiré l’attention d’ingénieurs développant des micro-robots volants.

L’analyse de phénomènes biologiques en question a nécessité le développement de trois nouveaux outils repoussant les limites technologiques: Premièrement, des capteurs de force MEMS ont été intégrés dans le système expérimental, permettant une amélioration de la résolution (< 1 µN), de la bande-passante (> 1 kHz) et du degré d’automatisation de mesures de forces en biologie. Deuxièmement, un système de vision a été développé permettant l’analyse en temps réel de la cinématique du battement d’ailes à plus de 6000 images par secondes, ce qui représente une amélioration significative par rapport aux méthodes existantes. Finalement, une arène de vol a été développée afin de stimuler visuellement la *Drosophila* à des fréquences supérieures à celle de son organe visuel (ca. 150 Hz), tout en couvrant la majorité de son champ visuel.

La combinaison des ces développements forment une plateforme biorobotique hautement automatisée permettant l’analyse du contrôle de vol biologique en boucle ouverte ou fermée. Les procédés ont aussi été étendus à plusieurs projets annexes, démontrant leur large champ d’application.

Afin d’appliquer ces outils à l’analyse du contrôle de vol biologique, j’ai utilisé une approche de "biologie du système". Cette approche considère l’organisme comme une boîte noire et nécessite donc une connaissance détaillée des entrées et sorties. Les
ouils développés dans le cadre de cette thèse sont idéaux pour une telle analyse.

L’approche de "biologie du système" a été appliquée dans quatre principales études: 1) Pour la première fois, un modèle aérodynamique du vol a été validé expérimentalement à travers des mesures instantanées de forces. 2) Le contrôle de la portance a été caractérisé par sa réponse fréquentielle et analysé à l’aide d’outils d’identification du système. 3) Le contrôle de vitesse a été caractérisé en mesurant la portance, la poussée et le tangage en réponse à des stimulations visuelles de vitesse. 4) Un robot mobile a été dynamiquement couplé au comportement de la Drosophile afin d’analyser les effets de différentes stratégies de couplage.

En conclusion, ces applications contribuent à une meilleure compréhension des mécanismes bio-physiques et neuronaux liés au contrôle du vol biologique. Du point de vue technique, les outils développés ont ouvert de nouvelles possibilités en matière de mesures à haute résolution et à hautes fréquences.
## Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASCII</td>
<td>American Standard Code for Information Interchange</td>
</tr>
<tr>
<td>AFM</td>
<td>Atomic Force Microscope</td>
</tr>
<tr>
<td>DAQ</td>
<td>Data Acquisition</td>
</tr>
<tr>
<td>DLP</td>
<td>Digital Light Processor</td>
</tr>
<tr>
<td>DOF</td>
<td>Degree Of Freedom</td>
</tr>
<tr>
<td>DWBA</td>
<td>Digital Wing Beat Analyzer</td>
</tr>
<tr>
<td>EKF</td>
<td>Extended Kalman Filter</td>
</tr>
<tr>
<td>FOV</td>
<td>Field of View</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>I2C</td>
<td>Inter-Integrated Circuit</td>
</tr>
<tr>
<td>INI</td>
<td>Institute of Neuroinformatics, ETH/Uni Zürich</td>
</tr>
<tr>
<td>IRIS</td>
<td>Institute of Robotics and Intelligent Systems, ETH Zürich</td>
</tr>
<tr>
<td>LED</td>
<td>Light Emitting Diode</td>
</tr>
<tr>
<td>MAV</td>
<td>Micro Aerial Vehicle</td>
</tr>
<tr>
<td>MEMS</td>
<td>Micro-electro-mechanical systems</td>
</tr>
<tr>
<td>PCB</td>
<td>Printed Circuit Board</td>
</tr>
<tr>
<td>ROI</td>
<td>Region of Interest</td>
</tr>
<tr>
<td>SOI</td>
<td>Silicon on Insulator</td>
</tr>
<tr>
<td>TWI</td>
<td>Two Wire Interface</td>
</tr>
</tbody>
</table>
Chapter 1

Motivation

This thesis lies in the exciting interaction between biology and robotics: novel micro-
robotic tools are developed and applied in a reverse-engineering approach to better
understand micro aerial flight in biology.

With the advent of microrobotics in the past 20 years, novel tools are allowing the
exploration of spatial and temporal scales that were completely inaccessible before.
Engineering is also providing a set of analytical tools that can be applied to quanti-
tatively characterize biological responses. By analyzing the biological organism with
the same techniques employed for man-made devices, such approaches facilitate the
transfer back to engineering [Abbott, 2007]. Such systems biology approaches have
gained a lot of momentum in recent years.

Another field that is emerging is ”robot-enabled biology”, where robot imple-
mentations of biological organisms are used to test hypotheses about the biological
entity, in cases where it is unpractical or even impossible to directly test on the
organism [Webb, 2006, Datteri and Tamburrini, 2007].

While engineering can improve biology, the converse is also true. As engineers
strive to build smaller, more autonomous micro robots, they are faced with a series
of challenges. The first one to come to mind is the miniaturization effort. Sensors,
actuators, power supplies and controllers must be built smaller. However, even with
an exact scaled copy of a successful ‘macro’ device, the device is unlikely to operate
successfully, because the various interaction forces have scaled differently [Purcell,
1977]. In recent years, engineers designing micro-flying and micro-swimming robots
have been exiting the high Reynolds number regime, where inertial effects dominate
the interaction forces, and have entered the world of intermediate Reynolds regime,
where both inertial and viscous forces must be taken into account. The physical
interaction with the environment becomes highly non-intuitive. When faced with a
new challenge, engineers have often turned to nature to seek design inspirations from the solutions that have evolved through millions of years of natural selection\footnote{There is no fundamental difference between "biomimetics" and more traditional engineering. Recently, many engineers have preferred "bioinspired" approaches, where biological solutions are used as starting points to traditional design. I will discuss Bioinspiration and biomimetics in more detail in Section\ref{sec:bioinspiration}.}. 

Another challenge linked to miniaturization lies in the control issues that arise: smaller devices are typically less stable than their macro counterparts because the inertial effects that play an important role in passive stability have scaled unfavorably compared to other phenomena. At the same time, computing power has also shrunk because the available payload has diminished. Through scaling, the problem is becomes more difficult on two levels: the controller has less computational power and the plant has become less stable [Rafal Zbikowski and Knowles, 2006]. Therefore, a simple yet efficient controller must be found to assure stability. Nature has evolved efficient sensory motor pathways that can generate stable yet maneuverable flight using only limited neural resources. The organism studied in this thesis is a prime example of such performance.

In summary, an exciting era has begun in the last decade: robotics and MEMS technologies are at a stage that is often compared to the IC era in the late 1970s, where they are enjoying an exponential growth and are only starting to realize their true potential [Gates, 2007]. Medicine and biology are using these improved tools to explore new areas. At the same time, interest in Life sciences has also recently boomed, driven by the huge potential underlying biotechnologies. This thesis is at this intersection, where robotics and biology mutually benefit each other.
Chapter 2

Introduction

The thesis is structured as followed. The introduction (Chapter 2) will first look at the main biological thematic, insect flight control. It will then focus on the main research questions addressed in this thesis. The approach (Chapter 3) will describe my strategy and how it compares to other strategies.

The rest of the dissertation is divided into two main parts; the "Developed technologies" chapter where the development of the micro technologies are described in the context of the biological application (Chapter 4), and the "Reverse-engineering biological flight" chapter where the technologies are applied to reverse-engineer flight control (Chapter 5).

2.1 Insect flight control

Insects possess a highly developed flight control system that allows them to perform stable yet maneuverable flight despite limited neural resources. This flight control system results from a highly interlinked process involving the sensory systems, the neural pathways, the motor controls and the biomechanics of flight [Dickinson, 2006].

A review of all the research done on each of these subsystems lies outside of the scope of this chapter, so I will concentrate on the more general aspects of flight control, underlining the seminal results, and point to review papers whenever possible. I will start with visual flight behavior, as it one of the most important modalities of flight control. It was also historically the first to be studied in detail.
2.1.1 Visual flight behavior

Vision is an essential sensory modality for flight control. There have been entire books devoted to insect vision, so I will only present the most important papers that were related to our research. For a more complete overview of insect vision, see [Nakayama, 1985, Wehner, 1981]. For an overview of the relationship between vision and behavior, see [Borst and Egelhaaf, 1989, Buchner, 1984, Collett et al., 1993].

Historically, early research on flight behavior focused on vision. Kennedy was one of the first to pioneer accurate, objective quantification of animals’ behavioral responses to visual stimuli [Kennedy, 1940]. Among other, he developed two important tools that are still in use today: a tethered experiment stage and a free flight wind tunnel.

The tethered experiment consisted in a tether that could freely rotate. Kennedy employed this to analyze basic optomotor responses in mosquitoes, where the insect was shown a simple visual stimulus and its body orientation was recorded. This allowed him to infer the preferred direction of flight.

Kennedy complemented these experiments with a free-flight wind tunnel, where he discovered important strategies about how visual and wind flow queues are integrated in the mosquito. These precursor technologies are remarkably similar to modern day tethered setups [Bender and Dickinson, 2006b] and wind tunnels [Fry et al., 2008].

Based on a similar systems approach, [Hassenstein and Reichardt, 1956] made important discoveries on the elementary motion detectors (EMD) of vision in tethered beetles: the famous Reichardt-Hassenstein correlator. Similar results were found in other insect species, such as the Musca [Eckert, 1973], Calliphora [Zaagman et al., 1977], apis [Kunze, 1961], Drosophila: [Götz, 1964], which have made this model one of the most influential in insect vision. Interestingly, this model has always been tested on tethered insects because it was technologically challenging to test it on free-flying animals. Fry recently developed a free-flight setup that allowed to test the correlator [Fry et al., 2008].

Researchers got interested in characterizing how the biomechanical performance was affected by the visual input. In his influential paper, Götz disclosed his helicopter model, where he showed that the fruit fly, Drosophila, could only control the magnitude but not the direction of its average flight force with respect to its body [Götz, 1964, Götz, 1968]. In his experiments, flies could control the azimuth of a vertical stripe by varying their torque. He showed that they elicited a strong stripe-fixation behavior, and he derived a first neural model of it.

The optomotor response and the neural circuits underlying it were further refined.
Heisenberg did important experiments analyzing the role of reafference in the fly under closed-loop conditions [Heisenberg and Wolf, 1988]. Electrophysiology on the large visual neurons, the lobula plate tangential cells, provided a better understanding of the visual processing [Joesch et al., 2008]. Electrophysiology was also performed on the motor output of flies and revealed the use of small control muscles, driven typically once per wing beat, to control the large power muscles in the thorax [Heide and Götz, 1996, Balint and Dickinson, 2001].

2.1.2 Mechanosensory feedback

There are numerous mechanosensors along the fly body. They fulfill different roles. Most notably, the hind wings of the fly, the halteres, lost their aerodynamic function and are employed as gyroscopes [Nalbach and Hengstenberger, 1986].

[Dickinson, 1999] showed that the halteres were responsible for much of the low-level stabilization at high frequencies. To be able to maneuver, the fly has to temporally destabilize. Dickinson proposed a general model for how different sensor modalities affect the halteres to create maneuvers [Dickinson, 2006].

Since this thesis is not specifically focused on mechanosensors, please refer to [Fayyazuddin and Dickinson, 1996] for more information.

2.1.3 Insect flight aerodynamics

Humans have always been fascinated by the ability of insects to produce enough lift to stay aloft. A flying insect and its reciprocating wings constitute a complex physical system, intricately linking wing motion, forces and behavior. Early back-of-the-envelope estimations led to the expression still popularly employed nowadays that ”science proved that bumblebees cannot fly”, although no scientific paper was directly published on this (see review by [Weisfogh and Jensen, 1956]). Researchers started using theory originally developed for airplanes and applied it to the study of insect dynamics [Jensen, 1956]. An important assumption inherent to this work was that the flapping wing could be decomposed as a series of instantaneous steady-state conditions. This assumption is called the quasi-steady assumption\footnote{Weis-Fogh did include one “unsteady” effect in his models: the clap and fling effect where the wings generate extra lift through the physical interaction at the end of the upstroke.}

The quasi-steady assumption was controversial, because it could not be tested experimentally at the time: all force measurements on live animals were made by averaging the forces over several wingbeats e.g. [Götz, 1964]. The measurement of
instantaneous forces first appeared at the end of the 1970s in larger insect species, such as the locusts [Cloupeau et al., 1979], or blow flies [Buckholz, 1981]. A detailed review of the available work led Ellington to oppose the "quasi-steady" assumption made by Weis-Fogh [Ellington, 1984a]. In his influential series of papers, Ellington developed a comprehensive analytic theory based on the unsteady vortices [Ellington, 1984c]. The model, however, could not estimate the instantaneous forces on airfoils and was used to characterize average forces.

Instantaneous force measurements on insect remained sparse. In fruit flies, the first instantaneous measurements were published by [Zanker and Götz, 1990], who used a taut wire to record the vibrations induced by the forces. However, these first measurements were of limited use because they were uncalibrated (forces given in relative units) and the dynamic effects of the wire’s vibration were not taken into account.

Dickinson employed a similar taut-wire system to measure lift and thrust responses in fruit flies [Dickinson and Götz, 1996]. This time, the system was calibrated and the measurements were synchronized with wake dynamic measurements to provide a general view of the aerodynamic and inertial effects taking place.

While these measurements paved the way, they also showed how difficult it was to accurately measure forces directly on an insect. These challenges led to the application of "indirect" force measurement techniques: computational fluid dynamics (CFD) and physical wing models.

These "indirect" techniques provided powerful analytic tools. Hypotheses about lift enhancing mechanisms can be tested over large parameter spaces, where it would be impractical or even impossible to test on the insect directly. For the results to be meaningful, the physical or computational models must obviously rely on precise measurements of the insect’s geometric and kinematic parameters, a task that remains challenging given the small dimensions and high temporal frequencies involved [Sane, 2003]. For CFD, intermediate Reynolds number simulations are computationally more expensive than other simulations, because neither the inertial nor the viscous components of the Navier-Stokes equation can be neglected [Sun and Lan, 2004, Ramamurti and Sandberg, 2007].

The physical wing model paradigm, in particular, allowed a quantum leap in the understanding of insect aerodynamics. By dynamically scaling the wing model to a larger size, force sensors could be directly integrated at the base of the robotic wing. This is an advantage compared to tethered force measurements, where one measures the resultant force on the insect’s thorax and therefore cannot isolate the individual body or wing components. Furthermore, by replacing the wings with
equivalent masses, one could further isolate the aerodynamic from the inertial components [Dickinson et al., 1999]. These measurements led to the formulation of the so-called Revised quasi-steady model of insect flight [Sane and Dickinson, 2001, Sane and Dickinson, 2002], that reintroduced quasi-steady assumptions as the leading model of lift generation.

The dynamically-scaled wing was also used to replay kinematics measured in free flight, allowing to reconstruct for the first time flight forces in free flight [Fry et al., 2003]. By comparing the forces generated in free flight to those generated in tethered flight, a quantitative comparison was made of the effects of tethering [Fry et al., 2005]. Fry showed that these disruptions were stereotyped and repeatable. In other words, tethered force measurements do not represent free flight forces, but a direct quantitative mapping exists between the two.

Although the aerodynamics of insect flight has largely been understood with the help of the dynamically scaled measurements, an experimental validation of the "revised quasi-steady" model with direct force measurements was never done, most likely because of the difficulty associated with simultaneously measuring instantaneous flight forces and kinematics. Such a validation is undertaken in this thesis, by using the novel tools we developed (see Section 5.1). For a more detailed review of insect flight aerodynamics, see [Sane, 2003].

2.1.4 Systems approach to flight control

The fly is increasingly becoming a model organism to explore systems-level processing [Dickinson, 2006]. This approach was motivated by the increasing knowledge that the subsystems constituting the fly’s sensorimotor pathways were highly interlinked [Dickinson et al., 2000]. A systems level approach could link what was happening at a very local level (neurons, muscles) to the overall behavior of the organism [Frye and Dickinson, 2004].

Aeronautical engineers started to get interested in the problem. Taylor applied state-space modeling to characterize the stability of tethered locusts using stroke-averaged forces [Taylor and Thomas, 2003]. In a subsequent paper, he performed the same analysis with instantaneous forces [Taylor and Zbikowski, 2005]. Zbikowski has been applying reverse-engineering approaches to the study of insect flight control [Rafal Zbikowski and Knowles, 2006]. Tanaka used a rigorous approach from aeronautics to characterize the lift controller in bumblebees [Tanaka and Kawachi, 2006].

A key aspect in such systems approaches lies in the ability to precisely define the
inputs and accurately measure the outputs [Taylor et al., 2008]. Given the speed at which control commands occur [Balint and Dickinson, 2004], stroke averaged measurements generally do not provide sufficient detail. Furthermore, while open-loop techniques provide a standard method to identify systems, closed-loop verification is often necessary to validate the identified system. These two characteristics require time-continuous and real-time measurement techniques, which will be an integral part of the technology development in this thesis.
Chapter 3

Approach

To address some of the relevant research questions presented in the previous chapter, a decision had to be made on the type of experiment to be implemented. A key choice is whether the experiments should use tethered or free-flying insects.

Fig 3.1: An overview of the research framework. See the text for details.
Taking a somewhat reductionist view, we can consider the fruit fly as a black-box encompassing sensory systems, neural pathways and motor output (see Fig 3.1). In free flight conditions, the fly interacts with the environment and receives natural feedback via its sensory systems (Fig 3.1 right). Free flight is intrinsically closed-loop, and open-loop conditions must be artificially created [Fry et al., 2008]. Tethered flight is intrinsically open-loop, and closed-loop conditions must be simulated (see Fig 3.1 left, and [Heisenberg and Wolf, 1988]).

3.1 Tethered vs. free flight

Tethered and free flight approaches each have their pros and cons. A nice thing is that these largely complement each other. Ideally, one should therefore verify results from tethered experiments in free-flight and vice-versa.

In tethered flight, the fly is held in place. This allows to employ more detailed measurement techniques that take advantage of the static position of the fly. It is also easier to provide controlled inputs to the fly. The condition is intrinsically open-loop: sensory feedback has been disrupted.

Free flight provides more naturalistic, less invasive, conditions and is therefore less prone to artifacts. Free flight is ideal for closed-loop experiments.

In my work, I worked with a tethered setup. The tethered approach allowed us to measure low-level flight control aspects that would have been impossible to measure in free flight: kinematics and dynamics of insect flight are notoriously difficult to measure due to the high beating frequency and low magnitude of changes. The tethered setup therefore provided an ideal method to explore flight control using a systems approach (See Section 2.1.4).

Tether artifacts

Tethering disrupts the natural feedback of flight. It is therefore essential to understand what these effects are and how one can cope with them. The insect’s various sensors all receive some form of feedback during flight. In tethered flight, however, the visual system, the antennae and especially the halteres do not receive the normal feedback during maneuvers. There have been workarounds implemented in the past, that all involve artificially closing the feedback loop. For vision, closed-loop turning response experiments have been made for many years, where the fly’s yaw torque is used to drive the azimuth position of a stripe [Götz, 1964]. For the antennae and halteres, flies have been tethered in a gimbal device that rotates the tethered fly up
3.1. TETHERED VS. FREE FLIGHT

to rotational speeds of 2000°/s [Sherman and Dickinson, 2003]. There have been experiments aimed at creating artificial Coriolis forces on the halteres by actuating the tether with force patterns that are phase-locked with the wing beat [Nalbach, 1993, Nalbach and Hengstenberg, 1994]. Flies have also been tethered to a needle with a special bearing that allowed it to freely move around the yaw axis [Bender and Dickinson, 2006b, Bender and Dickinson, 2006a]. All of these methods are powerful, but do not completely replace a free-flight verification because they always rely on a model of the behavior. The closed-loop turning response experiments, for instance, have been controversial: how should the turning torque affect the position of the fly? The biomechanics underlying this question are not so trivial, because a fly is beating its wings as it turns. In such artificial closed-loop experiments, there is therefore always the risk that one of the meaningful feedbacks is not provided correctly. [Taylor et al., 2008] argues that tethered experiments are most meaningful in open-loop, where the feedback is missing by design, but where it is acknowledged to be so.

Another effect of the tether is that it might alter the mechanical properties of the thorax. The thorax resonates at the wing beat frequency, driven by the large power muscles. This resonance is conveyed further to the wings via the control muscles acting on the wing hinge structure. The tether glued to the thorax obviously affects the biomechanical properties of this system. One would expect the rigidity of the structure to go up, which would result in a higher resonant frequency. This fact is supported by experimental evidence: tethered fruit flies often have wing beat frequencies around 230 Hz instead of about 200 Hz in free flight.

Finally, the tether imposes a fixed position to the body. During the course of a wingbeat in free flight, the body may move slightly to generate some inertial forces or change the center of gravity. These movements are not possible in tethered flight. If the fly is tethered to a device with a movable structure, such as a force sensor based on spring elements, then the dynamics of the movable structure add to the normal body dynamics.

A recent study obtained a good characterization of tethering effects by visually comparing wing stroke patterns in free and tethered flight in otherwise identical experimental conditions [Fry et al., 2005].

In summary, the artifacts created by tethered conditions must be carefully con-

\footnote{Taylor does not use the term open-loop and prefers to use the expression broken-loop to emphasize that the fly is still treating its sensor information as feedback. I will stick with “open-loop” as it is most widely used in the literature. This is purely a nomenclature definition and does not affect the interpretation of the results.}
sidered. The choice of experiment is crucial to avoid misinterpretations. Free flight verification is highly desirable. Obviously, it is usually not possible to exactly replicate the results in free flight (there would have been no need for a tethered approach), but the results can normally be generalized to a level where a comparison is possible.

3.2 Choice of model organism

We decided to work with the fruit fly, *Drosophila melanogaster*. The fruit fly has been a model organism in biology for over a century. It was originally popular because of the ease with which it could be grown and bred [Morgan, 1915]. Fruit flies are small enough so that several colonies can be raised in parallel in a small room; their gestation period is just around ten days, leading to reasonable time spans to go through several generations. Multiple phenotypes are also available to distinguish breeds. The fruit fly has therefore extensively been employed for genetic studies. The resulting genetic tools can benefit other areas of biology as well, where sensorimotor processes can be genetically isolated. This makes the fruit fly one of the most studied organism in science, as shown by the willingness of leading scientific journals to consecrate entire issues on the organism [Gunter et al., 2007].

For flight control, fruit flies represent an ideal model system: they show robust and highly stereotyped reflexive behaviors. Their neural control is essential to maintain stability and is therefore highly specialized.

For robotics, fruit flies represent an interesting, but still distant, model for autonomous devices. Current micro-aerial vehicles operate at scales that are at least an order of magnitude larger [Wood, 2008]. Nonetheless, certain aspects of fruit fly flight, such as specific lift generation mechanisms, have been imitated in flapping micro aerial vehicles. Micro-swimmers may also benefit from a better understanding of the biological processes at play, as their dynamic regime overlaps that of insect flight [Abbott et al., 2007].

The control techniques employed by biological organisms are especially interesting, because they use limited neural resources to produce stable yet highly maneuverable flight patterns [Rafal Zbikowski and Knowles, 2006]. Such efficient control techniques are very relevant to MAVs, where the small payload limits the complexity of onboard processing.
3.3 The importance of time-continuous, real-time, information

In the past, insect flight studies focused mostly on large time domains, since they are much easier to access experimentally. The flight forces and wing kinematics were averaged out over several wing beats.

The neural control mechanisms, however, act within a stroke. The neural signal phase-activates flight control muscles that change how the strains in the thorax are conveyed to the wings. A lot of relevant information is discarded by only studying the average effect of these signals. Furthermore, an instantaneous analysis of the wing kinematics and forces is necessary to study the important question of how the wing movements generate the intended forces.

In this thesis, I therefore decided to develop time-continuous, real-time measurement techniques. *Time-continuous* refers to a sufficient temporal resolution to measure discrete motor control commands. *Real-time* refers to the ability to extract behavioral information in real-time so that it can be used to drive external equipment or create closed-loop conditions. The real-time capability also allows to measure over extended periods of time, since only the extracted features are stored. This opens up the possibility to explore large parameter spaces in an automated fashion.

These two characteristics define key requirements for the technologies that were developed during this thesis (see Fig 3.1).
Chapter 4

Developed technologies

Analyzing micro aerial flight of an object the size of *Drosophila* is a challenging task. As dimensions scale down, accessibility becomes difficult. Wing beat frequency increases [Woods et al., 2001], putting temporal constraints on the measurement tools. Due to the increasing flight instability, the changes in kinematics required to generate rapid maneuvers are very subtle. Aerodynamic forces range in the micro Newtons. Furthermore, the sensory system has evolved to sample at high rates, making artificial sensory stimulation more difficult.

To overcome these challenges, we developed a set of novel microtechnologies that cope with these requirements (see Fig 3.1 for an overview). The first consisted in MEMS force sensors, which were designed to measure the instantaneous flight forces generated within a wing stroke. The second was a custom-designed high speed vision system, which was designed to measure the subtle changes in the temporal and spatial properties of the wing kinematics. The third was a visual flight simulator. It was built to take into account the near-panoramic field of view of flies and the ca. 150 Hz flicker rate of vision in fruit flies [Miall, 1978].

In the following sections, I will systematically discuss, for each of these three technologies, 1) the design requirements, 2) the current state-of-the-art, and 3) the implementation of the chosen design.

4.1 MEMS micro force sensors

Micro-electro-mechanical systems (MEMS) were pioneered in the 1980s, when engineers took micro fabrication techniques developed for the IC industry and applied them to the design of mechanical parts. MEMS technologies have since enjoyed a
formidable growth, leading to industrial applications such as the accelerometers featured in the airbag systems of cars and opening up new research avenues by allowing scientists to probe the micro world. Biology has benefited a lot from the new tools that MEMS has to offer [Abbott et al., 2007]. This thesis is one example of these applications.

4.1.1 Design requirements

The goal was to measure the instantaneous flight forces, which are composed of aerodynamic and inertial forces. Fruit flies weigh about 1 mg, and must therefore generate an average lift force of 10 \( \mu \text{N} \) to stay aloft. Since we are not only interested in the forces' mean value, but also in the time continuous effects, we must set temporal and force resolutions limits.

Temporal and force resolutions

Based on previous measurements of lift forces in fruit flies [Dickinson and Götz, 1996], the forces range from about -30 \( \mu \text{N} \) to +60 \( \mu \text{N} \). The required resolution is more difficult to estimate: no one has accurately measured the minimal magnitude of voluntary steering force commands. We decided that a sensor capable of measuring changes of at least 10% of the body weight should be sufficient to capture the most significant responses of the fly. These estimations set the minimum resolution to around 1 \( \mu \text{N} \).

Regarding the temporal aspect, the forces generated by a flapping wing show large local peaks: Both ends of the stroke, pronation and supination, cause sudden peaks in the forces. Control muscles act at precise wing beat phases to alter the way in which the energy from the large power muscles is conveyed to the wings. To measure these effects, the bandwidth of the force measurement device must be well above the 200 Hz wing beat frequency. Measurements by [Dickinson and Götz, 1996] showed that significant information was still present at frequencies as high as 4 kHz.

Relevant flight force/torque components

Given the fly’s ability to maneuver in all six translational and rotational degrees of freedom, it is useful to review the role of the different forces and torques to flight control. Historically, yaw torque has been the most measured component, because it corresponds to the fly’s intention to turn and therefore represents the most robust
behavioral response. However, yaw turning in tethered flies is likely to be very different than yaw turning in free-flight, because the feedback to the halteres has been disrupted. Although a few groups have artificially recreated this feedback, either by rotating the entire tethered setup [Sherman and Dickinson, 2003], or by generating pseudo-Coriolis forces to the fly [Nalbach and Hengstenberg, 1994], these techniques are cumbersome to use with a concurrent measurement of forces because they generate an additional force that must be separated from the insect’s forces. For these reasons, I decided not to work on yaw forces and concentrated on other components of flight.

Lift force is the largest component of flight forces. The fly must continuously adjust lift throughout maneuvers to compensate for gravity. Much attention has been focused on lift in fruit flies: from stroke-averaged tethered measurements [Götz, 1968, Hesselberg and Lehmann, 2007], to calibrated instantaneous forces [Dickinson and Götz, 1996], to free-flight force estimations through robotic wing models [Fry et al., 2003]. In this thesis, I will continue to look at lift, but I will put more focus on the control of lift as a response to visual inputs. The automated setup that I developed in the course of this thesis allows to explore a large parameter space.

Thrust forces are important to understand the sensorimotor pathways involved in forward flight. Götz studied stroke-averaged thrust components and put it in relationship with the lift forces: he measured a constant angle of 24 degrees between the body axis and the mean flight force vector [Götz, 1968]. Dickinson measured the instantaneous components of thrust forces in Drosophila [Dickinson and Götz, 1996]. As a consequence of the constant angle in between flight force and body axis, forward speed should be initiated first by a pitch angle adjustment, followed by an increase in overall force production; a prediction confirmed by [David, 1982]. This makes a tethered analysis of forward velocity challenging, because the fly cannot change pitch angle, so the measured thrust response might just be an artifact of the fly trying to pitch forward (and not making it) and might have very little in common with a natural thrust response.

As just mentioned, pitch torque is a central component forward velocity control. [Götz, 1968] measured average pitch torques, but no measurement of instantaneous pitch forces have been done to our knowledge.

The two remaining components of flight, side slip force and roll torque, have a more minor role in insect flight. They are typically not associated with maneuvers (although some coupling may exist), and are therefore controlled mainly to adjust stability along those axis. I have decided not to concentrate on either of these components of flight.
4.1.2 State-of-the-art

Here I will discuss the suitability of various force sensing techniques to the problem at hand. I will also review previous force measurements on *Drosophila*.

**General classes of force sensing**

Various micro electromechanical systems (MEMS) force sensor have been realized:

- **Piezoelectric** [Rao et al., 1995]: Piezoelectric force sensors have micro-Newton resolutions. Their main advantage is that they are robust with respect to handling.

- **Piezoresistive** [Fahlbusch et al., 2002]: Piezoresistive force sensors have been widely used in micro force sensing and gripping. [Rangelow et al., 2002] have built 3DOF micro force sensors with a resolution of 1 nN.

- **Capacitive** [Sun et al., 2003]: 10 nN force resolutions have been reported. Compared to the other methods, capacitive microforce sensing has the advantage of low power requirements, no hysteresis and low sensitivity to temperature variation.

- **Magnetic** [Gibbs et al., 2004]: Force sensors based on magnetostriction have been developed, but to our knowledge, resolutions below 0.1 N have not yet been attained.

- **Scanning probe microscopy** Scanning probe microscopy, a technique to map the surface of a material, relies on the interaction force of a probe and the specimen. The most common type of scanning probe microscope, the atomic force microscope, measures this force through the deflection of a cantilever beam. Different optical methods, such as laser speckle or interference, are used to measure the deflection. Scanning probe microscopes have extremely fine force resolutions. Their main disadvantage lies in the limited accessibility of the probe, which is surrounded by the scanning actuation mechanism, and the complexity and cost of the whole device.

In view of the characteristics cited above, piezoelectric and magnetostriction are not a viable option for fruit fly force measurement because their resolutions are currently too coarse. The other three transduction modalities, piezoresistive, capacitive and scanning probe microscopy, have all been used to measure forces produced by tethered fruit flies:
4.1. MEMS MICRO FORCE SENSORS

<table>
<thead>
<tr>
<th>Reference</th>
<th>Force component</th>
<th>Temporal res.</th>
<th>Sensor type</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Götz, 1964]</td>
<td>yaw</td>
<td>stroke averaged</td>
<td>“torque meter”</td>
</tr>
<tr>
<td>Götz, 1968</td>
<td>lift; thrust;</td>
<td>stroke averaged</td>
<td>“force meter”</td>
</tr>
<tr>
<td>[Heisenberg and Wolf, 1979]</td>
<td>yaw</td>
<td>stroke averaged</td>
<td>closed-loop</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>“torque meter”</td>
</tr>
<tr>
<td>[Zanker and Götz, 1990]</td>
<td>lift</td>
<td>non calibrated</td>
<td>taut wire</td>
</tr>
<tr>
<td></td>
<td></td>
<td>instantaneous</td>
<td></td>
</tr>
<tr>
<td>[Dickinson and Götz, 1996]</td>
<td>lift &amp; thrust</td>
<td>instantaneous</td>
<td>taut wire</td>
</tr>
<tr>
<td>[Nasir et al., 2005]</td>
<td>lift, thrust</td>
<td>no force trace</td>
<td>piezoresistive</td>
</tr>
<tr>
<td></td>
<td>&amp; pitch</td>
<td></td>
<td>sensor</td>
</tr>
</tbody>
</table>

Previous applications to fruit fly force/torque measurements

As we can see in this Table, there have been three main strategies employed to measure forces and torques in tethered *Drosophila*:

1. The **torque meter** uses the reciprocal physical principle of a galvanometer, in which an electrical current creates a rotary deflection, i.e. the torque generated by the tethered fly is measured through the electrical current needed to keep the fly pointed in the same direction [Götz, 1964]. The current-to-torque relationship was shown to be linear. In a subsequent paper, this same principle was adapted to measure separately lift and thrust [Götz, 1968]. Numerous studies have used improved implementations of the Götz apparatus to measure stroke-averaged turning torques [Heisenberg and Wolf, 1988, Hesselberg and Lehmann, 2007].

2. The **taut-wire** device employs a similar principle to the atomic force microscope. The fruit fly is tethered to a wire under tension. The lift forces produced by the fly cause the wire to vibrate. These tiny vibrations are picked up by shining a laser on the wire and observing how the diffraction pattern moves with the help of a phototransducer [Zanker and Götz, 1990]. The taut-wire method was somewhat cumbersome to employ since it required careful calibration of the dynamic characteristics for each new fly [Dickinson and Götz, 1996]. The size of the device also prohibited its integration into flight simulators.

3. **Piezoresistive MEMS sensors** have been employed to measure flight forces of tethered fruit flies [Nasir et al., 2005]. The paper’s results show that the
resolution of the sensor was too coarse to measure the flight forces reliably. The only detectable signal was a predominant 200 Hz component in the FFT.

4.1.3 Sensor design & fabrication

A capacitive MEMS sensor was chosen to measure the instantaneous flight forces of fruit flies. Capacitive sensors have the advantage of high sensitivity, low drift and temperature invariance. Sun was the first to demonstrate direct measurements of fruit fly flight forces using MEMS sensors [Sun et al., 2005].

The majority of MEMS design and the totality of MEMS fabrication was done by Felix Beyeler and Simon Muntwyler at IRIS. A close collaboration was maintained to ensure that the sensors would meet the requirements for the fruit fly measurements. There were two types of sensors used for the fruit fly project: the 1 DOF, employed for lift measurements, and the 3 DOF, employed for concurrent lift, thrust and pitch measurements. The key parameters of design are the spring shapes and the inter comb distances. The sensors are fabricated from SOI wafers and go through a four step fabrication process. For more detail about sensor design and fabrication, see [Beyeler et al., 2008].

4.1.4 Sensor integration and readout

Once fabricated, the sensors are individually removed from the wafer and glued to a PCB with epoxy glue. The sensor is electrically connected to the PCB by manually wire bonding the connection pads.

An electronic readout was designed to measure the vast varying capacitance, as a part of the MSc thesis of Mathias Moser. The readout circuitry was an important
4.1. MEMS MICRO FORCE SENSORS

**Fig 4.2:** Wafer with completed sensors. This image shows different 3DOF force sensors after the four micro fabrication steps have been completed.

<table>
<thead>
<tr>
<th># axis:</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resolution:</td>
<td>0.3 (\mu)N</td>
</tr>
<tr>
<td>Range:</td>
<td>160 (\mu)N</td>
</tr>
<tr>
<td>Resonant frequency:</td>
<td>1800 Hz</td>
</tr>
</tbody>
</table>

(with mass of fly)

<table>
<thead>
<tr>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>(2 forces, 1 torque)</td>
</tr>
<tr>
<td>Fx: 15 (\mu)N</td>
</tr>
<tr>
<td>Fy: 4 (\mu)N</td>
</tr>
<tr>
<td>Mz: 8 nNm</td>
</tr>
<tr>
<td>Fx: 500 (\mu)N</td>
</tr>
<tr>
<td>Fy: 600 (\mu)N</td>
</tr>
<tr>
<td>Mz: 1700 nNm</td>
</tr>
<tr>
<td>370 Hz</td>
</tr>
</tbody>
</table>

**Fig 4.3:** Overview of the two types of MEMS sensors used to measure flight forces on the fruit fly. a) The single axis force sensor produced by the ETH startup FemtoTools (www.femtotools.com) was used to measure lift forces. b) The three axis force sensor IRIS FF1 designed and fabricated by Simon Muntwyler during his Master thesis [Muntwyler, 2006]
part of this project because the fruit fly application combined fine resolution with high sampling rate requirements. The readout circuitry had to sample very rapidly while keeping the electrical noise low. As opposed to static applications, it was not possible to improve the signal-to-noise ratio by low-pass filtering the data because the measured signal contained important information at high frequencies. Finally, the multi-axis measurements decreased the effective sampling rate because each capacitive pair had to be measured sequentially.

Two readout circuitries were implemented. The first is ideal for single axis sensors, but is not adapted for multi-axis. The second one was designed for multi-axis.

**Single-axis readout circuit**

The first readout is based on the MS3110 chip, which converts the difference in capacitance of two capacitors into a voltage. The main advantages of the readout circuitry are that it is simple to set up and the voltage output is easy to measure. The main drawbacks are that its sampling rate is limited to 8 kHz and it is only applicable for one capacitance pair. For multi-axis sensors, a MS3110 would be required for each capacitance pair. Furthermore, the ground voltages would have to be separated for each capacitance pair, because MS3110 varies this voltage for each measurement cycle. The separation of the grounds becomes very challenging for MEMS sensors above 2 DOF, because there are only limited physical connections to the movable part of the sensor, which acts as the ground. The total number of electrical pads connecting the sensor to the PCB is $3 \cdot n$, where $n$ is the number of DOFs.
4.1. MEMS MICRO FORCE SENSORS

4.1. MEMS MICRO FORCE SENSORS

Fig 4.5: Multi-axis readout concept. The MEMS sensor is plugged into the multi-axis readout that uses the chip PSØ21 (see Fig 4.6). An intermediate device packages the measurements into manageable sets for a normal computer (see Fig 4.9).

Fig 4.6: A: Multi-axis force sensors. B: multi-axis readout circuits. The force sensors plug into the readout. The readout sends out digital data to an acquisition box (see Fig 4.9).

Multi-axis readout circuit

The second readout system measures the discharge time of each capacitor individually. The main advantages of this readout are that it is extremely fast at measuring the capacitance, and doesn’t require a separation of the ground voltages. Its main drawbacks are that it is more complex, and provides results in a digital format (SPI) that has to be interpreted by some subsequent IC. This makes this readout ideal for multi-axis sensors. The number of electrical pads connecting the sensor to the PCB is $2 \cdot n + 1$.

Acquisition

For both readout circuits, digital acquisition boards were required to record the data. For the single axis readout circuit, a standard DAQ card was sufficient to capture the full range and resolution of the MS3110 chip at 8 kHz. For the multi-axis, the DAQ card needed to be more specific (see Fig 4.9), since it had to communicate to the readout via a SPI interface. The SPI interface is first used to set the measurement registers on the readout chip. The readout is then ready to acquire data. Each
Fig 4.7: The "nutshell" was designed during Mathias Moser’s MSc thesis to gather the SPI measurements sent from the multi-axis readout and package them into UDP packets that are sent via ethernet to a host computer. It contains a commercial microcontroller board (red) with an additional application-specific PCB layer to facilitate the connections.

measurement starts with the DAQ asking the readout to perform a measurement and the DAQ responding with the data.

4.1.5 Sensor calibration

The raw data from the readout has a relationship to a force. This relationship depends on the mechanical characteristics of the sensor, such as its spring stiffness, and on the electrical characteristics of the sensor and readout circuit, such as the size of the capacitors and the strategy employed to convert this capacitance to a voltage. Because each sensor is slightly different, the force/signal relationship must be characterized individually.

The MEMS sensor measures forces in an indirect way. The forces applied at the probe tip deform the spring elements and cause a relative displacement between the "moveable" combs and the "rigid" combs. In turn, this displacement changes the electrical capacitance of the comb pairs, which can be sensed by the electronics mentioned above. When low frequency forces are being applied to the tip, the forces and displacements can be considered to be proportional. As the frequency of the input forces increases, the forces start exciting vibrations in the sensor. The forces and displacements are not proportional anymore. This requires two types of calibrations, depending on the frequencies one is measuring. The first calibration procedure is
the static calibration, and assumes a proportional force/displacement dependency. It is a simple calibration procedure, because it does not depend on the time course of the forces. The second calibration procedure is called the dynamic calibration. It is intrinsically more complex, since it involves sweeping through the frequencies and characterizing the dynamic response of the sensor. The following table gives an overview on the three main calibration strategies that were employed.

| Table 4.1: Main characteristics of the static and dynamic calibration for single-axis and multi-axis sensors. The mechanical, electronic and resulting calibration equations are presented. $F$ is the applied force on the probe. $x$ is the resulting deformation of the sensor. $k$ and $K$ represent the spring constants, $g$ the dynamical model of the force/displacement characteristic. MS3110 represents the single-axis readout circuit, PS021 the multi-axis one. $k_1$ and $K_1$ represent the static calibration gains. $g$ represents the force-to-deflection dynamic transfer function and $f$ represents the force-to-voltage dynamic transfer function, as described in Section A.2 |
| --- | --- | --- |
| # axis | Static | Dynamic |
| Mechanic | $F = k \cdot x$ | $F = K \cdot x$ | $F = g^{-1}(x, \dot{x}, \ddot{x}, t)$ |
| Electronic | $V = MS3110(x)$ | $C = PS021(x)$ | $V = MS3110(x)$ |
| Calibration | $F = k_1 \cdot V$ | $F = K_1 \cdot C$ | $F = f^{-1}(V, \dot{V}, \ddot{V}, t)$ |

**Static calibration**

The static calibration procedure assumes the sensor’s electrical and mechanical characteristics can be reduced to a time-independent force/signal relationship. For the single axis case, the sensor transduction is furthermore linear, meaning this relationship can be completely described through a gain and an offset. To determine the gain and offset, a reference sensor is used. The reference sensor is brought in contact with the probe of the MEMS sensor, and the sensors’ outputs are recorded synchronously. A line is then fit through the force/signal plot.

The multi-axis calibration is a little more complicated for several reasons. First, there are no multi-axis reference sensors, so our strategy was to use a single-axis reference sensor and repeat the calibration for multiple directions. Second, pure torques are difficult to generate, so the torque calibration must be done by applying forces at precise points. Finally, the transduction properties of a given axis are not
independent of the forces/torques acting on the other axis, so that a calibration
matrix is in theory necessary for all combinations of forces and torques.

The multi-axis force/torque calibration is described in Section A.1.

Dynamic calibration

The dynamic calibration is more complicated than the static calibration. Ideally, a sinusoidal force source would be used, and the frequency response of the sensor characterized by varying the frequency of the sinusoid. Unfortunately, such an ideal force source does not exist, at least at the force scales we are looking at. We therefore used an alternative method to identify the mechanical and electrical models of the sensor. These models are then implemented into a set of differential equations to compute the forces from the capacitance measurements. The model identification is explained in detail in the appendix Section A.2.

4.1.6 Experimental implementation: fly tethering

Once the force sensor has been mounted on the PCB and the whole system calibrated, the biological experiments could begin. Much care had to be taken to design an appropriate tethering stage, because this was the most critical process in view of the sensors’ fragility. The tethering stage consisted of a spherical gimbal with three rotational degrees-of-freedom and a micromanipulator with three translational degrees-of-freedom, giving full control of the positioning (see Fig 4.8). These devices were mounted below an optical microscope.

Flies were first cooled down to 4 °C in a refrigerator. At this temperature, flies are immobilized but remain alive and will recover within a few minutes. They were then placed in the gimbal, which consisted in an aluminum block with a semi-spherical cavity in which a semi-sphere was placed. The semi-sphere has a small slit at its center, designed to hold the fly in position. The aluminum block was cooled to 4 °C to keep the fly immobile. It was cooled with a Peltier thermoelectric heat pump. The heat from the aluminum block was transported away by a fluid cooling system, originally designed for high end computer cooling. The semi-sphere could be rotated in its casing, allowing the fly’s angular position to be adjusted. This was extremely important to ensure a proper tethering angle. Furthermore, with the three-axis sensor, the fly had to be tethered from a sideway angle, because otherwise the sensor would have obstructed the vertical field-of-view of the microscope. The micromanipulator consisted of a three axis linear stage with a MEMS holder at the end.
4.1. MEMS MICRO FORCE SENSORS

Fig 4.8: Tethering setup. A: The fly is placed in the sarcophagus of an aluminium semi-sphere that allows the tethering angle to be precisely adjusted. The sphere is placed in an aluminium bloc that is cooled by a Peltier heat pump.

To tether the fly, a tiny droplet of UV curable glue (Loctite Clear Glass) was placed on the fly’s thorax. The tip of the MEMS sensor was then brought in contact with the glue. A UV light source was used to cure the glue. The sensor with the attached fly was lifted up and the fly was given a few minutes to recover. Flies usually initiated flight spontaneously or with a gentle air puff.

After the experiments were performed, the fly was removed from the sensor, either by gently brushing it off, or by dissolving the glue using a small amount of Acetone.

4.1.7 Data analysis

The measurements were saved in ASCII text format and interpreted using a set of MATLAB routines. Each measurement series was logged into a Research book and was then digitized in a ”measurement selection” file, where all measurement timestamps were entered. The routines would use the measurement selection file as list of measurements to analyze.

To organize the analysis among multiple users, the data was saved to a centralized, backed-up, server. The analysis scripts were organized on a subversion server, that allowed users to have an up-to-date set of routines, but still be able to revert to older versions if necessary.
Fig 4.9: A: Fruit fly tethered to force sensor. B: Instantaneous force measurements (mean ± S.D. for 100 consecutive wing strokes)

4.2 Digital wing beat analyzer

The measurement of instantaneous flight forces are of limited value if one doesn’t know the corresponding phase of the wingstroke. A detailed analysis of the wing kinematics also provides us with a more direct view of the flight motor output of the fruit fly. A kinematic analysis is therefore the perfect complement to the MEMS force measurements. The combination of the two allows to measure and understand i) the motor output from the neuromuscular system ii) the biomechanics of flight and iii) the resulting flight maneuvers. Here, I present a novel system developed in the scope of this thesis to extract the wing kinematics in real time.

4.2.1 Approach

The measurement of wing motion has typically been performed using an infrared light that projected the shadow of each wing through a crescent mask onto a photodiode receptor [Dickinson et al., 1993, Reynolds and Riley, 2002]. While this setup can measure the relative stroke amplitude of both wings, it is fundamentally limited by the reduction of a three degree of freedom movement, i.e. the wing angles, to a single dimensional analog value, i.e. the photodiode readout voltage. Furthermore, construction of the analogue hardware and the requirement for a precise alignment of
the fly prior to measurement have hindered a more wide-spread use of this technique.

We opted for a new system based on computer vision. Computer vision has
the advantage of being highly flexible and allowing additional functionality, such as
automatic calibration features, to be implemented. In addition, the two dimensional
view combined with prior knowledge of an organism’s dimensions could be used in the
future to obtain new metrics, such as a full 3D reconstruction of the wings’ positions.

4.2.2 High speed vision overview

Many processes in technology and science require systems to be controlled in real
time using a contact-free measurement based on visual techniques. Production line
processes may require faulty components to be identified visually and subsequently
removed. Visual servoing requires manipulator and object positions to be extracted
from live images to generate appropriate control commands [Imai et al., 2004]. Ex-
ternal vision systems are also employed to locate micro-robots which are typically not
equipped with position sensors [Yesin et al., 2006]. Real-time, contact-free measurement
techniques also play an increasingly important role in biology to track moving
organisms while controlling aspects of the experimental process [Ogawa et al.,
2005,Fry et al., 2004].

These processes share the requirement to sample consecutive images, process them
on a frame-by-frame basis and use the result of the image analysis to take appropriate
action for the process control. Real-time high speed image analysis is technically de-
manding and underlies constraints with respect to the amount of information that can
be transferred and processed between frames. The system bandwidth BW [bytes/s] is given by

\[
BW = \frac{f \cdot d \cdot FOV}{\sigma^2} < BW_{\text{max}}
\]  

(4.1)

where \( f \) [Hz] is the sampling frequency, \( d \) [bytes] the depth of each pixel, \( FOV \) [m²]
the field-of-view and \( \sigma \) [m] the smallest resolvable object size. The amount of data
generated at each frame is dictated by the application specific requirements for the
spatial resolution and FOV, which are related to the vision system’s pixel array size
and optics as follows:

\[1\text{For these equations, we assume that the smallest detectable change of position is one pixel wide (no subpixel resolution). For simplicity, we also assume a FOV and camera sensor that have the same width and height.}\]
\[
\begin{align*}
\sigma &= (z \cdot p)/F < \sigma_{\text{max}} \\
\text{FOV} &= N \sigma^2 > \text{FOV}_{\text{min}}
\end{align*}
\] (4.2)

where \( F \) is the lens focal length, \( z \) is the lens working distance, \( p \) is the pixel side length on the camera sensor and \( N \) is the total number of pixels on the camera sensor.

Bandwidth limitations become particularly restrictive when tracking small objects, which tend to move at higher relative velocities and cover larger areas relative to their size than larger objects. Consequently, \( f \) and \( \text{FOV}/\sigma^2 \) need to be increased, respectively, and the bandwidth, therefore, quickly becomes a limiting factor in practical applications.

To avoid bandwidth limitations, a general strategy of transferring only relevant information is clearly beneficial. Generally speaking, relevant information in tracking applications is often localized in space and time, and, therefore, selective image sampling and analysis should provide a powerful strategy to avoid bandwidth limitations in real-time vision tracking applications. Here, I describe concepts and techniques applied to implement a high performance real-time high-speed vision application based on a selective image sampling and analysis strategy. This concept is implemented using standard digital vision hardware and, thus, provides a flexible and affordable solution implemented in software.

The solution relies on a process model that provides the spatial and temporal information required to select relevant pixel locations for analysis. I only expose and transfer the relevant portion of the image using a dynamic region-of-interest (ROI) that varies in both position and size, which avoids being limited by image transfer bandwidth. In quantitative terms, I can replace the global FOV in (4.1) with a smaller local FOV, allowing the BW budget gain \( \text{FOV}_{\text{global}}/\text{FOV}_{\text{local}} \) to be used to achieve both a fine spatial resolution and a high frame rate, thus making efficient use of the available system bandwidth (see comparative diagram of Fig 4.10). The applied concepts are general and can be implemented in standard high speed video hardware in applications where bandwidth is a limitation.

### 4.2.3 Existing solutions

Taking inspiration from early tracking work [Weiss et al., 1987, Allen et al., 1993], various approaches have emerged that take advantage of recent advances in imaging technology and computing power. One common approach is to circumvent the bandwidth problem by employing specific knowledge about the task to reduce the
data generated at each image. For instance, Nagle Research (Cedar Park, TX, USA) inspects the 3D surface of 250'000 pharmaceutical tablets per hour using a linear beam of structured laser light to only illuminate a profile of the tablet at each frame. The deviation from the reference line is used as a measure of the tablet’s height. The disadvantage of such highly specialized solutions is the difficulty in adapting them to different applications.

A second strategy is to increase the effective bandwidth from parallelization of image acquisition and processing. For example, multiple temporally and spatially synchronized image acquisition systems have been used to increase the effective frame rate [Wilburn et al., 2005, Schuurman and Capson, 2004]. System bandwidth is increased only in proportion to the number of employed vision systems, however. The advantage of multiple camera systems is particularly problematic for small-scale applications, where space constraints are inherent.

Third, the bandwidth problem can be overcome by integrating image computations at the pixel level [Cembrano et al., 2004, Oku et al., 2005, Kagami et al., 2006, Miao et al., 2007]. Massively parallelized pixel level computations, such as linear convolutions [Cembrano et al., 2004], have been implemented in VLSI (Very Large Scale Integration) vision chips. In this way, relevant information is already computed on chip and only the higher order information needs to be transferred for further processing. An interesting variant of this concept has been implemented in a temporal contrast vision sensor that transmits the addresses (i.e. pixel locations) only of pixels experiencing supra-threshold variation in relative intensity [Zhang and Liu, 2006, Lichtsteiner et al., 2006]. By filtering relevant information content at the hardware level, only relevant image information is transferred with sub-ms latency to the image processing software, thus reducing the transfer requirements dramatically. VLSI techniques are bound to go beyond proof-of-concept experiments and will be increasingly integrated in complex applications as their pixel array sizes increase and as they become more available as packaged products.

4.2.4 Design requirements

A high speed vision system was required to obtain a time resolved and spatially detailed concurrent measurement of both wing positions. We also desired a real-time read-out of the stroke parameters to control external hardware. For example, to implement a visual flight 'simulator', I intended to use the measured kinematic parameters to provide the fly with realistic visual feedback in real time, as normally experienced in free flight. Similarly, I wanted to simulate realistic force feedback stim-
ulation to specialized gyroscopic organs (halteres), which required phase-coupling the mechanical stimulus with the wing stroke [Nalbach, 1991]. The challenge, therefore, was to perform a real-time measurement of the complex movements of the fruit fly’s two tiny wings (each only about 3 mm long), which beat back and forth more than 200 times each second.

To faithfully reconstruct the wing motion and identify relevant kinematic control parameters, the spatial resolution had to be sufficient to distinguish the subtle changes in wing kinematics, but coarse enough to allow the camera to run at sufficiently high framerates and for both wing paths to stay within the global FOV.

To determine the required spatial resolution, I turned to earlier studies of flight control. The wing kinematics of tethered and free flying fruit flies were captured by Fry et al. at a frame rate of 5 kHz (about 25 samples per wing stroke) [Fry et al., 2003,Fry et al., 2005]. A semi-manual outline fitting procedure was applied to measure the time course of wing motion ( [Fry et al., 2003], their figure 1.B). In free flight, the changes of stroke amplitude occurring during fast turning maneuvers (body saccades) were measured in the range of 10°. We therefore opted for an angular resolution of 1°, which is sufficient to distinguish even subtle steering commands. Given a wing span of about 3 mm, the spatial resolution corresponded to about 26 µm in the world coordinate frame

The global FOV needed to be large enough to cover the wing stroke path on each side, which required a FOV of at least 7x7 mm, or, given the spatial resolution above, 270x270 pixels.

\[2\] These calculations assume that wing position is measured halfway down the wing, and that the FOV encompasses the full wing.
4.2. DIGITAL WING BEAT ANALYZER

**Fig 4.11:** Calibration procedure a) Raw sequence images (1024x1024 pixels) b) Statistical analysis c) Area segmentation extraction of body and wings d) Wing path extraction: the circular arcs represent the extracted wing paths

Within the constraints of the system bandwidth, which in our case were limited by the camera, I was able to obtain sampling rates between 4 and 7 kHz, corresponding to 10 to 17 samples per stroke for each wing. The latter value represents an eight-fold increase to using a static ROI with the same hardware.

### 4.2.5 Implementation

We captured images of the tethered flying flies using a MVD-1024-Trackcam digital high speed video camera (Photonfocus AG, Pfäffikon, Switzerland), equipped with a 1024 by 1024 pixels CMOS image sensor. At full resolution, the camera achieves only 75 fps, however, the frame rate increases in proportion to a decrease in the chosen image size as predicted by Eq 4.1 for a constant BW. The ROI and exposure settings could be updated on a frame by frame basis. The camera was connected to a Silicon Software (Mannheim, Germany) Micro Enable III Framegrabber via a Camera Link interface. We used a standard commercial PC with a Intel Pentium IV Dualcore 2.8 GHz processor and 1024MB of RAM. The operating system was Windows XP. To avoid system interrupts from causing delays (typically on the order of 1 ms), the program’s thread was assigned real-time priority and ran separately from the operating system’s threads on one of the dual processors, thus achieving real-time performance. The software was programmed using Visual C++ 6.0 and made use of
OpenCV and Intel Performance Primitives (IPP).

The experiments were performed on an optical table. The camera was equipped with an Edmund Optics (New Jersey, USA) VZM 300i zoom lens with a primary magnification set to 0.75:1. The aperture was fully closed to 1.5 mm to maximize focal depth. The camera’s exposure time was set to 50 µs to minimize wing blurring [Fry et al., 2003]. Due to the small aperture and short exposure time, extremely bright lighting conditions were required. We achieved this by backlighting with a randomized bundle fiber light guide attached to a 150 W halogen light source (Schott ACE, Mainz, Germany). Diffusive tracing paper was placed 15 mm in front of the light source to provide a homogenous light distribution. A 650 nm long pass filter (RG 715 Longpass, Edmund Optics) eliminated light in the range visible to the flies [Juusola and Hardie, 2001] to avoid behavioral artifacts.

Flies were immobilized at 5 °C on a custom built Peltier cooled stage and glued to a tungsten probe with UV cured glue (Loctite, Duro Clear Glass Adhesive) using standard techniques [Tammero and Dickinson, 2002]. The probe with the attached fly was positioned along six degrees of freedom with a micromanipulator (Sutter MP285), such that the stroke plane roughly coincided with the camera image plane and the wing stroke path was not occluded by the tether.

4.2.6 Calibration

ROI based wing tracking and an optimized analysis of the sampled images requires detailed knowledge of the wings’ expected paths at the time of tracking. Because the tethering and lighting conditions vary considerably between consecutive preparations with different flies, detailed image information needs to be acquired between consecutive measurements. For this, we employ an automatic calibration procedure that a) segments the FOV into functionally relevant areas containing the fly body, the tether and the wing stroke envelope; b) determines the optimal threshold to distinguish the wings from the background; c) determines the pixel paths along which the wing will be followed during the tracking.

Image segmentation

The algorithms for image segmentation are based on an analysis of the varying image characteristics between functionally distinct regions (body, tether and stroke envelope). First, a set of 100 full-frame (1024*1024 pixels) images are acquired at 75 Hz, whereas the wing phase in each image is arbitrary (see Fig 4.11). Next, a statistical analysis is performed of the pixel values throughout the sequence. This results in two
images (see Fig 4.11b). The first defines a mask for the stroke envelope based on a binarized image of the variance. The second defines a background image based on the median pixel values.

The dark halo caused by the background lighting is then removed using a vignetting filter and the tether is extracted using a Hough transform. Finally, the body blob is extracted from size and position criteria and its orientation and heading used to separate the left and right sides of the fly (Fig 4.11c Body).

**Wing segmentation characteristics**

The appropriate choice of the threshold used to track the wing edges during tracking is critical. To accommodate for the variation between individual fly preparations, the wing edge detection threshold also needs to be calculated during the calibration phase.

The threshold is determined in two steps. First, the background image is subtracted to increase the signal-to-noise ratio by rendering the system robust to spatial variations in back-lighting. Second, the Max-Lloyd quantization algorithm is applied to the pixels belonging to the wing envelope (white pixels in the variance image Fig 4.11b). This results in an optimal threshold in the least square’s sense (see Fig 4.11c Wing).
**Wing path extraction**

The wing tracking paths are the key information extracted during calibration. For each wing, a circular arc with an optimal radius and centered on the wing hinge is chosen. These arcs correspond to a list of pixel locations used to locate the wing edges and position the ROI in subsequent frames.

To obtain the wing hinge position of each wing, the wing envelope is binarized using the wing edge detection threshold (see above), which provides a representation of the isolated wing. A morphological opening is performed to eliminate possible holes within the wing blob caused by the semi-transparent sections of the wing. Next, a Canny edge detector is applied to isolate the wing edges (see Fig 4.11: Wing edge). A Hough transform is applied to extract the strongest line. In most of the cases, this line corresponds to the leading edge of the wing. The intersection between each strongest line of the sequence is calculated. The median intersection is taken as an estimate of the wing hinge position (marked with full circles in Fig 4.11).

Combining the knowledge of the wing hinge positions and the stroke envelopes, the optimal circular path segments for subsequent wing tracking are chosen by applying an angular range criterion (see the circular arcs in Fig 4.11). The sequence of pixel locations of the path segments are stored together with the corresponding value describing the angle subtended with respect to the body, providing a lookup table for wing position in angular coordinates of the body.

### 4.2.7 Real-time data acquisition

To track the wings robustly, the ROI needs to be placed in appropriate positions along the wing path and the wing position in the ROI determined using efficient image analysis. To initiate and perform wing tracking, as well as to recover from loss of tracking (e.g. because of flight interruption), we implemented the tracking state machine shown in Fig 4.12. The process is described in detail below.

**Wing edge detection**

The tracking process begins with a detection mode, in which the ROI is placed at one of the two limits of the wing path (maximum or minimum point). This point has two advantages; first, because it is bijective, it corresponds to a unique phase of the wing beat kinematics, creating an unambiguous starting point to the tracking system. Second, the low wing velocity at these points allows the parametric model to adjust itself without losing track. Images are continuously acquired and analyzed.
using the static ROI until both the leading and trailing edge of the wing are detected. The parametric model is then initialized and the state machine switches to the wing edge tracking mode.

**Wing edge tracking**

In the wing edge tracking mode, the ROI is dynamically positioned according to the prediction of a parametric model. The ROI is placed at the position of the expected wing traversal, which can be several wing chords distant from the previous measurement. The current ROI is grabbed and the image processed (see below) to obtain a measure of the wing position, which is then used to update the parametric model of the wing kinematics. The ROI is then placed according to the prediction of the updated model and the process is repeated. If the process fails to detect the wing edges, it continues to estimate the ROI position according to its last measurement. After about 500 $\mu$s, the system reverses to the detection mode and sets the ROI back to its initial position. The grace period gives the system a chance to recover from faulty predictions, which typically happen during the first wing beat after the Kalman initialization. The longer interruptions were usually the result of an interruption of flight.

**Image analysis: wing extraction**

The time budget allotted to the processing of the individual images does not allow CPU intensive image analysis, such as blob analysis and pattern matching, to be applied. Using instead only the circular paths extracted during calibration (see Fig 4.11), it is possible to reduce the number of pixels to be processed from 2500 to about 60 and perform the necessary image analysis within a time frame of about 50 $\mu$s (see Fig 4.16 for details).

After transferring the image from the camera to the frame grabber, the following procedure is used to detect the wing edges: The pixel values of the wing path are copied from the frame grabber’s onboard memory to an array in the computer’s RAM (Fig 4.13.1). The background image captured during calibration is then subtracted from the sequence (Fig 4.13.2) and the resulting sequence binarized using the wing detection threshold (Fig 4.13.3). A morphological opening is applied to fill in gaps caused by the semi-transparent sections of the wing (Fig 4.13.4). Finally, an edge detector extracts the leading and trailing edge of the wing (Fig 4.13.5).
CHAPTER 4. DEVELOPED TECHNOLOGIES

1. Raw data extraction
2. Background subtraction
3. Binarization
4. Opening
5. Edge detection

Fig 4.13: Edge extraction algorithm applied along the wing path during tracking. The binarization threshold is based on the Max-Lloyd quantization performed during calibration. The morphological opening width is manually chosen to remove objects that are the size of the legs, but not objects that are the size of the wings.

Fig 4.14: Significance of the state vector: the shape of the model is defined by the relative amplitudes $\alpha_i$ and relative phases $\omega_i$ of each Fourier component ($N = 3$). This shape is scaled and offset, both in time ($\omega$ and $\xi$) and space ($M$ and $a_0$) by the EKF to fit the measurements.
### 4.2. DIGITAL WING BEAT ANALYZER

**Fig 4.15:** Sample from a tracking sequence, the squares and circles represent the extracted positive and negative edges, respectively. The rectangles show the borders of the ROI inside of which pixels were exposed and transferred during a given frame. The median image extracted during calibration is shown in the background for ease of interpretation.

**Process model: implementation of an Extended Kalman filter**

The parametric model used to predict wing motion was implemented using a Kalman filter, by virtue of its computational efficiency (due to its recursiveness) and optimality with respect to most statistical criteria [Maybeck, 1979].

The Kalman filter is initialized prior to entering the wing tracking mode. Its initial state estimate is based on parameters extracted during calibration and on previous knowledge of wing kinematics, e.g. typical wing beat frequency. During tracking, four Kalman filters are used simultaneously to track the positive and negative edges of both left and right wings. The measured wing edge positions are used to update the state estimate and thus increase the accuracy of the prediction for the subsequent measurements. To position the ROI in the subsequent frame, the current state estimates are used to predict the future positions of the positive and negative edges of the wing. The ROI is then centered around the mean value. As a further kinematic parameter, the difference of the wing edge coordinates provide a measure of the wings’ angle of attack.

Because the Kalman filter state estimate represents an online parametrization of the stroke kinematics, it can be used to control external processes in real time. An example for its use as a behavioral metric for real time feedback control is given in Section 4.2.8.

The periodic motion of each wing edge is modeled using a Fourier series:
\[ \theta = a_0 + \sum_{i=1}^{N} a_i \cos(i\omega t) + b_i \sin(i\omega t) \]  
(4.3)

where \( \theta \) represents the 1D angular position of the edge with respect to the hinge and \( N \) is the number of Fourier terms.

By constraining all parameters except amplitude, offset, frequency and phase, I maintained the original shape of the function, while scaling it in time and space. (4.3) is rewritten to incorporate these state variables (also see Fig 4.14):

\[
\begin{cases}
\theta = a_0 + \sum_{i=1}^{N} \alpha_i M \sin(i\xi + \varphi_i) \\
\xi = \omega t
\end{cases}
\]  
(4.4)

where \( \alpha_i \) is the relative amplitude of the Fourier coefficient \( i \), \( M \) is the amplitude of the function, \( \xi \) contains the phase information of the function with respect to the frequency \( 2\pi\omega \) (phase is dependent on frequency), and \( \varphi_i \) is the relative phase of the Fourier coefficient \( i \). The \( \alpha_i \) and \( \varphi_i \) coefficients are static values describing the shape of the model. The state vector is \( x_k = [ a_0 \ M \ \xi \ \omega ]^T \), and the derivation of the discrete state dynamics from (4.4) yields:

\[
x_k = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & T_s \\
0 & 0 & 0 & 1
\end{bmatrix} x_{k-1} + W
\]  
(4.5)

\[ W \sim N(0, Q) \]

where \( T_s \) is the discrete time step, \( A \) relates the state at the previous time step to the state at the current step, \( W \) is the process noise and \( Q \) is the process noise covariance. This noise is introduced to take into account the changes in kinematics performed by the fly (which to measure is our goal) and the imprecision inherent to any model.

The measurement equation is nonlinear. Therefore, an Extended Kalman Filter (EKF) is used, which linearizes the measurement equation at each time step:

\[
\theta = H x_k + v_k
\]

\[
H = \begin{bmatrix}
\frac{\partial \theta}{\partial a_0} & \frac{\partial \theta}{\partial M} & \frac{\partial \theta}{\partial \xi} & \frac{\partial \theta}{\partial \omega}
\end{bmatrix}
\]

\[ v_k \sim N(0, R) \]  
(4.6)

where \( v_k \) is the measurement noise and \( R \) is the measurement noise covariance. In the experiments, the static parameters \( \alpha_i \) and \( \varphi_i \) with a total of three Fourier terms \( (N = 3) \) were chosen to fit the kinematic measurements of the wing beat presented
in [Fry et al., 2005]. The two parameters of the Kalman filter, $R$ and $Q$, were determined empirically from a first series of measurements. The components of $Q$ affect how much the state variables will vary given a new measurement. These weights must be chosen to achieve a correct sensitivity. $R$ represents the confidence that a measurement is correct (not a false-positive). Together, $R$ and $Q$ determine how the model parameters will get affected by the new measurement: the state vector estimate is updated for each measurement by solving the Ricatti equations.

**Process time line**

The different processes must work together efficiently in a synchronized way to achieve the required frame rates given the hardware constraints. During wing tracking, three pieces of hardware are running in parallel: the camera, the frame-grabber and the computer’s CPU. These three components must be able to determine where the ROI must be placed before the next frame is transferred. The camera (Fig 4.16, top row) initially exposes the image during 50 $\mu$s. The camera then transfers the image to the framegrabber (Fig 4.16, middle row). The computer receives a signal once the transfer is complete (Fig 4.16, bottom row) and starts the image processing. The wing position is extracted as described in Section 4.2.7. The Kalman filter uses the measured position to update its state vector and the following ROI position is calculated (see previous section). This ROI is sent to the framegrabber which updates the camera once the current frame has been transferred.

Although the process for a single frame is serial, up to three of these processes are running simultaneously on the different hardware components to maximize the speed. For instance, the next frame is exposed and transferred while the current frame is being processed. This does not affect the tracking performance because the left and right sides are exposed sequentially, giving an extra frame period to update the ROI on each side.

**Data logging and hardware control**

During tracking, the Kalman filter state vector, the position of the ROI and the position of the leading and trailing edge of the wings are stored in the computer’s heap for each captured frame. The camera data (ROI pixel values) are left on the frame grabber where they are overwritten during the subsequent buffer cycle. While this procedure does not retain the majority of the image data due to bandwidth requirements, the last 1GB of data remain available on the framegrabber and were retrieved and saved to disk for post processing (Fig 4.15, see also Discussion).
4.2.8 Results

The resolution of the wing position was about 1° in our experiments. Various combinations of ROI sizes and frame rates were used to explore optimal parametrization for robust and detailed tracking. We obtained robust tracking performance in a range of FOVs and sampling rates between 4000 Hz and 7000 Hz, which was sufficient for our application.

In this system, the current limitation was the bandwidth of image acquisition and not image processing. At 7000 Hz, the size of the ROI is barely larger than the full width of the wing. Therefore, a further increase in temporal resolution could only be achieved at the cost of spatial resolution by reducing the magnification of the lens.

As shown with a representative example in Fig 4.17, I was able to measure the positions of the two wing edges reliably. Our data closely resemble earlier published data on stroke angle ([Fry et al., 2005], their Fig 4.A).

We reconstructed the sampling procedure of ten consecutive frames based on data from the image buffer saved at the end of a measurement (See Fig 4.15). The sequence shows a half a stroke cycle starting with an early down stroke (image 1) to ventral stroke reversal (image 9).

Measurements up to 120 s, or 840'000 samples, were routinely performed. A representative example of such a measurement is shown in Fig 4.18. The fly was measured as it was optically stimulated with a vertically-oscillating pattern. The sinusoidal response of the fly is clearly visible.

Table 4.2 shows the tracking statistics collected with \( n = 10 \) flies. The robustness of tracking is high with over 80% of frames in which a full wing was extracted and
4.2. DIGITAL WING BEAT ANALYZER

Fig 4.17: Tracking results for one wing with an effective sampling rate of 6250/2 Hz. Positive and negative wing edge positions are marked with circles and stars, respectively. The positive and negative edge Kalman fits used to estimate wing position is represented with a solid and dashed line, respectively.

over 99% of time spent in "wing edge tracking" mode[3]. Occasionally, tracking fails during mid-stroke, where the wings move the fastest, but it typically recovers within the same stroke as the wing slowed down toward stroke reversal. The mistracking is caused by a mismatch between the fly’s kinematics and the EKF’s \textit{a priori} model. To address this issue, I developed an offline algorithm that can adjust the EKF parameters $\alpha_i$ and $\varphi_i$ given an initial measurement set. Incomplete wing extraction occurs when both wings temporarily overlap during dorsal flip, and only one of the wing edges can be extracted. Furthermore, flies sometimes extend their legs into the wing path, which led to false wing position readings. These effects are all visible in the visual data we collect at the end of the measurement (see Fig 4.15) and can be avoided by a better calibration and an appropriate choice of the opening width in Fig 4.13.

---

[3] The initial tests required to get a proper calibration were removed from the data set. With an improper calibration, the tracking performance is poor.
Fig 4.18: Typical stroke amplitude and frequency measurements for a fly stimulated with a pattern oscillating vertically at 1 Hz. The fly responds by sinusoidally modulating its wing beat frequency and amplitudes. The measurements were first interpolated to precisely extract both stroke reversals of each wing beat. The amplitude and frequency were then extracted from these extrema. Data on the left hand side were low pass filtered with a Butterworth filter to improve visibility. The right hand side shows a one second extract of the unfiltered data. The horizontal lines represent the mean values throughout the measurement.

Table 4.2: Tracking statistics based on $N > 28 \cdot 10^6$ frames measured over $n = 10$ flies.
Proof-of-concept experiments

As a proof-of-concept for the general applicability of the process model for real-time hardware control, we used the stroke phase estimate provided by the process model to trigger a stroboscopic flash at defined times during the stroke cycle on a stroke by stroke basis. By linearly varying the phase position between successive strokes, we generated a 'slow motion' impression of wing motion that could easily be verified by observation.

To control the strobe lighting, we estimated the wing phase after each Kalman update and calculated the remaining time to the subsequent strobe flash. If this time was calculated to be less than the camera’s frame time, a digital message was sent out that contained information about the precise timing of the flash. The timing information was decoded on dedicated hardware and used to trigger a flash of 50 µs duration. This application provided a live view of apparent slow motion that revealed induced kinematic changes from stimulation, but also problems occurring during tracking.

4.2.9 Discussion

Comparison with existing technologies

In previous research, several solutions have been proposed to overcome bandwidth limitations to perform real-time high speed tracking. Analog or hybrid analog/digital vision chips provide particularly intriguing solutions as they allow processing power to be integrated at the pixel level. Not only does this allow a massive parallelization of image pre-processing, but it also offers the opportunity of transferring image information selectively to increase the information transfer rate. While these technologies promise highly effective and affordable tracking solutions, implementations using standard digital and software development technology nevertheless offer several advantages.

First, standardized digital solutions have a high degree of flexibility to adapt to specific measurement or control requirements. In our solution, the parameters can readily be tuned to different experimental situations. Such flexibility is difficult to achieve in vision chips, in which a specific task is typically built into the hardware.

Furthermore, a graphical representation of the tracking process is maintained in digital systems, in our case in form of the final 1GB of acquired image data. The

\[ \text{The number of } \mu \text{s until the actual flash was encoded by seven bits and sent through the computer’s parallel port with a total delay of about 20 } \mu \text{s} \]
availability of raw visual data allows additional offline analysis to be performed e.g. as a control for meaningful position extraction. The analysis of the tracking process in analog systems, on the other hand, can be challenging and hard to interpret.

Concept analysis

The contributions can be summarized as the combination of three key concepts that together form a novel solution to high speed tracking: 1. Dynamic region-of-interest sampling 2. Real-time image processing and 3. Online process model.

1. Dynamic region-of-interest Our implementation in digital hardware makes use of the dynamic region-of-interest capabilities offered by recent CMOS based digital video cameras. The benefit from transferring only a small part of a FOV is of general relevance to tracking applications in which the bandwidth gain is worth the extra processing time involved with the prediction and update of the ROI. This is the case if the object of interest is small compared to the global FOV.

As sensors increase in pixel array size and frame rates, the bandwidth problem will certainly become more pronounced, making dynamic region-of-interest approaches more appealing. The concept of dynamic region-of-interest need not be limited to standard CMOS cameras. A scanning electron microscope (SEM), for example, can easily be configured to scan a specific area, making it ideal for a dynamic region-of-interest approach in time-critical cases. This is especially relevant because objects at small dimensions tend to move fast with respect to their size.

2. Real-time image processing The real-time image processing is inherent to our solution because the region of interest must be updated as a function of the current wing’s position. A given image had to be analyzed before the following one was completely transferred. By analyzing each image online, only the useful information has to be stored. As a consequence, the system consumes very little memory and can therefore run almost indefinitely. This is useful for applications that must test large parameters spaces, or that are waiting for a single event to occur.

To analyze each image within a few microseconds, I benefited from some simplifying assumptions inherent to our application: the fly was tethered, allowing us to extract the paths of the wings and only analyze relevant pixels and therefore drastically reducing the computation costs. Similar simplifications can be found
in many tracking applications: higher sampling rates decrease the distance an object has moved between two frames, such that the search of an object can first concentrate on the pixels most likely to contain it, and then extend to pixels with lower likelihoods. Limiting the analysis to a small amount of pixels allows even complex signal processing computations to be performed fast.

3. **Online process model** To be able to position the region-of-interest, the position of the object must be predictable from past measurements. The prediction model is therefore a key element of our tracking approach. Here, a tradeoff must be found on the level of complexity of the model. A more complex model will predict the position of the object more accurately, therefore allowing a smaller region of interest to be chosen and increasing the frame rate of the camera. A more complex model also involves more computations, however, and will add time to the processing. The level of complexity must therefore be chosen in an iterative way to find an optimal tradeoff between transfer and processing constraints.

In this application, I opted for an Extended Kalman Filter mainly due to its recursive implementation and its ability to precisely extract a state vector at each measurement instance, therefore providing real-time analysis of the measurement time-course. Other similar techniques, such as alpha-beta-gamma filters, do not provide a real-time analysis of the data.

**Concept synthesis**

This real-time functionality is crucial to applications that require the control of an external piece of hardware based on its current process state. This is of general relevance to behavioral research paradigms in biology, because it provides controllable sensory conditions that nevertheless depend on the animal’s (intended) behavior in a naturalistic way. As robotics become more autonomous and integrated into processes, the need for rapid and robust process estimations based on non-contact sensors will also increase. For instance, production lines are likely to increase in speed and complexity. Likewise, visual servoing will be applied to faster manipulators. In these cases, the complexity and structure of the process model is completely dictated by the required feedback.

In this application, two of the Kalman filter’s state variables were directly employed for the control of external hardware. Because the difference in stroke amplitude between the right and the left wings are associated with intended turning maneuvers
of the tethered fly [Dickinson et al., 1993], we could use them to control the fly’s visual panorama in a ‘flight simulator’ [Heisenberg and Wolf, 1988].

We also used the Kalman filter’s phase variable to stroboscopically illuminate the wings or excite the mechanosensors at precisely defined wing phases.

The EKF presented in this paper can be applied to any periodic motion. The shape of the fitted curve is entirely defined a-priori and is then scaled and offset both in time and space during the tracking, providing an estimate of the current amplitude, phase, frequency and offset of the periodic signal. For non-periodic motions, the EKF can easily be adapted to, for example, a constant acceleration model that predicts the future location of the object based on its present position, velocity and acceleration.

The combination of dynamic region-of-interest, real-time image processing and online process model promises broad applications in experimental research and process control, whenever the status of a system must be rapidly evaluated as part of a control loop.

4.3 Visual flight simulator

So far, I have described tools to measure the output of the fruit fly: the wing kinematics and the flight forces. These tools can be used by their own, but to analyze how flight is actively controlled, they must be used in combination with an input to the fly. The measurements then represent the response of the fly to a known stimulation.

In tethered flight, most of the fly’s sensory systems do not receive natural feedback because the body is immobilized. The sensory input must then be simulated. Here, I present a system that provides input to the fly’s most powerful sensory modality: vision.

4.3.1 Other insect visual displays

Presenting visual patterns to tethered insects has been done for quite some time. Initial experiments used patterned cylinders that were mechanically rotated. These experiments led to important discoveries of basic sensorimotor principles in insects [Von Holst and Mittlestaedt, 1950, Reichardt, 1961, Götz, 1964]. Due to the limited flexibility of the system, researchers then developed projector based systems. Götz used two independently controlled projectors for each eye [Götz, 1968]. Hausen used pattern projectors mounted on a gimbal [Hausen, 1982]. Krapp developed a system where a small dot could be positioned at different viewing angles [Krapp and
4.3. VISUAL FLIGHT SIMULATOR

Hengstenberg, 1997]. Afterwards, several groups developed custom built LED solutions [Dickinson and Lighton, 1995, Strauss et al., 1997, Lindemann et al., 2003], that provided more flexibility in the pattern choice as well as higher sampling rates. An improved, modular, LED panel system was developed by [Reiser and Dickinson, 2008], and the design was released as an open source project. At the same time, interest in projection-based systems has gone back up in recent times because of the emergence of faster Digital Light Processing (DLP) chips and the possibility to project on spherical domes [Taylor et al., 2008].

4.3.2 Design requirements

The basic requirement is to have a spatial resolution fine enough and a temporal resolution fast enough so that the fly doesn’t see the increments. These would produce artifacts. The fly’s interommatidial angle is about 4° [Götz, 1964] and the flicker fusion rate was found to be inferior to 150 Hz [Miall, 1978]. The field of view should cover as much of the fly’s natural field of view, which is almost panoramic. The system has to produce light at wavelengths visible to fruit flies. Heisenberg et al showed that the relative response of Drosophila was highest for wavelengths between 350 and 500 nm [Heisenberg, 1977]. The system should also be able to cover as much of the fly’s luminosity range as possible. Such a measure is difficult to perform in absolute terms and it is usually considered sufficient if the behavioral response to the visual input saturates as of a certain intensity.

4.3.3 Implementation

Given the large number of previous flight simulator implementations, it would have been inefficient to design our own from zero. We chose to base our simulator on the LED system designed by Reiser [Reiser and Dickinson, 2008]. This design was the only open-source one and fulfilled the specifications described above. Furthermore, we had direct contact with M.B. Reiser, making the transfer of know-how easier.

After a few months, however, we noticed certain limitations of the system in terms of the data rate, which, in certain cases, was as low as 15 Hz. At the heart of this problem was the Two Wire Interface (TWI) that was used to communicate from a single master to a total of 96 slaves. We decided to reduce this bottleneck by implementing a novel controller that used twelve TWIs to communicate with the slaves, based on the CompactRIO device developed by National Instruments. Apart from this and a few other small improvements, the system we used was a copy of the
**Fig 4.19:** Concept of panel system. The user interacts with the computer through a LabVIEW GUI. The computer sends over to the CRIOn the spatiotemporal parameters of the experiment. During the experiment, the Real-time controller of the CRIOn generates images that are transformed into I2C signals by the FPGA. The I2C signals are addressed to individual modules, which each show a 8x8 pixel pattern.

design by Reiser.

Our system is composed of three parts: i) a computer, where the user can specify the type of experiment and the visual parameters. The computer also logs the information of different events, such as the synchronization with external hardware or the image publication characteristics. ii) a CRIOn real-time controller, which generates each image, transforms it into individual instructions for each panel and sends these instructions to the panels via an FPGA circuit. iii) the individual panel modules, each consisting in a 8x8 LED matrix and a 8bit microcontroller. The different panels are assembled together to form the display.

**GUI computer interface**

I designed a GUI where the user could easily select the general type of experiment (open-loop dynamic stimulations, static images, closed-loop stimulations), and the different visual parameters, such as the temporal and spatial frequencies of a sine grating.

Once the experimental parameters have been chosen, and the experiment has been launched, the computer sends the information over to the real-time controller. At the end of the experiment, the real-time controller returns information about the experiment, such as the generation and publication times of images. The computer logs the information into data files for offline data analysis.
Fig 4.20: LabVIEW GUI of the panel system. The user can choose the spatiotemporal parameters of an experiment, launch it and observe the measurements.
Real-time controller

The real-time controller runs in the background and the user never has to directly interact with it. It contains a CPU running VxWorks (www.windriver.com/vxworks/) and a FPGA backplane that allows to parallelize most of the low level interfacing.

The real-time controller first waits for instructions from the computer. Once it receives the experimental parameters, it starts generating the image to be shown on the panel. The image is then segmented into the 8x8 pixels of each panel module and the I2C command for that panel is generated. The I2C commands are sent out on 12 parallel lines - one for each vertical row of panels. This significantly increases the speed of the system compared to the original design by Reiser. The real-time controller receives the image publication time stamps, and passes this information back to the computer. The real-time controller also logs the force data from the single axis force sensor and synchronizes itself with the Digital Wing Beat analyzer, so that the timestamps of both systems are directly comparable.

LED modules

The LED modules consist in a microcontroller that is programmed to receive the I2C commands and transform them into control signals to the LED, via a set of Darlington transistors. To achieve 8 different greenscale values, the LEDs are refreshed at
4.3. VISUAL FLIGHT SIMULATOR


approximately 2600 Hz in a cycle of 8 ON/OFF events, e.g. a pixel set to a greenscale of 4 will be on during 4 of the 8 cycles. The worst-case refresh rate, 372 Hz, is well above the flicker fusion rate recorded for fruit flies [Miall, 1978].

Completed arenas

Together with Mathias Moser, Vasco Medici and Jan Bartussek, we built two arenas, the flat and the circular arena. The flat arena used 96 individual LED modules and was designed to emulate conditions in the free-flight wind tunnel. The circular arena used 60 individual LED modules and was designed to cover the entire panoramic range of the fly. The code that I wrote to control the arenas was organized on a SVN server and can easily be accessed.

Performances

By parallelizing the I2C lines, I was able to increase the performance of the system. It achieves 50 Hz with a non-symmetrical, grayscale image, 150 Hz with non-symmetrical black/white images and over 250 Hz for images with vertical symmetry.
Fig 4.23: A: Circular arena B: planar arena
Chapter  5

Reverse-engineering micro-aerial flight in biology

I applied the technologies described in the previous chapter to a set of different biological problems.

Table  5.1:  Overview of the different projects and their dependency on the developed technologies

<table>
<thead>
<tr>
<th></th>
<th>DWBA</th>
<th>MEMS 1 DOF</th>
<th>MEMS 3 DOF</th>
<th>Planar LEDs</th>
<th>Circular LEDs</th>
<th>Piezo stim.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Biomechanics of tethered flight, §5.1</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td>(X)</td>
<td></td>
</tr>
<tr>
<td>2) Lift control, §5.2</td>
<td>(X)</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3) Thrust control, §5.3</td>
<td>(X)</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4) Role of mechanosensors [Bartussek et al., prep]</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>5) Exploration of bio/robot coupling, §C</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

1. The biomechanical analysis of tethered flight employed the instantaneous wing kinematics and wing forces to experimentally validate an aerodynamic model of force production. This model was developed using dynamically-scaled robotic
models of the wing but had never been compared to direct force measurements (see Section 5.1 page 56).

2. The study of lift control analyzed the equilibrium reflexes of flies for lift (altitude) control. In this study, the fly’s control response was characterized using system identification. This analysis provides a detailed, quantitative analysis of the fly’s reactions to changes in perceived altitude (see Section 5.2 page 60).

3. The study of thrust (forward velocity) control aimed at comparing the tethered data with free-flight experiments performed in a wind tunnel. It employed the 3DOF force sensor to measure the concurrent lift and thrust forces and pitch torque (see Section 5.3 page 69).

4. The role of mechanosensors was studied by Jan Bartussek in a collaboration that used the tethered setup. A first experiment analyzed the phase coherence of flies with and without ablated wing veins. In a second experiment, flies were stimulated with a piezo actuator to simulate Coriolis forces. The fly’s kinematic response was recorded using the digital wing beat analyzer [Bartussek et al., prep].

5. In a side experiments, the bio/robot coupling experiments explored the transfer of a biological control strategy into a robotic implementation. To this end, the fly’s output were coupled with the steering commands of a two wheeled robot, and the robot’s sensor data was fed back to the fly visually (see Section C).

Taken together, these applications show the breadth of biological applications made possible by the developed technologies. This list corresponds to the initial projects that could be undertaken within the time frame of this thesis. Novel types of experiments are currently planned as parts of collaborations and continuations. I will now go through each of the topics in more detail.

### 5.1 Biomechanics of tethered flight

In the past decade, the increasing interest of roboticists to build artificial microflappers as well as the availability of novel tools has brought a lot of interest to the lift generation mechanisms of small flying insects. These studies mainly used the dynamically scaled robotic wing approach and CFD simulations (see 2.1.3). They identified a set of models of how small insects generate sufficient lift to stay aloft
5.1. BIOMECHANICS OF TETHERED FLIGHT

Dickinson et al., 1999, Sane and Dickinson, 2001, Sane and Dickinson, 2002, Birch and Dickinson, 2001]. However, an experimental validation of these principles using direct force measurements was never performed. We therefore set off to compare the direct measurements with the model predictions, together with Vinzenz Schönfelder, who did his Master thesis under Steven Fry and my supervision.

To validate the model, we concurrently measured flight forces with the MEMS force sensor, and wing kinematics with the DWBA. The wing kinematics extracted from the DWBA were applied to the ”Revised quasi-steady model” which is the most detailed model available at the moment (see [Sane and Dickinson, 2002] and Section 2.1.3). We then compared the model’s predictions with the measured forces.

Revised quasi-steady model

Combining aerodynamic theory and empirical results from a robotic wing model, Sane and Dickinson proposed a revised quasi-steady model [Sane, 2003, Sane and Dickinson, 2001]. Besides steady-state forces, the model additionally included dynamic effects resulting from wing acceleration and rotation. Skin friction is neglected.

The revised quasi-steady model represents aerodynamic forces as the sum of three independent components:

1. **Translational force** corresponds to the standard quasi-steady model and is equivalent to the steady-state forces on airfoils.

2. **Rotational force** derives from the Magnus effect. [Dickinson et al., 1999]. The force emerges when a translating wing is rotating about its span-wise axis.

3. **Added (or virtual) mass force** arises from the inertia of the surrounding air [Ellington, 1984b]. It only acts when the wing accelerates (dragging the surrounding air with it) or rotates (which also results in an acceleration of the surrounding air).

A major shortcoming of the revised quasi steady model is that it only implicitly depends on time, i.e. it does not take into account the fluid flow pattern. Thus, it neglects the aerodynamic forces resulting from wake capture, which is considered a significant component of force production in *Drosophila* [Dickinson et al., 1999, Sane and Dickinson, 2002, Wang et al., 2004]. The details of the model implementation are described in Chapter B.
Fig 5.1: Comparison of modeled vs measured lift forces. The model details are described in Chapter B. The force data represents the average of $n = 100$ wingbeats measured on 1 fly.
5.1.1 Results

The modeled and measured forces were quite similar (see Fig 5.1). The magnitude of the forces are the same and the mid-stroke components match quite well. At stroke reversals, the modeled and measured forces differ. Here is a list of possible causes, with the most probable first:

- The revised quasi-steady model provides only a simplified estimation of aerodynamic forces. We had to omit the added mass component, we also neglected the inertial effects of the air surrounding the wing (see Section B.3). There was, however, no alternative, since this has been the most precise model published from the dynamically scaled robotic wing experiments.

- The wing beat analyzer can only provide the position of the wing in the camera plane. We extrapolated the full three dimensional movement of the wing by superposing the measured data with previously published wing kinematics [Fry et al., 2005]. This extrapolation obviously affects the precision of the model.

- All constants characterizing the insect’s morphology were not directly determined, but assumed to range in standard values found in the literature (see Section B.1). Wing mass, surface area and other morphological parameters all influence the instantaneous flight force production. Estimating these values by trying to improve the agreement between modeled and measured variables is not practical) due to the high-dimensionality of the parameter space and general uncertainties in the model. It is also not very desirable to fit the model to the measurements since the purpose of the experiment is to check if the two match.

- A direct instantaneous force measurement might contain artifacts due to other moving body parts. At least the muscles and halteres are known to move during flight, but also abdomen and head are not rigidly connected to the thorax. These organs have much larger masses than the wings. The model only accounts for aerodynamic and inertial effects of the wings.

Some of these issues are not easy to solve with the current setup. Nonetheless, these are the first attempts to directly compare modeled forces derived from a dynamically scaled robotic wing and instantaneous forces on the tethered insect. The similarity of the forces remains remarkable.
5.2 Lift control

Lift represents an important aspect of low-level flight control since it corresponds to the main component of flight force and must be continuously adjusted during flight maneuvers to compensate gravity. In this section, I apply a systems approach to reverse-engineering lift control.

A systems approach has the advantage of avoiding complexities arising from the low-level processes by treating the fly as a biological black box, where input-output relationships are described with a minimally-constrained model [Taylor et al., 2008].

Lift was chosen because it is one of the most suitable experimental paradigms for the tethered setup: Lift control is independent of the disruption of mechanosensor feedback in tethered experiments, because the fly, unlike airplanes or birds, does not rotate during lift adjustments (see the helicopter model by [Götz, 1964]).

Previous research on lift in fruit flies has mainly concentrated on the aerodynamic effects taking place to generate sufficient force. [Lehmann and Dickinson, 1997] correlated the changes in wing beat frequency and amplitude to the mean lift force. In a further study, they looked at the ability of flies to vary wing beat frequency and amplitude as a function of lift [Lehmann, 2001]. Fewer studies have looked at the control aspects of lift. In locusts, [Taylor and Thomas, 2003, Taylor and Zbikowski, 2005] studied flight stability by analyzing the state space model of how the locust would have flown had it been instantly released from its tether.

A rigorous approach inspired from aeronautical engineering was employed by [Tanaka and Kawachi, 2006] to study bumblebee flight control. Patterns were oscillated vertically at various frequencies and the responses were plotted in a Bode diagram. The frequency response was compared to frequency responses of other lift controllers.

Here, I extend this approach to a more detailed analysis of the fruit fly.

5.2.1 Methods

Fruit flies were tethered to a MEMS micro force sensors (see Section 4.1) and placed under a high speed camera (see Section 4.2) in a visual flight simulator (see Section 4.3). The flies’ reaction to vertically-oscillating patterns were recorded by measuring their lift forces and their kinematics. The oscillation frequency of the patterns was varied throughout the experiments.
Fig 5.2: Experimental setup and raw data. A: Wing kinematics extracted at 7000 fps. B: Lift forces measured by MEMS. C: Visual pattern position. The pattern is oscillated up and down (here at 0.8 Hz) D: Wing beat amplitude, extracted from instantaneous measurements. E: Wing beat frequency extracted from instantaneous measurements. F: Averaged lift forces. The forces were averaged for visualization purpose to remove the cyclic stroke components. Note the three orders of magnitude of time scales difference between the instantaneous (left) and stroke averaged (right) plots.
Visual patterns

Two types of patterns were used in the experiments. The first consisted of a sinus grating that oscillated vertically: the "greenscale" intensity $g$ of a LED located at $x, y$ at time $t$ is described as:

$$g(x, y, t) = \frac{C}{2} \left( \sin(2\pi SF(y + \frac{A_{in}}{2} \sin(2\pi TFt))) + 1 \right)$$

(5.1)

where $C$ is the contrast amplitude, SF is the spatial frequency [pixel$^{-1}$], TF is the temporal frequency [Hz] and $A_{in}$ is the oscillation amplitude [pixel].

The second visual pattern consisted of a sinusoid grating that underwent a step change in velocity:

$$g(x, y, t) = \frac{C}{2} \left( \sin (2\pi \cdot SF \cdot y + \phi_{step}(t)) + 1 \right)$$

(5.2)

$$\phi_{step}(t) = \begin{cases} 0 & \text{if } t < t_0 \\ 2\pi TF (t - t_0) & \text{if } t \geq t_0 \end{cases}$$

(5.3)

where $t_0$ is the step onset time. To calculate the position time course of a pattern, one must follow how a given phase in the grating moves through time.

Sinus gratings were chosen because they represent a single spatial frequency and therefore represent the most basic input to the visual system.

5.2.2 Experiments

We first characterized how the fly responded to vertically-oscillating patterns. For each measurement, the visual pattern properties, as defined in Eq 5.1, were randomly chosen in the following parameter space:

$$
\begin{align*}
TF & \in [0.1, 7.0] \text{ Hz} \\
A_{in} & \in [20, 75] \text{ mm} \\
SF & = 21 \text{ m}^{-1} \\
C & = 7
\end{align*}
$$

(5.4)

The visual pattern was presented to the fly for an interval of 20 s while the kinematics and forces were recorded. A typical measurement sample is shown in Fig 5.2.E-H. The average force response extracted from Fig 5.2.F-H is shown in Fig 5.5.A. To facilitate the extraction of a controller, the force response was then converted to the pseudo-position of a free-flying fly. The position corresponds to the altitude of a free flying fly producing the same force pattern. This conversion is used so that the
5.2. LIFT CONTROL

Fig 5.3: Average response to an oscillation (in this figure, TF = 0.8 Hz, n=25 oscillations). **A:** Segmented force data. The force response to each visual pattern oscillation is overlayed. **B:** Input and output positions. Each individual force response is numerically integrated to estimate the virtual positions of the fly. The virtual position of the fly is then put in relation to the visual pattern position by comparing the amplitudes $A_{\text{in}}, A_{\text{out}}$ and phase difference $\Delta \phi$ (shown here as the phase difference between maxima).

Input and output units are the same. To calculate the pseudo-position, the force is numerically integrated in an inertial newtonian model:

$$z = \frac{1}{m} \iint F(t) \, dt^2$$

where $z$ is the pseudo-position, $m$ is the mass of the fly kg and $F$ is the resultant force measured at the sensor tip ($F = F_{\text{fly}} - m \cdot g$). A similar methodology was employed by [Tanaka and Kawachi, 2006] and this overall strategy is discussed in section 5.2.4.

Fig 5.5.B shows the resulting pseudo-position pattern in comparison with the input stripe position.

5.2.3 Results

Given a sinusoidal stimulation, we expect the fly to respond with a periodic force trace. Indeed the response is approximately a sine of the same frequency as the input sine (Fig 5.2.E-H and 5.5.A). The double integration from Eq 5.5 causes the sine response to become even more clear (Fig 5.5.B). The amplitudes $A_{\text{in}}, A_{\text{out}}$ and phases $\phi_{\text{in}}, \phi_{\text{out}}$ are therefore sufficient to fully characterize the response.
Fig 5.4: Bode diagram of the fly’s lift controller (input: visual pattern position, output: pseudo-position of fly - see Fig 5.5). A: amplitude gain (Lin-log). B: amplitude gain (Log-log) C: phase difference. Each red point represents a single oscillation. Data for ca. 8000 visual pattern oscillations measured on n = 6 flies. Blue line shows directional mean ± directional S.D.. Orange line shows the extracted model $K(s)$. Green – – line shows bumblebee response (from [Tanaka and Kawachi, 2006]) and purple –·– line shows human pilot response (from [McRuer and Jex, 1967])
5.2. LIFT CONTROL

The gain $G = 20 \cdot \log(A_{\text{out}}/A_{\text{in}})$ and phase difference $\Delta \phi = \phi_{\text{out}} - \phi_{\text{in}}$ responses measured at different TF frequencies are then plotted in a Bode diagram (Fig 5.4). The gain response decreases from 40 dB at low frequencies with a slope of about forty decibels per decade. This is typical for a controller with a second order denominator. The variance of the gain is very low, which is remarkable for biological measurements.

The phase difference starts at $-100$ degrees and shows a linear decrease to ca. $-270$ degrees at 42 rad/s. This decay is expected for a system with a fixed time lag. The variance increases with frequency. To interpret the measured phase lags, we must consider the fact that insects react to pattern velocities, not position. In a sinusoidal stimulation, the velocity peak precedes the position peak by 90 degrees. Secondly, the flies reaction is a change in force output, which I integrated twice to get a position. The double integration (Eq 5.5) causes a $-180$ degree phase shift (compare Fig 5.5A and 5.5B). The ca. $-100$ degree phase difference measured at low frequencies in our position input/output diagram therefore corresponds to a 10 degree phase lag between velocity input and force output. The choice of input and output dimensions is discussed in detail below.

We used the Frequency Domain Identification Toolbox available for Matlab to find an appropriate model for the data. We considered models with one to four zeros, one to four poles and with or without a time delay. We fit the 32 resulting models to the Bode measurements using the toolbox. About a half-dozen models had small cost functions. We selected the following one, because it maintained a small cost function even though it had among the lowest orders:

$$K(s) = 5.24 \cdot 10^{-5}s^2 - 2.74 \cdot 10^{-5}s + 0.078 \over 0.0021s^2 + 0.00658s + 0.00513 \cdot e^{-0.03s}$$ (5.6)

This model has two zeros, two poles and a 30 ms time delay. The model fits well for the majority of the measurements, but diverges at both frequency extremes (see Fig 5.4). The good fit strengthens the hypothesis that the 40 dB/decade decay in the fly’s lift response can be modeled as a second order denominator in the transfer function. The 30 ms time delay is in the same order of magnitude as visual delays measured in houseflies [Borst, 1986].

We also complement our frequency measurements with a set of velocity step response measurements (Eq 5.2). The flies reacts with step increase in force, which, once converted to position, can be compared with the estimation of the extracted model (see Fig 5.5). This provides a rapid method to verify that the extracted model $K(s)$ is valid for a different type of input.
5.2.4 Discussion

In this chapter, I present a novel approach to reverse-engineering lift response in fruit flies. To this end, I employed state-of-the-art microrobotic tools to characterize the flight dynamics of lift responses. We analyzed these measurements in a control theory framework, where the fly was considered as a biological black box, with known inputs and measurable outputs. We presented the Bode diagram of the fly’s lift controller and compare it to lift controllers of other species and other man-made devices.

Analyzing the fly as a controller

By analyzing the fly from a control system’s perspective, we benefit from several aspects. The analytical framework that has been set up for man-made controllers is readily available. The field of system identification helps us identify models and parameters from the raw data. The extracted system can easily be characterized in terms of stability and robustness. Such results can directly be compared with measurements performed with other species or robots, therefore making it easier to transfer knowledge between the disparate fields of robotics and biology.

To benefit from these advantages, I had to quantify the input and output with the same units by deriving the pseudo-position of the fly from the force pattern measured with the MEMS. This was done to put the measurements in the framework of a controller, where a desired goal is fed in and the achieved output is measured. Such systems typically work in a feedback loop, where the desired and achieved are compared to generate an error signal for the controller. In this case, I was studying the open-loop characteristics of the system.

The systems approach also contains certain pitfalls. The first is to become too reductionist and forget that we are dealing with a biological entity. The biological complexity underlying the response in Fig 5.4 has much more to it than the extracted model \( K(s) \) would seem to contain. The sensorimotor pathway is highly-interlinked and it is unlikely that the visual and motor control can be easily separated.

Another pitfall lies in the way we analyze the fly’s controller. For instance, the use of the pattern’s metric feature size in the Bode diagram is dangerous, because flies do not perceive metric dimensions but angular ones. In a wider flight arena, a pattern of the same metric size would be perceived at a smaller angle. In other words, the offset of the gain magnitude in Fig 5.4 is flight-arena-dependent. In a larger flight arena, the gains would be offset to the bottom.

Interestingly, this fact does not invalidate our gain magnitude data, because the fly (and any other vision-based flyer) must also cope with this difference in the real
Fig 5.5: Lift response to a step increase in pattern velocity. **A:** After three seconds, the pattern starts moving upwards at a constant velocity. **B:** The fly reacts by adjusting its mean force. The forces were low-passed with a zero-phase filter to remove the wing beat oscillations. **C:** By integrating the force, the fly’s position is calculated and compared to the model $K(s)$ extracted from the Bode diagram.

Given the knowledge that flies react primarily to pattern velocities, one could ask why the input/output dimensions were not velocity. We chose position as an input/output dimension to facilitate the comparison with frequency response plots measured previously. Also, the Bode diagram for a velocity input/output (or any other time derivative of position) would be the same as Fig 5.4 as long as the measured
response remained sinusoidal, because magnitude gains and phase differences are preserved in the differentiation of sinusoids.

**Comparison with other lift controllers**

We can compare the extracted frequency response to frequency responses measured on other flying systems. As discussed above, the dependency of the gain magnitude on the setup size limits how much comparison can be made directly. However, the fitted models contain interesting similarities that are independent of setup size. For human controlled aerial-vehicles, the measurements pointed to a general first order system \( \frac{4}{s} \cdot e^{-\tau} \) [McRuer and Jex, 1967]. In bumblebees, a second order model fitted the measurements best (however, measurements were sparse [Tanaka and Kawachi, 2006]). These systems all had time delays that went from 200 ms for the bumblebee to over a second for humans. Smaller flying organisms face inherently less stable flight, because the instantaneous forces they generate become larger compared to their body masses. Similarly to highly maneuverable (and thus instable) fighter jets that require the help of rapid low-level controllers for humans to pilot them, small flying organisms are likely to have developed faster controllers to cope with their instable dynamics.

**5.2.5 Conclusion**

These experiments bring up a whole new set of relevant questions. For instance, we have assumed a linear time-invariant controller in the fruit fly. It would be interesting to measure the response to a superposition of two input (temporal) frequencies, to verify the superposition property expected from a LTI system. Closed-loop experiments could confirm the controller extracted here, and tackle the difficult question of feedback within the biological system itself. Finally, the extraction of spatial dimensions from optic flow is highly relevant to any vision based flying object. A broader analysis of step responses to different pattern sizes and speeds will help characterize this.

In summary, I have presented a rigorous approach to reverse-engineering micro-aerial flight control in biology. This approach generalizes the biological system so that it is directly comparable to artificial entities. Furthermore, this work shows how engineering can provide both strong technical tools and an analytical framework for the understanding of biological sensorimotor pathways, a task that is highly relevant to the design of artificial devices at similar dynamic scales.
5.3 Thrust control

The control of thrust in fruit flies is an important modality, because the fly spends most of its time flying longitudinally. We also decided to analyze thrust because a complementary experiment in the free flight setup, where flies fly in a wind tunnel, analyzes longitudinal flight control. The reproduction of a similar experiment in two different setups allows to compare the detailed measurements of the tethered setup to the naturalistic conditions in the free fly flight setup. In particular, the effect of the tether can be analyzed.

5.3.1 Methods

The experiments consisted in using the three-axis force sensor (see Section 4.1 and Chapter 4) to concurrently measure lift, thrust and pitch torque (see Fig 5.6). The fly was placed in the flight arena (see Section 4.3). The visual pattern shown to the fly consisted in vertical sine gratings. The sine gratings were initially immobile, but after a few seconds, they underwent a step increase in horizontal velocity, similarly to what was described for the step experiments in the lift paradigm (see Eq 5.2). We varied the spatial frequency of the sine gratings, as well as the temporal frequency of the velocity step.
CHAPTER 5. REVERSE-ENGINEERING FLIGHT CONTROL

5.3.2 Results

The flies typically responded with a step increase in lift, thrust and pitching torque (see Fig. 5.7). We therefore used the mean to quantify the responses to different visual parameters. Since I wanted to compare these results with the results in the wind tunnel, I divided the mean force response by the mass of the fly, and therefore obtained the acceleration the fly would have experienced had it not been tethered.\(^1\) The resulting ”pseudo-acceleration” is then color-plotted against TF and SF (see Fig. 5.8).

5.3.3 Discussion and conclusion

The data measured in the tethered setup does not give as clean of results as what was measured in free flight [Rohrseitz and Fry, ]. We believe the main cause of this is that the tether disrupts the normal pitch response of the fly, and I therefore do not get a very naturalistic response: without any feedback that it has pitched forward, the fly may be acting erratically to achieve this first goal.

Another potential cause lies in the way the fly was visually stimulated. For technical reasons, the experiments were all performed on the circular arena, with half the pixels showing the left eye pattern and half the pixels showing the right eye pattern. This does not represent very well what the fly sees in naturalistic conditions: there

\(^1\)This calculation is based on a purely inertial model of the fly, and therefore neglects drag.
5.3. THRUST CONTROL

**Fig 5.8:** Overview of the thrust experiments. **A:** Thrust pseudo-acceleration ($\Delta F_{\text{thrust}}/m_{\text{fly}}$ where $\Delta F_{\text{thrust}}$ is the mean difference in thrust force before and after the step occurs.) **B:** Lift pseudo-acceleration **C:** Pitch pseudo-acceleration. The '*' represent the points in the parameter space that were actually measured. The points in between '*' were interpolated. Definitions of SF and TF as in Eq 5.2.
is no distortion of elements far away and the expansion/contraction created in front and behind the fly might cause an escape/fixation response instead of the intended speed adjustment response. To overcome this problem, the easiest is to perform the experiments using the planar panels or even better, mounting the tethered fly inside the wind tunnel.

In conclusion, these experiments are more of a proof-of-principle of the thrust paradigm in tethered flight. The experiments can be extended to analyze the biomechanics of thrust: by measuring in the wind tunnel, the fly’s response to varying wing speeds can be analyzed in detail and compared with the fly’s visual response. Götz’s influential helicopter model [Götz, 1964] can also be put under scrutiny with these methods. Our preliminary results seem to go against his theory, however, more experiments are needed to make an unequivocal statement.
Chapter 6

Closing remarks

This thesis has explored the exciting interaction between biology and robotics. A set of novel technical tools were developed: a specifically-designed MEMS sensor, fabricated by IRIS, was calibrated and implemented in the experimental setup to measure the instantaneous flight forces of tethered fruit flies in up to three concurrent directions. A high speed vision system was developed to extract, in real time, the kinematics of both wings. An LED panel system was fabricated to generate appropriate visual stimuli.

These tools were applied in a reverse-engineering approach to identify low-level control strategies in fruit flies. A first experiment looked at the biomechanics of tethered flight, and compared the instantaneous force measurements with the predictions of a model fed with the fly’s measured kinematics. The second experiment used a systems approach and analyzed lift control. A bode diagram of the fly’s lift control was extracted and compared to other species. In the third experiment, the fly’s longitudinal speed control was analyzed by measuring the lift, thrust and pitching torque produced by the fly in response to a visual stimulation in forward speed. In a side project, the fly’s controller was used to drive a robot in closed-loop, allowing us to explore the coupling characteristics of both.

6.1 Contributions

Through the development and application of new engineering technologies to a biological problem, this thesis has made significant contributions to both fields.

For engineering, the contributions lie in the extension of measurement techniques into previously unreachable spatiotemporal scales: smaller, faster forces for
the MEMS, high image velocities for the DWBA and rapid control of visual stimuli for the flight simulator. Through the application of these technologies to a scientific problem, they went through the stringent questioning phase of the scientific method: each possible measurement artefact had to be considered and ruled out before a hypothesis could be validated or not. The amount of effort this represents compared to initial proof-of-principle experiments (pure technology with very little science) should not be underestimated. Unlike the linear development/application structure of this thesis, there was no such thing as a development phase followed by an application phase; both are intertwined. The technologies have gone through numerous iterations to reach a product level. At each iteration, biological measurements were performed and questions were asked that led to new implementation rules.

For biology, the benefits lie in the possibilities offered by the technology. Often, technical innovations do not just lead to better measurements for a pre-existing paradigm, but rather new possibilities create a different way of thinking that replaces the old one [Kuhn, 1962]. Technical innovations lie at the heart of new biological discoveries that were not even searched for in the first place. One of these paradigms that is being solidified here is the concept of linking control theory with a systems approach. In such an approach, the fly’s internal complexities are bundled up into a black box model and characterized quantitatively. The analysis is thus simplified as it focuses on the sensory inputs and behavioral outputs of the organism, ignoring for the most part the details of what is going on in between. Through specific experimental paradigm (choice of stimulation, use of genetic tools), a particular part of the black box can be identified, thus preventing the risk of being too general in the analysis. Another experimental paradigm rendered possible, or even imaginable, through the technologies developed in this thesis is the analysis of mechanosensors in tethered flight: the real-time capabilities of the DWBA are put to use to generate wing-phase-locked forces to the halteres, thus creating pseudo-Coriolis forces that simulate a body rotation.

6.2 Outlook

It is likely that the interest in small flying organisms will increase in coming years, as the technologies to analyze them and to build similarly-scaled artificial flyers becomes available. The contributions of this thesis will therefore serve future work in understanding important aspects of insect flight.

More pragmatically, genetics is likely to expand the horizon of possibilities. The ability of genetic tools to isolate a particular process in the sensorimotor pathway
combined with the technologies and the rigorous, quantitative, systems approach shown here, will be very powerful.

*Time is fun when you’re having flies* - Kermit the frog
Appendix A

MEMS calibration

A.1 Static calibration of three-axis sensors

As mentioned in 4.1.5, the calibration of the three-axis sensors involves using a single-axis reference sensor and measuring the force/force plots in different directions.

A.1.1 Calibration procedure

The two sensors are brought in contact under a microscope (see Fig A.1). The two electrical readouts are synchronized so that the forces can be measured concurrently on each (see Fig A.2).

Given that there are no reference 3DOF sensors at the range and resolution used here, the calibration is performed using a single axis reference sensor and the procedure is repeated in three configurations (see Fig A.3). In each of the three calibration positions, static measurements of 5000 data points are recorded. Between the measurements, the force applied is increased stepwise until it reaches the maximum value of 500 µN for the 3DOF sensor in direction X or 600 µN for the 3DOF sensor in Y direction, and is then decreased again.

The data from the four capacity readouts (one is not used for the 3DOF sensor) is averaged and collected with the data from the other increments (see Fig A.4 for an example of calibration data).
**Fig A.1:** 3DOF calibration setup: the 3DOF sensor is brought in contact with a reference 1DOF sensor. Forces on each are recorded simultaneously. The operation is repeated in different directions.

**Fig A.2:** Connection diagram for the multi-axis calibration.

**Fig A.3:** The three calibration steps: A) x calibration B) y calibration C) yy calibration.
A.1. Extract of calibration matrix

The further calculation is based on Simon Muntwyler’s Master thesis [Muntwyler, 2006]. The slopes of the first order least square fits in the plots in Fig A.4 are actually the entries of the matrix in Eq A.1. C2, C3 and C4 are the measured differential capacities, while Rx, Ry and Ryy represent the force values measured by the reference sensor in the three calibration positions. The MEMS sensor characteristic curve is supposed to be linear and the slopes of C2, C3 and C4 are approximated with first order fits. Fig A.4 shows that this is a good assumption for reference forces up to 300 µN.

\[
\begin{pmatrix}
C2 \\
C3 \\
C4
\end{pmatrix} =
\begin{pmatrix}
dC2 \\
dC3 \\
dC4
\end{pmatrix}
\begin{pmatrix}
dRx \\
dRy \\
Ryy
\end{pmatrix}
\begin{pmatrix}
Rx \\
Ry \\
Ryy
\end{pmatrix}
\]  
(A.1)

Eq A.1 is then numerically inverted. Eq A.2 shows the values of the inverted matrix as they were calculated for sensor C.

\[
\begin{pmatrix}
Rx \\
Ry \\
Ryy
\end{pmatrix} =
\begin{pmatrix}
0.0018 & -0.0093 & -0.0397 \\
-0.0267 & 0.0838 & 0.0004 \\
0.1013 & -0.0927 & 0.0007
\end{pmatrix}
\begin{pmatrix}
C2 \\
C3 \\
C4
\end{pmatrix}
\]  
(A.2)

Now, to convert the values Rx, Ry and Ryy to the actual forces and torques acting on the sensor, a simple matrix multiplication is performed:

\[
\begin{pmatrix}
F_x \\
F_y \\
M_z
\end{pmatrix} =
\begin{pmatrix}
1 & 0 & 0 \\
0 & 1 & 1 \\
0 & 0 & 2
\end{pmatrix}
\begin{pmatrix}
Rx \\
Ry \\
Ryy
\end{pmatrix}
\]  
(A.3)

Finally, combining Eq A.2 and Eq A.3 leads to the 3x3 calibration matrix shown in equation A.4 with the values calculated for sensor C.

\[
\begin{pmatrix}
F_x \\
F_y \\
M_z
\end{pmatrix} =
\begin{pmatrix}
0.0018 & -0.0093 & -0.0397 \\
0.0747 & -0.0089 & 0.0011 \\
0.2027 & -0.1853 & 0.0013
\end{pmatrix}
\begin{pmatrix}
C2 \\
C3 \\
C4
\end{pmatrix}
\]  
(A.4)
**Fig A.4:** Data from a full calibration. The data points marked ‘x’ are used in the calibration matrix extraction. The data marked ‘O’ are disregarded because they either are above the calibration range, or their standard deviation was too high.
A.2 Dynamic considerations in displacement/force sensors

A.2.1 Introduction

As mentioned in [4.1.5], the MEMS sensor measures forces indirectly: A force causes a deflection of the sensor probe. For static and low-frequent signals, this relationship behaves proportionally. However, this proportionality gets lost at high frequencies where dynamic effects take place.

Sensor characteristics

The sensors use a differential triplate comb drive configuration, as seen in Fig A.5. The two stationary comb drives were electrically isolated by placing the capacitor plates on each side of the moveable comb set.

Fig A.5: A: Schematic illustration of the micro force sensor. The two stationary comb drives are situated on each side of the moveable comb set (adapted from [Sun et al., 2005]). B: MS3110 MEMS capacitive readout circuit. A change in the capacity causes a change in the output voltage (adapted from www.irvine-sensors.com/pdf/MS3110DatasheetUSE.pdf).

Because of the differential configuration of the capacitive pair, the relationship between deflection and output voltage is quasi-linear. The force applied to the probe causes a deflection which changes the electrical capacitance of both capacities and finally the output voltage of the MS3110 Universal Capacitive Readout IC (Irvine Sensors Corporation). The functional block diagram of this readout chip is illustrated in Fig A.5B.
The limits of static calibration

At low frequencies, force and displacement can be assumed to be proportional. Since most users of these force sensors are interested in mostly static forces, FemtoTools performs a static calibration, where force and voltage are proportional. The S540 sensors have a gain of about 80 \( \mu \text{N/V} \).

When measuring rapidly-varying forces, such as the forces produced by the wing beat of a fruit fly, force and deflection are not proportional anymore. This happens as the frequency content of the input forces approaches the eigenfrequency of the MEMS’ mechanical structure. The oscillations of the MEMS’ movable structure cause amplitude and phase shifts between the input force and the resulting deflections. Given that the resonant frequency of the sensor without any extra mass added to the probe is around 2 kHz, we expect significant effects of the vibrations to already appear after several hundred Hertz, especially in the presence of an added mass such as the fly which brings the resonance frequency lower. This limit is quantified at the end of the section, see Fig A.13.

The difference between the static and a dynamic calibration can be seen in Table 4.1. The electronic readout transfer behavior is defined by the MS3110 MEMS readout chip of Irvine Sensors Corporation (www.irvine-sensors.com) and is assumed to be constant for all frequencies and sensors.

The goal of this chapter is to first gain sufficient theoretical knowledge and then experimentally identify \( f^{-1} \), the inverted frequency response of the sensor that will allow us to reconstruct the original force pattern that generated the recorded voltage signal.

A.2.2 Theoretical Modeling

The goal of this section is to identify a proper model of the force sensor so that we can calculate an estimate \( \hat{F}(t) \) of the original forces \( F(t) \) from the measured voltage pattern \( V(t) \) (see Fig A.6).

To determine this model, we split the system ”sensor” up into two subsystems: the mechanical and the electrical subsystems. The mechanical system \( G \) models how external forces displace the comb structure. The electrical system \( h(x) \) models how a displacement of the comb structure
affects the output voltage. To reconstruct the original force from the measured voltage, we must invert these two models.

**Fig A.6:** Overview of the dynamic calibration process. The force source (upper left) produces a force $F(t)$ that generates a displacement $x(t)$ on the force sensor. The relationship is characterized by the mechanical model $G$. The displacement further causes a change in capacitance which is picked up by the electrical readout and generates a voltage $V$. This transduction is characterized by the function $h(x)$. The modeling effort here tries to identify the inverse models $\hat{h}^{-1}$ and $\hat{G}^{-1}$ that allow to reconstruct the force $\hat{F}(t)$ from the measured voltage $V$. The $\hat{}$ denote estimated values.

**Model of the mechanical system**

As we can see in Fig A.5, the moveable part of the sensor is fixed with four springs. To characterize the behavior of this structure, we started with the commonly used mass-spring-damper model. As we will show in the following section, this model turned out to fit very well our experimental measurements and so we never had to search for alternative models.

Using Newton’s second law of motion, we can write the ordinary differential equation of a mass-spring-damper system as:

$$m\ddot{x} + b\dot{x} + kx = F(t) \quad (A.5)$$

where $k$ is the spring constant [N/m], $b$ is the damping constant [Ns/m] and $m$ is the mass of the sensor [kg]. To derive the transfer function
between the force $F(t)$ as input and the deflection $x(t)$ as output, we take the Laplace transform of Eq [A.5]. This yields:

$$m(s^2X(s) - sx(0) - x(0)) + b \cdot sX(s) - x(0) + k \cdot X(s) = F(s)$$

Taking $x(0) = x(0) = 0$ as initial condition yields:

$$ms^2X(s) + bsX(s) + kX(s) = F(s) \quad \text{(A.6)}$$

Which leads to:

$$G(s) = \frac{X(s)}{F(s)} = \frac{1/m}{s^2 + 2\delta \omega_0 s + \omega_0^2} \quad \text{(A.7)}$$

where $\delta = b/(2\sqrt{k \cdot m})$ is the damping coefficient and $\omega_0 = \sqrt{k/m}$ is the eigenfrequency. This transfer function can then be inverted in the reconstruction to obtain a force $F$ from a deflection $x$: $\hat{F}(s) = G^{-1}(s)\hat{X}(s)$.

**Model of the electronic readout system**

The force applied to the force sensor causes a deflection of the capacitors, which induces an output voltage of the readout circuit. The characteristics of this conversion is described in the datasheet of the MS3110 Universal capacitive readout IC by Irvine Sensors Corporation (www.irvinesensors.com):

$$V(t) = \text{Gain} \cdot V_{2P25} \cdot 1.14 \cdot \frac{C_{S2}(t) - C_{S1}(t)}{C_F} + V_{\text{ref}} \quad \text{(A.8)}$$

where $V(t)$ is the output Voltage and $V_{\text{ref}}$ is the reference voltage. Gain, $V_{2P25}$ are constants and $C_F$ is selected to optimize the input sense capacitance range. $C_{S2}$ and $C_{S1}$ are the MEMS capacitor pair and are defined as:

$$C_{S2} = n \cdot \epsilon \cdot \frac{A}{d_1 - x} + n \cdot \epsilon \cdot \frac{A}{d_2 + x} \quad C_{S1} = n \cdot \epsilon \cdot \frac{A}{d_1 + x} + n \cdot \epsilon \cdot \frac{A}{d_2 - x} \quad \text{(A.9)}$$

where $n$ is the number of capacitive comb drives, $\epsilon$ the permittivity of air and $A, d_1, d_2$ the geometric specification of the sensor as shown in Fig [A.5]A. The following table summarizes all constants and parameters:
A.2. DYNAMIC CALIBRATION

Table A.1: Parameters of the transfer equation of the electronic readout system and the values that are used for them in the Eq A.10 for both sensor types.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>FT-S270</th>
<th>FT-S540</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area A</td>
<td>2.5 · 10^{-8} m²</td>
<td>2.5 · 10^{-8} m²</td>
</tr>
<tr>
<td>n</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>d₁</td>
<td>5 µm</td>
<td>5 µm</td>
</tr>
<tr>
<td>d₂</td>
<td>20 µm</td>
<td>20 µm</td>
</tr>
<tr>
<td>ε</td>
<td>8.859 · 10^{-12} F/m</td>
<td>8.859 · 10^{-12} F/m</td>
</tr>
<tr>
<td>C_F</td>
<td>4 pF</td>
<td>4 pF</td>
</tr>
<tr>
<td>Gain</td>
<td>2 V</td>
<td>4 V</td>
</tr>
<tr>
<td>V_{2P25}</td>
<td>2.25 VDC</td>
<td>2.25 VDC</td>
</tr>
<tr>
<td>V_{ref}</td>
<td>2.25 V</td>
<td>2.25 V</td>
</tr>
</tbody>
</table>

Therefore the final transfer function of the MS3110 electronic readout system is defined as:

\[ V(t) = h(x(t)) = \frac{\text{Gain} \cdot V_{2P25} \cdot 1.14 \cdot 2 \cdot n \cdot \epsilon \cdot A \cdot (d_2^2 - d_1^2)}{C_F \cdot d_1^2 \cdot d_2^2} \cdot x(t) + V_{\text{ref}} \quad (A.10) \]

Again, to receive the reconstruction transfer function, we take the inverse of function \( h \)\(^{-1} \) \( (V(t)) = \frac{V(t) - V_{\text{ref}}}{K} \)

The complete transfer function

To calculate the overall voltage-to-force transfer function, we can simply combine the two transfer functions from above:

\[ \hat{F}(s) = f^{-1}(V(s)) = G^{-1}(s) \cdot h^{-1}(V) \]

\[ = m \cdot (s^2 + 2\delta\omega_0 s + \omega_0^2) \cdot \left( \frac{V - V_{\text{ref}}}{K} \right) \quad (A.12) \]

To get to the transfer behavior of the system in the time domain, we have

\(^1\)Since this function is only a gain and offset, it does not play a role whether we use it in time domain or frequency domain.
APPENDIX A. MEMS CALIBRATION

to apply an inverse Laplace transform of the Eq A.11. A separation of
the equation yields:

$$\hat{F}(s) = \frac{m}{K} \cdot \left( s^2 \cdot (V - V_{\text{ref}}) + 2\delta \omega_0 s \cdot (V - V_{\text{ref}}) + \omega_0^2 (V - V_{\text{ref}}) \right)$$  \hspace{1cm} (A.13)

Further, using the properties of Laplace-transformation yields:

$$\hat{F}(t) = \hat{f}^{-1}(V(t)) = \frac{m}{K} \cdot \left( \ddot{V}(t) + 2 \cdot \delta \cdot \dot{V}(t) + \omega_0^2 (V(t) - V_{\text{ref}}) \right)$$  \hspace{1cm} (A.14)

Once the model has been verified and its parameters identified, this function can be used to implement an off-line filter that reconstructs an estimate $\hat{F}(t)$ of the original forces $F(t)$ from the voltage measurements $V(t)$.

A.2.3 Experimental identification

In Eq A.14 certain model parameters are still unknown: the mass $m$, the damping coefficient $\delta$ and the eigenfrequency $\omega_0$. Ideally, to identify these parameters, one would use a sinusoidal force source and increase its frequency while keeping its amplitude constant. The frequency response of the sensor could then directly be measured, and the three parameters above could be extracted.

Unfortunately, there are no ideal force sources like the one described above, so we had to find another technique to extract the missing parameters.

The simplest technique is to use the sensor’s combs as electrostatic actuators: a sinusoidal voltage is applied between the combs, which generates an electrostatic force in between them. The resulting displacement of the sensor probe is precisely measured with a laser vibrometer. A major drawback of this technique is that the exact voltage/force relationship
is unknown. Depending on this relationship, the expected frequency response occurs sometimes at twice the frequency than the excitation. All in all, it is difficult to know exactly how the sensor is being excited.

We decided to implement another experimental paradigm to have a clear understanding of the system. The technique consisted in mounting the entire sensor on a piezo transducer (see Fig A.7). The piezo was aligned so that the movement it induced on the force sensor could be measured by the laser vibrometer. In this experiment, both the sensor’s body and its movable part are oscillating, so two lasers are necessary to measure the relative movement of the two.

![Dynamic calibration setup schematic](image)

**Fig A.7:** A: Dynamic calibration setup schematic. The piezo generates a displacement $e(t)$ of the entire sensor. This generates further oscillations of the sensor movable structure $x(t)$. The relative displacement between the sensor’s base and its movable part is denoted $x_{rel}(t)$. B: Assumed mass-spring-damper system for the experimental setup with the Piezo excitation under the scanning vibrometer.

The mass-damper-model of this system yields:

$$m\ddot{x} = -b\cdot\dot{x}_{rel} - k\cdot x_{rel}$$  \hspace{1cm} (A.16)

where $x_{rel} = x - e$ is the relative deflection between the sensor and the moveable part and $e(t)$ is the displacement initiated by the piezo, as shown in Fig A.7.

Solving this equation in the Laplace domain:

$$G_2(s) = \frac{X(s)}{E(s)} = \frac{2\delta\omega_0 s + \omega_0^2}{s^2 + 2\delta\omega_0 s + \omega_0^2}$$  \hspace{1cm} (A.17)

\(^2\)Such a confusion has probably led FemtoTools to under-estimate the resonance of their sensors by a factor two.
APPENDIX A. MEMS CALIBRATION

This is not exactly the same transfer function than the one we are trying to identify from the sensor: $G(s)$, see Eq A.7. However, the parameters are the same, so we can use $G_2(s)$ to identify $G(s)$.

**Frequency response experiments**

Using the experimental setup described in the previous section, several Bode diagrams of the system were measured. The measurements were done in three steps (see Fig A.8).

1. The reference laser is aimed on the sensor’s body. The measurement laser on a small wing that is rigidly connected to the sensor’s probe.

2. The reference laser is covered with a small cup with a mirror inside while the measurement laser is pointed on the movable part of the sensor. In this setup, the absolute deflection of the probe is measured.

3. The reference laser is still covered with a mirror-cup, but the measurement laser is pointed on the sensor’s body. Using the same excitation signal for the piezo as in experiment 2, the absolute deflection of the sensor’s body is measured.

Theoretically, only the first experiment is necessary. However, the quality of the data obtained with the two laser experiment was inferior to the experiments 2 and 3. We therefore mainly used experiments 2 and 3. Fig A.9 shows the result for the FT-S270 sensor. The vibrometer we used in these experiments measures only the velocities of the deflections. For

---

**Fig A.8:** Illustration of the measurements done on the laser vibrometer. The measurements are enumerated as in the text. a) is the reference laser. b) is the measurement laser.
A.2. DYNAMIC CALIBRATION

Fig A.9: Measurements of the deflection velocities of the FT-S270 sensor in frequency domain. The diagram is almost the same for the measurements with the double laser (blue), and the single laser pointed onto the moveable part (green). The red curve is the body part of the sensor and shows a much lower amplitude (spcb means laser pointing to the body).

For the FT-S270, it is easy to identify the resonance frequency, whereas for the Ft-S540 sensor it is more difficult, due to a second peak around 4.9 kHz and several lower peaks in the whole frequency spectrum. These peaks may appear from different modal forms, e.g. from different direction of deflection while oscillating with the piezo. The 4.9 kHz peak, in particular, shows up clearly in both sensors, and can therefore be assumed to be independent of the sensor’s springs. It is likely caused by the piezo or the connection between the piezo and the sensor, and is disregarded in the following analysis.
Based on these measurements, the resonance frequencies of the peaks are found to be $\approx 6.6$ kHz for the FT-S270 and $\approx 2.57$ kHz for the FT-S540 sensor. We will now discuss how we can extract the model parameters from the Bode diagram.

**Parameter identification**

The parameters that we want to identify from the transfer function $G_2$ (Eq A.17) and the measured Bode diagram are: the mass of the system $m$, the damping constant $b$ and the spring constant $k$.

**Mass of the sensor $m$** The easiest way of obtaining the mass of the sensor is simply to weight it. We removed the movable sensor part of an unused sensor and put it on a micro-scale. Due to the small size of the sensor, 5 capacitor plates broke off. Since their weight is small with respect to the rest of the probe, they were neglected. The final weight was found to be at about $m = 0.7 \pm 0.4$ mg. This value of the mass is assumed to be constant for all sensors of both types.

**Spring constant $k$** If we had the Bode diagram of the sensor system (Eq A.7), the spring constant would simply be the steady state gain (gain at $s = 0$). Unfortunately, the system we measured does not have this property. Its steady state gain is 1. We therefore have to adopt another approach. There are two possibilities to estimate the spring constant:

- Experimentally measuring the spring constant by adding a constant force by another force sensor to the sensor with zero frequency. In this case we know both the force and the displacement of the sensor, so we can calculate $k = F/x$

- Using the resonance frequency $\omega_{res} \approx \omega_0 = \sqrt{k/m}$ (for low damping) from the Bode diagram.

The second way is easier and less time consuming, because there is no need to setup another experiment.

---

This is about the same as the weight of a fruit fly ($\approx 1$ mg).
Damping constant $b$ It is more complex to determine the damping constant $b$, or further the damping coefficient $\delta$. Several ways are proposed:

- Using the bandwidth of the system, respectively the bandwidth of the measured Bode diagram. This method requires a clear phase response of the system.
- Trying to derive the numeric transfer function of the system using the System Identification Toolbox of MATLAB and then compare this transfer function with the model \[A.17\] to identify the parameters.
- If the eigenfrequency and mass is already defined (see identification of spring constant $k$), we can simply derive $\delta$ and furthermore $b$ of the system by combining $b = 2\delta\sqrt{k \cdot m}$ with the following relationship:

If $\delta < 1/\sqrt{2}$, then: (from [Guzella, 2007])

$$|G_2(\omega=\omega_0)| = 20 \log_{10}\left(\frac{1}{2\delta\sqrt{1 - \delta^2}}\right)$$  \hspace{1cm} (A.18)

The vibrometer measurements give us no clear information of the phase response of the system and no clear amplitude peak, so the first method doesn’t seem to be a good way. The second way is not very consistent either because we have no proof of correctness of the model and only a small frequency band of the whole spectrum. Given that the frequency response of the system provides us a clear peak, the third method is the best to estimate the damping constant.

Table A.2: Mechanical parameters of the sensors derived from the measurements

<table>
<thead>
<tr>
<th>Sensor</th>
<th>FT-S270</th>
<th>FT-S540</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mass $m$</td>
<td>0.7 mg</td>
<td>0.7 mg</td>
</tr>
<tr>
<td>Eigenfrequency $\omega_0$</td>
<td>41'928 rad/s</td>
<td>16148 rad/s</td>
</tr>
<tr>
<td>Eigenfrequency $f_0$</td>
<td>6673 Hz</td>
<td>2570 Hz</td>
</tr>
<tr>
<td>Spring constant $k$</td>
<td>1230.6 kg/s²</td>
<td>182.5257 kg/s²</td>
</tr>
<tr>
<td>$\delta$</td>
<td>0.0146</td>
<td>0.053</td>
</tr>
<tr>
<td>Damping constant $b$</td>
<td>$8.564 \cdot 10^{-4}$ kg/s</td>
<td>$11.99 \cdot 10^{-4}$ kg/s</td>
</tr>
</tbody>
</table>

With these Parameters defined, we obtain the following numerical values for the transfer function \[A.17\] of the mechanical model:
APPENDIX A. MEMS CALIBRATION

Fig A.10: Bode diagrams of the measured sensor with the modeled system, using the parameters collected by the experiments for the Sensor FT-S270.

\[
G_{270}(s) = \frac{1224.3s + 17.579 \cdot 10^9}{s^2 + 1224.3s + 17.579 \cdot 10^9} \quad (A.19)
\]

\[
G_{540}(s) = \frac{1711.688s + 260.76 \cdot 10^6}{s^2 + 1711.688s + 260.76 \cdot 10^6} \quad (A.20)
\]

Fig A.10 shows the Bode diagrams of the modeled sensor. The experimental models derived with the numerical transfer functions \( A.19 \) and \( A.20 \) show a remarkable fit to the main resonance response of the measured system. This proves our assumption of the mass-spring-damper model. Note that we did not show the phase information because it could only be measured very poorly with the technique applied here on the laser vibrometer.
A.2. DYNAMIC CALIBRATION

A.2.4 Dynamic calibration implementation

This section deals with the more practical aspects of the dynamic calibration: how do we choose the model parameters for a new sensor and how do we implement a digital filter to transform the voltages back into forces?

System parameter choice

Some of the model parameters are the same for all sensors. For instance, the electrical readout parameters can be considered constant (see Table A.1). The mechanical model parameters are all a little more sensitive to the microfabrication steps, and are therefore prone to variations from one sensor to another. However, the planar dimensions are the most sensitive to microfabrication parameters. In other words, of the three mechanical model parameters, the mass and the damping coefficient will not be very sensitive because they are hardly affected by changes in the planar dimensions. The spring constant, on the other hand, has a cubic dependency on the width of the springs (designed to be 5 µm wide). Therefore, the spring constant needs to be adjusted for each sensor. The technique we used in this section to identify the spring constant was to look at the resonant frequency in the frequency response of the force sensor. While this technique is precise, it is quite time-consuming to perform the laser vibrometer experiments for each sensor. Luckily, the spring constant can be inferred from the much simpler static calibration (already performed by FemtoTools). By combining Eq A.8 and Table 4.1, we obtain:

\[ k = k_1 \cdot K \]  

(A.21)

where \( k \) is the spring constant \( N/m \), \( k_1 \) is the static calibration constant \( N/V \) and \( K \) is the cumulative gain of the electrical readout \( V/m \) (see Eq A.8).

The mass of the probe and the damping coefficients can be assumed to be constant (see Fig A.11). The mass of the tethered object (in our case, the fruit fly) will reduce the resonant frequency and must therefore be taken into account: it is added to the mass of the probe.
Fig A.11: Overview of parameter choice and signal flow in the dynamic calibration implementation.

**Digital filter implementation**

To transform the voltage series $V(t)$ into a force estimate series $\hat{F}(t)$, we must apply $f^{-1}$ (Eq. A.14). The approach we used consisted in numerically differentiating the measurement. Numerical differentiation is very sensitive to noise amplification and we need a second derivative of our signal. To minimize this noise, we used a method called *symmetrical differential quotient*:

Combining the definition of the Fourier rows:

$$f(x + h) = f(x) + \frac{h}{1!} f'(x) + \frac{h^2}{2!} f''(x) + \ldots + \frac{h^k}{k!} f^k(x) \quad \text{(A.22)}$$

$$f(x - h) = f(x) - \frac{h}{1!} f'(x) + \frac{h^2}{2!} f''(x) - \ldots \quad \text{(A.23)}$$

and neglecting terms of higher order yields the symmetric differential quotients for first and second order:
\[ \Delta_h^1(x) = \frac{f(x + h) - f(x - h)}{2h} = f'(x) + \frac{h^2}{6} f'''(x) + O(h^2) \]  
\[ \Delta_h^2(x) = \frac{f(x + h) - 2f(x) + f(x - h)}{h^2} = f''(x) + O(h^2) \]

where the step size \( h \) is the sampling time \( t = 1/f_s = 1 \cdot 10^{-4} \) in our case.

### A.2.5 Validation of the dynamic calibration

For signals with low frequencies \((s \approx 0)\), we have a redundant calibration: the dynamic calibration should match the static calibration. This allows us to verify that the steady state gain of the dynamic calibration is correct.

#### Table A.3: Sensitivities of both model and Femto tools fact sheet and their error for both considered sensors.

<table>
<thead>
<tr>
<th>Sensor Nr.</th>
<th>Type</th>
<th>Sensitivity Model</th>
<th>Sensitivity Fact sheet</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1036</td>
<td>FT-S270</td>
<td>1155.27 µN/V</td>
<td>1050.8 µN/V</td>
<td>9%</td>
</tr>
<tr>
<td>1044</td>
<td>FT-S540</td>
<td>85.68 µN/V</td>
<td>73.2 µN/V</td>
<td>14.6%</td>
</tr>
</tbody>
</table>

These errors are in a very acceptable range and have different potential causes:

- Error in the mass of the sensor probe, either due to the broken capacitor plates or the resolution of the scale
- Error in the measurement of the eigenfrequency of the system (this would result in a different damping and spring constant)
- The transfer function of the electronic readout system (Eq A.8) was approximated to be linear.

### A.2.6 Results

To test the dynamic calibration of the sensors, the implemented filters have been used with data obtained with different fruit flies tethered to the probe. The flies were visually stimulated with light sources at different frequencies. Since the FT-S540 sensors are much more sensitive, most
measurements have been done using this sensor instead of the FT-S270 sensors.

The dynamic calibration filter has been tested on several flies using the FT-S540 sensors. The dynamic calibration effect is very visible: both the amplitude and the phase of the signal is corrected (see Fig A.12).

A.2.7 As of when is a dynamic calibration necessary?

We started off this chapter by saying that force sources containing high frequency information would require a dynamic calibration. We are now able to quantitatively determine as of which frequency the static calibration does not suffice. In Fig A.13 we show the calibration gain as a function of frequency for both static and dynamic calibrations. The static calibration is independent of frequency. The difference in between the two curves can be interpreted as the error of force amplitude made by employing a static calibration on a time-dependent signal. The point

Fig A.12: Comparison of the static calibration with the dynamic calibration. The phase and amplitude shift caused by the resonance effects of the sensor are very visible.
A.2. DYNAMIC CALIBRATION

Fig A.13: Calibration gain [N/V] as a function of frequency (mass of fly included).

where this error exceeds 3dB is marked with a dotted line. This happens above $\approx 1000$ Hz for the FT-S540 Sensor and above $\approx 2200$ Hz for the FT-S270. Note that in this case, the mass of the fly was taken into account ($\approx 1$ mg. For a heavier object, these thresholds would be lower.
Appendix B

Implementation of a quasi-steady model of lift generation

This chapter covers the model of lift generation that was compared to instantaneous forces and kinematics measured on the fruit fly (see Section 5.1). Vinzenz Schönenfelder implemented the model based on [Sane and Dickinson, 2001].

B.1 Model variables and constants

To describe stroke kinematics, the three instantaneous stroke angles $\alpha$, $\theta$, $\phi$ were defined similar to the standard literature [Sane and Dickinson, 2001]:

$$\alpha \quad \text{aerodynamic angle of attack}$$
$$\theta \quad \text{vertical stroke-deviation}$$
$$\phi \quad \text{horizontal stroke-position}$$

The aerodynamic angle of attack $\alpha$ is different from the physiological angle of attack. It is defined as the angle between the leading edge of the wing $\vec{x}_{le}$ and the direction of motion of the wing tip $\vec{U}_t$ (both unitary vectors):

$$\cos \alpha = \vec{x}_{le} \cdot \vec{U}_t$$
### APPENDIX B. QSS MODEL OF LIFT GENERATION

<table>
<thead>
<tr>
<th>variable</th>
<th>value</th>
<th>unit</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\rho_a$</td>
<td>1.2</td>
<td>kg/m$^3$</td>
<td>density of air [Lehmann and Dickinson, 1997]</td>
</tr>
<tr>
<td>$\rho_w$</td>
<td>1200</td>
<td>kg/m$^3$</td>
<td>density of wing material (solid cuticle) [Ellington, 1984b, Lehmann and Dickinson, 1997]</td>
</tr>
<tr>
<td>$R_w$</td>
<td>2.47E-3</td>
<td>m</td>
<td>wing length [Lehmann and Dickinson, 1997]</td>
</tr>
<tr>
<td>$\tau_w$</td>
<td>0.80E-3</td>
<td>m</td>
<td>mean wing chord [Ellington, 1984b, Lehmann and Dickinson, 1997]</td>
</tr>
<tr>
<td>$h_w$</td>
<td>1.33E-6</td>
<td>m</td>
<td>wing thickness [Ellington, 1984b, Lehmann and Dickinson, 1997]</td>
</tr>
<tr>
<td>$S$</td>
<td>1.98E-6</td>
<td>m$^2$</td>
<td>wing surface = $R_w\tau_w$</td>
</tr>
<tr>
<td>$m_w$</td>
<td>3.16E-9</td>
<td>kg</td>
<td>wing mass = $\rho_w R_w \tau_w h_w$ [Ellington, 1984b, Lehmann and Dickinson, 1997]</td>
</tr>
<tr>
<td>$x_{cm}$</td>
<td>0.7</td>
<td>$R_w$</td>
<td>relative length-wise centre of mass position [Fry et al., 2005]</td>
</tr>
</tbody>
</table>

Table B.1: Constants defining wing morphology, physiology and fluid properties, as they were used in the inertial and aerodynamic model.

Apart from these variables, all modeled forces contain constants deriving from wing morphology, physiology and fluid properties. The values were gathered from Ellington [Ellington, 1984b], Lehmann [Lehmann and Dickinson, 1997] and Fry [Fry et al., 2005] (see Table B.1)

### B.2 Inertial forces

According to standard mechanics, the inertial force that a flapping wing exerts on the fly’s body corresponds to

\[
\vec{F}_{acc} = -m_w \ddot{x}_{cm}
\]

where $m_w$ is the wing mass and $\ddot{x}_{cm}$ is the position of the centre of mass of the wing.

The length-wise position of the wing’s centre of mass is situated approximately at 70% of the line from base to tip ($\hat{d}_x = 0.7$) [Fry et al., 2005],
B.3. AERODYNAMIC FORCES

the chordwise position $\hat{d}_y$ was kept as a variable within the model at $0.2 - 0.5$ in units of average chord length measured from the leading edge (no measured value found in the literature).

B.3 Aerodynamic forces

The instantaneous forces generated by a flapping wing are represented as the sum of three separate force components: added mass, translational and rotational force. Each force component is assumed to act normal to the wing surface [Sane and Dickinson, 2002]. Therefore, only the magnitude of the different forces has to be summed to obtain the combined force:

$$F_{\text{aero}} = F_{\text{admass}} + F_{\text{trans}} + F_{\text{rot}}$$

- **Translational force** The translational force depends on the instantaneous wing tip velocity magnitude $U_t$ and the aerodynamic angle of attack $\alpha$. As customary the translational force is divided into a drag part (acting anti-parallel to the direction of motion) and lift part (acting perpendicular to the direction of motion). Drag and lift coefficients were determined empirically [Dickinson et al., 1999] and summed through vector addition:

  $$U_t^2 = R^2 (\dot{\theta}^2 + \dot{\phi}^2)$$
  $$C_l(\alpha) = -0.004 + 1.92 \sin(1.95\alpha + 0.04)$$
  $$C_d(\alpha) = 2.10 + 1.57 \sin(2.04\alpha - 1.64)$$
  $$F_{\text{trans}} = \frac{1}{2} \rho_a U_t^2 \dot{\alpha}^2 \sqrt{C_l^2(\alpha) + C_d^2(\alpha)}.$$

- **Rotational force** Using Kutta-Jukowski aerodynamic theory, rotational force has been estimated theoretically [Sane and Dickinson, 2002]. It depends on the wing tip velocity $U_t$ and the first derivative of the aerodynamic angle of attack $\omega = \dot{\alpha}$. Further, it depends on the axis

1 Different signs have to be taken into account.
2 modified values from personal communication with W.B. Dickson.
of $\alpha$-rotation of the wing determined by the non-dimensional axis of rotation $\hat{x}_0$: a rotation about the leading edge corresponds to $\hat{x}_0 = 0$, about the trailing edge to $\hat{x}_0 = 1$.

The rotational coefficient $C_r$ and corresponding force are given by:

$$C_{\text{rot}} = \pi (0.75 - \hat{x}_0)$$

$$F_{\text{rot}} = C_r \rho U_t \omega \bar{c}_w^2 \int_0^1 \hat{r} \hat{c}_w^2 (\hat{r}) d\hat{r}$$

- **Added mass force** The added mass component has been estimated theoretically [Sedov, 1965] to correspond to

$$F_{\text{admass}} = \rho \pi \frac{R^2 \bar{c}_w^2}{4} \int_0^1 \hat{r} \hat{c}_w^2 (\hat{r}) d\hat{r}$$

(The second summand in formula (3) in [Sane and Dickinson, 2002] has been eliminated since W.B. Dickson confirmed [personal communication] that it does not give accurate and physically plausible results).

- **Morphological integrals** The formulae for the revised quasi-steady model contain two integrals comprising the non-dimensional wing chord and non-dimensional radial position (for definitions see [Ellington, 1984b]). They depend on wing morphology and can be estimated assuming an elliptic wing shape. Instead, I used values that W.B. Dickson kindly disclosed [personal communication]:

<table>
<thead>
<tr>
<th>Integral</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\int_0^1 \hat{r} \hat{c}_w^2 (\hat{r}) d\hat{r}$</td>
<td>0.829</td>
</tr>
<tr>
<td>$\int_0^1 \hat{r}^2 \hat{c}_w (\hat{r}) d\hat{r}$</td>
<td>$\frac{R}{2}$ [Ellington, 1984b]</td>
</tr>
</tbody>
</table>
Appendix C

Exploration of bio/robot coupling

In this chapter, 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 C.1).

The idea to implement a Cyborg Fly came from Vasco Medici and Steven Fry when we were asked to create a workshop at a conference where both roboticists and biologists would take part. Our goal was to foster interactions of the two fields.

While robotics can benefit from biological research, robotic platforms can conversely enhance biological understanding by serving as testbeds to explore and validate hypothesis of sensorimotor pathways [Franceschini et al., 2007, Halloy et al., 2007]. 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 [Abbott, 2007].

Though such biorobotic implementations are intriguing, the transfer between a target biological system and its robotic counterpart is non-trivial and prone to misconceptions [Webb, 2006, Datteri and Tamburrini, 2007]. 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.

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

---

**Fig C.1:** Fly-in-the-loop: the fruit fly’s normal sensory feedback loop is replaced with a non-invasive biorobotic coupling (grey box).
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 making 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 C.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 [Hertz, 2008].

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 neuroprosthetics have been studying techniques to have animals directly control robotic devices [Kositsky et al., 2003, Sato et al., 2008, Wessberg et al., 2000, Abbott, 2006].

C.1 The Cyborg system

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

C.1.1 Fruit fly

We tethered wild-type fruit flies (*Drosophila melanogaster*) to a tungsten rod using standard procedures [Tammero and Dickinson, 2002].

C.1.2 Digital wing beat analyzer

The digital wing beat analyzer (see Section 4.2) 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 \( \mathbf{x} \) provides a real-time estimation of wing beat frequency \( f \), amplitude \( A \), phase \( \phi \) and mean angular position \( m \), which can be transformed into motor commands to the robot.
Fig C.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: Left wing amplitude measured from high speed camera E: Right wing amplitude F: Left wheel speed G: Right wheel speed. The wheel speed commands were generated from the wing kinematics and fly-to-robot transfer function in Eq C.3. Initially, the robot is turning right, but then a sudden decrease in the fly’s left wing amplitude makes it turn left.
C.1.3 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:

\[
\begin{pmatrix} \dot{\theta}_l \\ \dot{\theta}_r \end{pmatrix} = \text{TF}_{\text{fly-to-robot}}(x_l, x_r) = \text{TF}_{\text{fly-to-robot}}(A_l, A_r, m_l, m_r, \phi_l, \phi_r, f) \tag{C.1}
\]

where \(\dot{\theta}_l\) and \(\dot{\theta}_r\) are the angular speed commands to the left and right wheels of the robot [°/s], \(x = [A \ m \ \phi \ f]^T \in \mathbb{R}^4\) 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, \(f_l = f_r = f\) in (C.2). Note that each term in (C.1) and (C.2) are intrinsically time dependent and (C.1) may be a function of state vectors at different discrete times \((x(t), x(t-1), ..., x(t-n))\).

The choice of (C.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 \((A_l - A_r)\) [Fry et al., 2003]. Lift is controlled by increasing the mean speed of both wings \((A \cdot f)\) [Lehmann and Dickinson, 1997]. Thrust is controlled by a modulation of both pitch (induced by changes in the mean stroke position \(m\)) and total force magnitude (proportional to \(A \cdot f\)) [Fry et al., 2005].

As a note of caution, tethered flight is subject to some quantifiable artifacts [Fry et al., 2005], 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:

\[
\begin{pmatrix} \dot{\theta}_l \\ \dot{\theta}_r \end{pmatrix} = G_1 \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} A_l \\ A_r \end{bmatrix} + H_1 \tag{C.3}
\]
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:

\[
\begin{bmatrix}
    \dot{\theta}_l \\
    \dot{\theta}_r
\end{bmatrix} = G_2 \cdot f \cdot \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} A_l \\ A_r \end{bmatrix} + H_2
\]

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.

### C.1.4 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.

### C.1.5 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:

\[
I(x, y) = \text{TF}_{\text{robot-to-fly}}(i, p)
\]

where \( I(x, y) \) represents the image to be shown on the flight arena. \( i = (i_1, i_2, ..., i_{306}) \in \mathbb{R}^{306} \) is the linear image from the three cameras of the robot and \( p = [p_1, p_2, ..., p_8] \in \mathbb{R}^8 \) 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 C.1 gives a list of the ones we used.
Table C.1: Employed robot-to-fly transfer functions

<table>
<thead>
<tr>
<th>Robot-to-fly transfer function type</th>
<th>Description</th>
<th>Image</th>
</tr>
</thead>
<tbody>
<tr>
<td>A. Undistorted robot view</td>
<td>The fly directly sees the output of the linear cameras.</td>
<td></td>
</tr>
<tr>
<td>B. Nearest object defines azimuth and speed of expansion/contraction</td>
<td>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.</td>
<td></td>
</tr>
<tr>
<td>C. Brightest object defines stripe position</td>
<td>The azimuth of the brightest object in the linear camera image defines the azimuth of the bright stripe in the flight arena.</td>
<td></td>
</tr>
<tr>
<td>D. Object distance drives sinusoidal grating vertical speed</td>
<td>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.</td>
<td></td>
</tr>
</tbody>
</table>

C.1.6 LED visual flight simulator

The LED visual flight simulator (see Section 4.3) was refreshed at 30-400Hz, depending on the choice of the robot-to-fly transfer function.

C.1.7 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 user the possibility to vary the transfer function parameters (see Fig C.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
C.2 Experiments

C.2.1 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. (C.3) and (C.4) to have the robot a) turn proportionally to the difference in wing beat amplitude and b) advance at a speed proportional to the mean wing speed. Experimental data is shown in Fig C.3 E-H. The fly was shown a live view from the robot’s cameras (Table C.1.A). These feedbacks were similar to the feedback used in the cockroach experiments by Hertz et al [Hertz, 2008].

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 repeatedly crash into obstacles, not showing the desired behavior.

C.2.2 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 C.1.B). In this amplified situation, the Cyborg system showed robust obstacle avoidance behavior. After adjusting the transfer function gains, the robot drove through the arena for several minutes while avoiding walls and obstacles.

C.2.3 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 C.1.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.
C.2.4 Decoupled response feedback

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

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

(C.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 C.1.D). These experiments succeeded equally well in generating stable obstacle-avoidance behavior.

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 the robot does show similar behavior to that of a fly, it doesn’t necessarily represent a truly biomimetic implementation.

C.3 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 [Webb, 2006], 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 [Webb, 2006]! 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 (C.3) & (C.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 target 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 (C.2.3) or completely (C.2.4) decoupled processes to generate stable closed-loop turning behavior (Eq C.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.
Appendix D

Press coverage

- **Cover picture:** HFSP Journal. Frontiers of Interdisciplinary Research in the Life Sciences, April 2008
- **TV reportage:** "The humble fly helps in latest robotics research" featured on Euronews: Hi-Tech and TSR: Nouvo, February 2008
- **TV documentary:** "Robot volanti" featured on TSI: Il Giardino di Albert, November 2007
- **Cover story:** "CMOS captures fruit fly flight" in Systems Vision Design, February 2007
- **TV documentary:** "The flight of the fruit fly" featured on Deutsche Welle: Tomorrow Today, August 2006
- **Picture of the month:** Swiss National Science Foundation. Multiple radio and newspaper coverage, April 2006
High-Speed Image Processing
CMOS camera captures fruit fly flight
References


REFERENCES


[Rohrseitz and Fry, ] Rohrseitz, N. and Fry, S. In prep.


REFERENCES


REFERENCES


Chauncey GRAETZEL  
Malvenstrasse 12                                                                     
8057 Zürich  
Switzerland  
Tel: +41 76 391 4821  
Email: cgraetzel@ethz.ch

Birthday: 29.09.1980  
Civil Status: Married  
Dual Citizenship: Swiss, American

---

**RESEARCH / EDUCATION**

<table>
<thead>
<tr>
<th>Date</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>2004.08</td>
<td><strong>PhD degree, Mechanical engineering, ETH Zürich</strong></td>
</tr>
<tr>
<td>– 2008.11</td>
<td>Novel microrobotic tools applied to reverse-engineer micro-aerial flight in biology</td>
</tr>
<tr>
<td></td>
<td>Supervisors: Professor Bradley Nelson / Dr. Steven Fry</td>
</tr>
<tr>
<td></td>
<td>Institute of Robotics and Intelligent Systems / Institute of Neuroinformatics</td>
</tr>
<tr>
<td></td>
<td>Direct supervision of 8 Bachelor theses and 2 Master theses.</td>
</tr>
<tr>
<td>1999.10</td>
<td><strong>Diploma (MSc equivalent) in Micro-engineering, EPFL, Lausanne</strong></td>
</tr>
<tr>
<td>– 2004.03</td>
<td>Specialization: Robotics and Automation</td>
</tr>
<tr>
<td></td>
<td>Rank: 2nd/83</td>
</tr>
<tr>
<td></td>
<td>One year exchange program, University of Texas, Austin</td>
</tr>
<tr>
<td></td>
<td><strong>Master thesis</strong>: Medical rehabilitation robotics</td>
</tr>
<tr>
<td></td>
<td>Laboratory of Robotic Systems. Prof. Raymond Clavel</td>
</tr>
<tr>
<td>1996.08</td>
<td><strong>Federal Maturity</strong>, scientific section</td>
</tr>
<tr>
<td>– 1999.07</td>
<td>Gymnase Auguste Piccard, Lausanne</td>
</tr>
<tr>
<td></td>
<td>Award for best overall results. Physics prize</td>
</tr>
</tbody>
</table>

---

**TEACHING / PROFESSIONAL ACTIVITIES**

<table>
<thead>
<tr>
<th>Date</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>2005.03</td>
<td><strong>Studies on Mechatronics, ETHZ - Supervisor</strong></td>
</tr>
<tr>
<td>– 2008.06</td>
<td>Supervision of 6 student projects</td>
</tr>
<tr>
<td>2005.03</td>
<td><strong>Advanced Robotics and Mechatronics, ETHZ - Teaching assistant</strong></td>
</tr>
<tr>
<td>– 2006.07</td>
<td>Robotic competition organization. Supervision of student teams</td>
</tr>
<tr>
<td>2004.10</td>
<td><strong>Introduction to Robotics and Mechatronics, ETHZ - Teaching assistant</strong></td>
</tr>
<tr>
<td>– 2007.02</td>
<td>Replacement lecturer. Management of laboratory sessions</td>
</tr>
<tr>
<td>2003.01</td>
<td><strong>Virtual Reality and Active Interfaces group, EPFL - Research assistant</strong></td>
</tr>
<tr>
<td>– 2004.07</td>
<td>Development and clinical testing of a non-contact surgeon-computer user interface</td>
</tr>
</tbody>
</table>

---

**PEER-REVIEWED PUBLICATIONS**

**Journals / Book chapters**


International conferences

PRESS COVERAGE
Cover picture: HFSP Journal. Frontiers of Interdisciplinary Research in the Life Sciences, April 2008
Invited lecturer: Tufts University, February 2008
TV reportage: “The humble fly helps in latest robotics research” featured on Euronews: Hi-Tech and TSR: Nouvo, February 2008

LANGUAGES
French and English: native languages
German: fluent speaking and good writing skills.

COMPUTER SKILLS
Presenting & publishing: Office, Photoshop, Illustrator, LaTeX, PHP, HTML
Programming: C/C++, LabVIEW, LV Real Time & FPGA, Visual Studio, KDevelop
CAD: ProEngineer, I-DEAS, Unigraphics NX, Altium Designer
Analysis: Matlab, Simulink, Mathematica

EXTERNAL ACTIVITIES
Sports: Soccer player, competitive cyclist, mountaineering.
Music: Guitarist.
Other: Astronomy.