

Imperial College London
Department of Bioengineering

Bio-inspired robotics: efference copies and adaptive feedforward control

Benjamin P. Campbell

IMPERIAL

Submitted in part fulfilment of the requirements
for the degree of Doctor of Philosophy at
Imperial College London
December 10, 2025



I hereby declare that this thesis and the work reported herein was composed by and originated entirely from me. Information derived from the published and unpublished work of others has been acknowledged in the text and references are given in the list of sources.

Benjamin P. Campbell (2025)

The copyright of this thesis rests with the author. Unless otherwise indicated, its contents are licensed under a Creative Commons Attribution-non Commercial-No Derivatives 4.0 International Licence (CC BY-NC-ND). Under this licence, you may copy and redistribute the material in any medium or format on the condition that; you credit the author, do not use it for commercial purposes and do not distribute modified versions of the work. When reusing or sharing this work, ensure you make the license terms clear to others by naming the licence and linking to the licence text. Please seek permission from copyright holder for uses of this work that are not included in this licence or permitted under UK Copyright Law.

Publications

Some ideas and figures have appeared previously or are due to appear in the following publications. The contributions of others are discussed in the acknowledgements section.

Campbell, B. P., Lin, H. T., & Krapp, H. G. (2023). Weighting Elementary Movement Detectors Tuned to Different Temporal Frequencies to Estimate Image Velocity. In Conference on Biomimetic and Biohybrid Systems (pp. 398-410). Cham: Springer Nature Switzerland.

Aspects of the contents of Chapter 4 has been accepted for publication in the Institute for Automation and Control (IFAC) conference on adaptive and learning control systems (2025). Under the title: Implicit adaptive feedforward control. With authors: Campbell, B. P., Lin, H. T., & Krapp, H. G.

Aspects of the contents of Chapter 5 has been accepted for publication in the Conference on Biomimetic and Biohybrid Systems (2025). Cham: Springer Nature Switzerland. Under the title: Efference copy control for MIMO systems with multiple different time delays. With authors: Campbell, B. P., Lin, H. T., & Krapp, H. G.

Aspects of the contents of Chapter 2 are under review in Biological Cybernetics. Under the title: A control engineering perspective on the advantages of efference copies. With authors: Campbell, B. P., Lin, H. T., & Krapp, H. G. In addition, it is available on bioRxiv.

Aspects of the contents of Chapter 3 are under review in PLoS Biology. Under the title: Adaptive feedforward speed control in *Drosophila*. With authors: Campbell, B. P., Supple, J. A., Fabian, S. T., Lin, H. T., & Krapp, H. G. In addition, it is available on bioRxiv.

Abstract

This thesis investigates how insects control movement using adaptive feedforward control and efference copies, and how these principles can be applied to robotic systems. A phenomenological control architecture—the fully-separable-degrees-of-freedom (FSDoF) controller—is developed to formalise efference copies in a control-theoretic context. The FSDoF architecture is used in simulation to examine control advantages of efference copy-based systems. Specifically, it enables sensors to remain within their operating range, thereby optimising sensitivity to external disturbances, and is particularly effective in systems with time delays and significant sensor noise. When combined with high-pass-filter-like sensors, the FSDoF controller also exhibits a unique robustness to changes in sensor gain and bandwidth. Furthermore, a method is introduced to design FSDoF controllers for multi-input multi-output (MIMO) systems with multiple, distinct time delays, establishing its theoretical applicability to such systems. This work formalises the efference copy hypothesis in insects from an engineering perspective, offering a functional explanation for their evolutionary advantages, and enabling the generation of experimentally testable hypotheses.

One such hypothesis is that insect feedforward controllers adapt to changes in system dynamics. A series of free-flight experiments in *Drosophila melanogaster* supports this idea. Following unilateral wing damage, flies exhibited increased flight speeds in darkness compared to intact controls. When flying in light, with visual feedback, their speed was reduced to near-control levels. Importantly, this reduction persisted when the flies returned to darkness, suggesting a lasting change in their flight speed controller. This trend was also observed in flies that had their antennal arista removed, the primary airspeed sensor. Altogether, the results indicate that the adaptation was not solely due to changes in mechanosensory feedback, but suggest adaptive feedforward control.

Building on these insights, an implicit adaptive feedforward (IAFF) control strategy is proposed and implemented on a quadrotor platform. This controller uses adaptive sensitivity derivatives, so that it could, in theory, cope with an inversion of the system dynamics. IAFF control improved the reference tracking in a Parrot Mambo mini-drone within minutes of operation, starting from a randomised feedforward controller.

Acknowledgements

It is a real pleasure to thank my supervisors, Holger Krapp and Huai-Ti Lin. I got so lucky. Holger, not many people can keep me engaged in three-hour one-on-one meetings; your enthusiasm, knowledge, and support made these past four years enjoyable; your guidance and direction were instrumental in getting these results, and always balanced with the freedom to explore; in a symbolic effort to thank you for everything you have taught me, I hereby promise that this will be the last overly long, semicolon-cluttered sentence I ever write. Huai-Ti, your work ethic and long-term vision are inspiring. Your feedback has been extremely valuable, and I look forward to continuing to work with you in the future.

Throughout my PhD, I have been lucky to know and work with everyone in the Krapp Lab and NBits Lab. In particular, Jack Supple and Samuel Fabian, thank you for your enormous help in running the experiments and providing continuous feedback on the contents of Chapter 3. Sam, although you turned out to be completely wrong, without your initial assurance that the experiments would “be easy”, they would never have happened. Thank you for all your work on the experimental setup, collecting the preliminary data, and developing the tracking algorithm. Jack, later on, when it turned out that the preliminary data was just that—preliminary, you came to the rescue. You brought in controls, temperature measurements, rigorous timings, circadian considerations, and many hours of systematic wing and aristae clipping. The work we did was the first biological experiment I had done since I was 16, and I couldn’t have had two more helpful and supportive people to do it with. It was a genuine pleasure working with both of you. In addition, I would like to thank Myriam, Hendrik, Liv, and Haaris for their proofreading and feedback on this thesis.

Thank you to the Defence Science and Technology Laboratories (DSTL) for supporting this project. In particular, thank you to Glenn for all the feedback and advice over the past four years, and to Helen for continuous support throughout the project.

Mum and Dad, there will never be a time when I am not grateful for everything you both did for me. Oli, my baby bromine, you have inspired me more than you know.

To my housemates over the past four years, Ollie, Anna, Miles, Brian, Niall, Jizzle, and Liam, you have all contributed. I came home every day to a fun, positive, and caring family 2.0.

Nina, despite literally limiting the fresh air and sunlight I got these past few months, the figurative fresh air and sunlight you brought me were instrumental in the preparation of this thesis, and in my happiness across the board. Thank you.

Contents

Publications	3
Abstract	4
Acknowledgements	5
Nomenclature	11
1 Introduction	12
1.1 Overview	12
1.2 Control systems	13
1.2.1 Feedback control	14
1.2.2 Feedforward control	14
1.2.3 Other classical control methods	15
1.2.4 Deep-learning based control methods	18
1.2.5 Adaptive control	20
1.2.6 Summary	21
1.3 Control in insects	21
1.3.1 Insect sensors	22
1.3.2 The inner-outer-loop hypothesis	25
2 The advantages of efference copies	31
2.1 Formulation of the FSDoF insect sensorimotor control architecture	34
2.2 Methods	36
2.3 Results	39
2.3.1 Advantage 1: Mitigating the impact of sensor delays	40
2.3.2 Advantage 2: Mitigation of sensor noise	43
2.3.3 Advantage 3: Preventing sensor saturation	44
2.3.4 Controller robustness	45
2.4 Controller variants	49
2.5 Conclusions	53
3 Adaptive feedforward speed control following wing damage in <i>Drosophila</i>	56
3.1 Introduction	56
3.2 Methods	58
3.2.1 Animals and ablation procedures	58
3.2.2 Experimental setup and procedure	59
3.2.3 Tracking and analysis	62
3.3 Results	65

3.3.1	Experiment 1: Lasting recovery of flight kinematics following light exposure in clipped-wing flies	65
3.3.2	How could the controller change?	68
3.3.3	Experiment 2: Speed controller changes with bilaterally ablated aristae	71
3.4	Discussion	73
4	Implicit adaptive feedforward control	78
4.1	Introduction	78
4.2	Fully-separable-degrees-of-freedom control	80
4.3	Implicit adaptation	82
4.3.1	The feedforward filter	82
4.3.2	The gradient filter	84
4.3.3	Stability analysis	84
4.4	Example implementation	86
4.5	Drone implementation	90
4.5.1	Drone setup and parameters	91
4.5.2	Drone results	94
4.6	Discussion	97
5	FSDoF control of MIMO systems with multiple different time delays	101
5.1	Introduction	101
5.2	Methods	103
5.2.1	FSDoF on MDTD systems	104
5.2.2	Simulation and benchmarking methods	107
5.3	Results	110
5.4	Implicit adaptive feedforward control on multiple different time delay systems	113
5.5	Discussion	114
6	Conclusion	117
6.1	Summary of contributions	117
6.2	Limitations and open questions	119
A	Further stability analysis of IAFF	152
A.1	α_g upper-bound derivation	152
A.2	Resulting loss change without assuming constant input	153
A.3	Including regularisation in the learning rate upper-bounds	154
A.4	Numerical stability analysis	156
B	Analysis of temperature as a confounding factor in behavioural experiments in Ch. 3	160
C	Supporting data tables	163
C.1	SISO controller comparison data (Ch. 2)	164
C.1.1	Controller tuning	164
C.1.2	Results	165
C.2	Flight kinematics average data (Ch. 3)	168

List of Tables

2.1	Six structures of system dynamics for benchmarking FSDoF performance against the SP and PFB	36
3.1	Numbers of animals and inter-saccadic flight segments used in the statistical analysis and results	59
5.1	MIMO MDTD system dynamics used to benchmark MPC and FSDoF and the justification for their use	108
5.2	Costs of MPC and FSDoF on MIMO MDTD systems	113
C.1	Controller parameters for the SP, PFB, and FSDoF used in the simulations in Ch. 2	164
C.2	PFB results for simulations in Ch. 2	165
C.3	SP results for simulations in Ch. 2	166
C.4	FSDoF results for simulations in Ch. 2	167
C.5	Mean and standard deviation of flight kinematics for intact animals in Experiment 1 (Ch. 3)	168
C.6	Mean and standard deviation of flight kinematics for wing-damaged animals in Experiment 1 (Ch. 3)	168
C.7	Mean and standard deviation of kinematic data in wing-damage and ablated antennae animals in Experiment 2 (Ch. 3)	168

List of Figures

1.1	Reichardt detector model and response characteristics	24
1.2	Inner-outer-loop hypothesis diagram	26
2.1	Three control architectures: pure feedback (PFB), smith predictor (SP), and efference copy-based fully-separable degrees of freedom (FSDoF) controller . . .	32
2.2	Example step response of PFB, SP, and FSDoF controllers on the same system dynamics	39
2.3	Overshoot and settle time performance of PFB, SP, and FSDoF controlling systems with different sensor delays	40
2.4	Example step responses demonstrating instances where FSDoF can enact arbi- trarily fast desired state changes	42
2.5	Impact of sensor noise on reference tracking performance exemplified for FSDoF, two PFBs, and three SPs	44
2.6	Examples of how FSDoF, PFB, and SP respond to variable gain and bandwidth system and sensor dynamics	48
2.7	Five theoretically identical control architectures to the FSDoF	50
3.1	Clipped-wing (CW) <i>D. melanogaster</i>	59
3.2	Experimental setup and example <i>D. melanogaster</i> trajectories for behavioural experiment exploring adaptation to wing damage	60
3.3	The effect of circadian cycle, and two different dark-after-light filming procedures on free flight kinematic parameters	62
3.4	Tracking, reconstruction, and filtering procedure demonstrated on an example trajectory	64
3.5	Inter-saccadic kinematic parameters for different lighting conditions in intact and wing-damaged flies	67
3.6	Phenomenological control diagram highlighting two possible methods for adaption of the speed controller in <i>D. melanogaster</i>	69
3.7	Inter-saccadic kinematic parameters for different lighting conditions in wing- clipped ablated-aristae <i>D. melanogaster</i>	72
4.1	Implicit adaptive feedforward (IAFF) control diagram	81
4.2	Simulation of IAFF on a two-input-two-output system	88
4.3	Variation in adaption time due to different learning rate in IAFF	89
4.4	How the magnitude of the initial weights in IAFF determine the time to adapt and the initial error	90
4.5	Different drone control architectures implemented	92
4.6	Comparison of IAFF and feedback control on a Parrot Mambo minidrone in hardware and simulation	94

4.7	Hardware and simulation comparison between feedback control and IAFF when coping with propellor damage on a Parrot Mambo minidrone	97
5.1	FSDoF, model predictive control (MPC), and Padé approximation	103
5.2	Diagram of how multiple different sensor and muscle activation delays can be included and taken account of by the FSDoF control architecture.	106
5.3	Step responses of unique MPCs and FSDoF controllers on the different MIMO MDTD dynamics	111
5.4	MPC and FSDoF performance across the different MIMO MDTD dynamics used to benchmark	112
A.1	Process to calculate numerical estimates of the upper bounds on the learning rates for IAFF	156
A.2	Numerical estimates of the upper bounds of the learning rates in IAFF for different magnitude reference signals and different gain and bandwidth system dynamics .	158
A.3	Numerical estimates of the upper bounds of the learning rates in IAFF for different learning rates	159
B.1	ANCOVA for temperature as a confounding factor in behavioural experiments in Ch. 3	161

Nomenclature

List of Abbreviations

AA	Ablated antennae (flies)
CW	Clipped wing (flies)
DAL	Dark-after-light (lights off after having flown in the light-experimental condition)
D	Dark (lights off-experimental condition)
FEL	Feedback error learning
FSDoF	Fully separable degrees of freedom (efference copy-inspired controller)
HS	Horizontal system (HS-cells in the fly visual system respond to horizontal motion)
IAFF	Implicit adaptive feedforward (control)
I	Intact (flies)
INDI	Incremental nonlinear dynamic inversion
L	Light (lights on-experimental condition)
MDTD	Multiple different time delays
MIMO	Multi-input-multi-output (system dynamics)
MPC	Model predictive control
MRAC	Model reference adaptive control
PFB	Pure feedback control
PID	Proportional integral derivative (control)
RL	Reinforcement learning
RMSE	Root mean square error
SP	Smith predictor

Chapter 1

Introduction

Recent computational advances have enabled some robotics and machine learning algorithms to outperform humans in many cognitive challenges and games. However, animals are still far superior in sensorimotor learning and flexible control. A human can ride a bike to the beach, go swimming, play volleyball, and then drive a car to their violin lesson. The coordination and control for each activity is learned during life, and accuracy is honed through practice. Although robots frequently demonstrate incredible precision in manufacturing, surgical robotics, and drone flight; their ability to adapt to changing dynamics and learn new dynamics is still limited.

1.1 Overview

This thesis aims to further our understanding of the control architectures used by insects, how they might adapt, and test those hypothesised adaptive control schemes in robotic systems. Insects and humans provide the most useful biological platforms for understanding adaptive control: insects because of the accessibility of their brain, comparative simplicity, and availability, and humans because they are instructable.

The baseline control architectures used by insects, even without adaption, differ from those typically used in robotics. In [Chapter 2](#), I analyse a phenomenological control system that models a hypothesised insect control strategy. Through simulations, I show that the architecture is particularly adept at controlling systems where the sensors have noise or delays associated

with them. In addition, the architecture provides an innate error signal that can be used for learning.

In Chapter 3, I present behavioural evidence that the fruitfly *Drosophila melanogaster* may use this innate error signal to adapt their speed controller after wing damage, supporting the phenomenological control architecture. Chapter 4 proposes the mechanism by which insect feedforward controllers could be adapted, and shows the implementation of this method on a drone platform. Lastly, Chapter 5 describes a method for the phenomenological control architecture to cope with systems with multiple distinct time delays.

1.2 Control systems

In its simplest form, a control system is what goes in between actuators and sensors. For example, a control system that makes a drone steadily hover could receive information on the angle, velocity, acceleration, altitude, and position of the drone, the control system then determines what voltages to apply to the motors in order to keep the drone steadily hovering. In robotics and animals there is another component, some kind of decision making, meaning the controller sits between the actuators, sensors, and decision on what state to be in. This is called a *desired state* or *reference signal* and can consist of desired positions, velocities, angles, temperatures, volumes, or anything else that can be controlled (Ogata et al. 2010).

There are three main considerations in controller design: 1) reference tracking performance, 2) disturbance rejection performance, and 3) robustness. 1) The controller should ensure the actual state tracks the desired state (or reference signal) as well as possible. 2) If there is an external disturbance or perturbation to the system, i.e. a gust of wind tilts the hovering drone, the controller should actuate to return the drone to position and mitigate the error caused by the disturbance. Finally, robustness (3) is a measure of controller performance when the controller was designed around different dynamics to what it is controlling (Ogata et al. 2010). I will distinguish between *stability robustness* and *performance robustness* where stability robustness is a measure of the range of different dynamics a particular controller can stabilise, and performance robustness is how the controller's performance changes even if it remains

stable.

1.2.1 Feedback control

The most common method for control is based on feedback. Specifically, proportional integral derivative (PID) control makes up more than 90% in industrial machinery applications (Araki 2009). In almost all feedback control the error is calculated as the difference between the desired and actual state, and some form of feedback controller calculates an actuation command based on that error. For example, in PID the actuation command is calculated by summing the error, the integral of the error and the derivative of the error, where each of the three signals are multiplied by a gain (Eq. 1.1)

$$u(t) = k_p e(t) + k_d \frac{de(t)}{dt} + k_i \int_0^t e(t) dt \quad (1.1)$$

where $u(t)$ is the actuation command and $e(t)$ is the error signal between the desired and actual state.

1.2.2 Feedforward control

Feedback control can be combined with feedforward control to make a two-degree-of-freedom (2DoF) controller (Araki and Taguchi 2003). Feedforward components can be used to supplement the actuation of the feedback controller when the desired state changes. This directly improves the reference tracking performance and allows the feedback controller to be optimised for improved disturbance rejection and/or robustness (Araki and Taguchi 2003).

Feedforward control can also be used directly to improve disturbance rejection performance, as opposed to indirectly by allowing the feedback component to be optimised differently (Zhong, Pao, and Callafon 2012). To do this, the disturbance must be observable, with an auxiliary sensor or with an observer. In the drone example, an auxiliary sensor could be a wind sensor. A wind sensor cannot be used in feedback to control the position of the drone, but sensing a sudden gust of wind could allow a feedforward controller to compensate for the disturbance before the error is introduced. Whereas, feedback control requires an error before it generates

an actuation command (eq. 1.1).

Araïn and Kendoul (2014) implemented a feedforward disturbance rejection method on a drone with pitot tube wind sensors. They found that using this method reduced the wind-induced error by up to 65% (Araïn and Kendoul 2014).

Lastly, feedforward control for disturbance rejection can use an observer instead of auxiliary sensors. A gust of wind predictably affects a drone’s state, allowing a mathematical model to observe gusts of wind and a feedforward controller to preemptively respond (Azid et al. 2023). Recent deep reinforcement learning models that observe wind gusts without any additional sensors have also been able to reduce the wind-induced error by 65% (Ma et al. 2024).

1.2.3 Other classical control methods

There are many different control methods; however, most can be considered variations of a feedback controller. H-infinity (Gershon, Shaked, and Yaesh 2005), robust (Green and Limebeer 2012a), fuzzy-logic (Berenji 1992), and linear-quadratic regulators (LQR) (Ogata et al. 2010), can generally be categorised as feedback controllers, where each method refers to a way of choosing feedback gains or determining the actuation command based on the error. However, there are two additional classic control methods that are structurally different, model predictive control (MPC), and smith-predictor control (SP).

The smith predictor

Smith predictor control is a method for dealing with time delays in system dynamics or sensors (Mirzal 2017). It uses a model of the dynamics without a delay to generate the actuation command using feedback, and sometimes feedforward control (Mohammadzaheri and Tafreshi 2017). When there is an updated measurement from the sensors, the difference between the model with the delay and the sensor is used in feedback (Ch. 2, Fig. 2.1b).

The structure of the SP, if the model of the dynamics is exact and the feedback controller is unchanging, allows the reference tracking performance to be independent of the time delay in the dynamics or sensors. However, the robustness and disturbance rejection performance are not independent of the time delay, as they require feedback. Grimholt and Skogestad (2019)

did a comparison between standard feedback controllers and SPs using four different first-order systems with time delays. They fixed the robustness for the two controllers, using the peak sensitivity metric¹. Due to the additional complexity of the SP, in their study, they restrict the SP to only using a PI controller to make it a fair comparison with a PID feedback controller. The results showed that the performance of a PID controller was better than the SP (using PI control only) for a fixed robustness (Grimholt and Skogestad 2019). Even without the restriction on the SP, it is well established that SPs are generally less robust than feedback controllers alone (Adam, Latchman, and Crisalle 2000; Palmor 1980).

Smith predictors are used in Chapter 2 as a benchmark for determining the advantages and disadvantages of the insect-inspired control method.

Model predictive control

The model predictive controller also works fundamentally differently to any of the controllers mentioned. Model predictive control (MPC) optimises control inputs by solving a constrained optimisation problem over a finite prediction horizon. Unlike traditional feedback controllers, MPC predicts future system states using a model and determines optimal control actions to minimise a cost function while satisfying constraints. This makes MPC particularly effective for systems with constraints on states and inputs (Kouvaritakis and Cannon 2016). The dynamics of a system in MPC are typically described in a discrete-time state-space form:

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + B\mathbf{u}_k, \quad (1.2)$$

$$\mathbf{y}_k = C\mathbf{x}_k + D\mathbf{u}_k, \quad (1.3)$$

where $\mathbf{x}_k$ represents the state vector, $\mathbf{u}_k$ is the control input, and $\mathbf{y}_k$ is the system output at time step k . The matrices A , B , C , and D define the system dynamics.

To implement MPC, predictions over a finite time horizon (N time steps) are made by stacking multiple future states into a single matrix equation. Using the state-space model

¹The sensitivity function in control measures how much variations in system dynamics affect the closed-loop properties of the system and controller.

recursively, the future states can be written as:

$$\mathbf{X} = \mathcal{A}\mathbf{x} + \mathcal{B}\mathbf{U}, \quad (1.4)$$

where:

$$\mathcal{A} = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix}, \quad \mathcal{B} = \begin{bmatrix} B & 0 & \cdots & 0 \\ AB & B & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N-1}B & A^{N-2}B & \cdots & B \end{bmatrix}, \quad (1.5)$$

$\mathbf{X}$ and $\mathbf{U}$ contain the stacked state and input vectors, respectively. The control inputs $\mathbf{U}$ are obtained by solving an optimisation problem that minimises a cost function which can take many different forms, but is usually a quadratic function of controller error and actuation effort. The first control input from the optimised sequence is applied to the system, and the process repeats at the next time step.

A key advantage of MPC is its ability to handle state and input constraints explicitly. Unlike classical controllers, which require tuning and simulations to ensure constraints will not be broken, MPC incorporates the constraints within the optimisation framework. Typical constraints include upper and lower bounds on control inputs, such as actuator saturation limits, as well as state constraints, which can represent physical limitations like velocity or position limits in a robotic system (Camacho et al. 2007). Furthermore, MPCs can impose terminal constraints to ensure closed-loop stability, which are often designed based on Lyapunov functions or invariant sets (Rawlings et al. 2008).

Beyond linear MPC, significant efforts have been made in robust (Campo and Morari 1987) and non-linear (Allgower, Findeisen, Nagy, et al. 2004; Allgöwer and Zheng 2012) extensions to handle model uncertainties and system non-linearities. Robust MPC formulations account for worst-case disturbances or parameter variations by incorporating min-max or tube-based approaches, ensuring constraint satisfaction under uncertainty (Campo and Morari 1987). Nonlinear MPC (NMPC), on the other hand, extends the framework to systems governed by nonlinear dynamics. Unlike linear MPC, NMPC requires solving a nonlinear optimisation

problem at each time step, which can be computationally expensive but provides superior performance for highly dynamic and complex systems such as autonomous vehicles and robotic manipulators (Allgöwer and Zheng 2012). Advances in real-time optimisation algorithms and parallel computing have significantly improved the feasibility of NMPC in practical applications (Manenti 2011).

A standard implementation of MPC is used in Chapter 5 as a benchmark for multi-input-multi-output systems with multiple different time delays.

1.2.4 Deep-learning based control methods

Another different, but non-classical, control method is deep learning, where some form of neural network is trained to estimate the actuation command for given sensory and reference information. End-to-end neural network/deep-learning control methods require large training data sets, particularly for time-varying dynamics. This training data is not trivial to obtain since the ideal actuation command, which is needed to train the network, is not available. One method is to monitor how humans control systems and to use recorded observations as training data, as proposed for driverless cars (Chen and Huang 2017). However, these methods are limited in performance by the ability of humans and are not viable for robots that humans cannot control, like hexapods. Thus, reinforcement learning (RL) has emerged as a better solution, where agents learn how to maximise rewards based on the consequences of their actions. RL methods have now outperformed the best drone racers (Kaufmann et al. 2023) and Gran Turismo drivers (Wurman et al. 2022). However, in both instances, large amounts of time in advanced simulated environments are required for learning, and neither controller can adapt to changes in dynamics.

Reinforcement learning (RL) methods, despite their successes, face several challenges that limit their practicality in real-world control applications. One major challenge is the lack of formal guarantees on stability and robustness (Wang and Hong 2020). Unlike MPC, which explicitly enforces constraints through optimisation, RL agents learn purely from experience, meaning there is no assurance that they will behave safely in untested scenarios. This is particularly problematic in safety-critical systems where unpredictable behaviour in rare or edge

cases could lead to failure (Tipaldi, Iervolino, and Massenio 2022). Furthermore, RL policies are often trained in controlled simulations that do not fully capture real-world complexities, leading to difficulties when transferring the learned policies to physical systems. This simulation-to-reality gap can cause poor performance or unexpected behaviours when deployed on actual hardware (Salvato et al. 2021). Although it is possible to remove this gap by training on the hardware, this often takes a significant amount of time before it outperforms other simpler methods, and again is usually not possible in safety-critical or fragile systems. For instance, a bipedal robotic walker with two motors at the hip fell hundreds of times before learning to walk (Schuitema et al. 2010). To avoid this, specialised accelerated methods for hardware have been implemented (Shao et al. 2018), and transfer learning methods have been developed in which systems learn in simulation and make small adjustments on the hardware (Barrett, Taylor, and Stone 2010).

A further challenge with reinforcement learning (RL) in control applications is adapting to changes in system dynamics, as RL controllers are generally trained offline in a fixed simulation. Some efforts have been made to develop adaptive RL methods, such as meta-reinforcement learning, where agents are trained to learn new tasks quickly by leveraging past experiences (learning to learn; Gupta et al. 2018), and online RL, where policies are continuously updated as new data is collected (Shao et al. 2018). However, these approaches face significant difficulties, including slow adaptation times and the risk of catastrophic forgetting, where new experiences overwrite previously learned behaviours, leading to instability (Cahill 2011). In addition, updating RL policies in real-time can require significant computational resources, which can be impractical for high-speed control systems like drones or robotic manipulators; for this reason, in those high-speed systems, it is more common to see reinforcement learning used for path planning and navigation rather than flight control (Hong et al. 2021). While progress has been made in combining RL with elements of adaptive control, achieving fast, reliable adaptation in complex, dynamic environments remains an open research challenge (Kormushev, Calinon, and Caldwell 2013; Zhang and Mo 2021).

1.2.5 Adaptive control

Control schemes for robotics must maintain good performance in case of time-varying dynamics, and ideally, without complex simulations of every possible situation. Classical control offers three methods to do this. First, deriving a robust controller such that the closed-loop system meets certain criteria over a maximised range of possible dynamics (Dorato 1987; Green and Limebeer 2012b; Zhou and Doyle 1998). Second, gain scheduling of control gains for different dynamic regions (Rugh 1990; Fromion and Scorletti 2003). Thirdly, adaptive control, where controller parameters are adjusted in real-time to reduce control error.

One approach is feedback error learning (FEL), a method for adapting a feedforward controller when system dynamics change, originally hypothesised as a cerebellum learning mechanism (Miyamoto et al. 1988). FEL combines a feedback controller with an adaptive feedforward component that learns from the feedback control signal to gradually improve performance (Kawato 1990). Adaptation is typically achieved using estimates of the sensitivity derivatives (or system Jacobian), which provide gradient information to adjust the feedforward control signal and reduce error. This method has largely been applied in robotic applications requiring rapid adaptation to dynamic changes, such as robotic arms and legged locomotion (Casellato et al. 2012).

Incremental nonlinear dynamic inversion (INDI) is another adaptive control strategy that extends classical dynamic inversion by using real-time measurements to update control inputs. Traditional nonlinear dynamic inversion (NDI) relies on an accurate model of the system to cancel out nonlinearities, but inaccuracies in the model can degrade performance (Smith 1998). INDI overcomes this by using incremental control updates based on measured accelerations and system responses rather than relying solely on a predefined model (Sieberling, Chu, and Mulder 2010). The use of INDI is particularly prevalent in drone flight control, where aerodynamic effects are difficult to model precisely (Smeur, Chu, and De Croon 2016; Di Francesco, D’Amato, and Mattei 2015).

Model Reference Adaptive Control (MRAC) is another fundamental adaptive control method that ensures that a system follows a desired reference model by continuously adjusting controller parameters. The adaptation is typically based on Lyapunov stability theory, ensuring that

tracking errors converge to zero over time (Nguyen 2018). MRAC has been widely applied in aerospace and robotic systems where dynamic parameters change due to external forces, payload variations, or environmental conditions (Zhang and Wei 2017). However, MRAC can be sensitive to high adaptation rates, which can lead to instability, and often requires additional techniques such as robust adaptation laws or parameter projection methods to ensure safe operation (Lavretsky and Wise 2012). In addition, all of these methods require some prior knowledge of the system dynamics being controlled which can limit their flexibility when faced with significantly varying system dynamics.

1.2.6 Summary

There are many different classic control methods that offer accuracy, robustness, and good disturbance rejection for almost all different dynamics. Additional adaptive control methods extend these approaches to cope with time-varying system dynamics. However, these methods require some prior knowledge of the system being controlled, which can often restrict their flexibility. In contrast, reinforcement learning methods can learn complex system dynamics without prior knowledge. However, their large models require significant time in simulations or learning the dynamics in hardware. In addition, making them adapt quickly is challenging, and they do not provide any safety guarantees.

1.3 Control in insects

In this thesis, I identify aspects of insect control systems that could be applied to robotics. Insects are adaptable, robust, and, compared to other animals, easier to understand and experiment with. However, they operate under different constraints compared to typical robots. Understanding these differences is essential when aiming to transfer insect-inspired control methods to robotics. A major difference is the sensing ability of insects, given the significant energy constraints and the structure of the neuronal processing (Laughlin 2010).

1.3.1 Insect sensors

Mechanosensors

Insects use many more mechanosensors than robots. Fabian, Siwanowicz, et al. (2021) found that there were over 3000 sensors across the four wings of the small dragonfly *Perithemis tenera*, excluding all the sensors at the base of the wing. It is not fully understood what these sensors do, and how they are used in flight control. However, it is known that there are different types that respond to different inputs. Campaniform sensilla respond to strain in the wing structure and can be found: 1) alone around the edge of the wing, likely for measuring wing twist, 2) within some associated structure, or 3) in a field/cluster in specific locations (Fabian, Siwanowicz, et al. 2021). There are also flow sensors on the wings, which are thought to sense air flow (Uhrhan, Bompfrey, and Lin 2024), although there is no physiological evidence to support this.

On the head, body, and limbs there are yet more mechanosensors. Two, antennae and halteres, play an important role in flight control. The antennae have multiple functions, they are highly sensitive olfactory sensors that contain a high density of odorant binding proteins (Taylor and Krapp 2007; Hansson and Stensmyr 2011). In addition, crickets (Ifere et al. 2022) and tiger beetles (Zurek and Gilbert 2014) use their antennae to detect and examine obstacles. However, most notably for flight control, at the base of the antennae is the Johnston organ which responds to airflow on the antennae (Fuller et al. 2014). This has an important role in the speed control of flying insects and will be discussed in further detail in Chapter 3.

Halteres are mechanosensory organs unique to dipteran (two-winged) flies, that have evolved from the hindwings of ancestral insects (Yarger and Fox 2016). Over evolutionary time, these hindwings reduced to small, club-shaped structures that oscillate out of phase with the wingbeat. Their oscillations are driven by an independent neural oscillator that is weakly coupled to the wingbeat rhythm (Deora, Sane, and Sane 2021). Classic work by Fraenkel (1939) demonstrated that removal of the halteres rendered flies unstable and unable to maintain flight. However, stability could be restored by attaching small masses, such as bits of cotton wool, to the haltere bases, suggesting the primary role of the halteres is to provide inertial feedback.

Indeed, the primary function of the halteres is to sense body rotations through Coriolis

forces generated during angular acceleration (Yarger and Fox 2016; Dickinson 1999). This mechanosensory feedback provides rapid information about angular velocity and acceleration, which is critical for stabilising flight in the presence of dynamically unstable modes, common in insect flight dynamics (Ristroph et al. 2013). A comprehensive mathematical description of the forces and orientation dynamics experienced by the oscillating halteres is provided by Nalbach (1993).

Visual sensors

Insects have compound eyes consisting of many individual ommatidia which each contain a cornea, lens, and some photoreceptor cells. The density of ommatidia varies significantly across species (Land 1997), and the location on the eye (Kirschfeld 1976). *Drosophila* have around 750 ommatidia (Mishra and Knust 2013), whereas some dragonflies have up to 30,000 (Sherk 1978).

Whilst, in the case of *Drosophila* they have limited spatial resolution, they have significantly higher temporal resolution than humans, 200 Hz and 50 Hz flicker fusion frequency, respectively (Heisenberg and Wolf 2013; Mankowska et al. 2021). This is advantageous for using vision to sense motion, one of the major functions of the fly’s eyes (Borst and Egelhaaf 1989).

Flies perform a spatio-temporal correlation of local luminance levels to visually estimate their self-motion (Reichardt 1957) and the detection of looming objects (Card and Dickinson 2008b). The neuronal mechanisms by which this is done have been uncovered in the last decade (for a review, see Borst and Groschner 2023). However, long before that Reichardt (1957) described a phenomenological model (Fig. 1.1a) that captured most of the important properties of the fly motion detector (Hassenstein and Reichardt 1951; Hassenstein and Reichardt 1953).

The Reichardt detector works by measuring the luminance at two spatial locations on the eye (a_1 and a_2 in Fig. 1.1a), separated by $\Delta\phi$. The luminance levels are delayed with a temporal low-pass filter and multiplied by their undelayed counterpart from the other sampling location. The two outputs are subtracted meaning that a change in overall luminance does not induce any output; only when there is a spatio-temporal correlation will an output be produced. 70% of motion detectors in *Calliphora* use neighbouring ommatidia, meaning $\Delta\phi$ usually equals the inter-ommatidial angle (Buchner 1976; Schuling et al. 1989). The low-pass filters have been

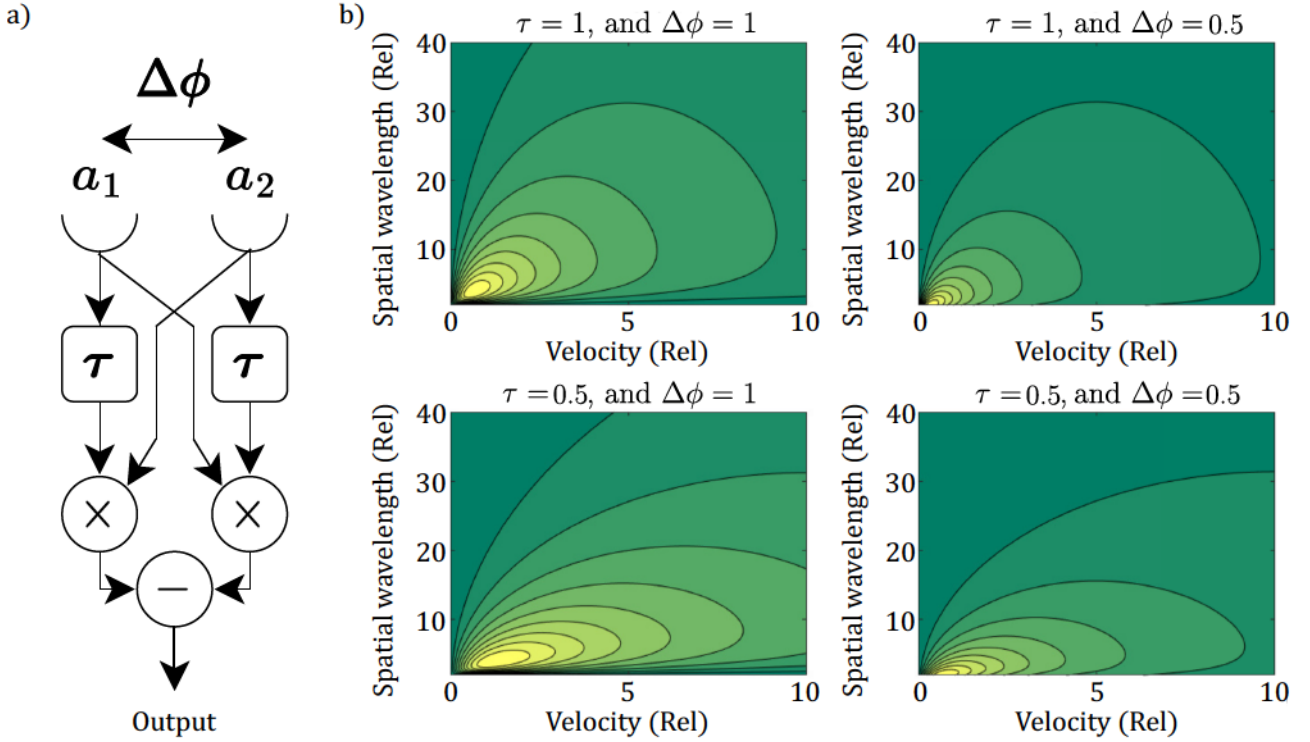


Figure 1.1: Adapted from Campbell, Lin, and Krapp (2023b) (a) Single Reichardt detector model parameterised by τ which is the time constant in a temporal low-pass filter; and $\Delta\phi$ which is the spatial sampling base, often referring to the angle between the optical axes of neighbouring ommatidia in the eye lattice. (b) Mean response of four differently parameterised simulated Reichardt detectors. The units have no absolute values; instead, the spatial wavelength is given relative to a single unit of sampling base, and velocity relative to a single unit of time constant. Hence, the values of the response have been omitted; yellow represents a large response, and dark green is a small response.

shown to have adaptable time constants (Maddess and Laughlin 1985), and crabs combine three channels of motion detectors tuned to different temporal frequencies (Nalbach 1989). Changing the parameters τ and $\Delta\phi$ affects the spatial frequency and velocity that maximally excite the detector (Fig. 1.1b).

Whilst the Reichardt detector is computationally efficient, robust to light conditions, and adaptable, it does have significant drawbacks. Firstly, it does not respond linearly with speed. There have been efforts to show how speed could be measured using variations/combinations of the Reichardt detector (Zanker, Srinivasan, and Egelhaaf 1999; Cope et al. 2016; Campbell, Lin, and Krapp 2023b). Despite this, it appears the spatial frequency has a large impact on animal behaviour (Kirchner and Srinivasan 1989). The second drawback is that each detector does not produce a smooth output and has large modulations around the average response due to local

contrast changes. The modulations can be reduced by combining the output of many detectors, but are still significant (Meyer, Lindemann, and Egelhaaf 2011).

Another property of these Reichardt detectors is that they are directionally selective, meaning they respond preferentially to motion in a specific direction. In the fly visual system, many such detectors are arranged across the visual field, each tuned to a different direction of motion. Together, they form spatially distributed vector fields called receptive field maps that function like matched filters (Krapp and Hengstenberg 1996; Taylor and Krapp 2007). This allows the fly to have specific cells that respond to relevant self-motion dynamics like roll, pitch and yaw.

Krapp, Taylor, and Humbert (2012) hypothesised that these receptive field maps would be organised to maximise a metric of controller observability². However, recent work found that the receptive field maps maximise energy transfer from actuator to sensor, as opposed to observability, which can be thought of as state to sensor energy transfer (Humbert et al. 2024).

1.3.2 The inner-outer-loop hypothesis

Although the motion vision pathway does not directly measure velocity, visual estimates of self-motion provide feedback for controlling various aspects of flight (McCann and MacGinitie 1965; Götz 1968). The optomotor response ensures that when the visual scene moves differently than expected, the fly adjusts its movements to minimise the discrepancy. For example, a tethered fly presented with a vertically oriented grating pattern moving horizontally in the frontal visual field will yaw in the corresponding direction to reduce visual slip (Cellini, Ferrero, and Mongeau 2024). Similar mechanisms regulate speed in a tunnel with moving wall gratings (Fry, Rohrseitz, et al. 2009) and pitch and roll adjustments in a setup that allows control of the rotation axis (Sherman and Dickinson 2004). These stabilising visual reactions can be thought of as inner-loop, feedback, reflexes (green in Fig. 1.2). When visual sensors detect an unanticipated movement of the world around them, the visual feedback pathway initiates motor commands to mitigate divergence from the expected.

What happens when a fly aligns to polarisation in the sky (Velez et al. 2014), chases a mate

²The observability measure used by the authors quantifies how easily a specific mode can be detected from sensor measurements and is described in more detail in Tarokh (1992).

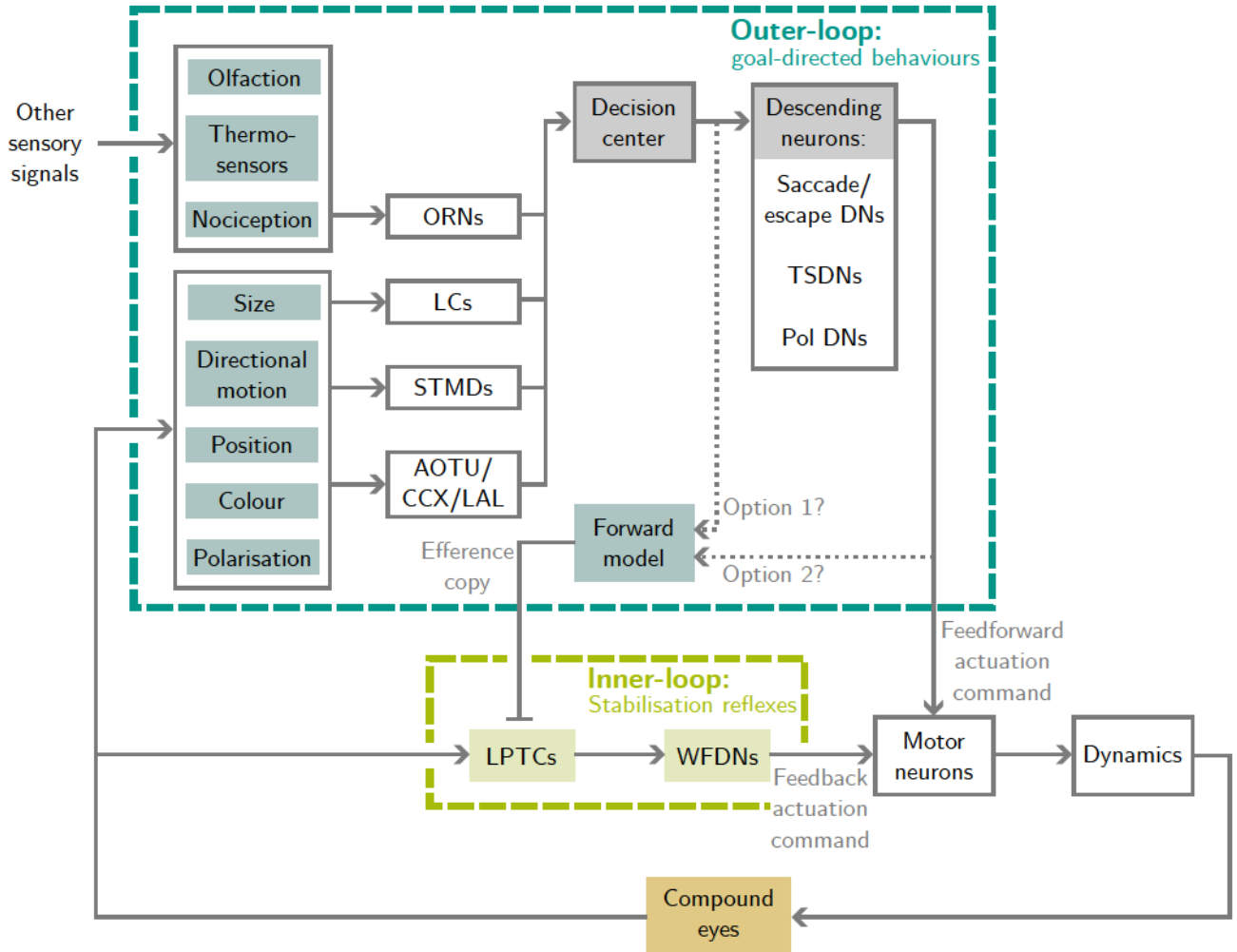


Figure 1.2: Inner-outer-loop hypothesis focused on the visual pathways. The inner loop functions as a stabilising feedback pathway, where lobula plate tangential cells (LPTCs) process wide-field optic flow in specific directions. Wide-field descending neurons (WFDNs) integrate the output of LPTCs and transmit them to motor neurons, acting as the visual feedback controller. The outer loop governs goal-directed behaviours. The compound eyes detect additional visual cues relevant to these behaviours, processed by specialised neural structures such as looming cells (LCs) and small target motion detectors (STMDs). Navigation-related processing occurs in the anterior optic tubercle (AOTU), central complex (CCX), and lateral accessory lobe (LAL). The outputs of these structures are assumed to contribute to a decision center that determines the appropriate behavioural response. Once a behaviour is initiated, corresponding descending neurons—such as target-selective descending neurons (TSDNs)—provide actuation commands to the motor neurons. Another crucial role of the outer loop is to prevent the inner loop from counteracting goal-directed movements. This is thought to involve a forward model that predicts the sensory consequences of intended actions. While the origin of this predictive input remains unknown, electrophysiological evidence indicates that the LPTCs receive such cancellation signal, thus avoiding the counteracting of self-induced motion. Finally, other sensory cues, like the olfactory inputs processed by olfactory response neurons (ORNs), are also likely to contribute to initiating goal-directed behaviours.

or prey (Fabian, Sumner, et al. 2018), tries to escape from a predator (Hammond and O’Shea

2007), follows an odour plume (Murlis, Elkinton, Carde, et al. 1992), or escapes pain (Neely et al. 2011)? I.e, what happens when a fly engages in goal-directed behaviours?

If insects used feedback control, the reference or desired state would change, and the same feedback controller that responds to disturbances would provide actuation commands to enact the goal-directed behaviours. However, Kim, Fitzgerald, and Maimon (2015) showed that in sensory deprivation, voluntary yaw movements of *Drosophila* resulted in an inhibition of the horizontal system north (HSN) cell that responds to horizontal optic-flow. This suggests that flies were preemptively suppressing the expected sensor response due to voluntary goal-directed behaviours. If, instead, flies moved purely via updating the desired state of a feedback controller, they would have actuated until the visual feedback error was eliminated. Under sensory deprivation, this error would never reach zero, leading the fly to turn indefinitely in the absence of corrective visual input. These results strongly supported the long-held hypothesis that biological systems differentiate between the sensory consequences of voluntary movements, termed *reafference*, and sensory signals due to external disturbances, termed *exafference* (Von Helmholtz 1867; Von Holst and Mittelstaedt 1950; Sperry 1950; Collett 1980).

This cancellation signal is called an efference copy³ and is likely generated by a forward model. Although there has not yet been direct evidence for a forward model, some signal processing is likely needed for appropriate sensory cancellation, which is effectively the forward model (Webb 2004b). This efference copy stops the inner-loop stabilisation reflexes from counteracting goal-directed movements (Kim, Fitzgerald, and Maimon 2015; Krapp 2015). Cancelling the expected sensor response is the first role of the outer-loop (Fig. 1.2, Krapp and Wicklein 2008; Collett 1980). The second role of the outer-loop is to provide the actuation commands to enact these goal-directed behaviours. Cases like saccading, escaping, taking off, and manoeuvring to find the source of an odorous plume are very likely to be enacted by feedforward control, as there is no direct feedback sensing pathway to use. However, for navigating relative to the polarisation pattern in the sky (Webb and Wystrach 2016) and pursuing a target, it is more ambiguous as they are somewhat dependent on direct feedback mechanisms. Indeed, target pursuit is frequently modelled as a feedback controller (Fabian, Sumner, et al. 2018). It leads us to a

³Whilst the term ‘efference copy’ originates from ‘motor copy’, in this thesis I will use the term to describe the cancellation signal that occurs at the level of the sensor.

subtle distinction in the role of the decision center⁴ block in Figure 1.2: it could use outer-loop sensing pathways to (i) determine a desired state which is actuated with feedforward control and efference copies, or (ii) activate specific feedback controllers whilst suppressing stabilising feedback controllers with efference copies. Most of the work in this thesis is based on option (i). Even for target pursuit, which has specific sensing pathways such as STMDs, Mischiati et al. (2015) suggests that dragonflies use internal predictive models of moving prey. Additionally, in housefly dogfighting, body saccades were enacted to point towards the target with almost no delay (Fabian and Lin 2023). These examples suggest that in target tracking/pursuit, switching to a different feedback controller is unlikely. Nevertheless, the assumption of option (i) is important context for interpreting the results in this thesis.

Figure 1.2 illustrates inner- and outer-loop control in flight, but similar behaviours exist in other contexts. For example, butterflies and moths (*Lepidoptera*) use a specialised mouthpart called a proboscis, which functions like a straw to drink nectar and other liquids. The hawkmoth exhibits an outer-loop behaviour by sweeping its proboscis across a flower to locate access to nectar, a strategy that can improve through learning (Deora, Ahmed, et al. 2021; Campos, Bradshaw Jr, and Daniel 2015). This is despite the existence of mechanosensory feedback pathways that allow the moth to track the movement of a flower caused by wind, resembling an inner-loop reflex (Stöckl et al. 2017).

Lastly, Figure 1.2 omits the roles of the halteres and antennae in flight control. Both are known to provide critical sensory feedback (Namiki et al. 2022; Fuller et al. 2014), but how they fit into the inner-outer-loop framework, particularly whether they are subject to efference copy or predictive cancellation of expected sensory input, remains unclear (halteres in flies, Chan, Prete, and Dickinson 1998; antennae of stick insects, Lepreux 2020). For example, the antennae are involved in speed regulation. But it remains unknown during take-off, when the animal transitions from rest to forward motion, whether the mechanosensory feedback from the antennae is simply used as an error signal, whether it is pre-emptively cancelled using an efference copy, or whether a set-point is adjusted.

⁴The term ‘decision center’ is taken from Von Holst and Mittelstaedt (1950) and Kim, Fitzgerald, and Maimon (2015), it is not intended to confirm or deny any level of consciousness. The same applies for the term ‘voluntary’.

Adaptive control in the inner-outer-loop framework

Adaptability is an important characteristic of biological control systems. The fly’s visual sensor systems adapt to contrast (Harris, O’Carroll, and Laughlin 2000), luminance (Laughlin and Hardie 1978), and adjust depending on the locomotive state of the animal, i.e. whether it is walking or flying (Jung, Borst, and Haag 2011a; Rosner, Egelhaaf, and Warzecha 2010; Chiappe et al. 2010; Niell and Stryker 2010; reviewed in: Longden and Krapp 2011). The inner-outer-loop hypothesis provides a useful framework for understanding adaptability in insect control systems, particularly because of the discrepancy between flexible goal-directed behaviours and robust stabilisation mechanisms.

There are two behaviours in flies that illustrate the discrepancy in adaptability between inner- and outer-loop control: (i) the wide-field optomotor response and (ii) the tendency to orient towards a visual bar. (i) The optomotor response is an inner-loop reflex in which flies attempt to stabilise their orientation relative to wide-field motion, such as a moving grating. (ii) When presented with a single visual bar whose angular position is controlled in closed-loop feedback by the fly, the fly actively attempts to keep the bar positioned in its frontal visual field. This is a goal-directed behaviour, often referred to as fixation, and can be considered an outer-loop control strategy.

Several studies have highlighted a key difference between these two behaviours in terms of their capacity for adaptation. To examine the optomotor response (behaviour i), Cellini, Ferrero, and Mongeau (2024) tethered *Drosophila* in an arena where they could manipulate the closed-loop feedback gain between yaw rotations and wide-field visual motion. When the authors changed the feedback gain in their augmented reality setup, they found that there was no change to the optomotor response (Cellini, Ferrero, and Mongeau 2024). This suggests that the optomotor response is time invariant and has no capacity for adaptation. In contrast, fixation to maintain a bar in the frontal visual field (behaviour ii) has been shown to be adaptive. Instead of wide-field optic flow, Heisenberg and Wolf (1984) used the measured yaw torque of a tethered *Drosophila* to adjust the position of a bar in closed-loop. They found that it could adapt to a complete inversion of the feedback gain to keep the bar in the frontal visual field (Heisenberg and Wolf 1984). Further, Wolf, Voss, et al. (1992) later showed that *Drosophila* could adapt

their controller to fixate the bar in their frontal visual field when the force from their leg was used to control the position of the bar (Wolf, Voss, et al. 1992)⁵. This suggests a highly flexible time-varying controller for goal-directed behaviours. These observations suggest that outer-loop behaviours in *Drosophila* are flexible and can be modified through experience and learning to achieve a goal, whereas inner-loop stabilising reflexes are governed by robust controllers that prioritise stability and immediate error correction over adaptability. This observation is revisited in Chapter 3, suggesting that this could be due to the use of adaptive feedforward control with an inflexible feedback controller.

⁵The lack of replication of these results is addressed in the [Discussion](#).

Chapter 2

The advantages of efference copies

As outlined in Section 1.3.2, biological systems use separate pathways to control stabilisation reflexes and goal-oriented behaviours, different from common engineering control methods. One can hypothesise that an inner-loop consists of a stabilising controller that compensates for any perceived change in state as a result of external perturbations (Fenk, Kim, and Maimon 2021). An outer-loop then uses feedforward control to actuate desired changes in state, and an efference copy is used to cancel the expected sensor response (Kim, Fitzgerald, and Maimon 2015; Hardcastle and Krapp 2016). In this Chapter, I formulate the efference copy-based inner-outer-loop hypothesis as a control diagram. I compare the control performance with a Smith predictor (SP, Fig. 2.1b) and a pure feedback controller (PFB, Fig. 2.1a) to identify the architecture's advantages. The PFB and SP architectures were chosen because they are fundamentally different control methods to the inner-outer-loop (unlike 2DoF), and are combinations of temporal filters making them possible to implement in the nervous system (unlike MPC).

Here, the term forward model is used to describe a function, or filter, that maps a desired state change to an expected sensor response (Webb 2004c; Webb 2004b). It is not an internal representation of the world or an understanding of what will happen when certain decisions are made, as Mischiati et al. (2015) describes. It is also different from the concept of a forward model as described by Wolpert (Wolpert, Miall, and Kawato 1998; Wolpert and Kawato 1998), where the forward model refers to a model of the dynamics of the system. Whilst there are many hypotheses for cerebellum control schemes, the most typical uses a model of the dynamics of the

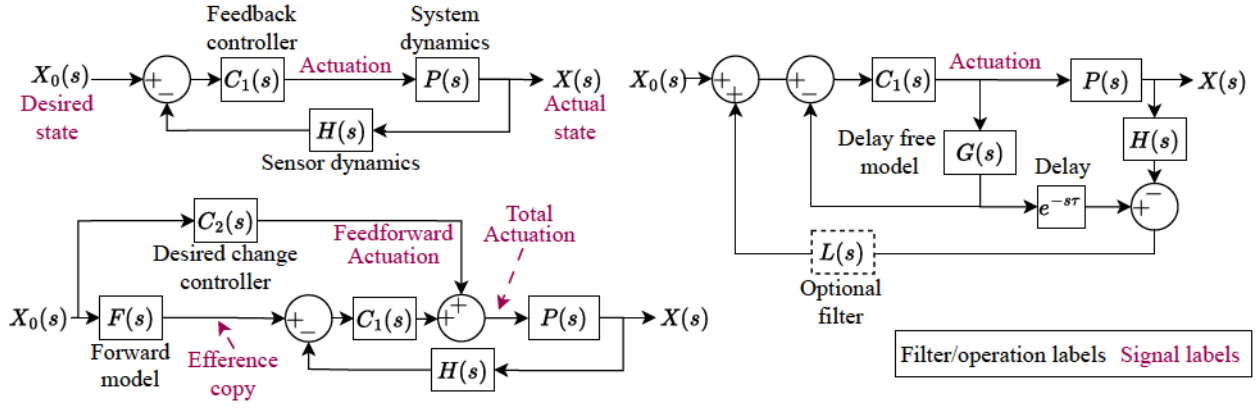


Figure 2.1: Three benchmarked control architectures. Blocks include Laplace transforms of the respective dynamics. Signals labelled “actuation” refers to actuation commands, the actuator dynamics are included in $P(s)$. (a) Pure feedback controller (PFB): the state is measured by sensors with some dynamics, this is subtracted from the desired state to give an error. The feedback controller, $C_1(s)$, generates an actuation command from the error. (b) Smith predictor (SP): a feedback controller, $C_1(s)$, controls an undelayed plant model and sensor dynamics ($G(s)$), to generate the command for the real system $P(s)$. The optional filter is discussed in Sec. 2.3.2. (c) FSDoF controller: the feedforward, desired change controller, $C_2(s)$, generates the motor command to actuate the desired state. The desired state is also used by a forward model to predict the sensory consequences and generate an efference copy. That efference copy signal cancels the expected sensor response. The feedback controller, $C_1(s)$, generates an output only if the sensor response and efference copy do not match.

system to yield an inverse model which can be used to generate a motor command (Wolpert, Miall, and Kawato 1998). Here, I refer to a forward model as a filter that predicts a sensor response from a desired state change. It is therefore not just a model of the dynamics, but a filter that models the insect controller, the dynamics, and the sensor properties. This form corresponds to option 1 in Figure 1.2 (Sec. 1.3.2), the implications of option 2 for the controller are discussed in Section 2.4.

The second aspect of the outer-loop is the generation of the actuation command to execute the desired state change or desired movement. This feedforward controller is another function, or filter, that maps a desired state change into a motor command resulting in muscle contraction. Mostly in engineering control theory, feedforward controllers are used to supplement the actuation of the feedback controller when changing the desired state, referred to as a two-degree-of-freedom (2DoF) controller (Araki and Taguchi 2003). I differentiate this type of feedforward control from what is hypothesised in the insect sensorimotor system by name and by a simple principle. I refer to the feedforward controller in the insect sensorimotor system as a desired change

controller; where the principle difference is that instead of supplementing the actuation of a feedback controller, it provides *all* of the actuation commands needed to change state.

To avoid confusion with other control architectures I will refer to this phenomenological control architecture as a fully-separable-degrees-of-freedom-controller (FSDoF); it is a specific formulation of a 2DoF controller (Araki and Taguchi 2003) where the roles of reference tracking, and disturbance rejection are completely separated. In the first half of this Chapter, I compare the performance of the FSDoF controller (Fig. 2.1c) with two common controllers in engineering to examine its advantages. Then, I evaluate the robustness of the FSDoF control scheme to varying system dynamics, and altered sensor properties. Finally, considering the design constraints of biological systems, I examine equivalent variations of the FSDoF controller with similar advantages.

Previous work has identified a clear advantage of efference copies: distinguishing between exafferent and reafferent sensor signals (Collett 1980; Webb 2004b; Von Holst and Mittelstaedt 1950). This is particularly beneficial for sensing pathways where self-generated signals may interfere with important exafferent information. Electric fish use electroreceptor organs to detect objects, prey, and other fish (Crampton 2019). However, mormyrids, which have self-generated electric fields, cancel reafferent signals to remain sensitive to exafferent sensory information (Bell 1989; Bell and Grant 1989; Crape and Sommer 2008). In addition, male crickets clap their wings to produce bursts of sound. To remain sensitive to hearing other males in the area, they cancel the expected sensor response in their auditory interneurons (Poulet and Hedwig 2002; Webb 2004b). In addition, robotic implementations mimicking cricket phonotaxis have shown that efference copy mechanisms may be the most effective method for tracking a sound source (Payne, Hedwig, and Webb 2010).

This is the first controller comparison involving the biology-inspired FSDoF controller to infer its advantages. The neural control structure has evolved over hundreds of millions of years enabling the integration of stabilisation reflexes and goal-oriented behaviours under severe energy constraints. By relating a control systems analysis of the structure, to the common sensor and dynamic properties of insects, I attempt to answer: what control advantages do biological systems gain from efference copy control methods?

2.1 Formulation of the FSDoF insect sensorimotor control architecture

By far the most widespread and convenient way to examine control architectures is using the complex frequency (Laplace) domain (Ogata et al. 2010). It is a mathematical tool that allows interactions between temporal filters to be treated algebraically, and convolution to be treated as multiplication.

By finding the transfer functions of the closed-loop controlled systems I can quantify three metrics for assessing a control system: reference tracking, disturbance rejection, and robustness (see Sec. 1.2). First, looking at disturbance rejection: the transfer function from a disturbance $U(s)$ to the actual state $X(s)$, is given by

$$\frac{X}{U}(s) = \frac{P_d}{1 + HC_1P} \quad (2.1)$$

where $P_d(s)$ is the transfer function for how the disturbance affects the state. The same equations are obtained for all three architectures in Figure 2.1 (one additional assumption is required for the SP). If all three architectures use the same feedback controller, they will all have approximately the same performance in handling disturbances. This is intuitive, since in order to reject a disturbance it must be sensed, and therefore feedback control must be used to remove its impact.

For performance in tracking the desired state, the three controllers have different transfer functions. The transfer functions for the pure feedback controller (PFB, Fig. 2.1a) and Smith predictor (SP, Fig. 2.1b) are given in equations 2.2 and 2.3, respectively.

$$\frac{X}{X_0}(s) = \frac{C_1P}{1 + HC_1P} \quad (2.2)$$

$$\frac{X}{X_0}(s) = \frac{C_1P}{1 + C_1G + C_1(PH - Ge^{-s\tau})} \quad (2.3)$$

$G(s)$ is an undelayed model of the undelayed system and sensor dynamics, $\bar{P}(s)$ and $\bar{H}(s)$, respectively. Therefore, I assume $G(s)e^{-\tau s} = \bar{P}(s)\bar{H}(s)e^{-\tau s} = P(s)H(s)$, that the estimated

model of the system is perfect. I will break this assumption when I discuss robustness to modelling errors in Section 2.3.4. For now, the resulting transfer function is given by

$$\frac{X}{X_0}(s) = \frac{C_1 P}{1 + C_1 G} \quad (2.4)$$

The difference between the SP and PFB is that the PFB controller is designed around the system and sensor dynamics including the delay, and the SP controller is designed around a delay-free system.

The FSDoF control architecture (Fig. 2.1c) has the single-input-single-output (SISO) transfer function

$$\frac{X}{X_0}(s) = \frac{C_2 P + F C_1 P}{1 + H C_1 P} \quad (2.5)$$

However, the forward model $F(s)$ is not a random filter, it should turn the desired state changes into the expected sensor response. By looking at Figure 2.1c, following the loop from desired state to sensor response, clearly the forward model should predict:

$$F(s) = C_2 P H \quad (2.6)$$

Substituting this into equation 2.5 yields

$$\frac{X}{X_0}(s) = C_2 P \quad (2.7)$$

which corresponds to a quasi-open-loop control. It is not traditional open-loop control because there is a feedback loop. The main disadvantages of *pure* open-loop control do not apply to the FSDoF architecture: the architecture can still compensate for external disturbances and control unstable dynamics.

In summary, for disturbance rejection, the above transfer functions show that all three controllers will have the exact same performance, given the same feedback component $C_1(s)$. This is not the case for the second control metric (ii), reference tracking, where the three architectures are formalised by different equations; the majority of the rest of the chapter will be on simulations showing the differences in performance. Finally (iii), the robustness of the

controller to differences in the real and expected system dynamics will be discussed in Section 2.3.4.

2.2 Methods

To determine the differences in reference tracking performance between the efferece copy-inspired FSDoF controller and the two typical engineering controllers described above, I follow a simple procedure: select some system and sensor dynamics, design all three controllers around controlling those dynamics, and compare the response to a step change in the desired state. The responses of the closed-loop systems were simulated using a discrete approximation (ode45) of continuous time in Simulink MATLAB 2023b.

The controller architectures were tested on 36 different system dynamics. Table 2.1 gives the structures, chosen to represent different common dynamics: 1 and 3 represent low pass filter dynamics, 2 includes an integrator, 4 and 6 include a *zero*, and 5 is an unstable system. Each of the six plant models was parameterised with different bandwidths (ω) and sensor delays (τ). This was to provide a range where either the sensor delay or the apparent delay due to the system dynamics dominated. I used three different sensor delays: 0 ms, 50 ms, and 100 ms. Each with two bandwidths: 10 rad s^{-1} , and 50 rad s^{-1} . This range was selected to approximately reflect the range of bandwidths and delays in the insect control system from system identification results, where the animal is enacting a desired behaviour (Rohrseitz and Fry 2011; Stöckl et al. 2017). The frequency of the *zero* (ω_z) was set to be 30 rad s^{-1} , between 10 rad s^{-1} and 50 rad s^{-1}

Table 2.1: Six different combinations of transfer functions describing system and sensor dynamics $P(s)H(s)$ as shown in Figure 2.1, where $H(s)$ was the time delay, and $P(s)$ was the polynomial transfer function.

1 $\frac{e^{-s\tau}}{\left(\frac{s}{\omega} + 1\right)}$	2 $\frac{e^{-s\tau}}{s \left(\frac{s}{\omega} + 1\right)}$	3 $\frac{e^{-s\tau}}{\left(\frac{s}{\omega} + 1\right)^2}$
4 $\frac{e^{-s\tau} \left(\frac{s}{\omega_z} + 1\right)}{\left(\frac{s}{\omega} + 1\right)}$	5 $\frac{e^{-s\tau}}{\left(\frac{-s}{\omega} + 1\right)}$	6 $\frac{e^{-s\tau} \left(\frac{s}{\omega_z} - 1\right)}{\left(\frac{s}{\omega} + 1\right)}$

meaning the pole would be both dominant and non-dominant. To test each controller on the 36 dynamics, there were 84 different controllers: 36 different PFB and FSDoF controllers, and 12 different SPs since the SP is designed around the undelayed system.

For a fair comparison, the limits of the actuators for each of the three systems must be approximately the same. To ensure this, the maximum allowable actuation of the FSDoF control architecture was not allowed to exceed 1% of the maximum actuation used by the PFB controller. The feedback controllers were all proportional-integral-derivative (PID) controllers. For the PFB and the SP, the feedback controller was configured using MATLAB's optimal PID tuner (Gahinet, Chen, and Eryilmaz 2013) with a design focus on reference-tracking, and a desired phase margin of 45° . This meant there were three optimised parameters (gains) dictating the reference tracking performance for the SP and PFB. In addition, given that the FSDoF has more filters and parameters that can be optimised, the degrees of freedom in the FSDoF were deliberately reduced. By defining specific structures for $F(s)$ and $C_2(s)$, FSDoF's reference tracking was optimised using just one bandwidth parameter, labelled as q . Two conditions should be met for the FSDoF controller in the context of these simulations:

$$F(s) = C_2 P H(s) \quad (2.8)$$

$$\lim_{s \rightarrow 0} F(s) = 1 \quad (2.9)$$

Given that $H(s) = e^{-\tau s}$ in all the simulations, and the plant dynamics were fractions of the form:

$$P(s) = \frac{P_n(s)}{P_d(s)} \quad (2.10)$$

The roots of the numerator polynomial can be separated into stable and unstable zeros.

$$P_n(s) = P_n^s(s) P_n^u(s) \quad (2.11)$$

where $P_n^s(s)$ and $P_n^u(s)$ are the polynomials of all the negative zeros and positive zeros of the

numerator, respectively. Substituting $P(s)$ and $H(s)$ into Eq. 2.8 yields

$$F(s) = C_2 \frac{P_n^s P_n^u}{P_d} e^{-\tau s}. \quad (2.12)$$

P_n^u can not be inverted since it will lead to an unstable pole, therefore it must be in the numerator of $F(s)$. A form of $F(s)$ that satisfies Eq. 2.9 is

$$F(s) = \frac{P_n^u}{a \left(\frac{s}{q} - 1 \right)^2} e^{-\tau s} \quad (2.13)$$

where q is the parameter which is optimised, and

$$a = \lim_{s \rightarrow 0} P_n^u(s) \quad (2.14)$$

Following this, to satisfy Eq. 2.12, the structure for C_2 is

$$C_2(s) = \frac{P_d}{\left(\frac{s}{q} - 1 \right)^2 P_n^s} \quad (2.15)$$

These structures meet the two requirements in Equation 2.8 and 2.9, ensure $F(s)$ and $C_2(s)$ are internally stable, and most importantly the FSDoFs reference tracking performance is controlled by a single variable q . The cases in which the MATLAB PID tuner did not find a stabilising controller for the PFB and SP were not included in the analysis.

An example step response is given in Figure 2.2. The MATLAB optimal gains for the PFB controller were $K_p = 0.86$, $K_d = 0$, and $K_i = 5.66$, for the SP controller the feedback gains were $K_p = 1.72$, $K_d = 0$, and $K_i = 19.59$. The feedback gains in the FSDoF have negligible impact on the reference tracking behaviour. However, to obtain the results in Figure 2.2, a feedback controller was tuned with a design focus on disturbance rejection which had gains of $K_p = 0.70$, $K_d = 0$, and $K_i = 6.28$. More relevant for reference tracking is $C_2(s)$ where q was set to 29 rad s^{-1} . Correspondingly, the forward model became a second-order low-pass filter with the same poles. The data on time delays was collected by repeating this process for all 36 different systems.

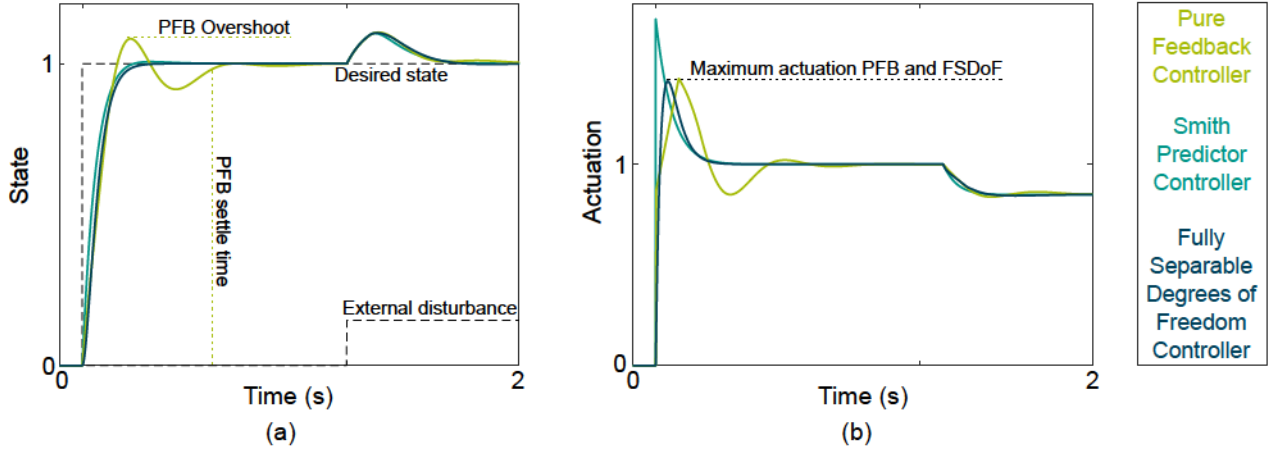


Figure 2.2: Step response of all three controllers on the first structure of dynamics in Table 2.1 with $\omega = 10 \text{ rad s}^{-1}$ and $\tau = 100 \text{ ms}$. (a) Desired state (dashed step function) and the actual state, the PFB overshoots 8.3 %, the SP overshoots 0.6 %, whilst the FSDoF controller does not overshoot. The FSDoF controller has a settle time of 20 ms, compared with 56 ms for the PFB and 18 ms for the SP controller. (b) shows the actuation used by the three controllers. The maximum actuation for the FSDoF and PFB is 1.4 units, and for the SP it is 1.7.

The demonstration of handling noisy systems was done by tuning two PFB controllers to the system dynamics and only altering the response time parameter in the PID tuner, which modified the bandwidth. It is not possible to perfectly simulate white noise due to its infinite frequency spectrum. Instead, a white noise approximation using a random sequence with a correlation time constant (MathWorks 2025a) was injected at the sensor. The correlation time constant must be significantly smaller than the smallest time constant in the system being simulated, therefore, I used a value of 0.1 ms and a power of 0.001. The FSDoF controller was equipped with the same feedback controller as the lower bandwidth PFB. The SP was equipped with the same feedback controller as the higher bandwidth PFB. The constraint of maximum actuation was relaxed for this demonstration, given the actuation used was largely determined by the response to noise.

Full tables of the controllers and performance metrics are provided in appendix C.

2.3 Results

An example of how the three controllers perform when designed around the same system dynamics is given in Figure 2.2. To assess the performance of the reference tracking I measured

the overshoot and the settle time. The overshoot measures how far the actual state exceeds the desired state before converging to the desired reference state; and the settle time is the time to reach and stay within 2 % of the desired state as shown in Figure 2.2a. I also measured the actuation costs in the form of the maximum actuation required as shown in Figure 2.2b.

2.3.1 Advantage 1: Mitigating the impact of sensor delays

The performance results for the overshoot and settle time at different sensor delays are shown in Figure 2.3. The fully-separable-degrees-of-freedom (FSDoF) controller never overshoots by design (Fig. 2.3b). The other two controllers do overshoot, and in the case of the pure feedback controller (PFB), the overshoot increases with the sensor delay.

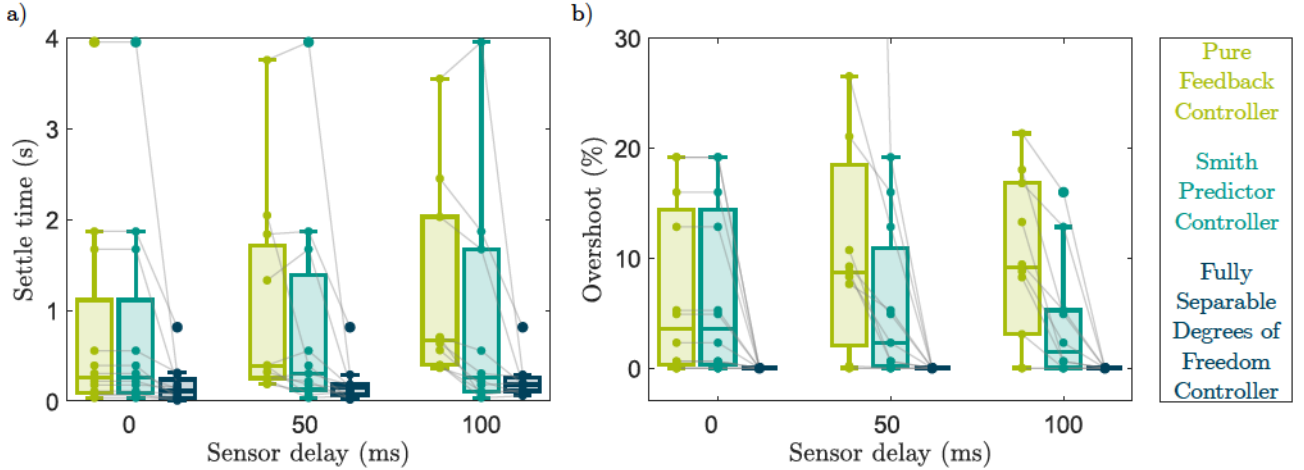


Figure 2.3: Performance at different sensor delays of three control architectures. The three architectures were tuned to one of the 36 dynamics (Tab. 5.1), and the step-response reference tracking performance was measured. This was repeated 36 times for each of the different system dynamics. (a) Settle time in seconds for each controller at different sensor delays. (b) Percentage overshoot of each controller at different sensor delays.

Whilst the FSDoF does not overshoot the desired state, for non-minimum phase systems i.e. plant structure 6 in Table 2.1, all controllers caused the state to initially go below zero before approaching the reference. In all cases, the FSDoF undershot by more than the SP and PFB for a similar maximum actuation. By adjusting FSDoF parameterisation the undershoot can be arbitrarily reduced at the expense of settle time (Fig. 2.4f).

The results from the Smith predictor (SP) should in theory be identical for all sensor delays. This is not the case for the results shown in Figure 2.3, as three cases were excluded from the

analysis due to the PFB being unstable. For plant structure five, three parametrisations were unstable: $\omega = 10 \text{ rad s}^{-1}$ $\tau = 100 \text{ ms}$, $\omega = 50 \text{ rad s}^{-1}$ $\tau = 50 \text{ ms}$, and $\omega = 50 \text{ rad s}^{-1}$ $\tau = 100 \text{ ms}$. When these are included in the analysis the Smith predictor has identical performance for all sensor delays—in line with the theory.

A similar argument can be applied to the FSDoF controller. As seen in equation 2.7 the transfer function from the desired state to the actual state is independent of the sensor properties. This means the reference tracking performance is independent of any delays in the sensors, yet there are small variations in settle time as the sensor delay is increased. The reason for this variation is that, due to the changing actuation required by the PFB, the FSDoF is allowed more or less actuation at different sensor delays. If the maximum actuation was fixed for all time delays, the settle time would be the same for different sensor delays, in line with the theory.

The settle time of the PFB increases significantly as the sensor delay increases. The mean settle time of the PFBs are 0.79 s, 1.00 s, and 1.18 s as the sensor delay increases (Fig. 2.3a). The Smith predictor has a mean settle time of 0.84 s which is reasonably consistent across different sensor delays. Both of these are much higher than the mean settle time of the FSDoF controller which has a mean settle time of 0.20 s; less than a quarter of the other two controllers. The FSDoF converges faster on the reference state than the PFB controller, and the performance gap increases with the sensor delay. Even compared to a controller specifically designed to handle time delay systems, the SP, the FSDoF offers a large improvement. The handling of sensor delays is an inherent advantage of the FSDoF controller. It is worth noting that a feedback component is required in all three control architectures for rejecting disturbances, the disturbance rejection performance will suffer from increased sensor delays for all controllers.

Another quality of the FSDoF controller is that the desired changes in state, when given more actuation, can often be arbitrarily fast (Fig. 2.4), depending on the dynamics. This is not the case for the SP or PFB where high gains can lead to instability, in particular for systems with sensor delays. The only trade-off with reducing the settle time of the FSDoF, besides the increased actuation, occurs for plant 6 where the undershoot significantly increases (Fig. 2.4f). Further biological system identification (Cowan et al. 2014) studies would shed more light on whether non-minimum phase systems are present in insect dynamics.

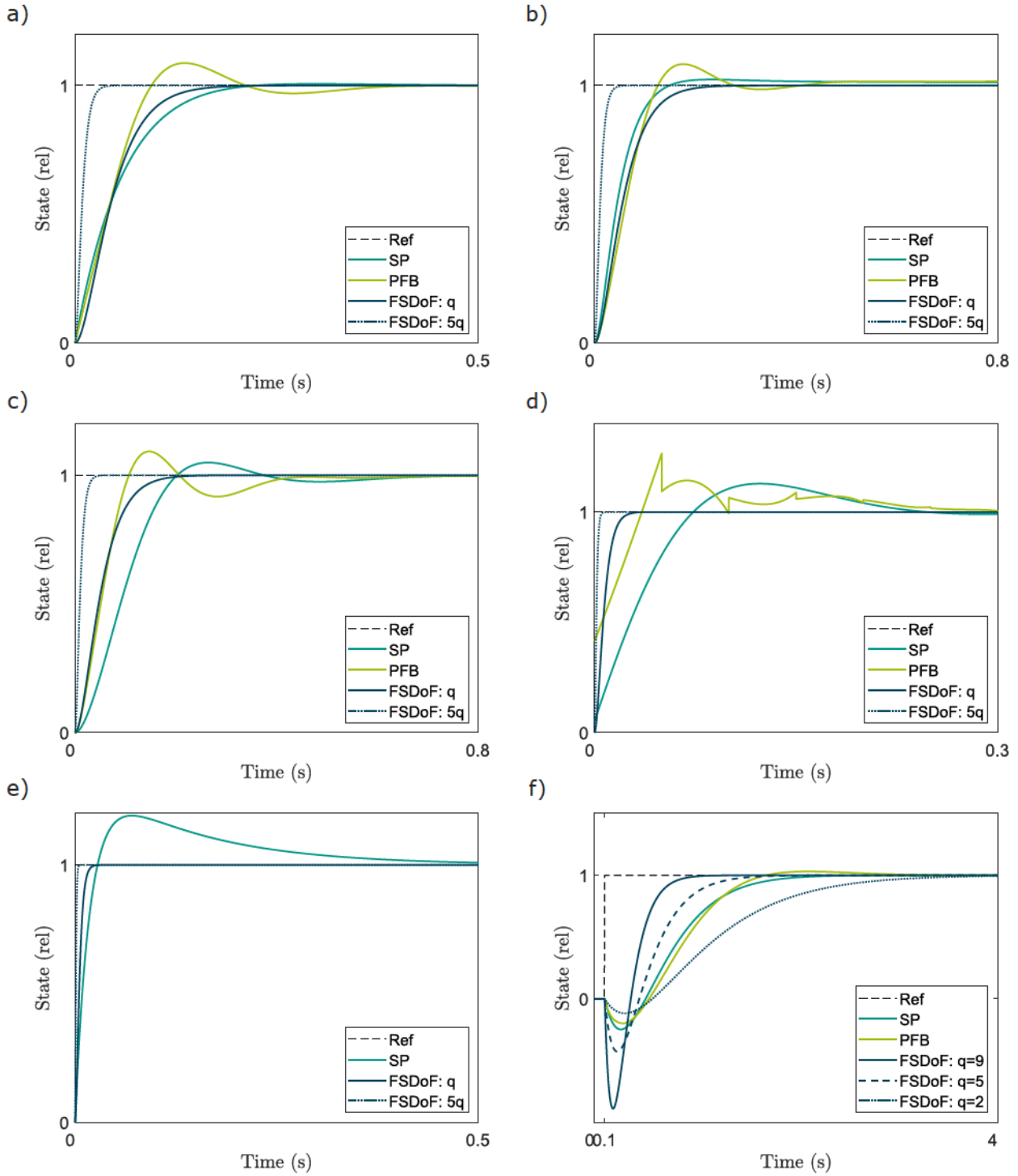


Figure 2.4: Step responses for $\omega = 10$ rad/s and $\tau = 50$ ms for all six dynamics. (a) to (f) represent dynamics one to six, respectively. For (a) to (e) the step response for the value of q used is plotted, and also the step response for when q is five times larger. (f) shows three different values of q to highlight the trade-off with undershoot and settle time for non-minimum phase systems.

2.3.2 Advantage 2: Mitigation of sensor noise

In classical feedback control, there is a trade-off between noise rejection and reference tracking. If the sensor signal is heavily filtered, the feedback control will be slowed down. If the feedback controller has a high bandwidth, it will track the desired state faster. However, it will come at the cost of a stronger response to noise from the sensors. A more formal mathematical evaluation of this trade-off is given in Astrom and Murray (2008).

The behaviour of two PFB controllers highlights the trade-off (Fig. 2.5a). One rises to the desired state (unity) quickly, but is highly sensitive to noise. Conversely, the other PFB rejects the noise effectively but rises to the desired state more slowly.

In contrast, the fully-separable-degrees-of-freedom (FSDoF) control architecture mitigates this trade-off (Fig. 2.5a). The inner loop filters noise, and the outer loop governs the desired changes in state. The outer loop is not affected by sensor noise. As a result, the reference tracking performance does not depend on sensor noise.

For the Smith predictor (SP), a low-pass filter can be used to mitigate the impact of noise. This does not slow the response to desired changes since the feedback commands to change state are largely driven by the undelayed model, which does not contain noise. The low-pass filter is shown in Figure 2.1b as an optional filter. So, this advantage is apparent in both the SP and the FSDoF architectures (Fig. 2.5b).

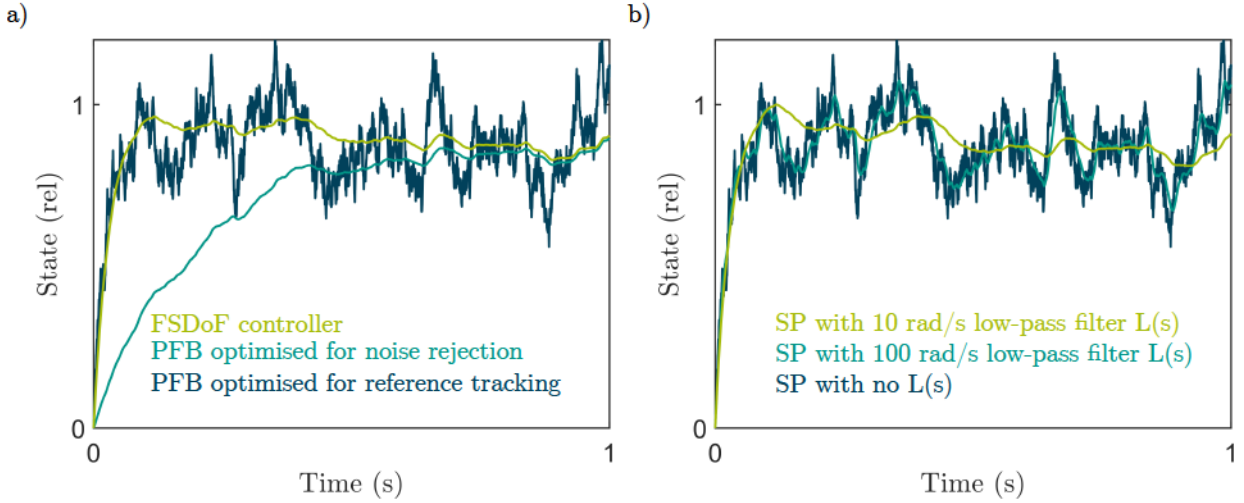


Figure 2.5: Impact of sensor noise on reference tracking performance exemplified for different controllers. All 6 systems have the same sensor noise. (a) FSDoF, and two PFBs with different bandwidths. There is no way to suppress noise in a PFB architecture without slowing the reference tracking, unlike with FSDoF control. (b) Three different SP controllers with different optional filters $L(s)$. The optional filter in the SP allows noise suppression without affecting the reference tracking performance.

2.3.3 Advantage 3: Preventing sensor saturation

In their study on efference copies in fruitflies, Kim, Fitzgerald, and Maimon (2015) found the cancellation of the expected sensor response during spontaneous yaw turns to take place in specific directional-selective interneurons of the animal's motion vision pathway. Those so-called HS-cells, specifically the HSN- and HSE-cells, have been suggested to sense yaw rotations by processing wide-field optic flow (Hausen 1993). Like any neuron in the nervous system, HS-cells have a limited electrical signalling range with which they encode motion information. By externally varying the membrane potential, the range of membrane currents of the HS-cells has been measured to cover approximately -25 to 10 nA (Haag, Theunissen, and Borst 1997). Where the bounds are determined by the equilibrium potentials and the number of open ion channels of contributing ion types. Cancelling the predicted sensory response of desired state changes, with an efference copy, keeps the signal of HS-cells within its operating range. Thus, it retains high sensitivity to external disturbances and effectively prevents saturation.

This is also the case for the output signals of spiking descending neurons which mediate sensory signals from the motion vision pathway and other modalities to the motor centres in the fly thoracic ganglion. Equation 2.16 is an *integrate and fire* model that can be used to

describe the spike rate of a descending neuron, where, I is the input current, C is the membrane capacitance, V_{th} is the threshold voltage, and t_f is a characteristic time for the neuron.

$$f_{spike} = \frac{I}{CV_{th} + t_f I} \quad (2.16)$$

Clearly, from this equation, if $I \gg \frac{CV_{th}}{t_f}$, the neuron begins to saturate. Large changes in input current result in small changes in spike rate.

If a fly did not cancel the expected sensor response during an intended bank turn, the visual neurons would be close to saturation and become less sensitive to external disturbances during the turn. The FSDoF and SP architectures cancel the expected sensor response, but the PFB does not provide this advantage.

2.3.4 Controller robustness

Integrating all physiological and biomechanical properties of a biological control system would require an exceptional level of complexity. What happens when the controller is built around an incorrect or approximate model of the sensors, actuators, and system dynamics?

In this section, the stability robustness is calculated, a common metric in robust control that describes how different the dynamics can be, for a given controller, such that the system remains stable. Secondly, an example of the performance robustness is presented where each controller uses high-pass filter sensors. These are given for the case that the system dynamics (Sec. 2.3.4) and the sensor dynamics (Sec. 2.3.4) differ from the dynamics that the controller was designed around.

System dynamics modelling error

Firstly, I consider the case where the controller has been designed around system dynamics that are inaccurate or have changed.

Mathematically, the stability robustness can be calculated by applying a multiplicative error to the system dynamics and calculating the condition for stability for the different controllers. A multiplicative error can be modelled in the Laplace domain by redefining the system dynamics

as $P(s)(1 + \Delta_P(s))$; where $\Delta_P(s)$ is the Laplace transform of the uncertainty in the dynamics and $P(s)$ represents the baseline system dynamics. The conditions for closed-loop stability for the three controllers can be derived as an extension to the Nyquist criterion using the small-gain theorem (Green and Limebeer 2012a). Despite the desired change controller and forward model, the PFB, SP, and FSDoF have the same condition on the uncertainty and controller to remain stable, Eq. 2.17.

$$\|\Delta_P\|_\infty \|HC_1P(1 - HC_1P)^{-1}\|_\infty < 1 \quad (2.17)$$

This can be understood by considering the inner-loop of the FSDoF controller as a mechanism that uses feedback to mitigate any unanticipated sensor response. Including those as a result of modelling errors: if the system dynamics differ from those expected by the forward model, the feedback controller will act to obtain the expected sensor response.

Despite the three controllers having identical conditions for stability robustness, their architectures may have different feedback controllers $C_1(s)$. For example, in an SP the feedback controller is usually designed around the undelayed system, leading to a different $C_1(s)$. As outlined in the introduction, it is generally accepted that the Smith predictor is less robust than a PFB controller (Adam, Latchman, and Crisalle 2000; Palmor 1980), and some argue that for a fixed robustness a PFB offers better performance than a SP (Grimholt and Skogestad 2019).

Secondly, in the PFB the feedback controller must be designed around balancing robustness, reference tracking, and disturbance rejection. However, as with all two degree-of-freedom controllers, the feedback component of the FSDoF only needs to consider robustness and disturbance rejection, which generally leads to superior robustness (Araki and Taguchi 2003).

To assess the performance robustness I will use the transfer functions for the reference tracking of the three architectures on the uncertain system dynamics are given for the FSDoF, PFB, and SP:

$$\frac{X}{X_0} = C_2P \left(1 + \frac{\Delta_P}{1 + HC_1P(1 + \Delta_P)} \right) \quad (2.18)$$

$$\frac{X}{X_0} = \frac{C_1P(1 + \Delta_P)}{1 + HC_1P(1 + \Delta_P)} \quad (2.19)$$

$$\frac{X}{X_0} = \frac{C_1P(1 + \Delta_P)}{1 + C_1G + C_1H\Delta_P} \quad (2.20)$$

An interesting feature of the FSDoF architecture is that instead of stabilising state units, it stabilises sensory units. For example, in the fly visual system, instead of stabilising roll angle, a possible internal representation of the state, the inner loop would be stabilising the response of a neuron that is tuned to sense roll rotation. This has some interesting effects: if the sensors act as high-pass filters, or derivatives, for the state units, this could result in steady-state errors. This can be explained intuitively: the stabilising inner loop may not see the unexpected different sensor response for long enough to be able to mitigate it. With the SP and PFB, sensors that act like high-pass filters are generally followed by an inversion of the sensor dynamics to obtain an estimate of the state, which means that most changes in system dynamics will not lead to steady-state reference tracking errors. However, for the FSDoF controller the high-pass filter dynamics are taken into account in the forward model. Figures 2.6a and 2.6b illustrate an example system where changes occur in the system dynamics. When the gain of the system dynamics changes, the FSDoF controller exhibits a small steady-state error in reference tracking (Fig. 2.6b). To determine whether or not this will occur, the final limit theorem can be applied to equation 2.18.

Sensor dynamics modelling error

When using high-pass filter sensors, it is possible that system dynamics changes cause steady-state errors when using FSDoF control. This may appear to be a disadvantage. However, the same principle works as an advantage for a different type of error. In biological systems, sensors are less reliable. Returning to the fly visual system again, the sensor response depends on the temporal and spatial properties of the environment (Taylor and Krapp 2007). If instead of the system dynamics being different to the expectation, the sensor response is, the resulting error due to this could be reduced.

In the SP and PFB architectures, if the sensors are incorrect, the feedback controller will settle to obtain an incorrect sensor response, thus the system will be at the wrong setpoint. With the FSDoF controller and a transient sensor response, a trade-off can be made. By permitting errors resulting from incorrect system dynamics, errors from incorrect sensor signals will be reduced. In Figures 2.6c and 2.6d this is exemplified, the steady-state error is significantly larger

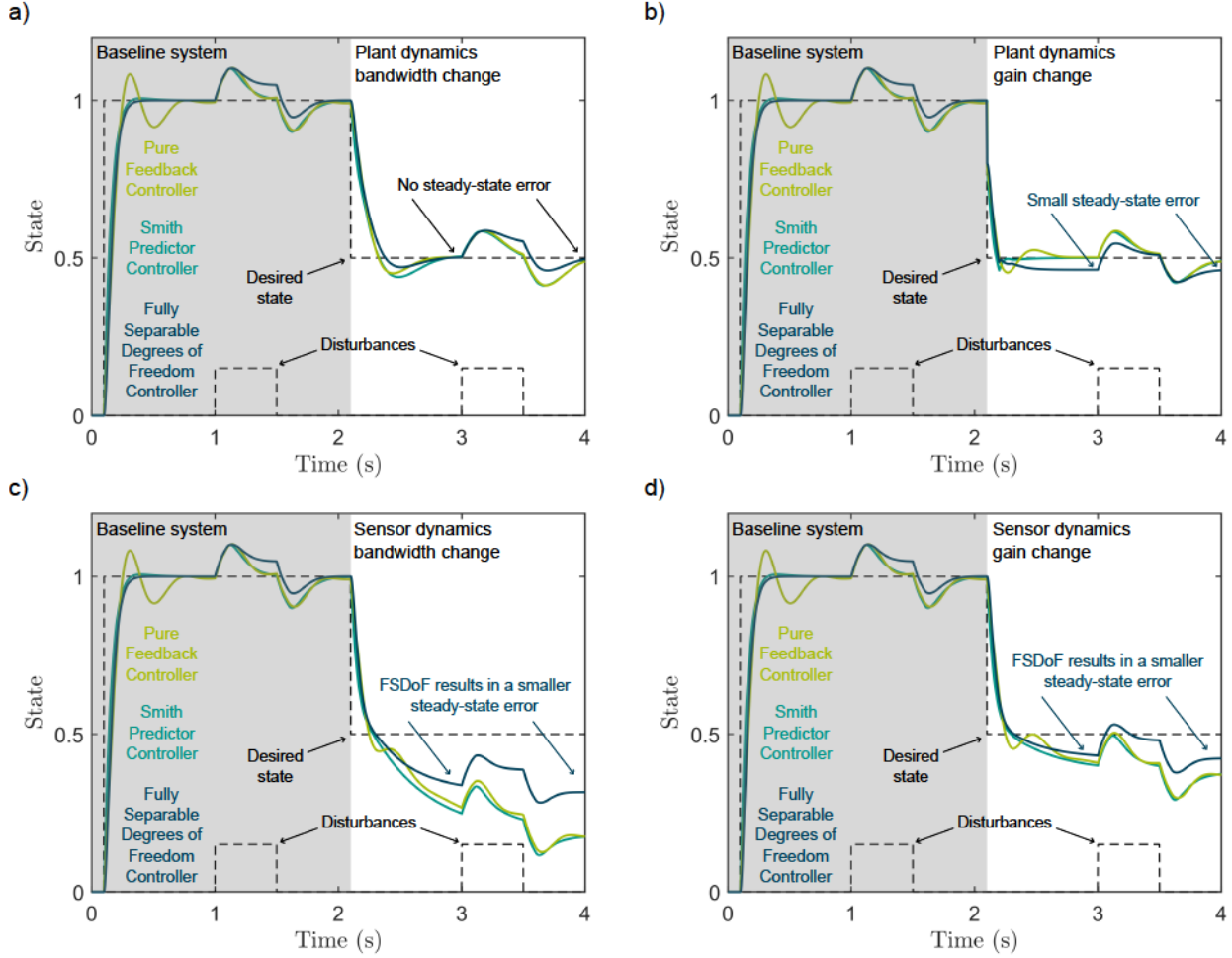


Figure 2.6: Three controllers with high-pass filter sensors responding to systems where either the system or sensor dynamics change instantaneously at 2.1 s. The controllers were designed around the baseline dynamics of structure 1 in table 2.1 with $\omega = 10 \text{ rad s}^{-1}$ and $\tau = 100 \text{ ms}$. The baseline high-pass filter sensor is first-order with a bandwidth of 3 rad s^{-1} and a unity gain. The SP and PFB are calculated in the same way as described in Sec. 2.2 with an additional block that inverts the baseline sensor dynamics to obtain the state estimation. The FSDoF is also the same as described in Sec. 2.2 but with the baseline sensor dynamics included in the forward model and the feedback component adjusted accordingly. (a) System dynamics bandwidth changes to 16.7 rad s^{-1} . (b) System dynamics gain changes to 0.8. (c) Sensor dynamics bandwidth changes to 5 rad s^{-1} . (d) Sensor dynamics gain changes to 0.8.

for the PFB and SP than for the FSDoF when the sensor dynamics change.

To summarise, the risk of instability due to unexpected system dynamics for the FSDoF control architecture is identical to the PFB and SP controller. Secondly, in the PFB and SP, a system dynamics modelling error will generally not result in any steady state error between the actual state and the desired state. But if there is an error in the sensor model, they will stabilise around incorrect sensor values. The same applies to the FSDoF architecture when the sensors measure the absolute value of a given state variable. However, biological sensors often act like high-pass filters or derivatives (Fuller et al. 2014; Nalbach 1993; Taylor and Krapp 2007). In this case, with the FSDoF architecture, small steady-state errors may be permitted when the system dynamics are different to the expectation. The advantage of permitting that steady-state error is a reduced steady-state error when the sensor signal is incorrect.

2.4 Controller variants

There are demonstrable advantages of the FSDoF controller over a PFB. Given that the FSDoF is a form of the two-degree-of-freedom controller, there are equivalent variations shown in Figure 2.7. All five of those variants have the same transfer function as the FSDoF given by Equation 2.5, and therefore should have theoretically identical reference tracking performance. The first three (Fig. 2.7a, 2.7b, 2.7c) have different structures to the FSDoF, using either a feedforward controller (Fig. 2.7a and 2.7c), or a reference filter alone (Fig. 2.7b). I will refer to these three variants as being structurally different. The final two equivalent variants (Fig. 2.7d and 2.7e) use both a reference filter, and feedforward controller, akin to the FSDoF in Figure 2.1c, although in a different order. Experimental evidence suggests that biological control architectures use a feedforward command and cancel the expected sensor response (Kim, Fitzgerald, and Maimon 2015; Kim, Fenk, et al. 2017; Fenk, Kim, and Maimon 2021). This rules out the structurally different variants in Figure 2.7. However, there is no biological evidence in support of the original variant in Figure 2.1c, which does not support the two structurally similar variants in Figure 2.7 as well. The purpose of this section is to (i) explain the intuition behind these variants, (ii) explore reasons why biological systems might not use the structurally different

variants, and (iii) assess practical differences of the structurally similar variants.

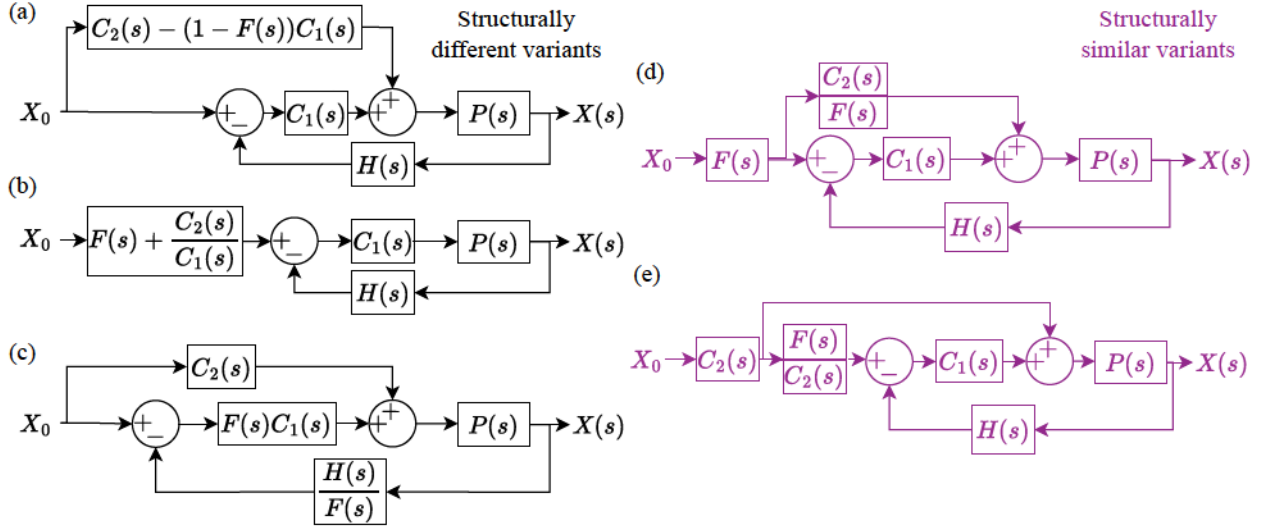


Figure 2.7: Five control architectures that give identical reference tracking equations to the FSDoF. Structurally different variants are given in black, and structurally similar variants in purple. I label the variants structurally similar if they are compatible with the electrophysiology results in (Kim, Fitzgerald, and Maimon 2015; Kim, Fenk, et al. 2017; Fenk, Kim, and Maimon 2021). (a) a feedforward motor command only. (b) a filter on the desired-state only. (c) The forward model is inside the feedback loop. (d) and (e) are equivalent structures to Figure 2.1c, both provide the full actuation command in feedforward, and have forward models cancelling the expected sensor signal, just in an altered order.

Qualitative descriptions of the variants' operation

The original structure in Figure 2.1c has two separate pathways from the desired state, one sending the actuation command, and one cancelling the expected sensor signal. In the first variant in Figure 2.7a the feedforward controller uses the forward model to provide the actuation required for the desired state change, minus, the actuation that the feedback controller is expected to generate based on the resulting sensor output. In this case, the cancellation occurs at the level of the actuator as opposed to the sensor. For example, if a fly intends to turn 10 rad s^{-1} to the left, it would “know” what motor command must be sent to the flight motor to produce the required torque. When the reafferent visual feedback triggered an optomotor response, it would also provide a feedforward command to the muscles that inhibited that, i.e., cancel the expected motor command instead of the expected sensor response.

The variation in Figure 2.7b operates in the opposite way, it has no feedforward controller, just a filter on the reference. This reference filter cancels (i) the expected sensor response

as a result of the desired state change, and (ii) the sensor response required to trigger the feedback controller to generate the actuation command. So, if a fly wanted to turn left, it would internally generate a sensor output that would trigger the optomotor left turn and then cancel the reafferent visual feedback.

Figure 2.7c shows a variant where the desired change controller has the same role, but the forward model has been moved inside the control loop. The sensor response is transformed into an estimation of the state through a division by the forward model. This signal is compared to the desired state, and then feedback control is used to reduce the error.

A structurally similar variant where the forward model still cancels the expected sensor signal, and the feedforward controller provides all the actuation commands to change state, is shown in Figure 2.7d. However, this time, the feedforward controller generates the actuation command based on the expected sensor response. If a fly wanted to turn left, it would cancel the expected sensor response—the efference copy. Then, based on the efference copy signal it would generate the motor command to get that sensor response. The second structurally similar variant in Figure 2.7e operates the other way round. It generates the actuation command, and the forward model predicts the sensor response based on the actuation command. This final variant reflects more directly the original etymology of *efference copy*, meaning motor copy (Webb 2004a), and corresponds to option 2 in Fig. 1.2 (Ch. 1). A fly turning left would provide the motor command to the actuator, and then, based on the motor command signal it would generate the expected sensor response it should cancel.

Why not cancel the expected motor command?

Flies appear to cancel the expected sensor signal as opposed to the motor command initiated by the reafferent sensor response as suggested in Figure 2.7a. The first reason may be that this variant does not always exhibit internal stability, whilst it is input-to-output stable, there are conditions under which individual blocks are not. These scenarios arise when C_1 has an integral component. When there is integral action, any combination of input signal (X_0) and forward model (F) such that $(1 - F)X_0$ has a non-zero average, will lead to internal instability. The integral components in the feedforward and feedback paths will tend towards $\pm\infty$. A

non-zero average in $(1 - F)X_0$ can arise when (i) there are high-pass filter sensors, (ii) there is a non-linearity, for example, if there is a limit on the rate at which $F(s)$ can change its output, or (iii) if the implemented $F(s)$ does not have a perfect unity DC gain.

The variant in Figure 2.7c also has a potential problem, with the inversion of the forward model $F(s)$. Firstly, if there is a positive zero in the system dynamics, a positive zero in $F(s)$ is required for $F(s) = C_2PH$ to hold. An inversion of $F(s)$ under this condition, as in Fig. 2.7c, will result in an unstable pole. Secondly, if there is a time delay, the inversion will create the need for a positive time delay, which is not possible.

Lastly, and perhaps most significantly, none of the three structurally different architectures prevents sensor output saturation, as described in Section 2.3.3.

Does the order matter?

Considering the two structurally similar variants, they both solve the sensor saturation problem. The first of the two variants generates the feedforward motor command based on the efference copy. This variant suffers the same challenges with inverting $F(s)$ as explained above. If the system dynamics had a positive zero or there was a sensor delay, the inversion of the forward model would lead to internal instability or infeasibility.

The final structurally similar variant (Fig. 2.7e) is more plausible and reflects the etymological origins of how efference copies may be used in sensorimotor control. One practical challenge with this architecture however is its handling of unstable system dynamics. Unstable system dynamics have a positive pole in $P(s)$. The filter $\frac{F(s)}{C_2(s)}$, is a model of the system and sensor dynamics, from Equation 2.6. Is it possible for animals to have a neural filter that represents unstable dynamics?

In the structure analysed in Figure 2.1c each block in the diagram is causal. In addition, when controlling unstable dynamics and a stabilising desired change controller, the forward model will be internally stable. It is worth noting that there have not been physiological experiments showing the exact origin of the feedforward control command generated by the desired change controller $C_2(s)$, or the efference copy generated by the forward model $F(s)$. In stable biological systems, it is perfectly possible to have a structure similar to that shown in

Figure 2.7e (resembling option 2 in Fig. 1.2).

2.5 Conclusions

Motivated by recent studies in the insect sensorimotor system I developed a phenomenological control architecture and benchmarked its performance against common control engineering approaches. I find that for accurate fully-separable-degrees-of-freedom (FSDoF) controllers, the performance in tracking the desired state is independent of the sensor output. This means that the control architecture is particularly good at handling sensors with time delays and significant noise.

Sensory delays are common in biological systems (Saijo et al. 2005; Matthews 1991; Borst and Egelhaaf 1989), yet many animals employ high-performance control systems. This work suggests how an FSDoF controller with forward models (efference copies) and desired change controllers could achieve this. In particular for systems with fast dynamics but significant sensor delays, like insects. This combination of fast dynamics and comparatively slow sensors is present in the fly visual system (Borst and Egelhaaf 1989; Taylor and Krapp 2007).

Another factor to consider is noise in the nervous system (Faisal, Selen, and Wolpert 2008). I show that the FSDoF architecture allows fast goal-directed behaviours despite sensor noise. This is due to the inherent property of the outer loop control signals being sensor-independent. Secondly, by subtracting the expected sensor response the sensors are kept in their operating range, maintaining sensitivity to external perturbations. This could cause a stronger signal power, and therefore, for a fixed level of noise, a better signal-to-noise ratio for disturbance rejection.

Further, the visual system in flies does not respond linearly to velocity, and has significant transient modulations that respond to the spatio-temporal frequency distribution of the environment (Meyer, Lindemann, and Egelhaaf 2011; Borst and Egelhaaf 1989). Despite this, flies remain incredibly agile and fast at enacting goal-directed behaviours. The results presented here could explain how the insect flight control system works effectively despite these sensory challenges. Firstly, the unwanted modulations could be treated like sensor noise, and with

the FSDoF architecture, they can be suppressed in feedback without impacting high-speed goal-oriented manoeuvres. Secondly, given high-pass filter sensing properties, the FSDoF had better sensor performance robustness. I.e, if the sensor gain and bandwidth change, the resulting error in the desired state can be made smaller with FSDoF and high-pass filter-like sensors.

The FSDoF control architecture does not overshoot, which is consistent with experiments in which humans are tasked with reaching an object and the position of their hand is tracked (Botzer and Karniel 2013). There is no work suggesting that insect control does or does not overshoot as monitoring the desired state of insects is not yet possible. However, in a study comparing voluntary take-offs with the escape response to take-off, getting up to horizontal speed following a voluntary take-off does not appear to overshoot (Figure 7A in Card and Dickinson 2008a).

A point that raises interesting questions is that the feedback controller does not control parameters in state units, but instead controls parameters in sensory units. This could potentially lend itself to a distributed control system. Instead of having many sensors integrated to provide the current state and then having a single, complicated, feedback controller. It allows each sensor system to sense something unanticipated and feedback directly to a corresponding motor system. This could potentially reduce computational overheads, inverting and combining information from multiple sensors, and provide a shorter neural pathway from sensor to muscle. Thereby, reducing feedback delays, and improving disturbance rejection performance. There is another potential advantage of this strategy. Different sensing modalities have different response delays. The FSDoF control architecture could provide a mechanism to utilise sensors operating at different speeds. A similar strategy was found in the fly gaze stabilisation system where the fast mechanoreceptor signals provide muscle commands quickly, while the slower visual system incurs a longer response delay before activating the neck muscles (Schwyn et al. 2011). This argument is developed further in Chapter 5.

Recent work makes a distinction between “graded” and “all-or-none” efference copies (Canelo et al. 2023); where a graded efference copy is the output of a forward model predicting the sensor response to cancel, and all-or-none is effectively turning off the sensory feedback controller when enacting a desired state change. In this chapter, I have analysed a graded efference

copy model, however, animals likely employ both methods. The results can be extrapolated to all-or-none efferece copy control systems; the three advantages would still be obtained. The two primary differences in performance are that with the all-or-none efferece copy (i) it becomes theoretically impossible to control unstable systems, and (ii) external disturbances during desired state changes are not sensed.

This Chapter demonstrates some potential advantages to biological systems using control structures like the efferece copy-based controller described here. Although the results show comparative stability robustness for the FSDoF scheme, for optimal performance the desired change controller and forward model should be accurate. How insects maintain the agility for high-speed manoeuvres required in nature, when their system dynamics change, may be due to adaptation. Insects adapt their sensors (Laughlin and Hardie 1978; Van Hateren 1997; Harris, O'Carroll, and Laughlin 2000) and their controllers (Wolf, Voss, et al. 1992; Rosner, Egelhaaf, and Warzecha 2010; Chiappe et al. 2010). One possible method for this adaptation in the presence of variable system dynamics could be to adapt the feedforward controller, which is investigated further in the following chapter.

Chapter 3

Adaptive feedforward speed control following wing damage in *Drosophila*

3.1 Introduction

Animals must contend with coordinating self-motion despite the ever-changing dynamics of both the neuromuscular system and the external world. Such changes include variations in either the relationship between the external world and physiological sensor responses, or the relationship between physiological motor commands and the resultant forces. In principle, the nervous system could maintain performance under these variable dynamics either by using a robust and unchanging sensorimotor controller, or by adapting the controller to the changed dynamics. However, animals across phyla appear to adapt their sensorimotor systems (Wolf, Voss, et al. [1992](#); Tuthill and Wilson [2016](#)). For example, systematic perturbation of human subjects reaching for a target results in new reaching dynamics that are sustained following further changes in experimental dynamics (Wolpert, Ghahramani, and Jordan [1994](#)).

In vertebrates, the cerebellum is thought to underly adaptive sensorimotor control through a combination of forward and inverse internal models (Wolpert, Miall, and Kawato [1998](#)). Forward models predict proprioceptive and sensory feedback as a result of executing a specific motor action (Wolpert, Miall, and Kawato [1998](#)), thereby enabling the nervous system to distinguish between the sensory consequences of self-motion (re-afference) and sensory input arising from

external sources (ex-afference) (von Holst 1973). This distinction prevents stabilisation reflexes from countering volitional movements, whilst allowing the same reflexes to operate continuously to maintain stability. Feedforward commands are estimates of the motor commands required to enact desired movements or, at a minimum, supplement the actuation commands of a feedback controller. The feedforward component within the control architecture avoids some of the decreased performance as a result of sensory delays, as well as being more robust to sensor saturation and noise (Ch. 2; Dean et al. 2010). Furthermore, by separating re- and ex-afferent signals, the feedforward controller can theoretically be adapted based on an error signal defined by the difference between the expected and actual sensor response, rather than the desired and actual state. This is advantageous as it is not subject to errors that result simply from time delays between control commands and the resulting sensory feedback (Kawato 1990). Importantly, both forward and inverse models are adaptive filters (Dean et al. 2010), enabling them to account for the variable dynamics between the external world and the sensory (forward) and motor systems (inverse).

Insects are also capable of adapting their sensorimotor control systems (Wolf, Voss, et al. 1992; Wolf and Heisenberg 1991; Salem, Cellini, Jaworski, et al. 2023). Insects can compensate for changes in dynamics due to limb loss (Isakov et al. 2016), wing damage resulting from collisions or predation (Jantzen and Eisner 2008; Muijres et al. 2017; Salem, Cellini, Kabutz, et al. 2022; Rajabi, Dirks, and Gorb 2020), mass loading (Salem, Cellini, Jaworski, et al. 2023; Davidovich and Ribak 2018; Evans 1962), temperature changes (Lehmann 1999), and nutritional state (Lin, Senapati, and Tsao 2019). Tethered flying *D. melanogaster* orient towards a vertical stripe under closed-loop conditions in which their yaw torque controls the azimuthal position of the stripe, even when the feedback sign between yaw torque and stripe position is inverted (Heisenberg and Wolf 2013). Such sensorimotor adaptation is transferable to other motor systems under novel dynamics unlikely to be experienced in nature. For example, flies continue to fixate a vertical stripe by using their prothoracic legs to control a lever (Wolf, Voss, et al. 1992), or using tethered flight yaw torque to turn off a high-temperature laser (Heisenberg and Wolf 2013).

Whilst insects can adapt to changes in dynamics, it is unclear whether insects learn to

reconfigure a feedback controller or update a feedforward controller—akin to cerebellar inverse models. Evidence for feedforward control has been demonstrated in the fly (Schwyn et al. 2011) and wasp (Viollet and Zeil 2013) gaze stabilisation systems, and mantis shrimp strikes (Kagaya and Patek 2016). However, whether such feedforward controllers can be adapted in insects remains unknown. Recent behavioural work in *D. melanogaster* demonstrated that the optomotor feedback controller does not adapt (Cellini, Ferrero, and Mongeau 2024). The frequency and phase responses of the yaw optomotor response of tethered flies in an augmented reality arena were found to be invariant for 30 minutes despite experimental changes in visual feedback gains (Cellini, Ferrero, and Mongeau 2024). This suggests that the insect visual feedback controller is unchanging, implying that control flexibility in goal-directed behaviours may indeed arise from adaptive feedforward control.

In this Chapter, I provide evidence supporting that feedforward mechanisms may adapt to changing system dynamics, using experiments that control the availability of visual feedback following wing damage in *Drosophila melanogaster*. The results confirm that changes in flight trajectories due to wing damage can be compensated for in the presence of visual feedback, as previously reported (Leonte et al. 2021). However, I also find that the compensation for flight speed persists upon returning to the dark, whereas other parameters such as flight curvature and velocity vector elevation angle do not. These lasting changes are still observed after disruption of the antennal mechanosensory system, suggesting that they arise from adaptation of the feedforward component of the flight speed controller, rather than adaptation of the antennal feedback system.

3.2 Methods

3.2.1 Animals and ablation procedures

A lab-reared colony of recently wild-caught (i.e. within 9 months) *Drosophila melanogaster* fruit flies were maintained on fruit fly media (Advanced Husbandry, Pets and Pieces Ltd., UK) under a 12/12 hr on/off light cycle (8am/8pm). A total of 681 7-28 day old flies were used for the experiments. For clipped-wing conditions (581 flies), flies were cold anaesthetised, and one

wing per fly was clipped chordwise using micro forceps from the intersection of the longitudinal veins L1 and L2 (Fig. 3.1, following previous work: Leonte et al. 2021). For experiments with bilaterally disrupted antennal mechanosensation (241 flies), both antennal aristae were removed from the antennae (following previous work: Fuller et al. 2014).

Table 3.1: Numbers of animals used in each experimental condition and the number of intersaccadic flight segments included in the analysis.

# Flies	# Included intersaccadic flight segments			Animal condition	
					
	Dark (D)	Light (L)	Dark-after-light (DAL)		
206	85	1517	143		Clipped wing (CW)
241	12	1113	20		Ablated-antennae clipped wing (AA-CW)
134	*	61	36		Clipped wing (CW)
100			 } $\times 2$		Intact (I)
	24 hours				



Figure 3.1: Clipped-wing (CW) *D. melanogaster*

3.2.2 Experimental setup and procedure

Flies were inserted into a custom made $400 \times 400 \times 300$ mm clear acrylic enclosure with 6 mm wall thickness, via an opaque transfer tube (Figure 3.2a). An LED strip (white light, 9.6 W m^{-1} , CRI > 80) positioned around the four upper edges of the enclosure provided visible illumination, and a checker pattern was positioned on two side walls to provide visual contrast. The enclosure was continuously illuminated with two infrared lights (96 850 nm wavelength LED, 12 W) diffused with tracing paper directed into the remaining two side walls to permit videography in both light and dark conditions. Two fastcam Sa3 high-speed cameras (Photron Europe Ltd.) with 24 mm lenses (Nikon, AF Nikkor 1:2.8 D, Japan) were positioned beneath the enclosure. The cameras were triggered to capture flight trajectories at 125 fps (aperture F/4, 1 ms exposure time), resulting in a 21.6 seconds (2700 frames) stereo recording every 10

minutes. Each frame was 1024 by 1024 pixels.

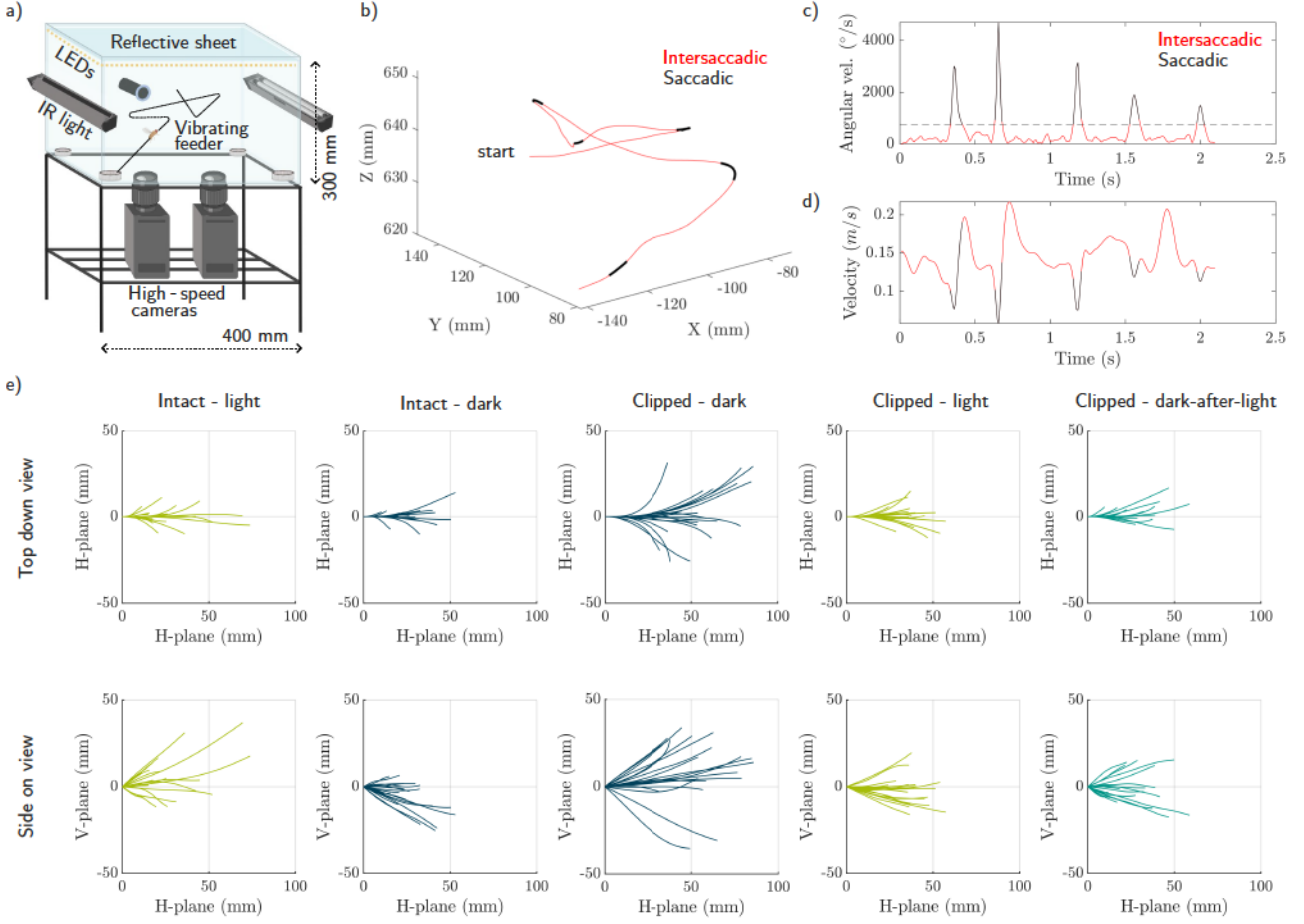


Figure 3.2: (a) Experimental enclosure for recording flight trajectories whilst controlling the available lighting. Omitted from this figure are: a checkerboard texture on the near and far enclosure walls, four blackout blankets that kept the entire setup in darkness, wet tissue paper in a cup that provided water and some humidity to the setup, and a temperature sensor inside the enclosure. The tube on the back wall had a mesh for air circulation and was used for inserting the animals. (b) Example trajectory from a *D. melanogaster* with intact wings flying in the light (I-L condition). (c) Angular velocity of the velocity vector during a flight trajectory where saccadic turns are defined as turns exceeding 750°s^{-1} (horizontal dashed line). (d) Translational velocity of the animal. (e) 20 intersaccadic sections of 160 ms from each condition. The intersaccadic sections were rotated to initially start in the same direction and are viewed from the top down and side on in the top and bottom row, respectively.

Four lures containing single-use disposable tissue soaked in malt vinegar were positioned at each of the lower four corners of the enclosure and were covered by a fine steel mesh so the flies could not access it. Underneath each lure was a Parallax 28822 vibration motor capsule that could vibrate the lure to encourage flight. A custom-designed printed circuit board (PCB) controlled the trigger for the stereo camera setup, the vibrating lures, and the strip of LEDs that went around the top of the enclosure. Before each experiment, a 274-frame calibration

(≈ 12.5 fps) sequence for the stereo-cameras was acquired using a 5×7 40 mm checkerboard. A temperature sensor (RS PRO-USB-1) was configured to record the temperature every 10 minutes and placed inside the enclosure, along with the vibrating feeders, and damp tissue to provide water and humidity.

Three visible lighting conditions were used for wing- and/or aristae-damaged flies throughout the study: LED lights continuously off from the start of the experiment (Dark, D), LED lights on (Light, L), and subsequent dark-after-light (DAL) where LED lights were turned off during video acquisition and on when the trajectories were not being recorded. The DAL condition always followed the L condition.

To eliminate confounding effects of local luminance adaption or contrast adaption, preliminary data collection involved two dark-after-light (DAL) conditions: (i) lights off for the entire 8-hour duration, and (ii) lights off only during video recordings. The only distinction was increased proclivity for flight under the latter condition (Figs. 3.3c and 3.3d) - hence the results shown are collected under condition (ii).

For flies with intact wings, no distinction was made between animals in the dark and the dark-after-light since there was no change to the flight dynamics with which they had been reared. Flight trajectories were recorded in the light (4 hrs), then in the dark (4 hrs), then for 4 hrs where LED lights were turned off during video acquisition. This cycle of 12 hrs was repeated to complete one 24 hr period. Initially 4 hr intervals were chosen to fit within the circadian cycle; however, there were no noticeable differences in flight propensity or trajectory kinematics for flies subjected to three 8-hour intervals. Therefore, 8 hr intervals under arista- and/or clipped-wing conditions were used to maximise the number of trajectories collected per experiment (since the dark/light sequence could not be repeated).

Clipped-wing flies were subjected to dark conditions for 8 hrs, light for 8 hrs, and then DAL for 8 hrs. To exclude nutritional state as a confounding factor a final control experiment was included in which clipped-wing flies stayed the full 24 hrs in the dark.

Finally, the internal body clock of *D. melanogaster* has been shown to influence the activity of walking animals (Dubowy and Sehgal 2017). To further exclude the possibility of circadian rhythm affecting the results, the start time of the 24 hr experiments was staggered between

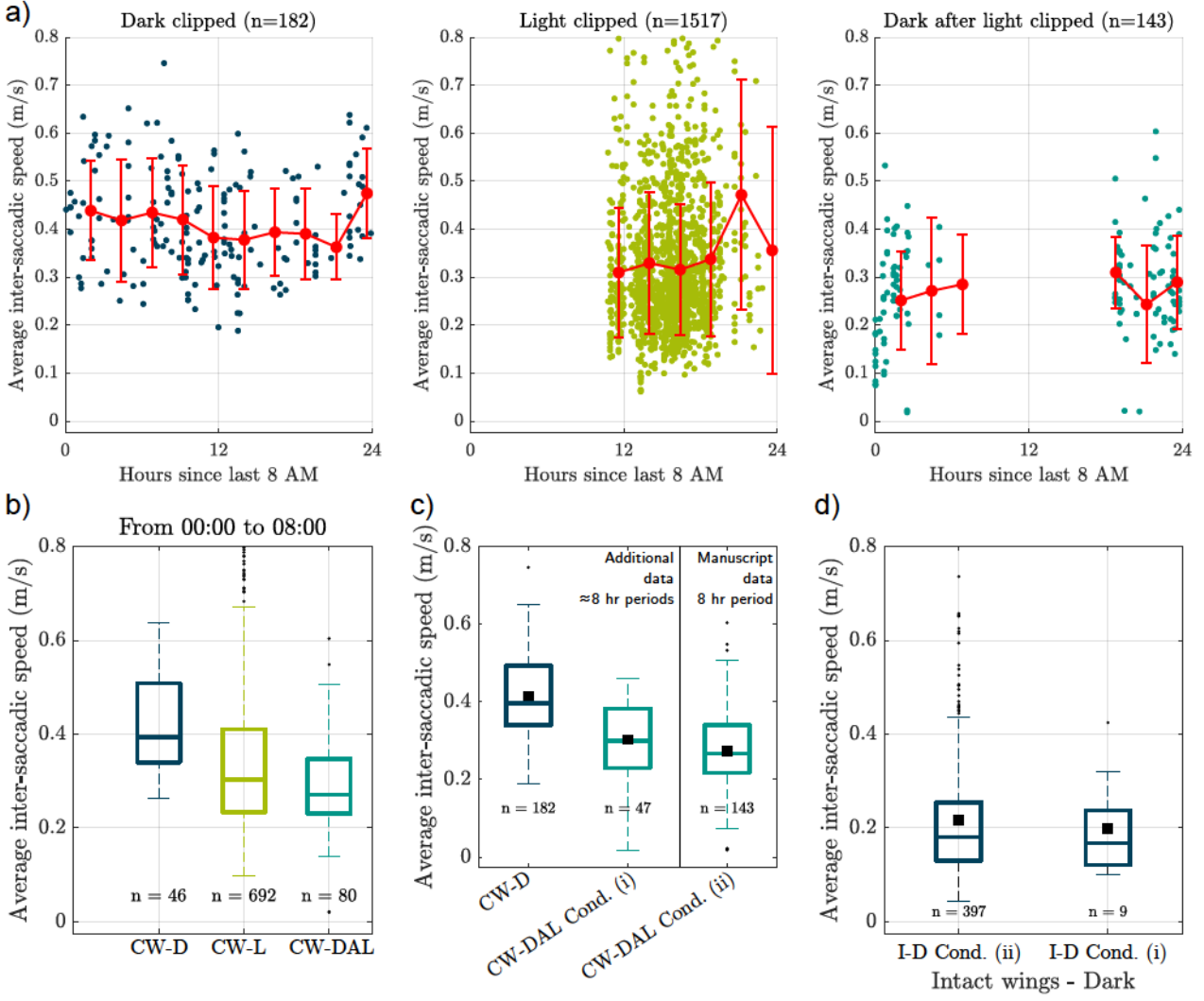


Figure 3.3: (a) Average intersaccadic speed at different times for clipped wing animals in each lighting condition. (b) Average inter-saccadic speeds for observations that all occurred between midnight and 8 am at the same point in the circadian cycle. (c) The average intersaccadic velocity of clipped wing animals flying in the dark (CW-D); and in the dark-after-light (CW-DAL). Condition (i) is the lights off the whole time, and condition (ii) is they are only off when recording videos. (d) Condition (i) and (ii) for the intact animals in the dark.

9:30 and 18:30. Looking at a subset of data where the three conditions occurred at the same time between midnight and 8 am the trend was consistent, thus excluding circadian rhythm as an explanation for these results (Figs. 3.3a and 3.3b).

3.2.3 Tracking and analysis

The first step in the tracking was the camera calibration of the stereo configuration, to allow pixel locations in the two cameras to be transformed into an XYZ position in a global coordinate

frame. The points between the checkers were extracted using the method described in (Geiger et al. 2012) with a minimum corner metric threshold of 0.2. Using the extracted points, the intrinsic and extrinsic camera parameters were estimated assuming no lens distortion. This is followed by a nonlinear least squares minimisation where all parameters are estimated with no assumptions about the lens distortion. The mean reprojection error of all the calibrations was 0.33 pixels, and the maximum mean reprojection error of any of the calibrations was 0.83 pixels. Then, a mask was determined that highlights the tracking area.

Each pair of videos was run through the tracking algorithm. For each video pair, the relevant mask and camera parameters were loaded. Then, 20 frames equally spaced from the beginning to the end of each video were averaged to generate a background average frame. Each video frame from each camera had the average subtracted and was binarised with a threshold scalar luminance value of 0.05 (Otsu 1979). Closed areas between 3 and 16 pixels were stored. For each detected object, a Kalman filter was used to determine when consecutive detections were a trajectory by assigning costs based on not being assigned to a trajectory and the distance between the center of consecutive detections. When the trajectories identified on both cameras had significant temporal overlap, they were triangulated for reconstruction in 3D (Fig. 3.4a). The 3D trajectories were then saved provided that the root mean square error of the reconstruction was less than 10 mm.

Following the extraction of trajectories, a cubic smoothing spline (using MATLAB's `csaps` function (MathWorks 2025b)) was applied with a smoothing parameter set to 0.99999. This parameter was deliberately chosen as it represented the highest degree of smoothing achievable without leading to the underfitting of saccades (Fig. 3.4b). Trajectories were removed if the flies were walking, which was determined by speed and proximity to the sides of the enclosure, and removed if they were ascending or descending more than 45° from horizontal. A threshold of 750° s^{-1} was applied to the turning rate of the velocity vector during each trajectory to determine if the animal was saccading as shown in Figure 3.2. This was chosen to differentiate spikes in turn rate from curvature in the CW-D condition. The only result that significantly varied when this threshold was changed was the number of intersaccadic data points. Finally, intersaccadic flight segments that lasted less than 80 ms were omitted from the analysis. Following these

omissions the number of intersaccadic flight segments is given in Table 3.1. The statistical

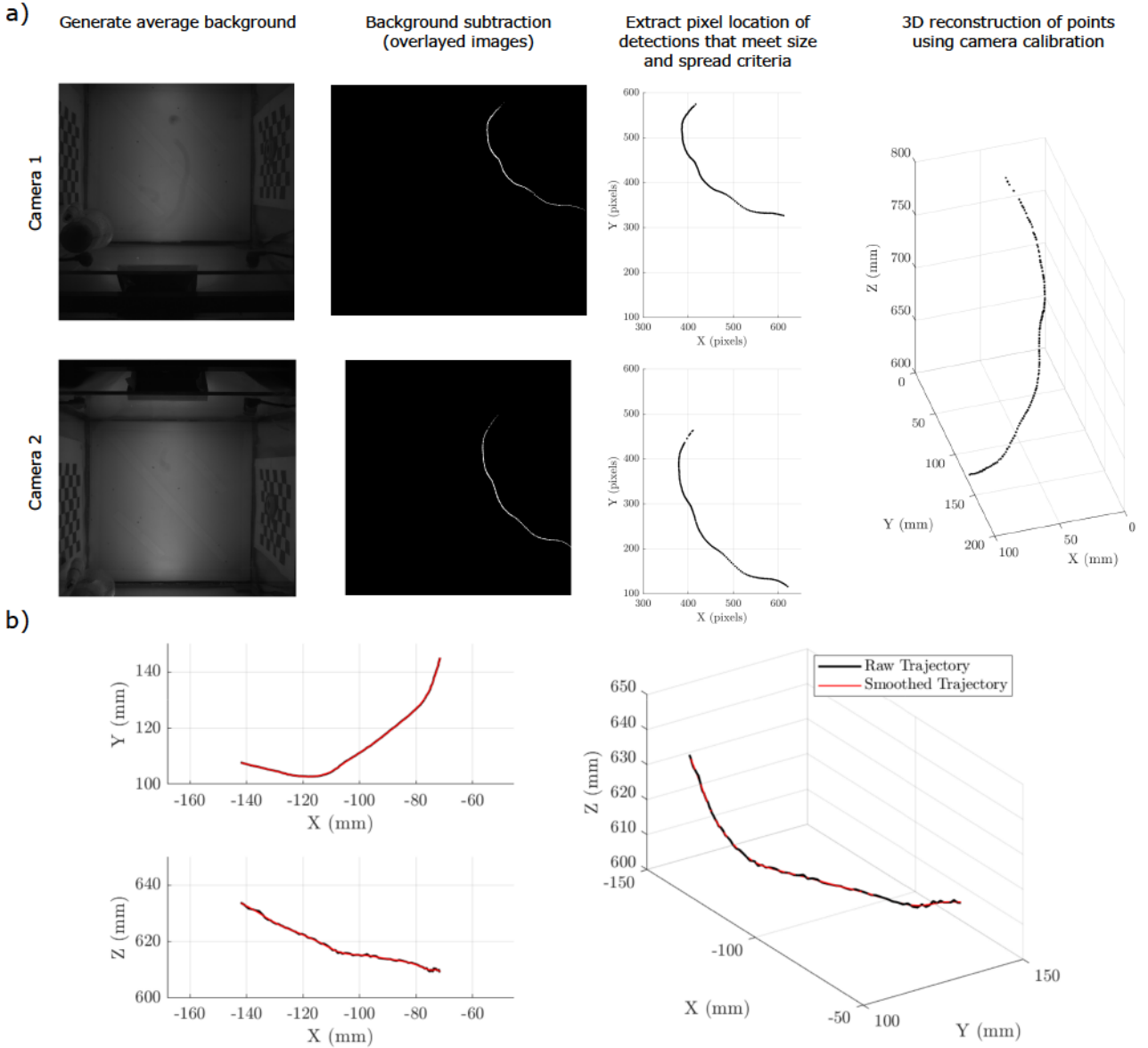


Figure 3.4: (a) The tracking and reconstruction procedure with an example trajectory. (b) Smoothed and raw trajectories, due to the camera placement the XY axis reconstruction contained very little noise. The Z-axis contained more noise but the difference between reconstructed and smooth rarely exceeded 2.5 mm.

significances of differences in the intersaccadic kinematics were calculated using a two-sample t-test (Welch's unequal variance test) where the variance was not assumed to be the same and, therefore, Satterthwaites approximation was used.

3.3 Results

3.3.1 Experiment 1: Lasting recovery of flight kinematics following light exposure in clipped-wing flies

Flies with asymmetric wing damage were previously shown to follow curved flight trajectories in the dark, but correct their wing stroke kinematics to maintain straighter flight in the light (Leonte et al. 2021). Any adaptive changes to the flight controller based on feedback would fail to correct flight deviations in the dark due to the lack of input to the motion vision system. But any adaptive changes to a feedforward controller would persist in the dark, as this does not rely on visual feedback. This hypothesis was tested by comparing the free-flight kinematics of wing-intact (I-) and clipped-wing (CW-) flies flying in the dark (D), in the light (L), and again in the dark after having flown in the light (dark-after-light, DAL). Intersaccadic flight segments of 160 ms in each condition are shown in Figure 3.2e.

In agreement with previous work (Leonte et al. 2021), the curvature, a measure of the deviation from a straight line, of inter-saccadic flight trajectories (Figure 3.5d) was significantly ($P < 10^{-4}$) larger for clipped-wing flies in the dark (CW-D) (mean $\pm$ std: $4.3 \pm 1.9 \text{ m}^{-1}$), compared with wing-intact flies flying in the light or the dark ($3.4 \pm 1.3 \text{ m}$, and $3.5 \pm 1.3 \text{ m}^{-1}$, respectively). This increase in curvature in clipped-wing flies was compensated for in the light ($2.9 \pm 1.3 \text{ m}^{-1}$), consistent with (Leonte et al. 2021). However, flies returned to the dark, after flying in the light, in the CW-DAL condition flew with curvature ($4.0 \pm 1.6 \text{ m}^{-1}$) insignificantly ($P > 0.05$) different from clipped-wing flies in CW-D. This indicates that compensation for inter-saccadic curvature does not persist after returning to the dark, contrary to the original hypothesis.

However, the curvature describes the resulting geometry of flight trajectories but does not account for the dynamics at which flies progress through these trajectories. Centripetal acceleration is the component of acceleration normal to the velocity vector (Figure 3.5a). The results show that the centripetal acceleration of clipped-wing flies in the dark was significantly higher ($1.8 \pm 1.0 \text{ ms}^{-2}$; $P < 10^{-4}$) than wing-intact flies flying in either the light or dark (0.7 ± 0.6 and $0.7 \pm 0.4 \text{ ms}^{-2}$, respectively). When the clipped-wing flies were exposed to

visual contrast in the light their centripetal acceleration reduced ($0.9 \pm 0.6 \text{ m s}^{-2}$) closer to the controls, and significantly lower than in the dark ($P < 10^{-4}$). Furthermore, the reduction of centripetal acceleration persisted upon returning to the dark, after flying in the light, ($1.1 \pm 0.6 \text{ m s}^{-2}$). CW-DAL condition flies flew with a significantly lower average inter-saccadic centripetal acceleration than D condition flies ($P < 10^{-4}$). This suggests that whilst flight trajectories geometrically are similarly curved for CW-D and CW-DAL, the centripetal acceleration (and therefore turning force) is lower for the CW-DAL condition than CW-D. Since centripetal acceleration corresponds to velocity squared multiplied by curvature, this indicates that the only substantial persistent change in trajectory kinematics relates to flight speed, rather than curvature.

Indeed, Figure 3.5b shows the average flight speed of clipped-wing flies in the dark was significantly higher ($41 \pm 10 \text{ cm s}^{-1}$; $P < 10^{-4}$) than wing-intact flies flying in either the light or dark (22 ± 14 and $20 \pm 9 \text{ cm s}^{-1}$, respectively, in line with reported values in Rohrseitz and Fry 2011). Clipped-wing flies flying in the light (CW-L) were able to compensate for the increased flight speed, flying at a significantly lower flight speed ($33 \pm 16 \text{ cm s}^{-1}$) than clipped-wing flies in the dark ($P < 10^{-4}$) and at a speed closer to wing-intact controls. The reduction of flight speed persisted upon returning to the dark, after flying in the light (CW-DAL) ($27 \pm 10 \text{ cm s}^{-1}$). This indicates that compensation for flight speed does persist after returning to the dark, which accounts for the observed changes in centripetal acceleration.

Why do clipped-wing flies fly faster than wing-intact or light-exposed flies? One explanation could be that clipped-wing flies in the dark have ineffective pitch control and lose altitude with gravity, resulting in higher speeds. However, the elevation angle of the velocity vector relative to the xy plane (Figure 3.5e) was significantly higher for clipped-wing flies flying in the dark ($8.3 \pm 18 \text{ deg}$; $P < 10^{-4}$) compared to wing-intact controls ($1.3 \pm 14 \text{ deg}$, and $0.0 \pm 16 \text{ deg}$ for light and dark, respectively). Clipped-wing flies therefore flew on average upward against gravity. This elevated angle of the velocity vector was returned to control levels in clipped-wing flies flying in the light. However, the reduction did not persist when they were returned to the dark-after-light; there was no significant difference between clipped-wing flies flying in CW-D or CW-DAL conditions ($P > 0.05$). This suggests that gravity does not explain the persistent

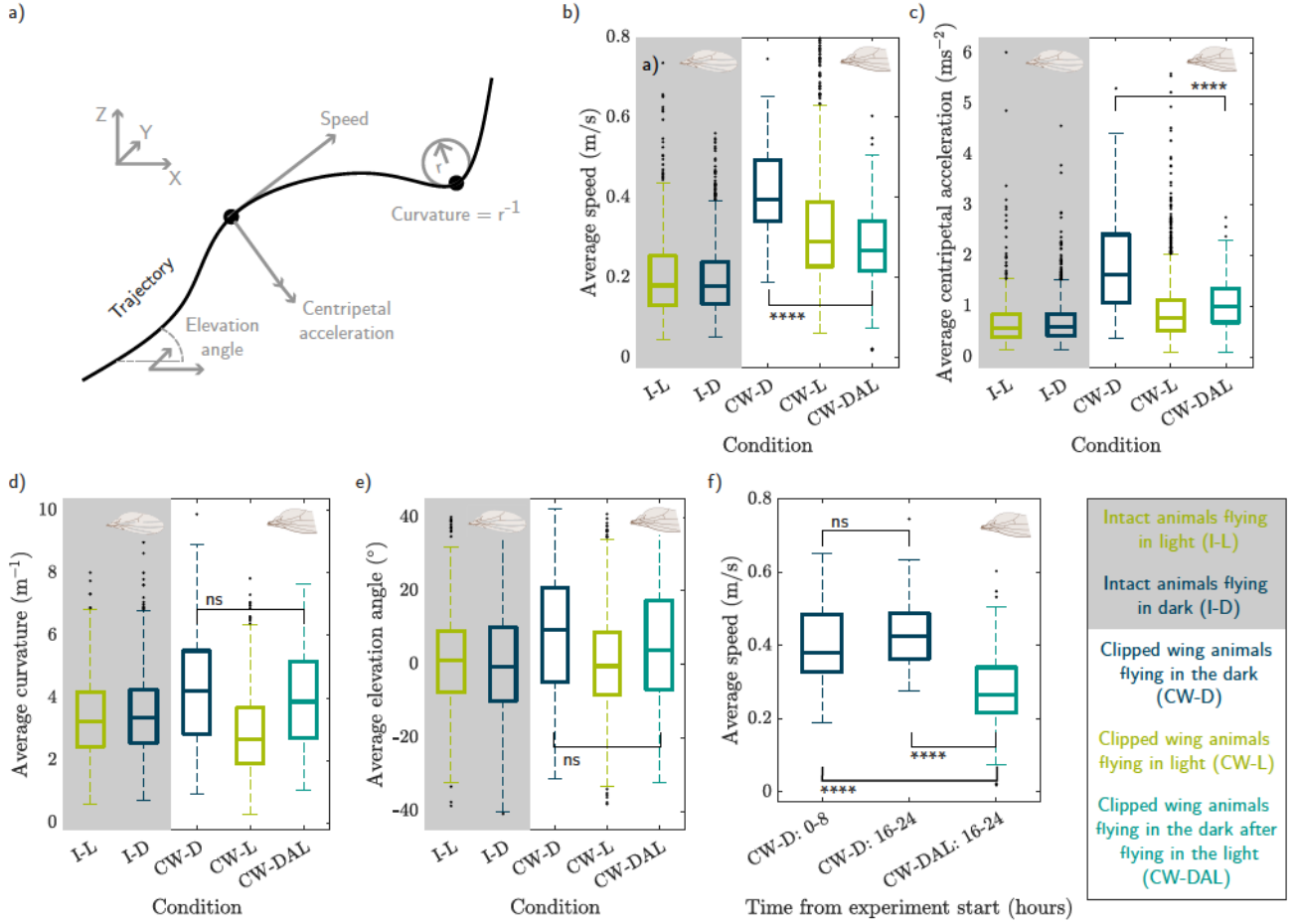


Figure 3.5: Intersaccadic kinematics for intact and clipped-wing *Drosophila* flying in the light, dark, and dark-after-light. **** indicates $P < 10^{-4}$, and ns indicates $P > 0.05$. The whiskers indicate the upper and lower quartiles. (a) Is a diagram of a demonstrative trajectory showing the centripetal acceleration, curvature, and speed. All the variables were calculated in 3D. (b) shows average intersaccadic speed for the 5 conditions: intact wings in the light (I-L); intact wings in the dark (I-D); segments from animals with a clipped wing flying in the dark since the wing was clipped (CW-D); clipped wing in the light (CW-L); and clipped wing in the dark, after having flown in the light (CW-DAL). (c), (d), and (e) show average intersaccadic centripetal acceleration, curvature, and elevation angle, respectively. (f) shows the average intersaccadic speed for flies in the CW-D condition for the first 0 - 8 hours of the experiment and the final 16-24 hours of the experiment, next to flies in the CW-DAL condition all from 16-24 hours of the experiment.

change in speed from the dark to the dark-after-light condition. The reason for increased flight in CW-D remains unresolved and is addressed in the discussion.

What changes between CW-D and CW-DAL for the speed to reduce so significantly? One potential explanation for the reduced flight speeds in CW-DAL conditions, compared to CW-D, was that clipped-wing flies reduced their speed as a function of time, either due to fatigue or longer periods of food deprivation. Thus, CW-DAL condition flight speeds were compared with

CW-D flies flying in the same time window (16-24 hours) but having experienced only dark for the full 24 hrs (Figure 3.5f). There was no significant difference in the flight speed of flies in continual dark compared with the initial 8 hrs of dark ($P > 0.05$), whereas flight speed for CW-DAL was statistically significantly different from both the 0-8 and 16-24 hrs dark-only condition ($P < 10^{-4}$). This suggests that the trend of reduced flight speed in the clipped-wing DAL condition was not a function of time, rather it corresponds to a lasting change in the flight speed controller.

To summarise, there is a clear change in speed from flying in the dark with one wing clipped, to flying in the dark with one clipped after having flown in the light. Clipped-wing flies flying in the dark have a 104 % increase in average speed from wing-intact controls. In the CW-DAL condition, this increase is 35 % from the control group. The adaptation that occurred when visual feedback was available therefore mitigated 67 % of the increase in speed caused by the change in system dynamics.

3.3.2 How could the controller change?

How can the persistent reduction in free-flight speed in clipped-wing flies returning to the dark after exposure to the light (CW-DAL) be accounted for? Figure 3.6 shows a hypothesised macroscopic model for the insect speed control system based on Chapter 2 (Campbell, Lin, and Krapp 2023a; Krapp and Wicklein 2008; Kim, Fitzgerald, and Maimon 2015; Kim, Fenk, et al. 2017; Taylor and Krapp 2007). The flight dynamics block represents a model for the resulting movement following a given motor command. The resulting combination of movement and external factors is sensed by the mechanosensory and the visual system. For stabilising reflexes, the mechanosensory and visual control systems sense disturbances, and initiate corrective muscle commands to mitigate course deviation. For goal-directed movements, it is hypothesised that a feedforward controller sends the muscle commands to execute the movement; the expected visual sensor response is then cancelled by the forward model. The mechanism for adaptive feedforward control is not dependent on the presence of a forward model. However, a forward model would likely provide a more convenient error signal to guide adaptation.

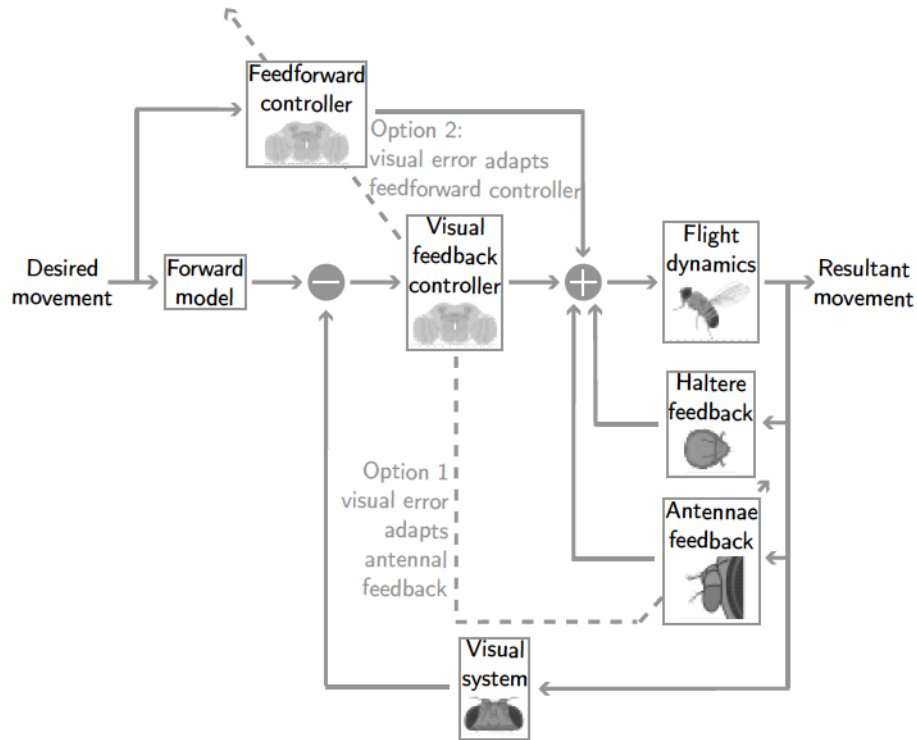


Figure 3.6: Hypothesised phenomenological model of the insect control system containing visual and mechanosensory stabilising systems in the inner-loop, with a feedforward and forward model component in the outer-loop to accommodate desired movements. Given that a change in speed control was observed, the haltere feedback pathway can be excluded leaving two possible adaptation mechanisms. Option 1 is that the visual error adjusts the antennal feedback system to better control speed when vision is denied. Option 2 is that the induced visual feedback error adapts a feedforward controller - which will also still operate when vision is denied.

Option 1: Adaptation of Mechanosensory Feedback

The first of the two plausible adaptation schemes in Figure 3.6 uses the visual feedback error to adapt the mechanosensory feedback system. Mechanosensation of flight speed is measured by the antennae, which can sense changes in airflow (Taylor and Krapp 2007). Tethered flying flies adjust their wingbeat amplitude (and thus thrust) in response to changes in frontal airflow velocity, but this response is absent following antennae removal (Burkhardt and Gewecke 1965; Niehaus 1981). However, more recent work examining free-flight behaviour in a wind tunnel where both airspeed and visual scenery could be controlled showed the antennae in *D. melanogaster* have only a transient effect on flight control (Fuller et al. 2014). Another mechanosensor thought to be involved in speed control are wind-sensitive hairs. Most of the work on wind-sensitive hairs has been done in *Locusta* (Taylor and Krapp 2007), and there has been little work in *D. melanogaster*. However, one study showed that tethered *D. melanogaster*

with their antennae bilaterally deafferented were no better at orienting upwind than dead flies in the same tethered setup (Budick, Reiser, and Dickinson 2007).

The halteres sense coriolis forces and therefore the angular acceleration/velocity. The halteres are not laid out for measuring translational velocities, and there is no evidence that they are involved in sensing speed (Yarger and Fox 2016). The halteres will be playing a critical role in stabilising flight, and the *D. melanogaster* would not be able to fly in the dark, with a damaged wing, without the halteres. But it is unlikely that they are responsible for the persistent change in the speed controller observed in Experiment 1.

To summarise, to my knowledge, there is no literature stating that *D. melanogaster* have wind-sensitive hairs that are involved in speed control, and halteres do not measure translational velocity. However, antennae have been found to respond to wind speed, where the response was described as transient by Fuller et al. (2014), Taylor and Krapp (2007), and Mamiya et al. (2011), and steady-state by Burkhardt and Gewecke (1965) and Niehaus (1981). Therefore, it is possible that visual motion information is used to adjust the antennae feedback control system to enhance its performance in case vision is unavailable (option 1, Fig. 3.6).

Option 2: Adaptation of Feedforward Control

The second of the two plausible adaptation schemes uses the visual feedback error to adapt the feedforward controller (Fig. 3.6). Adaptive feedforward control is well documented in humans (Bastian 2006; Pisotta and Molinari 2014; Fujita 2005). In insects, feedforward control has been demonstrated (Kukillaya and Holmes 2009; Schwyn et al. 2011; Viollet and Zeil 2013), although its capacity for adaptation is hitherto unknown. In the adaptive feedforward model, when clipped-wing animals fly in the dark, they fly faster than intact animals. When exposed to visual contrast in the light, there is an error between the expected and actual optic flow. This is compensated for by the visual feedback controller which acts to reduce the speed. The consistent visual feedback error is used to adapt the feedforward controller until the feedback controller is no longer compensating. Thus, when the flies are returned to the dark, their feedforward muscle commands more accurately reflect the changed system dynamics. As a result, they fly at speeds closer to the control group with intact wings.

3.3.3 Experiment 2: Speed controller changes with bilaterally ablated aristae

[Experiment 1](#) shows that the average intersaccadic speed is significantly different between clipped-wing animals flying in the dark (CW-D), and in the dark-after-light (CW-DAL). After exposure to visual contrast during flight, their speed is much closer to the speed of control flies. There are two possible mechanisms for how the speed control system could have changed to explain the results. The first mechanism is adapting the weights of the mechanosensory feedback system, so that mechanosensors involved in speed control better reflect the changed dynamics due to wing damage. The second mechanism is adaptive feedforward control.

To distinguish between two potential mechanisms for how the controller could be adapting, the experiments were repeated with animals that had the aristae on their antennae bilaterally ablated, as these are the primary mechanosensor involved in wind-speed sensing.

Figure [3.7a](#) shows the average intersaccadic speed for animals that have had one clipped wing (CW), and the arista on their antennae bilaterally ablated (ablated antennae, AA). When flying in the dark, the AA-CW-D average speed had a mean $\pm$ std, $89 \pm 22 \text{ cm s}^{-1}$; this is significantly faster than the speed of the control flies (I-D and I-L) presented in the first experiment ($P < 10^{-4}$). It is also significantly faster than flies in CW-D with wing damage only ($41 \pm 10 \text{ cm s}^{-1}$; $P < 10^{-4}$). The removal of the arista caused an increase in the average speed of clipped-wing animals flying in the dark. In addition, there was a significant increase in variance of speed for clipped-wing animals flying in the dark when the aristae were removed (Levenes: $P < 10^{-4}$) consistent with results reported by Fuller et al. ([2014](#)). When the clipped-wing arista-ablated animals were in the light (AA-CW-L), the visual feedback allowed them to compensate for the increased speed such that they flew significantly slower than AA-CW-D ($43 \pm 21 \text{ cm s}^{-1}$; $P < 10^{-4}$); this was still significantly faster, and significantly more variable than the controls ($P < 10^{-4}$). When they were returned to the dark, in dark-after-light (AA-CW-DAL), the lower speed persisted ($52 \pm 32 \text{ cm s}^{-1}$), and was not significantly different from AA-CW-L ($P > 0.05$), but lower than AA-CW-D ($P < 10^{-3}$). This implies that the speed controller has changed between AA-CW-D and AA-CW-DAL conditions despite not having aristae.

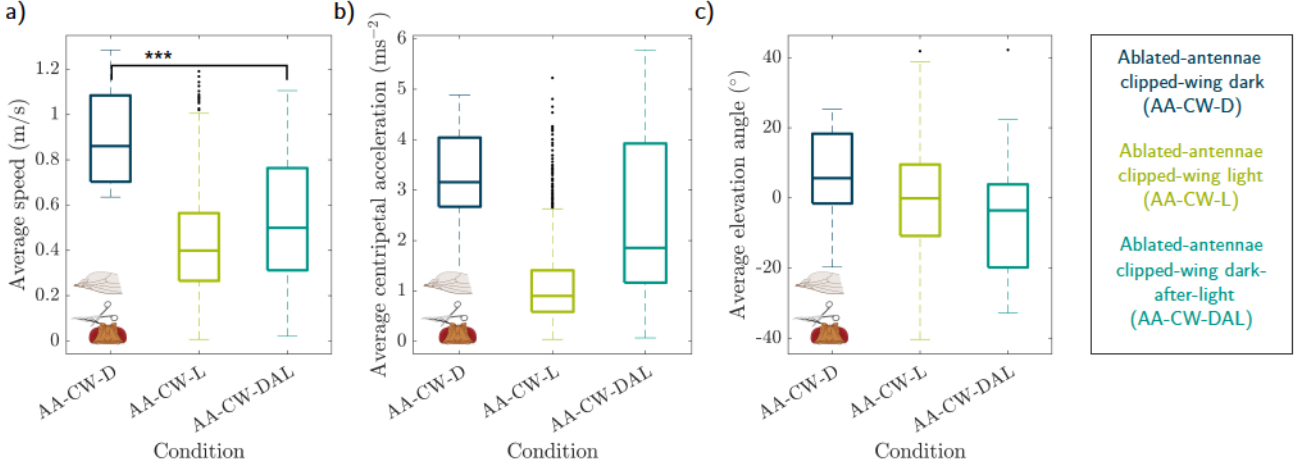


Figure 3.7: Intersaccadic kinematics for animals with one wing clipped and bilaterally ablated arista under the same experimental paradigm (D → L → DAL). *** indicates $P < 10^{-3}$, and the whiskers indicate the upper and lower quartiles. (a) The average inter-saccadic speed for animals flying in the dark (D), then flying in the light (AA-CW-L), and then flying back in the dark (AA-CW-DAL). (b), and (c) shows average intersaccadic centripetal acceleration and elevation angle, respectively.

The arista-ablated and clipped-wing flies had similar trends for centripetal acceleration (Figure 3.7b) and curvature as those with clipped wings only. In AA-CW-D the centripetal acceleration and curvature were higher than the controls ($3.2 \pm 1.1 \text{ ms}^{-2}$ and $3.7 \pm 1.4 \text{ m}^{-1}$, respectively; $P < 10^{-4}$). These were subsequently reduced when exposed to visual contrast in AA-CW-L ($1.1 \pm 0.8 \text{ ms}^{-2}$ and $2.8 \pm 1.4 \text{ m}^{-1}$, respectively). When the flies were returned to dark, AA-CW-DAL, the curvature subsequently increased again to be no different from AA-CW-D ($4.5 \pm 1.9 \text{ m}^{-1}$, $P > 0.05$). However, this time the centripetal acceleration for AA-CW-DAL was not significantly lower than AA-CW-D ($2.4 \pm 1.8 \text{ ms}^{-2}$ and $3.2 \pm 1.1 \text{ ms}^{-2}$, respectively, $P > 0.05$).

Again, the increased speed could not be explained by the animals in the dark flying preferentially downwards. Figure 3.7c shows that the average elevation angle of the velocity vector for animals in the dark was acting against gravity ($7.0 \pm 13 \text{ deg}$). This average elevation angle of the velocity vector was slightly higher than in AA-CW-L and in AA-CW-DAL, but not significantly (0.2 ± 15 , and $-4.5 \pm 19 \text{ deg}$, respectively, $P > 0.05$). Thus, the increase in speed in AA-CW-D compared to controls, AA-CW-L, and AA-CW-DAL conditions is not due to falling with gravity.

To summarise, it appears the antennae play a role in both steady-state and transient speed

control. When clipped wing animals with and without arista are compared, there are changes in steady-state speed and the variance of speed. Secondly, it appears that without the arista, the animal’s speed control system still adapts. The first experiment shows that adaptation occurs in the speed control system outside of the visual feedback pathway; here, I have shown that it also occurs outside of the antennal feedback pathway. These experiments further suggest that adaptation occurs in a feedforward controller, displayed as option 2 in Figure 3.6.

3.4 Discussion

In this chapter, I examined the ability of *Drosophila melanogaster* to adapt their free-flight kinematics following changes in flight dynamics due to asymmetric wing damage. [Experiment 1](#) found that clipped-wing flies fly faster in the dark compared with wing-intact controls, but that this increase in speed was compensated for when flying in the light. The speed compensation was sustained when light-exposed clipped-wing flies were subsequently returned to the dark (DAL), despite the absence of visual feedback. Overall, the change to the controller under DAL conditions compared to dark mitigated 67% of the speed increase associated with damaging the wing.

Two suggested adaptive control mechanisms could explain these experimental results (Fig. 3.6): (i) the animals could use the visual feedback error whilst they are in the light to adjust antennae feedback gains, or (ii) they could use the visual feedback error to adjust a feedforward controller. In a [second experiment](#), the experimental protocol was repeated with flies subjected to both asymmetric wing damage and bilateral removal of antennal aristae, which—besides vision—are the only sensors known to contribute to the control of flight speed in *Drosophila melanogaster* (Fuller et al. 2014; Taylor and Krapp 2007; Burkhardt and Gewecke 1965). Indeed, aristae-ablated clipped-wing flies fly with much higher and more variable speeds than aristae-intact clipped-wing flies, suggesting disruption of speed mechanosensation, in agreement with previously reported observations (Fuller et al. 2014). However, when aristae are removed, the distinct change in speed control from dark to dark-after-light conditions persists to a similar extent as in aristae-intact flies. These persistent changes observed to the flight speed controller

in DAL suggests that wind-sensing does not *solely* underlie this adaptation. While we cannot rule out the possibility that the adaptation occurs within the now-redundant antennal feedback pathway, any motor adjustment in the absence of sensory input would still constitute adaptive feedforward control.

Interestingly, the compensation of other free-flight trajectory parameters was not sustained upon returning clipped-wing flies to the dark. Both the inter-saccadic trajectory curvature and elevation angle of the velocity vector were significantly higher in clipped-wing flies flying in the dark compared to controls. However, whilst both these kinematic parameters were compensated for in the light, flight trajectories in the dark-after-light were indistinguishable from trajectories in the initial dark condition.

Why is there only evidence for adaptive feedforward control in speed, but not for trajectory curvature or elevation? One of the main advantages of cancelling expected sensor responses with a forward model is to prevent sensor saturation (see Sec. 2.3.4). Neurons have finite output ranges, but by cancelling expected sensor responses to re-afference, any constant response offsets are removed, thus maximising the sensor output range to signal non-anticipated inputs. When *D. melanogaster* are flying straight between saccades, any forward model of roll, yaw, and pitch movements should be zero, because the desired trajectory curvature and elevation is zero. However, when controlling for a non-zero speed, the visual sensors will be constantly measuring expansive translational optic flow. To ensure maximum sensitivity and avoid sensor saturation, a forward model would need to cancel this constant translational optic flow. The consequence of such forward model input is that the set-point of any feedback control becomes fixed at zero, i.e. the sensory error available for feedback corresponds only to the difference between expected and actual neuronal response to the optic flow. To actuate a non-zero speed, an additional non-zero feedforward command is therefore required. Adaptation of this feedforward command is what these results suggest accounts for the persistent compensatory reduction in flight speed in DAL.

In the case of trajectory curvature and elevation, since the desired state is zero, a feedforward command may not be required. Feedback alone may therefore be sufficient to control roll, pitch, and yaw. This difference in desired set-point may explain why there were no kinematic differences between dark and DAL in curvature and elevation, whereas these changes were

observed for speed.

This proposed combination of feedforward actuation commands with a forward model cancellation of the expected sensor response for controlling speed (Fig. 3.6) is consistent with previous results. It is well established that octopamine increases the response gain of optic flow-sensitive interneurons in flies (Longden and Krapp 2009; Suver, Mamiya, and Dickinson 2012; Maimon, Straw, and Dickinson 2010; Jung, Borst, and Haag 2011b). Whilst silencing octopaminergic neurons in *D. melanogaster* results in no change in average flight speed, corrective accelerations following unexpected optic flow perturbation was lower for octopamine-silenced animals compared to controls (Breugel, Suver, and Dickinson 2014). If speed was controlled with a fixed set-point and a feedback controller alone, octopamine-silenced flies would be expected to fly significantly faster than controls. This is because a larger magnitude of optic flow would be required to sufficiently activate the lower-gain sensors (Breugel, Suver, and Dickinson 2014). Alternatively, if average speed is controlled in feedforward (independent of the sensor response) and motion-sensitive interneurons receive forward model input (to cancel the expected sensory consequences of flying with a given speed), then these neurons would only signal unexpected deviations from the expected feedback caused by the feedforward command. This potentially explains why average flight speed is maintained but corrective accelerations are reduced in octopamine-silenced flies (Breugel, Suver, and Dickinson 2014), because the response of motion-sensitive interneurons to a given unexpected optic flow perturbation is reduced, resulting in a smaller error input to the feedback controller.

Although the hypothesis from these experiments may explain some previous results, they also show minor discrepancies with previous findings. Leonte et al. (2021) reported horizontal flight speeds of approximately 0.15 m/s (light) and 0.25 m/s (dark) in *Drosophila*, whereas the results here show median 3D flight speeds of about 0.18 m/s for both conditions. This difference in 3D versus 2D measurement does not account for the similar speeds we found between I-L and I-D (median horizontal speeds: 0.17 m/s for both). Two additional factors may explain the discrepancy with Leonte et al. (2021). First, their setup used a 5×5 cm checkerboard pattern on all four walls, producing an 11.4° visual stimulus at the arena centre, while the setup here used a 2.5×2.5 cm pattern on two walls, producing a 7.2° stimulus. Second, an analysis of covariance

(ANCOVA) revealed that ambient temperature affects flight speed differently in light and dark conditions (see Appendix B). It is possible that more similar light/dark temperatures would have resulted in speed trends closer to those reported by Leonte et al. (2021).

A second discrepancy with previous work is found in Muijres et al. (2017), who report average flight speeds of 0.15 m/s for intact and 0.11 m/s for clipped-wing *Drosophila hydei*, showing a decrease in speed after wing damage—unlike the I-L to CW-L results observed here. Firstly, the difference is unlikely to be due to species (*D. hydei* vs. *D. melanogaster*). Secondly, they used a 21 cm cylindrical arena and filmed only a 4×4×4 cm volume at the centre, providing a higher resolution than in this setup but restricting the analysis to a much smaller flight region. As a result, they did not separate saccadic and intersaccadic flight, instead excluding segments with large acceleration ($|a| > 2 \text{ m/s}^2$). Third, their higher frame rate allowed speed estimates based on only ~ 6 s of flight in the CW-L condition, whereas over 4.5 minutes of intersaccadic segments in CW-L are used here. Although, when repeatedly sampling 1% of the CW-L data, I-L appears faster than CW-L only 0.20% of the time, making it unlikely that their result is a statistical anomaly. A further difference is the wing-clipping method: here, one wing was clipped chordwise at the intersection of longitudinal veins L1 and L2, following Leonte et al. (2021). In contrast, Muijres et al. (2017) included both chordwise and spanwise cuts of different magnitudes. This likely accounts for the discrepancy; it is plausible that minor wing damage increases speed, while more extensive damage reduces it. It is also possible that spanwise cuts reduce speed and chordwise cuts increase it. Given that Muijres et al. (2017) do not make any distinction between the two types of damage in reporting their speed it is not possible to determine the impact of chordwise damage only. Furthermore, Leonte et al. (2021) also observes an increase in flight speed after clipping the wing, in-line with the results presented here.

Unlike Muijres et al. (2017), this setup was not designed to examine the biomechanics of speed control, however, the results leave two interesting biomechanical questions. Why do they fly faster, and why do they fly up, when in the dark with a clipped wing? The mechanism for controlling speed is done by adjusting the body pitch angle (Medici and Fry 2012). A higher body pitch angle is associated with a larger wing-damping force, which decelerates the animal (Medici and Fry 2012). Controlling vertical flight speed is not thought to be done by

changing body angle, but instead, wing-stroke (Fry, Sayaman, and Dickinson 2003; Sun and Tang 2002). Thus, these two observations are not contradictory and are likely a combination of the wing-stroke and body angle.

Adaptive feedforward control, rather than adaptive feedback, accounts for the discrepancy that goal-directed behaviours in insects can be highly adaptive (Wolf and Heisenberg 1991), but stabilising reflexes are not (Cellini, Ferrero, and Mongeau 2024), see Section 1.3.2. This is exemplified by the observation that insects can cope with an inversion of dynamics for orienting towards a bar (Wolf, Voss, et al. 1992), but were observed to spin uncontrollably for over an hour when wide-field background motion was inverted (Von Erich von Holst 1950). When a forward model cancels expected sensor responses from goal-directed behaviours a feedforward controller must correctly actuate the corresponding movements. In contrast, stabilisation reflexes by definition can only be executed with feedback control. The combination of adaptive feedforward control and inflexible feedback control may therefore explain long-standing observations that goal-directed behaviours, but not stabilisation reflexes, are adaptive.

Finally, there remains a significant open question on how this adaptation is achieved. Once there is a visual error, it is not clear how the animal knows which muscle, and feedforward motor command adjustments should be made. The two predominant engineering methods for adaptive feedforward control that could provide insights are model reference adaptive control (MRAC) and feedback error learning (FEL). Both methods use estimates of the system/system-gradients (or sensitivity derivatives) to determine the direction of adaption—as introduced in Section 1.2.5. However, these methods can not adapt to inversions (i.e. a sign change) of the dynamics, which insects on the other hand, were demonstrated to achieve (Heisenberg and Wolf 1984; Wolf, Voss, et al. 1992; Heisenberg and Wolf 2013). This is because if the dynamics invert, the sensitivity derivatives also do, meaning these classical adaptive control methods would adapt in the wrong direction. This challenge is addressed in the following chapter.

Chapter 4

Implicit adaptive feedforward control

4.1 Introduction

Individuals with facial palsy, which involves damage to the facial nerve that controls cheek muscles, can undergo treatment through hypoglossal nerve transposition. A surgeon cuts the damaged facial nerve and attaches it to a branch of the nerve to the tongue. Initially, when the patient moves their tongue, the cheek muscle is also activated. But, after time, the patient learns to control the tongue and cheek muscles independently (Socolovsky et al. 2016; Abdelghani, Lillicrap, and Tweed 2008). It has been suggested that flies can learn how to orient a bar in front of their visual field after an inversion of the dynamics (Wolf and Heisenberg 1986). As Chapter 3 suggests, insects may use adaptive feedforward control to obtain this level of controller flexibility—which is not yet matched by robotics. This chapter introduces implicit adaptive feedforward control as a method that (i) could potentially explain this flexibility, and (ii) could achieve some of that flexibility in robotic systems.

Many adaptive control methods (introduced in Sec. 1.2.5) such as model reference adaptive control (MRAC) (Nguyen 2018; Zhang and Wei 2017), feedback-error learning (Kawato 1990), and adaptive incremental nonlinear dynamic inversion (INDI) (Smeur, Chu, and De Croon 2016) require some prior knowledge of the system dynamics, or approximate internal models of the system, even if the models have variable parameters. I will refer to these methods as *explicit* adaptive control methods due to the prior knowledge, or approximate model, that determines the

direction in which they adapt (the sensitivity derivatives). Implicit adaptive feedforward control does not require any prior knowledge of the sensitivity derivatives and implicitly calculates the direction in which to adapt. In theory, this allows it to cope with inverted dynamics and to associate two previously unrelated dynamics.

Abdelghani and Tweed (2010) demonstrated that humans can adapt to new dynamics in joystick control experiments when the sensitivity derivatives are reversed. They proposed that this adaptability requires real-time learning of sensitivity derivatives, and can not be achieved using explicit adaptation techniques with fixed sensitivity derivatives (Abdelghani and Tweed 2010). Abdelghani, Lillicrap, and Tweed (2008) further suggested a mechanism for this learning process compatible with the nervous system, which does not involve the transport of filter weights (Abdelghani and Tweed 2010). This chapter introduces an implicit adaptive feedforward control mechanism based on a similar principle. The approach in this Chapter uses weight transport, which, unlike in the nervous system, is simple to implement in robotics¹. This approach of implicit adaptive feedforward control that uses weight transport (unlike the method in Abdelghani, Lillicrap, and Tweed 2008, which does not use weight transport) improves learning speed, reduces computational cost, and thereby increases applicability to engineered systems.

There are five motivating principles: (i) specifically *feedforward* control enables learning on hardware with the safety of the feedback controller ensuring stability, thus eliminating the simulation-to-reality gap, (ii) fast learning improves performance whilst remaining adaptive when the system dynamics change, unlike deep RL methods, (iii) less prior knowledge of the system decreases development time and increases flexibility (Hou and Jin 2013; Abdelghani, Lillicrap, and Tweed 2008), (iv) the feedback controller can be optimised for stability robustness without negatively impacting reference tracking performance, as the latter is largely determined by the feedforward component, and (v) highly flexible adaption mitigates scalability challenges for controlling different dynamics e.g. a drone carrying differently sized and shaped boxes (Dai, Lee, and Bernstein 2014).

¹In engineered adaptive filters, weights can be directly read, copied, or transferred from memory, whereas in biological neural systems synaptic strengths, though analogous to weights, cannot be accessed, duplicated, or inspected in such a straightforward way.

In this chapter, I describe the hypothesised insect fully-separable-degrees-of-freedom (FSDoF) control architecture for multi-input-multi-output (MIMO) systems. I then outline a novel implicit adaptive feedforward (IAFF) component of the controller, that unlike in Abdelghani, Lillicrap, and Tweed (2008), does use weight transport to improve performance. Finally, I demonstrate IAFF control in drone simulations, as well as on a drone platform for the first hardware demonstration of IAFF.

4.2 Fully-separable-degrees-of-freedom control

As outlined in Section 1.3.2 insects are thought to control their behaviours by combining forward models, feedback, and feedforward control (Kim, Fitzgerald, and Maimon 2015; Collett 1980; Webb 2004b; Fenk, Kim, and Maimon 2021). In this chapter, I will refer to the forward model as a reference filter in-line with typical control theory nomenclature. The fully separable degrees-of-freedom (FSDoF) control scheme (Fig. 4.1a) is described by the following equations for an M input and N output system:

$$\mathbf{y}(t) = P\mathbf{u}(t), \quad \mathbf{y} \in \mathbb{R}^N \quad (4.1)$$

$$\mathbf{u}(t) = \mathbf{u}_{\text{fb}}(t) + \mathbf{u}_{\text{ff}}(t), \quad \mathbf{u} \in \mathbb{R}^M \quad (4.2)$$

$$\mathbf{u}_{\text{fb}}(t) = C_1 \mathbf{e}(t), \quad C_1 \in \mathbb{R}^{M \times N} \quad (4.3)$$

$$\mathbf{e}(t) = \mathbf{y}_{\text{m}}(t) - \mathbf{y}(t), \quad \mathbf{e} \in \mathbb{R}^N \quad (4.4)$$

$$\mathbf{y}_{\text{m}}(t) = F\mathbf{r}(t), \quad F \in \mathbb{R}^{N \times N} \quad (4.5)$$

$$\mathbf{u}_{\text{ff}}(t) = C_2 \mathbf{r}(t), \quad \mathbf{r} \in \mathbb{R}^N \quad (4.6)$$

where each parameter is defined in Fig. 4.1a. As in Chapter 2, when the FSDoF controller is accurate, the reference filter (or forward model) should define the behaviour of the closed-loop reference tracking. The challenge is finding a feedforward controller and reference filter such that the following is true.

$$F\mathbf{r}(t) = PC_2\mathbf{r}(t) \quad (4.7)$$

If that equation holds, the reference tracking is defined by the following equations.

$$\mathbf{y}_m(t) = \mathbf{y}(t) \quad (4.8)$$

$$e(t) = u_{\text{fb}}(t) = 0 \quad (4.9)$$

$$\mathbf{y}(t) = P\mathbf{u}_{\text{ff}}(t) \quad (4.10)$$

Therefore, the reference tracking behaviour is determined by the feedforward actuation command

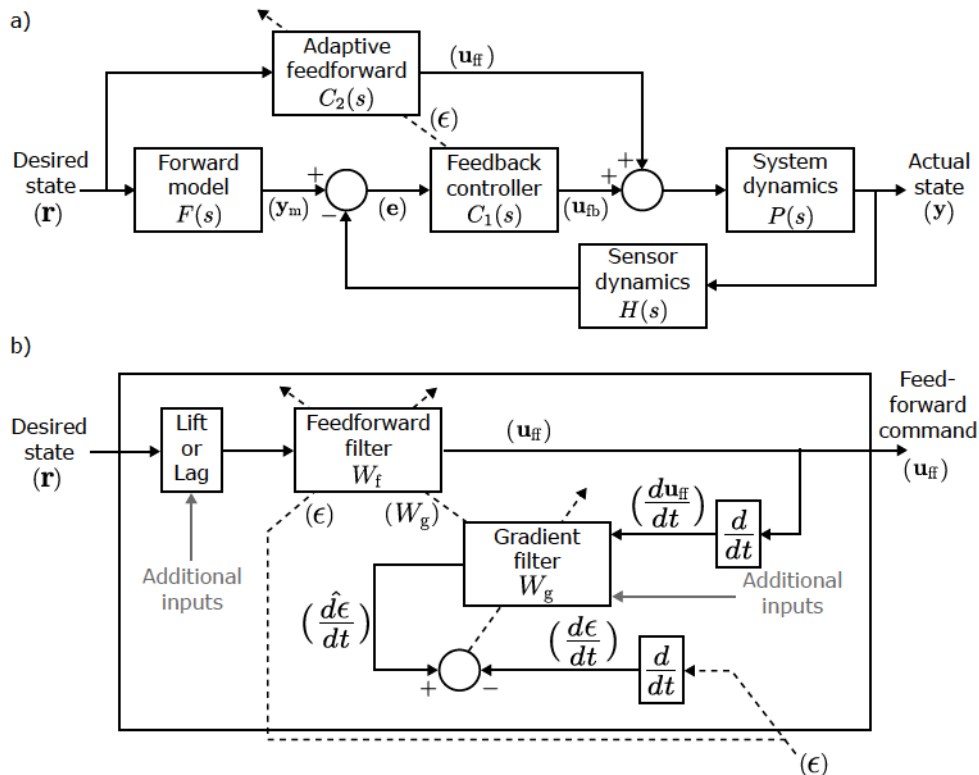


Figure 4.1: Implicit adaptive feedforward (IAFF) control. (a) Diagram of the fully-separable-degrees-of-freedom (FSDoF) control scheme. C_2 is a feedforward controller that should provide all actuation commands for any desired state change; F is a reference filter, previously referred to as a forward model that predicts the sensor response induced by the desired movement. C_1 is a feedback controller that adjusts the actuation command to remove error (ϵ) resulting from disturbances or inaccurate combinations of F and C_2 . The error (ϵ) is used to adapt the feedforward controller and can be set to either e , u_{fb} , or both. (b) Shows the implicit adaptation mechanism. The gradient filter estimates the $\frac{d\epsilon}{dt}$ based on $\frac{du_{ff}}{dt}$ and other inputs. From the chain rule, this makes part of the weights W_g an estimate of $\frac{d\epsilon}{du_{ff}}$. The resulting sensitivity derivative estimates adapt the feedforward filter, which generates the actuation command based on the desired state.

only (eq. 4.10). Given system dynamics $P(s)$, $C_2(s)$ should make the combined system act like the reference filter $F(s)$ (eq. 4.7). The approach of FSDoF control is to choose an $F(s)$ such

that (i) it reflects the desired overall system behaviour, and (ii) there is a realisable $C_2(s)$ for a given $P(s)$ such that Equation 4.7 will hold.

Typically, the desired system behaviour could be low-pass filter properties along the main diagonal of $F(s)$ and zero non-diagonal entries such that each state is controlled independently. When the system dynamics are known, determining what reference filter allows a realisable $C_2(s)$ to be learned can be derived using the concept of an interactor matrix (see: Kase and Mutoh 2009; Basel, Hirata, and Sugimoto 2007; Falb and Wolovich 1967).

Here, I assume that $P(s)$ is unknown and learn/adapt $C_2(s)$ to minimise some error $\epsilon(t) \in \mathbb{R}^K$ to satisfy Equation 4.7. The signal $\epsilon(t)$ can be the feedback error $e(t)$, the feedback actuation $u_{fb}(t)$, a combination of both, or any other measurable signal within C_1 . The key requirement is that if $\epsilon(t) \rightarrow 0$, then both $e(t) \rightarrow 0$ and $u_{fb}(t) \rightarrow 0$.

4.3 Implicit adaptation

To achieve fast adaptation, trial-and-error is not a viable solution for finding a feedforward filter C_2 that minimises the error. A systematic estimate of how to change $u_{ff}(t)$ to reduce the error $\epsilon(t)$ is required. When the system dynamics are known, the so-called sensitivity derivatives ($\frac{d\epsilon}{du}$) can be analytically derived (as done in: Basel, Hirata, and Sugimoto 2007; Kawato 1990; Miyamoto et al. 1988). Here this is done without a model, instead, using a time-evolving estimate of the sensitivity derivatives. The estimate is obtained using a second filter: the gradient filter (Fig. 4.1b) which estimates $\frac{d\epsilon}{dt}$ based on $\frac{du_{ff}}{dt}$ meaning, as a result of the chain rule, the filter weights provide an estimate for $\frac{d\epsilon}{du_{ff}}$.

4.3.1 The feedforward filter

The feedforward filter (Fig. 4.1b) is given an input vector $z_f(t) \in \mathbb{R}^{L_1}$. This input consists of the reference signal that has undergone processes known as lifting and/or lagging, as well as any additional information required by the system. Lagging involves the use of a tapped delay line, where past values of the reference signal are included, thus creating a time-expanded version of the reference signal. Lifting, on the other hand, applies multiple filters or operations

to the reference signal. Both methods transform the reference signal (desired state) into a higher-dimension representation. Without increasing the input space through lifting and/or lagging, the filter would lack sufficient input to optimally weight and generate the feedforward command $\mathbf{u}_{\text{ff}}(t) \in \mathbb{R}^M$. The feedforward filter weight matrix $W_{\text{f}} \in \mathbb{R}^{M \times L_1}$ is adapted to minimise the loss function $L(t)$ of the error $\epsilon(t) \in \mathbb{R}^K$.

$$\mathbf{u}_{\text{ff}} = W_{\text{f}} \mathbf{z}_{\text{f}} \quad (4.11)$$

$$L(t) = \frac{1}{2} \epsilon(t)^T \epsilon(t) \quad (4.12)$$

The first step is deriving the gradient descent update equations to find how to change the weights to minimise the loss function, with a learning rate α_{f} .

$$\begin{aligned} \frac{dW_{\text{f}}}{dt} &= -\alpha_{\text{f}} \frac{dL}{dW_{\text{f}}} \\ &= -\alpha_{\text{f}} \frac{dL}{d\epsilon} \frac{d\epsilon}{d\mathbf{u}_{\text{ff}}} \frac{d\mathbf{u}_{\text{ff}}}{dW_{\text{f}}} \\ &= -\alpha_{\text{f}} \left(\underbrace{\epsilon^T}_{1 \times K} \underbrace{\frac{d\epsilon}{d\mathbf{u}_{\text{ff}}}}_{K \times M} \right)^T \underbrace{\mathbf{z}_{\text{f}}^T}_{1 \times L_1} \in \mathbb{R}^{M \times L_1} \end{aligned} \quad (4.13)$$

Without an accurate model, the sensitivity derivatives can not be calculated in advance. Therefore, the gradient filter is used to estimate them, leaving the final update equation for the feedforward filter in Equation 4.15:

$$\boxed{\frac{dW_{\text{f}}}{dt} = -\alpha_{\text{f}} \left(\epsilon^T \frac{\hat{d\epsilon}}{d\mathbf{u}_{\text{ff}}} \right)^T \mathbf{z}_{\text{f}}^T} \quad (4.14)$$

Where $\frac{\hat{d\epsilon}}{d\mathbf{u}_{\text{ff}}}$ is an estimate of $\frac{d\epsilon}{d\mathbf{u}_{\text{ff}}}$ calculated by the gradient filter. In practice, the results shown here all include regularisation on each input-to-output:

$$\frac{dW_{\text{f}}^{m,l(n)}}{dt} = -\alpha_{\text{f}} \sum_{k=1}^K \epsilon_k \frac{\frac{\hat{d\epsilon}_k}{d\mathbf{u}_{\text{ff}}^m} z_f^{l(n)}}{1 + \left\| \frac{\hat{d\epsilon}_k}{d\mathbf{u}_{\text{ff}}^m} z_f^{l(n)} \right\|_2} \quad (4.15)$$

where, $m = 1, \dots, M$, $l = 1, \dots, L_1$, and $k = 1, \dots, K$.

4.3.2 The gradient filter

The gradient filter aims to estimate the sensitivity derivatives. The input to the gradient filter is denoted by $\mathbf{z}_g \in \mathbb{R}^{L_2}$ and is a concatenated vector of $\frac{du_{ff}}{dt}$ and additional useful variables for predicting $\dot{\epsilon}$ (Eq. 4.16). In practice, $\mathbf{z}_f$ was used as the additional input to the gradient filter. The weight matrix of the gradient filter $W_g \in \mathbb{R}^{K \times L_2}$ should, based on $\mathbf{z}_g(t)$, predict $\dot{\epsilon}(t)$ (Eq. 4.17) and can be trained on the measured $\dot{\epsilon}(t)$ using gradient descent (Eq. 4.18)

$$\mathbf{z}_g = \begin{bmatrix} \left[\frac{du_{ff}}{dt} \right] \\ [\mathbf{z}_{add}] \end{bmatrix} \quad (4.16)$$

$$\frac{\hat{d}\epsilon}{dt} = W_g \mathbf{z}_g \quad (4.17)$$

$$\boxed{\frac{dW_g}{dt} = -\alpha_g \left(\frac{\hat{d}\epsilon}{dt} - \frac{d\epsilon}{dt} \right) \mathbf{z}_g} \quad (4.18)$$

If the gradient filter error $\frac{\hat{d}\epsilon}{dt} - \frac{d\epsilon}{dt}$ is small, from the chain rule a subset of the weights matrix W_g will approximate the sensitivity derivatives.

$$W_g = \left[\frac{\hat{d}\epsilon}{du_{ff}} \quad \sim \right] \quad (4.19)$$

Again, in practice, regularisation was applied to this update equation for all implementations shown here:

$$\frac{dW_g}{dt} = -\alpha_g \left(\frac{\hat{d}\epsilon}{dt} - \frac{d\epsilon}{dt} \right) \frac{\mathbf{z}_g^T}{1 + \|\mathbf{z}_g\|_2} \quad (4.20)$$

4.3.3 Stability analysis

The use of two tightly coupled adaptive filters—where a subset of the gradient filter’s weights estimates the sensitivity derivatives for the feedforward filter—complicates standard Lyapunov analysis. Here, I discretise the filters’ loss functions to derive upper bounds on the learning rates. Neglecting regularisation, and letting $\hat{J} = \frac{\hat{d}\epsilon}{du_{ff}}$ and $J = \frac{d\epsilon}{du_{ff}}$ for notational simplicity, the

weight update and loss change at time step n are:

$$\Delta W_f[n] = -\alpha_f \hat{J}^T[n] \epsilon[n] z_f[n]^T \quad (4.21)$$

$$\Delta L[n] = \frac{1}{2} \epsilon[n+1]^T \epsilon[n+1] - \frac{1}{2} \epsilon[n]^T \epsilon[n] \quad (4.22)$$

The error at time $n+1$ can be approximated using a first-order Taylor expansion:

$$\epsilon[n+1] \approx \epsilon[n] + J \Delta \mathbf{u}_f[n] \quad (4.23)$$

Given that $\mathbf{u}_f[n] = W_f[n] z_f[n]$, and assuming z_f is slowly varying, i.e. $\dot{z}_f \approx 0$, the change in feedforward actuation can be approximated by:

$$\Delta \mathbf{u}_f[n] \approx \Delta W_f[n] z_f[n] \quad (4.24)$$

Dropping the notation $[n]$ and substituting Equations 4.21 4.23, and 4.24 into Equation 4.22 gives:

$$\begin{aligned} \Delta L(n) &= \epsilon[n]^T J \Delta W_f[n] z_f[n] + \frac{1}{2} (\Delta W_f[n] z_f[n])^T (J^T J) (\Delta W_f[n] z_f[n]) \\ &= -\alpha_f \epsilon^T J \hat{J}^T \epsilon \|z_f\|_2^2 + \frac{1}{2} \alpha_f^2 \|z_f\|_2^4 \epsilon^T \hat{J} J^T J \hat{J}^T \epsilon \end{aligned} \quad (4.25)$$

Imposing that the change in the loss function must be negative, a bound on the learning rate α_f can be obtained, given $\alpha_f > 0$ and $z_f \neq 0$:

$$\begin{aligned} 0 &\leq -\alpha_f \epsilon^T J \hat{J}^T \epsilon \|z_f\|_2^2 + \frac{1}{2} \alpha_f^2 \|z_f\|_2^4 \epsilon^T \hat{J} J^T J \hat{J}^T \epsilon \\ \alpha_f \epsilon^T J \hat{J}^T \epsilon \|z_f\|_2^2 &\leq \frac{1}{2} \alpha_f^2 \|z_f\|_2^4 \epsilon^T \hat{J} J^T J \hat{J}^T \epsilon \\ \epsilon^T J \hat{J}^T \epsilon &\leq \frac{1}{2} \alpha_f \|z_f\|_2^2 \epsilon^T \hat{J} J^T J \hat{J}^T \epsilon \end{aligned} \quad (4.26)$$

Assuming that $J\hat{J}^T \succ 0$, which is true when $\hat{J} \approx J$, and by following a similar procedure for the gradient filter (see App. A.1) the following upper bounds on the learning rates are obtained:

$$\alpha_f \leq \frac{2}{\|z_f\|_2^2 \|\hat{J}J^T\|_2} \quad \text{and,} \quad \alpha_g \leq \frac{2}{\|z_g\|_2^2} \quad (4.27)$$

The upper bounds on the learning rates given the use of regularisation are given in Appendix A.3.

The assumption that $\dot{z}_f \approx 0$ is not accurate; in practice, the filter input tends to vary more than the weights. As a result, the loss function does not continuously decrease; instead, it has temporary increases over time—as shown later in Fig. 4.3a. Nevertheless, for stability analysis, only the components of the loss function related to weight changes are relevant. The resulting loss change, without making this assumption, is provided in the Appendix A.2.

In addition, the assumption that $J\hat{J}^T \succ 0$ may not initially be true, as it requires that $\text{sign}(J) = \text{sign}(\hat{J})$. Whilst that may result in initial increases in the loss, the gradient filter is adapting to obtain $\hat{J} \approx J$. Therefore, if the sign of J changes, the loss will increase until the gradient filter has learned the new sensitivity derivatives and $\text{sign}(J) = \text{sign}(\hat{J})$ again.

The most significant challenge is that the limit on α_f is provided in terms of the sensitivity derivatives and their estimate (Eq. 4.27). These values are functions of time, the initial random weights, the dynamics, and the learning rates themselves (see App. A.4). This imposes a challenge on pre-determining a safe learning rate that guarantees stability.

4.4 Example implementation

Simulation parameters

The implicit adaptive feedforward (IAFF) controller is implemented on a two-input-two-output system defined by the transfer matrix in Equation 4.28.

$$P(s) = \begin{bmatrix} \frac{3}{60s+1} & \frac{6s}{25s+1} \\ \frac{10s}{50s+1} & \frac{2}{40s+1} \end{bmatrix} \quad (4.28)$$

where s is the complex frequency domain Laplace variable. The plant dynamics were modelled in MATLAB Simulink using ODE45 with a relative tolerance $< 10^{-6}$. The discrete controller used zero-order-hold which operated at 1 kHz. The feedback controller was optimised using the loop tuning method, with a decoupler and two PID controllers. The reference filter was a discretised diagonal matrix where each diagonal entry was a third-order low-pass filter with a cut-off frequency of 2 kHz.

The initial weights for the IAFF were a matrix of zeros for the gradient filter. The feedforward filter is initialised with a random matrix of values ranging from 0 to 0.02. The feedforward filter can not be initialised with zeros as it requires a non-zero temporal derivative of $\mathbf{u}_f(t)$ to adapt. The learning coefficients were 0.01 for both α_g and α_f (the affect of changing these values is detailed in the supplementary information). The input to the feedforward filter was a 100-element vector which consisted of two tapped-delay lines of length 50, one for each reference. The reference signal was a 300 Hz and a 200 Hz sine wave with magnitudes 1 and 1.2 for the two states, respectively. The error vector used to adapt the feedforward controller was the feedback actuation command $\mathbf{u}_{fb}$.

Simulation results

The gradient filter of the IAFF mechanism accurately predicts $\dot{\epsilon}$ after ≈ 0.2 s (Fig. 4.2a). This is despite the gradient filter weights continuously changing for the first ≈ 0.8 s (Fig. 4.2c). The difference highlights that as the feedforward filter changes, the sensitivity derivatives estimated by the gradient filter also change.

The IAFF reference tracking for both states starts with similar performance to the feedback controller (Fig. 4.2b) despite the randomly initialised feedforward filter. After 0.9 s the reference tracking by the IAFF was almost perfect (Fig 4.2b), and an improvement on the feedback controller. This was when the feedforward filter weights began to stabilise (Fig. 4.2d).

Whilst the improved performance of the IAFF required more actuation, by lowering the cut-off frequency of the reference filter the actuation/error trade-off can be determined for the given plant dynamics.

The initial weights in the gradient filter were always initialised as an array of zeros. As

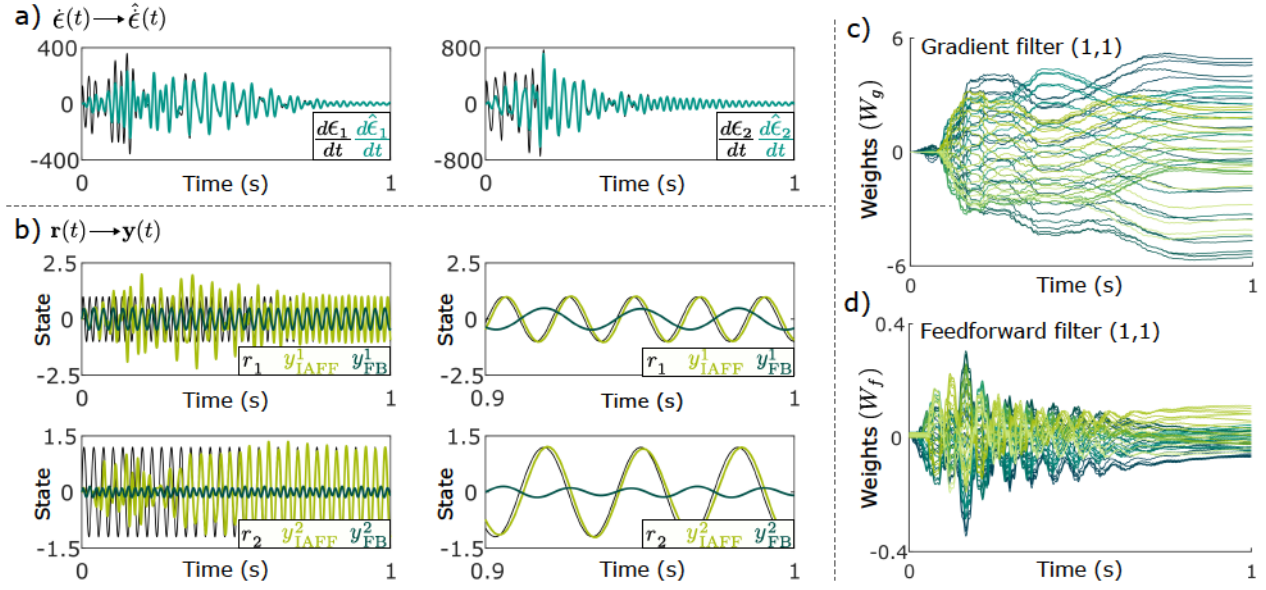


Figure 4.2: Example implementation of implicit adaptive feedforward (IAFF) control on a two-input-two-output system. (a) Rate-of-change of the two errors into the IAFF and their respective predictions by the gradient filter. (b) Reference tracking of both states by the IAFF system, and a feedback controller comparison over the 1 s simulation. (c) Gradient filter weight adaptation over time corresponding to $\frac{du_{ff}^{(1)}}{de_1}$. (d) Resulting weights of the feedforward filter from $r^{(1)}$ to $u_{ff}^{(1)}$

mentioned above, the feedforward filter must generate a non-zero output to have a signal that can be used to adapt. The standard initialisation for the feedforward filter consisted of random values between 0 and $S_W L_1^{-1}$, where S_W is an initial weights scale, set to unity unless specified otherwise. These random initial weights affect the learning process. Figure 4.3a shows the filtered mean square error (MSE) over time for ten different initial weight vectors in the feedforward filter. As suggested by the stability analysis, firstly, there are initial increases in the loss function before it reduces, probably due to an initial period where $\text{sign}(\hat{J}) \neq \text{sign}(J)$. Secondly, there are transient oscillations from the variable input vector $\dot{z}_f \neq 0$. As denoted by the dashed lines, the different initial weights led to variations in the time taken to adapt. As the learning rate increases, the time to adapt reduces as expected (Fig. 4.3b). All $\alpha_f \leq 0.0077$ were stable, independent of the initial weight vector for the 20 initial weight vectors tested. When the learning rate was 0.01, 65% of the 20 trials were stable, but 7 of the initial weight vectors led to unstable solutions. This highlights the clear upper bound on the learning rate to guarantee stability.

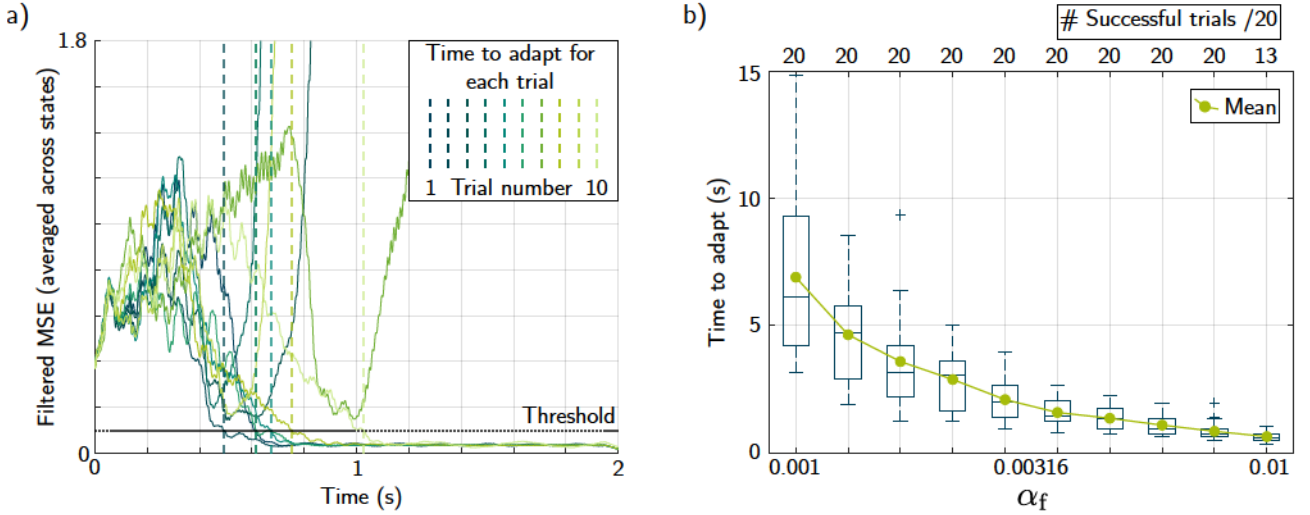


Figure 4.3: Evaluation of implicit adaptive feedforward control applied to the plant dynamics as described in Eq. 4.28. (a) Mean square error (MSE) averaged across states, filtered over time, using ten different randomly initialised weight vectors for the feedforward filter. The vertical lines indicates the adaptation time, defined as when the filtered MSE falls and remains below 0.1 units. Each of the ten trials is given a different colour. (b) This process of random initial weights is repeated 20 times, with 10 different logarithmically spaced learning rates for the feedforward filter (α_f) ranging from 0.001 to 0.01 (200 total simulations). The number out of 20 trials that successfully adapt is indicated at the top.

Figure 4.4 shows the results for the same simulation trials as in Figure 4.3 but with a learning rate of 0.001. In Figure 4.4b the Euclidean distance of the feedforward filter weight vectors from the mean weight vector over time is shown. Initially, the weights diverge considerably, indicating different trajectories in weight space. However, as the error reduces, the weights converge towards the mean. This suggests that the initial random weights affect the learning trajectory but have less impact on the final weights.

Lastly, the magnitude of the initial weights in the feedforward filter also affects the rate at which it adapts. Increasing the size of the initial weight vector reduced the time to adapt (Fig. 4.4c), but increased the initial control error (Fig. 4.4d). This is intuitive because increasing the magnitude of the feedforward command increases the strength of the relationship between the feedforward command and the reference tracking error which increases the accuracy of the prediction in the gradient filter. Thus, making the feedforward filter adaption direction more accurate. Due to the lower learning rate of 0.001 the unsuccessful trials in Figure 4.4c were not due to instability but were caused by the error not dropping below the threshold within the 25 s time window of the simulation.

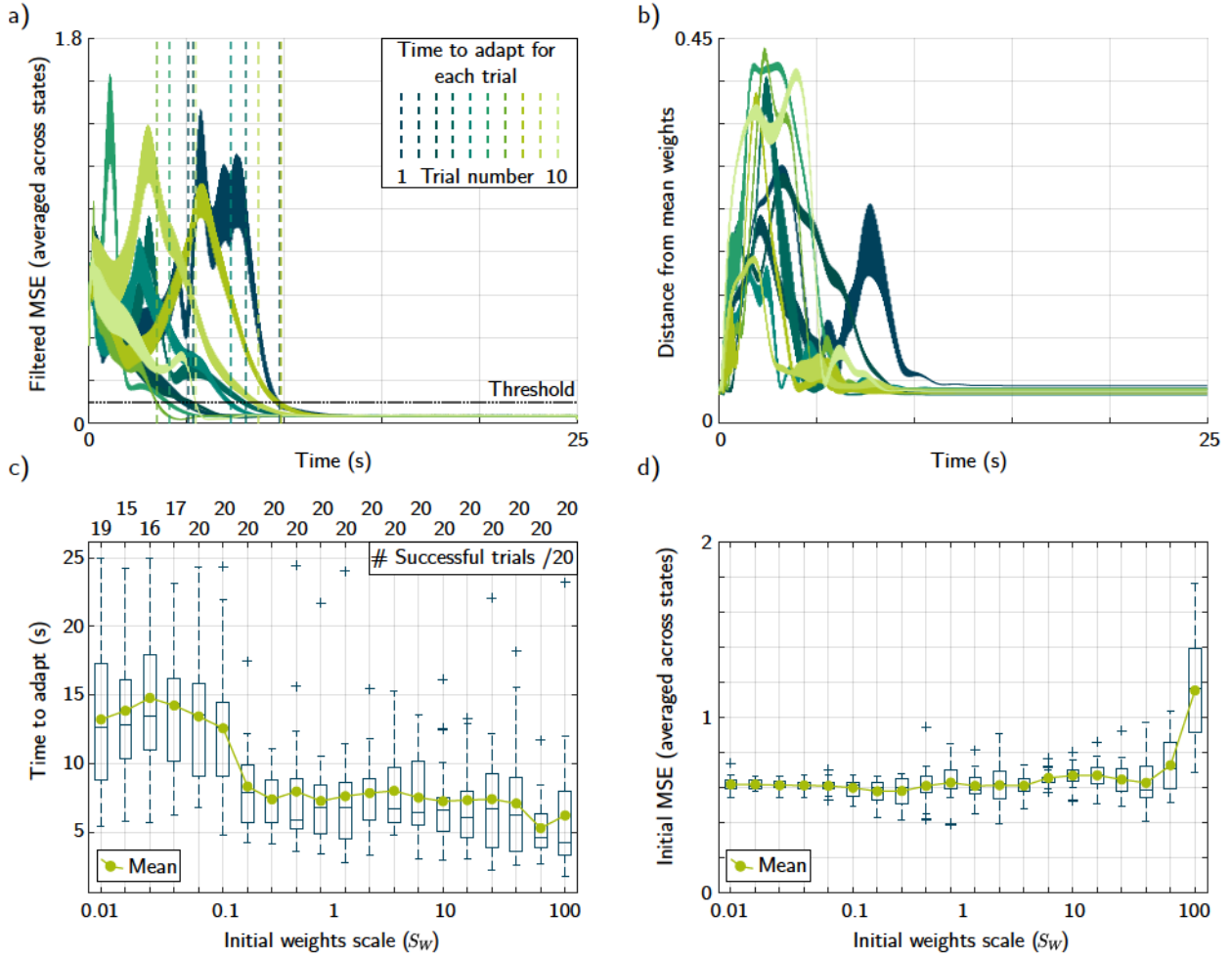


Figure 4.4: The effect of initial weight values on the implicit adaptive feedforward control of plant dynamics in Eq. 4.28. A feedforward filter learning rate of 0.001 is used to improve stability. (a) The filtered mean square error (MSE), averaged across states over 10 instances of randomly initialised feedforward filter weights. Each of the ten trials is given a different colour. (b) Euclidean distances between weight vectors from the mean over time. Initially, random initial weights deviate from the mean vector but converge closer to it after approximately 10 seconds. (c) The adaptation time relative to the magnitude of initial weight vectors. With a filter length of 50, unscaled initial weights are randomly and uniformly distributed between 0 and 0.02 ($1/50$). Scaling the initial weights by a factor of 10, results in a distribution between 0 and 0.2. (d) Initial MSE during the first second of simulation for variable scaling of the initial weight values.

4.5 Drone implementation

The implementation of implicit adaptive feedforward (IAFF) control is demonstrated on a Parrot Mambo minidrone. Both simulations and hardware demonstrations were used to assess the performance of IAFF in three tests: 1) pitch and roll control, 2) altitude control, and 3) handling damage to a propeller. The section is split into the setup and parameters used in

the simulations and on the hardware (Sec. 4.5.1). Then the three test results indicating the performance, are described in Section 4.5.2.

4.5.1 Drone setup and parameters

Simulations

The drone simulations were done in MATLAB Simulink and were derived from the `asbQuadcopter` model in the aerospace and support package for parrot minidrones (Inc. 2023b; Inc. 2023a; Sayre-McCord et al. 2018). The simulation includes: an environment model, non-linear airframe model, sensor model, and the flight control system which could be transferred easily between hardware and simulation. The model solver was ode3 Bogacki-Shampine run at 200 Hz, which is the same as the controller frequency.

The static environment model determined gravity, air pressure, temperature, and density.

The non-linear airframe model calculated the thrust, aerodynamic drag forces, and moments including the effect of rotor tilt due to flap motion (Pounds 2007; Riether 2016).

The simulation uses models of the sensors on the Parrot Mambo minidrone to mimic the measurement of the current state done by the real drone. Models were used for the inertial measurement unit (IMU), sonar altitude sensor, and optical flow sensor.

Finally, a flight controller connected the sensor outputs with the non-linear airframe model. The baseline feedback control system program was identical between the simulation and the real drone. The first stage of the flight controller was the state estimator. Sensor biases were subtracted from the IMU and the sonar altitude sensor, and the resulting signals were low-pass filtered. A complimentary filter combined visual and inertial sensor systems and was followed by Kalman filters, as described by Riether (2016). Following state estimation were six feedback PID controllers (Fig. 4.5), one controlled yaw, and another altitude. Further PID controllers controlled X and Y positions in a global coordinate frame, where the output was transformed into local coordinates for the pitch and roll PID controllers. The output of this baseline feedback control system consisted of actuation commands for pitch, roll, yaw, and thrust. The 6 PIDs used the default tuning (from Inc. 2023b), with the exception of the altitude controller, where the

integral action was included and the proportional and derivative gains were adjusted accordingly. The four actuation commands were put into a motor mixing matrix to calculate the commands to each of the drone's four motors (Fig. 4.5).

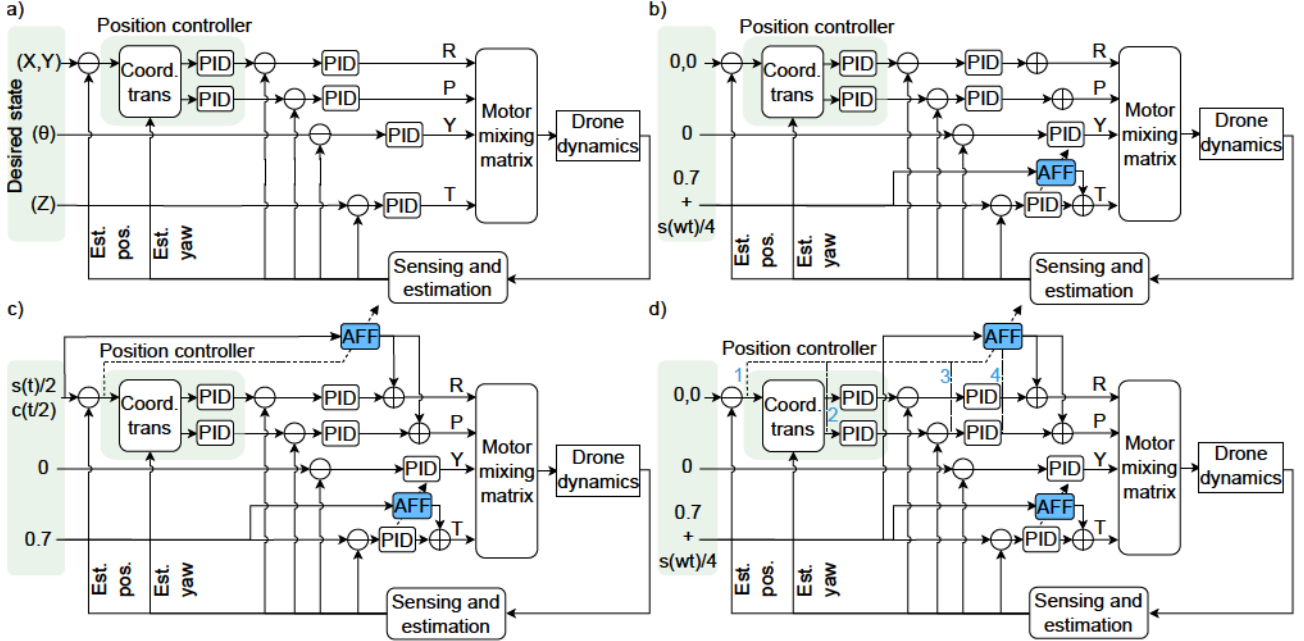


Figure 4.5: Drone control architectures used in simulation and on hardware. R, P, Y, and T refer to roll, pitch, yaw and thrust commands, respectively. $s()$ and $c()$ refer to $\sin()$ and $\cos()$, respectively. The coordinate transform in the position controller is from world to body coordinates. AFF refers to (implicit) adaptive feedforward. (a) Baseline feedback control architecture with no implicit adaptive feedforward mechanisms (IAFF). (b) Architecture used for the IAFF altitude control. (c) Pitch and roll IAFF scheme. (d) Different IAFF architectures were tested for the damaged propeller simulation and hardware tests, where each option for the error is denoted with a number. In both simulations and implementation on the drone all four error signals performed similarly. The results shown below were obtained with the position error used to adapt the IAFF (signal 1).

The two simulated implicit adaptive feedforward (IAFF) demonstrations, pitch/roll and damage handling, used slightly different architectures. For the pitch/roll simulation, the input into the feedforward architecture were the desired X and Y positions (Fig. 4.5c). In one simulation the desired position was lifted by nine discrete first-order high-pass filters with approximately logarithmically spread cut-off frequencies (bandwidths of: 100 rad s^{-1} , 70 rad s^{-1} , 30 rad s^{-1} , 10 rad s^{-1} , 7 rad s^{-1} , 3 rad s^{-1} , 1 rad s^{-1} , 0.7 rad s^{-1} , and 0.3 rad s^{-1}). Each discrete high-pass filter had a sample time of 0.005, the same as the controller loop frequency. In the other simulation the position reference was lagged by 30 tapped delays (Fig. 4.6a). For the damage control simulation, the input was the altitude (Fig. 4.5d) lifted by the same filters. In

all simulation results shown here, the position error was used for adapting. However, for the damage control simulation all four options in Figure 4.5d were tested. All four options yielded similar results. In the simulation, propeller damage was modelled with a 10% reduction in power on one propeller.

All experiments had a randomly initialised feedforward filter and a gradient filter where the weights were initialised as a matrix of zeros.

Hardware

The drone used to demonstrate IAFF altitude control and adapting to damage was a Parrot Mambo minidrone. It has a Linux operating system on an 800 MHz Arm A9 microprocessor. All the state estimation, optic flow calculations (Lucas-Kanade), control loops, and IAFF mechanisms were calculated on board in real time. The controller loop frequency was 200 Hz, the same as the simulation. It has a 0.3 MP downward-facing camera that operates at 60 fps. It has a low-energy Bluetooth module (BLE) which connected to an external computer. The control code was uploaded to the drone over the BLE connection and the estimated state variables by the drone were transmitted at 10 Hz to the computer.

The baseline flight control system was the same on the hardware as in the simulation (Sec. 4.5.1). All IAFF demonstrations on the drone used 9 discrete first-order high-pass filters to lift the reference, as this yielded faster learning in the prior simulations.

For adapting to propeller damage, the control architecture on the drone and in the simulations were identical. However, instead of a 10% reduction in power on one propeller, the tip of a propeller was cut off to achieve the same effect.

The altitude control demonstration used the altitude reference lifted by the 9 high-pass filters as input. The errors used for IAFF were the proportional, and derivative errors. The derivative error was not always an exact derivative of the proportional error because they were obtained from the state estimator. Using the output of the feedback controller as the error worked in that it tended towards zero. But the reference would not equal the actual state. It was manipulating the different sensor signals such that the derivative and proportional errors cancel out.

4.5.2 Drone results

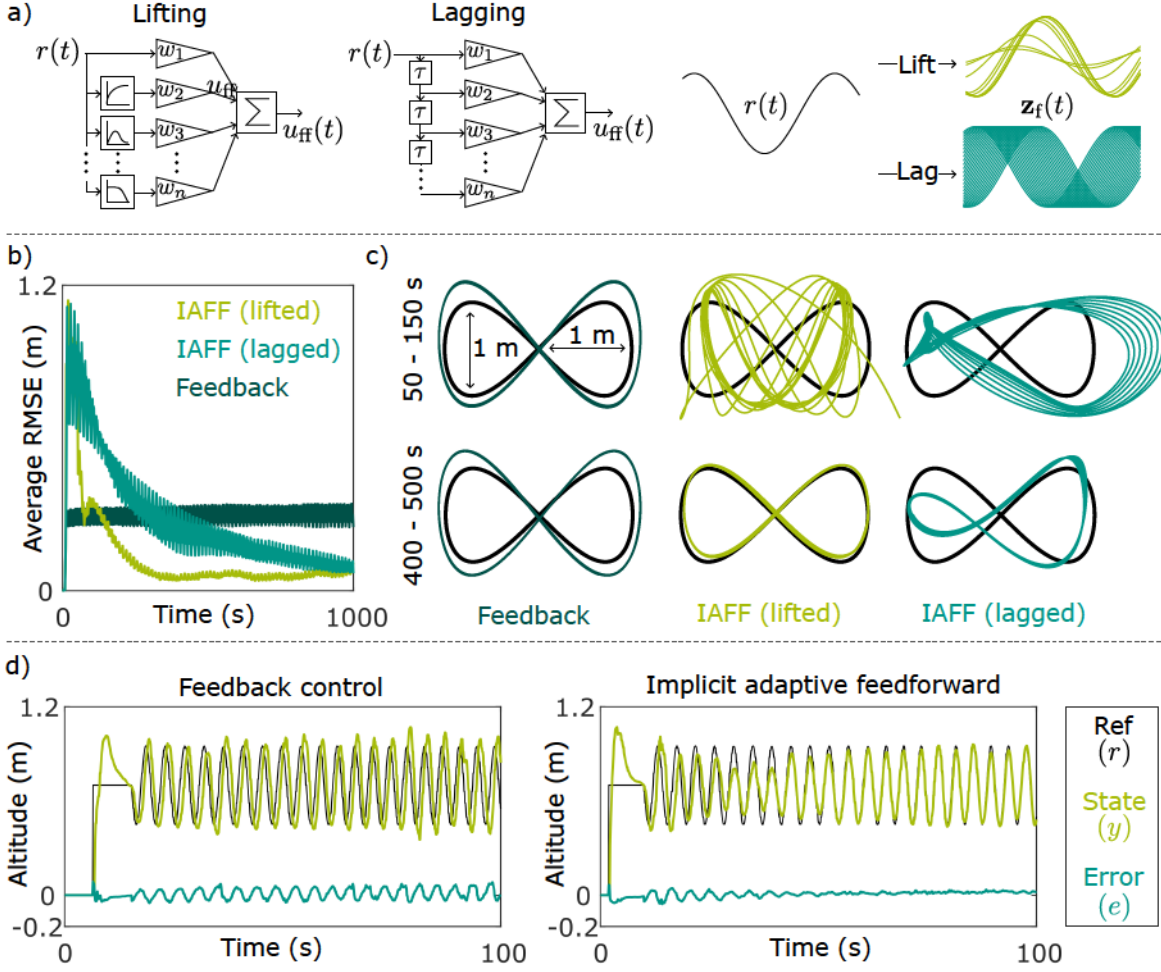


Figure 4.6: Implementation of IAFF on Parrot Mambo minidrone compared with feedback controller. (a) Lifting and lagging mechanisms that generate the input values to the feedforward filter. (b) Window-average root-mean-square-error (RMSE) over time for drone simulation flying in a figure-of-8. (c) Trajectories for the drone simulation flying in a figure-of-8 at two different time windows. (d) Comparison of feedback and IAFF altitude control on a Parrot Mambo minidrone hardware.

Pitch and roll control

When the reference position of the drone was a 2×1 m figure of eight, the standard feedback controller on the simulated drone had a maximum error of 0.46 m, and a time-invariant root mean square error (RMSE) centred around 0.29 m (Fig. 4.6b).

From the period of 50 to 150 s the average RMSE was 0.69 and 0.35 m for the lagged and lifted IAFF methods, respectively. For the lagged IAFF this error is significantly higher than the feedback controller alone. However, for the lifted IAFF the error is 0.06 m higher despite

looking much less like a figure of eight (Fig. 4.6c) than the feedback controllers. This is due to the time component: whilst the shape of the feedback controllers figure of eight looks close to the reference, it is consistently lagging behind the reference in time.

After more time to adapt, in the 400 to 500 s time interval, the average errors drop to 0.22 and 0.05 m for the lagged and lifted IAFF controllers, respectively. The lagged IAFF controller then continues to improve (Fig. 4.6b) and has an average error of 0.10 m in the final 100 s of the simulation.

The error does not reduce further for the two adaptive methods when given more time, probably due to the limited latent space of the input to the feedforward filter. Providing more inputs and increasing the number of weights generally resulted in a lower steady-state error, although the time to adapt became longer.

Altitude control

Implicit adaptive feedforward (IAFF) control of thrust/altitude was tested on the Parrot Mambo drone hardware and is compared to the feedback controller in Figure 4.6d. In both cases, all control computation was done on-board the drone and had a preprogrammed altitude reference.

The feedback controller (Fig. 4.6d) has a phase lag between the reference and the actual state, leading to a consistent sinusoidal error. For the IAFF controller, the phase lag is initially the same due to the random initialisation of the feedforward filter. However, in the first 30 s the state sinusoid reduces the magnitude and eliminates the phase delay. In the subsequent 10 s the magnitude of the oscillations increased to match the reference. This was followed by 50 s of low error and good reference tracking performance.

Handling a damaged propeller

The simulated drone flying sinusoidally up and down with one propeller operating at 90% power had a consistent oscillating pitch and roll error (Fig. 4.7a). Initially, the oscillations were large, before reaching a stable limit cycle after ≈ 30 s. This corresponded to an oscillating position and an induced drift in the Y axis (Fig. 4.7a).

The same simulation, using implicit adaptive feedforward control (IAFF), showed larger

initial pitch and roll oscillations (Fig. 4.7b). Again, this was because the feedforward filter weights were randomly initialised. After ≈ 30 s the oscillations were smaller than those under feedback control alone (Fig. 4.7a). Despite a single weakened propeller, after 40 s the oscillations in pitch and roll were negligible as the drone flew up and down. The position error of the drone (4.7b) does not have the same oscillations or drift as that of the feedback controller after the initial learning period.

When the same feedback controller was implemented in hardware and the same test was performed (Fig. 4.7c) the position oscillations were similar to the simulation. The drift observed in the Y-axis of the simulated drone, was present in both axes on the real drone, and was larger. Additionally, the transient fluctuations in position error were less predictable in amplitude.

This was the same for the drone under IAFF control with the damaged propeller (Fig. 4.7d). Unlike the simulations, the IAFF scheme did not mitigate the drift on the hardware platform, such that the feedback controller and IAFF had comparable drift in position. As has consistently been the case, the initial error was larger for the IAFF than for the feedback controller due to the random initialization of the feedforward weights. The error then quickly reduced. After ≈ 30 s there was a calibration period in which the error increased. After this, the transient effect of the sinusoidally varying altitude of the drone under IAFF control was reduced. The percentage of the frequency spectrum that the altitude reference induced in the position error is 88.7% lower for the IAFF than the feedback controller in the last 20 s.

The discrepancy in performance between the simulated and hardware drone could be due to the imbalanced propeller in the hardware inducing vibrations; something that the simulation does not consider. In addition, Figure 4.7d highlights that the implicit adaptive controller can only decrease the error when the input to the IAFF filter has the appropriate frequency content. The IAFF filter reduced the error frequencies present in the input by 88.7%. However, Figure 4.7d shows that there are still residual oscillations present at different frequencies. These are not removed because the feedforward filter is a linear combination of the inputs and therefore can not apply actuation commands that are a different frequency to those in the input. For hardware damaged propeller tests, neither the feedback nor the IAFF controller led to a stable limit cycle. This highlights a second limitation, IAFF cannot stabilise a system that cannot be

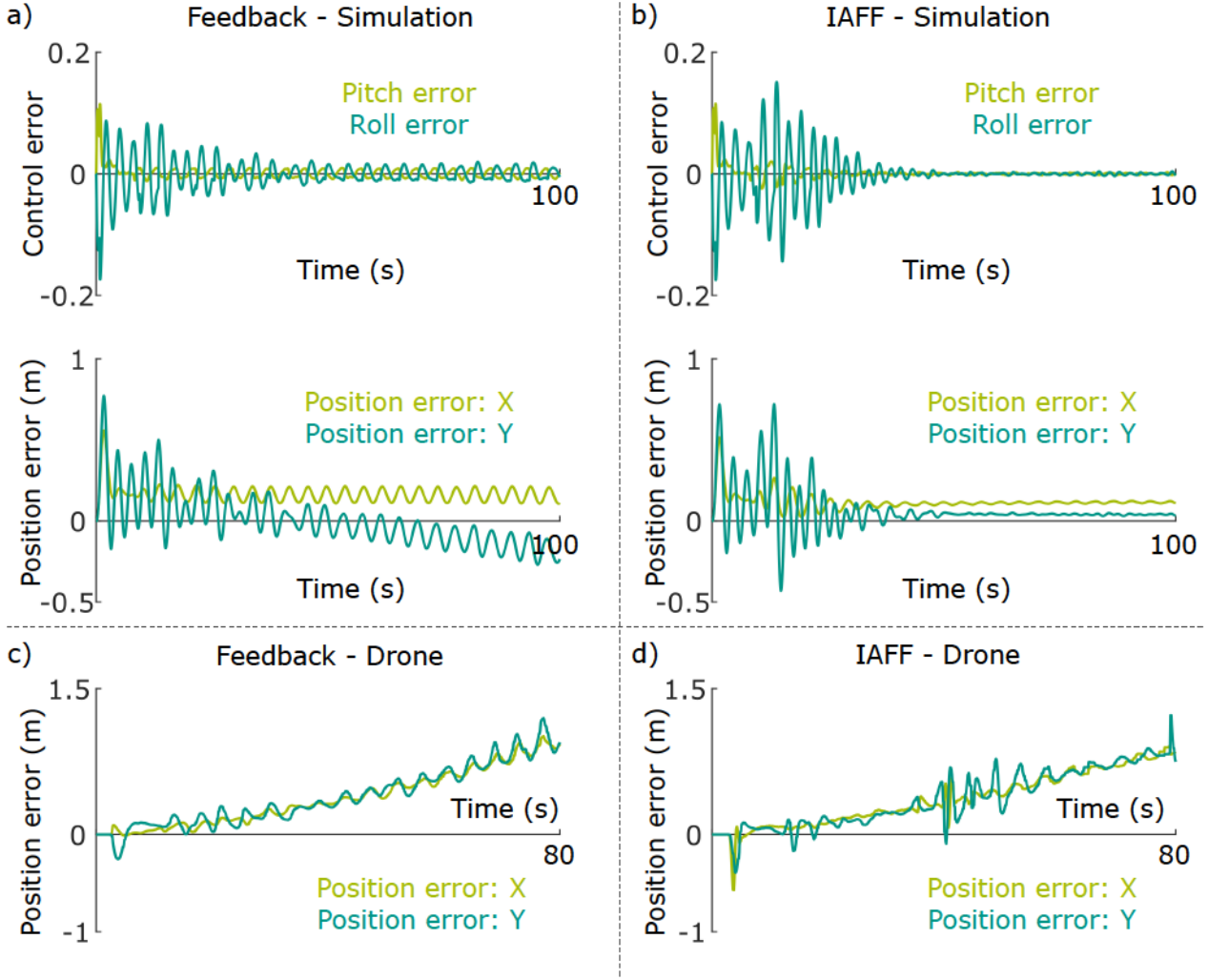


Figure 4.7: How IAFF control copes with a damaged propeller in simulation and on hardware when trying to fly up and down sinusoidally. (a) Pitch, roll, and position error of simulated Parrot Mambo minidrone in feedback control. The damaged propeller was modelled with a 10% reduction in power on one of the propellers. (b) Simulation using IAFF control. (c) Position error with a feedback controller on the Parrot Mambo minidrone hardware with one propeller cut to reduce the swept area by $\approx 10\%$ whilst sinusoidally flying up and down. (d) Position error of the same hardware and test, but controlled using IAFF.

stabilised by a feedback controller (see Sec. 2.3.4, Ch. 2).

4.6 Discussion

In this chapter, I described an implicit adaptive feedforward (IAFF) control scheme. The IAFF scheme uses two filters, it learns the sensitivity derivatives in the *gradient filter*, and uses them to adapt the *feedforward filter* that generates the actuation command. Since few dynamics lend themselves to perfect reference tracking (without infinite actuation), and the

IAFF scheme adapts to reduce the feedback error to zero, the adaptive method was combined with the fully-separable-degrees-of-freedom (FSDoF) controller. This uses a reference filter, or forward model, to ensure that the feedforward component will be realisable, and not demand infinite actuation.

The feasibility of the architecture was demonstrated in simulations and on a Parrot Mambo minidrone. The results show that despite random initialisation, the adaptive controller can learn to improve the reference tracking significantly. Whilst a feedback controller is still required, the improvements in reference tracking with the IAFF controller allows the feedback component to be optimised for robustness as in all two-degree-of-freedom controllers (Araki and Taguchi 2003).

Although it is unlikely that the IAFF control scheme adapts faster than model reference adaptive control (MRAC) (Nguyen 2018), in particular when optimised for fast adaptation (Nguyen, Krishnakumar, and Boskovic 2008), the IAFF scheme has two advantages over MRAC. Firstly, it does not require prior knowledge of the system dynamics which reduces design time. Secondly, it is more flexible for systems that may face significant changes in dynamics (Abdelghani, Lillicrap, and Tweed 2008). This increased flexibility is due to not using fixed sensitivity derivatives and instead learning them online. As a result, IAFF has different applications to MRAC and the other traditional adaptive feedforward methods. In cases where the dynamics are approximately known, detailed dynamic models already exist, and/or the dynamics are unlikely to change drastically, MRAC is a more suitable adaptive controller due to faster adaptation times. However, cases like delivery drones, soft robotics, or any form of robotics that require learning new sensorimotor coordination skills, IAFF and future improved versions of IAFF may be more suitable.

Improvements to the IAFF scheme could involve using a more advanced feedforward filter, for example, radial basis functions, instead of a simple weighted input (Lavretsky and Wise 2012). Or indeed, any type of feedforward neural network (FNN) may enable the encoding of non-linearities, provide better control precision, and better generalizability across a range of inputs (Lavretsky and Wise 2012; Li et al. 2013). I do not consider the generalisability in this account. It is likely that when a wide range of reference signals is applied, some form of more

advanced feedforward filter, like an FNN, would further improve performance across a wider range of reference signals.

The focus of this chapter was using the IAFF scheme to improve the reference tracking. Another application for feedforward control is disturbance rejection (Zhong, Pao, and Callafon 2012; Messineo and Serrani 2009) using an observer or auxiliary sensors (as described in Sec. 1.2.2). For example, an oven uses feedback control to regulate its internal temperature based on a temperature sensor. A feedforward control mechanism could use an auxiliary door sensor to anticipate temperature drops when the door is opened. The IAFF scheme requires an input and an error, and learns to change the output to reduce the error, regardless of the dynamics. This means that sensors that can not be used in feedback control, like an oven door sensor, could be used in IAFF. However, the limitation of using auxiliary sensors as input to an IAFF filter is that the sensor must have low levels of noise. Noise at the input to the feedforward filter can propagate to the output. When the derivative of the output (Fig. 4.1b) is calculated this noise will be magnified, thus limiting the ability to adapt.

macro:

macro:

The content of this chapter demonstrates significant potential for engineered systems, but how can it inform our understanding of biology? It is likely that this description of IAFF is not implementable in the brain due to the use of weight transport. The other mathematical framework for IAFF control by Abdelghani, Lillicrap, and Tweed (2008) does not use weight transport and is therefore possible to implement in a nervous system. The contribution of this chapter to biology is that it validates the principle of IAFF in a hardware system for the first time. IAFF is more likely to be a principle used by biological systems if it is a principle that works.

Whether or not IAFF principles are used in insects can be determined by whether they can adapt to sign changes in their dynamics as demonstrated by Wolf, Voss, et al. (1992). Undoubtedly, humans can adapt to sign changes in dynamics, and a similar principle may be used in the cerebellum as previously suggested (Abdelghani and Tweed 2010). However, it is clear that in humans and probably other mammals, these strategies are combined with many

others such as the coactivation of muscle pairs (impedance control, Burdet et al. [2001](#)), feedback error learning (Kawato [1990](#)), and more advanced state estimation (Miall and King [2008](#)).

Chapter 5

FSDoF control of MIMO systems with multiple different time delays

5.1 Introduction

As discussed in Chapter 2, the use of efference copies and feedforward control is particularly advantageous when dealing with time-delayed systems. When insects move to intercept prey or mates, forage or navigate, and evade predators, they control a complex neuromuscular and aerodynamic system with multiple inputs and outputs, each with a different time delay. Within the leg of a stick insect alone, there is a different time delay associated with measuring movement compared to load (Gebehart and Büschges 2021). In the visual system of the fruit fly *Drosophila*, delays range from 500 ms when responding to second-order motion of a small-field stimulus (Theobald, Shoemaker, et al. 2010), to 30 ms for the wide-field optomotor response (Theobald, Ringach, and Frye 2010). *Drosophila* then have antennae which respond to airflow on the order of 20 ms (Fuller et al. 2014). The system, sensor, and muscle dynamics that insects control are multi-input-multi-output (MIMO) with multiple different time delays (MDTDs).

Time delays are a common property of many engineered systems (Normey-Rico and Camacho 2007). Industrial processes are often described by first-order-plus-dead-time (FOPDT) systems (Normey-Rico and Camacho 2007). As robotics hardware improves, delays in sensors, actuators, and processing will become more relevant for the design of their control architectures.

Four main methods for controlling MDTD systems show significant performance improvements over classical feedback control and loop-tuning methods: (i) model predictive control (MPC, Fig. 5.1b), which uses an internal model of the transfer matrix to optimise future controller outputs (see Sec. 1.2.3 and Kouvaritakis and Cannon 2016). (ii) Smith predictor (SP) methods (and other dead-time compensators, see García and Albertos 2010), where a model predicts the states allowing the controller to be designed around a delay-free model (see Sec. 1.2.3 and Santos, Flesch, and Normey-Rico 2014). (iii) Decoupling methods, where MIMO MDTD systems are approximated by multiple single-input-single-output (SISO) systems using coordinate transforms (Giraldo et al. 2018). Lastly, (iv) the Padé approximation, which replaces time delays with rational transfer functions that closely approximate the system behaviour, as shown in Figure 5.1c. The controller can then be designed using standard methods for non-delayed systems. The errors from this approximation can limit the performance of the controller.

There has been some work on the inclusion of feedforward control into the Smith predictor architecture. However, these methods have generally revolved around SISO systems (Rodríguez, Normey-Rico, et al. 2016), additionally, the focus has been on using feedforward control to compensate for disturbances (Rodríguez, Guzmán, et al. 2014; Rodríguez, Normey-Rico, et al. 2016).

In this chapter, I extend previous work on the efference copy-based fully-separable degrees of freedom (FSDoF, Fig. 5.1a) control—in Chapter 2—to MIMO systems with multiple different time delays. This provides a framework for analysing biological control methods in the context of time delays and another method for control engineers to use on MIMO MDTD systems. The method is benchmarked against MPCs. MPCs are the industry standard for MDTD dynamics and performed considerably better than the other methods in initial analysis. In Section 5.4 I hypothesise a possible method for adjusting to variable delays using the implicit adaptive feedforward framework from Chapter 4.

5.2 Methods

This section is in two parts, Sec. 5.2.1 outlines the mathematical method for including the delays in the dynamics into the FSDoF controller. In Sec. 5.2.2 I describe the simulations, benchmarking, and specifics of the controller designs used to generate the results.

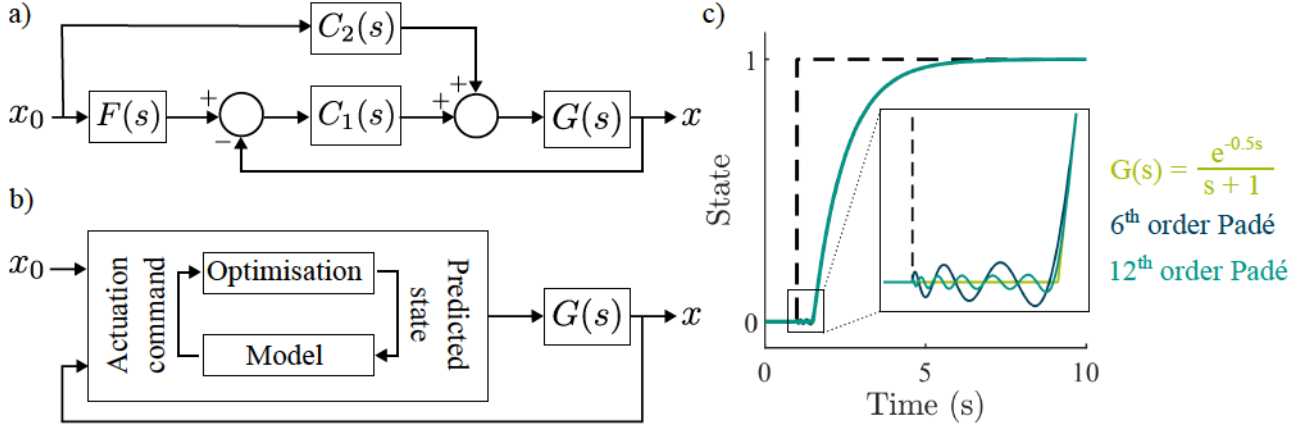


Figure 5.1: (a) Block diagram of FSDoF control scheme. $G(s)$ describes the system dynamics where s is the complex frequency domain variable. $C_1(s)$ and $C_2(s)$ describe the feedback and feedforward controller, respectively. $F(s)$ describes the forward model, x_0 is the desired state, and x is the actual state. The forward model should predict the sensory feedback that results from the desired change in state: $F(s) = GC_2(s)$. (b) Model predictive control (MPC) is used as a benchmark. (c) Step response of a first-order-plus-dead-time system, and the step response of a 6- and 12-order Padé approximation.

Unlike in Chapter 2, the closed-loop transfer function is now a transfer matrix given by Eq.

5.1:

$$\frac{x}{x_0}(s) = (I + GC_1)^{-1}G(C_2 + C_1F) \quad (5.1)$$

where I indicates the identity matrix. The forward model remains the same:

$$F(s) = GC_2(s). \quad (5.2)$$

Substituting Eq. 5.2 into Eq. 5.1 gives the closed-loop reference tracking transfer matrix:

$$\frac{x}{x_0}(s) = (I + GC_1)^{-1}G(C_2 + C_1GC_2) \quad (5.3)$$

$$= (I + GC_1)^{-1}(GC_2 + GC_1GC_2) \quad (5.4)$$

$$= (I + GC_1)^{-1}(I + GC_1)GC_2 \quad (5.5)$$

$$\frac{x}{x_0}(s) = GC_2(s) \quad (5.6)$$

5.2.1 FSDoF on MDTD systems

This method starts with a transfer matrix containing how each input relates to each output, or each actuation command affects each sensor signal. Assuming a square system with n inputs and outputs, there will be an $n \times n$ transfer matrix of the form:

$$G(s) = \begin{bmatrix} L_{1,1}(s)e^{-\tau_{1,1}s} & \dots & L_{1,n}(s)e^{-\tau_{1,n}s} \\ \vdots & \ddots & \vdots \\ L_{n,1}(s)e^{-\tau_{n,1}s} & \dots & L_{n,n}(s)e^{-\tau_{n,n}s} \end{bmatrix} \quad (5.7)$$

where $L_{i,j}(s)$ is the transfer function of the dynamics from actuation input j to sensor output i ; and $\tau_{i,j}$ is the time delay from actuation input j to sensor output i .

The proposed method consists of two key steps: (1) decomposing the transfer matrix delays into input and output delays to obtain a dynamics matrix that is either non-delayed or minimally delayed, and (2) incorporating these delays into the forward model and feedforward controller.

Each transfer delay τ represents a pure time delay in signal propagation from input to output. However, in Eq. 5.7, the origin of these delays is not explicitly clear. I will categorise them as either input delays, output delays, and process delays: input delays affect the actuation command signal, output delays affect the sensor signal, and process delays occur within the system dynamics. Since these delays can often be mathematically rearranged, they do not tell us anything about the cause of these delays. Notably, input delays do not necessarily correspond to actuator delays, nor do output delays necessarily correspond to sensor delays.

In biological sensorimotor control, delays primarily arise from sensory processing and muscle

activation, which naturally align with output and input delays, respectively. In contrast, significant process delays within the dynamics are unlikely in insect locomotion. For example, in insect flight, increasing the force on one wing immediately induces a body rotation, with the true delays residing in the neuronal activation of muscles and the sensory reception of rotation. Process delays are more relevant in industrial systems, such as chemical processes with inherent reaction times. Consequently, the decomposition of transfer delays may not be necessary for biological systems, as sensory and actuator delays already fit the framework of input and output delays, eliminating the need for further transformation.

Decomposing Process Delays into Input and Output Delays

In cases where it is not immediately clear which delays correspond to input or output, the process delays can be transformed, allowing all delays to be considered as input or output delays. To achieve this, the transfer matrix $G(s)$ is decomposed into input delays $T_i(s)$ and output delays $T_o(s)$, along with an undelayed or minimally delayed transfer matrix $\bar{G}(s)$:

$$G(s) = T_o(s)\bar{G}(s)T_i(s). \quad (5.8)$$

The decomposition process follows a straightforward procedure: $T_o(s)$ and $T_i(s)$ are diagonal matrices, where the n^{th} diagonal element of the output delay matrix $T_o(s)$ is the smallest delay in the n^{th} row of $G(s)$. The remaining delays are then assigned to the input delay matrix $T_i(s)$, whose n^{th} diagonal element is the minimum delay in the n^{th} column of $G(s)$. Depending on whether the input delay matrix (Eq. 5.9) or the output delay matrix is decomposed first (Eq. 5.10), different decompositions can be obtained.

$$\begin{bmatrix} L_{1,1}(s)e^{-4s} & L_{1,2}(s)e^{-7s} \\ L_{2,1}(s)e^{-6s} & L_{2,2}(s)e^{-8s} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & e^{-1s} \end{bmatrix} \begin{bmatrix} L_{1,1}(s) & L_{1,2}(s) \\ L_{2,1}(s)e^{-1s} & L_{2,2}(s) \end{bmatrix} \begin{bmatrix} e^{-4s} & 0 \\ 0 & e^{-7s} \end{bmatrix} \quad (5.9)$$

$$\begin{bmatrix} L_{1,1}(s)e^{-4s} & L_{1,2}(s)e^{-7s} \\ L_{2,1}(s)e^{-6s} & L_{2,2}(s)e^{-8s} \end{bmatrix} = \begin{bmatrix} e^{-4s} & 0 \\ 0 & e^{-6s} \end{bmatrix} \begin{bmatrix} L_{1,1}(s) & L_{1,2}(s)e^{-1s} \\ L_{2,1}(s) & L_{2,2}(s) \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & e^{-2s} \end{bmatrix} \quad (5.10)$$

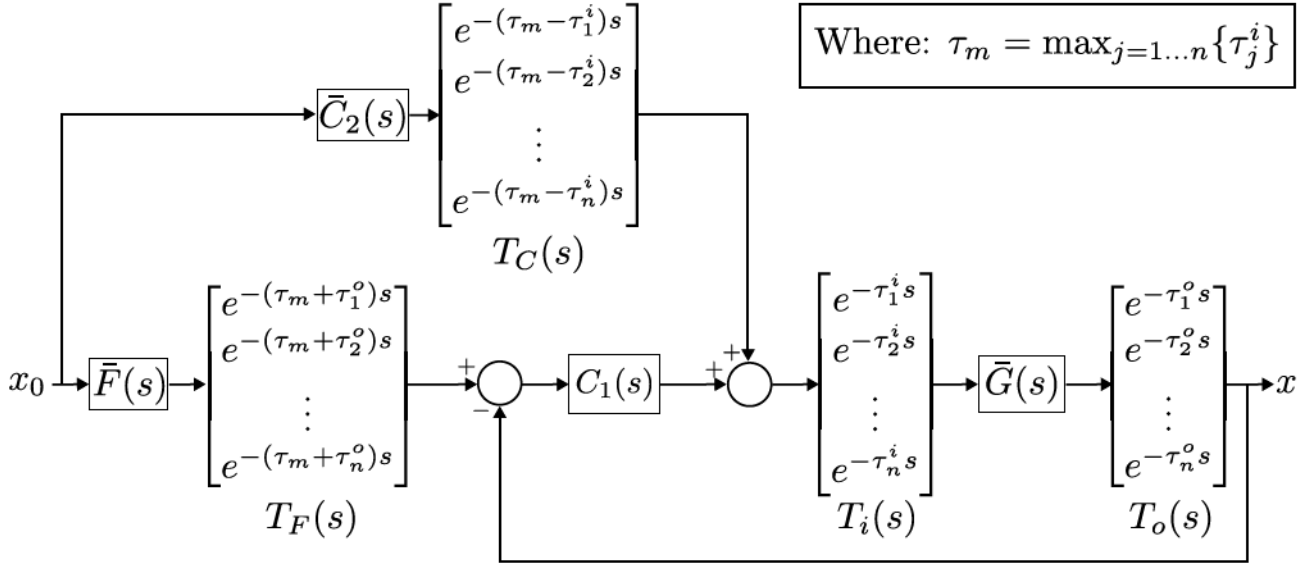


Figure 5.2: Accounting for the system dynamics input/actuator and output/sensor delays, in the FSDoF control scheme. T_F , T_C , T_o , and T_i are depicted as delay vectors in the Laplace domain for compactness, they are diagonal matrices.

If and only if condition 5.11 is met, then there will be no remaining delays in $\bar{G}$, as demonstrated in Eq. 5.12

$$\{(\bar{\tau}^{(in)}, \bar{\tau}^{(out)}) \mid \bar{\tau}^{(in)}, \bar{\tau}^{(out)} \in (\mathbb{R}^+)^n \text{ and } \forall i, j \in \{1, 2, \dots, n\}, \tau_i^{(in)} + \tau_j^{(out)} = \tau_{i,j}\}$$

$$\begin{aligned} \text{Where, } T_o(s) &= \text{diag}(e^{-s\bar{\tau}^{(out)}}) \\ T_i(s) &= \text{diag}(e^{-s\bar{\tau}^{(in)}}) \end{aligned} \quad (5.11)$$

$$\begin{bmatrix} L_{1,1}(s)e^{-4s} & L_{1,2}(s)e^{-6s} \\ L_{2,1}(s)e^{-6s} & L_{2,2}(s)e^{-8s} \end{bmatrix} = \begin{bmatrix} e^{-4s} & 0 \\ 0 & e^{-6s} \end{bmatrix} \begin{bmatrix} L_{1,1}(s) & L_{1,2}(s) \\ L_{2,1}(s) & L_{2,2}(s) \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & e^{-2s} \end{bmatrix} \quad (5.12)$$

Constructing the FSDoF controller using the input and output delays

The input and output delays can be used by the feedforward controller and the forward model to meet the FSDoF requirement as explained in Chapters 2 and 4, by substituting in Eq. 5.8.

$$F(s) = T_o(s)\bar{G}(s)T_iC_2(s) \quad (5.13)$$

To find the appropriate delays for $F(s)$ and $C_2(s)$, they are re-written as delay-free transfer functions followed by delays.

$$F(s) = T_F \bar{F}(s) \quad (5.14)$$

$$C_2(s) = T_C \bar{C}_2(s) \quad (5.15)$$

Inlcuding these substitutions in equation 5.13 gives equation 5.16:

$$T_F \bar{F}(s) = T_o(s) \bar{G}(s) T_i T_C \bar{C}_2(s) \quad (5.16)$$

There are two steps to find T_F and T_C in terms of T_i and T_o so that $\bar{C}_2$ can be determined from $\bar{F}$ and $\bar{G}$. The first step is to choose the feedforward delays such that equation 5.17 holds, putting the combined delays in a commutable form. In the second step, the forward model delays are set to the combination of the output, the input, and the feedforward delays, as described in 5.18.

$$T_i T_C = e^{-\tau_m s} I \quad (5.17)$$

$$T_F = T_o e^{-\tau_m s} \quad (5.18)$$

Once the forward model and feedforward delays satisfy Eqs. 5.17 and 5.18, equation 5.16 can be simplified to:

$$\bar{F} e^{-\tau_m s} = \bar{G} \bar{C}_2 e^{-\tau_m s} \quad (5.19)$$

The FSDoF controller can then be designed based on the minimally delayed transfer matrix. All states will be delayed by τ_m . The minimum τ_m that can be chosen is the maximum delay in T_i . The system with the delays is shown in Figure 5.2.

5.2.2 Simulation and benchmarking methods

Simulations are used to demonstrate the effectiveness of FSDoF control on MIMO systems with multiple different time delays (MDTD), benchmarked against model predictive control (MPC). The section is split into the simulation environment and system dynamics tested, the MPC design used for benchmarking, and the FSDoF controller design.

Simulation environment and system dynamics

The simulations were carried out in MATLAB Simulink 2024b. Continuous transfer matrices described the system dynamics (G), and the FSDoF controller filters. These were approximated using the `ode45` solver with a relative tolerance of 10^{-5} . The run time of the simulation varied depending on the dynamics (as seen in Fig. 5.3), but were chosen to ensure that for every change in the desired state there was sufficient time for both controllers to settle. The reference signal was designed so that both states initially target a value of 1. At one-third of the simulation duration, the second state's reference drops to 0.5, and at two-thirds of the duration, the first state's reference increases to 1.5 (see Fig. 5.3).

Five different MIMO MDTD system dynamics were tested in simulations, shown in Table 5.1. Dynamics 1, 2, and 3 are taken from the literature as common examples for benchmarking MDTD systems, and 4 and 5 are contrived examples for the purposes of demonstrating characteristics of FSDoF control. The justification for choosing each of the dynamics is given in Table 5.1.

Table 5.1: MIMO MDTD system dynamics used to benchmark MPC and FSDoF and the justification for their use

Dynamics	Justification
$\begin{bmatrix} \frac{12.8}{16.7s+1}e^{-s} & \frac{-18.9}{21s+1}e^{-3s} \\ \frac{6.6}{10.9s+1}e^{-7s} & \frac{-19.4}{14.4s+1}e^{-3s} \end{bmatrix}$	Example 1: The Wood and Berry distillation column (Wood and Berry 1973), commonly used as a benchmark for controlling MDTD systems. (Giraldo et al. 2018; Normey-Rico and Camacho 2007).
$\begin{bmatrix} \frac{3.5}{s}e^{-s} & \frac{-1}{2s+1}e^{-5s} \\ \frac{2}{1.5s+1}e^{-7s} & \frac{-1}{3.2s+1}e^{-5s} \end{bmatrix}$	Example 2: A three-stage evaporator process (Normey-Rico and Camacho 2007), discussed in (Santos, Flesch, and Normey-Rico 2014; García and Albertos 2010; Flesch et al. 2011), used as an example of dynamics with an integrator term.
$\begin{bmatrix} \frac{1}{s-1}e^{-0.5s} & \frac{0.5}{s+1}e^{-0.7s} \\ \frac{0.1}{10s+1}e^{-0.3s} & \frac{1}{s}e^{-0.7s} \end{bmatrix}$	Example 3: Used in (García and Albertos 2010) as an example for controlling unstable dynamics.
$\begin{bmatrix} \frac{1}{10s+1}e^{-5s} & \frac{3}{2s+1}e^{-7s} \\ \frac{0.1}{0.1s+1}e^{-4s} & \frac{-4}{5s+1}e^{-6s} \end{bmatrix}$	Example 4: A contrived example used to demonstrate control of systems that satisfy the fully separable delay condition given in equation 5.11.
$\begin{bmatrix} \frac{1}{10s+1}e^{-5s} & \frac{3}{2s+1}e^{-7s} \\ \frac{0.1}{0.1s+1}e^{-4s} & \frac{-4}{(5s+1)s}e^{-6s} \end{bmatrix}$	Example 5: A modification of Example 4 that highlights a limitation of the proposed method, where it is not possible to find a low-pass filter pre-filter that satisfies internal stability.

Model predictive controller benchmark

MPCs (introduced in Sec. 1.2.3) are used to benchmark the FSDoF system, as they are widely used in industry Garcia, Prett, and Morari 1989; Ding et al. 2018; Allgower, Findeisen, Nagy, et al. 2004. Figure 5.1b shows that MPCs use a model to optimise the actuation commands to minimise a cost function. The cost function can combine many objectives like limits, danger zones, actuation, and error. Here, the actuation rate and tracking error are used:

$$J = \sum_{k=0}^{N-1} ((x(k+1) - x_0)^T Q (x(k+1) - x_0) + \Delta u_k^T R \Delta u_k) \quad (5.20)$$

where: N is the number of future steps that the controller is optimising for (the time horizon), Δu_k is the actuation rate vector for step k , Q and R are matrices to weight the importance of the two states and actuation rate signals, respectively. In these simulations $Q = R\beta^{-1} = I$ where β is a scalar that determines how aggressive the MPC is. A large β means that the actuation cost is lower priority. The MPCs all had a time step of 0.2 s to be significantly smaller than the fastest dynamics in any of the systems. The time horizon (N) was set to be greater than ten times the largest delay divided by the time step, as is common practice (MathWorks n.d.). The optimisation in the MPC was run at every time step using updated sensor information.

Time delays were included in the MPC model by using a state-space model with time-shifted inputs, so the overall model approximates the time-delayed system.

FSDoF controllers

Following the method described in section 5.2.1, $G(s)$ can be transformed into a minimally delayed version $\bar{G}(s)$, and the FSDoF controller can be designed without delays around $\bar{G}(s)$ and then have the delays added afterwards. There is no single method for determining $\bar{F}(s)$ and $\bar{C}_2(s)$ that works best for every system. The approach used was to take a 6-order Padé approximation of the remaining delays in $\bar{G}(s)$, if there were remaining delays. $\bar{F}(s)$ was given diagonal third-order low-pass filter entries with bandwidth ω . In cases where $\bar{G}^{-1}(s)$ contained unstable poles (example 5, Tab. 5.1) these poles were included as zeros in $(\bar{F}(s))$. $\bar{C}_2(s)$ was then calculated as $\bar{G}^{-1}\bar{F}$.

The bandwidth ω then acted similarly to β in the MPC, where a larger bandwidth made the FSDoF faster and more aggressive. The bandwidth was tuned to be as fast as possible without introducing too much oscillatory error caused by the Padé approximation. For $G_4(s)$ the delays perfectly cancelled and no approximation was needed. It is worth noting that using a higher order Padé approximation would enable an increase of the controller bandwidth without inducing unwanted oscillations.

5.3 Results

Figure 5.3 shows the step responses for all the dynamics tested. For $G_4(s)$ where the delays were fully decomposed, Fig. 5.3e shows how the FSDoF controller can be arbitrarily fast.

Figure 5.3f shows the step response for $G_5(s)$ where the forward model included a zero and could not consist only of low-pass filters. The result is that the step response does not show smooth low-pass filter characteristics as it does for the other dynamics, but includes a deviation similar to that of the MPC. In all other examples in Fig. 5.3 the FSDoF has smooth low-pass filter properties.

When the cost function (Eq. 5.20) is applied to both controllers and the costs are averaged across the five dynamics, the FSDoF has a lower total cost than the MPC (Fig. 5.4a). Although the tracking error cost is marginally higher for the FSDoF than the MPC (Fig. 5.4a), the settle time (within 2% of the reference) was considerably lower for the FSDoF (Fig. 5.4c). Even when β was adjusted for a more aggressive MPC, it was never able to reach the shorter settle-time of the FSDoF (Fig. 5.4b). The more aggressive MPC with $\beta = 10$ had a lower state tracking error: 16 for MPC ($\beta = 1$), 14 for MPC ($\beta = 10$), and 21 for the FSDoF. However, for simulations with $\beta = 10$, the MPC had an average actuation rate cost over 100 times larger than the FSDoF.

In addition, the diagonal forward model structure of the FSDoF led to a lower cross-coupling error compared to the MPC. In the MPC, when one state reference signal changed, the other state deviated from the constant reference signal (highlighted in orange in Fig. 5.4d). The integral of the mean-squared cross-coupling error averaged across the simulations is shown in Figure 5.4e.

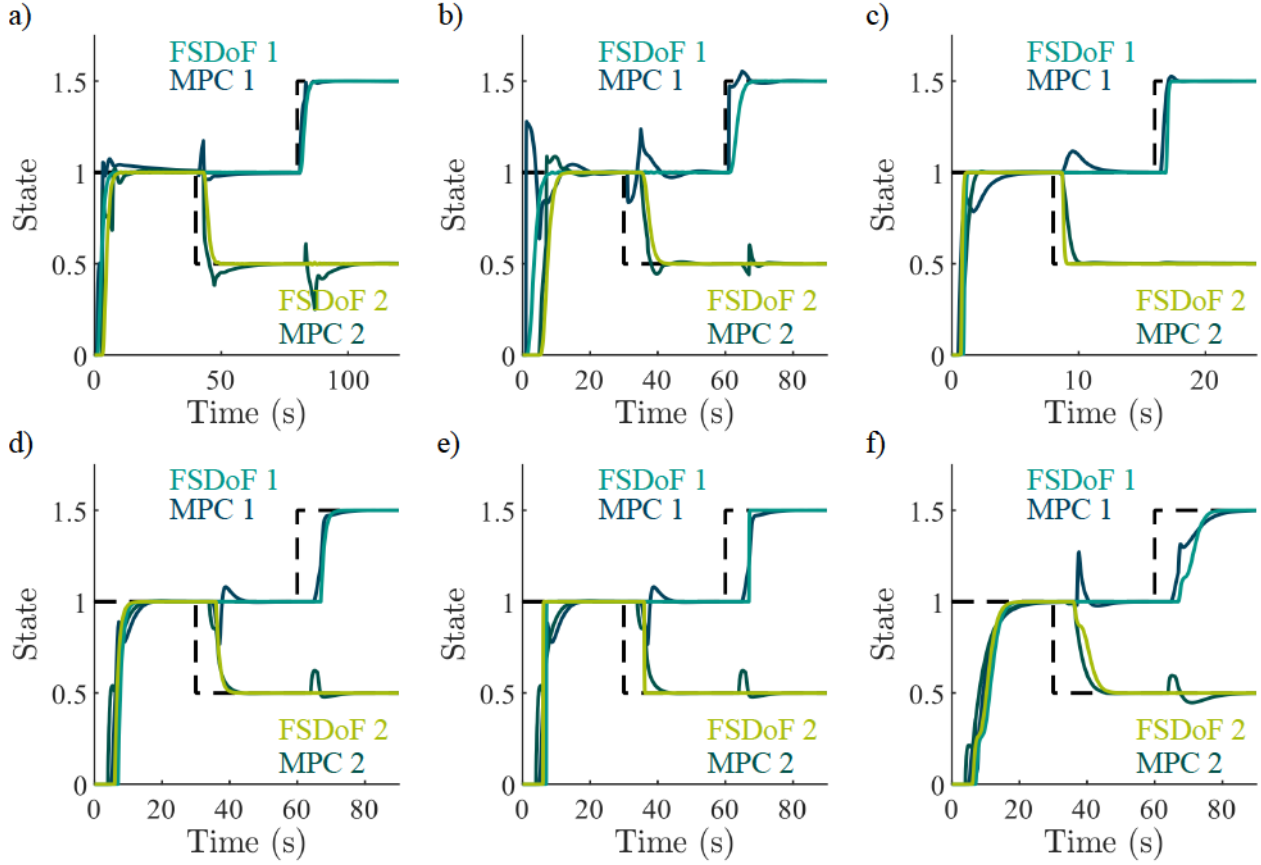


Figure 5.3: Step responses of unique MPCs and FSDoF controllers on the different dynamics in Table 5.1. The reference, or desired state, is given by the black dashed line. At time 0 s, both desired states of the MIMO systems are set to one. A third of the way through the simulations the signal for desired state 2 drops to 0.5, and at two thirds time the desired state 1 signal increases to 1.5. (a, b, c) State responses under the two control schemes for dynamics 1, 2, and 3, respectively. (d, e) State responses for dynamics 4 where delays are fully decomposed in the FSDoF controller: (d) uses a realistic FSDoF with similar actuation capabilities as MPC, while (e) demonstrates that the FSDoF controller can operate at an arbitrary speed. (f) State response for dynamics 5.

Table 5.2 shows the benchmarking results for the different dynamics. Although the average cost for the MPC is higher than for the FSDoF, in dynamics 3, 4, and 5 the MPC has a lower total cost. For all dynamics, the tracking error cost is lower for the MPC, despite the FSDoF having lower settle times. This is because a lower squared error value is obtained by rising quickly and overshooting, as opposed to slower rising without overshoot. The poor performance of the MPC on G_2 has a large impact on the average results. For G_3 and G_4 the lower cost of the MPC can be largely attributed to the actuation rate cost. Reducing the bandwidth (ω) of the FSDoF reduces this actuation cost, making the performance more comparable.

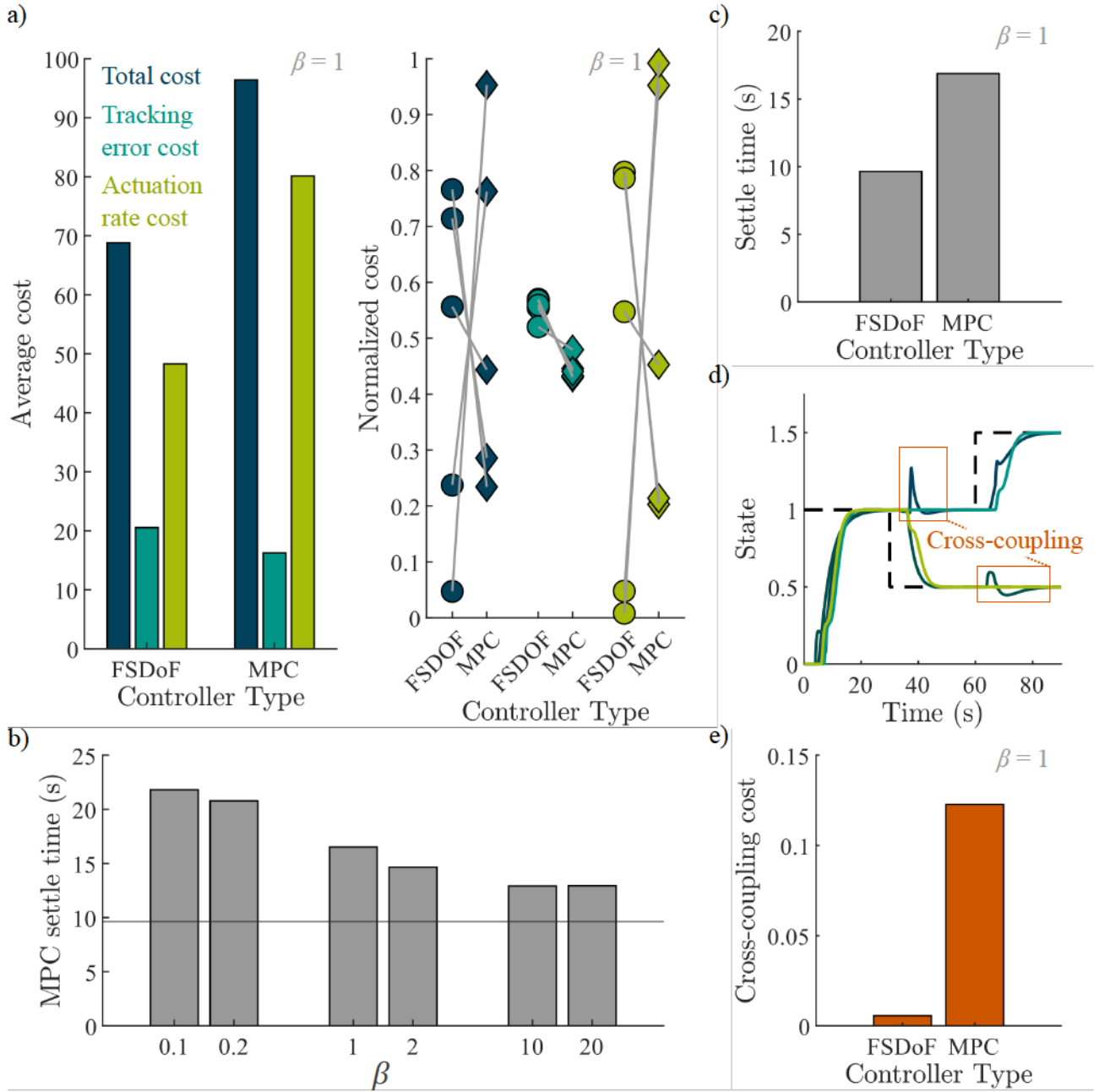


Figure 5.4: (a) Breakdown of costs as defined by the MPC cost function for both the FSDoF and MPC. The left shows the absolute costs averaged across the five different dynamic systems, and the right is the normalised costs for each system. (b) Settle time of the MPC over varying β in the cost function of MPC. A lower β optimises to reduce the actuation cost at the expense of reference tracking error. A higher β makes the controller more aggressive. (c) Comparison of settle time for FSDoF and MPC. (d) Simulation on dynamics number 5 with the cross coupling highlighted by orange rectangles. (e) The squared error of the tracking costs due to cross-coupling.

Table 5.2: Different costs for the simulations in Fig. 5.3 ($\beta = 1$).

$G(s)$	Total cost		Tracking cost		Actuation cost		Cross-coupl. cost		Settle time	
	MPC	FSDoF	MPC	FSDoF	MPC	FSDoF	MPC	FSDoF	MPC	FSDoF
1	36.29	11.30	7.94	9.88	28.35	1.42	0.15	≈ 0.00	25.64	8.00
2	313.79	15.53	9.88	13.08	303.91	2.45	0.13	≈ 0.00	18.10	11.08
3	43.98	143.97	10.16	11.03	33.81	132.94	0.02	0.01	4.40	1.22
4	50.53	126.41	25.08	32.87	25.45	93.54	0.16	≈ 0.00	14.73	10.93
5	37.40	46.92	28.22	35.82	9.19	11.10	0.15	0.02	21.52	17.00

5.4 Implicit adaptive feedforward control on multiple different time delay systems

As stated above, in insects, input and output delays could correspond directly to actuation and sensory delays, respectively. As seen in Figure 5.2, the delays in the feedforward controller $C_2(s)$ depend only on the input/actuation delays. In insects, these delays arise from the time it takes for neural signals to reach and contract the muscle and are therefore unlikely to change. The sensory delays, in particular visual sensor delays, vary considerably depending on the luminance levels (Weckström 1989; Miall 1978; Laughlin 2010).

Assuming fixed feedforward delays, and variable forward model delays, how can an implicit adaptive feedforward controller (IAFF, see Ch. 4) help adapt the variable forward model delays? Using a lagged desired state vector as input to the IAFF controller (as opposed to lifted, see Fig. 4.6a), as shown in Equation 5.21 below.

$$u_{\text{ff},i} = \langle \mathbf{W}_i, \mathbf{X}_0 \rangle \quad (5.21)$$

$$\underbrace{\underbrace{\begin{bmatrix} w_{1,1} & w_{1,2} & \dots & w_{1,L} \\ w_{2,1} & w_{2,2} & \dots & w_{2,L} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n,1} & w_{n,2} & \dots & w_{n,L} \end{bmatrix}}_{\mathbf{W}_i \ (n \times L)}}_{\underbrace{\underbrace{\begin{bmatrix} x_{0,1}(t) & x_{0,1}(t - \Delta t) & \dots & x_{0,1}(t - (L-1)\Delta t) \\ x_{0,2}(t) & x_{0,2}(t - \Delta t) & \dots & x_{0,2}(t - (L-1)\Delta t) \\ \vdots & \vdots & \ddots & \vdots \\ x_{0,n}(t) & x_{0,n}(t - \Delta t) & \dots & x_{0,n}(t - (L-1)\Delta t) \end{bmatrix}}_{\mathbf{X}_0 \ (n \times L)}}_{\substack{u_{\text{ff},i} \\ (1 \times 1)}} = \left\langle \begin{bmatrix} w_{1,1} & w_{1,2} & \dots & w_{1,L} \\ w_{2,1} & w_{2,2} & \dots & w_{2,L} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n,1} & w_{n,2} & \dots & w_{n,L} \end{bmatrix}, \begin{bmatrix} x_{0,1}(t) & x_{0,1}(t - \Delta t) & \dots & x_{0,1}(t - (L-1)\Delta t) \\ x_{0,2}(t) & x_{0,2}(t - \Delta t) & \dots & x_{0,2}(t - (L-1)\Delta t) \\ \vdots & \vdots & \ddots & \vdots \\ x_{0,n}(t) & x_{0,n}(t - \Delta t) & \dots & x_{0,n}(t - (L-1)\Delta t) \end{bmatrix} \right\rangle$$

where $\mathbf{X}_0$ is the matrix input into the IAFF controller, consisting of the lagged desired state vector, $u_{\text{ff},i}$ is the i th feedforward command. $\mathbf{W}_i$ is the weight matrix for the i th feedforward command in the IAFF controller, and $\langle \mathbf{W}_i, \mathbf{X}_0 \rangle$ denotes the inner product between the two matrices.

Assuming the baseline sensor delays are known and are considered in the forward model (as in Fig. 5.2), the forward model delays can be artificially increased by Δt . Upon perfect adaption of the IAFF controller, the first column of weights corresponding to the most recent desired state vector should be zero, provided the sensor delays are at the baseline. If this is initially the case, an algorithm could monitor whether that first column of zero weights became non-zero, meaning the sensory delays have increased. In such case, the forward model could increase the delay. Alternatively, if the second column of weights were adapted to zero, this would indicate the sensory delays had been reduced, the forward model could then reduce the delays. Knowing which delay in the forward model to increase or decrease is only possible if the forward model is diagonal. The proposed method of adapting the FSDoF controller to variable sensory delays would not work for $G(s)$ where there is no realisable $C_2(s)$ that leads to a diagonal $F(s)$.

If the forward model is diagonal and in all $\mathbf{W}_i$ matrices for $i = 1, \dots, n$ the element $w_{j,1}$ is non-zero, it would be the j th delay in the forward model that should be increased. One limitation of this method is that it slows down the controller by Δt .

5.5 Discussion

This Chapter introduced a method for designing efferece copy (FSDoF) controllers for multi-input-multi-output (MIMO) systems with multiple different time delays (MDTDs). FSDoF control was then benchmarked against model predictive control (MPC) on such MIMO MDTD systems. Although efferece copy mechanisms have been identified in the auditory system of crickets (Poulet and Hedwig 2003; Poulet and Hedwig 2006; Webb 2004b) and the electrosensory system of electric fish (Bell 1989), their role in sensorimotor control has so far only been physiologically demonstrated in the visual systems of flies (Kim, Fitzgerald, and Maimon 2015; Kim, Fenk, et al. 2017) and ants (Dauzere-Peres and Wystrach 2024). Understanding

how efference copy-based architectures manage multiple different delays remains important for several reasons. First, different visual processing pathways are known to have different latencies (Theobald, Ringach, and Frye 2010; Theobald, Shoemaker, et al. 2010). Second, it remains plausible that efference copies are also used in other sensory modalities, such as mechanosensation. Third, there are additional time delays in muscle activation, and these delays can vary between motor units. Thus, even in the absence of widespread evidence for efference copies in sensorimotor control beyond visual systems, insects must contend with multiple different time-delay systems. This work provides a framework for addressing these challenges, and a practically implementable control strategy.

The analysis also reveals that FSDoF control is particularly advantageous when delays are due to sensory and actuator latencies—for instance during insect locomotion. In these cases, depending on the system dynamics, FSDoF can often facilitate arbitrarily fast state changes.

A key limitation of this study is that it only compares square systems, where the number of sensory signals matches the number of actuator commands. Future work could explore how FSDoF can be extended to non-square systems, as biological systems typically feature more sensory inputs than motor outputs. This is not an inherent limitation to FSDoF control, it merely complicates the design of FSDoF controllers due to the challenge of inverting a non-square transfer matrix. Additionally, the benchmarking in this Chapter does not use identified insect flight dynamics. This is partly because there is no widely accepted transfer matrix formulation of insect flight dynamics, which includes different delays and sensory modalities. Furthermore, while FSDoF is tested across a range of system dynamics, the slower dynamics in this analysis are somewhat arbitrary. Since the system equations are time-scalable, shifting the time units from seconds to milliseconds would not alter the underlying behavior, meaning the absolute timescales should not be overinterpreted.

Furthermore, I hypothesised how the FSDoF architecture could be adapted if the sensory delays were to change, but this method has not yet been validated. It is possible that due to auto-correlation in the desired state signal, even if the sensory delays had not increased, the first column of the weight matrix would be adapted to be non-zero. However, it is shown that if the sensory delays change, then delays in the forward model can be adjusted to correct for

that. Thus, it presents an interesting hypothesis for biology, do insects adapt their controller to different sensory time delays, and does that adaptation take place in the forward model?

Finally, while MPC explicitly optimises a cost function, FSDoF can achieve a lower total cost in certain cases. This counterintuitive result arises due to discretization errors in MPC, which relies on a discrete state-space model for optimisation. Furthermore, MDTD systems often lead to the optimisation not being strictly convex, as decision variables appear at multiple, non-consecutive time steps (Chen, Li, et al. [2022](#)).

Chapter 6

Conclusion

6.1 Summary of contributions

Chapter 2 represents the first formalisation of the efference copy and inner-outer loop hypothesis within a control engineering framework. Prior to this work, no phenomenological control model had been proposed to describe insect efference copy-based architectures, despite their widespread presence in biological systems (Crapse and Sommer 2008). By recasting these biological principles in control engineering terms, Chapters 2 and 5 provided several insights. First, they suggest a functional explanation for the evolution of efference copy-based control in insects by identifying advantages compared to other control architectures. The efference-copy based architecture effectively handles systems with time delays, multiple different time delays, and sensor noise. It maintains sensitivity to external disturbances by avoiding sensor saturation through the use of an efference copy. When equipped with high-pass filter sensors, its performance robustness is improved when faced with variable gain and bandwidth sensors. These advantages are particularly useful for insects, that: (1) have significant sensor delays compared to their dynamics. System identification of the *Drosophila* closed-loop speed controller showed that the propagation delay was just under half of the smallest time constant in the dynamics (Fry, Rohrseitz, et al. 2009). (2) Have variable gain and bandwidth visual sensors that are highly dependent on the luminance, contrast, and spatial frequencies in the environment (Taylor and Krapp 2007). Lastly, (3) use sensors with significant noise (Tarawneh et al. 2017;

Faisal, Selen, and Wolpert 2008).

In addition to suggesting functional explanations for the efference copy-based control architecture, Chapters 2 and 5 established a framework that offered insights for designing focused biological experiments. Third, it allows explicit simulation and analysis of how different sensor errors and dynamics changes affect performance, enabling hypothesis generation about biological control strategies.

One example of a hypothesis that followed the phenomenological controller modelling is that adaptive feedforward control may aid the sensorimotor control performance after changes in system dynamics. This hypothesis—tested in Chapter 3—offers a novel perspective not previously studied in insects, although it is reported in the cerebellum literature (Wolpert and Kawato 1998; Kawato 1990). This is in part because earlier studies focused primarily on the forward model and error cancellation pathways as opposed to the origin of the motor command (Kim, Fitzgerald, and Maimon 2015; Crapse and Sommer 2008). Although the behavioural results cannot prove the existence of adaptive feedforward control, they are consistent with its predictions and open a new direction for interpreting sensorimotor adaptation.

In addition to supporting the adaptive feedforward control hypothesis, the free-flight kinematic experiments presented in Chapter 3 provided additional insights. One particularly noteworthy finding concerns the functional role of the antennae. Fuller et al. (2014) proposed that the antennae act as high-pass filters, primarily detecting changes in airspeed. In contrast, earlier studies (Burkhardt and Gewecke 1965; Niehaus 1981) suggest a role for antennal input in steady-state flight speed regulation. The results in Chapter 3 are more consistent with the latter interpretation, indicating that antennal feedback contributes to tonic speed control, both in the presence and absence of visual input. The second additional insight is that flight speed increases in *Drosophila melanogaster* with unilateral wing damage when flying in darkness compared to intact controls. This is inconsistent with previous work in *Drosophila hydei*, where Muijres et al. (2017) observed a 27% decrease in flight speed after similar wing damage.

The adaptive feedforward hypothesis also enables a reinterpretation of prior experimental findings. For instance, Wolf, Voss, et al. (1992) demonstrated that insects can adjust their outer-loop control in response to dynamic inversions, yet the underlying mechanism within a

control-theoretic framework remained unclear. An adaptive feedback controller is unlikely in this context, as inverting the plant dynamics in a feedback loop typically induces rapid instability. If, instead, the outer-loop adaptation occurred in a feedforward controller, a key question arises: how can such a controller adapt appropriately to an inversion of the system dynamics? Fixed sensitivity derivatives would result in maladaptive changes, driving the system further from the desired state. This motivated the hypothesis that the feedforward controller may rely on adaptive sensitivity gradients.

In Chapter 4, I describe the first robotic implementation of an implicit adaptive feedforward (IAFF) mechanism which uses adaptive sensitivity derivatives. In addition, I derive bounds on the learning rates that determine the stability of the adaptation. Although neural mechanisms for a similar concept have been described in the cerebellum literature (Abdelghani and Tweed 2010), Chapter 4 is the first mathematical description for practical applications and the first time the concept has been demonstrated in hardware. The hardware demonstration both shows how the method can improve the control of robotic systems and validates the hypothesised concept of implicit sensorimotor learning in humans.

6.2 Limitations and open questions

Cancellation signal or all-or-none efference copies

In the FSDoF architecture used throughout this thesis to model efference copy-based control, I assume that the efference copy operates as a continuous signal. An alternative is an all-or-none switch, where inner-loop sensory feedback is effectively suppressed during a movement. These two mechanisms can be distinguished by determining whether the sensor detects disturbances during active movements, or by observing whether there is an inhibition in the sensor signal during goal-directed movements in sensory deprivation.

Efference copies likely serve as cancellation signals for at least some biological control tasks. For example, in speed control, behavioural evidence presented in Chapter 3 supports the hypothesis that speed is regulated using a combination of feedforward control and efference copies. In addition, as demonstrated in wind tunnel experiments where moving wall gratings

influence flight speed (Fuller et al. 2014; Fry, Rohrseitz, et al. 2009), it is known that vision is also used in feedback to correct for deviations from the desired speed. This suggests that, at least for speed control, the efference copy functions as a graded cancellation signal rather than a complete sensory suppression mechanism. In addition, for voluntary yaw turns in sensory deprivation, an inhibition was found in the HSN cell (Kim, Fitzgerald, and Maimon 2015), suggesting a cancellation signal efference copy. For other goal-directed behaviours such as target pursuit and aligning to polarised light, it is unclear whether sensory input remains active for disturbance detection. In contrast, Canelo et al. (2023) have presented evidence suggesting that when *Drosophila* engages in bar fixation, they employ an all-or-none efference copy, ignoring wide-field background motion during alignment.

The all-or-none mechanism offers some advantages over graded efference copy signals. It eliminates the need for a forward model, reducing energy expenditure and neural complexity. Additionally, it remains robust to changes in sensor properties, does not require adaptation, and fully cancels sensory input regardless of context. It has also been successfully implemented in a robot built to model cricket behaviour (Webb and Harrison 2000). It may also be the only mechanism capable of supporting adaptation to inverted dynamics. In such cases, an unsuppressed feedback controller would introduce instability, even if the feedforward controller could adapt appropriately. To date, bar fixation is the only documented instance in which insects have successfully adapted to an inversion of the dynamics (Wolf, Voss, et al. 1992), and notably, it is also the only confirmed example of an all-or-none efference copy mechanism being employed (Canelo et al. 2023). This initial observation supports the hypothesis that only all-or-none efference copy mechanisms enable adaptation to dynamic inversions. Further investigation into other goal-directed behaviours is required to strengthen this claim.

The all-or-none approach has notable limitations: it prevents responses to external disturbances during goal-directed movements and may struggle with unstable dynamics due to the absence of feedback. It may also block access to the error signal that could be used to adapt any feedforward controllers.

Despite these differences, both graded and all-or-none efference copies retain key advantages discussed in Chapter 2. For systems with stable dynamics, both mechanisms enable rapid

execution of desired movements without reliance on high-gain feedback control, making them well-suited for scenarios with sensor delays and noise. Both methods are likely employed depending on the control task.

Limitations of using a phenomenological model

The FSDoF control architecture used in this thesis serves as a phenomenological model to formalise the inner-outer-loop hypothesis from a control-theoretic perspective. However, the actual neural circuits that govern the motor control of insects are significantly more complex. For example, representing the entire sensory system as a single functional block is a substantial simplification. In reality, insects possess diverse sensor modalities that integrate information in highly nonlinear and context-dependent ways (Huston and Krapp 2008; Rosner, Egelhaaf, and Warzecha 2010). Moreover, mechanosensory pathways vary widely: some signals ascend to higher-order brain regions, others directly influence motor neurons, and some—such as those from the halteres—participate in both ascending and reflexive circuits (Chan and Dickinson 1996; Tuthill and Wilson 2016).

Despite these simplifications, the FSDoF framework captures a coherent subset of functional roles that the nervous system may perform. As such, it provides a useful baseline model upon which greater biological realism can be layered. Future work can incorporate additional pathways, nonlinearities, and feedback loops to more accurately reflect the distributed and redundant nature of insect motor control.

Stability bounds for implicit adaptive feedforward control

In Section 4.3.3, I derive bounds on the learning rates of the IAFF control scheme using a discretised loss function, showing that below a certain threshold, monotonically decreasing loss can be guaranteed. This is the first time such stability guarantees have been established for this class of adaptive controller (Abdelghani, Lillicrap, and Tweed 2008). Although the derivation relies on two assumptions that are often violated in practice, any resulting increases in error are typically transient, and the overall error trend still decreases when the learning rates remain below the derived thresholds.

A more significant limitation is that the analytical bounds depend on the norms of the filter inputs and both the estimated and actual sensitivity derivatives. While the input norms are typically easy to estimate based on context, the sensitivity derivative norms are harder to pre-determine. Numerical studies (App. A.4 and Sec. 4.4) illustrate how sensitive these stability bounds are to factors such as the initial weights, the feedback controller, the chosen error to reduce, and the learning rates themselves. As a result, these analytical bounds are difficult to apply without making conservative assumptions, which reduces their practical usefulness.

Future work could involve reformulating the IAFF method in state-space form and incorporating the adaptive filters into the state vector. This might enable a more conventional Lyapunov-based stability analysis, as used in other adaptive control frameworks (Behera, Kumar, and Patnaik 2006; Kumar, Srivastava, and Gupta 2018). The lack of stability guarantees in terms of more accessible and commonly estimated parameters may limit the adoption of the method, especially in safety-critical or high-reliability applications where formal performance guarantees are essential.

IAFF controls relevance to insects

Implicit adaptive feedforward (IAFF) control has been proposed in theoretical models of cerebellar learning, particularly following observations that humans can adapt to inverted dynamics (Abdelghani, Lillicrap, and Tweed 2008; Abdelghani and Tweed 2010). However, prior to this thesis, these principles had not been translated into a practical control architecture. To my knowledge, this thesis presents both the first control design and hardware implementation of IAFF.

I initially hypothesised that this principle might extend to insect motor adaptation, inspired by widely cited findings from Heisenberg and Wolf (1984), Wolf and Heisenberg (1986), and Wolf, Voss, et al. (1992). In particular, Heisenberg and Wolf (1984) reported that *Drosophila* can adapt to maintain bar fixation under positive feedback—effectively, an inversion of natural visuomotor dynamics. Similarly, Wolf, Voss, et al. (1992) suggested that flies can use a single leg on a force-sensing platform to fixate on a bar in the frontal visual field. Again, this scenario would involve significantly altered sensitivity derivatives. Together, the work from

the Heisenberg lab suggested insect adaptive control may use a strategy that either learns the sensitivity derivatives implicitly—as in IAFF (Chapter 4)—or uses a trial-and-error process such as reinforcement learning.

However, more recently, these results have been met with some scepticism (Mongeau et al. 2024). After checking all papers citing Wolf, Voss, et al. (1992) and Wolf and Heisenberg (1986) (as of May 2025), to my knowledge there were no successful replications of their findings, nor evidence of any invertebrate species adapting to inverted dynamics. Instead, Robert (1988) found that locusts were unable to adapt to inverted bar fixation dynamics under similar experimental conditions, and Strauss, Renner, and Götz (2001) showed no adaptation to inverted bar fixation in two *Drosophila* mutants.

To date, no publication has directly refuted the results of Wolf, Voss, et al. (1992) using identical methodology, so their findings remain a key reference throughout this thesis. Nevertheless, the connection between IAFF and insect motor learning would be strengthened by additional independent evidence that insects can adapt to inverted dynamics. Regardless of this open question, Chapter 4 presents a novel control architecture for robotics, and the first hardware implementation of the hypothesised cerebellar learning system. In addition, the hardware demonstration at least demonstrates viability for insect sensorimotor adaptation, which is one ability of using robotics to understand biology (Webb 2000).

Adaptive forward models?

Chapters 3 and 4 demonstrated that adaptive feedforward control can improve reference tracking when system dynamics change. In such cases, adapting the feedforward controller helps ensure that the actual state follows the desired state. However, if sensor gains change rather than the system dynamics, the appropriate response is not to modify the feedforward controller—since the required motor commands remain the same—but to adjust the forward model and efference copy to account for the altered reafferent sensory feedback.

This is particularly relevant for visual sensors, which exhibit variable gains depending on environmental factors such as contrast and luminance (Borst and Egelhaaf 1989; Laughlin and Hardie 1978). It remains an open question whether biological systems adapt the gain of their

forward models to account for that variability. Furthermore, in Section 5.4, I proposed a method for compensating for variable sensor delays by adapting delay terms in the forward model. Given that insects experience changes in sensory latency (Laughlin and Hardie 1978), it is possible that they compensate for this by adjusting the delays in their forward models.

Physiological experiments using augmented reality paradigms—in which visual feedback delays are artificially manipulated—could reveal whether insects actively adapt these forward model delays to maintain control performance. It is possible that they do not adapt their forward model, and that the resulting errors remain within tolerable limits. However, if they do adapt their forward models, it raises a further question: How do they determine when to adapt the forward model versus the feedforward controller, given that the only information available is the sensory error?

Adapting the forward model therefore carries substantial risk. In the experiment in Ch. 3, the change in system dynamics produces an increased response to expansive optic flow. If the forward model were adjusted to cancel this altered response, the animal would effectively mask the change and continue to fly faster, preventing corrective adjustments and undermining robust control. If the forward model does change, I propose it is more likely that they are context-dependent *updates* as opposed to *adaptive*. For example, illumination is readily measurable and could be used to select an appropriate gain and delay for the expected sensor response. In short, it seems more likely that forward model variation is constrained by contextual cues rather than by indiscriminate adaptation to cancel every sensory signal that results from a goal-directed behaviour.

macro:

macro:

Where is the forward model?

One of the principal remaining questions is the neuroanatomical location of the forward model. Although anatomy was not the primary focus of this thesis, the theoretical results reported may provide two insights that relate to where the forward model might plausibly reside. The first insight is that it is more appropriate to use a representation of the desired behaviour as input

to the forward model, and not the motor command (options i and ii in Fig. 1.2, respectively). The second insight is that the forward model does not need to be complicated.

macro:

macro:

Key information for locating the forward model is the anatomy of its inputs and outputs. Experimentally, Kim, Fitzgerald, and Maimon (2015) show in *Drosophila* that one output of the forward model is in the HSN cells, where the predicted sensory responses are cancelled. Extrapolating this finding to the wider visual system suggests that the site of cancellation is at an early stage of sensory processing (i.e., lobula plate tangential cells, LPTCs). The input to the forward model is unknown. Although the term *efference copy* implies a direct motor copy, the analysis in Sec. 3.3.2 indicates that using a representation of the desired behaviour as input to the forward model may be more appropriate than using raw motor commands, particularly when controlling unstable system dynamics. This may be more practical for two other reasons: (i) it would be a shorter neural pathway and therefore introduce smaller delays and less noise, and (ii) Fenk, Kim, and Maimon (2021) showed that no efference copies were created for stabilising manoeuvres. If the input to the forward model did ascend from motor neurons, a determination would be needed about whether this motor command was associated with a goal-directed movement or a reflex before providing it to the forward model. Desired behaviours are likely determined in the mushroom body (Webb 2024). Therefore, it is likely that the input to the forward model originates in the mushroom body.

Previous authors have proposed the mushroom body as a candidate site for a forward model (Webb 2019), partly on account of its cerebellum-like architecture (Farris 2011). The second insight from this thesis is that the forward model need not be a large, highly adaptive network: if it primarily implements a (possibly non-adaptive) temporal prediction of sensory consequences, then its function can be realised by the dynamics of only a few neurons acting as a temporal filter. This reduces the requirement that the structure be extensive, even if the mushroom body remains a plausible site.

Adaptive feedforward control of turning in blowflies?

Chapter 3 provides evidence consistent with the use of adaptive feedforward control in regulating flight speed. However, no such evidence was found for the control of trajectory curvature or elevation. As outlined in Section 2.3.3, one of the proposed advantages of forward models is their ability to cancel expected sensory input, thereby preventing sensory saturation and enhancing sensitivity to external disturbances.

During straight flight between saccades, the desired trajectory curvature and elevation are effectively zero, meaning the expected rotational motion (i.e., roll, pitch, and yaw) is minimal. In this case, feedback alone is sufficient for control. In contrast, controlling for non-zero speed creates continuous exposure to expansive translational optic flow. To maintain sensitivity and prevent saturation, an efference copy must subtract this expected sensory input, shifting the feedback loop's operating point to zero. Consequently, a sustained non-zero feedforward motor command becomes necessary to achieve the desired speed. Chapter 3 suggests that it is the adaptation of this feedforward controller that underlies the persistent changes in speed.

The absence of adaptive changes in curvature and elevation may be explained by the absence of a non-zero desired setpoint in these degrees of freedom; since the target trajectory is straight, no feedforward contribution is required. Therefore, feedback control alone may suffice, explaining the lack of observed adaptation in curvature.

This interpretation leads to a testable hypothesis: if insects perform behaviours with sustained non-zero curvature or elevation—such as banked turns or climbing manoeuvres—adaptive feedforward components may emerge in the control of these trajectory components. Repeating the experimental paradigm used in Chapter 3 in blowflies (*Calliphora*), which exhibit banking turns, could reveal whether similar adaptation patterns occur in turning dynamics. Confirmation of this would provide evidence for adaptive feedforward control that extends beyond speed regulation to include control of curvature and turning.

What are the *quantitative* benefits of efference copies to insects?

The relevance of the results presented in Chapters 2 and 5 to insect sensorimotor control is limited by the fact that the demonstrated FSDoF architecture was applied to various generic,

rather than insect-specific, system dynamics. Consequently, while these findings support the conceptual benefits of efference copies, they do not permit quantitative predictions about their advantages in insect flight control.

In principle, a similar controller comparison could be performed using system dynamics more closely based on insect flight. This could include (i) simplified equations of motion derived from empirical perturbation studies, as in Gao, Aono, and Liu (2011), or (ii) computational fluid dynamics models of insect flight dynamics, such as those used in Humbert et al. (2024). However, these approaches face significant limitations. Current models of insect flight dynamics neglect key features such as sensor noise, muscle dynamics, and environmental variability—all of which substantially affect control design. Crucially, most existing models are valid only near a fixed operating point and do not capture the nonlinearities associated with large state changes. Because FSDoF is fundamentally about goal-directed state changes, models restricted to local linearisations are insufficient. As a result, it is unlikely that applying FSDoF to current insect-derived models would provide meaningful quantitative insight into the value of efference copies in insects.

A more promising alternative may lie in goal-directed system identification. Using speed as an example, Fry, Rohrseitz, et al. (2009) demonstrated that feedback controllers governing flight speed can be estimated directly from free-flight data. Extending this approach, one could analyse behavioural data during goal-directed speed changes—such as take-off, landing, or the characteristic acceleration that occurs 140 ms before a saccade (Mronz and Lehmann 2008)—to estimate the dynamics of feedforward controller adjustments to speed. Comparing the timescales and bandwidths of feedback and presumed feedforward responses could quantify the gap between feedback performance and observed behaviour. This comparison could reveal whether the already identified feedback controller alone is sufficient to explain the rate of desired speed changes, or whether efference copies and feedforward control is required. Notably, Schwyn et al. (2011) describes a visual feedback controller for gaze stabilisation that is highly robust, with unusually large gain and phase margins. While this makes it resistant to noise, it comes at the cost of tracking performance. Similar trade-offs may apply in speed control. Although assumptions about the animal’s internal goals (e.g., desired speed) are unavoidable in such

analyses, an initial approximation could be made by inferring the desired speed from the final observed state. This behavioural approach, particularly when extended to goal-directed actions across different goal-directed systems, may offer a more tractable route toward estimating the quantitative benefits of efference copies.

Concluding remarks

This thesis used modelling and simulation to explore the advantages of insect-inspired control methods, specifically inner–outer-loop control. These insights motivated behavioural experiments that supported the hypothesis that insects may use adaptive feedforward control. Building on this hypothesis and previous findings, the implicit adaptive feedforward (IAFF) control method was developed. Despite certain limitations, the findings presented in this thesis open promising avenues for future research. These include investigating whether other insect behaviours exhibit patterns consistent with adaptive feedforward control, or seeking electrophysiological evidence that support the underlying neural mechanisms. Further work could also aim to improve the IAFF method, demonstrate it in a broader range of robotic systems, and compare its strengths and limitations with other adaptive control strategies. Such efforts will help control engineers make more informed choices when selecting control methods for specific applications. Overall, this work represents a small step towards two long-term goals: (i) understanding the control principles that enable insects to exhibit remarkable agility and manoeuvrability despite their limited neural resources, and (ii) developing robots that can match humans and animals in sensorimotor learning

References

- Abdelghani, MN, Timothy P Lillicrap, and Douglas B Tweed (2008). “Sensitivity derivatives for flexible sensorimotor learning”. In: *Neural computation* 20.8, pp. 2085–2111.
- Abdelghani, Mohamed N and Douglas B Tweed (2010). “Learning course adjustments during arm movements with reversed sensitivity derivatives”. In: *BMC neuroscience* 11, pp. 1–10.
- Adam, EJ, HA Latchman, and OD Crisalle (2000). “Robustness of the Smith predictor with respect to uncertainty in the time-delay parameter”. In: *Proceedings of the 2000 American Control Conference. ACC (IEEE Cat. No. 00CH36334)*. Vol. 2. IEEE, pp. 1452–1457.
- Allgower, Frank, Rolf Findeisen, Zoltan K Nagy, et al. (2004). “Nonlinear model predictive control: From theory to application”. In: *Journal-Chinese Institute Of Chemical Engineers* 35.3, pp. 299–316.
- Allgöwer, Frank and Alex Zheng (2012). *Nonlinear model predictive control*. Vol. 26. Birkhäuser.
- Arain, Bilal and Farid Kendoul (2014). “Real-time wind speed estimation and compensation for improved flight”. In: *IEEE Transactions on Aerospace and Electronic Systems* 50.2, pp. 1599–1606.
- Araki, M (2009). “PID control”. In: *Control Systems, Robotics and Automation: System Analysis and Control: Classical Approaches II*, pp. 58–79.
- Araki, Mituhiko and Hidefumi Taguchi (2003). “Two-degree-of-freedom PID controllers”. In: *International Journal of Control, Automation, and Systems* 1.4, pp. 401–411.
- Astrom, KJ and RM Murray (2008). “Frequency Domain Design”. In: *Feedback Systems: An Introduction for Scientists and Engineers*, pp. 315–346.

- Azid, Sheikh Izzal, Sohail Ahmad Ali, Meshach Kumar, Maurizio Cirrincione, and Adriano Fagiolini (2023). “Precise trajectory tracking of multi-rotor UAVs using wind disturbance rejection approach”. In: *IEEE access*.
- Barrett, Samuel, Matthew E Taylor, and Peter Stone (2010). “Transfer learning for reinforcement learning on a physical robot”. In: *Ninth International Conference on Autonomous Agents and Multiagent Systems-Adaptive Learning Agents Workshop (AAMAS-ALA)*. Vol. 1.
- Basel, Alali, Kentaro Hirata, and Kenji Sugimoto (2007). “Generalization of feedback error learning (FEL) to MIMO systems”. In: *Transactions of the Society of Instrument and Control Engineers* 43.4, pp. 293–302.
- Bastian, Amy J (2006). “Learning to predict the future: the cerebellum adapts feedforward movement control”. In: *Current opinion in neurobiology* 16.6, pp. 645–649.
- Behera, Laxmidhar, Swagat Kumar, and Awhan Patnaik (2006). “On adaptive learning rate that guarantees convergence in feedforward networks”. In: *IEEE transactions on neural networks* 17.5, pp. 1116–1125.
- Bell, CC and K Grant (1989). “Corollary discharge inhibition and preservation of temporal information in a sensory nucleus of mormyrid electric fish”. In: *Journal of Neuroscience* 9.3, pp. 1029–1044.
- Bell, Curtis C (1989). “Sensory coding and corollary discharge effects in mormyrid electric fish”. In: *Journal of Experimental Biology* 146.1, pp. 229–253.
- Berenji, Hamid R (1992). “Fuzzy logic controllers”. In: *An introduction to fuzzy logic applications in intelligent systems*. Springer, pp. 69–96.
- Borst, Alexander and Martin Egelhaaf (1989). “Principles of visual motion detection”. In: *Trends in neurosciences* 12.8, pp. 297–306.
- Borst, Alexander and Lukas N Groschner (2023). “How flies see motion”. In: *Annual review of neuroscience* 46.1, pp. 17–37.
- Botzer, Lior and Amir Karniel (2013). “Feedback and feedforward adaptation to visuomotor delay during reaching and slicing movements”. In: *European Journal of Neuroscience* 38.1, pp. 2108–2123.

- Breugel, Floris van, Marie P Suver, and Michael H Dickinson (2014). “Octopaminergic modulation of the visual flight speed regulator of *Drosophila*”. In: *Journal of Experimental Biology* 217.10, pp. 1737–1744.
- Buchner, Erich (1976). “Elementary movement detectors in an insect visual system”. In: *Biological cybernetics* 24.2, pp. 85–101.
- Budick, Seth A, Michael B Reiser, and Michael H Dickinson (2007). “The role of visual and mechanosensory cues in structuring forward flight in *Drosophila melanogaster*”. In: *Journal of Experimental Biology* 210.23, pp. 4092–4103.
- Burdet, Etienne, Rieko Osu, David W Franklin, Theodore E Milner, and Mitsuo Kawato (2001). “The central nervous system stabilizes unstable dynamics by learning optimal impedance”. In: *Nature* 414.6862, pp. 446–449.
- Burkhardt, Dietrich and Michael Gewecke (1965). “Mechanoreception in Arthropoda: the chain from stimulus to behavioral pattern”. In: *Cold Spring Harbor symposia on quantitative biology*. Vol. 30. Cold Spring Harbor Laboratory Press, pp. 601–614.
- Cahill, Andy (2011). “Catastrophic forgetting in reinforcement-learning environments”. PhD thesis. University of Otago.
- Camacho, Eduardo F, Carlos Bordons, Eduardo F Camacho, and Carlos Bordons (2007). *Constrained model predictive control*. Springer.
- Campbell, Benjamin P, Huai-Ti Lin, and Holger G Krapp (2023a). “A control engineering perspective on the advantages of efference copies”. In: *bioRxiv*, pp. 2023–07.
- (2023b). “Weighting elementary movement detectors tuned to different temporal frequencies to estimate image velocity”. In: *Conference on Biomimetic and Biohybrid Systems*. Springer, pp. 398–410.
- Campo, Peter J and Manfred Morari (1987). “Robust model predictive control”. In: *1987 American control conference*. IEEE, pp. 1021–1026.
- Campos, Eric O, Harvey D Bradshaw Jr, and Thomas L Daniel (2015). “Shape matters: corolla curvature improves nectar discovery in the hawkmoth *Manduca sexta*”. In: *Functional ecology* 29.4, pp. 462–468.

- Canelo, Angel, Yeon Kim, Jeongmin Park, and Anmo Kim (2023). “A visual efference copy-based navigation algorithm in *Drosophila* for complex visual environments”. In: *bioRxiv*, pp. 2023–10.
- Card, Gwyneth and Michael Dickinson (2008a). “Performance trade-offs in the flight initiation of *Drosophila*”. In: *Journal of Experimental Biology* 211.3, pp. 341–353.
- Card, Gwyneth and Michael H Dickinson (2008b). “Visually mediated motor planning in the escape response of *Drosophila*”. In: *Current Biology* 18.17, pp. 1300–1307.
- Casellato, Claudia, A Pedrocchi, JA Garrido, NR Luque, Giancarlo Ferrigno, E D’Angelo, and E Ros (2012). “An integrated motor control loop of a human-like robotic arm: feedforward, feedback and cerebellum-based learning”. In: *2012 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*. IEEE, pp. 562–567.
- Cellini, Benjamin, Marioalberto Ferrero, and Jean-Michel Mongeau (2024). “*Drosophila* flying in augmented reality reveals the vision-based control autonomy of the optomotor response”. In: *Current Biology* 34.1, pp. 68–78.
- Chan, Wai Pang and Michael H Dickinson (1996). “Position-specific central projections of mechanosensory neurons on the haltere of the blow fly, *Calliphora vicina*”. In: *Journal of Comparative Neurology* 369.3, pp. 405–418.
- Chan, Wai Pang, Frederick Prete, and Michael H. Dickinson (1998). “Visual Input to the Efferent Control System of a Fly’s “Gyroscope””. In: *Science* 280.5361, pp. 289–292. DOI: [10.1126/science.280.5361.289](https://doi.org/10.1126/science.280.5361.289). eprint: <https://www.science.org/doi/pdf/10.1126/science.280.5361.289>. URL: <https://www.science.org/doi/abs/10.1126/science.280.5361.289>.
- Chen, Shaoru, Ning-Yuan Li, Victor M Preciado, and Nikolai Matni (2022). “Robust model predictive control of time-delay systems through system level synthesis”. In: *2022 IEEE 61st conference on decision and control (CDC)*. IEEE, pp. 6902–6909.
- Chen, Zhilu and Xinming Huang (2017). “End-to-end learning for lane keeping of self-driving cars”. In: *2017 IEEE intelligent vehicles symposium (IV)*. IEEE, pp. 1856–1860.

- Chiappe, M Eugenia, Johannes D Seelig, Michael B Reiser, and Vivek Jayaraman (2010). “Walking modulates speed sensitivity in *Drosophila* motion vision”. In: *Current Biology* 20.16, pp. 1470–1475.
- Collett, TS (1980). “Angular tracking and the optomotor response an analysis of visual reflex interaction in a hoverfly”. In: *Journal of Comparative Physiology* 140, pp. 145–158.
- Cope, Alex J, Chelsea Sabo, Kevin Gurney, Eleni Vasilaki, and James AR Marshall (2016). “A model for an angular velocity-tuned motion detector accounting for deviations in the corridor-centering response of the bee”. In: *PLoS computational biology* 12.5, e1004887.
- Cowan, Noah J, Mert M Ankarali, Jonathan P Dyhr, Manu S Madhav, Eatai Roth, Shahin Sefati, Simon Sponberg, Sarah A Stamper, Eric S Fortune, and Thomas L Daniel (2014). “Feedback control as a framework for understanding tradeoffs in biology”. In: *American Zoologist* 54.2, pp. 223–237.
- Crampton, William GR (2019). “Electroreception, electrogenesis and electric signal evolution”. In: *Journal of Fish Biology* 95.1, pp. 92–134.
- Crapse, Trinity B and Marc A Sommer (2008). “Corollary discharge across the animal kingdom”. In: *Nature Reviews Neuroscience* 9.8, pp. 587–600.
- Dai, Shicong, Taeyoung Lee, and Dennis S Bernstein (2014). “Adaptive control of a quadrotor UAV transporting a cable-suspended load with unknown mass”. In: *53rd IEEE Conference on Decision and Control*. IEEE, pp. 6149–6154.
- Dauzere-Peres, Océane and Antoine Wystrach (2024). “Ants integrate proprioception as well as visual context and efference copies to make robust predictions”. In: *Nature Communications* 15.1, p. 10205.
- Davidovich, Hilla and Gal Ribak (2018). “Loaded flight in male *Ischnura elegans* and its relationship to copulatory flight”. In: *Journal of insect physiology* 110, pp. 44–56.
- Dean, Paul, John Porrill, Carl-Fredrik Ekerot, and Henrik Jörntell (2010). “The cerebellar microcircuit as an adaptive filter: experimental and computational evidence”. In: *Nature Reviews Neuroscience* 11.1, pp. 30–43.
- Deora, Tanvi, Mahad A Ahmed, Thomas L Daniel, and Bing W Brunton (2021). “Tactile active sensing in an insect plant pollinator”. In: *Journal of Experimental Biology* 224.4, jeb239442.

- Deora, Tanvi, Siddharth S Sane, and Sanjay P Sane (2021). “Wings and halteres act as coupled dual oscillators in flies”. In: *Elife* 10, e53824.
- Di Francesco, Gabriele, Egidio D’Amato, and Massimiliano Mattei (2015). “INDI control with direct lift for a tilt rotor UAV”. In: *IFAC-PapersOnLine* 48.9, pp. 156–161.
- Dickinson, Michael H (1999). “Haltere-mediated equilibrium reflexes of the fruit fly, *Drosophila melanogaster*”. In: *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences* 354.1385, pp. 903–916.
- Ding, Ying, Liang Wang, Yongwei Li, and Daoliang Li (2018). “Model predictive control and its application in agriculture: A review”. In: *Computers and Electronics in Agriculture* 151, pp. 104–117.
- Dorato, Peter (1987). “A historical review of robust control”. In: *IEEE Control Systems Magazine* 7.2, pp. 44–47.
- Dubowy, Christine and Amita Sehgal (2017). “Circadian rhythms and sleep in *Drosophila melanogaster*”. In: *Genetics* 205.4, pp. 1373–1397.
- Evans, Howard E (1962). “A review of nesting behavior of digger wasps of the genus *Aphilandrops*, with special attention to the mechanics of prey carriage”. In: *Behaviour*, pp. 239–260.
- Fabian, Joseph, Igor Siwanowicz, Myriam Uhrhan, Masateru Maeda, Richard J Bomphrey, and Huai-Ti Lin (2021). “Complete neuroanatomy and sensor maps of Odonata wings for fly-by-feel flight control”. In: *bioRxiv*, pp. 2021–04.
- Fabian, Samuel T, Mary E Sumner, Trevor J Wardill, Sergio Rossoni, and Paloma T Gonzalez-Bellido (2018). “Interception by two predatory fly species is explained by a proportional navigation feedback controller”. In: *Journal of The Royal Society Interface* 15.147, p. 20180466.
- Fabian, ST and HT Lin (2023). “The Dogfighting Tactics of Houseflies”. In: *INTEGRATIVE AND COMPARATIVE BIOLOGY*. Vol. 62. OXFORD UNIV PRESS INC JOURNALS DEPT, 2001 EVANS RD, CARY, NC 27513 USA, S90–S90.
- Faisal, A Aldo, Luc PJ Selen, and Daniel M Wolpert (2008). “Noise in the nervous system”. In: *Nature reviews neuroscience* 9.4, pp. 292–303.
- Falb, P and William Wolovich (1967). “Decoupling in the design and synthesis of multivariable control systems”. In: *IEEE transactions on automatic control* 12.6, pp. 651–659.

- Farris, Sarah M (2011). “Are mushroom bodies cerebellum-like structures?” In: *Arthropod structure & development* 40.4, pp. 368–379.
- Fenk, Lisa M, Anmo J Kim, and Gaby Maimon (2021). “Suppression of motion vision during course-changing, but not course-stabilizing, navigational turns”. In: *Current Biology* 31.20, pp. 4608–4619.
- Flesch, Rodolfo CC, Bismark C Torrico, Julio E Normey-Rico, and Marcus U Cavalcante (2011). “Unified approach for minimal output dead time compensation in MIMO processes”. In: *Journal of Process Control* 21.7, pp. 1080–1091.
- Fraenkel, Gottfried (1939). “The function of the halteres of flies (Diptera)”. In: *Proceedings of the Zoological Society of London*. Vol. 109. 1. Wiley Online Library, pp. 69–78.
- Fromion, Vincent and Gérard Scorletti (2003). “A theoretical framework for gain scheduling”. In: *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal* 13.10, pp. 951–982.
- Fry, Steven N, Nicola Rohrseitz, Andrew D Straw, and Michael H Dickinson (2009). “Visual control of flight speed in *Drosophila melanogaster*”. In: *Journal of Experimental Biology* 212.8, pp. 1120–1130.
- Fry, Steven N, Rosalyn Sayaman, and Michael H Dickinson (2003). “The aerodynamics of free-flight maneuvers in *Drosophila*”. In: *Science* 300.5618, pp. 495–498.
- Fujita, Masahiko (2005). “Feed-forward associative learning for volitional movement control”. In: *Neuroscience research* 52.2, pp. 153–165.
- Fuller, Sawyer Buckminster, Andrew D Straw, Martin Y Peek, Richard M Murray, and Michael H Dickinson (2014). “Flying *Drosophila* stabilize their vision-based velocity controller by sensing wind with their antennae”. In: *Proceedings of the National Academy of Sciences* 111.13, E1182–E1191.
- Gahinet, Pascal, Rong Chen, and Bora Eryilmaz (June 2013). “Automated PID controller design”. US8467888B2. URL: <https://patents.google.com/patent/US8467888B2>.
- Gao, Na, Hikaru Aono, and Hao Liu (2011). “Perturbation analysis of 6DoF flight dynamics and passive dynamic stability of hovering fruit fly *Drosophila melanogaster*”. In: *Journal of theoretical biology* 270.1, pp. 98–111.

- Garcia, Carlos E, David M Prett, and Manfred Morari (1989). “Model predictive control: Theory and practice—A survey”. In: *Automatica* 25.3, pp. 335–348.
- García, Pedro and Pedro Albertos (2010). “Dead-time-compensator for unstable MIMO systems with multiple time delays”. In: *Journal of Process Control* 20.7, pp. 877–884.
- Gebehart, Corinna and Ansgar Büschges (2021). “Temporal differences between load and movement signal integration in the sensorimotor network of an insect leg”. In: *Journal of Neurophysiology* 126.6, pp. 1875–1890.
- Geiger, Andreas, Frank Moosmann, Ömer Car, and Bernhard Schuster (2012). “Automatic camera and range sensor calibration using a single shot”. In: *2012 IEEE international conference on robotics and automation*. IEEE, pp. 3936–3943.
- Gershon, Eli, Uri Shaked, and Isaac Yaesh (2005). *H-infinity control and estimation of state-multiplicative linear systems*. Vol. 318. Springer science & business media.
- Giraldo, Sergio Andres Castaño, Rodolfo César Costa Flesch, Julio Elias Normey-Rico, and Miriam Zareth Parra Sejas (2018). “A method for designing decoupled filtered smith predictor for square MIMO systems with multiple time delays”. In: *IEEE Transactions on Industry Applications* 54.6, pp. 6439–6449.
- Götz, Karl Georg (1968). “Flight control in Drosophila by visual perception of motion”. In: *Kybernetik* 4, pp. 199–208.
- Green, Michael and David JN Limebeer (2012a). *Linear robust control*. Courier Corporation.
- (2012b). *Linear robust control*. Courier Corporation.
- Grimholt, Chriss and Sigurd Skogestad (2019). “Should we forget the smith predictor?” In.
- Gupta, Abhishek, Russell Mendonca, YuXuan Liu, Pieter Abbeel, and Sergey Levine (2018). “Meta-reinforcement learning of structured exploration strategies”. In: *Advances in neural information processing systems* 31.
- Haag, Juergen, Frederic Theunissen, and Alexander Borst (1997). “The intrinsic electrophysiological characteristics of fly lobula plate tangential cells: II. Active membrane properties”. In: *Journal of computational neuroscience* 4, pp. 349–368.
- Hammond, Sarah and Michael O’Shea (2007). “Escape flight initiation in the fly”. In: *Journal of Comparative Physiology A* 193, pp. 471–476.

- Hansson, Bill S and Marcus C Stensmyr (2011). “Evolution of insect olfaction”. In: *Neuron* 72.5, pp. 698–711.
- Hardcastle, Ben J and Holger G Krapp (2016). “Evolution of biological image stabilization”. In: *Current Biology* 26.20, R1010–R1021.
- Harris, Robert A, David C O’Carroll, and Simon B Laughlin (2000). “Contrast gain reduction in fly motion adaptation”. In: *Neuron* 28.2, pp. 595–606.
- Hassenstein, Bernhard and Werner Reichardt (1951). “Funktionsanalyse der Bewegungsperzeption eines Käfers”. In: *Naturwissenschaften* 38.21, pp. 507–507.
- (1953). “Der schluss von reiz-reaktions-funktionen auf system-strukturen”. In: *Zeitschrift für Naturforschung B* 8.9, pp. 518–524.
- Hausen, Klaus (1993). “Decoding of retinal image flow in insects”. In: *Visual Motion and its Role in the Stabilization of Gaze*, pp. 203–235.
- Heisenberg, M and R Wolf (1984). “Vision in Drosophila. Genetics of microbehaviour.” In: Heisenberg, Martin and Reinhard Wolf (2013). *Vision in Drosophila: genetics of microbehavior*. Vol. 12. Springer-Verlag.
- Hong, Dooyoung, Seonhoon Lee, Young Hoo Cho, Donkyu Baek, Jaemin Kim, and Naehyuck Chang (2021). “Energy-efficient online path planning of multiple drones using reinforcement learning”. In: *IEEE Transactions on Vehicular Technology* 70.10, pp. 9725–9740.
- Hou, Zhongsheng and Shangtai Jin (2013). *Model free adaptive control*. CRC press Boca Raton, FL, USA:
- Humbert, J Sean, Holger G Krapp, James D Baeder, Camli Badrya, Inés L Dawson, Jiaqi V Huang, Andrew Hyslop, Yong Su Jung, Cosima Lutkus, Beth Mortimer, et al. (2024). “Motion vision is tuned to maximize sensorimotor energy transfer in blowfly flight”. In: *bioRxiv*, pp. 2024–03.
- Huston, Stephen J. and Holger G. Krapp (July 2008). “Visuomotor transformation in the fly gaze stabilization system”. In: *PLoS Biology* 6.7, pp. 1468–1478. ISSN: 15449173. DOI: [10.1371/journal.pbio.0060173](https://doi.org/10.1371/journal.pbio.0060173).

- Ifere, Nwuneke Okereke, Hisashi Shidara, Nodoka Sato, and Hiroto Ogawa (2022). “Spatial perception mediated by insect antennal mechanosensory system”. In: *Journal of Experimental Biology* 225.4, jeb243276.
- Inc., The MathWorks (2023a). *Aerospace Toolbox version: 4.2 (R2022a)*. Natick, Massachusetts, United States. URL: <https://www.mathworks.com>.
- (2023b). *Simulink support package for parrot minidrones: 22.1.1 (R2022a)*. Natick, Massachusetts, United States. URL: <https://www.mathworks.com>.
- Isakov, Alexander, Sean M Buchanan, Brian Sullivan, Akshitha Ramachandran, Joshua KS Chapman, Edward S Lu, L Mahadevan, and Benjamin de Bivort (2016). “Recovery of locomotion after injury in *Drosophila melanogaster* depends on proprioception”. In: *Journal of Experimental Biology* 219.11, pp. 1760–1771.
- Jantzen, Benjamin and Thomas Eisner (2008). “Hindwings are unnecessary for flight but essential for execution of normal evasive flight in *Lepidoptera*”. In: *Proceedings of the National Academy of Sciences* 105.43, pp. 16636–16640.
- Jung, Sarah Nicola, Alexander Borst, and Juergen Haag (2011a). “Flight activity alters velocity tuning of fly motion-sensitive neurons”. In: *Journal of Neuroscience* 31.25, pp. 9231–9237.
- (2011b). “Flight activity alters velocity tuning of fly motion-sensitive neurons”. In: *Journal of Neuroscience* 31.25, pp. 9231–9237.
- Kagaya, K and SN Patek (2016). “Feed-forward motor control of ultrafast, ballistic movements”. In: *Journal of Experimental Biology* 219.3, pp. 319–333.
- Kase, Wataru and Yasuhiko Mutoh (2009). “A simple derivation of the interactor matrix and its applications”. In: *International Journal of Systems Science* 40.11, pp. 1197–1205.
- Kaufmann, Elia, Leonard Bauersfeld, Antonio Loquercio, Matthias Müller, Vladlen Koltun, and Davide Scaramuzza (2023). “Champion-level drone racing using deep reinforcement learning”. In: *Nature* 620.7976, pp. 982–987.
- Kawato, Mitsuo (1990). “Feedback-error-learning neural network for supervised motor learning”. In: *Advanced neural computers*. Elsevier, pp. 365–372.
- Kim, Anmo J, Lisa M Fenk, Cheng Lyu, and Gaby Maimon (2017). “Quantitative predictions orchestrate visual signaling in *Drosophila*”. In: *Cell* 168.1, pp. 280–294.

- Kim, Anmo J, Jamie K Fitzgerald, and Gaby Maimon (2015). “Cellular evidence for efference copy in *Drosophila* visuomotor processing”. In: *Nature neuroscience* 18.9, pp. 1247–1255.
- Kirchner, WH and MV Srinivasan (1989). “Freely flying honeybees use image motion to estimate object distance”. In: *Naturwissenschaften* 76.6, pp. 281–282.
- Kirschfeld, K (1976). “The resolution of lens and compound eyes”. In: *Neural principles in vision*. Springer, pp. 354–370.
- Kormushev, Petar, Sylvain Calinon, and Darwin G Caldwell (2013). “Reinforcement learning in robotics: Applications and real-world challenges”. In: *Robotics* 2.3, pp. 122–148.
- Kouvaritakis, Basil and Mark Cannon (2016). “Model predictive control”. In: *Switzerland: Springer International Publishing* 38, pp. 13–56.
- Krapp, H. G. and M. Wicklein (2008). “Central processing of visual information in insects”. In: *The Senses: A Comprehensive Reference, Volume 1*. Ed. by R. Masland and T. D. Albright. San Diego: Academic Press, pp. 131–204.
- Krapp, Holger G (2015). “How a fly escapes the reflex trap”. In: *Nature neuroscience* 18.9, pp. 1192–1194.
- Krapp, Holger G and Roland Hengstenberg (1996). “Estimation of self-motion by optic flow processing in single visual interneurons”. In: *Nature* 384.6608, pp. 463–466.
- Krapp, Holger G, Graham K Taylor, and J Sean Humbert (2012). “The mode-sensing hypothesis: matching sensors, actuators and flight dynamics”. In: *Frontiers in sensing*. Springer, pp. 101–114.
- Kukillaya, Raghavendra P and Philip Holmes (2009). “A model for insect locomotion in the horizontal plane: feedforward activation of fast muscles, stability, and robustness”. In: *Journal of theoretical biology* 261.2, pp. 210–226.
- Kumar, Rajesh, Smriti Srivastava, and JRP Gupta (2018). “Comparative study of neural networks for control of nonlinear dynamical systems with lyapunov stability-based adaptive learning rates”. In: *Arabian Journal for Science and Engineering* 43, pp. 2971–2993.
- Land, Michael F (1997). “Visual acuity in insects”. In: *Annual review of entomology* 42.1, pp. 147–177.

- Laughlin, Simon B (2010). “Insects as a Guide to Neural Circuit Design”. In: *Handbook of Brain Microcircuits*, p. 404.
- Laughlin, Simon B and Roger C Hardie (1978). “Common strategies for light adaptation in the peripheral visual systems of fly and dragonfly”. In: *Journal of comparative physiology* 128, pp. 319–340.
- Lavretsky, Eugene and Kevin A Wise (2012). “Robust adaptive control”. In: *Robust and adaptive control: With aerospace applications*. Springer, pp. 317–353.
- Lehmann, F-O (1999). “Ambient temperature affects free-flight performance in the fruit fly *Drosophila melanogaster*”. In: *Journal of Comparative Physiology B* 169, pp. 165–171.
- Leonte, Maria-Bianca, Aljoscha Leonhardt, Alexander Borst, and Alex S Mauss (2021). “Aerial course stabilization is impaired in motion-blind flies”. In: *Journal of Experimental Biology* 224.14, jeb242219.
- Lepreux, Gaetan (2020). “Mediating Tactually Guided Behaviour: State-dependent and Sensory Modulation of Activity in an Identified Descending Interneuron”. PhD thesis. University of Bielefeld.
- Li, Guang, Jing Na, David P Stoten, and Xuemei Ren (2013). “Adaptive neural network feedforward control for dynamically substructured systems”. In: *IEEE Transactions on Control Systems Technology* 22.3, pp. 944–954.
- Lin, Suewei, Bhagyashree Senapati, and Chang-Hui Tsao (2019). “Neural basis of hunger-driven behaviour in *Drosophila*”. In: *Open biology* 9.3, p. 180259.
- Longden, Kit D and Holger G Krapp (2009). “State-dependent performance of optic-flow processing interneurons”. In: *Journal of neurophysiology* 102.6, pp. 3606–3618.
- (2011). “Sensory neurophysiology: motion vision during motor action”. In: *Current Biology* 21.17, R650–R652.
- Ma, Qun, Yibo Wu, Muhammad Usman Shoukat, Yukai Yan, Jun Wang, Long Yang, Fuwu Yan, and Lirong Yan (2024). “Deep Reinforcement Learning-Based Wind Disturbance Rejection Control Strategy for UAV”. In: *Drones* 8.11, p. 632.

- Maddess, T and Simon Barry Laughlin (1985). “Adaptation of the motion-sensitive neuron H1 is generated locally and governed by contrast frequency”. In: *Proceedings of the Royal society of London. Series B. Biological sciences* 225.1239, pp. 251–275.
- Maimon, Gaby, Andrew D Straw, and Michael H Dickinson (2010). “Active flight increases the gain of visual motion processing in *Drosophila*”. In: *Nature neuroscience* 13.3, pp. 393–399.
- Mamiya, Akira, Andrew D Straw, Egill Tómasson, and Michael H Dickinson (2011). “Active and passive antennal movements during visually guided steering in flying *Drosophila*”. In: *Journal of Neuroscience* 31.18, pp. 6900–6914.
- Manenti, Flavio (2011). “Considerations on nonlinear model predictive control techniques”. In: *Computers & Chemical Engineering* 35.11, pp. 2491–2509.
- Mankowska, Natalia D, Anna B Marcinkowska, Monika Waskow, Rita I Sharma, Jacek Kot, and Pawel J Winklewski (2021). “Critical flicker fusion frequency: a narrative review”. In: *Medicina* 57.10, p. 1096.
- MathWorks (2025a). *Band-Limited White Noise*. Simulink block documentation, accessed Online. MathWorks. URL: <https://uk.mathworks.com/help/simulink/slref/bandlimitedwhitenoise.html> (visited on 08/22/2025).
- (2025b). *csaps Function Documentation*. Accessed: August 26, 2025. The MathWorks, Inc. URL: <https://uk.mathworks.com/help/curvefit/csaps.html>.
- (n.d.). *MPC Designer - Specify Prediction and Control Horizons*. URL: <https://uk.mathworks.com/help/mpc/ug/plant-models-with-delays.html>.
- Matthews, Peter BC (1991). “The human stretch reflex and the motor cortex”. In: *Trends in neurosciences* 14.3, pp. 87–91.
- McCann, GD and GF MacGinitie (1965). “Optomotor response studies of insect vision”. In: *Proceedings of the Royal Society of London. Series B. Biological Sciences* 163.992, pp. 369–401.
- Medici, V and Steven N Fry (2012). “Embodied linearity of speed control in *Drosophila melanogaster*”. In: *Journal of The Royal Society Interface* 9.77, pp. 3260–3267.
- Messineo, Saverio and Andrea Serrani (2009). “Adaptive feedforward disturbance rejection in nonlinear systems”. In: *Systems & control letters* 58.8, pp. 576–583.

- Meyer, Hanno Gerd, Jens Peter Lindemann, and Martin Egelhaaf (2011). “Pattern-Dependent response modulations in motion-sensitive visual interneurons—a model study”. In: *PLoS One* 6.7, e21488.
- Miall, R Chris and Dominic King (2008). “State estimation in the cerebellum”. In: *The Cerebellum* 7.4, pp. 572–576.
- Miall, RC (1978). “The flicker fusion frequencies of six laboratory insects, and the response of the compound eye to mains fluorescent ‘ripple’”. In: *Physiological Entomology* 3.2, pp. 99–106.
- Mirzal, Andri (2017). “Delay compensation using the Smith predictor: a brief review with numerical examples”. In: *International Journal of Computer-aided Mechanical Design and Implementation* 3.1, pp. 1–8.
- Mischiati, Matteo, Huai-Ti Lin, Paul Herold, Elliot Imler, Robert Olberg, and Anthony Leonardo (2015). “Internal models direct dragonfly interception steering”. In: *Nature* 517.7534, pp. 333–338.
- Mishra, Monalisa and Elisabeth Knust (2013). “Analysis of the Drosophila compound eye with light and electron microscopy”. In: *Retinal Degeneration: Methods and Protocols*, pp. 161–182.
- Miyamoto, Hiroyuki, Mitsuo Kawato, Tohru Setoyama, and Ryoji Suzuki (1988). “Feedback-error-learning neural network for trajectory control of a robotic manipulator”. In: *Neural networks* 1.3, pp. 251–265.
- Mohammadzaheri, M and Reza Tafreshi (2017). “An enhanced Smith predictor based control system using feedback-feedforward structure for time-delay processes”. In: *The Journal of Engineering Research [TJER]* 14.2, pp. 156–165.
- Mongeau, Jean-Michel, Yu Yang, Ignacio Escalante, Noah Cowan, and Kaushik Jayaram (2024). “Moving in an uncertain world: robust and adaptive control of locomotion from organisms to machine intelligence”. In: *Integrative and comparative biology* 64.5, pp. 1390–1407.
- Mronz, Markus and Fritz-Olaf Lehmann (2008). “The free-flight response of Drosophila to motion of the visual environment”. In: *Journal of Experimental Biology* 211.13, pp. 2026–2045.
- Muijres, Florian T, Nicole A Iwasaki, Michael J Elzinga, Johan M Melis, and Michael H Dickinson (2017). “Flies compensate for unilateral wing damage through modular adjustments of wing and body kinematics”. In: *Interface Focus* 7.1, p. 20160103.

- Murlis, John, Joseph S Elkinton, Ring T Carde, et al. (1992). "Odor plumes and how insects use them". In: *Annual review of entomology* 37.1, pp. 505–532.
- Nalbach, G (1993). "The halteres of the blowfly *Calliphora*: I. Kinematics and dynamics". In: *Journal of Comparative Physiology A* 173, pp. 293–300.
- Nalbach, H-O (1989). "Three temporal frequency channels constitute the dynamics of the optokinetic system of the crab, *Carcinus maenas* (L.)" In: *Biological Cybernetics* 61.1, pp. 59–70.
- Namiki, Shigehiro, Ivo G Ros, Carmen Morrow, William J Rowell, Gwyneth M Card, Wyatt Korff, and Michael H Dickinson (2022). "A population of descending neurons that regulates the flight motor of *Drosophila*". In: *Current Biology* 32.5, pp. 1189–1196.
- Neely, G Gregory, Alex C Keene, Peter Duchek, Elaine C Chang, Qiao-Ping Wang, Yagiz Alp Aksoy, Mark Rosenzweig, Michael Costigan, Clifford J Woolf, Paul A Garrity, et al. (2011). "TrpA1 regulates thermal nociception in *Drosophila*". In: *PloS one* 6.8, e24343.
- Nguyen, Nhan, Kalmanje Krishnakumar, and Jovan Boskovic (2008). "An optimal control modification to model-reference adaptive control for fast adaptation". In: *AIAA Guidance, Navigation and Control Conference and Exhibit*, p. 7283.
- Nguyen, Nhan T (2018). *Model-reference adaptive control*. Springer.
- Niehaus, Monika (1981). "Flight and flight control by the antennae in the small tortoiseshell (*Aglais urticae* L., Lepidoptera) II. Flight mill and free flight experiments". In: *Journal of comparative physiology* 145, pp. 257–264.
- Niell, Cristopher M and Michael P Stryker (2010). "Modulation of visual responses by behavioral state in mouse visual cortex". In: *Neuron* 65.4, pp. 472–479.
- Normey-Rico, Julio Elias and Eduardo F Camacho (2007). *Control of dead-time processes*. Vol. 462. Springer.
- Ogata, Katsuhiko et al. (2010). *Modern control engineering*. Vol. 5. Prentice hall Upper Saddle River, NJ.
- Otsu, Nobuyuki (1979). "A threshold selection method from gray-level histograms". In: *IEEE transactions on systems, man, and cybernetics* 9.1, pp. 62–66.

- Palmor, Z (1980). “Stability properties of Smith dead-time compensator controllers”. In: *International Journal of Control* 32.6, pp. 937–949.
- Payne, Mark, Berthold Hedwig, and Barbara Webb (2010). “Multimodal Predictive Control in Crickets”. In: *From Animals to Animats 11*. Ed. by Stéphane Doncieux, Benoît Girard, Agnès Guillot, John Hallam, Jean-Arcady Meyer, and Jean-Baptiste Mouret. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 167–177. ISBN: 978-3-642-15193-4.
- Pisotta, Iolanda and Marco Molinari (2014). “Cerebellar contribution to feedforward control of locomotion”. In: *Frontiers in human neuroscience* 8, p. 475.
- Poulet, James FA and Berthold Hedwig (2002). “A corollary discharge maintains auditory sensitivity during sound production”. In: *Nature* 418.6900, pp. 872–876.
- (2006). “The cellular basis of a corollary discharge”. In: *Science* 311.5760, pp. 518–522.
- Poulet, JFA and B Hedwig (2003). “A corollary discharge mechanism modulates central auditory processing in singing crickets”. In: *Journal of neurophysiology* 89.3, pp. 1528–1540.
- Pounds, Paul Edward Ian (2007). “Design, Construction and Control of a Large Quadrotor Micro Air Vehicle”. Ph.D. thesis. Canberra, Australia: The Australian National University. DOI: [10.25911/5d651066ac8bd](https://doi.org/10.25911/5d651066ac8bd).
- Rajabi, Hamed, Jan-Henning Dirks, and Stanislav N Gorb (2020). “Insect wing damage: causes, consequences and compensatory mechanisms”. In: *Journal of Experimental Biology* 223.9, jeb215194.
- Rawlings, James B, Dennis Bonné, John B Jorgensen, Aswin N Venkat, and Sten Bay Jorgensen (2008). “Unreachable setpoints in model predictive control”. In: *IEEE Transactions on Automatic Control* 53.9, pp. 2209–2215.
- Reichardt, Werner (1957). “Autokorrelations-Auswertung als Funktionsprinzip des Zentralnervensystems: (bei der optischen Bewegungswahrnehmung eines Insektes)”. In: *Zeitschrift für Naturforschung B* 12.7, pp. 448–457.
- Riether, Fabian (2016). “Agile quadrotor maneuvering using tensor-decomposition-based globally optimal control and onboard visual-inertial estimation”. PhD thesis. Massachusetts Institute of Technology.

- Ristroph, Leif, Gunnar Ristroph, Svetlana Morozova, Attila J Bergou, Song Chang, John Guckenheimer, Z Jane Wang, and Itai Cohen (2013). “Active and passive stabilization of body pitch in insect flight”. In: *Journal of The Royal Society Interface* 10.85, p. 20130237.
- Robert, Daniel (1988). “Visual steering under closed-loop conditions by flying locusts: flexibility of optomotor response and mechanisms of correctional steering”. In: *Journal of Comparative Physiology A* 164, pp. 15–24.
- Rodríguez, C, JE Normey-Rico, JL Guzmán, and M Berenguel (2016). “On the filtered Smith predictor with feedforward compensation”. In: *Journal of Process Control* 41, pp. 35–46.
- Rodríguez, Carlos, José Luis Guzmán, Manuel Berenguel, and Tore Hägglund (2014). “Optimal feedforward compensators for systems with right-half plane zeros”. In: *Journal of Process Control* 24.4, pp. 368–374.
- Rohrseitz, Nicola and Steven N Fry (2011). “Behavioural system identification of visual flight speed control in *Drosophila melanogaster*”. In: *Journal of the Royal Society Interface* 8.55, pp. 171–185.
- Rosner, Ronny, Martin Egelhaaf, and A-K Warzecha (2010). “Behavioural state affects motion-sensitive neurones in the fly visual system”. In: *Journal of Experimental Biology* 213.2, pp. 331–338.
- Rugh, Wilson J (1990). “Analytical framework for gain scheduling”. In: *1990 American Control Conference*. IEEE, pp. 1688–1694.
- Saijo, Naoki, Ikuya Murakami, Shin’ya Nishida, and Hiroaki Gomi (2005). “Large-field visual motion directly induces an involuntary rapid manual following response”. In: *Journal of Neuroscience* 25.20, pp. 4941–4951.
- Salem, Wael, Benjamin Cellini, Eric Jaworski, and Jean-Michel Mongeau (2023). “Flies adaptively control flight to compensate for added inertia”. In: *Proceedings of the Royal Society B* 290.2008, p. 20231115.
- Salem, Wael, Benjamin Cellini, Heiko Kabutz, Hari Krishna Hari Prasad, Bo Cheng, Kaushik Jayaram, and Jean-Michel Mongeau (2022). “Flies trade off stability and performance via adaptive compensation to wing damage”. In: *Science Advances* 8.46, eabo0719.

- Salvato, Erica, Gianfranco Fenu, Eric Medvet, and Felice Andrea Pellegrino (2021). “Crossing the reality gap: A survey on sim-to-real transferability of robot controllers in reinforcement learning”. In: *IEEE Access* 9, pp. 153171–153187.
- Santos, Tito LM, Rodolfo CC Flesch, and Julio E Normey-Rico (2014). “On the filtered Smith predictor for MIMO processes with multiple time delays”. In: *Journal of Process Control* 24.4, pp. 383–400.
- Sayre-McCord, Thomas, Winter Guerra, Amado Antonini, Jasper Arneberg, Austin Brown, Guilherme Cavalheiro, Yajun Fang, Alex Gorodetsky, Dave McCoy, Sebastian Quilter, et al. (2018). “Visual-inertial navigation algorithm development using photorealistic camera simulation in the loop”. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 2566–2573.
- Schuitema, Erik, Martijn Wisse, Thijs Ramakers, and Pieter Jonker (2010). “The design of LEO: a 2D bipedal walking robot for online autonomous reinforcement learning”. In: *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 3238–3243.
- Schuling, FH, HAK Mastebroek, R Bult, and BPM Lenting (1989). “Properties of elementary movement detectors in the fly *Calliphora erythrocephala*”. In: *Journal of Comparative Physiology A* 165.2, pp. 179–192.
- Schwyn, Daniel A, Francisco JH Heras, Gino Bolliger, Matthew M Parsons, Holger G Krapp, and Reiko J Tanaka (2011). “Interplay between feedback and feedforward control in fly gaze stabilization”. In: *IFAC Proceedings Volumes* 44.1, pp. 9674–9679.
- Shao, Shengjia, Jason Tsai, Michal Mysior, Wayne Luk, Thomas Chau, Alexander Warren, and Ben Jeppesen (2018). “Towards hardware accelerated reinforcement learning for application-specific robotic control”. In: *2018 IEEE 29th International Conference on Application-specific Systems, Architectures and Processors (ASAP)*. IEEE, pp. 1–8.
- Sherk, Truman E (1978). “Development of the compound eyes of dragonflies (Odonata). III. Adult compound eyes”. In: *Journal of Experimental Zoology* 203.1, pp. 61–79.
- Sherman, Alana and Michael H Dickinson (2004). “Summation of visual and mechanosensory feedback in *Drosophila* flight control”. In: *Journal of experimental biology* 207.1, pp. 133–142.

- Sieberling, S, QP Chu, and JA Mulder (2010). “Robust flight control using incremental nonlinear dynamic inversion and angular acceleration prediction”. In: *Journal of guidance, control, and dynamics* 33.6, pp. 1732–1742.
- Smeur, Ewoud JJ, Qiping Chu, and Guido CHE De Croon (2016). “Adaptive incremental nonlinear dynamic inversion for attitude control of micro air vehicles”. In: *Journal of Guidance, Control, and Dynamics* 39.3, pp. 450–461.
- Smith, P (1998). “A simplified approach to nonlinear dynamic inversion based flight control”. In: *23rd atmospheric flight mechanics conference*, p. 4461.
- Socolovsky, Mariano, Roberto S Martins, Gilda Di Masi, Gonzalo Bonilla, and Mario Siqueira (2016). “Treatment of complete facial palsy in adults: comparative study between direct hemihypoglossal-facial neuroorrhaphy, hemihypoglossal-facial neuroorrhaphy with grafts, and masseter to facial nerve transfer”. In: *Acta neurochirurgica* 158, pp. 945–957.
- Sperry, Roger Wolcott (1950). “Neural basis of the spontaneous optokinetic response produced by visual inversion.” In: *Journal of comparative and physiological psychology* 43.6, p. 482.
- Stöckl, Anna L, Klara Kihlström, Steven Chandler, and Simon Sponberg (2017). “Comparative system identification of flower tracking performance in three hawkmoth species reveals adaptations for dim light vision”. In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 372.1717, p. 20160078.
- Strauss, Roland, Michael Renner, and Karl Götz (2001). “Task-specific association of photoreceptor systems and steering parameters in *Drosophila*”. In: *Journal of Comparative Physiology A* 187, pp. 617–632.
- Sun, Mao and Jian Tang (2002). “Lift and power requirements of hovering flight in *Drosophila virilis*”. In: *Journal of Experimental Biology* 205.16, pp. 2413–2427.
- Suver, Marie P, Akira Mamiya, and Michael H Dickinson (2012). “Octopamine neurons mediate flight-induced modulation of visual processing in *Drosophila*”. In: *Current Biology* 22.24, pp. 2294–2302.
- Tarawneh, Ghaith, Vivek Nityananda, Ronny Rosner, Steven Errington, William Herbert, Bruce G Cumming, Jenny CA Read, and Ignacio Serrano-Pedraza (2017). “Invisible noise obscures visible signal in insect motion detection”. In: *Scientific Reports* 7.1, p. 3496.

- Tarokh, M (1992). “Measures for controllability, observability and fixed modes”. In: *IEEE Transactions on Automatic Control* 37.8, pp. 1268–1273.
- Taylor, Graham K and Holger G Krapp (2007). “Sensory systems and flight stability: what do insects measure and why?” In: *Advances in insect physiology* 34, pp. 231–316.
- Theobald, Jamie C, Dario L Ringach, and Mark A Frye (2010). “Dynamics of optomotor responses in *Drosophila* to perturbations in optic flow”. In: *Journal of Experimental Biology* 213.8, pp. 1366–1375.
- Theobald, Jamie C, Patrick A Shoemaker, Dario L Ringach, and Mark A Frye (2010). “Theta motion processing in fruit flies”. In: *Frontiers in behavioral neuroscience* 4, p. 1751.
- Tipaldi, Massimo, Raffaele Iervolino, and Paolo Roberto Massenio (2022). “Reinforcement learning in spacecraft control applications: Advances, prospects, and challenges”. In: *Annual Reviews in Control* 54, pp. 1–23.
- Tuthill, John C and Rachel I Wilson (2016). “Mechanosensation and adaptive motor control in insects”. In: *Current Biology* 26.20, R1022–R1038.
- Uhrhan, Myriam J, Richard J Bomphrey, and Huai-Ti Lin (2024). “Flow sensing on dragonfly wings”. In: *Annals of the New York Academy of Sciences* 1536.1, pp. 107–121.
- Van Hateren, JH (1997). “Processing of natural time series of intensities by the visual system of the blowfly”. In: *Vision research* 37.23, pp. 3407–3416.
- Velez, Mariel M, Mathias F Wernet, Damon A Clark, and Thomas R Clandinin (2014). “Walking *Drosophila* align with the e-vector of linearly polarized light through directed modulation of angular acceleration”. In: *Journal of Comparative Physiology A* 200, pp. 603–614.
- Viollet, Stéphane and Jochen Zeil (2013). “Feed-forward and visual feedback control of head roll orientation in wasps (*Polistes humilis*, Vespidae, Hymenoptera)”. In: *Journal of Experimental Biology* 216.7, pp. 1280–1291.
- Von Erich von Holst, Horst Mittelstaedt (1950). “Das reafferenzprinzip”. In: *Die Naturwissenschaften* 37, pp. 464–476.
- Von Helmholtz, Hermann (1867). *Handbuch der physiologischen Optik*. Vol. 9. L. Voss.
- von Holst, Erich (1973). “The reafference principle”. In: *The Behavioral Physiology of Animals and Man*.

- Von Holst, Erich and Horst Mittelstaedt (1950). “Das reafferenzprinzip: Wechselwirkungen zwischen zentralnervensystem und peripherie”. In: *Naturwissenschaften* 37.20, pp. 464–476.
- Wang, Zhe and Tianzhen Hong (2020). “Reinforcement learning for building controls: The opportunities and challenges”. In: *Applied Energy* 269, p. 115036.
- Webb, Barbara (2000). “What does robotics offer animal behaviour?” In: *Animal behaviour* 60.5, pp. 545–558.
- (May 2004a). *Neural mechanisms for prediction: Do insects have forward models?* DOI: [10.1016/j.tins.2004.03.004](https://doi.org/10.1016/j.tins.2004.03.004).
- (2004b). “Neural mechanisms for prediction: do insects have forward models?” In: *Trends in neurosciences* 27.5, pp. 278–282.
- (2004c). “Small brains and minimalist emulation: When is an internal model no longer a model?” In: *Behavioral and Brain Sciences* 27.3, pp. 421–422.
- (2019). “The minds of insects”. In: *Andy Clark and his critics*, pp. 254–265.
- (2024). “Beyond prediction error: 25 years of modeling the associations formed in the insect mushroom body”. In: *Learning & Memory* 31.5, a053824.
- Webb, Barbara and Antoine Wystrach (2016). “Neural mechanisms of insect navigation”. In: *Current Opinion in Insect Science* 15. Pests and resistance * Behavioural ecology, pp. 27–39. ISSN: 2214-5745. DOI: <https://doi.org/10.1016/j.cois.2016.02.011>. URL: <https://www.sciencedirect.com/science/article/pii/S221457451630013X>.
- Webb, Barbara H. and Reid R. Harrison (Oct. 2000). “Integrating sensorimotor systems in a robot model of cricket behavior”. In: *Proceedings of SPIE 4196, Sensor Fusion and Decentralized Control in Robotic Systems III*, pp. 92–99. DOI: [10.1117/12.403710](https://doi.org/10.1117/12.403710). URL: <https://doi.org/10.1117/12.403710>.
- Weckström, Matti (1989). “Light and dark adaptation in fly photoreceptors: duration and time integral of the impulse response”. In: *Vision research* 29.10, pp. 1309–1317.
- Wolf, Reinhard and Martin Heisenberg (1986). “Visual orientation in motion-blind flies is an operant behaviour”. In: *Nature* 323.6084, pp. 154–156.
- (1991). “Basic organization of operant behavior as revealed in *Drosophila* flight orientation”. In: *Journal of Comparative Physiology A* 169, pp. 699–705.

- Wolf, Reinhard, Andreas Voss, Sigrid Hein, and Martin Heisenberg (1992). “Can a fly ride a bicycle?” In: *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences* 337.1281, pp. 261–269.
- Wolpert, Daniel M, Zoubin Ghahramani, and Michael I Jordan (1994). “Perceptual distortion contributes to the curvature of human reaching movements”. In: *Experimental brain research* 98, pp. 153–156.
- Wolpert, Daniel M and Mitsuo Kawato (1998). “Multiple paired forward and inverse models for motor control”. In: *Neural networks* 11.7-8, pp. 1317–1329.
- Wolpert, Daniel M, R Chris Miall, and Mitsuo Kawato (1998). “Internal models in the cerebellum”. In: *Trends in cognitive sciences* 2.9, pp. 338–347.
- Wood, RK and MW Berry (1973). “Terminal composition control of a binary distillation column”. In: *Chemical Engineering Science* 28.9, pp. 1707–1717.
- Wurman, Peter R, Samuel Barrett, Kenta Kawamoto, James MacGlashan, Kaushik Subramanian, Thomas J Walsh, Roberto Capobianco, Alisa Devlic, Franziska Eckert, Florian Fuchs, et al. (2022). “Outracing champion Gran Turismo drivers with deep reinforcement learning”. In: *Nature* 602.7896, pp. 223–228.
- Yarger, Alexandra M and Jessica L Fox (2016). *Dipteran halteres: perspectives on function and integration for a unique sensory organ*.
- Zanker, Johannes M, Mandyam V Srinivasan, and Martin Egelhaaf (1999). “Speed tuning in elementary motion detectors of the correlation type”. In: *Biological cybernetics* 80.2, pp. 109–116.
- Zhang, Dan and Bin Wei (2017). “A review on model reference adaptive control of robotic manipulators”. In: *Annual Reviews in Control* 43, pp. 188–198.
- Zhang, Tengting and Hongwei Mo (2021). “Reinforcement learning for robot research: A comprehensive review and open issues”. In: *International Journal of Advanced Robotic Systems* 18.3, p. 17298814211007305.
- Zhong, Hua, Lucy Pao, and Raymond de Callafon (2012). “Feedforward control for disturbance rejection: Model matching and other methods”. In: *2012 24th Chinese Control and Decision Conference (CCDC)*. IEEE, pp. 3528–3533.

- Zhou, Kemin and John Comstock Doyle (1998). *Essentials of robust control*. Vol. 104. Prentice hall Upper Saddle River, NJ.
- Zurek, Daniel B and Cole Gilbert (2014). “Static antennae act as locomotory guides that compensate for visual motion blur in a diurnal, keen-eyed predator”. In: *Proceedings of the Royal Society B: Biological Sciences* 281.1779, p. 20133072.

Appendix A

Further stability analysis of IAFF

A.1 α_g upper-bound derivation

Section 4.3.3 shows the derivation for the upper-bound on α_f , and presents the upper-bound on α_g , which I fully derive here. I define the error between the predicted and actual rate-of-change of the error ϵ in the gradient filter as $\zeta \in \mathbb{R}^K$, and again use a discretised loss function at time step n :

$$\zeta := \hat{\epsilon} - \dot{\epsilon} = W_g z_g - \dot{\epsilon} \quad (\text{A.1})$$

$$L[n] := \frac{1}{2} \zeta^T \zeta[n] \quad (\text{A.2})$$

Using the discrete weight update law for the gradient filter and taking a first-order Taylor series of the error (ζ)

$$W_g[n+1] = W_g[n] - \alpha_g \zeta[n] z_g^T[n] \quad (\text{A.3})$$

$$\zeta[n+1] = W_g[n+1] z_g[n+1] - \dot{\epsilon}[n+1] \quad (\text{A.4})$$

Again, assuming that the input (z_g) and the error (ϵ) are slowly varying the future error can be approximated by:

$$\begin{aligned}
 \zeta[n+1] &= (W_g[n] - \alpha_g \zeta z_g^T[n]) z_g[n] - \dot{\epsilon}[n] \\
 &= W_g z_g - \alpha_g \zeta \|z_g\|^2 - \dot{\epsilon} \\
 &= \zeta - \alpha_g \|z_g\|^2 \zeta
 \end{aligned} \tag{A.5}$$

Then, the change in the loss function in each step is:

$$\begin{aligned}
 \Delta L &= \frac{1}{2} \zeta[n+1]^T \zeta[n+1] - \frac{1}{2} \zeta^T \zeta \\
 &= \frac{1}{2} (\zeta - \alpha_g \|z_g\|_2^2 \zeta)^T (\zeta - \alpha_g \|z_g\|_2^2 \zeta) - \frac{1}{2} \zeta^T \zeta \\
 &= \frac{1}{2} (1 - \alpha_g \|z_g\|_2^2)^2 \zeta^T \zeta - \frac{1}{2} \zeta^T \zeta \\
 &= \frac{1}{2} \left[(1 - \alpha_g \|z_g\|_2^2)^2 - 1 \right] \zeta^T \zeta \\
 &= \frac{1}{2} [\alpha_g^2 \|z_g\|_2^4 - 2\alpha_g \|z_g\|_2^2] \zeta^T \zeta
 \end{aligned} \tag{A.6}$$

Imposing the inequality that the loss function must be reducing ($\Delta L \leq 0$), and assuming $\alpha_g > 0$ and $z_g \neq 0$:

$$\begin{aligned}
 \alpha_g^2 \|z_g\|_2^4 - 2\alpha_g \|z_g\|_2^2 &\leq 0 \\
 \alpha_g \|z_g\|_2^2 - 2 &\leq 0 \\
 \alpha_g &\leq \frac{2}{\|z_g\|_2^2}
 \end{aligned} \tag{A.7}$$

A.2 Resulting loss change without assuming constant input

Here, I repeat the derivation of the learning rate bound on α_f without the assumption that the input vector z_f is slowly varying. As before, I denote $\hat{J} = \frac{d\epsilon}{du_{ff}}$ and $J = \frac{d\epsilon}{du_{ff}}$ for notational

simplicity. The weight update and loss change at time step n are the same:

$$\Delta W_f[n] = -\alpha_f \hat{J}^T[n] \epsilon[n] z_f[n]^T \quad (\text{A.8})$$

$$\Delta L[n] = \frac{1}{2} \epsilon[n+1]^T \epsilon[n+1] - \frac{1}{2} \epsilon[n]^T \epsilon[n] \quad (\text{A.9})$$

Using a first-order Taylor expansion of the error:

$$\epsilon[n+1] \approx \epsilon[n] + J \Delta \mathbf{u}_{ff}[n] \quad (\text{A.10})$$

Now, rather than assuming that $\dot{z}_f \approx 0$, the full derivative of $\mathbf{u}_{ff} = W_f z_f$ is used:

$$\Delta \mathbf{u}_{ff}[n] = \Delta W_f[n] z_f[n] + W_f[n] \Delta z_f[n] \quad (\text{A.11})$$

Substituting Equations A.8, A.10, and A.11 into A.9, and dropping the $[n]$ notation, gives:

$$\begin{aligned} \Delta L &= \epsilon^T J (\Delta W_f z_f + W_f \Delta z_f) + \frac{1}{2} (\Delta W_f z_f + W_f \Delta z_f)^T J^T J (\Delta W_f z_f + W_f \Delta z_f) \\ &= \underbrace{-\alpha_f \epsilon^T J \hat{J}^T \epsilon \|z_f\|_2^2 + \frac{1}{2} \alpha_f^2 \|z_f\|_2^4 \epsilon^T \hat{J} J^T J \hat{J}^T \epsilon}_{\text{Same as before in Sec. 4.3.3}} \\ &\quad + \underbrace{\epsilon^T J W_f \Delta z_f - \alpha_f \|z_f\|_2^2 \epsilon^T \hat{J} J^T J W_f \Delta z_f + \frac{1}{2} (W_f \Delta z_f)^T J^T J (W_f \Delta z_f)}_{\text{Additional terms resulting from } \dot{z} \neq 0} \end{aligned} \quad (\text{A.12})$$

This more general expression includes cross-terms involving the product of weight updates and changes in z_f , as well as a second-order term in Δz_f .

A.3 Including regularisation in the learning rate upper-bounds

Section 4.3.3 presents the upper-bounds on the learning rates for IAFF assuming no regularisation. The regularisation terms, since they are positive scalars, can be included in the numerator of each bound.

Gradient filter

Equations A.13 and A.14 give the gradient filter update equation with and without regularisation, respectively.

$$\frac{dW_g}{dt} = -\alpha_g \left(\frac{\hat{d}\epsilon}{dt} - \frac{d\epsilon}{dt} \right) \frac{z_g^T}{1 + \|z_g\|_2} \quad (\text{A.13})$$

$$\frac{dW_g}{dt} = -\alpha_g \left(\frac{\hat{d}\epsilon}{dt} - \frac{d\epsilon}{dt} \right) z_g^T \quad (\text{A.14})$$

Making the regularisation term $(1 + \|z_g\|_2)$ and therefore the resulting upper-bound limit on the learning rate α_g when using regularisation in that way:

$$\alpha_g \leq \frac{2 + 2\|z_g\|_2}{\|z_g\|_2^2} \approx \begin{cases} \frac{1}{\|z_g\|_2}, & \text{if } \|z_g\|_2 \gg 1 \\ \frac{1}{\|z_g\|_2^2}, & \text{if } \|z_g\|_2 \ll 1 \end{cases} \quad (\text{A.15})$$

Feedforward filter

The feedforward filter has a more complicated regularisation term, where Equations A.16 and A.17 show the weight update with and without regularisation, respectively.

$$\frac{dW_f^{m,l(n)}}{dt} = -\alpha_f \sum_{k=1}^K \epsilon_k \frac{\frac{\hat{d}\epsilon_k}{du_{ff}^m} z_f^{l(n)}}{1 + \left\| \frac{\hat{d}\epsilon_k}{du_{ff}^m} \right\|_2 \left\| z_f^{l(n)} \right\|_2} \quad (\text{A.16})$$

$$\frac{dW_f}{dt} = -\alpha_f \left(\epsilon^T \frac{\hat{d}\epsilon}{du_{ff}} \right)^T z_f^T \quad (\text{A.17})$$

With the same assumptions as made in Sec 4.3.3, the resulting conservative bound on the learning rate α_f becomes:

$$\alpha_f \leq \frac{2 + 2 \sup_{k,m,n} \left(\left\| \frac{\hat{d}\epsilon_k}{du_{ff}^m} \right\|_2 \left\| z_f^{l(n)} \right\|_2 \right)}{\|z_f\|_2^2 \|\hat{J}J^T\|_2} \quad (\text{A.18})$$

In practice, however, these values are variable and difficult to obtain.

A.4 Numerical stability analysis

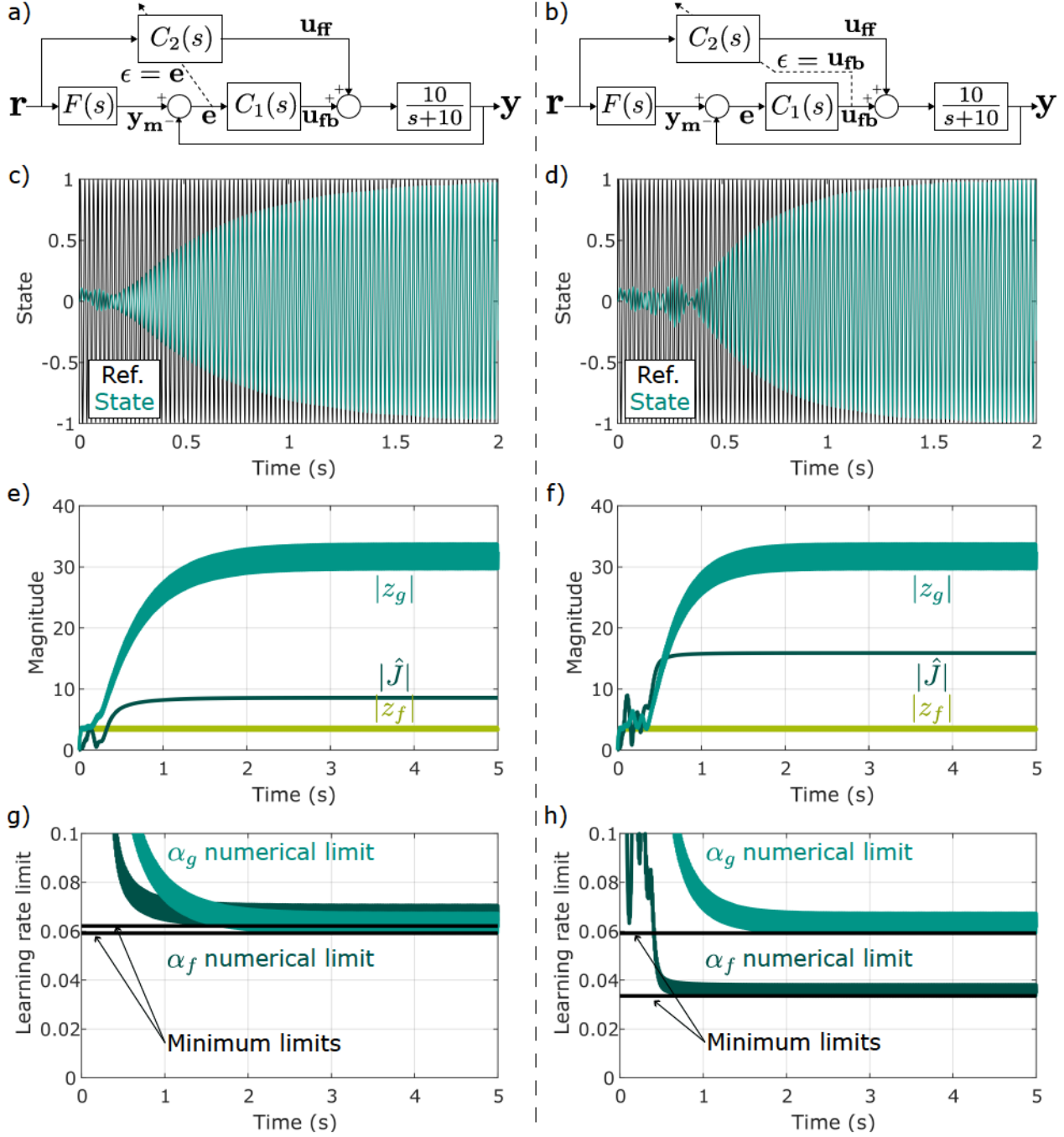


Figure A.1: Numerical estimations of the upper bounds on the learning rates. (a) and (b) show the two architectures used and the system dynamics tested in simulation. The subfigures on the left and right use $\epsilon = e$ and $\epsilon = u_{fb}$, respectively. (c) and (d) show the state reference tracking for the two architectures. (e) and (f) plot numerical calculations of the norms of the relevant parameters for determining stability bounds over time. (g) and (h) show the resulting numerical limits for the learning rates over time and their respective minimums.

The stability analysis in Section 4.3.3 revealed that a constantly reducing loss function could be guaranteed by ensuring that the two learning rates were below two thresholds, given various assumptions. However, the guaranteed limits were functions of the norms of the input vectors $\|z_g\|_2$, $\|z_f\|_2$, and the estimated and actual sensitivity derivatives $\|\hat{J}J^T\|_2$. Here, I obtain numerical estimates of the learning rate bounds using simulations of a single-input-single-output (SISO) system. Since I do not have access to J , $\hat{J}$ is used instead in the calculations. One simulation setup (Fig. A.1a) uses the feedback error as the adaptation error (ϵ) and the other (Fig. A.1b) uses the feedback actuation command. The adaptive state reference tracking by IAFF is similar for both (Figs. A.1c and A.1d). For numerically calculating the stability bounds, the two input vector norms are approximately the same for the two different architectures, however, the norm of the sensitivity derivatives is different (Figs. A.1e and A.1f). This results in the same limit for α_g for both architectures, but a different limit for α_f (Figs. A.1g and A.1h).

Following the procedure outlined in Figure A.1, numerical stability limits were calculated as the minimum maximum limit over time. As suggested by the analytical limits, when the reference signal magnitude is increased the resulting filter input norms increase. Therefore, the learning rate limit decreases (Figs. A.2a and A.2b). In addition, the feedback controller $C_1(s)$ has an impact on the α_f limit. The bandwidth of the feedback controller had no consistent impact on the limit when $\epsilon = e$ (Fig. A.2c). But, when $\epsilon = u_{ff}$, increasing the bandwidth of the feedback controller, decreased the learning limit for α_f (Fig. A.2d). This is likely because increasing the feedback controller's bandwidth increases the magnitude of the sensitivity derivatives. For both architectures, there was no clear relationship between the phase margin and the learning rate limits. In addition, the gradient filter learning rate was not affected by either the phase margin or the bandwidth of the feedback controller (Figs. A.2e and A.2f).

Lastly, the learning rates used in the simulation α_f and α_g , affected the calculation of the numerical limits of the learning rate α_f but not α_g (Fig. A.3). In particular, for both architectures, as α_g increased, the numerically calculated limit on α_f decreased (Figs. A.3c and A.3d).

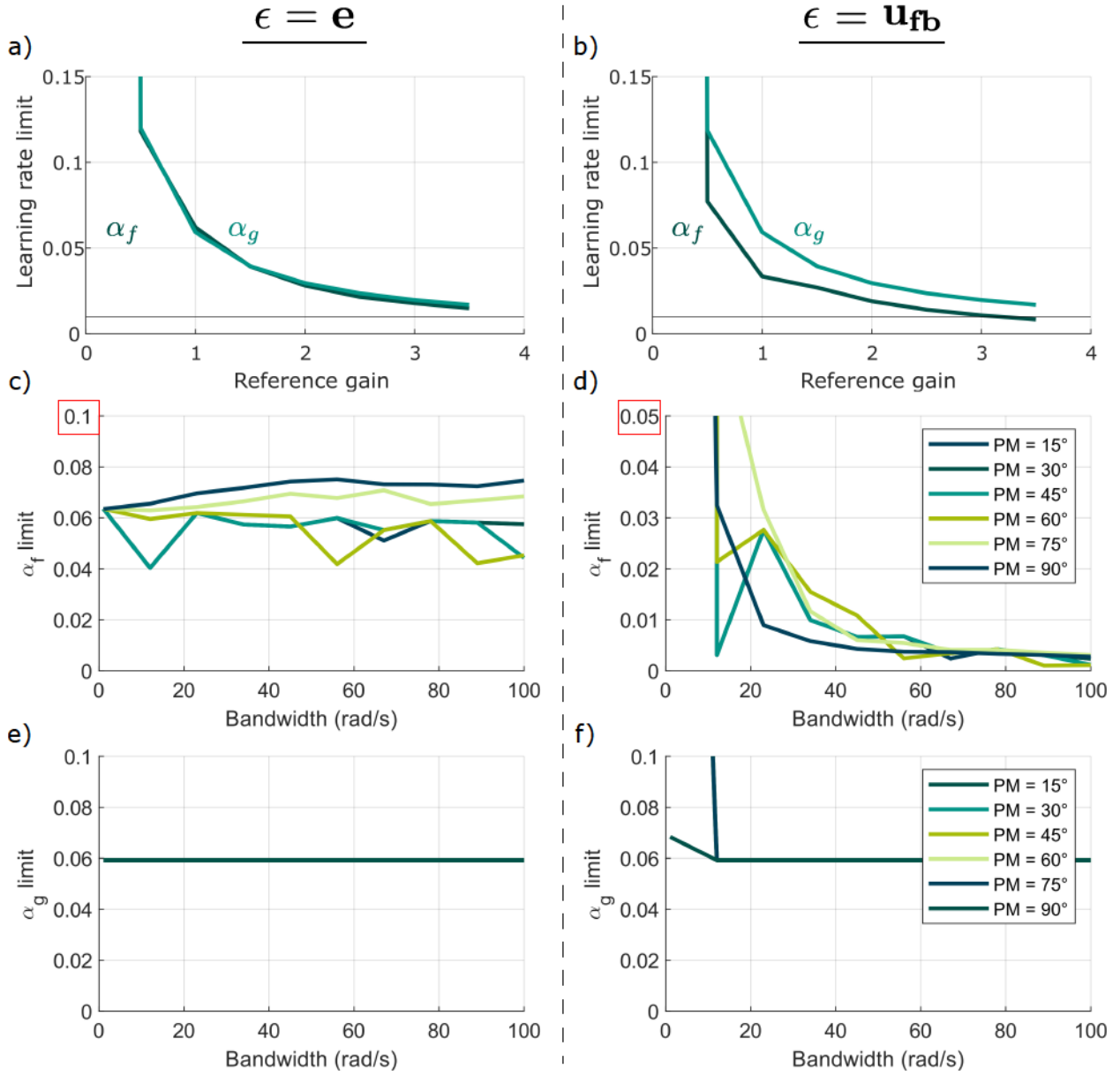


Figure A.2: Simulations demonstrating the dependencies of the numerical estimates of the learning rates. The subfigures on the left and right use $\epsilon = e$ and $\epsilon = u_{fb}$, respectively. (a) and (b) show the numerically estimated learning rate limits for different magnitudes of the reference signal. (c) and (d) show the α_f limit for different bandwidth and phase margin (PM) feedback controllers. (e) and (f) show the same for α_g .

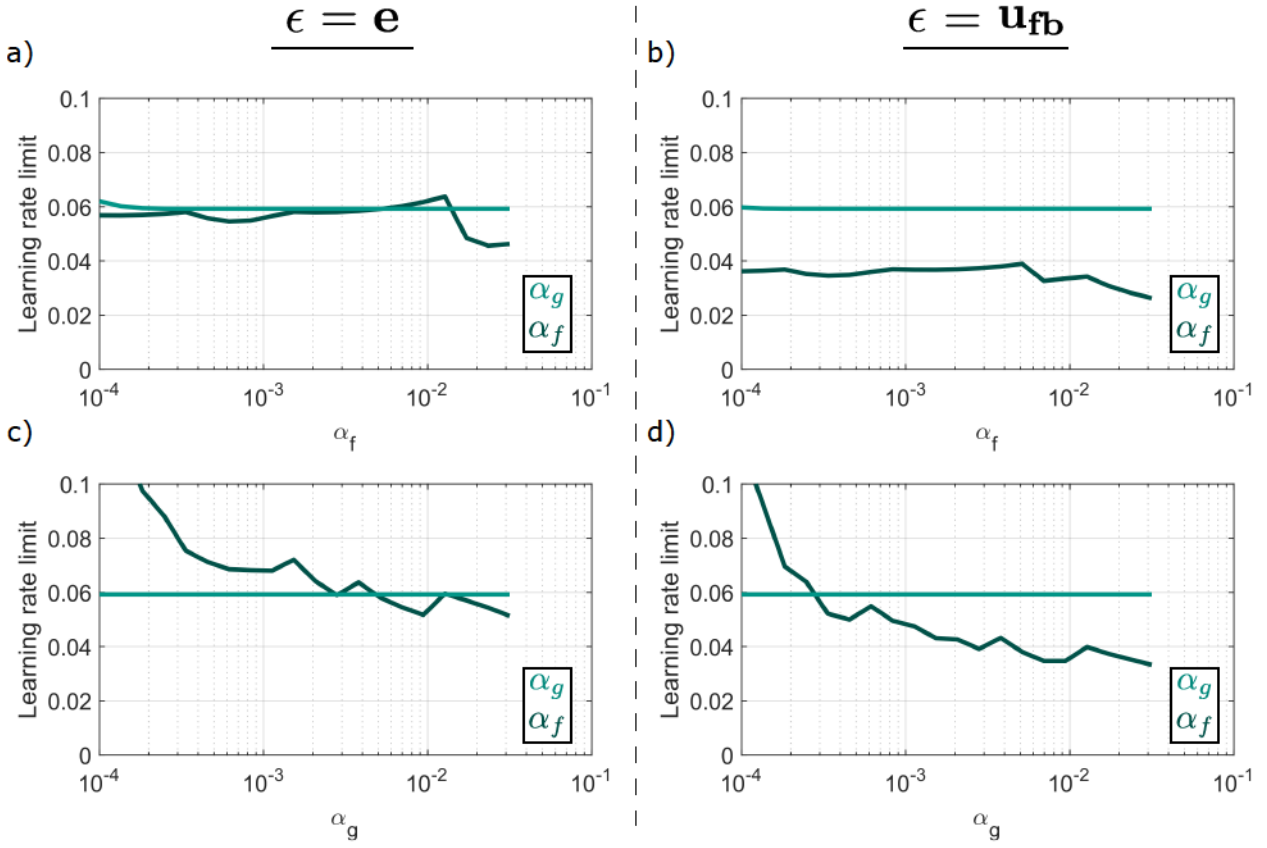


Figure A.3: Variability of the numerical estimates of the learning rates when calculated in simulations using different learning rates. (a) and (c) use $\epsilon = e$ and show variable α_f and α_g , respectively. (b) and (d) show the same for $\epsilon = u_{fb}$.

Appendix B

Analysis of temperature as a confounding factor in behavioural experiments in Ch. 3

The recorded temperatures inside the experimental flight arena in Chapter 3 were slightly different across different conditions in our experiments (Fig. B.1a). It was not clear whether temperature could explain the trends in our data from plotting the speed and temperature across the different conditions (Fig. B.1b). So, to account for the potential confounding effect of ambient temperature on flight speed, I conducted an analysis of covariance (ANCOVA) using a linear model. The aim is to compare intersaccadic average flight speeds across the five experimental conditions while adjusting for the influence of temperature. Two different models were used:

$$\text{Speed}_{i,j} = \mu + \beta_i + \beta_T \cdot T_{i,j} + \varepsilon_{i,j} \quad (\text{B.1})$$

$$\text{Speed}_{i,j} = \mu + \beta_i + \beta_{T,i} \cdot T_{i,j} + \varepsilon_{i,j} \quad (\text{B.2})$$

In these models, i, j represent indicators for each of the five conditions and each observation within that condition, respectively. μ is the predicted speed in the *reference condition* (I-D) at $T = 0^\circ\text{C}$. β_i captures the difference in the mean speed between each condition and the reference condition, at $T = 0^\circ\text{C}$. $T_{i,j}$ represents the measured ambient temperature in degrees Celsius for each intersaccadic flight segment. ε denotes the residual error term, assumed to be

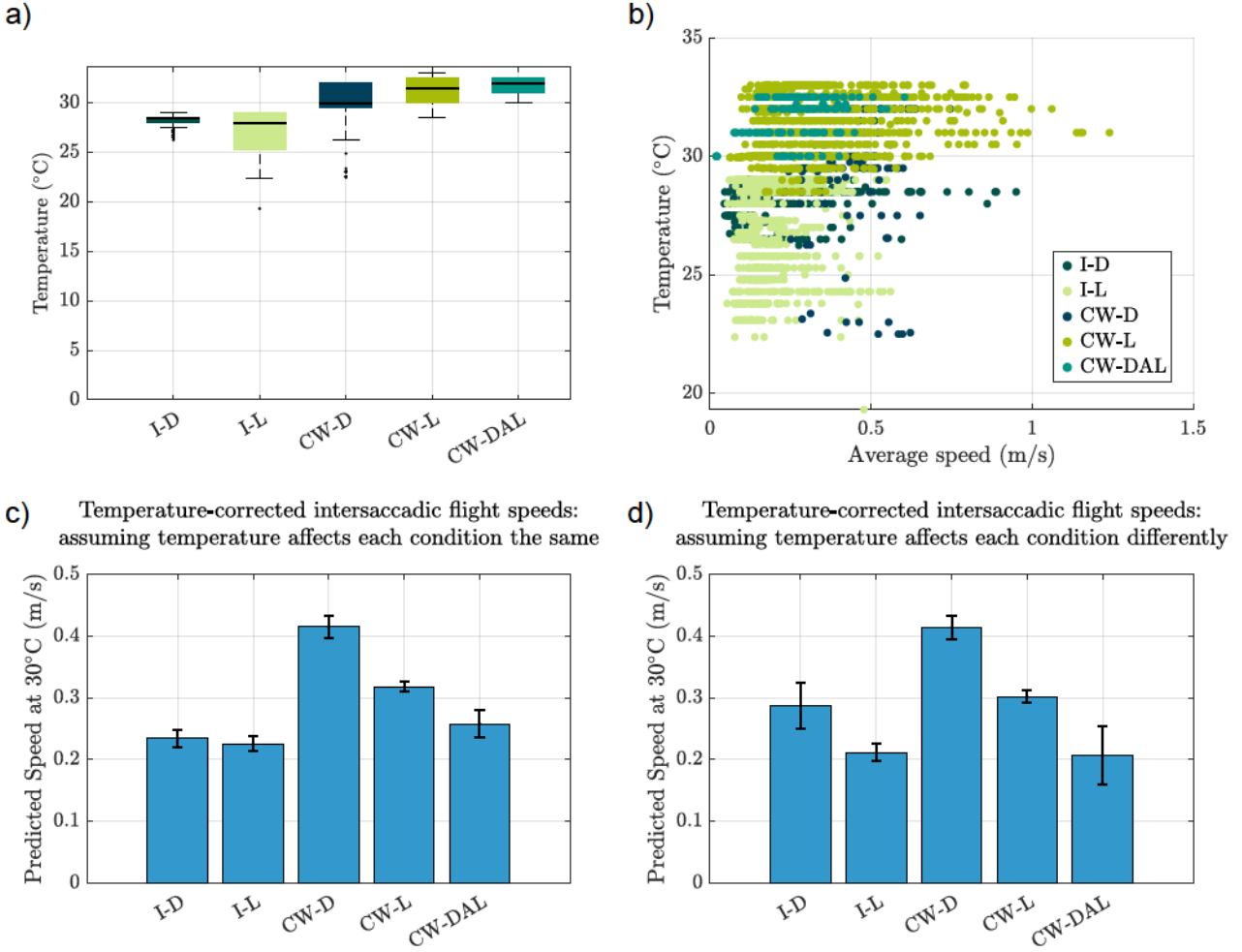


Figure B.1: (a) Temperatures across the different observations for each of the conditions in Experiment 1 in Chapter 3. (b) Average intersaccadic speeds at different temperatures for each condition. (c, d) Temperature-corrected average intersaccadic speed. Where the average speed is predicted at 30 degrees Celsius, and the error bars show the 95% confidence intervals. (c) Assumes temperature has the same linear effect on all conditions. (d) Assumes each condition is affected differently by the temperature.

normally distributed with zero mean and constant variance, $\varepsilon \sim \mathcal{N}(0, \sigma^2)$. In the first model (Eq. B.1), the coefficient β_T represents the common slope across all conditions, quantifying the linear effect of temperature on speed. Then in the second model (Eq. B.2), $\beta_{T,i}$ quantifies a condition-specific temperature slope.

The first model applies if the effect of temperature on flight speed is assumed to be the same across all conditions. The second model allows temperature to affect each condition differently. In both cases, a similar trend for CW-D, CW-L, and CW-DAL is observed (Figs. B.1c and B.1d). Thereby confirming our observed trend is not a result of temperature variations. However, for

intact animals there is a difference between the two plots. Indeed, according to the fitted second model there is a difference in how temperature affects flight speed in the dark and the light. The fitted gradients are 0.0368 and 0.0046 for $\beta_{T,(I-D)}$ and $\beta_{T,(I-L)}$, respectively.

Appendix C

Supporting data tables

C.1 SISO controller comparison data (Ch. 2)

C.1.1 Controller tuning

Table C.1: Controller parameters used in controller comparison in Ch. 2. The Smith predictor had the same gains as the PFB when $\tau = 0$ for all values of τ .

P struct.	ω (rad/s)	τ (s)	PFB: K_p	PFB: K_i	PFB: K_d	FSDoF: q
1	10	0	1.7212	19.5871	0	37.6369
1	10	0.1	0.86007	5.6566	0	28.9758
1	10	0.05	1.2761	11.2087	0	40.9076
1	50	0	1.7212	97.9357	0	187.825
1	50	0.1	0.40866	7.3928	0	100.4709
1	50	0.05	0.42975	13.1689	0	100.5642
2	10	0	21.2534	6.4618	1.6781	43.0316
2	10	0.1	5.0023	0.42403	0.45127	22.1563
2	10	0.05	9.2896	1.5187	0.99159	32.8344
2	50	0	106.2671	161.5462	1.6781	126.7761
2	50	0.1	7.3683	0.97547	0.3081	44.9396
2	50	0.05	12.8405	2.9942	0.41317	54.7867
3	10	0	1.9691	10.5361	0.014843	18.6125
3	10	0.1	1.4511	6.1648	0.085395	31.5524
3	10	0.05	2.1159	9.1138	0.10817	35.8477
3	50	0	1.9691	52.6805	0.0029686	75.6006
3	50	0.1	0.43646	6.7586	0.0051959	52.8529
3	50	0.05	0.51857	11.5895	0.0051033	53.141
4	10	0	0.16939	56.601	0	63.7718
4	10	0.1	0.8655	8.7584	0	76.3081
4	10	0.05	1.2446	22.0426	0	263.9988
4	50	0	0	0.99964	0	20
4	50	0.1	0	0.99964	0	20
4	50	0.05	0	0.99964	0	20
5	10	0	-7.1377	-40.482	0	206.5766
5	10	0.05	-1.5703	-2.2557	-0.072806	337.1645
5	50	0	-7.1377	-202.4101	0	1031.6281
6	10	0	0	1.0905	0	9
6	10	0.1	0	1.0005	0	9
6	10	0.05	0	0.99738	0	9
6	50	0	0	1.3688	0	49
6	50	0.1	0	1.0093	0	49
6	50	0.05	0	1.3153	0	49

C.1.2 Results

Table C.2: Pure feedback controller (PFB) table of results simulations in Ch. 2

P struct.	ω (rad/s)	τ (s)	Max actuation	Total actuation	Settle time (s)	Overshoot (%)
1	10	0	1.7212	0.22457	0.1784	0.61067
1	10	0.1	1.4257	0.58452	0.56067	8.25
1	10	0.05	1.8365	0.38207	0.32206	8.7177
1	50	0	1.7212	0.044918	0.035685	0.61067
1	50	0.1	1.1479	0.38212	0.39842	13.2661
1	50	0.05	1.1483	0.22819	0.23443	10.716
2	10	0	199.2938	2.529	0.3046	2.3031
2	10	0.1	50.9481	1.3599	0.63634	8.8128
2	10	0.05	114.9587	2.242	0.25202	8.3084
2	50	0	333.6711	3.1107	0.065618	15.9754
2	50	0.1	40.5438	1.5825	0.69682	21.328
2	50	0.05	59.7381	1.8048	0.38318	21.039
3	10	0	3.4534	0.65981	0.55488	4.902
3	10	0.1	9.9907	0.8442	0.70429	18.0198
3	10	0.05	12.9328	0.53905	0.39502	9.2305
3	50	0	2.2945	0.12968	0.10865	5.2519
3	50	0.1	1.1253	0.38351	0.39139	9.4171
3	50	0.05	1.1381	0.1905	0.18627	7.6357
4	10	0	1.6443	0.27218	0.21915	12.8313
4	10	0.1	1.7413	0.41552	0.36451	16.7991
4	10	0.05	2.3466	0.32315	0.25	26.516
4	50	0	0.99995	2.9595	3.9527	-0.0051387
4	50	0.1	0.99998	2.6544	3.5496	-0.0016679
4	50	0.05	0.99997	2.8112	3.7553	-0.0030613
5	10	0	8.4098	0.47791	0.39104	19.1577
5	10	0.05	13.2699	2.7653	2.0451	157.6398
5	50	0	8.4098	0.095586	0.078212	19.1577
6	10	0	1	1.386	1.8682	0.0014718
6	10	0.1	1.0269	1.9828	2.4505	3.0725
6	10	0.05	1.0018	1.3164	1.8371	0.20327
6	50	0	1	1.2987	1.6725	-4.0536e-10
6	50	0.1	1	1.4947	2.028	-4.3091e-09
6	50	0.05	1	0.9747	1.3301	-3.6593e-10

Table C.3: Smith predictor (SP) table of results simulations in Ch. 2

P struct.	ω (rad/s)	τ (s)	Max actuation	Total actuation	Settle time (s)	Overshoot (%)
1	10	0	1.7212	0.22457	0.1784	0.61067
1	10	0.1	1.7212	0.22457	0.1784	0.61067
1	10	0.05	1.7212	0.22457	0.1784	0.61067
1	50	0	1.7212	0.044918	0.035685	0.61067
1	50	0.1	1.7212	0.044918	0.035685	0.61067
1	50	0.05	1.7212	0.044918	0.035685	0.61067
2	10	0	199.2938	2.529	0.3046	2.3031
2	10	0.1	199.2938	2.529	0.3046	2.3031
2	10	0.05	199.2938	2.529	0.3046	2.3031
2	50	0	333.6711	3.1107	0.065619	15.9754
2	50	0.1	333.6711	3.1107	0.065619	15.9754
2	50	0.05	333.6711	3.1107	0.065619	15.9754
3	10	0	3.4534	0.65981	0.55488	4.902
3	10	0.1	3.4534	0.65981	0.55488	4.902
3	10	0.05	3.4534	0.65981	0.55488	4.902
3	50	0	2.2945	0.12968	0.10865	5.2519
3	50	0.1	2.2945	0.12968	0.10865	5.2519
3	50	0.05	2.2945	0.12968	0.10865	5.2519
4	10	0	1.6443	0.27218	0.21915	12.8313
4	10	0.1	1.6443	0.27218	0.21915	12.8313
4	10	0.05	1.6443	0.27218	0.21915	12.8313
4	50	0	0.99995	2.9595	3.9527	-0.0051387
4	50	0.1	0.99995	2.9595	3.9527	-0.0051387
4	50	0.05	0.99995	2.9595	3.9527	-0.0051387
5	10	0	8.4098	0.47791	0.39104	19.1577
5	10	0.05	8.4098	0.47791	0.39104	19.1577
5	50	0	8.4098	0.095586	0.078212	19.1577
6	10	0	1	1.386	1.8682	0.0014718
6	10	0.1	1	1.386	1.8682	0.0014718
6	10	0.05	1	1.386	1.8682	0.0014718
6	50	0	1	1.2987	1.6725	-4.0536e-10
6	50	0.1	1	1.2987	1.6725	-4.0536e-10
6	50	0.05	1	1.2987	1.6725	-4.0536e-10

Table C.4: Fully-seperable-degrees-of-freedom (FSDoF) controller table of results simulations in Ch. 2

P struct.	ω (rad/s)	τ (s)	Max actuation	Total actuation	Settle time (s)	Overshoot (%)
1	10	0	1.708	0.22461	0.15501	-1.5787e-11
1	10	0.1	1.4121	0.59165	0.20134	-2.2826e-11
1	10	0.05	1.8227	0.37316	0.14262	-1.5743e-11
1	50	0	1.7055	0.044904	0.031065	-3.9191e-12
1	50	0.1	1.1379	0.39851	0.05807	-7.816e-12
1	50	0.05	1.1384	0.23454	0.058016	-7.7827e-12
2	10	0	199.3839	2.7954	0.13558	-1.6875e-11
2	10	0.1	50.6912	1.3929	0.26331	-2.7456e-11
2	10	0.05	114.3585	2.0797	0.17768	-1.3689e-11
2	50	0	335.1802	1.5862	0.046018	-5.9841e-12
2	50	0.1	40.3913	1	0.12982	-1.7808e-11
2	50	0.05	60.0318	1	0.10649	-1.2257e-11
3	10	0	3.4643	0.64743	0.31345	-3.4639e-11
3	10	0.1	9.9555	0.8409	0.1849	-2.2804e-11
3	10	0.05	12.8506	0.53923	0.16275	-1.7275e-11
3	50	0	2.2862	0.1222	0.077169	-7.1942e-12
3	50	0.1	1.1174	0.39355	0.11039	-1.0014e-11
3	50	0.05	1.1296	0.18863	0.10979	-9.9698e-12
4	10	0	1.6288	0.25412	0.091489	2.2427e-12
4	10	0.1	1.7247	0.40496	0.076459	5.5067e-12
4	10	0.05	2.3247	0.30905	0.022104	2.065e-12
4	50	0	1	3.8393	0.2917	-1.3323e-12
4	50	0.1	1	3.4363	0.2917	-1.3323e-12
4	50	0.05	1	3.642	0.2917	-1.3323e-12
5	10	0	8.3439	0.44429	0.028242	-2.8422e-12
5	10	0.05	13.1447	2.1131	0.017309	-2.3537e-12
5	50	0	8.3347	0.088853	0.0056623	-1.2212e-13
6	10	0	1	1.746	0.81366	-1.5312e-10
6	10	0.1	1	2.3283	0.81366	-1.5312e-10
6	10	0.05	1	1.7149	0.81366	-1.5312e-10
6	50	0	1	1.6516	0.18295	-1.7124e-10
6	50	0.1	1	2.0072	0.18295	-1.7124e-10
6	50	0.05	1	1.3093	0.18295	-1.7124e-10

C.2 Flight kinematics average data (Ch. 3)

Table C.5: Mean and standard deviation of flight kinematics for intact animals in [Experiment 1](#) (Ch. 3)

Parameter	I-L	I-D
Curvature (m^{-1})	3.4121(± 1.3430)	3.4719(± 1.2921)
Cent acc (m/s^2)	0.7292(± 0.5919)	0.6988(± 0.4359)
Speed (m/s)	0.2160(± 0.1384)	0.1976(± 0.0881)
Elevation ang ($^\circ$)	1.2948(± 14.1018)	0.0118(± 15.6421)

Table C.6: Mean and standard deviation of flight kinematics for wing-damaged animals in [Experiment 1](#) (Ch. 3)

Parameter	CW-D	CW-L	CW-DAL
Curvature (m^{-1})	4.3074(± 1.8582)	2.8605(± 1.3088)	3.9804(± 1.6197)
Cent acc (m/s^2)	1.8307(± 0.9752)	0.9340(± 0.6232)	1.0743(± 0.5662)
Speed (m/s)	0.4145(± 0.1072)	0.3309(± 0.1554)	0.2737(± 0.1042)
Elevation ang ($^\circ$)	8.3062(± 17.9614)	0.2151(± 13.3420)	5.3497(± 15.7842)

Table C.7: Mean and standard deviation of kinematic data in wing-damage and ablated antennae animals in [Experiment 2](#) (Ch. 3)

Parameter	AA-CW-D	AA-CW-L	AA-CW-DAL
Curvature (m^{-1})	3.7425(± 1.3811)	2.8122(± 1.3813)	4.5371(± 1.9442)
Cent acc (m/s^2)	3.2396(± 1.0699)	1.1208(± 0.7727)	2.3569(± 1.7969)
Speed (m/s)	0.8902(± 0.2221)	0.4255(± 0.2146)	0.5199(± 0.3207)
Elevation ang ($^\circ$)	7.0102(± 13.0224)	0.2489(± 14.9748)	-4.4830(± 19.0975)